



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2018-0018535  
(43) 공개일자 2018년02월21일

- |  |   |
|--|---|
| (51) 국제특허분류(Int. Cl.)<br><i>H04N 19/52</i> (2014.01) <i>H04N 19/172</i> (2014.01)<br><i>H04N 19/176</i> (2014.01) <i>H04N 19/61</i> (2014.01)<br><i>H04N 19/70</i> (2014.01)<br>(52) CPC특허분류<br><i>H04N 19/52</i> (2015.01)<br><i>H04N 19/172</i> (2015.01)<br>(21) 출원번호 10-2017-7035367<br>(22) 출원일자(국제) 2016년06월09일<br>심사청구일자 없음<br>(85) 번역문제출일자 2017년12월07일<br>(86) 국제출원번호 PCT/US2016/036682<br>(87) 국제공개번호 WO 2016/201094<br>국제공개일자 2016년12월15일<br>(30) 우선권주장<br>62/174,393 2015년06월11일 미국(US)<br>(뒷면에 계속) | (71) 출원인<br><b>켈컴 인코포레이티드</b><br>미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775<br>(72) 발명자<br><b>치엔 웨이-정</b><br>미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775<br><b>왕 상린</b><br>미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775<br>(뒷면에 계속)<br>(74) 대리인<br><b>특허법인코리아나</b> |
|--|---|

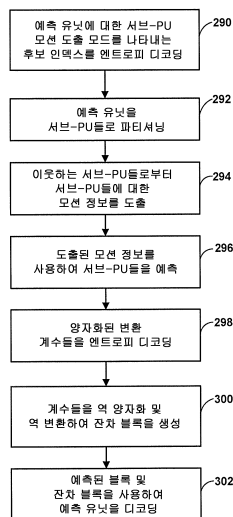
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 **공간적 및/또는 시간적 모션 정보를 사용하는 서브-예측 유닛 모션 벡터 예측**

**(57) 요약**

일 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리 및, 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다고 결정하고, 결정에 응답하여: 현재 블록을 서브-블록들로 파티셔닝하고, 서브-블록들 각각에 대해, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하며, 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하도록 구성된 비디오 디코더를 포함한다.

**대표도** - 도13



(52) CPC특허분류

*HO4N 19/176* (2015.01)

*HO4N 19/61* (2015.01)

*HO4N 19/70* (2015.01)

(72) 발명자

**장 리**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

**리우 홍빈**

중국 102208 베이징 창핑 디스트릭트 휘룽관 퉁화 위안 2번 섹션 37번 빌딩 룸 9-202

**천 지안레**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

**카르체비츠 마르타**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장

62/295,329 2016년02월15일 미국(US)

15/176,790 2016년06월08일 미국(US)

## 명세서

### 청구범위

#### 청구항 1

비디오 데이터를 디코딩하는 방법으로서,

비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 상기 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다고 결정하는 단계;

상기 결정에 응답하여:

상기 현재 블록을 상기 서브-블록들로 파티셔닝하는 단계;

상기 서브-블록들 각각에 대해, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하는 단계; 및

각각의 도출된 상기 모션 정보를 사용하여 상기 서브-블록들을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 2

제 1 항에 있어서,

상기 적어도 2 개의 이웃하는 블록들은 상부-이웃하는 블록, 좌측-이웃하는 블록, 및 시간적으로 이웃하는 블록을 포함하는 그룹으로부터 선택되는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 3

제 2 항에 있어서,

상기 상부-이웃하는 블록은 상기 현재 블록 내의 상부-이웃하는 서브-블록을 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 4

제 2 항에 있어서,

상기 좌측-이웃하는 블록은 상기 현재 블록 내의 좌측-이웃하는 서브-블록을 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 5

제 2 항에 있어서,

상기 상부-이웃하는 블록은 상기 현재 블록 밖의 상부-이웃하는 서브-블록을 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 6

제 2 항에 있어서,

상기 좌측-이웃하는 블록은 상기 현재 블록 밖의 좌측-이웃하는 서브-블록을 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 7

제 2 항에 있어서,

상기 시간적으로 이웃하는 블록은 이전에 디코딩된 픽처에서 상기 현재 블록과 공동-위치되는 블록에 이웃하는

상기 이전에 디코딩된 픽처에서의 블록을 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 8**

제 2 항에 있어서,

상기 모션 정보를 도출하는 단계는, 상기 상부-이웃하는 블록, 상기 좌측-이웃하는 블록, 및 상기 시간적으로 이웃하는 블록에 대한 상기 모션 정보의 평균을 사용하여 상기 모션 정보를 도출하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 9**

제 1 항에 있어서,

상기 이웃하는 블록들은 상기 서브-블록들의 사이즈 이하의 사이즈를 갖는, 비디오 데이터를 디코딩하는 방법.

**청구항 10**

제 1 항에 있어서,

상기 서브-블록들의 사이즈를 나타내는 데이터를 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 11**

제 10 항에 있어서,

상기 서브-블록들의 사이즈를 나타내는 데이터를 디코딩하는 단계는, 슬라이스 헤더, 시퀀스 파라미터 세트 (SPS), 또는 픽처 파라미터 세트 (PPS) 중 적어도 하나에서 데이터를 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 12**

제 1 항에 있어서,

상기 모션 정보를 도출하는 단계는 이웃하는 블록들에 대한 모션 정보를 공통의 레퍼런스 픽처로 스케일링하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 13**

제 1 항에 있어서,

상기 모션 정보를 도출하는 단계는 다음의 식에 따라 상기 서브-블록들 각각에 대한 모션 벡터 (MV) 를 도출하는 단계를 포함하고,

$$(MV_x, MV_y) = ((\sum_{i=0}^2 MV_{xi} + \text{sign}(\sum_{i=0}^2 MV_{xi}) * 1) * 43/128, (\sum_{i=0}^2 MV_{yi} + \text{sign}(-\sum_{i=0}^2 MV_{yi}) * 1) * 43/128)).$$

여기서  $w_i$  는 수평적 가중 인자를 나타내고,  $w_j$  는 수직적 가중 인자를 나타내고,  $0_i$  는 수평적 오프셋 값을 나타내며,  $0_j$  는 수직적 오프셋 값을 나타내는, 비디오 데이터를 디코딩하는 방법.

**청구항 14**

제 1 항에 있어서,

상기 모션 예측 후보를 후보 리스트에 추가하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 15**

제 1 항에 있어서,

상기 서브-블록들을 디코딩하기 전에 상기 서브-블록들을 인코딩하는 단계를 더 포함하고,

상기 방법은, 상기 현재 블록에 대한 상기 모션 예측 후보를 식별하는 데이터를 인코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 16**

제 1 항에 있어서,

상기 방법은 무선 통신 디바이스 상에서 실행 가능하고,

상기 디바이스는,

상기 비디오 데이터를 저장하도록 구성된 메모리;

상기 메모리에 저장된 상기 비디오 데이터를 프로세싱하기 위한 명령들을 실행하도록 구성된 프로세서; 및

상기 비디오 데이터의 인코딩된 버전을 수신하도록 구성된 수신기를 포함하는, 비디오 데이터를 디코딩하는 방법.

**청구항 17**

제 16 항에 있어서,

상기 무선 통신 디바이스는 셀룰러 전화기이고, 상기 인코딩된 비디오 데이터는 상기 수신기에 의해 수신되고 셀룰러 통신 표준에 따라 변조되는, 비디오 데이터를 디코딩하는 방법.

**청구항 18**

비디오 데이터를 디코딩하기 위한 디바이스로서,

비디오 데이터를 저장하도록 구성된 메모리; 및

비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 상기 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다고 결정하고;

상기 결정에 응답하여:

상기 현재 블록을 상기 서브-블록들로 파티셔닝하고;

상기 서브-블록들 각각에 대해, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하며;

각각의 도출된 상기 모션 정보를 사용하여 상기 서브-블록들을 디코딩하도록 구성된 비디오 디코더를 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 19**

제 18 항에 있어서,

상기 모션 정보를 도출하기 위해, 상기 비디오 디코더는, 상기 서브-블록들 각각에 대해, 상부-이웃하는 서브-블록, 좌측-이웃하는 서브-블록, 및 시간적으로 이웃하는 서브-블록을 사용하여 상기 모션 정보를 도출하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 20**

제 19 항에 있어서,

상기 상부-이웃하는 서브-블록은 상기 현재 블록 내 또는 상기 현재 블록 밖의 상부-이웃하는 서브-블록을 포함하며,

상기 좌측-이웃하는 서브-블록은 상기 현재 블록 내 또는 상기 현재 블록 밖의 좌측-이웃하는 서브-블록을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 21**

제 18 항에 있어서,

상기 비디오 디코더는 또한, 슬라이스 헤더, 시퀀스 파라미터 세트 (SPS), 또는 픽처 파라미터 세트 (PPS) 중 적어도 하나로부터의 상기 서브-블록들의 사이즈를 나타내는 데이터를 디코딩하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 22**

제 18 항에 있어서,

상기 비디오 디코더는 또한, 이웃하는 블록들에 대한 모션 정보를 공통의 레퍼런스 픽처로 스케일링하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 23**

제 18 항에 있어서,

상기 비디오 디코더가 상기 서브-블록들을 디코딩하기 전에, 상기 서브-블록들을 인코딩하도록 구성된 비디오 인코더를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 24**

제 18 항에 있어서,

상기 디바이스는, 상기 비디오 데이터의 인코딩된 버전을 수신하도록 구성된 수신기를 더 포함하는, 무선 통신 디바이스인, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 25**

제 24 항에 있어서,

상기 무선 통신 디바이스는 셀룰러 전화기이고, 상기 인코딩된 비디오 데이터는 상기 수신기에 의해 수신되고 셀룰러 통신 표준에 따라 변조되는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 26**

비디오 데이터를 디코딩하기 위한 디바이스로서,

비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 상기 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다고 결정하기 위한 수단;

상기 결정에 응답하여, 상기 현재 블록을 상기 서브-블록들로 파티셔닝하기 위한 수단;

상기 서브-블록들 각각에 대해, 상기 결정에 응답하여 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하기 위한 수단; 및

상기 결정에 응답하여 각각의 도출된 상기 모션 정보를 사용하여 상기 서브-블록들을 디코딩하기 위한 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 27**

제 26 항에 있어서,

상기 모션 정보를 도출하기 위한 수단은, 상기 서브-블록들 각각에 대해, 상부-이웃하는 서브-블록, 좌측-이웃하는 서브-블록, 및 시간적으로 이웃하는 서브-블록을 사용하여 상기 모션 정보를 도출하기 위한 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 28**

제 27 항에 있어서,

상기 상부-이웃하는 서브-블록은 상기 현재 블록 내 또는 상기 현재 블록 밖의 상부-이웃하는 서브-블록을 포함하며,

상기 좌측-이웃하는 서브-블록은 상기 현재 블록 내 또는 상기 현재 블록 밖의 좌측-이웃하는 서브-블록을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 29**

제 26 항에 있어서,

상기 디코딩하기 위한 수단이 상기 서브-블록들을 디코딩하기 전에, 상기 서브-블록들을 인코딩하기 위한 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

**청구항 30**

명령들이 저장되어 있는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 때 프로세서로 하여금,

비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 상기 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다고 결정하게 하고;

상기 결정에 응답하여:

상기 현재 블록을 상기 서브-블록들로 파티셔닝하게 하고;

상기 서브-블록들 각각에 대해, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하게 하며;

각각의 도출된 상기 모션 정보를 사용하여 상기 서브-블록들을 디코딩하게 하는, 컴퓨터 판독가능 저장 매체.

**발명의 설명**

**기술 분야**

[0001] 본 출원은 2015 년 6 월 11 일자로 출원된 미국 가출원 제 62/174,393 호, 및 2016 년 2 월 15 일자로 출원된 미국 가출원 제 62/295,329 호의 이익을 주장하며, 그 각각의 전체 내용들은 참조로서 여기에 포함된다.

[0002] **기술 분야**

[0003] 본 개시물은 비디오 코딩에 관한 것이다.

**배경 기술**

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 방송 시스템들, 무선 방송 시스템들, PDA (personal digital assistant) 들, 랩톱이나 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더들, 디지털 카메라들, 디지털 녹음 디바이스들, 디지털 미디어 재생기들, 비디오 게임 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화들, 소위 "스마트 폰들", 비디오 원격화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 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 (HEVC) 표준 (또한, ITU-T H.265 로서 지칭됨) 에 의해 정의된 표준들, 및 이러한 표준들의 확장들에서 설명된 바와 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써, 디지털 비디오 정보를 보다 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 코딩 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 제거하기 위한 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 포함한다. 블록-기반 비디오 코딩에 있어서, 비디오 슬라이스 (예를 들어, 비디오 프레임 또는 비디오 프레임의 일부) 는 비디오 블록들로 파티셔닝될 수도 있으며, 이 비디오 블록들은 일부 기법들에 대해 또한 트리블록들, 코딩 유닛 (CU) 들 및/또는 코딩 노드들로 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스 내의 비디오 블록들은 동일한 픽처 내의 이웃하는 블록들 내의 레퍼런스 샘플들에 대한 공간적 예측을 사용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽처 내의 이웃하는 블록들 내의 레퍼런스 샘플들에 대하여 공간적 예측을, 또는 다른 레퍼런스 픽처들 내의 레퍼런스 샘플들에 대하여 시간적 예측을 사용할 수도 있다. 픽처들은 프레임들로서 지칭될 수도 있고, 레퍼런스 픽처들은 레퍼런스 프레임들로 지칭될 수도 있다.

[0006] 공간 또는 시간적 예측은 코딩될 블록에 대한 예측 블록을 초래한다. 잔차 데이터는 코딩될 원래의 블록과 예측 블록 간의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 레퍼런스 샘플들의 블록을 가리키는 모션 벡터, 및 코딩된 블록과 예측 블록 간의 차이를 나타내는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위해, 잔차 데이터는 픽셀 도메인에서 변환 도메인으로 변환되어, 잔차 변환 계수들을 초래하고 이 변환 계수들은, 그 후 양자화될 수도 있다. 처음에 2 차원 어레이로 배열된 양자화된 변환 계수들은, 변환 계수들의 1 차원 벡터를 생성하도록 스캐닝될 수도 있고, 엔트로피 코딩이 적용되어 보다 많은 압축을 달성할 수도 있다.

**발명의 내용**

**과제의 해결 수단**

[0007] 일반적으로, 본 개시물의 기법들은 비디오 데이터의 블록들의 서브-블록들에 대한 모션 정보 (예를 들어, 모션 벡터들) 의 도출에 관한 것이다. 예를 들어, 기법들은 예측 유닛 (PU) 들 또는 PU들의 서브-예측 유닛 (서브-PU) 들에 대한 모션 정보를 도출하는데 사용될 수도 있다. 일반적으로, 이들 기법들은 이웃하는 서브-블록들의 모션 정보로부터 서브-블록들 각각에 대한 모션 정보를 도출하는 것을 포함한다. 이웃하는 서브-블록들은 공간적으로 및/또는 시간적으로 이웃하는 서브-블록들을 포함할 수도 있다. 예를 들어, 소정의 서브-블록에 대해, 비디오 코더 (예컨대 비디오 인코더 또는 비디오 디코더) 는 좌측-이웃하는 서브-블록, 상부-이웃하는 서브-블록, 및/또는 시간적으로 이웃하는 서브-블록, 예컨대 하부-우측 시간적으로 이웃하는 서브-블록의 모션 정보를 조합 (예를 들어, 평균) 함으로써 모션 정보를 도출할 수도 있다. 또한, 서브-블록들에 대한 이러한 모션 정보의 도출은 모션 정보 예측을 위한 후보 리스트 중 특정 후보를 사용하여 시그널링될 수도 있다.

[0008] 일 예에서, 비디오 데이터를 디코딩하는 방법은, 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다고 결정하는 단계, 이 결정에 응답하여, 현재 블록을 서브-블록들로 파티셔닝하는 단계, 서브-블록들 각각에 대해, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하는 단계, 및 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하는 단계를 포함한다.

[0009] 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리, 및 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다는 것을 결정하고, 이 결정에 응답하여, 현재 블록을 서브-블록들로 파티셔닝하고, 서브-블록들 각각에 대해, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하며, 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하도록 구성된 비디오 디코더를 포함한다.

[0010] 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는, 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다고 결정하기 위한 수단, 이 결정에 응답하여, 현재 블록을 서브-블록들로 파티셔닝하기 위한 수단, 서브-블록들 각각에 대해, 이 결정에 응답하여, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하기 위한 수단, 및 이 결정에 응답하여, 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하기 위한 수단을 포함한다.

[0011] 다른 예에서, 컴퓨터 판독가능 저장 매체는 명령들이 저장되어 있고, 이 명령들은, 실행되는 경우 프로세서로 하여금, 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다는 것을 결정하게 하고, 결정에 응답하여, 현재 블록을 서브-블록들로 파티셔닝하게 하고, 서브-블록들 각각에 대해, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하게 하며, 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하게 한다.

[0012] 하나 이상의 예들의 상세들이 첨부되는 도면들 및 하기의 상세한 설명들에서 기술된다. 다른 특성들, 목적들 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구범위로부터 명확해질 것이다.

**도면의 간단한 설명**

[0013] 도 1 은 어드밴스드 시간 모션 벡터 예측 (ATMVP) 을 구현하기 위한 기법들을 이용할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 어드밴스드 시간 모션 벡터 예측 (ATMVP) 에 대한 기법들을 구현할 수도 있는 비디오 인코더의 일 예를



예시하는 블록도이다.

도 3 은 어드밴스드 시간 모션 벡터 예측 (ATMVP) 에 대한 기법들을 구현할 수도 있는 비디오 디코더의 일 예를 예시하는 블록도이다.

도 4 는 고 효율 비디오 코딩 (HEVC) 에서 공간적 이웃하는 후보들을 예시하는 개념도이다.

도 5 는 HEVC 에서 시간 모션 벡터 예측 (TMVP) 을 예시하는 개념도이다.

도 6 은 3D-HEVC 에 대한 예시의 예측 구조를 예시한다.

도 7 은 3D-HEVC 에서 서브-PU 기반의 인터-뷰 모션 예측을 예시하는 개념도이다.

도 8 은 레퍼런스 픽처로부터 서브-PU 모션 예측을 예시하는 개념도이다.

도 9 는 (TMVP 와 유사한) ATMVP 에서 관련 픽처들을 예시하는 개념도이다.

도 10 은 예시의 공간-시간적 모션 벡터 예측자 (STMVP) 도출 프로세스를 예시하는 플로우차트이다.

도 11a 및 도 11b 는 PU 의 서브-PU들, 뿐만 아니라 PU 에 이웃하는 서브-PU들의 예들을 예시하는 개념도들이다.

도 12 는 본 개시물의 기법들에 따라 비디오 데이터를 인코딩하는 예시의 방법을 예시하는 플로우차트이다.

도 13 은 본 개시물의 기법들에 따라 비디오 데이터를 디코딩하는 방법의 예이다.

**발명을 실시하기 위한 구체적인 내용**

[0014] 일반적으로, 본 개시물은 비디오 코덱들에서의 모션 벡터 예측에 관련된다. 보다 구체적으로, 어드밴스드 모션 벡터 예측은 공간 및 시간적 이웃 블록들로부터 소정의 블록 (예를 들어, 예측 유닛 (PU)) 에 대한 서브-블록들 (예를 들어, 서브-예측 유닛 (PU)들) 에 대해 모션 벡터들을 도출함으로써 달성될 수도 있다. 하나의 예에서, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더) 는 현재 블록 (예를 들어, 현재 PU) 을 서브-블록들 (예를 들어, 서브-PU들) 로 파터셔닝하고, 각각의 서브-PU 에 대해, 공간적으로 및/또는 시간적으로 이웃하는 블록들을 포함할 수도 있는 이웃하는 블록들로부터의 서브-PU들 각각에 대한 모션 벡터를 포함하는 모션 정보를 도출할 수도 있다. 예를 들어, 서브-블록들 각각에 대해, 비디오 코더는 좌측-이웃하는 공간 블록, 상부-이웃하는 공간 블록, 및/또는 하부-우측 이웃하는 시간 블록으로부터 모션 정보를 도출할 수도 있다. 공간적으로 이웃하는 블록들은 서브-블록에 바로 인접한, 또는 서브-블록을 포함하는 현재 블록 밖에 있는 서브-블록들일 수도 있다. 현재 블록 밖의 서브-블록들을 사용하는 것은, 서브-블록들에 대한 모션 정보가 병렬로 도출되는 것을 허용할 수도 있다.

[0015] 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 및 그 자신의 SVC (Scalable Video Coding) 및 MVC (Multiview Video Coding) 확장들을 포함하는 ITU-T H.264 (또한, ISO/IEC MPEG-4 AVC 로 알려짐) 을 포함한다. MVC 의 최초의 공동 초안은 『"Advanced video coding for generic audiovisual services," ITU-T 권고 H.264, 2010 년 3 월』 에서 설명된다.

[0016] 또한, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 및 ISO/IEC 모션 픽처 전문가 그룹 (MPEG) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 개발되고 있는 새롭게 개발된 비디오 코딩 표준, 즉 고효율 비디오 코딩 (HEVC) 이 있다. HEVC 의 최근 초안은 [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip) 로부터 이용 가능하다. HEVC 표준은 또한, 양자 모두 "High efficiency video coding" 라는 제목으로 양자 모두 2014 년 10 월에 공개된 권고 ITU-T H.265 및 국제 표준 ISO/IEC 23008-2 에서 공동으로 제시된다.

[0017] 모션 정보: 각각의 블록에 대해, 모션 정보의 세트가 이용 가능할 수 있다. 모션 정보의 세트는 순방향 및 역방향 예측 방향들에 대한 모션 정보를 포함한다. 여기서, 순방향 및 역방향 예측 방향들은 현재 픽처 또는 슬라이스의 레퍼런스 픽처 리스트 0 (RefPicList0) 및 레퍼런스 픽처 리스트 1 (RefPicList1) 에 대응하는 2 개의 예측 방향들이다. 용어들 "순방향" 및 "역방향" 은 반드시 기하학적 의미를 가질 필요는 없다. 대신에, 이들은, 모션 벡터가 어느 레퍼런스 픽처 리스트에 기초하는지를 구별하는데 사용된다. 순방향 예측은 레퍼런스 리스트 0 에 기초하여 형성된 예측을 의미하는 한편, 역방향 예측은 레퍼런스 리스트 1 에 기초하여 형성된 예측을 의미한다. 레퍼런스 리스트 0 및 레퍼런스 리스트 1 양자 모두가 소정의 블록에 대한 예

측을 형성하도록 사용되는 경우에서, 이것은 양-방향 예측으로 지칭된다.

- [0018] 소정의 픽처 또는 슬라이스에 대해, 단지 하나의 레퍼런스 픽처 리스트가 사용되면, 픽처 또는 슬라이스 내의 모든 블록이 순방향 예측된다. 양자 모두의 레퍼런스 픽처 리스트들이 소정의 픽처 또는 슬라이스에 대해 사용되면, 픽처 또는 슬라이스 내의 블록은 순방향 예측, 또는 역방향 예측, 또는 양-방향으로 예측될 수도 있다.
- [0019] 각각의 예측 방향에 대해, 모션 정보는 레퍼런스 인덱스 및 모션 벡터를 포함한다. 레퍼런스 인덱스는 대응하는 레퍼런스 픽처 리스트 (예를 들어, RefPicList0 또는 RefPicList1) 에서 레퍼런스 픽처를 식별하는데 사용된다. 모션 벡터는 수평 및 수직 컴포넌트 양자 모두를 갖고, 이 컴포넌트들 각각은 수평 및 수직 방향 각각에 따른 오프셋 값을 나타낸다. 일부 상세한 설명들에서, 간략함을 위해, 용어 "모션 벡터" 는, 모션 벡터 및 그 연관된 레퍼런스 인덱스 양자 모두를 나타내도록 모션 정보와 상호교환적으로 사용될 수도 있다.
- [0020] 픽처 순서 카운트 (POC) 는 픽처의 디스플레이 순서를 식별하기 위해 비디오 코딩 표준들에서 광범위하게 사용된다. 하나의 코딩된 비디오 시퀀스 내에서 2 개의 픽처들이 동일한 POC 값을 가질 수도 있는 경우들이 존재하지만, 이것은 통상적으로 코딩된 비디오 시퀀스 내에서는 일어나지 않는다. 다수의 코딩된 비디오 시퀀스들이 비트스트림으로 존재하는 경우, POC 의 동일한 값을 갖는 픽처들은 디코딩 순서의 관점에서 서로 더 가까울 수도 있다. 픽처들의 POC 값들은 통상적으로, HEVC 및 모션 벡터 스케일링에서와 같이 레퍼런스 픽처 세트의 도출, 레퍼런스 픽처 리스트 구성을 위해 사용된다.
- [0021] 어드밴스드 비디오 코딩 (AVC) (H.264) 에서의 매크로블록 (MB) 구조: H.264/AVC 에서, 각각의 인터 매크로블록 (MB) 은 4 개의 상이한 방식들로 파티셔닝될 수도 있다:
  - [0022] • 하나의 16x16 MB 파티션
  - [0023] • 2 개의 16x8 MB 파티션들
  - [0024] • 2 개의 8x16 MB 파티션들
  - [0025] • 4 개의 8x8 MB 파티션들
- [0026] 하나의 MB 에서 상이한 MB 파티션들은 각각의 방향에 대해 상이한 레퍼런스 인덱스 값들 (RefPicList0 또는 RefPicList1) 을 가질 수도 있다.
- [0027] MB 가 4 개의 8x8 MB 파티션들로 파티셔닝되지 않는 경우, 이것은 각각의 방향에서 각각의 MB 파티션에 대해 단지 하나의 모션 벡터를 갖는다.
- [0028] MB 가 4 개의 8x8 MB 파티션들로 파티셔닝되는 경우, 각각의 8x8 MB 파티션은 또한, 서브-블록들로 파티셔닝될 수 있고, 이 서브 블록들 각각은 각각의 방향에서 상이한 모션 벡터를 가질 수 있다. 8x8 MB 파티션으로부터 서브-블록들을 얻기 위해 4 개의 상이한 방식들이 존재한다:
  - [0029] • 하나의 8x8 서브-블록
  - [0030] • 2 개의 8x4 서브-블록들
  - [0031] • 2 개의 4x8 서브-블록들
  - [0032] • 4 개의 4x4 서브-블록들
- [0033] 각각의 서브-블록은 각각의 방향에서 상이한 모션 벡터를 가질 수 있다. 따라서, 모션 벡터는 서브-블록과 동일한 또는 이보다 더 높은 레벨에 존재한다.
- [0034] AVC 에서의 시간적 직접 모드: AVC 에서, 시간적 직접 모드는 B 슬라이스들에서 스킵 또는 직접 모드에 대해 MB 또는 MB 파티션 레벨 중 어느 하나에서 인에이블될 수 있다. 각각의 MB 파티션에 대해, 현재 블록의 RefPicList1[ 0 ] 에서 현재 MB 파티션과 공동-위치된 블록의 모션 벡터들은 모션 벡터들을 도출하는데 사용된다. 공동-위치된 블록에서 각각의 모션 벡터는 POC 거리들에 기초하여 스케일링된다.
- [0035] AVC 에서의 공간적 직접 모드: AVC 에서, 직접 모드는 또한, 공간적 이웃들로부터 모션 정보를 예측할 수 있다.

- [0036] 고효율 비디오 코딩 (HEVC) 에서의 코딩 유닛 (CU) 구조: HEVC 에서, 슬라이스의 최대 코딩 유닛은 코딩 트리 블록 (CTB) 또는 코딩 트리 유닛 (CTU) 으로 지칭된다. CTB 는 쿼드-트리를 포함하고, 이 쿼드-트리의 노드들은 코딩 유닛들이다.
- [0037] CTB 의 사이즈는 HEVC 메인 프로파일에서 16x16 내지 64x64 의 범위일 수 있다 (하지만, 통상적으로 8x8 CTB 사이즈들이 지원될 수 있다). 코딩 유닛 (CU) 은 CTB 의 동일한 사이즈이고, 8x8 만큼 작을 수 있다. 각각의 코딩 유닛은 하나의 모드로 코딩된다. CU 가 인터 코딩되는 경우, 이것은 또한 2 또는 4 개의 예측 유닛들 (PUs) 로 파티셔닝되거나 또는 추가의 파티션이 적용되지 않는 경우 단지 하나의 PU 가 될 수도 있다. 하나의 CU 에서 2 개의 PU들이 존재하는 경우, 이들은 CU 의 절반 사이즈 직사각형들 또는 1/4 또는 3/4 사이즈를 갖는 2 개의 직사각형 사이즈일 수 있다.
- [0038] CU 가 인터 코딩되는 경우, 모션 정보의 하나의 세트가 각각의 PU 에 대해 존재한다. 또한, 각각의 PU 는 고유한 인터-예측 모드로 코딩되어 모션 정보의 세트를 도출한다.
- [0039] HEVC 에서의 모션 예측: HEVC 표준에서, 예측 유닛 (PU) 에 대해 머지 (머지의 특별한 경우로서 스킵이 고려됨) 및 어드밴스드 모션 벡터 예측 (AMVP) 모드들로 각각 명명된, 2 개의 모션 벡터 예측 모드들이 존재한다.
- [0040] AMVP 또는 머지 모드에서, 모션 벡터 (MV) 후보 리스트는 다수의 모션 벡터 예측자들에 대해 유지된다. 현재 PU 의 모션 벡터(들), 뿐만 아니라 머지 모드에서의 레퍼런스 인덱스들은 MV 후보 리스트로부터 하나의 후보를 취함으로써 생성된다.
- [0041] MV 후보 리스트는 머지 모드에 대해 최대 5 개의 후보들 및 AMVP 모드에 대해 단지 2 개의 후보들을 포함한다. 머지 후보는 모션 정보의 세트, 예를 들어 레퍼런스 픽처 리스트들 (리스트 0 및 리스트 1) 양자 모두에 대응하는 모션 벡터들 및 레퍼런스 인덱스들을 포함할 수도 있다. 머지 후보가 머지 인덱스에 의해 식별되면, 레퍼런스 픽처들은 현재 블록들의 예측을 위해 사용되고, 뿐만 아니라 연관된 모션 벡터들이 결정된다. 그러나, 리스트 0 또는 리스트 1 중 어느 하나로부터 각각의 잠재적인 예측 방향에 대해 AMVP 모드 하에서, 레퍼런스 인덱스는, AMVP 후보가 단지 모션 벡터를 포함하기 때문에 MV 후보 리스트로 MVP 인덱스와 함께 명백하게 시그널링될 필요가 있다. AMVP 모드에서, 예측된 모션 벡터들은 또한, 리파이닝될 수 있다.
- [0042] 위에서 알 수 있는 바와 같이, 머지 후보는 모션 정보의 풀 세트에 대응하는 한편, AMVP 후보는 레퍼런스 인덱스 및 특정 예측 방향에 대한 단지 하나의 모션 벡터를 포함한다.
- [0043] 양자 모두의 모드들에 대한 후보들은 동일한 공간적 및 시간적 이웃 블록들로부터 유사하게 도출된다.
- [0044] 도 1 은 어드밴스드 시간 모션 벡터 예측 (ATMVP) 을 구현하기 위한 기법들을 이용할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 컴퓨터 판독가능 매체 (16) 를 통해 목적지 디바이스 (14) 에 비디오 데이터를 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 구비될 수도 있다.
- [0045] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 컴퓨터 판독가능 매체 (16) 를 통해 수신할 수도 있다. 컴퓨터 판독가능 매체 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 유형의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 컴퓨터 판독가능 매체 (16) 는 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 실시간으로 직접 송신하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷-기반 네트워크, 예컨대 근거리 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0046] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22)로부터 저장 디바이스로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리와 같은 임의의 다양한 분산된 또는 로컬하게 액세스된 데이터 저장 매체, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체를 포함할 수도 있다. 추가의 예에서, 저장 디바이스는 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 다른 중간 저장 디바이스 또는 파일 서버에 대응할 수도 있다. 목적지 디바이스 (14)는 스트리밍 또는 다운로드를 통해 저장 디바이스로부터의 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예시의 파일 서버들은 (예를 들어, 웹사이트 용의) 웹서버, FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14)는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버에 저장된 인코딩된 비디오 데이터를 액세스하는데 적합한 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 양자 모두의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들의 조합일 수도 있다.

[0047] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정들에 반드시 제한되는 것은 아니다. 기법들은 다양한 멀티미디어 애플리케이션들, 예컨대 지상파 (over-the-air) 텔레비전 방송들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, HTTP를 통한 동적 적응형 스트리밍 (DASH)과 같은 인터넷 스트리밍 비디오 송신들, 데이터 저장 매체 상에서 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들 중 임의의 것을 지원하여 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은, 비디오 스트리밍, 비디오 재생, 비디오 방송, 및/또는 비디오 텔레포니와 같은 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0048] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 본 개시물에 따르면, 소스 디바이스 (12)의 비디오 인코더 (20)는 어드밴스드 시간 모션 벡터 예측 (ATMVP)을 위한 기법들을 적용하도록 구성될 수도 있다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 어레이먼트들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12)는 외부 비디오 소스 (18), 예컨대 외부 카메라로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14)는 통합된 디스플레이 디바이스를 포함하기 보다는, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0049] 도 1의 예시된 시스템 (10)은 단지 일 예이다. 어드밴스드 시간 모션 벡터 예측 (ATMVP)에 대한 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로, 본 개시물의 기법들은 비디오 인코딩 디바이스에 의해 수행되지만, 그 기법들은 통상적으로 "CODEC"으로서 지칭된 비디오 인코더/디코더에 의해 또한 수행될 수도 있다. 또한, 본 개시물의 기법들은 비디오 프리프로세서에 의해 또한 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는, 소스 디바이스 (12)가 목적지 디바이스 (14)로의 송신을 위한 코딩된 비디오 데이터를 생성하는 이러한 코딩 디바이스들의 단지 예들이다. 일부 예들에서, 디바이스들 (12, 14)은, 디바이스들 (12, 14) 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (10)은, 예를 들어 비디오 스트리밍, 비디오 재생, 비디오 방송, 또는 비디오 텔레포니를 위해 비디오 디바이스들 (12, 14) 간의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0050] 소스 디바이스 (12)의 비디오 소스 (18)는 비디오 캡처 디바이스, 예컨대 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스를 포함할 수도 있다. 다른 대안으로서, 비디오 소스 (18)는 소스 비디오와 같은 컴퓨터 그래픽 기반 데이터, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합을 생성할 수도 있다. 일부 경우들에서, 비디오 소스 (18)가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14)는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 전술된 바와 같이, 본 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용될 수도 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에서, 캡처된, 미리캡처된, 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20)에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 그 다음에, 컴퓨터 판독가능 매체 (16) 상으로 출력 인터페이스

(22) 에 의해 출력될 수도 있다.

[0051] 컴퓨터 판독가능 매체 (16) 는 트랜시언트 (transient) 매체, 예컨대 무선 브로드캐스트 또는 유선 네트워크 송신, 또는 저장 매체 (즉, 비일시적 저장 매체), 예컨대 하드 디스크, 플래시 드라이브, 콤팩트 디스크, 디지털 비디오 디스크, 블루레이 디스크, 또는 다른 컴퓨터 판독가능 매체를 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (미도시) 는 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신하고, 예를 들어 네트워크 송신을 통해 인코딩된 비디오 데이터를 목적지 디바이스 (14) 에 제공할 수도 있다. 유사하게, 디스크 스탬핑 설비와 같은 매체 생성 설비의 컴퓨팅 디바이스가 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신하고, 인코딩된 비디오 데이터를 포함하는 디스크를 생성할 수도 있다. 따라서, 컴퓨터 판독가능 매체 (16) 는, 다양한 예들에서, 다양한 형태들의 하나 이상의 컴퓨터 판독가능 매체를 포함하는 것으로 이해될 수도 있다.

[0052] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 판독가능 매체 (16) 로부터 정보를 수신한다. 컴퓨터 판독가능 매체 (16) 의 정보는 비디오 디코더 (30) 에 의해 또한 사용되는, 비디오 인코더 (20) 에 의해 정의된 신택스 정보를 포함할 수도 있고, 이 정보는 블록들 및 다른 코딩된 유닛들, 예를 들어, GOP 들의 특징들 및/또는 프로세싱을 설명하는 신택스 엘리먼트들을 포함한다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0053] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 코딩 표준, 예컨대 고효율 비디오 코딩 (HEVC) 표준, HEVC 표준으로의 확장들, 또는 후속의 표준들, 예컨대 ITU-T H.266 에 따라 동작할 수도 있다. 대안으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 다르게는 MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC) 으로도 지칭된 ITU-T H.264 표준과 같은 다른 사실 표준 또는 산업 표준들, 또는 이러한 표준들의 확장들에 따라 동작할 수도 있다. 그러나, 본 개시의 기법들은 임의의 특정 코딩 표준에 제한되지 않는다. 비디오 코딩 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 을 포함한다. 도 1 에 도시되지 않았으나, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통 데이터 스트림 또는 별개의 데이터 스트림들에서 오디오 및 비디오 양자 모두의 인코딩을 핸들링하기 위해 적합한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용 가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜을 준수할 수도 있다.

[0054] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 임의의 다양한 적합한 인코더 회로부, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적 회로들 (ASIC) 들, 필드 프로그램가능 게이트 어레이 (FPGA) 들, 이산 로직, 소프트웨어, 하드웨어, 펌웨어, 또는 이들의 임의의 조합들로서 구현될 수도 있다. 이 기법들이 부분적으로 소프트웨어에서 구현되는 경우, 디바이스는 그 소프트웨어에 대한 명령들을 적합한, 비일시적 컴퓨터 판독가능 매체에 저장하고, 하나 이상의 프로세서들을 사용하는 하드웨어에서 그 명령들을 실행하여 본 개시물의 기법들을 수행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이들 중 어느 하나가 결합된 인코더/디코더 (CODEC) 의 부분으로서 각각의 디바이스에 통합될 수도 있다.

[0055] 일반적으로, HEVC 표준은 루마 및 크로마 샘플들 양자 모두를 포함하는 최대 코딩 유닛들 (LCU) 또는 트리블록들의 시퀀스로 분할될 수도 있다. 비트스트림 내의 신택스 데이터는 LCU 를 정의할 수도 있고, 이것은 픽셀들의 수의 관점에서 가장 큰 코딩 유닛이다. 슬라이스는 코딩 순서에서 다수의 연속적인 트리블록들을 포함한다. 비디오 프레임 또는 픽처는 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 쿼드트리에 따라 코딩 유닛 (CU) 들로 스플리팅될 수도 있다. 일반적으로, 쿼드트리 데이터 구조는, 트리블록에 대응하는 루트 노드를 갖는, CU 당 하나의 노드를 포함한다. CU 가 4 개의 서브-CU들로 스플리팅되면, 그 CU 에 대응하는 노드는 4 개의 리프 노드들을 포함하고, 노드들 각각은 서브-CU들 중 하나에 대응한다.

[0056] 쿼드트리 데이터 구조의 각 노드는 대응하는 CU 에 대한 신택스 데이터를 제공할 수도 있다. 예를 들어, 쿼드트리 내의 노드는, 그 노드에 대응하는 CU 가 서브-CU 들로 스플리팅되는지의 여부를 나타내는 스플릿 플래그를 포함할 수도 있다. CU 에 대한 신택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU 가 서브 CU 들로 스플리팅되는지 여부에 의존할 수도 있다. CU 가 더 스플리팅되지 않으면, 그것은 리프-CU 로서 지칭된다. 본 개시물에서, 리프-CU 의 4 개의 서브 CU들은, 원래의 리프-CU 의 명백한 스플릿이 없더라도 리프-CU 들로

서 또한 지칭될 것이다. 예를 들어, 16x16 사이즈에서 CU 가 더 이상 스플리팅되지 않으면, 4 개의 8x8 서브-CU들은, 16x16 CU 가 절대 스플리팅되지 않더라도, 리프-CU 들로서 또한 지칭될 것이다.

[0057] CU 가 사이즈 구별을 갖지 않는다는 점을 제외하면, H.265 에서의 CU 는 H.264 표준의 매크로 블록과 유사한 목적을 갖는다. 예를 들어, 트리블록은 (서브-CU 들로도 지칭된) 4 개의 자식 (child) 노드들로 스플리팅될 수도 있고, 각각의 자식 노드는 이어서 부모 (parent) 노드일 수도 있고 다른 4 개의 자식 노드들로 스플리팅될 수도 있다. 쿼드트리 리프 노드로서 지칭된 최종의 스플리팅되지 않는 자식 노드는 리프-CU 로도 지칭된 코딩 노드를 포함한다. 코딩된 비트스트림과 연관된 선택스 데이터는, 최대 CU 심도로서 지칭된, 트리블록이 스플리팅될 수도 있는 최대 횡수를 정의할 수도 있고, 코딩 노드들의 최소 사이즈를 또한 정의할 수도 있다. 따라서, 비트스트림은 또한, 최소 코딩 유닛 (SCU) 을 정의할 수도 있다. 본 개시물은 HEVC 의 맥락에서 CU, PU, 또는 TU, 또는 다른 표준들의 맥락에서 유사한 데이터 구조들 (예를 들어, H.264/AVC 에서의 매크로블록들 및 그 서브-블록들) 중 임의의 것을 지칭하기 위해 용어 "블록" 을 사용한다.

[0058] CU 는 코딩 노드 및 코딩 노드와 연관된 변환 유닛 (TU) 들 및 예측 유닛 (PU) 들을 포함한다. CU 의 사이즈는 코딩 노드의 사이즈에 대응하고 정사각형 형상이어야 한다. CU 의 사이즈는 8x8 픽셀들에서 64x64 픽셀들 이상의 최대값을 갖는 트리블록의 사이즈까지의 범위에 있을 수도 있다. 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다. CU 와 연관된 선택스 데이터는, 예를 들어 CU 를 하나 이상의 PU 들로 파티셔닝하는 것을 설명할 수도 있다. 파티셔닝 모드들은, CU 가 스킵 또는 다이렉트 모드 인코딩되는지, 인트라-예측 모드 인코딩되는지, 또는 인터-예측 모드 인코딩되는지 여부 사이에서 상이할 수도 있다. PU 들은 비정사각형의 형상으로 파티셔닝될 수도 있다. CU 와 연관된 선택스 데이터는, 예를 들어 CU 를 쿼드트리에 따라 하나 이상의 TU 들로 파티셔닝하는 것을 또한 설명할 수도 있다. TU 는 정사각형 또는 비-정사각형 (예를 들어, 직사각형) 형상일 수 있다.

[0059] HEVC 표준은 TU 들에 따른 변환들을 허용하는데, 이것은 상이한 CU 들에 대해 상이할 수도 있다. TU 들은 통상적으로, 파티셔닝된 LCU 에 대해 정의된 소정의 CU 내에서의 PU 들의 사이즈에 기초하여 크기가 정해지지만, 이것이 항상 그러한 경우는 아닐 수도 있다. TU 들은 통상적으로, PU 들과 동일한 사이즈이거나 또는 더 작다. 일부 예들에서, CU 에 대응하는 잔차 샘플들은, "잔차 쿼드 트리 (residual quad tree; RQT)" 로서 알려진 쿼드트리 구조를 사용하여 더 작은 유닛들로 세분될 수도 있다. RQT 의 리프 노드들은 변환 유닛들 (TU들) 로서 지칭될 수도 있다. TU 들과 연관된 픽셀 차이 값들이 변환되어 변환 계수들을 생성할 수도 있고, 변환 계수는 양자화될 수도 있다.

[0060] 리프-CU 는 하나 이상의 예측 유닛 (PU) 들을 포함할 수도 있다. 일반적으로, PU 는 대응하는 CU 의 전체 또는 일부에 대응하는 공간 영역을 나타내고, PU 에 대한 레퍼런스 샘플을 추출하기 위한 데이터를 포함할 수도 있다. 또한, PU 는 예측과 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라-모드 인코딩된 경우, PU 에 대한 데이터는 잔차 쿼드트리 (RQT) 에 포함될 수도 있으며, 잔차 쿼드트리는 PU 에 대응하는 TU 에 대한 인트라-예측 모드를 설명하는 데이터를 포함할 수도 있다. 다른 예로서, PU 가 인터-모드 인코딩되는 경우, PU 는 PU 에 대한 하나 이상의 모션 벡터를 정의하는 데이터를 포함할 수도 있다. PU 에 대한 모션 벡터를 정의하는 데이터는, 예를 들어 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터에 대한 레졸루션 (예를 들어, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 레퍼런스 픽처, 및/또는 모션 벡터에 대한 레퍼런스 픽처 리스트 (예를 들어, 리스트 0, 리스트 1, 또는 리스트 C) 를 설명할 수도 있다.

[0061] 하나 이상의 PU 들을 갖는 리프-CU 는 하나 이상의 변환 유닛 (TU) 들을 또한 포함할 수도 있다. 변환 유닛들은, 위에서 논의된 바와 같이, (TU 쿼드트리 구조로도 지칭된) RQT 를 사용하여 명시될 수도 있다. 예를 들어, 스플릿 플래그는, 리프-CU 가 4 개의 변환 유닛들로 스플리팅되는지의 여부를 나타낼 수도 있다. 그 후, 각각의 변환 유닛은 추가의 서브-TU들로 더 스플리팅될 수도 있다. TU 가 더 이상 스플리팅되지 않는 경우, 이것은 리프-TU 로서 지칭될 수도 있다. 일반적으로, 인트라 코딩에 대해, 리프-CU 에 속하는 모든 리프-TU들은 동일한 인트라-예측 모드를 공유한다. 즉, 리프-CU 의 모든 TU들에 대한 예측된 값들을 계산하기 위해 동일한 인트라-예측 모드가 일반적으로 적용된다. 인트라 코딩에 대해, 비디오 인코더는, TU 에 대응하는 CU 의 부분과 원래의 블록 간의 차이로서, 인트라-예측 모드를 사용하여 각각의 리프-TU 에 대한 잔차 값을 계산할 수도 있다. TU 는 PU 의 사이즈에 반드시 제한되는 것은 아니다. 따라서, TU 들은 PU 보다 더 크거나 더 작을 수도 있다. 인트라 코딩에 대해, PU 는 동일한 CU 에 대한 대응하는 리프-TU 와 병치 (collocate) 될 수도 있다. 일부 예들에서, 리프-TU 의 최대 사이즈는 대응하는 리프-CU 의 사이즈에 대응할 수도 있다.

- [0062] 또한, 리프-CU 들의 TU 들은 또한 잔차 쿼드트리 (RQT) 들로서 지칭된, 각각의 쿼드트리 데이터 구조들과 연관 될 수도 있다. 즉, 리프-CU는 리프-CU가 TU들로 어떻게 파티셔닝되는지를 나타내는 쿼드트리를 포함할 수도 있다. TU 쿼드트리의 루트 노드는 일반적으로 리프-CU 에 대응하지만, CU 쿼드트리의 루트 노드는 일반적으로 트리블록 (또는, LCU) 에 대응한다. 스플리팅되지 않은 RQT 의 TU 들은 리프-TU 들로서 지칭된다. 일반적으로, 다르게 언급되지 않는 한, 본 개시물은 리프-CU 및 리프-TU를 각각 지칭하기 위해 용어들 CU 및 TU 를 사용한다.
- [0063] 비디오 시퀀스는 통상적으로, 일련의 비디오 프레임들 또는 픽처들을 포함한다. 픽처들의 그룹 (GOP) 은 일반적으로, 일련의 하나 이상의 비디오 픽처들을 포함한다. GOP 는 GOP 의 헤더, 하나 이상의 픽처들의 헤더, 또는 그 외의 곳에 GOP 에 포함된 픽처들의 수를 설명하는 선택스 데이터를 포함할 수도 있다. 픽처의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 설명하는 슬라이스 선택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 통상적으로, 비디오 데이터를 인코딩하기 위해 개개의 비디오 슬라이스들 내의 비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정된 또는 가변적인 사이즈들을 가질 수도 있고, 지정된 코딩 표준에 따라 사이즈가 상이할 수도 있다.
- [0064] 일 예로써, HM 은 다양한 PU 사이즈들에서의 예측을 지원한다. 특정 CU 의 사이즈가  $2N \times 2N$  이라고 가정하면, HM 은  $2N \times 2N$  또는  $N \times N$  의 PU 사이즈들에서의 인트라-예측, 및  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ , 또는  $N \times N$  의 대칭적 PU 사이즈들에서의 인터-예측을 지원한다. HM 은  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$ , 및  $nR \times 2N$  의 PU 사이즈들에서의 인터-예측에 대한 비대칭적 파티셔닝을 또한 지원한다. 비대칭적 파티셔닝에서, CU 의 일 방향은 파티셔닝되지 않는 한편, 다른 방향은 25% 및 75% 로 파티셔닝된다. 25% 파티션에 대응하는 CU 의 일부는 "n" 다음에 "위", "아래", "좌측", 또는 "우측" 의 표시에 의해 표시된다. 따라서, 예를 들어 " $2N \times nU$ " 은 위쪽의  $2N \times 0.5N$  PU 와 아래쪽의  $2N \times 1.5N$  PU 로 수평적으로 파티셔닝되는  $2N \times 2N$  CU 를 지칭한다.
- [0065] 본 개시물에서, " $N \times N$ " 및 "N 바이 N", 예를 들어  $16 \times 16$  픽셀들 또는 16 바이 16 개의 픽셀들은 수직 및 수평 치수들 관점에서 비디오 블록의 픽셀 치수들을 지칭하기 위해 상호 교환적으로 사용될 수도 있다. 일반적으로,  $16 \times 16$  블록은 수직 방향으로 16 개의 픽셀들 ( $y=16$ ) 및 수평 방향으로 16 개의 픽셀들 ( $x=16$ ) 을 구비할 것이다. 마찬가지로,  $N \times N$  블록은 일반적으로, 수직 방향으로 N 개의 픽셀들 및 수평 방향으로 N 개의 픽셀들을 구비하는데, 여기서 N 은 음이 아닌 정수 값을 나타낸다. 블록에서의 픽셀들은 행들 및 열들로 배열될 수도 있다. 또한, 블록들은 수평 방향에서의 픽셀들의 수가 수직 방향에서의 것과 반드시 동일할 필요는 없다. 예를 들어, 블록들은  $N \times M$  픽셀들을 포함할 수도 있으며, 여기서 M 은 N 과 반드시 동일하지는 않다.
- [0066] CU 의 PU 들을 사용하는 인트라-예측 또는 인터-예측 코딩 다음에, 비디오 인코더 (20) 는 CU 의 TU 들에 대한 잔차 데이터를 계산할 수도 있다. PU 들은 공간 도메인 (픽셀 도메인으로도 지칭됨) 에서의 예측 픽셀 데이터를 생성하는 방법 또는 모드를 설명하는 데이터를 포함할 수도 있고, TU 들은 잔차 비디오 데이터에 대한, 예를 들어 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환과 같은 변환의 적용을 뒤따르는 변환 도메인에서의 계수들을 포함할 수도 있다. 잔차 데이터는 인코딩되지 않은 픽처와 PU 들에 대응하는 예측 값들의 픽셀들 간의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔차 데이터를 포함하는 TU 들을 형성하고, 그 후 TU 들을 변환하여 CU 에 대한 변환 계수들을 생성할 수도 있다.
- [0067] 변환 계수들을 생성하기 위한 임의의 변환들 다음에, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로, 계수들을 표현하기 위해 사용된 데이터의 양을 가능한 감소시키기 위해 변환 계수들이 양자화되어, 추가의 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 값은 양자화 동안 m-비트 값으로 내림 (round down) 될 수도 있고, 여기서 n 은 m 보다 더 크다.
- [0068] 양자화 다음에, 비디오 인코더는 변환 계수들을 스캐닝하여, 양자화된 변환 계수들을 포함하는 2 차원 매트릭스로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 어레이의 앞쪽에 보다 높은 에너지 (따라서 보다 낮은 주파수) 계수들을 배치하고, 어레이의 뒤쪽에 보다 낮은 에너지 (따라서 보다 높은 주파수) 계수들을 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해 양자화된 변환 계수들을 스캔하도록 미리 정의된 스캔 순서를 활용할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응 스캔 (adaptive scan) 을 수행할 수도 있다. 양자화된 변환 계수들을 스캐

닝하여 1 차원 벡터를 형성한 후에, 비디오 인코더 (20) 는, 예를 들어 콘텍스트-적응 가변 길이 코딩 (CAVLC), 콘텍스트-적응 이진 산술 코딩 (CABAC), 신택스-기반 콘텍스트-적응 이진 산술 코딩 (SBAC), 확률 인터벌 파티셔닝 엔트로피 (PIPE) 코딩, 또는 다른 엔트로피 인코딩 방법론에 따라 1 차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30) 에 의해 사용하기 위한 인코딩된 비디오 데이터와 연관된 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0069] CABAC 를 수행하기 위해, 비디오 인코더 (20) 는 콘텍스트 모델 내의 콘텍스트를 송신될 심볼에 할당할 수도 있다. 콘텍스트는, 예를 들어 심볼의 이웃하는 값들이  $n$ -제로인지 또는 아닌지 여부에 관련될 수도 있다. CAVLC 를 수행하기 위해, 비디오 인코더 (20) 는 송신될 심볼에 대한 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드들은, 상대적으로 더 짧은 코드들이 고향률 (more probable) 심볼들에 대응하는 한편, 상대적으로 더 긴 코드들이 저확률 (less probable) 심볼들에 대응하도록 구성될 수도 있다. 이 방식에서, VLC 의 사용은 예를 들어 송신될 각각의 심볼에 대해 동일-길이의 코드워드들을 사용하는 것을 통해 비트 절감을 달성할 수도 있다. 확률 결정은 심볼들에 할당된 콘텍스트에 기초할 수도 있다.

[0070] 2016 년 1 월 25 일자로 출원된, 미국출원 제 15/005,564 (이하에서, " '564 출원") 은, 본 개시물의 기법들에 추가하여, 단독으로 또는 임의의 조합으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있는 다음의 기법들을 설명하였다. 특히, '564 출원은, 예를 들어 머지 후보 리스트로서 삽입되는 경우 ATMVP 후보의 포지션에 관련된 기법들을 설명한다. 공간적 후보들 및 TMVP 후보가 소정 순서로 머지 후보 리스트 안에 삽입된다고 가정한다. ATMVP 후보는 이들 후보들의 임의의 상대적으로 고정된 포지션에 삽입될 수도 있다. 일 대안으로, 예를 들어 ATMVP 후보는 첫 번째 2 개의 공간적 후보들, 예를 들어 A1 및 B1 후에 머지 후보 리스트에 삽입될 수 있다. 일 대안으로, 예를 들어 ATMVP 후보는 첫 번째 3 개의 공간적 후보들, 예를 들어 A1 및 B1 및 B0 후에 삽입될 수 있다. 일 대안으로, 예를 들어 ATMVP 후보는 첫 번째 4 개의 공간적 후보들, 예를 들어 A1, B1, B0, 및 A0 후에 삽입될 수 있다. 일 대안으로, 예를 들어 ATMVP 후보는 TMVP 후보 전에 우측에서 삽입될 수 있다. 일 대안으로, 예를 들어 ATMVP 후보는 TMVP 후보 후에 우측에서 삽입될 수 있다. 대안으로, 후보 리스트에서 ATMVP 후보의 포지션은 비트스트림으로 시그널링될 수 있다. TMVP 후보를 포함하는, 다른 후보들의 포지션들은 추가적으로 시그널링될 수 있다.

[0071] '564 출원은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 수행하도록 구성될 수도 있는, 모션 정보의 단 하나의 세트에 액세스함으로써 적용할 수 있는 ATMVP 후보의 이용 가능성 체크에 관련된 기법들을 설명한다. 이러한 정보의 세트가 이용 가능하지 않은 경우, 예를 들어 하나의 블록이 인트라-코딩되는 경우, 전체 ATMVP 후보는 이용 가능하지 않은 것으로서 고려된다. 그 경우에서, ATMVP 는 머지 리스트 안에 삽입되지 않을 것이다. 센터 포지션, 또는 센터 서브-PU 는 순전히 ATMVP 후보의 이용 가능성을 체크하도록 사용된다. 센터 서브-PU 가 사용되는 경우, 센터 서브-PU 는 센터 포지션 (예를 들어, PU 의 상부-좌측 샘플에 대해 (W/2, H/2) 의 상대적 좌표를 갖는 센터 3 포지션, 여기서  $W \times H$  는 PU 의 사이즈임) 을 커버하는 것이도록 선택된다. 이러한 포지션 또는 센터 서브-PU 는 모션 소스 픽처에서 대응하는 블록을 식별하도록 시간적 벡터와 함께 사용될 수도 있다. 대응하는 블록의 센터 포지션을 커버하는 블록으로부터 모션 정보의 세트가 식별된다.

[0072] '564 출원은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 수행하도록 구성될 수도 있는, 서브-PU 로부터 ATMVP 코딩된 PU 에 대한 모션 정보의 대표 세트에 관련된 기법들을 설명한다. ATMVP 후보를 형성하기 위해, 모션 정보의 대표 세트가 먼저, 형성된다. 모션 정보의 이러한 대표 세트는 고정된 포지션 또는 고정된 서브-PU 로부터 도출될 수도 있다. 이것은, 전술된 바와 같이, ATMVP 후보의 이용 가능성을 결정하는데 사용된 모션 정보의 세트와 동일한 방식으로 선택될 수 있다. 서브-PU 가 모션 정보의 그 자신의 세트를 식별하였고 이용 가능하지 않으면, 이것은 모션 정보의 대표 세트와 동일한 것으로 설정된다. 모션 정보의 대표 세트가 서브-PU 의 것이도록 설정되면, 최악의 경우의 시나리오에서 현재의 CTU 또는 슬라이스에 대해 디코더 측에서 추가의 모션 스토리지가 전혀 필요하지 않다. 프로세스가 결합된 양방향-예측 머지 후보들을 생성하는데 사용되도록, 전체 PU 가 프루닝을 포함하는, 모션 정보의 하나의 세트에 의해 표현될 것을 디코딩 프로세스들이 요구하는 경우, 모든 시나리오들에서 모션 정보의 이러한 대표 세트가 사용된다.

[0073] '564 출원은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있는, ATMVP 후보가 TMVP 후보와 프루닝될 수도 있는 방법 및 TMVP 와 ATMVP 간의 상호작용들이 고려될 수 있는 방법에 관련된 기법들을 설명한다. 서브-PU 기반의 후보, 예를 들어 정규 후보를 갖는 ATMVP 후보의 프루닝은 이러한 서브-PU 기반의 후보에 대한 (블릿 #3 에서와 같은) 모션 정보의 대표적인 세트를 사용함으로써 구성될 수도 있다. 모션 정보의 이러한 세트가 정규 머지 후보와 동일한 것이라면, 2 개의 후보들은 동일한 것으로서 고려된다.



대안으로, 또는 추가하여 ATMVP 가 다수의 서브-PU들에 대한 모션 정보의 다수의 상이한 세트들을 포함하는지 여부를 결정하도록 체크가 수행된다; 적어도 2 개의 상이한 세트가 식별되면, 서브-PU 기반의 후보는 프루닝을 위해 사용되지 않고, 즉 임의의 다른 후보와 상이한 것으로 고려되고; 그렇지 않으면, 이것은 프루닝을 위해 사용될 수도 있다 (예를 들어, 프루닝 프로세스 동안 프루닝될 수도 있다). 대안으로, 또는 추가하여, ATMVP 후보는 공간적 후보들, 예를 들어 A1 및 B1 으로서 표기된 포지션들을 갖는 단지 좌측 및 상부 후보들과 프루닝될 수도 있다. 대안으로, 단지 하나의 후보가 ATMVP 후보 또는 TMVP 후보 중 어느 하나인, 시간적 레퍼런스로부터 형성된다. ATMVP 가 이용 가능한 경우, 후보는 ATMVP 이고; 그렇지 않으면, 후보는 TMVP 이다. 이러한 후보는 TMVP 의 포지션과 유사한 포지션에서 머지 후보 리스트 안에 삽입된다. 이 경우에서, 후보들의 최대 수는 변하지 않고 유지될 수도 있다. 대안으로, TMVP 는 항상, ATMVP 가 이용 가능하지 않은 경우에도 디스에이블된다. 대안으로, TMVP 는, ATMVP 가 이용 가능하지 않은 경우에만 사용된다. 대안으로, ATMVP 가 이용 가능하고 TMVP 가 이용 가능하지 않은 경우, 하나의 서브-PU 의 모션 정보의 하나의 세트가 TMVP 후보로서 사용된다. 이 경우에서, 또한, ATMVP 와 TMVP 간의 프루닝 프로세스가 적용되지 않는다. 대안으로 또는 부가적으로, ATMVP 에 대해 사용된 시간적 벡터는 또한, HEVC 에서 현재 TMVP 에 대해 사용된 바와 같이 하부-우측 포지션 또는 센터 3 포지션이 사용될 필요가 없도록, TMVP 에 대해 사용될 수도 있다. 대안으로, 시간적 벡터에 의해 식별된 포지션 및 하부-우측 및 센터 3 포지션들은 이용 가능한 TMVP 후보를 제공하도록 합동으로 고려된다.

[0074] '564 출원은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있는, 더 정확하고 효율적이도록 ATMVP 후보에 대한 더 높은 기회들을 제공하기 위해 ATMVP 에 대한 다수의 이용 가능성 체크들이 지원될 수 있는 방법을 설명한다. (예를 들어, 도 9 에 도시된 바와 같이) 제 1 시간적 벡터에 의해 식별된 바와 같은 모션 소스 픽처로부터 현재의 ATMVP 후보가 이용 가능하지 않은 경우, 다른 픽처들이 모션 소스 픽처로서 고려될 수 있다. 다른 픽처가 고려되는 경우, 이것은 상이한 제 2 의 시간적 벡터와 연관될 수도 있고, 또는 이용 가능하지 않은 ATMVP 후보를 가리키는 제 1 시간적 벡터로부터 스케일링된 제 2 시간적 벡터와 단순히 연관될 수도 있다. 제 2 시간적 벡터는 제 2 모션 소스 픽처에서 ATMVP 후보를 식별할 수 있고, 동일한 이용 가능성 체크가 적용할 수 있다. 제 2 모션 소스 픽처로부터 도출된 바와 같은 ATMVP 후보가 이용 가능하면, ATMVP 후보가 도출되고 다른 픽처들은 체크될 필요가 없다; 그렇지 않으면, 모션 소스 픽처들과 같은 다른 픽처들은 체크될 필요가 있다. 체크될 픽처들은, 소정 순서로 현재 픽처의 레퍼런스 픽처 리스트들에서의 픽처들일 수도 있다. 각각의 리스트에 대해, 픽처들은 레퍼런스 인덱스의 올림 차순으로 체크된다. 리스트 X 가 먼저 체크되고, (1-X 인) 리스트 Y 에서의 픽처들이 뒤따른다. 리스트 X 는, 리스트 X 가 TMVP 에 대해 사용된 병치된 픽처를 포함하는 리스트이도록 선택된다. 대안으로, X 는 단순히, 1 또는 0 으로 설정된다. 체크될 픽처들은, 소정 순서로 공간적 이웃들의 모션 벡터들에 의해 식별된 것들을 포함할 수도 있다. 현재 ATMVP 가 적용할 PU 의 파티션은 2Nx2N, NxN, 2NxN, Nx2N 또는 다른 AMP 파티션들, 예컨대 2NxN/2 일 수도 있다. 대안으로, 또는 추가하여 다른 파티션 사이즈들이 허용될 수 있으면, ATMVP 도 또한 지원될 수 있고, 이러한 사이즈는 예를 들어, 64x8 을 포함할 수도 있다. 대안으로, 모드는 소정의 파티션들, 예를 들어 2Nx2N 에만 적용될 수도 있다.

[0075] '564 출원은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 수행하도록 구성될 수도 있는, 머지 모드의 상이한 유형을 사용하여 ATMVP 후보가 마킹될 수도 있는 방법을 설명한다.

[0076] 이웃들로부터 벡터 (제 1 스테이지에서와 같은 시간적 벡터) 를 식별하는 경우, 다수의 이웃하는 포지션들, 예를 들어 머지 후보 리스트 구성에서 사용된 것들이 순서대로 체크될 수 있다. 이웃들 각각에 대해, 레퍼런스 픽처 리스트 0 (list 0) 또는 레퍼런스 픽처 리스트 1 (list 1) 에 대응하는 모션 벡터들이 순서대로 체크될 수 있다. 2 개의 모션 벡터들이 이용 가능한 경우, 리스트 X 가 TMVP 에 대해 사용된 병치된 픽처를 포함하는 리스트이도록, 리스트 X 에서 모션 벡터들이 먼저, 그 다음에 리스트 Y (Y 는 1-X 과 동일함) 에서 체크될 수 있다. ATMVP 에서, 시간적 벡터는 서브-PU 의 임의의 센터 포지션의 시프트로서 추가되도록 사용되고, 여기서 시간적 벡터의 컴포넌트들은 정수들로 시프트될 필요가 있을 수도 있다. 이러한 시프트된 센터 포지션은, 모션 벡터들이, 예를 들어 현재의 센터 포지션을 커버하는 4x4 의 사이즈에, 할당될 수 있는 최소 유닛을 식별하는데 사용된다. 대안으로, 리스트 0 에 대응하는 모션 벡터들은, 리스트 1 에 대응하는 것들 전에 체크될 수도 있다. 대안으로, 리스트 1 에 대응하는 모션 벡터들은, 리스트 0 에 대응하는 것들 전에 체크될 수도 있다. 대안으로, 모든 공간적 이웃들에서 리스트 X 에 대응하는 모든 모션 벡터들, 그 다음에 리스트 Y (Y 는 1-X 과 동일함) 에 대응하는 모션 벡터들이 순서대로 체크된다. 여기서, X 는, 병치된 픽처가 속하는, 또는 단지 단순히 0 또는 1 로 설정되는 곳을 나타내는 리스트일 수 있다. 공간적 이웃들의 순서는

HEVC 머지 모드에서 사용된 것과 동일할 수 있다.

[0077] '564 출원은 또한, 식별하는 제 1 단계에서 시간적 벡터가 레퍼런스 픽처를 식별하는 것을 포함하지 않는 경우, 도 9 에 도시된 바와 같은 모션 소스 픽처가, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있는, 고정된 픽처, 예를 들어 TMVP 에 대해 사용된 병치된 픽처인 것으로 단순히 설정될 수도 있는 것에 관련한 기법들을 설명한다. 이러한 경우에서, 벡터는 단지, 이러한 고정된 픽처를 가리키는 모션 벡터들로부터 식별될 수도 있다. 이러한 경우에서, 벡터는 단지, 임의의 픽처를 가리키는 모션 벡터들로부터 식별되지만 고정된 픽처를 향해 추가로 스케일링될 수도 있다. 벡터를 식별하는 것이 레퍼런스 픽처를 식별하는 것으로 이루어지는 제 1 스테이지의 경우, 도 9 에 도시된 바와 같은 모션 소스 픽처, 다음의 추가의 체크들 중 하나 이상이 후보 모션 벡터에 대해 적용할 수도 있다. 모션 벡터가 인트라 코딩되는 픽처 또는 슬라이스와 연관되면, 이러한 모션 벡터는 이용 가능하지 않는 것으로 고려되고 벡터로 컨버팅되도록 사용될 수 없다. 모션 벡터가 연관된 픽처에서 (예를 들어, 모션 벡터로 현재 센터 좌표를 추가함으로써) 인트라 블록을 식별하면, 이러한 모션 벡터는 이용 가능하지 않은 것으로 고려되고 벡터로 변환되는데 사용될 수 없다.

[0078] '564 출원은 또한, 벡터를 식별하는 제 1 스테이지에서, 벡터의 컴포넌트들이, 이것이 모션 소스 픽처에서 하부-우측 픽셀 포지션을 식별하도록, (현재 PU 의 절반 폭, 현재 PU 의 절반 높이) 이도록 설정될 수도 있는 것에 관련한 기법들을 설명하고, 이 기법들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있다. 여기서, (x, y) 는 하나의 모션 벡터의 수평 및 수직 컴포넌트들을 나타낸다. 대안으로, 벡터의 컴포넌트들은 (sum(현재 PU 의 절반 폭, M), sum(현재 PU 의 절반 높이, N)) 이도록 설정될 수도 있고, 여기서 함수 sum(a, b) 는 a 및 b 의 합을 리턴한다. 일 예에서, 모션 정보가 4x4 유닛에 저장되는 경우, M 및 N 은 양자 모두 2 와 동일하도록 설정된다. 다른 예에서, 모션 정보가 8x8 유닛에 저장되는 경우, M 및 N 은 양자 모두 4 와 동일하도록 설정된다.

[0079] '564 출원은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있는, 파라미터 세트, 예를 들어 픽처 파라미터 세트의 시퀀스 파라미터 세트에서 ATMVP 가 적용할 때 시그널링되고 있는 서브-블록/서브-PU 사이즈에 관련한 기법들을 설명한다. 이 사이즈는 최소 PU 사이즈에서 CTU 사이즈까지의 범위이다. 사이즈는 또한, 미리-정의 또는 시그널링될 수 있다. 사이즈는, 예를 들어 4x4 만큼 작을 수 있다. 대안으로, 서브-블록/서브-PU 사이즈는 PU 또는 CU 의 사이즈에 기초하여 도출될 수 있다. 예를 들어, 서브-블록/서브-PU 는  $\max(4x4, (CU \text{ 의 폭}) \gg M)$  와 동일하게 설정될 수 있다. M 의 값은 미리정의되거나 또는 비트스트림으로 시그널링될 수 있다.

[0080] '564 출원은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있는, ATMVP 가 새로운 머지 후보로서 고려될 수 있다는 사실로 인해 머지 후보들의 최대 수가 1 만큼 증가되는 것에 관련한 기법들을 설명한다. 예를 들어, 프루닝 후에 머지 후보 리스트에서 최대 5 개의 후보들을 취하는 HEVC 에 비교하여, 머지 후보들의 최대 수는 6 으로 증가될 수 있다. 대안으로, 종래의 TMVP 후보와 프루닝 또는 종래의 TMVP 후보와의 통합은, 머지 후보들의 최대 수가 변하지 않고 유지될 수 있도록 ATMVP 에 대해 수행될 수도 있다. 대안으로, ATMVP 가 이용 가능한 것으로 식별되는 경우, 공간적 이웃하는 후보는 머지 후보 리스트로부터 배제되고, 예를 들어 폐칭 순서에서 최종의 공간적 이웃 후보가 배제된다.

[0081] '564 출원은 또한, 다수의 공간적 이웃 모션 벡터들이 시간적 벡터를 도출하도록 고려되는 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있는, 모션 벡터 유사성이 현재 PU 의 이웃하는 모션 벡터들 뿐만 아니라 모션 벡터와 동일하게 설정되는 특정한 시간적 벡터에 의해 식별된 이웃하는 모션 벡터들에 기초하여 계산될 수도 있는 것에 관련한 기법들을 설명한다. 최고 모션 유사성을 초래하는 것은 최종 시간적 벡터로서 선택될 수도 있다. 하나의 대안에서, 이웃하는 포지션 N 으로부터의 각각의 모션 벡터에 대해, 모션 벡터는 모션 소스 픽처에서 블록 (현재 PU 와 동일한 사이즈) 을 식별하고, 여기서 그 이웃하는 포지션 N 은 모션 정보의 세트를 포함한다. 모션 벡터의 이 세트는 현재 블록의 이웃하는 포지션 N 에서와 같이 모션 정보의 세트와 비교된다. 다른 대안에서, 이웃하는 포지션 N 으로부터의 각각의 모션 벡터에 대해, 모션 벡터는 모션 소스 픽처에서 블록을 식별하고, 여기서 그 이웃하는 포지션들은 모션 정보의 세트들을 포함한다. 모션 벡터의 이들 다수의 세트들은 동일한 상대적 포지션들에서 현재 PU 의 이웃하는 포지션들로부터 모션 정보의 다수의 세트들과 비교된다.

[0082] 모션 정보 유사성은 상기 기법들에 따라 계산될 수도 있다. 예를 들어, 현재 PU 는 MIA1, M1B1, MIA0 및 M1B0 로서 표기된, A1, B1, A0 및 B0 로부터의 모션 정보의 다음의 세트들을 갖는다. 시간적 벡터 (TV) 에 대해, 이것은 모션 소스 픽처에서 PU 에 대응하는 블록을 식별한다. 이러한 블록은 동일한 상대적 A1, B1,

A0 및 B0 포지션들로부터의 모션 정보를 갖고, TMIA1, TMIB1, TMIA0 및 TMIB0 로 표기된다. TV 에 의해 결  
정된 바와 같은 모션 유사성은  $MStv = \sum_{N \in \{A1, B1, A0, B0\}} MVSIm(MI_N, TMI_N)$  로서 계산되고, 여기서  
MVSIm() 은 모션 정보의 2 개의 세트들 (MIN, TMIN) 간의 유사성을 정의한다. 상기 경우들 양자 모두에서,  
모션 유사성 (MVSIm) 이 사용될 수 있고, 여기서 2 개의 입력 파라미터들은 2 개의 모션 정보이고, 각각은 최대  
2 개의 모션 벡터들 및 2 개의 레퍼런스 인덱스들을 포함한다. 리스트 X 에서 모션 벡터들의 각각의 쌍은  
실제로, 상이한 픽처들의 상이한 리스트 X 에서 레퍼런스 픽처들, 현재 픽처 및 모션 소스 픽처와 연관된다.

[0083] 2 개의 모션 벡터들 (MVXN 및 TMVXN) (X 는 0 또는 1 과 동일함) 각각에 대해, 모션 벡터 차이 (MVDXN) 은 상  
기 기법들에 따라,  $MVXN - TMVXN$  로서 계산될 수 있다. 그 후에, 차이 MVSImX 는, 예를 들어  
 $abs(MVDX_N[0]) + abs(MVDX_N[1])$ , 또는  $(MVDX_N[0]*MVDX_N[0] + MVDX_N[1]*MVDX_N[1])$  로서 계산된다. 모션 정보의  
세트들 양자 모두가 이용 가능한 모션 벡터들을 포함하면, MVSIm 은  $MVSIm0 + MVSIm1$  과 동일하게 설정된다.  
모션 차이의 통합된 계산을 갖기 위해, 모션 벡터들 양자 모두는 동일한 고정된 픽처를 향해 스케일링될 필요  
가 있고, 이것은, 예를 들어 현재 픽처의 리스트 X 의 제 1 레퍼런스 픽처 (RefPicListX[0]) 일 수 있다.  
제 1 세트로부터 리스트 X 에서 모션 벡터의 이용 가능성 및 제 2 세트로부터 리스트 X 에서 모션 벡터의 이용  
가능성이 상이하면, 즉 하나의 레퍼런스 인덱스가 -1 인 한편 다른 것은 -1 이 아니면, 이러한 모션 정보의 2  
개의 세트들은 방향 X 에서 유사하지 않은 것으로서 고려된다.

[0084] 2 개의 세트들이 양자 모두의 세트들에서 유사하지 않으면, 상기 기법들에 따라, 최종 MVSIm 함수는, 예를  
들어, 무한대로서 고려될 수도 있는, 큰 값 T 를 리턴할 수도 있다. 대안으로, 모션 정보의 세트들의 쌍에  
대해, 하나가 리스트 X (X 는 0 또는 1 과 동일함) 로부터 예측되지만 리스트 Y (Y 는 1-X 과 동일함) 로부터  
예측되지 않고 다른 것이 동일한 스테이터스를 가지면, 1 과 2 사이의 가중화 (예를 들어, MVSIm 은  $MVSImX * 1.5$   
과 동일함) 가 사용될 수도 있다. 하나의 세트가 단지, 리스트 X 로부터 예측되고 다른 세트가 단지,  
리스트 Y 로부터 예측되는 경우, MVSIm 은 큰 값 (T) 으로 설정된다. 대안으로, 모션 정보의 임의의 세트에  
대해, 하나의 모션 벡터가 이용 가능한 한, 양자 모두의 모션 벡터들이 생성될 것이다. 단지 하나의 모션  
벡터가 (리스트 X 에 대응하여) 이용 가능한 경우에서, 이것은 다른 리스트 Y 에 대응하는 모션 벡터를 형성하  
도록 스케일링된다. 대안으로, 모션 벡터는 현재 PU 의 이웃하는 픽셀들과 (현재 PU 와 동일한 사이즈의)  
블록의 이웃하는 픽셀들 간의 차이들에 기초하여 측정될 수도 있다. 최소 차이를 초래하는 모션 벡터는 최  
종 시간적 벡터로서 선택될 수도 있다.

[0085] '564 출원은 또한, 현재 블록의 시간적 벡터를 도출하는 경우, ATMVP 로 코딩되는 이웃하는 블록들로부터의 시  
간적 벡터들 및/또는 모션 벡터들이 다른 이웃하는 블록들로부터의 모션 벡터들보다 더 높은 우선순위를 가질  
수도 있다는 것에 관한 기법들을 설명하고, 이 기법들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의  
해 수행될 수도 있다. 일 예에서, 이웃하는 블록들 중 단지 시간적 벡터들이 먼저 체크되고, 제 1 이용 가  
능한 블록은 현재 블록의 시간적 벡터로 설정될 수 있다. 단지 이러한 시간적 벡터들이 존재하지 않는  
경우, 정규 모션 벡터들이 추가로 체크된다. 이 경우에서, ATMVP 코딩된 블록들에 대한 시간적 벡터들이 저  
장될 필요가 있다. 다른 예에서, ATMVP 코딩된 이웃하는 블록들로부터의 단지 모션 벡터들이 먼저  
체크되고, 제 1 이용 가능한 블록은 현재 블록의 시간적 벡터로 설정될 수 있다. 단지 이러한 시간적 벡터  
들이 존재하지 않는 경우, 정규 모션 벡터들이 추가로 체크된다. 다른 예에서, ATMVP 코딩된 이웃하는 블록  
들로부터의 단지 모션 벡터들이 먼저 체크되고, 제 1 이용 가능한 블록은 현재 블록의 시간적 벡터로 설정될 수  
있다. 이러한 모션 벡터들이 이용 가능하지 않으면, 시간적 벡터의 체크가 위에서 논의된 방식과 유사하게  
계속된다. 다른 예에서, 이웃하는 블록들로부터의 시간적 벡터들이 먼저 체크되고, 제 1 이용 가능한 블록  
은 현재 블록의 시간적 벡터로 설정될 수 있다. 이러한 모션 벡터들이 이용 가능하지 않으면, 시간적 벡터  
의 체크가 위에서 논의된 방식과 유사하게 계속된다. 다른 예에서, ATMVP 코딩된 이웃하는 블록들의 모션  
벡터들 및 시간적 벡터들이 먼저 체크되고, 제 1 이용 가능한 블록은 현재 블록의 시간적 벡터로 설정될 수 있  
다. 단지 이러한 시간적 벡터들 및 모션 벡터들이 존재하지 않는 경우, 정규 모션 벡터들이 추가로 체크된  
다.

[0086] '564 출원은 또한, 다수의 공간적 이웃 모션 벡터들이 시간적 벡터를 도출하도록 고려되는 경우, 모션 벡터는  
그것이, 픽셀 도메인으로부터 계산되는 왜곡을 최소화하도록 선택될 수도 있고, 예를 들어 템플릿 매칭은, 최소  
의 매칭 비용을 초래하는 것이 최종 시간적 벡터로서 선택되도록 시간적 벡터를 도출하는데 사용될 수도 있다는  
것에 관련한 기법들을 설명한다. 이들 기법들은 또한, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에

의해 수행될 수도 있다.

- [0087] '564 출원은 또한, (모션 소스 픽처에서) 대응하는 블록으로부터 모션 정보의 세트의 도출은, 모션 벡터가 임의의 리스트 X에 대해 대응하는 블록에서 이용 가능한 경우 (모션 벡터를 MVX 이도록 표기함), ATMVP 후보의 현재 서브-PU에 대해, 모션 벡터가 (MVX를 스케일링함으로써) 리스트 X에 대해 이용 가능한 것으로서 고려되는 방식으로 수행된다는 것에 관련한 기법들을 설명한다. 임의의 리스트 X에 대해 대응하는 블록에서 모션 벡터가 이용 가능하지 않으면, 모션 벡터는 리스트 X에 대해 이용 가능하지 않은 것으로 고려된다. 대안으로, 대응하는 블록에서 모션 벡터가 리스트 X에 대해 이용 가능하지 않지만 리스트 1-X에 대해서는 이용 가능한 경우 (1-X가 Y로 표기되고, 모션 벡터를 MVY 이도록 표기함), 모션 벡터는 여전히, (리스트 X에서 타겟 레퍼런스 픽처를 향해 MVY를 스케일링함으로써) 리스트 X에 대해 이용 가능한 것으로서 고려된다. 대안으로, 또는 추가하여, 리스트 X 및 리스트 Y (1-X와 동일함)에 대한 대응하는 블록에서 양자 모두의 모션 벡터들이 이용 가능한 경우, 리스트 X 및 리스트 Y로부터의 모션 벡터들은 직접적으로 스케일링되어, 스케일링에 의해 현재 서브-PU의 2개의 모션 벡터들을 생성하는데 사용될 필요는 없다. 일 예에서, ATMVP 후보를 포물레이팅할 때, TMVP에서 행해진 것과 같은 저-지연 체크가 각각의 서브-PU에 적용한다. 현재 슬라이스의 모든 레퍼런스 픽처 리스트에서 (refPic로 표기된) 모든 픽처에 대해, refPic의 픽처 순서 카운트(POC) 값이 현재 슬라이스의 POC보다 더 작으면, 현재 슬라이스는 저-지연 모드로 고려된다. 이 저-지연 모드에서, 리스트 X 및 리스트 Y로부터의 모션 벡터들이 스케일링되어, 리스트 X 및 리스트 Y 각각에 대한 현재 서브-PU의 모션 벡터들을 생성한다. 저-지연 모드에 있지 않은 경우, MVX 또는 MVY로부터의 단지 하나의 모션 벡터(MVZ)가 선택되고 스케일링되어, 현재 서브-PU에 대한 2개의 모션 벡터들을 생성한다. TMVP와 유사하게, 이러한 경우에서 Z는 collocated\_from\_l0\_flag와 동일하게 설정되고, 이것은 TMVP에서와 같이 병치된 픽처가 현재 픽처의 리스트 X 또는 리스트 Y에 있는지 여부에 의존한다는 것을 의미한다. 대안으로, Z는 다음과 같이 설정된다: 모션 소스 픽처가 리스트 X로부터 식별되면, Z는 X로 설정된다. 대안으로, 추가하여, 모션 소스 픽처들이 양자 모두의 레퍼런스 픽처 리스트들에 속하고 RefPicList0[idx0]이 리스트 0에 먼저 존재하는 모션 소스 픽처이고 RefPicList(1)[idx1]이 리스트 1에 먼저 존재하는 모션 소스 픽처인 경우, idx0가 idx1 이하이면 Z는 0인 것으로 설정되고, 그렇지 않은 경우 Z는 1인 것으로 설정된다.
- [0088] '564 출원은 또한, 모션 소스 픽처를 시그널링하기 위한 기법들을 설명하고, 이 기법들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 수행될 수도 있다. 상세에서, 모션 소스 픽처가 리스트 0 또는 리스트 1로부터의 것인지 여부를 나타내는 플래그는 B 슬라이스에 대해 시그널링된다. 대안으로, 추가하여, 현재 픽처의 리스트 0 또는 리스트 1에 대한 레퍼런스 인덱스는 모션 소스 픽처를 식별하도록 시그널링될 수도 있다.
- [0089] '564 출원은 또한, 시간적 벡터를 식별하는 경우, 벡터가, 그것이 연관된 모션 소스 픽처에서 인트라 코딩된 블록을 가리키면 이용 가능하지 않은 것으로서 고려되는 (따라서, 다른 것들이 고려될 수 있는) 것에 관련된 기법들을 설명하고, 이 기법들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 수행될 수도 있다.
- [0090] 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 공간적 및 시간적 이웃 블록들로부터 블록 (예를 들어, PU)의 서브-블록 (예를 들어, 서브-PU)에 대한 모션 벡터들을 도출하도록 구성될 수도 있다. 이하에서 논의되는 바와 같이, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))는 3 차원 도메인에서 이웃하는 블록들의 정보로부터 PU의 각각의 서브-PU에 대한 모션 벡터를 도출할 수도 있다. 이것은, 이웃하는 블록들이 이전의 코딩된 픽처들에서의 시간적 이웃들 또는 현재 픽처에서의 공간적 이웃들일 수 있다는 것을 의미한다. 이하에서 더 상세히 논의되는 도 10은 예시의 공간-시간적 모션 벡터 예측자 (STMVP) 도출 프로세스를 예시하는 플로우차트이다. 또한, 블릿들 #1, #2, #3, #4, #6, #7, #12, 및 #13에 대하여 전술된 방법들은 STMVP로 직접적으로 확장될 수 있다.
- [0091] 다음의 설명에서, 용어 "블록"은 예측 관련된 정보, 예를 들어 인트라 또는 인트라 예측, 인트라-예측 모드, 모션 정보 등의 저장을 위한 블록-유닛을 지칭하는데 사용된다. 이러한 예측 정보는 저장되고, 미래의 블록들을 코딩, 예를 들어 미래의 블록들에 대한 예측 모드 정보를 예측하기 위해 사용될 수도 있다. AVC 및 HEVC에서, 이러한 블록의 사이즈는 4x4이다.
- [0092] 다음의 설명에서, 'PU'는 이웃하는 블록들로부터 모션 정보를 도출하는 유닛을 나타내기 위해 인트라-코딩된 블록 유닛 및 서브-PU를 나타낸다는 것이 주목된다.
- [0093] 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 단독으로 또는 임의의 조합으로, 다음의 방법들 중 어느 하나를 적용하도록 구성될 수도 있다.

- [0094] 서브-PU들 및 이웃하는 블록들의 사이즈들: 다수의 서브-PU들을 갖는 PU 를 고려하면, 서브-PU 의 사이즈는 대개, 그 이웃하는 블록 사이즈보다 더 크거나 동일하다. 일 예에서, 도 11a 에 도시된 바와 같이, 음영처리된 스퀘어들은 현재 PU 밖에 있는 (소문자들 a, b, ..., i 을 사용하여 표현된) 이웃하는 블록들을 표현하고, (대문자들 A, B, ..., P 을 사용하여 표현된) 나머지 스퀘어들은 현재 PU 에서의 서브-PU들을 표현한다. 서브-PU 및 그 이웃하는 블록들의 사이즈들은 동일하다. 예를 들어, 그 사이즈는 4x4 와 동일하다. 도 11b 는 서브-PU들이 이웃하는 블록들보다 큰 다른 예를 나타낸다. 이 방식에서, 모션 정보 도출을 위해 사용된 이웃하는 블록들의 사이즈들은 모션 정보가 도출되는 서브-블록들의 사이즈들과 동일하거나 이보다 더 작을 수도 있다. 대안으로, 서브-PU들은 비-스퀘어 형상들, 예컨대 장방형 또는 삼각형 형상들을 취할 수도 있다. 또한, 서브-PU 의 사이즈는 슬라이스 헤더에서 시그널링될 수도 있다. 일부 예들에서, 예를 들어, 파라미터 세트에서 서브-블록 또는 서브-PU 사이즈들을 시그널링하는 것에 관하여 위에서 논의된 프로세스는 이들 기법들로 확장될 수도 있다. 예를 들어, 서브-PU 사이즈는 파라미터 세트, 예컨대 시퀀스 파라미터 세트 (SPS) 또는 픽처 파라미터 세트 (PPS) 에서 시그널링될 수도 있다.
- [0095] 도 11a 의 예에 대하여, 비디오 코더가 서브-PU들에 래스터 스캔 순서 (A, B, C, D, E, 등) 를 적용하여 서브-블록들에 대한 모션 예측을 도출한다는 것을 가정한다. 그러나, 다른 스캔 순서들이 또한 적용될 수도 있고, 이들 기법들은 래스터 스캔 순서에만 제한되지 않는다는 것에 주목해야 한다.
- [0096] 이웃하는 블록들은 2 개의 상이한 유형들: 공간적 및 시간적으로 분류될 수도 있다. 공간적 이웃 블록은 현재 서브-PU 에 이웃하고 현재 픽처 또는 슬라이스에 있는 이미 스캐닝된 서브-PU 이거나 이미 코딩된 블록이다. 시간적 이웃 블록은 현재 서브-PU 의 공동-위치된 블록에 이웃하고 이전 코딩된 픽처에서의 블록이다. 일 예에서, 비디오 코더는 현재 PU 와 연관된 모든 레퍼런스 픽처들을 사용하여 시간적 이웃 블록을 획득한다. 다른 예에서, 비디오 코더는 STMVP 도출을 위한 레퍼런스 픽처들의 서브-세트, 예를 들어 각각의 레퍼런스 픽처 리스트의 제 1 엔트리 만을 사용한다.
- [0097] 이들 정의들에 따르면, 서브-PU (A) 에 대해, 도 11a 를 추가로 참조하여, 이전에 코딩된 픽처들에서의 모든 하위 블록들 (a, b, ..., i) 및 그 병치된 블록들은 이용 가능한 것으로서 처리되는 공간적 및 시간적 이웃 블록들이다. 래스터 스캔 순서에 따르면, 블록들 B, C, D, E..., P 는 서브-PU (A) 에 대해 공간적으로 이용 가능하지 않다. 그러나, 모든 서브-PU들 (A 내지 P) 은, 그 모션 정보가 이전의 코딩된 픽처들에서 그 병치된 블록들에서 발견될 수 있기 때문에 서브-PU (A) 에 대해 시간적으로 이용 가능한 이웃 블록들이다. 다른 예로서 서브-PU (G) 를 취한다: 이용 가능한 그 공간적 이웃 블록들은 a, b... 내지 i 로부터, 그리고 A 내지 F 로부터의 것을 포함한다. 또한, 일부 예들에서 소정의 제한이 공간적 이웃 블록들에 적용될 수도 있고, 예를 들어 공간적 이웃 블록들 (즉, a, b... 내지 i) 은 동일한 LCU/슬라이스/타일에 있도록 제한될 수도 있다.
- [0098] 본 개시물의 기법들에 따르면, 비디오 코더 (비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 모든 이용 가능한 이웃 블록들의 서브세트를 선택하여, 각각의 서브-PU 에 대한 모션 정보 또는 모션 필드를 도출할 수도 있다. 각각의 PU 의 도출을 위해 사용된 서브세트는 미리-정의될 수도 있다; 대안으로, 비디오 인코더 (20) 는 슬라이스 헤더, PPS, SPS, 등에서 고 레벨 신호로서 서브세트를 시그널링할 수도 있다 (그리고, 비디오 디코더 (30) 는 이를 나타내는 시그널링된 데이터를 수신할 수도 있다). 코딩 성능을 최적화하기 위해, 서브세트는 각각의 서브-PU 에 대해 상이할 수도 있다. 실제로, 서브세트에 대한 로케이션의 고정된 패턴이 간략함을 위해 바람직하다. 예를 들어, 각각의 서브-PU 는 그 중간 상부의 공간적 이웃, 그 중간 좌측의 공간적 이웃 및 그 중간 하부-우측의 시간적 이웃을 서브세트로서 사용할 수도 있다. 도 11a 의 예에 대하여, (수평으로 해싱된) 서브-PU (J) 를 고려하는 경우, 블록 상부 (F) 및 블록 좌측 (I)(대각선으로 하방-좌측 해싱됨) 은 공간적으로 이용 가능한 블록들이고, 하부-우측 블록 (O)(양자 모두의 방향들에서 대각선으로 해싱됨) 은 시간적으로 이용 가능한 이웃 블록이다. 이러한 서브세트로, 현재 PU 에서의 서브-PU들은 프로세싱 의존성으로 인해 (래스터 스캔 순서와 같은 정의된 순서에서) 순차적으로 프로세싱될 것이다.
- [0099] 부가적으로 또는 대안으로, 서브-PU (J) 를 고려하는 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록 상부 (F) 및 블록 좌측 (I) 을 공간적으로 이용 가능한 이웃 블록들로서, 그리고 하부 블록 (N) 및 우측 블록 (K) 을 시간적으로 이용 가능한 이웃 블록들로서 처리할 수도 있다. 이러한 서브세트로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 프로세싱 의존성으로 인해 순차적으로 현재 PU 에서의 서브-PU들을 프로세싱할 수도 있다.
- [0100] 현재 PU 에서 각각의 서브-PU 의 병렬 프로세싱을 허용하기 위해, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 모션 예측 도출을 위해 일부 서브-PU들에 대해 이웃하는 블록들의 상이한 서브세트를 사용할 수도 있다.

일 예에서, 현재 PU 에 속하지 않는 공간적 이웃 블록들, 예를 들어 블록들  $a, b, \dots, i$  만을 포함하는 서브세트가 정의될 수도 있다. 이 경우에서, 병렬 프로세싱이 가능할 것이다.

[0101] 다른 예에서, 소정의 서브-PU 에 대해, 서브-PU 의 공간적 이웃 블록이 현재 PU 내에 있으면, 그 공간적 이웃 블록의 병치된 블록은 서브세트에 높이고 현재 서브-PU 의 모션 정보를 도출하는데 사용될 수도 있다. 예를 들어, 서브-PU (J) 를 고려하는 경우, 상부 블록 (F) 및 좌측 블록 (I) 및 하부-우측 블록 (O) 의 시간 병치된 블록들은 서브-PU (J) 의 모션을 도출하도록 서브세트로서 선택된다. 이 경우에서, 서브-PU (J) 의 서브세트는 3 개의 시간적 이웃 블록들을 포함한다. 다른 예에서, 부분적으로-병렬의 프로세스가 인에이블될 수도 있고, 여기서 하나의 PU 가 여러 영역들로 스플리팅되고 (여러 서브-PU들을 커버하는) 각각의 영역은 독립적으로 프로세싱될 수 있다.

[0102] 때때로, 이웃하는 블록들이 인트라-코딩되고, 여기서 더 좋은 모션 예측 및 코딩 효율성을 위해 이들 블록들에 대한 대체 모션 정보를 결정하기 위한 규칙을 갖는 것이 바람직하다. 예를 들어, 서브-PU (A) 를 고려하면, 블록들  $b, c$ , 및/또는  $f$  가 인트라-코딩되고,  $a, d, e, g, h$ , 및  $i$  가 인터-코딩되는 경우들이 존재할 수도 있다. 공간적 이웃들에 대해, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 미리-정의된 순서를 사용하여, 먼저 발견된 인터 코딩된 블록의 모션 정보와 인트라-코딩된 블록들의 모션 정보를 집산화할 수도 있다. 예를 들어, 상부의 이웃들의 검색 순서는,  $b, c, d$ , 및  $e$  의 순서를 의미하는, 중간 상부의 이웃에서부터 시작하여 우측으로 최우측 이웃까지로서 설정될 수 있다. 좌측 이웃들의 검색 순서는,  $f, g, h$ , 및  $i$  의 순서를 의미하는, 중간 좌측 이웃에서부터 시작하여 하방으로 최하단 이웃까지로서 설정될 수 있다. 검색 프로세스 전체에 걸쳐 어떤 인터-코딩된 블록도 발견되지 않으면, 상부 또는 좌측 공간적 이웃은 이용 가능하지 않은 것으로 고려된다. 시간적 이웃들에 대해, TMVP 도출에서 지정된 바와 동일한 규칙이 사용될 수 있다. 그러나, 다른 규칙들, 예를 들어 모션 방향, 시간적 거리 (상이한 레퍼런스 픽처들에서의 검색) 및 공간적 로케이션 등에 기초한 규칙들이 또한 사용될 수 있다는 것을 주목해야 한다.

[0103] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물의 기법들에 따라 소정의 서브-PU 에 대한 모션 정보를 도출하기 위해 다음의 방법을 사용할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 먼저, 타겟 레퍼런스 픽처를 결정하고, 모션 벡터 스케일링을 수행할 수도 있다. 각각의 이웃하는 블록에 대해, 모션 벡터 스케일링은 각각의 리스트에서 동일한 레퍼런스 픽처에 이웃하는 블록들의 모션 벡터들 모두를 맵핑하기 위해 각각의 레퍼런스 픽처 리스트에 기초하여 그 모션 벡터에 적용될 수도 있다. 2 개의 단계들이 존재한다: 먼저, 스케일링을 위해 사용될 소스 모션 벡터를 결정한다. 두 번째, 소스 모션 벡터가 프로젝팅되는 타겟 레퍼런스 픽처를 결정한다. 제 1 단계 동안, 여러 방법들이 사용될 수 있다:

[0104] a) 각각의 레퍼런스 리스트에 대해, 모션 벡터 스케일링은 다른 레퍼런스 리스트에서의 모션 벡터로부터 독립적이고; 소정 블록의 모션 정보에 대해, 레퍼런스 리스트에서 모션 벡터가 존재하지 않으면 (예를 들어, 양방향-예측 모드 대신에 단방향-예측 모드), 그 리스트에 대해 어떤 모션 벡터 스케일링도 수행되지 않는다.

[0105] b) 모션 벡터 스케일링은 다른 레퍼런스 리스트에서의 모션 벡터로부터 독립적이지 않다; 소정 블록의 모션 정보에 대해, 레퍼런스 리스트에서 모션 벡터가 이용 가능하지 않으면, 그것은 다른 레퍼런스 리스트에서의 하나로부터 스케일링될 수 있다.

[0106] c) 양자 모두의 모션 벡터들은 (TMVP 에서와 같이) 하나의 미리-정의된 레퍼런스 리스트로부터 스케일링된다.

[0107] 일 예에서, 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 공간적 이웃 블록들의 모션 벡터들을 스케일링하기 위해 상기의 방법 a), 및 시간적 이웃 블록들의 모션 벡터들을 스케일링하기 위해 상기의 방법 c) 를 사용한다. 그러나, 다른 조합들이 다른 예들에서 사용될 수도 있다.

[0108] 제 2 단계에 관해 말하자면, 타겟 레퍼런스 픽처가 이용 가능한 공간적 이웃 블록들의 모션 정보 (예를 들어, 레퍼런스 픽처들) 에 기초하여 소정 규칙에 따라 선택될 수 있다. 이러한 규칙의 일 예는 다수결 규칙, 즉 다수의 블록들에 의해 공유된 레퍼런스 픽처를 선택하는 것이다. 이 경우에서, 동일한 정보가 또한 동일한 규칙을 사용하여 디코더 측에서 추론될 수 있기 때문에, 인코더로부터 디코더로의 타겟 레퍼런스 픽처에 대해 필요한 시그널링이 존재하지 않는다. 대안으로, 이러한 레퍼런스 픽처는 또한, 슬라이스 헤더에서 명시적으로 지정되거나, 또는 일부 다른 방법들에서 디코더로 시그널링될 수도 있다. 일 예에서, 타겟 레퍼런스 픽처는 각각의 레퍼런스 리스트의 제 1 레퍼런스 픽처 ( $refidx = 0$ ) 로서 결정된다.

[0109] 타겟 레퍼런스 픽처를 결정하고 필요에 따라 모션 벡터들을 스케일링한 후에, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 소정의 서브-PU 에 대한 모션 정보를 도출한다. 소정의 서브-PU 에 대한 모션 정보를 갖는 N

개의 이용 가능한 이웃하는 블록들이 존재한다고 가정한다. 먼저, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예측 방향 (*InterDir*) 을 결정한다. 예측 방향을 결정하는 하나의 단순한 방법은 다음과 같다:

[0110] a. *InterDir* 은 0 으로서 초기화되고, 그 후 N 개의 이용 가능한 이웃하는 블록들의 모션 정보를 반복 실행 (loop through);

[0111] b. 리스트 0 에서 적어도 하나의 모션 벡터가 존재하는 경우,  $InterDir = (InterDir \text{ bitwiseOR } 1)$ ;

[0112] c. 리스트 1 에서 적어도 하나의 모션 벡터가 존재하는 경우,  $InterDir = (InterDir \text{ bitwiseOR } 2)$ .

[0113] 여기서, "bitwiseOR" 은 비트 OR 연산 (bitwise OR operation) 을 나타낸다. *InterDir* 의 값은, 이 예에서, 0 (인터 예측이 없음), 1 (리스트 0 에 기초한 인터 예측), 2 (리스트 1 에 기초한 인터 예측), 및 3 (리스트 0 및 리스트 1 에 기초한 인터 예측) 으로서 정의된다.

[0114] 대안으로, 전송된 모션 벡터 스케일링에 대한 타겟 레퍼런스 픽처에 대한 결정과 유사하게, 다수결 규칙은 모든 이용 가능한 이웃하는 블록들의 모션 정보에 기초하여 소정의 서브-PU 에 대한 *InterDir* 의 값을 결정하도록 사용될 수도 있다.

[0115] *InterDir* 이 결정된 후에, 모션 벡터들이 도출될 수 있다. 도출된 *InterDir* 에 기초한 각각의 레퍼런스 리스트에 대해, 위에서 논의된 바와 같이 타겟 레퍼런스 픽처로의 모션 벡터 스케일링을 통해 이용 가능한 M 개의 모션 벡터들 ( $M \leq N$ ) 이 존재할 수도 있다. 레퍼런스 리스트에 대한 모션 벡터는 다음과 같이 도출될 수 있다:

[0116] 
$$(MV_x, MV_y) = ((\sum_{i=0}^M w_i * MV_{xi} + O_i) / \sum_{i=0}^M w_i, (\sum_{j=0}^M w_j * MV_{yj} + O_j) / \sum_{j=0}^M w_j) \quad (1)$$

[0117] 여기서,  $w_i$  및  $w_j$  는 수평 및 수직 모션 컴포넌트 각각에 대한 가중 인자들이고,  $O_i$  및  $O_j$  는 가중 인자들에 의 존적인 오프셋 값들이다.

[0118] 가중 인자들은 다양한 인자들에 기초하여 결정될 수도 있다. 일 예에서, 동일한 규칙이 하나의 PU 내의 모든 서브-PU들에 적용될 수도 있다. 이 규칙은 다음과 같이 정의될 수도 있다:

[0119] • 예를 들어, 가중 인자는 현재 서브-PU 및 대응하는 이웃 블록의 로케이션 거리에 기초하여 결정될 수 있다.

[0120] • 다른 예에서, 가중 인자는 또한, 스케일링 전에 대응하는 이웃 블록의 모션 벡터와 연관된 레퍼런스 픽처와 타겟 레퍼런스 픽처 간의 POC 거리에 기초하여 결정될 수 있다.

[0121] • 또 다른 예에서, 가중 인자는 모션 벡터 차이 또는 일관성에 기초하여 결정될 수 있다.

[0122] • 단순함을 위해, 모든 가중 인자들은 또한, 1 로 설정될 수도 있다.

[0123] 대안으로, 상이한 규칙들이 하나의 PU 내의 서브-PU들에 적용될 수도 있다. 예를 들어, 상기 규칙은, 또한 제 1 로우/제 1 컬럼에 위치한 서브-PU들에 대해 적용될 수도 있고, 시간적 이웃 블록들로부터 도출된 모션 벡터들에 대한 가중 인자들은 0 으로 설정되는 한편, 나머지 블록들에 대해 공간적 이웃 블록들로부터 도출된 모션 벡터들에 대한 가중 인자들은 0 으로 설정된다.

[0124] 실제로, 위의 방정식들은 있는 그대로 구현되거나, 또는 쉬운 구현을 위해 단순화될 수도 있다는 것을 주목해야 한다. 예를 들어, 나눗셈 또는 부동 소수점 연산을 회피하기 위해, 고정 소수점 연산이 사용되어 상기 방정식을 근사할 수도 있다. 하나의 인스턴스는, 3 으로 나누는 것을 회피하기 위해, 대신에 곱셈 및 비트-시프트로 나눗셈 연산을 대체하도록 43/128 로 곱하는 것을 선택할 수도 있다. 이들 구현에서의 변화들은 본 개시물의 기법들의 동일한 사상 하에서 커버되는 것으로 고려되어야 한다.

[0125] 부가적으로 또는 대안으로, 프로세스가 2 개의 모션 벡터들을 인코딩하는 경우, 방정식 (1) 은 이하의 방정식 (2) 로 치환될 수도 있다:

[0126] 
$$(MV_x, MV_y) = ((\sum_{i=0}^1 MV_{xi}) / 2, (\sum_{i=0}^1 MV_{yi}) / 2) \quad (2)$$

[0127] 부가적으로 또는 대안으로, 프로세스가 3 개의 모션 벡터들을 인코딩하는 경우, 방정식 (1) 은 이하의 방정식 (3) 으로 치환될 수도 있다:

$$(MV_x, MV_y) = ((\sum_{i=0}^2 MV_{xi} + \text{sign}(\sum_{i=0}^2 MV_{xi}) * 1) * 43/128, (\sum_{i=0}^2 MV_{yi} + \text{sign}(\sum_{i=0}^2 MV_{yi}) * 1) * 43/128)) \quad (3)$$

[0128]

[0129]

부가적으로 또는 대안으로, 프로세스가 4 개의 모션 벡터들을 인코딩하는 경우, 방정식 (1) 은 이하의 방정식 (4) 로 치환될 수도 있다:

$$(MV_x, MV_y) = ((\sum_{i=0}^3 MV_{xi} + \text{sign}(\sum_{i=0}^3 MV_{xi}) * 2)/4,$$

[0130]

$$(\sum_{i=0}^3 MV_{yi} + \text{sign}(\sum_{i=0}^3 MV_{yi}) * 2)/4) \quad (4)$$

[0131]

여기서,  $\text{sign}(t)$  는  $t$  가 양의 값인 경우 1 이고,  $t$  가 음의 값인 경우 -1 이다.

[0132]

부가적으로 또는 대안으로, 비-선형 연산이 또한, 중간값 필터 (median filter) 와 같은 모션 벡터들을 도출하도록 적용될 수도 있다.

[0133]

비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 이들 기법들에 대한 모션 벡터 이용 가능성을 결정할 수도 있다. 각각의 서브-PU 의 모션 벡터 예측자들이 이용 가능한 경우에도, STMVP 모드는 하나의 PU 에 대해 이용 가능하지 않는 것으로 리셋될 수도 있다. 예를 들어, 일단 각각의 서브-PU 의 모션 벡터 예측자가 소정의 PU 에 대해 도출되면, 일부 이용 가능성 체크들이 수행되어 STMVP 모드가 소정의 PU 에 대해 이용 가능해져야 하는지를 결정한다. 이러한 연산은, STMVP 모드가 소정의 PU 에 대해 최종적으로 선택되는 것이 매우 쉽지 않은 경우들을 제거하는데 사용된다. STMVP 모드가 이용 가능하지 않은 경우, 모드 시그널링은 STMVP 을 포함하지 않는다. STMVP 모드가 머지 리스트에 STMVP 를 삽입함으로써 구현되는 경우에서, STMVP 모드가 이용 가능하지 않은 것으로 결정되는 경우 머지 리스트는 이 STMVP 후보를 포함하지 않는다. 결과적으로, 시그널링 오버헤드가 감소될 수도 있다.

[0134]

하나의 PU 를 M 개의 서브-PU들로 파티셔닝하는 것을 고려한다. 일 예에서, M 개의 서브-PU들 중에서 N1 ( $N1 \leq M$ ) 개의 서브-PU들이 동일한 모션 벡터 예측자 (즉, 동일한 모션 벡터들 및 동일한 레퍼런스 픽처 인덱스들) 을 가지면, N1 이 임계보다 작고 또는 예측자가 머지 리스트에서 (더 작은 머지 인덱스를 갖는) 다른 모션 벡터 예측자들과 상이한 경우, STMVP 만이 이용 가능해진다. 다른 예에서, STMVP 모드 하의 N2 ( $N2 \leq M$ ) 개의 서브-PU들이 ATMVP 하의 대응하는 서브-PU들과 동일한 모션 벡터 예측자들을 공유하면, N2 가 다른 임계보다 더 작은 경우 STMVP 만이 이용 가능해진다. 일 예에서, N1 및 N2 에 대한 양자 모두의 임계들은 M 과 동일하게 설정된다.

[0135]

비디오 인코더 (20) 및 비디오 디코더 (30) 는 그 후, 도출된 모션 예측자들을 후보 리스트, 예를 들어 머지 리스트 안에 삽입할 수도 있다. STMVP 후보가 이용 가능하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 STMVP 후보를 후보 리스트 (예를 들어, 머지 리스트) 안에 삽입할 수도 있다. 상기 블럭 #1 에서의 프로세스는 확장될 수 있고, STMVP 후보는 ATMVP 후보 전 또는 후에 삽입될 수 있다. 일 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 머지 리스트에서 ATMVP 후보 직후에 STMVP 를 삽입한다.

[0136]

비디오 인코더 (20) 는 또한, 신택스 데이터, 예컨대 블록-기반의 신택스 데이터, 프레임-기반의 신택스 데이터, 및 GOP-기반의 신택스 데이터를 예를 들어, 프레임 헤더, 블록 헤더, 슬라이스 헤더, 또는 GOP 헤더에서 비디오 디코더 (30) 로 전송할 수도 있다. GOP 신택스 데이터는 각각의 GOP 에서 다수의 프레임들을 설명할 수도 있고, 프레임 신택스 데이터는 대응하는 프레임을 인코딩하는데 사용된 인코딩/예측 모드를 나타낼 수도 있다.

[0137]

비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 임의의 다양한 적합한 인코더 또는 디코더 회로부, 적용 가능한 바와 같이, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적 회로 (ASIC) 들, 필드 프로그램가능 게이트 어레이 (FPGA) 들, 이산 로직 회로부, 소프트웨어, 하드웨어, 펌웨어, 또는 이들의 임의의 조합들로서 구현될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이들 중 어느 하나는 결합형 비디오 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 예컨대 셀룰러 전화기를 포함할 수도 있다.

[0138]

도 2 는 어드밴스드 시간적 모션 벡터 예측 (ATMVP) 에 대한 기법들을 구현할 수도 있는 비디오 인코더 (20) 의



일 예를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내에서 비디오 블록들의 인트라- 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 소정의 비디오 프레임 또는 픽처 내의 비디오에서 공간적 리던던시를 감소시키거나 제거하기 위해 공간적 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내의 비디오에서 시간적 리던던시를 감소시키거나 제거하기 위해 시간적 예측에 의존한다. 인트라-모드 (I 모드) 는 여러 공간 기반의 코딩 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 및 양방향 예측 (B 모드) 과 같은 인터-모드들은 여러 시간 기반의 코딩 모드들 중 임의의 것을 지칭할 수도 있다.

[0139] 도 2 에 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩될 비디오 프레임 내의 현재 비디오 블록을 수신한다. 도 2 의 예에서, 비디오 인코더 (20) 는 모드 선택 유닛 (40), 레퍼런스 픽처 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 모드 선택 유닛 (40) 은 이어서, 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라-예측 유닛 (46), 및 파티션 유닛 (48) 을 포함한다. 비디오 블록 복원을 위해, 비디오 인코더 (20) 는 또한, 역 양자화 유닛 (58), 역 변환 프로세싱 유닛 (60), 및 합산기 (62) 를 포함한다. 디블록킹 필터 (도 2 에 도시되지 않음) 는 또한, 복원된 비디오로부터 블록화 아티팩트들을 제거하기 위해 블록 경계들을 필터링하도록 포함될 수도 있다. 원하는 경우, 디블록킹 필터는 통상적으로 합산기 (62) 의 출력을 필터링할 것이다. 추가의 필터들 (인 루프 또는 포스트 루프) 이 또한, 디블록킹 필터에 추가되어 사용될 수도 있다. 이러한 필터들은 간결함을 위해 도시되지 않았으나, 원한다면 (인-루프 필터로서) 합산기 (50) 의 출력을 필터링할 수도 있다.

[0140] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 다수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 하나 이상의 레퍼런스 프레임들에서의 하나 이상의 블록들에 대한 수신된 비디오 블록의 인트라-예측 코딩을 수행하여, 시간적 예측을 제공한다. 인트라-예측 유닛 (46) 은 대안으로, 코딩될 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 대한 수신된 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 예측을 제공할 수도 있다. 비디오 인코더 (20) 는, 예를 들어 비디오 데이터의 각각의 블록에 대한 적합한 코딩 모드를 선택하기 위해 다수의 코딩 과정들을 수행할 수도 있다.

[0141] 또한, 파티션 유닛 (48) 은 이전의 코딩 과정들에서 이전의 파티셔닝 스킴들의 평가에 기초하여, 비디오 데이터의 블록들을 서브-블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛 (48) 은 초기에, 프레임 또는 슬라이스를 LCU들로 파티셔닝하고, 이 LCU들 각각을 레이트-왜곡 분석 (예를 들어, 레이트-왜곡 최적화) 에 기초하여 서브-CU들로 파티셔닝할 수도 있다. 모드 선택 유닛 (40) 은 또한, 서브-CU 들로의 LCU 의 파티셔닝을 나타내는 쿼트트리 데이터 구조를 생성할 수도 있다. 쿼트트리의 리프-노드 CU들은 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다.

[0142] 모드 선택 유닛 (40) 은, 예를 들어 여러 결과들에 기초하여 코딩 모드들, 인트라 또는 인터 중 하나를 선택하고, 이 결과의 인트라- 또는 인터-코딩된 블록을 합산기 (50) 에 제공하여 잔차 블록 데이터를 생성하며 합산기 (62) 에 제공하여 레퍼런스 프레임으로서 사용하기 위한 인코딩된 블록을 복원할 수도 있다. 모드 선택 유닛 (40) 은 또한, 선택스 엘리먼트들, 예컨대 모션 벡터들, 인트라-모드 표시자들, 파티션 정보, 및 다른 이러한 선택스 정보를 엔트로피 인코딩 유닛 (56) 에 제공한다.

[0143] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적 목적들을 위해 별개로 예시되어 있다. 모션 추정 유닛 (42) 에 의해 수행된 모션 추정은 모션 벡터들을 생성하는 프로세스이며, 이것은 비디오 블록들에 대한 모션을 추정한다. 모션 벡터는, 예를 들어 현재 프레임 (또는 다른 코딩된 유닛) 내의 코딩되고 있는 현재 블록에 대한 레퍼런스 프레임 (또는 다른 코딩된 유닛) 내의 예측 블록에 대해 현재 비디오 프레임 또는 픽처 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다. 예측 블록은 픽셀 차이의 관점에서 코딩될 블록과 밀접하게 일치하는 것으로 발견된 블록이고, 픽셀 차이는 절대 차의 합 (SAD), 제곱 차의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 레퍼런스 픽처 메모리 (64) 에 저장된 레퍼런스 픽처들의 서브-정수 픽셀 포지션들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 레퍼런스 픽처의 1/4 픽셀 포지션들, 1/8 픽셀 포지션들, 또는 다른 분수의 픽셀 포지션들의 값들을 보간할 수도 있다. 따라서, 모션 추정 유닛 (42) 은 전픽셀 (full pixel) 포지션들 및 분수적 픽셀 포지션들에 대한 모션 검색을 수행하고 분수적 픽셀 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0144] 모션 추정 유닛 (42) 은 PU 의 포지션을 레퍼런스 픽처의 예측 블록의 포지션에 비교함으로써 인트라-코딩된 슬라

이스에서 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 레퍼런스 픽처는 제 1 레퍼런스 픽처 리스트 (List 0) 또는 제 2 레퍼런스 픽처 리스트 (List 1) 로부터 선택될 수도 있고, 이들 각각은 레퍼런스 픽처 메모리 (64) 에 저장된 하나 이상의 레퍼런스 픽처들을 식별한다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0145] 모션 보상 유닛 (44) 에 의해 수행된 모션 보상은 모션 추정 유닛 (42) 에 의해 결정된 모션 벡터에 기초한 예측 블록의 페치 (fetch) 또는 생성을 수반할 수도 있다. 다시, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 일부 예들에서, 기능적으로 통합될 수도 있다. 현재 비디오 블록의 PU 에 대한 모션 벡터를 수신 시에, 모션 보상 유닛 (44) 은 레퍼런스 픽처 리스트들 중 하나에 모션 벡터가 가리키는 예측 블록을 위치시킬 수도 있다. 합산기 (50) 는, 이하에서 논의되는 바와 같이, 픽셀 차이 값들을 형성하는, 코딩되고 있는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산함으로써 잔차 비디오 블록을 형성한다. 일반적으로, 모션 추정 유닛 (42) 은 루마 컴포넌트들에 대해 모션 추정을 수행하고, 모션 보상 유닛 (44) 은 크로마 컴포넌트들 및 루마 컴포넌트들 양자 모두에 대한 루마 컴포넌트들에 기초하여 계산된 모션 벡터들을 사용한다. 모드 선택 유닛 (40) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0146] 모드 선택 유닛 (40) 은 또한, 블록 (예를 들어, PU) 에 대한 서브-블록 (예를 들어, 서브-PU) 모션 도출 모드를 선택할 수도 있다. 즉, 모드 선택 유닛 (40) 은, 어느 인코딩 과정 (및 이에 따른 예측 모드를 포함하는 인자들의 세트) 이 바람직한 레이트-왜곡 최적화 (RDO) 특징을 산출하는지를 결정하기 위해 인코딩 과정들의 범위 중에서, 예측 모드들을 포함하는 다양한 인코딩 인자들을 비교할 수도 있다. 모드 선택 유닛 (40) 이 비디오 데이터의 블록 (예를 들어, PU) 에 대한 서브-블록 모션 정보 도출 모드를 선택하는 경우, 모션 보상 유닛 (44) 은 본 개시물의 기법들을 사용하여 블록을 예측할 수도 있다.

[0147] 특히, 서브-블록 모션 정보 도출 모드를 사용하여, 모션 보상 유닛 (44) 은 블록의 서브-블록들에 대한 모션 정보를 도출할 수도 있다. 예를 들어, 모션 보상 유닛 (44) 은, 각각의 서브-블록에 대해 2 이상의 이웃하는 서브-블록들에 대한 모션 정보를 결정하고, 이웃하는 서브-블록들에 대한 모션 정보로부터 서브-블록에 대한 모션 정보를 도출할 수도 있다. 이웃하는 서브-블록들은, 예를 들어 공간적 및/또는 시간적 이웃하는 서브-블록들을 포함할 수도 있다. 일 예에서, 모션 보상 유닛 (44) 은 도 11a 에 대하여 더 상세히 이하에서 논의되는 바와 같이, 좌측-이웃하는 공간적 서브-블록, 상부-이웃하는 공간적 서브-블록, 및 하부-우측의 시간적 이웃하는 서브-블록에 대한 모션 정보 (예를 들어, 모션 벡터들) 를 평균함으로써 각각의 서브-블록에 대한 모션 정보를 도출한다. 다른 예들에서, 모션 보상 유닛 (44) 은, 예를 들어 식 (1)-(4) 중 하나를 사용하여 각각의 서브-블록에 대한 모션 정보를 도출할 수도 있다. 모션 보상 유닛 (44) 은 서브-블록들 각각에 대한 도출된 모션 정보를 사용하여, 서브-블록들에 대한 예측 데이터를 결정할 수도 있다. 서브-블록들 각각에 대해 이 예측 데이터를 취출함으로써, 모션 보상 유닛 (44) 은 서브-블록 모션 정보 도출 모드를 사용하여 현재 블록에 대한 예측된 블록을 생성한다.

[0148] 인트라-예측 유닛 (46) 은, 전송된 바와 같이 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 수행된 인트라-예측에 대한 대안으로서, 현재 블록을 인트라-예측할 수도 있다. 특히, 인트라-예측 유닛 (46) 은 현재 블록을 인코딩하는데 사용할 인트라-예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라-예측 유닛 (46) 은, 예를 들어 별개의 인코딩 과정들 동안 다양한 인트라-예측 모드들을 사용하여 현재 블록을 인코딩할 수도 있고, 인트라-예측 유닛 (46) (또는, 일부 예들에서 모드 선택 유닛 (40)) 은 테스트된 모드들로부터 사용할 적합한 인트라-예측 모드를 선택할 수도 있다.

[0149] 예를 들어, 인트라-예측 유닛 (46) 은 다양한 테스트된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 사용하여 레이트-왜곡 값들을 계산하고, 테스트된 모드들 중에서 최선의 레이트-왜곡 특징들을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로 인코딩된 블록과 원래의, 인코딩된 블록을 생성하도록 인코딩되었던 인코딩되지 않은 블록 간의 왜곡 (또는 에러) 의 양, 뿐만 아니라 인코딩된 블록을 생성하는데 사용된 비트레이트 (즉, 비트들의 수) 를 결정한다. 인트라-예측 유닛 (46) 은 다양한 인코딩된 블록들에 대한 레이트들 및 왜곡들로부터의 비율들을 계산하여, 어느 인트라-예측 모드가 블록에 대한 최선의 레이트-왜곡 값을 보이는지를 결정할 수도 있다.

[0150] 블록에 대한 인트라-예측 모드를 선택한 후에, 인트라-예측 유닛 (46) 은 블록에 대해 선택된 인트라-예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (56) 에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 선택

된 인트라-예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 송신된 비트스트림에서 구성 데이터를 포함할 수도 있고, 이 구성 데이터는 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 변경된 인트라-예측 모드 인덱스 테이블들 (또한, 코드워드 맵핑 테이블들로서 지칭됨), 각종 블록들에 대한 인코딩 컨텍스트들의 정의들, 및 가장 가능성 있는 인트라-예측 모드의 표시들, 인트라-예측 모드 인덱스 테이블, 및 컨텍스트들 각각에 대해 사용하는 변경된 인트라-예측 모드 인덱스 테이블을 포함할 수도 있다.

[0151] 비디오 인코더 (20) 는 코딩되고 있는 원래의 비디오 블록으로부터 모드 선택 유닛 (40) 으로부터의 예측 데이터를 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50) 는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 프로세싱 유닛 (52) 은 변환, 예컨대 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 유닛 (52) 은 개념적으로 DCT 와 유사한 다른 변환들을 수행할 수도 있다. 웨이블릿 변환들, 정수 변환들, 서브-대역 변환들 또는 다른 유형들의 변환들이 또한 사용될 수 있다. 임의의 경우에서, 변환 프로세싱 유닛 (52) 은 잔차 블록에 변환을 적용하여, 잔차 변환 계수들의 블록을 생성한다. 변환은 픽셀 값 도메인에서 변환 도메인, 예컨대 주파수 도메인으로 잔차 정보를 컨버팅할 수도 있다. 변환 프로세싱 유닛 (52) 은 결과의 변환 계수들을 양자화 유닛 (54) 으로 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킬 수도 있다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 일부 예들에서, 양자화 유닛 (54) 은 그 후, 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안으로, 엔트로피 인코딩 유닛 (56) 은 스캔을 수행할 수도 있다.

[0152] 양자화 다음에, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트-적응적 가변 길이 코딩 (CAVLC), 컨텍스트-적응적 이진 산술 코딩 (CABAC), 신덱스 기반 컨텍스트-적응적 이진 산술 코딩 (SBAC), 확률 구간 파티셔닝 엔트로피 (PIPE) 코딩, 또는 다른 엔트로피 코딩 기법을 수행할 수도 있다. 컨텍스트-기반 엔트로피 코딩의 경우에서, 컨텍스트는 이웃하는 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 코딩 다음에, 인코딩된 비트스트림은 다른 디바이스 (예를 들어, 비디오 디코더 (30)) 로 송신되거나 나중의 송신 또는 취출을 위해 아카이빙될 수도 있다.

[0153] 또한, 엔트로피 인코딩 유닛 (56) 은 비디오 데이터의 다양한 블록들에 대한 다양한 다른 신덱스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 비디오 데이터의 각각의 CU 의 각각의 PU 에 대한 예측 모드를 나타내는 신덱스 엘리먼트들을 인코딩할 수도 있다. 인트라-예측이 PU 에 대해 표시되는 경우, 엔트로피 인코딩 유닛 (56) 은, 모션 벡터가 머지 모드 또는 어드밴스드 모션 벡터 예측 (AMVP) 을 사용하여 인코딩되는지 여부를 포함할 수도 있는, 모션 정보를 인코딩할 수도 있다. 어느 한 경우에서, 비디오 인코더 (20) 는, 모션 정보가 예측될 수도 있는 후보들 (PU 에 공간적 및/또는 시간적 이웃 블록들) 을 포함하는 후보 리스트를 형성한다. 본 개시물의 기법들에 따르면, 후보 리스트는, 서브-블록 모션 정보 도출 모드가 PU 에 대해 사용된다는 것을 나타내는 후보를 포함할 수도 있다. 또한, 엔트로피 인코딩 유닛 (56) 은, 후보들 중 어느 것이 사용될지를 나타내는 후보 리스트 안에 후보 인덱스를 인코딩할 수도 있다. 따라서, 서브-블록 모션 정보 도출 모드가 선택되면, 엔트로피 인코딩 유닛 (56) 은 서브-블록 모션 정보 도출 모드를 나타내는 후보를 지칭하는 후보 인덱스를 인코딩한다.

[0154] 역 양자화 유닛 (58) 및 역 변환 유닛 (60) 은, 각각 역 양자화 및 역 변환을 적용하여, 예를 들어 레퍼런스 블록으로서 추후 사용을 위해 픽셀 도메인에서 잔차 블록을 복원한다. 모션 보상 유닛 (44) 은 레퍼런스 픽처 메모리 (64) 의 프레임들 중 하나의 예측 블록에 잔차 블록을 가산함으로써 레퍼런스 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한, 모션 추정에서 사용하기 위한 서브-정수 픽셀 값들을 계산하기 위해 복원된 잔차 블록에 하나 이상의 보간 필터들을 적용할 수도 있다. 합산기 (62) 는 복원된 잔차 블록을 모션 보상 유닛 (44) 에 의해 생성된 모션 보상된 예측 블록에 가산하여 레퍼런스 픽처 메모리 (64) 에 저장하기 위한 복원된 비디오 블록을 생성한다. 복원된 비디오 블록은 후속하는 비디오 프레임에서의 블록을 인트라 코딩하기 위한 레퍼런스 블록으로서 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 사용될 수도 있다.

[0155] 이 방식에서, 비디오 인코더 (20) 는, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타내는 비디오 데이터의 현재 블록에 대한 모션 예측 후보를 결정하고, 그 결정에 응답하여 서브-블록들 각각에 대해 현재 블록을 서브-블록들로 파티셔닝하고, 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하며, 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하도록 구성된 비디오 인코더의 예를 나타낸다. 즉, 비디오 인코더 (20) 는 본 개시물의 기법들을 사용하여 비디오 데이터의 블록들을 인코딩

및 디코딩한다.

- [0156] 도 3 은 어드밴스드 시간적 모션 벡터 예측 (ATMVP) 에 대한 기법들을 구현할 수도 있는 비디오 디코더 (30) 의 일 예를 예시하는 블록이다. 도 3 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (70), 모션 보상 유닛 (72), 인트라-예측 유닛 (74), 역 양자화 유닛 (76), 역변환 유닛 (78), 레퍼런스 픽처 메모리 (82) 및 합산기 (80) 를 포함한다. 비디오 디코더 (30) 는, 일부 예들에서 비디오 인코더 (20)(도 2) 에 대하여 설명된 인코딩 과정에 일반적으로 상반되는 디코딩 과정을 수행할 수도 있다. 모션 보상 유닛 (72) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있는 한편, 인트라-예측 유닛 (74) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 인트라-예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.
- [0157] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 표현하는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 비트스트림을 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들 또는 인트라-예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (70) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 모션 보상 유닛 (72) 으로 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서의 신택스 엘리먼트들을 수신할 수도 있다.
- [0158] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩되는 경우, 인트라-예측 유닛 (74) 은 현재 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터 및 시그널링된 인트라-예측 모드에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오-프레임이 인터 코딩된 (즉, B, P 또는 GPB) 슬라이스로서 코딩되는 경우, 모션 보상 유닛 (72) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 다른 신택스 엘리먼트들 및 모션 벡터들에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 레퍼런스 픽처 리스트들 중 하나 내의 레퍼런스 픽처들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는, 레퍼런스 픽처 메모리 (82) 에 저장된 레퍼런스 픽처들에 기초하여 디폴트 구성 기법들을 사용하여 레퍼런스 프레임 리스트들, 리스트 0 및 리스트 1 을 구성할 수도 있다.
- [0159] 모션 보상 유닛 (72) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (72) 은 비디오 슬라이스의 비디오 블록들을 코딩하기 위해 사용된 예측 모드 (예를 들어, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 유형 (예를 들어, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 하나 이상의 레퍼런스 픽처 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 스테이터스, 및 현재 비디오 슬라이스의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위해, 수신된 신택스 엘리먼트들 중 일부를 사용한다.
- [0160] 모션 보상 유닛 (72) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (72) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 사용되는 것과 같이 보간 필터들을 사용하여 레퍼런스 블록들의 서브-정수 픽셀들에 대한 보간된 값들을 계산할 수도 있다. 이 경우에서, 모션 보상 유닛 (72) 은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 사용된 보간 필터들을 결정하고, 이 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.
- [0161] 본 개시물의 기법들에 따르면, 엔트로피 디코딩 유닛 (70) 은 후보 리스트를 지칭하는 후보 인덱스에 대한 값을 디코딩하고, 블록, 예컨대 PU 가 인터-예측을 사용하여 예측되는 경우, 후보 인덱스의 값을 모션 보상 유닛 (72) 으로 패스한다. 후보 인덱스의 값은, 서브-블록 모션 정보 도출 모드를 사용하여 블록이 예측된다는 것을 나타내는 후보 리스트에서의 후보를 지칭할 수도 있다. 후보 인덱스의 값이, 블록이 서브-블록 모션 정보 도출 모드를 사용하여 예측된다는 것을 나타내는 후보 리스트에서의 후보를 지칭하면, 모션 보상 유닛 (72) 은 서브-블록 모션 정보 도출 모드를 사용하여 블록에 대한 예측된 블록을 생성할 수도 있다.
- [0162] 보다 특히, 서브-블록 모션 정보 도출 모드를 사용하여, 모션 보상 유닛 (72) 은 블록의 서브-블록들에 대한 모션 정보를 도출할 수도 있다. 예를 들어, 모션 보상 유닛 (72) 은, 각각의 서브-블록에 대해 2 이상의 이웃하는 서브-블록들에 대한 모션 정보를 결정하고, 이웃하는 서브-블록들에 대한 모션 정보로부터 서브-블록에 대한 모션 정보를 도출할 수도 있다. 이웃하는 서브-블록들은, 예를 들어 공간적 및/또는 시간적 이웃하는 서브-블록들을 포함할 수도 있다. 일 예에서, 모션 보상 유닛 (72) 은 도 11a 에 대하여 더 상세히 이하에서 논의되는 바와 같이, 좌측-이웃하는 공간적 서브-블록, 상부-이웃하는 공간적 서브-블록, 및 하부-우측의 시간

적 이웃하는 서브-블록에 대한 모션 정보 (예를 들어, 모션 벡터들) 를 평균함으로써 각각의 서브-블록에 대한 모션 정보를 도출한다. 다른 예들에서, 모션 보상 유닛 (72) 은, 예를 들어 식 (1)-(4) 중 하나를 사용하여 각각의 서브-블록에 대한 모션 정보를 도출할 수도 있다. 모션 보상 유닛 (72) 은 서브-블록들 각각에 대한 도출된 모션 정보를 사용하여, 서브-블록들에 대한 예측 데이터를 결정할 수도 있다. 서브-블록들 각각에 대해 이 예측 데이터를 취출함으로써, 모션 보상 유닛 (72) 은 서브-블록 모션 정보 도출 모드를 사용하여 현재 블록에 대한 예측된 블록을 생성한다.

[0163] 역 양자화 유닛 (76) 은 비트스트림으로 제공되고 엔트로피 디코딩 유닛 (70) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉 양자화해제한다. 역 양자화 프로세스는 양자화의 정도, 및 마찬가지로 적용되어야 하는 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스의 각각의 비디오 블록에 대해 비디오 디코더 (30) 에 의해 계산된 양자화 파라미터 (QPY) 의 사용을 포함할 수도 있다.

[0164] 역변환 유닛 (78) 은, 픽셀 도메인에서 잔차 블록들을 생성하기 위해 변환 계수들에 대해 역 변환, 예를 들어, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 적용한다.

[0165] 모션 보상 유닛 (72) 이 모션 벡터들 및 다른 신텍스 엘리먼트들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후에, 비디오 디코더 (30) 는 역 변환 프로세싱 유닛 (78) 으로부터의 잔차 블록들을 모션 보상 유닛 (72) 에 의해 생성된 대응하는 예측 블록들과 합함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (80) 는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 원하는 경우, 블록화 아티팩트들을 제거하기 위해 디코딩된 블록들을 필터링하도록 디블록킹 필터가 또한, 적용될 수도 있다. (코딩 루프에서 또는 코딩 루프 후에) 다른 루프 필터들이 또한, 픽셀 트랜지션들을 평활화하거나, 그렇지 않으면 비디오 품질을 개선시키는데 사용될 수도 있다. 그 후, 소정의 프레임 또는 픽처에서의 디코딩된 비디오 블록들은 레퍼런스 픽처 메모리 (82) 에 저장되고, 이 레퍼런스 픽처 메모리 (82) 는 후속의 모션 보상을 위해 사용된 레퍼런스 픽처들을 저장한다. 레퍼런스 픽처 메모리 (82) 는 또한, 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에서의 추후의 프리젠테이션을 위해 디코딩된 비디오를 저장한다.

[0166] 이 방식에서, 비디오 디코더 (30) 는, 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다는 것을 결정하고, 그 결정에 응답하여 현재 블록을 서브-블록들로 파티셔닝하고, 서브-블록들 각각에 대해 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하며, 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하도록 구성된 비디오 디코더의 예를 나타낸다.

[0167] 도 4 는 HEVC 에서 공간적 이웃하는 후보들을 예시하는 개념도이다. 공간적 MV 후보들은 특정 PU (PU0) 에 대해, 도 4 상에 도시된 이웃하는 블록들로부터 도출되지만, 블록들로부터 후보들을 생성하는 방법들은 머지 및 AMVP 모드들에 대해 상이하다.

[0168] 머지 모드에서, 최대 4 개의 공간적 MV 후보들이 넘버들을 갖고 도 4(a) 에 도시된 순서들로 도출될 수 있고, 이 순서는 다음과 같다: 도 4(a) 에 도시된 바와 같이, 좌측 (0, A1), 상부 (1, B1), 상부-우측 (2, B0), 아래-좌측 (3, A0), 및 상부 좌측 (4, B2). 즉, 도 4(a) 에서, 블록 (100) 은 PU0 (104A) 및 PU1 (104B) 를 포함한다. 비디오 코더가 머지 모드를 사용하여 PU0 (104A) 에 대한 모션 정보를 코딩하는 경우, 비디오 코더는 공간적 이웃 블록들 (108A, 108B, 108C, 108D, 및 108E) 로부터의 모션 정보를, 그 순서로 후보 리스트에 추가한다. 블록들 (108A, 108B, 108C, 108D, 및 108E) 은 또한, HEVC 에서와 같이, 블록들 (A1, B1, B0, A0, 및 B2) 로서 각각 지칭될 수도 있다.

[0169] AVMP 모드에서, 이웃하는 블록들은 2 개의 그룹들로 분할된다: 도 4(b) 상에 도시된 바와 같이 좌측 그룹은 블록들 (0 및 1) 을 포함하고, 상부의 그룹은 블록들 (2, 3, 및 4) 를 포함한다. 이들 블록들은, 각각 도 4(b) 에서 블록들 (110A, 110B, 110C, 110D, 및 110E) 로서 라벨링된다. 특히, 도 4(b) 에서, 블록 (102) 은 PU0 (106A) 및 PU1 (106B) 를 포함하고, 블록들 (110A, 110B, 110C, 110D, 및 110E) 은 PU0 (106A) 에 대한 공간적 이웃들을 나타낸다. 각각의 그룹에 대해, 시그널링된 레퍼런스 인덱스에 의해 표시된 동일한 레퍼런스 픽처를 지칭하는 이웃하는 블록에서 잠재적인 후보는 그룹의 최종 후보를 형성하도록 선택될 최고 우선순위를 갖는다. 모든 이웃하는 블록들은 동일한 레퍼런스 픽처를 가리키는 모션 벡터를 포함하지 않는다는 것이 가능하다. 따라서, 이러한 후보가 발견될 수 없으면, 제 1 이용 가능한 후보가 최종 후보를 형성하도록 스케일링될 것이고; 따라서 시간적 거리 차이들이 보상될 수 있다.

[0170] 도 5 는 HEVC 에서 시간적 모션 벡터 예측을 예시하는 개념도이다. 특히, 도 5(a) 는 PU0 (122A) 및 PU1

(122B) 을 포함하는 예시의 CU (120) 를 예시한다. PUO (122A) 는 PU (122A) 에 대한 센터 블록 (126) 및 PUO (122A) 로의 하부-우측 블록 (124) 을 포함한다. 도 5(a) 는 또한, 이하에서 논의된 바와 같이 모션 정보가 PUO (122A) 의 모션 정보로부터 예측될 수도 있는 외부 블록 (128) 을 나타낸다. 도 5(b) 는, 모션 정보가 예측될 현재 블록 (138) 을 포함하는 현재 픽처 (130) 를 예시한다. 특히, 도 5(b) 는 (현재 블록 (138) 에 병치된 블록 (140) 을 포함하는) 현재 픽처 (130) 로의 병치된 픽처 (134), 현재 레퍼런스 픽처 (132), 및 병치된 레퍼런스 픽처 (136) 를 예시한다. 병치된 블록 (140) 은, 블록 (138) 의 모션 정보에 대한 시간적 모션 벡터 예측자 (TMVP)(142) 로서 사용되는 모션 벡터 (144) 를 사용하여 예측된다.

[0171] TMVP 가 인에이블되고 TMVP 후보가 이용 가능하면 임의의 공간 모션 벡터 후보들 후에, 비디오 코더는 TMVP 후보 (예를 들어, TMVP 후보 (142)) 를 MV 후보 리스트 안에 추가할 수도 있다. TMVP 후보에 대한 모션 벡터 도출의 프로세스는 머지 및 AMVP 모드들 양자 모두에 대해 동일하다. 그러나, 머지 모드에서 TMVP 후보에 대한 타겟 레퍼런스 인덱스는 HEVC 에 따라 0 으로 설정된다.

[0172] TMVP 후보 도출에 대한 프라이머리 블록 로케이션은, 공간적 이웃하는 후보들을 생성하는데 사용된 상부 및 좌측 블록들에 대한 바이어스를 보상하기 위해, PUO (122A) 에 대해 블록 (124) 과 같이 도 5(a) 에 도시된 바와 같이 병치된 PU 밖의 하부 우측 블록이다. 그러나, 블록 (124) 이 현재 CTB 로우 밖에 위치되거나 또는 모션 정보가 블록 (124) 에 대해 이용 가능하지 않으면, 블록은 도 5(a) 에 도시된 바와 같이 PU 의 센터 블록 (126) 으로 대체된다.

[0173] TMVP 후보 (142) 에 대한 모션 벡터는, 슬라이스 레벨 정보에 표시된 바와 같이, 병치된 픽처 (134) 의 공동-위치된 블록 (140) 으로부터 도출된다.

[0174] AVC 에서의 시간적 직접 모드와 유사하게, TMVP 후보의 모션 벡터는 모션 벡터 스케일링을 받을 수도 있고, 이것은 현재 픽처 (130) 와 현재 레퍼런스 픽처 (132), 및 병치된 픽처 (134) 와 병치된 레퍼런스 픽처 (136) 간의 픽처 순서 카운트 (POC) 거리 차이들을 보상하도록 수행된다. 즉, 모션 벡터 (144) 는, 이들 POC 차이들에 기초하여 TMVP 후보 (142) 를 생성하도록 스케일링될 수도 있다.

[0175] HEVC 의 머지 및 AMVP 모드들의 여러 양태들은 이하에서 논의된다.

[0176] 모션 벡터 스케일링: 모션 벡터의 값은 제시 시간에서 픽처들 간의 거리에 비례하는 것으로 가정된다. 모션 벡터는 2 개의 픽처들: 레퍼런스 픽처 및 모션 벡터를 포함하는 (즉, 픽처를 포함하는) 픽처를 연관시킨다. 모션 벡터가 다른 모션 벡터를 예측하기 위해 비디오 인코더 (20) 또는 비디오 디코더 (30) 에 의해 사용되는 경우, 포함하는 픽처와 레퍼런스 픽처 간의 차이는 픽처 순서 카운트 (POC) 값들에 기초하여 계산된다.

[0177] 예측될 모션 벡터에 대해, 그 연관된 포함하는 픽처 및 레퍼런스 픽처는 상이하다. 즉, 2 개의 별개의 모션 벡터들: 예측될 제 1 모션 벡터, 및 제 1 모션 벡터를 예측하는데 사용된 제 2 모션 벡터에 대해 2 개의 POC 차이 값들이 존재한다. 또한, 제 1 POC 차이는 제 1 모션 벡터의 현재 픽처와 레퍼런스 픽처 간의 차이이고, 제 2 POC 차이는 제 2 모션 벡터를 포함하는 픽처와 제 2 모션 벡터가 지칭하는 레퍼런스 픽처 간의 차이이다. 제 2 모션 벡터는 이들 2 개의 POC 거리들에 기초하여 스케일링될 수도 있다. 공간적 이웃하는 후보에 대해, 2 개의 모션 벡터들에 대한 포함하는 픽처들은 동일한 한편, 레퍼런스 픽처들은 상이하다. HEVC 에서, 모션 벡터 스케일링은 공간적 및 시간적 이웃하는 후보들에 대한 TMVP 및 AMVP 양자 모두에 적용한다.

[0178] 인공 모션 벡터 후보 생성: 모션 벡터 후보 리스트가 완성되지 않으면, 인공 모션 벡터 후보들이 생성되어 리스트가 미리결정된 수의 후보들을 포함할 때까지 리스트의 끝에 삽입될 수도 있다.

[0179] 머지 모드에는, 2 개 유형들의 인공 MV 후보들이 존재한다: 결합된 후보들은 단지 B-슬라이스들에 대해서만 도출되고 0 후보들은 제 1 유형이 충분한 인공 후보들을 제공하지 않으면 AMVP 에 대해서만 사용된다.

[0180] 이미 후보 리스트에 있고 필요한 모션 정보를 갖는 후보들의 각각의 쌍에 대해, 양-방향 결합된 모션 벡터 후보들은 리스트 0 에서의 픽처를 지칭하는 제 1 후보의 모션 벡터 및 리스트 1 에서의 픽처를 지칭하는 제 2 후보의 모션 벡터의 결합에 의해 도출된다.

[0181] 다음은 후보 삽입에 대한 예시의 프루닝 프로세스의 설명이다. 상이한 블록들로부터의 후보들은 동일한 것으로 발생할 수도 있고, 이것은 머지/AMVP 후보 리스트의 효율성을 감소시킨다. 프루닝 프로세스는 이 문제를 해결하기 위해 적용될 수도 있다. 프루닝 프로세스에 따르면, 비디오 코더는 소정 범위로, 동일한 후보를 삽입하는 것을 회피하기 위해 현재 후보 리스트에서의 다른 것들에 하나의 후보를 비교한다. 복잡성을 감소시키기 위해, 리스트에 이미 있는 모든 다른 기존의 후보들과 각각의 잠재적 후보를 비교하는 대신에 단지

제한된 수의 프루닝 프로세스들이 적용된다.

- [0182] 도 6 은 3D-HEVC 에 대한 예시의 예측 구조를 예시한다. 3D-HEVC 는 JCT-3V 에 의한 개발 하에서 HEVC 의 3D 비디오 확장이다. 본 개시물의 기법들에 관련된 중요한 기법들은 이 서브-섹션에서 설명된다.
- [0183] 도 6 은 3-뷰 케이스에 대한 멀티뷰 예측 구조를 나타낸다. V3 는 베이스 뷰이고, 비-베이스 뷰 (V1 또는 V5) 에서의 픽처는 동일한 시간 인스턴스의 종속적인 (베이스) 뷰에서의 픽처들로부터 예측될 수 있다.
- [0184] (복원된 샘플들로부터의) 인터-뷰 샘플 예측이 MV-HEVC에서 지원되는 것은 언급할 가치가 있으며, 이것의 통상적인 예측 구조가 도 8 에 도시된다.
- [0185] MV-HEVC 및 3D-HEVC 는, 베이스 (텍스처) 뷰가 HEVC (버전 1) 디코더에 의해 디코딩 가능한 방식으로 HEVC 와 호환 가능하다. MV-HEVC 및 3D-HEVC 에 대한 테스트 모델은 Zhang 등의, 『"Test Model 6 of 3D-HEVC and MV-HEVC," JCT-3V document ISO/IEC JTC1/SC29/WG11N13940』 에서 설명되고, 2015 년 1 월 26 일자로 [mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding/test-model-6-3d-hevc-and-mv-hevc](http://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding/test-model-6-3d-hevc-and-mv-hevc) 에서 이용 가능하다.
- [0186] MV-HEVC 에서, 비-베이스 뷰에서의 현재 픽처는, 픽처의 레퍼런스 픽처 리스트들에 이들 픽처들 모두를 놓음으로써, 동일한 시간 인스턴스의 레퍼런스 뷰에서의 픽처들 및 동일한 뷰에서의 픽처들 양자 모두에 의해 예측될 수도 있다. 따라서, 현재 픽처의 레퍼런스 픽처 리스트는 시간적 레퍼런스 픽처들 및 인터-뷰 레퍼런스 픽처들 양자 모두를 포함한다.
- [0187] 시간적 레퍼런스 픽처에 대응하는 레퍼런스 인덱스와 연관된 모션 벡터는 시간적 모션 벡터로 표기된다.
- [0188] 인터-뷰 레퍼런스 픽처에 대응하는 레퍼런스 인덱스와 연관된 모션 벡터는 디스패리티 모션 벡터로 표기된다.
- [0189] 3D-HEVC 는 MV-HEVC 에서의 모든 픽처들을 지원한다; 따라서, 위에서 언급된 바와 같은 인터-뷰 샘플 예측이 인에이블된다.
- [0190] 또한, 더 진보된 텍스처 전용 코딩 툴들 및 심도 관련/종속적인 코딩 툴들이 지원된다.
- [0191] 텍스처-전용 코딩 툴들은 종종, 동일한 오브젝트에 속할 수도 있는 (뷰들 간의) 대응하는 블록들의 식별을 요구한다. 따라서, 디스패리티 벡터 도출은 3D-HEVC 에서의 기본 기술이다.
- [0192] 도 7 은 3D-HEVC 에서 서브-PU 기반의 인터-뷰 모션 예측을 예시하는 개념도이다. 도 7 은 현재 뷰 (V1) 의 현재 픽처 (160) 및 레퍼런스 뷰 (V0) 에서의 병치된 픽처 (162) 를 나타낸다. 현재 픽처 (160) 는 4 개의 서브-PU들 (166A-166D)(서브-PU들 (166)) 을 포함하는 현재 PU (164) 를 포함한다. 각각의 디스패리티 벡터들 (174A-174D)(디스패리티 벡터들 (174)) 은 병치된 픽처 (162) 에서 서브-PU들 (166) 에 대응하는 서브-PU들 (168A-168D) 을 식별한다. 3D-HEVC 는 인터-뷰 머지 후보, 즉 레퍼런스 뷰에서 레퍼런스 블록으로부터 도출된 후보에 대한 서브-PU 레벨 인터-뷰 모션 예측 방법을 설명한다.
- [0193] 이러한 모드가 인에이블되는 경우, 현재 PU (164) 는 레퍼런스 뷰에서 (디스패리티 벡터에 의해 식별된 현재 PU 와 동일한 사이즈를 갖는) 레퍼런스 영역에 대응할 수도 있고, 레퍼런스 영역은 PU 에 대한 통상적인 모션 정보의 하나의 세트의 생성을 위해 필요한 것보다 더 많은 모션 정보를 가질 수도 있다. 따라서, 서브-PU 레벨 인터-뷰 모션 예측 (SPIVMP) 방법이 도 7 에 도시된 바와 같이, 사용될 수도 있다.
- [0194] 이 모드는 또한, 특별한 머지 후보로서 시그널링될 수도 있다. 서브-PU들 각각은 모션 정보의 풀 세트를 포함한다. 따라서, PU 는 모션 정보의 다수의 세트들을 포함할 수도 있다.
- [0195] 3D-HEVC 에서 서브-PU 기반의 모션 파라미터 계승 (MPI): 유사하게, 3D-HEVC 에서, MPI 후보는 또한, 서브-PU 레벨 인터-뷰 모션 예측과 유사한 방식으로 확장될 수 있다. 예를 들어, 현재 심도 PU 가 다수의 PU들을 포함하는 공동-위치된 영역을 가지면, 현재 심도 PU 는 서브-PU들로 분리될 수도 있고, 각각 PU 는 모션 정보의 상이한 세트를 가질 수도 있다. 이 방법은 서브-PU MPI 로 지칭된다. 즉, 대응하는 서브 PU들 (168A-168D) 의 모션 벡터들 (172A-172D) 은 도 7 에 도시된 바와 같은, 모션 벡터들 (170A-170D) 과 같이 서브-PU들 (166A-166D) 에 의해 계승될 수도 있다.
- [0196] 2D 비디오 코딩에 대한 서브-PU 관련 정보: 2015 년 3 월 26 일자로 미국 공개 제 2015/0086929 호로서 공개된, 2014 년 9 월 25 일자로 출원된 미국 출원 제 14/497,128 호에서, 서브-PU 기반의 어드밴스드 TMVP 설계가 설명된다. 단일-계층 코딩에서, 2-스테이지 어드밴스드 시간적 모션 벡터 예측 설계가 제안된다.

- [0197] 제 1 스테이지는 레퍼런스 픽처에서 현재 예측 유닛 (PU) 의 대응하는 블록을 식별하는 벡터를 도출하는 것이고, 제 2 스테이지는 대응하는 블록으로부터 다수의 세트 모션 정보를 추출하고 이들을 PU 의 서브-PU들에 할당하는 것이다. PU 의 각각의 서브-PU 는 따라서, 별개로 모션 보상된다. ATMVP 의 개념은 다음과 같이 요약된다:
- [0198] 1. 제 1 스테이지에서 벡터는 현재 PU 의 공간적 및 시간적 이웃 블록들로부터 도출될 수 있다.
- [0199] 2. 이 프로세스는 모든 다른 머지 후보들 중에서 머지 후보를 활성화하는 것으로 달성될 수도 있다.
- [0200] 단일-계층 코딩 및 서브-PU 시간적 모션 벡터 예측에 적용 가능한 것으로, PU 또는 CU 는 예측자들의 상단으로 전달될 모션 리파인먼트 데이터를 가질 수도 있다.
- [0201] 제 14/497,128 출원의 여러 설계 양태들은 다음과 같이 하이라이트된다:
- [0202] 1. 벡터 도출의 제 1 스테이지는 또한, 단지 0 벡터에 의해 단순화될 수 있다.
- [0203] 2. 벡터 도출의 제 1 스테이지는 모션 벡터 및 그 연관된 픽처를 합동으로 식별하는 것을 포함할 수도 있다. 연관된 픽처를 선택하고, 또한 모션 벡터를 제 1 스테이지 벡터인 것으로 결정하는 다양한 방식들이 제안되어 있다.
- [0204] 3. 위의 프로세스 동안 모션 정보가 이용 가능하지 않으면, "제 1 스테이지 벡터" 가 치환을 위해 사용된다.
- [0205] 4. 시간적 이웃으로부터 식별된 모션 벡터는 TMVP 에서의 모션 벡터 스케일링과 유사한 방식으로, 현재 서브-PU 에 대해 사용되도록 스케일링되어야 한다. 그러나, 이러한 모션 벡터가 스케일링 될 수도 있는 레퍼런스 픽처는 다음의 방식들 중 하나로 설계될 수 있다:
- [0206] a. 픽처는 현재 픽처의 고정된 레퍼런스 인덱스에 의해 식별된다.
- [0207] b. 픽처는 또한, 현재 픽처의 레퍼런스 픽처 리스트에서 이용 가능하다면 대응하는 시간적 이웃의 레퍼런스 픽처인 것으로 식별된다.
- [0208] c. 픽처는 제 1 스테이지에서 식별된 병치된 픽처인 것으로 설정되고, 여기서부터 모션 벡터들이 추출된다.
- [0209] 도 8 은 레퍼런스 픽처로부터 서브-PU 모션 예측을 예시하는 개념도이다. 이 예에서, 현재 픽처 (180) 는 현재 PU (184)(예를 들어, PU) 를 포함한다. 이 예에서, 모션 벡터 (192) 는 PU (184) 에 대해 레퍼런스 픽처 (182) 의 PU (186) 를 식별한다. PU (186) 는 서브-PU들 (188A-188D) 로 파티셔닝되고, 이들 각각은 각각의 모션 벡터들 (190A-190D) 을 갖는다. 따라서, 현재 PU (184) 가 별개의 서브-PU들로 실제로 파티셔닝되지 않지만, 이 예에서 현재 PU (184) 는 서브-PU들 (188A-188D) 로부터의 모션 정보를 사용하여 예측될 수도 있다. 특히, 비디오 코더는 각각의 모션 벡터들 (190A-190D) 을 사용하여 현재 PU (184) 의 서브-PU들을 코딩할 수도 있다. 그러나, 비디오 코더는, 현재 PU (184) 가 서브-PU 들로 스플리팅된다는 것을 나타내는 신택스 엘리먼트들을 코딩할 필요가 없다. 이 방식에서, 현재 PU (184) 는, 현재 PU (184) 를 다수의 서브-PU들로 스플리팅하는데 사용된 신택스 엘리먼트들의 시그널링 오버헤드 없이, 각각의 서브-PU들 (188A-188D) 로부터 계승된, 다수의 모션 벡터들 (190A-190D) 을 사용하여 효율적으로 예측될 수도 있다.
- [0210] 도 9 는 (TMVP 와 유사한) ATMVP 에서의 관련 픽처들을 예시하는 개념도이다. 특히, 도 9 는 현재 픽처 (204), 모션 소스 픽처 (206), 및 레퍼런스 픽처들 (200, 202) 을 예시한다. 보다 구체적으로, 현재 픽처 (204) 는 현재 블록 (208) 을 포함한다. 시간적 모션 벡터 (212) 는 현재 블록 (208) 에 대해 모션 소스 픽처 (206) 의 대응하는 블록 (210) 을 식별한다. 대응하는 블록 (210) 은 이어서, 레퍼런스 픽처 (202) 를 지칭하고 현재 블록 (208) 의 적어도 일부, 예를 들어 현재 블록 (208) 의 서브-PU들에 대한 어드밴스드 시간적 모션 벡터 예측자로서 작용하는 모션 벡터 (214) 를 포함한다. 즉, 모션 벡터 (214) 는 현재 블록 (208) 에 대한 후보 모션 벡터 예측자로서 추가될 수도 있다. 선택되면, 현재 블록 (208) 의 적어도 일부는 대응하는 모션 벡터, 즉 레퍼런스 픽처 (200) 를 지칭하는 모션 벡터 (216) 를 사용하여 예측될 수도 있다.
- [0211] 도 10 은 본 개시물의 기법들에 따른 예시의 방법을 예시하는 플로우차트이다. 도 10 의 방법은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있다. 일반화를 위해, 도 10 의 방법은 비디오 인코더 (20) 또는 비디오 디코더 (30) 중 어느 하나에 대응할 수도 있는, "비디오 코더" 에 의해 수행되는 것으로서 설명된다.



- [0212] 먼저, 비디오 코더는 PU의 현재 서브-PU에 대한 공간적 또는 시간적 이웃 블록들로부터 이용 가능한 모션 필드를 획득한다 (230). 비디오 코더는 그 후, 획득된 이웃하는 모션 필드로부터 모션 정보를 도출한다 (232). 비디오 코더는 그 후, PU의 모든 서브-PU들에 대한 모션 정보가 도출되었는지 여부를 결정한다 (234). 도출되지 않았으면 (234의 "아니오" 분기), 비디오 코더는 나머지 서브-PU에 대한 모션 정보를 도출한다 (230). 한편, 모든 서브-PU들에 대한 모션 정보가 도출되었으면 (234의 "예" 분기), 비디오 코더는, 위에서 설명된 바와 같이 공간-시간적 서브-PU 모션 예측자의 이용 가능성을 결정한다 (236). 비디오 코더는, 공간-시간적 서브-PU 모션 예측자가 이용 가능한 경우, 공간-시간적 서브-PU 모션 예측자를 머지 리스트에 삽입한다 (238).
- [0213] 도 10의 방법에 도시되지 않았으나, 비디오 코더는 그 후, 머지 후보 리스트를 사용하여 PU (예를 들어, PU의 서브-PU들 각각)를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)에 의해 수행되는 경우, 비디오 인코더 (20)는 서브-PU들을 예측자들로서 사용하여 PU에 대해 (예를 들어, 각각의 서브-PU에 대해) 잔차 블록(들)을 계산하고, 잔차 블록(들)을 변환 및 양자화하며, 결과의 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다. 비디오 디코더 (30)는, 유사하게 수신된 데이터를 엔트로피 디코딩하여 양자화된 변환 계수들을 재생성하고, 이들 계수들을 역 양자화 및 역 변환하여 잔차 블록(들)을 재생성하며, 그 후 대응하는 서브-PU들과 잔차 블록(들)을 결합하여 PU에 대응하는 블록을 디코딩할 수도 있다.
- [0214] 도 11a 및 도 11b는 도출된 모션 정보를 사용하여 예측되는 서브-블록들을 포함하는 블록들의 예를 예시하는 개념도들이다. 특히, 도 11a는, 블록 (250)이 PU인 경우 서브-PU들을 나타낼 수도 있는, 서브-블록들 (254A-254P)(서브-블록들 (254))을 포함하는 블록 (250)(예를 들어, PU)을 예시한다. 블록 (250)에 이웃하는 서브-블록들 (256A-256I)(이웃하는 서브-블록들 (256))은 또한, 도 11a에 도시되고 밝은 회색으로 음영처리된다.
- [0215] 일반적으로, 비디오 코더, 예컨대 비디오 인코더 (20) 또는 비디오 디코더 (30)는 2 이상의 이웃하는 블록들로부터의 모션 정보를 사용하여 블록 (250)의 서브-블록들 (254)에 대한 모션 정보를 도출할 수도 있다. 이웃하는 블록들은 공간적으로 이웃하는 및/또는 시간적으로 이웃하는 블록들을 포함할 수도 있다. 예를 들어, 비디오 코더는 서브-블록 (254O)의 포지션에 대응하는 공간적으로 이웃하는 서브-블록들 (254F 및 254I), 및 시간적으로 이웃하는 블록으로부터 서브-블록 (254J)에 대한 모션 정보를 도출할 수도 있다. 시간적으로 이웃하는 블록은 서브-블록 (254O)과 공동-위치되는 이전에 코딩된 픽처로부터의 것일 수도 있다. 서브-블록 (254J)에 대한 모션 정보의 모션 벡터를 도출하기 위해, 비디오 코더는 서브-블록 (254F), 서브-블록 (254I), 및 서브-블록 (254O)와 공동-위치되는 시간적으로 이웃하는 블록에 대한 모션 벡터들을 평균할 수도 있다. 대안으로, 비디오 코더는 위에서 논의된 바와 같이 식들 (1)-(4) 중 하나를 사용하여 모션 벡터를 도출할 수도 있다.
- [0216] 일부 예들에서, 비디오 코더는 블록 (250)밖의 서브-블록들, 예를 들어 이웃하는 서브-블록들 (256) 및/또는 시간적으로 이웃하는 서브-블록들로부터 모션 정보를 항상 도출하도록 구성될 수도 있다. 이러한 구성은 서브-블록들 (254)이 병렬로 코딩되는 것을 허용할 수도 있다. 예를 들어, 비디오 코더는 서브-블록들 (256B 및 256F), 및 서브-블록 (254F)과 공동-위치되는 시간적으로 이웃하는 서브-블록의 모션 정보로부터 서브-블록 (254A)에 대한 모션 정보를 도출할 수도 있다. 비디오 코더는 또한, 서브-블록들 (256C, 256B, 256F), 및 서브-블록들 (254F 및 254G)과 공동-위치되는 시간적으로 이웃하는 서브-블록들의 모션 정보를 사용하여 서브-블록 (254A)과 병렬로 서브-블록 (254B)에 대한 모션 정보를 도출할 수도 있다.
- [0217] 도 11b는, 다시 서브-PU들을 나타낼 수도 있는 서브-블록들 (264A-264D)(서브-블록들 (264))을 포함하는 블록 (260)(예를 들어, PU)을 예시한다. 도 11b는 또한, 이웃하는 서브-블록들 (266A-266I)(이웃하는 서브-블록들 (266))을 예시한다. 일반적으로, 도 11b의 예는, 블록, 예컨대 블록 (260)의 서브-블록들이 다양한 사이즈일 수도 있고, 모션 정보를 도출하는데 사용된 이웃하는 블록들보다 더 클 수도 있다는 것을 나타낸다. 이 예에서, 서브-블록들 (264)은 이웃하는 서브-블록들 (266)보다 더 크다. 그럼에도 불구하고, 비디오 코더들 (예컨대, 비디오 인코더 (20) 및 비디오 디코더 (30))은 서브-블록들 (254)에 대하여 위에서 논의된 바와 같은 서브-블록들 (264)에 유사한 기법들을 적용하도록 구성될 수도 있다.
- [0218] 도 12는 본 개시물의 기법들에 따라 비디오 데이터를 인코딩하는 예시의 방법을 예시하는 플로우차트이다. 도 12의 방법은 설명 및 예의 목적들을 위해, 비디오 인코더 (20)(도 1 및 도 2) 및 그 컴포넌트들에 대하여 설명된다. 그러나, 다른 비디오 인코딩 디바이스들은 이들 또는 유사한 기법들을 수행하도록 구성될 수도 있다는 것이 이해되어야 한다. 또한, 소정의 단계들은 생략되고, 상이한 순서로 수행되고, 및/또는 병렬로

수행될 수도 있다.

- [0219] 먼저, 비디오 인코더 (20) 는 코딩 유닛 (CU) 을 하나 이상의 예측 유닛 (PU)들로 파티셔닝한다 (270). 비디오 인코더 (20) 는 그 후, PU들 각각에 대한 다양한 예측 모드들 (예를 들어, 공간 또는 인트라-예측, 시간 또는 인터-예측, 및 서브-블록 모션 도출 예측) 을 테스트할 수도 있다 (272). 특히, 모드 선택 유닛 (40) 은 다양한 예측 모드들을 테스트하고, PU 에 대한 최선의 레이트-왜곡 특징을 산출하는 PU 에 대한 모드들 중 하나를 선택할 수도 있다. 비디오 인코더 (20) 는 CU 의 PU 에 대한 서브-PU 모션 도출 모드를 선택한다 (274) 는 것을 예시의 목적들을 위해 가정한다.
- [0220] 서브-PU 모션 도출 모드에 따르면, 비디오 인코더 (20) 는 PU 를 서브-PU들로 파티셔닝한다 (276). 일반적으로, 서브-PU들은 별개의 정보, 예컨대 모션 정보가 서브-PU들에 대해 코딩되지 않는다는 점에서 PU 와 구별된다. 대신에, 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 는 이웃하는 서브-PU들로부터 서브-PU들에 대한 모션 정보를 도출한다 (278). 이웃하는 서브-PU들은 공간적 및/또는 시간적 이웃하는 서브-PU들을 포함할 수도 있다. 예를 들어, 이웃하는 서브-PU들은 도 11a 에 대하여 논의된 바와 같이 선택될 수도 있다. 즉, 이 예에서, 각각의 서브-PU 에 대해, 비디오 인코더 (20) 는 상부-이웃하는 공간적 서브-PU, 좌측-이웃하는 공간적 서브-PU, 및 하부-우측의 시간적 이웃하는 서브-PU 를 포함하는 이웃하는 서브-PU들로부터 모션 정보를 도출한다. 비디오 인코더 (20) 는 위에서 논의된 식들 (1)-(4) 에 따라 또는 이웃하는 서브-PU들의 모션 정보의 평균으로서 도출된 모션 정보를 계산할 수도 있다.
- [0221] 비디오 인코더 (20) 는 그 후, 도출된 모션 정보를 사용하여 서브-PU들을 예측할 수도 있다 (280). 즉, 비디오 인코더 (20) 의 모션 보상 유닛 (44) 은 각각의 서브-PU들에 대한 도출된 모션 정보를 사용하여 PU 의 서브-PU들 각각에 대한 예측된 정보를 취출할 수도 있다. 비디오 인코더 (20) 는 PU 의 그 각각의 포지션들에서 예측된 서브-PU들 각각의 어셈블리로서 PU 에 대한 예측된 블록을 형성할 수도 있다.
- [0222] 비디오 인코더 (20) 는 그 후, PU 에 대한 잔차 블록을 계산할 수도 있다 (282). 예를 들어, 합산기 (50) 는 잔차 블록을 형성하는, PU 의 원래 버전과 예측 블록 간의 픽셀 별 차이들을 계산할 수도 있다. 그 후, 비디오 인코더 (20) 의 변환 프로세싱 유닛 (52) 및 양자화 유닛 (54) 은 잔차 블록을 각각 변환 및 양자화하여, 양자화된 변환 계수들을 생성할 수도 있다 (284). 엔트로피 인코딩 유닛 (56) 은 그 후, 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다 (286).
- [0223] 또한, 엔트로피 인코딩 유닛 (56) 은, PU 가 서브-PU 모션 도출 모드를 사용하여 예측된다는 것을 나타내는 PU 에 대한 후보 인덱스를 엔트로피 인코딩할 수도 있다 (286). 특히, 엔트로피 인코딩 유닛 (56) 은 복수의 모션 예측 후보들, 뿐만 아니라 서브-PU 모션 도출 모드를 나타내는 후보를 포함하는 후보 리스트를 구성할 수도 있다. 따라서, 비디오 인코더 (20) 가 서브-PU 모션 정보 도출 모드를 선택하는 경우, 엔트로피 인코딩 유닛 (56) 은 PU 에 대한 후보 리스트에서 서브-PU 모션 도출 모드를 나타내는 후보의 포지션을 식별하는 인덱스를 나타내는 값을 엔트로피 인코딩한다.
- [0224] 전술된 방식으로 PU 를 인코딩한 후에, 비디오 인코더 (20) 는 또한, 실질적으로 유사한, 비록 상호간의 방식으로 PU 를 디코딩한다. 도 12 에 도시되지 않았으나, 비디오 인코더 (20) 는 또한, 양자화된 변환 계수들을 역 변환 및 역 양자화하여 잔차 블록을 재생성하고, 후속의 예측 (예를 들어, 인트라- 및/또는 인터-예측) 동안 레퍼런스 블록으로서 사용하기 위해, PU 를 디코딩하도록 예측된 블록과 잔차 블록을 결합한다.
- [0225] 이 방식에서, 도 12 의 방법은, 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다는 것을 결정하는 단계, 그 결정에 응답하여: 현재 블록을 서브-블록들로 파티셔닝하는 단계, 서브-블록들 각각에 대해 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하는 단계, 및 각각의 도출된 모션 정보를 사용하여 서브-블록들을 인코딩 (및 디코딩) 하는 단계를 포함하는 방법의 예를 나타낸다.
- [0226] 도 13 은 본 개시물의 기법들에 따라 비디오 데이터를 디코딩하는 방법의 일 예이다. 도 13 의 방법은 설명 및 예의 목적들을 위해, 비디오 디코더 (30)(도 1 및 도 3) 및 그 컴포넌트들에 대하여 설명된다. 그러나, 다른 비디오 디코딩 디바이스들이 이들 또는 유사한 기법들을 수행하도록 구성될 수도 있다는 것이 이해되어야 한다. 또한, 소정의 단계들은 생략되고, 상이한 순서로 수행되고, 및/또는 병렬로 수행될 수도 있다. 먼저, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은, 서브-PU 모션 도출 모드가 예측 유닛에 대해 사용된다는 것을 나타내는 후보 리스트에서의 후보의 후보 인덱스를 엔트로피 디코딩한다 (290). 도시되지 않았으나, 초기에 비디오 디코더 (30) 는 후보 리스트를 구성하고 후보들을 후보 리스트에 추가한다는 것을 이해해

야 한다. 이 예에서, 설명의 목적들을 위해, 후보 인덱스는 서브-PU 모션 도출 모드를 나타내는 후보를 지칭한다. 그러나, 일반적으로, 후보 인덱스는 PU 에 대한 후보 리스트에서 후보들 중 어느 하나를 지칭한다는 것을 이해해야 한다.

[0227] 이 예에서, 후보 인덱스는 서브-PU 모션 도출 모드가 PU 에 대해 사용된다는 것을 나타내는 후보를 지칭하기 때문에, 비디오 디코더 (30) 는 PU 를 서브-PU들로 파티셔닝한다 (292). 비디오 디코더 (30) 의 모션 보상 유닛 (72) 은 그 후, 이웃하는 서브-PU들로부터 서브-PU들 각각에 대한 모션 정보를 도출한다 (294). 이웃하는 서브-PU들은 공간적 및/또는 시간적 이웃하는 서브-PU들을 포함할 수도 있다. 예를 들어, 이웃하는 서브-PU들은 도 11a 에 대하여 논의된 바와 같이 선택될 수도 있다. 즉, 이 예에서, 각각의 서브-PU 에 대해, 비디오 디코더 (30) 는 상부-이웃하는 공간적 서브-PU, 좌측-이웃하는 공간적 서브-PU, 및 하부-우측의 시간적 이웃하는 서브-PU 를 포함하는 이웃하는 서브-PU들로부터 모션 정보를 도출한다. 비디오 디코더 (30) 는 위에서 논의된 식들 (1)-(4) 에 따라 또는 이웃하는 서브-PU들의 모션 정보의 평균으로서 도출된 모션 정보를 계산할 수도 있다.

[0228] 비디오 디코더 (30) 는 그 후, 도출된 모션 정보를 사용하여 서브-PU들을 예측할 수도 있다 (296). 즉, 비디오 디코더 (30) 의 모션 보상 유닛 (72) 은 각각의 서브-PU들에 대한 도출된 모션 정보를 사용하여 PU 의 서브-PU들 각각에 대한 예측된 정보를 추출할 수도 있다. 비디오 디코더 (30) 는 PU 의 그 각각의 포지션들에서 예측된 서브-PU들 각각의 어셈블리로서 PU 에 대한 예측된 블록을 형성할 수도 있다.

[0229] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 또한, PU 의 양자화된 변환 계수들을 엔트로피 디코딩할 수도 있다 (298). 역 양자화 유닛 (76) 및 역 변환 유닛 (78) 은 양자화된 변환 계수들을 각각 역 양자화 및 역 변환하여, PU 에 대한 잔차 블록을 생성한다 (300). 비디오 디코더 (30) 는 그 후, 예측된 블록 및 잔차 블록을 사용하여 예측 유닛을 디코딩할 수도 있다 (302). 특히, 합산기 (80) 는 픽셀별 유닛으로 예측된 블록 및 잔차 블록을 결합하여, 예측 유닛을 디코딩할 수도 있다.

[0230] 이 방식에서, 도 13 의 방법은, 비디오 데이터의 현재 블록에 대한 모션 예측 후보가, 모션 정보가 현재 블록의 서브-블록들에 대해 도출된다는 것을 나타낸다는 것을 결정하는 단계, 그 결정에 응답하여: 현재 블록을 서브-블록들로 파티셔닝하는 단계, 서브-블록들 각각에 대해 적어도 2 개의 이웃하는 블록들에 대한 모션 정보를 사용하여 모션 정보를 도출하는 단계, 및 각각의 도출된 모션 정보를 사용하여 서브-블록들을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법의 예를 나타낸다.

[0231] 예에 따라, 본원에서 설명된 임의의 기법들의 소정의 액트들 또는 이벤트들은 상이한 시퀀스로 수행될 수도 있거나, 추가, 머지될 수도 있거나, 또는 함께 제거될 수도 있다 (예를 들어, 설명된 모든 액트들 또는 이벤트들이 기법의 실시예 반드시 필요한 것은 아니다) 는 것으로 인식되어야 한다. 또한, 소정 예들에서, 액트들 또는 이벤트들은, 순차적이기 보다는 예를 들어 멀티-스레디드 프로세싱, 인터럽트 프로세싱, 또는 다중 프로세서들을 통해 동시에 수행될 수도 있다.

[0232] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 이 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 송신될 수도 있고, 하드웨어-기반의 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예를 들어 통신 프로토콜에 따라, 한 장소에서 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수도 있다. 이 방식으로, 컴퓨터 판독가능 매체는 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에 설명된 기법들의 구현을 위한 명령들, 코드, 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0233] 비제한적인 예로서, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장 디바이스, 자기 디스크 저장 디바이스 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체라고 적절히 지칭된다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에

포함된다. 그러나, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속들, 반송파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않고, 대신에 비일시적인, 유형의 저장 매체들이다. 본원에서 사용된 디스크 (disk) 및 디스크 (disc) 는, 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 통상 자기적으로 데이터를 재생하는 반면, 디스크 (disc) 들은 레이저들을 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들이 또한, 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

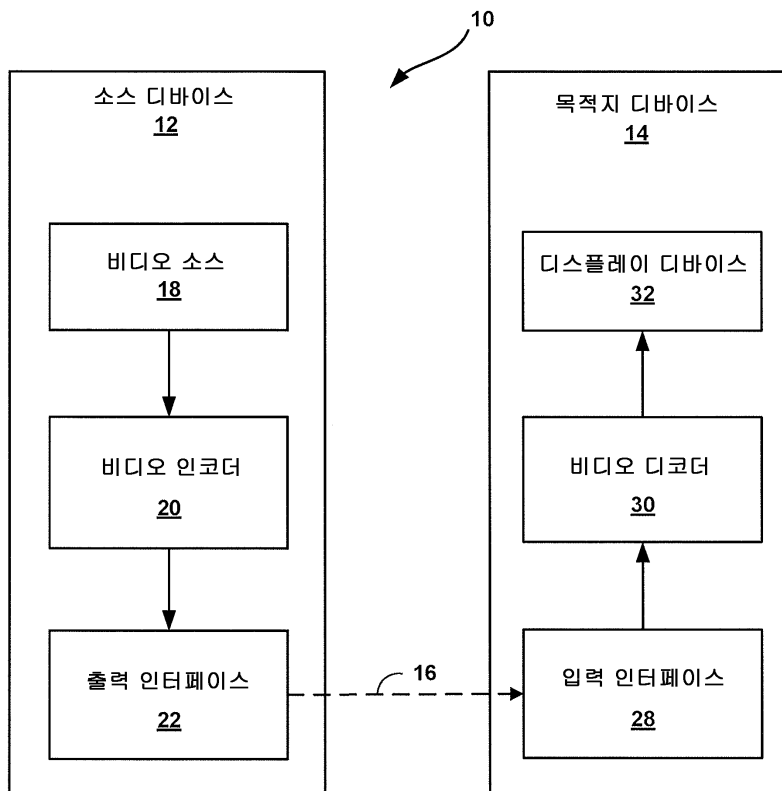
[0234] 명령들은, 하나 이상의 디지털 신호 프로세서 (DSP) 들, 범용 마이크로프로세서들, 주문형 집적 회로 (ASIC) 들, 필드 프로그래머블 로직 어레이 (FPGA) 들, 또는 다른 등가의 집적 또는 이산 로직 회로부와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서" 는 상기의 구조 또는 본원에 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 또한, 일부 양태들에서, 본원에 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있고, 또는 결합형 코덱에 통합될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0235] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들어, 칩 세트) 를 포함하는 광범위한 디바이스들 또는 장치들로 구현될 수도 있다. 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태를 강조하기 위해 다양한 컴포넌트들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 반드시 상이한 하드웨어 유닛들에 의한 실현을 요구할 필요는 없다. 차라리, 전술된 바와 같이 다양한 유닛들은 적합한 소프트웨어 및/또는 펌웨어와 관련되어, 전술된 하나 이상의 프로세서들을 포함하는, 상호 동작적인 하드웨어 유닛들의 집합에 의해 제공되거나 또는 코덱 하드웨어 유닛에 결합될 수도 있다.

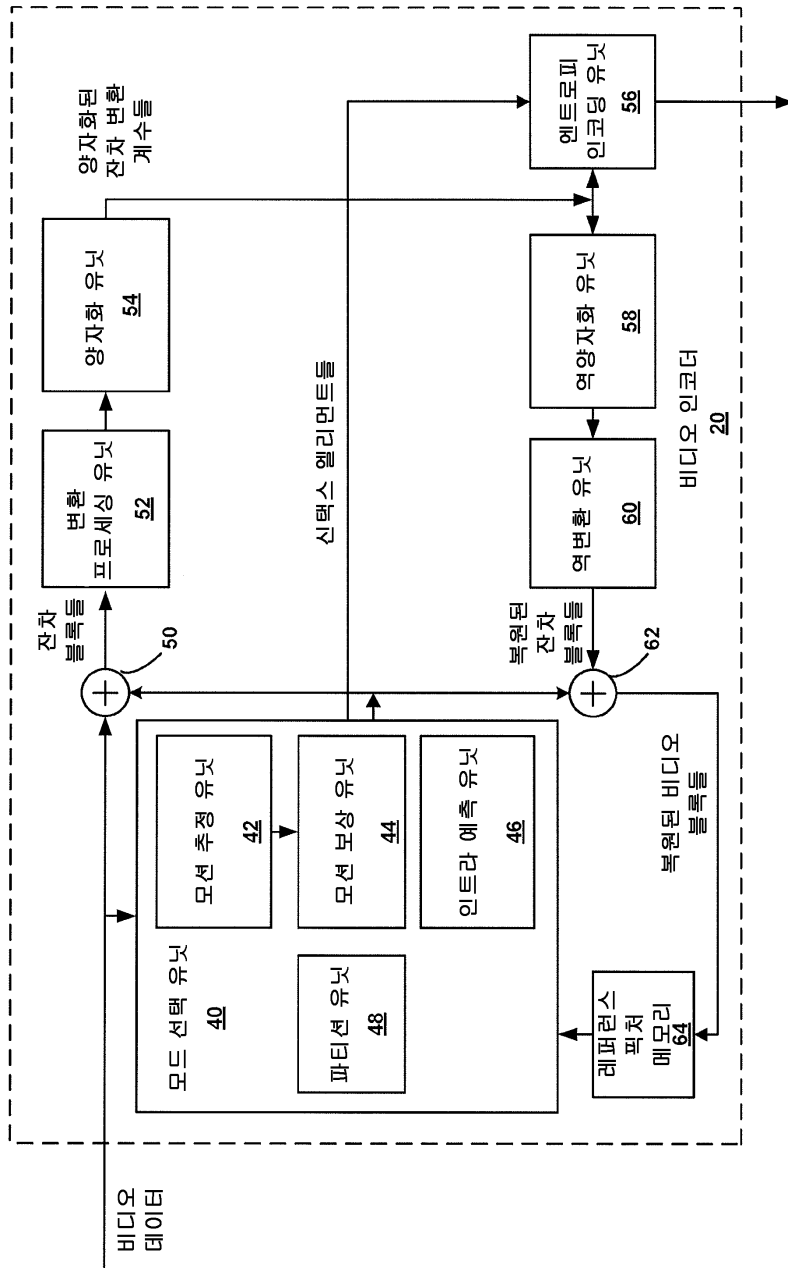
[0236] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 특허청구범위 내에 있다.

**도면**

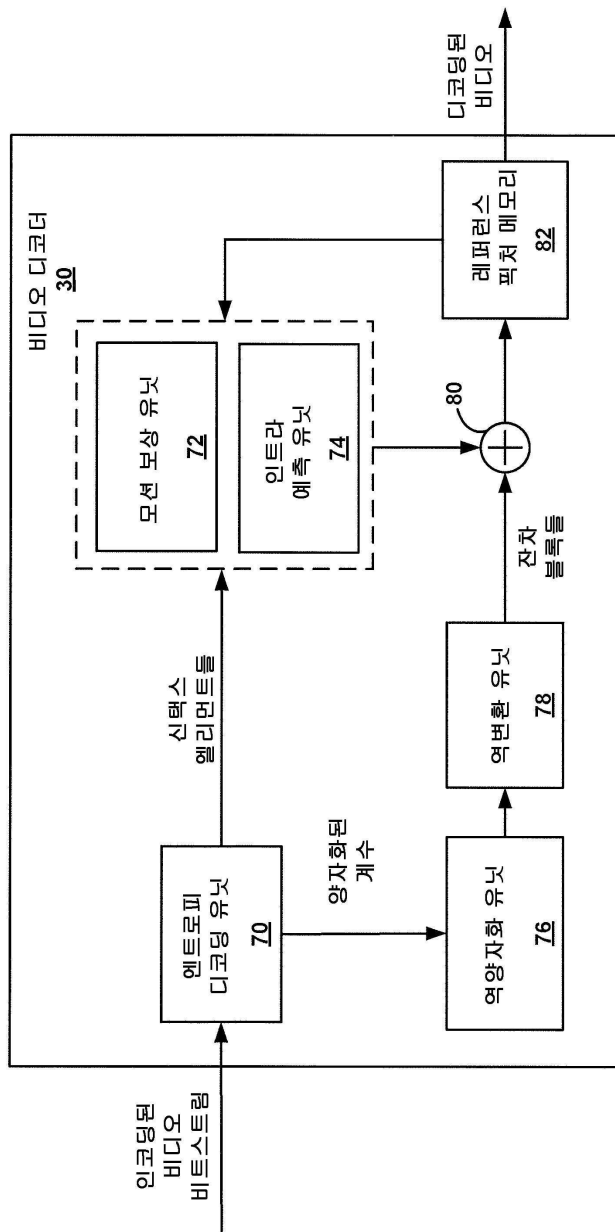
**도면1**



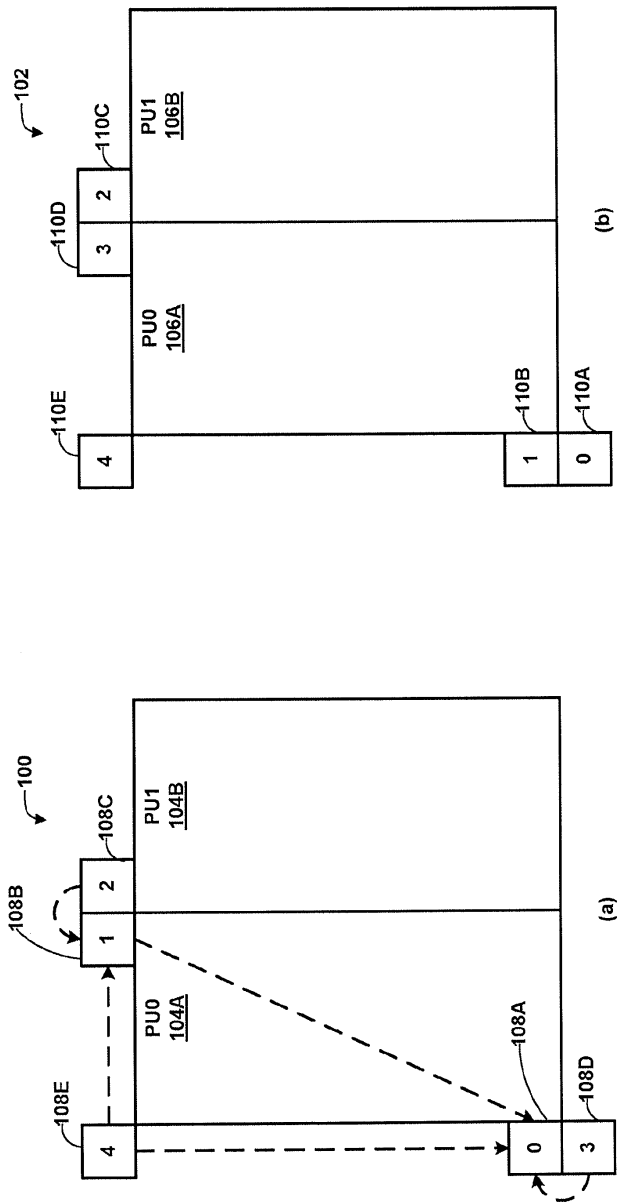
도면2



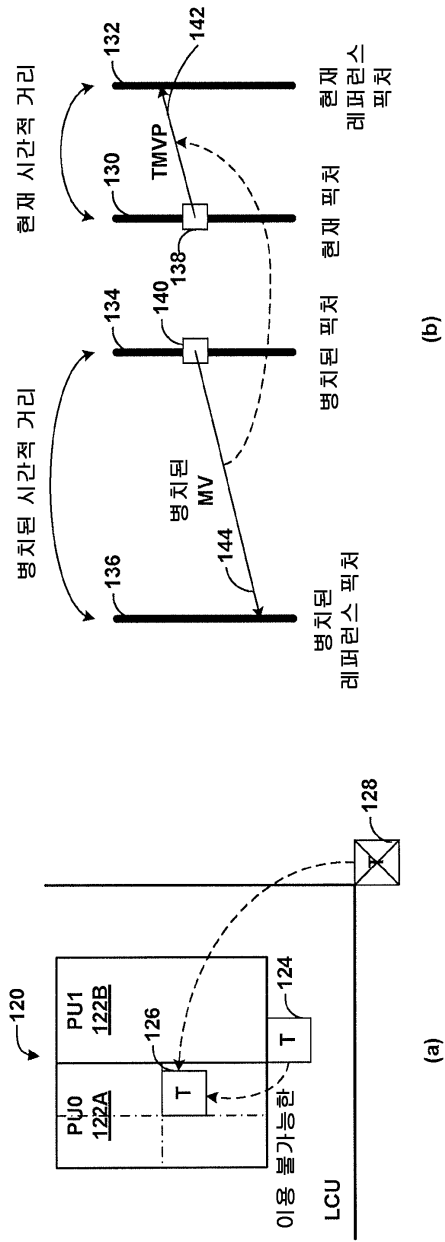
도면3



도면4

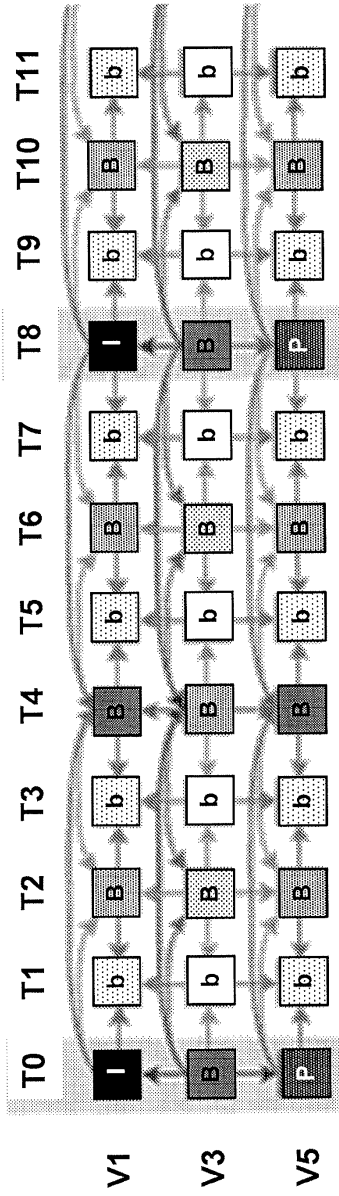


도면5

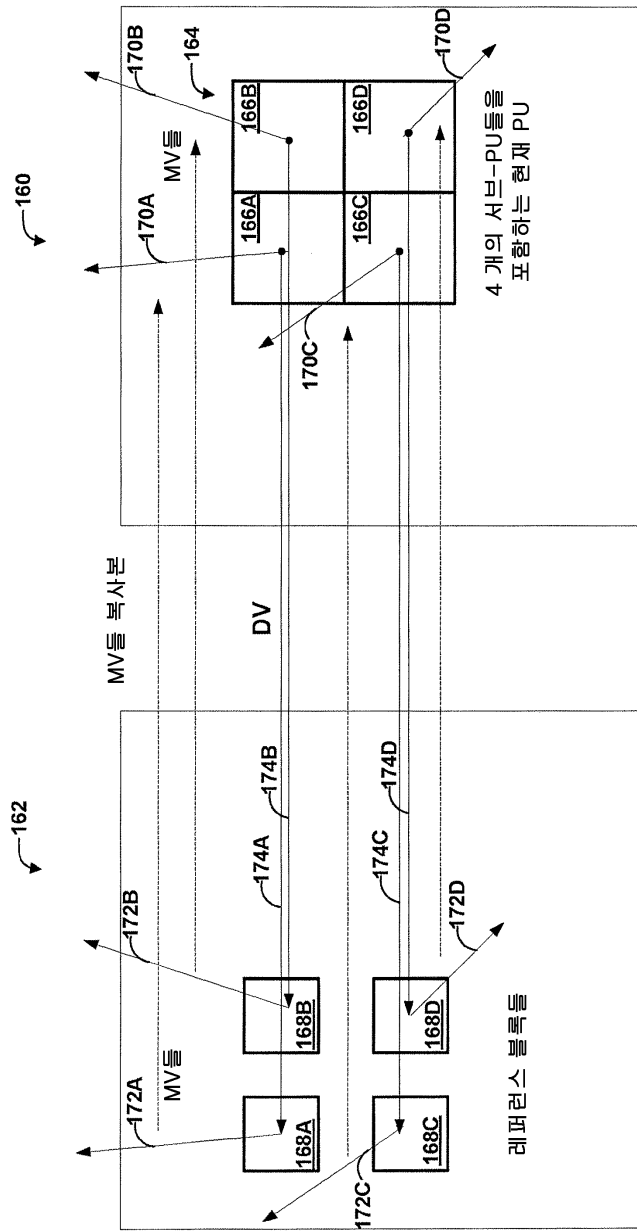




도면6



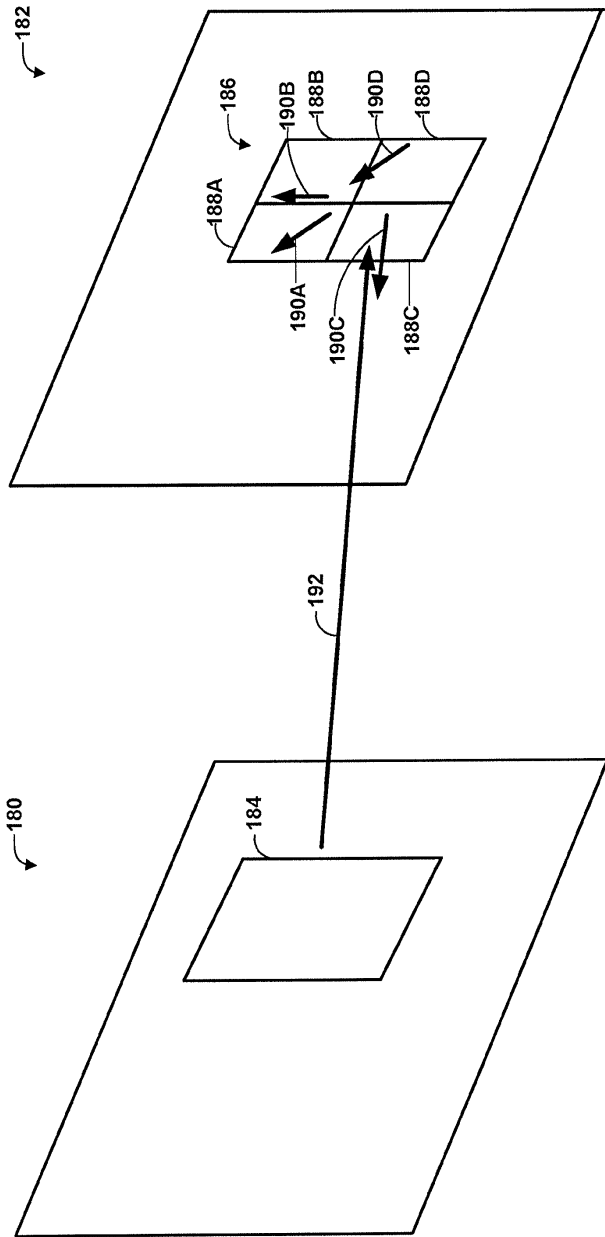
도면7



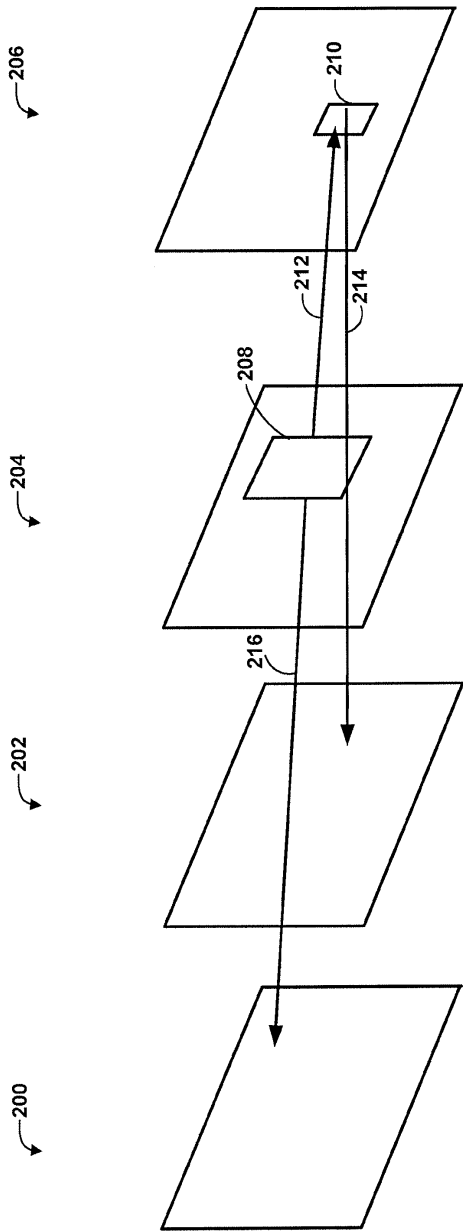
V0 (레퍼런스 뷰)

V1 (현재 뷰)

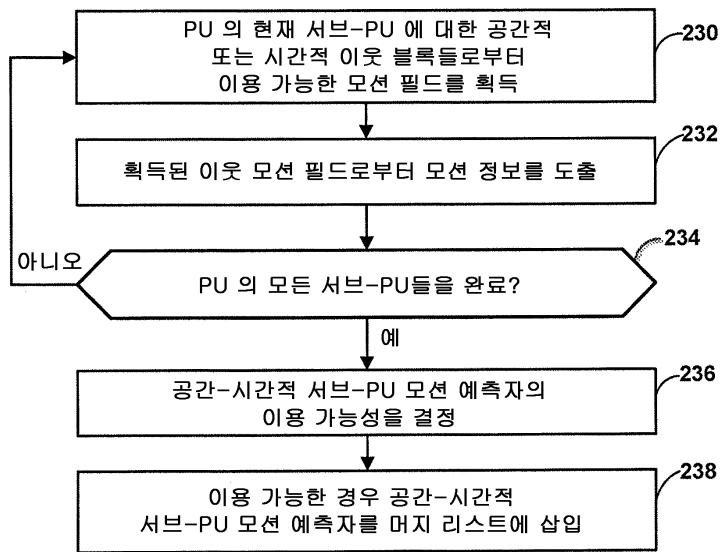
도면8



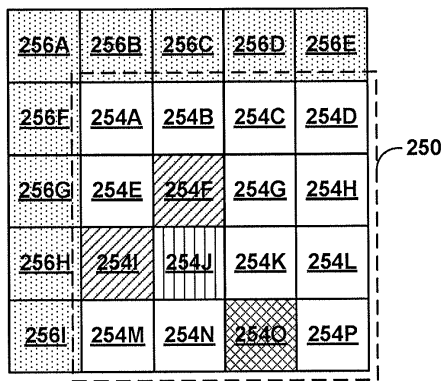
도면9



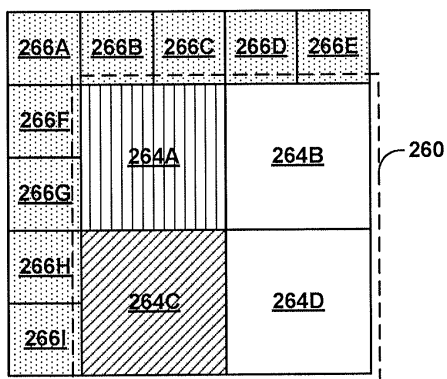
도면10



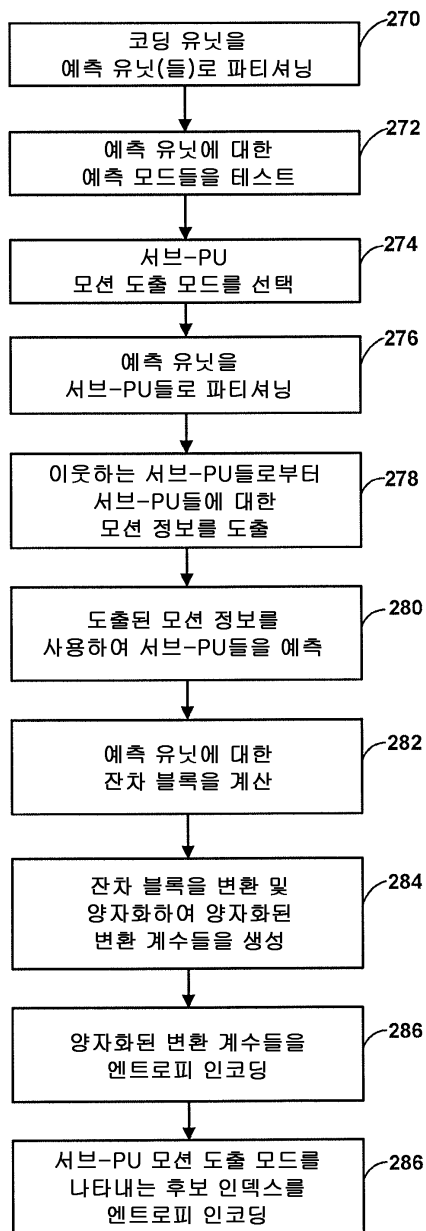
도면11a



도면11b



도면12



도면13

