



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2018년03월05일

(11) 등록번호 10-1834611

(24) 등록일자 2018년02월26일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/597 (2014.01) *HO4N 19/51* (2014.01)
- (21) 출원번호 10-2014-7028427
- (22) 출원일자(국제) 2013년03월14일
심사청구일자 2017년02월16일
- (85) 번역문제출일자 2014년10월08일
- (65) 공개번호 10-2014-0144210
- (43) 공개일자 2014년12월18일
- (86) 국제출원번호 PCT/US2013/031339
- (87) 국제공개번호 WO 2013/138593
국제공개일자 2013년09월19일
- (30) 우선권주장
13/798,997 2013년03월13일 미국(US)
(뒷면에 계속)
- (56) 선행기술조사문헌
WO2008007913 A1
Test Model under Consideration for HEVC based
3D video coding, MPEG MEETING, ISO/IEC
JTC1/SC29/WG11, no. N12559 (2012.02.11.)

- (73) 특허권자
웰컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
장 리
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- 천 잉
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- 카르체비츠 마르타
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- (74) 대리인
특허법인코리아나

전체 청구항 수 : 총 59 항

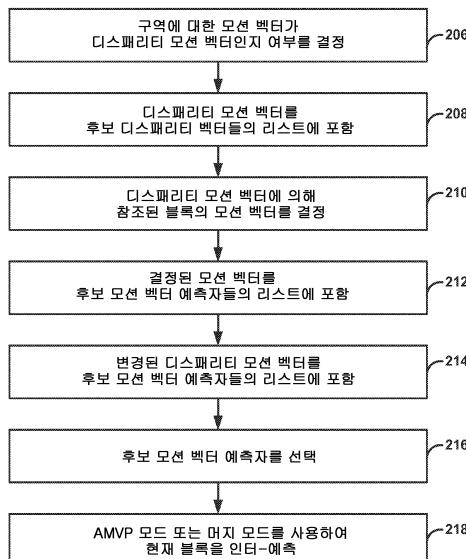
심사관 : 장석환

(54) 발명의 명칭 비디오 코딩에서 디스패리티 벡터 예측

(57) 요 약

예측될 현재 블록에 더 공간적 및 시간적으로 이웃하는 구역들 중 하나의 디스패리티 모션 벡터에 기초하여 현재 블록에 대한 디스패리티 벡터를 결정하는 기법들이 설명된다. 공간적 및 시간적으로 이웃하는 구역들은 하나 또는 복수의 블록들을 포함하고, 디스패리티 모션 벡터는 공간적 또는 시간적으로 이웃하는 구역 내의 복수의 블록들에 대한 하나의 레퍼런스 픽처 리스트에서의 단일 벡터를 나타낸다. 결정된 디스패리티 벡터는 상이한 뷰들, 예컨대 머지 모드, 어드밴스드 모션 벡터 예측 (AMVP) 모드, 인터-뷰 모션 예측, 및 인터-뷰 잔여 예측 간의 정보를 이용하는 코딩 도구들에 사용될 수 있다.

대 표 도 - 도16



(30) 우선권주장

61/610,961 2012년03월14일 미국(US)
61/621,929 2012년04월09일 미국(US)
61/623,041 2012년04월11일 미국(US)
61/658,754 2012년06월12일 미국(US)
61/682,221 2012년08월11일 미국(US)
61/682,224 2012년08월11일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 단계로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하는 단계;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하는 단계; 및

상기 현재 블록을 인터-예측 디코딩하는 단계로서, 상기 디코딩은 상기 구역에 대한 상기 디스패리티 모션 벡터를 상기 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 단계는, 상기 현재 블록을 공간적으로 이웃하는 공간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하는 단계를 포함하고,

상기 픽처는 상기 공간적 이웃 구역을 포함하고 상기 현재 블록을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 공간적 이웃 구역은 2 개의 4x4 블록들을 포함하는 8x4 구역을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 단계는, 상기 현재 블록을 시간적으로 이웃하는 시간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하는 단계를 포함하고,

상기 픽처는 상기 시간적 이웃 구역을 포함하고 상기 현재 블록을 포함하지 않는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 4 항에 있어서,

상기 시간적 이웃 구역은 16 개의 4x4 블록들을 포함하는 16x16 구역을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 현재 블록은 제 1 블록을 포함하고, 상기 구역은 제 1 구역을 포함하고, 상기 디스패리티 벡터는 제 1 디스패리티 벡터를 포함하고, 상기 방법은:

제 2 구역에 대한 디스패리티 벡터가 상기 제 2 구역을 이웃하는 블록의 디스패리티 모션 벡터로부터 도출되는 것에 기초하여 제 2 블록을 이웃하는 상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하는 단계;

상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하는 것에 응답하여, 상기 제 2 구역을 이웃하는 블록의 상기 디스패리티 모션 벡터가 상기 제 2 블록에 대한 제 2 디스패리티 벡터라고 결정하는 단계; 및

상기 제 2 디스패리티 벡터를 이용하여 상기 제 2 블록을 인터-예측 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 디스패리티 벡터가 지칭하는 블록을 결정하는 단계;

상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 모션 벡터를 결정하는 단계; 및

상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 상기 모션 벡터를 후보 모션 벡터 예측자들의 리스트에 포함시키는 단계를 더 포함하고,

상기 현재 블록을 인터-예측 디코딩하는 단계는 상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 상기 모션 벡터가 머지 모드 또는 어드밴스드 모션 벡터 예측 (advanced motion vector prediction; AMVP) 모드의 일부로서 후보 모션 벡터 예측자들의 리스트로부터 선택되는 것에 응답하여, 상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 상기 모션 벡터에 기초하여 상기 현재 블록을 인터-예측 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 디스패리티 벡터가 지칭하는 블록을 결정하는 단계를 더 포함하고,

상기 현재 블록을 인터-예측 디코딩하는 단계는 결정된 상기 블록의 잔여 정보를 사용하여 상기 현재 블록의 잔여 정보를 인터-예측하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 디스패리티 모션 벡터를 변경하는 단계; 및

변경된 상기 디스패리티 모션 벡터를 상기 현재 블록에 대한 후보 모션 벡터 예측자들의 리스트에 포함시키는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 10

제 9 항에 있어서,

상기 디스패리티 모션 벡터를 변경하는 단계는 상기 디스패리티 모션 벡터의 y-컴포넌트를 0 과 동일하게 설정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 11

제 1 항에 있어서,

상기 현재 블록은 순간 디코더 리프레시 (instantaneous decoder refresh; IDR) 또는 클린 랜덤 액세스 (clean random access; CRA) 꽉쳐의 블록을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 12

비디오 데이터를 인코딩하는 방법으로서,

상기 비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 단계로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하는 단계;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하는 단계; 및

상기 현재 블록을 인터-예측 인코딩하는 단계로서, 상기 인코딩은 상기 구역에 대한 디스패리티 모션 벡터를 상기 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 인코딩하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 13

제 12 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 단계는, 상기 현재 블록을 공간적으로 이웃하는 공간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 상기 픽처를 지칭한다고 결정하는 단계를 포함하고,

상기 픽처는 상기 공간적 이웃 구역을 포함하고 상기 현재 블록을 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 14

제 13 항에 있어서,

상기 공간적 이웃 구역은 2 개의 4x4 블록들을 포함하는 8x4 구역을 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 15

제 12 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 단계는, 상기 현재 블록을 시간적으로 이웃하는 시간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 상기 픽처를 지칭한다고 결정하는 단계를 포함하고,

상기 픽처는 상기 시간적 이웃 구역을 포함하고 상기 현재 블록을 포함하지 않는, 비디오 데이터를 인코딩하는 방법.

청구항 16

제 15 항에 있어서,

상기 시간적 이웃 구역은 16 개의 4x4 블록들을 포함하는 16x16 구역을 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 17

제 12 항에 있어서,

상기 현재 블록은 제 1 블록을 포함하고, 상기 구역은 제 1 구역을 포함하고, 상기 디스패리티 벡터는 제 1 디스패리티 벡터를 포함하고, 상기 방법은:

제 2 구역에 대한 디스패리티 벡터가 상기 제 2 구역을 이웃하는 블록의 디스패리티 모션 벡터로부터 도출되는 것에 기초하여 제 2 블록을 이웃하는 상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하는 단

계;

상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하는 것에 응답하여, 상기 제 2 구역을 이웃하는 블록의 상기 디스패리티 모션 벡터가 상기 제 2 블록에 대한 제 2 디스패리티 벡터라고 결정하는 단계; 및

상기 제 2 디스패리티 벡터를 이용하여 상기 제 2 블록을 인터-예측 인코딩하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 18

제 12 항에 있어서,

상기 디스패리티 벡터가 지칭하는 블록을 결정하는 단계를 더 포함하고,

상기 현재 블록을 인터-예측 인코딩하는 단계는 결정된 상기 블록의 잔여 정보를 사용하여 상기 현재 블록의 잔여 정보를 인터-예측하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 19

제 12 항에 있어서,

상기 디스패리티 모션 벡터를 변경하는 단계; 및

변경된 상기 디스패리티 모션 벡터를 상기 현재 블록에 대한 후보 모션 벡터 예측자들의 리스트에 포함시키는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 20

제 19 항에 있어서,

상기 디스패리티 모션 벡터를 변경하는 단계는 상기 디스패리티 모션 벡터의 y-컴포넌트를 0 과 동일하게 설정하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 21

제 12 항에 있어서,

상기 현재 블록은 순간 디코더 리프레시 (IDR) 또는 클린 랜덤 액세스 (CRA) 픽처의 블록인, 비디오 데이터를 인코딩하는 방법.

청구항 22

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하도록 구성되는 메모리 유닛; 및

비디오 디코더를 포함하고,

상기 비디오 디코더는:

비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것으로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하고;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하고; 그리고

상기 현재 블록을 인터-예측 디코딩하도록 구성되며,

상기 현재 블록을 디코딩하기 위해, 상기 비디오 디코더는 상기 구역에 대한 상기 디스패리티 모션 벡터를 상기 현재 블록에 대한 디스패리티 벡터로서 사용하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 23

제 22 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위해, 상기 비디오 디코더는, 상기 현재 블록을 공간적으로 이웃하는 공간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 상기 픽처를 지칭한다고 결정하도록 구성되고,

상기 픽처는 상기 공간적 이웃 구역을 포함하고 상기 현재 블록을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 24

제 23 항에 있어서,

상기 공간적 이웃 구역은 2 개의 4x4 블록들을 포함하는 8x4 구역을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 25

제 22 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위해, 상기 비디오 디코더는, 상기 현재 블록을 시간적으로 이웃하는 시간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 상기 픽처를 지칭한다고 결정하도록 구성되고,

상기 픽처는 상기 시간적 이웃 구역을 포함하고 상기 현재 블록을 포함하지 않는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 26

제 25 항에 있어서,

상기 시간적 이웃 구역은 16 개의 4x4 블록들을 포함하는 16x16 구역을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 27

제 22 항에 있어서,

상기 현재 블록은 제 1 블록을 포함하고, 상기 구역은 제 1 구역을 포함하고, 상기 디스패리티 벡터는 제 1 디스패리티 벡터를 포함하고, 상기 비디오 디코더는:

제 2 구역에 대한 디스패리티 벡터가 상기 제 2 구역을 이웃하는 블록의 디스패리티 모션 벡터로부터 도출되는 것에 기초하여 제 2 블록을 이웃하는 상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하고;

상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하는 것에 응답하여, 상기 제 2 구역을 이웃하는 블록의 상기 디스패리티 모션 벡터가 상기 제 2 블록에 대한 제 2 디스패리티 벡터라고 결정하고; 그리고

상기 제 2 디스패리티 벡터를 이용하여 상기 제 2 블록을 인터-예측 디코딩하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 28

제 22 항에 있어서,

상기 비디오 디코더는,

상기 디스패리티 벡터가 지칭하는 블록을 결정하고;

상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 모션 벡터를 결정하며;

상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 상기 모션 벡터를 후보 모션 벡터 예측자들의 리스트에 포

함시키도록 구성되고,

상기 현재 블록을 인터-예측 디코딩하기 위해, 상기 비디오 디코더는, 상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 상기 모션 벡터가 머지 모드 또는 어드밴스드 모션 벡터 예측 (AMVP) 모드의 일부로서 후보 모션 벡터 예측자들의 리스트로부터 선택되는 것에 응답하여, 상기 디스패리티 벡터가 지칭하는 상기 블록에 대한 상기 모션 벡터에 기초하여 상기 현재 블록을 인터-예측 디코딩하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 29

제 22 항에 있어서,

상기 비디오 디코더는,

상기 디스패리티 벡터가 지칭하는 블록을 결정하도록 구성되고,

상기 현재 블록을 인터-예측 디코딩하기 위해, 상기 비디오 디코더는, 결정된 상기 블록의 잔여 정보를 사용하여 상기 현재 블록의 잔여 정보를 인터-예측하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 30

제 22 항에 있어서,

상기 비디오 디코더는,

상기 디스패리티 모션 벡터를 변경하며;

변경된 상기 디스패리티 모션 벡터를 상기 현재 블록에 대한 후보 모션 벡터 예측자들의 리스트에 포함하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 31

제 30 항에 있어서,

상기 디스패리티 모션 벡터를 변경하기 위해, 상기 비디오 디코더는, 상기 디스패리티 모션 벡터의 y-컴포넌트를 0 과 동일하게 설정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 32

제 22 항에 있어서,

상기 현재 블록은 순간 디코더 리프레시 (IDR) 또는 클린 랜덤 액세스 (CRA) 픽처의 블록인, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 33

제 22 항에 있어서,

상기 디바이스는,

무선 통신 디바이스;

마이크로프로세서; 또는

집적 회로 중 하나를 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 34

비디오 데이터를 인코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하도록 구성되는 메모리 유닛; 및

비디오 인코더를 포함하고,

상기 비디오 인코더는:

비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것으로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하고;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하고; 그리고

상기 현재 블록을 인터-예측 인코딩하도록 구성되며,

상기 현재 블록을 인코딩하기 위해, 상기 비디오 인코더는, 상기 구역에 대한 디스패리티 모션 벡터를 상기 현재 블록에 대한 디스패리티 벡터로서 사용하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 35

제 34 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위해, 상기 비디오 인코더는, 상기 현재 블록을 공간적으로 이웃하는 공간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하도록 구성되고,

상기 픽처는 상기 공간적 이웃 구역을 포함하고 상기 현재 블록을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 36

제 35 항에 있어서,

상기 공간적 이웃 구역은 2 개의 4x4 블록들을 포함하는 8x4 구역을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 37

제 34 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위해, 상기 비디오 인코더는, 상기 현재 블록을 시간적으로 이웃하는 시간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하도록 구성되고,

상기 픽처는 상기 시간적 이웃 구역을 포함하고 상기 현재 블록을 포함하지 않는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 38

제 37 항에 있어서,

상기 시간적 이웃 구역은 16 개의 4x4 블록들을 포함하는 16x16 구역을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 39

제 34 항에 있어서,

상기 현재 블록은 제 1 블록을 포함하고, 상기 구역은 제 1 구역을 포함하고, 상기 디스패리티 벡터는 제 1 디스패리티 벡터를 포함하고, 상기 비디오 인코더는:

제 2 구역에 대한 디스패리티 벡터가 상기 제 2 구역을 이웃하는 블록의 디스패리티 모션 벡터로부터 도출되는 것에 기초하여 제 2 블록을 이웃하는 상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하고;

상기 제 2 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하는 것에 응답하여, 상기 제 2 구역을 이웃하

는 블록의 상기 디스패리티 모션 벡터가 상기 제 2 블록에 대한 제 2 디스패리티 벡터라고 결정하고; 그리고 상기 제 2 디스패리티 벡터를 이용하여 상기 제 2 블록을 인터-예측 인코딩하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 40

제 34 항에 있어서,

상기 비디오 인코더는,

상기 디스패리티 벡터가 지칭하는 블록을 결정하도록 구성되고,

상기 현재 블록을 인터-예측 인코딩하기 위해, 상기 비디오 인코더는, 결정된 상기 블록의 잔여 정보를 사용하여 상기 현재 블록의 잔여 정보를 인터-예측하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 41

제 34 항에 있어서,

상기 비디오 인코더는,

상기 디스패리티 모션 벡터를 변경하며;

변경된 상기 디스패리티 모션 벡터를 상기 현재 블록에 대한 후보 모션 벡터 예측자들의 리스트에 포함하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 42

제 41 항에 있어서,

상기 디스패리티 모션 벡터를 변경하기 위해, 상기 비디오 인코더는, 상기 디스패리티 모션 벡터의 y-컴포넌트를 0 과 동일하게 설정하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 43

제 34 항에 있어서,

상기 현재 블록은 순간 인코더 리프레시 (IDR) 또는 클린 랜덤 액세스 (CRA) 픽처의 블록인, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 44

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위한 수단으로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하기 위한 수단;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하기 위한 수단; 및

상기 현재 블록을 인터-예측 디코딩하기 위한 수단으로서, 상기 인터-예측 디코딩하기 위한 수단은 상기 구역에 대한 상기 디스패리티 모션 벡터를 상기 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 디코딩하기 위한 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 45

제 44 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위한 수단은, 상기 현

제 블록을 공간적으로 이웃하는 공간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하기 위한 수단을 포함하고,

상기 픽처는 상기 공간적 이웃 구역을 포함하고 상기 현재 블록을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 46

제 45 항에 있어서,

상기 공간적 이웃 구역은 2 개의 4x4 블록들을 포함하는 8x4 구역을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 47

제 44 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위한 수단은, 상기 현재 블록을 시간적으로 이웃하는 시간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하기 위한 수단을 포함하고,

상기 픽처는 상기 시간적 이웃 구역을 포함하고 상기 현재 블록을 포함하지 않는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 48

제 47 항에 있어서,

상기 시간적 이웃 구역은 16 개의 4x4 블록들을 포함하는 16x16 구역을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 49

비디오 데이터를 인코딩하기 위한 디바이스로서,

상기 비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위한 수단으로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하기 위한 수단;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하기 위한 수단; 및

상기 현재 블록을 인터-예측 인코딩하기 위한 수단으로서, 상기 인터-예측 인코딩하기 위한 수단은 상기 구역에 대한 디스패리티 모션 벡터를 상기 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 인코딩하기 위한 수단을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 50

제 49 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위한 수단은, 상기 현재 블록을 공간적으로 이웃하는 공간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하기 위한 수단을 포함하고,

상기 픽처는 상기 공간적 이웃 구역을 포함하고 상기 현재 블록을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 51

제 50 항에 있어서,

상기 공간적 이웃 구역은 2 개의 4x4 블록들을 포함하는 8x4 구역을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 52

제 49 항에 있어서,

상기 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하기 위한 수단은, 상기 현재 블록을 시간적으로 이웃하는 시간적 이웃 구역에 대한 상기 모션 벡터가 상기 제 2 뷰에서의 픽처를 지칭한다고 결정하기 위한 수단을 포함하고,

상기 픽처는 상기 시간적 이웃 구역을 포함하고 상기 현재 블록을 포함하지 않는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 53

제 52 항에 있어서,

상기 시간적 이웃 구역은 16 개의 4x4 블록들을 포함하는 16x16 구역을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 54

명령들이 저장되어 있는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행되는 경우 하나 이상의 프로세서들로 하여금,

비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하게 하는 것으로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하게 하고;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하게 하며; 그리고

상기 현재 블록을 인터-예측 디코딩하게 하고,

상기 하나 이상의 프로세서들로 하여금, 상기 디코딩하게 하는 명령들은, 상기 하나 이상의 프로세서들로 하여금, 상기 구역에 대한 디스패리티 모션 벡터를 상기 현재 블록에 대한 디스패리티 벡터로서 사용하게 하는 명령들을 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 55

명령들이 저장되어 있는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행되는 경우 하나 이상의 프로세서들로 하여금,

비디오 데이터의 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하게 하는 것으로서, 상기 구역은 예측될 현재 블록을 이웃하고, 상기 구역은 하나 이상의 블록들을 포함하고, 상기 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 상기 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내며, 상기 구역에 대한 모션 벡터는 상기 비디오 데이터의 상기 제 1 뷰와는 상이한 상기 비디오 데이터의 제 2 뷰에서의 픽처를 지칭하는, 상기 디스패리티 모션 벡터라고 결정하게 하고;

상기 픽처에서의 상기 구역에 대한 모션 벡터가 상기 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 상기 구역에 대한 상기 디스패리티 모션 벡터가 상기 비디오 데이터의 상기 현재 블록에 대한 디스패리티 벡터라고 결정하게 하며; 그리고

상기 현재 블록을 인터-예측 인코딩하게 하고,

상기 하나 이상의 프로세서들로 하여금, 상기 인코딩하게 하는 명령들은, 상기 하나 이상의 프로세서들로 하여금, 상기 구역에 대한 디스패리티 모션 백터를 상기 현재 블록에 대한 디스패리티 백터로서 사용하게 하는 명령들을 포함하는, 비일시적 컴퓨터 관독가능 저장 매체.

청구항 56

제 1 항에 있어서,

상기 방법은 무선 통신 디바이스 상에서 실행가능하며, 상기 디바이스는:

상기 비디오 데이터를 수신하도록 구성되는 수신기;

수신된 상기 비디오 데이터를 저장하도록 구성되는 메모리; 및

상기 메모리에 저장된 상기 비디오 데이터를 프로세싱하는 명령들을 실행하도록 구성되는 프로세서를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 57

제 56 항에 있어서,

상기 무선 통신 디바이스는 셀룰라 전화기이고, 상기 비디오 데이터는 상기 수신기에 의해 수신되어 셀룰라 통신 표준에 따라 변조되는, 비디오 데이터를 디코딩하는 방법.

청구항 58

제 22 항에 있어서,

상기 디바이스는 무선 통신 디바이스이고,

상기 디바이스는 상기 비디오 데이터를 수신하도록 구성되는 수신기를 더 포함하고,

상기 메모리 유닛은 수신된 상기 비디오 데이터를 저장하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 59

제 58 항에 있어서,

상기 무선 통신 디바이스는 셀룰라 전화기이고, 상기 비디오 데이터는 상기 수신기에 의해 수신되어 셀룰라 통신 표준에 따라 변조되는, 비디오 데이터를 디코딩하기 위한 디바이스.

발명의 설명

기술 분야

[0001]

본 출원은, 2012년 3월 14일자로 출원된 미국 가출원 제 61/610,961 호; 2012년 4월 9일자로 출원된 미국 가출원 제 61/621,929 호; 2012년 4월 11일자로 출원된 미국 가출원 제 61/623,041 호; 2012년 6월 12일자로 출원된 미국 가출원 제 61/658,754 호; 2012년 8월 11일자로 출원된 미국 가출원 제 61/682,221 호; 및 2012년 8월 11일자로 출원된 미국 가출원 제 61/682,224 호의 이익을 주장하며, 이 출원들 각각의 전체 내용은 참조로서 여기에 포함된다.

[0002]

본 개시물은 비디오 코딩, 및 보다 구체적으로는 멀티-뷰 및 3 차원 (3D) 비디오 데이터를 코딩하기 위한 기법들에 관한 것이다.

배경 기술

[0003]

디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인휴대 정보단말들 (PDA들), 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러

또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC)에 의해 규정된 표준들, 현재 개발 중인 고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장들에 기재된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 그러한 비디오 압축 기법들을 구현하는 것에 의해 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 압축 기법들은 공간적 (인트라-픽처) 예측 및/또는 시간적 (인터-픽처) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 제거한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분)는 비디오 블록들로 파티셔닝될 수도 있으며, 그 비디오 블록들은 또한 트리블록들 (treeblocks), 코딩 유닛들 (CU 들) 및/또는 코딩 노드들로서 지칭될 수도 있다. 픽처의 인트라 코딩된 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃 블록들에서의 레퍼런스 샘플들에 관한 공간적 예측을 이용하여 인코딩된다. 픽처의 인터 코딩된 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃 블록들에서의 레퍼런스 샘플들에 관한 공간적 예측 또는 다른 레퍼런스 픽처들에서의 레퍼런스 샘플들에 관한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들로서 지칭될 수도 있고, 레퍼런스 픽처들은 레퍼런스 프레임들로서 지칭될 수도 있다.

[0005] 공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록을 초래한다. 잔여 데이터는 코딩될 오리지널 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터 코딩된 블록은 예측 블록을 형성하는 레퍼런스 샘플들의 블록을 가리키는 모션 벡터와, 코딩된 블록 및 예측 블록 간의 차이를 나타내는 잔여 데이터에 따라 인코딩된다. 인트라 코딩된 블록은 인트라 코딩 모드 및 잔여 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔여 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 결과적으로 잔여 변환 계수들이 생겨나며, 그 계수들은 그 다음에 양자화될 수도 있다. 처음에 2차원 어레이로 배열된 양자화된 변환 계수들은, 변환 계수들의 1차원 벡터를 생성하기 위해 스캐닝될 수도 있고, 엔트로피 코딩이 더 많은 압축을 달성하기 위해 적용될 수도 있다.

발명의 내용

과제의 해결 수단

[0006] 본 개시물에 설명된 기법들은 일반적으로, 공간적 및/또는 시간적 이웃하는 블록들의 모션 정보에 기초하여 뷰의 픽처의 현재 블록에 대한 디스패리티 벡터를 결정하는 것에 관련된다. 이 기법들은, 이웃하는 블록들의 모션 정보가 현재 블록에 대한 디스패리티 벡터를 결정하기 위해 저장되는 방식을 고려한다. 예를 들어, 본 개시물에 설명된 기법들은 현재 블록을 공간적 및/또는 시간적으로 이웃하는 이웃하는 구역들의 모션 정보를 평가할 수도 있고, 여기서 구역은 하나 또는 복수의 이웃하는 블록들을 포함한다. 기법들은 구역에 대한 저장된 모션 정보에 기초하여 현재 블록에 대한 디스패리티 벡터를 결정할 수도 있다. 기법들은 또한, 랜덤 액세스 포인트 (RAP) 픽처들의 인터-뷰 모션 예측을 허용할 뿐만 아니라, 결정된 디스패리티 벡터들을 모션 벡터 예측자 후보들로서 사용할 수도 있다.

[0007] 일 예에서, 본 개시물은 비디오 데이터를 디코딩하는 방법을 설명한다. 방법은 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 방법은 또한, 픽처의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하는 단계, 및 현재 블록을 인터-예측 디코딩하는 단계로서, 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터라고 결정되는 경우 디코딩은 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 디코딩하는 단계를 포함한다.

[0008] 일 예에서, 본 개시물은 비디오 데이터를 인코딩하는 방법을 설명한다. 방법은 복수의 뷰들의 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정하는 단계를 포함한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 방법은 또한, 픽처의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결

정하는 것에 응답하여 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하는 단계, 및 현재 블록을 인터-예측 인코딩하는 단계로서, 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터라고 결정되는 경우 인코딩은 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 인코딩하는 단계를 포함한다.

[0009] 일 예에서, 본 개시물은 비디오 데이터를 디코딩하기 위한 디바이스를 설명한다. 디바이스는 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정하도록 구성된 비디오 디코더를 포함한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 비디오 디코더는 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하도록 구성된다. 비디오 디코더는 또한, 현재 블록을 인터-예측 디코딩하도록 구성되고, 비디오 디코더는 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터라고 결정되는 경우 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용한다.

[0010] 일 예에서, 본 개시물은 비디오 데이터를 인코딩하기 위한 디바이스를 설명한다. 디바이스는 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정하도록 구성된 비디오 인코더를 포함한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 비디오 인코더는 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하도록 구성된다. 비디오 인코더는 또한, 현재 블록을 인터-예측 인코딩하도록 구성되고, 비디오 인코더는 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터라고 결정되는 경우 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용한다.

[0011] 일 예에서, 본 개시물은 비디오 데이터를 디코딩하기 위한 디바이스를 설명한다. 디바이스는 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 디바이스는 또한, 픽처의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하기 위한 수단, 및 현재 블록을 인터-예측 디코딩하기 위한 수단으로서, 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터라고 결정되는 경우 인터-예측 디코딩하기 위한 수단은 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 디코딩하기 위한 수단을 포함한다.

[0012] 일 예에서, 본 개시물은 비디오 데이터를 인코딩하기 위한 디바이스를 설명한다. 디바이스는 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 디바이스는 또한, 픽처의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하기 위한 수단, 및 현재 블록을 인터-예측 인코딩하기 위한 수단으로서, 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터라고 결정되는 경우 인터-예측 인코딩하기 위한 수단은 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용하는, 상기 인터-예측 인코딩하기 위한 수단을 포함한다.

[0013] 일 예에서, 본 개시물은 명령들이 저장되어 있는 컴퓨터 판독가능 저장 매체를 설명하고, 상기 명령들은 실행되는 경우 하나 이상의 프로세서들로 하여금, 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정하게 한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 명령들은, 하나 이상의 프로세서들로 하여금, 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하게 한다. 명령들은 또한, 하나 이상의 프로세서들로 하여금, 현재 블록을 인터-예측 디코딩하게 하고, 하나 이상의 프로세서들로 하여금, 현재 블록을 디코딩하게 하는 상기 명령들은, 구역에 대한 디스패리티 모션 벡터가 현재 블록에

대한 디스패리티 벡터라고 결정되는 경우 하나 이상의 프로세서들로 하여금, 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용하게 하는 명령들을 포함한다.

[0014] 일 예에서, 본 개시물은 명령들이 저장되어 있는 컴퓨터 관독가능 저장 매체를 설명하고, 상기 명령들은 실행되는 경우 하나 이상의 프로세서들로 하여금, 복수의 뷰들 중 제 1 뷰의 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정하게 한다. 이 예에서, 구역은 예측될 현재 블록에 이웃하고, 구역은 하나 이상의 블록들을 포함하고, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트에 대한 단일의 모션 벡터를 나타내며, 디스패리티 모션 벡터는 제 2 의, 상이한 뷰에서의 픽처를 지칭한다. 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 명령들은, 하나 이상의 프로세서들로 하여금, 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터인지 여부를 결정하게 한다. 명령들은 또한, 하나 이상의 프로세서들로 하여금, 현재 블록을 인터-예측 인코딩하게 하고, 하나 이상의 프로세서들로 하여금, 현재 블록을 인코딩하게 하는 상기 명령들은, 구역에 대한 디스패리티 모션 벡터가 현재 블록에 대한 디스패리티 벡터라고 결정되는 경우 하나 이상의 프로세서들로 하여금, 구역에 대한 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 사용하게 하는 명령들을 포함한다.

[0015] 하나 이상의 예들의 상세들은 첨부 도면들 및 이하의 설명에서 기술된다. 다른 피처들, 오브젝트들, 및 이 점들은 상세한 설명 및 도면으로부터, 그리고 청구항들로부터 명백해질 것이다.

도면의 간단한 설명

[0016] 도 1 은 본 개시물에 설명된 하나 이상의 예들에 따른 예시의 비디오 인코딩 및 디코딩 시스템을 나타내는 블록도이다.

도 2 는 본 개시물에 설명된 하나 이상의 예들에 따른 예시의 멀티뷰 비디오 코딩 (multiview video coding; MVC) 인코딩 또는 디코딩 순서를 나타내는 그래픽도이다.

도 3 은 예시의 MVC 예측 패턴을 나타내는 그래픽도이다.

도 4 는 모션 벡터들이 후보 모션 벡터 예측자들일 수도 있는 공간적으로 이웃하는 블록들을 나타내는 개념도이다.

도 5 는 디스패리티 벡터에 기초하여 후보 모션 벡터 예측자를 결정하는 방식을 나타내는 개념도이다.

도 6 은 클린 랜덤 액세스 (clean random access; CRA) 픽처의 예를 나타내는 개념도이다.

도 7 은 후보 디스패리티 벡터를 결정하는 일 예를 나타내는 개념도이다.

도 8 은 본 개시물에 설명된 하나 이상의 예들에 따라 모션 벡터 정보가 저장되는 방식을 나타내는 개념도이다.

도 9 는 후보 디스패리티 벡터를 결정하는 일 예를 나타내는 개념도이다.

도 10 은 예측될 현재 블록과 함께 위치되는 함께 위치된 (co-located) 블록들의 예들을 나타내는 개념도이다.

도 11 은 후보 디스패리티 벡터를 결정하는 일 예를 나타내는 개념도이다.

도 12 는 본 개시물에 설명된 기법들을 구현할 수도 있는 비디오 인코더의 예를 나타내는 블록도이다.

도 13 은 본 개시물에 설명된 기법들을 구현할 수도 있는 비디오 디코더의 예를 나타내는 블록도이다.

도 14 는 본 개시물에 설명된 하나 이상의 기법들에 따른 비디오 디코더의 예시의 동작을 나타내는 플로우차트이다.

도 15 는 본 개시물에 설명된 하나 이상의 기법들에 따른 비디오 인코더의 예시의 동작을 나타내는 플로우차트이다.

도 16 은 본 개시물에 설명된 하나 이상의 기법들에 따른 비디오 코더의 예시의 동작을 나타내는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 개시물에 설명된 기법들은 일반적으로, 멀티뷰 텍스처 (또는 텍스처 및 심도) 코딩 (MVC), 및 보다 구체적으로는 비디오 코더 (예를 들어, 비디오 인코더 또는 비디오 디코더) 가 MVC 에 대한 현재 뷰의 현재 픽처 내의

현재 블록에 대한 디스파리티 벡터를 결정하는 방식에 관한 것이다. MVC 는 복수의 뷰들이 코딩되는 방식을 지칭하고, 여기서 각각의 뷰는 복수의 픽처들을 포함한다. MVC 는 3 차원 (3D) 비디오를 지원하는데 사용될 수도 있다. 3D 애플리케이션에서, 디스플레이가 2 이상의 뷰들로부터 픽처들을 제시하는 경우, 뷰어는 디스플레이의 2 차원 (2D)에 제약되는 이미지 보다는, 3D 볼륨을 포괄하는 이미지를 지각한다. 3D 볼륨을 포괄하는 이미지의 뷰어 지각은 상이한 뷰들의 픽처들에서 객체들 간의 수평 디스파리티에 의한 것이다.

[0018] 디스파리티 벡터는 현재 픽처의 현재 뷰와 상이한 뷰에 있는 대응하는 픽처 내의 블록을 지칭하는 벡터이다. 대응하는 픽처 내의 블록 및 현재 픽처에서의 현재 블록은 유사한 비디오 콘텐트를 포함할 수도 있다; 그러나 현재 픽처 내의 현재 블록의 로케이션 및 대응하는 픽처 내의 블록의 로케이션 간에는 수평 디스파리티가 존재한다. 현재 블록의 디스파리티 벡터는 대응하는 픽처에서의 블록과 현재 픽처에서의 현재 블록 간의 이 수평 디스파리티의 측정을 제공한다. 다른 예에서, 현재 픽처 내의 현재 블록의 로케이션과 대응하는 픽처 내의 블록의 로케이션 간에는 또한 수직 디스파리티가 존재할 수도 있다. 현재 블록의 디스파리티 벡터는 대응하는 픽처에서의 블록과 현재 픽처에서의 현재 블록 간의 이 수직 디스파리티의 측정을 제공할 수도 있다.

현재 뷰의 현재 픽처 및 상이한 뷰의 대응하는 픽처가 디스플레이되는 때의 시간은 동일할 수도 있다 (즉, 현재 뷰 및 상이한 뷰는 동일한 시간 인스턴스에 대한 뷰들이다).

[0019] 이하에서 더 상세히 설명되는 바와 같이, 비디오 코더는 현재 블록에 대해 하나 이상의 공간적 및 시간적으로 이웃하는 블록들에 대한 모션 정보에 기초하여 디스파리티 벡터를 결정한다. 그러나, 이들 공간적 및 시간적으로 이웃하는 블록들에 대한 모션 정보가 저장되는 방식 때문에, 일부 예들에서 비디오 코더는 공간적 및 시간적으로 이웃하는 블록 대 블록 (block-by-block) 단위로 모션 정보를 결정할 필요가 없을 수도 있다. 차라리, 비디오 코더는 구역 대 구역 (region-by-region) 단위로 모션 정보를 결정하도록 구성될 수도 있고, 여기서 하나의 구역은 하나 이상의 이웃하는 블록들을 포함하고 현재 블록의 사이즈보다 더 클 수도 있다. 이 예에서, 비디오 코더는, 공간적 및 시간적으로 이웃하는 블록들에 대한 모션 정보가 저장되어 비디오 코더가 블록 대 블록 단위로 모션 정보를 결정하는 것에 비해 모션 정보의 더 적은 결정들을 초래할 수도 있는 구역 대 구역 단위로 모션 정보를 결정하도록 저장되는 방식을 고려한다.

[0020] 일부 예들에서, 비디오 코더는 반드시 픽처에 대한 심도 맵을 도출하는 것을 필요로 하지 않고 또는 반드시 글로벌 디스파리티 벡터 (global disparity vector; GDV) 를 코딩하지 않고 디스파리티 벡터를 결정하는데, 여기서 심도 맵은 픽처에서의 화소들의 상대적인 심도를 나타낸다. 도출된 심도 맵들로부터 결정되는 디스파리티 벡터들은 평활한 시간적-뷰 예측된 (smooth temporal-view predicted; STV) 디스파리티 벡터들로 지칭된다. GDV 는 각각의 화소에 동일한 디스파리티 벡터를 할당하는 벡터이다. 일부 다른 예들에서, STV 디스파리티 벡터들은 각각의 예측 유닛/코딩 유닛/매크로블록/매크로블록 파티션들에 대해 적용될 수도 있고, 따라서 이를 각각은 그 자신의 STV 를 갖는다.

[0021] 픽처에 대한 심도 맵을 도출하는 것은 프로세싱 및 시간-집약적일 수 있다. 따라서, 픽처에 대한 심도 맵을 반드시 도출할 필요가 없는 블록에 대한 디스파리티 벡터를 결정하는 것은 효율적인 비디오 코딩을 촉진한다. 또한, 심도 맵이 이용 가능하더라도 본 개시물에 설명된 기법들이 이용될 수도 있다. 이 기법들은 또한, 심도 맵 코딩 도구들에 대해 유용할 수도 있다. 또한, GDV 는 상이한 뷰들에서 대응하는 블록들 간의 디스파리티의 정확한 설명을 제공하지 않는다. 예를 들어, 하나의 뷰에서의 픽처의 2 개의 블록들은 다른 뷰에서의 픽처에서 상이한 양들 만큼 변위될 수도 있다. GDV 는 모든 화소들에 동일한 디스파리티 벡터를 할당하기 때문에, GDV 는 2 개의 블록들의 디스파리티의 상이한 양들을 정확하게 정의하지 않는다.

[0022] 디스파리티 벡터는 여러 코딩 도구들, 예컨대 인터-뷰 모션 예측, 인터-뷰 잔여 예측에서 사용될 수 있다. 예를 들어, 인터-뷰 모션 예측은 디스파리티 벡터가 2 개의 방법들에서 사용될 수 있는 일 예이다. 제 1 방법에서, 인터-뷰 모션 예측을 위해, 현재 블록의 디스파리티 벡터가 지칭하는 블록에 대한 모션 벡터는 현재 블록의 모션 벡터에 대한 복수의 잠재적인 모션 벡터의 모션 벡터 예측자일 수도 있다. 비디오 코더가 이 모션 벡터 예측자 (예를 들어, 현재 블록의 디스파리티 벡터에 의해 지칭된 블록의 모션 벡터) 를 선택하는 경우, 비디오 코더는 이 모션 벡터 예측자를 머지 (merge) 모드 또는 스킵 모드 (간략화를 위해, 다음의 설명에서 머지 모드는 머지 및 스kip 모드 양자를 나타내도록 사용됨) 에서 현재 블록의 모션 벡터로서 이용할 수도 있고, 이 모션 벡터 예측자를 이용하여 어드밴스트 모션 벡터 예측 (advanced motion vector prediction; AMVP) 모드에서 현재 블록의 모션 벡터를 결정할 수도 있다. 이 경우에서, 디스파리티 벡터로부터 컨버팅된 모션 벡터는 AMVP 또는 머지 모드에 대한 추가의 후보로서 사용된다. 제 2 방법에서, 인터뷰 모션 예측을 위해, 디스파리티 벡터는 레퍼런스 뷰에서 하나의 대응하는 블록을 위치시키는데 사용되고, 대응하는 블록의 모션 정보는

AMVP 또는 머지 모드에 대한 후보로 컨버팅될 수도 있다.

[0023] 예를 들어, 머지 모드의 예들에서, 비디오 코더는 모션 벡터 예측자를 현재 블록의 모션 벡터로서 설정하고, 모션 벡터 예측자에 의해 지칭된 레퍼런스 픽처를 이용하여 현재 블록을 인터-예측할 수도 있다. AMVP 모드에서, 비디오 코더는 현재 블록의 실제 모션 벡터와 모션 벡터 예측자 간의 잔여분들을 코딩할 수도 있다. 이 예에서, 비디오 코더는 이 잔여분들을 모션 벡터 예측자에 가산하거나 이 모션 벡터 예측자로부터 잔여분들을 감산하여 현재 블록의 모션 벡터를 결정한다. 또한, AMVP 모드에서, 비디오 코더는 레퍼런스 인덱스를 제1 및 제2 레퍼런스 픽처 리스트들 중 적어도 하나로 코딩하여 현재 블록을 인터-예측하는데 사용되는 레퍼런스 픽처를 식별한다.

[0024] 반드시 심도 맵을 도출할 필요 없이 디스패리티 벡터를 결정함으로써, 비디오 코더는, 비디오 코더가 먼저 심도 맵을 도출하여 디스패리티 벡터를 결정하는 예들에 비해 더 효율적인 방식으로 디스패리티 벡터가 지칭하는 블록에 대한 모션 벡터를 결정하도록 구성될 수도 있다. 또한, GDV에 의존하지 않음으로써, 비디오 코더는 결정된 디스패리티 벡터를 더 정확한 양의 디스패리티에 제공하도록 구성될 수도 있고, 여기서 결정된 디스패리티 벡터는 머지 모드 또는 AMVP 모드에서 모션 벡터가 사용되는 블록을 더 정확하게 정의한다. 디스패리티가 지칭하는 블록에 대한 모션 벡터가 머지 모드 및 AMVP 모드에서 현재 블록에 대한 잠재적인 모션 벡터 예측자이기 때문에, 본 개시물에 설명된 기법들에서 비디오 코더는 일부 다른 기법들에 비해 더 효율적인 방식으로 머지 모드 및 AMVP 모드를 구현하도록 구성될 수도 있다.

[0025] 일부 예들에서, 비디오 코더는 디스패리티 벡터들의 후보 리스트에 기초하여 블록에 대한 디스패리티 벡터를 결정하도록 구성된다. 예를 들어, 심도 맵을 도출하고 이 심도 맵으로부터 디스패리티 벡터를 도출하기보다는, 일부 예들에서 비디오 코더는 후보 디스패리티 벡터들의 리스트를 구성하고, 후보 디스패리티 벡터들 중 하나를 선택하며, 선택된 후보 디스패리티 벡터에 기초하여 현재 블록에 대한 디스패리티 벡터를 결정한다. 비디오 코더가 후보 디스패리티 벡터들 중 하나를 선택하는 다양한 방법들이 존재할 수도 있고, 본 개시물에 설명된 기법들은 후보 디스패리티 벡터를 선택하는 임의의 특정 방법에 제한되지 않는다.

[0026] 심도 맵을 반드시 도출하지 않고 디스패리티 벡터를 결정하는 것은 여러 코딩 도구들, 예컨대 소정 유형들의 픽처들에 대한 인터-뷰 모션 예측/인터-뷰 잔여 예측을 허용할 수도 있다. 예를 들어, 인터-뷰 모션 예측은 본 개시물에서 설명된 기법들을 이용하는 클린 랜덤 액세스 (CRA) 및 순간 디코더 리프레시 (instantaneous decoder refresh; IDR) 픽처들과 같은 랜덤 액세스 포인트 (RAP) 픽처에 적용될 수도 있다. 일부 다른 기법들 (예를 들어, 본 개시물에 설명된 기법들에 따르지 않은 기법들)에서, RAP 픽처들은 RAP 픽처들의 추정된 심도 맵을 도출하기 위해 충분히 디코딩되었고 디스패리티 벡터가 추정된 심도 맵에 의존하기 때문에 인터-뷰 모션 예측은 소정의 비-기본 뷰들 (예를 들어, 의존적 뷰들)에서 RAP 픽처들에 적용될 수 없다. 심도 맵을 필요로 하지 않고 디스패리티 벡터들을 도출함으로써, 본 개시물에 설명된 기법들은 비-기본 뷰들의 RAP 픽처들에 대한 인터-뷰 모션 예측을 허용할 수도 있다.

[0027] 일부 예들에서, 비디오 코더는 공간적 및 시간적 이웃하는 블록들에 기초하여 후보 디스패리티 벡터들의 리스트를 구성한다. 예를 들어, 현재 블록에 공간적 또는 시간적 이웃하는 블록이 인터-뷰 예측을 사용하였으면, 이러한 블록들에 대한 디스패리티 모션 벡터는 후보 디스패리티 벡터로서 잠재적으로 포함될 수도 있다. 인터-뷰 예측은 현재 블록을 인터-예측하는데 사용된 레퍼런스 블록을 포함하는 레퍼런스 픽처가 현재 블록을 포함하는 현재 픽처가 위치되는 뷰와 상이한 뷰에 위치되는 인터-예측을 지칭한다. 디스패리티 모션 벡터는 제2의, 상이한 뷰에서의 픽처의 블록을 지칭하는 제1 뷰에서의 픽처의 블록에 대한 벡터이다.

[0028] 전술된 바와 같이, 본 개시물에 설명된 기법들에 따르면, 후보 디스패리티 벡터들의 리스트를 구성하는데 있어서, 비디오 코더는 공간적 및 시간적으로 이웃하는 블록들에 대한 모션 벡터 정보가 저장되는 방식을 고려할 수도 있다. 예를 들어, 시간적으로 이웃하는 블록을 포함하는 레퍼런스 픽처에 대해, 비디오 코더는 모든 예에서 레퍼런스 픽처에서의 각각의 4x4 블록에 대한 모션 벡터들을 저장하지 않고, 일부 예들에서 레퍼런스 픽처에서의 각각의 4x4 블록에 대한 모션 벡터들을 저장할 수도 있다. 일부 예에서, 비디오 코더는 16x16 구역에 대한 하나의 레퍼런스 픽처 리스트에 하나의 모션 벡터를 저장하도록 구성될 수도 있다 (여기서, 16x16 구역은 16 개의 4x4 블록들을 포함한다). 16x16 구역에 대한 이 하나의 모션 벡터는 레퍼런스 픽처 리스트에서 구역 내의 16 개의 4x4 블록들 각각에 대한 단일의 모션 벡터로서 고려될 수도 있다.

[0029] 간략화를 위해, 다음의 설명에서, 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트의 모션 벡터를 나타낸다. 또한, 레퍼런스 픽처로부터든 또는 현재 픽처 내로부터든 구역은 하나 이상의 블록들을 포함할 수도 있다. 예시의 용이함을 위해, 기법들은 구역이 1보다 많은 블록을 포함하는 예들을 설명하였으나, 구역이 하

나 이상의 블록들을 포함할 수도 있음을 이해해야 한다.

[0030] 상기 예에서, 레퍼런스 픽처에서의 16x16 구역의 각각의 4x4 블록은 상이한 모션 벡터들에 의해 코딩될 수도 있으나, 비디오 코더가 모션 벡터 정보를 저장하는 경우 16x16 구역의 각각의 4x4 블록에 대한 개별의 모션 벡터 정보가 손실되고, 각각의 4x4 블록에 대해 동일한 하나의 모션 벡터로 대체된다. 16x16 구역에서 모션 벡터를 저장하는 다른 이점은 메모리 요건이 상당히 감소된다는 것이다 (즉, 16x16 구역 내의 각각의 4x4 블록에 대한 모션 벡터를 저장할 필요가 없다). 비디오 코더가 16x16 구역에 대한 단일의 모션 벡터를 결정하는 상이한 방법들이 존재할 수도 있고, 본 개시물에 설명된 기법들은 비디오 코더가 16x16 구역에 대한 단일의 모션 벡터를 결정하는 임의의 특정 방법에 제한되지 않는다.

[0031] 비디오 코더가 블록 단위 보다는 구역에 대한 모션 정보를 저장하는 예들에서, 비디오 코더가 현재 블록에 시간적으로 이웃하는 매 단일의 4x4 블록의 모션 벡터 정보를 결정하는 것이 필요하지 않을 수도 있다. 예를 들어, 후보 디스패리티 벡터를 결정하기 위해, 비디오 코더는 레퍼런스 픽처 내의 다음의 로케이션들을 체크할 수도 있다: 현재 블록에 함께-위치된 구역, 함께-위치된 구역을 커버하는 최대 코딩 유닛 (LCU) 구역 내의 블록들, 및 하부-우측 블록. 그러나, 비디오 코더가 모션 벡터 정보를 저장했던 방식으로 인해 이들 체크된 영역들 모두에 대한 모션 벡터 정보가 동일한 것이 가능할 수도 있다. 따라서, 비디오 코더는 현재 블록에 대한 후보 디스패리티 벡터를 결정하기 위한 목적으로 이들 블록들 모두에 대한 모션 벡터 정보를 결정할 필요가 없을 수도 있다.

[0032] 대신에, 비디오 코더는 16x16 구역에 대한 모션 벡터 정보를 결정할 수도 있다. 이 16x16 구역에 대한 모션 벡터가 디스패리티 모션 벡터이면, 이 디스패리티 모션 벡터는 현재 블록에 대한 디스패리티 벡터를 결정하는데 사용되는 후보 디스패리티 벡터일 수도 있다. 이 예에서, 16x16 구역은 4x4 시간적으로 이웃하는 블록들 각각보다 사이즈가 더 크다. 또한, 16x16 구역은 현재 블록의 사이즈보다 사이즈가 더 클 수도 있다. 비디오 코더는 블록 대 블록 단위로 4x4 시간적으로 이웃하는 블록들을 체크할 수도 있는 것으로 이해되어야 한다.

[0033] 상기 예에서, 비디오 코더는 현재 블록에 시간적으로 이웃하는 구역에 기초하여 후보 디스패리티 벡터를 결정한다. 전술된 바와 같이, 일부 예들에서, 비디오 코더는 또한, 임의의 공간적으로 이웃하는 블록들이 인터-뷰 예측되는지 여부를 평가하고, 현재 블록의 디스패리티 벡터에 대한 후보들로서 이들 공간적으로 이웃하는 블록들에 대한 디스패리티 모션 벡터를 이용한다. 시간적으로 이웃하는 블록들의 경우에 유사하게, 일부 예들에서, 비디오 코더는 공간적으로 이웃하는 블록들에 대한 모든 모션 벡터 정보를 저장하지 않는다.

[0034] 예를 들어, 현재 블록을 포함하는 라인 위의 라인에서 블록들을 인터-예측하는 동안, 비디오 코더는 이 라인 (즉, 현재 블록을 포함하는 라인 위의 라인)에서의 블록들에 대한 모션 벡터 정보를 라인 버퍼에 저장한다. 라인 버퍼에서, 비디오 코더는 각각의 4x4 블록에 대한 모션 벡터 정보 모두를 저장하지 않을 수도 있다. 차라리, 2 개의 블록들이 동일한 모션 벡터 정보를 공유할 수도 있다. 그러나, 동일한 모션 벡터 정보를 공유하는 2 개의 블록들은 모든 예들에서 요구되지 않고, 비디오 코더는 각각의 4x4 블록에 대한 모션 벡터 정보 모두를 저장할 수도 있다.

[0035] 예를 들어, 4 개의 4x4 블록들이 라인에 배열된다고 가정한다. 라인에 배열된 이들 4 개의 4x4 블록들 각각에 대한 모션 벡터가 존재할 수도 있고, 이들 4 개의 4x4 블록들에 대한 모션 벡터들이 상이할 수도 있다.

[0036] 이 예에서, 제 1 및 제 2 4x4 블록들에 대한 모션 벡터들이 상이할 수도 있으나, 저장하는 경우 제 1 및 제 2 4x4 블록들은 동일한 모션 벡터 정보를 공유하고, 비디오 코더는 이들 블록들 양자에 대한 하나의 모션 벡터를 라인 버퍼에 저장한다. 이 하나의 모션 벡터는 제 1 및 제 2 4x4 블록들에 대한 단일의 모션 벡터를 나타낸다.

[0037] 일부 예들에서, 비디오 코더는 제 1 및 제 2 블록들 양자에 대한 모션 벡터로서 제 1 블록에 대한 모션 벡터 정보를 저장하여 제 2 4x4 블록에 대한 모션 벡터 정보가 손실되고, 제 1 4x4 블록에 대한 모션 벡터 정보가 보존된다. 또한, 이 예에서, 제 3 및 제 4 4x4 블록들에 대한 모션 벡터들이 상이할 수도 있으나, 저장하는 경우 제 3 및 제 4 4x4 블록들은 동일한 모션 벡터 정보를 공유하고, 비디오 코더는 이들 블록들 양자에 대한 하나의 모션 벡터를 라인 버퍼에 저장할 수도 있다. 이 하나의 모션 벡터는 제 3 및 제 4 4x4 블록들에 대한 단일의 모션 벡터를 나타낸다. 일부 예들에서, 비디오 코더는 제 3 및 제 4 블록들 양자에 대한 모션 벡터로서 제 4 블록에 대한 모션 벡터 정보를 저장하여 제 2 4x4 블록에 대한 모션 벡터 정보가 손실되고, 제 4 4x4 블록에 대한 모션 벡터 정보는 보존된다.

- [0038] 일부 예들에서, 저장되는 모션 벡터 정보는 제 1 및 제 2 4x4 블록들 또는 제 3 및 제 4 4x4 블록들로부터 도출될 수도 있고, 도출된 모션 벡터 정보는 2 개의 블록들 중 어느 하나와 동일하지 않을 수도 있다. 일반적으로, 비디오 코더는 라인 베퍼에 저장될 필요가 있는 모션 벡터 정보를 감소시키도록 임의의 기법을 이용할 수도 있고, 전술된 기법들 (예를 들어, 2 개의 블록들에 대한 모션 벡터 정보로서 하나의 블록에 대한 모션 벡터 정보를 저장하는 것, 또는 2 개의 블록들에 대한 모션 벡터 정보를 도출하는 것) 은 예시의 목적을 위해 제공되고 제한하는 것으로 고려되지 않아야 한다.
- [0039] 상기 예에서, 비디오 코더는 현재 블록을 포함하는 라인 위의 라인 내에 8x4 구역들에 대한 모션 벡터 정보를 저장하고, 여기서 각각의 8x4 구역은 2 개의 4x4 블록들을 포함한다. 이 경우에서, 비디오 코더는 현재 블록에 공간적으로 이웃하는 블록들 중에서 단일의 4x4 블록마다에 대해 모션 벡터 정보를 결정할 필요가 없을 수도 있다. 예를 들어, 공간적으로 이웃하는 블록들 중 일부에 대한 모션 벡터 정보는 동일한 것이 가능하다. 이 예에서, 블록들 중 단지 하나에 대한 모션 벡터 정보를 결정하는 것은 후보 디스패리티 벡터들의 리스트에 어느 후보들이 포함될 수 있는지를 결정하기에 충분할 수도 있다.
- [0040] 다시 말해, 비디오 코더는 공간적으로 이웃하는 블록들 중에서 4x4 블록에 대한 모션 벡터 정보를 결정하지 않을 수도 있다. 차라리, 비디오 코더는 (이 예에서) 8x4 구역에 대한 모션 벡터 정보를 결정할 수도 있고, 여기서 8x4 구역은 현재 블록을 각각 공간적으로 이웃하는 2 개의 4x4 블록들을 포함한다 (예를 들어, 2 개의 4x4 블록들을 나타내는 단일 모션 벡터). 이 8x4 구역에 대한 모션 벡터가 디스패리티 모션 벡터이면, 이 디스패리티 모션 벡터는 현재 블록에 대한 디스패리티 벡터를 결정하는데 사용되는 후보 디스패리티 벡터일 수도 있다. 이 예에서, 8x4 구역은 4x4 공간적으로 이웃하는 블록들 각각 보다 사이즈가 더 크다. 또한, 8x4 구역은 현재 블록의 사이즈보다 사이즈가 더 클 수도 있다.
- [0041] 더욱이, 본 개시물에 설명된 기법들은 일부 비디오 코딩 기법들에 존재하는 다른 이슈들을 극복할 수도 있다. 예를 들어, 공간적 및 시간적 이웃하는 블록들에 대한 디스패리티 모션 벡터들은 수평 컴포넌트 및 수직 컴포넌트 양자 (예를 들어, x-좌표 및 y-좌표 양자) 를 포함할 수도 있다. 그러나, 디스패리티 벡터들의 일부 예들은 단지 수평 컴포넌트만을 포함한다. 따라서, 일부 예들에서, 공간적 또는 시간적 이웃하는 블록의 디스패리티 모션 벡터를 사용하는 경우, 비디오 코더는 후보 디스패리티 벡터가 x-좌표만을 포함하도록 y-좌표 값을 0 으로 설정할 수도 있다. 다시, 모든 디스패리티 벡터들이 반드시 단지 수평 컴포넌트 만을 포함할 필요가 있는 것은 아니고, 수평 및 수직 컴포넌트들 양자를 포함할 수도 있다.
- [0042] 반대로, 이하에서 더 상세히 설명되는 바와 같이, 디스패리티 벡터가 디스패리티 모션 벡터로서 사용되는 경우, 디스패리티 모션 벡터는 y-컴포넌트를 포함하지 않을 수도 있다. 그러나, 일부 예들에서, 로케이션이 예측되고 있는 블록과 정확하게 동일한 y-좌표에 있지 않는 블록으로부터 현재 블록을 인터-뷰 예측하는 것이 유리할 수도 있다. 다시 말해, 인터-뷰 예측되고 있는 블록으로부터 단지 수평으로 변위되는 레퍼런스 뷰에서의 레퍼런스 픽처의 블록은 반드시 예측을 위한 최선의 블록이 아닐 수도 있다. 이 경우, 디스패리티 모션 벡터가 레퍼런스 뷰에서의 레퍼런스 픽처에서 수평으로 변위된 블록 위나 아래에 있는 블록을 식별하기 위해 y-좌표를 포함하는 것이 바람직할 수도 있다. 일부 예들에서, 본 개시물에 설명된 기법들은, 디스패리티 모션 벡터가 디스패리티 벡터로부터 도출되는 경우 디스패리티 모션 벡터가 y-좌표를 포함하는 것을 가능하게 한다.
- [0043] 도 1 은 본 개시물에서 설명된 하나 이상의 예들에 따른 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다. 예를 들어, 시스템 (10) 은 소스 디바이스 (12) 와 목적지 디바이스 (14) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소스 디바이스 (12) 및 목적지 디바이스 (14) 각각이 상이한 뷰들의 픽처들을 코딩하는 멀티뷰 코딩 (MVC) 을 구현하도록 구성된다. 상이한 뷰들의 픽처들이 함께 보여지는 경우, 뷰에는, 디스플레이의 2D 영역에 제한되는 이미지 대신, 3D 볼륨을 포괄하는 이미지를 지각한다.
- [0044] 시스템 (10) 은 멀티뷰 코딩의 상이한 비디오 코딩 표준들, 독점 표준, 또는 임의의 다른 방도에 따라 동작할 수도 있다. 다음은 비디오 코딩 표준들의 몇몇 예들을 설명하고, 제한하는 것으로 고려되지는 않는다. 예를 들어, 비디오 코딩 표준들은, 예컨대 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 그리고 SVC (Scalable Video Coding) 및 MVC (Multiview Video Coding) 확장들을 포함한 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC로 알려짐) 를 포함한다. 최근, MVC 공동 초안은 "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264, Mar 2010 에 설명된다. MVC 의 더 최근의, 공개적으로 입수 가능한 공동 초안이 "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264, June 2011 에 설명된다. MVC 의 현재 공동 초안은 2012년 1월 현재 승인되었다.

[0045] 또한, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 및 ISO/IEC 동 핵심 전문가 그룹 (MPEG) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 현재 개발 중인 새로운 비디오 코딩 표준, 즉 고 효율 비디오 코딩 (HEVC) 이 있다. HEVC WD9라고 지칭되는 HEVC 의 최근의 규격 초안 (WD) 은, 2013년 3월 16일, http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v10.zip 에서 입수가능하다.

[0046] 단지 예시의 목적으로, 본 개시물에 설명된 기법들은 H.264 및 HEVC 표준들에 따른 예들로 설명된다. 일부 예들에서, 본 개시물에 설명된 기법들은 H.264 3D-AVC 표준, H.264 MVC+Depth (H.246 MVC+D) 표준, 및 3D-HEVC 표준에 적용 가능할 수도 있다. 그러나, 본 개시물에 설명된 기법들은 이를 예시의 표준들에 제한되는 것으로 고려되지 않고, 멀티뷰 코딩 또는 3D 비디오 코딩을 위한 다른 비디오 코딩 표준들로, 또는 특정 비디오 코딩 표준에 반드시 기초하지 않는 멀티뷰 코딩 또는 3D 코딩에 관련된 기법들로 확장 가능할 수도 있다. 예를 들어, 본 개시물에 설명된 기법들은 멀티뷰 코딩을 위한 비디오 인코더들/디코더들 (코덱들) 에 의해 구현되고, 여기서 멀티뷰 코딩은 2 이상의 뷰들의 코딩을 포함한다.

[0047] 도 1에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는, 이른바 "스마트" 폰들, 이른바 "스마트" 패드들과 같은 무선 핸드셋, 또는 무선 통신을 위해 장착된 다른 이러한 무선 디바이스들을 포함한 넓은 범위의 디바이스들 중 임의의 것을 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 의 부가적인 예들은, 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템 내 디바이스, 무선 브로드캐스트 시스템 내 디바이스, 개인휴대 정보단말들 (PDA), 랩톱 컴퓨터, 테스크톱 컴퓨터, 태블릿 컴퓨터, e-북 리더, 디지털 카메라, 디지털 레코딩 디바이스, 디지털 미디어 플레이어, 비디오 게이밍 디바이스, 비디오 게임 콘솔, 셀룰러 무선 전화기, 위성 무선 전화기, 비디오 원격회의 디바이스, 및 비디오 스트리밍 디바이스 등을 포함하지만 그것들로 제한되지 않는다.

[0048] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 링크 (16) 를 통해 수신할 수도 있다. 링크 (16) 는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 이동시킬 수 있는 임의의 유형의 매체 또는 디바이스를 포함한다. 일 예에서, 링크 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접 실시간으로 송신하는 것을 가능하게 하는 통신 매체를 포함한다. 인코딩된 비디오 데이터는 통신 표준, 예컨대 무선 통신 프로토콜에 따라 변조되고 목적지 디바이스 (14) 로 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 기반 네트워크, 예컨대 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0049] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (33) 로 출력된다. 마찬가지로, 인코딩된 데이터는 입력 인터페이스 (28) 에 의해 저장 디바이스 (33) 로부터 액세스된다. 저장 디바이스 (33) 의 예들은 하드 드라이브, 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휴발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산형 또는 국소적으로 액세스되는 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다. 추가의 예에서, 저장 디바이스 (33) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 유지하는 파일 서버 또는 다른 중간 저장 디바이스에 대응한다. 이들 예들에서, 목적지 디바이스 (14) 는 저장 디바이스 (33) 로부터의 저장된 비디오 데이터에 스트리밍 또는 다운로드를 통해 액세스한다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 유형의 서버이다.

예시의 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, 네트워크 부속 스토리지 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스한다. 이는 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 양자 모두의 조합을 포함할 수도 있다. 저장 디바이스 (33) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양자 모두의 조합일 수도 있다.

[0050] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들에 반드시 제한되지는 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 예컨대 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, (예컨대, 인터넷을 통한) 스트리밍 비디오 송신들 중 임의의 것의 지원 하의

비디오 코딩, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 텔레포니와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성된다.

[0051] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22)를 포함한다. 일부 경우들에서, 출력 인터페이스 (22)는 변조기/복조기 (모뎀) 및/또는 송신기를 포함한다. 소스 디바이스 (12)에서, 비디오 소스 (18)는 비디오 캡처 디바이스 (예컨대, 비디오 카메라), 이전에 캡처된 비디오를 담고 있는 비디오 아카이브, 비디오 콘텐트 제공자로부터 비디오를 수신하는 비디오 피드 인터페이스, 및/또는 컴퓨터 그래픽 데이터를 소스 비디오로서 생성하는 컴퓨터 그래픽 시스템과 같은 소스, 그러한 소스들의 조합, 또는 임의의 다른 소스 중 하나 이상을 포함한다. 일 예로서, 비디오 소스 (18)가 비디오 카메라이면, 소스 디바이스 (12)와 목적지 디바이스 (14)는 이른바 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용가능하고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0052] 비디오 인코더 (20)는 캡처된, 사전 캡처된, 또는 컴퓨터 생성된 비디오를 인코딩한다. 소스 디바이스 (12)의 출력 인터페이스 (22)는 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신하도록 구성된다. 인코딩된 비디오 데이터는 또한 (또는 대안으로) 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 디코딩 및/또는 플레이백을 위한 나중의 액세스를 위해 저장 디바이스 (33)에 저장될 수도 있다.

[0053] 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 일부 경우들에서, 입력 인터페이스 (28)는 수신기 및/또는 모뎀을 포함한다. 목적지 디바이스 (14)의 입력 인터페이스 (28)는 링크 (16)를 통해 또는 저장 디바이스 (33)로부터 인코딩된 비디오 데이터를 수신한다. 링크 (16)를 통해 통신되거나 또는 저장 디바이스 (33) 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터의 디코딩 시에, 비디오 디코더, 예컨대 비디오 디코더 (30)에 의한 사용을 위해 비디오 인코더 (20)에 의해 생성된 다양한 선택스 엘리먼트들을 포함한다. 이러한 선택스 엘리먼트들은 통신 매체 상에서 송신되는, 저장 매체 상에 저장되는, 또는 파일 서버에 저장되는 인코딩된 비디오 데이터에 포함될 수도 있다.

[0054] 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합되거나, 또는 그것 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14)는 통합형 디스플레이 디바이스를 포함하고 또한 외부 디스플레이 디바이스와 인터페이싱하도록 구성된다. 다른 예들에서, 목적지 디바이스 (14)는 디스플레이 디바이스이다. 일 반적으로, 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함한다.

[0055] 비디오 인코더 (20) 및 비디오 디코더 (30)는 상기에서 나열된 예들과 같은 각종 비디오 코딩 표준들에 따라 동작할 수도 있다. 그러나, 본 개시물에 설명된 기법들은 이와 같이 제한하는 것으로 고려되지 않아야 한다. 설명의 목적을 위해, 비디오 인코더 (20) 및 비디오 디코더 (30)는 HEVC 또는 H.264 표준 및 이러한 표준들의 확장들의 맥락에서 설명되고, H.264 3D-AVC 표준, H.264 MVC+D 표준, 및 3D-HEVC 표준과 같은 표준들에 적용 가능할 수도 있다. 그러나, 본 개시물의 기법들은 임의의 특정 코딩 표준에 제한되지 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263을 포함한다. 0n2 VP6/VP7/VP8로서 지정된 것들과 같은 사유 코딩 기법들이 또한, 본원에 설명된 기법들 중 하나 이상을 구현할 수도 있다.

[0056] 비록 도 1에 도시되지 않았지만, 일부 양태들에서, 비디오 인코더 (20)와 비디오 디코더 (30)는 각각이 오디오 인코더 및 디코더와 통합되고, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함하여, 공통 데이터 스트림 또는 개별 데이터 스트림들에서의 오디오 및 비디오 양자 모두의 인코딩을 핸들링한다. 적용가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP)과 같은 다른 프로토콜들을 준수한다.

[0057] 비디오 인코더 (20)와 비디오 디코더 (30) 각각은 다양한 적합한 인코더 회로, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 별개의 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그것들의 임의의 조합 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행하기 위해, 적합한 비일시적 컴퓨터 판독가능 매체 내에 소프트웨어에 대한 명령들을 저장하고 하나 이상의 프

로 세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC)의 일부로서 개별 디바이스 내에 통합될 수도 있다. 본 개시물에서 설명된 기법들에 따라 비디오 인코더 (20)는 비디오 데이터를 인코딩하도록 구성되고 비디오 디코더 (30)는 비디오 데이터를 디코딩하도록 구성된다.

[0058] 도 2는 본 개시물에서 설명되는 하나 이상의 예들에 따른, 예시적인 멀티뷰 코딩 (MVC) 인코딩 또는 디코딩 순서를 예시하는 그래픽 도면이다. 예를 들어, 도 2에 예시된 디코딩 순서 배열은 시간 우선 (time-first) 코딩으로서 지칭된다. 도 2에서, S0 내지 S7 각각은 멀티뷰 비디오의 상이한 뷰들을 지칭한다. T0 내지 T8 각각은 하나의 출력 시간 인스턴스를 나타낸다. 액세스 유닛이 하나의 출력 시간 인스턴스에 대한 모든 뷰들의 코딩된 픽처들을 포함할 수도 있다. 예를 들어, 제 1 액세스 유닛은 시간 인스턴스 T0에 대한 뷰들 (S0 내지 S7)의 모두 (즉, 픽처들 (0 내지 7))를 포함하며, 제 2 액세스 유닛은 시간 인스턴스 T1에 대한 뷰들 (S0 내지 S7)의 모두 (즉, 픽처들 (8 내지 15))를 포함하는 등등이다. 이 예들에서, 픽처들 (0 내지 7)은 동일한 시간 인스턴스 (즉, 시간 인스턴스 T0)에 있고, 픽처들 (8 내지 15)은 동일한 시간 인스턴스 (즉, 시간 인스턴스 T1)에 있다. 동일한 시간 인스턴스를 갖는 픽처들은 일반적으로 동시에 디스플레이되고, 뷰어로 하여금 3D 볼륨을 포괄하는 이미지를 지각하게 하는 동일한 시간 인스턴스의 픽처들 내의 객체들 간에는 수평 디스패리티, 및 가능하게는 얼마간의 수직 디스패리티가 있다.

[0059] 도 2에서, 뷰들의 각각은 픽처들의 세트들을 포함한다. 예를 들어, 뷰 S0은 픽처들 (0, 8, 16, 24, 32, 40, 48, 56, 및 64)의 세트를 포함하며, 뷰 S1은 픽처들 (1, 9, 17, 25, 33, 41, 49, 57, 및 65)의 세트를 포함하는 등등이다. 각각의 세트는 2개의 픽처들을 포함하는데, 하나의 픽처는 텍스처 뷰 컴포넌트로서 지칭되고, 다른 픽처는 심도 뷰 컴포넌트로서 지칭된다. 뷰의 픽처들의 세트 내의 텍스처 뷰 컴포넌트 및 심도 뷰 컴포넌트는 서로 대응하는 것으로 간주될 수도 있다. 예를 들어, 뷰의 픽처들의 세트 내의 텍스처 뷰 컴포넌트는 뷰의 픽처들의 세트 내의 심도 뷰 컴포넌트에 대응하는 것으로 간주될 수 있고, 그 반대의 경우도 마찬가지인 것으로 간주될 수도 있다 (즉, 심도 뷰 컴포넌트는 세트에서의 그것의 텍스처 뷰 컴포넌트에 대응하고, 그 반대의 경우도 마찬가지이다). 본 개시물에서 사용된 바와 같이, 대응하는 텍스처 뷰 컴포넌트 및 심도 뷰 컴포넌트는 단일 액세스 유닛의 동일한 뷰의 부분이라고 간주될 수도 있다.

[0060] 텍스처 뷰 컴포넌트는 디스플레이되는 실제 이미지 콘텐트를 포함한다. 예를 들어, 텍스처 뷰 컴포넌트는 루마 (Y) 및 크로마 (Cb 및 Cr) 컴포넌트들을 포함할 수도 있다. 심도 뷰 컴포넌트는 그것의 대응하는 텍스처 뷰 컴포넌트에서 화소들의 상대 심도들을 나타낼 수도 있다. 일 예로서, 심도 뷰 컴포넌트는 루마 값들만을 포함하는 그레이 스케일 이미지로서 간주될 수도 있다. 다시 말해, 심도 뷰 컴포넌트는 임의의 이미지 콘텐트를 전달하지 않지만, 대신 텍스처 뷰 컴포넌트에서의 화소들의 상대 심도들의 측정치를 제공할 수도 있다.

[0061] 예를 들어, 심도 뷰 컴포넌트에서 순수 백색 화소에 대응하는 화소 값은 대응하는 텍스처 뷰 컴포넌트에서의 그것의 대응하는 화소 또는 화소들이 뷰어의 관점으로부터 더 가깝다는 것을 나타낼 수도 있고, 심도 뷰 컴포넌트에서의 순수 흑색 화소에 대응하는 화소 값은 대응하는 텍스처 뷰 컴포넌트에서의 그것의 대응하는 화소 또는 화소들이 뷰어의 관점으로부터 더 멀리 있다는 것을 나타낼 수도 있다. 흑색 및 백색 사이에서의 회색의 다양한 음영들은 상이한 심도 레벨들을 나타낸다. 예를 들어, 심도 뷰 컴포넌트에서 매우 회색 화소는 텍스처 뷰 컴포넌트에서의 그것의 대응하는 화소가 심도 뷰 컴포넌트에서의 약간 회색 화소보다 더 멀리 있다는 것을 나타낸다. 오직 회색 스케일이 화소들의 심도를 식별하는데 필요하기 때문에, 심도 뷰 컴포넌트는 크로마 컴포넌트들을 포함할 필요가 없어서, 심도 뷰 컴포넌트에 대한 컬러 값들이 어느 목적의 역할을 하지 않을 수도 있다.

[0062] 심도를 식별하기 위해 루마 값들 (예컨대, 세기 값들)만을 사용하는 심도 뷰 컴포넌트가 예시 목적으로 제공되고 제한하는 것으로 간주되지 않아야 한다. 일반적으로, 임의의 기법이 텍스처 뷰 컴포넌트에서 화소들의 상대 심도들을 나타내기 위해 이용될 수도 있다.

[0063] MVC에 따라, 텍스처 뷰 컴포넌트들은 동일한 뷰에서의 텍스처 뷰 컴포넌트들로부터 또는 하나 이상의 상이한 뷰들이지만 동일한 액세스 유닛들에서의 텍스처 뷰 컴포넌트들로부터 인터 예측된다. 텍스처 뷰 컴포넌트들은, "비디오 블록들"로서 지칭되는 그리고 H.264 콘택스트에서 "매크로블록들"이라 불리는 비디오 데이터의 블록들 내에 코딩될 수도 있다. 다른 비디오 코딩 표준들, 예컨대 HEVC 표준은, 비디오 블록들을 트리블록들 또는 코딩 유닛들 (CU 들)로서 지칭할 수도 있다.

- [0064] 동일한 시간 인스턴스의 픽처들에서의 비디오 콘텐트가 유사할 수도 있다. 그러나, 비디오 콘텐트는 수평 방향으로 가능하게는 수직 방향으로도 약간 변위될 수도 있다. 예를 들어, 블록이 뷰 S0의 픽처 0에서 (x, y)에 위치되면, 뷰 S1의 픽처 1에서 ($x+x'$, y)에 위치된 블록은 뷰 S0의 픽처 0에서 (x, y)에 위치된 블록과 유사한 비디오 콘텐트를 포함한다. 이 예에서, 뷰 S0의 픽처 0에서 (x, y)에 위치된 블록과 뷰 S1의 픽처 1에서 ($x+x'$, y)에 위치된 블록은 대응하는 블록들로서 간주된다. 일부 예들에서, 뷰 S1의 픽처 1에서 ($x+x'$, y)에 위치된 블록에 대한 디스파리티 벡터는 그것의 대응 블록의 로케이션을 지칭한다. 예를 들어, ($x+x'$, y)에 위치된 블록에 대한 디스파리티 벡터는 ($-x'$, 0)이다.
- [0065] 일부 예들에서, 비디오 인코더 (20) 또는 비디오 디코더 (30)는 제 2 뷰의 픽처에서 대응하는 블록을 식별하기 위해 제 1 뷰의 픽처에서의 블록의 디스파리티 벡터를 이용한다. 비디오 인코더 (20) 또는 비디오 디코더 (30)는 제 1 뷰의 픽처에서의 블록의 후보 모션 예측자들의 리스트에서 후보 모션 벡터 예측자로서 제 2 뷰의 픽처에서의 식별된 대응하는 블록에 대한 모션 정보를 이용할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 또는 비디오 디코더 (30)는 제 1 뷰의 픽처에서의 블록에 대한 모션 벡터 예측자로서 이 후보 모션 예측자 (예를 들어, 제 2 뷰의 픽처에서의 블록에 대한 모션 정보)를 선택한다. 이 예에서, 비디오 인코더 (20) 또는 비디오 디코더 (30)는 소위 머지 모드 또는 어드밴스드 모션 벡터 예측 (AMVP) 모드에 따라 모션 벡터 예측자에 기초하여 제 1 뷰에서의 픽처의 블록에 대한 모션 벡터를 결정하고, 모드를 양자 모두는 이하에서 더 상세히 설명된다. 일부 예에서, 디스파리티 벡터는 제 1 뷰의 현재 잔여 정보를 예측하기 위해 제 2 뷰에서 잔여 블록을 위치시키도록 사용될 수도 있다.
- [0066] 비디오 인코더 (20) 또는 비디오 디코더 (30)는 블록의 디스파리티 벡터를 결정하도록 구성된다. 일부 다른 기법들 (즉, 본 개시물에서 설명된 기법들에 따른 것들 외의 기법들)에서, 비디오 인코더 (20) 또는 비디오 디코더 (30)는 먼저, 텍스처 뷰 컴포넌트에 대한 심도 뷰 컴포넌트를 도출하고, 도출된 심도 뷰 컴포넌트에 기초하여 텍스처 뷰 컴포넌트에서 블록에 대한 디스파리티 벡터를 결정한다. 결과의 디스파리티 벡터는 평활한 시간적 뷰 예측된 (STV) 디스파리티 벡터로서 지칭된다. 또한, 일부 다른 기법들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각의 화소가 동일한 디스파리티 벡터를 할당하는 글로벌 디스파리티 벡터 (global disparity vector; GDV)를 코딩한다.
- [0067] 그러나, 심도 뷰 컴포넌트를 도출하는 것은 복잡할 수도 있다 (즉, 프로세싱 및 시간 집약적). 본 개시물에 설명된 기법들은 비디오 인코더 (20) 및 비디오 디코더 (30)가 심도 뷰 컴포넌트를 먼저 반드시 결정할 필요 없이 텍스처 뷰 컴포넌트에서의 블록에 대한 디스파리티 벡터를 결정하게 한다. 그러나, 비디오 인코더 (20) 및 비디오 디코더 (30)가 먼저 심도 뷰 컴포넌트를 도출하였더라도, 비디오 인코더 (20) 및 비디오 디코더 (30)는 본 개시물에 설명된 기법들을 여전히 이용할 수도 있다. 일부 경우들에서, 본 개시물에 설명된 기법들은 심도 맵 코딩 도구들로서 유용할 수도 있다.
- [0068] 또한, GDV는 실제 디스파리티 벡터의 정확한 측정이 아닐 수도 있고, 따라서 모션 벡터가 모션 벡터 예측자로서 사용되는 블록을 정확하게 지칭하지 않을 수도 있다. 다시 말해, GDV가 지칭하는 블록의 모션 벡터는, 실제 디스파리티 벡터가 지칭하는 블록의 모션 벡터보다 덜 정확한 모션 벡터이다.
- [0069] 도 3은 일 예의 MVC 예측 패턴을 나타내는 개념도이다. 도 3의 예에서, 8 개의 뷰들 (뷰 ID들인 "S0" 내지 "S7"을 가짐)이 예시되고, 12 개의 시간적 로케이션들 ("T0" 내지 "T11")이 각각의 뷰에 대해 예시된다. 다시 말하면, 도 3의 각각의 행은 뷰에 대응하는 한편, 각각의 열은 시간적 로케이션을 나타낸다.
- [0070] MVC가 소위 H.264/AVC 디코더들 또는 HEVC 디코더들에 의해 디코딩 가능한 기본 뷰이고, 스테레오 뷰 쌍이 MVC에 의해 또한 지원될 수 있더라도, MVC의 이점은, 2 보다 많은 뷰들을 3D 비디오 입력으로서 사용하고, 다중 뷰들에 의해 표현된 이 3D 비디오를 디코딩하는 예를 지원할 수 있다는 것이다. MVC 디코더 (예를 들어, H.264/MVC 디코더 또는 HEVC 디코더)를 갖는 클라이언트의 렌더러 (renderer)는 다중 뷰들을 갖는 3D 비디오 콘텐트를 기대할 수도 있다.
- [0071] 도 3에서, 뷰 S0은 기본 뷰로서 간주될 수도 있고, 뷰들 (S1 내지 S7)은 의존적 뷰들로서 간주될 수도 있다. 기본 뷰는 인터-뷰 예측되지 않은 픽처들을 포함한다. 기본 뷰에서의 픽처는 동일한 뷰에서의 다른 픽처들에 대하여 인터 예측될 수 있다. 예를 들어, 뷰 S0에서의 픽처들 중 어느 것도 뷰들 (S1 내지 S7) 중 임의의 뷰에서의 픽처에 대하여 인터 예측될 수 없지만, 뷰 S0에서의 픽처들의 일부는 뷰 S0에서의 다른 픽처들에 대하여 인터 예측될 수 있다.
- [0072] 의존적 뷰는 인터-뷰 예측되는 픽처들을 포함하다. 예를 들어, 뷰들 (S1 내지 S7) 중 각각의 뷰는 다른 뷰

에서의 픽처에 대하여 인터 예측되는 적어도 하나의 픽처를 포함한다. 의존적 뷰에서의 픽처들은 기본 뷰에서의 픽처들에 대하여 인터 예측될 수도 있거나, 또는 다른 의존적 뷰들에서의 픽처들에 대하여 인터 예측될 수도 있다.

[0073] 기본 뷰 및 의존적 뷰들 양자 모두를 포함하는 것은 비디오 디코더들의 상이한 유형들이 픽처들을 디코딩 가능할 수도 있다는 것을 보장한다. 예를 들어, 비디오 디코더의 하나의 기본 유형은 멀티뷰 코딩을 프로세싱하도록 구성될 수도 있다. 비디오 디코더의 유형은 기본 뷰를 디코딩할 수도 있는 한편, 멀티뷰 코딩을 프로세싱하도록 구성되는 비디오 디코더들은 뷰들 (S0 내지 S7) 의 각각을 디코딩할 수 있다.

[0074] 문서들인 m22570 및 m22571은 3D 코딩 (3D-HEVC)을 위한 일부 기법들을 정의하고, 2013년 3월 6일, 패스워드를 이용하여, http://wg11.sc29.org/doc_end_user/documents/98_Geneva/wg11/m22570-v2-m22570-v2.zip; http://wg11.sc29.org/doc_end_user/documents/98_Geneva/wg11/m22571-v2-m22571-v2.zip 으로부터 입수 가능하다. 이들 문서들에서 설명되는 도구들 중 하나는, 의존적 뷰에서의 블록의 모션 파라미터들 (즉, 모션 벡터들 및 레퍼런스 인덱스들) 이 동일한 액세스 유닛의 다른 뷰들에서의 이미 코딩된 모션 파라미터들에 기초하여 (예컨대, AMVP 모드 또는 머지모드에서) 예측되거나 또는 유추되는 인터-뷰 모션 예측이다. 더 상세히 설명되는 바와 같이, m22570 및 m22571 문서들에서 설명된 기법들에서 발생할 수 있는 특정한 제한들 및 문제점들이 있을 수도 있다. 본 개시물에서 설명된 기법들은 m22570 및 m22571 문서들에서 설명된 기법들에서 발생할 수 있는 제한들 및 문제점들 중의 일부를 잠재적으로 극복한다. 3D-HEVC에 대한 최근 레퍼런스 소프트웨어 설명은, 2012년 4월 스웨덴, 스톡홀름, Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, "3D-HEVC Test Model 1," http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/1_Stockholm_wg11/JCT3_V-A1005-v_1.zip에서 이용 가능하다. 3D-HEVC에 대한 최근 레퍼런스 소프트웨어는 https://hevc.hhi.fraunhofer.de/svn_svn_3DVCSoftware/trunk에서 이용 가능하다.

[0075] 도 3의 픽처들은 도 3에서 각각의 행 및 각각의 열의 교차부분에 표시된다. MVC 확장들을 갖는 H.264/AVC 표준은 프레임이란 용어를 사용하여 비디오의 부분을 나타낼 수도 있는 반면, HEVC 표준은 픽처라는 용어를 사용하여 비디오의 부분을 나타낼 수도 있다. 본 개시물은 용어 픽처 및 프레임을 상호교환적으로 사용한다.

[0076] 도 3의 픽처들은 대응 픽처가 인트라 코딩된 것 (다시 말하면, I-픽처) 인지, 또는 한 방향으로 인터 코딩된 것 (다시 말하면, P-픽처) 인지 또는 다수의 방향들로 인터 코딩된 것 (다시 말하면, B-픽처) 인지를 지정하는 글자를 포함하는 음영진 블록을 사용하여 예시된다. 일반적으로, 예측들은 화살표들로 나타내어지며, 가리켜진 (pointed-to) 픽처들은 예측 참조를 위해 가리키는 (pointed-from) 픽처를 이용한다. 예를 들어, 시간적 로케이션 T0에서의 뷰 S2의 P-픽처는 시간적 로케이션 T0에서의 뷰 S0의 I-픽처로부터 예측된다.

[0077] 단일 뷰 비디오 인코딩에서처럼, 멀티뷰 비디오 코딩 비디오 시퀀스의 픽처들은 상이한 시간적 로케이션들에서 픽처들에 대하여 예측하여 인코딩될 수도 있다. 예를 들어, 시간적 로케이션 T1에서의 뷰 S0의 B-픽처는 시간적 로케이션 T0에서 자신으로 향하는 뷰 S0의 I-픽처로부터의 화살표를 가져서, b-픽처가 I-픽처로부터 예측됨을 나타낸다. 그러나, 부가적으로, 멀티뷰 비디오 인코딩의 맥락에서, 픽처들은 인터-뷰 예측될 수도 있다. 다시 말하면, 뷰 컴포넌트 (예컨대, 텍스처 뷰 컴포넌트) 은 참조를 위해 다른 뷰들의 뷰 컴포넌트들을 이용할 수 있다. MVC에서는, 예를 들어, 인터-뷰 예측은 다른 뷰의 뷰 컴포넌트가 인터 예측 참조인 듯이 실현된다. 잠재적 인터-뷰 참조들은 시퀀스 파라미터 세트 (SPS) MVC 확장으로 시그널링되고 레퍼런스 픽처 리스트 구성 프로세스에 의해 변경될 수 있는데, 이 구성 프로세스는 인터 예측 또는 인터-뷰 예측 참조들의 유연한 순서화 (ordering) 를 가능하게 한다.

[0078] 도 3은 인터-뷰 예측의 다양한 예들을 제공한다. 뷰 S1의 픽처들은, 도 3의 예에서, 뷰 S1의 상이한 시간적 로케이션들에 있는 픽처들로부터 예측된 것으로서뿐만 아니라, 동일한 시간적 로케이션들에 있는 뷰들 (S0 및 S2)의 픽처들로부터 인터-뷰 예측된 것으로서 예시된다. 예를 들어, 시간적 로케이션 T1에 있는 뷰 S1의 B-픽처는 시간적 로케이션들 (T0 및 T2)에 있는 뷰 S1의 B-픽처들의 각각 뿐만 아니라 시간적 로케이션 T1에 있는 뷰들 (S0 및 S2)의 b-픽처들로부터 예측된다.

[0079] 도 3의 예에서, 대문자 "B" 및 소문자 "b"는 상이한 코딩 방법론들 보다는 픽처들 간의 상이한 계위적 관계들을 나타내는데 사용된다. 일반적으로, 대문자 "B" 픽처들은 소문자 "b" 프레임들보다 예측 계위에서 상대적으로 더 높다. 도 3은 예측 계위에서의 변화들을 상이한 음영 (shading) 레벨들을 이용하여 도시하는데, 음영 양이 더 큰 (다시 말하면, 상대적으로 더 어두운) 프레임들은 적은 음영을 갖는 (다시 말하면, 상대적으로 더 밝은) 프레임들보다 예측 계위에서 더 상위에 있다. 예를 들어, 도 3의 모든 I-픽처들은 짙은 음영으로 도시되는 반면, P-픽처들은 약간 더 밝은 음영을 가지고, B-픽처들 (및 소문자 b-픽처들) 은 서로 상대적이지만

P-픽처들 및 I-픽처들의 음영보다는 항상 더 밝은 여러 가지 음영 레벨들을 가진다.

[0080] 일반적으로, 예측 계위는, 계위에서 상대적으로 더 상위의 그들 픽처들이 계위에서 상대적으로 더 하위의 픽처들의 디코딩 동안에 레퍼런스 픽처들로서 사용될 수 있도록 예측 계위에서 상대적으로 더 상위의 픽처들이 그 계위에서 상대적으로 더 하위의 픽처들을 디코딩하기 전에 디코딩되어야 한다는 점에서, 뷰 순서 인덱스들에 관련될 수도 있다. 뷰 순서 인덱스는 액세스 유닛에서 뷰 컴포넌트들의 디코딩 순서를 나타내는 인덱스이다.

뷰 순서 인덱스들은 H.264/AVC (MVC 개정안) 의 부록 H에 명시된 바와 같이, 시퀀스 파라미터 세트 (SPS) MVC 확장에서 암시된다. SPS에서, 각각의 인덱스 i 에 대해, 대응하는 $view_id$ 가 시그널링된다. 뷰 컴포넌트들의 디코딩은 뷰 순서 인덱스의 오름 차순을 따를 것이다. 모든 뷰들이 제시된다면, 뷰 순서 인덱스들은 0부터 $num_views_minus_1$ 로의 연속하는 순서로 존재한다.

[0081] 이 방식으로, 레퍼런스 픽처들로서 사용되는 픽처들은 레퍼런스 픽처들에 의존하는 픽처들보다 먼저 디코딩된다. 뷰 순서 인덱스는 액세스 유닛에서 뷰 컴포넌트들의 디코딩 순서를 나타내는 인덱스이다. 각각의 뷰 순서 인덱스 i 에 대해, 대응하는 $view_id$ 가 시그널링된다. 뷰 컴포넌트들의 디코딩은 뷰 순서 인덱스들의 오름 차순을 따른다. 모든 뷰들이 제시된다면, 뷰 순서 인덱스들의 세트는 영부터 뷰들의 전체 수보다는 하나 작은 수까지의 연속 순서의 세트를 포함할 수도 있다.

[0082] 계위의 동일한 레벨들에 있는 특정한 픽처들의 경우, 디코딩 순서는 서로에 대해 중요하지 않을 수도 있다. 예를 들어, 시간적 로케이션 T0에 있는 뷰 S0의 I-픽처는 시간적 로케이션 T0에 있는 뷰 S2의 P-픽처에 대한 레퍼런스 픽처로서 사용될 수도 있으며, 이 P-픽처는 시간적 로케이션 T0에 있는 뷰 S4의 P-픽처에 대한 레퍼런스 픽처로서 사용될 수도 있다. 따라서, 시간적 로케이션 T0에 있는 뷰 S0의 I-픽처는 시간적 로케이션 T0에 있는 뷰 S2의 P-픽처보다 먼저 디코딩되어야 하고, 이 P-픽처는 시간적 로케이션 T0에 있는 뷰 S4의 P-픽처보다 먼저 디코딩되어야 한다. 그러나, 뷰들 (S1 및 S3) 사이에서, 디코딩 순서가 중요하지 않은데, 뷰들 (S1 및 S3)이 예측을 위해 서로 의존하지 않기 때문이고, 대신에 예측 계위에서 더 상위인 다른 뷰들만으로부터 예측된다. 더욱이, 뷰 S1이 뷰들 (S0 및 S2) 뒤에 디코딩되는 한, 뷰 S1은 뷰 S4보다 먼저 디코딩될 수도 있다.

[0083] 이 방식으로, 계위적 순서화가 뷰들 (S0 내지 S7)을 기술하는데 사용될 수도 있다. 본 개시물에서, 표기 "SA > SB"는 뷰 SA가 뷰 SB보다 먼저 디코딩되어야 한다는 것을 의미한다. 이 표기법을 이용하면, 도 2의 예에서, S0 > S2 > S4 > S6 > S7이다. 또한, 도 2의 예에 관하여, S0 > S1, S2 > S1, S2 > S3, S4 > S3, S4 > S5, 그리고 S6 > S5이다. 이 계위적 순서화를 위반하지 않는 뷰들에 대한 임의의 디코딩 순서라도 가능하다. 따라서, 제한들이 계위적 순서화에 기초하는 많은 상이한 디코딩 순서들이 가능하다.

[0084] 일부 예들에서, 도 3은 텍스처 뷰 컴포넌트들을 예시하는 것으로 보일 수도 있다. 이들 예들에서, 소위 멀지 모드 또는 AMVP 모드에서 인터-뷰 모션 예측을 구현하기 위해, 비디오 인코더 (20) 또는 비디오 디코더 (30)는 일부 경우들에서 인터 예측될 현재 블록의 디스패리티 벡터에 의해 지정된 블록에 대한 모션 벡터를 모션 벡터 예측자로서 이용할 수도 있다. 본 개시물에서 설명되는 기법들은 심도 뷰 컴포넌트를 도출하는 것을 반드시 필요로 하는 일 없이, 그리고 글로벌 디스패리티 벡터에 반드시 의존하는 일 없이 현재 블록의 디스패리티 벡터를 결정한다.

[0085] 비디오 시퀀스는 뷰 (예컨대, 도 2 및 도 3에 예시된 뷰들)로부터의 일련의 비디오 픽처들을 일반적으로 포함한다. 픽처들의 그룹 (GOP)은 일반적으로 일련의 하나 이상의 비디오 픽처들을 포함한다. GOP는 신팩스 데이터를 GOP의 헤더, GOP의 하나 이상의 픽처들의 헤더, 또는 GOP에 포함된 픽처들의 수를 기술하는 다른 곳에 포함할 수도 있다. 각각의 픽처는 개별 픽처에 대한 인코딩 모드를 기술하는 픽처 신팩스 데이터를 포함할 수도 있다. 비디오 인코더 (20)는 비디오 데이터를 인코딩하기 위해 개개의 비디오 픽처들 내의 비디오 블록들에 대해 통상 동작한다. 비디오 블록은 H.264 표준에서 정의된 바와 같은 매크로블록, 매크로블록의 파티션, 및 가능하게는 파티션의 서브블록 또는 HEVC 표준에서 정의된 바와 같은 최대 코딩 유닛들 (LCU들), 코딩 유닛들 (CU들), 예측 유닛들 (PU들), 또는 변환 유닛들 (TU들)에 대응할 수도 있다. 비디오 블록들은 고정된 또는 가변하는 사이즈들을 가질 수도 있고, 특정된 코딩 표준에 따라 사이즈를 달리할 수도 있다. 각각의 비디오 픽처는 복수의 슬라이스들을 포함할 수도 있다. 각각의 슬라이스는 복수의 블록들을 포함할 수도 있다.

[0086] 일 예로서, ITU-T H.264 표준은 다양한 블록 사이즈들, 예컨대 루마 (luma) 컴포넌트들을 위한 16 바이 16, 8 바이 8, 또는 4 바이 4, 및 크로마 (chroma) 컴포넌트들을 위한 8x8에서 인트라 예측을 지원할 뿐만 아니라 다양한 블록 사이즈들, 예컨대 루마 컴포넌트들을 위한 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 및 4x4 및 크로마 컴

포넌트들을 위한 대응하는 스케일의 사이즈들에서 인터 예측을 지원한다. 본 개시물에서, "NxN" 및 "N 바이 N"은 수직방향 및 수평방향 치수들의 관점에서 블록의 화소 치수들 (예컨대, 16x16 화소들 또는 16 바이 16 화소들) 을 지칭하는데 교환적으로 사용될 수도 있다. 일반적으로, 16x16 블록은 수직 방향의 16 개 화소들 ($y = 16$) 과 수평 방향의 16 화소들 ($x = 16$) 을 가질 것이다. 비슷하게, NxN 블록은 일반적으로 수직 방향의 N 개 화소들 및 수평 방향의 N 개 화소들을 가지며, 여기서 N은 음이 아닌 정수 값을 나타낸다. 블록에서의 화소들은 행들 및 열들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향에서와 동일한 수의 화소들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, 블록들은 M이 N과 반드시 동일하지는 않은 NxM 화소들을 포함할 수도 있다.

[0087] 블록이 인트라 모드 인코딩되는 (예컨대, 인트라 예측되는) 경우, 블록은 그 블록에 대한 인트라 예측 모드를 기술하는 데이터를 포함할 수도 있다. 다른 예로서, 블록이 인터 모드 인코딩되는 (예컨대, 인터 예측되는) 경우, 블록은 그 블록에 대한 모션 벡터를 정의하는 정보를 포함할 수도 있다. 이 모션 벡터는 동일한 뷰에서의 레퍼런스 픽처를 지칭하거나, 또는 다른 뷰에서의 레퍼런스 픽처를 지칭한다. 블록에 대한 모션 벡터를 정의하는 데이터는, 예를 들어, 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터에 대한 분해능 (예컨대, 1/4 화소 정밀도 또는 1/8 화소 정밀도) 을 기술한다. 또한, 인터 예측되는 경우, 블록은 모션 벡터가 가리키는 레퍼런스 픽처와 같은 레퍼런스 인덱스 정보, 및/또는 모션 벡터에 대한 레퍼런스 픽처 리스트 (예컨대, RefPicList0 또는 RefPicList1) 를 포함할 수도 있다.

[0088] JCT-VC는 HEVC 표준의 개발에 노력하고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로서 지칭되는 비디오 코딩 디바이스의 진화 모델에 기초하고 있다. HM 은, (예컨대, ITU-T H.264/AVC) 에 따른 기존 디바이스들에 비해 비디오 코딩 디바이스들의 몇몇 부가적인 능력을 상정한다. 예를 들어, H.264가 9 개의 인트라 예측 인코딩 모드들을 제공하는 반면, HM 은 33 개 정도의 방향성/각 (angular) 인트라 예측 인코딩 모드를 뿐만 아니라 DC 및 평면 (planar) 모드들을 제공할 수도 있다.

[0089] HM의 작업 모델은 비디오 픽처가 루마 및 크로마 샘플들 양자 모두를 포함하는 트리블록들 또는 최대 코딩 유닛 (largest coding unit; LCU) 들의 시퀀스로 분할될 수도 있다는 것을 설명한다. 트리블록은 H.264 표준의 매크로블록과 유사한 목적을 가진다. 슬라이스는 코딩 순서에서 연속적인 다수의 트리블록들을 포함한다. 비디오 픽처가 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 큐드트리에 따라 코딩 유닛들 (CU 들) 로 분할될 수도 있다. 예를 들어, 큐드트리의 루트 노드인 트리블록은 4 개의 자식 노드들로 분할될 수도 있고, 각각의 자식 노드는 다시 부모 노드가 될 수도 있고 다른 4 개의 자식 노드들로 분할될 수도 있다. 큐드트리의 리프 노드인, 최종의 분할되지 않는 자식 노드는 코딩 노드 (즉, 코딩된 비디오 블록) 를 포함한다. 코딩된 비트스트림에 연관된 신택스 데이터는 트리블록이 분할될 수도 있는 최대 횟수를 정의할 수도 있고, 또한 코딩 노드들의 최소 사이즈를 정의할 수도 있다. 트리블록들은 일부 예들에서 LCU 들로서 지칭된다.

[0090] CU 는 코딩 노드와 그 코딩 노드에 연관된 예측 유닛들 (PU 들) 및 변환 유닛들 (TU 들) 을 포함한다. CU의 사이즈는 코딩 노드의 사이즈에 대응하고 형상이 정사각형일 수도 있다. 일부 예들에서, CU 들의 사이즈는 8x8 화소들로부터 최대 64x64 화소들 또는 그 이상을 갖는 트리블록의 사이즈에 이르기까지의 범위이다. 일부 예들에서, 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함한다. CU 에 연관된 신택스 데이터는, 예를 들어, 하나 이상의 PU 들로의 CU의 파티셔닝을 서술한다. 파티셔닝 모드들은 CU 가 스kip되는지 또는 직접 모드 인코딩되는지, 인트라 예측 모드 인코딩되는지, 또는 인터 예측 모드 인코딩되는지의 사이에서 상이하다. PU 들은, 일부 예들에서, 형상이 정사각형이 아니도록 파티셔닝될 수 있다. CU 에 연관된 신택스 데이터는 또한, 예를 들어, 큐드트리에 따른 하나 이상의 TU 들로의 CU의 파티셔닝을 기술한다. TU 는 형상이 정사각형이거나 또는 정사각형이 아닐 수 있다.

[0091] HEVC 표준은 상이한 CU 들에 대해 상이할 수도 있는, TU 들에 따른 변환들을 허용한다. TU 들은, 항상 그러한 것은 아니지만, 파티셔닝된 LCU 에 대해 정의된 주어진 CU 내의 PU 들의 사이즈에 기초하여 통상 사이즈가 정해진다. TU 들은 통상 PU 들과 동일한 사이즈이거나 또는 그것들보다 작다. 일부 예들에서, CU 에 대응하는 잔여 샘플들은 "잔여 큐드 트리 (residual quad tree)" (RQT) 로서 알려진 큐드트리 구조를 사용하여 작은 유닛들로 세분된다. RQT의 리프 노드들은 변환 유닛들 (TU 들) 로서 지칭될 수 있다. TU 들에 연관된 화소 차이 값들은, 일부 예들에서, 양자화되는 변환 계수들을 생성하도록 변환된다.

[0092] 일반적으로 PU 는 예측 프로세스에 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라 모드 인코딩되는 경우, PU 는 PU 에 대한 인트라 예측 모드를 기술하는 데이터를 포함한다. 다른 예로서, PU 가 인터 모드

인코딩되는 경우, PU 는 PU 에 대한 모션 벡터를 정의하는 데이터를 포함한다. PU 에 대한 모션 벡터를 정의하는 데이터는, 예를 들어, 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터에 대한 분해능 (예컨대, 1/4 화소 정밀도 또는 1/8 화소 정밀도), 모션 벡터가 가리키는 레퍼런스 픽처, 및/또는 모션 벡터에 대한 레퍼런스 픽처 리스트 (예컨대, RefList 0, 또는 RefList 1) 를 기술한다.

[0093] 일반적으로, TU 는 변환 및 양자화 프로세스들을 위해 사용된다. 하나 이상의 PU 들을 갖는 주어진 CU 는 또한 하나 이상의 변환 유닛들 (TU 들) 을 포함할 수도 있다. 예측 다음에, 비디오 인코더 (20) 는 PU 에 대응하는 잔여 값들을 계산할 수도 있다. 그 잔여 값들은 엔트로피 코딩을 위한 직렬화된 (serialized) 변환 계수들을 생성하기 위해 TU 들을 사용하여 변환 계수들로 변환되며, 양자화되고, 스캐닝될 수도 있는 화소 차이 값들을 포함한다. 본 개시물은 통상 용어 "비디오 블록"을 CU의 코딩 노드를 지칭하기 위해 사용한다.

일부 특정 경우들에서, 본 개시물은 또한 용어 "비디오 블록"을 트리블록, 즉, LCU, 또는 코딩 노드와 PU 들 및 TU 들을 포함하는 CU 를 지칭하기 위해 사용할 수도 있다.

[0094] 일 예로서, HM 은 다양한 PU 사이즈들에서 예측을 지원한다. 특정 CU의 사이즈가 2Nx2N 이라고 가정하면, HM 은 2Nx2N 또는 NxN 의 PU 사이즈들에서의 인트라 예측과, 2Nx2N, 2NxN, Nx2N, 또는 NxN 의 대칭적 PU 사이즈들에서의 인터 예측을 지원한다. HM 은 또한 2NxN, 2NxN, nLx2N, 및 nRx2N 의 PU 사이즈들에서의 인터 예측을 위한 비대칭 파티셔닝을 지원한다. 비대칭 파티셔닝 시, CU의 하나의 방향은 구획되지 않는 반면, 다른 방향은 25% 및 75%로 파티셔닝된다. 25% 구획에 대응하는 CU의 부분은 "상부 (Up)", "하부 (Down)", "좌측 (Left)", 또는 "우측 (Right)"의 표시가 뒤따르는 "n" 에 의해 나타내어진다. 따라서, 예를 들어, "2NxN" 는 상단의 2Nx0.5N PU 및 하단의 2Nx1.5N PU 로 수평으로 구획되는 2Nx2N CU 를 지칭한다.

[0095] H.264 표준 또는 HEVC 표준 중 어느 하나에서, 인트라 예측 또는 인터 예측 코딩 다음에, 비디오 인코더 (20) 는, HEVC 에서 또는 H.264 에서의 매크로블록에 대해, CU의 TU 들에 대한 잔여 데이터를 계산한다. PU 들은 공간 도메인 (또한 화소 도메인으로서 지칭됨) 에서의 화소 데이터를 포함하고 TU 들은, 잔여 비디오 데이터에 대한 변환 (예컨대, 이산 코사인 변환 (DCT)), 정수 변환, 웨이브릿 변환, 또는 개념적으로 유사한 변환의 적용에 뒤따르는 변환 도메인에서의 계수들을 포함한다. 잔여 데이터는 인코딩되지 않은 픽처의 화소들 및 HEVC 에서의 PU 들에 대응하는 예측 값들 또는 H.264 에서의 매크로블록에 대한 예측 값들 사이의 화소 차이들에 대응할 수도 있다.

[0096] 변환 계수들을 생성하는 임의의 변환들 다음에, 비디오 인코더 (20) 는, 일부 예들에서, 변환 계수들의 양자화를 수행한다. 양자화는 변환 계수들이 그 계수들을 나타내는데 사용된 데이터의 양을 가능한 한 줄이도록 양자화되어서, 추가의 압축을 제공하는 프로세스를 일반적으로 말한다. 양자화 프로세스는 그 계수들의 일부 또는 전부에 연관된 비트 심도를 감소시킨다. 예를 들어, n-비트 값은 양자화 동안에 m-비트 값으로 버림되며 (rounded down), 여기서 n 은 m보다 크다.

[0097] 일부 예들에서, 비디오 인코더 (20) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해 미리 정의된 스캔 순서를 이용하여 양자화된 변환 계수들을 스캔한다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캔을 수행한다. 1차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캔한 후, 일부 예들에서, 비디오 인코더 (20) 는, 몇몇 예들로서, 콘텍스트 적응 가변 길이 코딩 (context adaptive variable length coding; CAVLC), 콘텍스트 적응 이진 산술 코딩 (context adaptive binary arithmetic coding; CABAC), 신팩스 기반 콘텍스트 적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 또는 확률 간격 파티셔닝 엔트로피 (Probability Interval Partitioning Entropy; PIPE) 코딩 또는 다른 엔트로피 인코딩 수법에 따라, 1차원 벡터를 엔트로피 인코딩한다. 비디오 인코더 (20) 는 또한 비디오 데이터의 디코딩에서 비디오 디코더 (30) 에 의한 사용을 위해 인코딩된 비디오 데이터에 연관된 신팩스 엘리먼트들을 엔트로피 인코딩한다.

[0098] CABAC 를 수행하기 위해, 비디오 인코더 (20) 는 콘텍스트 모델 내의 콘텍스트를 송신될 심볼에 할당할 수도 있다. 그 콘텍스트는, 예를 들어, 심볼의 이웃 값들이 영이 아닌지의 여부에 관련될 수도 있다. CAVLC 를 수행하기 위해, 비디오 인코더 (20) 는 송신될 심볼에 대한 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드들은 상대적으로 더 짧은 코드들이 더 가능성있는 (probable) 심볼들에 대응하는 반면, 더 긴 코드들은 덜 가능성있는 심볼들에 대응하도록 구성될 수도 있다. 이런 식으로, VLC 의 사용은, 예를 들어, 송신될 각각의 심볼에 대해 동일 길이 코드워드들을 사용하여, 비트 절약을 달성할 수도 있다. 확률 결정은 심볼에 할당된 콘텍스트에 기초할 수도 있다.

[0099] 전술된 바와 같이, MVC 에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제 2 뷰의 레퍼런스 픽처의 레퍼런

스 블록을 참조하여 제 1 뷰의 현재 픽처의 현재 블록을 인터 예측할 수도 있다. 이러한 인터 예측은 인터-뷰 예측으로서 지칭된다. 현재 픽처 및 레퍼런스 픽처의 시간 인스턴스는 개별 뷰들에서 동일할 수도 있다.

이 예에서, 비디오 인코더 (20) 또는 비디오 디코더 (30)는 동일한 액세스 유닛에서의 픽처들 전체에 걸쳐 인터-뷰 예측을 수행하는데, 여기서 동일한 액세스 유닛에서의 픽처들은 동일한 시간 인스턴스에 있다.

[0100] 현재 블록에 대해 인터-뷰 예측을 수행하기 위해, 비디오 인코더 (20) 또는 비디오 디코더 (30)는, 인터-뷰 예측을 위해 사용될 수 있는 픽처들을 포함하는, 인터 예측을 위해 사용될 수 있는 레퍼런스 픽처들을 식별하는 레퍼런스 픽처 리스트들을 구성한다. 인터 예측은 레퍼런스 픽처에서의 레퍼런스 블록에 대하여 현재 픽처에서의 현재 블록을 예측하는 것을 말한다. 인터-뷰 예측은, 인터-뷰 예측에서 레퍼런스 픽처가 현재 픽처의 뷰와는 상이한 뷰 내에 있다는 점에서 인터 예측의 서브세트이다. 따라서, 인터-뷰 예측을 위해, 비디오 인코더 (20) 및 비디오 디코더 (30)는 다른 뷰에서의 레퍼런스 픽처를 구성된 레퍼런스 픽처 리스트들의 하나 또는 양자 모두에 추가한다. 다른 뷰에서의 레퍼런스 픽처는 구성된 레퍼런스 픽처 리스트를 내의 임의의 로케이션에서 식별될 수 있다. 본 개시물에서 사용된 바와 같이, 비디오 인코더 (20) 가 블록에 대해 인터 예측을 수행하는 (예컨대, 인터 예측하는) 경우, 비디오 인코더 (20)는 블록을 인터 예측 인코딩하는 것으로서 간주될 수도 있다. 비디오 디코더 (30) 가 블록에 대해 인터 예측을 수행하는 (예컨대, 인터 예측하는) 경우, 비디오 디코더 (30)는 블록을 인터 예측 디코딩하는 것으로서 간주될 수도 있다.

[0101] 인터 예측에서, 현재 블록에 대한 모션 벡터가 현재 블록을 인터 예측하기 위한 레퍼런스 블록으로서 사용될 블록의 로케이션을 식별하고, 구성된 레퍼런스 픽처 리스트들 중 하나 또는 양자 모두로의 레퍼런스 인덱스는 현재 블록을 인터 예측하기 위한 레퍼런스 블록으로서 사용되는 블록을 포함하는 레퍼런스 픽처를 식별한다. MVC에서는, 모션 벡터들의 적어도 2 개의 유형들이 있다. 시간적 모션 벡터는, 시간적 레퍼런스 픽처가 예측될 블록을 포함하는 픽처와 동일한 뷰 내의 픽처인 경우, 및 시간적 레퍼런스 픽처가 예측될 블록을 포함하는 픽처보다 앞서 또는 나중에 디스플레이되는 경우, 시간적 레퍼런스 픽처를 지칭한다. 디스패리티 모션 벡터는 픽처가 예측될 블록을 포함하는 뷰와는 다른 뷰에서의 레퍼런스 픽처를 지칭한다. 본 개시물에서, 용어 "모션 벡터"는 시간적 모션 벡터나 디스패리티 모션 벡터 중 하나, 또는 단지 시간적 모션 벡터 또는 디스패리티 모션 벡터를 지칭할 수도 있고, 이는 설명의 맥락에서 더 명확해질 것이다.

[0102] 비디오 인코더 (20) 또는 비디오 디코더 (30) 가 시간적 모션 벡터들을 이용하는 경우, 비디오 인코더 (20) 및 비디오 디코더 (30)는 모션 보상 예측 (MCP) 을 구현하는 것으로서 간주된다. 비디오 인코더 (20) 또는 비디오 디코더 (30) 가 디스패리티 모션 벡터들을 이용하는 경우, 비디오 인코더 (20) 및 비디오 디코더 (30)는 디스패리티 보상 예측 (DCP), 또는 인터-뷰 예측을 구현하는 것으로서 간주된다.

[0103] 시그널링될 필요가 있는 비디오 데이터의 양을 감소시키기 위해, 비디오 인코더 (20)는 모든 예에서 예측될 현재 블록에 대한 모션 벡터를 시그널링 할 필요가 없다. 차라리, 비디오 디코더 (30)는 모션 벡터 예측자들에 기초하여 현재 블록에 대한 모션 벡터를 결정한다. 모션 벡터 예측자는 현재 블록 외의 블록에 대한 모션 벡터이다. 현재 블록 외의 블록은 현재 블록에 공간적으로 이웃하는 블록, 현재 블록에 시간적으로 이웃하는 블록, 또는 현재 블록의 뷰 외의 뷰에서 대응하는 블록일 수도 있다.

[0104] 비디오 디코더 (30)는 어드밴스드 모션 벡터 예측 (AMVP) 모드 또는 머지 모드에 따라 현재 블록의 모션 벡터를 결정하도록 모션 벡터 예측자를 이용한다. 머지 모드 및 AMVP 모드 양자에서, 일부 예들에서, 비디오 디코더 (30)는 모션 벡터 예측자들의 후보 리스트를 구성하도록 암시적 기법 (implicit technique) 을 구현한다.

암시적 기법은, 비디오 디코더 (30) 가 모션 벡터 예측자들의 후보 리스트를 구성하기 위한 방식으로 비디오 디코더 (30)에 명령하는 비디오 인코더 (20)로부터의 명령들을 반드시 수신할 필요는 없다는 것을 의미한다.

다른 예들에서, 비디오 인코더 (20) 가 모션 벡터 예측자들의 후보 리스트를 구성하기 위한 방식으로 비디오 디코더 (30)에 지시하는 것이 가능하다. 예시의 목적을 위해, 비디오 디코더 (30) 가 후보 모션 벡터 예측자들의 리스트를 구성하기 위해 암시적 기법을 구현하는 예들로 기법들이 설명된다.

[0105] 예를 들어, 비디오 디코더 (30)는 현재 블록에 하나 이상의 공간적으로 이웃하는 블록들의 모션 벡터를 결정하고, 모션 벡터 예측자들의 후보 리스트에서 후보 모션 벡터 예측자들로서 이들 모션 벡터들을 포함한다. 일부 예들에서, 비디오 디코더 (30)는 현재 블록에 하나 이상의 시간적으로 이웃하는 블록들의 모션 벡터를 결정하고, 모션 벡터 예측자들의 후보 리스트에서 후보 모션 벡터 예측자들로서 이들 모션 벡터들을 포함한다. 시간적으로 이웃하는 블록들은, 현재 블록이 그 픽처에서 포괄하는 픽처에서의 거의 동일한 영역을 포괄하는 현재 블록을 포함하는 픽처와 동일한 뷰에서 픽처에서의 블록들을 지칭한다. 시간적으로 이웃하는 블록들은 함께-위치된 블록들로 지칭될 수도 있다.

- [0106] 일부 예들에서, 비디오 디코더 (30)는 현재 블록의 POC 값 및 시간적으로 이웃하는 블록들의 핵처 순서 카운트 (picture order count; POC) 값에 기초하여 시간적으로 이웃하는 블록들에 대한 모션 벡터를 스케일링한다. POC 값은 핵처들의 디스플레이 또는 출력 순서를 나타낸다.
- [0107] 또한, 일부 예들에서, 비디오 디코더 (30)는 현재 블록의 디스패리티 벡터에 의해 지칭되는 블록의 모션 벡터를 결정한다. 현재 블록의 디스패리티 벡터에 의해 지칭되는 블록은 현재 블록의 뷰와 상이한 뷰 내에 있다. 디스패리티 벡터에 의해 지칭되는 블록은, 대응하는 블록 및 현재 블록이 상이한 뷰들의 관점으로부터 유사한 비디오 콘텐트를 포함하기 때문에 현재 블록에 대응하는 블록으로서 지칭된다. 이들 예들에서, 비디오 디코더 (30)는 후보 모션 벡터 예측자들의 리스트에서 후보 모션 벡터 예측자로서 대응하는 블록에 대한 모션 벡터를 포함한다.
- [0108] 일부 예들에서, 비디오 인코더 (20)는 인덱스 값을 모션 벡터 예측자들의 후보 리스트로 시그널링하고, 비디오 디코더 (30)는 모션 벡터 예측자들의 후보 리스트로의 인덱스 값에 기초한 모션 벡터 예측자들의 후보 리스트에서 식별된 모션 벡터 예측자를 선택한다. 머지 모드에서, 비디오 디코더 (30)는 현재 블록에 대한 모션 벡터로서 선택된 모션 벡터 예측자를 설정한다. 또한, 머지 모드에서, 비디오 디코더 (30)는 선택된 모션 벡터 예측자가 참조하는 레퍼런스 핵처를 이어받는다. 예를 들어, 모션 벡터 예측자는 현재 블록 외의 블록에 대한 모션 벡터이다. 현재 블록 외의 블록에 대한 모션 벡터는 제 1 레퍼런스 핵처 리스트 (RefPicList0) 또는 제 2 레퍼런스 핵처 리스트 (RefPicList1) 중 하나로의 인덱스 값에 의해 식별된 레퍼런스 핵처를 참조한다. 머지 모드에서, 비디오 디코더 (30)는 현재 블록에 대한 레퍼런스 핵처 리스트 안의 레퍼런스 인덱스로서 레퍼런스 핵처 리스트로의 레퍼런스 인덱스를 이어받는다. 일부 예들에서, 레퍼런스 인덱스는 그것이 이어받을 수 있기 전에 핵처 순서 카운트 (POC)에 기초하여 컨버팅되도록 요구할 수도 있다. 이 방식에서, 비디오 디코더 (30)는 머지 모드에서 현재 블록을 인터-예측하는데 사용되는 레퍼런스 핵처 및 현재 블록에 대한 모션 벡터를 결정한다.
- [0109] AMVP 모드에서, 후보 모션 벡터 예측자들의 리스트로의 인덱스 값의 시그널링에 추가하여, 비디오 인코더 (20)는 인덱스 값에 의해 식별된 모션 벡터 예측자와 현재 블록의 실제 모션 벡터 간의 잔여부를 시그널링한다. 비디오 디코더 (30)는 모션 벡터 예측자들의 후보 리스트로의 인덱스 값에 기초하여 모션 벡터 예측자를 선택하고, 현재 블록에 대한 모션 벡터를 결정하기 위해 현재 블록의 실제 모션 벡터 및 인덱스 값에 의해 식별된 모션 벡터 예측자 간의 시그널링된 잔여부를 가산 또는 감산한다.
- [0110] 일부 예들에서, AMVP 모드에서, 비디오 인코더 (20)는 또한, 비디오 디코더 (30)가 현재 블록을 인터-예측하기 위해 사용하는 레퍼런스 핵처를 식별하는 RefPicList0 또는 RefPicList1로 레퍼런스 인덱스를 시그널링한다. 다시 말해, 머지 모드와 달리, AMVP 모드에서는 비디오 디코더 (30)가 레퍼런스 인덱스를 이어받지 않고 차라리 레퍼런스 인덱스를 수신한다.
- [0111] 도 4는 모션 벡터들이 후보 모션 벡터 예측자들일 수도 있는 공간적으로 이웃하는 블록들을 나타내는 개념도이다. 도 4에서, 인터-예측될 현재 블록은 예측 유닛 (PU)(34)이다. 도 4는 PU (34)에 공간적으로 이웃하는 블록들 (A0, A1, B0, B1, 및 B2)(즉, PU (34)와 동일한 핵처 내의 블록들)을 나타낸다. 일부 예들에서, 비디오 디코더 (30)는 (A0, A1, B0, B1, 및 B2)에 대한 모션 벡터들을 결정하고, 이용 가능하다면 PU (34)에 대한 후보 모션 벡터의 리스트에서 후보 모션 벡터 예측자들로서, 가능한 시그널링 후에 이들 모션 벡터들 중 하나 이상을 포함한다.
- [0112] 블록들 (A0, A1, B0, B1, 및 B2)은 PU (34)에 공간적으로 이웃하는 블록들의 예들이고, PU (34)에 단지 공간적으로 이웃하는 블록들의 예들로서 고려되지 않아야 한다. 예를 들어, PU (34)에 공간적으로 이웃하는 블록들은 블록들 (A0, A1, B0, B1, 및 B2) 외의 로케이션에 위치될 수 있다. 블록들 (A0, A1, B0, B1, 및 B2)의 로케이션은 다음과 같이 정의된다.
- [0113] 도 4에서, 루마 로케이션 (xP, yP)는 PU (34)을 포함하는 현재 핵처의 좌측 상단 샘플에 대해 PU (34)의 좌측 상단 루마 샘플을 특정한다. 따라서, 현재 핵처의 좌측 상단 샘플에 대해 현재 PU "N"의 동일한 좌측 상단 루마는 (xN, yN)이다. 변수들 (nPSPW 및 nPSH)은 루마에 대한 PU (34)의 폭 및 높이를 나타낸다. 이 예에서, N이 A0, A1, B0, B1, 또는 B2에 의해 대체되는 (xN, yN)은, 각각 (xP - 1, yP + nPSH), (xP - 1, yP + nPSH - 1), (xP + nPSW, yP - 1), (xP + nPSW - 1, yP - 1) 또는 (xP - 1, yP - 1)로서 정의된다. 이 방식으로, 비디오 디코더 (30)는 공간적으로 이웃하는 블록들의 로케이션을 결정한다. 이것은, PU (34)의 모션 벡터를 결정하기 위해 모션 벡터 후보들의 후보 리스트에 비디오 디코더가 (30)가 포함하는 이들

블록들 중 하나 이상의 가능한 스케일링 후의 모션 벡터들이다.

[0114] 다음은 비디오 디코더 (30) 가 시간적으로 이웃하는 블록들에 대한 모션 벡터를 결정하는 방식을 설명한다. 전술된 바와 같이, 일부 예들에서, 시간적으로 이웃하는 블록들은 인터-예측될 현재 블록을 포함하는 픽처와 동일한 뷰의 픽처에 상주한다. 그러나, 시간적으로 이웃하는 블록들은 인터-예측될 현재 블록을 포함하는 픽처와 동일한 뷰의 픽처에 반드시 상주할 필요가 없다. 일부 예들에서, 시간적으로 이웃하는 블록들은 인터-예측될 현재 블록을 포함하는 픽처의 뷰와 상이한 뷰에서의 픽처에 상주할 수도 있다. 예를 들어, 시간적으로 이웃하는 블록들을 포함하는 픽처는 인터-예측될 현재 블록을 포함하는 픽처에 대한 레퍼런스 픽처 리스트들 중 하나 안의 collocated_ref_idx 인덱스 값에 의해 식별될 수도 있다. collocated_ref_idx 인덱스 값은 인터-예측될 블록을 포함하는 픽처와 동일한 뷰에서의 픽처, 또는 인터-예측될 블록을 포함하는 픽처와 상이한 뷰에서의 픽처를 식별할 수도 있다. 이들 예들 중 어느 하나에서, collocated_ref_idx 인덱스 값에 의해 식별된 픽처는 본 개시물에서 구절이 사용되는 바와 같이, "시간적으로 이웃하는 블록들" 을 포함할 수도 있다. 후보 모션 벡터 예측자들인 시간적으로 이웃하는 블록들에 대한 모션 벡터는 시간적 모션 벡터 예측자들 (temporal motion vector predictors; TMVPs) 로서 지칭된다.

[0115] 시간적으로 이웃하는 블록에 대한 모션 벡터를 결정하기 위해, 비디오 디코더 (30) 는 어느 픽처가 시간적으로 이웃하는 블록을 포함하는지를 결정할 수도 있다. 시간적으로 이웃하는 블록을 포함하는 픽처는 함께-위치된 픽처로서 지칭된다. 인터-예측될 블록 (예를 들어, 현재 블록) 이 현재 픽처의 슬라이스 내에 있으면, (여기서 슬라이스는 B-슬라이스 (예를 들어, RefPicList0, RefPicList1 에서의 레퍼런스 픽처에 대하여 또는 2 개의 레퍼런스 픽처들에 대하여 예측된, 여기서 하나는 RefPicList0 에서 식별되고 다른 하나는 RefPicList1 에서 식별됨)), 비디오 인코더 (20) 는 함께-위치된 픽처가 RefPicList0 또는 RefPicList1 에서 식별되는지 여부를 나타내는 플래그 (예를 들어, collocated from 10 flag) 를 시그널링한다. 예를 들어, 플래그 값이 1 이면, 비디오 디코더 (30) 는 함께-위치된 픽처가 RefPicList0 에서 식별된다고 결정하고, 플래그 값이 0 이면, 비디오 디코더 (30) 는 함께-위치된 픽처가 RefPicList1 에서 식별된다고 결정한다.

[0116] 일부 예들에서, B-슬라이스에 대해, 비디오 디코더 (30) 는 구성된 RefPicList0 및 RefPicList1 로부터 결합된 레퍼런스 픽처 리스트 (RefPicListC) 를 구성한다. 비디오 디코더 (30) 는, 비디오 코더가 레퍼런스 픽처 리스트 변경 신팩스를 시그널링하는 경우 RefPicListC 를 변경한다. 비디오 디코더 (30) 가 RefPicListC 를 구성하는 예들에서, 비디오 인코더 (20) 는 collocated_from_10_flag 를 시그널링하지 않을 수도 있다. 이들 예들에서, 비디오 디코더 (30) 는 함께-위치된 픽처가 RefPicListC 에서 식별된다는 것을 결정한다.

[0117] 인터-예측될 블록이, 슬라이스가 (예를 들어, RefPicList0 에서의 레퍼런스 픽처에 대하여 예측된) P-슬라이스 인, 현재 픽처의 슬라이스 내에 있으면, 비디오 인코더 (20) 는 플래그를 시그널링하지 않을 수도 있다. 이 경우에서, 비디오 디코더 (30) 는, 현재 블록 (즉, 인터-예측될 블록) 이 P-슬라이스 내에 있기 때문에 함께-위치된 픽처가 RefPicList0 에 위치된다고 결정한다.

[0118] 비디오 디코더 (30) 는 또한, 결정된 레퍼런스 픽처 리스트 안의 인덱스를 수신할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 인터-예측될 현재 블록을 포함하는 슬라이스의 슬라이스 헤더에서 collocated_ref_idx 신팩스를 시그널링한다. 전술된 바와 같이, collocated_ref_idx 신팩스는 인덱스 값은 결정된 레퍼런스 픽처 리스트에 제공한다. 비디오 디코더 (30) 는 레퍼런스 픽처 리스트 안의 인덱스 값에 기초하여 시간적으로 이웃하는 블록들을 포함하는 함께-위치된 픽처를 결정한다.

[0119] 비디오 디코더 (30) 는 그 후, 결정된 함께-위치된 픽처에서 함께-위치된 블록을 식별한다. 예를 들어, 비디오 디코더 (30) 는 그 픽처에서의 현재 블록의 로케이션에 기초하여 함께-위치된 픽처에서 함께-위치된 블록의 로케이션을 식별한다. 예를 들어, 현재 블록은 예측 유닛 (PU) 이라고 가정한다. 이 경우, 비디오 디코더 (30) 는 그 픽처에서 PU 의 로케이션에 기초하여 함께-위치된 픽처에서 코딩 유닛 (CU) 의 로케이션을 결정한다. 함께-위치된 픽처에서의 이 CU 는 함께-위치된 블록으로서 지칭된다.

[0120] 일부 예들에서, 비디오 디코더 (30) 는 함께-위치된 블록 내의 우측 하부 PU 에 대한 모션 벡터를 결정한다. 일부 예들에서, 비디오 디코더 (30) 는 함께-위치된 블록의 변경된 중간 포지션을 커버하는 PU 에 대한 모션 벡터를 결정한다. 함께-위치된 블록의 변경된 중간 포지션은 함께-위치된 블록의 센터 포인트로부터 함께-위치된 블록의 하부 우측 코너를 향해 연장되는 블록을 지칭할 수도 있다. 다시 말해, 함께-위치된 블록의 변경된 중간 포지션에 위치된 블록의 상부-좌측 코너는 함께-위치된 블록의 센터일 수도 있고, 함께-위치된 블록의 변경된 중간 포지션에 위치된 블록의 하부-우측 코너는 함께-위치된 블록의 센터에 대해 하방 및 우측 방향으로 연장될 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 현재 블록을 포함하는 픽처 및 함께-위치

된 픽처의 POC 값에 기초하여 이를 식별된 모션 벡터들을 스케일링한다.

[0121] HEVC 표준은 또한, 어느 픽처들이 함께-위치된 픽처들일 수 있는지를 정의한다. 다시 말해, HEVC 표준은 또 한, 시간적으로 이웃하는 블록들로부터 후보 모션 벡터 예측자들을 결정하는데 비디오 디코더 (30) 가 이용될 수도 있는 픽처들을 정의할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 픽처 파라미터 세트 (PPS) 에서 enable_temporal_mvp_flag 를 시그널링한다. 현재 블록을 포함하는 픽처의 시간적 식별 값 (temporal_id) 이 0 이고, 픽처에 대한 신팩스 엘리먼트들이 PPS 를 지정하며 여기서 enable_temporal_mvp_flag 가 0 인 경우, 비디오 디코더 (30) 는 비디오 디코더 (30) 의 디코딩된 픽처 버퍼 (DPB) 에 저장된 모든 레퍼런스 픽처들을 "시간적 모션 벡터 예측에 사용되지 않은" 것으로서 설정할 수도 있다. 이 경우에서, 비디오 디코더 (30) 는 그 후, 디코딩 순서에서 현재 블록을 포함하는 픽처를 뒤따르는 픽처들에 대해 그리고 현재 블록을 포함하는 픽처에 대한 함께-위치된 픽처로서 현재 블록을 포함하는 픽처보다 디코딩 순서에서 더 이른 임의의 픽처를 이용하지 않을 수도 있다.

[0122] 도 5 는 디스패리티 벡터에 기초하여 후보 모션 벡터 예측자를 결정하는 방식을 나타내는 개념도이다. 도 5 에 설명된 예는 반드시 본 개시물에 설명된 기법들을 따르지 않고, 여기서 본 개시물에 설명된 기법들에서 비디오 디코더 (30) 는 디스패리티 벡터를 결정하기 위해 심도 맵을 도출할 필요가 없을 수도 있다. 도 5 는 현재 뷰의 현재 픽처 (36) 를 나타낸다. 현재 픽처 (36) 는 현재 블록 (38) 을 포함하고, 이 블록은 인터-예측되는 블록이다. 현재 블록 (38) 에 대한 후보 모션 벡터 예측자를 결정하기 위해, 도 5 의 예에서 비디오 디코더 (30) 는 현재 블록 (38) 에 대한 디스패리티 벡터를 결정한다.

[0123] 블록 (38) 에 대한 디스패리티 벡터를 결정하기 위해, 비디오 디코더 (30) 는 현재 픽처 (36) 에서 현재 블록 (38) 의 로케이션 (40A) 을 결정한다. 비디오 디코더 (30) 는 심도 맵 (42) 에서 심도 값 (41) 을 결정한다. 예를 들어, 비디오 디코더 (30) 는 현재 블록 (38) 의 로케이션 (40A) 에 위치된 화소의 상대적 심도를 나타내는 심도 값 (41) 및 현재 픽처 (36) 의 심도 맵 (42) 을 도출해야 할 수도 있다.

[0124] 도 5 의 이 예에서, 비디오 디코더 (30) 는 디스패리티 벡터 (46) 를 결정한다. 예를 들어, 비디오 디코더 (30) 는 심도 값 (41) 에 의해 정의된 심도에서 블록 (38) 의 비디오 콘텐트를 뷰어가 지각하도록, 얼마나 많은 현재 블록 (38) 이 수평적으로 변위되어야 하는지를 결정할 수도 있다. 디스패리티 벡터 (46) 는 결정된 수평적 변위의 양을 나타낼 수도 있고, 따라서 심도 값 (41) 에 기초하여 결정될 수도 있다. 디스패리티 벡터 (46) 는 평활한 시간적 뷰 예측된 (smooth temporal view predicted; STV) 디스패리티 벡터의 예이다.

[0125] 예시된 바와 같이, 디스패리티 벡터 (46) 는 레퍼런스 뷰에서의 현재 픽처 (44) 의 레퍼런스 샘플 블록 (48) 을 지칭한다. 레퍼런스 샘플 블록 (48) 은 현재 블록 (38) 의 비디오 콘텐트와 유사한 비디오 콘텐트를 포함할 수도 있고, 레퍼런스 샘플 블록 (48) 의 로케이션은 현재 뷰의 현재 픽처 (36) 에서의 현재 블록 (38) 의 로케이션에 대하여 레퍼런스 뷰의 현재 픽처 (44) 에서 수평적으로 변위된다. 레퍼런스 블록 (48) 의 모션 벡터 (50) 는 레퍼런스 뷰의 레퍼런스 픽처 (54) 의 레퍼런스 블록 (52) 을 지칭한다.

[0126] 모션 벡터 (50) 는 현재 블록 (38) 의 모션 벡터에 대한 모션 벡터 예측자의 예이다. 예를 들어, 비디오 디코더 (30) 는 현재 블록 (38) 에 대한 모션 벡터 예측자들의 후보 리스트에 모션 벡터 (50) 를 포함시킨다.

[0127] 일부 예들에서, 비디오 디코더 (30) 는 또한, 현재 블록 (38) 에 대한 후보 모션 벡터 예측자로서 디스패리티 벡터 (46) 를 포함한다. 예를 들어, 도 5 는 레퍼런스 블록 (48) 이 인터-예측된다는 것을 나타낸다. 그러나, 예시의 목적을 위해, 레퍼런스 블록 (48) 이 인트라-예측되고 인터-예측되지 않는다고 가정한다. 이 경우에서, 비디오 디코더 (30) 가 모션 벡터 예측자로서 사용할 수도 있는 레퍼런스 블록 (48) 에 대한 모션 벡터는 존재하지 않는다. 따라서, 이 예에서, 비디오 디코더 (30) 는 모션 벡터 예측자들의 후보 리스트에서 모션 벡터 예측자로서 디스패리티 벡터 (46) 를 포함시킨다.

[0128] 후보 모션 벡터 예측자로서 포함되도록 모션 벡터 (50) 가 충족시켜야 할 필요가 있는 소정의 요건들이 존재할 수도 있다. 일 예로써, 모션 벡터 (50) 가 지칭하는 픽처 (즉, 레퍼런스 뷰에서의 픽처 (54)) 의 시간 인스턴스가 현재 블록 (38) 의 하나의 레퍼런스 픽처, 예컨대 픽처 (60) 의 시간 인스턴스와 동일한 경우에만 (즉, 픽처 (54) 및 픽처 (60) 는 동일한 액세스 유닛에 속함), 비디오 디코더 (30) 는 모션 벡터 (50) 가 모션 벡터 예측자들의 후보 리스트에 포함된다고 결정한다.

[0129] 도 5 에 예시된 예에서, 비디오 디코더 (30) 는 머지 모드를 구현한다. 예를 들어, 비디오 인코더 (20) 는 비디오 디코더 (30) 가 구성한 모션 벡터 예측자들의 후보 리스트로 인덱스를 시그널링한다. 도 5 에서, 모션 벡터 예측자들의 후보 리스트로의 인덱스가 모션 벡터 (50) 를 식별한다고 가정한다. 이 경우에서, 머지

모드에 있어서, 비디오 디코더 (30)는 모션 벡터 (50)를 선택하고, 모션 벡터 (50)를 블록 (38)에 대한 모션 벡터 (즉, 모션 벡터 (56))로서 설정한다.

[0130] 또한, 이 예에서, 비디오 디코더 (30)는 레퍼런스 뷰에서 레퍼런스 픽처 (54)를 식별한 레퍼런스 픽처 리스트로 레퍼런스 인덱스를 이어받을 수도 있다. 이 예에서, 비디오 디코더 (30)가 현재 뷰의 픽처 (36)에 대해 구성한 레퍼런스 픽처 리스트 안의 레퍼런스 인덱스는 현재 뷰에서 레퍼런스 픽처 (60)를 식별할 수도 있다. 현재 뷰에서의 레퍼런스 픽처 (60)는 레퍼런스 블록 (58)을 포함한다. 비디오 디코더 (30)는 레퍼런스 블록 (58)을 이용하여 현재 블록 (38)을 인터-예측한다.

[0131] 도 5에 대하여 설명된 AMVP 모드 및 머지 모드 기법들은 다음과 같이 요약될 수 있다. 예를 들어, AMVP 모드 및 머지 모드 양자 모두는 모션 벡터 예측자들의 후보 리스트에 포함될 인터-뷰 모션 벡터 예측자를 허용한다. 비디오 디코더 (30)는 현재 블록의 중간 샘플의 심도 추정치를 결정하고, 이 결정된 심도 추정치에 기초하여 레퍼런스 뷰에서 레퍼런스 블록 및 디스패리치 벡터를 결정한다.

[0132] AMVP 모드에서, 현재 블록에 대한 RefPicList1 및 RefPicList1 중 하나 또는 양자 모두 안의 레퍼런스 인덱스가 인터-뷰 레퍼런스 픽처를 지칭하면, 비디오 디코더 (30)는 결정된 디스패리티 벡터와 동일한 인터-뷰 모션 벡터 예측자를 설정한다. 또한, 디스패리티 벡터에 의해 지칭된 블록에 대한 RefPicList0 및 RefPicList1 중 하나 또는 양자 모두 안의 레퍼런스 인덱스가 시간적 레퍼런스 픽처 (즉, 디스패리티 벡터에 의해 지칭된 블록을 포함하는 픽처와 동일한 뷰에서의 픽처)를 지칭하고, 시간적 레퍼런스 픽처의 시간 인스턴스가 비디오 디코더 (30)가 인터-예측을 위해 사용하는 레퍼런스 픽처의 시간 인스턴스와 동일하면, 비디오 디코더 (30)는 모션 벡터 예측자로서 레퍼런스 블록에 대한 모션 벡터를 이용한다. 모든 다른 경우들에서, 비디오 디코더 (30)는 상이한 뷰로부터 모션 벡터를 이용하지 않을 수도 있다.

[0133] 머지 모드 (또는 스킵 모드)에서, 비디오 디코더 (30)는 픽처 (36)에 대해 구성된 레퍼런스 픽처 리스트에서 첫 번째 2개의 인덱스들을 이용한다. 레퍼런스 인덱스 0에 있어서, 비디오 디코더 (30)는 AMVP 모드에 대하여 전술된 바와 같은 동일한 방식으로 후보 모션 벡터 예측자를 결정한다 (즉, 레퍼런스 인덱스 0에서 식별된 픽처는 레퍼런스 픽처 (44)이다). 비디오 디코더 (30)가 레퍼런스 인덱스 0에서 식별된 픽처에 대한 후보 모션 벡터 예측자를 결정하면, 비디오 디코더 (30)는 후보 모션 벡터로서 모션 벡터 (50)를 포함하고, 모션 벡터 (50)의 레퍼런스 인덱스 (즉, 픽처 (54)를 식별하는 레퍼런스 인덱스)를 이어받는다.

[0134] 비디오 디코더 (30)가 레퍼런스 인덱스 0으로 식별된 픽처를 갖는 후보 모션 벡터 예측자를 결정하지 않으면, 비디오 디코더 (30)는 레퍼런스 인덱스 1로 식별된 픽처에 대해 머지 모드에서 상기 스텝들을 반복한다. 비디오 디코더 (30)가 레퍼런스 인덱스 1로 식별된 픽처를 갖는 후보 모션 벡터 예측자를 결정하지 않으면, 비디오 디코더 (30)는 모션 벡터 예측자들의 후보 리스트에 레퍼런스 뷰로부터의 블록에 대한 모션 벡터를 포함시키지 않는다.

[0135] 일부 예들에서, 비디오 디코더 (30)는 레퍼런스 인덱스 1에 의해 식별된 픽처로 시작하고, 그 후 레퍼런스 인덱스 0에 의해 식별된 픽처로 진행한다. 이것은, 인터-뷰 모션 예측 보다 더 많은 시간 예측을 구현하는 비디오 디코더 (30)를 초래할 수도 있다.

[0136] 도 5에 설명된 예에서, 비디오 디코더 (30)는 먼저, 디스패리티 벡터 (46)를 결정하기 위해 심도 맵 (42)을 도출할 필요가 있다. 비디오 디코더 (30)가 심도 맵 (42)을 도출하는 다양한 방식들이 존재한다. 예로써, 비디오 디코더 (30)는 의존적 뷰 (dependent view)에서의 픽처에 대한 심도 맵을 결정/업데이트하기 위해 기본 뷰에서의 픽처에 대응하는 기본 뷰에서의 심도 맵, 및 기본 뷰에서의 픽처의 시간적 모션 벡터들을 사용한다. 다른 예로써, 비디오 디코더 (30)는 의존적 뷰에서의 픽처에 대한 심도 맵을 결정/업데이트하기 위해 의존적 뷰에서의 픽처들 대상 디스패리티 모션 벡터들, 및 기본 뷰 및 의존적 뷰에서의 픽처들에 대한 시간적 모션 벡터들을 사용한다. 심도 맵을 도출하는 다른 방식들이 존재할 수도 있다.

[0137] 다른 예로써, 코딩 프로세스의 시작에서, 비디오 디코더 (30)는 기본 뷰에서의 제 1 픽처를 디코딩할 수도 있다. 기본 뷰는 다른 뷰에서의 픽처들로부터 예측되지 않는 픽처들을 포함할 수도 있다. 예를 들어, 도 3을 참조하면, 기본 뷰는 S0 일 수도 있는데, 뷰 S0에서의 픽처들이 뷰들 S1-S7 중 어느 하나에서의 픽처들로 인터-예측되지 않기 때문이다. 뷰 S0에서의 제 1 픽처는 인트라-예측된 픽처이다 (즉, I-픽처는 시간 T0에서 뷰 S0에서의 I로 표현됨). 또한, 비-기본 뷰들 (예를 들어, 의존적 뷰들) 중 하나에서의 제 1 픽처는 인터-뷰 모션 예측되거나 인트라-예측될 수도 있다. 비-기본 뷰 (의존적 뷰로서 지정됨)는 (예를 들어, 디스패리티 모션 벡터들을 갖는) 다른 뷰에서의 픽처들에 대하여 인터-예측되는 픽처들을 포함할 수도 있다.

코딩 프로세스의 이 스테이지에서, 심도 맵은 이용 가능하지 않다.

[0138] 비-기본 뷰에서의 제 1 꾹처의 디코딩 후에, 디스패리티 모션 벡터들이 이용 가능하다 (즉, 디스패리티 모션 벡터들은 비-기본 뷰의 제 1 꾹처에서 블록들을 인터-예측하는데 사용됨). 이들 디스패리티 모션 벡터들은 심도 값들로 컨버팅되어 심도 맵을 생성할 수 있다. 심도 맵은 기본 뷰로 맵핑하는데 사용되거나, 비-기본 뷰에서 다음의 꾹처들의 심도 맵들을 업데이트하는데 사용될 수 있다. 심도 맵으로부터, 비디오 디코더 (30)는 블록들에 대한 디스패리티 벡터들을 결정할 수도 있다.

[0139] 일반적으로, 심도 맵을 도출하기 위한 이러한 기법은 복잡할 수도 있고, 비디오 디코더 (30) 가 프로세싱 전력 및 시간을 소모하여 심도 맵을 도출하도록 요구할 수도 있다. 본 개시물에 설명된 예들에서, 비디오 디코더 (30) 는 현재 블록에 대한 디스패리티 벡터를 결정하기 위해 심도 맵을 먼저 도출하지 않을 수도 있다. 대신에, 비디오 디코더 (30) 는 공간적 및/또는 시간적으로 이웃하는 블록들의 디스패리티 모션 벡터들로부터 디스패리티 벡터를 결정할 수도 있는데, 이것은 도출된 심도 맵으로부터 디스패리티 벡터를 결정하는 것보다 덜 복잡한 태스크일 수도 있다.

[0140] 일부 다른 기법들에서, 심도 맵으로부터 디스패리티 벡터를 결정하기 보다는, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 GDV 를 인코딩 또는 디코딩할 수도 있다. GDV 가 모든 화소들에 대해 동일한 디스패리티 벡터를 제공하기 때문에, GDV 는 실제 디스패리티 벡터에 정확한 측정을 제공하지 않고, 따라서 GDV 로부터 결정된 모션 벡터 예측자는 블록의 실제 디스패리티 벡터로부터 결정되는 모션 벡터 예측자만큼 우수하지 않다. 따라서, 심도 맵들로부터 그리고 실제 디스패리티 벡터들에 대한 대체물로서 GDV 를 사용하는 것으로부터 디스패리티 벡터들을 결정하는데 있어서 결함이 존재할 수도 있다.

[0141] 일부 예들에서, 본 개시물에 설명된 기법들에 따른, 결정된 디스패리티 벡터는 현재 블록의 실제 디스패리티 벡터와 동일하지 않을 수도 있다. 일 예로써, 현재 블록의 실제 디스패리티 벡터는 x-컴포넌트를 포함하고, y-컴포넌트를 포함하지 않거나 y-컴포넌트에 대해 0 을 포함할 수도 있는데, 이것은 실제 디스패리티 벡터가 현재 블록에 대해 수평적으로만 변위되는 대응하는 블록을 지칭하기 때문이다. 일부 예들에서, 디스패리티 벡터를 결정하는데 사용된 디스패리티 모션 벡터가 x-컴포넌트 및 y-컴포넌트 양자 모두를 포함할 수도 있기 때문에 결정된 디스패리티 벡터는 x-컴포넌트 및 y-컴포넌트 양자 모두를 포함한다. 블록에 대해 결정된 디스패리티 벡터는, 인터-뷰 모션 예측에 사용되는 구성된 디스패리티 벡터이기 때문에 인터-뷰 모션 예측을 위한 디스패리티 벡터 (DVIVMP) 로서 지칭될 수도 있다.

[0142] 현재 블록에 대한 디스패리티 벡터를 결정하기 위해, 비디오 디코더 (30) 는 디스패리티 벡터들의 후보 리스트를 구성한다. 디스패리티 벡터들의 이 후보 리스트는 모션 벡터 예측자들의 후보 리스트와 혼동되지 않아야 한다. 디스패리티 벡터들의 후보 리스트는 현재 블록의 디스패리티 벡터를 결정하는데 잠재적으로 사용될 수 있는 벡터들을 식별한다. 모션 벡터 예측자들의 후보 리스트는 현재 블록의 모션 벡터를 결정하는데 잠재적으로 사용될 수 있는 벡터들을 식별한다.

[0143] 또한, 디스패리티 벡터 및 디스패리티 모션 벡터는 혼동되지 않아야 한다. 제 1 뷰에서 블록의 디스패리티 모션 벡터는 제 2 뷰에서의 블록을 지칭하는 모션 벡터이고, 여기서 제 2 뷰에서의 블록은 제 1 뷰에서 블록을 인터-예측하는데 사용되는 레퍼런스 블록이다. 디스패리티 벡터는 예측되고 있는 블록과 상이한 뷰에서의 블록을 지칭하고, 디스패리티 벡터에 의해 지칭된 블록을 포함하는 꾹처에 대해 예측되고 있는 블록의 변위를 나타낸다. 디스패리티 벡터에 의해 지칭된 블록은 반드시 현재 블록을 인터-예측하는데 사용될 필요는 없지만, 현재 블록에 대한 모션 벡터 예측자들을 결정하는데 사용될 수도 있다. 일부 예들에서, (결정된 디스패리티 벡터의 y-컴포넌트를 0 으로 설정함으로써) 변경된 디스패리티 벡터에 의해 지칭된 블록은 현재 블록을 인터-예측하는데 사용된다. 이들 예들에서, 변경된 디스패리티 벡터는 디스패리티 모션 벡터의 일 예로써 고려된다.

[0144] 다시 말해, 모든 디스패리티 벡터들이 디스패리티 모션 벡터들이 아니고, 디스패리티 벡터들에 의해 지칭된 블록들은 현재 블록을 인터-예측하는데 사용되는 경우, 이들 디스패리티 벡터들은 디스패리티 모션 벡터들로 컨버팅될 수 있다. 디스패리티 모션 벡터들은 인터-예측을 위한 레퍼런스 블록들로서 사용되는 블록들을 지칭하고, 반드시 현재 블록의 변위를 나타내지는 않는다.

[0145] 예를 들어, 비디오 인코더 (20) 는 현재 블록의 뷰와 상이한 뷰에서 꾹처에서의 검색을 수행하고, 현재 블록에 최선으로 매칭하는 상이한 뷰의 꾹처에서 블록을 찾는다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 그 후, 디스패리티 모션 벡터와의 "최선의 매칭" 블록의 로케이션을 식별함으로써 현재 블록을 인터-예측하기 위해

이 "최선의 매칭" 블록을 이용한다.

[0146] 디스패리티 벡터는 반드시, "최선의 매칭" 블록을 지칭하지는 않는다. 차라리, 디스패리티 벡터는 모션 벡터가 모션 벡터 예측자로서 사용될 수 있는 블록을 지칭한다. 그러나, 디스패리티 벡터가 지칭하는 블록이 인터-예측되지 않으면 (예를 들어, 디스패리티 벡터가 지칭하는 블록에 대한 모션 벡터가 존재하지 않음), 디스패리티 벡터는 후보 모션 벡터 예측자일 수도 있다. 일부 예들에서, 디스패리티 벡터가 지칭하는 블록이 인터-예측되더라도, 변경 후의 디스패리티 벡터는 추가의 후보 모션 벡터 예측자일 수도 있다.

[0147] 디스패리티 벡터가 (예를 들어, 모션 벡터 예측자들을 결정하기 위해) 사용되는 방법의 상기 예들은 예시의 목적을 위해서만 제공되며 제한하는 것으로 고려되어서는 안되는 것으로 이해되어야 한다. 다시 말해, 상기 예들은 결정된 디스패리티 벡터가 사용될 수 있는 일부 방식들을 설명하지만, 본 개시물에 설명된 기법들은 그렇게 제한하는 것으로 고려되지 않아야 한다. 일반적으로, 본 개시물에 설명된 기법들은 디스패리티 벡터를 결정하기 위한 예시의 방식들, 예컨대 일 예로써 심도 맵을 도출하는데 반드시 필요하지는 않지만 디스패리티 벡터를 결정하기 위한 기법들을 제공한다.

[0148] 디스패리티 벡터들의 후보 리스트를 구성하기 위해, 비디오 디코더 (30)는 공간적으로 이웃하는 블록들 및/또는 시간적으로 이웃하는 블록들이 디스패리티 모션 벡터로 예측된 인터-뷰 모션 예측되었는지 여부를 결정한다.

비디오 디코더 (30)는 디스패리티 벡터들의 후보 리스트에서 인터-뷰 모션 예측되었거나 암시적 디스패리티 벡터들을 사용하였던 이웃하는 블록들에 대한 디스패리티 모션 벡터들 중 하나 이상을 포함할 수도 있다. 암시적 디스패리티 벡터들은 이하에서 더 상세히 설명된다. 또한, 일부 예들에서, STV 또는 GDV는 또한, 디스패리티 벡터들의 후보 리스트를 구성하는데 사용될 수도 있다. 비디오 디코더 (30)는 복수의 상이한 기법들 중 어느 하나를 이용하여 후보 디스패리티 벡터들의 리스트로부터 하나를 선택하고, 이 선택된 후보 디스패리티 벡터를 현재 블록에 대한 디스패리티 벡터로서 설정할 수도 있다.

[0149] 상기 예에서, 비디오 디코더 (30)는 디스패리티 벡터들의 후보 리스트를 구성하기 위해 블록 대 블록 단위로 공간적으로 이웃하고 시간적으로 이웃하는 블록들의 모션 벡터 정보를 평가하였다. 그러나, 블록 대 블록 단위로 공간적으로 이웃하고 시간적으로 이웃하는 블록들의 모션 벡터 정보를 평가하는 것은, 비디오 디코더 (30)가 공간적 및 시간적으로 이웃하는 블록들의 모션 벡터 정보를 저장하는 방식에서 비효율적일 수도 있다.

더 상세히 설명되는 바와 같이, 일부 예들에서, 비디오 디코더 (30)는 공간적으로 이웃하고 시간적으로 이웃하는 블록들의 모션 벡터 정보가 저장되는 방식을 고려하므로, 비디오 디코더 (30)는 구역 대 구역 단위로 모션 벡터 정보를 평가하고, 여기서 구역은 복수의 이웃하는 블록들을 포함하고 현재 블록보다 크기가 더 클 수도 있다. 그러나, 비디오 디코더 (30)가 블록 대 블록 단위에서 공간적으로 또는 시간적으로 이웃하는 블록들의 모션 벡터 정보를 저장하면, 구역은 하나의 블록을 포함할 수도 있다. 다시 말해, 본 개시물에서, 구역은 하나 이상의 블록들을 포함할 수도 있다.

[0150] 디스패리티 모션 벡터들의 구역 대 구역 단위의 평가에 기초하여, 비디오 디코더 (30)는 후보 디스패리티 벡터들의 리스트를 구성하고, 후보 디스패리티 벡터들 중 하나를 선택하며, 선택된 후보 디스패리티 벡터에 기초하여 현재 블록에 대한 디스패리티 벡터를 결정한다. 비디오 디코더 (30)는 결정된 디스패리티 벡터로 치정된 블록을 이용하는 AMVP 모드 또는 머지 모드를 구현한다. 일부 예들에서, 비디오 디코더 (30)는 또한, 결정된 디스패리티 벡터로 지정된 블록을 사용하여 인터뷰 잔여 예측을 구현할 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 결정된 디스패리티 벡터로 지정된 블록을 이용하는 다른 코딩 도구들을 구현할 수도 있다. 일 예로써, 비디오 디코더 (30)는 이웃하는 구역 (예를 들어, 하나 이상의 공간적으로 또는 시간적으로 이웃하는 블록들을 포함하는 구역)의 디스패리티 모션 벡터를 블록에 대한 디스패리티 벡터로서 사용함으로써 블록을 인터-예측 디코딩할 수도 있다. 예를 들어, 비디오 디코더 (30)는 이웃하는 구역의 디스패리티 모션 벡터를 블록의 디스패리티 벡터로서 사용하여 AMVP 모드 또는 머지 모드를 구현할 수도 있다.

[0151] 본 개시물에 설명된 기법들에서, 구역들에 대한 디스패리티 모션 벡터들은 x-컴포넌트 및 y-컴포넌트 양자 모두를 포함한다. 따라서, 일부 예들에서, 결정된 디스패리티 벡터는 x-컴포넌트 및 y-컴포넌트 양자 모두를 포함한다. 일부 예들에서, 비디오 디코더 (30)는 또한, 모션 벡터 예측자들의 후보 리스트에 현재 블록에 대한 후보 모션 벡터 예측자로서 결정된 디스패리티 벡터를 포함시킨다.

[0152] 비디오 디코더 (30)는 결정된 디스패리티 벡터를 모션 벡터 예측자들의 후보 리스트에 포함시키기 전에 결정된 디스패리티 벡터를 변경할 수도 있다. 예를 들어, 비디오 디코더 (30)는 결정된 디스패리티 벡터를 변경하기 위해 결정된 디스패리티 벡터의 y-컴포넌트를 0과 동일하게 설정하고, 변경된 디스패리티 벡터를 모션 벡터 예측자들의 후보 리스트에 포함시킨다. 일부 예들에서, 변경된 디스패리티 벡터는 결정된 디스패리티 벡터

보다 현재 블록의 모션 벡터에 대한 더 좋은 예측자이다. 일부 예들에서, 0과 동일하지 않을 수도 있는 결정된 디스패리티 벡터의 y-컴포넌트는 변화되지 않고 유지되거나, 인터-뷰 모션 예측 및/또는 인터-뷰 잔여 예측에 사용된다.

[0153] 또한, 본 개시물에 설명된 일부 기법들에서, 공간적 및 시간적으로 이웃하는 블록들로부터 디스패리티 벡터를 결정함으로써, 픽처들의 소정 유형들은 디스패리티 벡터를 사용하는 다른 코딩 도구들 또는 인터-뷰 모션 예측 및/또는 인터-뷰 잔여 예측으로 코딩될 수 있다. 예를 들어, 일부 다른 기법들 (즉, 본 개시물에 설명된 기법들을 따르지 않는 기법들)에서, 의존적 뷰는 랜덤 액세스 포인트 (RAP) 픽처들, 예컨대 순간 디코더 리프레시 (instantaneous decoder refresh; IDR) 픽처들 및 클린 랜덤 액세스 (clean random access; CRA) 픽처들을 포함한다. 이들 다른 기법들에서, RAP 픽처들에 대한 심도 맵은 RAP 픽처들이 충분히 디코딩될 때까지 이용 가능하지 않다. 따라서, 이들 다른 기법들에서, RAP 픽처들은 심도 맵이 디스패리티 벡터를 결정하는데 이용 가능하지 않았기 때문에 머지 모드 또는 AMVP 모드를 사용하여 인터-뷰 예측되지 않을 수 있다.

[0154] 그러나, 본 개시물에 설명된 기법들에서, 비디오 디코더 (30)는 심도 맵을 도출하는 것을 필요로 하지 않고 디스패리티 벡터를 결정하도록 구성될 수도 있다. 따라서, 일부 예들에서, 비디오 디코더 (30)는 RAP 픽처들에 대해 머지 모드 및 AMVP 모드를 구현하도록 구성될 수 있고, 여기서 모션 벡터 예측자들의 후보 리스트는 RAP 픽처들의 뷰 외의 뷰들에서의 블록들로부터 모션 벡터들을 포함한다.

[0155] 다음은 픽처들의 예시의 유형들을 설명한다. HEVC에서 네트워크 추상 계층 (NAL) 유닛 유형에 의해 식별될 수 있는 4개의 픽처 유형들이 존재한다. 픽처들의 유형들의 예는 순간 디코더 리프레시 (IDR) 픽처, 클린 랜덤 액세스 (CRA) 픽처, 시간적 계층 액세스 (temporal layer access; TLA) 픽처, 및 IDR, CRA 또는 TLA 픽처가 아닌 코딩된 픽처를 포함하지만 이에 제한되는 않는다.

[0156] H.264/AVC 사양은 IDR 픽처를 정의하였고, HEVC 표준은 H.264/AVC 사양으로부터 IDR 픽처의 정의를 이어받는다. HEVC 표준은 또한, 클린 랜덤 액세스 (CRA), 브로큰 링크 액세스 (broken link access; BLA) 픽처, 및 시간적 계층 액세스 (TLA) 픽처를 정의하였다. 이들 픽처 유형들은 HEVC에서 새로우며, H.264/AVC 사양에서 이용 가능하지 않다.

[0157] CRA 픽처는 비디오 시퀀스의 중간에서 임의의 랜덤 액세스 포인트로부터 시작하는 디코딩을 용이하게 하는 픽처이고, 랜덤 액세스를 위해 IDR 픽처들을 사용하는 것보다 더 효율적일 수도 있다. HEVC에서, 이들 CRA 픽처들로부터 시작하는 비트스트림은 또한, 동조 (conforming) 비트스트림들이다. TLA 픽처는 유효한 시간적 계층 스위칭 포인트들을 나타내는데 사용될 수 있는 픽처 유형이다. 다음은 CRA 및 TLA 픽처들을 더 상세히 설명한다.

[0158] 브로드캐스팅 및 스트리밍과 같은 비디오 애플리케이션들에서, 뷰어들은 최소 지연을 갖는 비디오 내의 파트들을 지정하기 위해 점프하거나 상이한 채널들 간에 스위칭하기를 원할 수도 있다. 이러한 스위칭 또는 점핑을 허용하기 위해, 비디오 인코더 (20)는 비디오 비트스트림에서 규칙적인 인터벌들에 랜덤 액세스 픽처들을 포함할 수도 있다. 랜덤 액세스 픽처는 디코딩이 시작할 수 있는 픽처이다. 예를 들어, 사용자가 채널들을 스위칭하거나 비디오 내의 특정 파트로 점프하는 경우, 비디오는 랜덤 액세스 픽처에서 스위칭하거나 랜덤 액세스 픽처로 점프한다.

[0159] H.264/AVC 및 HEVC에서 지정된 IDR 픽처가 랜덤 액세스를 위해 사용될 수 있다. 그러나, IDR 픽처는 코딩된 비디오 시퀀스를 시작하고, 항상 비디오 디코더 (30)의 디코딩된 픽처 버퍼 (DPB)를 클린하기 때문에, 디코딩 순서에서 IDR 다음의 픽처들은 IDR 픽처 이전에 디코딩된 픽처들을 래퍼런스로서 사용할 수 없다. 예를 들어, DPB는, 비디오 디코더 (30)가 인터-예측 (예를 들어, 인터-예측 디코딩)의 목적을 위해 래퍼런스 픽처들로서 사용한 디코딩된 픽처들을 저장하였다. 비디오 디코더 (30)가 IDR 픽처에서 디코딩을 시작하는 경우 발생하는 것과 같이, 비디오 디코더 (30)가 DPB를 클리어하면, (선두 (leading) 픽처들로서 지정된) 디코딩 순서에서 IDR 픽처 다음의 픽처들이 래퍼런스 픽처들로서 사용될 수 있는 DPB에서의 임의의 픽처가 존재하지 않을 수도 있다.

[0160] 결과적으로, 랜덤 액세스를 위해 IDR 픽처들에 의존하는 비트스트림들은 상당히 낮은 코딩 효율성 (예를 들어, 6%)을 가질 수 있다. 코딩 효율성을 개선하기 위해, HEVC에서 CRA 픽처들은 디코딩 순서에서 CRA 픽처를 뒤따르지만 출력 순서에서 CRA 픽처를 선행하여, CRA 픽처 전에 디코딩된 픽처들을 래퍼런스로서 사용하는 픽처들을 허용한다.

[0161] 도 6은 클린 랜덤 액세스 (CRA) 픽처의 일 예를 나타내는 개념도이다. 예를 들어, 도 6은 그 디스플레이

순서에 의해 식별된 픽처들을 나타낸다. 디스플레이 순서는 픽처 순서 카운트 (picture order count; POC) 값들에 의해 식별된다. 예를 들어, 도 6에서, POC 값 (24)을 갖는 픽처는 CRA 픽처이고, 픽처들의 그룹 (GOP)에 속한다. GOP는 또한, POC 값들 (17-23)을 갖는 픽처들을 포함한다. POC 값들 (17-23)을 갖는 픽처들은 디코딩 순서에서 CRA 픽처 (즉, POC 값 (24)을 갖는 픽처)를 뒤따른다 (즉, POC 값들 (17-23)을 갖는 픽처들은 POC 값 (24)을 갖는 CRA 픽처에 의해 인터-예측되거나, POC 값 (24)을 갖는 CRA 픽처에 의해 인터-예측되었던 픽처들에 의해 인터-예측된다). 그러나, POC 값들 (17-23)을 갖는 픽처들은 출력 순서에서 POC 값 (24)을 갖는 CRA를 선행한다 (즉, POC 값들 (17-23)을 갖는 픽처들은 POC 값 (24)을 갖는 CRA 픽처보다 더 빨리 디스플레이된다).

[0162] 이 예에서, POC 값들 (17-23)을 갖는 픽처들은 POC 값 (24)을 갖는 CRA의 선두 픽처들로서 지정되고, 현재 CRA 픽처 전의 CRA 픽처 (즉, POC 값 (24)을 갖는 픽처) 또는 IDR로부터 비디오 디코더 (30)가 디코딩을 시작하는 경우 정확하게 디코딩될 수 있다. 그러나, 비디오 디코더 (30)는, 비디오 디코더 (30)가 POC 값 (24)을 갖는 CRA 픽처로부터 디코딩을 시작하는 경우 POC 값들 (17-23)을 갖는 픽처들을 적절히 디코딩하지 않을 수도 있다. 이들 경우들에서, 비디오 디코더 (30)는 랜덤 액세스 디코딩 동안 선두 픽처들 (즉, POC 값들 (17-23)을 갖는 픽처들)을 폐기한다.

[0163] 또한, 비디오 디코더 (30)가 디코딩을 시작하는 곳에 의존하여 이용 가능하지 않을 수도 있는 레퍼런스 픽처들로부터 예상 전파를 방지하기 위해, 디코딩 순서 및 출력 순서 양자 모두에서 POC 값 (24)을 갖는 CRA 픽처를 뒤따르는 다음의 GOP에서의 모든 픽처들은 디코딩 순서나 출력 순서에서 POC 값 (24)을 갖는 CRA 픽처를 선행하는 임의의 픽처를 레퍼런스로서 사용하지 않는다. 예를 들어, 전술된 바와 같이, 비디오 디코더 (30)가 POC 값 (24)을 갖는 CRA 픽처로부터 디코딩을 시작하면, 비디오 디코더 (30)는 POC 값들 (17-23)을 갖는 픽처들을 폐기한다. 이 경우에서, POC 값들 (17-23)을 갖는 픽처들이 이용 가능하지 않기 때문에, 디코딩 순서 및 출력 순서 양자 모두에서 POC 값 (24)을 갖는 CRA 픽처를 뒤따르는 픽처들은 POC 값들 (17-23)을 갖는 픽처들 중 어느 하나를 레퍼런스 픽처로서 사용하지 않는다.

[0164] 유사한 랜덤 액세스 기능성들은 복구 포인트 보충 강화 정보 (SEI) 메시지를 갖고 H.264/AVC에서 지원된다. H.264/AVC 표준을 따르는 모든 비디오 디코더들이 반드시 복구 포인트 SEI 메시지를 지원하지는 않는다.

[0165] HEVC에서, CRA 픽처로 시작하는 비트스트림은 동조 비트스트림으로서 고려된다. 전술된 바와 같이, 비트스트림이 CRA 픽처로 시작하는 경우, CRA 픽처의 선두 픽처들은 이용 가능하지 않은 레퍼런스 픽처들을 지정할 수 있고 따라서 정확하게 디코딩되지 않을 수 있다. 그러나, 또한 전술된 바와 같이, HEVC는 시작 CRA 픽처의 선두 픽처들이 출력되지 않고, 따라서 명칭 "클린 랜덤 액세스"를 지정한다.

[0166] 비트스트림 순응 요건의 확립을 위해, HEVC는 비-출력 선두 픽처들의 디코딩을 위해 이용 가능하지 않은 레퍼런스 픽처들을 생성하도록 디코딩 프로세스를 지정한다. 그러나, 비디오 디코더 (30)가 디코딩 프로세스가 비트스트림의 시작으로부터 수행되는 경우에 비교하여 동일한 출력을 생성할 수 있는 한, 비디오 디코더 (30)는 반드시 그 디코딩 프로세스를 따르지는 않는다. 더욱이, HEVC에서, 동조 비트스트림은 IDR 픽처들을 전혀 포함하지 않을 수도 있고, 결과적으로 불완전한 코딩된 비디오 시퀀스 또는 코딩된 비디오 시퀀스의 서브세트를 포함할 수도 있다.

[0167] 전술된 바와 같이, 일부 예들에서, 비디오 디코더 (30)는 현재 블록에 대한 디스패리티 벡터를 결정하기 위해 공간적 및 시간적으로 이웃하는 블록들의 모션 벡터들을 평가한다. 다음의 예들은, 비디오 디코더 (30)가 공간적 및 시간적으로 이웃하는 블록들의 모션 벡터들로부터 현재 블록에 대한 디스패리티 벡터를 결정하는 예시의 방식들을 나타낸다. 다음의 예들은 제한하는 것으로 고려되지 않아야 하고, 이 기법들은 현재 블록에 대한 디스패리티 벡터를 결정하기 위한 다른 가능한 기법들로 확장 가능하다.

[0168] 타겟 레퍼런스 뷰가 주어지면, 이 타겟 레퍼런스 뷰로부터 인터-뷰 모션 예측을 가능하게 하기 위해 비디오 디코더 (30)는 공간 디스패리티 벡터들, 시간 디스패리티 벡터들, 및/또는 암시적 디스패리티 벡터들을 이용하여 인터-뷰 잔여 예측 및/또는 인터-뷰 모션 예측을 위한 디스패리티 벡터 (DVIVMP)를 생성할 수도 있다. 공간 디스패리티 벡터들, 시간 디스패리티 벡터들, 및 암시적 디스패리티 벡터들은 모두 이하에서 더 상세히 설명된다. 일부 예들에서, 디스패리티 벡터는 주어진 타겟 레퍼런스 뷰에 의해 제한되지 않는데, 이것은 디스패리티 벡터가 이용 가능한 것으로서 고려되는 임의의 의존적 뷰에서의 픽처에 대응하고, 결정된 디스패리티 벡터로서 직접적으로 사용될 수 있다는 것을 의미한다. 일부 예들에서, 이 디스패리티 벡터는 동일한 액세스 유닛의 타겟 레퍼런스 뷰에서 이미 코딩된 모션 파라미터들에 기초하여 현재 뷰에서의 블록의 모션 파라미터들 (즉, 모션 벡터 예측자들)을 예측 또는 추론하는데 사용된다. 본 개시물의 기법들에 따르면, 비디오 디코

더 (30)는 결정된 암시적 디스패리티 벡터들 (IDV), 공간 디스패리티 벡터들 (SDV), 및 시간 디스패리티 벡터들 (TDV)에 기초하여 후보 디스패리티 벡터들의 리스트를 구성할 수도 있다. 결정된 후보 디스패리티 벡터들 중 하나 이상은, 예를 들어 머지 또는 AMVP에 대한 후보 리스트에 포함될 수도 있다.

[0169] 도 7은 후보 디스패리티 벡터를 결정하는 일 예를 나타내는 개념도이다. 도 7에서, 픽처 (62)는 제 1 뷰에 있고, 머지 모드 또는 AMVP 모드를 사용하여 인터-예측되는 블록 (64)을 포함한다. 블록 (64)에 대한 디스패리티 벡터를 결정하기 위해, 비디오 디코더 (30)는 공간적으로 이웃하는 블록들에 대한 모션 벡터들을 결정한다. 비디오 디코더 (30)는 공간적으로 이웃하는 블록들을 임의의 순서로 평가할 수도 있다. 각각의 공간적으로 이웃하는 블록에 대해, 비디오 디코더 (30)는 공간적으로 이웃하는 블록이 RefPicList0에서 레퍼런스 픽처를 식별하는 순방향 예측 벡터로 예측되는지 또는 RefPicList1에서 레퍼런스 픽처를 식별하는 역방향 예측 벡터로 예측되는지 여부를 결정한다. 결합된 레퍼런스 픽처 리스트 (즉, RefPicListC)가 사용되면, 비디오 디코더 (30)는 공간적으로 이웃하는 블록에 대한 모션 벡터가 RefPicListC에서 픽처를 식별하는지 여부를 결정한다. 이들 중 어느 하나가 디스패리티 모션 벡터로 인터-예측되는 공간적으로 이웃하는 블록들 이면, 비디오 디코더 (30)는 블록 (64)에 대한 후보 디스패리티 벡터들의 리스트에 디스패리티 모션 벡터를 포함시킨다.

[0170] 예를 들어, 도 7은 각각 블록 (64)에 공간적으로 이웃하는 공간적으로 이웃하는 블록 (66) 및 공간적으로 이웃하는 블록 (68)을 나타낸다 (예를 들어, 이웃하는 블록 (66/68)은 픽처 (62)인, 블록 (64)과 동일한 픽처에 속한다). 이 예에서, 블록 (66) 및 블록 (68)은 각각 4x4 블록들이다. 도 7에서, 블록 (66)은 픽처 (70)의 블록 (72)을 지칭하는 시간 모션 벡터 (78)에 의해 예시된 바와 같이, 픽처 (70)의 블록 (72)으로 인터-예측된다. 픽처 (70)는 픽처 (62)와 동일한 뷰에 있으나, 상이한 시간 인스턴스에 있다.

[0171] 블록 (68)은 픽처 (74)의 블록 (76)에 대하여 인터-뷰 예측된다. 예를 들어, 블록 (68)에 대한 모션 벡터 (80)는 픽처 (74)의 블록 (76)을 지칭하는 디스패리티 모션 벡터이다. 픽처들 (62 및 74)은 동일한 시간 인스턴스에서 상이한 뷰들에 있다. 모션 벡터 (80)는, 픽처 (62)가 위치되는 제 1 뷰와 상이한 제 2 뷰에 픽처 (74)가 위치되기 때문에 디스패리티 모션 벡터이다. 더욱이, 픽처 (62) 및 픽처 (74)의 시간 인스턴스는 동일하다. 따라서, 픽처 (62) 및 픽처 (74)는 동일한 액세스 유닛에 속한다.

[0172] 이 예에서, 비디오 디코더 (30)는 블록 (68)의 모션 벡터 (80) 및 블록 (66)의 모션 벡터 (78)를 평가하고, 모션 벡터 (80)가 디스패리티 모션 벡터라고 결정한다. 비디오 디코더 (30)는 그 후, 디스패리티 벡터들의 후보 리스트에 디스패리티 모션 벡터 (80)를 포함시킨다. 디스패리티 모션 벡터 (80)는 공간 디스패리티 벡터 (SDV)로서 지칭될 수도 있다.

[0173] 그러나, 비디오 디코더 (30)가 블록 (66) 및 블록 (68)의 모션 벡터 정보를 저장하기 때문에, 비디오 디코더 (30)가 블록 (66) 및 블록 (68) 양자 모두의 모션 벡터 정보를 평가할 필요가 없을 수도 있다. 예를 들어, 비디오 디코더 (30)가 블록 (66) 및 블록 (68)을 인터-예측한 후에, 비디오 디코더 (30)는 비디오 디코더 (30)의 라인 버퍼에 블록 (66) 및 블록 (6)에 대한 모션 벡터 정보를 저장한다. 일부 예들에서, 모션 벡터 정보를 라인 버퍼에 저장하기 전에, 비디오 디코더 (30)가 블록 (66) 및 블록 (68) 중 하나에 대한 모션 벡터 정보를 저장하거나, 블록들 (66 및 68)에 대한 모션 벡터 정보로부터 양자의 블록들 (66 및 68)에 대한 모션 벡터 정보를 도출하도록, 비디오 디코더 (30)는 블록 (66) 및 블록 (68)에 대한 모션 벡터 정보를 압축한다. 모션 벡터 정보의 압축은 도 8에 추가로 예시된다. 모션 벡터 정보의 압축은 모든 예에서 요구되지 않는다.

[0174] 도 8은 본 개시물에 설명된 하나 이상의 예들에 따라 모션 벡터 정보가 저장되는 방식을 나타내는 개념도이다. 도 8은 라인 버퍼 (82)를 나타낸다. 라인 버퍼 (82)는 예측될 현재 블록 위에 공간적으로 위치된 블록들에 대한 모션 벡터를 저장한다. 도 8의 블록들 (0-15)은 각각 4x4 블록들이다.

[0175] 모션 벡터 정보는 인터-예측 방향, 레퍼런스 픽처 인덱스, 및 모션 벡터 (MV)를 포함한다. 일부 예들에서, 비디오 디코더 (30)는 블록들 (0-15) 중 각 블록에 대한 모션 벡터 정보를 라인 버퍼 (82)에 저장하지 않는다. 차라리, 비디오 디코더 (30)는 매 4 개의 블록들 중 2 개의 블록에 대한 모션 벡터 정보를 라인 버퍼 (82) 내에 저장한다. 이것은, 하나의 라인에 대한 2:1 모션 벡터 정보 압축을 초래하고, 라인 버퍼 (82)에 저장될 필요가 있는 더 적은 정보를 초래한다.

[0176] 예를 들어, 블록들 (0-3) 중 각 블록에 대한 모션 벡터 정보는 상이할 수도 있다. 이 예에서, 블록 (0) 및 블록 (1)에 대한 모션 벡터 정보가 상이하더라도, 저장하는 경우, 블록 (0) 및 블록 (1)은 동일한 모션 벡터

정보를 공유하고, 비디오 디코더 (30) 는 블록들 (0 및 1) 양자에 대한 모션 벡터 정보의 하나의 세트를 라인 베퍼 (82) 에 저장한다. 도 8 에서, 비디오 디코더 (30) 는 블록 (0) 에 대한 모션 벡터 정보를 블록 (0) 및 블록 (1) 양자에 대한 모션 벡터 정보로서 저장한다. 이 경우에서, 블록 (1)에 대한 모션 벡터 정보는 손실될 수도 있고, 블록 (0)에 대한 모션 벡터 정보는 보존된다.

[0177] 유사하게, 이 예에서, 블록들 (2 및 3)에 대한 모션 벡터들이 상이하더라도, 저장하는 경우, 블록들 (2 및 3)은 동일한 모션 벡터 정보를 공유하고, 비디오 디코더 (30) 는 블록 (2) 및 블록 (3) 양자에 대한 모션 벡터 정보의 하나의 세트를 라인 베퍼에 저장할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 블록 (3)에 대한 모션 벡터 정보를 블록들 (2 및 3) 양자에 대한 모션 벡터로서 저장한다. 이 경우에서, 블록 (2)에 대한 모션 벡터 정보는 손실될 수도 있고, 블록 (3)에 대한 모션 벡터 정보는 보존된다.

[0178] 다시 말해, 라인 베퍼 감축을 위한 모션 벡터 정보 압축을 위해, 비디오 디코더 (30) 는 블록 (0) 및 블록 (1) 양자에 대한 모션 벡터 정보를 라인 베퍼 (82)에 저장하지 않는다. 차라리, 비디오 디코더 (30) 는 블록 (1)의 모션 벡터 정보를 라인 베퍼 (82)에 저장하고, 블록 (0)의 모션 벡터 정보는 손실될 수도 있다. 유사하게, 비디오 디코더 (30)는 블록 (3)의 모션 벡터 정보를 라인 베퍼 (82)에 저장하고, 블록 (2)의 모션 벡터 정보는 손실될 수도 있다.

[0179] 도 8에서, 어두운 블록들은 모션 벡터 정보가 저장되는 블록들을 나타낸다. 화살표는, 모션 벡터 정보가 이제 저장된 모션 벡터 정보에 의해 표현되는 블록을 나타낸다. 예시된 바와 같이, 블록들의 절반에 대한 모션 벡터 정보가 손실되고 블록들의 다른 절반에 대한 모션 벡터 정보에 의해 표현되기 때문에, 블록들 모두에 대한 모션 벡터 정보가 저장되는 경우에 비해 라인 베퍼 (82)에서 단지 절반의 메모리가 요구된다.

[0180] 도 8에 대하여 설명된 기법들은 모션 벡터 정보 압축을 구현하기 위한 하나의 방법이다. 라인 베퍼 감축을 위해 모션 벡터 정보 압축을 구현하기 위한 다른 방법들이 존재할 수도 있고, 본 개시물에 설명된 기법들은 라인 베퍼 감축을 위해 모션 벡터 정보 압축을 구현하기 위한 임의의 특정 방법에 제한되지 않는다.

[0181] 본 개시물에 설명된 기법들에 따르면, 비디오 디코더 (30)는, 라인 베퍼 (82)가 구역 내의 모든 블록들에 대해서가 아니라 블록들의 구역에 대해 모션 정보를 저장하기 때문에 블록들 (0-15)의 각 블록에 대한 모션 벡터 정보를 결정할 필요가 없을 수도 있다. 예를 들어, 구역들 (84A-84H) 각각은 화소들의 8x4 구역을 나타낸다. 예를 들어, 구역 (84A)은 제 1 8x4 구역을 나타내고, 여기서 제 1 8x4 구역은 블록들 (0 및 1)을 포함한다. 이 예에서, 블록 (0)의 모션 벡터 정보는 구역 (84A)의 모션 벡터 정보로서 고려된다. 구역 (84B)은 제 2 8x4 구역을 나타내고, 여기서 제 2 8x4 구역은 블록들 (2 및 3) 등을 포함한다. 이 예에서, 블록 (3)의 모션 벡터 정보는 구역 (84B) 등의 모션 벡터 정보로서 고려된다.

[0182] 또한, 도 8은 8x4 구역들을 예시하였으나, 본 개시물의 양태들은 그렇게 제한되지 않는다. 다른 예들에서, 구역들은 4x8 구역들일 수도 있고, 여기서 라인 베퍼 (82)는 수평적 라인들 보다는 수직적 라인들에 상주하는 블록들에 대한 모션 벡터 정보를 저장한다. 유사하게, 공간적으로 이웃하는 블록들의 4x4 블록 사이즈는 또한, 예시의 목적을 위해 제공되고 제한하는 것으로서 고려되지 않아야 한다.

[0183] 도 8의 예에서, 비디오 디코더 (30)는 구역들 (84A-84H) 중 하나가 현재 블록에 공간적으로 이웃하는 블록들을 포함하는지를 결정할 수도 있다. 그 후, 비디오 디코더 (30)는 구역 내의 공간적으로 이웃하는 블록들 중 각 블록에 대한 모션 벡터 정보를 결정하기 보다는, 현재 블록에 공간적으로 이웃하는 블록들을 포함하는 구역에 대한 모션 벡터 정보를 결정한다. 구역에 대한 모션 벡터가 디스패리티 모션 벡터이면, 비디오 디코더 (30)는 현재 블록에 대한 후보 디스패리티 벡터들의 리스트에 디스패리티 모션 벡터를 포함할 수도 있다. 이 예에서, 구역에 대한 모션 벡터는 그 구역 내의 복수의 블록들에 대한 단일의 모션 벡터를 나타낸다.

[0184] 일부 예들에서, 공간적으로 이웃하는 블록들, RefPicList0 을 지칭하는 하나 및 RefPicList1 을 지칭하는 하나를 포함하는 구역에 대한 2 개의 모션 벡터들이 존재할 수도 있다. 이들 예들에서, RefPicList1 을 지칭하는 공간적으로 이웃하는 블록들을 포함하는 구역 내의 모든 블록들에 대해 하나의 단일 모션 벡터 및 RefPicList0 을 지칭하는 공간적으로 이웃하는 블록들을 포함하는 구역 내의 모든 블록들에 대해 하나의 단일 모션 벡터가 존재한다.

[0185] 예를 들어, 도 7을 다시 참조하면, 도 7의 블록 (66)은 도 8의 블록 (2)에 대응하고, 블록 (68)은 도 8의 블록 (3)에 대응한다고 가정한다. 이 예에서, 라인 베퍼 (82)는 디스패리티 모션 벡터 (80)에 대한 모션 벡터 정보 및 픽처 (74)를 식별하는 레퍼런스 인덱스를 저장하고, 시간적 모션 벡터 (78)에 대한 모션 벡터 정보 및 픽처 (70)를 식별하는 레퍼런스 인덱스를 저장하지 않는다.

[0186]

이 예에서, 비디오 디코더 (30) 는 블록 (66) 에 대한 모션 벡터 정보 및 블록 (68) 에 대한 모션 벡터 정보를 결정할 필요가 없다. 대신에, 비디오 디코더 (30) 는 구역 (84B) 에 대한 모션 벡터 정보를 결정할 수도 있는데, 여기서 구역 (84B) 은 블록 (66) 및 블록 (68) 을 포함한다. 이것은, 비디오 디코더 (30) 가 모션 벡터 정보의 더 적은 결정들을 수행하여 이에 의해 효율적인 비디오 코딩을 촉진시키는 것을 허용한다. 이 경우에서, 구역 (84B) 에 대한 모션 벡터 정보는, 모션 벡터가 디스패리티 모션 벡터 (80) 인 것을 나타낸다. 구역 (84) 에 대한 모션 벡터가 디스패리티 모션 벡터이기 때문에, 비디오 디코더 (30) 는 블록 (64) 에 대한 후보 디스패리티 벡터들의 리스트에 블록 (64) 에 대한 후보 디스패리티 벡터로서 디스패리티 모션 벡터 (80) 를 포함한다. 디스패리티 모션 벡터 (80) 는, 디스패리티 모션 벡터 (80) 가 공간적으로 이웃하는 구역 (84) 에 대한 디스패리티 벡터이기 때문에 공간 디스패리티 벡터 (SDV) 로서 지정될 수도 있다. 또한, 공간적으로 이웃하는 블록들 (예를 들어, 공간적으로 이웃하는 구역) 에 대한 디스패리티 모션 벡터 (80) 가 블록 (64) 에 대한 디스패리티 벡터인 것으로 결정되면 (예를 들어, 그것이 후보 디스패리티 벡터들의 리스트로부터 선택되기 때문에), 비디오 디코더 (30) 는 인터-예측하는 블록 (64) 에 대한 블록 (64) 에 대한 디스패리티 벡터로서 디스패리티 모션 벡터 (80) 를 사용할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 블록 (64) 에 대한 디스패리티 벡터로서 디스패리티 모션 벡터 (80) 를 사용하여 블록 (64) 상에서 AMVP 또는 멀지 모드를 구현할 수도 있다.

[0187]

도 7 및 도 8 의 예에서, 비디오 디코더 (30) 는 현재 블록에 대한 후보 디스패리티 벡터를 결정하기 위해 공간적으로 이웃하는 블록들을 이용하였다. 일부 예들에서, 공간적으로 이웃하는 블록들을 이용하는 대신에 또는 이에 추가하여, 비디오 디코더 (30) 는 현재 블록에 대한 다른 후보 디스패리티 벡터를 결정하기 위해 시간적으로 이웃하는 블록들을 이용한다.

[0188]

도 9 는 후보 디스패리티 벡터를 결정하는 일 예를 나타내는 개념도이다. 도 9 는 인터-예측될 블록 (88) 을 포함하는 뷰 1 에서의 픽처 (86) 를 예시한다. 픽처 (86) 에 대한 3 개의 가능한 레퍼런스 픽처들이 존재한다: 이들 픽처들이 픽처 (86) 와 동일한 뷰에 속하기 때문에 픽처 (96) 및 픽처 (106), 및 픽처 (90) 는 픽처 (86) 와 상이한 뷰 (뷰 0) 에 있지만 동일한 액세스 유닛에 있기 때문에 픽처 (90) (예를 들어, 픽처 (86) 및 픽처 (90) 의 시간 인스턴스는 동일함).

[0189]

이 예에서, 비디오 디코더 (30) 는 레퍼런스 픽처들 (90, 96, 및 106) 에서 3 개의 함께-위치된 블록들 (즉, 블록 (92), 블록 (98), 및 블록 (108)) 에 대한 모션 벡터 정보를 결정한다. 예를 들어, 비디오 디코더 (30) 는, 함께-위치된 블록들 (92, 98, 및 108) 중 어느 하나에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정한다.

[0190]

도 9 에서, 블록 (108) 은 인트라-예측되고, 따라서 디스패리티 모션 벡터로 예측되지 않는다. 블록 (92) 은 모션 벡터 (94) 로 인터-예측된다. 그러나, 모션 벡터 (94) 는, 모션 벡터 (94) 가 픽처 (110) 에서의 블록 (112) 을 지정하고 픽처 (110) 가 픽처 (90) 와 동일한 뷰 (즉, 뷰 0) 에 있기 때문에 시간 모션 벡터이다.

[0191]

픽처 (96) 의 블록 (98) 은, 모션 벡터 (100) 에 의해 예시된 바와 같이 픽처 (102) 의 블록 (104) 에 대하여 인터-예측된다. 이 예에서, 픽처 (96) 는 뷰 1 에 있고 픽처 (102) 는 상이한 뷰 (즉, 뷰 0) 에 있다. 따라서, 모션 벡터 (100) 는 디스패리티 모션 벡터이다. 이 예에서, 비디오 디코더 (30) 는 후보 디스패리티 벡터들의 리스트에 후보 디스패리티 벡터로서 디스패리티 모션 벡터 (100) 를 포함한다. 디스패리티 모션 벡터 (100) 는, 디스패리티 모션 벡터 (100) 가 함께-위치된 블록에 대한 모션 벡터이기 때문에 시간 디스패리티 벡터 (TDV) 로서 지정된다.

[0192]

도 9 는 블록 (88) 과 함께-위치되는 함께-위치된 블록들의 일부 예들을 나타낸다. 함께-위치된 블록들의 다른 예들이 존재할 수도 있다. 예를 들어, 현재 블록에 함께-위치된 블록의 일 예는, 현재 블록이 그 현재 픽처에서 위치되는 동일한 구역에 위치되는 함께-위치된 픽처에서의 블록이다. 이 구역은 함께-위치된 구역 (CR) 로서 지정된다. 다시 말해, 함께-위치된 블록의 일 예는 함께-위치된 픽처의 CR 에서의 블록이다.

[0193]

함께-위치된 블록의 다른 예는 함께-위치된 구역을 커버하는 최대 코딩 유닛 (LCU) 이다. 예를 들어, LCU 는 현재 블록보다 사이즈가 더 클 수도 있고, 현재 픽처에서의 현재 블록에 의해 포함되는 영역에 비해 함께-위치된 픽처에서 더 많은 영역을 포함할 수도 있다. 이러한 LCU 가 커버하는 구역은 함께-위치된 LCU (CLCU) 로서 지정된다.

[0194]

함께-위치된 블록의 또 다른 예로써, 비디오 디코더 (30) 는 함께-위치된 픽처에서의 함께-위치된 구역의 하부-

우측에 위치된 블록을 식별한다. 함께-위치된 블록의 이 예는 BR로서 지칭되고, 이는 하부-우측 블록에 대한 약어이다. 함께-위치된 블록들의 예들은 도 10에 추가로 예시된다. 함께-위치된 블록의 또 다른 예로써, 함께-위치된 픽처에 위치된 임의의 블록은 후보 디스패리티 벡터를 도출하는데 사용될 수 있다.

[0195] 도 10은 예측될 현재 블록과 함께-위치되는 함께-위치된 블록들의 예들을 나타내는 개념도이다. 도 10은 픽처 (114) 및 픽처 (118)를 예시한다. 일부 예들에서, 픽처 (114) 및 픽처 (118)는 동일한 뷰 및 상이한 액세스 유닛들에 속한다. 일부 예들에서, 픽처 (114) 및 픽처 (118)는 상이한 뷰들 및 동일한 액세스 유닛에 속한다. 어느 하나의 예에서, 픽처 (118)는 픽처 (114)와 함께-위치되는 것으로 고려된다. 대안으로, 픽처 (118)는 제 1 예 (즉, 동일한 뷰 및 상이한 액세스 유닛들에 속함)에 대해서만 픽처 (114)와 함께-위치된 것으로 고려된다.

[0196] 픽처 (114)는 예를 들어, 머지 모드 또는 AMVP 모드 또는 잔여 예측을 사용하여 인터-예측되는 블록 (116)을 포함한다. 머지 모드 또는 AMVP 모드를 구현하기 위해, 비디오 디코더 (30)는 블록 (116)에 대한 후보 디스패리티 벡터들을 결정하고, 여기서 후보 디스패리티 벡터들 중 하나 이상은 인터-뷰 예측되는 함께-위치된 블록들로부터의 디스패리티 모션 벡터들이다.

[0197] 도 10에서, 블록 (120)은 블록 (116)과 함께-위치되는 함께-위치된 블록의 일 예이다. 예를 들어, 블록 (120)은, 블록 (116)이 픽처 (114)에서 커버하는 픽처 (118)에서의 동일한 구역을 커버한다. 블록 (120)은 함께-위치된 구역 (CR) 블록 (예를 들어, 함께-위치된 픽처에서 함께-위치된 구역을 커버하는 블록)으로서 지칭된다.

[0198] 블록 (122)은 블록 (116)과 함께-위치되는 함께-위치된 블록의 다른 예이다. 예를 들어, 블록 (122)은 사이즈가 블록 (116)의 사이즈와 동일하거나 이보다 큰 최대 코딩 유닛 (LCU)이다. 도 10의 예에서, 블록 (122)은 블록 (116)이 픽처 (114)에서 커버하는 영역보다 픽처 (118)에서의 더 많은 영역을 커버한다. 그러나, 블록 (122)은 적어도, 블록 (116)이 픽처 (114)에서 커버하는 픽처 (118)에서의 동일한 영역을 커버한다. 이 경우에서, 블록 (122)은 블록 (116)의 함께-위치된 구역을 커버하는 LCU이고, CLCU 블록으로서 지칭된다.

[0199] 블록 (124)은 블록 (116)과 함께-위치되는 함께-위치된 블록의 또 다른 예이다. 예를 들어, 블록 (124)은 블록 (116)이 상주하는 픽처 (114)에서의 구역과 함께-위치되는 픽처 (118)에서의 함께-위치된 구역의 하부-우측에 위치된다. 블록 (124)은 BR 블록으로서 지칭된다.

[0200] 일부 예들에서, 블록 (116)에 대한 후보 디스패리티 벡터들을 결정하기 위해, 비디오 디코더 (30)는 CR 블록 (120), CLCU 블록 (122), 및 BR 블록 (124) 중 하나 이상에 대한 모션 벡터 정보를 결정한다. 예를 들어, 비디오 디코더 (30)는 CR 블록 (120), CLCU 블록 (122), 및 BR 블록 (124) 내의 4x4 블록들 각각이 인터-뷰 예측되는지를 결정한다. 비디오 디코더 (30)는 블록 (116)에 대한 후보 디스패리티 벡터들의 리스트에 CR 블록 (120), CLCU 블록 (122), 및 BR 블록 (124) 내의 인터-뷰 예측된 4x4 블록들에 대한 디스패리티 모션 벡터들을 포함할 수도 있다.

[0201] 그러나, CR 블록 (120), CLCU 블록 (122), 및 BR 블록 (124) 내의 각각의 4x4 블록에 대한 모션 벡터 정보를 결정하는 것은 비디오 디코더 (30)가 필요한 것보다 더 많이 함께-위치된 블록들에 대한 모션 벡터 정보를 불필요하게 결정할 것을 요구할 수도 있다. 예를 들어, 공간적으로 이웃하는 블록들을 갖는 예와 유사하게, 비디오 디코더 (30)는 함께-위치된 픽처 내의 각각의 4x4 블록에 대한 모션 벡터 정보를 저장하지 않을 수도 있다. 그러나, 일부 예들에서, 비디오 디코더 (30)는 함께-위치된 픽처 내의 4x4 블록에 대한 모션 벡터 정보를 저장하는 것이 가능할 수도 있다.

[0202] 예를 들어, 비디오 디코더 (30)가 함께-위치된 픽처 내의 블록들을 인터- 또는 인트라-예측한 후에, 비디오 디코더 (30)는 함께-위치된 픽처에서 인터-예측된 4x4 블록들의 모션 벡터를 상에 16x 압축을 적용할 수도 있다. 일 예로써, 16x 압축을 적용하기 위해, 비디오 디코더 (30)는 16 개의 4x4 블록들 (수평에서의 4 개의 4x4 블록들 및 수직 방향에서의 4 개의 4x4 블록들)에 기초하여 모션 벡터 정보를 생성하여, 16x16 구역에 대한 모션 벡터를 발생시킬 수도 있다. 예를 들어, 비디오 디코더 (30)는 하나의 16x16 구역의 상부-좌측 4x4 블록의 모션 정보를 선택하여, 모든 16 개의 4x4 블록들을 나타낼 수도 있다. 비디오 디코더 (30)는 16x16 구역에 대한 모션 벡터를 비디오 디코더 (30)의 베피에 저장한다. 이 베피는 라인 베피 (82)와 동일하거나 비디오 디코더 (30)의 상이한 베피일 수도 있다.

[0203] 다시 말해, 함께-위치된 픽처에서의 16x16 구역은 16 개의 4x4 블록들을 포함한다. 이들 16 개의 4x4 블록

들에 대한 모션 벡터 정보는 상이할 수도 있다. 그러나, 메모리 대역폭 감축을 위한 모션 벡터 압축에 있어서, 이들 16 개의 4x4 블록들 각각에 대한 개별의 모션 벡터 정보는 손실되고, 16 개의 4x4 블록들을 포함하는 16x16 구역에 대한 하나의 모션 벡터로 대체될 수도 있다. 16x16 구역에 대한 이 모션 벡터는 16x16 구역 내의 복수의 4x4 블록들에 대한 단일의 모션 벡터를 나타낸다. 이 단일의 모션 벡터는 2 개의 레퍼런스 퍽처 리스트들 중 하나 (RefPicList0 또는 RefPicList1) 를 지칭할 수도 있다. 일부 예들에서, 16x16 구역에 대한 2 개의 모션 벡터들, RefPicList0 를 지칭하는 하나 및 RefPicList1 을 지칭하는 하나가 존재할 수도 있다. 이들 예들에서, RefPicList0 을 지칭하는 16x16 구역 내의 모든 블록들에 대한 하나의 단일 모션 벡터 및 RefPicList1 을 지칭하는 16x16 구역 내의 모든 블록들에 대한 하나의 단일 모션 벡터가 존재한다. 압축이 사용되지 않는 예들에서, 구역은 하나의 4x4 블록을 포함할 수도 있다.

[0204] 더욱이, 16 개의 4x4 블록들을 포함하는 16x16 구역은 예시의 목적으로 제공되며 제한하는 것으로서 고려되지 않아야 한다. 구역은 16x16 과 상이한 사이즈일 수도 있고, 비디오 디코더 (30) 는 16x 압축에 대한 수평 및 수직 방향 양자에서 블록 4 만큼 모션 벡터 필드를 다운샘플링할 수도 있다. 유사하게, 시간적으로 이웃하는 블록들의 4x4 블록 사이즈는 또한, 예시의 목적을 위해 제공되고 제한하는 것으로서 고려되지 않아야 한다.

[0205] 본 개시물에 설명된 기법들에 따르면, 비디오 디코더 (30) 는 함께-위치된 블록들 내의 각각의 4x4 블록에 대한 모션 벡터 정보를 결정할 필요가 없을 수도 있다. 차라리, 비디오 디코더 (30) 는 함께-위치된 블록들 내의 4x4 블록들을 포함하는 16x16 구역에 대한 모션 벡터 정보를 결정할 수도 있다. 예를 들어, 도 10 을 다시 참조하면, 비디오 디코더 (30) 는 CR 블록 (120), CLCU 블록 (122), 또는 BR 블록 (124) 내의 각각의 4x4 블록에 대한 모션 벡터 정보를 결정하여, 4x4 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터-뷰 예측되었는지 여부를 결정할 필요가 없을 수도 있다.

[0206] 이 예에서, 비디오 디코더 (30) 는 CR 블록 (120), CLCU 블록 (122), 또는 BR 블록 (124) 을 커버하는 구역들 또는 16x16 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정한다. CR 블록 (120), CLCU 블록 (122), 또는 BR 블록 (124) 을 커버하는 구역들 또는 16x16 구역에 대한 모션 벡터가 디스패리티 모션 벡터이면, 비디오 디코더 (30) 는 블록 (116) 의 후보 디스패리티 벡터들의 리스트에 후보 디스패리티 벡터로서 디스패리티 모션 벡터를 포함한다. 또한, CR 블록 (120), CLCU 블록 (122), 또는 BR 블록 (124) 을 커버하는 구역들 또는 구역 (예를 들어, 시간적으로 이웃하는 구역) 에 대한 디스패리티 모션 벡터가 블록 (88) 에 대한 디스패리티 벡터인 것으로 결정되면 (예를 들어, 그것은 후보 디스패리티 벡터들의 리스트로부터 선택되기 때문에), 비디오 디코더 (30) 는 블록 (88) 을 인터-예측하는 블록 (88) 에 대한 디스패리티 벡터로서 이 디스패리티 모션 벡터를 사용할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 블록 (88) 에 대한 디스패리티 벡터로서 디스패리티 모션 벡터를 사용하여 블록 (88) 상에서 AMVP 또는 머지 모드를 구현할 수도 있다.

[0207] 도 7 내지 도 10 에 예시된 예들은 비디오 디코더 (30) 가 후보 디스패리티 벡터들의 리스트에 추가되는 후보 디스패리티 벡터를 결정하는 기법들을 설명한다. 예를 들어, 도 7 및 도 8 은 후보 디스패리티 벡터들의 리스트에 공간 디스패리티 벡터 (SDV) 를 포함시키는 것을 설명한다. 도 9 및 도 10 은 후보 디스패리티 벡터들의 리스트에 시간 디스패리티 벡터 (TDV) 를 포함시키는 것을 설명한다. 비디오 디코더 (30) 가 현재 블록에 대한 디스패리티 벡터로서 SDV 또는 TDV 를 선택하면, 비디오 디코더 (30) 는 선택된 SDV 또는 TDV 디스패리티 벡터를 사용하여 현재 블록 상에서 머지 모드 또는 AMVP 모드를 구현하여, 모션 벡터 정보가 현재 블록에 대한 모션 벡터 정보를 결정하는데 사용되는 블록을 식별한다.

[0208] 그러나, 본 개시물에 설명된 기법들은 그렇게 제한되지는 않는다. 일부 예들에서, 비디오 디코더 (30) 는 또한, 암시적 디스패리티 벡터 (IDV) 를 결정하고, 이 IDV 를 후보 디스패리티 벡터들의 리스트에 포함한다. IDV 는 이웃하는 블록이 인터-뷰 모션 예측으로 코딩되는 이웃하는 블록의 디스패리티 벡터이다. 예를 들어, 예측될 현재 블록은 제 1 블록으로서 지칭되고, 제 1 블록에 이웃하는 블록은 제 2 블록으로서 지칭되며, 제 2 블록에 이웃하는 블록은 제 3 블록으로서 지칭된다고 가정한다.

[0209] 이 예에서, 제 2 블록은 제 2 블록과 동일한 뷰에서의 퍽처에 대하여 인터-예측된다. 그러나, 제 2 블록에 대한 모션 벡터는 제 3 블록으로부터 결정된다. 예를 들어, 제 2 블록에 대한 모션 벡터를 결정하기 위해, 비디오 디코더 (30) 는 전술된 유사한 기법들을 사용하여 제 2 블록에 대한 디스패리티 벡터를 결정한다. 비디오 디코더 (30) 는 제 2 블록의 디스패리티 벡터가 제 2 블록에 대한 모션 벡터로서 지칭하는 제 3 블록에 대한 모션 벡터를 사용한다. 이 경우에서, 제 2 블록의 디스패리티 벡터는 암시적 디스패리티 벡터 (IDV) 로서 지칭되고, 비디오 디코더 (30) 는 제 1 블록에 대한 후보 디스패리티 벡터로서 이 IDV 를 포함한다.

- [0210] 다시 말해, 비디오 디코더 (30) 는 현재 블록에 공간적 및 시간적으로 이웃하는 블록들에 기초하여 도 7 내지 도 10 에 대하여 전술된 SDV 및 TDV 를 결정한다. IDV 는, 비디오 디코더 (30) 가 블록을 공간적 및/또는 시간적으로 이웃하는 공간적 및/또는 시간적으로 이웃하는 블록들에 기초하여 IDV 를 결정하고, 이 블록은 현재 블록을 공간적 및/또는 시간적으로 이웃한다는 점에서 SDV 및 TDV 와 상이하다. 예를 들어, 비디오 디코더 (30) 가 제 2 블록의 디스패리티 벡터에 의해 위치된 제 3 블록의 모션 벡터에 기초하여 (제 2 블록으로서 지칭된) 공간적 및/또는 시간적으로 이웃하는 블록에 대한 모션 벡터를 결정하였으면, 비디오 디코더 (30) 는 (제 1 블록으로서 지칭된) 현재 블록에 대한 디스패리티 벡터들의 후보 리스트에 (IDV 로서 지칭된) 디스패리티 벡터를 포함시킨다. IDV 벡터는 도 11 에서 추가로 예시된다.
- [0211] 도 11 은 후보 디스패리티 벡터를 결정하는 일 예를 나타내는 개념도이다. 도 11 은 블록 (128) 을 포함하는 현재 픽처 (126) 를 나타내고, 여기서 블록 (128) 은 예측될 블록이다. 도 11 에서, 블록 (130) 은 블록 (128) 을 이웃하고, 모션 벡터 (138) 에 의해 예시된 바와 같이 픽처 (134) 의 블록 (136) 으로 인터-예측된다. 이 예에서, 비디오 디코더 (30) 는 AMVP 모드 또는 머지 모드를 사용하여 모션 벡터 (138) 를 결정하였고, 여기서 비디오 디코더 (30) 는 블록 (130) 에 대한 후보 모션 벡터 예측자를 구성하였다. 도 11 에서, 비디오 디코더 (30) 는 블록 (130) 에 대한 후보 모션 벡터 예측자를 포함하였고, 여기서 비디오 디코더 (30) 는 블록 (130) 을 공간적으로 이웃하는 공간적으로 이웃하는 블록의 디스패리티 벡터에 기초하여 블록 (130) 에 대한 후보 모션 벡터 예측자를 결정하였다. 다시 말해, 일부 예들에서, 비디오 디코더 (30) 는 블록 (130) 에 대한 후보 모션 벡터 예측자를 결정하기 위해 본 개시물에 설명된 기법들을 구현하였다.
- [0212] 예를 들어, 비디오 디코더 (30) 는, 블록 (130) 을 공간적으로 이웃하는 블록이 블록 (130) 을 포함하는 픽처와 상이한 뷰에서, 동일한 액세스 유닛에서 블록을 지칭하는 디스패리티 모션 벡터에 대하여 인터-뷰 예측된다고 결정할 수도 있다. 비디오 디코더 (30) 는 그 후, 상이한 뷰에서 블록에 대한 모션 벡터를 결정하였을 수도 있다. 상이한 뷰에서 블록에 대한 모션 벡터가 블록 (130) 을 인터-예측하는데 사용되는 레퍼런스 픽처와 동일한 액세스 유닛 (예를 들어, 동일한 시간 인스턴스) 에 있는 픽처를 지칭하면, 비디오 디코더 (30) 는 블록 (130) 을 인터-예측하기 위해 상이한 뷰에서 블록에 대한 모션 벡터 정보를 사용했을 수도 있다.
- [0213] 예를 들어, 도 11 에서 블록 (132) 은 블록 (130) 을 공간적으로 이웃한다. 또한, 도 11 에서, 블록 (132) 은 디스패리티 모션 벡터 (144) 로 나타내어진 바와 같은 픽처 (140) 의 블록 (142) 에 의해 인터-뷰 예측된다. 이 예에서, 픽처 (140) 및 픽처 (126) 는 동일한 액세스 유닛 (예를 들어, 동일한 시간 인스턴스) 에 있지만 상이한 뷰들에 있고; 따라서 모션 벡터 (144) 는 디스패리티 모션 벡터이다. 이 예에서, 비디오 디코더 (30) 는 디스패리티 모션 벡터 (144) 가 블록 (130) 에 대한 후보 디스패리티 벡터라고 결정하였다.
- [0214] 블록 (130) 에 대한 모션 벡터를 결정하기 위해, 비디오 디코더 (30) 는 블록 (142) 의 모션 벡터를 결정한다 (즉, 블록 (130) 에 대해 결정된 디스패리티 벡터로 지칭된 블록, 여기서 블록 (130) 에 대해 결정된 디스패리티 벡터는 디스패리티 모션 벡터 (144) 임). 예시된 바와 같이, 블록 (142) 은 모션 벡터 (150) 에 의해 나타낸 바와 같이, 픽처 (146) 의 블록 (148) 에 대하여 인터-예측된다. 이 예에서, 비디오 디코더 (30) 는 블록 (130) 에 대한 후보 모션 벡터 예측자로서 모션 벡터 (150) 를 포함한다. 예를 들어, 비디오 인코더 (20) 는 블록 (130) 이 픽처 (134) 와 인터-예측되어야 한다고 결정할 수도 있다. 이 예에서, 픽처 (146) 는 픽처 (134) 와 동일한 시간 인스턴스 (예를 들어, 동일한 액세스 유닛) 에 있다. 따라서, 모션 벡터 (150) 는 블록 (130) 에 대한 유효한 모션 벡터 예측자일 수 있다.
- [0215] 도 11 에서, 비디오 디코더 (30) 는, 모션 벡터 (150) 가 블록 (130) 에 대한 모션 벡터 예측자이고 블록 (130) 의 모션 벡터를 모션 벡터 (150) 와 동일하게 설정한다고 결정하였다. 예를 들어, 도 11 은 블록 (130) 의 모션 벡터인 모션 벡터 (138) 를 예시한다. 이 예에서, 비디오 디코더 (30) 는 모션 벡터 (150) 에서 모션 벡터 (138) 로 확장되는 점선에 의해 예시된 바와 같이, 모션 벡터 (150) 에 기초하여 모션 벡터 (138) 를 결정하였다. 예를 들어, 모션 벡터 (138) 는 모션 벡터 (150) 의 모션 벡터 정보를 이어받았다.
- [0216] 이 예에서, 비디오 디코더 (30) 는 결정된 디스패리티 벡터 (144) (즉, 블록 (130) 의 결정된 디스패리티 벡터인 블록 (132) 의 디스패리티 모션 벡터) 에 기초하여 블록 (130) 에 대한 모션 벡터를 결정하였기 때문에, 비디오 디코더 (30) 는 블록 (128) 에 대한 후보 디스패리티 벡터로서 디스패리티 모션 벡터 (144) 를 포함한다. 이 예에서, 디스패리티 모션 벡터 (144) 는 암시적 디스패리티 벡터 (IDV) 로서 지칭된다. 더욱이, 이 예에서, 비디오 디코더 (30) 는 암시적 디스패리티 벡터 (즉, 디스패리티 모션 벡터 (144)) 에 기초하여 블록 (130) 에 대한 모션 벡터를 결정하였기 때문에, 블록 (130) 은 암시적 디스패리티 예측된 (IDP) 블록으로서 지칭된다.
- [0217] 도 11 은 공간적으로 이웃하는 블록들에 기초하여 IDV 의 예를 예시하였으나, 본 개시물에 설명된 기법들은 그

령게 제한되지 않는다는 것에 유의해야 한다. 일부 예들에서, 블록 (130)은 시간적으로 이웃하는 블록일 수도 있고, 블록 (132)은 도 11에 예시된 공간적으로 이웃하는 블록들인 블록 (130) 및 블록 (132) 양자 대신에, 블록 (130)을 이웃하는 공간적 또는 시간적으로 이웃하는 블록일 수도 있다. 따라서, IDV의 일 예는 공간적으로 이웃하는 블록들에 기초하여 결정된 IDV이다. IDV의 다른 예는 시간적으로 이웃하는 블록들에 기초하여 결정된 IDV이다.

[0218] 더욱이, 본 개시물에 설명된 기법들에 따르면, 비디오 디코더 (30)는 현재 블록에 대한 IDV를 결정하는 경우 블록들에 대한 모션 정보가 저장되는 방식을 고려할 수도 있다. 예를 들어, IDV가 공간적으로 이웃하는 블록들에 기초하는 경우, 비디오 디코더 (30)는 블록들의 구역 (예를 들어, 블록들의 8x4 구역)에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정할 수도 있고, 여기서 구역은 인터-예측될 블록을 포함하는 픽처와 동일한 픽처에 복수의 블록들을 포함하고, 블록들 중 적어도 하나에 대한 모션 벡터는 다른 블록들에 대한 모션 벡터와 상이하다. 또한, IDV가 시간적으로 이웃하는 블록들에 기초하는 경우, 비디오 디코더 (30)는 블록들의 구역 (예를 들어, 블록들의 16x16 구역)에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정할 수도 있고, 여기서 구역은 인터-예측될 블록을 포함하는 픽처와 상이한 픽처에서 복수의 블록들을 포함한다.

[0219] 따라서, 비디오 디코더 (30)는, 구역에 대한 모션 벡터가 구역을 이웃하는 블록 (즉, 도 11의 블록 (132)은 도 11의 블록 (130)을 포함하는 블록들의 구역을 이웃함)의 디스패리티 모션 벡터로부터 도출되는 경우, 블록의 구역 (즉, 현재 블록을 이웃하는 구역)이 암시적 디스패리티 예측된 구역이라고 결정할 수도 있다. 구역이 암시적으로 디스패리티 예측된 구역이라고 결정하는 것에 응답하여, 비디오 디코더 (30)는 구역을 이웃하는 블록의 디스패리티 모션 벡터가 현재 블록 (예를 들어, 도 11의 블록 (128))에 대한 디스패리티 벡터인지 여부를 결정할 수도 있다. 예를 들어, 비디오 디코더 (30)는 후보 디스패리티 벡터들의 리스트에 구역을 이웃하는 블록의 디스패리티 모션 벡터를 포함하고, 이 디스패리티 모션 벡터를 선택할 수도 있다. 이 예에서, 비디오 디코더 (30)는, 디스패리티 모션 벡터 (즉, 디스패리티 모션 벡터 (144))가 블록 (128)에 대한 디스패리티 벡터라고 결정되는 경우, 블록 (128)에 대한 디스패리티 벡터로서 디스패리티 모션 벡터 (144)를 사용하여 블록 (128)을 인터-예측 디코딩할 수도 있다. 일부 예들에서, 이웃하는 구역에 대한 디스패리티 모션 벡터가 블록 (128)에 대한 디스패리티 벡터인 것으로 결정되지 않은 경우 (예를 들어, 공간적 또는 시간적 이웃하는 구역에 대한 디스패리티 모션 벡터가 블록 (128)에 대한 디스패리티 벡터인 것으로 결정되지 않거나, 공간적 또는 시간적 이웃하는 구역이 디스패리티 모션 벡터로 인터-예측되지 않음), 비디오 디코더 (30)는 블록 (128)에 대한 디스패리티 벡터로서 디스패리티 모션 벡터 (144)를 사용하여 블록 (128)을 인터-예측 디코딩할 수도 있다.

[0220] 이 방식으로, 비디오 디코더 (30)는, 비디오 디코더 (30)가 현재 블록을 공간적으로 및/또는 시간적으로 이웃하는 블록들에 대한 모션 벡터 정보를 저장하는 방식에 기초하여 예측될 현재 블록에 대한 후보 디스패리티 벡터들의 리스트를 구성할 수도 있다. 후보 디스패리티 벡터들의 이 리스트는 하나 이상의 공간 디스패리티 벡터들 (SDVs), 하나 이상의 시간 디스패리티 벡터들 (TDVs), 및/또는 하나 이상의 암시적 디스패리티 벡터들 (IDVs)을 포함할 수도 있다.

[0221] 비디오 디코더 (30)는 후보 디스패리티 벡터들의 리스트로부터 후보 디스패리티 벡터들 중 하나를 선택할 수도 있다. 본 개시물에 설명된 바와 같이, 현재 블록에 대한 후보 디스패리티 벡터들의 리스트를 결정하고 이 리스트로부터 후보 디스패리티 벡터를 선택하는 것은 STV들 및 GDV들에 대하여 전술된 이슈들을 극복할 수도 있으므로, 비디오 디코더 (30)는 시간 효율적인 방식으로 최소의 복잡성을 갖고 현재 블록에 대한 디스패리티 벡터를 결정할 수 있다.

[0222] 비디오 디코더 (30)는, 어느 후보 디스패리티 벡터가 선택되는지를 결정하는데 이용하는 각종 가능한 기준이 존재할 수도 있다. 일 예로써, 비디오 디코더 (30)는 가장 빈번하게 발생하는 후보 디스패리티 벡터를 선택한다. 다른 예로써, 비디오 디코더 (30)는 제 1 후보 디스패리티 벡터를 선택한다 (예를 들어, 후보 디스패리티 벡터가 발견되자마자, 비디오 디코더 (30)는 후보 디스패리티 벡터들을 찾는 것을 멈춘다). 이 예에서, 비디오 디코더 (30)는 먼저 SDV, 그 후 TDV 다음에 IDV를 찾을 수도 있다; 그러나, SDV, TDV, 및 IDV의 순서는 예시의 목적을 위해 제공되며 제한으로 고려되지 않아야 한다. 예를 들어, 비디오 디코더 (30)가 SDV를 찾으면, SDV는 디스패리티 벡터일 수도 있다. 그러나, SDV가 존재하지 않고 비디오 디코더 (30)가 TDV를 찾으면, TDV는 디스패리티 벡터 등일 수도 있다.

[0223] 다른 예로써, 비디오 디코더 (30)는 후보 디스패리티 벡터들의 x- 및 y-컴포넌트들의 제곱들의 합 및 그 합의 결과의 제곱근을 취하는 비디오 디코더 (30)에 의해 결정된 바와 같이, 최단 후보 디스패리티 벡터를

선택한다. 일반적으로, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터 후보 디스패리티 벡터에서의 인덱스 값을 수신하는 것을 포함하는, 후보 디스패리티 벡터들의 리스트로부터 후보 디스패리티 벡터를 선택하기 위해 임의의 기법을 이용할 수도 있다.

[0224] 비디오 디코더 (30) 는 선택된 후보 디스패리티 벡터에 기초하여 현재 블록의 디스패리티 벡터를 결정할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재 블록의 디스패리티 벡터로서 선택된 후보 디스패리티 벡터를 설정한다. 이 예에서, 현재 블록의 결정된 디스패리티 벡터는, 선택된 후보 디스패리티 벡터가 또한 x-컴포넌트 및 y-컴포넌트 양자를 포함했다면 x-컴포넌트 및 y-컴포넌트 양자를 포함할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 현재 블록의 결정된 디스패리티 벡터에 수직 컴포넌트가 존재하지 않도록, 선택된 후보 디스패리티 벡터의 y-컴포넌트를 0 과 동일하게 설정할 수도 있다.

[0225] 일부 예들에서, 비디오 디코더 (30) 는 현재 블록의 뷰와 상이한 뷰에 있는 픽처에서의 블록을 식별하기 위해 결정된 디스패리티 벡터를 이용한다. 비디오 디코더 (30) 는 현재 블록의 뷰와 상이한 뷰에 있는 픽처에서의 뷰에 대한 모션 벡터를 결정한다. 이 예에서, 비디오 디코더 (30) 는 현재 블록에 대한 후보 모션 벡터 예측자들의 리스트에 후보 모션 벡터 예측자로서 결정된 모션 벡터를 포함한다. 일부 예들에서, 결정된 모션 벡터가 현재 픽처를 인터-예측하는데 사용되는 픽처와 동일한 시간 인스턴스에서의 픽처를 지칭하면, 비디오 디코더 (30) 는 현재 픽처에 대한 후보 모션 벡터 예측자들의 리스트에 후보 모션 벡터 예측자로서 결정된 모션 벡터를 포함한다.

[0226] 비디오 디코더 (30) 는 현재 블록에 대한 모션 벡터 예측자들의 후보 리스트에 기초하여 현재 블록 상에서 머지 모드 또는 AMVP 모드를 구현할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 일 예로써, 비디오 인코더 (20) 로부터 예컨대 모션 벡터 예측자들의 리스트에서의 인덱스 값을 수신함으로써 모션 벡터 예측자들 중 하나를 선택할 수도 있다. 비디오 디코더 (30) 는 그 후, 전술된 방식으로 선택된 후보 모션 벡터 예측자에 대한 모션 벡터 정보를 이용한다.

[0227] 또한, 일부 예들에서, 비디오 디코더 (30) 는 또한, 후보 모션 벡터 예측자로서 결정된 디스패리티 벡터를 포함한다. 예를 들어, 전술된 바와 같이, 후보 모션 벡터 예측자들의 리스트는 모션 벡터를 정의하는 신택스 엘리먼트들을 반드시 수신하지 않고 현재 블록에 대한 모션 벡터를 결정하기 위해 비디오 디코더 (30) 에 대한 방법을 제공한다. 일부 경우들에서, 결정된 디스패리티 벡터 (예를 들어, 디스패리티 벡터들의 후보 리스트로부터 선택된 디스패리티 벡터) 는 또한, 현재 블록에 대한 적합한 후보 모션 벡터 예측자일 수도 있고, 후보 모션 벡터 예측자일 수도 있다.

[0228] 결정된 디스패리티 벡터가 후보 모션 벡터인 예들에서, 결정된 디스패리티 벡터는 디스패리티 모션 벡터로서 고려된다. 이것은, 이들 예들에서 디스패리티 벡터가 지칭하는 블록이 현재 블록을 인터-예측하는데 직접적으로 사용되기 때문이다 (예를 들어, 비디오 인코더 (20) 는 디스패리티 벡터가 지칭하는 블록과 현재 블록 간의 잔여부들을 시그널링한다).

[0229] 일부 예들에서, 비디오 디코더 (30) 는 결정된 디스패리티 벡터를 변경한다. 예를 들어, 전술된 바와 같이, 현재 블록에 대한 결정된 디스패리티 벡터는 x-컴포넌트 및 y-컴포넌트 양자를 포함할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 결정된 디스패리티 벡터의 y-컴포넌트를 0 과 동일하게 설정한다. 비디오 디코더 (30) 는 현재 블록에 대한 후보 모션 벡터 예측자로서 변경된 디스패리티 벡터를 포함한다.

[0230] 일부 예들에서, 비디오 디코더 (30) 는 머지 모드 또는 AMVP 모드의 병렬 프로세싱을 구현한다. 예를 들어, HEVC 에서, 비디오 디코더 (30) 는 최대 코딩 유닛 (LCU) 을 평행 모션 추정 구역 (motion estimation region; MER) 으로 분할할 수도 있고, 모션 벡터 예측자들의 리스트에 대한 구성 프로세스에 포함될 상이한 MER 들에 속하는, 현재 블록에 단지 이들 이웃하는 블록들을 허용할 수도 있다.

[0231] 비디오 인코더 (20) 는 픽처 파라미터 세트 (PPS) 에서 MER 의 사이즈를 정의할 수도 있다. 예를 들어, PPS 에서의 $\log_2_{\text{parallel_merge_level_minus2}}$ 신택스 엘리먼트는 MER 의 사이즈를 정의한다. 또한, MER 사이즈가 $N \times N$ 보다 커서 $2N \times 2N$ 이 코딩 유닛 (CU) 의 최소 사이즈이면, HEVC 사양은 공간적으로 이웃하는 블록이 그 것이 현재 블록과 동일한 MER 안에 있다면 이용 가능하지 않은 것으로 고려되는 방법에서 MER 이 발효한다는 것을 정의할 수도 있다.

[0232] 또한, 상기 기법들은 인터-뷰 모션 예측에 대하여 설명된다; 그러나, 이들 기법들은 인터-뷰 잔여 예측으로 확장 가능하다. 예를 들어, 인터-뷰 잔여 예측은 HEVC 기반 3DV 에서 인에이블된다. HEVC 기반 3DV 에서, 심도 맵으로부터 결정된 디스패리티 벡터는 잔여 레퍼런스 뷰에 잔여 레퍼런스 블록을 위치시키는데 사용된다.

잔여 예측이 잔여 블록에 대해 인에이블되는 경우, 잔여 블록은 이 블록에 대한 모션 보상된 예측자들 및 시그널링된 잔여부에 추가적으로 추가된다.

[0233] 일부 예들에서, 인터-뷰 잔여 예측에 대해, 비디오 디코더 (30)는 전술된 방식 (예를 들어, SDV, TDV, 및/또는 IDV에 기초하여) 디스패리티 벡터를 결정할 수도 있다. 비디오 디코더 (30)는 디스패리티 벡터가 지칭하는 블록을 결정할 수도 있다. 이 예에서, 인터-뷰 잔여 예측을 위해, 비디오 디코더 (30)는 현재 블록의 잔여 정보를 인터-예측하기 위해 결정된 블록 (즉, 디스패리티 벡터가 지칭하는 블록)의 잔여 정보를 사용할 수도 있다. 비디오 인코더 (20)는 유사한 기법을 구현할 수도 있다.

[0234] 전술된 바와 같이, 심도 맵으로부터 결정되는 디스패리티 벡터에 의존하는 것은 연산적으로 복잡하고 시간 소모적이다. 따라서, 비디오 디코더 (30)는 디스패리티 벡터를 결정하기 위해 전술된 기법들을 이용하고, 그 후 전술된 기법과 유사한 잔여 블록에 대한 인터-뷰 잔여 예측을 구현할 수도 있다.

[0235] 상기 예들은, 비디오 디코더 (30)가 공간적으로 이웃하는 블록들 및 시간적으로 이웃하는 블록들에 기초하여 현재 블록에 대한 디스패리티 벡터를 결정하는 방식을 설명하였다. 다음의 예들은, 비디오 디코더 (30)가 어느 이웃하는 블록들이 디스패리티 벡터로 인터뷰 예측될지를 결정하기 위해 어느 공간적으로 이웃하는 블록들 및 시간적으로 이웃하는 블록들이 평가되는지를 결정하는 방식을 설명한다.

[0236] 예를 들어, 소정의 현재 뷰 및 레퍼런스 뷰 (예를 들어, 타겟 레퍼런스 뷰 또는 임의의 다른 의존적 뷰)에 대해, 공간적으로 이웃하는, 시간적으로 이웃하는 블록들, 또는 이웃하는 블록들의 이웃하는 블록들 (예컨대, 암시적 디스패리티 벡터에 대해)로부터 후보 디스패리티 벡터가 발견될 때마다 비디오 디코더 (30)는 후보 디스패리티 벡터를 포함할 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 먼저, 공간적으로 이웃하는 블록들로부터 결정된 디스패리티 벡터들을 포함하고, 후보 디스패리티 벡터들의 리스트에서 시간적으로 이웃하는 블록들로부터 결정된 디스패리티 벡터를 포함한다. 일부 예들에서, 비디오 코더 (30)는 먼저, 시간적으로 이웃하는 블록들로부터 결정된 디스패리티 벡터들을 포함하고, 그 후 후보 디스패리티 벡터들의 리스트에서 공간적으로 이웃하는 블록들로부터 결정된 디스패리티 벡터, 다음에 이웃하는 블록들 또는 이웃하는 블록들로부터 결정된 디스패리티 벡터를 포함한다.

[0237] 일부 예들에서, 비디오 디코더 (30)는 먼저, 공간적으로 이웃하는 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정하고, 디스패리티 모션 벡터가 발견되면 비디오 디코더 (30)는 디스패리티 벡터들의 후보의 리스트에 그 디스패리티 모션 벡터를 포함할 수도 있고, 시간적으로 이웃하는 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정하는 것을 진행하지 않을 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 먼저, 시간적으로 이웃하는 블록들이 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정할 수도 있고, 디스패리티 모션 벡터가 발견되면 비디오 디코더 (30)는 디스패리티 벡터들의 후보의 리스트에 그 디스패리티 모션 벡터를 포함할 수도 있고, 공간적으로 이웃하는 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정하는 것을 진행하지 않을 수도 있다. 공간적/시간적 이웃하는 블록들 및 이웃하는 블록들의 이웃하는 블록들의 체크 순서는 임의의 순열에 있을 수 있다.

[0238] 공간적으로 이웃하는 블록 또는 시간적으로 이웃하는 블록에 대해, 비디오 디코더 (30)는 공간적으로 이웃하는 블록 또는 시간적으로 이웃하는 블록에 대한 모션 벡터가 RefPicList0에서 꽉쳐를 지칭하는지 여부를 결정할 수도 있고, 모션 벡터가 디스패리티 모션 벡터이면 비디오 디코더 (30)는 이 디스패리티 모션 벡터를 후보 디스패리티 벡터들의 리스트에 포함할 수도 있다. 그렇지 않은 경우, 공간적으로 이웃하는 블록 또는 시간적으로 이웃하는 블록에 대한 모션 벡터가 RefPicList1/RefPicListC에서의 꽉쳐를 지칭하면, 비디오 디코더 (30)는 이 디스패리티 모션 벡터를 후보 디스패리티 벡터들의 리스트에 포함할 수도 있다. 이전의 예에서, 비디오 디코더 (30)는 먼저 RefPicList0, 및 그 후 RefPicList1/RefPicListC를 체크하였다. 일부 예들에서, 비디오 디코더 (30)는 먼저 RefPicList1/RefPicListC를, 그 후 RefPicList0을 체크한다. 일부 예들에서, 단지 하나의 레퍼런스 꽉쳐 리스트, 예를 들어 RefPicList0 또는 RefPicList1가 체크될 수도 있다.

[0239] 공간적으로 이웃하는 블록들에 대해, 비디오 디코더 (30)는 이웃하는 구역에 대한 모션 벡터가 임의의 순서 (예를 들어, 상부-좌측에서 시작하여 상부-우측으로 연장되거나, 상부-우측에서 시작하여 상부-좌측으로 연장됨)에서의 디스패리티 모션 벡터인지 여부를 결정할 수도 있다. 비디오 디코더 (30)가 구역들에서의 모션 벡터들을 반드시 결정하지 않는 일부 예들에서, 비디오 디코더 (30)는 공간적으로 이웃하는 4x4 블록들 중 어느 하나의 모션 벡터가 AMVP 모드 또는 머지 모드에 대한 디스패리티 모션 벡터인지 여부를 임의의 순서로 결정할 수도 있다. 예를 들어, 비디오 디코더 (30)가 4x4 공간적으로 이웃하는 블록들을 체크하는 순서는 예를 들어 인터-뷰 모션 예측을 사용하지 않는 HEVC 표준에 의해 정의된 순서에 있을 수도 있다. 예를

들어, 머지 모드에 있어서, 인터-뷰 모션 예측을 사용하지 않는 예들에 대해, HEVC 는 비디오 디코더 (30) 가 도 4 에 예시된 바와 같은 A1, B1, B0, AO, 및 B2 의 순서로 모션 벡터들을 체크한다고 정의한다. 이 예에서, 비디오 디코더 (30) 는 그 순서에서의 A1, B1, B0, AO, 및 B2 중 어느 하나에 대한 모션 벡터가 디스패리티 모션 벡터들인지 여부를 결정할 수도 있다.

[0240] 대안으로, 비디오 디코더 (30) 는 상기 5 개의 블록들 (즉, A1, B1, B0, AO, 및 B2) 의 순서에서의 임의의 순열 또는 공간적으로 이웃하는 구역들의 임의의 순열을 이용할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는, 이웃하는 블록들 또는 이웃하는 구역들의 서브세트에 대한 모션 벡터들이 디스패리티 모션 벡터인지 여부 및 일부 예들에서 임의의 순열로 결정할 수도 있다.

[0241] 또한, 공간적으로 이웃하는 블록이 예측될 현재 블록과 동일한 MER 안에 있으면, 비디오 디코더 (30) 는 그 공간적으로 이웃하는 블록이 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정하지 않는다. 그러나, 본 개시물에 설명된 기법들은 그렇게 제한되지 않는다. 일부 다른 예들에서, 공간적으로 이웃하는 블록이 예측될 현재 블록과 동일한 MER 내에 있으면, 비디오 디코더 (30) 는 그 블록이 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정한다.

[0242] 전술된 바와 같이, 일부 예들에서, 비디오 디코더 (30) 는, 공간적으로 이웃하는 블록들이 블록 대 블록 단위로 인터-뷰 예측되는지 여부를 결정할 필요가 없고, 공간적으로 이웃하는 블록들이 구역 대 구역 단위로 인터-뷰 예측되는지 여부를 결정할 수도 있다. 예를 들어, 구역은 복수의 블록들을 포함할 수도 있다. 일 예로 써, 블록은 4x4 일 수도 있고, 구역은 8x4 구역을 포함하는 2 개의 블록들일 수도 있다. 블록들 및 구역들의 다른 사이즈들이 가능할 수도 있고, 본 개시물에 설명된 기법들은 4x4 블록들 및 8x4 구역들의 예들에 제한되는 것으로 고려되지 않아야 한다.

[0243] 시간적으로 이웃하는 블록들에 대해, 비디오 디코더 (30) 는 어느 함께-위치된 픽처들이 평가되고, 함께-위치된 픽처들 내의 어느 블록들이 평가되어 시간적으로 이웃하는 블록들이 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정할지를 결정할 수도 있다. 예를 들어, 일부 예들에서 함께-위치된 픽처를 선택하기 위해, 비디오 디코더 (30) 는 현재 블록을 포함하는 현재 픽처의 레퍼런스 픽처 리스트(들)에서 식별된 픽처들을 평가한다. 이들 예들에서, 비디오 디코더 (30) 는 레퍼런스 픽처 리스트(들)에서 식별된 픽처들을 오름 차순 (예를 들어, 리스트의 제 1 픽처에서 시작하고 최종 픽처에서 끝남) 으로 평가한다. 일부 예들에서, 비디오 디코더 (30) 는 레퍼런스 픽처 리스트(들)에서 식별된 픽처들을 내림 차순 (예를 들어, 리스트의 최종 픽처에서 시작하고 리스트의 제 1 픽처에서 끝남) 으로 평가한다.

[0244] 일 예로써, 비디오 인코더 (20) 에 의해 시그널링된 num_ref_idx_10_active_minus1 신팩스 엘리먼트는 RefPicList0 에서 다수의 엔트리들을 식별하고, 비디오 인코더 (20) 에 의해 시그널링된 num_ref_idx_11_active_minus1 또는 num_ref_idx_1C_active_minus1 는 RefPicList1 또는 RefPicListC 각각에서 다수의 엔트리들을 식별한다. num_ref_idx_1C_active_minus1 신팩스 엘리먼트는 현재 HEVC 표준의 일부가 아니고, 추가의 예로서 여기에 포함될 수도 있다. 예시의 목적을 위해, 이 기법들은 RefPicList0 및 RefPicList1 인 레퍼런스 픽처 리스트들로 설명된다. 이 예에서, N 을 레퍼런스 픽처 리스트(들)에서 다수의 엔트리들을 정의하는 이들 2 개의 시그널링된 값들 중 더 큰 것과 동일하게 한다. 또한, ref_idx 는 RefPicList0 및 RefPicList1 에서의 레퍼런스 인덱스를 지칭하게 한다.

[0245] 일 예에서, 비디오 디코더 (30) 는 전술된 collocated_from_10_flag 의 값에 기초하여 RefPicList0 및 RefPicList1 에서 식별된 픽처들을 체크한다. 예를 들어, collocated_from_10_flag 가 0 이면, 비디오 디코더 (30) 는 먼저, RefPicList0[ref_idx] 에서 식별된 픽처들이 (여기서, ref_idx 는 0 에서 N 까지의 범위에 이름) 디스패리티 모션 벡터에 대하여 인터-뷰 예측되는 함께-위치된 블록들을 포함하는지 여부를 결정한다.

RefPicList0 의 픽처들 중 어느 것에서의 함께-위치된 블록들 중 어느것도 디스패리티 모션 벡터로 인터뷰 예측되지 않으면, 비디오 디코더는 RefPicList1[ref_idx] 에서 식별된 픽처들이 (여기서, ref_idx 는 0 에서 N 까지의 범위에 이름) 디스패리티 모션 벡터에 대하여 인터뷰 예측되는 함께-위치된 블록들을 포함하는지 여부를 결정한다.

[0246] collocated_from_10_flag 이 1 이면, 비디오 디코더 (30) 는 먼저, RefPicList1[ref_idx] 에서 식별된 픽처들이 (여기서, ref_idx 는 0 에서 N 까지의 범위에 이름) 디스패리티 모션 벡터에 대하여 인터-뷰 예측되는 함께-위치된 블록들을 포함하는지 여부를 결정한다. RefPicList1 의 픽처들 중 어느 것에서의 함께-위치된 블록들 중 어느것도 디스패리티 모션 벡터로 인터뷰 예측되지 않으면, 비디오 디코더는 RefPicList0[ref_idx] 에서 식별된 픽처들이 (여기서, ref_idx 는 0 에서 N 까지의 범위에 이름) 디스패리티 모션 벡터에 대하여 인터뷰 예

측되는 함께-위치된 블록들을 포함하는지 여부를 결정한다.

[0247] 일부 예들에서, 비디오 디코더 (30)는 먼저, RefPicList0[ref_idx]에 의해 식별되는 바와 같은 RefPicList0에서의 임의의 픽처가 collocated_from_10_flag의 값에 관계없이 디스패리티 벡터로 인터뷰 예측되는 함께-위치된 블록을 포함하는지 여부를 체크하고, 그 후 RefPicList1[ref_idx]에 의해 식별되는 바와 같은 RefPicList1에서의 임의의 픽처가 collocated_from_10_flag의 값에 관계없이 디스패리티 벡터로 인터뷰 예측되는 함께-위치된 블록을 포함하는지 여부를 체크한다. 일부 예들에서, 비디오 디코더 (30)는 먼저, RefPicList1[ref_idx]에 의해 식별되는 바와 같은 RefPicList1에서의 임의의 픽처가 collocated_from_10_flag의 값에 관계없이 디스패리티 벡터로 인터뷰 예측되는 함께-위치된 블록을 포함하는지 여부를 체크하고, 그 후 RefPicList0[ref_idx]에 의해 식별되는 바와 같은 RefPicList0에서의 임의의 픽처가 collocated_from_10_flag의 값에 관계없이 디스패리티 벡터로 인터뷰 예측되는 함께-위치된 블록을 포함하는지 여부를 체크한다.

[0248] RefPicList0 또는 RefPicList1에서 픽처들을 체크하는 경우, 픽처들 중 하나가 현재 블록의 뷰와 상이한 뷰에서의 픽처에 대한 것이면, 비디오 디코더 (30)는 디스패리티 벡터로 인터뷰 예측되는 그러한 픽처 (예를 들어, 레퍼런스 픽처 리스트에서 상이한 뷰로부터의 픽처는 스킵될 수도 있음)에서 임의의 함께-위치된 블록들이 존재하는지 여부를 결정하지 않을 수도 있다. 대안으로, 비디오 디코더 (30)는 현재 픽처의 뷰와 상이한 뷔에서의 픽처가 디스패리티 모션 벡터로 인터-뷰 예측되는 함께-위치된 블록들을 포함하는지 여부를 결정할 수도 있다. 예를 들어, 현재 블록의 뷔와 상이한 뷔에서의 픽처는 현재 뷔를 포함하는 픽처와 동일한 액세스 유닛에서 이전에 디코딩된 픽처일 수도 있다.

[0249] 일부 예들에서, 비디오 디코더 (30)는 현재 픽처의 뷔와 상이한 뷔에서의 픽처가, 픽처가 RefPicList0 또는 RefPicList1에 포함되지 않더라도 디스패리티 모션 벡터로 인터-뷰 예측되는 함께-위치된 블록들을 포함하는지 여부를 결정할 수도 있다. 이들 예들에서, 일부 경우들에서, 현재 블록을 포함하는 픽처와 동일한 액세스 유닛에서의 픽처들을 갖는 다수의 뷔들이 존재하는 것이 가능할 수도 있다. 이들 경우들에 대해, 비디오 디코더 (30)는 뷔 순서 인덱스들의 내림 차순으로 픽처들을 체크할 수도 있다. 예를 들어, 비디오 디코더 (30)는 최상위 뷔 순서 인덱스를 갖는 뷔, 다음에 두 번째 최상위 뷔 순서 인덱스를 갖는 뷔 등에서의 픽처가 디스패리티 모션 벡터로 인터-뷰 예측되는 함께-위치된 블록들을 포함하는지 여부를 결정할 수도 있다.

[0250] 다른 예에서, 비디오 디코더 (30)는 현재 블록에 대한 뷔 식별자 및 상이한 뷔들에 있는 픽처들 각각에 대한 뷔 식별자들 간의 차이를 결정한다. 비디오 디코더 (30)는 이 차이에 기초하여 오름 차순으로 픽처들을 체크한다. 예를 들어, 현재 블록을 포함하는 현재 픽처와 동일한 액세스 유닛 내에 각각 있는 제 2 뷔에서의 픽처 및 제 1 뷔에서의 픽처가 존재하고, 제 1 뷔에 대한 뷔 식별자 및 현재 뷔에 대한 뷔 식별자 간의 차이는 제 2 뷔에 대한 뷔 식별자와 현재 뷔에 대한 뷔 식별자 간의 차이보다 더 작다고 가정한다. 이 경우에서, 비디오 디코더 (30)는 먼저 제 1 뷔에서의 픽처의 함께-위치된 블록들 중 어느 하나가 디스패리티 벡터로 인터-뷰 예측되는지 여부를 결정하고, 그렇지 않은 경우 제 2 뷔에서의 픽처의 함께-위치된 블록들 중 어느 하나가 디스패리티 벡터로 인터-뷰 예측되는지 여부를 결정할 수도 있다.

[0251] 일부 예들에서, 비디오 디코더 (30)는, 픽처들 중 어느 하나가 디스패리티 모션 벡터로 인터-뷰 예측되는 함께-위치된 블록들을 포함하는지 여부를 결정하기 위해 레퍼런스 픽처 리스트들에서 식별된 픽처들의 서브세트만을 체크한다. 일부 예들에서, 이 픽처들의 서브세트는 상대적으로 작은 레퍼런스 인덱스들로 식별되는 픽처들 (예를 들어, RefPicList0 및 RefPicList1에서 레퍼런스 인덱스 0 및 레퍼런스 인덱스 1로 식별되는 픽처들)을 포함할 수도 있다. 다시 말해, 비디오 디코더 (30)는 RefPicList0 및 RefPicList1의 생략 (truncated) 버전에서 식별된 픽처들을 체크할 수도 있다.

[0252] 상기에서는, 비디오 디코더 (30)가 디스패리티 모션 벡터로 인터-뷰 예측되는 함께-위치된 블록들을 식별하기 위해 함께-위치된 픽처들을 체크하는 순서의 일부 예시의 방법들을 설명한다. 다음은, 비디오 디코더 (30)가 디스패리티 모션 벡터로 인터-뷰 예측되는 함께-위치된 블록들을 식별하기 위해 각각의 픽처를 체크하는 함께-위치된 블록들을 설명한다.

[0253] 전술된 바와 같이, 함께-위치된 블록들의 3 개의 예들 (예를 들어, 도 10에 대하여 전술된 바와 같은 CR 블록, CLCU 블록, 및 BR 블록)이 존재할 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 CR 블록, CLCU 블록, 및 BR 블록의 각 블록을 체크하여, 어느 함께-위치된 블록들이 디스패리티 모션 벡터로 인터-뷰 예측되는지를 결정할 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 CR 블록, CLCU 블록, 및 BR 블록 중 하나 이상을 체크할 수도 있으나, 3 개 전부를 체크하지는 않는다. 예를 들어, 비디오 디코더 (30)는 일 경우에서는 BR

블록을 체크하지 않을 수도 있고, 일 경우에서는 CR 블록을 체크하지 않을 수도 있으며, 일 경우에서는 CLCU 블록을 체크하지 않을 수도 있다.

[0254] 일부 예들에서, 비디오 디코더 (30) 가 함께-위치된 블록들의 3 개의 예들을 체크하는 순서는 상이한 뷰들에서의 핵심들에 대해 상이하다. 예를 들어, 제 1 뷰에서의 핵심에 대해, 비디오 디코더 (30) 는 먼저 CR 블록, 다음에 CLCU 블록을 체크하고, 그 후 BR 블록을 체크하여 이를 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터뷰 예측되는지 여부를 결정한다. 제 2 뷰에서의 핵심에 대해, 비디오 디코더 (30) 는 먼저, BR 블록, 다음에 CR 블록을 체크하고 그 후 CLCU 블록을 체크하여, 이를 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터뷰 예측되는지 여부를 결정한다.

[0255] 각각의 CR 블록 또는 CLCU 블록에 대해, 비디오 디코더 (30) 는 4x4 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터뷰 예측되는지 여부를 결정하기 위해 CR 블록 또는 CLCU 블록에서 각각의 4x4 블록을 래스터 스캔할 수도 있다. 비디오 디코더 (30) 가 CLCU 블록을 래스터 스캔하는 예들에서, 비디오 디코더 (30) 는 CR 블록의 4x4 블록들을 스kip할 수도 있고, CR 블록의 4x4 블록들 중 어느 하나가 디스패리티 모션 벡터로 인터뷰 예측되는지 여부를 결정하지 않을 수도 있다.

[0256] 그러나, 각각의 4x4 시간적으로 이웃하는 블록을 체크하는 것이 필요하지 않을 수도 있다. 예를 들어, 전술된 바와 같이, 비디오 디코더 (30) 는 4x4 블록들 각각에 대한 모션 벡터 정보 보다는, 모션 벡터 압축으로 인해 16 개의 4x4 블록들을 포함하는 하나의 16x16 구역에 대한 모션 벡터 정보를 저장할 수도 있다. 이를 예들에서, 비디오 디코더 (30) 는 16x16 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정하기 위해 래스터 스캔 순서로 16x16 구역들을 체크할 수도 있다. 비디오 디코더 (30) 는 함께-위치된 핵심들이 상이한 액세스 유닛이지만 동일한 뷰에 있는 예들에서, 그리고 함께-위치된 핵심들이 동일한 액세스 유닛이지만 상이한 뷰에 있는 예들에서 16x16 구역들을 체크할 수도 있다.

[0257] 16x16 구역은 일 예로써 설명되고, 제한으로 고려되지 않아야 한다. 일부 예들에서, 구역은 16x16 보다 더 크거나 더 작을 수도 있고, 비디오 디코더 (30) 가 시간적으로 이웃하는 블록들 (예를 들어, 동일한 뷰에 있지만 상이한 시간 인스턴스에서의 핵심들의 블록들 또는 상이한 뷰이지만 동일한 시간 인스턴스에서의 핵심들의 블록들) 에 대한 모션 벡터 정보를 압축하는 방식에 기초할 수도 있다.

[0258] 일부 예들에서, 비디오 디코더 (30) 는 NxN 블록들 상에서 함께-위치된 블록들을 래스터 스캔할 수도 있고, 여기서 2Nx2N 은 최소 코딩 유닛 (CU) 의 사이즈이다. 일부 예들에서, 비디오 디코더 (30) 는 예측 유닛 (PU) 들 상에서 래스터 스캔할 수도 있다.

[0259] 본 개시물에 설명된 기법들에 따르면, (예를 들어, 공간적으로 이웃하는 블록들 및 시간적으로 이웃하는 블록들의 스캐닝에 기초하여) 적어도 하나의 디스패리티 모션 벡터를 식별한 후에, 비디오 디코더 (30) 는 상이한 목적들을 위해 디스패리티 모션 벡터를 이용할 수도 있다. 일 예로써, 비디오 디코더 (30) 는 현재 블록에 대한 후보 디스패리티 벡터들의 리스트에 디스패리티 모션 벡터들을 포함할 수도 있고, 선택된 후보 디스패리티 벡터에 의해 지정된 블록에 대한 모션 벡터 정보는 AMVP 모드 및 머지 모드에 대한 후보 모션 벡터 예측자들의 리스트에 포함된다. 다른 예로써, 비디오 디코더 (30) 는 디스패리티 모션 벡터를 후보 모션 벡터 예측자들의 리스트에 포함할 수도 있다. 이 예에서, 일부 경우들에서, 비디오 디코더 (30) 는 디스패리티 모션 벡터의 y-컴포넌트를 0 과 동일하게 설정함으로써 디스패리티 모션 벡터를 변경할 수도 있다.

[0260] 상기 예시의 기법은 비디오 디코더 (30) 의 관점으로부터 설명되었으나, 본 개시물에 설명된 기법들은 그렇게 제한되지 않는다. 일반적으로, 비디오 인코더 (20) 는 비디오 디코더 (30) 에 대하여 전술된 것들과 유사한 기법들을 구현하도록 구성될 수도 있다. 더욱이, 일부 예들에서, 비디오 디코더 (30) 는 본 개시물에 설명된 기법들을 구현하도록 구성되고, 비디오 인코더 (20) 는 그렇지 않다. 일부 예들에서, 비디오 인코더 (20) 는 본 개시물에 설명된 기법들을 구현하도록 구성되고, 비디오 디코더 (30) 는 그렇지 않다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 양자는 본 개시물에 설명된 기법들을 구현하도록 구성된다.

[0261] 따라서, 본 개시물은, 제 1 뷰에서 핵심의 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정하도록 구성되는 비디오 코더 (예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)) 를 설명한다. 이 예에서, 구역은 예측될 현재 블록을 시간적으로 또는 공간적으로 이웃한다. 예를 들어, 구역은 하나 이상의 시간적으로 또는 공간적으로 이웃하는 블록들을 포함하고, 이 구역에 대한 모션 벡터는 하나의 레퍼런스 핵심리스트에 대한 복수의 공간적 또는 시간적으로 이웃하는 블록들에 대한 단일의 모션 벡터를 나타낸다. 예를 들어, 2 개의 레퍼런스 핵심리스트들로부터 인터-예측된다면, 구역에 대한 하나의 모션 벡터는 하나의 레퍼런

스 팩처 리스트 (예를 들어, RefPicList0 또는 RefPicList1 중 하나)에 대한 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타내고, 구역에 대한 다른 모션 벡터는 다른 레퍼런스 팩처 리스트 (예를 들어, RefPicList0 또는 RefPicList1 중 다른 하나)에 대한 하나 이상의 블록들에 대한 단일의 모션 벡터를 나타낸다. 비디오 코더는 또한, 현재 블록에 대한 후보 디스패리티 벡터들의 리스트에 디스패리티 모션 벡터를 포함하도록 구성된다. 또한, 일부 예들에서, 현재 블록은 순간 디코더 리프레시 (IDR) 또는 클린 랜덤 액세스 (CRA) 팩처의 블록이다.

[0262] 구역이 공간적으로 이웃하는 구역인 예들에서, 구역은 2 개의 4x4 블록들을 포함하는 8x4 구역일 수도 있다.

이 예에서, 공간적으로 이웃하는 구역을 포함하는 팩처는 또한, 예측될 현재 블록을 포함한다. 구역이 시간적으로 이웃하는 구역인 예들에서, 구역은 16 개의 4x4 블록들을 포함하는 16x16 구역일 수도 있다. 이 예에서, 시간적으로 이웃하는 구역을 포함하는 팩처는 현재 블록을 포함하지 않는다. 전술된 바와 같이, 시간적 및/또는 공간적으로 이웃하는 블록들에 대한 모션 벡터 정보를 저장하기 위해 압축이 사용되지 않으면, 공간적으로 이웃하는 구역 및/또는 시간적으로 이웃하는 구역은 복수의 블록들 보다는 하나의 블록을 포함할 수도 있다.

[0263] 본 개시물에 설명된 기법들에 따르면, 비디오 코더는, 레퍼런스 팩처 리스트 (예를 들어, RefPicList0 및 RefPicList1 중 하나 또는 양자)에서 식별된 팩처들에서 시간적으로 이웃하는 구역들이 디스패리티 모션 벡터들로 인터-뷰 예측되는지 여부를 결정하도록 구성될 수도 있다. 예를 들어, 비디오 코더가, 제 1 팩처의 구역이 인터-뷰 예측되는지 여부를 결정한 후에, 비디오 코더는 제 2 팩처의 구역이 인터-뷰 예측되는지 여부를 결정한다. 제 2 팩처의 구역이 인터-뷰 예측되면, 비디오 코더는 현재 블록에 대한 후보 디스패리티 벡터들의 리스트에 제 2 팩처의 구역에 대한 디스패리티 모션 벡터를 포함한다.

[0264] 이 예에서, 제 1 팩처 및 제 2 팩처 양자 모두는 레퍼런스 팩처 리스트에 속한다. 또한, 일부 예들에서, 레퍼런스 팩처 리스트는 오름차순으로 제 1 팩처 및 제 2 팩처를 식별한다. 예를 들어, 레퍼런스 팩처 리스트에서 제 1 팩처를 식별하는 레퍼런스 인덱스는 레퍼런스 팩처 리스트에서 제 2 팩처를 식별하는 레퍼런스 인덱스보다 더 작다. 다시 말해, 제 1 팩처는 레퍼런스 팩처 리스트에서 제 2 팩처보다 더 빨리 식별된다.

[0265] 더욱이, 비디오 코더는 예컨대 머지 모드 및 AMVP 모드에 대한 인터-뷰 예측 목적들을 위해 디스패리티 모션 벡터를 이용할 수도 있다. 예를 들어, 비디오 코더는 디스패리티 모션 벡터를 현재 블록에 대한 디스패리티 벡터로서 설정한다. 비디오 코더는 디스패리티 벡터가 지칭하는 블록을 결정한다. 비디오 코더는 또한, 디스패리티 벡터가 지칭하는 블록에 대한 모션 벡터를 결정하고, 이 결정된 모션 벡터를 후보 모션 벡터 예측자들의 리스트에 포함한다. 비디오 코더는 그 후, 후보 모션 벡터 예측자들 중 하나에 기초하여 현재 블록을 인터-예측할 수도 있다.

[0266] 일부 예들에서, 비디오 코더는 후보 모션 벡터 예측자들의 리스트에 디스패리티 모션 벡터를 포함한다. 예를 들어, 비디오 코더는 디스패리티 모션 벡터의 y-컴포넌트를 0과 동일하게 설정함으로써 디스패리티 모션 벡터를 변경하고, 이 변경된 디스패리티 모션 벡터를 후보 모션 벡터 예측자들의 리스트에 포함한다.

[0267] 일부 예들에서, 비디오 코더는 인터-뷰 잔여 예측을 위해 본 개시물에 설명된 기법들을 구현할 수도 있다. 예를 들어, 비디오 코더는 디스패리티 벡터가 지칭하는 블록을 결정할 수도 있고, 현재 블록의 잔여 정보를 인터-예측하기 위해 디스패리티 벡터가 지칭하는 블록 (즉, 결정된 블록)의 잔여 정보를 사용할 수도 있다.

[0268] 도 12는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 예시하는 블록도이다. 예를 들어, 도 12는 비디오 슬라이스들 내의 비디오 블록들의 인트라 및 인터 코딩을 수행할 수도 있는 비디오 인코더 (20)를 예시한다. 예를 들어, 비디오 인코더 (20)는 인터 예측 인코딩 또는 인트라 예측 인코딩을 수행할 수도 있다. 인트라 코딩은 공간적 예측에 의존하여, 주어진 비디오 프레임 또는 팩처의 비디오에서 공간적 리던던시를 감소시키거나 또는 제거한다. 인터 코딩은 시간적 예측에 의존하여, 비디오 시퀀스의 인접한 프레임들 또는 팩처들 내의 비디오에서 시간적 리던던시를 감소시키거나 또는 제거한다. 인트라 모드 (I 모드)는 여러 공간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 및 양방향 예측 (B 모드)과 같은 인터 모드들은 여러 시간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다.

[0269] 도 12의 예에서, 비디오 인코더 (20)는 파티셔닝 유닛 (152), 예측 프로세싱 유닛 (154), 레퍼런스 팩처 메모리 (176), 합산기 (162), 변환 프로세싱 유닛 (164), 양자화 프로세싱 유닛 (166), 및 엔트로피 인코딩 유닛 (168)을 포함한다. 예측 프로세싱 유닛 (154)은 모션 추정 유닛 (156), 모션 보상 유닛 (158), 및 인트라

예측 유닛 (160) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역 양자화 프로세싱 유닛 (170), 역 변환 프로세싱 유닛 (172), 및 합산기 (174) 를 포함한다. 블록화제거 (deblocking) 필터 (도 8에서 미도시) 가 또한 블록 경계들을 필터링하여 재구성된 비디오로부터 블록형 (blockiness) 아티팩트들을 제거하기 위해 포함될 수도 있다. 원한다면, 블록화제거 필터는 통상 합산기 (174) 의 출력을 필터링 할 것이다. 부가적인 루프 필터들 (인 루프 또는 포스트 루프) 이 또한 블록화제거 필터에 부가적으로 사용될 수도 있다. 레퍼런스 픽처 메모리 (176) 는 디코딩된 픽처 베피 (DPB) 의 일 예이다.

[0270] 도 12 에 도시된 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고, 파티셔닝 유닛 (152) 은 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한 슬라이스들, 타일들, 또는 다른 큰 유닛들로의 파티셔닝, 뿐만 아니라, (예컨대, LCU 들 및 CU 들의 쿼드트리 구조에 따른) 비디오 블록 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 슬라이스는 다수의 비디오 블록들로 (및 가능하게는 타일들로서) 지정된 비디오 블록들의 세트들로 나누어질 수도 있다. 예측 프로세싱 유닛 (154) 은 복수의 가능한 코딩 모드들 중 하나, 예컨대 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나를 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨) 에 기초하여 현재 비디오 블록에 대해 선택할 수도 있다. 예측 프로세싱 유닛 (154) 은 결과적인 인트라 또는 인터 코딩된 블록을 잔여 블록 데이터를 생성하는 합산기 (162) 및 레퍼런스 픽처로서 사용하기 위해 인코딩된 블록을 재구성하는 합산기 (174) 에 제공할 수도 있다.

[0271] 예측 프로세싱 유닛 (154) 내의 인트라 예측 유닛 (160) 은 코딩될 현재 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들을 기준으로 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (154) 내의 모션 추정 유닛 (156) 및 모션 보상 유닛 (158) 은 하나 이상의 레퍼런스 픽처들 및/또는 레퍼런스 뷰들에서의 하나 이상의 예측 블록들을 기준으로 현재 비디오 블록의 인터 예측 코딩을 수행하여 시간적 압축을 제공한다.

[0272] 모션 추정 유닛 (156) 은 비디오 시퀀스에 대한 미리 결정된 패턴에 따라 비디오 슬라이스에 대한 인터 예측 모드를 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들, 또는 B 슬라이스들이라고 지정할 수도 있다. 모션 추정 유닛 (156) 과 모션 보상 유닛 (158) 은 고도로 통합될 수도 있지만 개념상의 목적들을 위해 별개로 예시된다. 모션 추정 유닛 (156) 에 의해 수행된 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어, 레퍼런스 픽처의 예측 블록에 대한 현재 비디오 프레임 또는 픽처의 비디오 블록의 PU의 변위 (displacement) 를 나타낼 수도 있다.

[0273] 예측 블록은 차의 절대값 합 (SAD), 차의 제곱 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 화소 차이의 관점에서 코딩될 비디오 블록의 PU 에 밀접하게 매칭되는 것으로 발견되는 블록이다. 일부 예들에서, 비디오 인코더 (20) 는 레퍼런스 픽처 메모리 (176) 에 저장된 레퍼런스 픽처들의 부 정수 (sub-integer) 화소 포지션들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 레퍼런스 픽처의 1/4 화소 포지션들, 1/8 화소 포지션들, 또는 다른 분수 화소 포지션들의 값들을 보간할 수도 있다. 따라서, 모션 추정 유닛 (156) 은 풀 (full) 화소 포지션들 및 분수 화소 포지션들에 대한 모션 검색을 수행하고 분수 화소 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0274] 모션 추정 유닛 (156) 은 PU의 포지션과 레퍼런스 픽처의 예측 블록의 포지션을 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 레퍼런스 픽처는 레퍼런스 픽처 메모리 (176) 에 저장된 하나 이상의 레퍼런스 픽처들을 각각 식별하는 제 1 레퍼런스 픽처 리스트 (RefPicList0) 또는 제 2 레퍼런스 픽처 리스트 (RefPicList1) 로부터 선택될 수도 있다. 모션 추정 유닛 (156) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (168) 및 모션 보상 유닛 (158) 으로 전송한다.

[0275] 모션 보상 유닛 (158) 에 의해 수행되는 모션 보상은 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐치하는 것 또는 생성하는 것, 가능하게는 부 화소 (sub-pixel) 정밀도로의 보간들을 수행하는 것을 수반할 수도 있다. 현재 비디오 블록의 PU 에 대한 모션 벡터를 수신 시, 모션 보상 유닛 (158) 은 레퍼런스 픽처 리스트들 중 하나에 모션 벡터가 가리키는 예측 블록을 위치시킬 수도 있다. 비디오 인코더 (20) 는, 코딩되고 있는 현재 비디오 블록의 화소 값들로부터 예측 블록의 화소 값들을 감산하여 화소 차이 값들을 형성함으로써 잔여 비디오 블록을 형성한다. 화소 차이 값들은 블록에 대한 잔여 데이터를 형성하고, 루마 및 크로마 차이 컴포넌트들 양자 모두를 포함할 수도 있다. 합산기 (162) 는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 모션 보상 유닛 (158) 은 또한 비디오 슬라이스의 비디오 블록들의 디코딩에서

비디오 디코더 (30)에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스에 연관된 선택스 엘리먼트들을 생성할 수도 있다. 모션 추정 유닛 (156) 및 모션 보상 유닛 (158)은 함께 또한, 블록이 다른 뷰에서의 블록에 대하여 인터 예측되는 인터 예측을 구현할 수도 있다.

[0276] 인트라 예측 유닛 (160)은, 전술된 바와 같이, 모션 추정 유닛 (156)과 모션 보상 유닛 (158)에 의해 수행되는 인터 예측에 대한 대안으로서 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 유닛 (160)은 현재 블록을 인코딩하는데 사용하기 위한 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 유닛 (160)은 예컨대, 개별 인코딩 과정들 동안에 다양한 인트라 예측 모드들을 사용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 유닛 (160) (또는 일부 예들에서, 모드 선택 유닛)은 테스트된 모드들로부터 사용할 적절한 인트라 예측 모드를 선택할 수도 있다. 예를 들어, 인트라 예측 유닛 (160)은 다양한 테스트된 인트라 예측 모드들에 대한 레이트 왜곡 분석을 사용하여 레이트 왜곡 값들을 계산하고, 테스트된 모드들 중에서 최상의 레이트 왜곡 특징들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록 및 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러)의 양, 뿐만 아니라 인코딩된 블록을 생성하는데 사용된 비트 레이트 (다시 말하면, 비트들의 수)를 결정한다. 인트라 예측 유닛 (160)은 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 계산하여 어느 인트라 예측 모드가 그 블록에 대해 최상의 레이트 왜곡 값을 보이는지를 결정할 수도 있다.

[0277] 임의의 경우에서, 블록에 대한 인트라 예측 모드를 선택한 후, 인트라 예측 유닛 (160)은 그 블록에 대한 선택된 인트라 예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (168)에 제공할 수도 있다. 엔트로피 인코딩 유닛 (168)은 본 개시물의 기법들에 따른 선택된 인트라 예측 모드를 나타내는 정보를 인코딩할 수도 있다.

비디오 인코더 (20)는 송신되는 비트스트림에 구성 데이터를 포함시킬 수도 있는데, 그 구성 데이터는 복수의 인트라 예측 모드 인덱스 표들 및 복수의 변경된 인트라 예측 모드 인덱스 표들 (또한 코드워드 매핑 표들로서 지칭됨), 다양한 블록들에 대한 콘텍스트들을 인코딩하는 정의들, 및 가장 있음직한 인트라 예측 모드의 표시들, 인트라 예측 모드 인덱스 표, 및 콘텍스트들의 각각에 대한 사용을 위한 변경된 인트라 예측 모드 인덱스 표를 포함할 수도 있다.

[0278] 예측 프로세싱 유닛 (154)이 인터 예측 또는 인트라 예측 중 어느 하나를 통해 현재 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 인코더 (20)는 현재 비디오 블록으로부터 예측 블록을 감산함으로써 잔여 비디오 블록을 형성한다. 잔여 블록에서의 잔여 비디오 데이터는, 하나 이상의 TU 들에 포함되고 변환 프로세싱 유닛 (164)에 적용될 수도 있다. 변환 프로세싱 유닛 (164)은 변환, 예컨대 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환을 사용하여 잔여 비디오 데이터를 잔여 변환 계수들로 변환시킨다. 변환 프로세싱 유닛 (164)은 잔여 비디오 데이터를 화소 도메인으로부터 변환 도메인, 예컨대 주파수 도메인으로 변환할 수도 있다.

[0279] 변환 프로세싱 유닛 (164)은 결과적인 변환 계수들을 양자화 프로세싱 유닛 (166)에 전송할 수도 있다. 양자화 프로세싱 유닛 (166)은 변환 계수들을 양자화하여 비트 레이트를 더욱 감소시킨다. 양자화 프로세스는 그 계수들의 일부 또는 전부에 연관된 비트 심도를 감소시킬 수도 있다. 양자화 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 일부 예들에서, 양자화 프로세싱 유닛 (166)은 그 후 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안으로, 엔트로피 인코딩 유닛 (168)이 그 스캔을 수행할 수도 있다.

[0280] 양자화 다음에, 엔트로피 인코딩 유닛 (168)은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (168)은 콘텍스트 적응 가변 길이 코딩 (CAVLC), 콘텍스트 적응 이진 산술 코딩 (CABAC), 선택스 기반 콘텍스트 적응 이진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 다른 엔트로피 인코딩 수법 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (168)에 의한 엔트로피 인코딩 다음에, 인코딩된 비트스트림은 비디오 디코더 (30)로 송신되거나 또는 나중의 송신 또는 비디오 디코더 (30)에 의한 취출을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (168)은 코딩되고 있는 현재 비디오 슬라이스에 대한 모션 벡터들 및 다른 선택스 엘리먼트들을 또한 엔트로피 인코딩할 수도 있다.

[0281] 역 양자화 프로세싱 유닛 (170)과 역 변환 프로세싱 유닛 (172)은 역 양자화 및 역 변환을 각각 적용하여, 레퍼런스 픽처의 레퍼런스 블록으로서 나중에 사용하기 위해 화소 도메인에서 잔여 블록을 재구성한다. 모션 보상 유닛 (158)은 레퍼런스 픽처 리스트들 중 하나의 레퍼런스 픽처 리스트 내의 레퍼런스 픽처들 중 하나의 레퍼런스 픽처의 예측 블록에 잔여 블록을 가산함으로써 레퍼런스 블록을 계산할 수도 있다. 모션 보상 유

닛 (158) 은 또한 하나 이상의 보간 필터들을 재구성된 잔여 블록에 적용하여 모션 추정에서 사용하기 위한 부정수 화소 값들을 계산할 수도 있다. 합산기 (174) 는 재구성된 잔여 블록을 모션 보상 유닛 (158) 에 의해 생성된 모션 보상된 예측 블록에 가산하여, 레퍼런스 픽처 메모리 (176) 에 저장하기 위한 레퍼런스 블록을 생성한다. 레퍼런스 블록은 모션 추정 유닛 (156) 과 모션 보상 유닛 (158) 에 의해 후속 비디오 프레임 또는 픽처에서 블록을 인터 예측하기 위한 레퍼런스 블록으로서 사용될 수도 있다.

[0282] 이 방식으로, 비디오 인코더 (20) 는 본 개시물에 설명된 하나 이상의 예시의 기법들을 구현하도록 구성되는 비디오 인코더의 일 예이다. 예를 들어, 예측 프로세싱 유닛 (154) 은 공간적 및 시간적으로 이웃하는 구역들의 디스패리티 모션 벡터들에 기초하여 현재 블록에 대한 디스패리티 벡터를 결정하도록 구성될 수도 있다. 예측 프로세싱 유닛 (154) 은 또한, 후보 모션 벡터 예측자들의 리스트에 대한 후보 모션 벡터 예측자들을 결정하기 위해 결정된 디스패리티 벡터를 이용하도록 구성될 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (154) 은 결정된 디스패리티 벡터를 변형하고, 이 변형된, 결정된 디스패리티 벡터를 후보 모션 벡터 예측자들의 리스트에 포함할 수도 있다. 예측 프로세싱 유닛 (154) 은 또한, 인터-뷰 모션 예측 및 인터-뷰 잔여 예측을 위해 구성될 수도 있다.

[0283] 일부 예들에서, 예측 프로세싱 유닛 (154) 외의 유닛은 전술된 예들을 구현할 수도 있다. 일부 다른 예들에서, 비디오 인코더 (20) 의 하나 이상의 다른 유닛들과 함께 예측 프로세싱 유닛 (154) 은 전술된 예들을 구현할 수도 있다. 일부 예들에서, (도 12 에 도시되지 않은) 비디오 인코더 (20) 의 유닛 또는 프로세서는 단독으로 또는 비디오 인코더 (20) 의 다른 유닛들과 함께 전술된 예들을 구현할 수도 있다.

[0284] 도 13 은 본 개시물에 설명된 기법들을 구현할 수도 있는 비디오 디코더의 예를 나타내는 블록도이다. 예를 들어, 비디오 디코더 (30) 는 인터-예측 디코딩 또는 인트라-예측 디코딩을 수행할 수도 있다. 도 13 은 비디오 디코더 (30) 를 예시한다. 도 13 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (180), 예측 프로세싱 유닛 (181), 역양자화 프로세싱 유닛 (186), 역변환 프로세싱 유닛 (188), 합산기 (190), 및 레퍼런스 픽처 메모리 (192) 를 포함한다. 예측 프로세싱 유닛 (181) 은 모션 보상 유닛 (182) 및 인트라 예측 유닛 (184) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서 도 12 로부터의 비디오 인코더 (20) 에 대하여 설명된 인코딩 과정에 일반적으로 상반되는 (reciprocal) 디코딩 과정을 수행할 수도 있다. 레퍼런스 픽처 메모리 (192) 는 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 의 일 예이다.

[0285] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (180) 은 그 비트스트림을 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (180) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 예측 프로세싱 유닛 (181) 에 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.

[0286] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩되는 경우, 예측 프로세싱 유닛 (181) 의 인트라 예측 프로세싱 유닛 (184) 은 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 현재 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 시그널링된 인트라 예측 모드 및 데이터에 기초하여 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B 또는 P) 슬라이스로서 코딩되는 경우, 예측 프로세싱 유닛 (181) 의 모션 보상 유닛 (182) 은 엔트로피 디코딩 유닛 (180) 으로부터 수신된 일시적인 모션 벡터들, 다른 신택스 엘리먼트들에 기초하여, 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 그 예측 블록들은 레퍼런스 픽처 리스트들 중 하나 내의 레퍼런스 픽처들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는, 레퍼런스 픽처 메모리 (192) 에 저장된 레퍼런스 픽처들에 기초하여 디풀트 구성 기법들을 사용하여, 레퍼런스 픽처 리스트들 (RefPicList0 및 RefPicList1) 을 구성할 수도 있다.

[0287] 모션 보상 유닛 (182) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 포함함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (182) 은 수신된 신택스 엘리먼트들의 일부를 사용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예컨대, 인트라 또는 인터 예측), 인터 예측 슬라이스 유형 (예컨대, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 레퍼런스 픽처 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터 코딩된 비디오 블록에 대한 인터 예측 스테이터스, 및 현재 비디오 슬라이스 내의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

- [0288] 모션 보상 유닛 (182) 은 또한 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (182) 은 비디오 블록들의 인코딩 동안에 비디오 인코더 (20) 에 의해 사용된 것과 같은 보간 필터들을 사용하여 레퍼런스 블록들의 부 정수 화소들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (182) 은 비디오 인코더 (20) 에 의해 사용된 보간 필터들을 수신된 신팩스 엘리먼트들로부터 결정하고 그 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.
- [0289] 역 양자화 프로세싱 유닛 (186) 은 비트스트림으로 제공되고 엔트로피 디코딩 유닛 (180) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉, 디퀀타이즈 (dequantize) 한다. 역 양자화 프로세스는 양자화 정도와, 마찬가지로 적용되어야 할 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다. 역 변환 프로세싱 유닛 (188) 은 화소 도메인에서 잔여 블록들을 생성하기 위해 역 변환 (예컨대, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환) 프로세스를 변환 계수들에 적용한다.
- [0290] 모션 보상 유닛 (182) 이 현재 비디오 블록에 대한 예측 블록을 모션 벡터들 및 다른 신팩스 엘리먼트들에 기초하여 생성한 후, 비디오 디코더 (30) 는 역 변환 프로세싱 유닛 (188) 으로부터의 잔여 블록들을 모션 보상 유닛 (182) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (90) 는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 원한다면, 블록화제거 필터가 또한 블록현상 아티팩트들을 제거하기 위해 디코딩된 블록들을 필터링하는데 적용될 수도 있다. 다른 루프 필터들이 (코딩 루프 내 또는 코딩 루프 후 중 어느 하나에서) 또한 화소 전환들을 부드럽게 하기 위해, 또는 그렇지 않으면 비디오 품질을 개선하기 위해 사용될 수도 있다. 주어진 픽처에서의 디코딩된 비디오 블록들은 그 후, 레퍼런스 픽처 메모리 (192) 에 저장되며, 그 레퍼런스 픽처 메모리는 후속 모션 보상을 위해 사용되는 레퍼런스 픽처들을 저장한다. 레퍼런스 픽처 메모리 (192) 는 또한 디스플레이 디바이스, 예컨대 도 1 의 디스플레이 디바이스 (32) 상의 나중의 프레젠테이션을 위해 디코딩된 비디오를 저장한다.
- [0291] 이 방식에서, 비디오 디코더 (30) 는 본 개시물에 설명된 하나 이상의 예시의 기법들을 구현하도록 구성되는 비디오 디코더의 일 예이다. 예를 들어, 예측 프로세싱 유닛 (181) 은 공간적 및 시간적으로 이웃하는 구역들의 디스패리티 모션 벡터들에 기초하여 현재 블록에 대한 디스패리티 벡터를 결정하도록 구성될 수도 있다. 예측 프로세싱 유닛 (181) 은 또한, 후보 모션 벡터 예측자들의 리스트에 대한 후보 모션 벡터 예측자들을 결정하기 위해 결정된 디스패리티 벡터를 이용하도록 구성될 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (181) 은 결정된 디스패리티 벡터를 변형할 수도 있고, 이 변형된, 결정된 디스패리티 벡터를 후보 모션 벡터 예측자들의 리스트에 포함할 수도 있다. 예측 프로세싱 유닛 (181) 은 또한, 인터-뷰 모션 예측 및 인터-뷰 잔여 예측을 수행하도록 구성될 수도 있다.
- [0292] 일부 예들에서, 예측 프로세싱 유닛 (181) 외의 유닛은 전술된 예들을 구현할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 의 하나 이상의 다른 유닛들과 함께 예측 프로세싱 유닛 (181) 은 전술된 예들을 구현할 수도 있다. 또 다른 일부 예들에서, (도 13 에 도시되지 않은) 비디오 디코더 (30) 의 유닛 또는 프로세서는 단독으로 또는 비디오 디코더 (30) 의 다른 유닛들과 함께 전술된 예들을 구현할 수도 있다.
- [0293] 도 14 는 본 개시물에 설명된 하나 이상의 기법들에 따른 비디오 디코더의 예시의 동작을 나타내는 플로우차트이다. 비디오 디코더의 일 예는 비디오 디코더 (30) 이다. 도 14 에 예시된 바와 같이, 비디오 디코더 (30) 는 구역에 대한 모션 벡터가 디스패리티 모션 벡터인지 여부를 결정할 수도 있다 (194). 예를 들어, 구역은 하나 이상의 블록들을 포함할 수도 있고, 복수의 뷰들의 제 1 뷰에서의 픽처 내에 있을 수도 있다. 구역에 대한 모션 벡터는 하나의 레퍼런스 픽처 리스트 (즉, RefPicList0 또는 RefPicList1) 에 대한 구역 내의 하나 이상의 블록에 대한 단일의 모션 벡터를 나타낼 수도 있다. 간략함을 위해, 이 기법들은 레퍼런스 픽처 리스트들 중 하나에 대한 것인 구역에 대한 모션 벡터로 설명된다. 일부 경우들에서, 구역에 대해 2 개의 모션 벡터들이 존재할 수도 있다. 이들 예들에서, 하나의 레퍼런스 픽처 리스트에 대한 구역 내의 블록들 모두에 대해 단일의 모션 벡터, 및 다른 레퍼런스 픽처 리스트에 대한 구역 내의 블록들 모두에 대해 단일의 모션 벡터가 존재할 수도 있다.
- [0294] 구역은 현재 블록을 이웃 (예를 들어, 공간적으로 이웃, 시간적으로 이웃) 할 수도 있다. 구역은 또한, 현재 블록을 공간적 또는 시간적으로 이웃하는 블록을 공간적 또는 시간적으로 이웃할 수도 있다. 또한, 모션 벡터는, 모션 벡터가 제 2 의 상이한 뷰에서의 (예를 들어, 구역을 갖는 픽처를 포함하는 뷰와 상이한 뷰에서의) 픽처를 지칭하는 경우, 디스패리티 모션 벡터일 수도 있다.
- [0295] 픽처에서의 구역에 대한 모션 벡터가 디스패리티 모션 벡터라고 결정하는 것에 응답하여, 비디오 디코더 (30)

는, 구역에 대한 디스파리티 모션 벡터가 현재 블록에 대한 디스파리티 벡터인지 여부를 결정할 수도 있다 (196). 비디오 디코더 (30) 가, 구역에 대한 디스파리티 모션 벡터가 현재 블록에 대한 디스파리티 벡터인지 여부를 결정할 수도 있는 다양한 방식들이 존재할 수도 있다. 일 예로써, 비디오 디코더 (30) 는 공간적 디스파리티 벡터들 (spatial disparity vectors; SDVs) 을 결정하기 위해 공간적으로 이웃하는 구역들을 체크하고, 시간적 디스파리티 벡터들 (temporal disparity vectors; TDVs) 을 결정하기 위해 시간적으로 이웃하는 구역들을 체크하고, 암시적 디스파리티 벡터들 (implicit disparity vectors; IDVs) 을 결정하기 위해 공간적/시간적으로 이웃하는 블록들의 공간적/시간적으로 이웃하는 구역들을 체크할 수도 있다. 비디오 디코더 (30) 는 SDVs, TDVs, 및 IDVs 을 포함하는 후보 디스파리티 벡터들의 리스트를 구성할 수도 있고, 현재 블록에 대한 디스파리티 벡터로서 후보 디스파리티 벡터들 중 하나를 선택할 수도 있다. 다른 예로써, 비디오 디코더 (30) 는 현재 블록에 대한 디스파리티 벡터로서 제 1 디스파리티 모션 벡터를 선택할 수도 있다. 비디오 디코더 (30) 가, 구역에 대한 디스파리티 모션 벡터가 현재 블록에 대한 디스파리티 벡터인지 여부를 결정하는 다른 예시의 방식들이 존재할 수도 있다.

[0296] 비디오 디코더 (30) 는 현재 블록을 인트라-예측 디코딩한다 (198). 예를 들어, 비디오 디코더 (30) 는, 구역에 대한 디스파리티 모션 벡터가 현재 블록에 대한 디스파리티 벡터라고 결정되는 경우 현재 블록에 대한 디스파리티 벡터로서 디스파리티 모션 벡터를 사용하여 현재 블록을 인터-예측 디코딩한다. 인터-예측 디코드는 비디오 디코더 (30) 에 의해 수행된 인터-예측을 지칭한다. 예를 들어, 비디오 디코더 (30) 는 디스파리티 벡터가 지칭하는 블록을 결정할 수도 있다. 비디오 디코더 (30) 는 디스파리티 벡터가 지칭하는 블록에 대한 모션 벡터를 결정하고, 디스파리티 벡터가 후보 모션 벡터 예측자들의 리스트에서 가리키는 블록에 대한 모션 벡터를 포함할 수도 있다. 이 예에서, 비디오 디코더 (30) 는, 머지 모드 또는 어드밴스드 모션 벡터 예측 (AMVP) 모드의 일부로서 디스파리티 벡터가 가리키는 블록에 대한 모션 벡터가 후보 모션 벡터 예측자들의 리스트로부터 선택되는 경우 디스파리티 벡터가 가리키는 블록에 대한 모션 벡터에 기초하여 현재 블록을 인터-예측 디코딩할 수도 있다.

[0297] 도 15 는 본 개시물에 설명된 하나 이상의 기법들에 따른 비디오 인코더의 예시의 동작을 나타내는 플로우차트이다. 비디오 인코더의 일 예는 비디오 인코더 (20) 이다. 일반적으로, 비디오 인코더 (20) 는 현재 블록에 대한 디스파리티 벡터를 결정하는 것에 대하여, 비디오 디코더 (30) 에 대해 도 14 에서 전술된 것과 같은 유사한 기법들을 수행할 수도 있다.

[0298] 예를 들어, 비디오 인코더 (20) 는 구역에 대한 모션 벡터가 디스파리티 모션 벡터인지 여부를 결정할 수도 있다 (200). 비디오 인코더 (20) 는, 도 14 의 비디오 디코더 (30) 에 대하여 전술된 것과 유사한 방식으로 구역에 대한 모션 벡터가 디스파리티 모션 벡터인지 여부를 결정할 수도 있다. 꾹처에서의 구역에 대한 모션 벡터가 디스파리티 모션 벡터라고 결정하는 것에 응답하여, 비디오 인코더 (20) 는 구역에 대한 디스파리티 모션 벡터가 현재 블록에 대한 디스파리티 벡터인지 여부를 결정한다 (202). 비디오 인코더 (20) 가, 구역에 대한 디스파리티 모션 벡터가 현재 블록에 대한 디스파리티 벡터인지 여부를 결정되는 경우 현재 블록에 대한 디스파리티 벡터를 사용하여 인터-예측 인코딩한다 (204). 예를 들어, 비디오 인코더 (20) 는, 구역에 대한 디스파리티 모션 벡터가 현재 블록에 대한 디스파리티 벡터라고 결정되는 경우 현재 블록에 대한 디스파리티 벡터로서 디스파리티 모션 벡터를 사용하여 인터-예측 인코딩한다 (204). 인터-예측 인코드는 비디오 인코더 (20) 에 의해 수행된 인터-예측을 지칭하고, 비디오 인코더 (20) 는 비디오 디코더 (30) 와 유사한 방식으로 현재 블록을 인터-예측 인코딩할 수도 있으나, 비디오 데이터를 인코딩하기 위한 것이다. 전술된 바와 같이, 비디오 디코더 (30) 는 비디오 데이터를 디코딩하기 위해 인터-예측 디코딩을 수행할 수도 있다.

[0299] 도 16 은 본 개시물에 설명된 하나 이상의 기법들에 따른 비디오 코더의 예시의 동작을 나타내는 플로우차트이다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 도 16 에 나타낸 예시의 기법들을 구현하도록 구성될 수도 있다.

[0300] 예를 들어, 비디오 코더는, 공간적 또는 시간적으로 이웃하는 구역에 대한 모션 벡터가 디스파리티 모션 벡터인지 여부를 결정한다 (206). 공간적으로 이웃하는 구역 (예를 들어, 2 개의 4x4 블록들을 포함하는 8x4 구역) 은 예측될 블록과 동일한 꾹처 내에 있다. 시간적으로 이웃하는 구역 (예를 들어, 16 개의 4x4 블록들을 포함하는 16x16 구역) 은 예측될 블록을 포함하는 꾹처 외의 꾹처에 있다. 일부 예들에서, 비디오 코더는, 공간적 또는 시간적으로 이웃하는 블록과 공간적 또는 시간적으로 이웃하는 구역에 대한 모션 벡터가 디스파리티 모션 벡터인지 여부를 결정할 수도 있다. 구역에 대한 모션 벡터는 그 구역 내에 있는 복수의 블

록들에 대한 단일의 모션 벡터를 나타낸다.

[0301] 또한, 시간적으로 이웃하는 구역들에 대해, 비디오 코더는 현재 블록에 대한 레퍼런스 픽처 리스트들 (즉, 현재 블록을 포함하는 픽처에 대한 구성된 레퍼런스 픽처 리스트들) 중 하나 또는 양자에서 각 픽처를 평가할 수도 있다. 각각의 픽처에 대해, 비디오 코더는, 픽처가 디스패리티 모션 벡터로 인터-예측되는 시간적으로 이웃하는 구역을 포함하는지 여부를 결정할 수도 있다. 일부 예들에서, 비디오 코더는 레퍼런스 픽처 리스트에서의 각각의 픽처를 오름 차순으로 평가한다. 또한, 비디오 코더는 collocated_from_10_flag 값에 기초하여 먼저 RefPicList0 또는 RefPicList1에서의 픽처들을 평가한다. 다른 예들에서, 비디오 코더는 collocated_from_10_flag의 값에 관계없이, 먼저 RefPicList0, 그리고 그 후 RefPicList1에서의 픽처를 평가하고, 또는 그 반대이다.

[0302] 도 16의 예에서, 비디오 코더는 후보 디스패리티 벡터들의 리스트에서 디스패리티 모션을 포함한다 (208). 비디오 코더가 후보 디스패리티 벡터들의 리스트로부터 이 디스패리티 모션 벡터를 선택하면, 비디오 코더는 디스패리티 모션 벡터로 지정된 블록의 모션 벡터를 결정한다 (210). 이 예에서, 비디오 코더는 현재 블록에 대한 후보 모션 벡터 예측자들의 리스트에서 결정된 모션 벡터를 포함한다 (212). 일부 예들에서, 비디오 코더는 또한, 후보 모션 벡터 예측자들의 리스트에서 디스패리티 모션 벡터의 변경된 버전 (예를 들어, y-컴포넌트가 0과 동일하게 설정된 디스패리티 모션 벡터)을 포함한다 (214).

[0303] 일부 경우들에서, 비디오 코더는 후보 모션 벡터 예측자들의 리스트 안의 인덱스를 나타내는 코딩된 신팩스 엘리먼트에 기초하여 후보 모션 벡터 예측자를 선택한다 (216). 비디오 코더는 그 후, 어드밴스드 모션 벡터 예측 (AMVP) 모드 또는 머지 모드를 사용하여 현재 블록을 인터-예측한다 (218).

[0304] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 관독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 관독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 관독가능 저장 매체들, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이 방식으로, 컴퓨터 관독가능 매체들은 일반적으로 (1) 비일시적 (non-transitory) 인 유형의 (tangible) 컴퓨터 관독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 관독가능 매체를 포함할 수도 있다.

[0305] 비제한적인 예로, 이러한 컴퓨터 관독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 관독가능 매체로 적절히 정해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 예컨대 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다. 그러나, 컴퓨터 관독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비일시적 (non-transient), 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc)는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들로써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 관독가능 매체들의 범위 내에 포함되어야 한다.

[0306] 명령들은 하나 이상의 프로세서들, 예컨대 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서"는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 또한, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec)으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들

은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

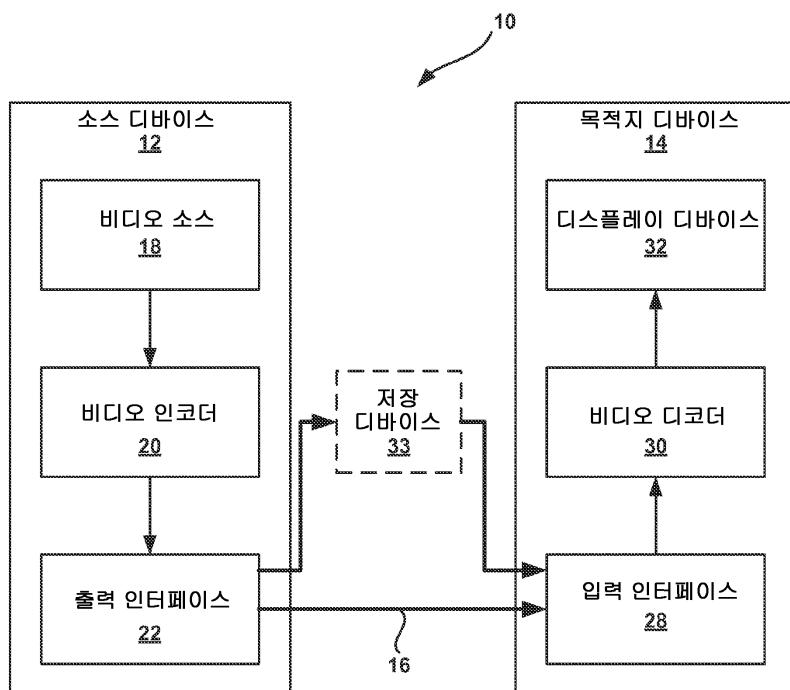
[0307] 본 개시물의 기법들은 무선 핸드셋, 접적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트) 을 포함하는, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 전술된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 전술된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

[0308] 일부 예들에서, 기법들은 비디오 디코더를 포함하는 디바이스 또는 비디오 인코더를 포함하는 디바이스로 설명된다. 이 디바이스는 무선 통신 디바이스와 같은 독립형 디바이스일 수도 있다. 일부 예들에서, 디바이스는 마이크로프로세서 또는 접적 회로일 수도 있다. 이를 예들에서, 마이크로프로세서 또는 접적 회로는 적용 가능하다면 비디오 디코더 또는 비디오 인코더를 포함할 수도 있다. 일부 예들에서, 비디오 디코더 및 비디오 인코더는 독립형 마이크로프로세서들 또는 접적 회로들일 수도 있다.

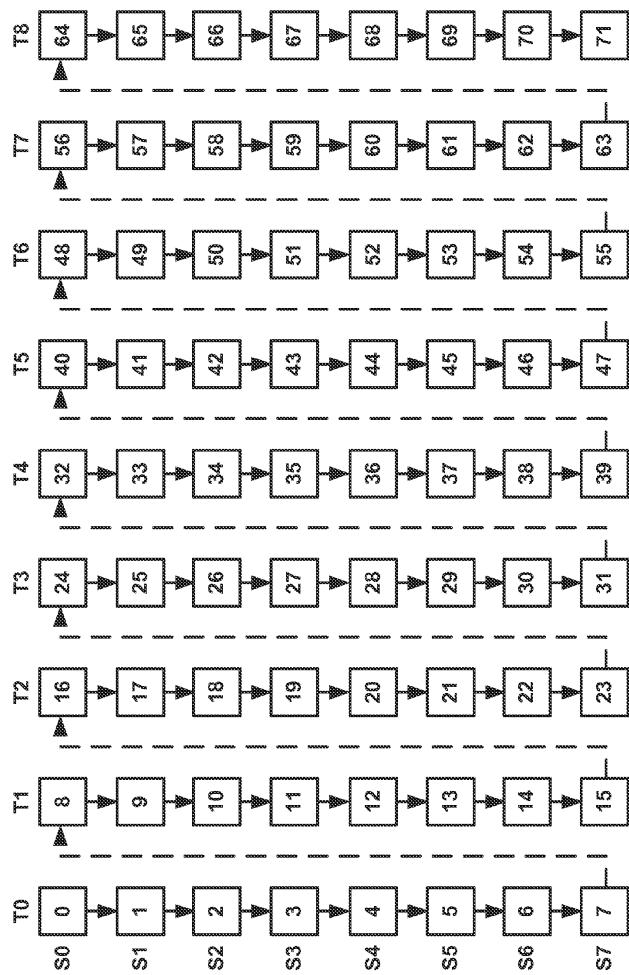
[0309] 다양한 예들이 설명되어 있다. 이를 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

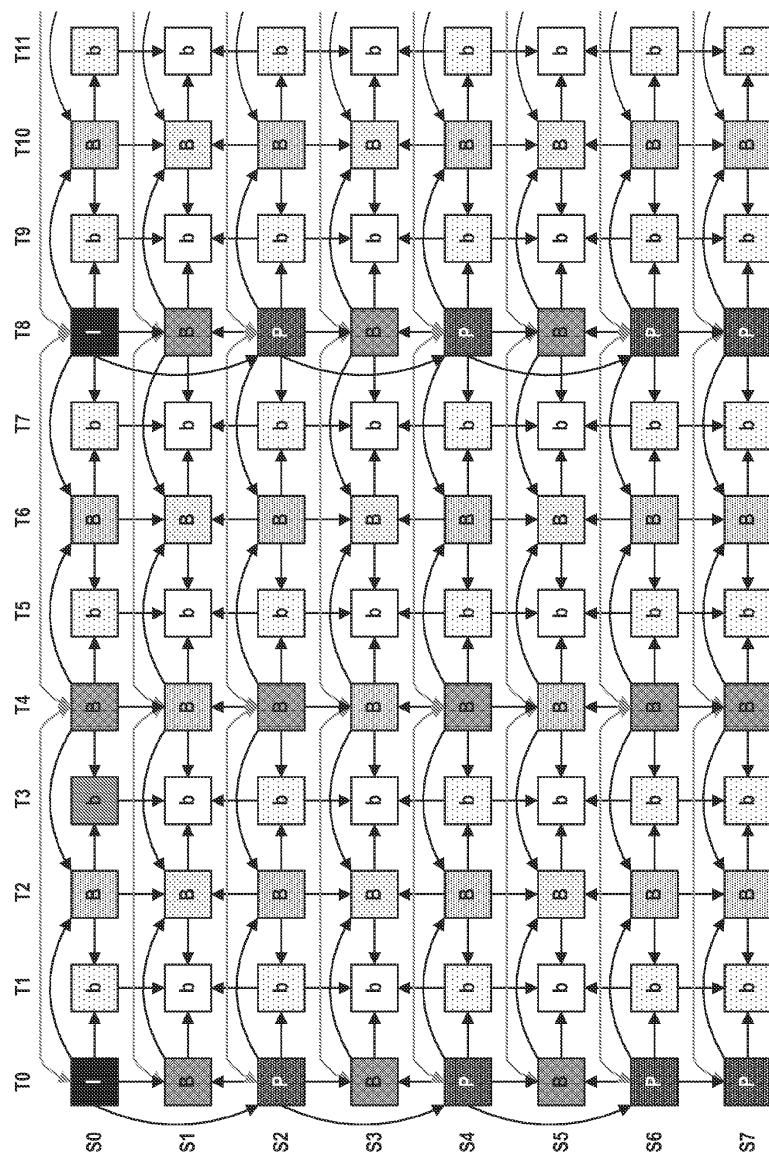
도면1



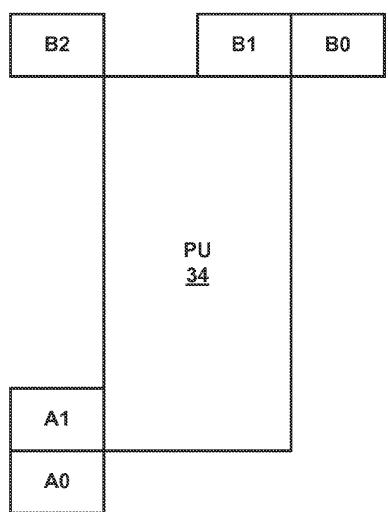
도면2



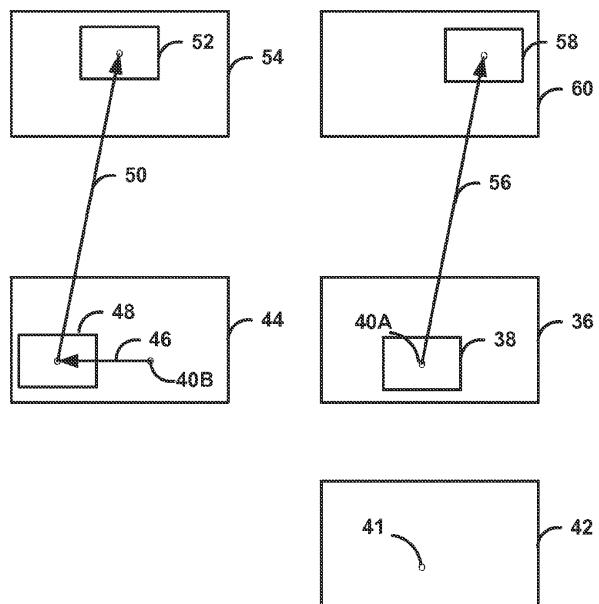
도면3



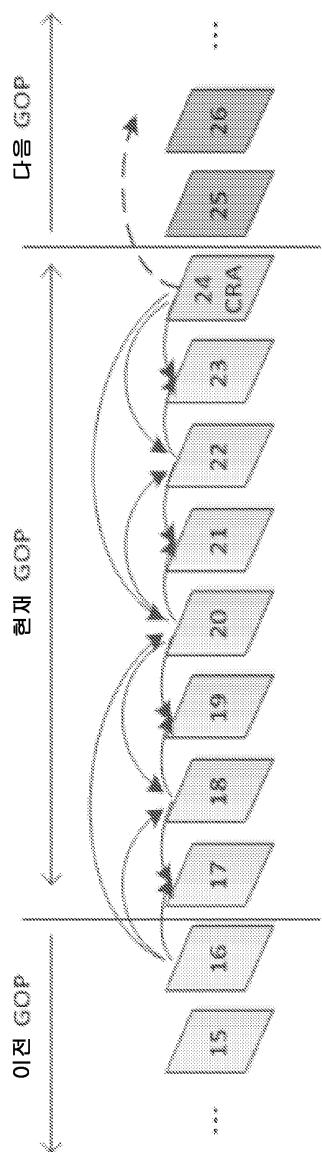
도면4



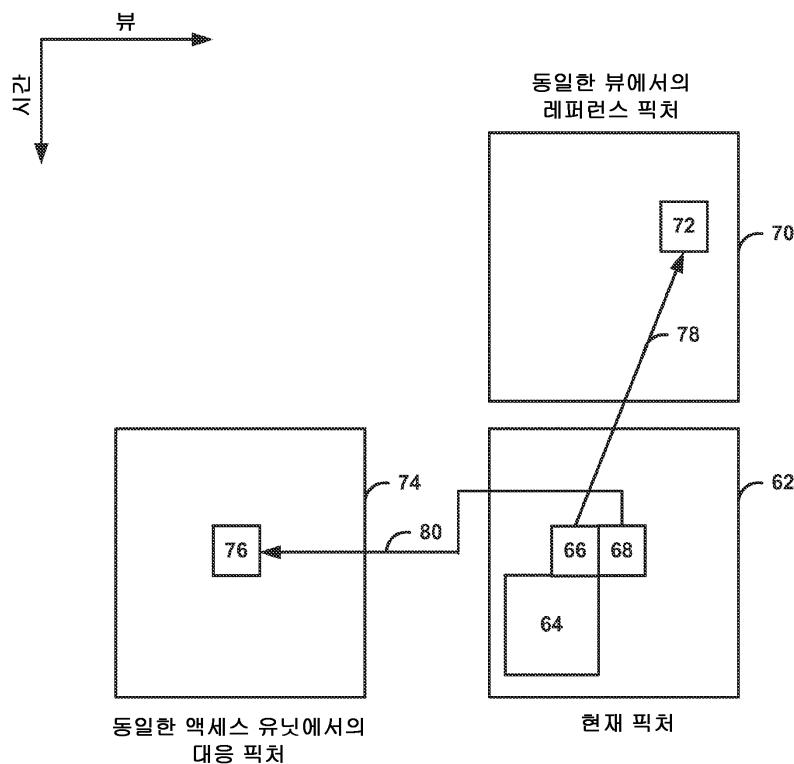
도면5



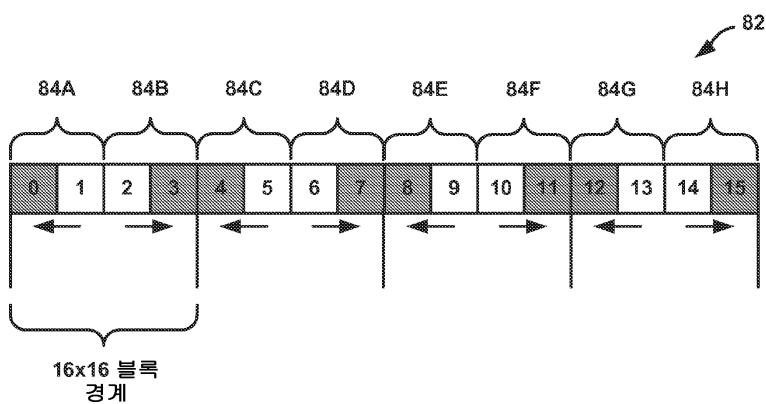
도면6



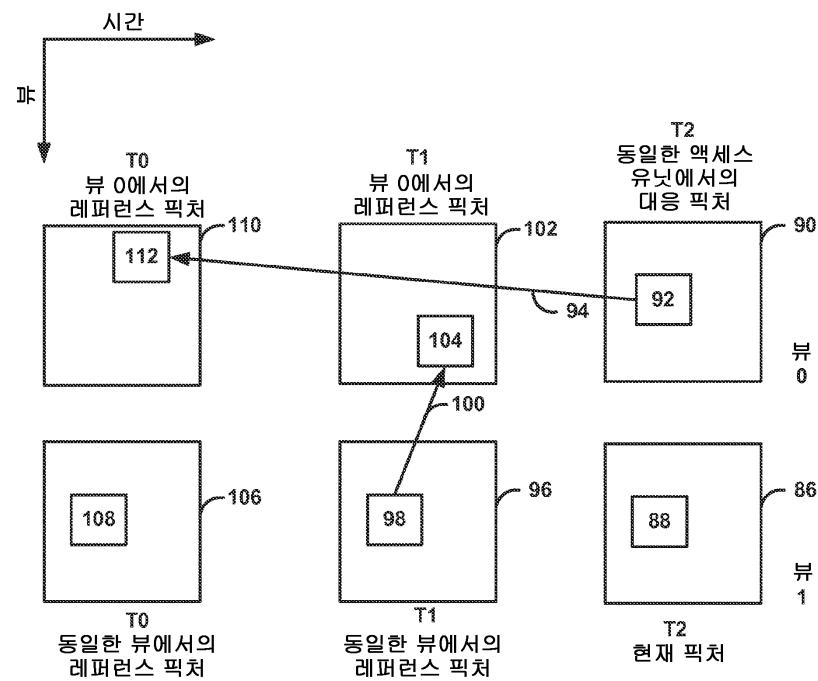
도면7



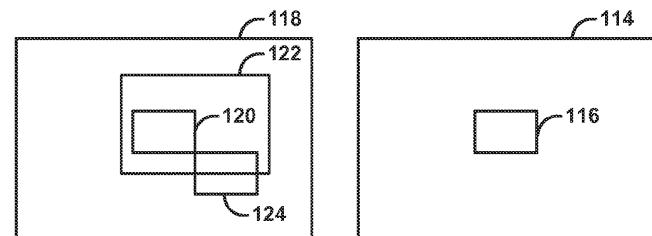
도면8



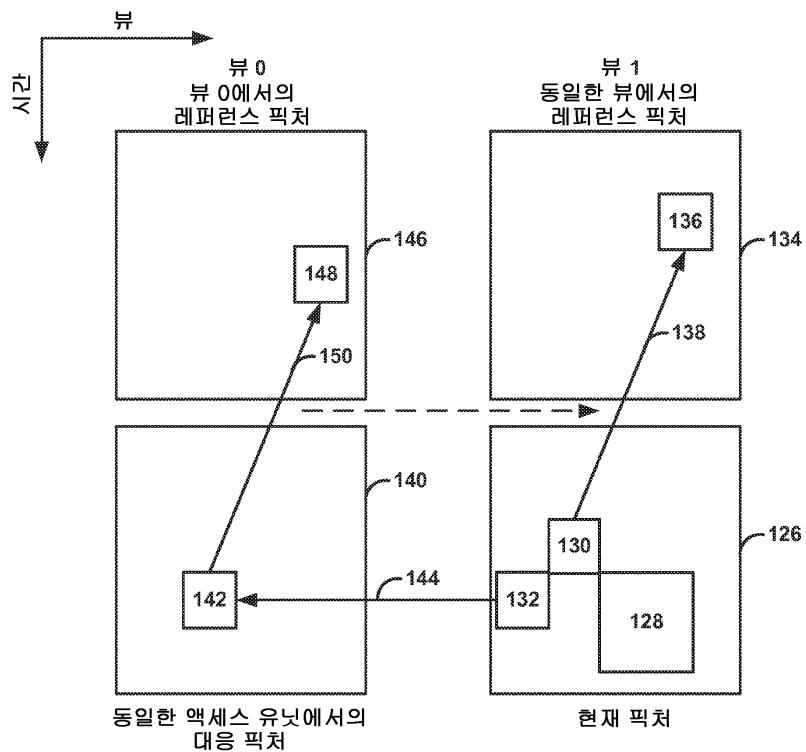
도면9



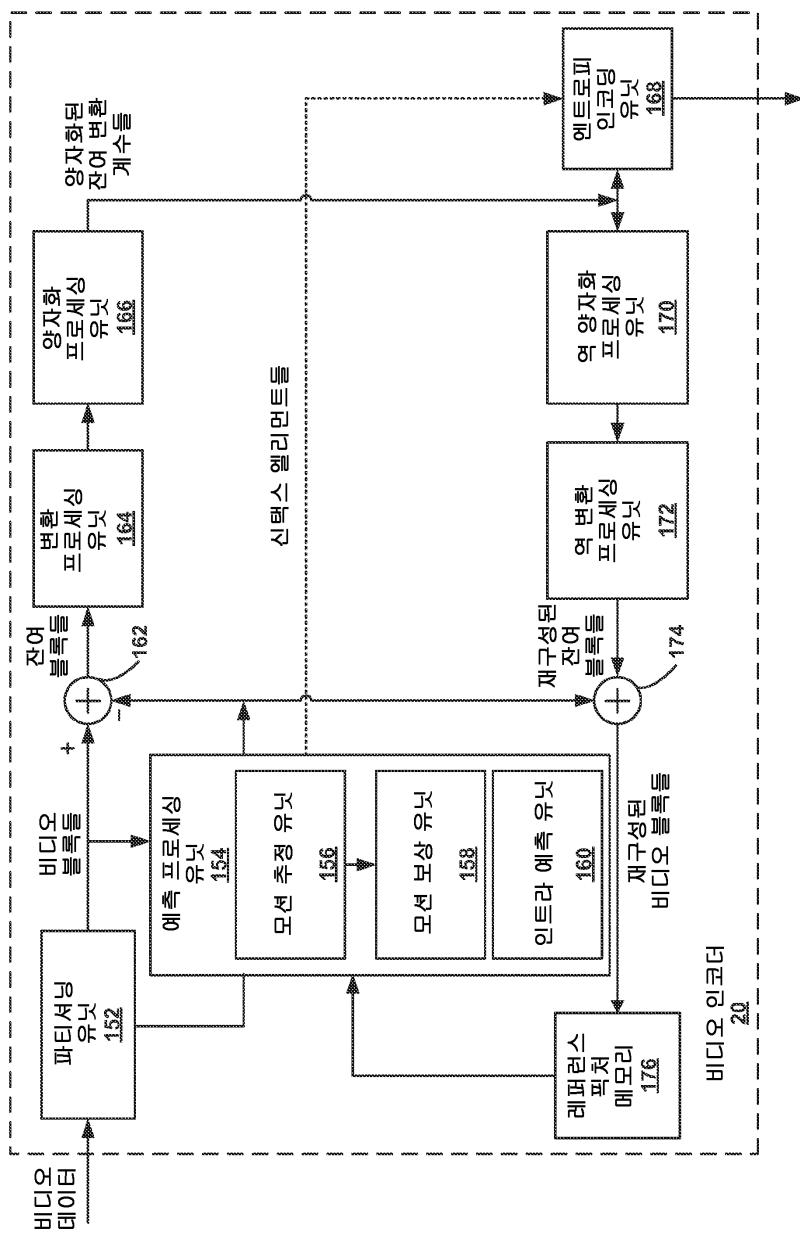
도면10



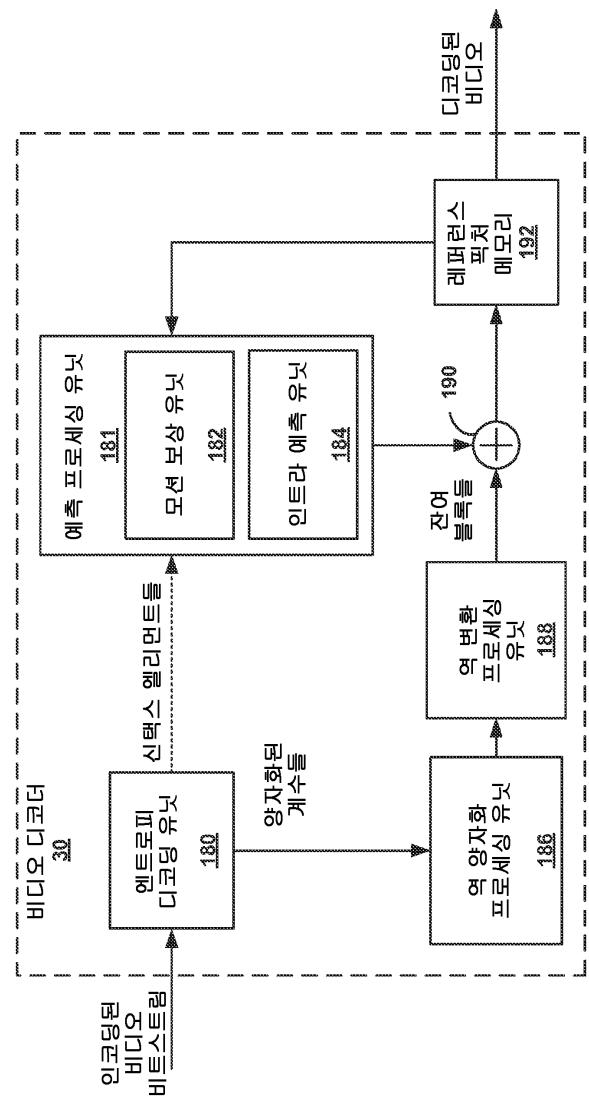
도면11



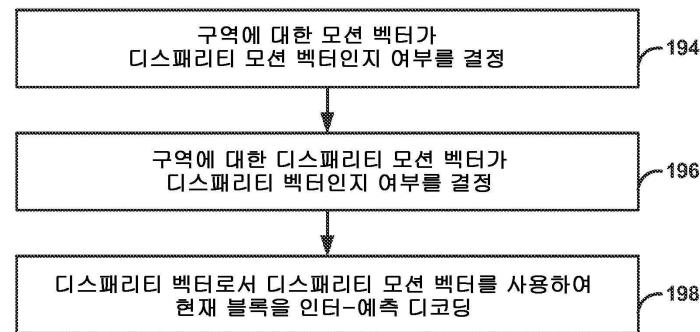
도면12



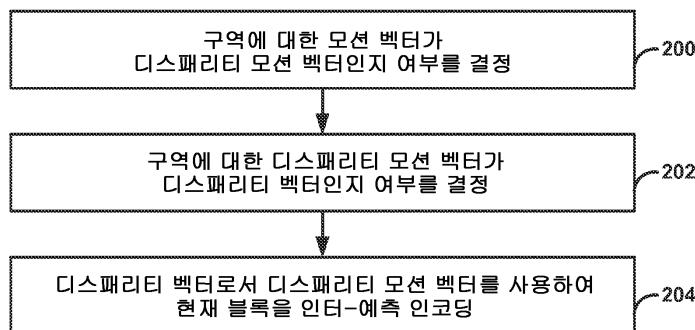
도면13



도면14



도면15



도면16

