



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0091190
(43) 공개일자 2021년07월21일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>H04N 19/105</i> (2014.01) <i>H04N 19/109</i> (2014.01)
 <i>H04N 19/124</i> (2014.01) <i>H04N 19/13</i> (2014.01)
 <i>H04N 19/167</i> (2014.01) <i>H04N 19/176</i> (2014.01)
 <i>H04N 19/52</i> (2014.01) <i>H04N 19/593</i> (2014.01)
 <i>H04N 19/61</i> (2014.01)</p> <p>(52) CPC특허분류
 <i>H04N 19/105</i> (2015.01)
 <i>H04N 19/109</i> (2015.01)</p> <p>(21) 출원번호 10-2021-7016269
 (22) 출원일자(국제) 2019년12월06일
 심사청구일자 없음
 (85) 번역문제출일자 2021년05월27일
 (86) 국제출원번호 PCT/US2019/064953
 (87) 국제공개번호 WO 2020/118191
 국제공개일자 2020년06월11일</p> <p>(30) 우선권주장
 62/776,373 2018년12월06일 미국(US)
 16/704,827 2019년12월05일 미국(US)</p> | <p>(71) 출원인
 헬컴 인코포레이티드
 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775</p> <p>(72) 발명자
 루사노브스키 드미트로
 미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
 셸리아코프 니콜라이 미하일로비치
 미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
 (뒷면에 계속)</p> <p>(74) 대리인
 특허법인코리아나</p> |
|---|---|

전체 청구항 수 : 총 30 항

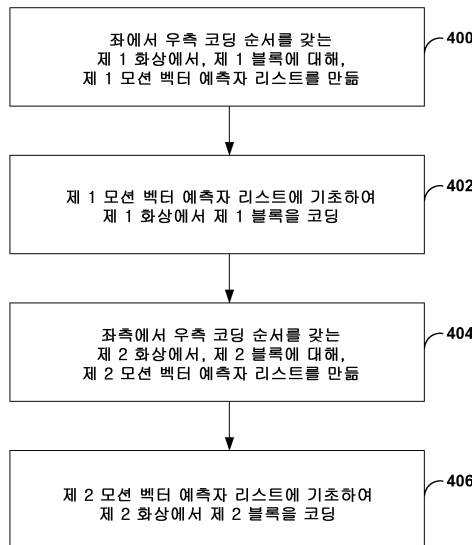
(54) 발명의 명칭 비디오 코딩을 위한 공간-시간적 모션 벡터 예측 패턴

(57) 요약

공간적으로 이웃하는 블록 및 콜로케이트된 블록에 기초하여 모션 벡터 예측자 리스트를 만들기 위한 기술이 설명된다. 비디오 데이터의 코딩 방법은, 좌측에서 우측으로 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 모션 벡터 예측자 리스트를 만드는 단계로서, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는

(뒷면에 계속)

대표도 - 도9



상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 1 모션 벡터 예측자 리스트를 만드는 단계, 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서의 제 1 블록을 코딩하는 단계, 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 모션 벡터 예측자 리스트를 만드는 단계로서, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 2 블록에 대해 우측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 2 모션 벡터 예측자 리스트를 만드는 단계, 및 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서의 제 2 블록을 코딩하는 단계를 포함한다.

(52) CPC특허분류

HO4N 19/124 (2015.01)

HO4N 19/13 (2015.01)

HO4N 19/167 (2015.01)

HO4N 19/176 (2015.01)

HO4N 19/52 (2015.01)

HO4N 19/593 (2015.01)

HO4N 19/61 (2015.01)

(72) 발명자

치엔 웨이-정

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 모션 벡터 예측자 리스트를 만드는 단계로서, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 1 모션 벡터 예측자 리스트를 만드는 단계;

상기 제 1 모션 벡터 예측자 리스트에 기초하여 상기 제 1 화상에서의 상기 제 1 블록을 코딩하는 단계;

우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 모션 벡터 예측자 리스트를 만드는 단계로서, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 2 블록에 대해 우측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 2 모션 벡터 예측자 리스트를 만드는 단계; 및

상기 제 2 모션 벡터 예측자 리스트에 기초하여 상기 제 2 화상에서의 제 2 블록을 코딩하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만드는 단계는 상기 모션 벡터 정보 이용 가능성에 기초하여 다음 순서:

상기 좌측 이웃하는 블록의 모션 벡터 정보;

상측 이웃하는 블록의 모션 벡터 정보;

우상측 이웃하는 블록의 모션 벡터 정보;

좌하측 이웃하는 블록의 모션 벡터 정보; 및

좌상측 이웃하는 블록의 모션 벡터 정보

로 모션 벡터 정보를 갖는 상기 제 1 모션 벡터 예측자 리스트를 만드는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 2 모션 벡터 예측자 리스트를 만드는 단계는 상기 모션 벡터 정보 이용 가능성에 기초하여 다음 순서:

상기 우측 이웃하는 블록의 모션 벡터 정보;

상측 이웃하는 블록의 모션 벡터 정보;

좌상측 이웃하는 블록의 모션 벡터 정보.

우하측 이웃하는 블록의 모션 벡터 정보; 및

우상측 이웃하는 블록의 모션 벡터 정보

로 모션 벡터 정보를 갖는 상기 제 2 모션 벡터 예측자 리스트를 만드는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 1 블록에 대한 상기 좌측 이웃하는 블록은 상기 제 1 블록의 하단 경계와 동일한 하단 경계를 갖는 상기 제 1 블록의 좌측 블록을 포함하고, 상기 우측 이웃하는 블록은 상기 제 2 블록의 하단 경계와 동일한 하단 경계를 갖는 상기 제 2 블록의 우측 블록을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만드는 단계는 병렬로 상기 제 1 모션 벡터 예측자 리스트를 만드는 데 사용되는 복수의 공간적으로 이웃하는 블록을 위한 모션 벡터 정보를 체크하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만드는 단계는

상기 제 1 화상 외의 화상에 위치하는 임의의 다른 콜로케이트된 블록에서 모션 벡터 정보가 이용 가능한지 여부를 결정하기 전에 상기 제 1 화상 외의 상기 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하는 단계; 및

상기 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 1 모션 벡터 예측자 리스트에 추가하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만드는 단계는

상기 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하는 단계;

상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 1 화상 외의 상기 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하는 단계; 및

상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 1 모션 벡터 예측자 리스트에 추가하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 8

제 7 항에 있어서,

상기 하단 콜로케이트된 블록의 우측 경계는 상기 제 1 화상에서 상기 제 1 블록의 우측 경계와 상기 제 1 화상 외의 상기 화상에서 동일한 위치에 있는, 비디오 데이터를 코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 제 2 모션 벡터 예측자 리스트를 만드는 단계는

상기 제 2 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하는 단계;

상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 2 화상 외의 상기 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하는 단

계; 및

상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 2 모션 벡터 예측자 리스트에 추가하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 10

제 9 항에 있어서,

상기 하단 콜로케이트된 블록의 좌측 경계는 상기 제 2 화상에서 상기 제 2 블록의 좌측 경계와 상기 제 2 화상 외의 상기 화상에서 동일한 위치에 있는, 비디오 데이터를 코딩하는 방법.

청구항 11

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만드는 단계는

상기 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하는 단계;

상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 1 화상 외의 상기 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하는 단계;

상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 1 화상 외의 상기 화상에 위치한 우측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하는 단계; 및

상기 우측 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 우측 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 1 모션 벡터 예측자 리스트에 추가하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 12

제 11 항에 있어서,

상기 우측 콜로케이트된 블록의 하단 경계는 상기 제 1 화상에서 상기 제 1 블록의 하단 경계와 상기 제 1 화상 외의 상기 화상에서 동일한 위치에 있는, 비디오 데이터를 코딩하는 방법.

청구항 13

제 1 항에 있어서,

상기 제 2 모션 벡터 예측자 리스트를 만드는 단계는

상기 제 2 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하는 단계;

상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 2 화상 외의 상기 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하는 단계;

상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 2 화상 외의 상기 화상에 위치한 좌측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하는 단계; 및

상기 좌측 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 좌측 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 2 모션 벡터 예측자 리스트에 추가하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 14

제 13 항에 있어서,

상기 좌측 콜로케이트된 블록의 하단 경계는 상기 제 2 화상에서 상기 제 2 블록의 하단 경계와 상기 제 2 화상 외의 상기 화상에서 동일한 위치에 있는, 비디오 데이터를 코딩하는 방법.

청구항 15

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만드는 단계는 상기 제 1 블록이 임계 크기보다 큰 크기를 갖는 것에 기초하여, 제 1 최대 크기를 갖는, 상기 제 1 모션 벡터 예측자 리스트를 만드는 단계를 포함하고, 상기 방법은

상기 제 1 화상에서 제 3 블록에 대해, 상기 제 3 블록이 상기 임계 크기보다 작은 크기를 갖는 것에 기초하여, 상기 제 1 최대 크기보다 작은 제 2 최대 크기를 갖는, 제 3 모션 벡터 예측자 리스트를 만드는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 16

제 15 항에 있어서,

상기 제 1 최대 크기는 6과 같고, 상기 제 2 최대 크기는 4 와 같고, 상기 임계 크기는 $N \times 4$ 또는 $4 \times N$ 와 같고, 여기서 N 은 8보다 작거나 같은, 비디오 데이터를 코딩하는 방법.

청구항 17

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트에 기초하여 상기 제 1 화상에서 상기 제 1 블록을 코딩하는 단계는 상기 제 1 모션 벡터 예측자 리스트에 기초하여 상기 제 1 화상에서의 상기 제 1 블록을 디코딩하는 단계를 포함하고, 상기 제 1 블록을 디코딩하는 단계는:

상기 제 1 모션 벡터 예측자 리스트의 엔트리를 나타내는 정보를 수신하는 단계;

상기 엔트리에 기초하여 상기 제 1 블록을 위한 모션 벡터를 결정하는 단계;

상기 모션 벡터에 기초하여 상기 제 1 블록을 위한 예측 블록을 결정하는 단계; 및

상기 예측 블록 및 상기 예측 블록과 상기 제 1 블록 사이의 차이를 나타내는 잔차 데이터를 나타내는 수신된 정보에 기초하여 상기 제 1 블록을 재구성하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 18

제 1 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트에 기초하여 상기 제 1 화상에서 상기 제 1 블록을 코딩하는 단계는 상기 제 1 모션 벡터 예측자 리스트에 기초하여 상기 제 1 화상에서의 상기 제 1 블록을 인코딩하는 단계를 포함하고, 상기 제 1 블록을 인코딩하는 단계는

상기 제 1 블록을 위한 예측 블록을 결정하는 단계;

상기 예측 블록을 식별하는 상기 제 1 블록을 위한 모션 벡터를 결정하는 단계;

상기 모션 벡터에 기초하여 상기 제 1 모션 벡터 예측자 리스트에서의 엔트리를 결정하는 단계;

상기 엔트리를 나타내는 정보를 시그널링하는 단계; 및

상기 제 1 블록과 상기 예측 블록 사이의 차이를 나타내는 잔차 데이터를 나타내는 정보를 시그널링하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 19

비디오 데이터를 코딩하기 위한 디바이스로서,
 모션 벡터 정보를 저장하도록 구성된 메모리; 및
 상기 메모리에 연결된 프로세싱 회로를 포함하고,
 상기 프로세싱 회로는:

좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 모션 벡터 예측자 리스트를 만드는 것으로서, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 메모리에 저장된 상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 1 모션 벡터 예측자 리스트를 만들고;

상기 제 1 모션 벡터 예측자 리스트에 기초하여 상기 제 1 화상에서의 상기 제 1 블록을 코딩하고;

우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 모션 벡터 예측자 리스트를 만드는 것으로서, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 메모리에 저장된 상기 제 2 블록에 대해 우측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 2 모션 벡터 예측자 리스트를 만들고;

상기 제 2 모션 벡터 예측자 리스트에 기초하여 상기 제 2 화상에서의 상기 제 2 블록을 코딩하게 하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 20

제 19 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만들기 위해, 상기 프로세싱 회로는 상기 모션 벡터 정보 이용 가능성에 기초하여 다음 순서:

상기 좌측 이웃하는 블록의 모션 벡터 정보;

상측 이웃하는 블록의 모션 벡터 정보;

우상측 이웃하는 블록의 모션 벡터 정보;

좌하측 이웃하는 블록의 모션 벡터 정보; 및

좌상측 이웃하는 블록의 모션 벡터 정보

로 모션 벡터 정보를 갖는 상기 제 1 모션 벡터 예측자 리스트를 만들도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 21

제 19 항에 있어서,

상기 제 2 모션 벡터 예측자 리스트를 만들기 위해, 상기 프로세싱 회로는 상기 모션 벡터 정보 이용 가능성에 기초하여 다음 순서:

상기 우측 이웃하는 블록의 모션 벡터 정보;

상측 이웃하는 블록의 모션 벡터 정보;

좌상측 이웃하는 블록의 모션 벡터 정보.

우하측 이웃하는 블록의 모션 벡터 정보; 및

우상측 이웃하는 블록의 모션 벡터 정보

로 모션 벡터 정보를 갖는 상기 제 2 모션 벡터 예측자 리스트를 만들도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 22

제 19 항에 있어서,

상기 제 1 블록에 대한 상기 좌측 이웃하는 블록은 상기 제 1 블록의 하단 경계와 동일한 하단 경계를 갖는 상기 제 1 블록의 좌측 블록을 포함하고, 상기 우측 이웃하는 블록은 상기 제 2 블록의 하단 경계와 동일한 하단 경계를 갖는 상기 제 2 블록의 우측 블록을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 23

제 19 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만들기 위해, 상기 프로세싱 회로는 병렬로 상기 제 1 모션 벡터 예측자 리스트를 만드는 데 사용되는 복수의 공간적으로 이웃하는 블록을 위한 모션 벡터 정보를 체크하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 24

제 19 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만들기 위해, 상기 프로세싱 회로는

상기 제 1 화상 외의 화상에 위치하는 임의의 다른 콜로케이트된 블록에서 모션 벡터 정보가 이용 가능한지 여부를 결정하기 전에 상기 제 1 화상 외의 상기 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고;

상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 1 모션 벡터 예측자 리스트에 추가하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 25

제 19 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만들기 위해, 상기 프로세싱 회로는:

상기 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고;

상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 1 화상 외의 상기 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고;

상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 1 모션 벡터 예측자 리스트에 추가하도록 구성되고,

상기 하단 콜로케이트된 블록의 우측 경계는 상기 제 1 화상에서 상기 제 1 블록의 우측 경계와 상기 제 1 화상 외의 상기 화상에서 동일한 위치에 있는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 26

제 19 항에 있어서,

상기 제 2 모션 벡터 예측자 리스트를 만들기 위해, 상기 프로세싱 회로는

상기 제 2 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고;

상기 중앙 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 2 화상 외의 상기 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고;

상기 하단 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 상기 제 2 화상 외의 상기 화상에 위치한 좌측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고;

상기 좌측 콜로케이트된 블록을 위한 상기 모션 벡터 정보가 이용 가능한 것에 기초하여, 상기 좌측 콜로케이트된 블록을 위한 상기 모션 벡터 정보를 상기 제 2 모션 벡터 예측자 리스트에 추가하도록 구성되고,

상기 좌측 콜로케이트된 블록의 하단 경계는 상기 제 2 화상에서 상기 제 2 블록의 하단 경계와 상기 제 2 화상 외의 상기 화상에서 동일한 위치에 있는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 27

제 19 항에 있어서,

상기 제 1 모션 벡터 예측자 리스트를 만들기 위해, 상기 프로세싱 회로는 상기 제 1 블록이 임계 크기보다 큰 크기를 갖는 것에 기초하여, 제 1 최대 크기를 갖는, 상기 제 1 모션 벡터 예측자 리스트를 만들도록 구성되고, 상기 프로세싱 회로는

상기 제 1 화상에서 제 3 블록에 대해, 상기 제 3 블록이 상기 임계 크기보다 작은 크기를 갖는 것에 기초하여, 상기 제 1 최대 크기보다 작은 제 2 최대 크기를 갖는, 제 3 모션 벡터 예측자 리스트를 만들도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 28

제 27 항에 있어서,

상기 제 1 최대 크기는 6과 같고, 상기 제 2 최대 크기는 4 와 같고, 상기 임계 크기는 $N \times 4$ 또는 $4 \times N$ 와 같고, 여기서 N 은 8보다 작거나 같은, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 29

제 19 항에 있어서,

상기 디바이스는 무선 통신 디바이스를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 30

명령들을 저장하는 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금,

좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 모션 벡터 예측자 리스트를 만드는 것으로서, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 1 모션 벡터 예측자 리스트를 만들고;

상기 제 1 모션 벡터 예측자 리스트에 기초하여 상기 제 1 화상에서의 상기 제 1 블록을 코딩하고;

우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 모션 벡터 예측자 리스트를 만드는 것으로서, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 2 블록에 대해 우측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 2 모션 벡터 예측자 리스트를 만들고;

상기 제 2 모션 벡터 예측자 리스트에 기초하여 상기 제 2 화상에서의 상기 제 2 블록을 코딩하게 하는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2018 년 12 월 6 일 출원된 미국 가출원 제 62/776,373 호의 이익을 주장하는, 2019 년 12 월 5 일 출원된 미국 출원 제 16/704,827 호에 대한 우선권을 주장하며, 이들 출원의 전체 내용은 참조로서 본원에 통합된다.

[0002] 기술 분야

[0003] 본 개시는 비디오 인코딩 및 비디오 디코딩에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, PDA(personal digital assistant)들, 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, 전자책 단말기,

디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰", 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, 어드밴스드 비디오 코딩 (AVC), 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, ITU-T H.265/고효율 비디오 코딩 (HEVC), 및 이러한 표준들의 확장들에 의해 정의된 표준들에 기재된 것들과 같은, 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 그러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 코딩 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소 또는 제거하기 위해 공간 (화상내) 예측 및/또는 시간 (화상간) 예측을 포함한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (예를 들어, 비디오 화상 또는 비디오 화상의 일부) 는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 코딩 트리 유닛들 (CTU들), 코딩 유닛들 (CU들) 및/또는 코딩 노드들로도 지칭될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 사용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스내의 비디오 블록들은 동일한 화상내의 이웃하는 블록들에 있는 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들 내의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수도 있고, 참조 화상들은 참조 프레임들로 지칭될 수도 있다.

발명의 내용

[0006] 개요

[0007] 일반적으로, 본 개시는 비디오 코딩에서 인터 예측 및 모션 벡터 예측을 위한 기술을 설명한다. 보다 구체적으로, 본 개시는 모션 벡터 예측자 리스트에 기초한 모션 벡터 예측과 같은 공간-시간적 모션 벡터 예측을 위한 기술을 설명한다. 본 개시의 기술은 H.265/HEVC (High Efficiency Video Coding) 과 같은 기존 비디오 코덱들 중 어느 것과도 함께 이용될 수 있거나, 또는 H.266/VVC (Versatile Video Coding) 와 같은 임의의 미래의 비디오 코딩 표준에서의 효율적인 코딩 툴일 수 있다.

[0008] 더 상세히 설명된 바와 같이, 본 개시는 공간적으로 그리고 시간적으로 이웃하는 블록의 모션 벡터 정보를 포함하는 모션 벡터 예측자 리스트를 구성하는 예시적인 기술을 설명한다. 일부 경우에, 화상의 코딩 순서 (예를 들어, 좌측에서 우측 또는 우측에서 좌측) 는 어느 공간적으로 이웃하는 블록들이 모션 벡터 예측자 리스트를 구성하는 데 이용 가능함에 영향을 줄 수 있다. 따라서, 본 개시는 비디오 코더 (예를 들어, 비디오 인코더 또는 비디오 디코더) 가 화상에 사용되는 코딩 순서에 기초하여 모션 벡터 예측자 리스트를 구성할 수 있는 예시적인 기술을 설명한다. 또한, 시간적으로 이웃하는 블록 (콜로케이트된 블록 (collocated block) 이라고도 함) 에 대해, 비디오 코더는 특정 순서로 시간적으로 이웃하는 블록의 모션 벡터 정보에 액세스하여 그의 이용 가능성을 체크하도록 구성될 수도 있다.

[0009] 일례에서, 본 개시는 비디오 데이터를 코딩하는 방법을 설명하며, 그 방법은: 좌측에서 우측으로 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 모션 벡터 예측자 리스트를 만드는 단계로서, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 1 모션 벡터 예측자 리스트를 만드는 단계, 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서의 제 1 블록을 코딩하는 단계, 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 모션 벡터 예측자 리스트를 만드는 단계로서, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 2 블록에 대해 우측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 2 모션 벡터 예측자 리스트를 만드는 단계, 및 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서의 제 2 블록을 코딩하는 단계를 포함한다.

[0010] 일례에서, 본 개시는 비디오 데이터를 코딩하기 위한 디바이스를 설명하며, 상기 디바이스는 모션 벡터 정보를 저장하도록 구성된 메모리 및 그 메모리에 연결된 프로세싱 회로를 포함한다. 프로세싱 회로는 다음과 같이 구성된다 좌측에서 우측으로 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 모션 벡터 예측자 리스트를 만드는 것으로서, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 메모리에 저장된 상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 1 모션 벡터 예측자 리스트를 만들고, 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서의 제 1 블록을 코딩하고, 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 모션 벡터 예측자 리스트를 만드는 것으로서, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 메모리에 저장된 상기 제 2 블록에 대해 우측 이웃하는 블

록의 모션 벡터 정보에 기초하는, 상기 제 2 모션 벡터 예측자 리스트를 만들고, 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서의 제 2 블록을 코딩한다.

[0011] 일례에서, 본 개시는 실행될 때 하나 이상의 프로세서들로 하여금, 좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 모션 벡터 예측자 리스트를 만들게 하는 것으로서, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 1 모션 벡터 예측자 리스트를 만들게 하고, 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서의 제 1 블록을 코딩하게 하고, 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 모션 벡터 예측자 리스트를 만들게 하는 것으로서, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 상기 제 2 블록에 대해 우측 이웃하는 블록의 모션 벡터 정보에 기초하는, 상기 제 2 모션 벡터 예측자 리스트를 만들게 하고, 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서의 제 2 블록을 코딩하게 하는 명령들을 저장한 컴퓨터 판독가능 저장 매체를 설명한다.

[0012] 하나 이상의 예들의 상세는 첨부 도면 및 아래의 설명에 제시되어 있다. 다른 특징, 목적 및 이점들은 상세한 설명, 도면, 및 청구항들로부터 분명해질 것이다.

도면의 간단한 설명

[0013] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록 도이다.

도 2 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코더를 예시하는 블록 도이다.

도 3 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더를 예시하는 블록 도이다.

도 4a 는 병합 모드를 위한 공간적 이웃하는 후보들을 도시하는 개념 도이다.

도 4b 는 AMVP (advanced motion vector prediction) 모드를 위한 공간적 이웃하는 후보들을 도시하는 개념 도이다.

도 5a 는 TMVP (temporal motion vector predictor) 후보를 도시하는 개념 도이다.

도 5b 는 TMVP를 위한 모션 벡터 스케일링을 도시하는 개념 도이다.

도 6 는 인접하지 않은 공간 병합 후보의 페칭 (fetching) 을 도시하는 개념 도이다.

도 7a 는 모션 벡터 예측에 사용되는 예시적인 공간적 그리고 시간적 후보를 예시하는 개념 도이다.

도 7b 는 공간적 후보를 위한 예시적인 방문 순서를 예시하는 개념 도이다.

도 7c 는 공간적 후보를 위한 다른 예시적인 방문 순서를 예시하는 개념 도이다.

도 8a 는 본 개시의 일례에 따른 예시적인 공간적 시간적 모션 벡터 예측자 패턴을 예시하는 개념 도이다.

도 8b 는 본 개시의 일례에 따른 예시적인 인버트된 공간적 시간적 모션 벡터 예측자 패턴을 예시하는 개념 도이다.

도 8c 는 본 개시의 일례에 따른 다른 예시적인 인버트된 공간적 시간적 모션 벡터 예측자 패턴을 예시하는 개념 도이다.

도 9 는 예시적인 코딩 방법을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0014] 상세한 설명

[0015] 일부 예에서, 비디오 코딩 (예를 들어, 비디오 인코딩 또는 비디오 디코딩) 은 인터 예측 및/또는 인트라 블록 카피 (IBC) 기술을 포함할 수도 있다. 인터 예측 또는 IBC 양자 모두에서, 비디오 인코더는 현재 블록을 위한 모션 벡터 (IBC의 경우, 모션 벡터는 블록 벡터일 수 있음) 에 기초하여 예측 블록을 결정하고, 예측 블록과 현재 블록 간의 잔차 정보 (예를 들어, 차이) 를 결정하고, 잔차 정보를 시그널링한다. 비디오 디코더는 잔차 정보를 수신한다. 또한, 비디오 디코더는 현재 블록을 위한 모션 벡터를 결정하고 모션 벡터에 기초하여 예측 블록을 결정한다. 비디오 디코더는 현재 블록을 재구성하기 위해 예측 블록에 잔차 정보를 더한다.

- [0016] 비디오 디코더가 현재 블록을 위한 모션 벡터를 결정하는 한 가지 방법은 모션 벡터 예측자 리스트에 기초한다. 비디오 인코더 및 비디오 디코더 양자 모두는 각각의 모션 벡터 예측자 리스트를 만들기 위해 유사한 프로세스를 이용하여 비디오 인코더에 의해 만들어진 모션 벡터 예측자 리스트와 비디오 디코더에 의해 만들어진 모션 벡터 예측자 리스트가 동일하다. 모션 벡터 예측자 리스트는 이전에 코딩된 블록, 이를테면 공간적으로 이웃하는 블록 (예를 들어, 현재 블록과 동일한 화상에서 현재 블록에 이웃하는 블록) 및 콜로케이트된 블록 (예를 들어, 다른 화상에서 특정 위치에 위치된 블록) 의 모션 벡터 정보를 포함한다. 일부 예에서, 모션 벡터 예측자 리스트는 인공적으로 생성된 모션 벡터 정보 (예를 들어, 이전에 코딩된 블록으로부터 유래되지 않은 모션 벡터 정보) 를 포함할 수도 있다.
- [0017] 비디오 인코더는 모션 벡터 예측자 리스트에서의 엔트리를 결정하고 엔트리를 나타내는 정보를 시그널링한다. 비디오 디코더는 엔트리에 기초하여 모션 벡터 예측자 리스트로부터 모션 벡터 정보를 결정하고 결정된 모션 벡터 정보에 기초하여 현재 블록을 위한 모션 벡터를 결정한다. 일례로서, 비디오 디코더는 (병합 모드에서와 같이) 결정된 모션 벡터 정보와 동일하게 현재 블록을 위한 모션 벡터를 설정할 수도 있다. 다른 예로서, 비디오 디코더는 비디오 인코더에 의해 시그널링된 모션 벡터 차이 (MVD) 를 결정된 모션 벡터 정보에 더하여 (AMVP (advanced motion vector prediction) 모드에서와 같이) 현재 블록을 위한 모션 벡터를 결정할 수도 있다.
- [0018] 전술한 바와 같이, 모션 벡터 예측자 리스트는 공간적으로 이웃하는 블록을 위한 모션 벡터 정보를 포함할 수도 있다. 일부 예에서, 공간적으로 이웃하는 블록의 모션 벡터 정보가 이용 가능하기 위해, 공간적으로 이웃하는 블록은 이전에 코딩 (예를 들어, 인코딩 또는 디코딩) 되었다. 이웃하는 블록이 아직 코딩되지 않은 경우, 비디오 인코더 및 비디오 디코더는 이 아직 코딩된 이웃하는 블록을 위한 모션 벡터 정보를 결정하기 위해 필요한 정보를 아직 결정하지 않았을 수 있으며 아직 갖지 않을 수 있다. 따라서, 일부 예에서, 모션 벡터 정보에 대해 체크될 수 있는 유일한 공간적으로 이웃하는 블록은 이전에 코딩된 블록들이다.
- [0019] 그러나, 어떤 블록이 이전에 코딩된 블록인지는 화상의 코딩 순서에 기초할 수도 있다. 가령, 비디오 코더 (예를 들어, 비디오 인코더 또는 비디오 디코더) 는 좌측에서 우측으로 그리고 상단에서 하단으로 제 1 화상을 코딩할 수도 있다. 좌측에서 우측으로 그리고 상단에서 하단으로 제 1 화상을 코딩하는 것은 비디오 코더가 제 1 화상에서 상단 좌측 블록부터 시작하여 제 1 화상 내에서 블록별 (block-by-block) 로 코딩하고 비디오 코더가 제 1 화상의 우측 경계에 도달할 때까지 오른쪽 방향으로 블록을 코딩하는 것을 말한다. 그런 다음 비디오 코더가 제 1 화상에서 상단 좌측 블록 아래의 블록으로 돌아가서 비디오 코더가 제 1 화상의 우측 경계에 도달할 때까지 블록별로 코딩한다. 비디오 코더는 비디오 코더가 제 1 화상의 하단 우측 블록을 코딩할 때까지 이러한 동작을 반복한다.
- [0020] 코딩 순서는 반드시 좌측에서 우측으로 그리고 상단에서 하단일 필요는 없다. 비디오 코더는 우측에서 좌측으로 그리고 상단에서 하단으로 제 2 화상을 코딩할 수도 있다. 우측에서 좌측으로 그리고 상단에서 하단으로 제 2 화상을 코딩하는 것은 비디오 코더가 제 2 화상에서 상단 우측 블록부터 시작하여 제 2 화상 내에서 블록별로 코딩하고 비디오 코더가 제 2 화상의 좌측 경계에 도달할 때까지 왼쪽 방향으로 블록을 코딩하는 것을 말한다. 그런 다음 비디오 코더가 제 2 화상에서 상단 우측 블록 아래의 블록으로 돌아가서 비디오 코더가 제 2 화상의 좌측 경계에 도달할 때까지 블록별로 코딩한다. 비디오 코더는 비디오 코더가 제 2 화상의 하단 좌측 블록을 코딩할 때까지 이러한 동작을 반복한다.
- [0021] 제 1 및 제 2 화상에 대한 상이한 코딩 순서로 인해, 제 2 화상에서 블록과 비교하여 제 1 화상에서의 블록에 대해 이전에 코딩된 상이한 공간적으로 이웃하는 블록이 있을 수도 있다. 예로서, 좌측에서 우측으로 코딩되는 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 제 1 블록에 대한 좌측 이웃하는 블록이 제 1 블록이 코딩되기 전에 이미 코딩되었을 수도 있으므로, 좌측 이웃하는 블록을 위한 모션 벡터 정보가 이용 가능할 수도 있다 (예를 들어, 이전에 결정됨). 또한, 제 1 블록의 경우, 제 1 블록에 대한 우측 이웃하는 블록이 아직 코딩되지 않았을 수도 있으므로, 우측 이웃하는 블록을 위한 모션 벡터 정보가 이용 가능하지 않을 수도 있다 (예를 들어, 아직 결정되지 않음).
- [0022] 우측에서 좌측으로의 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 1 화상에서의 제 1 블록의 반대 (opposite) 가 참일 수도 있다. 예를 들어, 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 제 2 블록에 대해 좌측 이웃하는 블록이 아직 코딩되지 않았을 수도 있으므로, 좌측 이웃하는 블록을 위한 모션 벡터 정보가 이용 가능하지 않을 수도 있다 (예를 들어, 아직 결정되지 않음). 또한, 제 2 블록에 대해, 제 2 블록에 대해 우측 이웃하는 블록은 제 2 블록이 코딩되기 전에 이미 코딩되었을 수도 있으므로, 우

측 이웃하는 블록을 위한 모션 벡터 정보가 이용 가능할 수도 있다 (예를 들어, 이전에 결정됨).

[0023] 어떤 공간적으로 이웃하는 블록이 코딩되었는지는 화상의 코딩 순서에 기초할 수 있기 때문에, 좌측에서 우측 코딩 순서를 갖는 제 1 화상에서의 제 1 블록을 위한 모션 벡터 예측자 리스트를 만드는 기술 및 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서 제 2 블록을 위한 모션 벡터 예측자 리스트를 만드는 기술은 상이할 수도 있다. 본 개시에 설명된 하나 이상의 예들에서, 모션 벡터 정보가 제 1 블록을 위한 모션 벡터 예측자 리스트를 형성하는 공간적으로 이웃하는 블록들의 세트의 위치에 모션 벡터 정보가 제 2 블록을 위한 모션 벡터 예측자 리스트를 형성하는 공간적으로 이웃하는 블록들의 세트의 위치에 대해 인버트 (invert) 될 수도 있다.

[0024] 가령, 좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 비디오 코더는, 상기 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리가 상기 제 1 블록에 대해 좌측 이웃하는 블록의 모션 벡터 정보에 기초하도록 제 1 모션 벡터 예측자 리스트를 만들 수도 있다. 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 비디오 코더는, 상기 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리가 상기 제 2 블록에 대해 우측 이웃하는 블록의 모션 벡터 정보에 기초하도록 제 2 모션 벡터 예측자 리스트를 만들 수도 있다. 제 1 블록에 대해 좌측 이웃하는 블록과 제 2 블록에 대해 우측 이웃하는 블록은 서로에 대해 인버트된 것으로 간주될 수도 있다 (예를 들어, 좌측 이웃하는 블록은 좌측에 있고 우측 이웃 블록은 우측에 있다).

[0025] 일부 예에서, 비디오 코더는 제 1 모션 벡터 예측자 리스트가 다음의 순서로 모션 벡터 정보를 갖도록 제 1 모션 벡터 예측자 리스트를 만들 수도 있다: 좌측 이웃하는 블록의 모션 벡터 정보, 상측 이웃하는 블록의 모션 벡터 정보, 우상측 이웃하는 블록의 모션 벡터 정보, 좌하측 이웃하는 블록의 모션 벡터 정보, 및 좌상측 이웃하는 블록의 모션 벡터 정보. 비디오 코더는 제 2 모션 벡터 예측자 리스트가 다음의 순서로 모션 벡터 정보를 갖도록 제 2 모션 벡터 예측자 리스트를 만들 수도 있다: 우측 이웃하는 블록의 모션 벡터 정보, 상측 이웃하는 블록의 모션 벡터 정보, 좌상측 이웃하는 블록의 모션 벡터 정보, 우하측 이웃하는 블록의 모션 벡터 정보, 및 우상측 이웃하는 블록의 모션 벡터 정보.

[0026] 위의 예에서, 제 1 및 제 2 모션 벡터 예측자 리스트는 공간적으로 이웃하는 블록을 위한 모션 벡터 정보가 모션 벡터 예측자 리스트(들)에 배열되는 특정 순서를 갖는다. 모션 벡터 정보가 배열되는 특정한 순서가 있지만, 비디오 코더는 공간적으로 이웃하는 블록들로부터 모션 벡터 정보에 병렬로 액세스하도록 구성될 수도 있다. 이것은 비디오 코더로 하여금 다수의 메모리 호출을 요구하기 보다는 하나의 메모리 호출로 (예를 들어, 일괄 요청 (batch request) 으로) 모션 벡터 정보에 액세스할 수 있게 한다. 그 다음, 비디오 코더는 위에서 설명된 예시적인 순서와 같은 특정 순서에 따라 모션 벡터 정보를 배열할 수도 있다.

[0027] 일부 기술에서, 특정 공간적으로 이웃하는 블록의 모션 벡터 정보가 모션 벡터 예측자 리스트에 포함되는지 여부는 다른 블록의 모션 벡터 정보의 이용 가능성에 기초한다. 달리 말하면, 특정 공간적으로 이웃하는 블록이 모션 벡터 예측자 리스트에 추가될지 여부는 조건부 (예를 들어, 다른 블록의 모션 벡터 정보의 이용 가능성에 기초하여 조건부) 이다. 공간적으로 이웃하는 것을 위한 모션 벡터 정보가 포함되는지 여부에 대한 조건이 존재하므로, 비디오 코더는 먼저 다른 블록의 모션 벡터 정보에 액세스하여 조건이 충족되는지 여부를 판단한 후, 조건이 충족되는지 여부에 기초하여, 공간적으로 이웃하는 블록의 모션 벡터 정보를 위한 다른 요청을 전송할 필요가 있을 수도 있다. 이러한 기술은 다수의 메모리 호출을 발생시켜, 블록을 코딩하는 데 걸리는 시간의 양을 증가시킬 수 있다.

[0028] 본 개시에서 설명된 하나 이상의 예들에서, 특정 공간적으로 이웃하는 블록의 모션 벡터 정보가 모션 벡터 예측자 리스트에 포함될지 여부는 다른 블록의 모션 벡터 정보가 이용 가능한지 여부에 대해 조건부가 아니다. 이러한 방식으로, 비디오 코더는 다수의 요청 대신 메모리에 대한 일괄 요청으로 공간적으로 이웃하는 블록의 모션 벡터 정보에 액세스 가능할 수도 있으며, 이는 블록을 코딩하는 데 걸리는 시간의 양을 줄이고 비디오 코더의 동작을 개선할 수 있다.

[0029] 전술한 바와 같이, 비디오 코더는 공간적으로 이웃하는 블록의 모션 벡터 정보 이외에, 하나 이상의 콜로케이트된 블록 (시간적으로 이웃하는 블록이라고도 함) 의 모션 벡터 정보를 포함할 수도 있다. 하나 이상의 콜로케이트된 블록은 코딩되는 블록을 포함하는 화상 외의 다른 화상 (참조 화상이라고 함) 에 있는 블록이다. 콜로케이트된 블록의 일례는 현재 화상에서 코딩되고 있는 현재 블록의 위치와 겹치는 참조 화상에서의 위치에 위치한 블록이다. 달리 말하면, 현재 블록은 현재 화상 내의 영역을 정의할 수도 있으며, 동일한 영역 내에 있지만 참조 화상에 있는 블록은 콜로케이트된 블록의 예이다. 이러한 콜로케이트된 블록의 예는 중앙 콜로케이트된 블록이라고 할 수도 있다.

- [0030] 콜로케이트된 블록의 다른 예는 현재 블록 아래에 위치하지만 참조 화상에 있는 (예를 들어, 현재 화상에서 현재 블록에 의해 정의된 영역 아래에 위치하지만 참조 화상에 있는) 블록이다. 이러한 콜로케이트된 블록의 예는 하단 콜로케이트된 블록이라고 할 수도 있다. 콜로케이트된 블록의 다른 예는 현재 블록 우측에 위치하지만 참조 화상에 있는 (예를 들어, 현재 화상에서 현재 블록에 의해 정의된 영역 우측에 위치하지만 참조 화상에 있는) 블록이다. 이러한 콜로케이트된 블록의 예는 우측 콜로케이트된 블록이라고 할 수도 있다. 콜로케이트된 블록의 다른 예는 현재 블록 좌측에 위치하지만 참조 화상에 있는 (예를 들어, 현재 화상에서 현재 블록에 의해 정의된 영역 좌측에 위치하지만 참조 화상에 있는) 블록이다. 이러한 콜로케이트된 블록의 예는 좌측 콜로케이트된 블록이라고 할 수도 있다.
- [0031] 하나 이상의 예들에서, 비디오 코더는 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 결정할 수도 있다. 중앙 콜로케이트된 블록을 위한 모션 벡터가 이용 가능한 경우, 비디오 코더는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가하고 다른 콜로케이트된 블록은 체크하지 않을 수도 있다. 중앙 콜로케이트된 블록을 위한 모션 벡터가 이용 가능하지 않은 경우, 비디오 코더는 하측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 판단할 수도 있다. 하측 콜로케이트된 블록을 위한 모션 벡터가 이용 가능한 경우, 비디오 코더는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가하고 다른 콜로케이트된 블록은 체크하지 않을 수도 있다.
- [0032] 하측 콜로케이트된 블록을 위한 모션 벡터가 이용 가능하지 않은 경우, 비디오 코더는 화상 코딩 순서에 기초하여 우측 콜로케이트된 블록 또는 좌측 콜로케이트된 블록 중 하나를 위한 모션 벡터 정보가 이용 가능한지 결정할 수도 있다. 일례로서, 코딩 순서가 좌측에서 우측이면, 비디오 코더는 우측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 결정할 수도 있다. 코딩 순서가 우측에서 좌측이면, 비디오 코더는 좌측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한지 결정할 수도 있다. 다른 예들에서, 비디오 코더는 역 (예를 들어, 좌측에서 우측 코딩 순서에 대한 좌측 콜로케이트된 블록을 위한 모션 벡터 정보 및 우측에서 좌측 코딩 순서에 대한 우측 콜로케이트된 블록을 위한 모션 벡터 정보) 을 수행할 수도 있다. 적용 가능한 바에 따라, 우측 또는 좌측 콜로케이트된 블록을 위한 모션 벡터가 이용 가능한 경우, 비디오 코더는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가하고 다른 콜로케이트된 블록은 체크하지 않을 수도 있다.
- [0033] 콜로케이트된 블록의 추가 예가 있을 수도 있다. 예를 들어, 콜로케이트된 블록은 현재 블록의 하단 그리고 우측 (예를 들어, 하단 우측 방향으로 대각선) 에 위치하지만 참조 화상에 있는 (예를 들어, 현재 화상에서 현재 블록에 의해 정의된 영역의 하단 우측에 있지만, 참조 화상에 있는) 하단 우측 블록일 수도 있다. 콜로케이트된 블록의 다른 예는, 현재 블록의 하단 그리고 좌측 (예를 들어, 하단 좌측 방향으로 대각선) 에 위치하지만 참조 화상에 있는 (예를 들어, 현재 화상에서 현재 블록에 의해 정의된 영역의 하단 좌측에 있지만, 참조 화상에 있는) 하단 좌측 블록일 수도 있다. 비디오 코더가 모션 벡터 정보를 위해 하단 우측 또는 하단 좌측 콜로케이트된 블록을 체크하는지 여부는 화상 코딩 순서 (예를 들어, 좌측에서 우측 코딩 순서에 대한 하단 우측 콜로케이트된 블록 및 우측에서 좌측 코딩 순서에 대한 하단 좌측 콜로케이트된 블록, 또는 그 반대) 에 기초할 수도 있다.
- [0034] 또한, 비디오 코더는 모션 벡터 예측자 리스트가 가득찰 때까지 (예를 들어, 모션 벡터 예측자 리스트에 있는 엔트리 수가 모션 벡터 예측자 리스트의 최대 크기와 같아질 때까지) 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있다. 일부 예들에서, 모션 벡터 예측자 리스트의 크기는 블록의 크기에 기초할 수도 있다. 예를 들어, 블록이 임계 크기보다 큰 크기를 갖는 경우, 모션 벡터 예측자 리스트의 최대 크기는 M 일 수도 있다 (예를 들어, 모션 벡터 예측자 리스트에 최대 M 개의 엔트리가 있을 수도 있다). 블록이 임계 크기보다 작은 크기를 갖는 경우, 모션 벡터 예측자 리스트의 최대 크기는 X 일 수도 있다 (예를 들어, 모션 벡터 예측자 리스트에 최대 X 개의 엔트리가 있을 수도 있다). X는 M보다 작을 수도 있다.
- [0035] 예로서, 블록이 $N \times 4$ 또는 $4 \times N$ 보다 큰 크기를 갖는 경우, 모션 벡터 예측자 리스트의 최대 크기는 6 일 수도 있다 (예를 들어, $M = 6$). 블록이 $N \times 4$ 또는 $4 \times N$ 보다 작은 크기를 갖는 경우, 모션 벡터 예측자 리스트의 최대 크기는 4 일 수도 있다 (예를 들어, $X = 4$). 임계 크기가 $N \times 4$ 또는 $4 \times N$ 인 경우, N 은 8보다 작거나 같을 수도 있다.
- [0036] 도 1 은 본 개시의 기술을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (100) 을 예시하는 블록도이다. 본 개시의 기법들은 일반적으로 비디오 데이터를 코딩 (인코딩 및/또는 디코딩) 하는 것과 관련된다. 일반적으로, 비디오 데이터는 비디오를 프로세싱하기 위한 임의의 데이터를 포함한다. 따라서, 비디오 데이터는 원시, 코딩되지 않은 비디오, 인코딩된 비디오, 디코딩된 (예를 들어, 재구성된) 비디오, 및 비

디오 메타데이터, 이를 태면 시그널링 데이터를 포함할 수도 있다.

[0037] 도 1 에 도시된 바와 같이, 시스템 (100) 은, 이 예에서 목적지 디바이스 (116) 에 의해 디코딩 및 디스플레이 될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (102) 를 포함한다. 특히, 소스 디바이스 (102) 는 비디오 데이터를 컴퓨터 관독가능 매체 (110) 를 통해 목적지 디바이스 (116) 에 제공한다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋탑 박스들, 전화기 핸드셋들, 예컨대 스마트폰들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 무선 통신을 위해 장비될 수도 있고, 따라서 무선 통신 디바이스들로 지칭될 수도 있다.

[0038] 도 1 의 예에 있어서, 소스 디바이스 (102) 는 비디오 소스 (104), 메모리 (106), 비디오 인코더 (200), 및 출력 인터페이스 (108) 를 포함한다. 목적지 디바이스 (116) 는 입력 인터페이스 (122), 비디오 디코더 (300), 메모리 (120), 및 디스플레이 디바이스 (118) 를 포함한다. 본 개시에 따르면, 소스 디바이스 (102) 의 비디오 인코더 (200) 및 목적지 디바이스 (116) 의 비디오 디코더 (300) 는 본 개시에 설명된 예시적인 기술 중 하나 이상을 사용하여 모션 벡터 예측자 리스트를 만드는 것과 같은 공간-시간적 모션 벡터 예측을 위한 기술을 적용하도록 구성될 수도 있다. 따라서, 소스 디바이스 (102) 는 비디오 인코딩 디바이스의 예를 나타내는 한편, 목적지 디바이스 (116) 는 비디오 디코딩 디바이스의 예를 나타낸다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102) 는 외부 카메라와 같은 외부 비디오 소스로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (116) 는 통합된 디스플레이 디바이스를 포함하기 보다는, 외부 디스플레이 디바이스와 인터페이스로 접속할 수도 있다.

[0039] 도 1에 도시된 바와 같은 시스템 (100) 은 하나의 예일뿐이다. 일반적으로, 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스는 공간-시간적 모션 벡터 예측을 위한 기술을 수행할 수도 있다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는, 소스 디바이스 (102) 가 목적지 디바이스 (116) 로의 송신을 위해 코딩된 비디오 데이터를 생성하는 그러한 코딩 디바이스들의 예들일 뿐이다. 본 개시는 "코딩" 디바이스를 데이터의 코딩 (인코딩 및/또는 디코딩) 을 수행하는 디바이스로서 지칭한다. 따라서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 코딩 디바이스들, 특히 각각 비디오 인코더 및 비디오 디코더의 예들을 나타낸다. 일부 예들에서, 디바이스들 (102, 116) 은 디바이스들 (102, 116) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (100) 은 예를 들어, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오 전화를 위해, 비디오 디바이스들 (102, 116) 간의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0040] 일반적으로, 비디오 소스 (104) 는 비디오 데이터 (즉, 원시, 코딩되지 않은 비디오 데이터) 의 소스를 나타내며 화상들을 위한 데이터를 인코딩하는 비디오 인코더 (200) 에 비디오 데이터의 순차적인 일련의 화상들 (또한 "프레임들" 로도 지칭됨) 을 제공한다. 소스 디바이스 (102) 의 비디오 소스 (104) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 원시 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (104) 는 컴퓨터 그래픽 기반 데이터를 소스 비디오로서, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합으로서 생성할 수도 있다. 각각의 경우에 있어서, 비디오 인코더 (200) 는 캡처된, 사전-캡처된, 또는 컴퓨터-생성된 비디오 데이터를 인코딩한다. 비디오 인코더 (200) 는 화상들을 수신된 순서 (때때로 "디스플레이 순서" 로 지칭됨) 로부터 코딩을 위한 코딩 순서로 재배열할 수도 있다. 비디오 인코더 (200) 는 인코딩된 비디오 데이터를 포함하는 비트스트림을 생성할 수도 있다. 소스 디바이스 (102) 는 그 후 예를 들어 목적지 디바이스 (116) 의 입력 인터페이스 (122) 에 의한 수신 및/또는 취출을 위해 인코딩된 비디오 데이터를 출력 인터페이스 (108) 를 통해 컴퓨터 관독 가능 매체 (110) 상으로 출력할 수도 있다.

[0041] 소스 디바이스 (102) 의 메모리 (106) 및 목적지 디바이스 (116) 의 메모리 (120) 는 범용 메모리들을 나타낸다. 일부 예에서, 메모리들 (106, 120) 은 원시 비디오 데이터, 예를 들어, 비디오 소스 (104) 로부터의 원시 비디오 및 비디오 디코더 (300) 로부터의 원시, 디코딩된 비디오 데이터를 저장할 수도 있다. 추가적으로 또는 대안적으로, 메모리들 (106, 120) 은 예를 들어, 각각 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해 실행가능한 소프트웨어 명령들을 저장할 수도 있다. 이 예에서는 비디오 인코더 (200) 및 비디오 디코더 (300) 와 별도로 도시되지만, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 또한 기능적으로 유

사하거나 또는 동등한 목적들을 위한 내부 메모리들을 포함할 수도 있음을 이해해야 한다. 또한, 메모리들 (106, 120) 은 예를 들어, 비디오 인코더 (200) 로부터 출력되고 비디오 디코더 (300) 에 입력되는 인코딩된 비디오 데이터를 저장할 수도 있다. 일부 예들에서, 메모리들 (106, 120) 의 부분들은 예를 들어, 원시, 디코딩된, 및/또는 인코딩된 비디오 데이터를 저장하기 위해 하나 이상의 비디오 버퍼들로서 할당될 수도 있다.

[0042] 컴퓨터 판독 가능 매체 (110) 는 인코딩된 비디오 데이터를 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로 전송할 수 있는 임의의 타입의 매체 또는 디바이스를 나타낼 수도 있다. 일례에서, 컴퓨터 판독 가능 매체 (110) 는, 소스 디바이스 (102) 로 하여금, 인코딩된 비디오 데이터를 직접 목적지 디바이스 (116) 에 실시간으로, 예를 들어 무선 주파수 네트워크 또는 컴퓨터 기반 네트워크를 통해 송신할 수 있게 하기 위한 통신 매체를 나타낸다. 무선 통신 프로토콜과 같은 통신 표준에 따라, 출력 인터페이스 (108) 는 인코딩된 비디오 데이터를 포함하는 송신 신호를 변조할 수도 있고, 입력 인터페이스 (122) 는 수신된 송신 신호를 변조할 수도 있다. 통신 매체는 무선 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷 기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로의 통신을 가능하게 하는 데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0043] 일부 예들에서, 소스 디바이스 (102) 는 출력 인터페이스 (108) 로부터 저장 디바이스 (112) 로 인코딩된 데이터를 출력할 수도 있다. 유사하게, 목적지 디바이스 (116) 는 입력 인터페이스 (122) 를 통해 저장 디바이스 (112) 로부터의 인코딩된 데이터에 액세스할 수도 있다. 저장 디바이스 (112) 는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 로컬 액세스 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다.

[0044] 일부 예들에서, 소스 디바이스 (102) 는 소스 디바이스 (102) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 (114) 또는 다른 중간 저장 디바이스에 인코딩된 비디오 데이터를 출력할 수도 있다. 목적지 디바이스 (116) 는 스트리밍 또는 다운로드를 통해 파일 서버 (114) 로부터의 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버 (114) 는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (116) 에 송신할 수 있는 임의의 타입의 서버 디바이스일 수도 있다. 파일 서버 (114) 는 (예를 들어, 웹사이트를 위한) 웹 서버, 파일 전송 프로토콜 (FTP) 서버, 콘텐츠 전달 네트워크 디바이스, 또는 NAS (network attached storage) 디바이스를 나타낼 수도 있다. 목적지 디바이스 (116) 는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 파일 서버 (114) 로부터 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 (114) 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한, 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 파일 서버 (114) 및 입력 인터페이스 (122) 는 스트리밍 송신 프로토콜, 다운로드 송신 프로토콜, 또는 이들의 조합에 따라 동작하도록 구성될 수도 있다.

[0045] 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 무선 송신기/수신기, 모뎀들, 유선 네트워킹 컴포넌트들 (예를 들어, 이더넷 카드들), 다양한 IEEE 802.11 표준들 중 임의의 것에 따라 동작하는 무선 통신 컴포넌트들, 또는 다른 물리적 컴포넌트들을 나타낼 수도 있다. 출력 인터페이스 (108) 및 입력 인터페이스 (122) 가 무선 컴포넌트를 포함하는 예들에 있어서, 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 4G, 4G-LTE (Long-Term Evolution), LTE 어드밴스드, 5G 등과 같은 셀룰러 통신 표준에 따라, 인코딩된 비디오 데이터와 같은 데이터를 전송하도록 구성될 수도 있다. 출력 인터페이스 (108) 가 무선 송신기를 포함하는 일부 예들에 있어서, 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 IEEE 802.11 사양, IEEE 802.15 사양 (예를 들어, ZigBee™), Bluetooth™ 표준 등과 같은 다른 무선 표준들에 따라, 인코딩된 비디오 데이터와 같은 데이터를 전송하도록 구성될 수도 있다. 일부 예들에서, 소스 디바이스 (102) 및/또는 목적지 디바이스 (116) 는 각각의 시스템-온-칩 (SoC) 디바이스들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102) 는 비디오 인코더 (200) 및/또는 출력 인터페이스 (108) 에 기인한 기능성을 수행하기 위한 SoC 디바이스를 포함할 수도 있고, 목적지 디바이스 (116) 는 비디오 디코더 (300) 및/또는 입력 인터페이스 (122) 에 기인한 기능성을 수행하기 위한 SoC 디바이스를 포함할 수도 있다.

[0046] 본 개시의 기술은 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 예컨대 DASH (dynamic adaptive streaming over HTTP), 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션

이션들과 같은 다양한 멀티미디어 애플리케이션들 중 임의의 것을 지원하여 비디오 코딩에 적용될 수도 있다.

[0047] 목적지 디바이스 (116) 의 입력 인터페이스 (122) 는 컴퓨터 판독 가능 매체 (110) (예를 들어, 저장 디바이스 (112), 파일 서버 (114) 등) 로부터 인코딩된 비디오 비트스트림을 수신한다. 컴퓨터 판독 가능 매체 (110) 로부터의 인코딩된 비디오 비트스트림은 비디오 블록들 또는 다른 코딩된 유닛들 (예를 들어, 슬라이스들, 화상들, 화상들의 그룹들, 시퀀스들 등) 의 프로세싱 및/또는 특징을 기술하는 값들을 갖는 구문 요소과 같은, 비디오 디코더 (300) 에 의해 또한 사용되는 비디오 인코더 (200) 에 의해 정의된 시그널링 정보를 포함할 수도 있다. 디스플레이 디바이스 (118) 는 디코딩된 비디오 데이터의 디코딩된 화상들을 사용자에게 디스플레이한다. 디스플레이 디바이스 (118) 는 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 어느 것을 나타낼 수도 있다.

[0048] 도 1 에 도시되지는 않았지만, 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각 오디오 인코더 및/또는 오디오 디코더와 통합될 수도 있고, 공통 데이터 스트림에서 오디오 및 비디오 양자 모두를 포함하는 멀티플렉싱된 스트림들을 핸들링하기 위해, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및/또는 소프트웨어를 포함할 수도 있다. 적용가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol; UDP) 과 같은 다른 프로토콜들을 준수할 수도 있다.

[0049] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각 다양한 적합한 인코더 및/또는 디코더 회로, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적 회로들 (ASIC들), 필드 프로그래밍 가능 게이트 어레이들 (FPGA들), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들 중 어느 것으로서 구현될 수도 있다. 기술이 부분적으로 소프트웨어에서 구현되는 경우, 디바이스는 적합한, 비일시적 컴퓨터 판독 가능 매체에 소프트웨어를 위한 명령들을 저장하고, 본 개시의 기술을 수행하기 위해 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽도 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다. 비디오 인코더 (200) 및/또는 비디오 디코더 (300) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 예컨대 셀룰러 전화기를 포함할 수도 있다.

[0050] 비디오 코딩 표준들은 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 ITU-T H.264 (ISO/IEC MPEG-4 AVC 으로도 알려짐) 을 포함하며, 그의 SVC (Scalable Video Coding) 및 MVC (Multi-view Video Coding) 확장판을 포함한다.

[0051] 또한, HEVC (High Efficiency Video Coding) 또는 ITU-T H.265 (G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no.12. pp. 1649-1668, December 2012 에 기재됨) 으로 명명되며, 그의 범위 확장, 멀티뷰 확장 (MV-HEVC) 및 스케일러블 확장 (SHVC) 을 포함하는, 비디오 코딩 표준이, ITU-T Video Coding Experts Group (VCEG) 및 ISO/IEC Motion Picture Experts Group (MPEG) 의 Joint Collaboration Team on 3D Video Coding Extension Development (JCT-3V) 뿐만 아니라 Joint Collaboration Team on Video Coding (JCT-VC) 에 의해 개발되었다. 이하 HEVC WD 로 지칭되는 HEVC 초안 사양은 http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip 에서 입수 가능하다. HEVC 의 국제 표준의 최종 드래프트 (FDIS) 의 최신 버전은 http://phenix.it-sudparis.eu/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip 에서 찾아볼 수 있다.

[0052] ITU-T VCEG (Q6/16) 및 ISO/IEC MPEG (JTC 1/SC 29/WG 11) 는, (스크린 콘텐츠 코딩 및 고-동적-범위 코딩을 위한 그의 현재 확장들 및 단기 확장들을 포함하는) 현재 HEVC 표준의 압축 능력을 현저하게 초과하는 압축 능력을 갖는 미래의 비디오 코딩 기술의 표준화에 대한 잠재적인 필요성을 지금 연구하고 있다. 그 그룹들은 이 영역에서 그들의 전문가들에 의해 제안된 압축 기술 설계들을 평가하기 위해 JVET (Joint Video Exploration Team) 로 알려진 공동 협력 노력에서 이 탐구 활동에 대해 함께 작업하고 있다. JVET 는 2015년 10월 19 - 21일간 처음 만났다. 레퍼런스 소프트웨어의 버전, 즉, JEM 7 (Joint Exploration Test Model 7) 는 https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-7.2/ 으로부터 다운로드될 수 있다.

Joint Exploration Test Model 7 (JEM-7) 의 알고리즘 설명은 J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, "Algorithm Description of Joint Exploration Test Model 7", JVET-G1001, July, 2017 에 기재되어 있다. VVC 표준의 최근 드래프트는 Bross 등의, "Versatile Video Coding (Draft 6)," Joint

Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Gothenburg, SE, 3-12 July 2019, JVET-02001-vE (이하 “VVC Draft 6”) 에 기재되어 있다. 하지만, 본 개시의 기법들은 어느 특정 코딩 표준에 한정되지 않는다.

[0053] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 고 효율 비디오 코딩 (HEVC) 으로 또한 지칭되는 ITU-T H.265 와 같은 비디오 코딩 표준 또는 그에 대한 확장들, 예컨대 멀티-뷰 및/또는 스케일러블 비디오 코딩 확장들에 따라 동작할 수도 있다. 대안적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 JEM (Joint Exploration Test Model) 및/또는 VVC 과 같은 다른 독점 또는 산업 표준들에 따라 동작할 수도 있다. 비디오 코딩 표준의 또 다른 예는 필수 비디오 코딩 (EVC) 표준이다. 하지만, 본 개시의 기법들은, 어느 특정 코딩 표준에 한정되지 않는다.

[0054] 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 화상들의 블록 기반 코딩을 수행할 수도 있다. 용어 "블록" 은 일반적으로 프로세싱될 (예를 들어, 인코딩될, 디코딩될, 또는 다르게는 인코딩 및/또는 디코딩 프로세스에서 사용될) 데이터를 포함하는 구조를 지칭한다. 예를 들어, 블록은 루미넌스 및/또는 크로미넌스 데이터의 샘플들의 2 차원 행렬을 포함할 수도 있다. 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 YUV (예를 들어, Y, Cb, Cr) 포맷으로 표현된 비디오 데이터를 코딩할 수도 있다. 즉, 화상의 샘플들을 위한 적색, 녹색, 및 청색 (RGB) 데이터를 코딩하기보다는, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 및 크로미넌스 성분들을 코딩할 수도 있고, 여기서 크로미넌스 성분은 적색 색조 및 청색 색조 크로미넌스 성분 양자 모두를 포함할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 인코딩 이전에 수신된 RGB 포맷화된 데이터를 YUV 표현으로 변환하고, 비디오 디코더 (300) 는 YUV 표현을 RGB 포맷으로 변환한다. 대안적으로, 전- 및 후-프로세싱 유닛들 (도시되지 않음) 이 이들 변환들을 수행할 수도 있다.

[0055] 본 개시는 일반적으로 화상의 데이터를 인코딩 또는 디코딩하는 프로세스를 포함하는 화상들의 코딩 (예를 들어, 인코딩 및 디코딩) 을 언급할 수도 있다. 유사하게, 본 개시는, 예를 들어, 예측 및/또는 잔차 코딩 과 같은, 블록들을 위한 데이터를 인코딩 또는 디코딩하는 프로세스를 포함하는 화상의 블록들의 코딩을 언급할 수도 있다. 인코딩된 비디오 비트스트림은 일반적으로 코딩 결정들 (예를 들어, 코딩 모드들) 및 화상들의 블록들로의 파티셔닝을 나타내는 구문 요소를 위한 일련의 값들을 포함한다. 따라서, 화상 또는 블록을 코딩하는 것에 대한 언급들은 일반적으로 화상 또는 블록을 형성하는 구문 요소를 위한 코딩 값들로서 이해되어야 한다.

[0056] HEVC 는 코딩 유닛들 (CU들), 예측 유닛들 (PU들), 및 변환 유닛들 (TU들) 을 포함한 다양한 블록들을 정의한다. HEVC 에 따르면, 비디오 코더 (예컨대 비디오 인코더 (200)) 는 쿼드트리 구조에 따라 코딩 트리 유닛 (CTU) 을 CU들로 파티셔닝한다. 즉, 비디오 코더는 CTU들 및 CU들을 4 개의 동일한, 오버랩하지 않는 정사각형들로 파티셔닝하고, 쿼드트리의 각각의 노드는 0 개 또는 4 개의 자식 노드들을 갖는다. 자식 노드들이 없는 노드들은 "리프 노드들" 로 지칭될 수도 있고, 그러한 리프 노드들의 CU들은 하나 이상의 PU들 및/또는 하나 이상의 TU들을 포함할 수도 있다. 비디오 코더는 PU들 및 TU들을 추가로 파티셔닝할 수도 있다. 예를 들어, HEVC 에서, 잔차 쿼드트리 (RQT) 는 TU들의 파티셔닝을 나타낸다. HEVC 에서, PU들은 인터-예측 데이터를 나타내는 한편, TU들은 잔차 데이터를 나타낸다. 인트라-예측되는 CU들은 인트라-모드 표시와 같은 인트라-예측 정보를 포함한다.

[0057] 다른 예로서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 JEM 및/또는 VVC 의 예에 따라 동작하도록 구성될 수도 있다. JEM/VVC 의 예에 따르면, 비디오 코더 (이를 테면 비디오 인코더 (200)) 는 화상을 복수의 코딩 트리 유닛들 (CTU들) 로 파티셔닝한다. 비디오 인코더 (200) 는 쿼드트리 이진 트리 (QTBT) 구조와 같은 트리 구조에 따라 CTU 를 파티셔닝할 수도 있다. JEM/VVC 의 예들의 QTBT 구조는 HEVC 의 CU들, PU들, 및 TU들 사이의 분리와 같은, 다중 파티션 타입들의 개념들을 제거한다. JEM/VVC 의 예들의 QTBT 구조는 2 개의 레벨들: 쿼드트리 파티셔닝에 따라 파티셔닝된 제 1 레벨, 및 이진 트리 파티셔닝에 따라 파티셔닝된 제 2 레벨을 포함한다. QTBT 구조의 루트 노드는 CTU 에 대응한다. 이진 트리들의 리프 노드들은 코딩 유닛들 (CU들) 에 대응한다.

[0058] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 및 크로미넌스 성분들의 각각을 나타내기 위해 단일 QTBT 구조를 사용할 수도 있는 한편, 다른 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 성분을 위한 하나의 QTBT 구조 및 양자의 크로미넌스 성분들을 위한 다른 QTBT 구조 (또는 각각의 크로미넌스 성분들을 위한 2 개의 QTBT 구조들) 와 같은 2 이상의 QTBT 구조들을 사용할 수도 있다.

[0059] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 HEVC 당 쿼드트리 파티셔닝, JEM/VVC의 예들에 따른 QTBT 파티

셔닝, 또는 다른 파티셔닝 구조들을 사용하도록 구성될 수도 있다. 설명의 목적을 위해, 본 개시의 기술의 설명은 QTBT 파티셔닝에 대하여 제시된다. 그러나, 본 개시의 기술은 또한, 쿼드트리 파티셔닝, 또는 다른 타입들의 파티셔닝도 역시 사용하도록 구성된 비디오 코더들에 적용될 수도 있음을 이해해야 한다.

[0060] 본 개시는 수직 및 수평 차원들의 측면에서 (CU 또는 다른 비디오 블록과 같은) 블록의 샘플 차원을 지칭하기 위해 "NxN" 및 "N 바이 (by) N", 예를 들어 16x16 샘플 또는 16 바이 16 샘플을 상호 교환적으로 사용할 수도 있다. 일반적으로, 16x16 CU 는, 수직 방향에서 16개 샘플들 ($y = 16$) 그리고 수평 방향에서 16개 샘플들 ($x = 16$) 을 가질 것이다. 마찬가지로, NxN CU 는 일반적으로 수직 방향에서 N개 샘플들 및 수평 방향에서 N개 샘플들을 갖고, 여기서 N 은 음이 아닌 정수 값을 나타낸다. CU 에서의 샘플들은 행들 및 열들로 배열 될 수도 있다. 더욱이, CU 들은 수직 방향에서와 동일한 수의 샘플들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들면, CU 들은 $N \times M$ 샘플들을 포함할 수도 있고, 여기서 M 은 N 과 반드시 동일한 것은 아니다.

[0061] 비디오 인코더 (200) 는 예측 및/또는 잔차 정보를 나타내는 CU 들을 위한 비디오 데이터, 및 다른 정보를 인코딩한다. 예측 정보는 CU 를 위한 예측 블록을 형성하기 위하여 CU 가 어떻게 예측될지를 표시한다. 잔차 정보는 일반적으로 인코딩 이전의 CU 의 샘플들과 예측 블록 사이의 샘플 별 (sample-by-sample) 차이들을 나타낸다.

[0062] CU 를 예측하기 위해, 비디오 인코더 (200) 는 일반적으로 인터 예측 또는 인트라 예측을 통해 CU 를 위한 예측 블록을 형성할 수도 있다. 인터-예측은 일반적으로 이전에 코딩된 화상의 데이터로부터 CU 를 예측하는 것을 지칭하는 반면, 인트라-예측은 일반적으로 동일한 화상의 이전에 코딩된 데이터로부터 CU 를 예측하는 것을 지칭한다. 인터-예측을 수행하기 위해, 비디오 인코더 (200) 는 하나 이상의 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 비디오 인코더 (200) 는 일반적으로, 예를 들어, CU 와 참조 블록 사이의 차이들의 측면에서, CU 에 근접하게 매칭하는 참조 블록을 식별하기 위해 모션 탐색을 수행할 수도 있다. 비디오 인코더 (200) 는 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of squared differences; SSD), 평균 절대차 (mean absolute difference; MAD), 평균 제곱차 (mean squared differences; MSD), 또는 참조 블록이 현재 CU 에 근접하게 매칭하는지 여부를 결정하기 위한 다른 그러한 차이 계산들을 사용하여 차이 메트릭을 계산할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 단방향 예측 또는 양방향 예측을 사용하여 현재 CU 를 예측할 수도 있다.

[0063] JEM 또는 VVC 은 또한, 인터-예측 모드로 고려될 수도 있는 아핀 모션 보상 모드를 제공한다. 아핀 모션 보상 모드에서, 비디오 인코더 (200) 는 줌 인 또는 아웃, 회전, 원근 모션 (perspective motion), 또는 다른 불규칙한 모션 타입들과 같은 비-병진 모션을 나타내는 2개 이상의 모션 벡터들을 결정할 수도 있다.

[0064] 인트라 예측을 수행하기 위해, 비디오 인코더 (200) 는 예측 블록을 생성하기 위해 인트라 예측 모드를 선택할 수도 있다. JEM 은 평면 모드 및 DC 모드 뿐만 아니라 다양한 방향성 모드들을 포함하여 67 개의 인트라-예측 모드들을 제공한다. 일반적으로, 비디오 인코더 (200) 는 현재 블록의 샘플들을 예측할 현재 블록 (예를 들어, CU 의 블록) 에 대한 이웃하는 샘플들을 기술하는 인트라-예측 모드를 선택한다. 그러한 샘플들은 일반적으로, 비디오 인코더 (200) 가 래스터 스캔 순서로 (좌측에서 우측, 상단에서 하단 코딩 순서 또는 우측에서 좌측, 상단에서 하단 코딩 순서로) CTU 들 및 CU 들을 코딩하는 것을 가정하여, 현재 블록과 동일한 화상에서 현재 블록의 상측, 좌상측, 또는 좌측에 있을 수도 있다.

[0065] 비디오 인코더 (200) 는 현재 블록을 위한 예측 모드를 나타내는 데이터를 인코딩한다. 예를 들어, 인터-예측 모드의 경우, 비디오 인코더 (200) 는 다양한 이용가능한 인터-예측 모드들 중 어느 것이 사용되는지를 나타내는 데이터 뿐만 아니라, 대응하는 모드를 위한 모션 정보를 인코딩할 수도 있다. 단방향 또는 양방향 인터-예측을 위해, 예를 들어, 비디오 인코더 (200) 는 어드밴스드 모션 벡터 예측 (AMVP) 또는 병합 모드를 사용하여 모션 벡터들을 인코딩할 수도 있다. 비디오 인코더 (200) 는 유사한 모드들을 사용하여 아핀 모션 보상 모드를 위한 모션 벡터들을 인코딩할 수도 있다.

[0066] 블록의 인트라-예측 또는 인터-예측과 같은 예측에 후속하여, 비디오 인코더 (200) 는 블록을 위한 잔차 데이터를 계산할 수도 있다. 잔차 블록과 같은 잔차 데이터는 대응하는 예측 모드를 사용하여 형성되는, 블록과 블록에 대한 예측 블록 사이의 샘플 별 차이를 나타낸다. 비디오 인코더 (200) 는 샘플 도메인 대신에 변환 도메인에서 변환된 데이터를 생성하기 위해, 잔차 블록에 하나 이상의 변환들을 적용할 수도 있다. 예를 들어, 비디오 인코더 (200) 는 이산 코사인 변환 (DCT), 정수 변환, 웨이브릿 변환, 또는 개념적으로 유사한 변환을 잔차 비디오 데이터에 적용할 수도 있다. 추가적으로, 비디오 인코더 (200) 는 MDNSST (mode-dependent non-separable secondary transform), 신호 의존적 변환, Karhunen-Loeve 변환 (KLT) 등과 같은 제 1 변환에

후속하는 2차 변환을 적용할 수도 있다. 비디오 인코더 (200) 는 하나 이상의 변환들의 적용에 후속하여 변환 계수들을 생성한다.

[0067] 위에 언급된 바와 같이, 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (200) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로, 변환 계수들이 그 계수들을 나타내는 데 사용되는 데이터의 양을 가능하게는 감소시키기 위해 양자화되어, 추가 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스를 수행함으로써, 비디오 인코더 (200) 는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, 비디오 인코더 (200) 는 양자화 동안 n-비트 값을 m-비트 값으로 라운딩 다운할 수도 있고, 여기서 n 은 m 보다 크다. 일부 예들에서, 양자화를 수행하기 위해, 비디오 인코더 (200) 는 양자화될 값의 비트단위 우측-시프트를 수행할 수도 있다.

[0068] 양자화에 후속하여, 비디오 인코더 (200) 는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 포함한 2 차원 행렬로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 벡터의 전방에 더 높은 에너지 (및 따라서 더 낮은 주파수) 계수들을 배치하고 벡터의 후방에 더 낮은 에너지 (및 따라서 더 높은 주파수) 변환 계수들을 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 사전 정의된 스캔 순서를 활용하여 양자화된 변환 계수들을 스캔함으로써 직렬화된 벡터를 생성하고, 그 후 벡터의 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (200) 는 적응적 스캔을 수행할 수도 있다. 1차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캔한 후, 비디오 인코더 (200) 는, 예를 들어, 컨텍스트-적응적 이진 산술 코딩 (CABAC) 에 따라, 1차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (200) 는 또한, 비디오 데이터를 디코딩하는 데 있어서 비디오 디코더 (300) 에 의한 사용을 위해 인코딩된 비디오 데이터와 연관된 메타데이터를 기술하는 구문 요소를 위한 값들을 엔트로피 인코딩할 수도 있다.

[0069] CABAC 을 수행하기 위해, 비디오 인코더 (200) 는 송신될 심볼에 컨텍스트 모델 내의 컨텍스트를 할당할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 제로 값인지 여부와 관련될 수도 있다. 확률 결정은 심볼에 할당된 컨텍스트에 기초할 수도 있다.

[0070] 비디오 인코더 (200) 는 구문 데이터, 예컨대 블록 기반 구문 데이터, 화상 기반 구문 데이터, 및 시퀀스 기반 구문 데이터를, 비디오 디코더 (300) 에, 예를 들어, 화상 헤더, 블록 헤더, 슬라이스 헤더에서, 또는 다른 구문 데이터, 예컨대 시퀀스 파라미터 세트 (SPS), 화상 파라미터 세트 (PPS), 또는 비디오 파라미터 세트 (VPS) 에서 추가로 생성할 수도 있다. 비디오 디코더 (300) 는 마찬가지로 대응하는 비디오 데이터를 디코딩하는 방법을 결정하기 위해 그러한 구문 데이터를 디코딩할 수도 있다.

[0071] 이러한 방식으로, 비디오 인코더 (200) 는 인코딩된 비디오 데이터, 예를 들어 화상의 블록들 (예를 들어, CU들) 로의 파티셔닝을 기술하는 구문 요소들 및 블록들을 위한 예측 및/또는 잔차 정보를 포함하는 비트스트림을 생성할 수도 있다. 궁극적으로, 비디오 디코더 (300) 는 비트스트림을 수신하고 인코딩된 비디오 데이터를 디코딩할 수도 있다.

[0072] 일반적으로, 비디오 디코더 (300) 는 비트스트림의 인코딩된 비디오 데이터를 디코딩하기 위해 비디오 인코더 (200) 에 의해 수행되는 것과 상반되는 프로세스를 수행한다. 예를 들어, 비디오 디코더 (300) 는 비디오 인코더 (200) 의 CABAC 인코딩 프로세스와 실질적으로 유사하지만, 상반되는 방식으로 CABAC 을 사용하여 비트스트림의 구문 요소들을 위한 값들을 디코딩할 수도 있다. 구문 요소들은 화상의 CTU들로의 파티셔닝 정보, 및 QTBT 구조와 같은 대응하는 파티션 구조에 따른 각각의 CTU 의 파티셔닝을 정의하여, CTU 의 CU들을 정의할 수도 있다. 구문 요소들은 비디오 데이터의 블록들 (예를 들어, CU들) 에 대한 예측 및 잔차 정보를 추가로 정의할 수도 있다.

[0073] 잔차 정보는 예를 들어 양자화된 변환 계수들에 의해 표현될 수도 있다. 비디오 디코더 (300) 는 블록을 위한 잔차 블록을 재생하기 위해 블록의 양자화된 변환 계수들을 역 양자화 및 역 변환할 수도 있다. 비디오 디코더 (300) 는 시그널링된 예측 모드 (인트라- 또는 인터-예측) 및 관련된 예측 정보 (예를 들어, 인터-예측을 위한 모션 정보) 를 사용하여 블록을 위한 예측 블록을 형성한다. 비디오 디코더 (300) 는 그 후 예측 블록과 잔차 블록을 (샘플 별 기반으로) 결합하여 원래 블록을 재생할 수도 있다. 비디오 디코더 (300) 는 블록의 경계들을 따라 시각적 아티팩트들을 감소시키기 위해 디블로킹 프로세스를 수행하는 것과 같은 추가적인 프로세싱을 수행할 수도 있다.

[0074] 본 개시의 기술들에 따르면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트를 만들도록 구성될 수도 있다. 전술한 바와 같이, 병합 모드 및 AMVP 모드의 경우, 비디오 디코더 (300) 는 모션

벡터 예측자 리스트에서의 모션 벡터 정보에 기초하여 디코딩되고 있는 현재 블록을 위한 모션 벡터를 결정할 수도 있다. 예를 들어, 비디오 인코더 (200) 는 모션 벡터 예측자 리스트에의 인덱스를 나타내는 정보를 시그널링할 수 있고, 비디오 디코더 (300) 는 인덱스에 기초하여 모션 벡터 예측자 리스트에 저장된 모션 벡터 정보를 추출할 수도 있다. 병합 모드의 경우, 비디오 디코더 (300) 는 현재 블록을 위한 모션 벡터를 추출된 모션 벡터 정보와 동일하게 설정할 수도 있다. AMVP 모드의 경우, 비디오 디코더 (300) 는 또한, 현재 블록을 위한 모션 벡터와 추출된 모션 벡터 정보 사이의 모션 벡터 차이 (MVD) 를 나타내는 정보를 수신할 수도 있다. 그러한 예에서, 비디오 디코더 (300) 는 현재 블록을 위한 모션 벡터를 결정하기 위해 추출된 모션 벡터 정보에 MVD 를 더할 수도 있다.

[0075] 병합 및 AMVP 모드에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 비디오 인코더 (200)에 의해 만들어진 모션 벡터 예측자 리스트 및 비디오 디코더 (300) 에 의해 만들어진 모션 벡터 예측자 리스트가 실질적으로 동일하도록 유사한 방식으로 모션 벡터 예측자 리스트를 만들도록 구성될 수도 있다. 모션 벡터 예측자 리스트를 만들기 위해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 공간적으로 이웃하는 블록 및 시간적으로 이웃하는 블록 (콜로케이트된 블록이라고도 함) 의 모션 벡터 정보를 추출할 수도 있다. 공간적으로 이웃하는 블록은 인코딩 또는 디코딩되고 있는 현재 블록을 포함하는 현재 화상에 위치한 블록을 말한다. 콜로케이트된 블록은 현재 화상과는 상이한 참조 화상에 위치한 블록을 말한다.

[0076] 그러나, 어느 공간적으로 이웃하는 블록을 비디오 인코더 (200) 및 비디오 디코더 (300) 가 모션 벡터 정보를 위해 체크하는지는 인코딩 또는 디코딩되고 있는 현재 블록이 위치한 현재 화상의 코딩 순서에 기초할 수도 있다. 이하에서 보다 상세히 설명하는 바와 같이, 현재 화상이 좌측에서 우측 코딩 순서를 갖는 예들에서, 체크되는 공간적으로 이웃하는 블록들의 예들이 도 8a 에 예시되어 있고, 블록 0 (802), 블록 1 (804), 블록 2 (806), 블록 3 (808) 및 블록 4 (810) 로 식별된다. 현재 화상이 우측에서 좌측 코딩 순서를 갖는 예들에서, 체크되는 공간적으로 이웃하는 블록들의 예들이 도 8b 에 예시되어 있고, 블록 0 (818), 블록 1 (820), 블록 2 (822), 블록 3 (824) 및 블록 4 (826) 로 식별된다. 이러한 방식으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 화상의 코딩 순서에 기초하여 서로 다른 공간적으로 이웃하는 블록들을 체크하도록 구성될 수도 있다.

[0077] 공간적으로 이웃하는 블록에 추가하여, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 콜로케이트된 블록의 모션 벡터 정보를 포함할 수도 있다. 일부 예들에서, 현재 화상의 좌측에서 우측 코딩 순서에 대해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 도 8a의 중앙 콜로케이트된 블록 5 (812) 의 모션 벡터 정보가 이용 가능한지 결정하고, 이용 가능하다면, 모션 벡터 예측자 리스트에 중앙 콜로케이트된 블록 5 (812) 의 모션 벡터 정보를 포함하고 어느 다른 콜로케이트된 블록의 모션 벡터 정보를 체크 또는 포함하지 않을 수도 있다.

[0078] 중앙 콜로케이트된 블록 5 (812) 의 모션 벡터 정보가 이용가능하지 않다면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 도 8a의 하단 콜로케이트된 블록 6 (814) 의 모션 벡터 정보가 이용 가능한지 결정하고, 이용 가능하다면, 모션 벡터 예측자 리스트에 하단 콜로케이트된 블록 6 (814) 의 모션 벡터 정보를 포함하고 어느 다른 콜로케이트된 블록의 모션 벡터 정보를 체크 또는 포함하지 않을 수도 있다. 예시된 바와 같이, 하단 콜로케이트된 블록 6 (814) 은 현재 블록 (800) 의 우측에 가장 가깝고 하단인 블록일 수도 있다 (예를 들어, 블록 6 (814) 은 도 8a의 현재 블록 (800) 아래에 있고 블록 6 (814) 의 우측 경계는 도 8a의 현재 블록 (800) 의 우측 경계와 동일하다). 즉, 하단 콜로케이트된 블록의 우측 경계는 현재 블록 (800) 의 우측 경계와 현재 블록 (800) 을 포함하는 화상 외의 화상에서 동일한 위치에 있다. 블록 6 (814) 에 대한 좌표는 (xColBot, yColBot) 일 수도 있으며, 여기서 $xColBot = xCb + cbWidth - 1$, 그리고 $yColBot = yCb + cbHeight$, 여기서 (xCb, yCb) 는 도 8a 의 현재 블록 (800) 에 대한 좌표이고 cbWidth는 도 8a의 현재 블록 (800) 의 폭이고, cbHeight는 도 8a의 현재 블록 (800) 의 높이이다.

[0079] 하단 콜로케이트된 블록 6 (814) 의 모션 벡터 정보가 이용가능하지 않다면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 도 8a의 우측 콜로케이트된 블록 7 (816) 의 모션 벡터 정보가 이용 가능한지 결정하고, 이용 가능하다면, 모션 벡터 예측자 리스트에 우측 콜로케이트된 블록 7 (816) 의 모션 벡터 정보를 포함하고 어느 다른 콜로케이트된 블록의 모션 벡터 정보를 체크 또는 포함하지 않을 수도 있다. 예시된 바와 같이, 우측 콜로케이트된 블록 7 (816) 은 도 8a 의 현재 블록 (800) 의 하단에 가장 가깝고 우측인 블록일 수도 있다 (예를 들어, 블록 7 (816) 은 도 8a의 현재 블록 (800) 의 우측에 있고 블록 7 (816) 의 하단 경계는 도 8a의 현재 블록 (800) 의 하단 경계와 동일하다). 블록 7 (816) 에 대한 좌표는 (xColBr, yColBr) 일 수도 있으며, 여기서 $xColBr = xCb + cbWidth$, 그리고 $yColBr = yCb + cbHeight - 1$, 여기서 (xCb, yCb) 는 도 8a 의 현재 블록 (800) 에 대한 좌표이고 cbWidth는 도 8a의 현재 블록 (800) 의 폭이고, cbHeight는 도 8a의 현재 블록 (800) 의 높이이다.

의 높이이다.

- [0080] 일부 예들에서, 현재 화상의 우측에서 좌측 코딩 순서에 대해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 도 8b의 중앙 콜로케이트된 블록 5 (828) 의 모션 벡터 정보가 이용 가능한지 결정하고, 이용 가능하다면, 모션 벡터 예측자 리스트에 중앙 콜로케이트된 블록 5 (828) 의 모션 벡터 정보를 포함하고 어느 다른 콜로케이트된 블록의 모션 벡터 정보를 체크 또는 포함하지 않을 수도 있다.
- [0081] 중앙 콜로케이트된 블록 5 (828) 의 모션 벡터 정보가 이용가능하지 않다면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 도 8b의 하단 콜로케이트된 블록 6 (830) 의 모션 벡터 정보가 이용 가능한지 결정하고, 이용 가능하다면, 모션 벡터 예측자 리스트에 하단 콜로케이트된 블록 6 (830) 의 모션 벡터 정보를 포함하고 어느 다른 콜로케이트된 블록의 모션 벡터 정보를 체크 또는 포함하지 않을 수도 있다. 예시된 바와 같이, 하단 콜로케이트된 블록 6 (830) 은 현재 블록 (800) 의 좌측에 가장 가깝고 하단인 블록일 수도 있다 (예를 들어, 블록 6 (830) 은 도 8b의 현재 블록 (800) 아래에 있고 블록 6 (830) 의 좌측 경계는 도 8b의 현재 블록 (800) 의 좌측 경계와 동일하다). 즉, 하단 콜로케이트된 블록의 좌측 경계는 현재 블록 (800) 의 좌측 경계와 현재 블록 (800) 을 포함하는 화상 외의 화상에서 동일한 위치에 있다. 블록 6 (830) 에 대한 좌표는 (xColBot, yColBot) 일 수도 있으며, 여기서 $xColBot = xCb$, 그리고 $yColBot = yCb + cbHeight$, 여기서 (xCb, yCb) 는 도 8b 의 현재 블록 (800) 에 대한 좌표이고 cbWidth는 도 8b의 현재 블록 (800) 의 폭이고, cbHeight 는 도 8b의 현재 블록 (800) 의 높이이다.
- [0082] 하단 콜로케이트된 블록 6 (830) 의 모션 벡터 정보가 이용가능하지 않다면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 도 8b의 좌측 콜로케이트된 블록 7 (832) 의 모션 벡터 정보가 이용 가능한지 결정하고, 이용 가능하다면, 모션 벡터 예측자 리스트에 좌측 콜로케이트된 블록 7 (832) 의 모션 벡터 정보를 포함하고 어느 다른 콜로케이트된 블록의 모션 벡터 정보를 체크 또는 포함하지 않을 수도 있다. 예시된 바와 같이, 좌측 콜로케이트된 블록 7 (832) 은 도 8b 의 현재 블록 (800) 의 하단에 가장 가깝고 좌측인 블록일 수도 있다 (예를 들어, 블록 7 (832) 은 도 8b의 현재 블록 (800) 의 좌측에 있고 블록 7 (832) 의 하단 경계는 도 8b의 현재 블록 (800) 의 하단 경계와 동일하다). 블록 7 (832)에 대한 좌표는 (xColBr, yColBr) 일 수도 있으며, 여기서 $xColBr = xCb - 1$, 그리고 $yColBr = yCb + cbHeight - 1$, 여기서 (xCb, yCb) 는 도 8b 의 현재 블록 (800) 에 대한 좌표이고 cbWidth는 도 8b의 현재 블록 (800) 의 폭이고, cbHeight는 도 8b의 현재 블록 (800) 의 높이이다.
- [0083] 일부 예에서, 도 8a의 블록 (814, 816) 및 도 8b의 블록 (830, 832) 의 콜로케이트된 위치는 입도 8x8 블록으로 정의될 수도 있다. 예를 들어, 현재 CU (예를 들어, 도 8a 또는 도 8b의 현재 블록 (800)) 이 64x64 인 경우, 하단, 좌측 또는 우측에서 이용 가능한 다수의 8x8 블록이 있을 수도 있다. 일부 예들에서, 코너의 8x8 블록 (예를 들어, 블록 (814, 816, 830, 832)) 만이 모션 벡터를 제공하는 데 사용될 수도 있다.
- [0084] 더욱이, 일부 예들에서, 모션 벡터 예측자 리스트의 크기는 현재의 블록의 크기에 기초할 수도 있다. 일례로서, 현재 블록의 크기가 임계치보다 더 크면, 모션 벡터 예측자 리스트의 최대 엔트리 수는 현재 블록의 크기가 임계치보다 작은 경우보다 더 클 수도 있다. 예를 들어, 현재 블록의 크기가 Nx4 또는 4xN보다 더 큰 경우 (여기서 N은 8보다 작거나 같음), 모션 벡터 예측자 리스트의 최대 크기는 6 일 수도 있다. 현재 블록의 크기가 Nx4 또는 4xN보다 더 작은 경우 (여기서 N은 8보다 작거나 같음), 모션 벡터 예측자 리스트의 최대 크기는 4 일 수도 있다. 위의 값은 이해를 돕기 위한 하나의 예일 뿐이며 제한적인 것으로 간주되서는 안된다.
- [0085] 본 개시는 일반적으로 구문 요소들과 같은, 소정의 정보를 "시그널링" 하는 것을 언급할 수도 있다. 용어 "시그널링" 은 인코딩된 비디오 데이터를 디코딩하는 데 사용되는 구문 요소 및/또는 다른 데이터의 값들의 통신을 지칭할 수도 있다. 즉, 비디오 인코더 (200) 는 비트스트림에서 구문 요소를 위한 값들을 시그널링할 수도 있다. 일반적으로, 시그널링은 비트스트림에서 값을 생성하는 것을 지칭한다. 상기 언급된 바와 같이, 소스 디바이스 (102) 는 목적지 디바이스 (116) 에 의한 추후 취출을 위해 저장 디바이스 (112) 에 구문 요소를 저장할 때 발생할 수도 있는 바와 같이, 비실시간으로 또는 실질적으로 실시간으로 비트스트림을 목적지 디바이스 (116) 로 전송할 수도 있다.
- [0086] 비디오 인코더 (200) 및 비디오 디코더 (300) 의 예가 도 2 및 도 3 과 관련하여 더 상세히 예시되고 설명된다. 도 2 및 도 3 을 설명하기 전에, 이하는 도 4a, 4b, 5a, 5b, 및 6 과 관련하여, 모션 벡터 예측과 같은 비디오 코딩 프로세스의 일부 추가 세부 사항을 설명한다.
- [0087] 보다 상세히 설명된 바와 같이, 도 7a 내지 7c는 비디오 인코더 (200) 및 비디오 디코더 (300) 가 모션 벡터 예

측자 리스트를 만들기 위해 평가하는 공간적으로 그리고 시간적으로 이웃하는 블록의 일부 예를 예시한다. 도 7a 내지 도 7c 에 예시된 기술을 사용하여 모션 벡터 예측자 리스트를 만드는 데 어떤 문제가 있을 수도 있다. 보다 상세히 설명된 바와 같이, 도 8a 내지 도 8c는 도 7a 내지 7c와 관련하여 설명된 모션 벡터 예측자 리스트를 만드는 문제를 해결할 수도 있는 모션 벡터 예측자 리스트를 만드는 일부 추가 예를 예시한다.

- [0088] 다음은 HEVC 에서의 CU 구조 및 모션 벡터 예측을 검토한다 HEVC 에서, 슬라이스의 최대 코딩 유닛은 코딩 트리 블록 (CTB) 또는 코딩 트리 유닛 (CTU) 으로 불린다. CTB 는 쿼드-트리를 포함하고, 그 노드들은 코딩 유닛들이다.
- [0089] CTB 의 크기는 (기술적으로 8x8 CTB 크기가 지원될 수 있지만) HEVC 메인 프로파일에서 16x16 내지 64x64 의 범위일 수 있다. 코딩 유닛 (CU) 은 CTB 의 동일한 사이즈일 수 있고 8x8 만큼 작을 수 있다. 각각의 코딩 유닛은 하나의 모드로 코딩, 이를테면 인터 코딩 또는 인트라 코딩된다. 인터 코딩되는 것은 또한, 인터 예측 코딩 또는 인터 예측되는 것이라고도 할 수도 있다. 인트라 코딩되는 것은 또한, 인트라 예측 코딩 또는 인트라 예측되는 것이라고도 할 수도 있다.
- [0090] CU 가 인터 코딩될 경우, CU 는 2 또는 4개의 예측 유닛들 (PU들) 로 추가로 파티셔닝될 수도 있거나, 또는 추가의 파티션이 적용되지 않을 경우 하나의 PU 일 수도 있다. 2 개의 PU 들이 하나의 CU 에 존재할 때, 그 PU 들은 CU 의 절반 크기의 직사각형 또는 1/4 또는 3/4 크기를 갖는 2 개의 직사각형 크기일 수 있다. CU 가 인터 코딩될 때, 각각의 PU 는 인터 예측 모드로 도출되는 한 세트의 모션 정보를 갖는다.
- [0091] 이하는 모션 벡터 예측을 검토한다. HEVC 표준에서, 예측 유닛 (PU) 에 대해 각각 병합 (스킵이 병합의 특수한 경우로 간주됨) 및 AMVP (advanced motion vector prediction) 모드로 명명된 2 개의 인터 예측 모드가 있다. AMVP 모드 또는 병합 모드 중 어느 하나에서, 모션 벡터 (MV) 후보 리스트 (MV 후보 리스트라고도 불림) 가 다수의 모션 벡터 예측자에 대해 유지된다. 모션 벡터 예측자 리스트는 모션 벡터 예측자 리스트로 지칭될 수도 있다. 현재의 PU 의 모션 벡터 (들) 뿐만 아니라, 병합 모드에서의 참조 인덱스들도 모션 벡터 예측자 리스트로부터 하나의 후보를 취함으로써 생성된다. 모션 벡터 예측자 리스트는 병합 모드에 대해 5 개까지의 후보들과, AMVP 모드에 대해 2 개의 후보들만을 포함할 수도 있다. 병합 후보는 모션 정보의 세트, 예컨대, 참조 화상 리스트들 (리스트 0 및 리스트 1) 및 참조 인덱스들 양자 모두에 대응하는 모션 벡터들을 포함할 수도 있다. 병합 후보가 병합 인덱스에 의해 식별되면, 현재 블록의 예측에 사용되는 참조 화상들 뿐만 아니라 연관된 모션 벡터들이 결정된다. 즉, 병합 인덱스에 의해 모션 벡터 예측자 리스트에서 식별된 모션 벡터 및 참조 화상은 현재 블록의 모션 벡터 및 참조 화상과 동일하게 설정된다.
- [0092] 한편, AMVP 모드 하에서, 리스트 0 또는 리스트 1 중 어느 하나로부터의 각각의 잠재적 예측 방향에 대해, AMVP 후보가 모션 벡터만을 포함하므로 모션 벡터 예측자 리스트에 MV 예측자 (MVP) 인덱스와 함께, 참조 인덱스가 명시적으로 시그널링될 필요가 있다. AMVP 모드에서, 예측된 모션 벡터는 (예를 들어, 위에서 설명된 모션 벡터 차이 (MVD) 에 기초하여) 더 정제 (refine) 될 수도 있다. 양자의 모드를 위한 후보들은 동일한 공간적 및 시간적 이웃하는 블록들로부터 유사하게 도출된다.
- [0093] 이하는 모션 벡터 정보가 모션 벡터 예측자 (모션 벡터 예측자 후보라고도 함) 로서 형성되는 공간적 이웃하는 블록을 검토한다. 일부 예들에서, 공간적 MV 후보들 (예를 들어, 공간적 모션 벡터 예측자) 은, 특정한 PU (PU₀) (434 및 438) 에 대해, 각각, 도 4a 및 도 4b 에 도시된 이웃하는 블록들로부터 도출되지만, 블록으로부터 후보를 생성하는 방법은 병합 및 AMVP 모드에 대해 상이하다. 도 4a 는 병합 모드를 위한 공간적 이웃하는 후보들을 도시하는 개념 도이다. 도 4b 는 AMVP 모드를 위한 공간적 이웃하는 후보들을 도시하는 개념 도이다.
- [0094] 병합 모드에서는, 일부 예들에서, 5 개까지의 공간적 MV 후보들이 도 4a 에 도시된 순서로 도출될 수 있고, 순서는 다음과 같다: 도 4a 에서 도시된 바와 같이, 좌측 (0, A1), 상측 (1, B1), 우상측 (2, B0), 좌하측 (3, A0), 및 좌상측 (4, B2). 예를 들어, PU0 (434) 의 경우, 블록 A1은 0으로 식별되고 PU0 (434) 의 좌측에 있고, 블록 B1 은 1로 식별되고 PU0 (434) 상측에 있고, 블록 B0 은 2로 식별되고 PU0 (434) 의 우상측에 있고 PU1 (436) 상측에 있으며, 블록 A0 은 3 으로 식별되고 PU0 (434) 의 좌하측에 있으며 블록 B2 는 4 로 식별되고 PU0 (434) 의 좌상측에 있다.
- [0095] AMVP 모드에서, 일부 예들에서, 이웃하는 블록들은 2 개의 그룹들로 분할된다: 도 4b 에 도시된 바와 같이, 각각 PU0 (438) 의 좌하측 및 좌측에 있는 블록 0 과 1 을 포함하는 좌측 그룹, 및 PU01 (438) 의 우상측, 상측, 및 좌상측에 있는 블록들 2, 3, 및 4 를 포함하는 상측 그룹. 블록 2 는 PU1 (440) 상측에 있다. 각각

의 그룹에 대해, 시그널링된 참조 인덱스에 의해 표시되는 동일한 참조 화상을 지칭하는 이웃하는 블록에서 잠재적인 후보는 그룹의 최종 후보를 형성하도록 선정될 가장 높은 우선순위를 갖는다. 모든 이웃하는 블록들은 동일한 참조 화상을 가리키는 모션 벡터를 포함하지 않을 수 있다. 그러므로, 그러한 후보가 구해질 수 없으면, 첫번째 이용가능한 후보는 최종 후보를 형성하기 위하여 스케일링될 수도 있으며, 이렇게 하여 시간적 거리 차이가 보상될 수 있다.

[0096] 이하는 도 5a 내지 도 5b 로 시간적 모션 벡터 예측을 검토한다. 도 5a 는 TMVP (temporal motion vector predictor) 후보를 도시하는 개념 도이다. 도 5b 는 TMVP를 위한 모션 벡터 스케일링을 도시하는 개념 도이다.

[0097] 일부 예에서, TMVP 후보가, 활성화되고 이용 가능한 경우, 공간 모션 벡터 후보 (예를 들어, 공간적으로 이웃하는 블록의 모션 벡터 정보) 후에 MV 후보 리스트 (예를 들어, 모션 벡터 예측자 리스트) 에 추가된다. TMVP 후보를 위한 모션 벡터 도출의 프로세스는 병합 및 AMVP 모드들 양자 모두에 대해 동일하지만, 병합 모드에서 TMVP 후보를 위한 타겟 참조 인덱스는 0 으로 설정된다.

[0098] 일부 예들에서, TMVP 후보 도출을 위한 1차 블록 위치는 도 5a 에서 블록 T (540) 로서 나타낸 바와 같이 콜로케이트된 PU 외부의 하단 우측 블록이다. 블록 T (540) 의 위치는 공간적 이웃하는 후보들을 생성하는 데 사용된 상측 및 좌측 블록들에 대한 편향 (bias) 을 보상할 수도 있다. 그러나, 블록 T (540) 가 현재 CTB 행의 외부에 위치되거나 또는 모션 정보가 블록 T (540) 에 대해 이용가능하지 않으면, 블록 T (540) 는 PU 의 중앙 블록 (541) 으로 치환된다.

[0099] 도 5b 에서, 현재 화상 (550) 에서의 TMVP 후보를 위한 모션 벡터 (548) 는, 슬라이스 레벨에서 표시된, 콜로케이트된 화상 (546) 의 콜로케이트된 PU 로부터 도출된다. 콜로케이트된 PU 를 위한 모션 벡터 (MV) 는 콜로케이트된 MV (542) 로 불린다. TMVP 후보 모션 벡터를 도출하기 위해, 콜로케이트된 MV (542) 는 도 5b 에 나타낸 바와 같이 시간적 거리 차이를 보상하기 위해 스케일링될 수도 있다. 예를 들어, 콜로케이트된 화상 (546) 과 콜로케이트된 참조 화상 (544) 사이의 시간적 차이 및 현재 화상 (550) 과 현재 참조 화상 (552) 사이의 시간적 차이는 모션 벡터 (548) 를 생성하기 위해 콜로케이트된 MV (542) 를 스케일링하는 데 사용된다.

[0100] 다음은 HEVC 에서의 모션 예측의 일부 다른 양태들을 검토한다 병합 모드 및 AMVP 모드의 여러 양태들이 아래에 설명된다.

[0101] **모션 벡터 스케일링:** 모션 벡터의 값은 그 프리젠테이션 시간에서의 화상의 거리에 비례할 수 있다. 모션 벡터는 2 개의 화상, 즉 참조 화상, 및 모션 벡터를 포함하는 화상 (즉, 포함하는 화상) 를 연관시킨다. 모션 벡터가 다른 모션 벡터를 예측하는 데 활용될 때, 포함하는 화상과 참조 화상의 거리는 화상 순서 카운트 (POC) 값들에 기초하여 계산된다.

[0102] 예측될 모션 벡터에 대해, 그 연관된 포함하는 화상 및 참조 화상 양자는 상이할 수 있다. 따라서, 새로운 거리 (POC 기반) 가 계산되고, 이들 두 POC 거리들에 기초하여 모션 벡터가 스케일링된다. 공간적 이웃하는 후보에 대해, 2개의 모션 벡터들을 위한 포함하는 화상들은 동일한 한편, 참조 화상들은 상이하다. HEVC 에서, 모션 벡터 스케일링은 공간적 및 시간적 이웃 후보들을 위해 TMVP 및 AMVP 양자 모두에 적용한다.

[0103] **인공 모션 벡터 후보 생성:** 모션 벡터 후보 리스트가 완료되지 않은 경우 (예를 들어, 미리 결정된 수보다 후보들이 더 적은 경우), 인공 모션 벡터 예측자가 생성되어 모션 벡터 예측자 리스트가 모든 후보를 가질 때까지 모션 벡터 예측자 리스트의 끝에 삽입된다. 병합 모드에 있어서, 2가지 타입의 인공 MV 후보들: 즉, B-슬라이스만을 위해 도출된 조합된 후보 및 B-슬라이스만을 위해 도출된 조합된 후보가 모션 벡터 예측자 리스트를 채우기에 충분한 인공 후보들을 제공하지 않는 경우 AMVP만을 위해 사용되는 제로 후보가 있다.

[0104] 이미 모션 벡터 예측자 리스트에 있고 필요한 모션 정보를 갖는 후보들의 각각의 쌍에 대해, 양방향 조합된 모션 벡터 후보들은 리스트 0 에서의 화상을 참조하는 제 1 후보의 모션 벡터 및 리스트 1 에서의 화상을 참조하는 제 2 후보의 모션 벡터의 조합에 의해 도출된다.

[0105] **후보 삽입을 위한 프루닝 (pruning) 프로세스:** 상이한 블록들로부터의 후보들이 우연히 동일할 수도 있으며, 이는 병합/AMVP 모션 벡터 후보 리스트의 효율을 감소시킨다. 이러한 문제를 해결하기 위해 프루닝 프로세스가 적용된다. 프루닝 프로세스는 동일한 후보를 삽입하는 것을 피하기 위해 현재 모션 벡터 후보 리스트에서 하나의 후보를 다른 후보들과 비교한다. 복잡성을 줄이기 위해, 제한된 수의 프루닝 프로세스만이 적용되어 각각의 잠재적 후보를 리스트에 있는 모든 다른 기존 후보들과 비교하는 것을 피한다.

- [0106] VVC 의 개발에는 향상된 모션 벡터 예측이 포함된다. 예를 들어, 현재 블록을 위한 모션 벡터 예측 또는 병합 예측의 모션 벡터 예측자 리스트를 도출하거나 정제하는 여러 인터 코딩 톨이 제안되었다. 몇 가지 예가 아래에 설명된다.
- [0107] 다음은 L. Zhang, K. Zhang, H. Liu, Y. Wang, P. Zhao, and D. Hong, “CE4-related: History-based Motion Vector Prediction”, JVET-K0104, July 2018 에 기재된 바와 같이 HMVP (history-based motion prediction) 를 설명한다. HMVP 는 비디오 인코더 (200) 및 비디오 디코더 (300) 가 바로 인접한 인과적 이웃하는 모션 필드에 있는 그러한 MV들 (예를 들어, 바로 공간적으로 이웃하는 블록의 MV는 바로 인접한 인과적 이웃하는 모션 필드에 있는 MV의 예이다) 에 더하여, 과거 (past) 로부터 디코딩된 MV 의 리스트로부터 각각의 블록에 대해 MV 예측자를 결정한다. HMVP는 비디오 인코더 (200) 및 비디오 디코더 (300) 가 HMVP 후보로서 이전에 디코딩된 모션 벡터에 대한 테이블을 만드는 것을 포함한다.
- [0108] 예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 인코딩/디코딩 프로세스 동안 다수의 HMVP 후보를 갖는 테이블을 만든다. 테이블 구성에는 테이블에 HMVP 후보를 추가하고 테이블로부터 HMVP 후보를 제거하는 것이 포함될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 인코딩 또는 디코딩을 위해 새로운 슬라이스에 직면할 때 테이블을 비우도록 (예를 들어, HMVP 후보들을 제거하도록) 구성될 수도 있다. 인터 코딩된 블록이 있을 때마다, 비디오 인코더 (200) 및 비디오 디코더 (300)는 새로운 HMVP 후보로서 FIFO (first-in-first-out) 방식으로 테이블에 연관된 모션 정보를 삽입하도록 구성될 수도 있다. 다음으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 제약 FIFO 규칙을 적용하도록 구성될 수도 있다. 일부 기술에서, HMVP 후보를 테이블에 삽입할 때, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 테이블에 동일한 HMVP 후보가 있는지 여부를 결정하기 위해 먼저 중복 검사 (예를 들어, 프루닝) 를 적용하도록 구성될 수도 있다. 발견되면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 테이블로부터 그 특정 HMVP 후보 및 그 후보가 이동된 후의 모든 HMVP 후보를 제거하도록 구성될 수도 있다.
- [0109] 비디오 인코더 (200) 및 비디오 디코더 (300)는 병합 후보 리스트 구성 프로세스에서 HMVP 후보를 사용하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 TMVP 후보 다음에 테이블에서 마지막 엔트리로부터 첫번째 엔트리까지 모든 HMVP 후보를 삽입하도록 구성될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 HMVP 후보에 대해 프루닝을 적용하도록 구성될 수도 있다. 이용 가능한 병합 후보의 총 수가 시그널링되거나 미리 결정된 최대 허용 병합 후보 수에 도달하면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 병합 후보 리스트 만들기 프로세스를 종료하도록 구성될 수도 있다. 병합 후보 리스트 만들기는 모션 벡터 예측자 리스트를 만드는 한 가지 예이다.
- [0110] 유사하게, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 AMVP 후보 리스트를 만들기 위해 AMVP 후보 리스트 만들기 프로세스에서 HMVP 후보를 사용하도록 구성될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 TMVP 후보 다음에 테이블에서 마지막 K 개 HMVP 후보들의 모션 벡터를 삽입하도록 구성될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 AMVP 후보 리스트를 만들기 위해 AMVP 타겟 참조 화상과 동일한 참조 화상을 가진 HMVP 후보들만을 사용하도록 구성될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 HMVP 후보에 대해 프루닝을 적용하도록 구성될 수도 있다. AMVP 후보 리스트 구성은 모션 벡터 예측자 리스트를 만드는 다른 예이다.
- [0111] 다음은 비인접 공간적 병합 후보를 설명한다. R. Yu, 등의, “CE 4-2.1: Adding non-adjacent spatial merge candidates,” Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 11th Meeting: Ljubljana, SI, 10-18 July 2018 (이하 “JVET-K0228”) 에 기재된 비인접 공간 병합 후보 만들기는 두 개의 비인접 이웃하는 위치로부터 새로운 공간적 후보를 도출하는 것을 포함한다. 예를 들어, 도 6 에 도시된 바와 같이, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 현재 블록 (648) 위의 가장 가까운 비인접 블록 (650) 으로부터 및/또는 현재 블록 (648) 의 좌측에 있는 가장 가까운 비인접 블록 (652) 으로부터 새로운 공간적 후보를 도출할 수도 있다. 가령, 도 6은 이웃하는 블록들 A0, A1, B2, B1 및 B0 을 갖는 현재 블록 (648) 을 예시한다. 일부 예들에서, 비인접 공간적 병합 후보들은 현재 블록 (648) 에 공간적으로 이웃하지 않은 도 6의 블록 HN (652) 및 블록 VN (650) 의 모션 정보를 포함한다. 즉, 도 6은 HMVP 후보 히스토리 테이블을 만들기 위해 모션 벡터 정보가 사용될 수도 있는 블록들의 예를 예시한다.
- [0112] 블록 (650 및 652) 은 현재 블록 (648) 에 대한 1 CTU 의 최대 거리 내에서 제한된다. 비인접 후보들의 폐쇄 프로세스는 수직 방향으로 이전에 디코딩된 블록을 추적하는 것으로 시작된다. 수직 역 추적은 인터 블록에 직면되거나 역추적된 거리가 1 CTU 크기에 달하면 중지된다.

- [0113] 그런 다음, 폐칭 프로세스는 수평 방향으로 이전에 디코딩된 블록을 추적한다. 수평 폐칭 프로세스를 중지하는 기준은 성공적으로 폐칭되는 수직 비인접 후보가 있는지 여부에 의존한다. 수직 비인접 후보가 폐칭되지 않는 경우, 수평 폐칭 프로세스는 인터 코딩된 블록에 직면되거나 역추적된 거리가 하나의 CTU 크기 임계치를 초과하는 경우 중지된다. 폐칭된 수직 비인접 후보가 있는 경우, 수평 폐칭 프로세스는 수직 비인접 후보로부터 상이한 MV 를 포함하는 인터 코딩된 블록에 직면되거나 역추적된 거리가 하나의 CTU 크기 임계치를 초과하는 경우 중지된다. 폐칭된 비인접 이웃하는 후보는 병합 후보 리스트에서 TMVP 후보 전에 추가된다.
- [0114] 전술한 바와 같이, 병합 모드 및 AMVP 모드의 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트이라고도 하는 모션 벡터 (MV) 후보 리스트를 만들 수도 있다. 모션 벡터 예측자 리스트는 공간적으로 이웃하는 블록 (예를 들어, 현재 블록에 이웃하고 현재 블록과 동일한 화상에 있는 블록), 시간적으로 이웃하는 블록 (예를 들어, 현재 블록에 이웃하지만 현재 블록과는 상이한 화상에 있는 블록) 의 모션 벡터 정보, 그리고 일부 경우에, HMVP 후보들의 모션 벡터 정보 (예를 들어, HMVP 후보들을 형성하는 이전에 디코딩된 모션 벡터들을 위한 테이블로부터의 엔트리) 를 포함한다. 시간적으로 이웃하는 블록은 또한, 콜로케이트된 블록이라고도 한다.
- [0115] 따라서, 모션 벡터 예측자 리스트를 만들기 위해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 이웃하는 블록들 (예를 들어, 공간적으로 그리고 시간적으로 이웃하는 블록들) 을 평가하여 이웃하는 블록들이 모션 벡터 정보를 가지고 있는지 여부를 결정할 수도 있다. (예를 들어, 이웃하는 블록이 인터 코딩되기 때문에) 이웃하는 블록이 모션 벡터 정보를 갖는 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트에 이웃하는 블록의 모션 벡터 정보를 포함할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 HMVP 후보들을 위해 최근 인코딩되거나 디코딩된 블록들의 모션 벡터 정보의 별도 테이블을 유지할 수도 있다.
- [0116] 비디오 인코더 (200) 및 비디오 디코더 (300) 가 이웃하는 블록이 모션 벡터 정보를 갖는지 여부를 결정하기 위해 평가 (예를 들어, 체크) 할 수 있는 복수의 이웃하는 블록이 있을 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 가 평가할 필요가 있을 수도 있는 이웃하는 블록의 수를 제한하기 위해, 다음은 공간 및 시간적 모션 벡터 예측을 위한 일부 예시적인 설계 (예를 들어, 평가되는 블록에 공간적으로 그리고 시간적으로 이웃하는 블록의 예) 를 설명한다.
- [0117] 병합 모드의 일부 구현에서 (일부 용어에서 이것은 스킵 또는 직접 모드로 불릴 수 있음), 공간 및 시간적 모션 벡터 예측자 (MVP) 후보는 모션 벡터 예측자 리스트를 포플레이트 (populate) 하기 위해 주어진 순서로 방문 (예를 들어, 체크) 될 수 있다. SMVP (공간적 MVP)는 현재 블록의 모션 벡터를 위한 모션 벡터 예측자로 사용될 수 있는 공간적으로 이웃하는 블록의 모션 벡터일 수도 있다. TMVP (시간적 MVP) 는, 전술한 바와 같이, 현재 블록의 모션 벡터를 위한 모션 벡터 예측자로서 사용될 수 있는 시간적으로 이웃하는 블록 (예를 들어, 콜로케이트된 블록) 을 위한 모션 벡터일 수도 있다.
- [0118] 일부 기술에서, SMVP 및 TMVP 를 결정하기 위해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 특정 순서로 이웃하는 블록 (예를 들어, 공간적으로 또는 시간적으로 이웃하는 블록) 을 평가할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 가 이웃하는 블록을 평가할 수도 있는 순서는 또한 방문 순서라고 지칭될 수도 있다.
- [0119] 도 7a 는 모션 벡터 예측에 사용되는 예시적인 공간적 그리고 시간적 후보를 예시하는 개념 도이다. 도 7b 는 공간적 후보를 위한 예시적인 방문 순서를 예시하는 개념 도이다. 도 7c 는 공간적 후보를 위한 다른 예시적인 방문 순서를 예시하는 개념 도이다.
- [0120] 도 7a는 MVP 후보로 활용되는 공간적 이웃의 예를 도시한다. 도 7a 에서, 현재 블록 (700) 의 공간적 이웃 (예를 들어, 공간적으로 이웃하는 블록) 은 블록 A (702), 블록 B (704), (블록 C (706), 블록 A1 (710A) | 블록 B1 (710B)), 블록 A0 (714) 및 블록 B2 (712) 를 포함한다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 2 단계 프로세스로 이웃하는 블록을 평가할 수도 있다. 공간 후보 (예를 들어, 공간적으로 이웃하는 블록) 가 체크되는 순서 (즉, 0-5 의 숫자로 표시된 방문 순서) 는 도 7b에 도시되어 있다.
- [0121] 도 7a의 그룹 1 후보에는 다음이 포함된다:
- [0122] a. (HEVC 표기법에서 블록 B0 (708) 과 콜로케이트되고) 해시 음영으로 표시되는 블록 A (702), 블록 B (704), 블록 C (706), 및

- [0123] b. 블록 A1 (710A) 또는 블록 B1 (710B), 여기서 블록 A1 (710A) 또는 블록 B1 (710B) 은 블록 C (706) 에서의 모션 벡터 예측자가 이용 가능한지 여부에 따라 포함된다.
- [0124] 그룹 2 후보에는 다음이 포함된다:
- [0125] a. 블록 A0 (714) 및 블록 B2 (712)
- [0126] 도 7a에서, 블록 C (706)는 블록 B0 (708) 과 콜로케이트된다. 가령, 블록 C (706) 및 블록 B0 (708) 은 동일한 블록일 수도 있지만, HEVC 와 VVC 간의 이름 변경으로 인한 두 개의 다른 이름이 도 7a에 주어진다. HEVC 표준은 블록 B0 (708) 으로 예시된 블록을 블록 "B0" 으로 지칭한다. VVC 표준은 블록 B0 (708) 로 예시된 블록을 블록 "C" 로 지칭한다. 따라서, 블록 C (706) 는 블록 B0 (708) 과 콜로케이트되고 현재 블록 (700) 과 동일한 화상에 있다.
- [0127] 그룹 1 및 그룹 2 에 따른 평가의 예로서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 그룹 1에 있는 블록을 평가 (예를 들어, 체크) 할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 A (702) 를 평가하고 블록 A (702) 를 위한 모션 벡터 정보가 있는지 여부를 결정할 수도 있다. (예를 들어, 블록 A (702) 가 모션 벡터로 인터 코딩되었기 때문에) 블록 A (702) 를 위한 모션 벡터 정보가 있는 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트에 블록 A (702) 의 모션 벡터 정보를 추가할 수도 있다. 블록 A (702) 가 모션 벡터 정보를 갖는지 여부에 관계 없이, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 B (704) 를 평가하고 블록 B (704) 를 위한 모션 정보가 있는지 여부를 결정할 수도 있다. (예를 들어, 블록 B (704) 가 모션 벡터로 인터 코딩되었기 때문에) 블록 B (704) 를 위한 모션 벡터 정보가 있는 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트에 블록 B (704) 의 모션 벡터 정보를 추가할 수도 있다. 블록 B (704) 가 모션 벡터 정보를 갖는지 여부에 관계 없이, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 C (706) 를 평가하고 블록 C (706) 를 위한 모션 정보가 있는지 여부를 결정할 수도 있다. (예를 들어, 블록 C (706) 가 모션 벡터로 인터 코딩되었기 때문에) 블록 C (706) 를 위한 모션 벡터 정보가 있는 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트에 블록 C (706) 의 모션 벡터 정보를 추가할 수도 있다.
- [0128] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 C (706) 를 위한 모션 벡터 정보가 있었는지에 기초하여 블록 A1 (710A) 또는 블록 B1 (710B) 중 하나를 평가할 수도 있다. 예를 들어, 블록 C (706) 를 위한 모션 벡터 정보가 있는 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 A1 (710A) 또는 블록 B1 (710B) 중 하나 (예를 들어, 블록 A1 (710A)) 를 평가할 수도 있다. 그러나, (예를 들어, 블록 C (706) 가 인트라 코딩되었기 때문에) 블록 C (706) 를 위한 모션 벡터 정보가 없는 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 A1 (710A) 또는 블록 B1 (710B) 중 다른 하나 (예를 들어, 블록 B1 (710B)) 를 평가할 수도 있다.
- [0129] 블록 A1 (710A) 또는 블록 B1 (710B) 중 하나를 평가한 후, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 전술한 바와 같이 그룹 2의 블록 (예를 들어, 블록 B2 (712) 및 블록 A0 (714)) 을 평가하고 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있다. 그 후, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 시간적 이웃들 (예를 들어, 시간적으로 이웃하는 블록들 또는 콜로케이트된 블록들) 을 평가할 수도 있다.
- [0130] 도 7a에서, 현재 블록 (700) 의 시간적 이웃 (예를 들어, 시간적으로 이웃하는 블록) 은 블록 C0 (716) 및 하나 이상의 블록 (718) 을 포함한다. 도 7a에 도시된 바와 같이, MVP 후보로서 활용되는 시간적으로 콜로케이트된 이웃들은 현재 블록의 중앙에 콜로케이트된 블록 (블록 C0 (716) 으로 표시됨) 및 현재 블록 외부의 가장 하단 우측 위치에 있는 (점선 윤곽선을 갖는) 블록 (718A) 에서 나온 것이다. 블록 C0 (716) 및 블록 (718) 은 현재 블록 (700) 을 포함하는 현재 화상과는 상이한 참조 화상일 수도 있다. 그룹 3 후보 블록 (점선 윤곽선으로 표시됨) 에는 다음이 포함된다:
- [0131] a. 블록 C0 (716), 블록 H (718A)
- [0132] b. H 위치 (예를 들어, 블록 H (718A)) 가 콜로케이트된 화상 외부에 있는 것으로 밝혀지면, 하나 이상의 폴백 (fall back) H 위치 (예를 들어, 블록 H (718B-718D)) 가 대신 사용될 수도 있다.
- [0133] 예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 C0 (716) 를 위한 모션 벡터 정보가 있는지 여부를 결정하기 위해 블록 C0 (716) 를 평가하고, 모션 벡터 예측자 리스트에 이용 가능하다면 블록 C0 (716) 를 위한 모션 벡터 정보를 추가할 수도 있다. 유사하게, 비디오 인코더 (200) 및 비디오 디코더 (300) 는

블록 H (718A) 을 위한 모션 벡터 정보가 있는지 여부를 결정하기 위해 블록 H (718A) 를 평가하고, 모션 벡터 예측자 리스트에 이용 가능하다면 블록 H (718A) 를 위한 모션 벡터 정보를 추가할 수도 있다. 블록 H (718A) 가 이용 가능하지 않은 경우 (예를 들어, 그것이 화상 외부에 있기 때문에), 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 H (718B-718D) 중 하나를 평가할 수도 있다.

[0134] 모션 벡터 예측자 리스트의 크기는 고정될 수도 있다 (즉, 모션 벡터 예측자 리스트에 고정된 수의 엔트리가 있을 수도 있다). 따라서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트의 최대 크기에 도달하지 않는 한 이웃하는 블록을 평가 (예를 들어, 체크) 하고 이웃하는 블록의 모션 벡터 예측자 리스트에 모션 벡터 정보를 추가할 수도 있다. 또한, 시간적으로 이웃하는 (예를 들어, 콜로케이트된) 블록 CO (716) 그리고 블록 H (718A-718D) 중 하나를 평가한 후에 모션 벡터 예측자 리스트의 크기에 도달하지 않으면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트에 HMVP 후보를 추가하거나 또는 인공 모션 벡터 후보를 모션 벡터 예측자 리스트에 추가할 수도 있다.

[0135] 전술한 바와 같이, 비디오 인코더 (200) 및 비디오 디코더 (300) 가 이웃하는 블록을 평가하는 순서가 있을 수도 있다. 도 7b는 예시적인 순서를 예시한다. 가령, 도 7b 에서, "0" 은 블록 A (702) 를 나타내며, 비디오 인코더 (200) 및 비디오 디코더 (300) 가 평가할 수도 있는 첫번째 블록이다. "1" 은 블록 B (704) 를 지칭하고, 비디오 인코더 (200) 및 비디오 디코더 (300) 가 평가할 수도 있는 두번째 블록인 등이다.

[0136] 도 7b에서 "3" 은 블록 A1 (710A) 및 블록 B1 (710B) 를 나타낸다. 이것은 비디오 인코더 (200) 및 비디오 디코더 (300) 가 블록 C (706) ("2" 로 식별됨) 를 위한 모션 벡터 정보가 있는지에 기초하여 블록 A1 (710A) 또는 블록 B1 (710B) 중 하나를 평가할 수도 있음을 표시하는 것이다.

[0137] 일부 예시적인 구현에서, 사용된 블록 파티셔닝 및 코딩 순서에 따라, 역 공간 MVP 후보 순서가 도 7c 에 도시된 바와 같이 사용될 수 있고, 여기서 체크 순서는 0 에서 5 로 진행된다. 예를 들어, 도 7a 및 도 7b 와 부합하게, 비디오 인코더 (200) 및 비디오 디코더 (300) 가 각각 화상에 있는 블록에 대해 좌측에서 우측 그리고 상단에서 하단 코딩 순서를 사용하여 인코딩 또는 디코딩하는 경우, 블록 A (702), 블록 B (704), 블록 C (706), 블록 A1 (710A), 블록 B1 (710B), 블록 B2 (712) 및 블록 A0 (714) 는 현재 블록 (700) 이 인코딩 또는 디코딩되기 전에 인코딩되거나 디코딩될 것이다. 블록 A (702), 블록 B (704), 블록 C (706), 블록 A1 (710A), 블록 B1 (710B), 블록 B2 (712) 및 블록 A0 (714) 는 현재 블록 (700) 이 인코딩 또는 디코딩되기 전에 인코딩 또는 디코딩되기 때문에, 현재 블록 (700) 을 인코딩 또는 디코딩하기 위해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 A (702), 블록 B (704), 블록 C (706), 블록 A1 (710A), 블록 B1 (710B), 블록 B2 (712), 및 블록 A0 (714) 을 위한 모션 벡터 정보를 평가 가능할 수도 있다. 가령, 블록 A (702) 가 현재 블록 (700) 이전에 좌측에서 우측, 상단에서 하단 코딩 순서로 인코딩 또는 디코딩되었기 때문에, 블록 A (702) 가 인터 코딩된 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 A (702) 를 위한 모션 벡터 정보에 액세스가능할 수도 있다.

[0138] 그러나, 일부 예들에서, 코딩 순서는 좌측에서 우측 및 상단에서 하단이 아닐 수도 있다. 오히려, 코딩 순서는 우측에서 좌측 및 상단에서 하단일 수도 있다. 그러한 예들에서, 도 7a 및 도 7b 에 예시된 이웃하는 블록들 중 일부는 현재 블록 (700) 이전에 인코딩 또는 디코딩되지 않았을 수도 있다. 따라서, 이웃하는 블록이 인터 코딩될 수 있더라도, 이웃하는 블록에 대한 모션 벡터 정보는, 도 7a 및 도 7b 에 있는 이웃하는 블록 중 일부가 우측에서 좌측 코딩 순서로 인코딩 또는 디코딩되지 않았을 것이기 때문에, 아직 이용할 수 없다.

[0139] 도 7c는 평가되는 공간적으로 이웃하는 블록의 예 및 공간적으로 이웃하는 블록이 화상에서 블록의 우측에서 좌측 그리고 상단에서 하단 코딩 순서로 평가되는 순서를 예시한다. 예를 들어, 도 7c에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 (720, 722, 724), 블록 (726A 또는 726B) 중 하나, 블록 (728 및 730) 를 평가할 수도 있다. 예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 는, 모션 벡터 정보가 있는지 여부를 결정하기 위해, 블록 (720) ("0"으로 식별됨) 으로부터 시작하여 블록 (722) ("1" 로 식별됨) 그리고 그 다음 블록 (724) ("2" 로 식별됨) 을 평가하는 것을 시작할 수도 있다.

[0140] 도 7a 및 도 7b 의 블록 A1 (710A) 및 블록 B1 (710B) 과 유사하게, 도 7c 에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 (724) 를 위한 모션 벡터 정보가 있는지 여부에 기초하여 (예를 들어, 블록 (724) 가 인터 코딩되었는지 여부에 기초하여) 블록 (726A 또는 726B) (둘 다 "3" 으로 식별됨) 중 하나를 평가할 수도 있다. 예시된 바와 같이, 블록 (726A 또는 726B) 중 하나 이후, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 (728) 을 평가한 다음 블록 (730) 을 평가할 수도 있다.

- [0141] 이상은 비디오 인코더 (200) 및 비디오 디코더 (300) 가 모션 벡터 예측자 리스트에 포함될 모션 벡터 정보가 있는지 여부를 결정하기 위해 이웃하는 블록들을 평가할 수도 있는 예시적인 순서를 설명한다. 일부 예들에서, 하나씩 평가하기 보다는, 비디오 인코더 (200) 및 비디오 디코더 (300) 가 복수의 이웃하는 블록들을 병렬로 평가하는 것이 가능할 수도 있다. 이러한 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300)는 모션 벡터 예측자 리스트에서 모션 벡터 정보를 도 7a 내지 도 7c 에 예시된 순서로 순서화 (order) 할 수도 있다.
- [0142] 일례로서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 A (702) 및 블록 B (704) 를 차례로 평가하기 보다는 병렬로 평가할 수도 있다. 이 예에서, 블록 A (702) 및 블록 B (704) 양자 모두가 모션 벡터 정보를 갖는 경우, 블록 A (702) 가 평가 순서에서 블록 B (704) 이전에 있기 때문에, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 A (702) 를 위한 모션 벡터 정보가 블록 B (704) 를 위한 모션 벡터 정보 이전에 있도록 모션 벡터 예측자 리스트를 만들 수도 있다.
- [0143] 그러나, 도 7a 내지 도 7c에 예시된 예에서, 모션 벡터 정보를 병렬로 평가하는 데 제한이 있을 수도 있으며, 이는 계산 효율성에 영향을 미칠 수 있다. 예를 들어, 블록 A1 (710A) 또는 블록 B1 (710B) 이 평가될 것인지 여부는 블록 C (706) 를 위한 모션 벡터 정보가 있는지 여부에 기초한다. 이러한 예에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 직렬적으로 블록 C (706) 를 평가하고 다음으로 블록 C (706) 를 평가한 결과에 기초하여 (예를 들어, 블록 C (706) 를 위한 모션 벡터 정보가 있는지 여부에 기초하여) 블록 A1 (710A) 또는 블록 B1 (710B) 중 하나를 평가할 수도 있다. 도 7c 에서, 병렬적 평가보다는, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 직렬적으로 블록 (724) 를 평가하고 다음으로 블록 (724) 를 평가한 결과에 기초하여 (예를 들어, 블록 (724) 를 위한 모션 벡터 정보가 있는지 여부에 기초하여) 블록 (726A) 또는 블록 (726B) 중 하나를 평가할 수도 있다.
- [0144] 유사하게, 비디오 인코더 (200) 및 비디오 디코더 (300) 가 블록 H (718B-718D) 중 하나를 평가하는지 여부는 블록 (718A) 이 (예를 들어, 화상 외부에 있는 것이 아니라) 이용 가능한지 여부에 기초할 수도 있다. 시간적으로 이웃하는 블록은 도 7c에 도시되어 있지 않지만, 도 7a 에서 블록 H (718A-718D) 에 적용되는 바처럼 시간적으로 이웃하는 블록에 대해 동일하게 적용될 수도 있다.
- [0145] 달리 말하면, 도 7a 내지 도 7c에 설명된 패턴과 같은 MVP 후보 설계는, 특정 MVP 후보의 이용 가능성에 기초하여 (예를 들어, 블록 C (706) 에 대한 모션 벡터 정보의 이용 가능성 (C 후보 이용 가능성이라고도 함) 에 기초하여), 블록 파티셔닝 유형 또는 MVP 후보의 공간적 위치, 이를테면 도 7a에 도시된 H 후보들의 그룹 (예를 들어, 블록 H (718A-718D)) 으로부터 선택하는 것에 기초하여, 조건부 MVP 폐칭 (A1|B1) (예를 들어, 블록 A1 (710A) 또는 블록 B1 (710B) 중 어느 일방) 을 사용한다. 이러한 조건부 폐칭은 계산 복잡성을 증가시키고 MVP 의 특정 구현을 위한 병렬화 또는 메모리 제약에 영향을 미칠 수도 있다.
- [0146] 일부 예에서 많은 수의 종속성 및 조건을 필요로 할 수도 있는 모션 벡터 예측자 리스트를 만드는 것은, 특히 작은 블록 크기, 예를 들어, 4x4 또는 4xN (N<=8) 에 대해 현저한 상대적 (픽셀 당) 복잡성을 도입할 수도 있다. 더 작은 크기의 블록의 경우, 공간적으로 인접한 블록을 위한 모션 벡터 정보에 있어서 그만큼의 다양성이 없을 수도 있다. 가령, 현재 블록에 대해 좌측 이웃하는 블록과 상측 이웃하는 블록은 더 큰 크기의 블록에 대한 것보다 더 작은 크기의 블록에 대해 서로 더 근접할 수도 있다. 예시로서, 도 7a에서, 현재 블록 (700) 이 더 클수록, 블록 A (702) 및 블록 B0 (708) 은 다른 블록으로부터 더 멀리 떨어지고, 현재 블록 (700) 이 더 작을수록, 블록 A (702) 및 블록 B0 (708) 은 다른 블록에 더 가깝다.
- [0147] 서로 더 가까운 두 블록은 유사한 모션 벡터 정보를 갖는 경향이 있다. 따라서, 더 작은 크기의 블록의 경우, 공간적으로 이웃하는 블록이 더 큰 크기의 블록에 대한 공간적으로 이웃하는 블록에 비해 유사한 모션 벡터 정보를 가질 가능성이 더 높다. 공간적으로 이웃하는 블록들이 유사한 모션 벡터 정보를 가질 가능성이 더 높기 때문에, 상대적으로 큰 크기의 모션 벡터 예측자 리스트를 갖는 것은 모션 벡터 정보에 있어서 다양성을 제공하지 않을 수도 있다. 즉, 더 작은 크기의 블록에 대한 상대적으로 더 큰 크기의 모션 벡터 예측자 리스트를 갖는 것이 많은 모션 벡터 예측자 후보를 제공할 수도 있지만, 모션 벡터 정보에 있어서 다양성은 크지 않을 수도 있다. 그러나, 더 큰 크기의 블록의 경우, 상대적으로 큰 모션 벡터 예측자 리스트를 갖는 것은 실제 모션 벡터에 상대적으로 가까운 값을 갖는 모션 벡터 예측자 후보의 포함을 허용하는 다양한 모션 벡터 정보를 제공할 수도 있다.
- [0148] 따라서, 모든 크기의 블록에 대해 동일한 크기의 모션 벡터 예측자 리스트를 갖는 것은 바람직하지 않을 수도 있다. 작은 크기의 블록의 경우, 큰 크기의 블록과 동일한 크기의 모션 벡터 예측자 리스트를 갖는다는 것

은 비디오 인코더 (200) 및 비디오 디코더 (300) 가 모션 벡터 정보에 있어서의 다양성 부족 때문에 다수의 모션 벡터 예측자 후보를 갖는 것으로부터 코딩 이득을 달성함이 없이 모션 벡터 예측자 리스트에 채우기 위한 다수의 동작들을 수행할 필요가 있다는 것을 의미한다. 따라서, 더 작은 크기의 블록의 경우, 더 작은 크기의 모션 벡터 예측자 리스트가 유리할 수도 있다. 그러나, 더 큰 크기의 블록의 경우, 모션 벡터 예측자 리스트의 작은 크기로 인해 다양한 모션 벡터 예측자 후보가 모션 벡터 예측자 리스트에서 제외될 수도 있기 때문에 상대적으로 작은 크기의 모션 벡터 예측자 리스트를 갖는 것은 불리할 수도 있다.

[0149] 이러한 단점을 고려하여, 본 개시는 공간-시간적 모션 벡터 예측자 리스트 만들기를 수행하기 위한 다음 기술을 설명한다. 예시적인 기법은 개별적으로 또는 조합하여 적용될 수도 있다. 또한, 예시적인 기술은 위에서 설명한 문제를 해결할 수 있다. 그러나, 예시적인 기술은 위에서 설명한 문제를 해결하는 것으로 제한되어서는 안된다.

[0150] 하나의 예에서, 본 개시는 공간적 및 시간적 MVP 후보가 생성되는 패턴을 결정하기 위한 기술을 설명한다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 공간-시간적 이웃 후보들로부터 다음의 모션 벡터 예측자 리스트 만들기 프로세스를 사용할 수도 있다. 모션 벡터 예측자 리스트에서 후보 배치의 우선 순위는 도 8a에 도시된 바와 같이 0...7 으로부터의 숫자로 표시된다.

[0151] 먼저, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 도 8a에서 실선 블록에 도시된 순서 0...4 로 공간 MVP 후보를 방문 (즉, 모션 정보의 이용 가능성을 체크) 한다. 예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 정보가 블록 0 (802), 블록 1 (804), 블록 2 (806), 블록 3 (808) 및 블록 4 (810) 에 대해 이용 가능한지 여부를 결정할 수도 있다.

[0152] 다음으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 점선 블록으로 도시된 시간적 콜로케이트된 위치 (5, 6 및 7) 로부터 모션 벡터 예측자 후보를 방문 (즉, 모션 정보의 이용 가능성을 체크) 한다. 예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 는, 모션 벡터 정보가 중앙 콜로케이트된 블록 5 (812) 라고도 하는 블록 5 (812) 또는 하측 콜로케이트된 블록 6 (814) 라고도 하는 블록 6 (814) (여기서 하측 콜로케이트된 블록 6 (814) 는 도 8a 의 현재 블록 (800) 의 우측에 가장 가깝고 하단일 수도 있다), 또는 우측 콜로케이트된 블록 7 (816) 이라고도 하는 블록 7 (816) (여기서 우측 콜로케이트된 블록 7 (816) 은 도 8b 의 현재 블록 (800) 의 하단에 가장 가깝고 우측일 수도 있다) 에 대해 이용 가능한지 여부를 결정할 수도 있다.

[0153] 도 8a 는 본 개시의 일례에 따른 예시적인 공간적 시간적 모션 벡터 예측자 패턴을 예시하는 개념 도이다. 도 8b 는 본 개시의 일례에 따른 예시적인 인버트된 공간적 시간적 모션 벡터 예측자 패턴을 예시하는 개념 도이다. 도 8c 는 본 개시의 일례에 따른 다른 예시적인 인버트된 공간적 시간적 모션 벡터 예측자 패턴을 예시하는 개념 도이다. 도 8a 내지 도 8c 는 (예를 들어, 블록에서 숫자로 표시되는 바처럼) 상이한 체크 순서 그리고 (예를 들어, 점선 블록으로 표시되는 바처럼) 시간적 콜로케이트된 블록에 대한 상이한 위치를 도시한다.

[0154] 코딩 모드, 파티셔닝 타입 및/또는 코딩 방향, 특히 역 스캔 순서 프로세싱을 포함하는 코딩 모드에 따라, 공간적으로 인버트된 (또는 미러링된) 패턴이 도 8b 에 도시된 바와 같이 사용될 수 있다. 일부 예에서, 도 8a 에 도시된 패턴의 일부가 인버트될 수 있으며, 예를 들어, 공간적 위치만이 인버트되고, 시간적 후보 위치는 동일하게 남는다. 또 다른 예에서, MVP 후보들의 패턴 (예를 들어, 상대적 위치) 은 동일하게 남을 수도 있다. 그러나, 방문 순서는 도 8c 에 도시된 바와 같이 특정 축 (예를 들어, 대각선 축) 을 따라 공간적으로 인버트 (미러링) 될 수도 있다.

[0155] 하나 이상의 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각 현재 블록 (800) 을 포함하는 화상의 코딩 순서에 기초하여 모션 벡터 예측자 리스트를 각각 만들 수도 있다. 코딩 순서의 한 가지 예는 좌측에서 우측 코딩 순서이며, 이 경우 비디오 인코더 (200) 및 비디오 디코더 (300) 는, 현재 화상의 우측 끝에 도달할 때까지 현재 화상의 상단 좌측 코너로부터 시작해서, 다음으로 상단 좌측 코너 아래 블록, 그리고 하단 우측에 위치한 블록에 도달할 때까지 그렇게 계속함으로써 블록 별로 인코딩 또는 디코딩한다. 코딩 순서의 다른 예는 우측에서 좌측 코딩 순서이며, 이 경우 비디오 인코더 (200) 및 비디오 디코더 (300) 는, 현재 화상의 좌측 끝에 도달할 때까지 현재 화상의 상단 우측 코너로부터 시작해서, 다음으로 상단 우측 코너 아래 블록, 그리고 하단 좌측에 위치한 블록에 도달할 때까지 그렇게 계속함으로써 블록 별로 인코딩 또는 디코딩할 수 있다.

[0156] 코딩 순서는 어느 공간적으로 이웃하는 블록이 체크에 이용 가능한지에 영향을 줄 수도 있다. 가령, 이전에

인코딩 또는 디코딩된 블록만의 모션 벡터 정보가 이용 가능할 수도 있다. 따라서, 블록이 아직 인코딩 또는 디코딩되지 않은 경우, 그 블록을 위한 모션 벡터 정보는 아직 이용 가능하지 않을 수도 있다. 코딩 순서는 블록이 인코딩 또는 디코딩되는 순서를 정의하기 때문에, 공간적으로 이웃하는 블록의 모션 벡터 정보를 결정할 수 있는지 여부는 코딩 순서에 기초한다.

[0157] 하나의 예로서, 도 8a 에서, 좌측에서 우측 코딩 순서에 대해, 블록 0 (802) 의 모션 벡터 정보는 현재 블록 (800) 을 인코딩 또는 디코딩할 때 이용 가능할 수도 있다. 그러나, 현재 블록 (800) 에 대한 우측 이웃하는 블록의 모션 벡터 정보는 이용 가능하지 않을 수도 있다. 도 8b 에서, 우측에서 좌측 코딩 순서에 대해, 블록 0 (818) 의 모션 벡터 정보는 현재 블록 (800) 을 인코딩 또는 디코딩할 때 이용 가능할 수도 있다. 그러나, 현재 블록 (800) 좌측 이웃하는 블록의 모션 벡터 정보는 이용 가능하지 않을 수도 있다.

[0158] 하나 이상의 예들에서, 좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 제 1 모션 벡터 예측자 리스트를 만들 수도 있다. 도 8a 에 도시된 바와 같이, 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 제 1 블록에 대한 좌측 이웃하는 블록 (예를 들어, 블록 0 (802)) 의 모션 벡터 정보에 기초한다. 제 1 블록에 대해 좌측 이웃하는 블록은 제 1 블록의 하단 경계와 동일한 하단 경계를 갖는 제 1 블록의 좌측 블록일 수도 있다. 좌측에서 우측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 제 2 모션 벡터 예측자 리스트를 만들 수도 있다. 도 8b 에 예시된 바와 같이, 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 제 2 블록에 대한 우측 이웃하는 블록 (예를 들어, 블록 0 (818)) 의 모션 벡터 정보에 기초한다. 제 2 블록에 대해 우측 이웃하는 블록은 제 2 블록의 하단 경계와 동일한 하단 경계를 갖는 제 2 블록의 우측 블록일 수도 있다.

[0159] 예시된 바와 같이, 블록 0 (802) 은 도 8a 의 현재 블록 (800) 의 좌측에 있다. 일부 예들에서, 블록 0 (802) 은 블록 0 (802) 의 하단 경계가 도 8a 의 현재 블록 (800) 의 하단 경계와 동일하도록 도 8a 의 현재 블록 (800) 의 좌측에 위치할 수도 있다. 예시된 바와 같이, 블록 0 (818) 은 도 8b 의 현재 블록 (800) 의 우측에 있다. 일부 예들에서, 블록 0 (818) 은 블록 0 (818) 의 하단 경계가 도 8b 의 현재 블록 (800) 의 하단 경계와 동일하도록 도 8b 의 현재 블록 (800) 의 우측에 위치할 수도 있다.

[0160] 제 1 모션 벡터 예측자 리스트의 경우, 제 1 모션 벡터 예측자 리스트에 있는 엔트리는 다음과 같을 수도 있다: 제 1 엔트리는 좌측 이웃하는 블록 0 (802) 이고, 제 2 엔트리는 상단 이웃하는 블록 1 (802) 이고, 제 3 엔트리는 상단 우측 이웃하는 블록 2 (806) 이고, 제 4 엔트리는 하단 좌측 이웃하는 블록 3 (808) 이고, 제 5 엔트리는 상단 좌측 이웃하는 블록 4 (810) 이다. 일부 예들에서, 상단 이웃하는 블록 1 (802) 은 도 8a 의 현재 블록 (800) 의 상단일 수도 있고 도 8a 의 현재 블록 (800) 과 우측 경계를 공유할 수도 있다.

[0161] 제 2 모션 벡터 예측자 리스트의 경우, 제 2 모션 벡터 예측자 리스트에 있는 엔트리는 다음과 같을 수도 있다: 제 1 엔트리는 우측 이웃하는 블록 0 (818) 이고, 제 2 엔트리는 상단 이웃하는 블록 1 (820) 이고, 제 3 엔트리는 상단 좌측 이웃하는 블록 2 (822) 이고, 제 4 엔트리는 하단 우측 이웃하는 블록 3 (824) 이고, 제 5 엔트리는 상단 우측 이웃하는 블록 4 (826) 이다. 일부 예들에서, 상단 이웃하는 블록 1 (820) 은 도 8b 의 현재 블록 (800) 의 상단일 수도 있고 도 8b 의 현재 블록 (800) 과 좌측 경계를 공유할 수도 있다.

[0162] 일부 예들에서, 공간적으로 이웃하는 블록을 체크한 후, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 콜로케이트된 블록을 체크할 수도 있다. 일례에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 중앙 콜로케이트된 블록 5 (812) 를 체크할 수도 있다. 중앙 콜로케이트된 블록 5 (812) 를 위한 모션 벡터 정보가 이용 가능한 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있고 임의의 다른 콜로케이트된 블록으로부터 모션 벡터 정보를 추가하지 않을 수도 있다.

[0163] 중앙 콜로케이트된 블록 5 (812) 을 위한 모션 벡터 정보가 이용가능하지 않은 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 하단 콜로케이트된 블록 6 (814) 을 체크할 수도 있다. 하단 콜로케이트된 블록 6 (814) 을 위한 모션 벡터 정보가 이용 가능한 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있고 임의의 다른 콜로케이트된 블록으로부터 모션 벡터 정보를 추가하지 않을 수도 있다. 예시된 바와 같이, 하단 콜로케이트된 블록 6 (814) 은 현재 블록 (800) 의 우측에 가장 가깝고 하단인 블록일 수도 있다 (예를 들어, 블록 6 (814) 은 도 8a 의 현재 블록 (800) 아래에 있고 블록 6 (814) 의 우측 경계는 도 8a 의 현재 블록 (800) 의 우측 경계와 동일하다). 블록 6 (814) 에 대한 좌표는 (xColBot, yColBot) 일 수도 있으며, 여기서 $xColBot = xCb + cbWidth - 1$, 그리고 $yColBot = yCb + cbHeight$, 여기서 (xCb, yCb) 는 도 8a 의 현재 블록 (800) 에 대한 좌표이고 cbWidth는 도 8a 의 현재 블록

(800)의 폭이고, cbHeight는 도 8a의 현재 블록 (800)의 높이이다.

[0164] 하단 콜로케이트된 블록 6 (814)을 위한 모션 벡터 정보가 이용가능하지 않은 경우, 비디오 인코더 (200) 및 비디오 디코더 (300)는 우측 콜로케이트된 블록 7 (816)을 체크할 수도 있다. 우측 콜로케이트된 블록 7 (816)을 위한 모션 벡터 정보가 이용 가능한 경우, 비디오 인코더 (200) 및 비디오 디코더 (300)는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있고 임의의 다른 콜로케이트된 블록으로부터 모션 벡터 정보를 추가하지 않을 수도 있다. 예시된 바와 같이, 우측 콜로케이트된 블록 7 (816)은 도 8a의 현재 블록 (800)의 하단에 가장 가깝고 우측인 블록일 수도 있다 (예를 들어, 블록 7 (816)은 도 8a의 현재 블록 (800)의 우측에 있고 블록 7 (816)의 하단 경계는 도 8a의 현재 블록 (800)의 하단 경계와 동일하다). 블록 7 (816)에 대한 좌표는 (xColBr, yColBr)일 수도 있으며, 여기서 $xColBr = xCb + cbWidth$, 그리고 $yColBr = yCb + cbHeight - 1$, 여기서 (xCb, yCb)는 도 8a의 현재 블록 (800)에 대한 좌표이고 cbWidth는 도 8a의 현재 블록 (800)의 폭이고, cbHeight는 도 8a의 현재 블록 (800)의 높이이다.

[0165] 다른 예에서, 비디오 인코더 (200) 및 비디오 디코더 (300)는 중앙 콜로케이트된 블록 5 (828)를 체크할 수도 있다. 중앙 콜로케이트된 블록 5 (828)를 위한 모션 벡터 정보가 이용 가능한 경우, 비디오 인코더 (200) 및 비디오 디코더 (300)는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있고 임의의 다른 콜로케이트된 블록으로부터 모션 벡터 정보를 추가하지 않을 수도 있다.

[0166] 중앙 콜로케이트된 블록 5 (828)을 위한 모션 벡터 정보가 이용가능하지 않은 경우, 비디오 인코더 (200) 및 비디오 디코더 (300)는 하단 콜로케이트된 블록 6 (830)을 체크할 수도 있다. 하단 콜로케이트된 블록 6 (830)을 위한 모션 벡터 정보가 이용 가능한 경우, 비디오 인코더 (200) 및 비디오 디코더 (300)는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있고 임의의 다른 콜로케이트된 블록으로부터 모션 벡터 정보를 추가하지 않을 수도 있다. 예시된 바와 같이, 하단 콜로케이트된 블록 6 (830)은 현재 블록 (800)의 좌측에 가장 가깝고 하단인 블록일 수도 있다 (예를 들어, 블록 6 (830)은 도 8b의 현재 블록 (800)아래에 있고 블록 6 (830)의 좌측 경계는 도 8b의 현재 블록 (800)의 좌측 경계와 동일하다). 블록 6 (830)에 대한 좌표는 (xColBot, yColBot)일 수도 있으며, 여기서 $xColBot = xCb$, 그리고 $yColBot = yCb + cbHeight$, 여기서 (xCb, yCb)는 도 8b의 현재 블록 (800)에 대한 좌표이고 cbWidth는 도 8b의 현재 블록 (800)의 폭이고, cbHeight는 도 8b의 현재 블록 (800)의 높이이다.

[0167] 하단 콜로케이트된 블록 6 (830)을 위한 모션 벡터 정보가 이용가능하지 않은 경우, 비디오 인코더 (200) 및 비디오 디코더 (300)는 좌측 콜로케이트된 블록 7 (832)을 체크할 수도 있다. 좌측 콜로케이트된 블록 7 (832)을 위한 모션 벡터 정보가 이용 가능한 경우, 비디오 인코더 (200) 및 비디오 디코더 (300)는 모션 벡터 정보를 모션 벡터 예측자 리스트에 추가할 수도 있고 임의의 다른 콜로케이트된 블록으로부터 모션 벡터 정보를 추가하지 않을 수도 있다. 예시된 바와 같이, 좌측 콜로케이트된 블록 7 (832)은 도 8b의 현재 블록 (800)의 하단에 가장 가깝고 좌측인 블록일 수도 있다 (예를 들어, 블록 7 (832)은 도 8b의 현재 블록 (800)의 좌측에 있고 블록 7 (832)의 하단 경계는 도 8b의 현재 블록 (800)의 하단 경계와 동일하다). 블록 7 (832)에 대한 좌표는 (xColBr, yColBr)일 수도 있으며, 여기서 $xColBr = xCb - 1$, 그리고 $yColBr = yCb + cbHeight - 1$, 여기서 (xCb, yCb)는 도 8b의 현재 블록 (800)에 대한 좌표이고 cbWidth는 도 8b의 현재 블록 (800)의 폭이고, cbHeight는 도 8b의 현재 블록 (800)의 높이이다.

[0168] 도 8a 및 도 8b에서 공간적으로 이웃하고 콜로케이트된 블록의 위치는 하나의 예이고 제한하는 것으로 간주해서는 안된다는 것이 이해되어야 한다. 예를 들어, 블록 (802)은 일반적으로 도 8a의 현재 블록 (800)의 좌측에 위치할 수도 있고 블록 (818)은 일반적으로 도 8b의 현재 블록 (800)의 우측에 위치할 수도 있다. 다른 공간적으로 그리고 콜로케이트된 블록에도 동일하게 적용된다.

[0169] 위의 예는 비디오 인코더 (200) 및 비디오 디코더 (300)가 모션 벡터 예측자 리스트에서 모션 벡터 정보를 순서화하는 예시적인 방법을 설명한다. 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300)가 특정 순서로 모션 벡터 예측자 리스트에서, 이용 가능한 경우, 모션 벡터 정보를 순서화하지만, 비디오 인코더 (200) 및 비디오 디코더 (300)는 병렬적으로 이웃하는 블록들 (예를 들어, 도 8a 및 도 8b에서의 공간적으로 이웃하는 블록 0..4)의 모션 벡터 정보에 액세스하도록 구성될 수도 있다. 즉, 비디오 인코더 (200) 및 비디오 디코더 (300)는 모션 벡터 예측자 리스트를 병렬로 만드는 데 사용되는 복수의 (예를 들어, 모든) 공간적으로 이웃하는 블록들을 위한 모션 벡터 정보를 체크할 수도 있다.

[0170] 예를 들어, 본 개시는 모션 벡터 예측자 리스트에 모션 벡터 예측자 후보를 포함하기 위한 기술을 설명한다. 모션 벡터 예측자 리스트 만들기의 다음 과정을 설명한다. 비디오 인코더 (200) 및 비디오 디코더 (300)

는 다른 모션 벡터 예측자 후보들의 이용 가능성에 독립적이고 무조건적의 순서로 모션 벡터 예측자 후보 위치들 (예를 들어, 모든 공간적으로 이웃하는 블록들) 을 체크할 수도 있다. 따라서, 모션 벡터 예측자 후보 체크는 병렬로 수행될 수 있다.

- [0171] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 이웃하는 블록의 위치가 화상 경계 내에 있는 경우 이웃하는 블록을 체크할 수도 있다. 또한, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 이웃하는 블록이 이미 인코딩 또는 디코딩되었던 위치에 있는 이웃하는 블록을 체크할 수도 있다. 또한, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 정보가 이웃하는 블록에 대해 이용 가능한지 여부를 결정할 수도 있다. 위의 예시적인 조건이 만족되면, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 모션 벡터 예측자 리스트에 이웃하는 블록의 모션 벡터 정보를 포함할 수도 있다.
- [0172] 전술한 바와 같이, 모션 벡터 예측자 리스트의 최대 크기가 있을 수도 있다. 예를 들어, 모션 벡터 예측자 리스트의 유효 크기가, 예를 들어, 모션 벡터 예측자 리스트의 최대 크기를 정의하는 MAX_MVP_LIST_SIZE 에 의해 지정되는, 지정된 상한값에 도달할 때까지, 폐칭된 모션 벡터 예측자 후보가 모션 벡터 예측자 리스트에 포함될 수도 있다. 일부 예에서, MAX_MVP_LIST_SIZE 의 값은 6 과 같을 수도 있다.
- [0173] 그러나, 전술한 바와 같이, 블록 크기를 고려하지 않고 모든 블록에 대해 동일한 최대 크기를 갖는 것은, 불필요하게 만들기 복잡하거나 충분한 후보를 포함하지 않는 모션 벡터 예측자 리스트를 초래할 수도 있다. 예를 들어, 전술한 바와 같이, 더 작은 크기의 블록에 대해, 계산 사이클을 줄이기 위해, 추가 후보가 많은 코딩 이득을 제공하지 않을 수도 있기 때문에 더 작은 크기의 모션 벡터 예측자 리스트를 갖는 것이 유리할 수도 있다. 그러나, 더 큰 크기의 블록의 경우, 모션 벡터 정보에 있어서 양호한 다양성을 보장하기 위해, 더 큰 크기의 모션 벡터 예측자 리스트를 갖는 것이 유리할 수도 있다.
- [0174] 일례에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 현재 코딩된 블록 크기에 따라 만들어진 모션 벡터 예측자 리스트의 최대 유효 크기를 결정 및/또는 제한하도록 구성될 수도 있다. 일부 예에서, 4xN 또는 Nx4 개의 픽셀 (여기서 N<=8) 을 포함하는 블록에 대해, 유효 MVP 리스트 크기는 SMALL_BLOCK_MAX_MVP_LIST<MAX_MVP_LIST_SIZE 와 동일하게 설정된다. 변수 SMALL_BLOCK_MAX_MVP_LIST 는 더 작은 크기의 블록을 위한 모션 벡터 예측자 리스트의 최대 크기를 정의하고, 여기서 더 작은 크기의 블록은, 샘플이 4xN 또는 Nx4보다 작거나 같고 N 이 8 인 블록일 수도 있다.
- [0175] 현재 블록이 작은 크기의 블록인지 여부를 나타내기 위해, 비디오 인코더 (200) 및/또는 비디오 디코더 (300) 는 smallBlockFlag (예를 들어, 블록이 작은지 여부를 나타내는 이진 값) 을 결정할 수도 있다. smallBlockFlag 를 결정하는 등식은 smallBlockFlag = (cuw *cuh < SMALL_BLOCK_SIZE); SMALL_BLOCK_SIZE = 32 일 수도 있고, 여기서 cuw는 현재 블록의 폭이고 cuh는 현재 블록의 높이이다.
- [0176] 일부 예들에서, 작은 블록 크기 SMALL_BLOCK_SIZE 의 기준은 16 픽셀과 동일한 것으로 정의될 수 있다. 따라서, 작은 블록 크기에 대한 32 의 예는 단지 하나의 예이고 32 및 16 외의 값이 작은 블록 크기를 위해 본 개시에 의해 고려된다.
- [0177] 일부 예에서, MAX_MVP_LIST_SIZE 의 값은 4 과 같을 수도 있다. 즉, 작은 블록에 대해, 모션 벡터 예측자 리스트의 최대 크기는 4 일 수도 있다. 일부 예에서, 작은 블록 크기의 기준은 블록 예지 길이 값을 통해 표현될 수 있다:
- [0178] smallBlockFlag = ((max(cuw, cuh) <= 8 && min(cuw, cuh) < 8))
- [0179] (smallBlockFlag) 이면
- [0180] SMALL_BLOCK_MAX_MVP_LIST 를 사용;
- [0181] 그렇지 않으면
- [0182] MAX_MVP_LIST_SIZE 를 사용
- [0183] 다른 예에서, 본 개시는 모션 벡터 예측자 후보 리스트 파라미터의 시그널링을 위한 기술을 설명한다. 예를 들어, 최대 가능한 모션 벡터 예측자 리스트 크기 (MAX_MVP_LIST_SIZE) 는 디코더 측 (예를 들어, 비디오 디코더 (300)) 에서 부가 정보 (예를 들어, 비디오 인코더 (200) 로부터 시그널링된 정보) 로서 지정될 수 있다.
- [0184] 일부 예들에서, 현재 코딩 조건, 예를 들어 현재 시퀀스, 화상, 슬라이스 또는 CU 그룹을 위해 애플리케이션에 의해 유효하게 사용되는 최대 모션 벡터 예측자 리스트 크기 (예를 들어, MAX_MVP_LIST_SIZE 및/또는

SMAL_BLOCK_MAX_MVP_LIST) 는, 예를 들어, 시퀀스/화상/슬라이스 헤더 또는 다른 레벨에서, 비트스트림의 구분 요소에 의해, 시그널링될 수도 있다.

- [0185] 일부 예들에서, 블록 크기 의존성의 파라미터들, 예를 들어 SMALL_BLOCK_SIZE 는 부가 정보로서 디코더 측에서 지정된다. 다른 예에서, 이 파라미터는 비트스트림을 통해 시그널링될 수도 있다.
- [0186] 도 2 는 본 개시의 기술을 수행할 수도 있는 예시적인 비디오 인코더 (200) 를 예시하는 블록 도이다. 도 2 는 설명의 목적으로 제공되며, 본 개시에서 대체로 예시화되고 설명된 바와 같은 기술의 한정으로서 고려되지 않아야 한다. 설명의 목적으로, 본 개시는 HEVC 비디오 코딩 표준 및 개발 중인 H.266/VVC 비디오 코딩 표준과 같은 비디오 코딩 표준들의 컨텍스트에서 비디오 인코더 (200) 를 설명한다. 그러나, 본 개시의 기술은 이들 비디오 코딩 표준들에 한정되지 않으며, 일반적으로 비디오 인코딩 및 디코딩에 적용가능하다.
- [0187] 도 2 의 예에서, 비디오 인코더 (200) 는 비디오 데이터 메모리 (230), 모드 선택 유닛 (202), 잔차 생성 유닛 (204), 변환 프로세싱 유닛 (206), 양자화 유닛 (208), 역 양자화 유닛 (210), 역 변환 프로세싱 유닛 (212), 재구성 유닛 (214), 필터 유닛 (216), 디코딩된 화상 버퍼 (DPB) (218), 및 엔트로피 인코딩 유닛 (220) 을 포함한다.
- [0188] 비디오 데이터 메모리 (230) 는 비디오 인코더 (200) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 인코더 (200) 는 예를 들어, 비디오 소스 (104) (도 1) 로부터 비디오 데이터 메모리 (230) 에 저장된 비디오 데이터를 수신할 수도 있다. DPB (218) 는 비디오 인코더 (200) 에 의한 후속 비디오 데이터의 예측에 사용하기 위해 참조 비디오 데이터를 저장하는 참조 화상 메모리로서 작용할 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동기식 동적 랜덤 액세스 메모리 (synchronous dynamic random access memory; SDRAM) 을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항성 RAM (magnetoresistive RAM; MRAM), 저항성 RAM (resistive RAM; RRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 어느 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동일한 메모리 디바이스 또는 개별 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (230) 는 예시된 바와 같이 비디오 인코더 (200) 의 다른 컴포넌트들과 온-칩되거나, 또는 그 컴포넌트들에 대해 오프-칩될 수도 있다.
- [0189] 본 개시에서, 비디오 데이터 메모리 (230) 에 대한 언급은 이처럼 구체적으로 설명되지 않으면 비디오 인코더 (200) 내부의 메모리, 또는 이처럼 구체적으로 설명되지 않으면 비디오 인코더 (200) 외부의 메모리에 한정되는 것으로 해석되지 않아야 한다. 오히려, 비디오 데이터 메모리 (230) 에 대한 언급은 비디오 인코더 (200) 가 인코딩을 위해 수신하는 비디오 데이터 (예를 들어, 인코딩될 현재 블록을 위한 비디오 데이터) 를 저장하는 메모리로서 이해되어야 한다. 도 1 의 메모리 (106) 는 또한 비디오 인코더 (200) 의 다양한 유닛들로부터의 출력들의 일시적 저장을 제공할 수도 있다.
- [0190] 도 2 의 다양한 유닛들은 비디오 인코더 (200) 에 의해 수행되는 동작들의 이해를 돕기 위해 예시된다. 이 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하며, 수행될 수 있는 동작들에 대해 미리설정된다. 프로그래밍가능 회로들은 다양한 태스크들을 수행하도록 프로그램될 수 있는 회로들을 지칭하며, 수행될 동작들에서 유연한 기능성을 제공한다. 예를 들어, 프로그래밍가능 회로들은 프로그래밍가능 회로들이 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 소프트웨어 명령들을 (예를 들어, 파라미터들을 수신하거나 파라미터들을 출력하기 위해) 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작 타입들은 일반적으로 불변이다. 일부 예들에서, 유닛들의 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그래밍가능) 일 수도 있고, 일부 예들에서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.
- [0191] 비디오 인코더 (200) 는 프로그래밍가능 회로들로부터 형성된, 산술 로직 유닛 (arithmetic logic unit; ALU) 들, 기본 기능 유닛 (elementary function unit; EFU) 들, 디지털 회로들, 아날로그 회로들, 및/또는 프로그래밍가능 코어들을 포함할 수도 있다. 비디오 인코더 (200) 의 동작이 프로그램 가능 회로에 의해 실행되는 소프트웨어를 사용하여 수행되는 예에서, 메모리 (106) (도 1) 는 비디오 인코더 (200) 가 수신하고 실행하는 소프트웨어의 오브젝트 코드를 저장할 수도 있거나, 또는 비디오 인코더 (200) 내의 다른 메모리 (미도시) 가 그러한 명령들을 저장할 수도 있다.
- [0192] 비디오 데이터 메모리 (230) 는 수신된 비디오 데이터를 저장하도록 구성된다. 비디오 인코더 (200) 는 비

디오 데이터 메모리 (230)로부터 비디오 데이터의 화상을 추출하고 비디오 데이터를 잔차 생성 유닛 (204) 및 모드 선택 유닛 (202)에 제공할 수도 있다. 비디오 데이터 메모리 (230)에서의 비디오 데이터는 인코딩될 원시 비디오 데이터일 수도 있다.

[0193] 모드 선택 유닛 (202)은 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라-예측 유닛 (226)을 포함한다. 모드 선택 유닛 (202)은 다른 예측 모드들에 따라 비디오 예측을 수행하기 위해 추가적인 기능 유닛들을 포함할 수도 있다. 예를로서, 모드 선택 유닛 (202)은 팔레트 유닛, 인트라-블록 카피 유닛 (이는 모션 추정 유닛 (222) 및/또는 모션 보상 유닛 (224)의 부분일 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다.

[0194] 모드 선택 유닛 (202)은 일반적으로 인코딩 파라미터들의 조합들 및 그러한 조합들에 대한 결과적인 레이트-왜곡 값들을 테스트하기 위해 다수의 인코딩 패스들을 조정한다. 인코딩 파라미터들은 CTU들의 CU들로의 파티셔닝, CU들을 위한 예측 모드들, CU들의 잔차 데이터를 위한 변환 타입들, CU들의 잔차 데이터를 위한 양자화 파라미터들 등을 포함할 수도 있다. 모드 선택 유닛 (202)은 궁극적으로 다른 테스트된 조합들보다 더 나은 레이트-왜곡 값들을 갖는 인코딩 파라미터들의 조합을 선택할 수도 있다.

[0195] 비디오 인코더 (200)는 비디오 데이터 메모리 (230)로부터 추출된 화상을 일련의 CTU들로 파티셔닝하고, 슬라이스 내에 하나 이상의 CTU들을 캡슐화할 수도 있다. 모드 선택 유닛 (202)은 위에 설명된 HEVC의 쿼드트리 구조 또는 QTBT 구조와 같은, 트리 구조에 따라 화상의 CTU를 파티셔닝할 수도 있다. 위에 설명된 바와 같이, 비디오 인코더 (200)는 트리 구조에 따라 CTU를 파티셔닝하는 것으로부터 하나 이상의 CU들을 형성할 수도 있다. 그러한 CU는 일반적으로 "비디오 블록" 또는 "블록"으로도 또한 지칭될 수도 있다.

[0196] 일반적으로, 모드 선택 유닛 (202)은 또한 그의 컴포넌트들 (예를 들어, 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라-예측 유닛 (226))을 제어하여 현재 블록 (예를 들어, 현재 CU, 또는 HEVC에서, PU 및 TU의 오버랩하는 부분)을 위한 예측 블록을 생성한다. 현재 블록의 인트라-예측을 위해, 모션 추정 유닛 (222)은 하나 이상의 참조 화상들 (예를 들어, DPB (218)에 저장된 하나 이상의 이전에 코딩된 화상들)에서 하나 이상의 밀접하게 매칭하는 참조 블록들을 식별하기 위해 모션 탐색을 수행할 수도 있다. 특히, 모션 추정 유닛 (222)은, 예를 들어 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of squared differences; SSD), 평균 절대차 (mean absolute difference; MAD), 평균 제곱차의 합 (mean squared differences; MSD)에 따라, 잠재적 참조 블록이 현재 블록에 얼마나 유사한지를 나타내는 값을 계산할 수도 있다. 모션 추정 유닛 (222)은 일반적으로 고려되는 참조 블록과 현재 블록 사이의 샘플 별 (sample-by-sample) 차이들을 사용하여 이러한 계산들을 수행할 수도 있다. 모션 추정 유닛 (222)은 현재 블록과 가장 밀접하게 매칭하는 참조 블록을 표시하는, 이러한 계산들로부터 비롯되는 최저 값을 갖는 참조 블록을 식별할 수도 있다.

[0197] 모션 추정 유닛 (222)은 현재 화상에서의 현재 블록의 위치에 대한 참조 화상들에서의 참조 블록들의 위치들을 정의하는 하나 이상의 모션 벡터 (MV) 들을 형성할 수도 있다. 모션 추정 유닛 (222)은 그 후 모션 벡터들을 모션 보상 유닛 (224)에 제공할 수도 있다. 예를 들어, 단방향 인트라-예측에 대해, 모션 추정 유닛 (222)은 단일 모션 벡터를 제공할 수도 있는 반면, 양방향 인트라-예측에 대해, 모션 추정 유닛 (222)은 2개의 모션 벡터들을 제공할 수도 있다. 그 후, 모션 보상 유닛 (224)은 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 예를 들어, 모션 보상 유닛 (224)은 모션 벡터를 사용하여 참조 블록의 데이터를 추출할 수도 있다. 다른 예로서, 모션 벡터가 분수 샘플 정밀도를 갖는다면, 모션 보상 유닛 (224)은 하나 이상의 보간 필터들에 따라 예측 블록을 위한 값들을 보간할 수도 있다. 또한, 양방향 인트라-예측에 대해, 모션 보상 유닛 (224)은 각각의 모션 벡터들에 의해 식별된 2개의 참조 블록들을 위한 데이터를 추출하고, 예를 들어 샘플 별 평균화 또는 가중된 평균화를 통해 추출된 데이터를 결합할 수도 있다.

[0198] 본 개시의 기술에 따라, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224)은 본 개시에 설명된 인트라-예측 및 모션 벡터 예측 기술을 수행할 수도 있다. 예를 들어, 아래에서 더 상세히 설명되는 바와 같이, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224)은 본 개시에서 설명된 예시적인 기술을 이용하여 모션 벡터 예측자 리스트를 만들도록 구성될 수도 있다.

[0199] 전술한 바와 같이, 일부 예들에서, 블록을 위한 모션 벡터 예측자 리스트는 그 블록 및 공간적으로 이웃하는 블록 및/또는 콜로케이트된 블록을 포함하는 화상의 코딩 순서에 기초할 수도 있다. 가령, 하나의 예에서, 5개의 공간적으로 이웃하는 블록은 좌측에서 우측 코딩 순서에 대해 도 8a에 예시된 것일 수도 있고 5개의 공간적으로 이웃하는 블록은 우측에서 좌측 코딩 순서에 대해 도 8b에 예시된 것일 수도 있다.

- [0200] 가령, 좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 모션 벡터 예측자 리스트를 만들 수도 있다. 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 제 1 블록에 대한 좌측 이웃하는 블록 (예를 들어, 도 8a 의 블록 0 (802)) 의 모션 벡터 정보에 기초할 수도 있다. 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 2 모션 벡터 예측자 리스트를 구성할 수도 있다. 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 제 2 블록에 대한 우측 이웃하는 블록 (예를 들어, 도 8b 의 블록 0 (818)) 의 모션 벡터 정보에 기초할 수도 있다.
- [0201] 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 병렬적으로, 복수의 (전부 포함) 공간적으로 이웃하는 블록을 위한 비디오 데이터 메모리 (230) 또는 디코딩된 화상 버퍼 (218) (2 개의 예로서) 에 저장된 모션 벡터 정보를 체크할 수도 있다. 즉, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 이 특정 공간적으로 이웃하는 블록을 위한 모션 벡터 정보를 체크하는지 여부는 임의의 다른 블록의 모션 벡터 정보에 의존하지 않을 수도 있다.
- [0202] 공간적으로 이웃하는 블록들의 모션 벡터 정보의 체크에 기초하여, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 모션 벡터 정보의 이용 가능성에 기초한 순서로 모션 벡터 정보를 갖는 모션 벡터 예측자 리스트를 만들 수도 있다. 좌측에서 우측 코딩 순서를 위한 모션 벡터 예측자 리스트에서 모션 벡터 정보의 순서는 다음과 같을 수도 있다: 좌측 이웃하는 블록 (예를 들어, 도 8a의 블록 0 (802)) 의 모션 벡터 정보, 상측 이웃하는 블록 (예를 들어, 도 8a의 블록 1 (804)) 의 모션 벡터 정보, 우상측 이웃하는 블록 (예를 들어, 도 8a의 블록 2 (806)) 의 모션 벡터 정보, 좌하측 이웃하는 블록 (예를 들어, 도 8a의 블록 3 (808)) 의 모션 벡터 정보, 및 좌상측 이웃하는 블록 (예를 들어, 도 8a의 블록 4 (810)) 의 모션 벡터 정보. 우측에서 좌측 코딩 순서를 위한 모션 벡터 예측자 리스트에서 모션 벡터 정보의 순서는 다음과 같을 수도 있다: 우측 이웃하는 블록 (예를 들어, 도 8b의 블록 0 (818)) 의 모션 벡터 정보, 상측 이웃하는 블록 (예를 들어, 도 8b의 블록 1 (820)) 의 모션 벡터 정보, 좌상측 이웃하는 블록 (예를 들어, 도 8b 의 블록 2 (822)) 의 모션 벡터 정보, 우하측 이웃하는 블록 (예를 들어, 도 8b의 블록 3 (824)) 의 모션 벡터 정보, 및 우상측 이웃하는 블록 (예를 들어, 도 8b의 블록 4 (826)) 의 모션 벡터 정보.
- [0203] 위의 순서는 모션 벡터 정보의 이용 가능성에 기초한다. 특정 블록에 대해 모션 벡터 정보가 이용 가능하지 않은 경우, 그 모션 벡터 정보는 모션 벡터 예측자 리스트에 추가되지 않는다. 일부 예들에서, 다음 이용 가능한 모션 벡터 정보는 모션 벡터 정보가 없는 블록에 대해 예약된 공간에 포함될 수도 있다. 예를 들어, 도 8a 의 블록 1 (804) 를 위한 모션 벡터 정보는 모션 벡터 예측자 리스트에서 두번째 엔트리일 수도 있고, 도 8a 의 블록 2 (806) 를 위한 모션 벡터 정보는 모션 벡터 예측자 리스트에서 세번째 엔트리일 수도 있다. 도 8a 의 블록 1 (804) 를 위한 모션 벡터 정보가 이용 가능하지 않고, 도 8a 의 블록 2 (806) 를 위한 모션 벡터 정보가 이용 가능하다면, 블록 2 (806) 를 위한 모션 벡터 정보는 모션 벡터 예측자 리스트의 두번째 엔트리일 수도 있다.
- [0204] 위의 예에서, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 공간적으로 이웃하는 블록에 기초하여 제 1 모션 벡터 예측자 리스트 및 제 2 모션 벡터 예측자 리스트를 결정하였다. 일부 예들에서, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 모션 벡터 예측자 리스트 및 제 2 모션 벡터 예측자 리스트에 콜로케이트된 블록의, (2 개의 예로서) 비디오 데이터 메모리 (230) 또는 디코딩된 화상 버퍼 (218) 에 저장된, 모션 벡터 정보를 포함할 수도 있다.
- [0205] 일례로, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은, 제 1 화상 외의 화상에 위치한 임의의 다른 콜로케이트된 블록에서의 모션 벡터 정보가 이용 가능한지 여부를 결정하기 전에 제 1 화상외의 화상에 위치한 중앙 콜로케이트된 블록 (예를 들어, 도 8a 의 블록 5 (812)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정할 수도 있고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 모션 벡터 예측자 리스트에 중앙 콜로케이트된 블록을 위한 모션 벡터 정보를 추가할 수도 있다.
- [0206] 일부 예들에서, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정할 수도 있다. 이러한 예들에서, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 화상 외의 화상에 위치한 하단 콜로케이트된 블록 (예를 들어, 도 8a 의 블록 6 (814)) 을 위한 모션 벡터 정보가 이용 가능한지를 결정할 수도 있고, 하단 콜로케이트된 블록을 위한 모션 벡터 정보

가 이용 가능한 것에 기초하여, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 하단 콜로케이트된 블록을 위한 모션 벡터 정보를 제 1 모션 벡터 예측자 리스트에 추가할 수도 있다.

[0207] 일부 예들에서, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 화상 외의 화상에 위치된 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정할 수도 있다. 일부 경우에서, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 화상 외의 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정할 수도 있다. 이러한 경우들에서, 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 화상 외의 화상에 위치한 우측 콜로케이트된 블록 (예를 들어, 도 8a 의 블록 7 (816)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정할 수도 있고, 우측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 우측 콜로케이트된 블록을 위한 모션 벡터 정보를 제 1 모션 벡터 예측자 리스트에 추가할 수도 있다.

[0208] 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 2 모션 벡터 예측자 리스트를 만들기 위해 콜로케이트된 블록을 위한 모션 벡터 정보를 추가하기 위한 유사한 동작을 수행할 수도 있다. 그러나, 제 2 모션 벡터 예측자 리스트에 대해, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 도 8b 의 중앙 콜로케이트된 블록 5 (828), 하단 콜로케이트된 블록 6 (830) 및 좌측 콜로케이트된 블록 7 (832) 의 이용 가능성에 기초하여 모션 벡터 정보를 추가할 수도 있다.

[0209] 모션 벡터 예측자 리스트를 만들 때, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 일부 예들에서 블록의 크기에 기초하여 모션 벡터 예측자 리스트를 만들 수도 있다. 예를 들어, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 블록이 임계 크기보다 큰 크기를 갖는 것에 기초하여 제 1 최대 크기를 갖는 제 1 모션 벡터 예측자 리스트를 만들 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 화상에서 제 3 블록에 대해, 제 3 블록이 임계 크기보다 작은 크기를 갖는 것에 기초하여, 제 1 최대 크기보다 작은 제 2 최대 크기를 갖는, 제 3 모션 벡터 예측자 리스트를 만들 수도 있다. 하나의 예로서, 제 1 최대 크기는 6과 같고, 제 2 최대 크기는 4 와 같고, 임계 크기는 $N \times 4$ 또는 $4 \times N$ 와 같고, 여기서 N 은 8보다 작거나 같다.

[0210] 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 블록 및 제 2 블록을 인코딩하도록 구성될 수도 있다. 예를 들어, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 1 블록을 위한 예측 블록을 결정하고 예측 블록을 식별하는 제 1 블록을 위한 모션 벡터를 결정할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 모션 벡터에 기초하여 제 1 모션 벡터 예측자 리스트에서 엔트리를 결정할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 엔트로피 인코딩 유닛 (220) 으로 하여금 제 1 모션 벡터 예측자 리스트에서의 엔트리를 나타내는 정보를 시그널링하게 할 수도 있다. 또한, 잔차 생성 유닛 (204) 은 제 1 블록과 예측 블록 사이의 차이를 나타내는 잔차 데이터를 결정할 수도 있다. 엔트로피 인코딩 유닛 (220) 은 잔차 데이터를 나타내는 정보를 시그널링할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 제 2 모션 벡터 예측자 리스트를 사용하여 제 2 블록을 인코딩하기 위한 유사한 동작을 수행할 수도 있다.

[0211] 다른 예로서, 인트라-예측, 또는 인트라-예측 코딩에 대해, 인트라-예측 유닛 (226) 은 현재 블록에 이웃하는 샘플들로부터 예측 블록을 생성할 수도 있다. 예를 들어, 방향성 모드들에 대해, 인트라-예측 유닛 (226) 은 일반적으로 이웃하는 샘플들의 값들을 수학적으로 결합하고 현재 블록에 걸쳐 정의된 방향에서 이들 계산된 값들을 파플레이트하여 예측 블록을 생성할 수도 있다. 또 다른 예로서, DC 모드에 대해, 인트라-예측 유닛 (226) 은 현재 블록에 대한 이웃하는 샘플들의 평균을 계산하고 예측 블록을 생성하여 예측 블록의 각각의 샘플에 대해 이러한 결과적인 평균을 포함할 수도 있다.

[0212] 모드 선택 유닛 (202) 은 예측 블록을 잔차 생성 유닛 (204) 에 제공한다. 잔차 생성 유닛 (204) 은 비디오 데이터 메모리 (230) 로부터의 현재 블록의 원시, 코딩되지 않은 버전 및 모드 선택 유닛 (202) 으로부터의 예측 블록을 수신한다. 잔차 생성 유닛 (204) 은 현재 블록과 예측 블록 사이의 샘플 별 차이들을 계산한다. 결과적인 샘플 별 차이들은 현재 블록을 위한 잔차 블록을 정의한다. 일부 예들에서, 잔차 생성 유닛 (204) 은 또한 잔차 차분 펄스 코드 변조 (residual differential pulse code modulation; RDPCM) 를 사용하여 잔차 블록을 생성하기 위해 잔차 블록에서의 샘플 값들 사이의 차이들을 결정할 수도 있다. 일부 예들에서, 잔차 생성 유닛 (204) 은 이진 감산 (binary subtraction) 을 수행하는 하나 이상의 감산기 회로들을 사용하여 형성될 수도 있다.

[0213] 모드 선택 유닛 (202) 이 CU들을 PU들로 파티셔닝하는 예들에서, 각각의 PU 는 루마 예측 유닛 및 대응하는 크

로마 예측 유닛들과 연관될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다양한 사이즈를 갖는 PU들을 지원할 수도 있다. 위에 나타낸 바와 같이, CU 의 크기는 CU 의 루마 코딩 블록의 크기를 지칭할 수도 있고 PU 의 크기는 PU 의 루마 예측 유닛의 크기를 지칭할 수도 있다. 특정 CU 의 크기가 2Nx2N 이라고 가정하면, 비디오 인코더 (200) 는 인트라 예측을 위해 2Nx2N 또는 NxN 의 PU 크기들을 지원하고, 인터 예측을 위해 2Nx2N, 2NxN, Nx2N, NxN, 또는 유사한 것의 대칭적인 PU 크기들을 지원할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 또한, 인터 예측을 위해 2NxnU, 2NxnD, nLx2N, 및 nRx2N 의 PU 크기들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

- [0214] 모드 선택 유닛 (202) 이 CU 를 PU들로 추가로 파티셔닝하지 않는 예들에서, 각각의 CU 는 루마 코딩 블록 및 대응하는 크로마 코딩 블록들과 연관될 수도 있다. 위와 같이, CU 의 크기는 CU 의 루마 코딩 블록의 크기를 지칭할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 2Nx2N, 2NxN, 또는 Nx2N 의 CU 크기들을 지원할 수도 있다.
- [0215] 몇개의 예로서, 인트라-블록 카피 모드 코딩, 아핀-모드 코딩, 및 선형 모델 (LM) 모드 코딩과 같은 다른 비디오 코딩 기술에 대해, 모드 선택 유닛 (202) 은 코딩 기술과 연관된 각각의 유닛들을 통해, 인코딩되는 현재 블록을 위한 예측 블록을 생성한다. 팔레트 모드 코딩과 같은 일부 예들에서, 모드 선택 유닛 (202) 은 예측 블록을 생성하지 않을 수도 있고, 대신 선택된 팔레트에 기초하여 블록을 재구성하는 방식을 표시하는 구문 요소들을 생성할 수도 있다. 그러한 모드들에서, 모드 선택 유닛 (202) 은 이들 구문 요소들을 인코딩될 엔트 로피 인코딩 유닛 (220) 에 제공할 수도 있다.
- [0216] 위에 설명된 바와 같이, 잔차 생성 유닛 (204) 은 현재 블록 및 대응하는 예측 블록을 위한 비디오 데이터를 수신한다. 잔차 생성 유닛 (204) 은 그 후 현재 블록을 위한 잔차 블록을 생성한다. 잔차 블록을 생성하기 위해, 잔차 생성 유닛 (204) 은 예측 블록과 현재 블록 사이의 샘플 별 차이들을 계산한다.
- [0217] 변환 프로세싱 유닛 (206) 은 잔차 블록에 하나 이상의 변환들을 적용하여 변환 계수들의 블록 (본 명세서에서는 "변환 계수 블록" 으로 지칭됨) 을 생성한다. 변환 프로세싱 유닛 (206) 은 다양한 변환들을 잔차 블록에 적용하여 변환 계수 블록을 형성할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (206) 은 이산 코사인 변환 (DCT), 방향성 변환, Karhunen-Loeve 변환 (KLT), 또는 개념적으로 유사한 변환을 잔차 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 대한 다중 변환들, 예를 들어 1 차 변환 및 2 차 변환, 이를 테면 회전 변환을 수행할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 변환들을 적용하지 않는다.
- [0218] 양자화 유닛 (208) 은 양자화된 변환 계수 블록을 생성하기 위해 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 유닛 (208) 은 현재 블록과 연관된 양자화 파라미터 (QP) 값에 따라 변환 계수 블록의 변환 계수들을 양자화할 수도 있다. 비디오 인코더 (202) 는 (예를 들어, 모드 선택 유닛 (202) 을 통해) CU 와 연관된 QP 값을 조정함으로써 현재 블록과 연관된 계수 블록들에 적용된 양자화도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서 양자화된 변환 계수들은 변환 프로세싱 유닛 (206) 에 의해 생성된 원래의 변환 계수들보다 더 낮은 정밀도를 가질 수도 있다.
- [0219] 역 양자화 유닛 (210) 및 역 변환 프로세싱 유닛 (212) 은 각각 양자화된 변환 계수 블록에 역 양자화 및 역 변환들을 적용하여, 변환 계수 블록으로부터 잔차 블록을 재구성할 수도 있다. 재구성 유닛 (214) 은 모드 선택 유닛 (202) 에 의해 생성된 예측 블록 및 재구성된 잔차 블록에 기초하여 (잠재적으로 어느 정도의 왜곡을 가짐에도 불구하고) 현재 블록에 대응하는 재구성된 블록을 생성할 수도 있다. 예를 들어, 재구성 유닛 (214) 은 재구성된 잔차 블록의 샘플들을, 모드 선택 유닛 (202) 에 의해 생성된 예측 블록으로부터의 대응하는 샘플들에 가산하여 재구성된 블록을 생성할 수도 있다.
- [0220] 필터 유닛 (216) 은 재구성된 블록에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (216) 은 CU들의 에지들을 따라 블록화 아티팩트들 (blockiness artifacts) 을 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (216) 의 동작들은 일부 예들에서 스킵될 수도 있다.
- [0221] 비디오 인코더 (200) 는 DPB (218) 에 재구성된 블록들을 저장한다. 예를 들어, 필터 유닛 (216) 의 동작들이 필요하지 않은 예들에서, 재구성 유닛 (214) 은 재구성된 블록들을 DPB (218) 에 저장할 수도 있다. 필터 유닛 (216) 의 동작들이 필요한 예들에서, 필터 유닛 (216) 은 필터링된 재구성된 블록들을 DPB (218) 에 저장할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 재구성된 (및 잠재적으로 필터링된) 블록들로부터 형성된 DPB (218) 로부터 참조 화상을 취출하여, 후속 인코딩된 화상들의 블록들을 인터-예측할 수

도 있다. 또한, 인트라-예측 유닛 (226) 은 현재 화상에서의 다른 블록들을 인트라-예측하기 위해 현재 화상의 DPB (218) 에서 재구성된 블록들을 사용할 수도 있다.

[0222] 일반적으로, 엔트로피 인코딩 유닛 (220) 은 비디오 인코더 (200) 의 다른 기능 컴포넌트들로부터 수신된 구문 요소들을 엔트로피 인코딩할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 양자화 유닛 (208) 으로부터 양자화된 변환 계수 블록들을 엔트로피 인코딩할 수도 있다. 다른 예로서, 엔트로피 인코딩 유닛 (220) 은 모드 선택 유닛 (202) 으로부터 예측 구문 요소들 (예를 들어, 인트라-예측을 위한 모션 정보 또는 인트라-예측을 위한 인트라-모드 정보) 을 엔트로피 인코딩할 수도 있다. 엔트로피 인코딩 유닛 (220) 은 엔트로피-인코딩된 데이터를 생성하기 위해, 비디오 데이터의 다른 예인, 구문 요소들에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 컨텍스트-적응적 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 구문 기반 컨텍스트-적응적 이진 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-골롬 인코딩 동작, 또는 다른 타입의 엔트로피 인코딩 동작을 데이터에 대해 수행할 수도 있다. 일부 예들에서, 엔트로피 인코딩 유닛 (220) 은 구문 요소들이 엔트로피 인코딩되지 않는 바이패스 모드에서 동작할 수도 있다.

[0223] 비디오 인코더 (200) 는 화상 또는 슬라이스의 블록들을 재구성하는 데 필요한 엔트로피 인코딩된 구문 요소들을 포함하는 비트스트림을 출력할 수도 있다. 특히, 엔트로피 인코딩 유닛 (220) 은 비트스트림을 출력할 수도 있다.

[0224] 위에 설명된 동작들은 블록에 대하여 설명된다. 그러한 설명은 루마 코딩 블록 및/또는 크로마 코딩 블록들을 위한 동작들이므로 이해되어야 한다. 위에 설명된 바와 같이, 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 CU 의 루마 및 크로마 성분들이다. 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 PU 의 루마 및 크로마 성분들이다.

[0225] 일부 예들에서, 루마 코딩 블록에 대해 수행되는 동작들은 크로마 코딩 블록들에 대해 반복될 필요가 없다. 하나의 예로서, 루마 코딩 블록을 위한 모션 벡터 (MV) 및 참조 화상을 식별하기 위한 동작들이 크로마 블록들을 위한 MV 및 참조 화상을 식별하기 위해 반복될 필요는 없다. 오히려, 루마 코딩 블록을 위한 MV 는 크로마 블록들을 위한 MV 를 결정하기 위해 스케일링될 수도 있고, 참조 화상은 동일할 수도 있다. 다른 예로서, 인트라-예측 프로세스는 루마 코딩 블록들 및 크로마 코딩 블록들에 대해 동일할 수도 있다.

[0226] 도 3 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더 (300) 를 예시하는 블록 도이다. 도 3 은 설명의 목적으로 제공되고 본 개시에 폭넓게 예시되고 설명된 바와 같은 기술에 대해 한정하지 않는다. 설명의 목적으로, 본 개시는 H.266/VVC, JEM, 및 HEVC 의 기술에 따라 비디오 디코더 (300) 를 설명한다. 그러나, 본 개시의 기술은 다른 비디오 코딩 표준들로 구성되는 비디오 코딩 디바이스들에 의해 수행될 수도 있다.

[0227] 도 3 의 예에서, 비디오 디코더 (300) 는, 코딩된 화상 버퍼 (CPB) 메모리 (320), 엔트로피 디코딩 유닛 (302), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 재구성 유닛 (310), 필터 유닛 (312), 및 디코딩된 화상 버퍼 (DPB) (314) 를 포함한다. 예측 프로세싱 유닛 (304) 은 모션 보상 유닛 (316) 및 인트라 예측 프로세싱 유닛 (318) 을 포함한다. 예측 프로세싱 유닛 (304) 은 다른 예측 모드들에 따라 예측을 수행하기 위해 추가 유닛들을 포함할 수도 있다. 예를 들어, 예측 프로세싱 유닛 (304) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 보상 유닛 (316) 의 일부를 형성할 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다. 다른 예들에서, 비디오 디코더 (300) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0228] CPB 메모리 (320) 는, 비디오 디코더 (300) 의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. CPB 메모리 (320) 에 저장된 비디오 데이터는, 예를 들어, 컴퓨터 판독가능 매체 (110) (도 1) 로부터 획득될 수도 있다. CPB 메모리 (320) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터 (예를 들어, 구문 요소) 를 저장하는 CPB 를 포함할 수도 있다. 또한, CPB 메모리 (320) 는 비디오 디코더 (300) 의 다양한 유닛들로부터의 출력들을 나타내는 일시적 데이터와 같은, 코딩된 화상의 구문 요소외의 비디오 데이터를 저장할 수도 있다. DPB (314) 는 일반적으로, 인코딩된 비디오 비트스트림의 후속 데이터 또는 화상들을 디코딩할 때 참조 비디오 데이터로서 비디오 디코더 (300) 가 출력 및/또는 사용할 수도 있는 디코딩된 화상들을 저장한다. CPB 메모리 (320) 및 DPB (314) 는 동기식 동적 랜덤 액세스 메모리 (SDRAM) 를 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 어느 것에 의해 형성될 수도 있다.

CPB 메모리 (320) 및 DPB (314) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, CPB 메모리 (320) 는 비디오 디코더 (300) 의 다른 컴포넌트들과 온-칩되거나, 또는 그 컴포넌트들에 대해 오프-칩될 수도 있다.

[0229] 추가적으로 또는 대안적으로, 일부 예들에서, 비디오 디코더 (300) 는 메모리 (120) (도 1) 로부터 코딩된 비디오 데이터를 추출할 수도 있다. 즉, 메모리 (120) 는 CPB 메모리 (320) 로 위에서 논의된 바와 같이 데이터를 저장할 수도 있다. 마찬가지로, 메모리 (120) 는 비디오 디코더 (300) 의 기능성의 일부 또는 전부가 비디오 디코더 (300) 의 프로세싱 회로에 의해 실행되도록 소프트웨어에서 구현될 때, 비디오 디코더 (300) 에 의해 실행될 명령들을 저장할 수도 있다.

[0230] 도 3 에 도시된 다양한 유닛들은 비디오 디코더 (300) 에 의해 수행되는 동작들의 이해를 돕기 위해 예시된다. 이 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 도 2 와 유사하게, 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하고, 수행될 수 있는 동작들에 대해 미리설정된다. 프로그래밍가능 회로들은 다양한 태스크들을 수행하도록 프로그래밍될 수 있는 회로들을 지칭하고, 수행될 수 있는 동작들에 있어서 유연한 기능성을 제공한다. 예를 들어, 프로그래밍가능 회로들은, 프로그래밍가능 회로들로 하여금, 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 (예를 들어, 파라미터들을 수신하거나 또는 파라미터들을 출력하기 위해) 소프트웨어 명령들을 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작들의 타입들은 일반적으로 불변이다. 일부 예들에서, 유닛들 중 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그래밍가능) 일 수도 있고, 일부 예들에서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.

[0231] 비디오 디코더 (300) 는 프로그래밍가능 회로들로부터 형성된, ALU들, EFU들, 디지털 회로들, 아날로그 회로들, 및/또는 프로그래밍가능 코어들을 포함할 수도 있다. 비디오 디코더 (300) 의 동작들이 프로그래밍가능 회로들 상에서 실행하는 소프트웨어에 의해 수행되는 예들에서, 온-칩 또는 오프-칩 메모리는 비디오 디코더 (300) 가 수신하고 실행하는 소프트웨어의 명령들 (예를 들어, 오브젝트 코드) 을 저장할 수도 있다.

[0232] 엔트로피 디코딩 유닛 (302) 은 인코딩된 비디오 데이터를 CPB 로부터 수신하고, 비디오 데이터를 엔트로피 디코딩하여 구문 요소들을 재생할 수도 있다. 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 재구성 유닛 (310), 및 필터 유닛 (312) 은 비트스트림으로부터 추출된 구문 요소들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0233] 일반적으로, 비디오 디코더 (300) 는 블록 별 (block-by-block) 기준으로 화상을 재구성한다. 비디오 디코더 (300) 는 개별적으로 각각의 블록에 대해 재구성 동작을 수행할 수도 있다 (여기서 현재 재구성되는, 즉 디코딩되는 블록은 "현재 블록" 으로 지칭될 수도 있음).

[0234] 엔트로피 디코딩 유닛 (302) 은 양자화 파라미터 (QP) 및/또는 변환 모드 표시(들)와 같은 변환 정보 뿐만 아니라, 양자화된 변환 계수 블록의 양자화된 변환 계수들을 정의하는 구문 요소들을 엔트로피 디코딩할 수도 있다. 역 양자화 유닛 (306) 은 양자화된 변환 계수 블록과 연관된 QP 를 사용하여, 양자화도 그리고, 마찬가지로, 역 양자화 유닛 (306) 이 적용할 역 양자화도를 결정할 수도 있다. 역 양자화 유닛 (306) 은 예를 들어, 양자화된 변환 계수들을 역 양자화하기 위해 비트단위 좌측-시프트 동작을 수행할 수도 있다. 이에 의해, 역 양자화 유닛 (306) 은 변환 계수들을 포함하는 변환 계수 블록을 형성할 수도 있다.

[0235] 역 양자화 유닛 (306) 이 변환 계수 블록을 형성한 후, 역 변환 프로세싱 유닛 (308) 은 현재 블록과 연관된 잔차 블록을 생성하기 위해 변환 계수 블록에 하나 이상의 역 변환들을 적용할 수도 있다. 예를 들어, 역 변환 프로세싱 유닛 (308) 은 역 DCT, 역 정수 변환, 역 Karhunen-Loeve 변환 (KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 계수 블록에 적용할 수도 있다.

[0236] 또한, 예측 프로세싱 유닛 (304) 은 엔트로피 디코딩 유닛 (302) 에 의해 엔트로피 디코딩된 예측 정보 구문 요소들에 따라 예측 블록을 생성한다. 예를 들어, 예측 정보 구문 요소들이 현재 블록이 인터-예측됨을 표시하면, 모션 보상 유닛 (316) 은 예측 블록을 생성할 수도 있다. 이 경우에, 예측 정보 구문 요소들은 참조 블록을 추출할 DPB (314) 에서의 참조 화상 뿐만 아니라 현재 화상에서의 현재 블록의 위치에 대한 참조 화상에서의 참조 블록의 위치를 식별하는 모션 벡터를 표시할 수도 있다. 모션 보상 유닛 (316) 은 일반적으로 모션 보상 유닛 (224) (도 2) 에 대하여 설명된 것과 실질적으로 유사한 방식으로 인터-예측 프로세스를 수행할 수도 있다.

[0237] 본 개시의 기술에 따르면, 모션 보상 유닛 (316) 은 본 개시에 설명된 블록을 코딩할 때 인터-예측 및 모션 벡

터 예측 기술을 수행할 수도 있다. 예를 들어, 아래에서 더 상세히 설명되는 바와 같이, 모션 보상 유닛 (316) 은 본 개시에서 설명된 예시적인 기술을 이용하여 모션 벡터 예측자 리스트를 만들도록 구성될 수도 있다.

- [0238] 전술한 바와 같이, 일부 예들에서, 블록을 위한 모션 벡터 예측자 리스트는 그 블록 및 공간적으로 이웃하는 블록 및/또는 콜로케이트된 블록을 포함하는 화상의 코딩 순서에 기초할 수도 있다. 가령, 하나의 예에서, 5 개의 공간적으로 이웃하는 블록은 좌측에서 우측 코딩 순서에 대해 도 8a에 예시된 것들일 수도 있고 5 개의 공간적으로 이웃하는 블록은 우측에서 좌측 코딩 순서에 대해 도 8b에 예시된 것일 수도 있다.
- [0239] 가령, 좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 모션 보상 유닛 (316) 은 제 1 모션 벡터 예측자 리스트를 구성할 수도 있다. 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 제 1 블록에 대한 좌측 이웃하는 블록 (예를 들어, 도 8a 의 블록 0 (802)) 의 모션 벡터 정보에 기초할 수도 있다. 우측에서 좌측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 모션 보상 유닛 (316) 은 제 2 모션 벡터 예측자 리스트를 만들 수도 있다. 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 제 2 블록에 대한 우측 이웃하는 블록 (예를 들어, 도 8b 의 블록 0 (818)) 의 모션 벡터 정보에 기초할 수도 있다.
- [0240] 모션 보상 유닛 (316) 은 병렬적으로, 복수의 (전부를 포함) 공간적으로 이웃하는 블록에 대해, (하나의 예로서) 디코딩된 화상 버퍼 (314) 에 저장된 모션 벡터 정보를 체크할 수도 있다. 즉, 모션 보상 유닛 (316) 이 특정 공간적으로 이웃하는 블록을 위한 모션 벡터 정보를 체크하는지 여부는 임의의 다른 블록의 모션 벡터 정보에 의존하지 않을 수도 있다.
- [0241] 공간적으로 이웃하는 블록들의 모션 벡터 정보의 체크에 기초하여, 모션 보상 유닛 (316) 은 모션 벡터 정보의 이용 가능성에 기초한 순서로 모션 벡터 정보를 가지는 모션 벡터 예측자 리스트를 만들 수도 있다. 좌측에서 우측 코딩 순서를 위한 모션 벡터 예측자 리스트에서 모션 벡터 정보의 순서는 다음과 같을 수도 있다: 좌측 이웃하는 블록 (예를 들어, 도 8a의 블록 0 (802)) 의 모션 벡터 정보, 상측 이웃하는 블록 (예를 들어, 도 8a 의 블록 1 (804)) 의 모션 벡터 정보, 우상측 이웃하는 블록 (예를 들어, 도 8a의 블록 2 (806)) 의 모션 벡터 정보, 좌하측 이웃하는 블록 (예를 들어, 도 8a의 블록 3 (808)) 의 모션 벡터 정보, 및 좌상측 이웃하는 블록 (예를 들어, 도 8a의 블록 4 (810)) 의 모션 벡터 정보. 우측에서 좌측 코딩 순서를 위한 모션 벡터 예측자 리스트에서 모션 벡터 정보의 순서는 다음과 같을 수도 있다: 우측 이웃하는 블록 (예를 들어, 도 8b의 블록 0 (818)) 의 모션 벡터 정보, 상측 이웃하는 블록 (예를 들어, 도 8b의 블록 1 (820)) 의 모션 벡터 정보, 좌상측 이웃하는 블록 (예를 들어, 도 8b 의 블록 2 (822)) 의 모션 벡터 정보, 우하측 이웃하는 블록 (예를 들어, 도 8b의 블록 3 (824)) 의 모션 벡터 정보, 및 우상측 이웃하는 블록 (예를 들어, 도 8b의 블록 4 (826)) 의 모션 벡터 정보.
- [0242] 위의 순서는 모션 벡터 정보의 이용 가능성에 기초한다. 특정 블록에 대해 모션 벡터 정보가 이용 가능하지 않은 경우, 그 모션 벡터 정보는 모션 벡터 예측자 리스트에 추가되지 않는다. 일부 예들에서, 다음 이용 가능한 모션 벡터 정보는 모션 벡터 정보가 없는 블록에 대해 예약된 공간에 포함될 수도 있다. 예를 들어, 도 8a 의 블록 1 (804) 를 위한 모션 벡터 정보는 모션 벡터 예측자 리스트에서 두번째 엔트리일 수도 있고, 도 8a 의 블록 2 (806) 를 위한 모션 벡터 정보는 모션 벡터 예측자 리스트에서 세번째 엔트리일 수도 있다. 도 8a 의 블록 1 (804) 를 위한 모션 벡터 정보가 이용 가능하지 않고, 도 8a 의 블록 2 (806) 를 위한 모션 벡터 정보가 이용 가능하다면, 블록 2 (806) 를 위한 모션 벡터 정보는 모션 벡터 예측자 리스트의 두번째 엔트리일 수도 있다.
- [0243] 위의 예에서, 모션 보상 유닛 (316) 은 공간적으로 이웃하는 블록에 기초하여 제 1 모션 벡터 예측자 리스트 및 제 2 모션 벡터 예측자 리스트를 결정하였다. 일부 예들에서, 모션 보상 유닛 (316) 은 제 1 모션 벡터 예측자 리스트 및 제 2 모션 벡터 예측자 리스트에 콜로케이트된 블록의, (하나의 예로서) 디코딩된 화상 버퍼 (314) 에 저장된, 모션 벡터 정보를 포함할 수도 있다.
- [0244] 일례로서, 모션 보상 유닛 (316) 은, 제 1 화상 외의 화상에 위치한 임의의 다른 콜로케이트된 블록에서의 모션 벡터 정보가 이용 가능한지 여부를 결정하기 전에 제 1 화상의 화상에 위치한 중앙 콜로케이트된 블록 (예를 들어, 도 8a 의 블록 5 (812)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정할 수도 있고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 모션 보상 유닛 (316) 은 제 1 모션 벡터 예측자 리스트에 중앙 콜로케이트된 블록을 위한 모션 벡터 정보를 추가할 수도 있다.
- [0245] 일부 예들에서, 모션 보상 유닛 (316) 은 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션

벡터 정보가 이용 가능하지 않다고 결정할 수도 있다. 이러한 예들에서, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 모션 보상 유닛 (316)은 제 1 화상 외의 화상에 위치한 하단 콜로케이트된 블록 (예를 들어, 도 8a의 블록 6 (814))을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정할 수도 있고, 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 모션 보상 유닛 (316)은 하단 콜로케이트된 블록을 위한 모션 벡터 정보를 제 1 모션 벡터 예측자 리스트에 추가할 수도 있다.

[0246] 일부 예들에서, 모션 보상 유닛 (316)은 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정할 수도 있다. 일부 경우에서, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 모션 보상 유닛 (316)은 제 1 화상 외의 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정할 수도 있다. 이러한 경우에, 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후, 모션 보상 유닛 (316)은 제 1 화상 외의 화상에 위치한 우측 콜로케이트된 블록 (예를 들어, 도 8a의 블록 7 (816))을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정할 수도 있고, 우측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 모션 보상 유닛 (316)은 우측 콜로케이트된 블록을 위한 모션 벡터 정보를 제 1 모션 벡터 예측자 리스트에 추가할 수도 있다.

[0247] 모션 보상 유닛 (316)은 제 2 모션 벡터 예측자 리스트를 만들기 위해 콜로케이트된 블록을 위한 모션 벡터 정보를 추가하기 위한 유사한 동작을 수행할 수도 있다. 그러나, 제 2 모션 벡터 예측자 리스트에 대해, 모션 보상 유닛 (316)은 도 8b의 중앙 콜로케이트된 블록 5 (828), 하단 콜로케이트된 블록 6 (830) 및 좌측 콜로케이트된 블록 7 (832)의 이용 가능성에 기초하여 모션 벡터 정보를 추가할 수도 있다.

[0248] 모션 벡터 예측자 리스트를 만들 때, 모션 보상 유닛 (316)은 일부 예들에서 블록의 크기에 기초하여 모션 벡터 예측자 리스트를 만들 수도 있다. 예를 들어, 모션 보상 유닛 (316)은 제 1 블록이 임계 크기보다 큰 크기를 갖는 것에 기초하여 제 1 최대 크기를 갖는 제 1 모션 벡터 예측자 리스트를 만들 수도 있다. 모션 보상 유닛 (316)은 제 1 화상에서 제 3 블록에 대해, 제 3 블록이 임계 크기보다 작은 크기를 갖는 것에 기초하여, 제 1 최대 크기보다 작은 제 2 최대 크기를 갖는, 제 3 모션 벡터 예측자 리스트를 만들 수도 있다. 하나의 예로서, 제 1 최대 크기는 6과 같고, 제 2 최대 크기는 4와 같고, 임계 크기는 $N \times 4$ 또는 $4 \times N$ 와 같고, 여기서 N 은 8보다 작거나 같다.

[0249] 모션 보상 유닛 (316)은, 재구성 유닛 (310)과 함께, 제 1 블록 및 제 2 블록을 디코딩하도록 구성될 수도 있다. 예를 들어, 모션 보상 유닛 (316)은 제 1 모션 벡터 예측자 리스트에서의 엔트리를 나타내는 정보를 수신하고 그 엔트리에 기초하여 제 1 블록을 위한 모션 벡터를 결정한다. 모션 보상 유닛 (316)은 모션 벡터에 기초하여 제 1 블록을 위한 디코딩된 화상 버퍼 (314)에 저장된 예측 블록을 결정할 수도 있다. 재구성 유닛 (310)은 예측 블록 및 그 예측 블록과 제 1 블록 사이의 차이를 나타내는 잔차 데이터를 나타내는 수신된 정보에 기초하여 제 1 블록을 재구성하도록 구성될 수도 있다. 모션 보상 유닛 (316) 및 재구성 유닛 (310)은, 제 2 블록을 유사하게 디코딩하도록 구성될 수도 있다.

[0250] 다른 예로서, 예측 정보 구문 요소가 현재 블록이 인트라-예측됨을 표시하면, 인트라-예측 유닛 (318)은 예측 정보 구문 요소에 의해 표시된 인트라-예측 모드에 따라 예측 블록을 생성할 수도 있다. 다시, 인트라-예측 유닛 (318)은 일반적으로 인트라-예측 유닛 (226) (도 2)에 대하여 설명된 것과 실질적으로 유사한 방식으로 인트라-예측 프로세스를 수행할 수도 있다. 인트라-예측 유닛 (318)은 DPB (314)로부터 현재 블록을 위한 이웃하는 샘플들의 데이터를 추출할 수도 있다.

[0251] 재구성 유닛 (310)은 예측 블록 및 잔차 블록을 사용하여 현재 블록을 재구성할 수도 있다. 예를 들어, 재구성 유닛 (310)은 잔차 블록의 샘플들을 예측 블록의 대응하는 샘플들에 더하여 현재 블록을 재구성할 수도 있다.

[0252] 필터 유닛 (312)은 재구성된 블록들에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (312)은 재구성된 블록들의 에지들을 따라 블록화 아티팩트를 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (312)의 동작들이 모든 예들에서 반드시 수행되는 것은 아니다.

[0253] 비디오 디코더 (300)는 DPB (314)에 재구성된 블록들을 저장할 수도 있다. 위에 논의된 바와 같이, DPB (314)는 예측 프로세싱 유닛 (304)에, 인트라-예측을 위한 현재 화상 및 후속 모션 보상을 위한 이전에 디코딩된 화상들의 샘플들과 같은 참조 정보를 제공할 수도 있다. 또한, 비디오 디코더 (300)는 도 1의 디스플레이 디바이스 (118)와 같은 디스플레이 디바이스 상에의 후속 프리젠테이션을 위해 DPB로부터 디코딩된 화

상들을 출력할 수도 있다.

- [0254] 도 9 는 데이터를 코딩하는 예시적인 방법을 나타내는 플로우차트이다. 설명의 편의를 위해, 도 9의 예는 프로세싱 회로와 관련하여 설명된다. 프로세싱 회로의 일례는 비디오 인코더 (200) 이다. 프로세싱 회로의 다른 예는 비디오 디코더 (300) 이다. 프로세싱 회로는 고정 기능 회로, 프로그램 가능 회로 또는 고정 기능과 프로그램 가능 회로의 조합일 수도 있다.
- [0255] 하나 이상의 예들에서, 메모리는 공간적으로 이웃하는 블록들 및 콜로케이트된 블록들의 모션 벡터 정보와 같은 모션 벡터 정보를 저장하도록 구성될 수도 있다. 일례로서, 메모리는 비디오 데이터 메모리 (230), 디코딩된 화상 버퍼 (218) 또는 메모리 (106) 일 수도 있다. 다른 예로서, 메모리는, 디코딩된 화상 버퍼 (314) 또는 메모리 (120) 일 수도 있다.
- [0256] 좌측에서 우측 코딩 순서를 갖는 제 1 화상에서, 제 1 블록에 대해, 프로세싱 회로는 제 1 모션 벡터 예측자 리스트 (400) 를 만들도록 구성될 수도 있다. 제 1 모션 벡터 예측자 리스트에서의 제 1 엔트리는 메모리에 저장된 제 1 블록에 대한 좌측 이웃하는 블록의 모션 벡터 정보에 기초할 수도 있다. 좌측 이웃하는 블록의 일례는 도 8a 의 블록 0 (802) 이고, 여기서, 도 8a 의 블록 0 (802) 는 도 8a 의 현재 블록 (800) 의 좌측에 위치하고 도 8a 의 현재 블록 (800) 의 좌측 경계를 공유한다 (예를 들어, 좌측 이웃하는 블록은 제 1 블록의 하단 경계와 동일한 하단 경계를 갖는 제 1 블록의 좌측 블록이다). 일부 예들에서, 프로세싱 회로는 모션 벡터 정보 이용 가능성에 기초하여 다음 순서로 모션 벡터 정보를 갖는 제 1 모션 벡터 예측자 리스트를 만들도록 구성될 수도 있다: 좌측 이웃하는 블록 (예를 들어, 블록 0 (802)) 의 모션 벡터 정보, 상측 이웃하는 블록 (예를 들어, 블록 1 (804)) 의 모션 벡터 정보, 우상측 이웃하는 블록 (예를 들어, 블록 2 (806)) 의 모션 벡터 정보, 좌하측 이웃하는 블록 (예를 들어, 블록 3 (808)) 의 모션 벡터 정보 및 좌상측 좌측 이웃하는 블록 (예를 들어, 블록 4 (810)) 의 모션 벡터 정보.
- [0257] 일부 예들에서, 제 1 모션 벡터 예측자 리스트를 만들기 위해, 프로세싱 회로는 제 1 화상 외의 화상에 위치한 임의의 다른 콜로케이트된 블록에서의 모션 벡터 정보가 이용 가능한지 여부를 결정하기 전에 제 1 화상의 화상에 위치한 중앙 콜로케이트된 블록 (예를 들어, 블록 812) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보를 제 1 모션 벡터 예측자 리스트에 추가하도록 구성될 수도 있다. 일부 예들에서, 제 1 모션 벡터 예측자 리스트를 만들기 위해, 프로세싱 회로는, 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후에, 제 1 화상 외의 화상에 위치한 하단 콜로케이트된 블록 (예를 들어, 도 8a의 현재 블록 (800) 의 우측에 가장 가깝고 하단인 블록 6 (814)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고, 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 하단 콜로케이트된 블록을 위한 모션 벡터 정보를 제 1 모션 벡터 예측자 리스트에 추가하도록 구성된다. 일부 예들에서, 하단 콜로케이트된 블록 (예를 들어, 블록 6 (814)) 의 우측 경계는 제 1 화상에서 제 1 블록의 우측 경계와 제 1 화상 외의 화상에서 동일한 위치에 있다.
- [0258] 일부 예들에서, 제 1 모션 벡터 예측자 리스트를 만들기 위해, 프로세싱 회로는, 제 1 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후에, 제 1 화상 외의 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고, 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후에, 제 1 화상 외의 화상에 위치한 우측 콜로케이트된 블록 (예를 들어, 도 8a의 현재 블록 (800) 의 하단에 가장 가깝고 우측인 블록 7 (816)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고, 우측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 우측 콜로케이트된 블록을 위한 모션 벡터 정보를 제 1 모션 벡터 예측자 리스트에 추가하도록 구성된다. 일부 예들에서, 우측 콜로케이트된 블록 (예를 들어, 블록 7 (816)) 의 하단 경계는 제 1 화상에서 제 1 블록의 하단 경계와 제 1 화상 외의 화상에서 동일한 위치에 있다.
- [0259] 프로세싱 회로는 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서 제 1 블록을 코딩하도록 구성될 수도 있다 (402). 일례로서, 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서 제 1 블록을 코딩하기 위해, 프로세싱 회로는 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서 제 1 블록을 디코딩하도록 구성될 수도 있다. 제 1 블록을 디코딩하기 위해, 프로세싱 회로는 제 1 모션 벡터 예측자 리스트에서의 엔트리를 나타내는 정보를 수신하고, 엔트리에 기초하여 제 1 블록을 위한 모션 벡터를 결정하고, 모션 벡터에 기초하

여 제 1 블록을 위한 예측 블록을 결정하고, 예측 블록 및 그 예측 블록과 제 1 블록 사이의 차이를 나타내는 잔차 데이터를 나타내는 수신된 정보에 기초하여 제 1 블록을 재구성하도록 구성될 수도 있다. 다른 예로서, 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서 제 1 블록을 코딩하기 위해, 프로세싱 회로는 제 1 모션 벡터 예측자 리스트에 기초하여 제 1 화상에서 제 1 블록을 인코딩하도록 구성될 수도 있다. 제 1 블록을 인코딩하기 위해, 프로세싱 회로는 제 1 블록을 위한 예측 블록을 결정하고, 예측 블록을 식별하는 제 1 블록을 위한 모션 벡터를 결정하고, 모션 벡터에 기초하여 제 1 모션 벡터 예측자 리스트에서의 엔트리를 결정하고, 엔트리를 나타내는 정보를 시그널링하고, 제 1 블록과 예측 블록 사이의 차이를 나타내는 잔차 데이터를 나타내는 정보를 시그널링하도록 구성될 수도 있다.

[0260] 좌측에서 우측 코딩 순서를 갖는 제 2 화상에서, 제 2 블록에 대해, 프로세싱 회로는 제 2 모션 벡터 예측자 리스트를 만들도록 구성될 수도 있다 (404). 제 2 모션 벡터 예측자 리스트에서의 제 1 엔트리는 메모리에 저장된 제 2 블록에 대한 우측 이웃하는 블록의 모션 벡터 정보에 기초할 수도 있다. 우측 이웃하는 블록의 일례는 도 8b 의 블록 0 (818) 이고, 여기서, 도 8b 의 블록 0 (818) 는 도 8b 의 현재 블록 (800) 의 우측에 위치하고 도 8b 의 현재 블록 (800) 의 우측 경계를 공유한다 (예를 들어, 우측 이웃하는 블록은 제 2 블록의 하단 경계와 동일한 하단 경계를 갖는 제 2 블록의 우측 블록이다). 일부 예들에서, 프로세싱 회로는 모션 벡터 정보 이용 가능성에 기초하여 다음 순서로 모션 벡터 정보를 갖는 제 2 모션 벡터 예측자 리스트를 만들도록 구성될 수도 있다: 우측 이웃하는 블록 (예를 들어, 블록 0 (818)) 의 모션 벡터 정보, 상측 이웃하는 블록 (예를 들어, 블록 1, (820)) 의 모션 벡터 정보, 좌상측 이웃하는 블록 (예를 들어, 블록 2 (822)) 의 모션 벡터 정보, 우하측 이웃하는 블록 (예를 들어, 블록 3 (824)) 의 모션 벡터 정보 및 우상측 이웃하는 블록 (예를 들어, 블록 4 (826)) 의 모션 벡터 정보.

[0261] 일부 예들에서, 제 2 모션 벡터 예측자 리스트를 만들기 위해, 프로세싱 회로는 제 2 화상 외의 화상에 위치한 임의의 다른 콜로케이트된 블록에서의 모션 벡터 정보가 이용 가능한지 여부를 결정하기 전에 제 2 화상외의 화상에 위치한 중앙 콜로케이트된 블록 (예를 들어, 블록 5 (828)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보를 제 2 모션 벡터 예측자 리스트에 추가하도록 구성될 수도 있다. 일부 예들에서, 제 2 모션 벡터 예측자 리스트를 만들기 위해, 프로세싱 회로는, 제 2 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후에, 제 2 화상 외의 화상에 위치한 하단 콜로케이트된 블록 (예를 들어, 도 8b의 현재 블록 (800) 의 좌측에 가장 가깝고 하단인 블록 6 (830)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고, 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 하단 콜로케이트된 블록을 위한 모션 벡터 정보를 제 2 모션 벡터 예측자 리스트에 추가하도록 구성된다. 일부 예들에서, 하단 콜로케이트된 블록 (예를 들어, 블록 6 (830)) 의 좌측 경계는 제 2 화상에서 제 2 블록의 좌측 경계와 제 2 화상 외의 화상에서 동일한 위치에 있다.

[0262] 일부 예들에서, 제 2 모션 벡터 예측자 리스트를 만들기 위해, 프로세싱 회로는, 제 2 화상 외의 화상에 위치한 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고, 중앙 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후에, 제 2 화상 외의 화상에 위치한 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정하고, 하단 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능하지 않다고 결정한 후에, 제 2 화상 외의 화상에 위치한 좌측 콜로케이트된 블록 (예를 들어, 도 8b의 현재 블록 (800) 의 하단에 가장 가깝고 좌측인 블록 7 (832)) 을 위한 모션 벡터 정보가 이용 가능한지 여부를 결정하고, 좌측 콜로케이트된 블록을 위한 모션 벡터 정보가 이용 가능한 것에 기초하여, 좌측 콜로케이트된 블록을 위한 모션 벡터 정보를 제 2 모션 벡터 예측자 리스트에 추가하도록 구성된다. 일부 예들에서, 좌측 콜로케이트된 블록의 하단 경계 (예를 들어, 블록 7 (832)) 는 제 2 화상에서 제 2 블록의 하단 경계와 제 2 화상 외의 화상에서 동일한 위치에 있다.

[0263] 프로세싱 회로는 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서 제 2 블록을 코딩하도록 구성될 수도 있다 (406). 일례로서, 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서 제 2 블록을 코딩하기 위해, 프로세싱 회로는 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서 제 2 블록을 디코딩하도록 구성될 수도 있다. 제 2 블록을 디코딩하기 위해, 프로세싱 회로는 제 2 모션 벡터 예측자 리스트에서의 엔트리를 나타내는 정보를 수신하고, 엔트리에 기초하여 제 2 블록을 위한 모션 벡터를 결정하고, 모션 벡터에 기초하여 제 2 블록을 위한 예측 블록을 결정하고, 예측 블록 및 그 예측 블록과 제 2 블록 사이의 차이를 나타내는 잔차 데이터를 나타내는 수신된 정보에 기초하여 제 2 블록을 재구성하도록 구성될 수도 있다. 다른

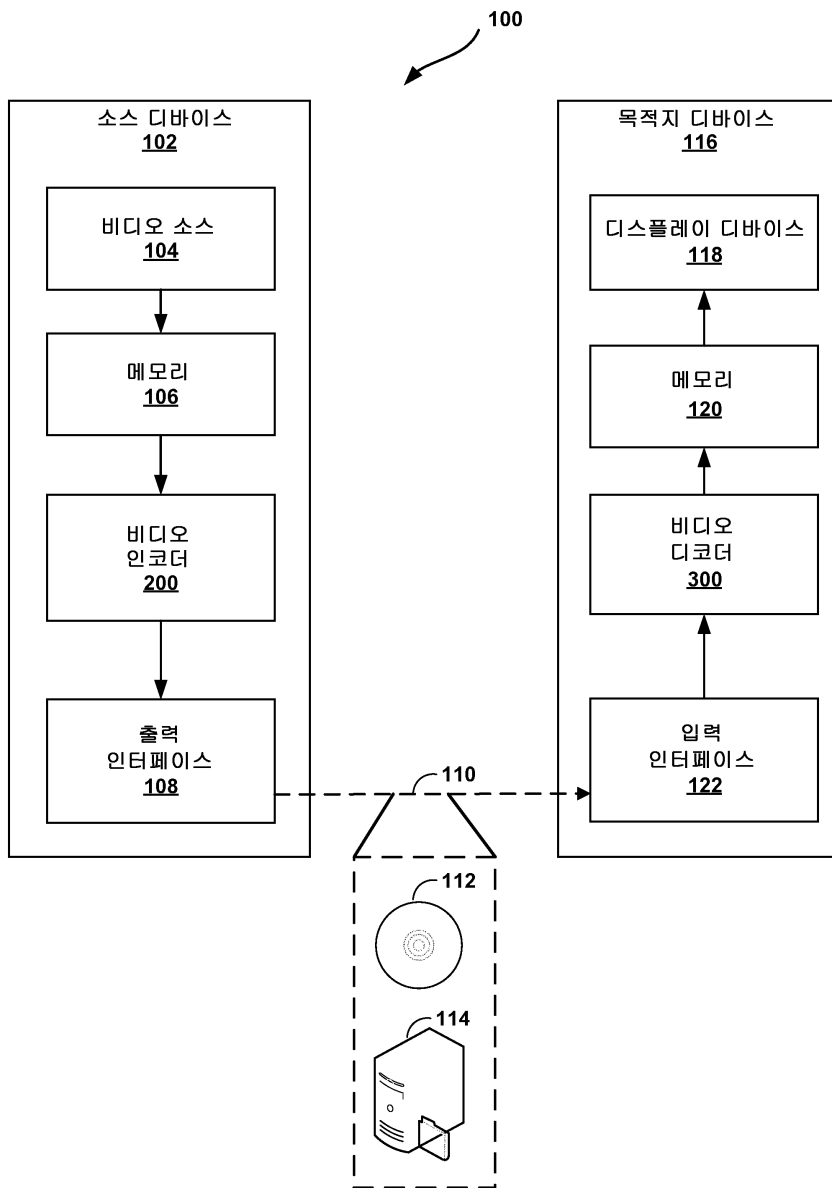
예로서, 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서 제 2 블록을 코딩하기 위해, 프로세싱 회로는 제 2 모션 벡터 예측자 리스트에 기초하여 제 2 화상에서 제 2 블록을 인코딩하도록 구성될 수도 있다. 제 2 블록을 인코딩하기 위해, 프로세싱 회로는 제 2 블록을 위한 예측 블록을 결정하고, 예측 블록을 식별하는 제 2 블록을 위한 모션 벡터를 결정하고, 모션 벡터에 기초하여 제 2 모션 벡터 예측자 리스트에서의 엔트리를 결정하고, 엔트리를 나타내는 정보를 시그널링하고, 제 2 블록과 예측 블록 사이의 차이를 나타내는 잔차 데이터를 나타내는 정보를 시그널링하도록 구성될 수도 있다.

- [0264] 다음은 본 개시에 설명된 임의의 기술과 함께 및/또는 조합하여 수행될 수도 있는 일부 예시적인 기술이다. 다음 예는 일부 예일 뿐이며 제한적이거나 모든 예에서 필요한 것으로 간주해서는 안된다.
- [0265] 예 1. 비디오 데이터의 코딩 방법으로서, 5 개의 공간 후보 및 3 개 이하의 콜로케이트된 시간적 후보를 포함하는 모션 벡터 예측자 리스트를 만드는 단계, 및 모션 벡터 예측자 리스트를 사용하여 비디오 데이터의 블록을 코딩하는 단계를 포함하는, 코딩 방법.
- [0266] 예 2. 예 1에 있어서, 모션 벡터 예측자 리스트를 만드는 단계는 콜로케이트된 시간적 후보의 이용 가능성을 체크하기 전에 공간적 후보의 이용 가능성을 체크하는 단계를 포함하는, 코딩 방법.
- [0267] 예 3. 예 1에 있어서, 5 개의 공간적 후보 및 3 개의 콜로케이트된 시간적 후보는 도 8a에 도시된 패턴으로 위치되는, 코딩 방법.
- [0268] 예 4. 예 1에 있어서, 5 개의 공간적 후보 및 3 개의 콜로케이트된 시간적 후보는 도 8b에 도시된 패턴으로 위치되는, 코딩 방법.
- [0269] 예 5. 예 1에 있어서, 5 개의 공간적 후보 및 3 개의 콜로케이트된 시간적 후보는 도 8c에 도시된 패턴으로 위치되는, 코딩 방법.
- [0270] 예 6. 예 1에 있어서, 블록을 위한 코딩 모드, 블록의 파티셔닝 유형 또는 블록의 코딩 방향 중 하나 이상에 기초하여 5 개의 공간적 후보 및 3 개의 콜로케이트된 시간적 후보를 결정하는 단계를 더 포함하는, 코딩 방법.
- [0271] 예 7. 예 1에 있어서, 모션 벡터 예측자 리스트를 만드는 단계는 콜로케이트된 시간적 후보의 이용 가능성을 체크하는 것과 병행하여 공간적 후보의 이용 가능성을 체크하는 단계를 포함하는, 코딩 방법.
- [0272] 예 8. 예 7에 있어서, 공간적 후보 및 콜로케이트된 시간적 후보의 각각의 위치의 유효성을 결정하는 단계, 및 유효한 것으로 결정된 후보들로부터 모션 정보를 모션 벡터 예측자 리스트에 추가하는 단계를 더 포함하는, 코딩 방법.
- [0273] 예 9. 예 1에 있어서, 모션 벡터 예측자 리스트를 만드는 단계는 비디오 데이터 블록의 크기에 기초하여 모션 벡터 예측자 리스트의 최대 유효 크기를 결정하는 단계 및 최대 유효 크기에 기초하여 모션 벡터 예측자 리스트를 만드는 단계를 포함하는, 코딩 방법.
- [0274] 예 10. 예 1 내지 예 9의 기술을 조합한, 코딩 방법.
- [0275] 예 11. 예 1 내지 예 10 중 어느 한 예에 있어서, 코딩은 디코딩을 포함하는, 코딩 방법.
- [0276] 예 12. 예 1 내지 예 10 중 어느 한 예에 있어서, 코딩은 인코딩을 포함하는, 코딩 방법.
- [0277] 예 13. 비디오 데이터를 코딩하기 위한 디바이스로서, 예 1 내지 예 10 중 어느 한 예의 방법을 수행하기 위한 하나 이상의 수단을 포함하는, 디바이스.
- [0278] 예 14. 예 13에 있어서, 하나 이상의 수단은 회로에서 구현되는 하나 이상의 프로세서를 포함하는, 디바이스.
- [0279] 예 15. 예 13 및 예 14 중 어느 한 예에 있어서, 비디오 데이터를 저장하기 위한 메모리를 더 포함하는, 디바이스.
- [0280] 예 16. 예 13 내지 예 15의 임의의 조합의 디바이스로서,
- [0281] 디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함하는, 디바이스.
- [0282] 예 17. 예 13 내지 예 16의 임의의 조합의 디바이스로서,
- [0283] 디바이스는 카메라, 컴퓨터, 이동 디바이스, 브로드캐스트 수신기 디바이스, 또는 셋탑 박스 중 하나 이상을 포함하는, 디바이스.

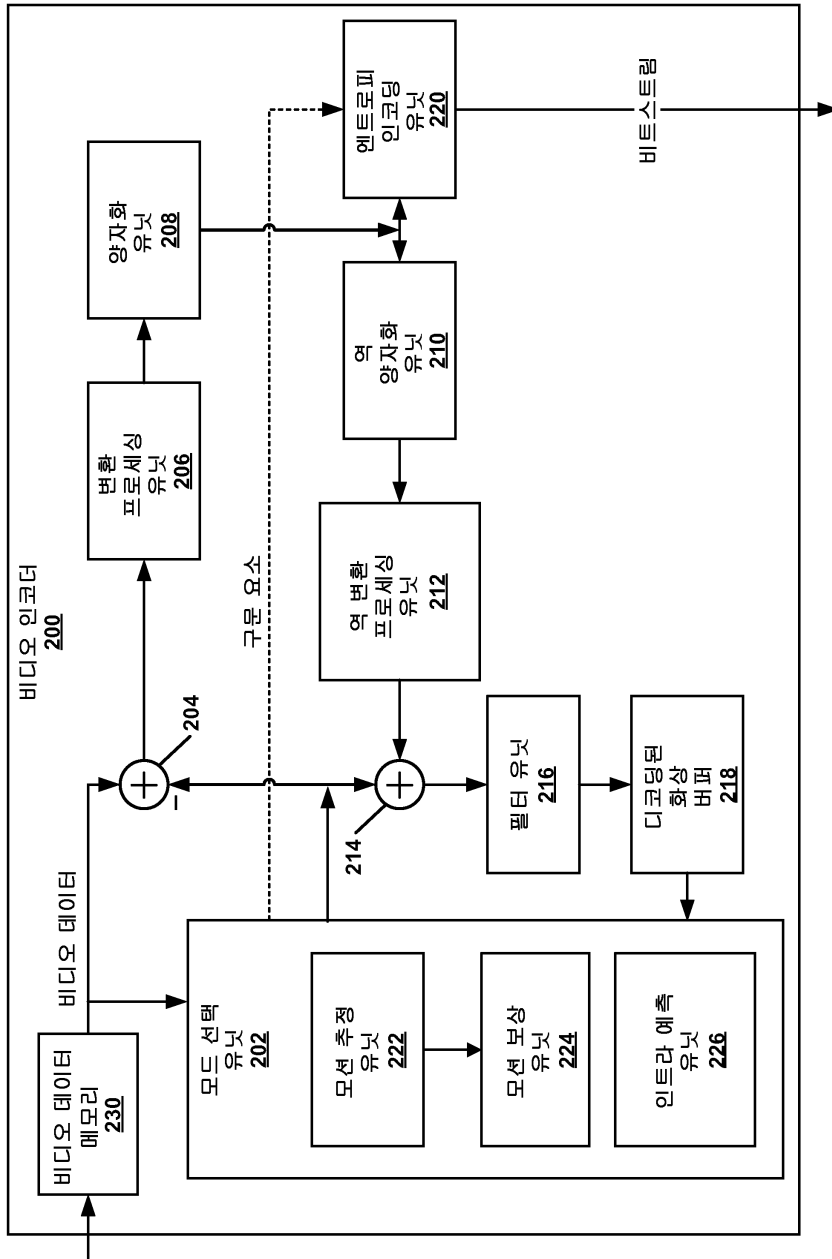
- [0284] 예 18. 예 13 내지 예 17의 임의의 조합의 디바이스로서, 디바이스는 비디오 디코더를 포함하는, 디바이스.
- [0285] 예 19. 예 13 내지 예 18의 임의의 조합의 디바이스로서, 디바이스는 비디오 인코더를 포함하는, 디바이스.
- [0286] 예 20. 명령들이 저장된 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 실행될 경우, 하나 이상의 프로세서들로 하여금 예 1 내지 예 10 중 어느 한 예의 방법을 수행하게 하는, 컴퓨터 판독가능 저장 매체.
- [0287] 예 21. 본 개시에 설명된 기술들의 임의의 조합.
- [0288] 예에 따라, 여기에 기재된 기술 중 어느 것의 특정 행위들 또는 이벤트들이 상이한 시퀀스에서 수행될 수 있거나, 추가될 수 있거나, 병합될 수 있거나, 또는 전부 생략될 수 있다 (예를 들어, 모든 설명된 행위들 또는 이벤트들이 기술의 실시를 위해 필요한 것은 아니다) 는 것이 인식되어야 한다. 또한, 특정 예들에서, 행위들 또는 이벤트들은, 예를 들어, 순차적으로 보다는 멀티 스레드 프로세싱, 인터럽트 프로세싱 또는 다수의 프로세서들을 통해, 동시적으로 수행될 수도 있다.
- [0289] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 그 기능들은 컴퓨터 판독가능 매체 상의 하나 이상의 명령 또는 코드로서 저장되거나 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 예를 들면, 통신 프로토콜에 따라, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체 또는 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체를 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체는 일반적으로, (1) 비밀스러운 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는, 본 개시에서 설명된 기술들의 구현을 위해 명령들, 코드 및/또는 데이터 구조들을 추출하기 위하여 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수 있다.
- [0290] 비한정적 예로서, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령 또는 데이터 구조의 형태로 원하는 프로그램 코드를 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 라인 (DSL), 또는 적외선, 전파 (radio), 및 마이크로파와 같은 무선 기술을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되면, 그 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 전파, 및 마이크로파와 같은 무선 기술은 매체의 정의 내에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속, 캐리어 파, 신호 또는 다른 일시적 매체를 포함하는 것이 아니라, 대신에 비밀시적, 유형의 저장 매체에 관한 것이 이해되어야 한다. 여기에 사용된, 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, DVD (digital versatile disc), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서, 디스크 (disk) 는 보통 데이터를 자기적으로 재생하지만, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 또한, 상기의 조합은 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.
- [0291] 명령들은 하나 이상의 프로세서, 예컨대 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 이산 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에 사용된 용어 "프로세서" 는 전술한 구조 중 임의의 것 또는 본원에 설명된 기술의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 추가로, 일부 양태들에서, 여기서 설명된 기능은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나 또는 결합된 코덱에 포함될 수도 있다. 또한, 그 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.
- [0292] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩 세트) 를 포함하여, 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기술을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실행을 반드시 요구하는 것은 아니다. 오히려, 위에 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호 작용 가능한 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.
- [0293] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

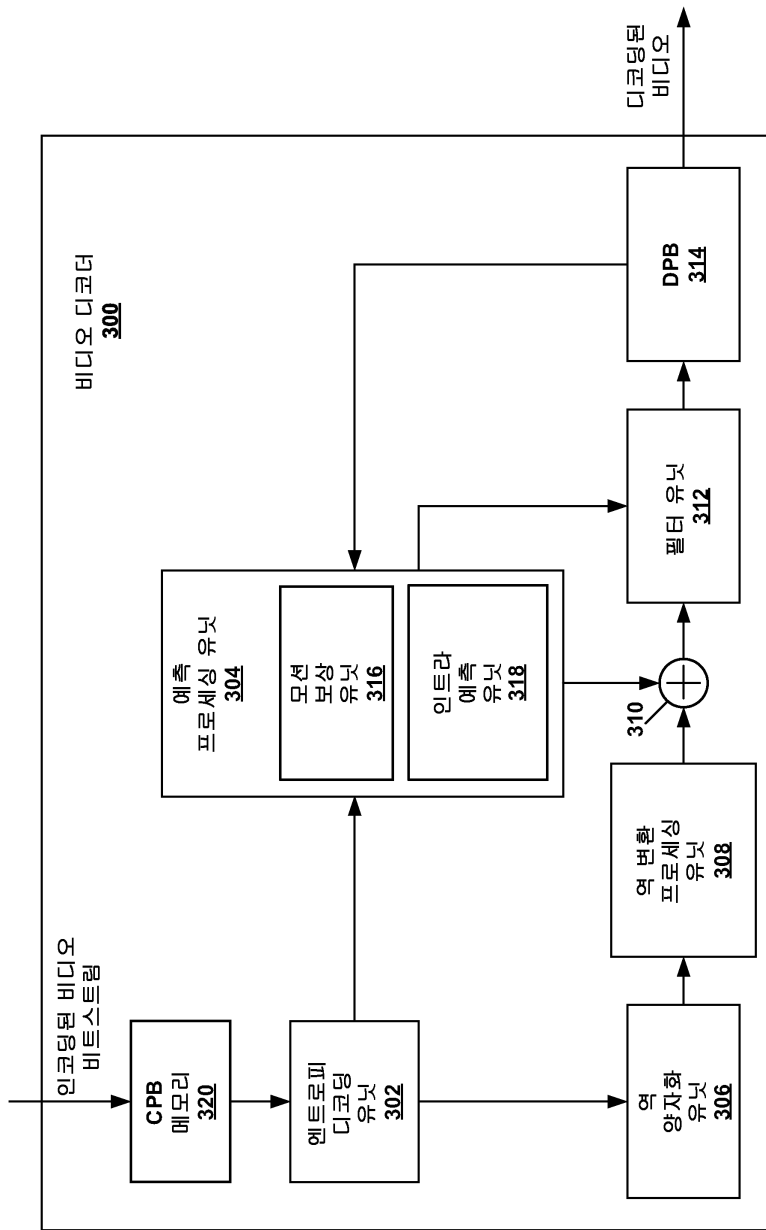
도면1



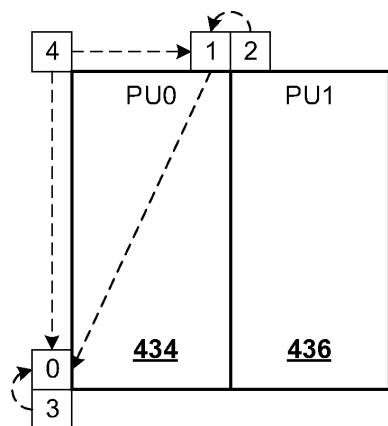
도면2



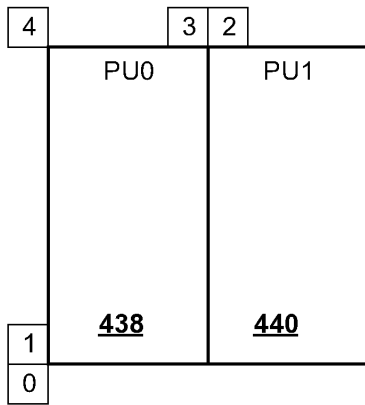
도면3



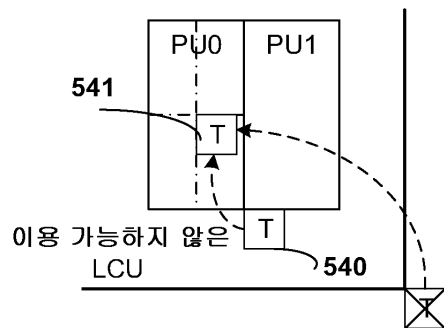
도면4a



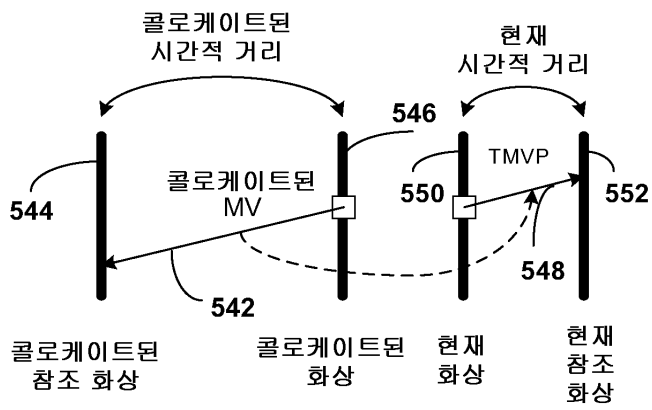
도면4b



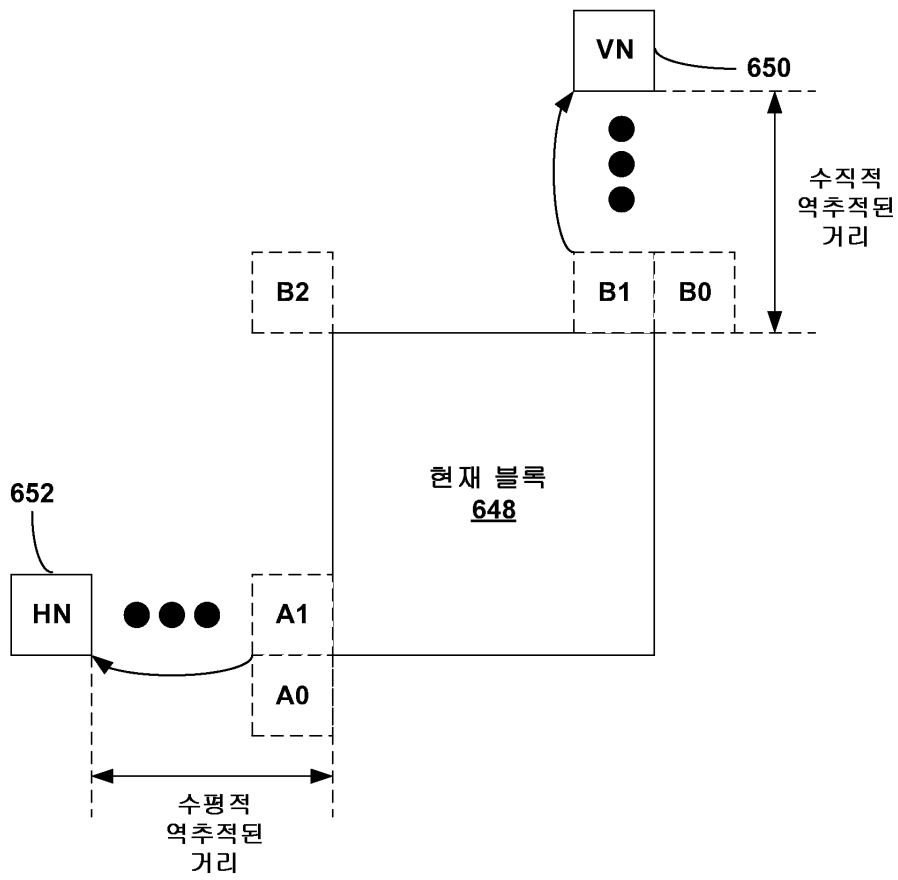
도면5a



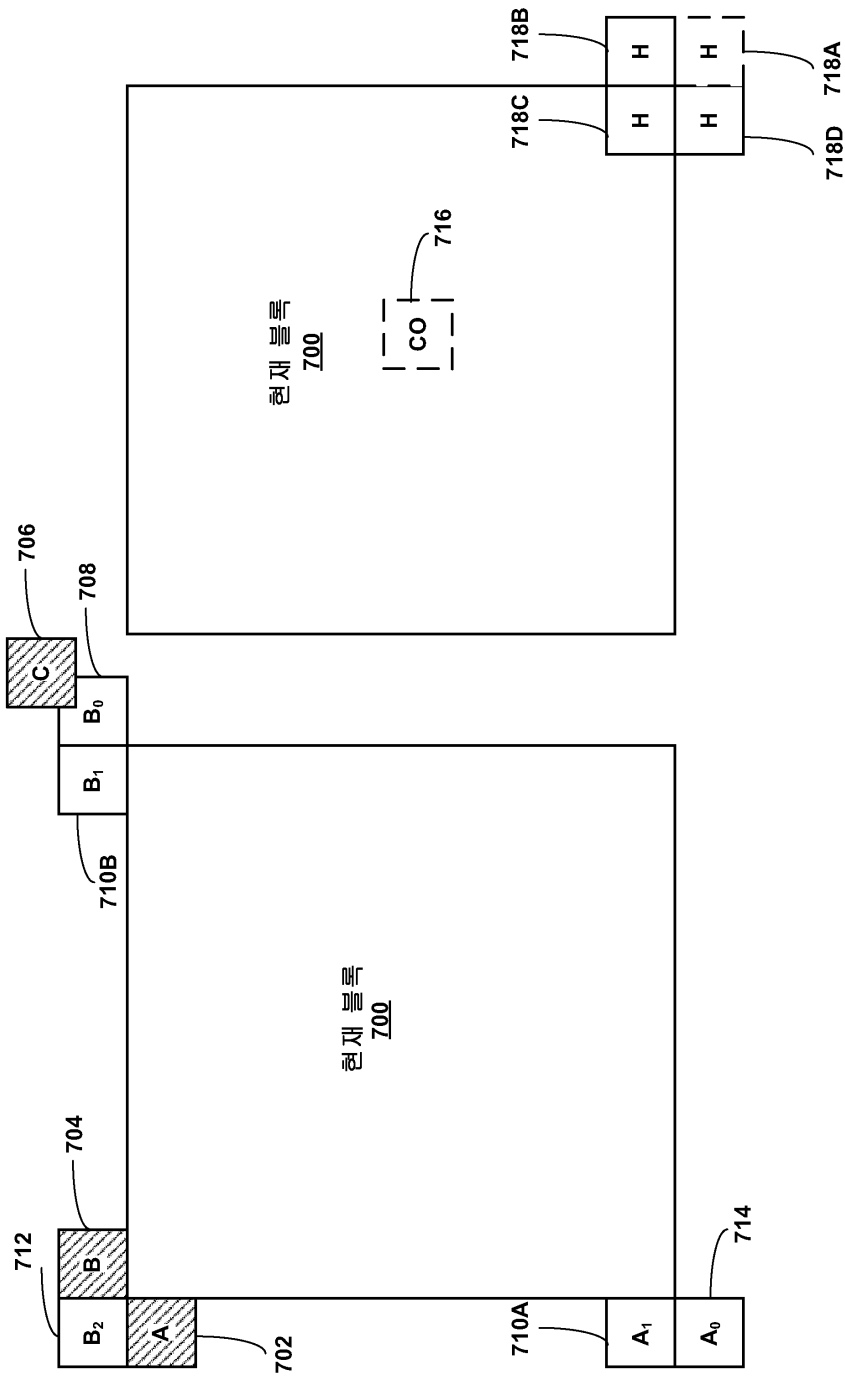
도면5b



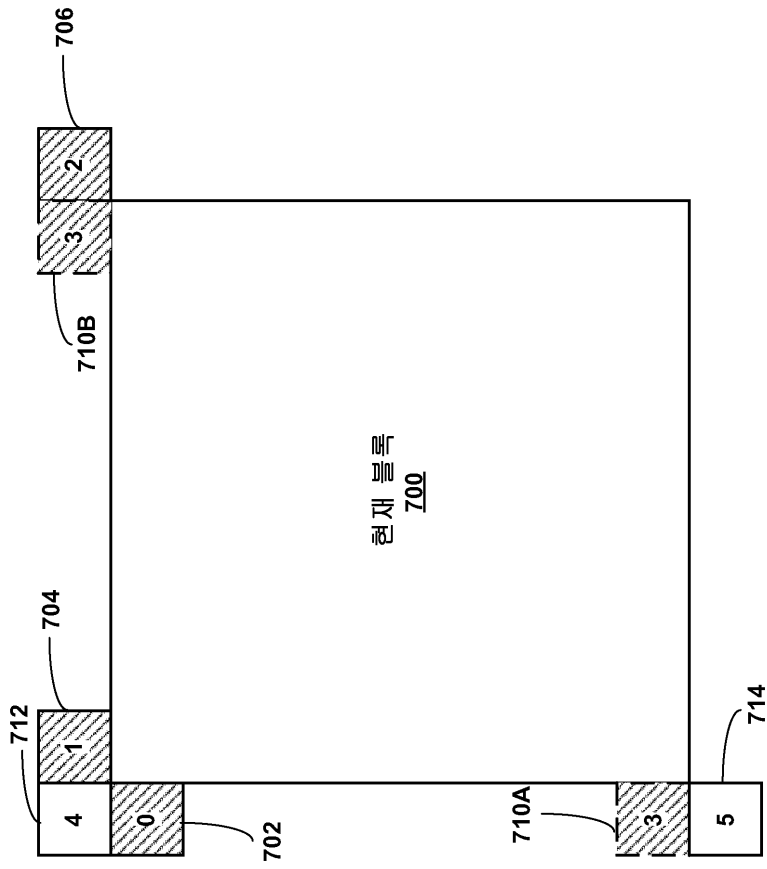
도면6



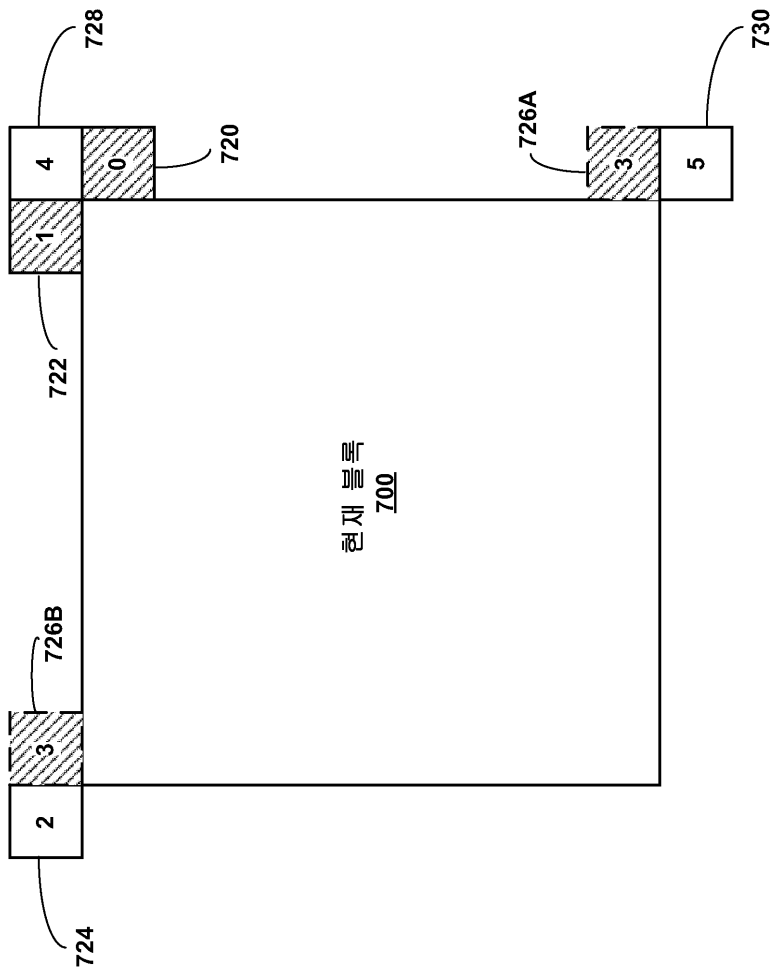
도면7a



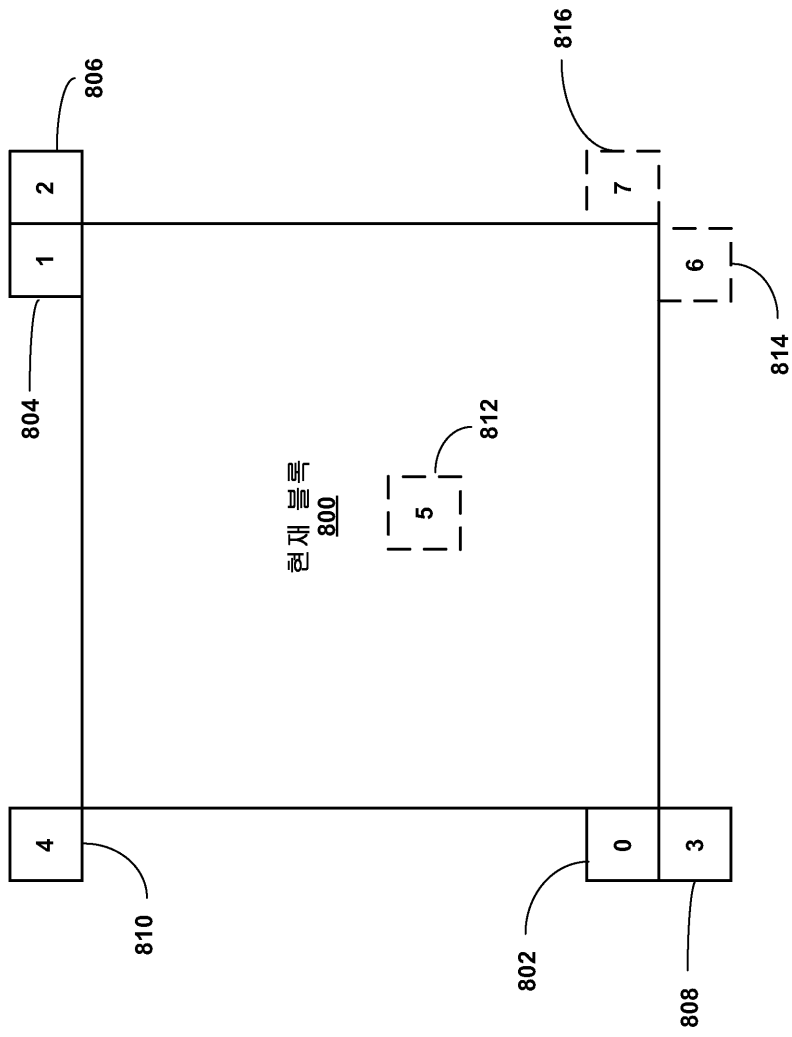
도면7b



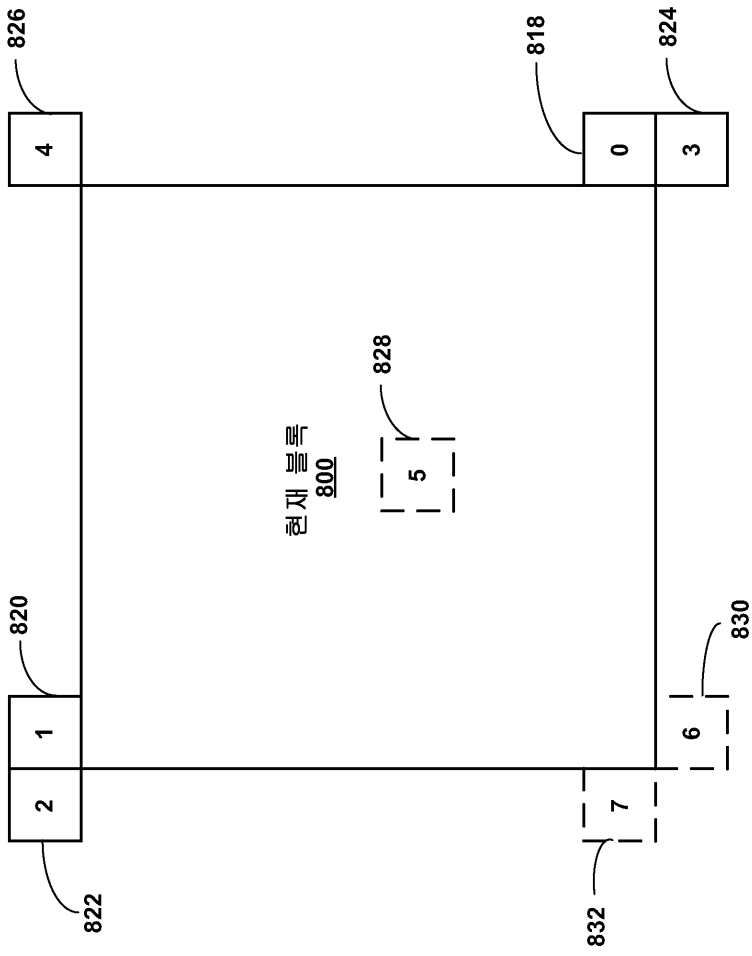
도면7c



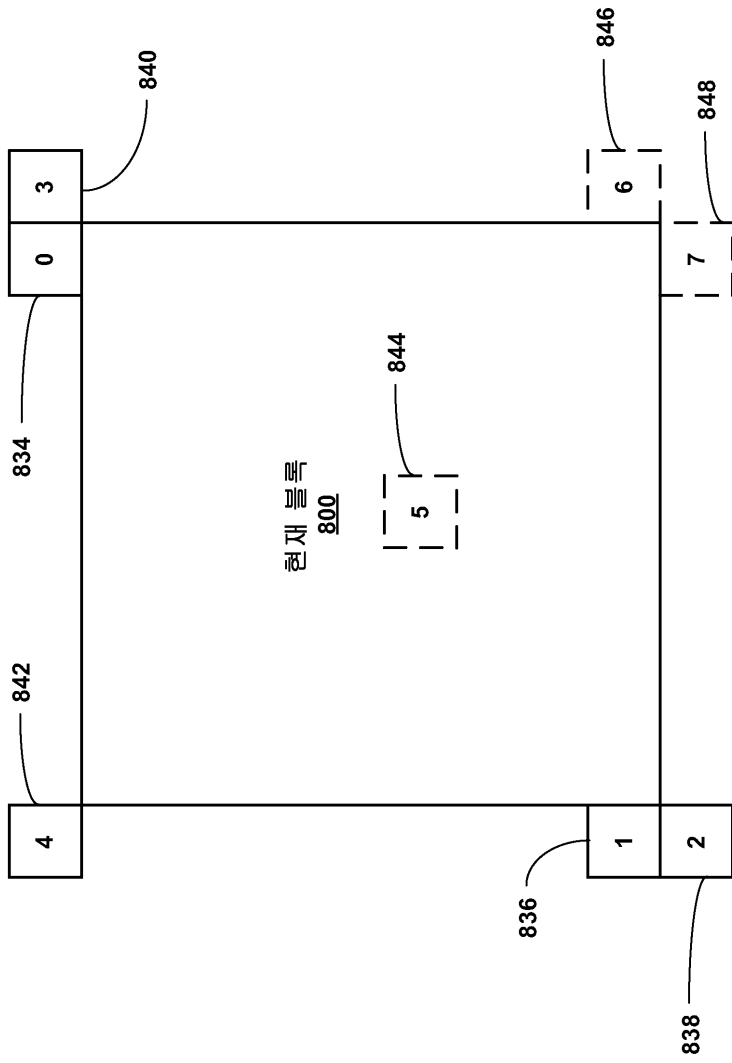
도면8a



도면 8b



도면8c



도면9

