(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년06월20일
(11) 등록번호 10-1749437
(24) 등록일자 2017년06월14일

(51) 국제특허분류(Int. Cl.)
H04N 19/597 (2014.01) *H04N 19/105* (2014.01)
H04N 19/156 (2014.01) *H04N 19/176* (2014.01)
H04N 19/52 (2014.01)
(52) CPC특허분류
H04N 19/597 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2015-7018363
(22) 출원일자(국제) 2013년12월13일
심사청구일자 2016년08월12일
(85) 번역문제출일자 2015년07월08일
(65) 공개번호 10-2015-0097591
(43) 공개일자 2015년08월26일
(86) 국제출원번호 PCT/US2013/074981
(87) 국제공개번호 WO 2014/093805
국제공개일자 2014년06월19일
(30) 우선권주장
61/737,639 2012년12월14일 미국(US)
14/104,095 2013년12월12일 미국(US)

(56) 선행기술조사문헌
3D-HEVC Test Model 1, The Joint Collaborative Team on 3D video coding extension development, JCT3V-A1005 (2012.09.20.)
3D-CE5.h related: Disparity vector derivation for multiview video and 3DV, Mpeg Meeting, no. m24937 (2012.05.01.)

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
강 제원
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
천 잉
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 19 항 심사관 : 장석환

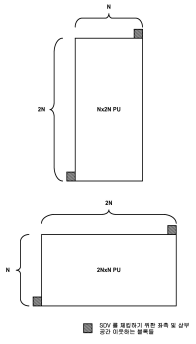
(54) 발명의 명칭 **디스패리티 벡터 유도**

(57) 요 약

디바이스는 현재의 블록에 대한 디스패리티 벡터를 결정하기 위해 디스패리티 벡터 유도 프로세스를 수행한다.
디스패리티 벡터 유도 프로세스를 수행하는 것의 일부로서, 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 디바이스는 디스패리티 모션 벡터 또는 암시적인 디스패리

(뒷면에 계속)

대 표 도 - 도13



티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환한다. 디스패리티 벡터 유도 프로세스에서 체킹되는 이웃하는 블록들의 개수가 감소되어, 잠재적으로 감소된 복잡성 및 메모리 대역폭 요구사항들을 초래한다.

(52) CPC특허분류
H04N 19/156 (2015.01)
H04N 19/176 (2015.01)
H04N 19/52 (2015.01)

(72) 발명자
장 리
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명 세 서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

하나 이상의 프로세서들 또는 회로들에 의해, 상기 비디오 데이터의 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하는 단계로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 단계는:

제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 상기 하나 이상의 프로세서들 또는 회로들에 의해, 상기 디스패리티 모션 벡터 또는 상기 암시적인 디스패리티 벡터를 상기 현재의 블록에 대한 디스패리티 벡터로 변환하는 단계를 포함하며,

상기 디스패리티 벡터 유도 프로세스는 설령 상기 제 1 또는 상기 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 상기 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 디스패리티 벡터 유도 프로세스를 수행하는 단계; 및

상기 하나 이상의 프로세서들 또는 회로들에 의해, 상기 현재의 블록에 대한 인터-뷰 모션 예측을 부분적으로 수행함으로써, 상기 현재의 블록에 대한 샘플 블록을 재구성하는 단계로서, 상기 현재의 블록은 제 1 뷰에 있고, 인터-뷰 모션 예측을 수행하는 것은:

상기 하나 이상의 프로세서들 또는 회로들에 의해, 상기 유도된 디스패리티 벡터에 기초하여, 상기 제 1 뷰와는 상이한 제 2 뷰에서의 대응하는 블록을 식별하는 것; 및

상기 하나 이상의 프로세서들 또는 회로들에 의해, 리스트의 후보로서 상기 대응하는 블록의 모션 벡터들을 사용하는 것으로서, 상기 리스트는 상기 현재의 블록에 대한 병합 리스트 및 상기 현재의 블록에 대한 진보된 모션 벡터 예측 (AMVP) 후보 리스트로 이루어진 그룹 내에 있는, 상기 대응하는 블록의 모션 벡터들을 사용하는 것을 포함하는, 상기 현재의 블록에 대한 샘플 블록을 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 공간 이웃하는 블록은 상기 현재의 블록의 상부 에지에 인접하며, 상기 제 2 공간 이웃하는 블록은 상기 현재의 블록의 좌측 에지에 인접한, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 디스패리티 벡터 유도 프로세스를 수행하는 단계는, 상기 하나 이상의 프로세서들 또는 회로들에 의해, 디스패리티 모션 벡터에 대해 상기 제 1 공간 이웃하는 블록을 체킹하고 그후 디스패리티 모션 벡터에 대해 상기 제 2 공간 이웃하는 블록을 체킹하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 2 항에 있어서,

상기 디스패리티 벡터 유도 프로세스를 수행하는 단계는, 상기 하나 이상의 프로세서들 또는 회로들에 의해, 디스패리티 모션 벡터에 대해 상기 제 2 공간 이웃하는 블록을 체킹하고 그후 디스패리티 모션 벡터에 대해 상기 제 1 공간 이웃하는 블록을 체킹하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

비디오 데이터를 인코딩하는 방법으로서,

하나 이상의 프로세서들 또는 회로들에 의해, 상기 비디오 데이터의 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하는 단계로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 단계는: 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 상기 디스패리티 모션 벡터 또는 상기 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 단계를 포함하며,

상기 디스패리티 벡터 유도 프로세스는 설령 상기 제 1 또는 상기 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 상기 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 디스패리티 벡터 유도 프로세스를 수행하는 단계; 및

상기 하나 이상의 프로세서들 또는 회로들에 의해, 상기 현재의 블록의 인코딩된 표현을 발생하는 단계로서, 상기 현재의 블록은 제 1 뷰에 있고, 상기 현재의 블록의 상기 인코딩된 표현을 발생하는 단계는 인터-뷰 모션 예측을 수행하는 단계를 포함하고, 상기 인터-뷰 모션 예측을 수행하는 단계는:

상기 하나 이상의 프로세서들 또는 회로들에 의해, 상기 유도된 디스패리티 벡터에 기초하여, 상기 제 1 뷰와는 상이한 제 2 뷰에서의 대응하는 블록을 식별하는 단계; 및

상기 하나 이상의 프로세서들 또는 회로들에 의해, 리스트의 후보로서 상기 대응하는 블록의 모션 벡터들을 사용하는 단계로서, 상기 리스트는 상기 현재의 블록에 대한 병합 리스트 및 상기 현재의 블록에 대한 진보된 모션 벡터 예측 (AMVP) 후보 리스트로 이루어진 그룹 내에 있는, 상기 대응하는 블록의 모션 벡터들을 사용하는 단계를 포함하는, 상기 현재의 블록의 인코딩된 표현을 발생하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 6

제 5 항에 있어서,

상기 제 1 공간 이웃하는 블록은 상기 현재의 블록의 상부 에지에 인접하며, 상기 제 2 공간 이웃하는 블록은 상기 현재의 블록의 좌측 에지에 인접한, 비디오 데이터를 인코딩하는 방법.

청구항 7

제 6 항에 있어서,

상기 디스패리티 벡터 유도 프로세스를 수행하는 단계는, 상기 하나 이상의 프로세서들 또는 회로들에 의해, 디스패리티 모션 벡터에 대해 상기 제 1 공간 이웃하는 블록을 체킹하고 그후 디스패리티 모션 벡터에 대해 상기 제 2 공간 이웃하는 블록을 체킹하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 8

제 6 항에 있어서,

상기 디스패리티 벡터 유도 프로세스를 수행하는 단계는, 상기 하나 이상의 프로세서들 또는 회로들에 의해, 디스패리티 모션 벡터에 대해 상기 제 2 공간 이웃하는 블록을 체킹하고 그후 디스패리티 모션 벡터에 대해 상기 제 1 공간 이웃하는 블록을 체킹하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 9

비디오 데이터를 인코딩하거나 디코딩하는 디바이스로서,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은:

상기 비디오 데이터의 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하는 것으로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은:

제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 상기 디스패리티 모션 벡터 또는 상기 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며,

상기 디스패리티 벡터 유도 프로세스는 설령 상기 제 1 또는 상기 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 상기 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 디스패리티 벡터 유도 프로세스를 수행하고; 및

상기 현재의 블록에 대한 인터-뷰 모션 예측을 부분적으로 수행함으로써, 상기 현재의 블록에 대한 샘플 블록을 재구성하는 것; 또는 상기 현재의 블록에 대한 인터-뷰 모션 예측을 부분적으로 수행함으로써 상기 현재의 블록의 인코딩된 표현을 발생하는 것 중 하나를 수행하도록 구성되고,

상기 현재의 블록은 제 1 뷰에 있고, 상기 하나 이상의 프로세서들은, 상기 현재의 블록에 대한 인터-뷰 모션 예측을 수행하는 것의 부분으로서, 상기 하나 이상의 프로세서들이:

상기 유도된 디스패리티 벡터에 기초하여, 상기 제 1 뷰와는 상이한 제 2 뷰에서의 대응하는 블록을 식별하고; 및

리스트의 후보로서 상기 대응하는 블록의 모션 벡터들을 사용하는 것으로서, 상기 리스트는 상기 현재의 블록에 대한 병합 리스트 및 상기 현재의 블록에 대한 진보된 모션 벡터 예측 (AMVP) 후보 리스트로 이루어진 그룹 내에 있는, 상기 대응하는 블록의 모션 벡터들을 사용하도록

구성되는, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

청구항 10

제 9 항에 있어서,

상기 제 1 공간 이웃하는 블록은 상기 현재의 블록의 상부 에지에 인접하며, 상기 제 2 공간 이웃하는 블록은 상기 현재의 블록의 좌측 에지에 인접한, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

청구항 11

제 10 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 하나 이상의 프로세서들이 상기 디스패리티 벡터 유도 프로세스를 수행할 때, 상기 하나 이상의 프로세서들이 디스패리티 모션 벡터에 대해 상기 제 1 공간 이웃하는 블록을 체킹하고 그 후 디스패리티 모션 벡터에 대해 상기 제 2 공간 이웃하는 블록을 체킹하도록 구성되는, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

청구항 12

제 10 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 하나 이상의 프로세서들이 상기 디스패리티 벡터 유도 프로세스를 수행할 때, 상기 하나 이상의 프로세서들이 디스패리티 모션 벡터에 대해 상기 제 2 공간 이웃하는 블록을 체킹하고 그 후 디스패리티 모션 벡터에 대해 상기 제 1 공간 이웃하는 블록을 체킹하도록 구성되는, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

청구항 13

비디오 데이터를 인코딩하거나 디코딩하는 디바이스로서,

상기 비디오 데이터의 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하는 수단으로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은:

제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 상기 디스패리티 모션 벡터 또는 상기 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며,

상기 디스패리티 벡터 유도 프로세스는 설령 상기 제 1 또는 상기 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 상기 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 디스패리티 벡터 유도 프로세스를 수행하는 수단; 및

인터-뷰 모션 예측을 수행하기 위해 상기 현재의 블록에 대한 디스패리티 벡터를 이용하는 수단; 및

상기 현재의 블록에 대한 인터-뷰 모션 예측을 부분적으로 수행함으로써, 상기 현재의 블록에 대한 샘플 블록을 재구성하는 수단; 또는 상기 현재의 블록에 대한 인터-뷰 모션 예측을 부분적으로 수행함으로써 상기 현재의 블록의 인코딩된 표현을 발생하는 수단 중 하나를 포함하고,

상기 현재의 블록은 제 1 뷰에 있고, 상기 샘플 블록을 재구성하는 수단 또는 상기 현재의 블록의 인코딩된 표현을 발생하는 수단은,

상기 유도된 디스패리티 벡터에 기초하여, 상기 제 1 뷰와는 상이한 제 2 뷰에서의 대응하는 블록을 식별하는 수단; 및

리스트의 후보로서 상기 대응하는 블록의 모션 벡터들을 사용하는 수단으로서, 상기 리스트는 상기 현재의 블록에 대한 병합 리스트 및 상기 현재의 블록에 대한 진보된 모션 벡터 예측 (AMVP) 후보 리스트로 이루어진 그룹 내에 있는, 상기 대응하는 블록의 모션 벡터들을 사용하는 수단을 포함하는, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

청구항 14

제 13 항에 있어서,

상기 제 1 공간 이웃하는 블록은 상기 현재의 블록의 상부 에지에 인접하며, 상기 제 2 공간 이웃하는 블록은 상기 현재의 블록의 좌측 에지에 인접한, 디바이스.

청구항 15

명령들을 저장하고 있는 비일시성 컴퓨터-판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금,

비디오 데이터의 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하게 하는 것으로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은:

제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 상기 디스패리티 모션 벡터 또는 상기 암시적인 디스패리티 벡터를 상기 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며,

상기 디스패리티 벡터 유도 프로세스는 설령 상기 제 1 또는 상기 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 상기 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 디스패리티 벡터 유도 프로세스를 수행하게 하고; 및

상기 현재의 블록에 대한 인터-뷰 모션 예측을 부분적으로 수행함으로써, 상기 현재의 블록에 대한 샘플 블록을 재구성하는 것; 또는 상기 현재의 블록에 대한 인터-뷰 모션 예측을 부분적으로 수행함으로써 상기 현재의 블록의 인코딩된 표현을 발생하는 것 중 하나를 수행하게 하며,

상기 현재의 블록은 제 1 뷰에 있고, 상기 하나 이상의 프로세서들은, 상기 현재의 블록에 대한 인터-뷰 모션 예측을 수행하는 것의 부분으로서, 상기 하나 이상의 프로세서들이:

상기 유도된 디스패리티 벡터에 기초하여, 상기 제 1 뷰와는 상이한 제 2 뷰에서의 대응하는 블록을 식별하고; 및

리스트의 후보로서 상기 대응하는 블록의 모션 벡터들을 사용하는 것으로서, 상기 리스트는 상기 현재의 블록에 대한 병합 리스트 및 상기 현재의 블록에 대한 진보된 모션 벡터 예측 (AMVP) 후보 리스트로 이루어진 그룹 내에 있는, 상기 대응하는 블록의 모션 벡터들을 사용하도록

구성되는, 비일시성 컴퓨터-판독가능 저장 매체.

청구항 16

제 15 항에 있어서,

상기 제 1 공간 이웃하는 블록은 상기 현재의 블록의 상부 에지에 인접하며, 상기 제 2 공간 이웃하는 블록은 상기 현재의 블록의 좌측 에지에 인접한, 비일시성 컴퓨터-판독가능 저장 매체.

청구항 17

제 9 항에 있어서,

상기 디바이스는:

집적회로;

마이크로프로세서; 또는

무선 통신 디바이스

중 적어도 하나를 포함하는, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

청구항 18

제 9 항에 있어서,

디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함하는, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

청구항 19

제 9 항에 있어서,

상기 비디오 데이터를 캡처하도록 구성된 카메라를 더 포함하는, 비디오 데이터를 인코딩하거나 디코딩하는 디바이스.

발명의 설명

기 술 분 야

[0001] 본 출원은 2012년 12월 14일자에 출원된 미국 가특허 출원번호 제 61/737,639호의 이익을 주장하며, 이의 전체 내용이 본원에 참고로 포함된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 인코딩 및 디코딩에 관한 것이다.

배 경 기 술

[0004] 디지털 비디오 능력들은, 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDAs), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트폰들", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), 현재 개발중인 HEVC (High Efficiency Video Coding) 표준, 및 이런 표준들의 확장판들에 의해 정의된 표준들에서 설명되는 비디오 코딩 기법들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이런 비디오 압축 기법들을 구현함으로써, 디지털 비디오 정보를 좀더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 고유한 리던던시를 감소시키거나 또는 제거하기 위해 공간 (인트라-화

상) 예측 및/또는 시간 (인터-화상) 예측을 수행한다. 블록-기반 비디오 코딩에 있어, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 일부) 는 블록들로 파티셔닝될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스에서 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수 있으며, 참조 화상들은 참조 프레임들로서 지칭될 수도 있다.

[0006] 공간 또는 시간 예측은 코딩되는 블록에 대한 예측 블록을 초래한다. 잔여 데이터는 코딩되는 원래 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라서 인코딩되며, 잔여 데이터는 코딩된 블록과 예측 블록 사이의 차이를 나타낸다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔여 데이터에 따라서 인코딩된다. 추가적인 압축을 위해, 잔여 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 그 결과 잔여 계수들이 되고, 그후 양자화될 수도 있다. 2차원 어레이로 처음에 배열된 양자화된 계수들은 계수들의 1차원 벡터를 발생하기 위해 스캐닝될 수도 있으며, 더욱 더 많은 압축을 달성하기 위해 엔트로피 코딩이 적용될 수도 있다.

[0007] 멀티-뷰 코딩 비트스트림은 뷰들을, 예컨대, 다수의 관점들로부터 인코딩함으로써 발생될 수도 있다. 멀티-뷰 코딩 양태들을 이용하는 일부 3차원 (3D) 비디오 표준들이 개발되었다. 예를 들어, 상이한 뷰들은 3D 비디오를 지원하기 위해 좌측 및 우측 눈 뷰들을 송신할 수도 있다. 이의 대안으로, 일부 3D 비디오 코딩 프로세스들은 소위 멀티-뷰 플러스 심도 코딩을 적용할 수도 있다. 멀티-뷰 플러스 심도 코딩에서, 3D 비디오 비트스트림은 텍스쳐 뷰 성분들 뿐만 아니라, 심도 뷰 성분들도 포함할 수도 있다. 예를 들어, 각각의 뷰는 하나의 텍스쳐 뷰 성분 및 하나의 심도 뷰 성분을 포함할 수도 있다.

발명의 내용

과제의 해결 수단

[0008] 일반적으로, 본 개시물은 멀티-뷰 및 3차원 비디오 코딩에 관한 것이다. 특히, 본 개시물은 현재의 블록에 대한 디스패리티 벡터를 결정하는 디스패리티 벡터 유도 프로세스를 기술한다. 디스패리티 벡터 유도 프로세스를 수행하는 것의 일부로서, 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터가 현재의 블록에 대한 디스패리티 벡터로 변환된다. 일부 예들에서, 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는다. 이러한 방법으로, 디스패리티 벡터 유도 프로세스에서 체킹되는 이웃하는 블록들의 개수가 감소되어, 잠재적으로, 감소된 복잡성 및 메모리 대역폭 요구사항들을 초래할 수도 있다.

[0009] 일 예에서, 본 개시물은 비디오 데이터를 디코딩하는 방법을 기술하며, 본 방법은 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하는 단계로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며, 상기 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 수행하는 단계; 및 유도된 디스패리티 벡터에 기초하여, 현재의 블록에 대한 인터-뷰 예측을 부분적으로 수행함으로써 현재의 블록에 대한 샘플 블록을 재구성하는 단계를 포함한다.

[0010] 또 다른 예에서, 본 개시물은 비디오 데이터를 인코딩하는 방법을 기술하며, 본 방법은 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하는 단계로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며, 상기 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 제 1 및 제 2

공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 수행하는 단계; 및 현재의 블록에 대한 유도된 디스패리티 벡터에 부분적으로 기초하여, 현재의 블록의 인코딩된 표현을 발생하는 단계를 포함한다.

[0011] 또 다른 예에서, 본 개시물은, 비디오 데이터를 저장하도록 구성된 메모리; 및 비디오 데이터의 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하도록 구성된 하나 이상의 프로세서들을 포함하며, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며, 상기 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 디바이스를 기술한다.

[0012] 또 다른 예에서, 본 개시물은 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하는 수단으로서, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며, 상기 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는, 상기 수행하는 수단; 및 인터-뷰 예측을 수행하기 위해 현재의 블록에 대한 디스패리티 벡터를 이용하는 수단을 포함하는 디바이스를 기술한다.

[0013] 또 다른 예에서, 본 개시물은 명령들을 안에 저장하고 있는 컴퓨터-판독가능 저장 매체를 기술하며, 상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금, 현재의 블록에 대한 디스패리티 벡터를 유도하기 위해 디스패리티 벡터 유도 프로세스를 수행하도록 하고; 인터-뷰 예측을 수행하기 위해 현재의 블록에 대한 디스패리티 벡터를 이용하도록 하며, 상기 디스패리티 벡터 유도 프로세스를 수행하는 것은 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하며, 상기 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는다.

[0014] 본 개시물의 하나 이상의 예들의 세부 사항들은 첨부도면 및 아래의 상세한 설명에서 개시된다. 다른 특성들, 목적들, 및 이점들은 설명, 도면들, 및 청구범위로부터 명백히 알 수 있을 것이다.

도면의 간단한 설명

[0015] 도 1 은 본 개시물에서 설명하는 기법들을 활용할 수도 있는 예시적인 비디오 코딩 시스템을 예시하는 블록도이다.

도 2 는 현재의 PU 에 대한 예시적인 공간적으로-이웃하는 예측 유닛들 (PUs) 을 예시하는 개념도이다.

도 3 은 예시적인 멀티-뷰 디코딩 순서를 예시하는 개념도이다.

도 4 는 멀티-뷰 코딩을 위한 예시적인 예측 구조를 예시하는 개념도이다.

도 5 는 시간 후보 화상들의 대응하는 PU 에서 예시적인 시간 이웃들을 예시하는 개념도이다.

도 6 은 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 7 은 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 8 은 코딩 유닛 (CU) 과 동일한 2Nx2N PU 에서 예시적인 공간 및 시간 이웃들을 예시하는 개념도이다.

도 9 는 CU 의 제 1 PU 가 CU 에 대한 대표 PU 인, 이웃하는 블록들-기반의 디스패리티 벡터 (NBDV) 유도 프로세스에서 사용되는 예시적인 공간 및 시간 이웃하는 블록들을 예시하는 개념도이다.

도 10 은 본 개시물의 하나 이상의 기법들에 따른, 비디오 코더의 예시적인 동작을 예시하는 플로우차트이다.

도 11a 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 모션 예측을 수행하는 비디오 인코더의 예시적인 동작을 예시하는 플로우차트이다.

도 11b 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 모션 예측을 수행하는 비디오 디코더의 예시적인 동작을 예시하는 플로우차트이다.

도 12a 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 잔여 예측을 수행하는 비디오 인코더의 예시적인 동작을 예시하는 플로우차트이다.

도 12b 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 잔여 예측을 수행하는 비디오 디코더의 예시적인 동작을 예시하는 플로우차트이다.

도 13 은 2개의 PU들에 대한 예시적인 좌측 공간 이웃하는 블록들 및 상부 공간 이웃하는 블록들을 예시하는 개념도이다.

도 14 는 NBDV 유도 프로세스에 사용되는 좌측, 상부, 및 좌상부 공간 이웃하는 블록들을 예시하는 개념도이다.

도 15 는 본 개시물의 하나 이상의 기법들에 따른, 비디오 코더의 예시적인 동작을 예시하는 플로우차트이다.

도 16 은 본 개시물의 일 예에 따른, 현재의 블록에 대한 디스패리티 벡터를 결정하는 비디오 코더의 예시적인 동작을 예시하는 플로우차트이다.

도 17 은 본 개시물의 하나 이상의 기법들에 따른, 현재의 블록에 대한 디스패리티 벡터를 결정하는 비디오 코더의 예시적인 동작을 예시하는 플로우차트이다.

도 18 은 도 17 에 나타낸 동작의 예시적인 연속부분을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0016] 고-효율 비디오 코딩 (HEVC) 은 새로-개발된 비디오 코딩 표준이다. 3D-HEVC 는 3차원 (3D) 비디오 데이터에 대한 HEVC 의 확장판이다. 3D-HEVC 는 상이한 관찰 지점들로부터의 동일한 장면의 다수의 뷰들을 제공한다. 3D-HEVC 에 대한 표준화 노력들은 HEVC 에 기초한 멀티-뷰 비디오 코덱의 표준화를 포함한다. 3D-HEVC 에서, 상이한 뷰들로부터의 재구성된 뷰 성분들 (즉, 재구성된 화상들) 에 기초한 인터-뷰 예측이 이용가능하게 된다. 더욱이, 3D-HEVC 는 인터-뷰 모션 예측 및 인터-뷰 잔여 예측을 구현한다.

[0017] 비디오의 동일한 시간 인스턴스를 나타내는 각각의 뷰의 화상들은 유사한 비디오 콘텐츠를 포함한다. 그러나, 뷰들의 비디오 콘텐츠는 서로에 대해 공간적으로 변위될 수도 있다. 특히, 뷰들의 비디오 콘텐츠는 상이한 관점들을 나타낼 수도 있다. 예를 들어, 제 1 뷰 내 화상에서의 비디오 블록은 제 2 뷰 내 화상에서의 비디오 블록과 유사한 비디오 콘텐츠를 포함할 수도 있다. 이 예에서, 제 1 뷰 내 화상에서의 비디오 블록의 로케이션 및 제 2 뷰 내 화상에서의 비디오 블록의 로케이션은 상이할 수도 있다. 예를 들어, 상이한 뷰들에서의 비디오 블록들의 로케이션들 사이에 어떤 변위가 있을 수도 있다.

[0018] 비디오 블록에 대한 디스패리티 벡터는 이 변위의 측정치를 제공한다. 예를 들어, 제 1 뷰 내 화상의 비디오 블록은 제 2 뷰 내 화상에서의 대응하는 비디오 블록의 변위를 나타내는 디스패리티 벡터와 연관될 수도 있다.

[0019] 비디오 블록에 대한 디스패리티 벡터를 유도하는 하나의 기법은 이웃하는 블록 디스패리티 벡터 (NBDV) 유도로서 지칭된다. NBDV 유도 기법들에서, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더) 는 이웃하는 블록들 중 임의의 블록이 상이한 뷰에서의 화상으로 인터-예측되었는지 여부를 결정하기 위해, 현재의 블록에 이웃하는 블록들을 평가한다. 예를 들어, 멀티-뷰 코딩은 인터-뷰 예측을 위해 제공한다. 인터-뷰 예측에서, 제 1 뷰 내 화상에서의 비디오 블록은 상이한 제 2 뷰 내 화상에서의 데이터 (예컨대, 비디오 블록) 로 인터-예측된다. 인터-뷰 예측은 인터-예측과 유사하며, 인터-예측 및 인터-뷰 예측 양쪽에서, 화상에서의 비디오 블록은 또 다른 화상에서의 비디오 블록으로 인터-예측된다. 그러나, 규칙적인 인터-예측에서, 화상들은 동일한 뷰로부터 유래하며, 반면, 인터-뷰 예측에서, 화상들은 상이한 뷰들로부터 유래한다.

[0020] 인터-뷰 모션 보상, 인터-뷰 모션 예측, 및 인터-뷰 잔여 예측을 포함한, 인터-뷰 예측의 여러 형태들이 존재한다. 인터-뷰 모션 보상에서, 비디오 코더는 인터-뷰 예측되고 있는 블록에 대한 모션 벡터를 결정할 수도 있다. 혼란을 피하기 위해, 인터-뷰 예측에 사용되는 모션 벡터는 이 모션 벡터가 상이한 뷰에서의 화상을

참조하기 때문에, 디스패리티 모션 벡터로서 지칭된다. 규칙적인 인터-예측을 위한 모션 벡터는 이 모션 벡터가 동일한 뷰에서 그러나 상이한 시간 인스턴스에서의 화상을 참조하기 때문에, 시간 모션 벡터로서 지칭된다.

[0021] 디스패리티 모션 벡터는 현재의 블록의 인터-뷰 모션 보상에 사용되는 블록을 식별한다. 디스패리티 벡터는, 한편, 현재의 블록의 비디오 콘텐츠와 유사한 비디오 콘텐츠를 포함하는 상이한 뷰 내 화상에서의 블록을 식별한다. 디스패리티 모션 벡터의 목적과는 상이한 디스패리티 벡터의 여러 목적들이 있을 수도 있다.

그러나, 디스패리티 벡터는 디스패리티 모션 벡터와 동일한 목적을 위해 사용되는 것이 가능할 수도 있으며, 이 경우, 디스패리티 벡터 및 디스패리티 모션 벡터는 동일하게 된다. 일반적으로, 그러나, 디스패리티 벡터 및 디스패리티 모션 벡터는 상이한 벡터들일 수도 있으며, 상이한 목적들을 위해 사용될 수도 있다.

[0022] NBDV 유도 기법들에서, 비디오 코더는 현재의 블록의 이웃하는 블록들을 평가하고 이웃하는 블록들 중 임의의 블록이 디스패리티 모션 벡터로 인터-뷰 예측되었는지 여부를 결정한다. 이웃하는 블록이 디스패리티 모션 벡터로 인터-뷰 예측되면, 비디오 코더는 이웃하는 블록에 대한 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 예를 들어, 비디오 코더는 디스패리티 모션 벡터의 수평 성분을 유지하고 디스패리티 모션 벡터의 수직 성분을 0 으로 설정함으로써 이웃하는 블록에 대한 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 수 있다. 다른 예들에서, 비디오 코더는 이웃하는 블록에 대한 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 때 수직 성분을 0 이외의 값으로 설정할 수도 있다. 이웃하는 블록들은 현재의 블록과 동일한 화상에서의 블록들 (예컨대, 공간적으로 이웃하는 블록들) 을 포함할 수도 있다. 이러한 공간적으로 이웃하는 블록들은 디스패리티 모션 벡터로 인터-뷰 예측될 수도 있으며, 현재의 블록에 대한 디스패리티 벡터를 형성할 수도 있는 공간적으로 이웃하는 블록들에 대한 디스패리티 모션 벡터는 공간 디스패리티 벡터 (SDV) 로 지칭된다. 일부 예들에서, 이웃하는 블록들은 현재의 블록이 상주하는 화상과는 상이한 화상에서의 블록들 (예컨대, 시간적으로 이웃하는 블록들) 을 포함할 수도 있다. 이러한 시간적으로 이웃하는 블록들은 디스패리티 모션 벡터로 인터-뷰 예측될 수도 있으며, 현재의 블록에 대한 디스패리티 벡터를 형성할 수도 있는 시간적으로 이웃하는 블록들에 대한 디스패리티 모션 벡터는 시간 디스패리티 벡터 (TDV) 로 지칭된다.

[0023] 일부 예들에서, 비디오 코더는 NBDV 유도를 수행하는 것의 일부로서, 이웃하는 블록의 암시적인 디스패리티 벡터 (IDV) 에 기초하여 현재의 블록에 대한 디스패리티 벡터를 결정할 수도 있다. 위에서 나타낸 바와 같이, 비디오 코더가 NBDV 유도를 수행할 때, 비디오 코더는 디스패리티 모션 벡터들에 대해 이웃하는 블록들을 체킹한다. 비디오 코더가 디스패리티 모션 벡터에 대해 이웃하는 블록을 체킹할 때, 비디오 코더는 또한 이웃하는 블록이 IDV 를 가지는지 여부를 결정할 수도 있다. 이웃하는 블록이 IDV 를 가지면, 비디오 코더는 이웃하는 블록이 IDV 를 가진다는 것을 표시하기 위해 이웃하는 블록에 대해 IDV 플래그를 설정할 수도 있다. 디스패리티 모션 벡터들에 대해 이웃하는 블록들의 각각을 체킹한 후, 비디오 코더가 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터를 가지지 않는다고 결정하면, 비디오 코더는 이웃하는 블록들의 IDV 플래그들을 이용하여 IDV 를 가지는 이웃하는 블록을 식별할 수도 있다. 비디오 코더는 그후 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 이웃하는 블록에 대한 IDV 는 이웃하는 블록의 예측 (즉, 코딩) 동안 이웃하는 블록에 대해 유도된 디스패리티 벡터일 수도 있다.

[0024] 현재의 블록에 대한 디스패리티 벡터를 유도하기 위한 상기 설명으로부터, 현재의 블록에 대한 디스패리티 벡터의 유도가 계산 집약적일 수도 있으며 이전에-코딩된 이웃하는 블록들에 대한 디스패리티 모션 벡터들 및 IDVs 를 결정하기 위해 메모리에 대한 다수의 호출들을 필요로 할 수도 있음을 알 수 있다. 본 개시물에서 설명되는 하나 이상의 기법들에 따르면, 디스패리티 벡터를 블록 단위로 유도하는 대신, 비디오 코더는 블록들의 그룹에 대한 디스패리티 벡터를 유도할 수도 있으며, 비디오 코더 그 유도된 디스패리티 벡터를 블록들의 그룹 내 블록들의 각각에 대한 디스패리티 벡터로서 이용할 수도 있다.

[0025] 예를 들어, 위에서 설명한 바와 같이, 코딩 유닛 (CU) 은 복수의 예측 유닛들 (PUs) 로 파티셔닝될 수도 있다. 본 개시물에서 설명되는 하나 이상의 기법들에 따르면, CU 내 각각의 PU 에 대한 디스패리티 벡터를 유도하기 보다는, 비디오 코더는 CU 의 대표 PU 에 대한 디스패리티 벡터를 유도할 수도 있다. 이 예에서, 비디오 코더는 CU 의 각각의 PU 에 대한 디스패리티 벡터들을 따로 유도하기 보다는, 그 유도된 디스패리티 벡터 (즉, 대표 PU 에 대한 디스패리티 벡터) 를 CU 내 PU들의 각각에 할당할 수도 있다. 더욱이, 비디오 코더는 그 유도된 디스패리티 벡터에 기초하여 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측을 수행할 수

도 있다.

[0026] 위에서 설명된 바와 같이, 비디오 코더는 이웃하는 블록들 중 임의의 블록이 디스패리티 모션 벡터로 인터-뷰 예측되는지 여부를 결정하기 위해 이웃하는 블록들을 평가한다. 일부 본 개시물의 예시적인 기법들은 비디오 코더가 NBDV 유도 동안 평가하는 이웃하는 블록들의 개수를 제한한다. 예를 들어, 비디오 코더가 디스패리티 벡터 유도 프로세스 (예컨대, NBDV 유도) 를 수행하고 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 IDV 를 가질 때, 비디오 코더는 디스패리티 모션 벡터 또는 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 그러나, 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 IDV 를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 IDVs 를 가지는 여부를 결정하지 않는다. 따라서, 제 1 및 제 2 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 IDVs 를 가지는지 여부에 상관없이, 비디오 코더는 임의의 추가적인 공간 이웃하는 블록들을 체킹하지 않는다.

[0027] 더욱이, 본 개시물의 하나 이상의 기법들은 IDV 가 처리되는 방법을 수정할 수도 있다. 예를 들어, 일부 기법들에서, 비디오 코더는 이웃하는 블록이 IDV 를 가지는지 여부를 표시하기 위해 IDV 플래그를 이용한다. IDV 플래그를 저장하는 것은 요구되지 않을 수도 있다. 대신, 본 개시물의 하나 이상의 기법들에 따르면, 비디오 코더는 IDVs 가 이전에-코딩된 이웃하는 블록들에 존재하는지 여부를 결정하기 위해 경계구역밖 (out-of-bounds) 정보 (예컨대, 불법 값들) 를 이용할 수도 있다. 예를 들어, 이전에-코딩된 이웃하는 블록이 그런 경계구역밖 정보를 가지는지 여부에 따라서, 비디오 코더는 IDV 가 이웃하는 블록에 대해 존재하거나 또는 존재하지 않는다고 결정할 수도 있다.

[0028] 도 1 은 본 개시물의 기법들을 이용할 수도 있는 예시적인 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 사용될 때, 용어 "비디오 코더" 는 비디오 인코더들 및 비디오 디코더들 양쪽을 포괄적으로 지칭한다. 본 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 비디오 인코딩 또는 비디오 디코딩을 포괄적으로 지칭할 수도 있다.

[0029] 도 1 에 나타낸 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 및 목적지 디바이스 (14) 를 포함한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 발생한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치로서 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 발생된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치로서 지칭될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.

[0030] 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예컨대, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 자동차용 컴퓨터들, 또는 기타 등등을 포함한, 광범위한 디바이스들을 포함할 수도 있다.

[0031] 목적지 디바이스 (14) 는 소스 디바이스 (12) 로부터 채널 (16) 을 통해서 인코딩된 비디오 데이터를 수신할 수도 있다. 채널 (16) 은 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시키는 것이 가능한 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 일 예에서, 채널 (16) 은 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 무선 통신 프로토콜과 같은 통신 표준에 따라서, 인코딩된 비디오 데이터를 변조할 수도 있으며, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 과 같은, 패킷-기반 네트워크의 일부를 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.

[0032] 또 다른 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 발생되는 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 저장 매체에, 예컨대, 디스크 액세스 또는 카드 액세스를 통해서, 액세스할 수도 있다. 저장 매체는 Blu-ray 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 로컬-액세스되는 데이터 저장 매체들을 포함할 수도 있다.

[0033] 추가 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 발생된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 또 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해서 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신가능한 서버의 형태일 수도 있다. 예시적인 파일 서버들은 (예컨대, 웹사이트용) 웹 서버들, 파일 전송 프로토콜 (FTP) 서버들, NAS (network attached storage) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.

[0034] 목적지 디바이스 (14) 는 인코딩된 비디오 데이터에 인터넷 접속과 같은 표준 데이터 접속을 통해서 액세스할 수도 있다. 데이터 접속들의 예시적인 유형들은 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, 디지털 가입자 회선 (DSL), 케이블 모뎀, 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 양자의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다.

[0035] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정들에 한정되지 않는다. 이 기법들은 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 예컨대, 인터넷을 통한 위성 텔레비전 송신들, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들의 지원 하에서, 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 지원 애플리케이션들로의 1-방향 또는 2-방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0036] 도 1 은 단지 예이며, 본 개시물의 기법들은 인코딩 디바이스와 디코딩 디바이스 사이의 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들 (예컨대, 비디오 인코딩 또는 비디오 디코딩) 에 적용할 수도 있다. 다른 예들에서, 데이터 (예컨대, 인코딩된 및/또는 디코딩된 비디오 데이터와 같은, 비디오 데이터) 는 로컬 메모리로부터 취출되어, 네트워크를 통해서 스트리밍되거나, 또는 기타 등등이다. 비디오 인코딩 디바이스는 데이터를 인코딩하여 메모리에 저장할 수도 있으며, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 취출하여 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은 서로 통신하지 않지만 간단히 데이터를 메모리로 인코딩하거나 및/또는 메모리로부터 데이터를 취출하여 디코딩하는 디바이스들에 의해 수행된다.

[0037] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 일부 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 비디오 소스 (18) 는 비디오 캡쳐 디바이스, 예컨대, 비디오 카메라, 이전에-캡쳐된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하는 비디오 공급 인터페이스, 및/또는 비디오 데이터를 발생하기 위한 컴퓨터 그래픽스 시스템, 또는 비디오 이런 데이터의 소스들의 조합을 포함할 수도 있다.

[0038] 비디오 인코더 (20) 는 비디오 소스 (18) 로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로부터 출력 인터페이스 (22) 를 통해서 직접 송신한다. 다른 예들에서, 인코딩된 비디오 데이터는 또한 디코딩 및/또는 플레이백을 위해 목적지 디바이스 (14) 에 의한 추후 액세스를 위해 저장 매체 또는 파일 서버 상으로 저장될 수도 있다.

[0039] 도 1 의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함한다. 입력 인터페이스 (28) 은 인코딩된 비디오 데이터를 채널 (16) 을 통해서 수신할 수도 있다. 비디오 디코더 (30) 는 인코딩된 비디오 데이터를 디코딩할 수도 있다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이할 수도 있다. 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 또는 그 외부에 있을 수도 있다. 디스플레이 디바이스 (32) 는 다양한 디스플레이 디바이스들, 예컨대 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스를 포함할 수도 있다.

[0040] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 집적 회로들 (ASICs), 필드-프로그래밍가능 게이트 어레이들 (FPGAs), 이산 로직, 하드웨어, 또는 임의의 이들의 조합들과 같은, 다양한 적합한 회로 중 임의의 회로로서 구현될 수도 있다. 기법들이 소

프트웨어로 부분적으로 구현되면, 디바이스는 소프트웨어용 명령들을 적합한 비일시성 컴퓨터-판독가능 저장 매체에 저장할 수도 있으며, 본 개시물의 기법들을 수행하기 위해 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 이용하여 실행할 수도 있다. (하드웨어, 소프트웨어, 하드웨어와 소프트웨어의 조합 등을 포함한) 전술한 것 중 임의의 것이 하나 이상의 프로세서들로 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있다.

[0041] 본 개시물은 일반적으로 어떤 정보를 비디오 디코더 (30) 와 같은 또 다른 디바이스로 "시그널링하는" 비디오 인코더 (20) 를 참조할 수도 있다. 용어 "시그널링" 은 일반적으로 압축된 비디오 데이터를 디코딩하는데 사용되는 구문 엘리먼트들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 이런 통신은 실시간 또는 거의 -실시간으로 일어날 수도 있다. 대안적으로, 이런 통신은 어떤 기간에 걸쳐서 일어날 수도 있으며, 예컨대 인코딩 시에 구문 엘리먼트들을 컴퓨터-판독가능 저장 매체에 인코딩된 비트스트림으로 저장할 때에 발생할지도 모르며, 이 구문 엘리먼트들은 그후 이 매체에 저장되어진 후 언제라도 디코딩 디바이스에 의해 취출될 수도 있다.

[0042] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ISO/IEC MPEG-4 Visual 및 ITU-T H.264 (또한, ISO/IEC MPEG-4 AVC 로서 알려짐) 과 같은, 그의 스케일러블 비디오 코딩 (SVC) 확장판, 멀티-뷰 비디오 코딩 (MVC) 확장판, 및 MVC-기반의 3DV 확장판을 포함한, 비디오 압축 표준에 따라서 동작한다. 더욱이, H.264/AVC 에 대한 3차원의 비디오 (3DV) 코딩 확장판, 즉 AVC-기반의 3DV 를 만들어 내려는 노력이 진행 중에 있다. H.264 의 MVC 확장판의 합동 초안은 2010 년 3월, ITU-T 권고안 H.264, "Advanced Video Coding for generic audiovisual services" 에 설명되어 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, 및 ITU-T H.263, ISO/IEC-4 Visual 에 따라서 동작할 수도 있다. 따라서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual 및 (ISO/IEC MPEG-4 AVC 로서 또한 알려진) 그의 SVC 및 MVC 확장판들을 포함한, ITU-T H.264 를 포함하는, 비디오 코딩 표준들에 따라서 동작할 수도 있다.

[0043] 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 과 ISO/IEC 동화상 전문가 그룹 (MPEG) 의 비디오 코딩에 관한 합동 작업팀 (JCT-VC) 에 의해 개발된 HEVC (High Efficiency Video Coding) 표준에 따라서 동작할 수도 있다. "HEVC Working Draft 8" 또는 "HEVC base specification" 으로 지칭되는, HEVC 표준의 초안은 ITU-T SG16 WP3 와 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 2012년 7월, 스웨덴, 스톡홀름, 10차 회의, Bross 등, "High Efficiency Video Coding (HEVC) text specification draft 8" 에 설명되어 있다. 더욱이, HEVC 에 대한 스케일러블 비디오 코딩, 멀티-뷰 코딩, 및 3DV 확장판들을 만들어 내려는 노력이 진행 중에 있다. HEVC 의 스케일러블 비디오 코딩 확장판은 SHEVC 로서 지칭될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC 표준에 대한 이러한 확장판들에 따라서 동작할 수도 있다.

[0044] 현재, VCEG 및 MPEG 의 3D 비디오 코딩 (JCT-3C) 에 관한 합동 연구팀은 HEVC 에 기초한 3DV 표준을 개발하고 있으며, 표준화 노력들의 부분은 HEVC (MV-HEVC) 에 기초한 멀티-뷰 비디오 코덱의 표준화 및 HEVC (3D-HEVC) 에 기초한 3D 비디오 코딩을 위한 또 다른 부분을 포함한다. 3D-HEVC 에 대해, 텍스쳐 및 심도 뷰들 양쪽에 대해, 코딩 유닛/예측 유닛 레벨에서의 툴들을 포함한, 새로운 코딩 툴들이 포함되고 지원될 수도 있다. 2013년 12월 5일 현재, 3D-HEVC (즉, 3D-HTM 버전 6.0) 에 대한 소프트웨어는 다음 링크로부터 다운로드될 수 있다: [3D-HTM]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-6.0/

[0045] 일반적으로, HEVC 의 모션 보상 루프는 H.264/AVC 에서의 모션 보상 루프와 동일하다. 예를 들어, 현재의 프레임 ($\hat{I}$)의 재구성은 역양자화된 계수들 (r) 플러스 시간 예측 (P) 과 동일할 수도 있다:

[0046] $$\hat{I} = r + P$$

[0047] 상기 공식에서, P 는 P 프레임들에 대한 단방향 예측 또는 B 프레임들에 대한 양방향 예측을 나타낸다.

[0048] 그러나, HEVC 에서의 모션 보상의 단위는 이전 비디오 코딩 표준들에서의 모션 보상의 단위와는 상이하다. 예를 들어, 이전 비디오 코딩 표준들에서 매크로블록의 컨셉이 HEVC 에 존재하지 않는다. 대신, 매크로블록

들은 일반적인 쿼드트리 방식에 기초하여 매우 유연한 계층적 구조로 대체된다. 이 방식에 의하면, 블록들의 3개의 유형들, 즉, 코딩 유닛들 (CUs), 예측 유닛들 (PUs), 및 변환 유닛들 (TUs) 이 정의된다. CU 는 영역 분할의 기본적인 단위이다. CU 의 컨셉은 매크로블록의 컨셉과 유사하며, 그러나 CU 는 최대 사이즈에 제한되지 않으며 CU 는 콘텐츠 적응성을 향상시키기 위해 4개의 동일-사이즈로된 CU들로의 회귀적인 분할을 허용한다. PU 는 인터/인트라 예측의 기본적인 단위이며, PU 는 불규칙적인 이미지 패턴들을 효과적으로 코딩하기 위해 단일 PU 에 다수의 임의-형상의 파티션들을 포함할 수도 있다. TU 는 변환의 기본적인 단위이다. CU 의 TU들은 CU 의 PU들과는 따로 정의될 수 있다. 그러나, TU 의 사이즈는 TU 가 속하는 CU 에 제한된다. 3개의 상이한 컨셉들로의 블록 구조의 이 분리는 각각이 그의 역할에 따라서 최적화될 수 있게 할 수도 있으며, 이것은 향상된 코딩 효율을 초래할 수도 있다.

[0049] HEVC 및 다른 비디오 코딩 사양들에서, 비디오 시퀀스는 일반적으로 일련의 화상들을 포함한다. 화상들은 또한 "프레임들" 로서 지칭될 수도 있다. 화상은 SL, SCb, 및 SCr 로 표기되는, 3개의 샘플 어레이들을 포함할 수도 있다. SL 은 루마 샘플들의 2차원 어레이 (즉, 블록) 이다. SCb 는 Cb 색차 샘플들의 2차원 어레이이다. SCr 은 Cr 색차 샘플들의 2차원 어레이이다. 색차 샘플들은 또한 본원에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우, 화상은 단색일 수도 있으며 단지 루마 샘플들의 어레이를 포함할 수도 있다.

[0050] 화상의 인코딩된 표현을 발생하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛들 (CTUs) 의 세트를 발생할 수도 있다. CTUs 의 각각은 루마 샘플들의 코딩 트리 블록 (CTB), 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용되는 구문 구조들을 포함할 수도 있다. 단색 화상들 또는 3개의 별개의 칼라 플레인들을 가지는 화상들에서, CTU 는 단일 코딩 트리 블록 및 코딩 트리 블록의 샘플들을 코딩하는데 사용되는 구문 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로서 지칭될 수도 있다. HEVC 의 CTUs 는 H.264/AVC 와 같은, 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 특정의 사이즈에 반드시 제한되지 않으며, 하나 이상의 CU들을 포함할 수도 있다.

[0051] 슬라이스는 래스터 스캐닝 순서로 연속적으로 순서화된 CTUs 의 정수를 포함할 수도 있다. 코딩된 슬라이스는 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스의 슬라이스 헤더는 슬라이스에 관한 정보를 제공하는 구문 엘리먼트들을 포함하는 구문 구조일 수도 있다. 슬라이스 데이터는 슬라이스의 코딩된 CTUs 를 포함할 수도 있다

[0052] 본 개시물은 하나 이상의 샘플 블록들 및 하나 이상의 샘플들의 블록들의 샘플들을 코딩하는데 사용되는 구문 구조들을 지칭하기 위해, 용어 "비디오 유닛" 또는 "비디오 블록" 또는 "블록" 을 이용할 수도 있다. 비디오 유닛들의 예시적인 유형들은 CTUs, CU들, PU들, 변환 유닛들 (TUs), 매크로블록들, 매크로블록 파티션들 등을 포함할 수도 있다. 일부 상황들에서, PU들의 논의는 매크로블록 파티션들의 매크로블록들의 논의와 상호 교환될 수도 있다.

[0053] 코딩된 CTU 를 발생하기 위해, 비디오 인코더 (20) 는 코딩 트리 블록들을 코딩 블록들, 따라서 이름 "코딩 트리 유닛들" 로 분할하기 위해 CTU 의 코딩 트리 블록들에 관해 쿼드-트리 파티셔닝을 회귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다. CU 는 루마 샘플들의 코딩 블록 및 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 가지는 화상의 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 코딩 블록들의 샘플들을 코딩하는데 사용되는 구문 구조들을 포함할 수도 있다. 단색 화상들 또는 3개의 별개의 칼라 플레인들을 가지는 화상들에서, CU 는 단일 코딩 블록 및 코딩 블록의 샘플들을 코딩하는데 사용되는 구문 구조들을 포함할 수도 있다.

[0054] 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록이다. CU 의 PU 는 루마 샘플들의 예측 블록, 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록들을 예측하는데 사용되는 구문 구조들을 포함할 수도 있다. 단색 화상들 또는 3개의 별개의 칼라 플레인들을 가지는 화상들에서, PU 는 단일 예측 블록 및 예측 블록을 예측하는데 사용되는 구문 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb, 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 발생할 수도 있다. 그러므로, 본 개시물에서, CU 는 하나 이상의 PU들로 파티셔닝되는 것으로 말할 수도 있다. 설명의 용이성을 위해, 본 개시물은 PU 의 예측 블록의 사이즈를 간단히 PU 의 사이즈로서 지칭할 수도 있다.

[0055] 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 이용하여, PU 에 대한 예측 블록들을 발생할 수도 있다.

비디오 인코더 (20) 가 PU 의 예측 블록들을 발생하기 위해 인트라 예측을 이용하면, 비디오 인코더 (20) 는 PU 와 연관되는 화상의 샘플들에 기초하여 PU 의 예측 블록들을 발생할 수도 있다. 본 개시물에서, 어구 "에 기초하여" 는 "적어도 부분적으로 기초하여" 를 나타낼 수도 있다.

[0056] 비디오 인코더 (20) 가 PU 의 예측 블록들을 발생하기 위해 인터 예측을 이용하면, 비디오 인코더 (20) 는 PU 와 연관되는 화상 이외의 하나 이상의 화상들의 디코딩된 샘플들에 기초하여, PU 의 예측 블록들을 발생할 수도 있다. 인터 예측이 블록 (예컨대, PU) 의 예측 블록들을 발생하는데 사용되면, 본 개시물은 그 블록을 "인터-코딩되는" 또는 "인터 예측되는" 것으로 지칭할 수도 있다. 인터 예측은 단방향(즉, 단방향-예측) 또는 양방향(즉, 양방향-예측) 일 수도 있다. 단방향-예측 또는 양방향-예측을 수행하기 위해, 비디오 인코더 (20) 는 현재의 화상 에 대한 제 1 참조 화상 리스트 (RefPicList0) 및 제 2 참조 화상 리스트 (RefPicList1) 를 발생할 수도 있다. 참조 화상 리스트들의 각각은 하나 이상의 참조 화상들을 포함할 수도 있다. 참조 화상 리스트가 구성된 후 (즉, 이용가능하면, RefPicList0 및 RefPicList1), 참조 화상 리스트로의 참조 인덱스가 참조 화상 리스트에 포함된 임의의 참조 화상을 식별하는데 사용될 수 있다.

[0057] 단방향-예측을 이용할 때, 비디오 인코더 (20) 는 참조 화상 내 참조 로케이션을 결정하기 위해 RefPicList0 및 RefPicList1 중 어느 하나 또는 양쪽에서 참조 화상들을 탐색할 수도 있다. 더욱이, 단방향-예측을 이용할 때, 비디오 인코더 (20) 는 참조 로케이션에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU 에 대한 예측 블록들을 발생할 수도 있다. 더욱이, 단방향-예측을 이용할 때, 비디오 인코더 (20) 는 PU 의 예측 블록과 참조 로케이션 사이의 공간 변위를 나타내는 단일 모션 벡터를 발생할 수도 있다. 모션 벡터는 PU 의 예측 블록과 참조 로케이션 사이의 수평 변위를 규정하는 수평 성분을 포함할 수도 있으며, PU 의 예측 블록과 참조 로케이션 사이의 수직 변위를 규정하는 수직 성분을 포함할 수도 있다.

[0058] PU 를 인코딩하는데 양방향-예측을 이용할 때, 비디오 인코더 (20) 는 RefPicList0 에서의 참조 화상에서 제 1 참조 로케이션을, 그리고 RefPicList1 에서의 참조 화상에서 제 2 참조 로케이션을 결정할 수도 있다. 비디오 인코더 (20) 는 제 1 및 제 2 참조 로케이션들에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU 에 대한 예측 블록들을 발생할 수도 있다. 더욱이, PU 를 인코딩하기 위해 양방향-예측을 이용할 때, 비디오 인코더 (20) 는 PU 의 예측 블록과 제 1 참조 로케이션 사이의 공간 변위를 나타내는 제 1 모션 벡터, 및 PU 의 예측 블록과 제 2 참조 로케이션 사이의 공간 변위를 나타내는 제 2 모션 벡터를 발생할 수도 있다.

[0059] 비디오 인코더 (20) 가 PU 의 예측 블록들을 발생하기 위해 인터 예측을 이용하면, 비디오 인코더 (20) 는 PU 와 연관되는 화상 이외의 하나 이상의 화상들의 샘플들에 기초하여 PU 의 예측 블록들을 발생할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 PU 에 관해 단방향 인터 예측 (즉, 단방향-예측) 또는 양방향 인터 예측 (즉, 양방향-예측) 을 수행할 수도 있다.

[0060] 비디오 인코더 (20) 가 PU 에 관해 단방향-예측을 수행하는 경우에, 비디오 인코더 (20) 는 PU 의 모션 벡터에 기초하여, 참조 화상에서 참조 로케이션을 결정할 수도 있다. 비디오 인코더 (20) 는 그후 PU 에 대한 예측 블록을 결정할 수도 있다. PU 에 대한 예측 블록에서의 각각의 샘플은 참조 로케이션과 연관될 수도 있다. 일부 예들에서, PU 에 대한 예측 블록에서의 샘플은 샘플이 PU 와 동일한 사이즈를 가지는 샘플들의 블록 내에 있고 그의 좌상단 모서리가 참조 로케이션일 때 참조 로케이션과 연관될 수도 있다. 예측 블록에서의 각각의 샘플은 참조 화상의 실제 또는 내삽된 샘플일 수도 있다. 예측 블록의 루마 샘플들이 참조 화상의 내삽된 루마 샘플들에 기초하는 경우에, 비디오 인코더 (20) 는 8-탭 내삽 필터를 참조 화상의 실제 루마 샘플들에 적용함으로써, 내삽된 루마 샘플들을 발생할 수도 있다. 예측 블록의 크로마 샘플들이 참조 화상의 내삽된 크로마 샘플들에 기초하는 경우에, 비디오 인코더 (20) 는 4-탭 내삽 필터를 참조 화상의 실제 크로마 샘플들에 적용함으로써, 내삽된 크로마 샘플들을 발생할 수도 있다. 일반적으로, 필터의 탭들 (taps) 의 개수는 필터를 수학적으로 표현하기 위해 요구되는 계수들의 개수를 나타낸다. 더 높은 탭 개수를 가진 필터는 일반적으로 낮은 탭 개수를 가진 필터보다 더 복잡하다.

[0061] 비디오 인코더 (20) 가 PU 에 관해 양방향-예측을 수행하는 경우에, PU 는 2개의 모션 벡터들을 갖는다. 비디오 인코더 (20) 는 PU 의 모션 벡터들에 기초하여, 2개의 참조 화상들에서 2개의 참조 로케이션들을 결정할 수도 있다. 비디오 인코더 (20) 는 그후 위에서 설명된 방법으로, 2개의 참조 로케이션들과 연관되는 참조 블록들을 결정할 수도 있다. 비디오 인코더 (20) 는 그후 PU 에 대한 예측 블록을 결정할 수도 있다. 예측 블록에서의 각각의 샘플은 참조 블록들에서의 대응하는 샘플들의 가중 평균일 수도 있다. 샘플들의 가중 (weighting) 은 PU 를 포함하는 화상으로부터의 참조 화상들의 시간 거리들에 기초할 수도 있다.

[0062] 비디오 인코더 (20) 는 CU 를 하나 이상의 PU들로 여러 파티셔닝 모드들에 따라서 파티셔닝할 수도 있다.

예를 들어, 인트라 예측이 CU 의 PU들에 대한 예측 블록들을 발생하는데 사용되면, CU 는 PART_2Nx2N 모드 또는 PART_NxN 모드에 따라서 파티셔닝될 수도 있다. PART_2Nx2N 모드에서, CU 는 단지 하나의 PU 를 갖는다. PART_NxN 모드에서, CU 는 직사각형의 예측 블록들을 가지는 4개의 동일-사이즈로된 PU들을 갖는다. 인터 예측이 CU 의 PU들에 대한 예측 블록들을 발생하는데 사용되면, CU 는 PART_2Nx2N 모드, PART_NxN 모드, PART_2NxN 모드, PART_Nx2N 모드, PART_2NxnU 모드, PART_2NxuD 모드, PART_nLx2N 모드, 또는 PART_nRx2N 모드에 따라서 파티셔닝될 수도 있다. PART_2NxN 모드 및 PART_Nx2N 모드에서, CU 는 직사각형의 예측 블록들을 가지는 2개의 동일-사이즈로된 PU들로 파티셔닝된다. PART_2NxnU 모드, PART_2NxuD 모드, PART_nLx2N 모드, 및 PART_nRx2N 모드의 각각에서, CU 는 직사각형의 예측 블록들을 가지는 2개의 un동일-사이즈로된 PU들로 파티셔닝된다.

[0063] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 루마, Cb, 및 Cr 블록들을 발생한 후, 비디오 인코더 (20) 는 CU 에 대한 루마 잔여 블록을 발생할 수도 있다. CU 의 루마 잔여 블록에서의 각각의 샘플은 CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 원래 루마 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낸다. 게다가, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔여 블록을 발생할 수도 있다. CU 의 Cb 잔여 블록에서의 각각의 샘플은 CU 의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 원래 Cb 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한 CU 에 대한 Cr 잔여 블록을 발생할 수도 있다. CU 의 Cr 잔여 블록에서의 각각의 샘플은 CU 의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 원래 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다.

[0064] 더욱이, 비디오 인코더 (20) 는 쿼드-트리 파티셔닝을 이용하여, CU 의 루마, Cb, 및 Cr 잔여 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형의 (예컨대, 정사각형 또는 비-정사각형) 블록이다. CU 의 TU 는 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용되는 구문 구조들을 포함할 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관되는 루마 변환 블록은 CU 의 루마 잔여 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔여 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔여 블록의 서브-블록일 수도 있다. 단색 화상들 또는 3개의 별개의 칼라 플레인들을 가지는 화상들에서, TU 는 단일 변환 블록 및 변환 블록의 샘플들을 변환하는데 사용되는 구문 구조들을 포함할 수도 있다.

[0065] 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용하여, TU 에 대한 루마 계수 블록을 발생할 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용하여, TU 에 대한 Cb 계수 블록을 발생할 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용하여, TU 에 대한 Cr 계수 블록을 발생할 수도 있다.

[0066] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록, 또는 Cr 계수 블록) 을 발생한 후, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로 변환 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감축하기 위해 변환 계수들이 양자화되는 프로세스를 지칭하며, 추가적인 압축을 제공한다. 비디오 인코더 (20) 는 CU 와 연관되는 양자화 파라미터 (QP) 값에 기초하여 CU 의 TU 와 연관되는 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 CU 와 연관되는 QP 값을 조정함으로써 CU 와 연관되는 계수 블록들에 적용되는 양자화의 정도를 조정할 수도 있다. 일부 예들에서, CU 와 연관되는 QP 값은 현재의 화상 또는 슬라이스와 전체로서 연관될 수도 있다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 구문 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 구문 엘리먼트들에 관해 컨텍스트-적응 2진 산술 코딩 (CABAC) 을 수행할 수도 있다.

[0067] 비디오 인코더 (20) 는 코딩된 화상들 및 연관되는 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 네트워크 추상화 계층 (NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은 NAL 유닛에서의 데이터의 형태, 및 그 데이터를 미가공 바이트 시퀀스 페이로드 (RBSP) 의 유형으로 에뮬레이션 방지 비트들과 필요에 따라 섞어서 포함하는 바이트들의 표시를 포함하는 구문 구조이다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하며 RBSP 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 구문 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 규정된 NAL 유닛 유형 코드는 NAL 유닛의 형태를 나타낸다. RBSP 는 NAL 유닛 내에 캡슐화된 정수의 바이트들을 포함하는 구문 구

조일 수도 있다. 일부의 경우, RBSP 는 제로 비트들을 포함한다.

[0068] 상이한 유형들의 NAL 유닛들이 상이한 유형들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, 상이한 유형들의 NAL 유닛은 비디오 파라미터 세트들 (VPSs), 시퀀스 파라미터 세트들 (SPS들), 화상 파라미터 세트들 (PPS들), 코딩된 슬라이스들, SEI 등에 대해 상이한 RBSP들을 캡슐화할 수도 있다. (파라미터 세트들 및 SEI 메시지들에 대한 RBSP들과는 반대로) 비디오 코딩 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 비디오 코딩 계층 (VCL) NAL 유닛들로서 지칭될 수도 있다.

[0069] HEVC 에서, SPS들은 코딩된 비디오 시퀀스 (CVS) 의 모든 슬라이스들에 적용되는 정보를 포함할 수도 있다. HEVC 에서, CVS 는 동시 디코딩 리프레시 (IDR) 화상, 또는 깨진 링크 액세스 (BLA) 화상, 또는 IDR 또는 BLA 화상이 아닌 모든 후속 화상들을 포함하여, 비트스트림에서 제 1 화상인 깨끗한 무작위 액세스 (CRA) 화상으로부터 시작할 수도 있다. 즉, HEVC 에서, CVS 는 디코딩 순서에서, 비트스트림에서 제 1 액세스 유닛인 CRA 액세스 유닛, IDR 액세스 유닛 또는 BLA 액세스 유닛, 뒤이어서, 모든 후속 액세스 유닛들까지 포함하지만 임의의 후속 IDR 또는 BLA 액세스 유닛은 포함하지 않는 0개 이상의 비-IDR 및 비-BLA 액세스 유닛들로 이루어질 수도 있는 액세스 유닛들의 시퀀스를 포함할 수도 있다.

[0070] VPS 는 0개 이상의 전체 CVSs 에 적용하는 구문 엘리먼트들을 포함하는 구문 구조이다. SPS 는 SPS 가 활성일 때 활성인 VPS 를 식별하는 구문 엘리먼트를 포함할 수도 있다. 따라서, VPS 의 구문 엘리먼트들은 일반적으로 SPS 의 구문 엘리먼트들보다 더 많이 적용가능할 수도 있다. PPS 는 0개 이상의 코딩된 화상들에 적용하는 구문 엘리먼트들을 포함하는 구문 구조이다. PPS 는 PPS 가 활성일 때 활성인 SPS 를 식별하는 구문 엘리먼트를 포함할 수도 있다. 슬라이스의 슬라이스 헤더는 슬라이스가 코딩되고 있을 때 활성인 PPS 를 식별하는 구문 엘리먼트를 포함할 수도 있다.

[0071] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 발생되는 비트스트림을 수신할 수도 있다. 게다가, 비디오 디코더 (30) 는 비트스트림을 파싱하여, 비트스트림으로부터 구문 엘리먼트들을 획득할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 구문 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 화상들을 재구성할 수도 있다. 비디오 데이터를 재구성하는 프로세스는 일반적으로 비디오 인코더 (20) 에 의해 수행되는 프로세스와 반대일 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재의 CU 의 PU들에 대한 예측 블록들을 결정하기 위해 PU들의 모션 벡터들을 이용할 수도 있다. 게다가, 비디오 디코더 (30) 는 현재의 CU 의 TU들과 연관되는 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 계수 블록들에 관해 역변환들을 수행하여, 현재의 CU 의 TUs 과 연관되는 변환 블록들을 재구성할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 PU들에 대한 예측 블록들의 샘플들을 현재의 CU 의 TUs 의 변환 블록들의 대응하는 샘플들에 가산함으로써, 현재의 CU 의 코딩 블록들을 재구성할 수도 있다. 화상의 각각의 CU 에 대해 코딩 블록들을 재구성함으로써, 비디오 디코더 (30) 는 그 화상을 재구성할 수도 있다.

[0072] 각각의 코딩된 화상은 코딩된 화상 또는 그 코딩된 화상에 (미래에) 뒤따르는 화상에 의해 참조를 위해 사용될 수도 있는 모든 화상들을 포함하는 참조 화상 세트를 갖는다. 비디오 코더는 어느 화상들이 미래 화상의 참조로서 단지 사용될 수 있는지를 구별할 수도 있다. 참조 화상 리스트들은 미래 화상들의 참조들로서 단지 사용될 수 있는 화상들이 아닌, 현재의 화상에 대해 사용될 수 있는 참조 화상 세트 ("RPS") (즉 "현재용 RPS") 에서의 화상들에 기초하여 구성된다.

[0073] 일부 예들에서, 비디오 인코더 (20) 가 현재의 화상을 인코딩하기 시작할 때, 비디오 인코더 (20) 는 현재의 화상에 대한 참조 화상들 (즉, 참조 화상 서브세트들) 의 5개의 서브세트들을 발생할 수도 있다. 일부 예들에서, 이들 5개의 참조 화상 서브세트들은 다음과 같다: RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr, 및 RefPicSetLtFoll. 본 개시물은 RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll 에서의 참조 화상들을 "단기 참조 화상들", "단기 화상들", 또는 "STRPs" 로서 지칭할 수도 있다. 따라서, "단기 참조 화상" 은 (예컨대, RefPicSetStCurrBefore, RefPicSetStCurrAfter, 또는 RefPicSetStFoll 에 있기 때문에) 단기 참조용으로 사용되는 것으로 표시되는 화상일 수도 있다. 본 개시물은 RefPicSetLtCurr 및 RefPicSetLtFoll 에서의 참조 화상들을 "장기 참조 화상들", "장기 화상들", 또는 "LTRPs" 로서 지칭할 수도 있다. 비디오 인코더는 각각의 화상에 대해 5개의 참조 화상 서브세트들을 재발생할 수도 있다.

[0074] 더욱이, 현재의 화상의 현재의 슬라이스가 P 슬라이스 (즉, 인트라 예측 및 단방향 인터 예측이 이용가능하게 된 슬라이스) 일 때, 비디오 인코더 (20) 는 현재의 화상의 RefPicStCurrAfter, RefPicStCurrBefore, 및 RefPicStLtCurr 참조 화상 서브세트들로부터의 참조 화상들을 이용하여, 현재의 슬라이스에 대한 단일 참조 화

상 리스트 (RefPicList0) 를 발생할 수도 있다. 현재의 슬라이스가 B 슬라이스 (즉, 인트라 예측, 단방향 인터 예측, 및 양방향 인터 예측이 이용가능하게 된 슬라이스) 이면, 비디오 인코더 (20) 는 현재의 화상의 RefPicStCurrAfter, RefPicStCurrBefore, 및 RefPicStLtCurr 참조 화상 서브세트들로부터의 참조 화상들을 이용하여, 현재의 슬라이스에 대해 2개의 참조 화상 리스트들 (RefPicList0 및 RefPicList1) 을 발생할 수도 있다. 비디오 인코더 (20) 는 현재의 화상의 제 1 슬라이스에 대한 슬라이스 헤더에, 비디오 디코더 (30) 가 현재의 화상의 참조 화상 서브세트들을 결정하는데 이용될 수도 있는 구문 엘리먼트들을 포함시킬 수도 있다. 비디오 디코더 (30) 가 현재의 화상의 현재의 슬라이스를 디코딩할 때, 비디오 디코더 (30) 는 현재의 화상의 참조 화상 서브세트들을 결정할 수도 있으며 RefPicList0 및/또는 RefPicList1 를 재발생할 수도 있다.

[0075] 위에서 나타낸 바와 같이, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 화상의 현재의 슬라이스를 코딩하기 시작할 때, 비디오 코더는 제 1 참조 화상 리스트 (즉, RefPicList0) 를 초기화할 수도 있다. 더욱이, 현재의 슬라이스가 B 슬라이스이면, 비디오 코더는 제 2 참조 화상 리스트 (즉, RefPicList1) 를 초기화할 수도 있다. 일부 예들에서, 참조 화상 리스트 초기화는 참조 화상들의 POC (picture order count) 값들의 순서에 기초하여 참조 화상 메모리 (즉, 디코딩 화상 버퍼) 에서의 참조 화상들을 리스트에 삽입하는 명시적인 메카니즘이다. POC 값은 동일한 코딩된 비디오 시퀀스에서 다른 화상들의 출력 순서 위치들에 대한, 출력 순서에서의 연관된 화상의 위치를 나타내는 각각의 화상과 연관되는 변수이다.

[0076] RefPicList0 를 발생하기 위해, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더) 는 RefPicList0 의 초기, 디폴트 버전을 발생할 수도 있다. RefPicList0 의 초기 버전에서, RefPicSetStCurrBefore 에서의 참조 화상들이 첫번째로 리스트되고, 뒤이어서 RefPicSetStCurrAfter 에서의 참조 화상들, 뒤이어서 RefPicSetLtCurr 에서의 참조 화상들이 리스트된다. 이와 유사하게, RefPicList1 를 발생하기 위해, 비디오 코더는 RefPicList1 의 초기 버전을 발생할 수도 있다. RefPicList1 의 초기 버전에서, RefPicSetStCurrAfter 에서의 참조 화상들이 리스트된 첫번째로 리스트되고, 뒤이어서 RefPictSetStCurrBefore 에서의 참조 화상들, 뒤이어서 RefPicSetLtCurr 에서의 참조 화상들이 리스트된다.

[0077] 비디오 코더가 참조 화상 리스트 (예컨대, RefPicList0 또는 RefPicList1) 를 초기화한 후, 비디오 코더는 참조 화상 리스트에서의 참조 화상들의 순서를 수정할 수도 있다. 다시 말해서, 비디오 코더는 참조 화상 리스트 변경 (RPLM) 프로세스를 수행할 수도 있다. 비디오 코더는 참조 화상들의 순서를 하나의 특정의 참조 화상이 참조 화상 리스트에서 하나 보다 많은 위치에 나타날 수도 있는 경우를 포함하여, 임의의 순서로, 수정할 수도 있다. 다시 말해서, 참조 화상 리스트 재순서정렬 (reordering) 메카니즘은 참조 화상 리스트 초기화 동안 리스트에 삽입된 화상의 위치를 임의의 새로운 위치로 수정하거나, 또는 설령 화상이 그 초기화된 리스트에 속하지 않더라도, 임의의 참조 화상을 참조 화상 메모리에 임의의 위치에 삽입할 수도 있다. 그러나, 화상의 위치가 리스트의 활성 참조 화상들의 개수를 초과하면, 화상은 최종 참조 화상 리스트의 엔트리로서 간주되지 않는다. 슬라이스 헤더는 참조 화상 리스트들에서 활성 참조 화상들의 개수를 나타내는 하나 이상의 구문 엘리먼트들을 포함할 수도 있다.

[0078] 일부 예들에서, 비디오 코더는 비디오 코더가 최종 참조 화상 리스트들 (즉, RefPicList0 및 RefPicList1) 을 구성한 후 B 슬라이스에 대해, 결합된 리스트 (예컨대, RefPicListC) 를 구성한다. 비디오 코더는 하나 이상의 참조 화상 리스트 변경 구문 엘리먼트들이 결합된 리스트에 존재하면, 추가로 그 결합된 리스트를 더 수정할 수도 있다.

[0079] 일부 예들에서, 비디오 인코더 (20) 는 병합/스킵 모드 또는 진보된 모션 벡터 예측 (AMVP) 모드를 이용하여 PU 의 모션 정보를 시그널링할 수도 있다. 다시 말해서, HEVC 에서, 모션 파라미터들의 예측을 위한 2개의 모드들이 존재하며, 하나는 병합/스킵 모드이고 다른 하나는 AMVP 모드이다. 모션 예측은 하나 이상의 다른 블록들의 모션 정보에 기초한 블록 (예컨대, PU) 의 모션 정보의 결정을 포함할 수도 있다. PU 의 모션 정보는 PU 의 모션 벡터(들), PU 의 참조 인덱스(들), 및 하나 이상의 예측 방향 표시자들을 포함할 수도 있다.

[0080] 비디오 인코더 (20) 가 병합 모드를 이용하여 현재의 PU 의 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 병합 후보 리스트를 발생한다. 다시 말해서, 비디오 인코더 (20) 는 모션 벡터 예측자 리스트 구성 프로세스를 수행할 수도 있다. 병합 후보 리스트는 현재의 PU 에 공간적으로 또는 시간적으로 이웃하는 PU들의 모션 정보를 나타내는 병합 후보들의 세트를 포함한다. 즉, 병합 모드에서, 모션 파라미터들 (예컨대, 참조 인덱스들, 모션 벡터들, 등) 의 후보 리스트는, 후보가 공간 및 시간 이웃하는 블록들로부터 유래할 수 경우에 구성된다.

[0081] 더욱이, 병합 모드에서, 비디오 인코더 (20) 는 병합 후보 리스트로부터 병합 후보를 선택할 수도 있으며 선택된 병합 후보에 의해 표시되는 모션 정보를 현재의 PU 의 모션 정보로서 이용할 수도 있다. 비디오 인코더 (20) 는 선택된 병합 후보의 병합 후보 리스트에서의 위치를 시그널링할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 그 선택된 병합 후보의 후보 리스트 내 위치를 나타내는 인덱스 (즉, 병합 후보 인덱스) 를 송신함으로써, 선택된 모션 벡터 파라미터들을 시그널링할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터, 후보 리스트로의 인덱스 (즉, 병합 후보 인덱스) 를 획득할 수도 있다. 게다가, 비디오 디코더 (30) 는 동일한 병합 후보 리스트를 발생할 수도 있으며, 병합 후보 인덱스에 기초하여, 선택된 병합 후보를 결정할 수도 있다. 비디오 디코더 (30) 는 그후 그 선택된 병합 후보의 모션 정보를 이용하여, 현재의 PU 에 대한 예측 블록들을 발생할 수도 있다. 즉, 비디오 디코더 (30) 는 후보 리스트 인덱스에 적어도 부분적으로 기초하여, 후보 리스트에서, 선택된 후보를 결정할 수도 있으며, 여기서, 선택된 후보는 현재의 PU 에 대한 모션 벡터를 규정한다. 이러한 방법으로, 디코더 측에서, 일단 인덱스가 디코딩되면, 인덱스가 가리키는 대응하는 블록의 모든 모션 파라미터들은 현재의 PU 에 의해 상속될 수도 있다.

[0082] 스킵 모드는 병합 모드와 유사하다. 스킵 모드에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 병합 후보 리스트를, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 병합 모드에서 병합 후보 리스트를 이용하는 동일한 방법으로 발생하여 이용한다. 그러나, 비디오 인코더 (20) 가 스킵 모드를 이용하여 현재의 PU 의 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 현재의 PU 에 대한 임의의 잔여 데이터를 시그널링하지 않는다. 따라서, 비디오 디코더 (30) 는 병합 후보 리스트에서의 선택된 후보의 모션 정보에 의해 표시되는 참조 블록에 기초하여, 잔여 데이터의 사용 없이, PU 에 대한 예측 블록을 결정할 수도 있다.

[0083] AMVP 모드는 비디오 인코더 (20) 가 후보 리스트를 발생할 수도 있으며 후보 리스트로부터 후보를 선택할 수도 있다는 점에서, 병합 모드와 유사하다. 그러나, 비디오 인코더 (20) 가 AMVP 모드를 이용하여 현재의 PU 의 RefPicListX (여기서, X 는 0 또는 1 임) 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 현재의 PU 에 대한 RefPicListX 모션 벡터 예측자 (MVP) 플래그를 시그널링하는 것에 더해서, 현재의 PU 에 대한 RefPicListX 모션 벡터 차이 (MVD) 및 현재의 PU 에 대한 RefPicListX 참조 인덱스를 시그널링할 수도 있다. 현재의 PU 에 대한 RefPicListX MVP 플래그는 AMVP 후보 리스트에서, 선택된 AMVP 후보의 위치를 나타낼 수도 있다. 현재의 PU 에 대한 RefPicListX MVD 는 현재의 PU 의 RefPicListX 모션 벡터와 선택된 AMVP 후보의 모션 벡터 사이의 차이를 나타낼 수도 있다. 이러한 방법으로, 비디오 인코더 (20) 는 RefPicListX MVP 플래그, RefPicListX 참조 인덱스 값, 및 RefPicListX MVD 를 시그널링함으로써, 현재의 PU 의 RefPicListX 모션 정보를 시그널링할 수도 있다. 다시 말해서, 현재의 PU 에 대한 모션 벡터를 나타내는 비트스트림에서의 데이터는 참조 인덱스, 후보 리스트로의 인덱스, 및 MVD 를 나타내는 데이터를 포함할 수도 있다. 따라서, 선택된 모션 벡터들은 후보 리스트로의 인덱스를 송신함으로써 시그널링될 수도 있다. 게다가, 참조 인덱스 값들 및 모션 벡터 차이들이 또한 시그널링될 수도 있다.

[0084] 더욱이, 현재의 PU 의 모션 정보가 AMVP 모드를 이용하여 시그널링될 때, 비디오 디코더 (30) 는 비트스트림으로부터, 현재의 PU 에 대한 MVD 및 MVP 플래그를 획득할 수도 있다. 비디오 디코더 (30) 는 동일한 AMVP 후보 리스트를 발생할 수도 있으며, MVP 플래그에 기초하여, 선택된 AMVP 후보를 결정할 수도 있다. 다시 말해서, AMVP 에서, 각각의 모션 가설에 대한 모션 벡터 예측자들의 후보 리스트가 그 코딩된 참조 인덱스에 기초하여 유도된다. 앞서와 같이, 이 리스트는 시간 참조 화상에서 동일 장소에 배치된 블록의 이웃하는 블록의 모션 파라미터들에 기초하여 유도되는 시간 모션 벡터 예측자 뿐만 아니라, 동일한 참조 인덱스와 연관되는 이웃하는 블록들의 모션 벡터들을 포함할 수도 있다. 비디오 디코더 (30) 는 MVD 를 선택된 AMVP 후보에 의해 표시되는 모션 벡터에 가산함으로써, 현재의 PU 의 모션 벡터를 복원할 수도 있다. 즉, 비디오 디코더 (30) 는 그 선택된 AMVP 후보에 의해 표시되는 모션 벡터 및 MVD 에 기초하여, 현재의 PU 의 모션 벡터를 결정할 수도 있다. 비디오 디코더 (30) 는 그후 현재의 PU 의 모션 벡터들 또는 복구된 모션 벡터를 이용하여, 현재의 PU 에 대한 예측 블록들을 발생할 수도 있다.

[0085] 비디오 코더가 현재의 PU 에 대한 AMVP 후보 리스트를 발생할 때, 비디오 코더는 현재의 PU 에 공간적으로 이웃하는 로케이션들을 커버하는 PU들 (즉, 공간적으로-이웃하는 PU들) 의 모션 정보에 기초하여 하나 이상의 AMVP 후보들을 유도하고, 그리고 현재의 PU 에 시간적으로 이웃하는 PU들의 모션 정보에 기초하여 하나 이상의 AMVP 후보들을 유도할 수도 있다. 본 개시물에서, PU (또는, 다른 유형의 비디오 유닛) 은 PU 와 연관되는 예측 블록 (또는, 비디오 유닛과 연관되는 다른 유형의 샘플 블록) 이 그 로케이션을 포함하면, 로케이션을 "커버한다" 고 말할 수도 있다. 후보 리스트는 시간 참조 화상에서 동일 장소에 배치된 블록의 이웃하는 블록의 모션 파라미터들 (즉, 모션 정보) 에 기초하여 유도되는 시간 모션 벡터 예측자 뿐만 아니라, 동일한 참조 인덱스

와 연관되는 이웃하는 블록들의 모션 벡터들을 포함할 수도 있다.

[0086] 도 2 는 현재의 PU (40) 에 대한, 예시적인 공간적으로-이웃하는 PU들을 예시하는 개념도이다. 도 2 의 예에서, 공간적으로-이웃하는 PU들은 A_0, A_1, B_0, B_1, 및 B_2 로서 표시된 로케이션들을 커버하는 PU들일 수도 있다.

다시 말해서, 현재의 PU (40) 와 그의 공간 이웃하는 PU들 사이의 예시적인 관계가 도 2 에 도시된다.

[0087] 공간 이웃 PU들과 관련하여, 다음 심볼들이 정의될 수도 있다:

[0088] - 루마 로케이션 (xP, yP) 는 현재의 화상의 좌상단 샘플에 대한 현재의 PU 의 좌상단 루마 샘플을 규정하는데 사용된다;

[0089] - 변수들 nPSW 및 nPSH 는 루마에 대해 PU 의 폭 및 높이를 표시한다;

[0090] - 현재의 화상의 좌상단 샘플에 대한, 현재의 PU N 의 좌상단 루마 샘플은 (xN, yN) 이다.

[0091] 상기 정의들에서, (xN, yN) (여기서, N 은 A_0, A_1, B_0, B_1 또는 B_2 로 대체됨) 은 (xP - 1, yP + nPSH), (xP - 1, yP + nPSH - 1), (xP + nPSW, yP - 1), (xP + nPSW - 1, yP - 1) 또는 (xP - 1, yP - 1) 으로서 각각 정의된다.

[0092] 현재의 PU (즉, 현재의 PU 와는 상이한 시간 인스턴스와 연관되는 PU) 와 시간적으로 이웃하는 PU 의 모션 정보에 기초하는 병합 후보 리스트 또는 AMVP 후보 리스트에서의 후보는 TMVP 로서 지칭될 수도 있다. TMVP 는 HEVC 의 코딩 효율을 향상시키는데 이용될 수도 있으며, 다른 코딩 툴들과는 달리, TMVP 는 디코딩 화상 버퍼 내 프레임의 모션 벡터에 액세스할 필요가 있을 수도 있다. 좀더 구체적으로는, TMVP 는 참조 화상 리스트에서의 화상의 모션 벡터에 액세스할 필요가 있을 수도 있다.

[0093] TMVP 를 결정하기 위해, 비디오 코더는 먼저 현재의 PU 와 동일 장소에 배치되는 PU 를 포함하는 참조 화상을 식별할 수도 있다. 다시 말해서, 비디오 코더는 소위 "동일 장소에 배치된 화상" 을 식별할 수도 있다. 현재의 화상의 현재의 슬라이스가 B 슬라이스 (즉, 양방향으로 인터 예측된 PU들을 포함하도록 허용되는 슬라이스) 이면, 비디오 인코더 (20) 는 슬라이스 헤더에서, 동일 장소에 배치된 화상이 RefPicList0 또는 RefPicList1 로부터 유래하는지 여부를 나타내는 구문 엘리먼트 (예컨대, collocated_from_l0_flag) 를 시그널링할 수도 있다. 다시 말해서, TMVPs 의 사용이 현재의 슬라이스에 대해 이용가능하게 되고 현재의 슬라이스가 B 슬라이스 (예컨대, 양방향으로 인터 예측된 PU들을 포함하도록 허용되는 슬라이스) 일 때, 비디오 인코더 (20) 는 동일 장소에 배치된 화상이 RefPicList0 또는 RefPicList1 내에 있는지 여부를 나타내기 위해, 슬라이스 헤더로 구문 엘리먼트 (예컨대, collocated_from_l0_flag) 를 시그널링할 수도 있다.

[0094] 슬라이스 헤더에서 구문 엘리먼트 (예컨대, collocated_ref_idx) 는 그 식별된 참조 화상 리스트에서 동일 장소에 배치된 화상을 나타낼 수도 있다. 따라서, 비디오 디코더 (30) 가 동일 장소에 배치된 화상을 포함하는 참조 화상 리스트를 식별한 후, 비디오 디코더 (30) 는 슬라이스 헤더로 시그널링될 수도 있는, collocated_ref_idx 를 이용하여, 그 식별된 참조 화상 리스트에서 동일 장소에 배치된 화상을 식별할 수도 있다.

[0095] 비디오 코더는 동일 장소에 배치된 화상을 체킹함으로써 동일 장소에 배치된 PU 를 식별할 수도 있다. TMVP 는 동일 장소에 배치된 PU 를 포함하는 CU 의 우하단 PU 의 모션 정보, 또는 이 PU 를 포함하는 CU 의 중심 PU들 내 우하단 PU 의 모션 정보를 나타낼 수도 있다. 따라서, 이 PU 를 포함하는 CU 의 우하단 PU 의 모션, 또는 이 PU 를 포함하는 CU 의 중심 PU들 내 우하단 PU 의 모션이 사용된다. 동일 장소에 배치된 PU 를 포함하는 CU 의 우하단 PU 는 PU 의 예측 블록의 우하단 샘플의 바로 하부 및 우측의 로케이션과 동일 장소에 배치된 로케이션을 커버하는 PU 일 수도 있다. 다시 말해서, TMVP 는 현재의 PU 의 우하단 모서리와 동일 장소에 배치되는 로케이션을 커버하는 참조 화상에 있는 PU 의 모션 정보를 나타낼 수도 있거나, 또는 TMVP 는 현재의 PU 의 중심 (즉, 현재의 PU 의 예측 블록의 중심) 과 동일 장소에 배치되는 로케이션을 커버하는 참조 화상에 있는 PU 의 모션 정보를 나타낼 수도 있다.

[0096] 비디오 코더가 시간 참조 화상에서 TMVP 의 모션 벡터를 규정하는 모션 벡터 후보 (예컨대, AMVP 후보 리스트의 병합 리스트에서의 후보) 를 발생할 때, 비디오 코더는 시간 참조 화상의 (POC 값에 의해 반영된) 시간 로케이션에 기초하여 TMVP 의 모션 벡터를 스케일링할 수도 있다. 다시 말해서, 비디오 코더는 현재의 화상과 참조 화상 사이의 POC 거리에 기초하여 모션 벡터 후보의 모션 벡터를 스케일링할 수도 있다. 예를 들어, 비디오 코더가 제 1 화상과 제 2 화상 사이의 POC 거리에 기초하여 모션 벡터를 스케일링할 때, 비디오 코더는 제 1 화상 및 제 2 화상의 POC 값들 사이의 차이가 더 적을 때보다 제 1 화상 및 제 2 화상의 POC 값들 사이의 차

이가 더 클 때 더 큰 양들 만큼 모션 벡터의 크기를 증가시킬 수도 있다.

[0097] 멀티-뷰 코딩에서, 상이한 관찰 지점들로부터의 동일한 장면의 다수의 뷰들이 존재할 수도 있다. 용어 "액세스 유닛" 은 동일한 시간 인스턴스에 대응하는 화상들의 세트를 지칭하기 위해 사용된다. 따라서, 비디오 데이터는 시간 경과에 따라 발생하는 액세스 유닛들의 시리즈로서 개념화될 수도 있다. "뷰 성분" 은 단일 액세스 유닛에서의 뷰의 코딩된 표현일 수도 있다. 본 개시물에서, "뷰" 는 동일한 뷰 식별자와 연관되는 뷰 성분들의 시퀀스를 지칭할 수도 있다.

[0098] 멀티-뷰 코딩은 인터-뷰 예측을 지원한다. 인터-뷰 예측은 HEVC 에 사용되는 인터 예측과 유사하며, 동일한 구문 엘리먼트들을 이용할 수도 있다. 그러나, 비디오 코더가 (PU 와 같은) 현재의 비디오 유닛에 관해 인터-뷰 예측을 수행할 때, 비디오 인코더 (20) 는 참조 화상으로서, 현재의 비디오 유닛과 동일한 액세스 유닛에 있지만 상이한 뷰에 있는 화상을 이용할 수도 있다. 다시 말해서, 멀티-뷰 코딩에서, 인터-뷰 예측은 뷰들 사이의 상관을 제거하기 위해서 동일한 액세스 유닛의 (즉, 동일한 시간 인스턴스 내) 상이한 뷰들에서 캡쳐되는 화상들 사이에 수행된다. 이에 반해, 종래의 인터 예측은 단지 상이한 액세스 유닛들에서의 화상들을 참조 화상들로서 이용한다.

[0099] 멀티-뷰 코딩에서, 비트스트림은 복수의 계층들을 가질 수도 있다. 계층들은 상이한 뷰들에 대응할 수도 있다. 뷰는 비디오 디코더 (예컨대, 비디오 디코더 (30)) 가 임의의 다른 뷰에서의 화상들에 대한 참조 없이 뷰와 연관되는 화상들을 디코딩할 수 있으면, "베이스 뷰" 로서 지칭될 수도 있다. 뷰는 뷰의 디코딩이 하나 이상의 다른 뷰들과 연관되는 화상들의 디코딩에 의존하면, 비-베이스 뷰로서 지칭될 수도 있다.

[0100] 예를 들어, NAL 유닛들은 헤더들 (즉, NAL 유닛 헤더들) 및 페이로드들 (예컨대, RBSP들) 을 포함할 수도 있다. NAL 유닛 헤더들은 nuh_reserved_zero_6bits 구문 엘리먼트들을 포함할 수도 있다. 상이한 값들을 규정하는 nuh_reserved_zero_6bit 구문 엘리먼트들을 가지는 NAL 유닛들은 비트스트림의 상이한 "계층들" 에 속한다. 따라서, 멀티-뷰 코딩, 3DV, 또는 SVC 에서, NAL 유닛의 nuh_reserved_zero_6bits 구문 엘리먼트는 NAL 유닛의 계층 식별자 (즉, 계층 ID) 를 규정할 수도 있다. 일부 예들에서, NAL 유닛의 nuh_reserved_zero_6bits 구문 엘리먼트는, NAL 유닛이 멀티-뷰 코딩, 3DV 코딩, 또는 SVC 에서의 기초 계층에 관련되면 0 과 동일하다. 비트스트림의 기초 계층에서의 데이터는 비트스트림의 임의의 다른 계층에서의 데이터에 대한 참조 없이 디코딩될 수도 있다. NAL 유닛이 멀티-뷰 코딩, 3DV, 또는 SVC 에서의 기초 계층에 관련되지 않으면, nuh_reserved_zero_6bits 구문 엘리먼트는 비-제로 값을 가질 수도 있다. 위에서 나타낸 바와 같이, 멀티-뷰 코딩 및 3DV 코딩에서, 비트스트림의 상이한 계층들은 상이한 뷰들에 대응할 수도 있다. SVC 에서, 기초 계층 이외의 계층들은 "향상 계층들" 로서 지칭될 수도 있으며, 비트스트림으로부터 디코딩되는 비디오 데이터의 시각적 품질을 향상시키는 정보를 제공할 수도 있다.

[0101] 더욱이, 계층 내 일부 화상들은 동일한 계층 내 다른 화상들에 대한 참조 없이 디코딩될 수도 있다. 따라서, 계층의 어떤 화상들의 데이터를 캡슐화하는 NAL 유닛들은 그 계층에서의 다른 화상들의 디코딩성 (decodability) 에 영향을 미침이 없이 비트스트림으로부터 제거될 수도 있다. 이러한 화상들의 데이터를 캡슐화하는 NAL 유닛들을 제거하는 것은 비트스트림의 프레임 레이트를 감소시킬 수도 있다. 계층 내 다른 화상들을 참조하지 않고 디코딩될 수도 있는 계층 내 화상들의 서브세트는 본원에서 "서브-계층" 또는 "시간 서브-계층" 로서 지칭될 수도 있다.

[0102] NAL 유닛들은 또한 temporal_id 구문 엘리먼트들을 포함할 수도 있다. NAL 유닛의 temporal_id 구문 엘리먼트는 NAL 유닛의 시간 식별자를 규정한다. NAL 유닛의 시간 식별자는 NAL 유닛이 연관되는 서브-계층을 식별한다. 따라서, 비트스트림의 각각의 서브-계층은 상이한 시간 식별자와 연관될 수도 있다. 제 1 NAL 유닛의 시간 식별자가 제 2 NAL 유닛의 시간 식별자 미만이면, 제 1 NAL 유닛에 의해 캡슐화되는 데이터는 제 2 NAL 유닛에 의해 캡슐화되는 데이터에 대한 참조 없이 디코딩될 수도 있다.

[0103] 비-베이스 뷰들 중 하나에서의 화상을 코딩할 때, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 화상이 비디오 코더가 현재의 코딩하고 있는 화상과는 상이한 뷰와 연관되지만 비디오 코더가 현재 코딩하고 있는 화상과 동일한 시간 인스턴스 (즉, 액세스 유닛) 와 연관되면, 화상을 참조 화상 리스트에 추가할 수도 있다. 다른 인터 예측 참조 화상들과 유사하게, 비디오 코더는 인터-뷰 예측 참조 화상을 참조 화상 리스트의 임의의 위치에 인터 예측 참조 화상에서와 동일한 방법으로 삽입할 수도 있다. 다시 말해서, 인터-뷰 예측으로 코딩되는 화상은 다른 비-베이스 뷰들의 인터-뷰 예측을 위해 참조 화상 리스트에 추가될 수도 있다.

[0104] H.264/AVC 의 MVC 확장판에서, 인터-뷰 예측은, H.264/AVC 모션 보상의 구문을 이용하지만 상이한 뷰에서의 화상이 참조 화상으로서 사용되도록 허용하는 디스패리티 모션 보상 (즉, 인터-뷰 모션 예측) 에 의해 지원된다. 2개의 뷰들의 코딩은 또한 H.264/AVC 의 MVC 확장판에 의해 지원될 수도 있다. H.264/AVC 의 MVC 확장판의 이점들 중 하나는, MVC 인코더가 2개보다 많은 뷰들을 3D 비디오 입력으로서 취할 수도 있으며 MVC 디코더가 이러한 멀티-뷰 표현을 디코딩할 수도 있다는 점이다. 그 결과, MVC 디코더를 가진 임의의 렌더러는 2개보다 많은 뷰들을 가진 3D 비디오 콘텐츠를 기대할 수도 있다.

[0105] MVC 에서, 인터-뷰 예측은 동일한 액세스 유닛에서 (즉, 동일한 시간 인스턴스를 가진) 화상들 사이에 허용된다. 비-베이스 뷰에서의 화상을 코딩할 때, 비디오 코더는 참조 화상 리스트에, 인터-뷰 참조 화상 (즉, 현재의 화상과는 상이한 뷰와 연관되지만 현재의 화상과 동일한 시간 인스턴스와 연관되는 화상) 을 포함할 수도 있다. 인터-뷰 참조 화상은 마치 임의의 인터 예측 참조 화상처럼, 참조 화상 리스트의 임의의 위치에 삽입될 수 있다. 인터-뷰 참조 화상이 모션 보상을 위해 사용될 때, 대응하는 모션 벡터는 "디스패리티 모션 벡터" 로서 지칭된다.

[0106] 멀티-뷰 비디오 코딩의 상황에서, 2종류의 모션 벡터들이 존재한다. 모션 벡터의 한 종류는 시간 참조 화상을 가리키는 법선 (normal) 모션 벡터이다. 법선, 시간 모션 벡터에 대응하는 인터 예측의 유형은 모션-보상된 예측 (MCP) 으로서 지칭될 수도 있다. 인터-뷰 예측 참조 화상이 모션 보상을 위해 사용될 때, 대응하는 모션 벡터는 "디스패리티 모션 벡터" 로서 지칭될 수도 있다. 다시 말해서, 디스패리티 모션 벡터는 상이한 뷰에서의 화상 (즉, 디스패리티 참조 화상 또는 인터-뷰 참조 화상) 을 가리킨다. 디스패리티 모션 벡터에 대응하는 인터 예측의 유형은 "디스패리티-보상된 예측" 또는 "DCP" 로서 지칭될 수도 있다.

[0107] 도 3 은 예시적인 멀티-뷰 디코딩 순서를 예시하는 개념도이다. 멀티-뷰 디코딩 순서는 비트스트림 순서일 수도 있다. 도 3 의 예에서, 각각의 정사각형은 뷰 성분에 대응한다. 정사각형들의 칼럼들은 액세스 유닛들에 대응한다. 각각의 액세스 유닛은 시간 인스턴스의 모든 뷰들의 코딩된 화상들을 포함하도록 정의될 수도 있다. 정사각형들의 로우들은 뷰들에 대응한다. 도 3 의 예에서, 액세스 유닛들은 T0…T8 로 라벨링되며, 뷰들은 S0…S7 로 라벨링된다. 액세스 유닛의 각각의 뷰 성분이 다음 액세스 유닛의 임의의 뷰 성분 이전에 디코딩되기 때문에, 도 3 의 디코딩 순서는 시간-우선 코딩으로서 지칭될 수도 있다. 액세스 유닛들의 디코딩 순서는 출력 또는 디스플레이 순서와 동일하지 않을 수도 있다.

[0108] 멀티-뷰 코딩은 인터-뷰 예측을 지원할 수도 있다. 인터-뷰 예측은 H.264/AVC, HEVC, 또는 다른 비디오 코딩 사양들에 사용되는 인터 예측과 유사하며, 동일한 구문 엘리먼트들을 이용할 수도 있다. 그러나, 비디오 코더가 (매크로블록 또는 PU 와 같은)현재의 비디오 유닛에 관해 인터-뷰 예측을 수행할 때, 비디오 코더는 참조 화상으로서, 현재의 비디오 유닛과 동일한 액세스 유닛에 있지만 상이한 뷰에 있는 화상을 이용할 수도 있다. 이에 반해, 종래의 인터 예측은 단지 상이한 액세스 유닛들에서의 화상들을 참조 화상들로서 이용한다.

[0109] 도 4 는 멀티-뷰 코딩을 위한 예시적인 예측 구조를 예시하는 개념도이다. 도 4 의 멀티-뷰 예측 구조는 시간 및 인터-뷰 예측을 포함한다. 도 4 의 예에서, 각각의 정사각형은 뷰 성분에 대응한다. 도 4 의 예에서, 액세스 유닛들은 T0…T11 로 라벨링되며, 뷰들은 S0…S7 로 라벨링된다. "I" 로 라벨링된 정사각형들은 인트라 예측된 뷰 성분들이다. "P" 로 라벨링된 정사각형들은 단방향 인터 예측된 뷰 성분들이다. "B" 및 "b" 로 라벨링된 정사각형들은 양방향으로 인터 예측된 뷰 성분들이다. "b" 로 라벨링된 정사각형들은 "B" 로 라벨링된 정사각형들을 참조 화상들로서 이용할 수도 있다. 제 1 정사각형으로부터 제 2 정사각형까지 가리키는 화살표는 제 1 정사각형이 제 2 정사각형에 대한 참조 화상으로서 인터 예측에서 이용가능하는 것을 나타낸다. 도 4 에서 수직 화살표들로 표시된 바와 같이, 동일한 액세스 유닛의 상이한 뷰들에서의 뷰 성분들은 참조 화상들로서 이용가능할 수도 있다. 따라서, 도 4 는 멀티-뷰 비디오 코딩을 위한 (각각의 뷰 내 인터-화상 예측 및 인터-뷰 예측 양쪽을 포함한) 전형적인 MVC 예측 구조를 나타내며, 여기서 예측들은 화살표들로 표시되며, 지시도달 오브젝트는 예측 참조를 위해 지시출발 오브젝트를 이용한다. 동일한 액세스 유닛의 또 다른 뷰 성분에 대한 참조 화상으로서의 액세스 유닛의 하나의 뷰 성분의 사용은 인터-뷰 예측로서 지칭될 수도 있다.

[0110] 위에서 언급한 바와 같이, HEVC 의 멀티-뷰 확장판 (즉, MV-HEVC) 및 HEVC 의 3DV 확장판 (즉, 3D-HEVC) 이 개발되고 있다. 즉, VCEG 및 MPEG 의 3D 비디오 코딩 (JCT-3V) 에 관한 합동 연구팀은 HEVC 에 기초한 3DV 표준을 개발하고 있으며, 표준화 노력들의 부분은 HEVC (MV-HEVC) 에 기초한 멀티-뷰 비디오 코덱의 표준화 및 HEVC (3D-HEVC) 에 기초한 3D 비디오 코딩을 위한 또 다른 부분을 포함한다.

[0111] Gerhard Tech 등, "3D-HEVC Test Model Description draft 1", JCT3V-A1005, ITU-T SG 16 WP 3 와 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장판 개발에 관한 합동 협업팀, 1차 회의: 2012년 7월 16-20일, 스웨덴, 스톡홀름 (이하, "3D-HEVC 테스트 모델 1") 는, 3D-HEVC 의 참조 소프트웨어 설명 뿐만 아니라, 작업 초안을 제공한다. 3D-HEVC, 즉 3DV-HTM 에 대한 참조 소프트웨어는 https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/trunk 에서 입수가능하다. Gerhard Tech 등, "MV-HEVC Working Draft 1", JCT3V-A1004, ITU-T SG 16 WP 3 와 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장판 개발에 관한 합동 협업팀, 1차 회의: 2012년 7월 16-20일, 스웨덴, 스톡홀름, (이하, "JCT3V-A1004") 는 MV-HEVC 에 대한 작업 초안을 제공한다.

[0112] HEVC 의 무작위 액세스 컨셉이 멀티-뷰 및 3DV 확장판들로 확장된다. JCT3V-A1004 는 무작위 액세스 뷰 성분들 뿐만 아니라 무작위 액세스 지점 액세스 유닛들의 상세한 정의를 포함한다. 예를 들어, JCT3V-A1004 는, 단지 슬라이스들을 포함하고 각각의 슬라이스가 7 내지 12 의 범위에서의 nal_unit_type 을 가지는 뷰 성분으로서, 무작위 액세스 지점 (RAP) 뷰 성분을 정의한다. 무작위 액세스 지점 뷰 성분은 인터 예측으로부터 예측되지 않는다. 따라서, HEVC 의 멀티-뷰 또는 3D 확장판 (즉, MV-HEVC 또는 3D-HEVC) 에서, 뷰 성분이 무작위 액세스 지점인지 여부는 뷰 성분의 NAL 유닛 유형에 의존한다. 유형이 무작위 액세스 지점 화상들에 대한 HEVC 기본 사양에 정의되어 있는 것들에 속하면, 현재의 뷰 성분은 무작위 액세스 지점 뷰 성분 (또는, 간결성을 위해, 현재의 뷰의 무작위 액세스 지점 화상) 이다. 무작위 액세스 지점 (RAP) 뷰 성분은 P 또는 B 화상일 수도 있다. P 화상은 단방향-예측적 인터 예측 및 인트라 예측이 허용되지만 양방향-예측적 인터 예측이 허용되지 않는 화상이다 B 화상은 단방향-예측적 인터 예측, 양방향-예측적 인터 예측, 및 인트라 예측이 허용되는 화상이다. 더욱이, JCT3V-A1004 는 무작위 액세스 지점 액세스 유닛을 그 코딩된 화상이 RAP 화상인 액세스 유닛로서 정의한다.

[0113] 일부 예들에서, 무작위 액세스 기능이, 시간 치수에서의 (따라서, 뷰 내) 어떤 예측들이 HEVC 기본 사양에서와 같이 유사하게 이용불능되거나 또는 구속되는 방법으로, 단지 시간 예측에, 적용된다. 그러나, 무작위 액세스 지점 뷰 성분에 대한 인터-뷰 예측이 H.264/MVC 에서의 앵커 화상과 유사하게, 코딩 효율을 향상시키는데, 여전히 가능하고 전형적일 수도 있다.

[0114] 비디오 코더는 블록들 (예컨대, PU들, CU들, 등) 에 대한 디스패리티 벡터들을 결정할 수도 있다. 일반적으로, 디스패리티 벡터는 2개의 뷰들 사이의 변위의 추정자로서 사용된다. 비디오 코더는 인터-뷰 모션 또는 잔여 예측을 위해 블록에 대한 디스패리티 벡터를 이용하여 참조 블록을 또 다른 뷰에 로케이트할 수도 있거나, 또는 비디오 코더는 인터-뷰 모션 예측을 위해 디스패리티 벡터를 디스패리티 모션 벡터로 변환할 수도 있다.

[0115] 비디오 코더가 현재의 블록 (예컨대, 현재의 PU, CU, 등) 에 대해 인터-뷰 모션 예측을 수행할 때, 비디오 코더는 현재의 블록에 대한 디스패리티 벡터를 이용하여, 참조 뷰에서 대응하는 블록 (예컨대, 참조 PU) 을 식별할 수도 있다. 비디오 코더는 그후 대응하는 블록의 모션 정보 (예컨대, 모션 벡터, 참조 인덱스, 등) 를 병합 모드 또는 AMVP 모드에 대한 후보 리스트에서의 후보로서 이용할 수도 있다. 다시 말해서, 현재의 블록의 대응하는 블록은 디스패리티 벡터에 의해 식별되며, 그의 모션 벡터들이 현재의 블록의 AMVP 또는 병합 리스트의 추가적인 후보로서 이용될 수도 있다. 더욱이, 일부 예들에서, 비디오 코더는 현재의 블록에 대한 디스패리티 벡터를 병합 모드 또는 AMVP 모드에 대한 후보 리스트에서의 후보로서 이용할 수도 있다. 즉, 디스패리티 벡터가 디스패리티 모션 벡터로 변환되어 AMVP 또는 병합 리스트에 추가될 수도 있다.

[0116] 더욱이, 비디오 코더는 블록 (예컨대, PU, CU, 등) 의 디스패리티 벡터를 이용하여 인터-뷰 잔여 예측을 수행할 수도 있다. 비디오 인코더 (20) 가 인터-뷰 잔여 예측을 수행할 때, 비디오 인코더 (20) 는 현재의 블록에 대한 디스패리티 벡터를 이용하여 참조 뷰에서 대응하는 블록을 식별할 수도 있다. 비디오 인코더 (20) 는 그후 대응하는 블록에 대한 잔여를 결정할 수도 있다. 비디오 인코더 (20) 는 그후 대응하는 블록에 대한 잔여와 현재의 블록에 대한 잔여 사이의 차이를 시그널링할 수도 있다. 비디오 디코더 (30) 는 현재의 블록에 대한 디스패리티 벡터를 이용하여 대응하는 블록을 식별할 수도 있으며 대응하는 블록에 대한 잔여를 결정할 수도 있다. 비디오 코더는 그후 그 시그널링된 잔여의 샘플들을 대응하는 블록에 대한 잔여의 대응하는 샘플들 및 현재의 블록에 대한 예측 블록의 대응하는 샘플들에 가산하여, 현재의 블록의 샘플들을 재구성할 수도 있다. 이러한 방법으로, 현재의 블록이 현재의 CU 이면 그리고 현재의 CU 의 대응하는 블록이 비-제로 잔여 픽셀들을 포함하면, 블록들의 나머지는 현재의 CU 의 나머지를 예측하는데 사용된다.

[0117] 일부 예들에서, 비디오 코더는 이웃하는 블록들 기반의 디스패리티 벡터 (NBDV) 의 방법을 이용하여 블록에 대한 디스패리티 벡터를 유도할 수도 있다. 3D-HEVC 는 L. Zhang 등, "3D-CE5.h: Disparity vector

generation results", ITU-T SG 16 WP 3 와 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장판 개발에 관한 합동 협업팀, 1차 회의: 2012년 7월 16-20일, 스웨덴, 스톡홀름, 문서 JCT3V-A0097 (이하, "JCT3V-A0097") 에서 제안된 NBDV 유도 프로세스를 먼저 채택하였다. NBDV 유도 프로세스가 그 이후 추가로 채택되었다. 예를 들어, 암시적인 디스패리티 벡터들 (IDVs) 은 Sung 등, "3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding", ITU-T SG 16 WP 3 와 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장판 개발에 관한 합동 협업팀, 1차 회의: 2012년 7월 16-20일, 스웨덴, 스톡홀름, 문서 JCT3V-A0126 (이하, "JCT3V-A0126") 에서의 단순화된 NBDV 와 함께 포함되었다. 더욱이, Kang 등, "3D-CE5.h related: Improvements for disparity vector derivation", ITU-T SG 16 WP 3 와 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장판 개발에 관한 합동 협업팀, 2차 회의: 2012년 10월 13-19일, 중국, 상하이, 문서 JCT3V-B0047 (이하, "JCT3V-B0047") 에서, NBDV 유도 프로세스는 디코딩 화상 버퍼에 저장되는 IDVs 를 제거함과 동시에, RAP 화상 선택에 향상된 코딩 이득을 제공함으로써, 더욱 단순화된다.

[0118] 멀티-뷰 비디오 코딩에서 디스패리티 모션 보상은 디스패리티 모션 벡터들로 수행된다. 이웃하는 블록들 (예컨대, 현재의 블록에 공간적으로 또는 시간적으로 이웃하는 블록들) 이 비디오 코딩에서 거의 동일한 모션/디스패리티 정보를 공유하기 때문에, 현재의 블록은 이웃하는 블록들에서의 모션 벡터 정보를 우수한 예측자로서 이용하여, 코딩 이득을 향상시킬 수 있다. 이 아이디어에 뒤이어서, NBDV 유도 프로세스는 상이한 뷰들에서의 디스패리티 벡터를 추정하기 위해 이웃하는 디스패리티 정보를 이용한다. 구체적으로 설명하면, NBDV 유도 프로세스는 공간 및 시간 이웃하는 블록들로부터의 디스패리티 모션 벡터들을 이용하여, 현재의 블록에 대한 디스패리티 벡터를 유도한다.

[0119] 비디오 코더가 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 이웃하는 블록들의 2개의 세트들을 이용할 수도 있다. 하나의 세트는 공간적으로-이웃하는 블록들로부터 유래하며, 다른 세트는 시간적으로-이웃하는 블록들로부터 유래한다. 비디오 코더는 공간 이웃하는 블록들 및 시간 이웃하는 블록들을 사전-정의된 순서로 체킹할 수도 있다. 사전-정의된 순서는 현재의 블록과 후보 블록 사이의 상관의 우선순위에 의해 결정될 수도 있다. 일단 디스패리티 모션 벡터가 후보들에서 발견되면, 비디오 코더는 디스패리티 모션 벡터를 디스패리티 벡터로 변환할 수도 있다.

[0120] NBDV 유도 프로세스의 일부 버전들에서, 비디오 코더는 디스패리티 벡터 유도를 위해 5개의 공간 이웃하는 블록들을 이용한다. 예를 들어, 비디오 코더는 다음 공간적으로-이웃하는 블록들, 즉, 현재의 블록의 좌하측 공간적으로-이웃하는 블록, 좌측 공간적으로-이웃하는 블록, 상부-우측 공간적으로-이웃하는 블록, 상부 공간적으로-이웃하는 블록, 및 좌상부 공간적으로-이웃하는 블록을 체킹할 수도 있다. 더욱이, NBDV 유도 프로세스의 일부 버전들에서, 5개의 공간적으로-이웃하는 블록들이 디스패리티 벡터 유도에 사용되며, 블록들은 도 2 에 나타낸 바와 같이 로케이션들 A_0, A_1, B_0, B_1, 및 B_2 을 각각 커버할 수도 있다. 일부 예들에서, NBDV 유도 프로세스에 사용되는 공간적으로-이웃하는 블록들은 HEVC 에서 병합 모드들에서 사용되는 것들과 동일하다. 따라서, 일부 이러한 예들에서, 어떤 추가적인 메모리 액세스도 요구되지 않는다.

[0121] 더욱이, 위에서 언급한 바와 같이, 비디오 코더는 시간적으로-이웃하는 PU들을, 현재의 블록 (예컨대, 현재의 PU) 에 대한 디스패리티 벡터를 결정하는 프로세스의 일부로서 체킹할 수도 있다. 비디오 코더가 시간 이웃하는 블록들 (예컨대, 시간 이웃하는 PU들) 을 체킹할 때, 비디오 코더는 후보 화상 리스트의 구성 프로세스를 먼저 수행할 수도 있다. 비디오 코더가 후보 화상 리스트의 구성 프로세스를 수행할 때, 비디오 코더는 현재의 뷰로부터의 모든 참조 화상들 (즉, 현재의 블록과 연관되는 뷰) 를 후보 화상들로서 취급할 수도 있다. 더욱이, 비디오 코더가 후보 화상 리스트의 구성 프로세스를 수행할 때, 비디오 코더는 먼저 소위 "동일 장소에 배치된 화상" 을 후보 화상 리스트에, 다음으로 후보 화상들의 나머지를 참조 인덱스의 오름 차순으로 삽입할 수도 있다. 즉, 비디오 코더는 나머지 후보 화상들이 현재의 화상의 참조 화상 리스트들 (예컨대, RefPicList0 및 RefPicList1) 에서 발생하는 순서에 따라서, 나머지 후보 화상들을 후보 화상 리스트에 삽입할 수도 있다. 현재의 블록을 포함하는 슬라이스의 슬라이스 헤더에서의 하나 이상의 구문 엘리먼트들은 동일 장소에 배치된 화상을 나타낼 수도 있다. 양쪽의 참조 화상 리스트들 (예컨대, RefPicList0 및 RefPicList1) 에서 동일한 참조 인덱스를 가진 참조 화상들이 NDBV 유도 프로세스에 사용하기 위해 이용가능할 때, 동일 장소에 배치된 화상과 동일한 참조 화상 리스트에서의 참조 화상은 후보 화상 리스트에서, 다른 참조 화상보다 선행한다.

[0122] 후보 화상 리스트를 발생한 후, 비디오 코더는 후보 화상 리스트에서 후보 화상들 내 후보 영역들을 결정할 수도 있다. 비디오 코더는 후보 영역들을 이용하여, 시간적으로-이웃하는 블록들을 결정할 수도 있다. 위

에서 나타낸 바와 같이, 비디오 코더는 시간적으로-이웃하는 블록의 디스패리티 모션 벡터 또는 IDV 에 기초하여 현재의 블록에 대한 디스패리티 벡터를 유도할 수도 있다. 일부 예들에서, 후보 화상 리스트에서의 각각의 후보 화상에 대해, 비디오 코더는 3개의 후보 영역들을 결정할 수도 있다. 3개의 후보 영역들은 다음과 같이 정의될 수도 있다:

[0123] - CPU: 현재의 PU 또는 현재의 CU 의 동일 장소에 배치된 영역.

[0124] - CLCU: 현재의 PU 의 동일 장소에 배치된 영역을 커버하는 최대 코딩 유닛 (LCU).

[0125] - BR: CPU 의 우하단 4x4 블록.

[0126] 16x16 블록에서의 더 작은 블록들이 모션 압축의 결과로서 동일한 모션 정보를 공유하기 때문에, 비디오 코더는 오직 하나의 샘플 블록을 디스패리티 벡터에 대해 체킹할 수도 있다. 후보 영역이 하나 보다 많은 16x16 블록을 커버할 때, 비디오 코더는 래스터 스캐닝 순서에 따라서 후보 영역에서 모든 16x16 블록들을 체킹할 수도 있다. 예를 들어, 시간적으로 동일 장소에 배치된 블록에 대한 모션 벡터는 참조 화상의 16x16 블록으로 저장되며, 일반적으로, 비디오 코더는 4x4 블록에 액세스하여 모션 벡터를 찾는다. 따라서, 비디오 코더가 후보 블록을 16x16 블록으로 배치하면, 모든 4x4 블록들이 공통 모션 벡터를 포함하며, 비디오 코더는 모든 4x4 블록들을 체킹하여 상이한 모션 벡터를 찾을 필요가 없다. 한편, 후보 영역이 16x16 보다 더 크면, 16x16 블록 외부의 4x4 블록들은 상이한 모션 벡터를 포함할 수도 있다.

[0127] 비디오 코더가 후보 영역 (또는, 후보 영역 내 16x16 블록) 을 체킹할 때, 비디오 코더는 후보 영역을 커버하는 PU 가 디스패리티 모션 벡터를 규정하는지 여부를 결정할 수도 있다. 후보 영역을 커버하는 PU 가 디스패리티 모션 벡터를 규정하면, 비디오 코더는 PU 의 디스패리티 모션 벡터에 기초하여 현재의 블록의 디스패리티 벡터를 결정할 수도 있다.

[0128] 일부 예들에서, 비디오 코더는 NBDV 유도 프로세스를 수행하는 것의 일부로서 우선순위-기반의 디스패리티 벡터 결정을 수행할 수도 있다. 예를 들어, 비디오 코더는, 일단 비디오 코더가 디스패리티 모션 벡터를 포함하는 이웃하는 블록을 식별하면 비디오 코더가 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환하도록, 디스패리티 벡터를 유도할 수도 있다. 비디오 코더는 그후 그 디스패리티 벡터를 인터-뷰 모션 예측 및/또는 인터-뷰 잔여 예측에 이용할 수도 있다. 일부 예들에서, 이웃하는 블록들의 체킹 순서는 이웃하는 블록들과 현재의 블록 사이의 상관에 기초하여 정의된다. 예를 들어, 비디오 코더는 먼저 공간 이웃하는 블록들을 하나하나씩 체킹할 수도 있다. 일단 비디오 코더가 디스패리티 모션 벡터를 식별하였으면, 비디오 코더는 디스패리티 모션 벡터를 디스패리티 벡터로서 반환한다. 일부 예들에서, 5개의 공간 이웃하는 블록들의 체킹 순서는 A_1, B_1, B_0, A_0 및 B_2 로서 정의된다.

[0129] 더욱이, 후보 화상 리스트에서의 각각의 후보 화상에 대해, 비디오 코더는 이 후보 화상에서 순서대로 3개의 후보 영역들을 체킹할 수도 있다. 3개의 영역들의 체킹 순서는 다음과 같이 정의된다: 제 1 비-베이스 뷰에 대해 CPU, CLCU 및 BR 또는 제 2 비-베이스 뷰에 대해 BR, CPU, CLU. 이 예에서, 제 1 비-베이스 뷰와 연관되는 화상들의 디코딩은 베이스 뷰와 연관되는 화상들의 디코딩에 의존할 수도 있으나, 다른 뷰들과 연관되는 화상들의 디코딩에는 의존하지 않을 수도 있다. 더욱이, 이 예에서, 제 2 비-베이스 뷰와 연관되는 화상들의 디코딩은 베이스 뷰와 연관되는 화상들의 디코딩에 의존할 수도 있으며, 그러나, 일부 경우, 제 1 비-베이스 뷰는, 다른 뷰들과 연관되는 화상들에, 존재하면, 의존하지 않을 수도 있다. 간결성을 위해, 공간 이웃하는 블록들에서의 디스패리티 모션 벡터들은 SDVs 로서 표시될 수도 있으며, 시간 이웃하는 블록들에서의 디스패리티 모션 벡터들은 TDVs 로서 표시될 수도 있다.

[0130] 비디오 코더가 블록 (즉, 공간적으로-이웃하는 블록, 후보 화상의 후보 영역, 또는 후보 화상의 후보 영역의 16x16 블록) 의 모션 벡터(들) 을 체킹할 때, 비디오 코더는 블록의 모션 벡터(들) 이 디스패리티 모션 벡터들인지 여부를 결정할 수도 있다. 화상의 블록의 디스패리티 모션 벡터는 화상의 디스패리티 참조 화상 내 로케이션을 가리키는 모션 벡터이다. 주어진 화상의 디스패리티 참조 화상은 주어진 화상과 동일한 액세스 유닛과 연관되지만 주어진 화상과는 상이한 뷰와 연관되는 화상일 수도 있다. 비디오 코더가 디스패리티 모션 벡터를 식별할 때, 비디오 코더는 체킹 프로세스를 종료할 수도 있다. 비디오 코더는 그 반환된 디스패리티 모션 벡터를 디스패리티 벡터로 변환할 수도 있으며 디스패리티 벡터를 인터-뷰 모션 예측 및 인터-뷰 잔여 예측에 이용할 수도 있다. 예를 들어, 비디오 코더는 현재의 블록에 대한 디스패리티 벡터의 수평 성분을 디스패리티 모션 벡터의 수평 성분과 동일하게 설정할 수도 있으며, 디스패리티 벡터의 수직 성분을 0 으로 설정할 수도 있다.

[0131] 비디오 코더가 공간적으로-이웃하는 PU 를 체킹할 때, 비디오 코더는 이웃하는 PU 가 디스패리티 모션 벡터를 가지는지 여부를 먼저 체킹할 수도 있다. 공간적으로-이웃하는 PU들 중 어느 것도 디스패리티 모션 벡터를 갖지 않으면, 비디오 코더는 공간적으로-이웃하는 PU들 중 임의의 PU 가 IDV 를 가지는지 여부를 결정할 수도 있다. IDV 는 공간적으로- 또는 시간적으로-이웃하는 PU (예컨대, 인터-뷰 모션 예측 또는 인터-뷰 잔여 예측을 이용하여 코딩되는 PU) 의 디스패리티 벡터일 수도 있다. 예를 들어, 비디오 코더가 인터-뷰 모션 예측으로 블록을 코딩할 때, 비디오 코더는 상이한 뷰에서 대응하는 블록을 선택하기 위해 디스패리티 벡터를 유도할 필요가 있을 수도 있다. 이 예에서, 용어 "IDV" 는 인터-뷰 모션 예측에서 유도된 디스패리티 벡터를 지칭할 수도 있다. 설령 블록이 시간 모션 예측으로 코딩될 수 있더라도, 비디오 코더는 하나 이상의 다음 블록들을 코딩하는 목적을 위해 그 유도된 디스패리티 벡터를 폐기하지 않는다. 이러한 방법으로, IDV 는 디스패리티 벡터 유도의 목적을 위해 그 블록에 저장될 수도 있다.

[0132] 비디오 코더가 NBDV 유도 프로세스의 일부로서 IDVs 를 체킹하는 일부 예들에서, 비디오 코더는 다음 단계들을 수행할 수도 있다. 다음 단계들 중 임의의 단계가 디스패리티 벡터를 발견하면, 비디오 코더는 유도 프로세스를 종료할 수도 있다.

[0133] - 단계 1: 5개의 공간 이웃하는 블록들을 A_1, B_1, B_0, A_0 및 B_2 의 순서로 체킹하여, 디스패리티 모션 벡터를 찾는다. 일단 디스패리티 모션 벡터가 발견되면, 비디오 코더는 디스패리티 모션 벡터를 디스패리티 벡터로 변환한다. 공간 이웃하는 블록들이 IDVs 를 포함하면, 비디오 코더는 그들의 IDV 플래그들을 "IDV 사용됨" 으로 표시하고 IDV 플래그들의 연관된 값들을 저장한다.

[0134] - 단계 2: 시간 모션 벡터 예측이 이용가능하게 될 때, 다음이 적용된다:

[0135] a) 현재의 코딩 모드가 AMVP 이면, 목표 참조 화상 리스트에서 목표 참조 인덱스를 가진 참조 화상이 동일 장소에 배치된 화상으로서 사용된다. 동일 장소에 배치된 화상에서 2개의 블록들이 정의된다 (즉, 동일 장소에 배치된 PU (BR) 의 우하단 블록 및 동일 장소에 배치된 PU (CB) 의 중심 블록). 이 예에서, 비디오 코더는 동일 장소에 배치된 화상의 블록들을 다음 순서로 체킹한다:

[0136] 1) BR 을 체킹하여 BR 이 디스패리티 모션 벡터를 포함하는지 여부를 확인한다. 예이면, 비디오 코더는 디스패리티 모션 벡터를 디스패리티 벡터로 변환한다. 그렇지 않으면, BR 이 스킵 모드로서 코딩되고 BR 이 IDV 을 포함하면 (즉, IDV 의 플래그가 1 과 동일하면), 비디오 코더는 BR 을 "IDV 사용됨" 으로 표시하고, 그 연관된 IDV 를 저장한다. 비디오 코더는 그후 단계 3 을 수행할 수도 있다.

[0137] 2) CB 를 체킹하여, CB 가 디스패리티 모션 벡터를 포함하는지 여부를 확인한다. 예이면, 비디오 코더는 디스패리티 모션 벡터를 디스패리티 벡터로 변환한다. 그렇지 않으면, BR 이 스킵 모드로서 코딩되고 BR 이 IDV 를 포함하면 (즉, IDV 의 플래그가 1 과 동일하면), 비디오 코더는 BR 를 "IDV 사용됨" 으로 표시하고, 비디오 코더는 그 연관된 IDV 를 저장한다. 비디오 코더는 그후 단계 3 을 수행할 수도 있다.

[0138] b) 현재의 코딩 모드가 스킵/병합이면, 비디오 코더는 적용가능하면, 각각의 참조 화상 리스트에서 2개의 동일 장소에 배치된 참조 화상들을 사용한다. 동일 장소에 배치된 참조 화상들을 나타내는 참조 인덱스들은 좌측 이웃하는 PU 의 참조 인덱스 또는 0 과 동일할 수도 있다. 참조 화상 리스트들 0 및 1 에서 동일 장소에 배치된 화상들의 각각에 대해, 비디오 코더는 a) 1) 및 a) 2) 에서의 단계들을 순서대로 수행한다.

[0139] - 단계 3: 5개의 공간 이웃하는 블록들 중 하나가 스킵 모드를 이용하여 코딩되고 공간 이웃하는 블록이 IDV 를 포함하면 (즉, 공간 이웃하는 블록이 "IDV 사용됨" 으로 마크된 플래그를 가지면), 비디오 코더는 IDV 를 디스패리티 벡터로서 반환한다. 이 예에서, IDVs 에 대한 공간 이웃하는 블록들의 체킹 순서는 A_0, A_1, B_0, B_1, 및 B_2 이다.

[0140] - 단계 4: 시간 모션 벡터 예측이 이용가능하게 되고 "IDV 사용됨" 으로 표시되는 동일 장소에 배치된 화상 (즉, BR 또는 CB) 에서 하나의 블록이 존재하면, 비디오 코더는 블록과 연관되는 IDV 를 디스패리티 벡터로 변환한다.

[0141] 디코딩 화상 버퍼 (DPB) 내 IDV 에 액세스하는 것과 연관되는 메모리 대역폭 및 복잡성 요구사항들이 클 수도 있다. 예를 들어, 이웃하는 블록이 NBDV 유도 프로세스 동안 IDV 를 갖는지 여부를 결정하기 위해, 비디오 코더는 이웃하는 블록의 AMVP 또는 병합 후보 리스트를 재구성하고, 그 재구성된 AMVP 또는 병합 후보 리스트에

서 인터-뷰 모션 벡터 후보를 식별할 필요가 있을 수도 있다. 이웃하는 블록의 AMVP 또는 병합 후보 리스트의 재구성은 상대적으로 계산적 및 대역폭 집약적일 수도 있다. 따라서, 비디오 코더는 낮은-복잡성 NBDV 유도 프로세스를 수행할 수도 있다. 비디오 코더는 비디오 코더가 낮은 복잡성 NBDV 유도 프로세스를 수행할 때 더 적은 블록 후보들을 고려한다. 예를 들어, 비디오 코더는 DPB 에, IDVs 에 대한 정보를 저장할 수도 있다. 이 예에서, IDVs 에 대한 여분의 정보는 모든 이전에-코딩된 화상들에 대한 IDV 플래그들 및 벡터들을 포함할 수도 있다. 더욱이, 낮은 복잡성 NBDV 유도 프로세스에서, DPB 에서 IDV 후보들을 제거하는 것은 메모리 대역폭을 감소시킬 수 있다. 다시 말해서, 비디오 코더는 IDV-관련된 정보를 DPB 에 저장하지 않는다.

[0142] 일부 낮은-복잡성 NBDV 유도 프로세스들에서, 비디오 코더는 위에서 설명된 NBDV 유도 프로세스에서보다 후보 화상들의 더 적은 후보 영역들을 체킹한다. 예를 들어, 도 5 는 시간 후보 화상의 대응하는 PU 에서 시간 이웃들을 예시하는 개념도이다. 도 5 의 예에서, 비디오 코더는 "Pos. A" 및 "Pos. B" 로 표시되는 이 위치들을 커버하는 후보 영역들을 체킹할 수도 있다. 더욱이, 일부 낮은-복잡성 NBDV 유도 프로세스들에서, 비디오 코더는 단지 동일 장소에 배치된 화상 및 무작위 액세스 화상의 후보 영역들을 체킹할 수도 있다. 따라서, 일부 예들에서, 동일 장소에 배치된 화상 및 무작위 액세스 화상이 시간 블록 체크들을 위해 고려된다 (즉, 도 5 에 나타낸 바와 같이 저부-하부 및 중심 블록들).

[0143] 더욱이, 일부 낮은-복잡성 NBDV 유도 프로세스들에서, 비디오 코더는 후보 화상 유도를 슬라이스 또는 화상 레벨에서 한번 수행할 수도 있다. 다시 말해서, 비디오 코더는 NBDV 유도 프로세스에서 사용하기 위한 후보 화상 리스트를 화상 또는 슬라이스 당 한번 발생할 수도 있다. 그 결과, 이러한 낮은-복잡성 NBDV 유도 프로세스들에서, 비디오 코더는 후보 화상 유도 프로세스를 PU 또는 CU 레벨에서 더 이상 호출하지 않는다.

[0144] 3D-HEVC 에서 디스패리티 벡터를 유도하는 프로세스들은 잠재적으로 다음 이슈들을 가질 수도 있다. 예를 들어, NBDV 유도 프로세스가 인터-뷰 모션 예측을 위해 모든 PU 에서 호출되며, 이것은 컴퓨터 복잡성을 현저하게 증가시킬 수도 있다. 더욱이, NBDV 유도 프로세스 동안 체킹되는 공간 이웃하는 블록들의 개수는 잠재적으로 크며, 일부 공간 이웃하는 블록들은 동일한 또는 유사한 코딩 효율을 달성하도록 체킹될 필요가 없을지도 모른다.

[0145] 본 개시물의 기법들은 3D-HEVC 에서의 디스패리티 벡터 유도 프로세스 (즉, NBDV) 에 여러 향상들 및 단순화들을 제공할 수도 있다. 예를 들어, CU-기반의 디스패리티 벡터 유도는 디스패리티 유도 프로세스가 CU 에 대해 오직 한번 호출되는 방법일 수도 있으며, CU 내부에 있는 모든 PU들은 예컨대, 인터-뷰 모션 예측, 및 인터-뷰 잔여 예측을 포함한, 인터-뷰 예측에 있어, 동일한 유도된 디스패리티 벡터를 공유한다. CU-기반의 디스패리티 벡터 유도는 컴퓨터 복잡성의 증가 및 병렬 프로세싱의 향상의 향상을 초래할 수도 있다.

[0146] 예를 들어, 비디오 인코더 (20) 는 비디오 데이터의 코딩된 표현을 포함하는 비트스트림을 발생할 수도 있다. 비트스트림을 발생하는 것의 일부로서, 비디오 인코더 (20) 는 디스패리티 벡터 유도 프로세스를 수행하여, 복수의 블록들에서 대표 블록에 대한 디스패리티 벡터를 유도할 수도 있다. 이 예에서, 비디오 데이터는 복수의 블록들로 파티셔닝되는 부모 블록을 포함한다. 더욱이, 이 예에서, 비디오 인코더 (20) 는 그 유도된 디스패리티 벡터에 기초하여, 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들 (즉, 부모 블록의 2개 이상의 블록들) 에 대해 인터-뷰 예측을 부분적으로 수행함으로써, 비디오 데이터의 코딩된 표현을 포함하는 비트스트림을 발생할 수도 있다.

[0147] 또 다른 예에서, 비디오 디코더 (30) 는 디스패리티 벡터 유도 프로세스를 수행하여, 복수의 블록들에서 대표 블록에 대한 디스패리티 벡터를 유도할 수도 있다. 이 예에서, 비디오 데이터는 복수의 블록들로 파티셔닝되는 부모 블록을 포함한다. 더욱이, 이 예에서, 비디오 디코더 (30) 는 그 유도된 디스패리티 벡터에 기초하여, 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측을 부분적으로 수행함으로써, 복수의 블록들 (예컨대, 부모 블록) 에서의 2개 이상의 블록들에 대해 샘플 블록들을 재구성할 수도 있다.

[0148] 또 다른 본 개시물의 예시적인 기법에서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 수정된 NBDV 유도 프로세스를 수행하여, PU 에 대한 디스패리티 벡터를 유도할 수도 있다. 비디오 코더가 수정된 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 3D-HEVC 테스트 모델 1 에서 설명되는 NBDV 유도 프로세스보다, SDVs 및/또는 IDVs 를 유도하기 위해 더 적은 공간 이웃하는 블록들을 체킹한다. 예를 들어, 비디오 코더가 수정된 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 단지 좌측 블록 및 상부 블록 (도 2 에

서 A_1 및 B_1 로 표시됨) 을 체킹한다. 더욱이, 이 예에서, 비디오 코더는 좌측 블록을 체킹하고 그후 상부 블록을 체킹할 수도 있다. 다른 예들에서, 비디오 코더는 상부 블록을 체킹하고 그후 좌측 블록을 체킹할 수도 있다. 본 개시물의 수정된 NBDV 유도 프로세스는 3D-HEVC 테스트 모델 1 에서 설명되는 NBDV 유도 프로세스에 비해 코딩 이득에 대한 최소의 변화 및 감소된 복잡성을 초래할 수도 있다.

[0149] 예를 들어, 비디오 인코더 (20) 는 현재의 블록에 대한 디스패리티 벡터를 결정하는 디스패리티 벡터 유도 프로세스를 수행할 수도 있다. 디스패리티 벡터 유도 프로세스를 수행하는 것의 일부로서, 비디오 인코더 (20) 는 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 IDV 를 가질 때, 디스패리티 모션 벡터 또는 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 이 예에서, 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 IDV 를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 IDVs 를 가지는 여부를 결정하지 않는다. 일부 예들에서, 제 1 공간 이웃하는 블록은 현재의 블록 (즉, 상기 이웃하는 블록) 의 상부 에지에 인접하며, 제 2 공간 이웃하는 블록은 현재의 블록 (즉, 좌측 이웃하는 블록) 의 좌측 에지에 인접한다. 본 개시물은 블록 (예컨대, CTU 의 코딩 트리 블록, CU 의 코딩 블록, PU 의 예측 블록, 등) 에 대응하는 샘플들의 블록의 외부 에지를 블록의 에지로서 지칭할 수도 있다. 더욱이, 이러한 예들에서, 비디오 인코더 (20) 는 현재의 블록에 대한 유도된 디스패리티 벡터에 부분적으로 기초하여, 현재의 블록의 인코딩된 표현을 발생할 수도 있다.

[0150] 또 다른 예에서, 비디오 디코더 (30) 는 현재의 블록에 대한 디스패리티 벡터를 결정하는 디스패리티 벡터 유도 프로세스를 수행할 수도 있다. 디스패리티 벡터 유도 프로세스를 수행하는 것의 일부로서, 비디오 디코더 (30) 는 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 IDV 를 가질 때, 디스패리티 모션 벡터 또는 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 IDV 를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 IDVs 를 가지는 여부를 결정하지 않는다. 일부 예들에서, 제 1 공간 이웃하는 블록은 현재의 블록 (즉, 좌측 이웃하는 블록) 의 좌측 에지에 인접하며, 제 2 공간 이웃하는 블록은 현재의 블록 (즉, 상기 이웃하는 블록) 의 상부 에지에 인접한다. 더욱이, 비디오 디코더 (30) 는 그 유도된 디스패리티 벡터에 기초하여, 현재의 블록에 대한 인터-뷰 예측을 부분적으로 수행함으로써, 현재의 블록에 대한 샘플 블록을 재구성할 수도 있다.

[0151] 3D-HEVC 디스패리티 벡터를 유도하는 프로세스에 대한 또 다른 예시적 잠재 이슈로서, 비디오 코더는 공간 이웃하는 블록이 IDV 를 포함하는지 여부를 표시하기 위해 모든 블록과 연관되는 플래그를 저장한다. 이러한 플래그들을 저장하는 것은 시스템에서 메모리 요구사항들을 증가시킬 수도 있다. 더욱이, 각각의 블록은 최고 2개의 IDVs (즉, RefPicList0 에 대응하는 IDV 및 RefPicList1 에 대응하는 IDV) 를 포함할 수도 있다. 본 개시물에서 설명하는 기법들에서, 이러한 플래그들의 저장은 필요하지 않을 수도 있다.

[0152] 본 개시물의 예시적인 기법에 따르면, IDVs 는 공간 이웃하는 블록이 IDV 를 포함하는지 여부를 시그널링하는 것에 더해서, 사용되는 어떤 플래그도 존재하지 않도록, 간단한 방법으로 표현된다. 더욱이, 참조 인덱스, 참조 뷰 인덱스 또는 참조 뷰 인덱스의 차이가 블록에 대해 시그널링되며, 불법 참조 인덱스 (즉, 1), 불법 참조 뷰 인덱스 (예컨대, 현재의 뷰와 동일한 것), 및/또는 불법 참조 뷰 인덱스의 차이 (예컨대, 0) 는 암시적인 디스패리티 벡터가 블록에 대해 존재하지 않는다는 것을 나타낸다. 게다가, 오직 하나의 디스패리티 벡터는 하나의 공간 이웃하는 블록에 대해 존재할 수도 있다.

[0153] 따라서, 일부 이러한 예들에서, 공간 이웃하는 블록이 IDV 를 포함하는지 여부를 표시하기 위해, 어떤 플래그도 시그널링되지 않는다. 예를 들어, 비디오 인코더 (20) 는 공간 이웃하는 블록이 IDV 를 포함하는지 여부를 표시하기 위해 어떤 플래그도 시그널링하지 않을 수도 있다. 더욱이, 일부 이러한 예들에서, 다음 중 하나는 IDV 가 이웃하는 블록에 대해 존재한다는 것을 나타낸다: 불법 참조 인덱스; 불법 참조 뷰 인덱스; 및 불법 참조 뷰 인덱스들의 차이. 예를 들어, 비디오 인코더 (20) 는 IDV 가 이웃하는 블록에 대해 존재한다는 것을 나타내기 위해 다음 중 하나를 시그널링할 수도 있다: 불법 참조 인덱스; 불법 참조 뷰 인덱스; 및 불법 참조 뷰 인덱스들의 차이.

[0154] 본 개시물의 예시적인 기법들은 별개로 또는 조합하여 사용될 수도 있다. 즉, 본 개시물의 기술적인 양태들 중 임의의 양태가 디스패리티 벡터 유도를 위한 완전한 솔루션으로 결합될 수 있다.

[0155] 도 6 은 본 개시물의 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 도

6 은 설명의 목적들을 위해 제공되며 본 개시물에서 넓게 예시되고 설명된 바와 같은 기법들의 한정으로 간주되지 않아야 한다. 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 상황에서 비디오 인코더 (20) 를 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0156] 도 6 의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 유닛 (100), 잔여 발생 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 프로세싱 유닛 (110), 재구성 유닛 (112), 필터 유닛 (114), 디코딩 화상 버퍼 (116), 및 엔트로피 인코딩 유닛 (118) 을 포함한다. 예측 프로세싱 유닛 (100) 은 인터-예측 프로세싱 유닛 (120) 및 인트라-예측 프로세싱 유닛 (126) 을 포함한다. 인터-예측 프로세싱 유닛 (120) 은 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 을 포함한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능적 구성요소들을 포함할 수도 있다.

[0157] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터의 화상의 슬라이스에서 각각의 CTU 를 인코딩할 수도 있다. CTUs 의 각각은 화상의 동일-사이즈로된 루마 코딩 트리 블록들 (CTBs) 및 대응하는 CTBs 와 연관될 수도 있다. CTU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 쿼드-트리 파티셔닝을 수행하여, CTU 의 CTBs 를 계속해서-더 작은 블록들로 분할할 수도 있다. 더 작은 블록들은 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 CTU 와 연관되는 CTB 를 4개의 동일-사이즈로된 서브-블록들로 파티셔닝하고, 서브-블록들 중 하나 이상을 4개의 동일-사이즈로된 서브-서브-블록들로, 그리고 기타 등등으로 파티셔닝할 수도 있다.

[0158] 비디오 인코더 (20) 는 CU들의 인코딩된 표현들 (즉, 코딩된 CU들) 을 발생하기 위해 CTU 의 CU들을 인코딩할 수도 있다. CU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU들 중에서 CU 와 연관되는 코딩 블록들을 파티셔닝할 수도 있다. 따라서, 각각의 PU 는 루마 예측 블록 및 대응하는 크로마 예측 블록들과 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 여러 사이즈들을 가지는 PU들을 지원할 수도 있다. 위에서 나타낸 바와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있으며, PU 의 사이즈는 루마 PU 의 예측 블록의 사이즈를 지칭할 수도 있다. 특정의 CU 의 사이즈가 2Nx2N 이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인트라 예측에 대해서는 2Nx2N 또는 NxN 의 PU 사이즈들을, 그리고 인터 예측에 대해서는 2Nx2N, 2NxN, Nx2N, NxN, 또는 등등의 대칭 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한 인터 예측에 대해서 2NxnU, 2NxnD, nLx2N, 및 nRx2N 의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

[0159] 인터-예측 프로세싱 유닛 (120) 은 CU 의 각각의 PU 에 대해 인터 예측을 수행함으로써, PU 에 대한 예측 데이터를 발생할 수도 있다. 예측 PU 에 대한 데이터는 PU 의 예측 블록들 및 PU 에 대한 모션 정보를 포함할 수도 있다. 인터-예측 프로세싱 유닛 (120) 은 PU 가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 내에 있는지에 따라서, CU 의 PU 에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들은 인트라 예측된다. 그러므로, PU 가 I 슬라이스에 있으면, 인터-예측 프로세싱 유닛 (120) 은 PU 에 관해 인터 예측을 수행하지 않는다.

[0160] PU 가 P 슬라이스 내에 있을 때, 모션 추정 유닛 (122) 은 참조 화상들의 리스트 (예컨대, "RefPicList0") 에서의 참조 화상들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있다. PU 에 대한 참조 영역은 참조 화상 내에서, PU 의 예측 블록들에 가장 가깝게 대응하는 샘플들을 포함하는 영역일 수도 있다. 모션 추정 유닛 (122) 은 PU 에 대한 참조 영역을 포함하는 참조 화상의 RefPicList0 에서의 위치를 나타내는 참조 인덱스를 발생할 수도 있다. 게다가, 모션 추정 유닛 (122) 은 PU 의 예측 블록과 참조 영역과 연관되는 참조 로케이션 사이의 공간 변위를 나타내는 모션 벡터를 발생할 수도 있다. 예를 들어, 모션 벡터는 현재의 화상에서의 좌표들로부터 참조 화상에서의 좌표들까지 오프셋을 제공하는 2차원 벡터일 수도 있다. 모션 추정 유닛 (122) 은 참조 인덱스 및 모션 벡터를 PU 의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터에 의해 표시되는 참조 로케이션에서의 실제 또는 내삽된 샘플들에 기초하여, PU 의 예측 블록들을 발생할 수도 있다.

[0161] PU 가 B 슬라이스 내에 있을 때, 모션 추정 유닛 (122) 은 PU 에 대해 단방향-예측 또는 양방향-예측을 수행할 수도 있다. PU 에 대한 단방향-예측을 수행하기 위해, 모션 추정 유닛 (122) 은 RefPicList0 또는 제 2 참조 화상 리스트 ("RefPicList1") 의 참조 화상들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있다. 모션 추정 유닛 (122) 은 PU 의 모션 정보로서, 참조 영역을 포함하는 참조 화상의 RefPicList0 또는 RefPicList1 에서의 위치를 나타내는 참조 인덱스, PU 의 예측 블록과 참조 영역과 연관되는 참조 로케이션 사이의 공간 변위를 나타내는 모션 벡터, 및 참조 화상이 RefPicList0 또는 RefPicList1 에 있는지 여부를 나타내는 하나 이상

의 예측 방향 표시자들을 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터에 의해 표시되는 참조 로케이션에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여, PU 의 예측 블록들을 발생할 수도 있다.

[0162] PU 에 대한 양방향 인터 예측을 수행하기 위해, 모션 추정 유닛 (122) 은 RefPicList0 에서의 참조 화상들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있으며, 또한 RefPicList1 에서의 참조 화상들을 PU 에 대한 또 다른 참조 영역에 대해 탐색할 수도 있다. 모션 추정 유닛 (122) 은 참조 영역들을 포함하는 참조 화상들의 RefPicList0 및 RefPicList1 에서의 위치들을 나타내는 참조 인덱스들을 발생할 수도 있다. 게다가, 모션 추정 유닛 (122) 은 참조 영역들과 연관되는 참조 로케이션들과 PU 의 예측 블록 사이의 공간 변위들을 나타내는 모션 벡터들을 발생할 수도 있다. PU 의 모션 정보는 PU 의 참조 인덱스들 및 모션 벡터들을 포함할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터들에 의해 표시되는 참조 로케이션들에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여, PU 의 예측 블록들을 발생할 수도 있다.

[0163] 인트라-예측 프로세싱 유닛 (126) 은 PU 에 대해 인트라 예측을 수행함으로써, PU 에 대한 예측 데이터를 발생할 수도 있다. 예측 PU 에 대한 데이터는 PU 에 대한 예측 블록들 및 여러 구문 엘리먼트들을 포함할 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU들에 대해 인트라 예측을 수행할 수도 있다.

[0164] PU 상에 인트라 예측을 수행하기 위해, 인트라-예측 프로세싱 유닛 (126) 은 다수의 인트라 예측 모드들을 이용하여 PU 에 대한 예측 블록들의 다수의 세트들을 발생할 수도 있다. 특정의 인트라 예측 모드를 이용하여 인트라 예측을 수행할 때, 인트라-예측 프로세싱 유닛 (126) 은 이웃하는 블록들로부터의 샘플들의 특정의 세트를 이용하여, PU 에 대한 예측 블록들을 발생할 수도 있다. 이웃하는 블록들은 PU들, CU들, 및 CTUs 에 대해 좌우, 상하 인코딩 순서를 가정하면, PU 의 예측 블록들의 상부, 상부 우측, 상부 좌측, 또는 좌측일 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 다수의 인트라 예측 모드들, 예컨대, 33개의 방향 인트라 예측 모드들을 이용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 개수는 PU 의 예측 블록들의 사이즈에 의존할 수도 있다.

[0165] 예측 프로세싱 유닛 (100) 은 PU들에 대해 인터-예측 프로세싱 유닛 (120) 에 의해 발생된 예측 데이터, 또는 PU들에 대해 인트라-예측 프로세싱 유닛 (126) 에 의해 발생된 예측 데이터 중으로부터, CU 의 PU들에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여, CU 의 예측 PU 에 대한 데이터들을 선택한다. 선택된 예측 데이터의 예측 블록들은 본원에서 그 선택된 예측 블록들로서 지칭될 수도 있다.

[0166] 잔여 발생 유닛 (102) 은 CU 의 루마, Cb 및 Cr 코딩 블록들 및 CU 의 PU들의 선택된 예측 루마, Cb 및 Cr 블록들에 기초하여, CU 의 루마, Cb 및 Cr 잔여 블록들을 발생할 수도 있다. 예를 들어, 잔여 발생 유닛 (102) 은 잔여 블록들에서의 각각의 샘플이 CU 의 코딩 블록에서의 샘플과 CU 의 PU 의 대응하는 선택된 예측 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 갖도록, CU 의 잔여 블록들을 발생할 수도 있다.

[0167] 변환 프로세싱 유닛 (104) 은 쿼드-트리 파티셔닝을 수행하여, CU 의 잔여 블록들을 CU 의 TU들과 연관되는 변환 블록들로 파티셔닝할 수도 있다. 따라서, TU 는 루마 변환 블록 및 2개의 대응하는 크로마 변환 블록들과 연관될 수도 있다. CU 의 TU들의 루마 및 크로마 변환 블록들의 사이즈들 및 위치들은 CU 의 PU들의 예측 블록들의 사이즈들 및 위치들에 기초하거나 또는 기초하지 않을 수도 있다.

[0168] 변환 프로세싱 유닛 (104) 은 하나 이상의 변환들을 TU 의 변환 블록들을 적용함으로써, CU 의 각각의 TU 에 대한 변환 계수 블록들을 발생할 수도 있다. 변환 프로세싱 유닛 (104) 은 여러 변환들을 TU 와 연관되는 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 이산 코사인 변환 (DCT), 방향성 변환, 또는 개념적으로-유사한 변환을 변환 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (104) 은 변환들을 변환 블록에 적용하지 않는다. 이러한 예들에서, 변환 블록은 변환 계수 블록으로서 취급될 수도 있다.

[0169] 양자화 유닛 (106) 은 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 그 변환 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안 m-비트 변환 계수로 절사될 수도 있으며, 여기서, n 은 m 보다 더 크다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서 양자화된 변환 계수들은 원래 정밀도들보다 낮은 정밀도를 가질 수도 있다.

[0170] 역양자화 유닛 (108) 및 역변환 프로세싱 유닛 (110) 은 역양자화 및 역변환들을 계수 블록에 각각 적용하여,

계수 블록으로부터 잔여 블록을 재구성할 수도 있다. 재구성 유닛 (112) 은 그 재구성된 잔여 블록을 예측 프로세싱 유닛 (100) 에 의해 발생되는 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 가산하여, TU 와 연관되는 재구성된 변환 블록을 생성할 수도 있다. 이 방법으로 CU 의 각각의 TU 에 대한 변환 블록들을 재구성함으로써, 비디오 인코더 (20) 는 CU 의 코딩 블록들을 재구성할 수도 있다.

[0171] 필터 유닛 (114) 은 CU 와 연관되는 코딩 블록들에서 블록킹 아티팩트들을 감소시키기 위해, 하나 이상의 디블록킹 동작들을 수행할 수도 있다. 디코딩 화상 버퍼 (116) 는 필터 유닛 (114) 이 재구성된 코딩 블록들에 관해 하나 이상의 디블록킹 동작들을 수행한 후 그 재구성된 코딩 블록들을 저장할 수도 있다. 인터-예측 프로세싱 유닛 (120) 은 그 구성된 코딩 블록들을 포함하는 참조 화상을 이용하여, 다른 화상들의 PU들에 관해 인터 측을 수행할 수도 있다. 게다가, 인트라-예측 프로세싱 유닛 (126) 은 디코딩 화상 버퍼 (116) 에서의 재구성된 코딩 블록들을 이용하여, CU 와 동일한 화상에서의 다른 PU들에 관해 인트라 예측을 수행할 수도 있다.

[0172] 엔트로피 인코딩 유닛 (118) 은 비디오 인코더 (20) 의 다른 기능적 구성요소들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 양자화 유닛 (106) 으로부터 계수 블록들을 수신할 수도 있으며 예측 프로세싱 유닛 (100) 으로부터 구문 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (118) 은 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여, 엔트로피-인코딩된 데이터를 발생할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 컨텍스트-적응 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, 변수-대-변수 (V2V) 길이 코딩 동작, 구문-기반 컨텍스트-적응 2진 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-Golomb 인코딩 동작, 또는 또다른 유형의 엔트로피 인코딩 동작을 데이터에 대해 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩 유닛 (118) 에 의해 발생된 엔트로피-인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다.

[0173] 본 개시물의 하나 이상의 기법들에 따르면, 부모 블록은 복수의 블록들로 파티셔닝되며, 디스패리티 벡터 유도 프로세스가 복수의 블록들에서 대표 블록에 대한 디스패리티 벡터를 유도하도록 수행된다. 비디오 인코더 (20) 는 그 유도된 디스패리티 벡터에 기초하여 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측 (예컨대, 인터-뷰 모션 예측 및/또는 인터-뷰 잔여 예측) 을 부분적으로 수행함으로써, 비디오 데이터의 코딩된 표현을 포함하는 비트스트림을 발생할 수도 있다. 더욱이, 본 개시물의 예시적인 일부 기법들에서, 비디오 인코더 (20) 는 현재의 블록에 대한 디스패리티 벡터를 결정하는 디스패리티 벡터 유도 프로세스를 수행할 수도 있다. 디스패리티 벡터 유도 프로세스를 수행하는 것의 일부로서, 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가질 때, 비디오 인코더 (20) 는 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 암시적인 디스패리티 벡터를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 암시적인 디스패리티 벡터들을 가지는지 여부를 결정하지 않는다.

[0174] 도 7 은 본 개시물의 기법들을 구현하도록 구성된 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 7 은 설명의 목적들을 위해 제공되며 본 개시물에 넓게 예시되고 설명된 것과 같은 기법들에 한정하는 것이 아니다. 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 컨텍스트에서의 비디오 디코더 (30) 를 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0175] 도 7 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 필터 유닛 (160), 및 디코딩 화상 버퍼 (162) 를 포함한다. 예측 프로세싱 유닛 (152) 은 모션 보상 유닛 (164) 및 인트라-예측 프로세싱 유닛 (166) 을 포함한다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능적 구성요소들을 포함할 수도 있다.

[0176] 코딩 화상 버퍼 (CPB) (151) 는 비트스트림의 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 을 수신하여 저장할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 CPB (151) 로부터 NAL 유닛들을 수신하여, NAL 유닛들을 파싱하여, 비트스트림으로부터 구문 엘리먼트들을 획득할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 NAL 유닛들에서의 엔트로피-인코딩된 구문 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 및 필터 유닛 (160) 은 비트스트림으로부터 추출된 구문 엘리먼트들에 기초하여, 디코딩된 비디오 데이터를 발생할 수도 있다.

[0177] 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 것의 일부로서, 엔트로피 디코딩 유닛 (150) 은 코딩된 슬라이스 NAL 유닛들로부터 구문 엘리먼트들을 추출하여 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관련된 구문 엘리먼트들을 포함할 수도 있다.

[0178] 비트스트림으로부터 구문 엘리먼트들을 디코딩하는 것에 더해서, 비디오 디코더 (30) 는 CU 에 관해 디코딩 동작을 수행할 수도 있다. CU 에 관해 디코딩 동작을 수행함으로써, 비디오 디코더 (30) 는 CU 의 코딩 블록들을 재구성할 수도 있다.

[0179] CU 에 관해 디코딩 동작을 수행하는 것의 일부로서, 역양자화 유닛 (154) 은 CU 의 TU들과 연관되는 계수 블록들을 역양자화할 수도 있다, 즉 양자화 해제할 수도 있다. 역양자화 유닛 (154) 은 TU 의 CU 와 연관되는 QP 값을 이용하여, 적용할 역양자화 유닛 (154) 에 대한 양자화의 정도 및 이와 유사하게, 역양자화의 정도를 결정할 수도 있다. 즉, 압축 비, 즉, 원래 시퀀스 및 압축된 시퀀스를 표현하는데 사용되는 비트수의 비는, 변환 계수들을 양자화할 때 사용되는 QP 의 값을 조정함으로써 제어될 수도 있다. 압축 비는 또한 채용되는 엔트로피 코딩의 방법에 의존할 수도 있다.

[0180] 역양자화 유닛 (154) 이 계수 블록을 역양자화한 후, 역변환 프로세싱 유닛 (156) 은 TU 와 연관되는 잔여 블록을 발생하기 위해, 하나 이상의 역변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (156) 은 역 DCT, 역 정수 변환, 역 Karhunen-Loeve 변환 (KLT), 역 회전 변환, 역 방향 변환, 또는 또 다른 역변환을 계수 블록에 적용할 수도 있다.

[0181] PU 가 인트라 예측을 이용하여 인코딩되면, 인트라-예측 프로세싱 유닛 (166) 은 인트라 예측을 수행하여, PU 에 대한 예측 블록들을 발생할 수도 있다. 인트라-예측 프로세싱 유닛 (166) 은 공간적으로-이웃하는 PU들의 예측 블록들에 기초하여, PU 에 대한 예측 루마, Cb, 및 Cr 블록들을 발생하기 위해, 인트라 예측 모드를 이용할 수도 있다. 인트라-예측 프로세싱 유닛 (166) 은 비트스트림으로부터 디코딩된 하나 이상의 구문 엘리먼트들에 기초하여, PU 에 대한 인트라 예측 모드를 결정할 수도 있다.

[0182] 예측 프로세싱 유닛 (152) 은 비트스트림으로부터 추출된 구문 엘리먼트들에 기초하여, 제 1 참조 화상 리스트 (RefPicList0) 및 제 2 참조 화상 리스트 (RefPicList1) 를 구성할 수도 있다. 더욱이, PU 가 인터 예측을 이용하여 인코딩되면, 엔트로피 디코딩 유닛 (150) 은 PU 에 대한 모션 정보를 획득할 수도 있다. 모션 보상 유닛 (164) 은 PU 의 모션 정보에 기초하여, PU 에 대한 하나 이상의 참조 영역들을 결정할 수도 있다. 모션 보상 유닛 (164) 은 PU 에 대한 하나 이상의 참조 블록들에서의 샘플들에 기초하여, PU 에 대한 예측 루마, Cb, 및 Cr 블록들을 발생할 수도 있다.

[0183] 재구성 유닛 (158) 은 CU 의 루마, Cb, 및 Cr 코딩 블록들을 재구성하기 위해, 적용가능한 경우, CU 의 TU들과 연관되는 루마, Cb, 및 Cr 변환 블록들 및 CU 의 PU들의 예측 루마, Cb, 및 Cr 블록들로부터의 잔여 값들, 즉, 인트라-예측 데이터 또는 인터-예측 데이터를 이용할 수도 있다. 예를 들어, 재구성 유닛 (158) 은 루마, Cb, 및 Cr 변환 블록들의 샘플들을 예측 루마, Cb, 및 Cr 블록들의 대응하는 샘플들에 가산하여, CU 의 루마, Cb, 및 Cr 코딩 블록들을 재구성할 수도 있다.

[0184] 필터 유닛 (160) 은 CU 의 루마, Cb, 및 Cr 코딩 블록들과 연관되는 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행할 수도 있다. 비디오 디코더 (30) 는 CU 의 루마, Cb, 및 Cr 코딩 블록들을 디코딩 화상 버퍼 (162) 에 저장할 수도 있다. 디코딩 화상 버퍼 (162) 는 후속 모션 보상, 인트라 예측, 및 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에의 프리젠테이션을 위해, 참조 화상들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 디코딩 화상 버퍼 (162) 에서의 루마, Cb, 및 Cr 블록들에 기초하여, 다른 CU들의 PU들에 관해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다. 이러한 방법으로, 비디오 디코더 (30) 는 비트스트림으로부터, 유의한 루마 계수 블록의 변환 계수 레벨들을 추출하고, 변환 계수 레벨들을 역양자화하고, 변환을 변환 계수 레벨들에 적용하여 변환 블록을 발생하고, 그 변환 블록에 적어도 부분적으로 기초하여, 코딩 블록을 발생하고, 그리고 디스플레이를 위해 코딩 블록을 출력할 수도 있다.

[0185] 본 개시물의 하나 이상의 기법들에 따르면, 부모 블록 (예컨대, CTU, CU, 등) 은 복수의 블록들을 파티셔닝될 수도 있으며, 디스패리티 벡터 유도 프로세스가 복수의 블록들에서 대표 블록에 대한 디스패리티 벡터를 유도하도록 수행된다. 비디오 디코더 (30) 는 그 유도된 디스패리티 벡터에 기초하여 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측을 부분적으로 수행함으로써, 복수의 블록들에서의 2개 이상의 블록들에 대해

샘플 블록들을 재구성할 수도 있다. 더욱이, 일부 예들에서, 비디오 디코더 (30) 는 현재의 블록에 대한 디스패리티 벡터를 결정하는 디스패리티 벡터 유도 프로세스를 수행할 수도 있다. 디스패리티 벡터 유도 프로세스를 수행하는 것의 일부로서, 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 IDV 를 가질 때, 비디오 디코더 (30) 는 디스패리티 모션 벡터 또는 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 IDV 를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 IDVs 를 가지는 여부를 결정하지 않는다.

[0186] 위에서 설명된 바와 같이, 비디오 코더는 CU-기반의 NBDV 유도 프로세스를 수행하여, CU 의 각각의 PU 에 대한 디스패리티 벡터들을 유도할 수도 있다. 비디오 코더가 CU 의 각각의 PU 에 대한 디스패리티 벡터들을 유도하기 위해 CU-기반의 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 NBDV 유도 프로세스를 CU 에 대해 한번 수행하며, 비디오 코더는 그 유도된 디스패리티 벡터를 CU 의 모든 PU들에 할당한다. 다시 말해서, CU-기반의 NBDV 유도 프로세스는 모든 PU들 및 CU 자신이 NBDV 유도 프로세스에 기초하여, 예컨대, 3D-HEVC 에서 유도된 바와 같이, 한번 유도된 동일한 디스패리티 벡터를 공유하는 방법으로 이루어질 수 있다. NBDV 유도 프로세스는 대표 블록 (예컨대, PU 또는 CU) 에 적용가능하다. 예를 들어, CU 는 복수의 PU들로 파티셔닝될 수도 있으며, 비디오 코더는 디스패리티 벡터 유도 프로세스를 수행하여, 복수의 PU들에서 (예컨대, CU 에서) 대표 PU 에 대한 디스패리티 벡터를 유도할 수도 있다.

[0187] 더욱이, CU-기반의 NBDV 유도 프로세스는 간단한 (straightforward) 확장판으로서의 LCU-기반의 NBDV 유도 프로세스, 또는 임의의 블록 사이즈에 대한 NBDV 유도 프로세스로 확장될 수 있다. 예를 들어, LCU 는 복수의 CU들로 파티셔닝될 수도 있으며, 비디오 코더는 디스패리티 벡터 유도 프로세스를 수행하여, 복수의 CU들에서의 대표 CU 에 대해 디스패리티 벡터를 유도할 수도 있다.

[0188] 위에서 설명된 바와 같이, 비디오 코더는 NBDV 유도 프로세스를 수행할 때 5개의 공간 이웃하는 블록들을 체킹할 수도 있다. 5개의 공간 이웃하는 블록들은 전체 CU 에 대해 오직 하나의 대표 PU 를 항상 고려함으로써, NBDV 에서 처럼 로케이트될 수도 있다.

[0189] 본 개시물에서, 복수의 PU들의 대표 PU 는 NBDV-프로세싱 유닛 (NPU) 으로 불려질 수도 있다. 다른 예들에서, 비디오 코더는 NPU 를 여러 방법들로 설정할 수도 있다. 예를 들어, NPU 는 현재의 CU 일 수도 있다. 또 다른 예에서, NPU 는 현재의 CU 에서의 제 1 PU 일 수도 있다. 이 예에서, 제 1 PU 는 CU 의 좌상단 픽셀을 커버하는 PU 인 것으로 정의된다.

[0190] 또 다른 예에서, 비디오 코더는 대표 PU 인 CU 의 우하단 픽셀을 커버하는 PU 를, 대표 블록으로서, 선택할 수도 있다. 다시 말해서, 비디오 코더는 NPU 로서 CU 의 우하단 픽셀을 커버하는 PU 를 선택할 수도 있다. 또 다른 예에서, 비디오 코더는 대표 PU 인 CU 의 우상단 픽셀을 커버하는 PU 를, 대표 블록으로서, 선택할 수도 있다. 다시 말해서, 비디오 코더는 NPU 로서 CU 의 우상단 픽셀을 커버하는 PU 를, 대표 블록으로서, 선택할 수도 있다. 또한 또 다른 예에서, 비디오 코더는 대표 PU 인 CU 의 좌하단 픽셀을 커버하는 PU 를 선택할 수도 있다. 다시 말해서, 비디오 코더는 NPU 로서 CU 의 좌하단 픽셀을 커버하는 PU 를 선택할 수도 있다. 본 개시물에서, 설명의 용이성을 위해, CU 의 우하단 픽셀, 우상단 픽셀, 또는 좌하단 픽셀은 각각 CU 의 코딩 블록의 우하단 픽셀, 우상단 픽셀, 또는 좌하단 픽셀일 수도 있다.

[0191] 도 8 은 CU 와 동일한 2Nx2N PU 에서 예시적인 공간 및 시간 이웃들을 예시하는 개념도이다. 도 8 의 예에서, 비디오 코더는 비디오 코더가 디스패리티 모션 벡터들 및 IDVs 에 대한 로케이션들 (200, 202, 204, 206, 및 208) 을 커버하는 블록들을 체킹하는 NBDV 유도 프로세스를 수행할 수도 있다. 더욱이, 도 8 의 예에서, 비디오 코더가 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 후보 참조 화상들 내에 있고 로케이션들 (210 및 212) 과 동일 장소에 배치된 블록들을 체킹할 수도 있다.

[0192] 도 9 는 CU 의 제 1 PU 가 CU 에 대한 대표 PU 인, NBDV 유도 프로세스에 사용되는 예시적인 공간 및 시간 이웃하는 블록들을 예시하는 개념도이다. 도 9 의 예에서, 대표 블록은 CU 의 좌측에서 Nx2N PU 이다. 다시 말해서, 도 9 는 PU 가 Nx2N 형태를 갖는 예이다. 현재의 CU 의 주어진 파티션 정보 (예컨대, NX2N) 에 의해, 블록 위치들은 제 1 PU (예컨대, 좌측 PU) 의 위치들로 결정될 수 있다. 도 9 는 블록 위치들 (즉, 이웃하는 블록들에 의해 커버되는 로케이션들) 이 제 1 PU 의 형태에 기초하여 결정될 수 있는 방법의 일 예이다.

[0193] 특히, 도 9 의 예에서, 비디오 코더는 비디오 코더가 디스패리티 모션 벡터들 및 IDVs 에 대한 로케이션들 (250, 252, 254, 256, 및 258) 을 커버하는 블록들을 체킹하는 NBDV 유도 프로세스를 수행할 수도 있다.

더욱이, 도 9 의 예에서, 비디오 코더가 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 후보 참조 화상들 내에 있고 로케이션들 (260 및 262) 과 동일 장소에 배치된 블록들을 체킹할 수도 있다. 도 8 및 도 9 의 예들로부터 볼 수 있는 바와 같이, NBDV 유도 프로세스에서 사용되는 이웃하는 블록들에 의해 커버되는 로케이션들은, CU 가 PU들로 파티셔닝되는 방법 그리고 PU들 중 어느 것이 대표 블록인지에 따라서, CU 의 PU들에 대해 상이할 수도 있다. 따라서, 비디오 코더는 제 1 PU 의 형태에 기초하여 복수의 블록 위치들을 결정할 수도 있으며, 여기서, 복수의 블록 위치들에서 각각의 각각의 블록 위치는 복수의 이웃하는 블록들에서 각각의 이웃하는 블록에 의해 커버된다. 더욱이, 비디오 코더는 공간 디스패리티 모션 벡터, 암시적인 디스패리티 벡터, 또는 시간 디스패리티 모션 벡터에 대해 이웃하는 블록들 중 하나 이상을 체킹할 수도 있다.

[0194] 도 10 은 본 개시물의 하나 이상의 기법들에 따른, 비디오 코더의 예시적인 동작을 예시하는 플로우차트이다. 도 10 의 예에 예시된 바와 같이, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 디스패리티 벡터 유도 프로세스를 수행하여 복수의 블록들에서 대표 블록에 대한 디스패리티 벡터를 유도할 수도 있다 (300). 도 10 의 예에서, 비디오 데이터는 복수의 블록들로 파티셔닝되는 부모 블록을 포함할 수도 있다. 예를 들어, 부모 블록은 CU 일 수도 있으며, 복수의 블록들에서의 각각의 블록은 CU 의 PU 일 수도 있다. 이 예에서, 대표 블록은 CU 의 제 1 PU 일 수도 있다. 또 다른 예에서, 부모 블록은 LCU 일 수도 있으며, 복수의 블록들에서의 각각의 블록은 CU 일 수도 있으며, 대표 블록은 LCU 의 CU들 중 하나일 수도 있다.

[0195] 더욱이, 비디오 코더는 그 유도된 디스패리티 벡터에 기초하여, 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측을 수행할 수도 있다 (302). 비디오 코더가 비디오 디코더인 일부 예들에서, 비디오 디코더는 그 유도된 디스패리티 벡터에 기초하여, 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측을 부분적으로 수행함으로써, 복수의 블록들에서의 2개 이상의 블록들에 대해 샘플 블록들을 재구성할 수도 있다. 이와 유사하게, 비디오 코더가 비디오 인코더인 일부 예들에서, 비디오 인코더는 그 유도된 디스패리티 벡터에 기초하여 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측을 부분적으로 수행함으로써, 비디오 데이터의 코딩된 표현을 포함하는 비트스트림을 발생할 수도 있다. 비디오 코더가 2개 이상의 블록들에 대해 인터-뷰 예측을 수행할 때, 비디오 코더는 그 유도된 디스패리티 벡터에 기초하여, 2개 이상의 블록들에 대해 인터-뷰 모션 예측 및 인터-뷰 잔여 예측 중 하나 이상을 수행할 수도 있다.

[0196] 도 11a 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 모션 예측을 수행하는 비디오 인코더의 예시적인 동작 (350) 을 예시하는 플로우차트이다. 도 11a 의 예에 예시된 바와 같이, 비디오 인코더 (예컨대, 비디오 인코더 (20)) 는 현재의 CU 의 파티셔닝 모드를 결정할 수도 있다 (352). 본 개시물의 하나 이상의 기법들에 따르면, 비디오 인코더는 디스패리티 벡터 유도 프로세스를 수행하여, 현재의 CU 의 대표 PU 에 대한 디스패리티 벡터를 결정할 수도 있다 (354). 대표 블록이 CU 인 일부 예들에서, 비디오 인코더는 복수의 이웃하는 블록들을 적어도 부분적으로 결정함으로써 디스패리티 벡터 유도 프로세스를 수행할 수도 있으며, 여기서, 복수의 이웃하는 블록들은 CU 에 인접한 공간 이웃하는 블록들, CU 의 중심과 동일 장소에 배치된 로케이션을 커버하는 시간 이웃하는 블록, 및 CU 의 직하부 및 우측 로케이션을 커버하는 시간 이웃하는 블록을 포함한다.

더욱이, 이러한 예들에서, 비디오 인코더는 공간 디스패리티 모션 벡터, 암시적인 디스패리티 벡터, 또는 시간 디스패리티 모션 벡터에 대해 이웃하는 블록들 중 하나 이상을 체킹할 수도 있다.

[0197] 다음으로, 비디오 인코더는 대표 PU 에 대한 디스패리티 벡터에 기초하여, 현재의 PU 에 대한 인터-뷰 예측 모션 벡터 후보 (IPMVC) 를 결정할 수도 있다 (356). IPMVC 는 현재의 화상과는 상이한 뷰와 연관되는 참조 화상에서의 로케이션을 나타내는 모션 벡터 후보이다. 현재의 PU 는 현재의 CU 의 PU들 중 하나이다. 비디오 인코더는 그후 IPMVC 를 현재의 PU 에 대한 후보 리스트 (예컨대, 병합 후보 리스트 또는 AMVP 후보 리스트) 에 포함시킬 수도 있다 (358). 그 후에, 비디오 인코더는 후보 리스트로부터 후보를 선택할 수도 있다 (360). 일부의 경우, 선택된 후보는 IPMVC 이다. 비디오 인코더는 그후, 비트스트림에, 선택된 후보의 인덱스를 나타내는 데이터를 포함시킬 수도 있다 (362). 예를 들어, 후보 리스트가 병합 후보 리스트이면, 비디오 인코더는 비트스트림에, merge_idx 구문 엘리먼트를 표시하는 데이터를 포함시킬 수도 있다. 또 다른 예에서, 후보 리스트는 AMVP 후보 리스트이고 비디오 인코더는 비트스트림에, mvp_l0_flag 구문 엘리먼트 또는 mvp_l1_flag 구문 엘리먼트를 표시하는 데이터를 포함시킬 수도 있다. 이 예에서, 비디오 인코더는 또한 현재의 PU 에 대한 참조 인덱스 (예컨대, ref_idxl0 구문 엘리먼트 또는 ref_idxl1 구문 엘리먼트) 를 표시

하는 데이터 및 현재의 PU 에 대한 모션 벡터 차이 (예컨대, mvd_코딩 구문 구조) 를 표시하는 데이터를 포함시킬 수도 있다.

[0198] 비디오 인코더는 그후 현재의 CU 에 임의의 나머지 PU들이 존재하는지 여부를 결정할 수도 있다 (364). 현재의 CU 에서 하나 이상의 나머지 PU들이 존재하면 (364 의 "예"), 비디오 인코더는 현재의 PU 처럼 나머지 PU들 중 하나에 대해 액션들 356-364 을 반복할 수도 있다. 한편, 현재의 CU 에서 어떤 나머지 PU들도 존재하지 않으면 (364 의 "아니오"), 동작 350 이 종료되고 비디오 인코더는 다른 인코딩 액션들을 수행할 수도 있다.

[0199] 도 11b 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 모션 예측을 수행하기 위한 비디오 디코더의 예시적인 동작 (380) 을 예시하는 플로우차트이다. 도 11b 의 예에 예시된 바와 같이, 비디오 디코더 (예컨대, 비디오 디코더 (30)) 는 현재의 CU 의 파티셔닝 모드를 결정할 수도 있다 (382). 일부 예들에서, 비디오 디코더는 현재의 CU 의 PART_mode 구문 엘리먼트에 기초하여 현재의 CU 의 파티셔닝 모드를 결정할 수도 있다.

[0200] 본 개시물의 하나 이상의 기법들에 따르면, 비디오 디코더는 디스패리티 벡터 유도 프로세스를 수행하여 현재의 CU 의 대표 PU 에 대한 디스패리티 벡터를 결정할 수도 있다 (384). 예를 들어, 비디오 디코더는 디스패리티 벡터 유도 프로세스를 수행하여, 현재의 CU 의 제 1 PU 에 대한 디스패리티 벡터를 결정할 수도 있다. 대표 블록이 CU 인 일부 예들에서, 비디오 디코더는 복수의 이웃하는 블록들을 적어도 부분적으로 결정함으로써 디스패리티 벡터 유도 프로세스를 수행할 수도 있으며, 여기서, 복수의 이웃하는 블록들은 CU 에 인접한 공간 이웃하는 블록들, CU 의 중심과 동일 장소에 배치된 로케이션을 커버하는 시간 이웃하는 블록, 및 CU 의 직하부 및 우측 로케이션을 커버하는 시간 이웃하는 블록을 포함한다. 더욱이, 이러한 예들에서, 비디오 디코더는 공간 디스패리티 모션 벡터, IDV, 또는 시간 디스패리티 모션 벡터에 대해 이웃하는 블록들 중 하나 이상을 체킹할 수도 있다.

[0201] 다음으로, 비디오 디코더는 대표 PU 에 대한 디스패리티 벡터에 기초하여, 현재의 PU 에 대한 IPMVC 를 결정할 수도 있다 (386). 현재의 PU 는 현재의 CU 의 PU들 중 하나이다. 비디오 디코더는 그후 현재의 PU 에 대한 후보 리스트 (예컨대, 병합 후보 리스트 또는 AMVP 후보 리스트) 에 IPMVC 를 포함시킬 수도 있다 (388). 그 후에, 비디오 디코더는 비트스트림에서의 데이터에 기초하여, 후보 리스트에서의 선택된 후보를 결정할 수도 있다 (390). 예를 들어, 후보 리스트가 병합 후보 리스트이면, 비디오 디코더는 비트스트림으로부터 획득된 merge_idx 구문 엘리먼트에 기초하여 그 선택된 후보를 결정할 수도 있다. 또 다른 예에서, 후보 리스트가 AMVP 후보 리스트이면, 비디오 디코더는 비트스트림으로부터 획득된 mvp_l0_flag 구문 엘리먼트 또는 mvp_l1_flag 구문 엘리먼트에 기초하여, 그 선택된 후보를 결정할 수도 있다. 일부의 경우, 선택된 후보는 IPMVC 이다. 비디오 디코더는 그후 그 선택된 후보에 기초하여, 현재의 PU 에 대한 모션 벡터를 결정할 수도 있다 (392).

[0202] 비디오 디코더는 그후 현재의 CU 에 임의의 나머지 PU들이 존재하는지 여부를 결정할 수도 있다 (394). 현재의 CU 에서 하나 이상의 나머지 PU들이 존재하면 (394 의 "예"), 비디오 디코더는 현재의 PU 처럼 나머지 PU들 중 하나에 대해 액션들 386-394 을 반복할 수도 있다. 한편, 현재의 CU 에서 어떤 나머지 PU들 도 존재하지 않으면 (394 의 "아니오"), 동작 380 이 종료되고 비디오 디코더는 다른 디코딩 액션들을 수행할 수도 있다.

[0203] 도 12a 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 잔여 예측을 수행하는 비디오 인코더의 예시적인 동작 (400) 을 예시하는 플로우차트이다. 도 12a 의 예에 예시된 바와 같이, 비디오 인코더 (예컨대, 비디오 인코더 (20)) 는 부모 블록의 대표 블록에 대한 디스패리티 벡터를 결정할 수도 있다 (402). 일부 예들에서, 부모 블록은 부모 블록에 대한 대표 블록을 포함한, 복수의 블록들로 파티셔닝된다. 예를 들어, 대표 블록은 CU 일 수도 있으며 부모 블록은 CTU 일 수도 있다. 다른 경우, 대표 블록은 PU 일 수도 있으며 부모 블록은 CU 일 수도 있다. 다른 예들에서, 부모 블록은 서브-블록들로 파티셔닝되지 않으며, 부모 블록에 대한 대표 블록은 부모 블록 자신이다. 예를 들어, 대표 블록 및 부모 블록은 동일한 PU, CU, CTU, 등일 수도 있다.

[0204] 더욱이, 도 12a 의 예에서, 비디오 인코더는 대표 블록에 대한 디스패리티 벡터에 기초하여, 현재의 블록에 대한 디스패리티 참조 블록을 결정할 수도 있다 (404). 위에서 나타낸 바와 같이, 부모 블록은 하나 이상의 서브-블록들로 파티셔닝될 수도 있다. 현재의 블록은 서브-블록들 중 하나일 수도 있다. 더욱이, 현재의 블록에 대한 디스패리티 참조 블록은 디스패리티 참조 화상에서의 샘플들에 기초한 샘플들의 블록을 포함할 수도 있다. 예를 들어, 디스패리티 참조 블록은 디스패리티 참조 화상의 샘플들을 포함할 수도 있다. 디스패리티 참조 화상은 현재의 블록과 동일한 시간 인스턴스와 연관되지만, 현재의 블록과는 상이한 뷰와 연관

된다.

[0205] 비디오 인코더는 디스패리티 참조 블록에 대한 잔여 데이터를 결정할 수도 있다 (406). 디스패리티 참조 블록에 대한 잔여 데이터는 디스패리티 참조 블록의 샘플들과, 디스패리티 참조 블록에 대한 하나 이상의 예측 블록들의 대응하는 샘플들 사이의 차이들을 나타내는 샘플들을 포함할 수도 있다. 비디오 인코더는 그후 현재의 블록에 대한 잔여 데이터를 결정할 수도 있다 (408). 현재의 블록에 대한 잔여 데이터는 현재의 블록에 대한 원래 샘플 블록에서의 대응하는 샘플들 사이의 차이들을 나타내는 샘플들, 현재의 블록에 대한 디스패리티 참조 블록에 대한 잔여 데이터, 및 하나 이상의 예측 블록들을 포함할 수도 있다. 그 후에, 비디오 인코더는 부모 블록에 임의의 나머지 서브-블록들이 존재하는지 여부를 결정할 수도 있다 (410). 부모 블록에 하나 이상의 나머지 서브-블록들이 존재하면 (410 의 "예"), 비디오 인코더는 현재의 블록 처럼 나머지 서브-블록들 중 하나에 대해 액션들 404-410 을 반복할 수도 있다. 이러한 방법으로, 비디오 인코더는 대표 블록에 대한 디스패리티 벡터를 이용하여, 부모 블록의 서브-블록들의 각각에 대한 디스패리티 참조 블록들을 결정할 수도 있다.

[0206] 한편, 부모 블록에서 어떤 나머지 서브-블록들도 없으면 (410 의 "아니오"), 비디오 인코더는 비트스트림에, 부모 블록에 대한 잔여 데이터를 나타내는 데이터를 포함시킬 수도 있다 (412). 예를 들어, 부모 블록이 CTU 이고 서브-블록들의 각각이 CU 이면, 부모 블록에 대한 잔여 데이터를 나타내는 데이터는 CTU 의 CU들의 각각의 잔여 데이터의 변환된 및 양자화된 버전들을 나타내는 데이터를 포함할 수도 있다. 또 다른 예에서, 부모 블록이 CU 이고 서브-블록들의 각각이 PU 이면, CU 에 대한 잔여 데이터를 나타내는 데이터는 CU 의 PU들의 각각의 잔여 데이터의 변환된 및 양자화된 버전들을 나타내는 데이터를 포함할 수도 있다.

[0207] 도 12b 는 본 개시물의 하나 이상의 기법들에 따른, 인터-뷰 잔여 예측을 수행하는 비디오 디코더의 예시적인 동작 (450) 을 예시하는 플로우차트이다. 도 12b 의 예에 예시된 바와 같이, 비디오 디코더 (예컨대, 비디오 디코더 (30)) 는 부모 블록의 대표 블록에 대한 디스패리티 벡터를 결정할 수도 있다 (452). 일부 예들에서, 부모 블록은 부모 블록에 대한 대표 블록을 포함한, 복수의 블록들로 파티셔닝된다. 예를 들어, 대표 블록은 CU 일 수도 있으며 부모 블록은 CTU 일 수도 있다. 다른 경우, 대표 블록은 PU 일 수도 있으며 부모 블록은 CU 일 수도 있다. 다른 예들에서, 부모 블록은 서브-블록들로 파티셔닝되지 않으며, 부모 블록에 대한 대표 블록은 부모 블록 자신이다. 예를 들어, 대표 블록 및 부모 블록은 동일한 PU, CU, CTU, 등일 수도 있다.

[0208] 더욱이, 도 12b 의 예에서, 비디오 디코더는 대표 블록에 대한 디스패리티 벡터에 기초하여, 현재의 블록에 대한 디스패리티 참조 블록을 결정할 수도 있다 (454). 위에서 나타낸 바와 같이, 부모 블록은 하나 이상의 서브-블록들로 파티셔닝될 수도 있다. 현재의 블록은 서브-블록들 중 하나일 수도 있다. 더욱이, 현재의 블록에 대한 디스패리티 참조 블록은 디스패리티 참조 화상에서의 샘플들에 기초한 샘플들의 블록을 포함할 수도 있다. 예를 들어, 디스패리티 참조 블록은 디스패리티 참조 화상의 샘플들을 포함할 수도 있다. 디스패리티 참조 화상은 현재의 블록과 동일한 시간 인스턴스와 연관되지만, 현재의 블록과는 상이한 뷰와 연관된다.

[0209] 비디오 디코더는 디스패리티 참조 블록에 대한 잔여 데이터를 결정할 수도 있다 (456). 디스패리티 참조 블록에 대한 잔여 데이터는 디스패리티 참조 블록의 샘플들과, 디스패리티 참조 블록에 대한 하나 이상의 예측 블록들의 대응하는 샘플들 사이의 차이들을 나타내는 샘플들을 포함할 수도 있다. 그 후에, 비디오 디코더는 부모 블록에 임의의 나머지 서브-블록들이 존재하는지 여부를 결정할 수도 있다 (458). 부모 블록에 하나 이상의 나머지 서브-블록들이 존재하면 (458 의 "예"), 비디오 디코더는 현재의 블록 처럼 나머지 서브-블록들 중 하나에 대해 액션들 454-458 을 반복할 수도 있다. 이러한 방법으로, 비디오 디코더는 대표 블록에 대한 디스패리티 벡터를 이용하여, 부모 블록의 서브-블록들의 각각에 대한 디스패리티 참조 블록들을 결정할 수도 있다.

[0210] 한편, 부모 블록에서 어떤 나머지 서브-블록들도 없으면 (458 의 "아니오"), 비디오 디코더는 부모 블록에 대한 샘플 블록 (예컨대, 코딩 트리 블록, 코딩 블록, 등) 을 결정할 수도 있다 (460). 부모 블록에 대한 샘플 블록은 부모 블록의 하나 이상의 서브-블록들에 대한 디스패리티 참조 블록들에 대한 잔여 데이터에서의 대응하는 샘플들의 합계인 샘플들, 하나 이상의 예측 블록들, 및 비트스트림으로 시그널링된 잔여 데이터를 포함할 수도 있다.

[0211] 위에서 설명된 바와 같이, 본 개시물의 기법들에 따르면, 비디오 코더는 3D-HEVC 테스트 모델 1 에서 설명되는 NBDV 유도 프로세스보다 더 적은 공간 이웃하는 블록들을 이용하는 수정된 NBDV 유도 프로세스를 수행할 수도

있다. 일 예에서, 비디오 코더가 수정된 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 단지 좌측 공간 이웃하는 블록 및 상부 공간 이웃하는 블록 (즉, 도 2 에서 A_1 및 B_1 로서 표시되는 블록들) 을 고려할 수도 있다. 도 13 은 2개의 PU들에 대한 예시적인 좌측 공간 이웃하는 블록들 및 상부 공간 이웃하는 블록들을 예시하는 개념도이다. 특히, 도 13 은 Nx2N PU 및 2NxN PU 에 대한 좌측 및 상부 공간 이웃하는 블록들을 나타낸다. 도 13 의 예에 나타낸 바와 같이, 좌측 공간 이웃하는 블록은 현재의 블록 (예컨대, Nx2N PU 또는 2NxN PU) 의 좌측 에지에 인접하며, 상부 공간 이웃하는 블록은 현재의 블록 (예컨대, Nx2N PU 또는 2NxN PU) 의 상부 에지에 인접한다.

[0212] 그러나, 좌측 및 상부 4x4 공간 이웃하는 블록들을 항상 고려하는 대신, 체킹될 블록들의 개수가 더 작아지도록, 여러 대안들이 있을 수 있다. 예를 들어, 하나의 솔루션에서, 비디오 코더는 3개의 공간 이웃하는 블록들을 체킹한다. 예를 들어, 도 14 에 나타낸 바와 같이, 비디오 코더는 단지 상부, 좌측, 및 좌상부 공간 이웃하는 블록들을 체킹한다. 즉, 도 14 는 NBDV 유도 프로세스에 사용되는 좌측, 상부, 및 좌상부 공간 이웃하는 블록들을 예시하는 개념도이다. 이 예에서, 비디오 코더는 상부, 좌측, 및 좌상부 공간 이웃하는 블록들을 그 순서대로 체킹할 수도 있다. 다른 예들에서, 비디오 코더는 상부, 좌측, 및 좌상부 공간 이웃하는 블록들을 임의의 다른 순서로 체킹할 수도 있다.

[0213] 다른 예들에서, 비디오 코더는 NBDV 유도 프로세스에서 공간 이웃하는 블록들 중 임의의 블록을 체킹할 수도 있다. 다시 말해서, 임의의 다른 공간 이웃하는 블록들은 도 2 에 나타낸 5개의 공간 이웃하는 블록들 A_0, A_1, B_0, B_1, 및 B_2 사이에 데시메이션될 수도 있다. 일부 예들에서, 데시메이션 이후 체킹될 블록들의 총 개수는 제로 이상 및 5개 미만일 수 있다. 더욱이, 일부 예들에서, 비디오 코더는 디스패리티 모션 벡터들에 대해 5개의 공간 이웃하는 블록들의 서브세트를 체킹할 수도 있지만, IDVs 에 대해서는 모든 5개의 공간 이웃하는 블록들을 체킹할 수도 있다. 다시 말해서, 이 예에서, 감소는 IDVs 가 탐색되어야 하는 블록들에 대해 적용될 수도 있거나 또는 적용되지 않을 수도 있다. 더욱이, 일부 예들에서, 공간 이웃하는 블록들의 체킹 순서는 재순서정렬될 수 있다.

[0214] 공간 이웃하는 블록들의 감소는 본 개시물에서 다른 어딘가에서 설명되는 CU-기반의 NBDV 유도 프로세스들과 결합될 수 있다. 예를 들어, 도 13 의 예에서, NPU 는 좌상단 픽셀을 커버하는 PU 이고, NPU 의 공간 이웃하는 블록들은 단지 NPU 의 상부 공간 이웃하는 블록 및 좌측 공간 이웃하는 블록이다. 또 다른 예에서, NPU 는 CU 이고, 공간 이웃하는 블록들은 도 14 에 나타낸 바와 같이, CU 의 상부, 좌측 및 좌상부 공간 이웃하는 블록들이다.

[0215] 도 15 는 본 개시물의 하나 이상의 기법들에 따른, 비디오 코더의 예시적인 동작 (600) 을 예시하는 예시하는 플로우차트이다. 도 15 의 예에서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 현재의 블록에 대한 디스패리티 벡터를 결정하는 디스패리티 벡터 유도 프로세스를 수행한다 (602). 본 개시물의 하나 이상의 기법들에 따르면, 디스패리티 벡터 유도 프로세스를 수행하는 것의 일부로서, 비디오 코더는 제 1 또는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터 또는 IDV 를 가질 때, 디스패리티 모션 벡터 또는 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다. 도 15 의 예에서, 디스패리티 벡터 유도 프로세스는 설령 제 1 또는 제 2 공간 이웃하는 블록들 중 어느 것도 디스패리티 모션 벡터 또는 IDV 를 가지지 않더라도, 제 1 및 제 2 공간 이웃하는 블록들에 더해서 임의의 공간 이웃하는 블록들이 디스패리티 모션 벡터들 또는 IDVs 를 가지는 여부를 결정하지 않는다. 일부 예들에서, 제 1 공간 이웃하는 블록은 현재의 블록의 상부 에지에 인접하며, 제 2 공간 이웃하는 블록은 현재의 블록의 좌측 에지에 인접한다. 일부 예들에서, 비디오 코더는 디스패리티 모션 벡터 (또는, IDV) 에 대해 제 1 공간 이웃하는 블록을 먼저 체킹할 수도 있으며 그후 디스패리티 모션 벡터 (또는, IDV) 에 대해 제 2 공간 이웃하는 블록을 체킹할 수도 있다. 다른 예들에서, 비디오 코더는 디스패리티 모션 벡터 (또는, IDV) 에 대해 제 2 공간 이웃하는 블록을 먼저 체킹할 수도 있으며, 그후 디스패리티 모션 벡터 (또는, IDV) 에 대해 제 1 공간 이웃하는 블록을 체킹할 수도 있다.

[0216] 더욱이, 도 15 의 예에서, 비디오 코더는 그 유도된 디스패리티 벡터에 기초하여, 그리고 대표 블록 이외의 복수의 블록들에서의 임의의 블록에 대한 디스패리티 벡터들을 따로 유도함이 없이, 복수의 블록들에서의 2개 이상의 블록들에 대해 인터-뷰 예측을 수행한다 (604). 비디오 코더가 비디오 디코더를 포함하는 일부 예들에서, 비디오 디코더는 그 유도된 디스패리티 벡터에 기초하여, 현재의 블록에 대한 인터-뷰 예측을 부분적으로 수행함으로써, 현재의 블록에 대한 샘플 블록을 재구성한다. 비디오 코더가 비디오 인코더를 포함하는 일부 예들에서, 비디오 인코더는 현재의 블록에 대한 유도된 디스패리티 벡터에 부분적으로 기초하여, 현재의 블록의

인코딩된 표현을 발생한다.

[0217] 도 16 은 본 개시물의 일 예에 따른, 현재의 블록에 대한 디스패리티 벡터를 결정하는 비디오 코더의 예시적인 동작 (620) 을 예시하는 플로우차트이다. 도 16 의 예에서, 비디오 코더는 현재의 블록에 대한 디스패리티 벡터를 결정할 때 최고 2개의 공간적으로 이웃하는 블록들을 이용한다. 구체적으로 설명하면, 도 16 의 예에서, 비디오 코더는 제 1 공간 이웃하는 블록이 디스패리티 모션 벡터를 가지는지 여부를 결정할 수도 있다 (622). 일부 예들에서, 제 1 공간 이웃하는 블록은 현재의 블록의 좌측 에지에 인접한다. 다른 예들에서, 제 1 공간 이웃하는 블록은 현재의 블록의 상부 에지에 인접한다. 제 1 공간 이웃하는 블록이 디스패리티 모션 벡터를 갖는다고 결정하는 것에 응답하여 (622 의 "예"), 비디오 코더는 제 1 공간 이웃하는 블록의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다 (624).

[0218] 한편, 제 1 공간 이웃하는 블록이 디스패리티 모션 벡터를 갖지 않는다고 결정하는 것에 응답하여 (622 의 "아니오"), 비디오 코더는 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터를 가지는지 여부를 결정할 수도 있다 (626). 일부 예들에서, 제 2 공간 이웃하는 블록은 현재의 블록의 좌측 에지에 인접한다. 다른 예들에서, 제 2 공간 이웃하는 블록은 현재의 블록의 상부 에지에 인접한다. 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터를 갖는다고 결정하는 것에 응답하여 (626 의 "예"), 비디오 코더는 제 2 공간 이웃하는 블록의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환한다 (628). 그렇지 않으면, 제 2 공간 이웃하는 블록이 디스패리티 모션 벡터를 가지지 않는다고 결정하는 것에 응답하여 (626 의 "아니오"), 비디오 코더는 현재의 블록에 대한 디스패리티 벡터가 이용불가능하다고 표시할 수도 있다 (630).

[0219] 위에서 설명된 바와 같이, 양방향-예측 인터 예측에서, 최고 2개의 참조 화상 리스트들 (즉, RefPicList0 및 RefPicList1) 이 존재할 수 있다. 더욱이, 양방향-예측 블록 (예컨대, PU) 는 2개의 모션 벡터들을 포함한다. 블록의 양쪽의 모션 벡터들 (즉, RefPicList0 및 RefPicList1 에 각각 대응하는 모션 벡터들) 이 디스패리티 모션 벡터들일 때, RefPicList1 에 대응하는 모션 벡터는 여분일 수도 있다.

[0220] 3D-HEVC 테스트 모델 1 에서, 비디오 코더는 이웃하는 블록이 RefPicList0 에서의 참조 화상을 이용하는지 여부를 결정한다. 예를 들어, 비디오 코더는 PredFlagL0[xN][yN] 가 1 과 동일한지 여부를 체킹할 수도 있으며, 여기서, (xN, yN) 는 이웃하는 블록에 의해 커버되며, PredFlagL0 는 특정의 좌표들을 커버하는 블록들이 RefPicList0 에서의 참조 화상들을 이용하는지 여부를 나타내는 값들의 어레이이다.

[0221] 더욱이, 3D-HEVC 테스트 모델 1 에서 설명된 바와 같이, 이웃하는 블록이 RefPicList0 에서의 참조 화상을 이용하면, 비디오 코더는 이웃하는 블록에 대한 RefPicList0 참조 인덱스가 RefPicList0 에서의 인터-뷰 참조 화상을 나타내는지 여부를 결정할 수도 있으며, 만약 그렇다면, 인터-뷰 참조 화상이 특정의 참조 뷰 인덱스와 동일한 뷰 순서 인덱스를 가지는지 여부를 결정할 수도 있다. 예를 들어, 비디오 코더는 RefPicList0[RefIdxL0[xN][yN]] 가 refViewIdx 와 동일한 ViewOrderIndex 를 가진 인터-뷰 참조 화상인지 여부를 결정할 수도 있으며, 여기서, RefIdxLx0[xN][yN] 는 이웃하는 블록에 대한 RefPicList0 참조 인덱스이며, refViewIdx 는 특정의 뷰 인덱스이다. 그렇다면, 비디오 코더는 디스패리티 벡터를 이웃 블록에 대한 RefPicList0 모션 벡터와 동일하게 설정할 수도 있다. 그렇지 않으면, 이웃하는 블록의 예측 모드가 스킵 모드이고 이웃하는 블록이 이웃하는 블록이 (예컨대, 이웃하는 블록이 IDV 를 가질 때 발생할 수도 있는) 인터-뷰 모션 벡터를 가진다고 표시하는 인터-뷰 예측 모션 벡터 플래그를 가지면, 비디오 코더는 변수 (예컨대, ivpMVDispN0) 를 이웃하는 블록 (예컨대, IvpMvDispL0[xN, yN]) 의 디스패리티 벡터로 설정할 수도 있다. 비디오 코더는 그후 RefPicList1 에 대해 이 프로세스를 반복할 수도 있다. 뷰 순서 인덱스는 액세스 유닛에서 뷰 성분들의 디코딩 순서를 나타내는 값 (즉, 인덱스) 일 수도 있다.

[0222] 따라서, JCT3V-A1005 의 섹션 G.8.5.2.1.13 에서 설명된 바와 같이:

[0223] 0 내지 1 의 각각의 X 에 대해, availableN 이 1 과 동일하고 PredFlagLX[xN][yN] 이 1 과 동일하면, 다음이 적용된다

[0224] a. RefPicListX[RefIdxLX[xN][yN]] 이 refViewIdx 과 동일한 ViewOrderIndex 를 가진 인터-뷰 참조 화상이면, 다음이 적용된다:

[0225] mvDisp = mvLXN[xN][yN]

[0226] availableDV = 1

[0227] b. 그렇지 않고, PredMode[xN][yN] 이 MODE_SKIP 와 동일하고 IvpMvFlagLX[xN, yN] 이 1

과 동일하면, 다음이 적용된다:

[0228] ivpMvDispNX = IvpMvDispLX[xN, yN]

[0229] availableFlagIvpMvNX = 1

[0230] 본 개시물의 일부 기법들에 따르면, 비디오 코더는 비디오 코더가 3D-HEVC 테스트 모델 1 에서 설명된 바와 같은 RefPicList0 및 RefPicList1양쪽을 체킹하지 않는 프루닝 프로세스 (pruning process) 를 수행할 수도 있다. 대신, 비디오 코더는 단지 참조 화상 리스트들 중 하나를 체킹할 수도 있다. 예를 들어, 비디오 코더는 단지 RefPicList0 를 체킹할 수도 있다. 다른 예들에서, 비디오 코더는 단지 RefPicList1 를 체킹할 수도 있다. 따라서, 이 프루닝 프로세스에서, 공간 또는 시간 이웃하는 블록에 대해, RefPicList0 및 RefPicList1 양쪽에 대응하는 모션 벡터들을 체킹하는 대신 이 블록으로부터 디스패리티 모션 벡터를 유도하기 위해, 비디오 코더는 단지 RefPicList0 (RefPicList1) 에 대응하는 모션 벡터를 체킹한다.

[0231] 예를 들어, 비디오 코더가 디스패리티 벡터 유도 프로세스를 수행할 때, 비디오 코더는 이웃하는 블록에 대해, 이웃하는 블록이 제 2 참조 화상 리스트 (예컨대, RefPicList0 또는 RefPicList1) 에 대응하는 모션 벡터를 가지는지 여부에 관계없이, 제 1 참조 화상 리스트 (예컨대, RefPicList0 또는 RefPicList1) 에 대응하는 이웃하는 블록의 모션 벡터가 디스패리티 모션 벡터인지 여부를, 단지 체킹할 수도 있다. 더욱이, 이 예에서, 제 1 참조 화상 리스트에 대응하는 이웃하는 블록의 모션 벡터가 디스패리티 모션 벡터일 때, 비디오 코더는 디스패리티 모션 벡터를 대표 블록에 대한 디스패리티 벡터로 변환할 수도 있다.

[0232] 도 17 은 본 개시물의 하나 이상의 기법들에 따른, 현재의 블록에 대한 디스패리티 벡터를 결정하는 비디오 코더의 예시적인 동작 (650) 을 예시하는 플로우차트이다. 도 16 및 도 17 의 예들과 같은, 일부 예들에서, IDVs 는 어떤 플래그도 이웃하는 블록이 IDV 를 포함하는지 여부를 시그널링하는데 이용되지 않게, 간단한 방법으로, 표현될 수도 있다. 대신, 이러한 예들에서, 비디오 코더는 이웃하는 블록이 IDV 를 포함한다는 것을 나타내기 위해 이웃하는 블록 (예컨대, 참조 인덱스, 참조 뷰 인덱스, 등) 의 하나 이상의 기존 변수들을 설정할 수도 있다.

[0233] 도 17 의 예에 나타낸 바와 같이, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 현재의 공간 이웃하는 블록이 디스패리티 모션 벡터를 가지는지 여부를 결정할 수도 있다 (652). 현재의 공간 이웃하는 블록이 디스패리티 모션 벡터를 가진다고 결정하는 것에 응답하여 (652 의 "예"), 비디오 코더는 현재의 공간 이웃하는 블록의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다 (654). 디스패리티 모션 벡터를 디스패리티 벡터로 변환한 후, 동작 650 이 종료된다.

[0234] 한편, 현재의 공간 이웃하는 블록이 디스패리티 모션 벡터를 가지지 않는다고 결정하는 것에 응답하여 (652 의 "아니오"), 비디오 코더는 임의의 나머지 공간 이웃하는 블록들이 존재하는지 여부를 결정할 수도 있다 (660). 하나 이상의 공간 이웃하는 블록들이 존재한다고 결정하는 것에 응답하여 (660 의 "예"), 비디오 코더는 공간 이웃하는 블록들의 또 다른 하나에 대해 액션들 652-656 을 반복할 수도 있다. 이러한 방법으로, 비디오 코더는 비디오 코더가 공간 이웃하는 블록들 모두 또는 식별된 공간 모션 벡터를 가질 때까지 액션들 652-656 을 수행할 수도 있다. 비디오 코더는 공간 이웃하는 블록들을 A_1, B_1, B_0, A_0 및 B_2 의 순서로 체킹할 수도 있다.

[0235] 어떤 나머지 공간 이웃하는 블록들도 존재하지 않는다고 결정하는 것에 응답하여 (656 의 "아니오"), 비디오 코더는 시간 모션 벡터 예측이 이용가능하게 되는지 여부를 결정할 수도 있다 (658). 시간 모션 벡터 예측이 이용가능하게 되어 있다고 비디오 코더가 결정하면 (658 의 "예"), 비디오 코더는 도 18 에 나타낸 동작 650 의 부분 (도 17 에서 "A" 로서 표시됨) 을 수행할 수도 있다. 도 18 에 나타낸 동작 650 의 일부를 수행한 후 또는 시간 모션 벡터 예측이 이용불가능하게 되어 있다고 결정한 후 (658 의 "아니오"), 비디오 코더는 IDV 에 대해 공간 이웃하는 블록들을 체킹할 수도 있다. 비디오 코더는 공간 이웃하는 블록들을 A_0, A_1, B_0, B_1 및 B_2 의 순서로 체킹할 수도 있다. 따라서, 도 17 의 예에서, 비디오 코더는 현재의 공간 이웃하는 블록이 스킵 모드에서 코딩되는지 그리고 불법 값으로 설정된 특정의 변수 세트를 가지는지 여부를 결정할 수도 있다 (660). 현재의 공간 이웃하는 블록에 대한 특정의 변수가 불법 값으로 설정되면, 현재의 공간 이웃하는 블록은 IDV 를 갖는다. 예를 들어, 현재의 블록에 대한 디스패리티 벡터를 결정하기 전에, 비디오 코더는 현재의 공간 이웃하는 블록을 이미 코딩하였다. 현재의 공간 이웃하는 블록을 코딩하는 것의 일부로서, 비디오 코더는 모션 예측 프로세스 (예컨대, 병합 모드 프로세스 또는 AMVP 프로세스) 를 이용하여, 현재의 공간 이웃하는 블록에 대한 모션 정보를 결정할 수도 있다. 3D-HEVC 작업 초안 1 에서, 비디오 코더가 모션 예측

프로세스를 수행할 때, 비디오 코더는 현재의 공간 이웃하는 블록이 IDV 를 가지는지 여부를 나타내기 위해 인터-뷰 예측 모션 벡터 플래그 (예컨대, IvpMvFlagLX) 를 설정한다. 그러나, 본 개시물의 하나 이상의 기법들에 따르면, 비디오 코더는 현재의 공간 이웃하는 블록이 IDV 를 가진다는 것을 표시하기 위해, 현재의 공간 이웃하는 블록의 특정의 변수를 불법 값으로 설정할 수도 있다. 위에서 나타낸 바와 같이, 특정의 변수는 참조 인덱스, 참조 뷰 인덱스, 및 다른 변수일 수도 있다. 비디오 코더가 현재의 공간 이웃하는 블록에 대한 모션 정보를 결정한 후, 비디오 코더는 특정의 변수를 다시 필요로 하지 않을 수도 있다. 따라서, 특정의 변수는 현재의 공간 이웃하는 블록이 IDV 를 가지는지 여부를 표시하기 위해 재사용될 수 있다.

[0236] 현재의 공간 이웃하는 블록이 스킵 모드에서 코딩되고 불법 값으로 설정된 특정의 변수를 갖는다고 결정하는 것에 응답하여 (660 의 "예"), 비디오 코더는 현재의 공간 이웃하는 블록의 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다 (662). 현재의 공간 이웃하는 블록의 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환한 후, 동작 650 이 종료된다.

[0237] 한편, 현재의 공간 이웃하는 블록이 스킵 모드에서 코딩되지 않는다고 또는 현재의 공간 이웃하는 블록에 대한 특정의 변수가 불법 값으로 설정되지 않는다고 결정하는 것에 응답하여 (660 의 "아니오"), 비디오 코더는 임의의 나머지 공간 이웃하는 블록들이 존재하는지 여부를 결정할 수도 있다 (664). 하나 이상의 나머지 공간 이웃하는 블록들이 존재한다고 결정하는 것에 응답하여 (664 의 "예"), 비디오 코더는 현재의 공간 이웃하는 블록 처럼 나머지 공간 이웃하는 블록들 중 하나에 대해 액션들 660-664 을 반복할 수도 있다.

[0238] 어떤 나머지 공간 이웃하는 블록들도 존재하지 않는다고 결정하는 것에 응답하여 (664 의 "아니오"), 비디오 코더는 현재의 블록에 대한 디스패리티 벡터가 이용불가능하다고 표시할 수도 있다 (666). 동작 650 이 그후 종료될 수도 있다.

[0239] 도 18 은 도 17 에 나타낸 동작 (650) 의 예시적인 연속부분을 예시하는 플로우차트이다. 도 18 의 예에 나타낸 바와 같이, 비디오 코더는 현재의 블록의 코딩 모드가 AMVP 인지 여부를 결정할 수도 있다 (690). 코딩 모드가 AMVP 라고 결정하는 것에 응답하여 (690 의 "예"), 비디오 코더는 동일 장소에 배치된 화상에서 BR 및 CB 영역들을 정의할 수도 있다 (692). 다음으로, 비디오 코더는 BR 영역을 커버하는 블록 (예컨대, PU) 이 디스패리티 모션 벡터를 가지는지 여부를 결정할 수도 있다 (694). BR 영역을 커버하는 블록이 디스패리티 모션 벡터를 가진다고 결정하는 것에 응답하여 (694 의 "예"), 비디오 코더는 BR 영역을 커버하는 블록의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다 (696). 비디오 코더는 그후 동작 650 의 디스패리티 벡터 유도 프로세스를 종료할 수도 있다.

[0240] 한편, BR 영역을 커버하는 블록이 디스패리티 모션 벡터를 가지지 않는다고 결정하는 것에 응답하여 (694 의 "아니오"), 비디오 코더는 CB 영역을 커버하는 블록이 디스패리티 모션 벡터를 가지는지 여부를 결정할 수도 있다 (698). CB 영역을 커버하는 블록이 디스패리티 모션 벡터를 가진다고 결정하는 것에 응답하여 (698 의 "예"), 비디오 코더는 CB 영역을 커버하는 블록의 디스패리티 모션 벡터를 현재의 블록의 디스패리티 벡터로 변환할 수도 있다 (700). 비디오 코더는 그후 동작 650 의 디스패리티 벡터 유도 프로세스를 종료할 수도 있다. 그러나, CB 영역을 커버하는 블록이 디스패리티 모션 벡터를 가지지 않으면 (698 의 "아니오"), 비디오 코더는 "B" 로 표시된 원에 뒤따르는 도 17 에 나타낸 동작 650 의 일부를 수행할 수도 있다.

[0241] 코딩 모드가 AMVP 가 아니라고 결정하는 것에 응답하여 (690 의 "아니오"), 비디오 코더는 후보 화상 리스트에서 현재의 동일 장소에 배치된 화상에서 BR 및 CB 영역들을 정의할 수도 있다 (702). 다음으로, 비디오 코더는 현재의 후보 화상의 BR 영역을 커버하는 블록 (예컨대, PU) 이 디스패리티 모션 벡터를 가지는지 여부를 결정할 수도 있다 (704). 현재의 후보 화상의 BR 영역을 커버하는 블록이 디스패리티 모션 벡터를 가진다고 결정하는 것에 응답하여 (704 의 "예"), 비디오 코더는 현재의 후보 화상의 BR 영역을 커버하는 블록의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환할 수도 있다 (706). 비디오 코더는 그후 동작 650 의 디스패리티 벡터 유도 프로세스를 종료할 수도 있다.

[0242] 한편, 현재의 후보 화상의 BR 영역을 커버하는 블록이 디스패리티 모션 벡터를 가지지 않는다고 결정하는 것에 응답하여 (704 의 "아니오"), 비디오 코더는 현재의 후보 화상의 CB 영역을 커버하는 블록이 디스패리티 모션 벡터를 가지는지 여부를 결정할 수도 있다 (708). 현재의 후보 화상의 CB 영역을 커버하는 블록이 디스패리티 모션 벡터를 가진다고 결정하는 것에 응답하여 (708 의 "예"), 비디오 코더는 현재의 후보 화상의 CB 영역을 커버하는 블록의 디스패리티 모션 벡터를 현재의 블록의 디스패리티 벡터로 변환할 수도 있다 (710). 비디오 코더는 그후 동작 650 의 디스패리티 벡터 유도 프로세스를 종료할 수도 있다.

[0243] 그러나, 현재의 후보 화상의 CB 영역을 커버하는 블록이 디스패리티 모션 벡터를 가지지 않는다고 결정하는 것에 응답하여 (708 의 "아니오"), 비디오 코더는 후보 화상 리스트에 임의의 나머지 후보 화상들이 존재하는지 여부를 결정할 수도 있다 (712). 후보 화상 리스트에 하나 이상의 나머지 후보 화상들이 존재한다고 결정하는 것에 응답하여 (712 의 "예"), 비디오 코더는 현재의 후보 화상 처럼 후보 화상들의 또 다른 하나에 대해 액션 702-712 을 수행할 수도 있다. 한편, 후보 화상 리스트에 어떤 나머지 후보 화상들도 존재하지 않는다고 결정하는 것에 응답하여 (712 의 "아니오"), 비디오 코더는 "B" 으로 표시된 원에 뒤따르는, 도 17 에 나타낸 동작 650 의 일부를 수행할 수도 있다.

[0244] 예 1. 비디오 데이터를 코딩하는 방법으로서, 본 방법은 하나 이상의 예측 유닛들 (PUs) 을 포함하는 비디오 데이터의 코딩 유닛 (CU) 에 대해 디스패리티 벡터를 결정하는 단계로서, 상기 디스패리티 벡터는 CU 의 하나 이상의 PU들의 각각에 공통인, 상기 결정하는 단계; 및 결정된 디스패리티 벡터에 기초하여 CU 에 대한 인터-뷰 모션 예측 및 인터-뷰 잔여 예측 중 하나 이상을 수행하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0245] 예 2. 예 1 의 방법에 있어서, 본 방법은 디스패리티 벡터를 디스패리티 모션 벡터로서 채용하는 단계를 더 포함하며, 결정된 디스패리티 벡터에 기초하여 CU 에 대한 인터-뷰 모션 예측 및 인터-뷰 잔여 예측 중 하나 이상을 수행하는 것은 디스패리티 모션 벡터를 이용하여, CU 에 대한 인터-뷰 모션 예측 및 인터-뷰 잔여 예측 중 하나 이상을 수행하는 것을 포함하는, 예 1 의 방법.

[0246] 예 3. 예 1 의 방법에 있어서, CU 에 대한 디스패리티 벡터를 결정하는 단계는 비디오 데이터의 참조 블록과 연관되는 암시적인 디스패리티 벡터 (IDV) 및 디스패리티 모션 벡터 (DMV) 중 하나에 기초하여, 디스패리티 벡터를 결정하는 단계를 포함하는, 예 1 의 방법.

[0247] 예 4. 예 3 의 방법에 있어서, 상기 참조 블록은 CU 와 연관되는 공간 이웃하는 블록; 및 CU 와 연관되는 시간 이웃하는 블록 중 하나를 포함하는, 예 3 의 방법.

[0248] 예 5. 예 4 의 방법에 있어서, 상기 CU 는 비디오 데이터의 프레임의 복수의 CU들 중 하나를 포함하며, 상기 공간 이웃하는 블록은 그 CU 이외의, 복수의 CU들의 또 다른 CU 내에 포함되고 프레임 내 CU 에 인접하게 로케이트되는 비디오 데이터의 블록을 포함하는, 예 4 의 방법.

[0249] 예 6. 예 4 의 방법에 있어서, 상기 CU 는 비디오 데이터의 제 1 프레임의 제 1 복수의 CU들 중 하나를 포함하며, 상기 시간 이웃하는 블록은 비디오 데이터의 제 2 프레임 내에 포함되는 비디오 데이터의 블록을 포함하며, 상기 제 1 프레임은 제 2 프레임과 상이한, 예 4 의 방법.

[0250] 예 7. 예 6 의 방법에 있어서, 상기 제 2 프레임 내에 포함되는 블록은 제 2 프레임의 제 2 복수의 CU들의 CU 에 포함되는 비디오 데이터의 블록; 및 제 2 프레임의 제 2 복수의 CU들의 CU 에 인접하게 로케이트되는 비디오 데이터의 블록 중 하나를 포함하며, 여기서 제 2 프레임의 제 2 복수의 CU들의 CU 는 제 1 프레임의 제 1 복수의 CU들의 CU 와 연어를 이루는, 예 6 의 방법.

[0251] 예 8. 예 3 의 방법에 있어서, 상기 CU 는 비디오 데이터의 프레임의 복수의 CU들 중 하나를 포함하며, 상기 참조 블록은 프레임 내 CU 의 좌하단 모서리에 인접하게 로케이트되는 그 CU 이외의, 복수의 CU들의 제 1 CU 내에 포함되는 제 1 이웃하는 PU; 프레임 내 CU 의 좌측에 바로 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 2 CU 내에 포함되는 제 2 이웃하는 PU; 프레임 내 CU 의 상부 좌측 모서리에 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 3 CU 내에 포함되는 제 3 이웃하는 PU; 프레임 내 CU 의 상부에 바로 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 4 CU 내에 포함되는 제 4 이웃하는 PU; 및 프레임 내 CU 의 우상단 모서리에 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 5 CU 내에 포함되는 제 5 이웃하는 PU 중 하나를 포함하는, 예 3 의 방법.

[0252] 예 9. 예 8 의 방법에 있어서, 상기 참조 블록은 제 1 이웃하는 PU, 제 2 이웃하는 PU, 제 3 이웃하는 PU, 제 4 이웃하는 PU, 및 제 5 이웃하는 PU 의 서브세트 중 하나를 포함하는, 예 8 의 방법.

[0253] 예 10. 예 3 의 방법에 있어서, 상기 참조 블록은 CU 와 연어를 이루는 제 2 프레임의 부분 내에 포함되는 비디오 데이터의 제 2 프레임의 제 1 영역; CU 의 PU 와 연어를 이루는 제 2 프레임의 부분 내에 포함되는 제 2 프레임의 제 2 영역; 제 2 프레임의 최대 CU (LCU) 에 대응하고 그리고 제 1 영역 및 제 2 영역의 하나 또는 양자를 포함하는 제 2 프레임의 제 3 영역; 및 제 1 영역 및 제 2 영역의 하나 이상의 우하단 모서리에 인접하게 로케이트되는 제 2 프레임의 제 4 영역 중 하나를 포함하는, 예 3 의 방법.

[0254] 예 11. 예 3 의 방법에 있어서, 상기 참조 블록은 CU 의, 그 CU 이외의, 공간 이웃하는 CU 및 시간 이웃하는 CU 중 하나의 PU 를 포함하는, 예 3 의 방법.

[0255] 예 12. 예 3 의 방법에 있어서, 상기 참조 블록은 CU 의 복수의 PU들의 대표 PU 에 기초하여 결정되는, 예 3 의 방법.

[0256] 예 13. 예 3 의 방법에 있어서, 상기 디스패리티 모션 벡터는 CU 와 연관되는 하나 이상의 참조 블록들의 각각의 디스패리티 모션 벡터 및 암시적인 디스패리티 벡터 중 하나를 미리 결정된 순서에 따라서 결정함으로써 결정되는, 예 3 의 방법.

[0257] 예 14. 예 13 의 방법에 있어서, 상기 미리 결정된 순서는 CU 와 하나 이상의 참조 블록들의 각각 사이의 상관의 양에 기초하여 정의되는, 예 13 의 방법.

[0258] 예 15. 예 1 의 방법에 있어서,

[0259] 참조 인덱스; 참조 뷰 인덱스; 참조 뷰 인덱스의 차이; 불법 참조 인덱스; 불법 참조 뷰 인덱스; 및 불법 참조 뷰 인덱스의 차이 중 하나를 시그널링함으로써, 암시적인 디스패리티 벡터 (IDV) 가 CU 와 연관되는 참조 블록에 대해 존재하는지 여부를 표시하는 단계를 더 포함하는, 예 1 의 방법.

[0260] 예 16. 예 15 의 방법에 있어서, IDV 가 참조 블록에 존재하는지 여부를 단일-비트 플래그를 이용하여 표시하는 것을 회피하는 단계를 더 포함하는, 예 15 의 방법.

[0261] 예 17. 예들 1 - 16 의 임의의 조합의 방법.

[0262] 예 18. 예들 1 - 16 중 임의의 예 또는 이들의 조합들의 방법에 있어서, 상기 방법은 인코더에 의해 수행되는, 예들 1 - 16 중 임의의 예 또는 이들의 조합들의 방법.

[0263] 예 19. 예들 1 - 16 중 임의의 예 또는 이들의 조합들의 방법에 있어서, 상기 방법은 디코더에 의해 수행되는, 예들 1 - 16 중 임의의 예 또는 이들의 조합들의 방법.

[0264] 예 20. 예들 1 - 16 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 시스템.

[0265] 예 21. 실행될 때, 하나 이상의 프로세서들로 하여금, 예들 1 - 16 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 하는 명령들을 포함하는 비일시성 컴퓨터 판독가능 저장 매체.

[0266] 예 22. 예들 1 - 16 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 인코딩 디바이스.

[0267] 예 23. 예 1 - 16 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 디코딩 디바이스.

[0268] 예 24. 하나 이상의 예측 유닛들 (PUs) 을 포함하는 비디오 데이터의 코딩 유닛 (CU) 에 대해 디스패리티 벡터를 결정하는 수단으로서, 상기 디스패리티 벡터는 CU 의 하나 이상의 PU들의 각각에 공통인, 상기 결정하는 수단; 및 결정된 디스패리티 벡터에 기초하여 CU 에 대한 인터-뷰 모션 예측 및 인터-뷰 잔여 예측 중 하나 이상을 수행하는 수단을 포함하는, 비디오 데이터를 코딩하는 디바이스.

[0269] 예 25. 예 24 의 디바이스에 있어서, 디스패리티 벡터를 디스패리티 모션 벡터로서 채용하는 수단을 더 포함하며, 결정된 디스패리티 벡터에 기초하여 CU 에 대한 인터-뷰 모션 예측 및 인터-뷰 잔여 예측 중 하나 이상을 수행하는 수단은 디스패리티 모션 벡터를 이용하여, CU 에 대한 인터-뷰 모션 예측 및 인터-뷰 잔여 예측 중 하나 이상을 수행하는 수단을 포함하는, 예 24 의 디바이스.

[0270] 예 26. 예 24 의 디바이스에 있어서, 상기 CU 에 대한 디스패리티 벡터를 결정하는 수단은 비디오 데이터의 참조 블록과 연관되는 암시적인 디스패리티 벡터 (IDV) 및 디스패리티 모션 벡터 (DMV) 중 하나에 기초하여 디스패리티 벡터를 결정하는 수단을 포함하는, 예 24 의 디바이스.

[0271] 예 27. 예 26 의 디바이스에 있어서, 상기 참조 블록은 CU 와 연관되는 공간 이웃하는 블록; 및 CU 와 연관되는 시간 이웃하는 블록 중 하나를 포함하는, 예 26 의 디바이스.

[0272] 예 28. 예 27 의 디바이스에 있어서, 상기 CU 는 비디오 데이터의 프레임의 복수의 CU들 중 하나를 포함하고, 공간 이웃하는 블록이 그 CU 이외의, 복수의 CU들의 또 다른 CU 내에 포함되고 프레임 내 CU 에 인접하게 로케이트되는 비디오 데이터의 블록을 포함하는, 예 27 의 디바이스.

[0273] 예 29. 예 27 의 디바이스에 있어서, 상기 CU 는 비디오 데이터의 제 1 프레임의 제 1 복수의 CU들 중 하나

를 포함하고, 상기 시간 이웃하는 블록은 비디오 데이터의 제 2 프레임 내에 포함되는 비디오 데이터의 블록을 포함하며, 상기 제 1 프레임은 제 2 프레임과 상이한, 예 27 의 디바이스.

[0274] 예 30. 예 29 의 디바이스에 있어서, 제 2 프레임 내에 포함되는 블록은 제 2 프레임의 제 2 복수의 CU들의 CU 내에 포함되는 비디오 데이터의 블록, 및 제 2 프레임의 제 2 복수의 CU들의 CU 에 인접하게 로케이트되는 비디오 데이터의 블록 중 하나를 포함하며, 상기 제 2 프레임의 제 2 복수의 CU들의 CU 는 제 1 프레임의 제 1 복수의 CU들의 CU 와 연어를 이루는, 예 29 의 디바이스.

[0275] 예 31. 예 26 의 디바이스에 있어서, 상기 CU 는 비디오 데이터의 프레임의 복수의 CU들 중 하나를 포함하며, 상기 참조 블록은 프레임 내 CU 의 좌하단 모서리에 인접하게 로케이트되는 그 CU 이외의, 복수의 CU들의 제 1 CU 내에 포함되는 제 1 이웃하는 PU; 프레임 내 CU 의 좌측에 바로 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 2 CU 내에 포함되는 제 2 이웃하는 PU; 프레임 내 CU 의 상부 좌측 모서리에 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 3 CU 내에 포함되는 제 3 이웃하는 PU; 프레임 내 CU 의 상부에 바로 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 4 CU 내에 포함되는 제 4 이웃하는 PU; 및 프레임 내 CU 의 우상단 모서리에 인접하게 로케이트되는, 그 CU 이외의, 복수의 CU들의 제 5 CU 내에 포함되는 제 5 이웃하는 PU 중 하나를 포함하는, 예 26 의 디바이스.

[0276] 예 32. 예 31 의 디바이스에 있어서, 상기 참조 블록은 제 1 이웃하는 PU, 제 2 이웃하는 PU, 제 3 이웃하는 PU, 제 4 이웃하는 PU, 및 제 5 이웃하는 PU 의 서브세트 중 하나를 포함하는, 예 31 의 디바이스.

[0277] 예 33. 예 26 의 디바이스에 있어서, 상기 참조 블록은 CU 와 연어를 이루는 제 2 프레임의 부분 내에 포함되는 비디오 데이터의 제 2 프레임의 제 1 영역; CU 의 PU 와 연어를 이루는 제 2 프레임의 부분 내에 포함되는 제 2 프레임의 제 2 영역; 제 2 프레임의 최대 CU (LCU) 에 대응하고 그리고 제 1 영역 및 제 2 영역의 하나 또는 양자를 포함하는 제 2 프레임의 제 3 영역; 및 제 1 영역 및 제 2 영역의 하나 이상의 우하단 모서리에 인접하게 로케이트되는 제 2 프레임의 제 4 영역 중 하나를 포함하는, 예 26 의 디바이스.

[0278] 예 34. 예 26 의 디바이스에 있어서, 상기 참조 블록은 CU 의, 그 CU 이외의, 공간 이웃하는 CU 및 시간 이웃하는 CU 중 하나의 PU 를 포함하는, 예 26 의 디바이스.

[0279] 예 35. 예 26 의 디바이스에 있어서, 상기 참조 블록은 CU 의 복수의 PU들의 대표 PU 에 기초하여 결정되는, 예 26 의 디바이스.

[0280] 예 36. 예 26 의 디바이스에 있어서, 상기 디스패리티 모션 벡터를 결정하는 수단은 CU 와 연관되는 하나 이상의 참조 블록들의 각각의 디스패리티 모션 벡터 및 암시적인 디스패리티 벡터 중 하나를 미리 결정된 순서에 따라서 결정하는 수단을 포함하는, 예 26 의 디바이스.

[0281] 예 37. 예 36 의 디바이스에 있어서, 상기 미리 결정된 순서는 CU 와 하나 이상의 참조 블록들의 각각 사이의 상관의 양에 기초하여 정의되는, 예 36 의 디바이스.

[0282] 예 38. 예 24 의 디바이스에 있어서, 참조 인덱스; 참조 뷰 인덱스; 참조 뷰 인덱스의 차이; 불법 참조 인덱스; 불법 참조 뷰 인덱스; 및 불법 참조 뷰 인덱스의 차이 중 하나를 시그널링함으로써, 암시적인 디스패리티 벡터 (IDV) 가 CU 와 연관되는 참조 블록에 대해 존재하는지 여부를 표시하는 수단을 더 포함하는, 예 24 의 디바이스.

[0283] 예 39. 예 38 의 디바이스에 있어서, IDV 가 참조 블록에 대해 존재하는지 여부를 단일-비트 플래그를 이용하여 표시하는 것을 회피하는 수단을 더 포함하는, 예 38 의 디바이스.

[0284] 예 40. 본 개시물에서 설명된 임의의 디바이스 또는 방법.

[0285] 상기 예들 중 임의의 예의 임의의 세부 사항들이 본 개시물과 부합하는 다른 예들과 결합될 수도 있다. 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 또는 컴퓨터-판독가능 매체를 통해서 송신될 수도 있으며, 하드웨어-기반의 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터-판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터-판독가능 저장 매체들, 또는 예컨대, 통신 프로토콜에 따라서 한 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함한 통신 매체들을 포함할 수도 있다. 이런 방법으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비일시성 유형의 컴퓨터-판독가능 저장 매체, 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명하는 기법들의 구현

을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독가능 매체를 포함할 수도 있다.

[0286] 일 예로서, 이에 한정하지 않고, 이런 컴퓨터-판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광디스크 스토리지, 자기디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터-판독가능 매체로 적절히 지칭된다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 무선 기술들, 예컨대 적외선, 라디오, 및 마이크로파를 이용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들 예컨대 적외선, 라디오, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체 및 데이터 저장 매체는 접속부들, 반송파들, 신호들, 또는 다른 일시성 매체를 포함하지 않고, 그 대신, 비-일시성 유형의 저장 매체로 송신되는 것으로 해석되어야 한다. 디스크 (disk) 및 디스크 (disc) 는, 본원에서 사용할 때, 컴팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 Blu-ray 디스크를 포함하며, 디스크들 (disks) 은 데이터를 자기적으로 보통 재생하지만, 디스크들 (discs) 은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 결합들이 또한 컴퓨터-판독가능 매체들의 범위 내에 포함되어야 한다.

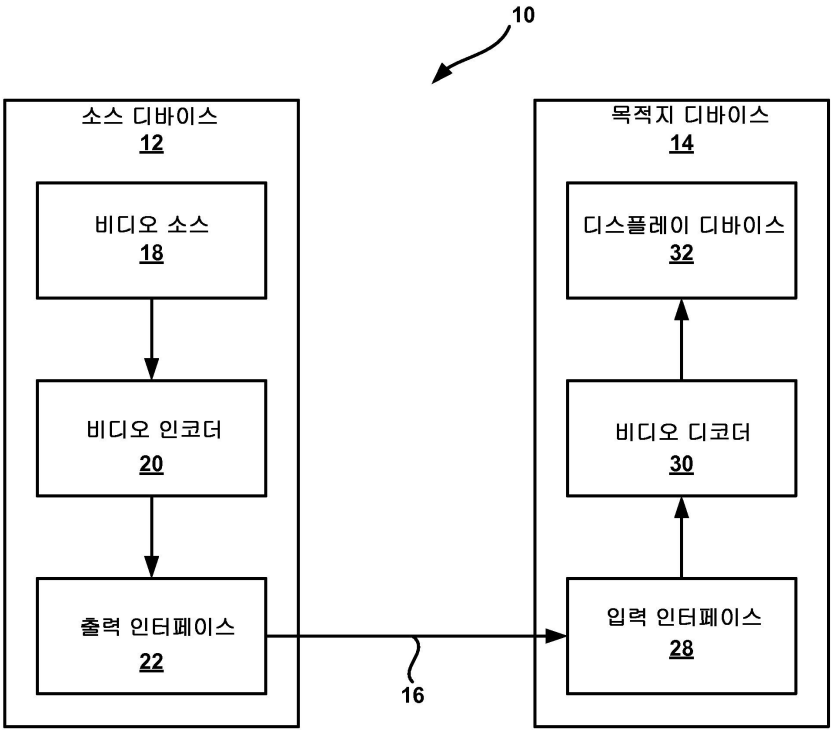
[0287] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 로직 어레이들 (FPGAs), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서" 는, 본원에서 사용될 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능은 전용 하드웨어 및/또는 인코딩 및 디코딩을 위해 구성되는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.

[0288] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트) 를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 여러 구성요소들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 대신, 위에서 설명한 바와 같이, 여러 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

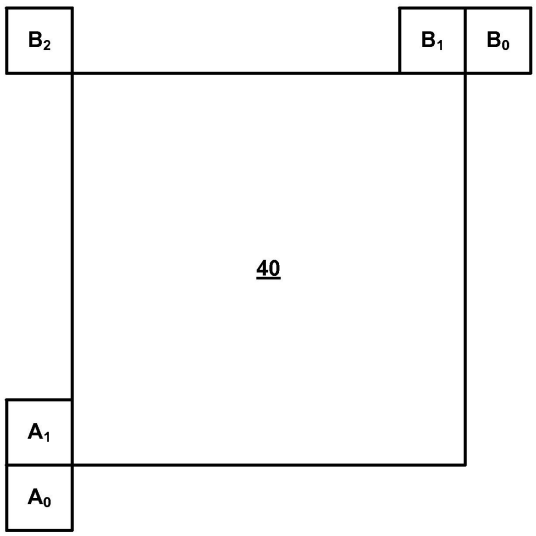
[0289] 여러 예들이 설명되었다. 이들 및 다른 예들은 다음 청구항들의 범위 이내이다.

도면

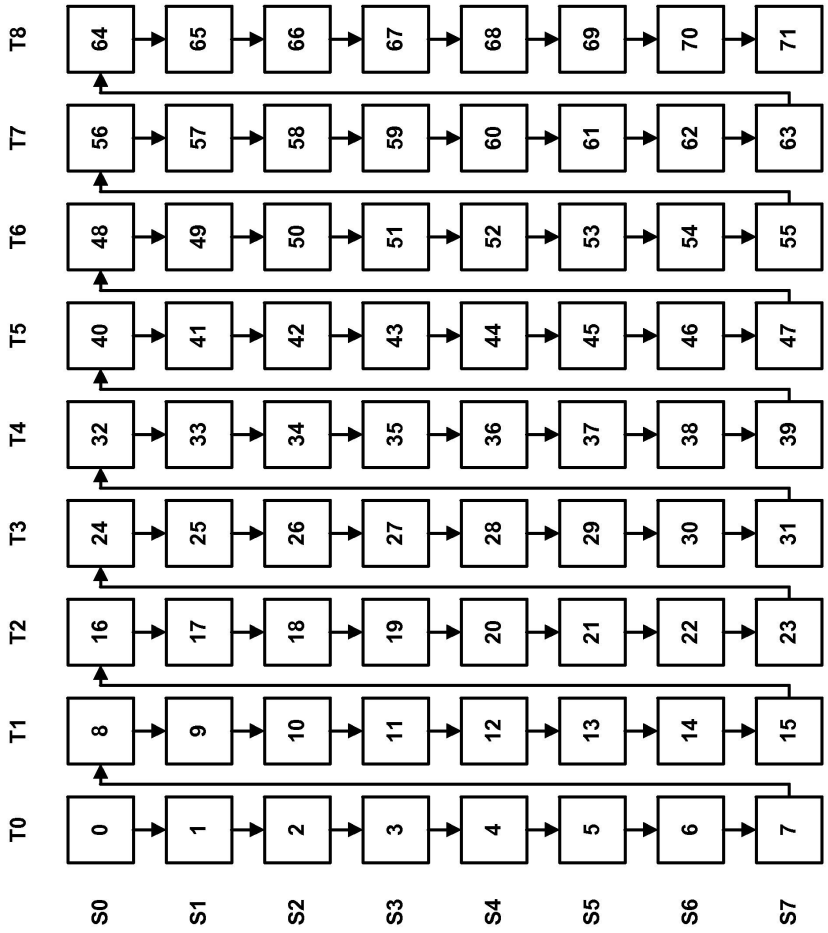
도면1


10
소스 디바이스
12
비디오 소스
18
비디오 인코더
20
출력 인터페이스
22
16
목적지 디바이스
14
디스플레이 디바이스
32
비디오 디코더
30
입력 인터페이스
28

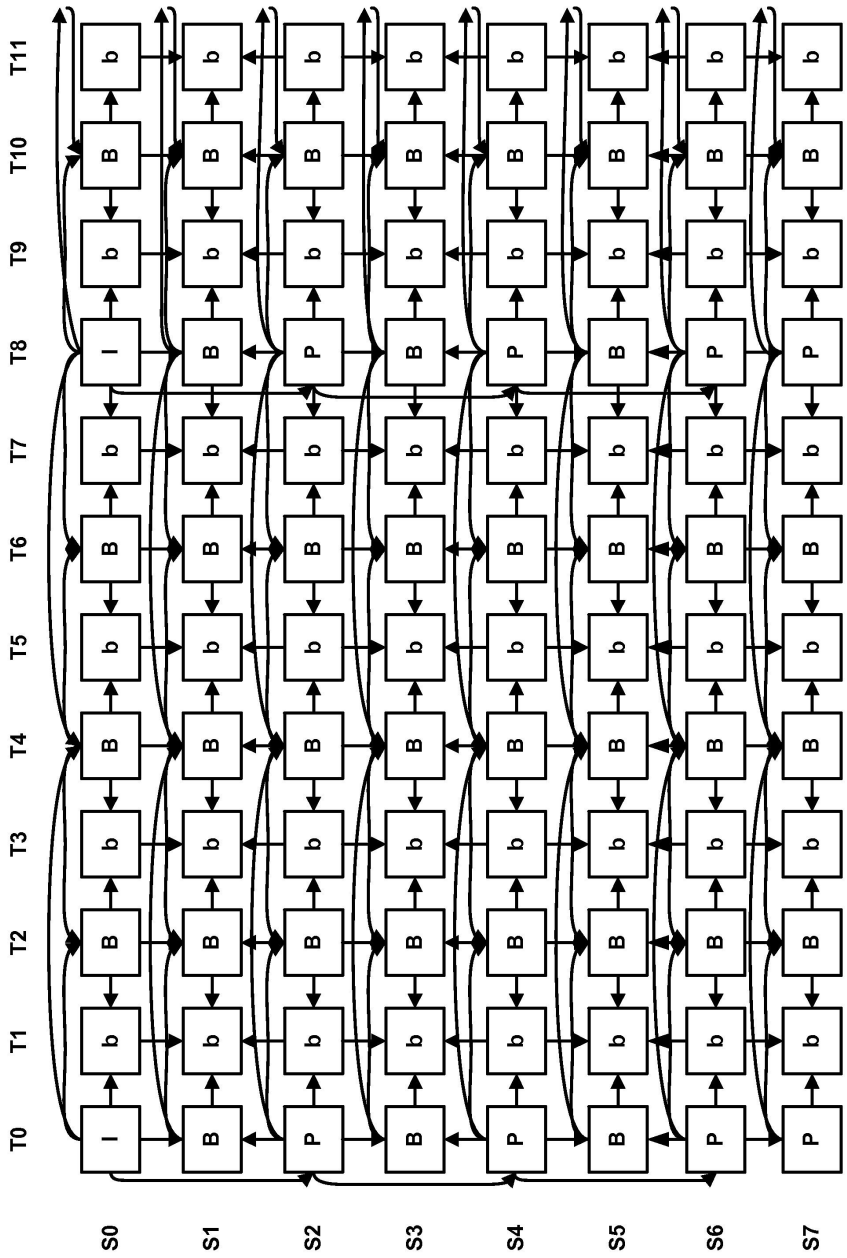
도면2


B2
B1
B0
40
A1
A0

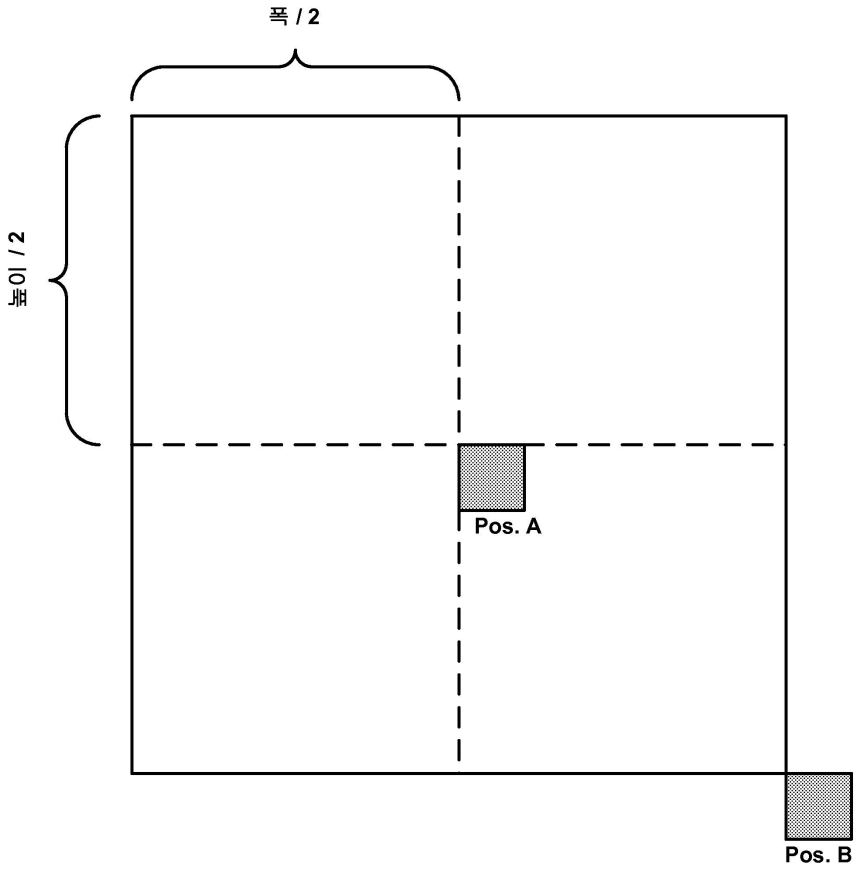
도면3


T8
T7
T6
T5
T4
T3
T2
T1
T0
64 65 66 67 68 69 70 71
56 57 58 59 60 61 62 63
48 49 50 51 52 53 54 55
40 41 42 43 44 45 46 47
32 33 34 35 36 37 38 39
24 25 26 27 28 29 30 31
16 17 18 19 20 21 22 23
8 9 10 11 12 13 14 15
0 1 2 3 4 5 6 7
S0 S1 S2 S3 S4 S5 S6 S7

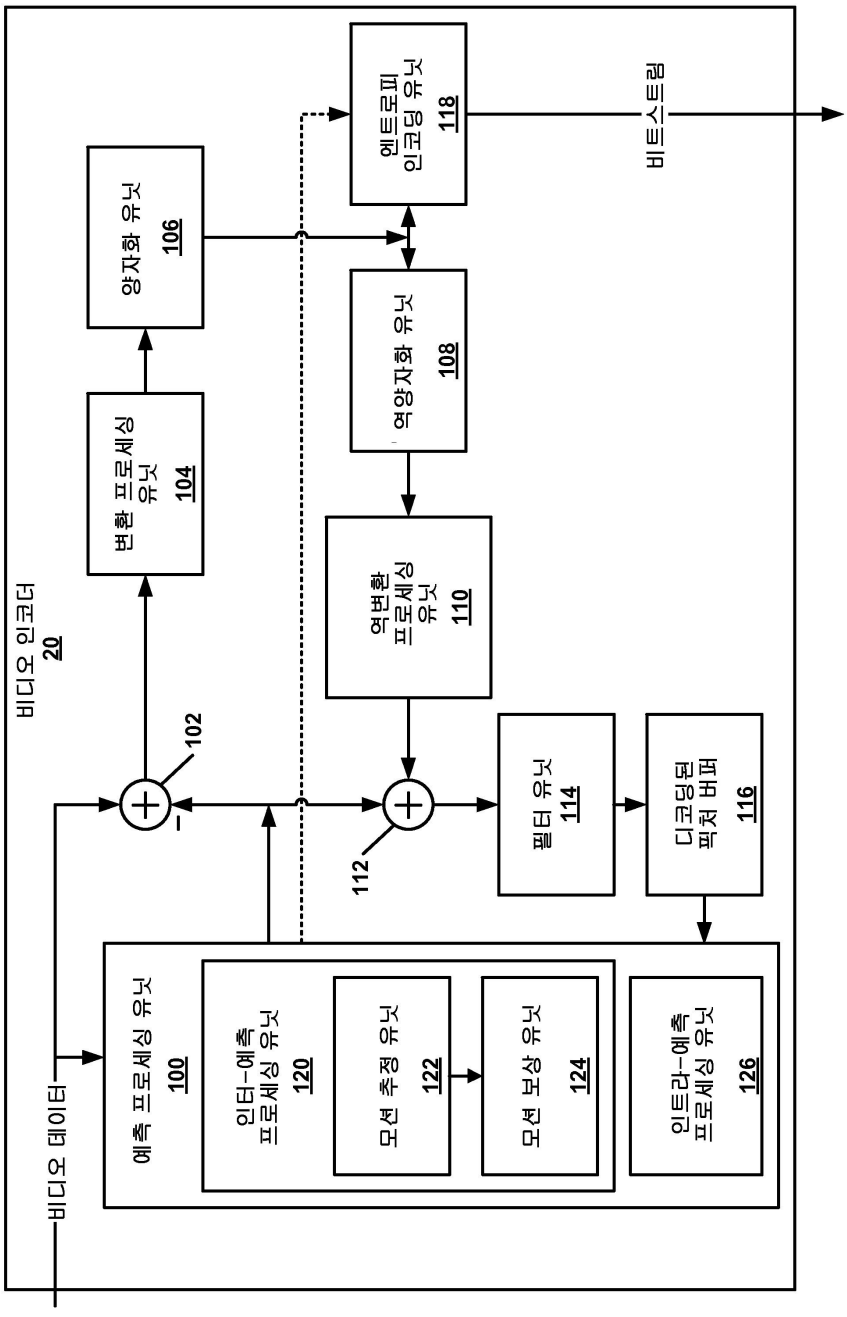
도면4



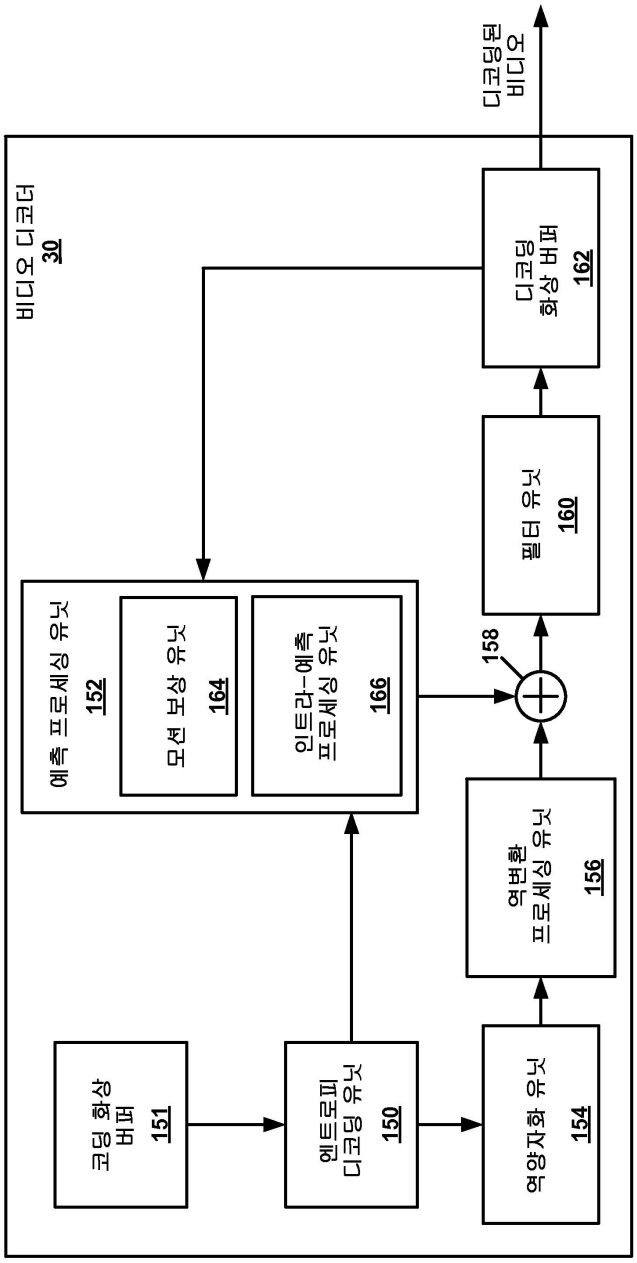
도면5



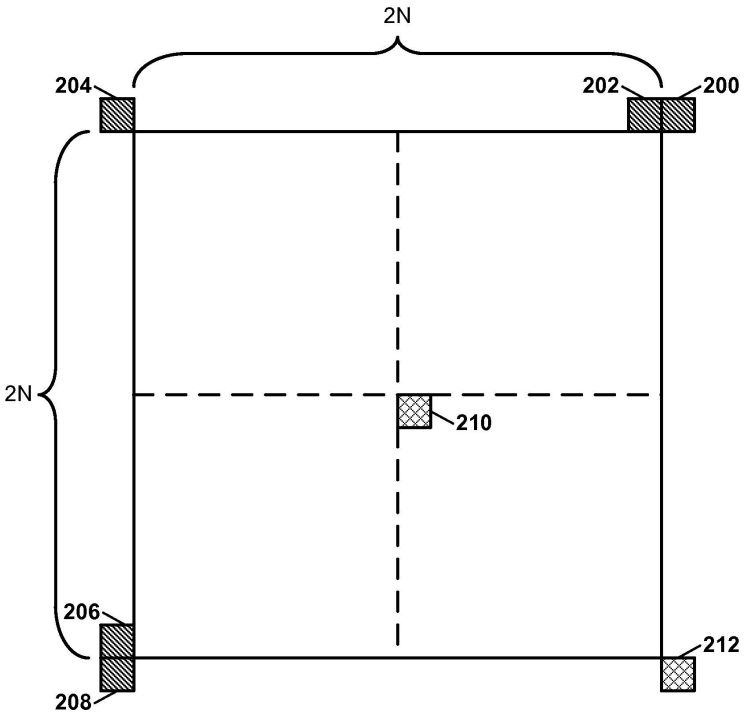
도면6


비디오 인코더
20
비디오 데이터
예측 프로세싱 유닛
100
인터-예측
프로세싱 유닛
120
모션 추정 유닛
122
모션 보상 유닛
124
인트라-예측
프로세싱 유닛
126
102
변환 프로세싱
유닛
104
양자화 유닛
106
역양자화 유닛
108
역변환
프로세싱
유닛
110
112
필터 유닛
114
디코딩된
픽처 버퍼
116
엔트로피
인코딩 유닛
118
비트스트림

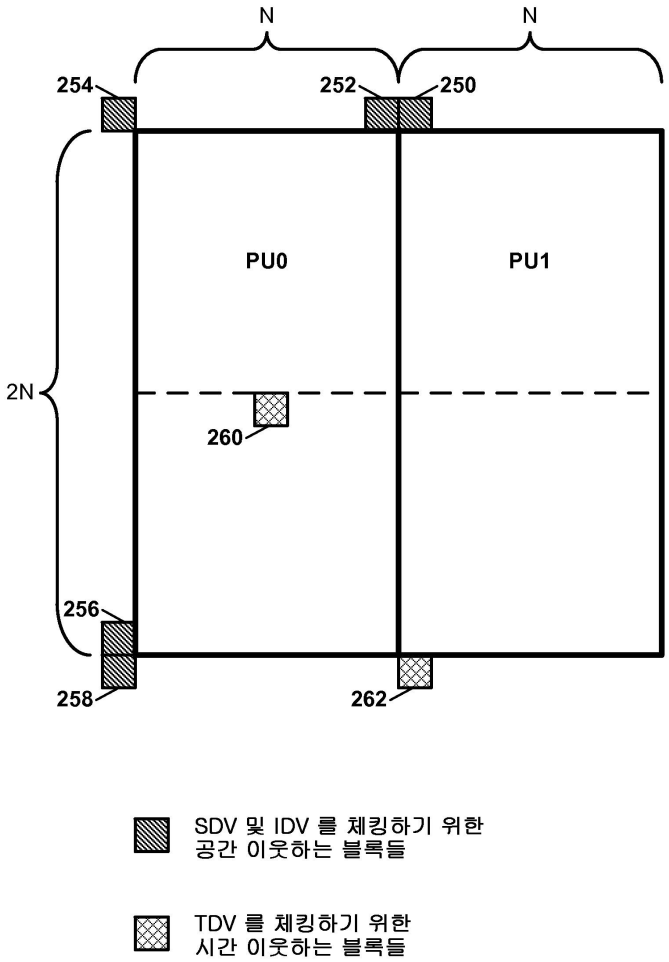
도면7


디코딩된
비디오
비디오 디코더
30
디코딩
화상 버퍼
162
필터 유닛
160
158
예측 프로세싱 유닛
152
모션 보상 유닛
164
인트라-예측
프로세싱 유닛
166
역변환
프로세싱 유닛
156
역양자화 유닛
154
엔트로피
디코딩 유닛
150
코딩 화상
버퍼
151

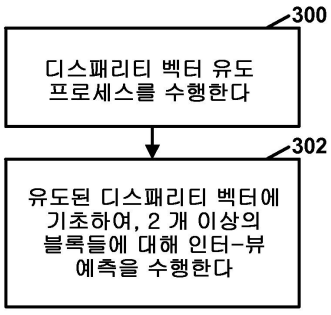
도면8


2N
204
202
200
2N
210
206
212
208
SDV 및 IDV 를 체킹하기 위한
공간 이웃하는 블록들
TDV 를 체크하기 위한
시간 이웃하는 블록들

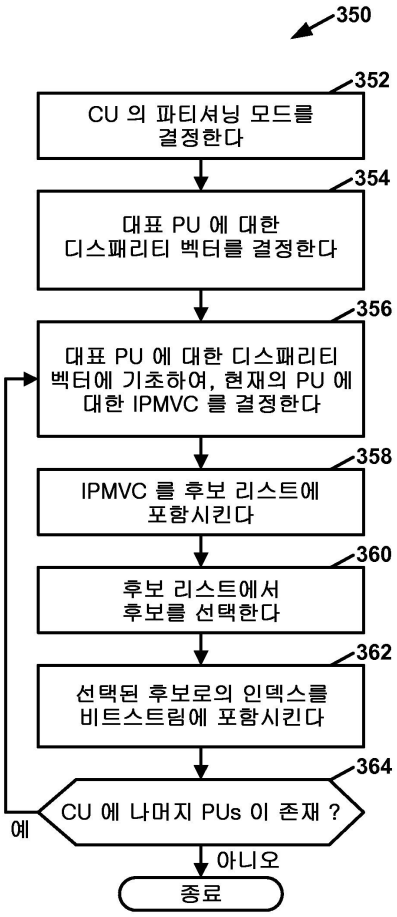
도면9


N
N
254
252
250
PU0
PU1
2N
260
256
258
262
SDV 및 IDV 를 체킹하기 위한 공간 이웃하는 블록들
TDV 를 체킹하기 위한 시간 이웃하는 블록들

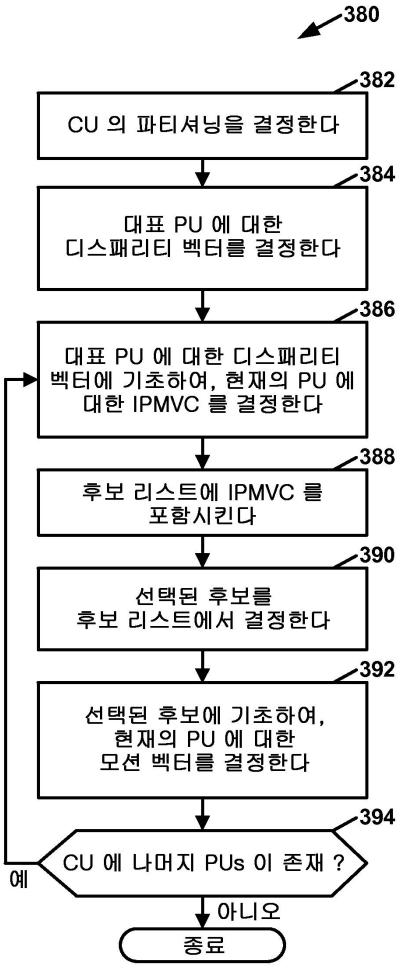
도면10


300
디스패리티 벡터 유도 프로세스를 수행한다
302
유도된 디스패리티 벡터에 기초하여, 2 개 이상의 블록들에 대해 인터-뷰 예측을 수행한다

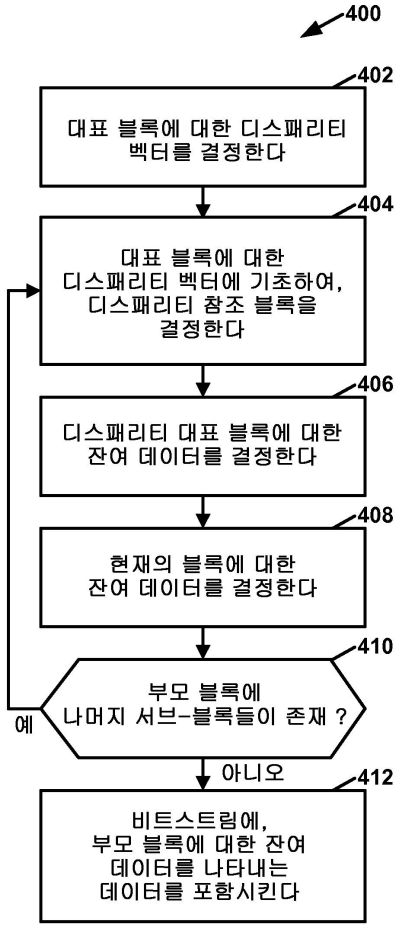
도면11a


350
352
CU 의 파티셔닝 모드를 결정한다
354
대표 PU 에 대한 디스패리티 벡터를 결정한다
356
대표 PU 에 대한 디스패리티 벡터에 기초하여, 현재의 PU 에 대한 IPMVC 를 결정한다
358
IPMVC 를 후보 리스트에 포함시킨다
360
후보 리스트에서 후보를 선택한다
362
선택된 후보로의 인덱스를 비트스트림에 포함시킨다
364
CU 에 나머지 PUs 이 존재 ?
예
아니오
종료

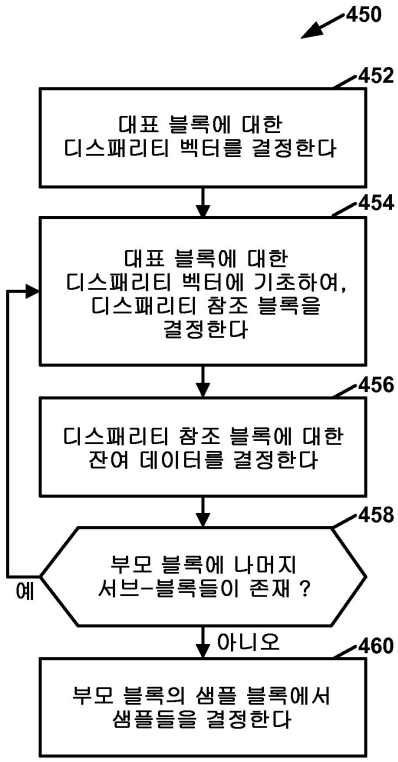
도면11b


380
382
CU 의 파티셔닝을 결정한다
384
대표 PU 에 대한
디스패리티 벡터를 결정한다
386
대표 PU 에 대한 디스패리티
벡터에 기초하여, 현재의 PU 에
대한 IPMVC 를 결정한다
388
후보 리스트에 IPMVC 를
포함시킨다
390
선택된 후보를
후보 리스트에서 결정한다
392
선택된 후보에 기초하여,
현재의 PU 에 대한
모션 벡터를 결정한다
394
CU 에 나머지 PUs 이 존재 ?
예
아니오
종료

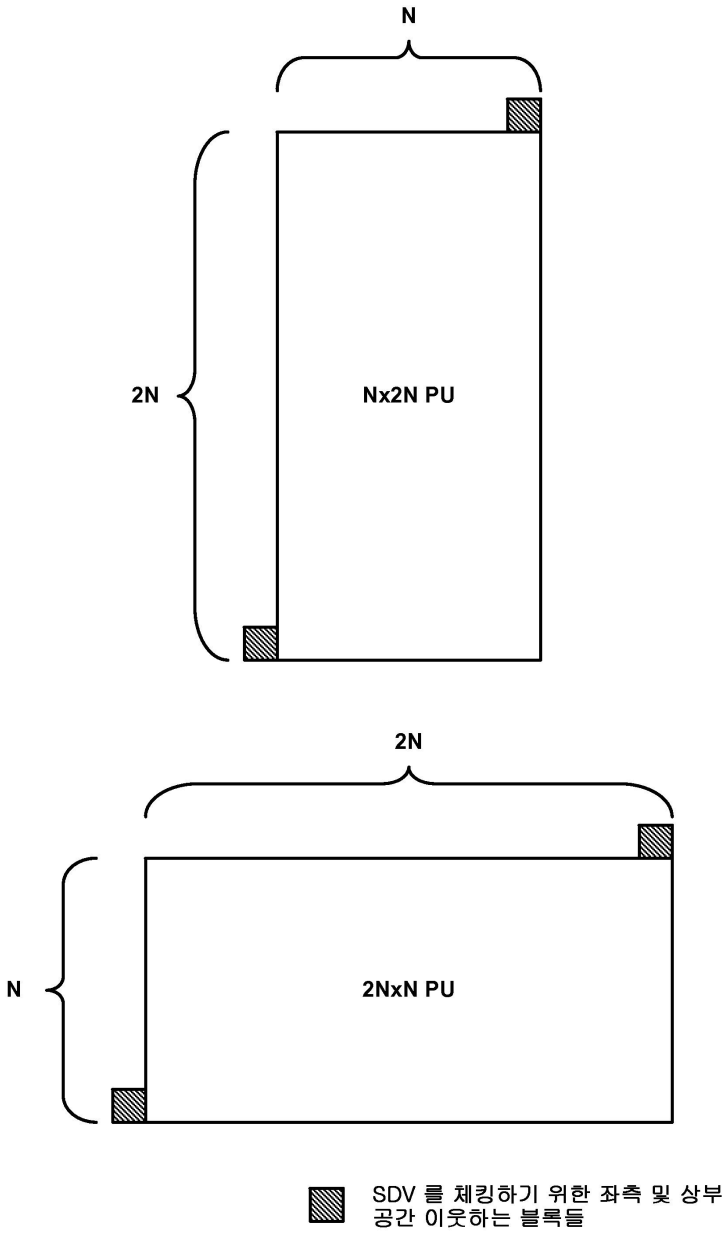
도면12a


400
402
대표 블록에 대한 디스패리티 벡터를 결정한다
404
대표 블록에 대한 디스패리티 벡터에 기초하여, 디스패리티 참조 블록을 결정한다
406
디스패리티 대표 블록에 대한 잔여 데이터를 결정한다
408
현재의 블록에 대한 잔여 데이터를 결정한다
410
부모 블록에 나머지 서브-블록들이 존재 ?
예
아니오
412
비트스트림에, 부모 블록에 대한 잔여 데이터를 나타내는 데이터를 포함시킨다

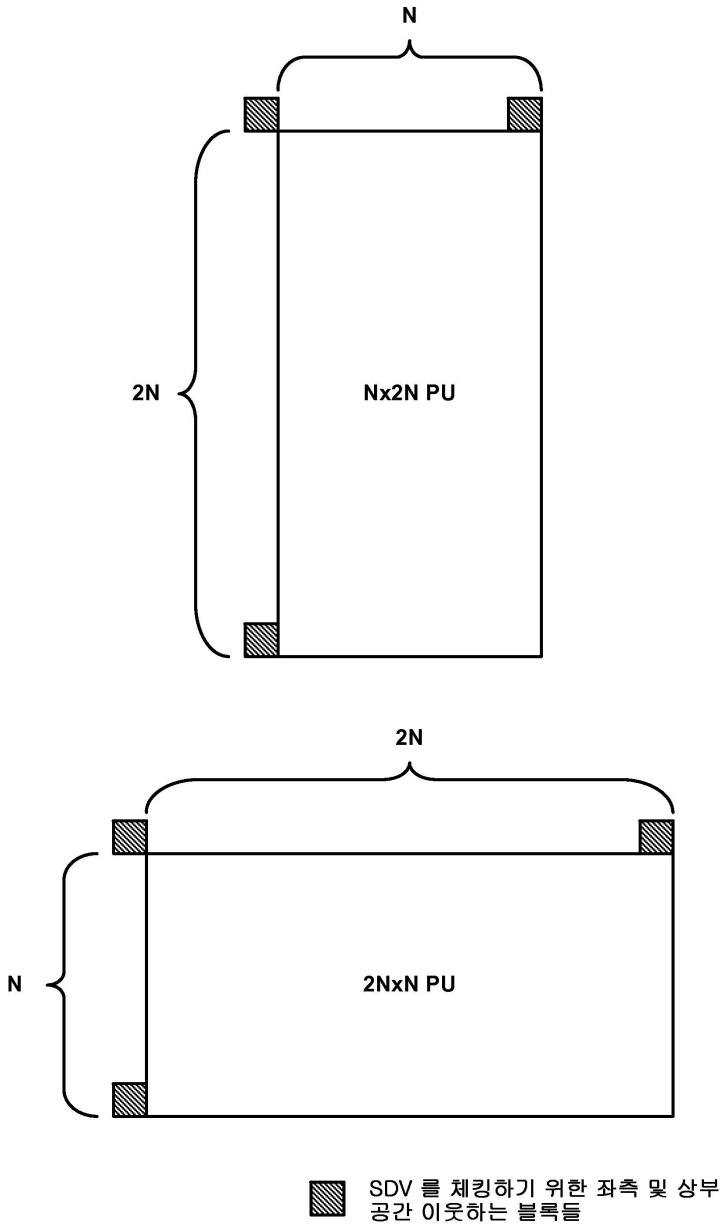
도면12b


450
452
대표 블록에 대한 디스패리티 벡터를 결정한다
454
대표 블록에 대한 디스패리티 벡터에 기초하여, 디스패리티 참조 블록을 결정한다
456
디스패리티 참조 블록에 대한 잔여 데이터를 결정한다
458
부모 블록에 나머지 서브-블록들이 존재 ?
예
아니오
460
부모 블록의 샘플 블록에서 샘플들을 결정한다

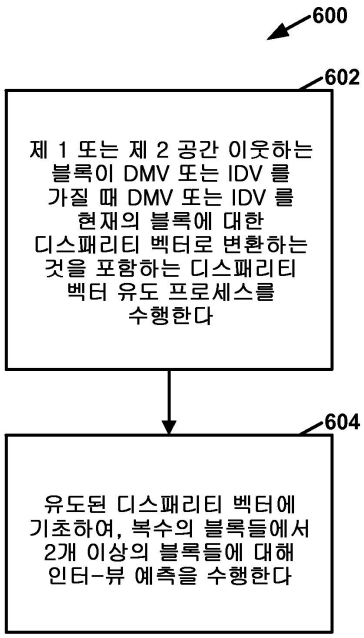
도면13



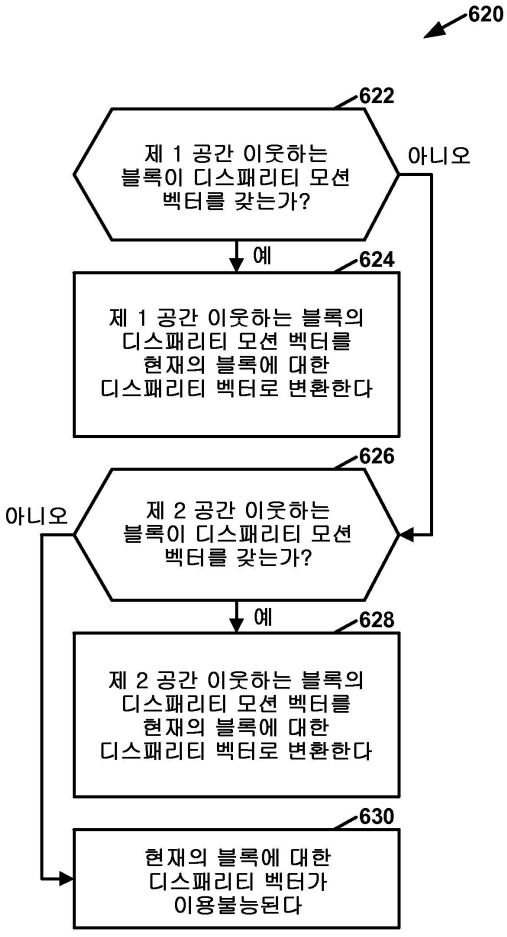
도면14



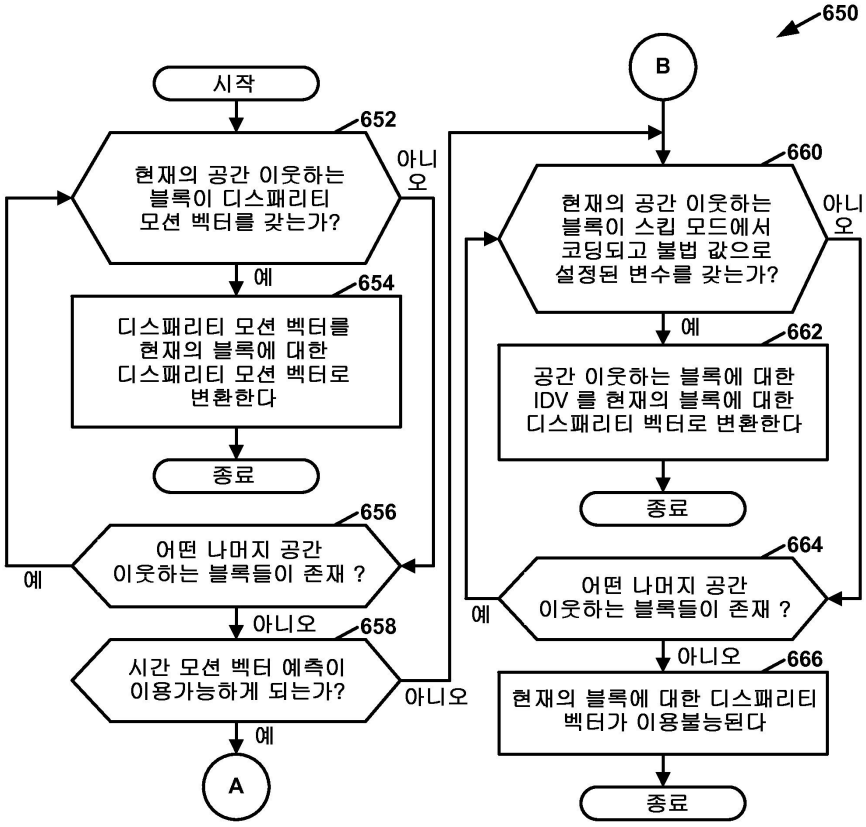
도면15


600
602
제 1 또는 제 2 공간 이웃하는 블록이 DMV 또는 IDV 를 가질 때 DMV 또는 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환하는 것을 포함하는 디스패리티 벡터 유도 프로세스를 수행한다
604
유도된 디스패리티 벡터에 기초하여, 복수의 블록들에서 2개 이상의 블록들에 대해 인터-뷰 예측을 수행한다

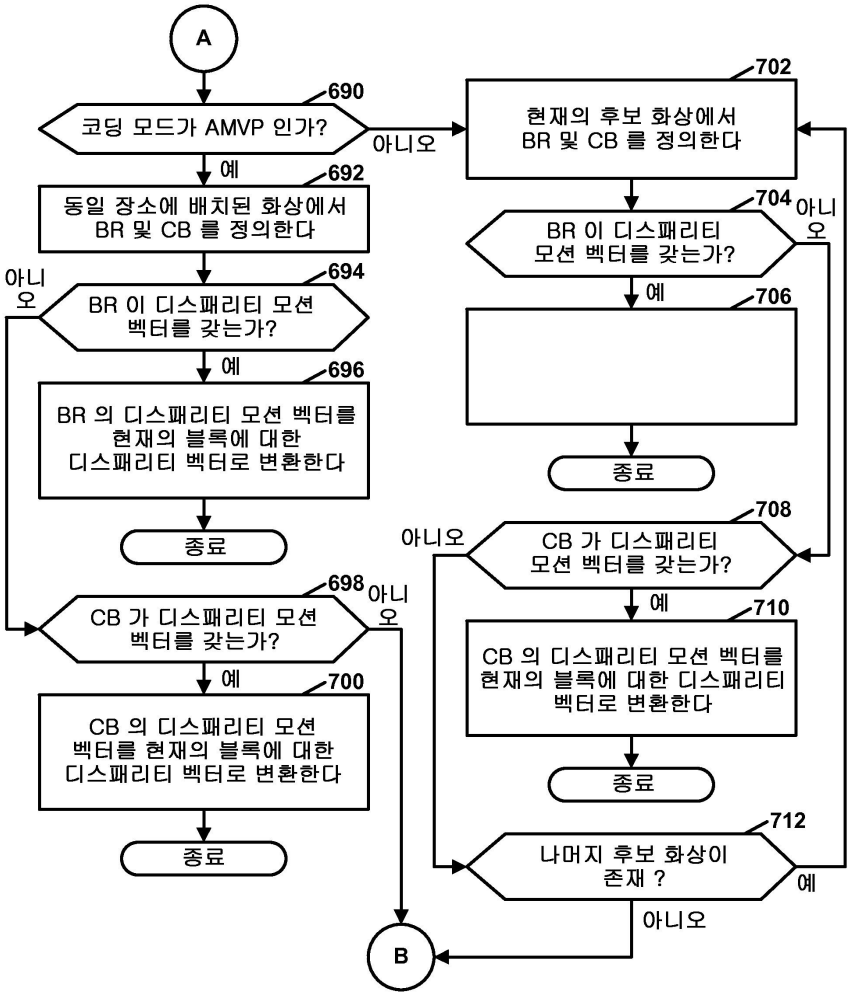
도면16


620
622
제 1 공간 이웃하는 블록이 디스패리티 모션 벡터를 갖는가?
아니오
예
624
제 1 공간 이웃하는 블록의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환한다
626
아니오
제 2 공간 이웃하는 블록이 디스패리티 모션 벡터를 갖는가?
예
628
제 2 공간 이웃하는 블록의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환한다
630
현재의 블록에 대한 디스패리티 벡터가 이용불능된다

도면17


650
시작
652
현재의 공간 이웃하는 블록이 디스패리티 모션 벡터를 갖는가?
아니오
예
654
디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 모션 벡터로 변환한다
종료
656
어떤 나머지 공간 이웃하는 블록들이 존재 ?
예
아니오
658
시간 모션 벡터 예측이 이용가능하게 되는가?
아니오
예
A
B
660
현재의 공간 이웃하는 블록이 스킵 모드에서 코딩되고 불법 값으로 설정된 변수를 갖는가?
아니오
예
662
공간 이웃하는 블록에 대한 IDV 를 현재의 블록에 대한 디스패리티 벡터로 변환한다
종료
664
어떤 나머지 공간 이웃하는 블록들이 존재 ?
예
아니오
666
현재의 블록에 대한 디스패리티 벡터가 이용불능된다
종료

도면18


A
690
코딩 모드가 AMVP 인가?
아니오
예
692
동일 장소에 배치된 화상에서 BR 및 CB 를 정의한다
694
아니오
BR 이 디스패리티 모션 벡터를 갖는가?
예
696
BR 의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환한다
종료
698
아니오
CB 가 디스패리티 모션 벡터를 갖는가?
예
700
CB 의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환한다
종료
B
702
현재의 후보 화상에서 BR 및 CB 를 정의한다
704
아니오
BR 이 디스패리티 모션 벡터를 갖는가?
예
706
종료
708
아니오
CB 가 디스패리티 모션 벡터를 갖는가?
예
710
CB 의 디스패리티 모션 벡터를 현재의 블록에 대한 디스패리티 벡터로 변환한다
종료
712
나머지 후보 화상이 존재 ?
예
아니오