



등록특허 10-2331683



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2021년11월25일  
(11) 등록번호 10-2331683  
(24) 등록일자 2021년11월23일

- (51) 국제특허분류(Int. Cl.)  
*HO4N 19/577* (2014.01) *HO4N 13/00* (2018.01)  
*HO4N 19/105* (2014.01) *HO4N 19/176* (2014.01)  
*HO4N 19/52* (2014.01) *HO4N 19/597* (2014.01)
- (52) CPC특허분류  
*HO4N 19/577* (2015.01)  
*HO4N 13/161* (2018.05)
- (21) 출원번호 10-2016-7018926
- (22) 출원일자(국제) 2015년01월09일  
심사청구일자 2019년12월23일
- (85) 번역문제출일자 2016년07월13일
- (65) 공개번호 10-2016-0106604
- (43) 공개일자 2016년09월12일
- (86) 국제출원번호 PCT/US2015/010878
- (87) 국제공개번호 WO 2015/106141  
국제공개일자 2015년07월16일
- (30) 우선권주장  
61/926,290 2014년01월11일 미국(US)  
14/592,633 2015년01월08일 미국(US)
- (56) 선행기술조사문헌

Li Zhang ET AL., “CE4: Further improvements on advanced residual prediction”, Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 6th Meeting: Geneva, CH, 2013.10.18., JCT3V-F0123.\*

Shiori Sugimoto ET AL., “3D-CE4.h: Simplification of inter-view ARP”, Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 7th Meeting: San Jose, US, 2014.01.09., JCT3V-G0158.\*

\*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 13 항

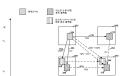
심사관 : 박상철

(54) 발명의 명칭 3D 비디오 코딩을 위한 블록-기반 진보된 잔차 예측

**(57) 요 약**

비디오 코딩에서의 진보된 잔차 예측 (advanced residual prediction; ARP) 을 위한 기법들은, 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 수신하는 것으로서, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 인터-뷰 예측을 이용하여 인코딩된, 상기 비디오 데이터의 제 1 인코딩된 블록을 수신하는 것, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하는 것, 및 제 1 예측 방향에 대해 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하는 것으로서, 참조 블록은 제 2 액세스 유닛 내에 있는, 상기 제 2 예측 방향에 대한 참조 블록들을 식별하는 것을 포함할 수도 있다.

**대 표 도**



(52) CPC특허분류

*H04N 19/105* (2015.01)

*H04N 19/176* (2015.01)

*H04N 19/52* (2015.01)

*H04N 19/597* (2015.01)

---

## 명세서

### 청구범위

#### 청구항 1

멀티뷰 비디오 데이터를 디코딩하기 위한 방법으로서,

제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 수신하는 단계로서, 비디오 데이터의 상기 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었고, 상기 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터-뷰 예측을 포함하고, 상기 진보된 잔차 예측은 잔차 예측자 블록의 잔차에 대해 비디오 데이터의 상기 제 1 인코딩된 블록의 잔차의 예측을 수행하는 것을 포함하는, 상기 비디오 데이터의 제 1 인코딩된 블록을 수신하는 단계;

비디오 데이터의 상기 제 1 인코딩된 블록을 위한 상기 제 1 예측 방향에 대한 제 1 시간적 모션 정보를 결정하는 단계;

상기 제 1 예측 방향에 대한 상기 제 1 시간적 모션 정보를 이용하여, 상기 제 1 예측 방향에 대한 제 1 시간적 참조 블록을 식별하는 단계로서, 상기 제 1 시간적 참조 블록은 상기 제 1 뷰에 있고, 상기 제 1 액세스 유닛과 상이한 제 2 액세스 유닛에 있는, 상기 제 1 시간적 참조 블록을 식별하는 단계;

비디오 데이터의 상기 제 1 인코딩된 블록을 위한 상기 제 1 예측 방향에 대한 디스패리티 백터를 유도하는 단계;

상기 제 1 예측 방향에 대한 상기 디스패리티 백터를 이용하여, 제 1 인터-뷰 참조 블록을 식별하는 단계로서, 상기 제 1 인터-뷰 참조 블록은 상기 제 1 액세스 유닛에 있고, 상기 제 1 뷰와 상이한 제 2 뷰에 있는, 상기 제 1 인터-뷰 참조 블록을 식별하는 단계;

상기 제 1 인터-뷰 참조 블록을 이용하여, 상기 제 1 인터-뷰 참조 블록을 식별하는 단계로서, 상기 제 1 예측 방향에 대한 상기 제 1 시간적 모션 정보와 상기 제 1 인터-뷰 참조 블록을 이용하여, 제 2 인터-뷰 참조 블록을 식별하는 단계로서, 상기 제 2 인터-뷰 참조 블록은 상기 제 2 뷰에 있고, 상기 제 2 액세스 유닛에 있는, 상기 제 2 인터-뷰 참조 블록을 식별하는 단계;

상기 제 1 인터-뷰 참조 블록 및 상기 제 2 인터-뷰 참조 블록 사이의 픽셀-대-픽셀 (pixel-by-pixel) 차이로부터 제 1 잔차 예측자 블록을 결정하는 단계;

비디오 데이터의 상기 제 1 인코딩된 블록을 위한 상기 제 2 예측 방향에 대한 제 1 디스패리티 모션 정보를 결정하는 단계;

상기 제 2 예측 방향에 대한 상기 제 1 디스패리티 모션 정보를 이용하여 제 3 인터-뷰 참조 블록을 식별하는 단계로서, 상기 제 3 인터-뷰 참조 블록은 상기 제 2 뷰에 있고, 상기 제 1 액세스 유닛에 있는, 상기 제 3 인터-뷰 참조 블록을 식별하는 단계;

상기 제 1 예측 방향에 대한 상기 제 1 시간적 모션 정보 및 상기 제 3 인터-뷰 참조 블록을 이용하여, 제 4 인터-뷰 참조 블록을 식별하는 단계로서, 상기 제 4 인터-뷰 참조 블록은 상기 제 2 뷰에 있고, 상기 제 2 액세스 유닛에 있는, 상기 제 4 인터-뷰 참조 블록을 식별하는 단계;

상기 제 1 시간적 참조 블록 및 상기 제 4 인터-뷰 참조 블록 사이의 픽셀-대-픽셀 차이로부터 제 2 잔차 예측자 블록을 결정하는 단계; 및

상기 제 1 시간적 참조 블록, 상기 제 1 잔차 예측자 블록 및 상기 제 2 잔차 예측자 블록을 이용하여 비디오 데이터의 상기 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행함으로써 비디오 데이터의 상기 제 1 인코딩된 블록을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하기 위한 방법.

#### 청구항 2

제 1 항에 있어서,

상기 비디오 데이터는 멀티뷰 고효율 비디오 코딩 (MV-HEVC) 비디오 데이터인, 비디오 데이터를 디코딩하기 위

한 방법.

### 청구항 3

제 1 항에 있어서,

상기 비디오 데이터는 멀티뷰-플러스-심도 (multiview-plus-depth) 비디오 데이터인, 비디오 데이터를 디코딩하기 위한 방법.

### 청구항 4

제 3 항에 있어서,

상기 비디오 데이터는 3D 고효율 비디오 코딩 (3D-HEVC) 비디오 데이터인, 비디오 데이터를 디코딩하기 위한 방법.

### 청구항 5

제 1 항에 있어서,

상기 제 2 뷰는 기본 뷰인, 비디오 데이터를 디코딩하기 위한 방법.

### 청구항 6

멀티뷰 비디오 데이터를 디코딩하도록 구성된 장치로서,

제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 저장하도록 구성된 비디오 데이터 메모리로서, 비디오 데이터의 상기 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었고, 상기 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터-뷰 예측을 포함하고, 상기 진보된 잔차 예측은 잔차 예측자 블록의 잔차에 대해 비디오 데이터의 상기 제 1 인코딩된 블록의 잔차의 예측을 수행하는 것을 포함하는, 상기 비디오 데이터 메모리; 및

상기 비디오 데이터 메모리와 통신하는 하나 이상의 프로세서들로서, 상기 하나 이상의 프로세서들은,

비디오 데이터의 상기 제 1 인코딩된 블록을 위한 상기 제 1 예측 방향에 대한 제 1 시간적 모션 정보를 결정하고;

상기 제 1 예측 방향에 대한 상기 제 1 시간적 모션 정보를 이용하여, 상기 제 1 예측 방향에 대한 제 1 시간적 참조 블록을 식별하는 것으로서, 상기 제 1 시간적 참조 블록은 상기 제 1 뷰에 있고, 상기 제 1 액세스 유닛과 상이한 제 2 액세스 유닛에 있는, 상기 제 1 시간적 참조 블록을 식별하며;

비디오 데이터의 상기 제 1 인코딩된 블록을 위한 상기 제 1 예측 방향에 대한 디스패리티 벡터를 유도하고;

상기 제 1 예측 방향에 대한 상기 디스패리티 벡터를 이용하여, 제 1 인터-뷰 참조 블록을 식별하는 것으로서, 상기 제 1 인터-뷰 참조 블록은 상기 제 1 액세스 유닛에 있고, 상기 제 1 뷰와 상이한 제 2 뷰에 있는, 상기 제 1 인터-뷰 참조 블록을 식별하며;

상기 제 1 예측 방향에 대한 상기 제 1 시간적 모션 정보와 상기 제 1 인터-뷰 참조 블록을 이용하여, 제 2 인터-뷰 참조 블록을 식별하는 것으로서, 상기 제 2 인터-뷰 참조 블록은 상기 제 2 뷰에 있고, 상기 제 2 액세스 유닛에 있는, 상기 제 2 인터-뷰 참조 블록을 식별하고;

상기 제 1 인터-뷰 참조 블록 및 상기 제 2 인터-뷰 참조 블록 사이의 픽셀-대-픽셀 (pixel-by-pixel) 차이로부터 제 1 잔차 예측자 블록을 결정하며;

비디오 데이터의 상기 제 1 인코딩된 블록을 위한 상기 제 2 예측 방향에 대한 제 1 디스패리티 모션 정보를 결정하고;

상기 제 2 예측 방향에 대한 상기 제 1 디스패리티 모션 정보를 이용하여 제 3 인터-뷰 참조 블록을 식별하는 것으로서, 상기 제 3 인터-뷰 참조 블록은 상기 제 2 뷰에 있고, 상기 제 1 액세스 유닛에 있는, 상기 제 3 인터-뷰 참조 블록을 식별하며;

상기 제 1 예측 방향에 대한 상기 제 1 시간적 모션 정보 및 상기 제 3 인터-뷰 참조 블록을 이용하여, 제 4 인터-뷰 참조 블록을 식별하는 것으로서, 상기 제 4 인터-뷰 참조 블록은 상기 제 2 뷰에 있고, 상기 제 2 액세스 유닛에 있는, 상기 제 4 인터-뷰 참조 블록을 식별하고;

상기 제 1 시간적 참조 블록 및 상기 제 4 인터-뷰 참조 블록 사이의 픽셀-대-픽셀 차이로부터 제 2 잔차 예측자 블록을 결정하며; 그리고

상기 제 1 잔차 예측자 블록 및 상기 제 2 잔차 예측자 블록을 이용하여 비디오 데이터의 상기 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행함으로써 비디오 데이터의 상기 제 1 인코딩된 블록을 디코딩하도록 구성되는, 상기 하나 이상의 프로세서들을 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

#### **청구항 7**

제 6 항에 있어서,

비디오 데이터의 상기 디코딩된 블록을 디스플레이하도록 구성된 디스플레이를 더 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

#### **청구항 8**

제 6 항에 있어서,

상기 비디오 데이터 메모리 및 상기 하나 이상의 프로세서들은 데스크톱 컴퓨터, 노트북 컴퓨터, 랩톱 컴퓨터, 셋톱 박스, 전화 핸드셋, 스마트폰, 스마트 패드, 태블릿 컴퓨터, 텔레비전, 카메라, 디지털 미디어 플레이어, 비디오 게임용 콘솔, 또는 비디오 스트리밍 디바이스 내에 실장되는, 비디오 데이터를 디코딩하도록 구성된 장치.

#### **청구항 9**

제 6 항에 있어서,

상기 비디오 데이터는 멀티뷰 고효율 비디오 코딩 (MV-HEVC) 비디오 데이터인, 비디오 데이터를 디코딩하도록 구성된 장치.

#### **청구항 10**

제 6 항에 있어서,

상기 비디오 데이터는 멀티뷰-플러스-심도 (multiview-plus-depth) 비디오 데이터인, 비디오 데이터를 디코딩하도록 구성된 장치.

#### **청구항 11**

제 10 항에 있어서,

상기 비디오 데이터는 3D 고효율 비디오 코딩 (3D-HEVC) 비디오 데이터인, 비디오 데이터를 디코딩하도록 구성된 장치.

#### **청구항 12**

제 6 항에 있어서,

상기 제 2 뷰는 기본 뷰인, 비디오 데이터를 디코딩하도록 구성된 장치.

#### **청구항 13**

명령들을 저장하는 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 실행될 경우, 비디오 데이터를 디코딩하도록 구성된 디바이스의 하나 이상의 프로세서들로 하여금 제 1 항 내지 제 5 항 중 어느 한 항에 따른 방법을 수행하게 하는, 컴퓨터 판독가능 저장 매체.

#### **청구항 14**

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

## 발명의 설명

### 기술 분야

- [0001] 이 출원은 2014년 1월 11일자로 출원된 미국 가출원 제 61/926,290호의 이익을 주장하고, 그 전체 내용은 그 전체적으로 참조로 본원에 편입된다.
- [0002] 이 개시물은 비디오 코딩에 관련된다.

### 배경 기술

- [0003] 디지털 비디오 능력들은, 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템 (digital direct broadcast system) 들, 무선 브로드캐스트 시스템들, 개인 정보 단말 (personal digital assistant; PDA) 들, 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더 (e-book reader) 들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들 새로 편입될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, 진보된 비디오 코딩 (Advanced Video Coding; AVC) 에 의해 정의된 표준들, 현재 개발 중에 있는 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, 및 이러한 표준들의 확장들에서 설명된 것들과 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신하고, 수신하고, 인코딩하고, 디코딩하고, 및/또는 저장할 수도 있다.

- [0004] 비디오 코딩 기법들은 비디오 시퀀스들에 내재된 중복성을 감소시키거나 제거하기 위한 공간적 (인트라-픽처 (intra-picture)) 예측 및/또는 시간적 (인터-픽처 (inter-picture)) 예측을 포함한다. 블록-기반 비디오 코딩을 위하여, 비디오 슬라이스 (예컨대, 비디오 프레임 또는 비디오 프레임의 부분) 는 비디오 블록들로 파티셔닝 (partitioning) 될 수도 있으며, 이 비디오 블록들은 또한, 트리블록 (treeblock) 들, 코딩 유닛 (coding unit; CU) 들 및/또는 코딩 노드들로서 지정될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처에서의 이웃하는 블록들 내의 참조 샘플들에 대한 공간적 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽처에서의 이웃하는 블록들 내의 참조 샘플들에 대한 공간적 예측, 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들로서 지정될 수도 있고, 참조 픽처들은 참조 프레임들로서 지정될 수도 있다.

- [0005] 공간적 또는 시간적 예측은 코딩되어야 할 블록에 대한 예측 블록으로 귀착된다. 잔차 데이터 (residual data) 는 코딩되어야 할 원래의 블록과 예측 블록과의 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 지시하는 모션 벡터와, 코딩된 블록과 예측 블록과의 사이의 차이를 표시하는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위하여, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어 잔차 변환 계수들로 귀착될 수도 있고, 그 후에 이러한 잔차 변환 계수들은 양자화될 수도 있다. 초기에 2 차원 어레이로 배열된 양자화된 변환 계수들은 변환 계수들의 1 차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩은 훨씬 더 많은 압축을 달성하기 위하여 적용될 수도 있다.

## 발명의 내용

### 과제의 해결 수단

- [0006] 일반적으로, 이 개시물은 코딩된 비디오 데이터가 2개 이상의 뷰 (view) 들을 포함하는 멀티뷰 비디오 코딩 (multiview video coding) 에 관련된다. 특히, 이 개시물은 진보된 잔차 예측 (advanced residual prediction; ARP) 에 관련된 다양한 기법들을 설명한다. 이 개시물의 기법들은 (잠재적인 보간 동작들로 예측 블록들을 생성하기 위한 배정된 모션 벡터들을 이용하여) ARP 또는 임의의 기초적인 모션 보상 프로세스들을 수행하기 위하여, 비디오 코더 (예컨대, 비디오 인코더 및/또는 비디오 디코더) 가 모션 정보를 액세스하는 횟수를 감소시킬 수도 있다. 이러한 방법으로, 모션 정보에 대한 더 적은 메모리 액세스들이 수행되므로, 비디오 코딩 (즉, 인코딩 또는 디코딩) 의 속도는 증가될 수도 있다.

[0007] 개시물의 일 예에서, 비디오 데이터를 디코딩하기 위한 방법은, 제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 수신하는 단계로서, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었고, 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터-뷰 (inter-view) 예측을 포함하는, 상기 비디오 데이터의 제 1 인코딩된 블록을 수신하는 단계, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보 (temporal motion information)를 결정하는 단계, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보 (disparity motion information)를 결정하는 단계, 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하는 단계로서, 참조 블록들은 제 1 액세스 유닛과는 상이한 액세스 유닛 내에 있는, 상기 제 2 예측 방향에 대한 참조 블록들을 식별하는 단계, 및 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행하는 단계를 포함한다.

[0008] 개시물의 또 다른 예에서, 비디오 데이터를 디코딩하도록 구성된 장치는, 제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 저장하도록 구성된 비디오 데이터 메모리로서, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었고, 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터-뷰 예측을 포함하는, 상기 비디오 데이터 메모리, 및 상기 비디오 데이터 메모리와 통신하는 하나 이상의 프로세서들로서, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하고, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보를 결정하고, 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하는 것으로서, 참조 블록들은 제 1 액세스 유닛과는 상이한 액세스 유닛 내에 있는, 상기 제 2 예측 방향에 대한 참조 블록들을 식별하고, 그리고 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행하도록 구성된 상기 하나 이상의 프로세서들을 포함한다.

[0009] 개시물의 또 다른 예에서, 비디오 데이터를 디코딩하도록 구성된 장치는, 제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 수신하기 위한 수단으로서, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었고, 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터-뷰 예측을 포함하는, 상기 비디오 데이터의 제 1 인코딩된 블록을 수신하기 위한 수단, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하기 위한 수단, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보를 결정하기 위한 수단, 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하기 위한 수단으로서, 참조 블록들은 제 1 액세스 유닛과는 상이한 액세스 유닛 내에 있는, 상기 제 2 예측 방향에 대한 참조 블록들을 식별하기 위한 수단, 및 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행하기 위한 수단을 포함한다.

[0010] 또 다른 예에서, 개시물은 명령들을 저장하는 컴퓨터-관독가능 저장 매체를 설명하고, 상기 명령들은, 실행될 경우, 비디오 데이터를 디코딩하도록 구성된 디바이스의 하나 이상의 프로세서들로 하여금, 제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 수신하게 하는 것으로서, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었고, 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터-뷰 예측을 포함하는, 상기 비디오 데이터의 제 1 인코딩된 블록을 수신하게 하고, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하게 하고, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보를 결정하게 하고, 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하게 하는 것으로서, 참조 블록들은 제 1 액세스 유닛과는 상이한 액세스 유닛 내에 있는, 상기 제 2 예측 방향에 대한 참조 블록들을 식별하게 하고, 그리고 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행하게 한다.

[0011] 개시물의 하나 이상의 예들의 세부사항들은 첨부한 도면들 및 이하의 설명에서 기재된다. 다른 특징들, 목적들, 및 장점들은 설명, 도면들, 및 청구항들로부터 명백할 것이다.

### 도면의 간단한 설명

[0012] 도 1 은 이 개시물에서 설명된 기법들을 사용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하

는 블록도이다.

도 2 는 예시적인 멀티뷰 인코딩 또는 디코딩 순서를 예시하는 그래픽 도면이다.

도 3 은 멀티뷰 비디오 코딩을 위한 예시적인 시간적 및 인터-뷰 예측 패턴을 예시하는 개념도이다.

도 4 는 3D 비디오에 대한 텍스처 (texture) 및 심도 (depth) 값들을 예시하는 개념도이다.

도 5 는 현재의 블록의 모션 정보의 예측을 위한 현재의 블록에 대한 이웃하는 블록들의 예시적인 관계를 예시하는 개념도이다.

도 6 은 현재의 블록의 모션 정보의 예측을 위한 인터-뷰 예측된 모션 벡터 후보 및 인터-뷰 디스파리티 모션 벡터 후보의 유도의 예를 예시하는 개념도이다.

도 7 은 공간적으로 이웃하는 블록들로부터, 현재의 비디오 블록에 대한 디스파리티 벡터가 현재의 비디오 블록에 관한 이웃하는 블록 기반 디스파리티 벡터 (Neighboring Block Based Disparity Vector; NBDV) 유도를 이용하여 유도될 수도 있는, 예시적인 공간적으로 이웃하는 블록들을 예시하는 개념도이다.

도 8 은 서브-예측 유닛 (prediction unit; PU) 인터-뷰 모션 예측을 예시하는 개념도이다.

도 9 는 시간적으로-예측된 비디오 블록의 시간적 진보된 잔차 예측 (ARP) 을 위한 예시적인 예측 구조를 예시하는 개념도이다.

도 10 은 시간적 ARP 를 위한 예시적인 양방향 예측 구조를 예시하는 개념도이다.

도 11 은 이 개시물에서 설명된 기법들에 따라 인터-뷰 예측된 비디오 블록의 인터-뷰 ARP 를 위한 예시적인 예측 구조를 예시하는 개념도이다.

도 12 는 하나의 참조 픽처 리스트에 대한 인터-뷰 예측과, 또 다른 참조 픽처 리스트에 대한 시간적 예측을 이용하는 양방향 ARP 를 위한 예시적인 예측 구조를 예시하는 개념도이다.

도 13 은 이 개시물의 기법들에 따라, 하나의 참조 픽처 리스트에 대한 인터-뷰 예측과, 또 다른 참조 픽처 리스에 대한 시간적 예측을 이용하는 양방향 ARP 를 위한 예시적인 예측 구조를 예시하는 개념도이다.

도 14 는 블록-기반 시간적 ARP 를 예시하는 개념도이다.

도 15 는 블록-기반 인터-뷰 ARP 를 예시하는 개념도이다.

도 16 은 서브-PU 병합 후보를 갖는 블록-기반 ARP 를 예시하는 개념도이다.

도 17 은 이 개시물에서 설명된 기법들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 18 은 이 개시물에서 설명된 기법들을 사용할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 19 는 이 개시물에서 설명된 기법들에 따라 비디오 블록을 인코딩하기 위한 예시적인 ARP 방법을 예시하는 흐름도이다.

도 20 은 이 개시물에서 설명된 기법들에 따라 비디오 블록을 디코딩하기 위한 예시적인 ARP 방법을 예시하는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

[0013]

일반적으로, 이 개시물은 코딩된 비디오 데이터가 2 개 이상의 뷰들을 포함하는 멀티뷰 비디오 코딩에 관련된다. 일부의 예들에서, 멀티뷰 비디오 코딩은 멀티뷰-플러스-심도 (multiview-plus-depth) 비디오 코딩 프로세스를 포함한다. 일부의 예들에서, 멀티뷰 코딩은 3D 비디오 코딩으로서 지칭될 수도 있는 3 차원 (3D) 비디오의 코딩을 포함할 수도 있다. 이 개시물의 다양한 예들에서는, 멀티-뷰 및/또는 3D 비디오 코딩 시퀀스들의 비-기본 (non-base) 뷰에서의 진보된 잔차 예측 (ARP) 을 위한 기법들이 설명된다. 이 개시물의 기법들은 ARP 또는 임의의 기초적인 인터-예측 (예컨대, 시간적 및/또는 인터-뷰 인터-예측 및 양방향-예측 (bi-prediction)) 을 수행하기 위하여, 비디오 코더 (예컨대, 비디오 인코더 및/또는 비디오 디코더) 가 예컨대, 메모리로부터 모션 정보를 액세스하는 횟수를 감소시킬 수도 있다. 이러한 방법으로, 모션 정보에 대한 더 적은 메모리 액세스들이 수행되므로, 비디오 코딩 (즉, 인코딩 또는 디코딩) 의 속도는 증가될 수도 있다.

[0014] 예를 들어, 이 개시물은 비디오 데이터를 디코딩하기 위한 방법을 설명하고, 상기 방법은, 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 수신하는 단계로서, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 인터-뷰 예측을 이용하여 인코딩된, 상기 비디오 데이터의 제 1 인코딩된 블록을 수신하는 단계, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하는 단계, 및 제 1 예측 방향에 대해 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하는 단계로서, 참조 블록은 제 2 액세스 유닛 내에 있는, 상기 제 2 예측 방향에 대한 참조 블록들을 식별하는 단계를 포함한다.

[0015] 도 1 은 이 개시물의 기법들을 사용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 에서 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 더 이후의 시간에 디코딩되어야 할 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 컴퓨터-판독가능 매체 (16) 를 통해 비디오 데이터를 목적지 디바이스 (14) 에 제공할 수도 있다.

소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 (set-top) 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들 중의 임의의 것을 포함할 수도 있다. 일부의 경우들에는, 소스 디바이스 (12) 및 목적지 디바이스 (14) 가 무선 통신을 위해 구비될 수도 있다.

[0016] 목적지 디바이스 (14) 는 컴퓨터-판독가능 매체 (16) 를 통해, 디코딩되어야 할 인코딩된 비디오 데이터를 수신 할 수도 있다. 컴퓨터-판독가능 매체 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 일부 예에서, 컴퓨터-판독가능 매체 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 직접 송신하는 것을 가능하게 하기 위한 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조될 수도 있고, 목적지 디바이스 (14) 로 송신될 수도 있다.

통신 매체는 라디오 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 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하기 위해 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0017] 일부의 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, 블루-레이 (Blu-ray) 디스크들, DVD 들, CD-ROM 들, 플래시 메모리 (flash memory), 휴발성 또는 비-휴발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적당한 디지털 저장 매체들과 같은, 다양한 분산되거나 국소적으로 액세스된 데이터 저장 매체들 중의 임의의 것을 포함할 수도 있다.

추가의 예에서, 저장 디바이스는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 저장 디바이스로부터 저장된 비디오 데이터를 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있으며 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 예시적인 파일 서버들은 (예컨대, 웹사이트를 위한) 웹 서버, FTP 서버, 네트워크 연결 저장 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터를 액세스할 수도 있다. 이것은, 파일 서버 상에 저장된 인코딩된 비디오 데이터를 액세스하기 위해 적당한 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다.

저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 그 조합일 수도 있다.

[0018] 이 개시물의 기법들은 무선 애플리케이션들 또는 세팅들로 반드시 제한되는 것은 아니다. 기법들은 오버-더-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, HTTP 를 통한 동적 적응 스트리밍 (dynamic adaptive streaming over HTTP; DASH) 과 같은 인터넷 스트리밍 비디오 송신들, 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중의 임의의 것의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부의 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐

스팅, 및/또는 화상 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위하여 일방향 (one-way) 또는 양방향 (two-way) 비디오 송신을 지원하도록 구성될 수도 있다.

[0019] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 심도 추정 유닛 (19), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 심도 이미지 기반 렌더링 (depth image based rendering; DIBR) 유닛 (31), 및 디스플레이 디바이스 (32) 를 포함한다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 외부 카메라와 같은 외부 비디오 소스 (18) 로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하는 것이 아니라, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0020] 도 1 의 예시된 시스템 (10) 은 하나의 예에 불과하다. 이 개시물의 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로, 이 개시물의 기법들은 비디오 인코딩 디바이스에 의해 수행되지만, 기법들은 또한, "CODEC" 으로서 전형적으로 지정된 비디오 인코더/디코더에 의해 수행될 수도 있다. 또한, 이 개시물의 기법들은 또한, 비디오 프리프로세서 (video preprocessor) 에 의해 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 예들에 불과하다. 일부의 예들에서, 디바이스들 (12, 14) 은, 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 이 때문에, 시스템 (10) 은 예컨대, 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 또는 화상 통화를 위하여, 비디오 디바이스들 (12, 14) 사이의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0021] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 및/또는 비디오 컨텐츠 제공자로부터 비디오를 수신하기 위한 비디오 공급 인터페이스를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (18) 는 소스 비디오로서, 또는 라이브 비디오 (live video), 아카이빙된 비디오 (archived video), 및 컴퓨터 생성된 비디오의 조합으로서, 컴퓨터 그래픽-기반 (computer graphics-based) 데이터를 생성할 수도 있다. 일부의 경우들에는, 비디오 소스 (18) 가 비디오 카메라일 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 위에서 언급된 바와 같이, 이 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에 있어서, 캡처된, 프리-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 다음으로, 인코딩된 비디오 정보는 출력 인터페이스 (22) 에 의해 컴퓨터-판독가능 매체 (16) 상으로 출력될 수도 있다.

[0022] 비디오 소스 (18) 는 비디오 데이터의 하나 이상의 뷰들을 비디오 인코더 (20) 에 제공할 수도 있다. 예를 들어, 비디오 소스 (18) 는 카메라들의 어레이에 대응할 수도 있고, 카메라들의 각각은 촬영되고 있는 특정 장면과 관련된 고유의 수평 위치를 가질 수도 있다. 대안적으로, 비디오 소스 (18) 는 예컨대, 컴퓨터 그래픽을 이용하여 이질적인 수평 카메라 관점들로부터 비디오 데이터를 생성할 수도 있다. 심도 추정 유닛 (19) 은 텍스처 이미지에서의 픽셀들에 대응하는 심도 픽셀들에 대한 값들을 결정하도록 구성될 수도 있다. 예를 들어, 심도 추정 유닛 (19) 은 사운드 내비게이션 및 탐지 (Sound Navigation and Ranging; SONAR) 유닛, 광검출 및 탐지 (Light Detection and Ranging; LIDAR) 유닛, 또는 장면의 비디오 데이터를 레코딩하면서 실질적으로 동시에 심도 값들을 직접적으로 결정할 수 있는 다른 유닛을 나타낼 수도 있다.

[0023] 추가적으로 또는 대안적으로, 심도 추정 유닛 (19) 은 상이한 수평 카메라 관점들로부터 실질적으로 동일한 시간에 캡처되었던 2 개 이상의 이미지들을 비교함으로써 심도 값들을 간접적으로 계산하도록 구성될 수도 있다. 이미지들에서의 실질적으로 유사한 픽셀 값들 사이의 수평 디스패리티를 계산함으로써, 심도 추정 유닛 (19) 은 장면에서의 다양한 객체들의 심도를 근사화할 수도 있다. 심도 추정 유닛 (19) 은 일부의 예들에서, 비디오 소스 (18) 와 기능적으로 통합될 수도 있다. 예를 들어, 비디오 소스 (18) 가 컴퓨터 그래픽 이미지를 생성할 때, 심도 추정 유닛 (19) 은 예컨대, 텍스처 이미지들을 렌더링하기 위하여 이용된 픽셀들 및 객체들의 z-좌표들을 이용하여 그래픽 객체들에 대한 실제적인 심도 맵들을 제공할 수도 있다.

[0024] 컴퓨터-판독가능 매체 (16) 는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 순시적 매체 (transient medium) 들, 또는 하드 디스크, 플래시 드라이브, 컴팩트 디스크, 디지털 비디오 디스크, 블루-레이 디스크, 또는 다른 컴퓨터-판독가능 매체들과 같은 저장 매체들 (즉, 비-일시적 저장 매체들) 을 포함할 수도 있다.

일부의 예들에서, 네트워크 서버 (도시되지 않음)는 예컨대, 네트워크 송신을 통해, 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신할 수도 있으며 인코딩된 비디오 데이터를 목적지 디바이스 (14)에 제공할 수도 있다. 유사하게, 디스크 스텬핑 (disc stamping) 서비스와 같은 매체 생산 설비의 컴퓨팅 디바이스는 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신할 수도 있고, 인코딩된 비디오 데이터를 포함하는 디스크를 생산할 수도 있다. 그러므로, 컴퓨터-판독가능 매체 (16)는 다양한 예들에서, 다양한 형태들의 하나 이상의 컴퓨터-판독가능 매체들을 포함하는 것으로 이해될 수도 있다.

[0025] 목적지 디바이스 (14)의 입력 인터페이스 (28)는 컴퓨터-판독가능 매체 (16)로부터 정보를 수신한다. 컴퓨터-판독가능 매체 (16)의 정보는, 블록들 및 다른 코딩된 유닛들, 예컨대, GOP들의 특성을 및/또는 프로세싱을 설명하는 선택스 엘리먼트들을 포함하는 선택스 정보로서, 비디오 인코더 (20)에 의해 정의되며 비디오 디코더 (30)에 의해 또한 이용되는 상기 선택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관 (cathode ray tube; CRT), 액정 디스플레이 (liquid crystal display; LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode; OLED) 디스플레이, 또는 또 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중의 임의의 것을 포함할 수도 있다. 일부의 예들에서, 디스플레이 디바이스 (32)는 예컨대, 시청자를 위한 3D 시각적 효과를 생성하기 위하여, 2개 이상의 뷰들을 동시에 또는 실질적으로 동시에 디스플레이할 수 있는 디바이스를 포함할 수도 있다.

[0026] 목적지 디바이스 (14)의 DIBR 유닛 (31)은 비디오 디코더 (30)로부터 수신된 디코딩된 뷰들의 텍스처 및 심도 정보를 이용하여 합성된 뷰들을 렌더링할 수도 있다. 예를 들어, DIBR 유닛 (31)은 대응하는 심도 맵들에서의 픽셀들의 값들의 함수로서, 텍스처 이미지들의 픽셀 데이터에 대한 수평 디스패리티를 결정할 수도 있다. 다음으로, DIBR 유닛 (31)은 결정된 수평 디스패리티에 의해 텍스처 이미지에서의 픽셀들을 좌측 또는 우측으로 오프셋시킴으로써 합성된 이미지를 생성할 수도 있다. 이러한 방식으로, 디스플레이 디바이스 (32)는 임의의 조합으로 디코딩된 뷰들 및/또는 합성된 뷰들에 대응할 수도 있는 하나 이상의 뷰들을 디스플레이할 수도 있다. 이 개시물의 기법들에 따르면, 비디오 디코더 (30)는 심도 범위들 및 카메라 파라미터들에 대한 원래의 그리고 업데이트된 정밀도 값들을, 뷰들을 적당하게 합성하기 위하여 심도 범위들 및 카메라 파라미터들을 이용할 수도 있는 DIBR 유닛 (31)에 제공할 수도 있다.

[0027] 도 1에서 도시되지 않았지만, 일부의 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통의 데이터 스트림 또는 별도의 데이터 스트림들에서 오디오 및 비디오 양자의 인코딩을 처리하기 위한 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용 가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol; UDP)과 같은 다른 프로토콜들을 준수할 수도 있다.

[0028] 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (digital signal processor; DSP)들, 애플리케이션 특정 집적 회로 (application specific integrated circuit; ASIC)들, 필드 프로그래밍 가능한 게이트 어레이 (field programmable gate array; FPGA)들, 개별로직, 소프트웨어, 하드웨어, 펌웨어 또는 그 임의의 조합들과 같은 다양한 적당한 인코더 회로부 중의 임의의 것으로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현될 때, 디바이스는 소프트웨어를 위한 명령들을 적당한 비-일시적 (non-transitory) 컴퓨터-판독가능 매체에서 저장할 수도 있고, 이 개시물의 기법들을 수행하기 위하여 하나 이상의 프로세서들을 이용하여 명령들을 하드웨어로 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 이들 중 어느 것도 조합된 인코더/디코더 (CODEC)의 일부로서 각각의 디바이스 내에 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 셀룰러 전화와 같은 무선 통신 디바이스를 포함할 수도 있다.

[0029] 개시물의 일 예에서, 비디오 디코더 (30)는 제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 수신하도록 구성될 수도 있고, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었고, 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터뷰 예측을 포함한다. 비디오 디코더 (30)는 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하고, 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보를 결정하도록 추가로 구성될 수도 있다. 비디오 디코더 (30)는 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하도록 추가로 구성될 수도 있고, 참조 블록들은 제 1 액세스 유닛과는 상이한 액세스 유닛 내에 있고, 제 2 예측

방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행한다. 이러한 방법으로, 제 1 예측 방향에 대한 시간적 모션 정보는 제 2 예측 방향에 대하여 재이용된다. 이와 같이, 제 2 예측 방향에 대응하는 제 1 인코딩된 블록의 모션 벡터에 의해 식별된 블록의 시간적 모션 정보가 액세스될 필요가 없으므로, 시간적 모션 정보의 더 적은 메모리 액세스들이 행해질 필요가 있고, 이에 따라, 더욱 신속한 비디오 디코딩을 허용한다. 게다가, ARP 를 수행할 때에 이용된 참조 블록들의 총 수는 6 으로부터 5 로 감소될 수도 있고, 이것은 승산 (multiplication) 및 가산 (addition) 동작들을 이용한 보간의 측면에서 더 적은 연산 부잡도로 귀착된다. 마찬가지로, 양방향 인터-예측을 수행할 때, 비디오 인코더 (20) 는 제 2 예측 방향을 인코딩할 때에 제 1 예측 방향에 대한 시간적 모션 정보를 재이용하도록 구성될 수도 있다.

[0030] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 개발 중에 있는 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 코딩 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HEVC Test Model; HM) 을 준수할 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MPEG-4, Part 10, 진보된 비디오 코딩 (AVC) 으로서 대안적으로 지칭된 ITU-T H.264 표준, 또는 ITU-T H.264/AVC 의 MVC 확장과 같은 이러한 표준들의 확장들과 같은 다른 전용 또는 산업 표준들에 따라 동작할 수도 있다. MVC 의 최신 합동 초안은 "Advanced video coding for generic audiovisual services (일반적인 시청각 서비스들을 위한 진보된 비디오 코딩)" 2010년 3 월자 ITU-T 추천안 H.264 에서 설명되어 있다. 특히, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC 표준의 3D 확장 (예컨대, 3D-HEVC) 을 포함하는 3D 및/또는 멀티-뷰 코딩 표준에 따라 동작할 수도 있다.

[0031] "HEVC 작업 초안 10" 또는 "WD10" 으로서 지칭된 HEVC 표준의 하나의 초안은 문서 JCTVC-L1003v34, Bross 등, "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call) {{FDIS & 최후 호출에 대한} 고효율 비디오 코딩 (HEVC) 텍스트 초안 10}", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11 의 비디오 코딩에 관한 합동 협력 팀 (Joint Collaborative Team on Video Coding; JCT-VC), 12차 회의: Geneva, CH, 2013년 1월 14-23 일에 설명되어 있고, 이 문서는 2015년 1월 5 일자로, [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 다운로드 가능하다.

[0032] HEVC 표준의 또 다른 초안은, 2015년 1월 5 일자로, [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/wg11/JCTVC-M0432-v3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip)로부터 입수 가능한, Bross 등, "Editors' proposed corrections to HEVC version 1 (HEVC 버전 1에 대한 편집자들의 제안된 정정들)", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11 의 비디오 코딩에 관한 합동 협력 팀 (JCT-VC), 13차 회의, Incheon, KR, 2013년 4월에서 설명된 "WD10 수정본들"로서 본원에서 지칭된다. HEVC에 대한 멀티뷰 확장, 즉, MV-HEVC 는 또한, JCT-3V에 의해 개발되고 있다.

[0033] 현재, VCEG 및 MPEG 의 3D 비디오 코딩에 관한 합동 협력 팀 (JCT-3C) 은 HEVC에 기초한 3DV 표준을 개발하고 있고, 그에 대한 표준화 노력들의 일부는 HEVC에 기초한 멀티-뷰 비디오 코덱 (MV-HEVC) 과, HEVC에 기초한 3D 비디오 코딩 (3D-HEVC)에 대한 또 다른 일부의 표준화를 포함한다. MV-HEVC를 위하여, 그것에는 하이-레벨 신택스 (high-level syntax; HLS) 변경들만이 있어서, HEVC 내의 코딩 유닛/예측 유닛 레벨에서의 어떤 모듈도 재설계될 필요가 없고 MV-HEVC를 위해 완전히 재이용될 수 없다는 것이 보장되어야 한다. 3D-HEVC를 위하여, 텍스처 및 심도 뷰들 양자에 대한, 코딩 유닛/예측 유닛 레벨에서의 코딩 툴들을 포함하는 새로운 코딩 툴들이 포함될 수도 있고 지원될 수도 있다.

[0034] 3D-HEVC를 위한 참조 소프트웨어 3D-HTM의 하나의 버전은 다음의 링크로부터 다운로드될 수 있다: [3D-HTM 버전 9.01r1]: [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSoftware/tags/HTM-9.0r1/](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-9.0r1/). 참조 소프트웨어 설명의 하나의 버전은 Li Zhang, Gerhard Tech, Krzysztof Wegner, Sehoon Yea, "Test Model 6 of 3D-HEVC and MV-HEVC (3D-HEVC 및 MV-HEVC의 테스트 모델 6)" JCT3V-F1005, ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11의 3D 비디오 코딩 확장 개발에 관한 합동 협력 팀, 6차 회의: Geneva, CH, 2013년 11월 (JCT3V-F1005)에서 설명되어 있다. JCT3V-F1005는 [http://phenix.int-sudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=1636](http://phenix.int-sudparis.eu/jct2/doc_end_user/current_document.php?id=1636)로부터 다운로드될 수도 있다.

[0035] 3D-HEVC의 하나의 작업 초안은 Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, "3D-HEVC Draft Text 2 (3D-HEVC 초안 텍스트 2)" JCT3V-F1001, ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11의 3D 비디오 코딩 확장 개발에 관한 합동 협력 팀, 6차 회의: Geneva, CH, 2013년 11월 (JCT3V-F1-001)에서 설명되어 있다. JCT3V-F1001은 다음으로부터 입수 가능하다: [http://phenix.int-sudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=1361](http://phenix.int-sudparis.eu/jct2/doc_end_user/current_document.php?id=1361). 최신 소프트웨어 설명 (문서 번호: E1005)

는 다음으로부터 입수 가능하다: [http://phenix.int-sudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=1360](http://phenix.int-sudparis.eu/jct2/doc_end_user/current_document.php?id=1360).

[0036] 3D-HEVC 를 위한 소프트웨어 3D-HTM 의 더욱 최신 버전은 다음의 링크로부터 다운로드될 수 있다: [3D-HTM 버전 12.0]: [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSoftware/tags/HTM-12.0/](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-12.0/). 3D-HEVC 의 대응하는 작업 초안 (문서 번호: I1001) 은 다음으로부터 입수 가능하다: [http://phenix.int-evry.fr/jct3v/doc\\_end\\_user/current\\_document.php?id=2299](http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=2299). 최신 소프트웨어 설명 (문서 번호: I1005) 은 다음으로부터 입수 가능하다: [http://phenix.int-evry.fr/jct3v/doc\\_end\\_user/current\\_document.php?id=2301](http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=2301).

[0037] 먼저, HEVC 의 예시적인 코딩 기법들이 논의될 것이다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로서 지정된 비디오 코딩 디바이스의 진화형 모델에 기초하였다. HM 은 예컨대, ITU-T H.264/AVC 에 따라 기존의 디바이스들에 관하여 비디오 코딩 디바이스들의 몇몇 추가적인 능력들을 추정하였다. 예를 들어, H.264 가 9 개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM 은 33 개만큼 많은 각도 인트라-예측 인코딩 모드들과, DC 및 평면 모드들을 제공할 수도 있다.

[0038] HEVC 및 다른 비디오 코딩 사양들에서, 비디오 시퀀스는 전형적으로 일련의 팩처들을 포함한다. 팩처들은 또한, "프레임들" 로서 지정될 수도 있다. 팩처는  $S_L$ ,  $S_{Cb}$ , 및  $S_{Cr}$  로 나타낸 3 개의 샘플 어레이들을 포함할 수도 있다.  $S_L$  은 루마 샘플 (luma sample) 들의 2 차원 어레이 (즉, 블록) 이다.  $S_{Cb}$  는 Cb 크로미넌스 샘플 (chrominance sample) 들의 2 차원 어레이이다.  $S_{Cr}$  은 Cr 크로미넌스 샘플들의 2 차원 어레이이다. 크로미넌스 샘플들은 또한, 본원에서 "크로마 (chroma)" 샘플들로서 지정될 수도 있다. 다른 사례들에서, 팩처는 단색일 수도 있고 루마 샘플들의 어레이를 포함하기만 할 수도 있다.

[0039] 팩처의 인코딩된 표현을 생성하기 위하여, 비디오 인코더 (20) 는 코딩 트리 유닛 (coding tree unit; CTU) 들의 세트를 생성할 수도 있다. CTU 들의 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2 개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 단색 팩처들, 또는 3 개의 별도의 컬러 평면들을 포함하는 팩처들에서는, CTU 가 단일 코딩 트리 블록과, 코딩 트리 블록의 샘플들을 코딩하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한, "트리 블록" 또는 "최대 코딩 유닛 (largest coding unit)" (LCU) 으로서 지정될 수도 있다. HEVC 의 CTU 들은 H.264/AVC 와 같은 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 반드시 특정 사이즈 (size) 로 제한되는 것은 아니고, 하나 이상의 코딩 유닛 (CU) 들을 포함할 수도 있다. 슬라이스는 레스터 스캔 순서 (raster scan order) 로 연속적으로 순서화된 정수 (integer number) 의 CTU 들을 포함할 수도 있다.

[0040] 코딩된 CTU 를 생성하기 위하여, 비디오 인코더 (20) 는 코딩 트리 블록들을 코딩 블록들, 이 때문에, 명칭 "코딩 트리 유닛들" 로 분할하기 위하여, CTU 의 코딩 트리 블록들에 대해 쿼드-트리 파티셔닝 (quad-tree partitioning) 을 재귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다. 코딩 유닛 (CU) 은, 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 가지는 팩처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2 개의 대응하는 코딩 블록들과, 코딩 블록들의 샘플들을 코딩하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 단색 팩처들, 또는 3 개의 별도의 컬러 평면들을 가지는 팩처들에서는, CU 가 단일 코딩 블록과, 코딩 블록의 샘플들을 코딩하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다.

[0041] 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록이다. CU 의 예측 유닛 (PU) 은 루마 샘플들의 예측 블록, 크로마 샘플들의 2 개의 대응하는 예측 블록들, 및 예측 블록들을 예측하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 단색 팩처들, 또는 3 개의 별도의 컬러 평면들을 가지는 팩처들에서는, PU 가 단일 예측 블록과, 예측 블록을 예측하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb, 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.

[0042] 비디오 인코더 (20) 는 PU 에 대한 예측 블록들을 생성하기 위하여 인트라 예측 또는 인터 예측을 이용할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위하여 인트라 예측을 이용할 경우, 비디오 인코더 (20) 는 PU 와 연관된 팩처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. HEVC 의 일부의 버전들에서는, 각각의 PU 의 루마 컴포넌트에 대하여, 인트라 예측 방법이 33 개의 각도 예측 모드들 (2 로부터 34 까지 인덱싱됨), DC 모드 (1 로 인덱싱됨), 및 평면 모드 (0 으로 인덱싱됨) 와 함께 사용

된다.

[0043] 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위하여 인터 예측을 이용할 경우, 비디오 인코더 (20) 는 PU 와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 인터 예측은 단방향 인터 예측 (즉, 단방향-예측 (uni-prediction) 또는 단방향-예측방식 예측 (uni-predictive prediction)) 또는 양방향 인터 예측 (즉, 양방향-예측 (bi-prediction) 또는 양방향-예측방식 예측 (bi-predictive prediction)) 일 수도 있다. 단방향-예측 또는 양방향-예측을 수행하기 위하여, 비디오 인코더 (20) 는 현재의 슬라이스에 대한 제 1 참조 픽처 리스트 (RefPicList0) 및 제 2 참조 픽처 리스트 (RefPicList1) 를 생성할 수도 있다. 참조 픽처 리스트들의 각각은 하나 이상의 참조 픽처들을 포함할 수도 있다. 단방향-예측을 이용할 때, 비디오 인코더 (20) 는 참조 픽처 내에서 참조 로케이션을 결정하기 위하여 RefPicList0 및 RefPicList1 의 어느 하나 또는 양자에서 참조 픽처들을 검색할 수도 있다. 또한, 단방향-예측을 이용할 때, 비디오 인코더 (20) 는 참조 로케이션에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU 에 대한 예측 샘플 블록들을 생성할 수도 있다. 또한, 단방향-예측을 이용할 때, 비디오 인코더 (20) 는 PU 의 예측 블록과 참조 로케이션 사이의 공간적 변위를 표시하는 단일 모션 벡터 (motion vector) 를 생성할 수도 있다. PU 의 예측 블록과 참조 로케이션 사이의 공간적 변위를 표시하기 위하여, 모션 벡터는 PU 의 예측 블록과 참조 로케이션 사이의 수평 변위를 특정하는 수평 컴포넌트를 포함할 수도 있고, PU 의 예측 블록과 참조 로케이션 사이의 수직 변위를 특정하는 수직 컴포넌트를 포함할 수도 있다.

[0044] PU 를 인코딩하기 위하여 양방향-예측을 이용할 때, 비디오 인코더 (20) 는 RefPicList0 내의 참조 픽처에서의 제 1 참조 로케이션, 및 RefPicList1 내의 참조 픽처에서의 제 2 참조 로케이션을 결정할 수도 있다. 다음으로, 비디오 인코더 (20) 는 제 1 및 제 2 참조 로케이션들에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU 에 대한 예측 블록들을 생성할 수도 있다. 또한, PU 를 인코딩하기 위하여 양방향-예측을 이용할 때, 비디오 인코더 (20) 는 PU 의 샘플 블록과 제 1 참조 로케이션 사이의 공간적 변위를 표시하는 제 1 모션 벡터, 및 PU 의 예측 블록과 제 2 참조 로케이션 사이의 공간적 변위를 표시하는 제 2 모션 벡터를 생성할 수도 있다.

[0045] 전형적으로, B 픽처의 제 1 또는 제 2 참조 픽처 리스트 (예컨대, RefPicList0 또는 RefPicList1) 에 대한 참조 픽처 리스트 구성은 2 개의 단계들을 포함한다: 참조 픽처 리스트 초기화 및 참조 픽처 리스트 재순서화 (수정). 참조 픽처 리스트 초기화는, (또한, 디코딩된 픽처 버퍼로서 알려진) 참조 픽처 메모리 내의 참조 픽처들을 POC (픽처의 디스플레이 순서와 정렬된 픽처 순서 카운트 (Picture Order Count)) 값들의 순서에 기초한 리스트 내에 놓는 명시적 메커니즘이다. 참조 픽처 리스트 재순서화 메커니즘은 참조 픽처 리스트 초기화 단계 동안에 리스트 내에 놓여졌던 픽처의 위치를 임의의 새로운 위치로 수정할 수 있거나, 픽처가 초기화된 리스트에 속하지 않더라도 참조 픽처 메모리에서의 임의의 참조 픽처를 임의의 위치에 놓을 수 있다. 참조 픽처 리스트 재순서화 (수정) 후의 일부 픽처들은 리스트에서 바로 추가의 위치에 놓여질 수도 있다. 그러나, 픽처의 위치가 리스트의 활성 참조 픽처들의 수를 초과할 경우, 픽처는 최종 참조 픽처 리스트의 엔트리 (entry) 로서 고려되지 않는다. 활성 참조 픽처들의 수는 각각의 리스트에 대한 슬라이스 헤더에서 시그널링될 수도 있다.

[0046] 참조 픽처 리스트들이 구성된 후 (즉, 이용가능하다면, RefPicList0 및 RefPicList1), 참조 픽처 리스트에 대한 참조 인덱스는 참조 픽처 리스트 내에 포함된 임의의 참조 픽처를 식별하기 위하여 이용될 수 있다.

[0047] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU 들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성한 후, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에서의 각각의 샘플은 CU 의 예측 루마 블록들 중의 하나에서의 루마 샘플과, CU 의 원래의 루마 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 표시한다. 게다가, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 생성할 수도 있다.

CU 의 Cb 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cb 블록들 중의 하나에서의 Cb 샘플과, CU 의 원래의 Cb 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 표시할 수도 있다. 비디오 인코더 (20) 는 또한, CU 에 대한 Cr 잔차 블록을 생성할 수도 있다. CU 의 Cr 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cr 블록들 중의 하나에서의 Cr 샘플과, CU 의 원래의 Cr 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 표시할 수도 있다.

[0048] 또한, 비디오 인코더 (20) 는 CU 의 루마, Cb, 및 Cr 잔차 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해하기 위하여 퀼드-트리 파티셔닝을 이용할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형 (예컨대, 정사각형 또는 비-정사각형) 블록이다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하기 위하여 이용된 신택스

구조들을 포함할 수도 있다. 이에 따라, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록 (sub-block) 일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다. 단색 픽처들, 또는 3 개의 별도의 컬러 평면들을 가지는 픽처들에서는, TU 가 단일 변환 블록과, 변환 블록의 샘플들을 변환하기 위하여 이용된 신택스 구조들을 포함할 수도 있다.

[0049] 비디오 인코더 (20) 는 TU 에 대한 루마 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용할 수도 있다. 계수 블록은 변환 계수들의 2 차원 어레이일 수도 있다. 변환 계수는 스칼라 량 (scalar quantity) 일 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cb 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용할 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cr 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용할 수도 있다.

[0050] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록, 또는 Cr 계수 블록) 을 생성한 후, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로, 변환 계수들을 나타내기 위하여 이용된 데이터의 양을 가능한 대로 감소시키기 위하여 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 신택스 엘리먼트들에 대해 컨텍스트-적응 이진 산술 코딩 (Context-Adaptive Binary Arithmetic Coding; CABAC) 을 수행할 수도 있다.

[0051] 비디오 인코더 (20) 는 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 네트워크 추상화 계층 (network abstraction layer; NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은 에뮬레이션 방지 비트 (emulation prevention bit) 들로 필요에 따라 산재된 원시 바이트 시퀀스 페이로드 (raw byte sequence payload; RBSP) 의 형태로, NAL 유닛에서의 데이터의 태입의 표시 및 그 데이터를 포함하는 바이트들을 포함하는 신택스 구조이다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하고 RBSP 를 캡슐화 (encapsulate) 한다. NAL 유닛 헤더는 NAL 유닛 태입 코드를 표시하는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 태입 코드는 NAL 유닛의 태입을 표시한다. RBSP 는 NAL 유닛 내에서 캡슐화되는 정수의 바이트들을 포함하는 신택스 구조일 수도 있다. 일부의 사례들에서, RBSP 는 제로 비트 (zero bit) 들을 포함한다.

[0052] 상이한 태입들의 NAL 유닛들은 상이한 태입들의 RBSP 들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 태입은 픽처 파라미터 세트 (PPS) 에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 2 태입은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있으며, NAL 유닛의 제 3 태입은 SEI 에 대한 RBSP 를 캡슐화할 수도 있는 등등과 같다. (파라미터 세트들 및 SEI 메시지들에 대한 RBSP들과 대조적으로) 비디오 코딩 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 비디오 코딩 계층 (NCL) NAL 유닛들로서 지칭될 수도 있다.

[0053] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 게다가, 비디오 디코더 (30) 는 비트스트림으로부터 신택스 엘리먼트들을 획득하기 위하여 비트스트림을 파싱 (parse) 할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 재구성할 수도 있다. 비디오 데이터를 재구성하기 위한 프로세스는 비디오 인코더 (20) 에 의해 수행된 프로세스와 일반적으로 상반적일 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재의 CU 의 PU 들에 대한 예측 블록들을 결정하기 위하여 PU 들의 모션 벡터들을 이용할 수도 있다. 게다가, 비디오 디코더 (30) 는 현재의 CU 의 TU 들과 연관된 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 TU 들과 연관된 변환 블록들을 재구성하기 위하여 계수 블록들에 대해 역변환들을 수행할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 PU 들에 대한 예측 블록들의 샘플들을 현재의 CU 의 TU 들의 변환 블록들의 대응하는 샘플들에 추가함으로써, 현재의 CU 의 코딩 블록들을 재구성할 수도 있다. 픽처의 각각의 CU 에 대한 코딩 블록들을 재구성함으로써, 비디오 디코더 (30) 는 픽처를 재구성할 수도 있다.

[0054] 일부의 예들에서, 비디오 인코더 (20) 는 병합 모드 또는 진보된 모션 벡터 예측 (advanced motion vector prediction; AMVP) 모드를 이용하여 PU 의 모션 정보를 시그널링할 수도 있다. 다시 말해서, HEVC 에서는, 모션 파라미터들의 예측을 위한 2 개의 모드들이 있고, 하나는 병합 모드이고 다른 하나는 AMVP 이다. 모션 예측은 하나 이상의 다른 비디오 유닛들의 모션 정보에 기초한 비디오 유닛 (예컨대, PU) 의 모션 정보의 결정을 포함할 수도 있다. PU 의 모션 정보는 PU 의 모션 벡터 (들) 및 PU 의 참조 인덱스 (들) 를 포함할 수도

있다.

[0055] 비디오 인코더 (20) 가 병합 모드를 이용하여 현재의 PU 의 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 병합 후보 리스트를 생성한다. 다시 말해서, 비디오 인코더 (20) 는 모션 벡터 예측자 리스트 구성 프로세스를 수행할 수도 있다. 병합 후보 리스트는 현재의 PU 와 공간적으로 또는 시간적으로 이웃하는 PU 들의 모션 정보를 표시하는 병합 후보들의 세트를 포함한다. 즉, 병합 모드에서는, 모션 파라미터들 (예컨대, 참조 인덱스들, 모션 벡터들 등) 의 후보 리스트가 구성되고, 여기서, 후보는 공간적 및 시간적으로 이웃하는 블록들로부터 나올 수 있다. 일부의 예들에서, 후보들은 또한 인위적으로 생성된 후보를 포함할 수도 있다.

[0056] 또한, 병합 모드에서, 비디오 인코더 (20) 는 병합 후보 리스트로부터 병합 후보를 선택할 수도 있고, 선택된 병합 후보에 의해 표시된 모션 정보를 현재의 PU 의 모션 정보로서 이용할 수도 있다. 비디오 인코더 (20) 는 선택된 병합 후보의 병합 후보 리스트에서의 위치를 시그널링할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 인덱스를 후보 리스트로 송신함으로써 선택된 모션 벡터 파라미터들을 시그널링할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터, 후보 리스트로의 인덱스 (즉, 후보 리스트 인덱스) 를 획득할 수도 있다. 게다가, 비디오 디코더 (30) 는 동일한 병합 후보 리스트를 생성할 수도 있고, 선택된 병합 후보의 위치의 표시에 기초하여, 선택된 병합 후보를 결정할 수도 있다. 다음으로, 비디오 디코더 (30) 는 현재의 PU 에 대한 예측 블록들을 생성하기 위하여 선택된 병합 후보의 모션 정보를 이용할 수도 있다. 즉, 비디오 디코더 (30) 는 후보 리스트 인덱스에 적어도 부분적으로 기초하여, 후보 리스트에서의 선택된 후보를 결정할 수도 있고, 여기서, 선택된 후보는 현재의 PU 에 대한 모션 벡터를 특정한다. 이러한 방법으로, 디코더 측에서는, 일단 인덱스가 디코딩되면, 인덱스가 지시하는 대응하는 블록의 모든 모션 파라미터들이 현재의 PU 에 의해 승계될 수도 있다.

[0057] 스kip 모드 (skip mode) 는 병합 모드와 유사하다. 스kip 모드에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 병합 모드에서 병합 후보 리스트를 이용하는 것과 동일한 방법으로 병합 후보 리스트를 생성 및 이용한다. 그러나, 비디오 인코더 (20) 가 스kip 모드를 이용하여 현재의 PU 의 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 현재의 PU 에 대한 임의의 잔차 데이터를 시그널링하지 않는다. 따라서, 비디오 디코더 (30) 는 잔차 데이터의 이용 없이, 병합 후보 리스트에서의 선택된 후보의 모션 정보에 의해 표시된 참조 블록에 기초하여 PU 에 대한 예측 블록을 결정할 수도 있다.

[0058] AMVP 모드는, 비디오 인코더 (20) 가 후보 리스트를 생성할 수도 있고 후보 리스트로부터 후보를 선택할 수도 있다는 점에서 병합 모드와 유사하다. 그러나, 비디오 인코더 (20) 가 AMVP 모드를 이용하여 현재의 PU 의 RefPicListX 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 현재의 PU 에 대한 RefPicListX MVP 플래그를 시그널링하는 것에 추가하여, 현재의 PU 에 대한 RefPicListX 모션 벡터 차이 (motion vector difference; MVD) 및 현재의 PU 에 대한 RefPicListX 참조 인덱스를 시그널링할 수도 있다. 현재의 PU 에 대한 RefPicListX MVP 플래그는 AMVP 후보 리스트에서 선택된 AMVP 후보의 위치를 표시할 수도 있다. 현재의 PU 에 대한 RefPicListX MVD 는 현재의 PU 의 RefPicListX 모션 벡터와, 선택된 AMVP 후보의 모션 벡터와의 사이의 차이를 표시할 수도 있다. 이러한 방법으로, 비디오 인코더 (20) 는 RefPicListX 모션 벡터 예측자 (motion vector predictor; MVP) 플래그, RefPicListX 참조 인덱스 값, 및 RefPicListX MVD 를 시그널링함으로써 현재의 PU 의 RefPicListX 모션 정보를 시그널링할 수도 있다. 다시 말해서, 현재의 PU 에 대한 모션 벡터를 나타내는 비트스트림에서의 데이터는 참조 인덱스, 후보 리스트로의 인덱스, 및 MVD 를 나타내는 데이터를 포함할 수도 있다.

[0059] 또한, 현재의 PU 의 모션 정보가 AMVP 모드를 이용하여 시그널링될 때, 비디오 디코더 (30) 는 비트스트림으로부터, 현재의 PU 에 대한 MVD 및 MVP 플래그를 획득할 수도 있다. 비디오 디코더 (30) 는 동일한 AMVP 후보 리스트를 생성할 수도 있고, MVP 플래그에 기초하여, 선택된 AMVP 후보를 결정할 수도 있다. 비디오 디코더 (30) 는 선택된 AMVP 후보에 의해 표시된 모션 벡터에 MVD 를 추가함으로써 현재의 PU 의 모션 벡터를 복원할 수도 있다. 즉, 비디오 디코더 (30) 는 선택된 AMVP 후보에 의해 표시된 모션 벡터 및 MVD 에 기초하여, 현재의 PU 의 모션 벡터를 결정할 수도 있다. 다음으로, 비디오 디코더 (30) 는 현재의 PU 에 대한 예측 블록들을 생성하기 위하여 현재의 PU 의 복원된 모션 벡터 또는 모션 벡터들을 이용할 수도 있다.

[0060] 비디오 디코더 (30) 가 현재의 PU 에 대한 AMVP 후보 리스트를 생성할 때, 비디오 디코더 (30) 는 현재의 PU 와 공간적으로 이웃하는 로케이션들을 커버하는 PU 들 (즉, 공간적으로 이웃하는 PU 들) 의 모션 정보에 기초하여 하나 이상의 AMVP 후보들을 유도할 수도 있다. PU 의 예측 블록이 로케이션을 포함할 때, PU 는 로케이션을 커버할 수도 있다.

- [0061] 현재의 PU 와 시간적으로 이웃하는 PU (즉, 현재의 PU 와는 상이한 시간 인스턴스에 있는 PU) 의 모션 정보에 기초하고 있는 병합 후보 리스트 또는 AMVP 후보 리스트에서의 후보는 TMVP 로서 지칭될 수도 있다. 즉, TMVP 는 HEVC 의 코딩 효율을 개선시키기 위해 이용될 수도 있고, 다른 코딩 툴들과는 상이하게, TMVP 는 디코딩된 픽처 베패, 더욱 구체적으로 참조 픽처 리스트에서 프레임의 모션 벡터를 액세스할 필요가 있을 수도 있다.
- [0062] TMVP 들의 이용은 CVS(coded video sequence; 코딩된 비디오 시퀀스)-대-CVS 에 기초하여, 슬라이스-대-슬라이스에 기초하여, 또는 또 다른 것에 기초하여 인에이블 (enable) 또는 디스에이블 (disable) 될 수도 있다. SPS 에서의 신택스 엘리먼트 (예컨대, sps\_temporal\_mvp\_enable\_flag) 는 TMVP 들의 이용이 CVS 에 대해 인에이블되는지 여부를 표시할 수도 있다. 또한, TMVP 들의 이용이 CVS 에 대해 인에이블될 때, TMVP 들의 이용은 CVS 내의 특정한 슬라이스들에 대해 인에이블 또는 디스에이블될 수도 있다. 예를 들어, 슬라이스 헤더에서의 신택스 엘리먼트 (예컨대, slice\_temporal\_mvp\_enable\_flag) 는 TMVP 들의 이용이 슬라이스에 대해 인에이블되는지 여부를 표시할 수도 있다. 이에 따라, 인터 예측된 슬라이스에서, TMVP 가 전체의 CVS 에 대해 인에이블될 때 (예컨대, SPS 에서의 sps\_temporal\_mvp\_enable\_flag 가 1 로 설정됨), TMVP 가 현재의 슬라이스에 대해 인에이블되는지 여부를 표시하기 위하여, slice\_temporal\_mvp\_enable\_flag 가 슬라이스 헤더에서 시그널링된다.
- [0063] TMVP 를 결정하기 위하여, 비디오 코더는 먼저, 현재의 PU 와 공동-위치되는 (co-located) PU 를 포함하는 참조 픽처를 식별할 수도 있다. 다시 말해서, 비디오 코더는 공동-위치된 픽처를 식별할 수도 있다. 현재의 픽처의 현재의 슬라이스가 B 슬라이스 (즉, 양방향으로 인터 예측된 PU 들을 포함하도록 허용되는 슬라이스) 일 경우, 비디오 인코더 (20) 는 슬라이스 헤더에서, 공동-위치된 픽처가 RefPicList0 또는 RefPicList1 로부터인지 여부를 표시하는 신택스 엘리먼트 (예를 들어, collocated\_from\_10\_flag) 를 시그널링할 수도 있다. 다시 말해서, TMVP 들의 이용이 현재의 슬라이스에 대해 인에이블되고, 현재의 슬라이스가 B 슬라이스 (예컨대, 양방향 인터 예측된 PU 들을 포함하도록 허용되는 슬라이스) 일 때, 비디오 인코더 (20) 는 공동-위치된 픽처가 RefPicList0 또는 RefPicList1 에 있는지 여부를 표시하기 위하여 슬라이스 헤더에서 신택스 엘리먼트 (예컨대, collocated\_from\_10\_flag) 를 시그널링할 수도 있다. 다시 말해서, TMVP 를 얻기 위하여, 먼저, 공동-위치된 픽처가 식별되어야 한다. 현재의 픽처가 B 슬라이스일 경우, collocated\_from\_10\_flag 는 공동-위치된 픽처가 RefPicList0 또는 RefPicList1 로부터인지 여부를 표시하기 위하여 슬라이스 헤더에서 시그널링된다.
- [0064] 비디오 디코더 (30) 가 공동-위치된 픽처를 포함하는 참조 픽처 리스트를 식별한 후, 비디오 디코더 (30) 는 식별된 참조 픽처 리스트에서 픽처 (즉, 공동-위치된 픽처) 를 식별하기 위하여, 슬라이스 헤더에서 시그널링될 수도 있는 또 다른 신택스 엘리먼트 (예컨대, collocated\_ref\_idx) 를 이용할 수도 있다. 즉, 참조 픽처 리스스트가 식별된 후, 슬라이스 헤더에서 시그널링된 collocated\_ref\_idx 는 참조 픽처 리스트에서 픽처를 식별하기 위하여 이용된다.
- [0065] 비디오 코더는 공동-위치된 픽처를 검사함으로써 공동-위치된 PU 를 식별할 수도 있다. TMVP 는 공동-위치된 PU 를 포함하는 CU 의 우측-하부 PU 의 모션 정보, 또는 이 PU 를 포함하는 CU 의 중심 PU 들 내의 우측-하부 PU 의 모션 정보 중의 어느 하나를 표시할 수도 있다. 이에 따라, 이 PU 를 포함하는 CU 의 우측-하부 PU 의 모션, 또는 이 PU 를 포함하는 CU 의 중심 PU 들 내의 우측-하부 PU 의 모션이 이용된다. 공동-위치된 PU 를 포함하는 CU 의 우측-하부 PU 는 PU 의 예측 블록의 하부-우측 샘플의 바로 하부 및 우측의 로케이션을 커버하는 PU 일 수도 있다. 다시 말해서, TMVP 는, 참조 픽처에 있으며 현재의 PU 의 하부 우측 코너와 공동-위치되어 있는 로케이션을 커버하는 PU 의 모션 정보를 표시할 수도 있거나, TMVP 는, 참조 픽처에 있으며 현재의 PU 의 중심과 공동-위치되어 있는 로케이션을 커버하는 PU 의 모션 정보를 표시할 수도 있다.
- [0066] 상기 프로세스에 의해 식별된 모션 벡터들 (즉, TMVP 의 모션 벡터들) 이 병합 모드 또는 AMVP 모드에 대한 모션 후보를 생성하기 위하여 이용될 때, 비디오 코더는 (POC 값에 의해 반영된) 시간적 로케이션에 기초하여 모션 벡터들을 스케일링 (scaling) 할 수도 있다. 예를 들어, 비디오 코더는 현재의 픽처 및 참조 픽처의 POC 값들의 차이가 더 클 때에 더 큰 양들만큼 모션 벡터의 크기 (magnitude) 를 증가시킬 수도 있고, 현재의 픽처 및 참조 픽처의 POC 값들 사이의 차이가 더 작을 때에 더 작은 양들만큼 모션 벡터의 크기를 증가시킬 수도 있다.
- [0067] TMVP 로부터 유도된 시간적 병합 후보에 대한 모든 가능한 참조 픽처 리스트들의 타겟 참조 인덱스는 항상 0 으로 설정될 수도 있다. 그러나, AMVP 에 대하여, 모든 가능한 참조 픽처들의 타겟 참조 인덱스는 디코딩된 참조 인덱스와 동일하게 설정된다. 다시 말해서, TMVP 로부터 유도된 시간적 병합 후보에 대한 모든 가능한

참조 픽처 리스트들의 타겟 참조 인덱스는 항상 0 으로 설정되는 반면, AMVP 에 대하여, 그것은 디코딩된 참조 인덱스와 동일하게 설정된다. HEVC 에서, SPS 는 플래그 (예컨대, sps\_temporal\_mvp\_enable\_flag) 를 포함할 수도 있고, 슬라이스 헤더는 sps\_temporal\_mvp\_enable\_flag 가 1 과 동일할 때에 플래그 (예컨대, pic\_temporal\_mvp\_enable\_flag) 를 포함할 수도 있다. pic\_temporal\_mvp\_enable\_flag 및 temporal\_id 의 양자가 특정한 픽처에 대해 0 과 동일할 때, 디코딩 순서에서 그 특정한 픽처 전의 픽처들로부터의 모션 벡터는 특정한 픽처, 또는 디코딩 순서에서의 특정한 픽처 이후의 픽처의 디코딩 시에 TMVP 로서 이용되지 않는다.

[0068] 일부의 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)(도 1) 는 멀티뷰 및/또는 3D 비디오 코딩, 예컨대, 2 개 이상의 뷰들을 포함하는 비디오 데이터의 코딩을 위한 기법들을 채용할 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 는 2 개 이상의 뷰들에 대한 인코딩된 비디오 데이터를 포함하는 비트스트림을 인코딩할 수도 있고, 비디오 디코더 (30) 는 2 개 이상의 뷰들을 예컨대, 디스플레이 디바이스 (32) 에 제공하기 위하여 인코딩된 비디오 데이터를 디코딩할 수도 있다. 일부의 예들에서, 비디오 디코더 (30) 는 디스플레이 디바이스 (32) 가 3D 비디오를 디스플레이하는 것을 가능하게 하기 위하여 비디오 데이터의 다수의 뷰들을 제공할 수도 있다. 일부의 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 예컨대, 멀티뷰 코딩 또는 멀티뷰 플러스 심도 코딩 프로세스들이 이용되는 HEVC 표준의 3D-HEVC 확장을 준수할 수도 있다. 멀티뷰 및/또는 3D 비디오 코딩은 2 개 이상의 텍스처 뷰들, 및/또는 텍스처 및 심도 컴포넌트들을 포함하는 뷰들의 코딩을 수반할 수도 있다. 일부의 예들에서, 비디오 인코더 (20) 에 의해 인코딩되고 비디오 디코더 (30) 에 의해 디코딩된 비디오 데이터는 즉, "액세스 유닛" 내의 임의의 소정의 시간 인스턴스에서의 2 개 이상의 픽처들, 또는 임의의 소정의 시간 인스턴스에서의 2 개 이상의 픽처들이 유도될 수 있는 데이터를 포함한다.

[0069] 일부의 예들에서, 디바이스, 예컨대, 비디오 소스 (18) 는 공통 장면을 캡처하기 위하여, 예를 들어, 2 개 이상의 공간적으로 오프셋된 카메라들, 또는 다른 비디오 캡처 디바이스들을 이용함으로써 2 개 이상의 픽처들을 생성할 수도 있다. 약간 상이한 수평 위치들로부터 동시에, 또는 거의 동시에 캡처된 동일한 장면의 2 개의 픽처들은 3 차원 효과를 생성하기 위하여 이용될 수 있다. 일부의 예들에서, 비디오 소스 (18) (또는 소스 디바이스 (12) 의 또 다른 컴포넌트) 는 소정의 시간 인스턴스에서의 제 1 뷰의 제 1 픽처로부터, 소정의 시간 인스턴스에서의 제 2 (또는 다른 추가적인) 뷰의 제 2 (또는 다른 추가적인) 픽처를 생성하기 위하여 심도 정보 또는 디스패리티 정보를 이용할 수도 있다. 이 경우, 액세스 유닛 내의 뷰는 제 1 뷰에 대응하는 텍스처 컴포넌트와, 제 2 뷰를 생성하기 위하여 텍스처 컴포넌트와 함께 이용될 수 있는 심도 컴포넌트를 포함할 수도 있다. 심도 또는 디스패리티 정보는 예컨대, 카메라 파라미터들과, 비디오 캡처 디바이스의 구성 및 제 1 뷰에 대한 비디오 데이터의 캡처에 관해 알려진 다른 정보에 기초하여 제 1 뷰를 캡처하는 비디오 캡처 디바이스에 의해 결정될 수도 있다. 심도 또는 디스패리티 정보는 추가적으로 또는 대안적으로, 카메라 파라미터들 및/또는 제 1 뷰에서의 비디오 데이터로부터, 예컨대, 비디오 소스 (18) 또는 소스 디바이스 (12) 의 또 다른 컴포넌트에 의해 계산될 수도 있다.

[0070] 3D 비디오를 제시하기 위하여, 디스플레이 디바이스 (32) 는 동시에 또는 거의 동시에 캡처되었던 공통 장면의 상이한 뷰들과 연관된 2 개의 픽처들을 동시에 또는 거의 동시에 디스플레이할 수도 있다. 일부의 예들에서, 목적지 디바이스 (14) 의 사용자는 좌측 및 우측 렌즈들을 신속하게 그리고 교대로 셔터동작을 하기 위한 능동형 안경 (active glasses) 을 착용할 수도 있고, 디스플레이 디바이스 (32) 는 능동형 안경과 동기되어 좌측 뷰와 우측 뷰 사이에서 신속하게 전환할 수도 있다. 다른 예들에서, 디스플레이 디바이스 (32) 는 2 개의 뷰들을 동시에 디스플레이할 수도 있고, 사용자는 적당한 뷰들이 사용자의 눈들을 통과하게 하기 위하여 뷰들을 필터링하는 예컨대, 편광된 렌즈를 갖는 수동형 안경을 착용할 수도 있다. 다른 예들에서, 디스플레이 디바이스 (32) 는 사용자가 3D 효과를 인지하기 위한 안경을 요구하지 않는 자동입체영상 디스플레이 (autostereoscopic display) 를 포함할 수도 있다.

[0071] 멀티뷰 비디오 코딩은 복수의 뷰들이 코딩되는 방식을 지칭한다. 3D 비디오 코딩의 경우, 복수의 뷰들은 예를 들어, 좌안 뷰 (left-eye view) 및 우안 뷰 (right-eye view) 에 대응할 수도 있다. 복수의 뷰들의 각각의 뷰는 복수의 픽처들을 포함한다. 3D 장면의 뷰어 (viewer) 의 인지는 상이한 뷰들의 픽처들에서의 물체들 사이의 수평 디스패리티로 인한 것이다.

[0072] 현재의 픽처의 현재의 블록에 대한 디스패리티 벡터 (disparity vector; DV) 는, 현재의 픽처와는 상이한 뷰에 있는 대응하는 픽처에서의 대응하는 블록을 지시하는 벡터이다. 이에 따라, DV 를 이용하면, 비디오 코더는 대응하는 픽처에서, 현재의 픽처의 현재의 블록에 대응하는 블록을 위치시킬 수 있다. 이 경우, 대응하는 픽처는, 현재의 픽처와 동일한 시간적 인스턴스이지만, 상이한 뷰에 있는 픽처이다. 대응하는 픽처에서의 대응하는 블록 및 현재의 픽처에서의 현재의 블록은 유사한 비디오 컨텐츠를 포함할 수도 있지만; 그러나, 현재

의 픽처에서의 현재의 블록의 로케이션과, 대응하는 픽처에서의 대응하는 블록의 로케이션 사이의 수평 디스페리티가 적어도 있다. 현재의 블록의 DV는 대응하는 픽처에서의 블록과, 현재의 픽처에서의 현재의 블록 사이의 이 수평 디스페리티의 척도 (measure) 를 제공한다.

[0073] 일부의 사례들에서는, 또한, 대응하는 픽처 내의 블록의 로케이션과, 현재의 픽처 내의 현재의 블록의 로케이션 사이의 수직 디스페리티가 있을 수도 있다. 현재의 블록의 DV는 또한, 대응하는 픽처에서의 블록과, 현재의 블록에서의 현재의 블록 사이의 이 수직 디스페리티의 척도를 제공할 수도 있다. DV는 2 개의 컴포넌트들 (x-컴포넌트 및 y-컴포넌트) 을 포함하지만; 많은 사례들에서, 수직 컴포넌트는 제로 (zero) 와 동일할 것이다. 현재의 뷰의 현재의 픽처 및 상이한 뷰의 대응하는 픽처가 디스플레이될 때의 시간은 동일할 수도 있고, 이것은 현재의 픽처 및 대응하는 픽처들이 동일한 시간적 인스턴스의 픽처들인 것을 말하기 위한 것이다.

[0074] 비디오 코딩에서는, 인트라-예측 및 인터 예측으로서 보편적으로 지칭된 2 개의 타입들의 예측이 일반적으로 있다. 인트라-예측에서, 비디오 코더는 동일한 픽처에서의 이미-코딩된 블록에 기초하여 픽처에서의 비디오의 블록을 예측한다. 인터 예측에서는, 비디오 코더가 상이한 픽처 (즉, 참조 픽처) 의 이미-코딩된 블록에 기초하여 픽처에서의 비디오의 블록을 예측한다. 이 개시물에서 이용된 바와 같은 참조 픽처는 일반적으로, 디코딩 순서에서의 후속 픽처들의 디코딩 프로세스에서 인터 예측을 위하여 이용될 수도 있는 샘플들을 포함하는 임의의 픽처를 지칭한다. 현재의 픽처에 관하여, 예컨대, 3D-HEVC 에 따라 멀티뷰 컨텐츠를 코딩할 때, 참조 픽처들은 동일한 시간적 인스턴스이지만 상이한 뷰에 있을 수도 있거나, 동일한 뷰 내에 있지만 상이한 시간적 인스턴스일 수도 있다. 예컨대, 3D-HEVC 에서의 멀티뷰 코딩의 경우, 인터-픽처 예측은 현재의 픽처와 동일한 액세스 유닛 내에 있지만, 현재의 픽처와는 상이한 뷰와 연관된 상이한 픽처로부터의 예측뿐만 아니라, 시간적으로 상이한 픽처에서의 또 다른 비디오 블록으로부터, 즉, 현재의 픽처와는 상이한 액세스 유닛으로부터의 현재의 비디오 블록, 예컨대, CU 의 현재의 코딩 노드의 예측을 포함할 수도 있다.

[0075] 인터 예측의 후자의 경우, 그것은 인터-뷰 코딩 또는 인터-뷰 예측으로서 지칭될 수 있다. 현재의 픽처와 동일한 액세스 유닛 내에 있지만, 현재의 픽처와는 상이한 뷰와 연관되는 참조 픽처는 인터-뷰 참조 픽처로서 지칭될 수도 있다. 멀티뷰 코딩에서, 인터-뷰 예측은 뷰들 사이의 상관을 제거하기 위하여 동일한 액세스 유닛의 (즉, 동일한 시간 인스턴스를 갖는) 상이한 뷰들에서 캡처된 픽처들 사이에서 수행된다. 비-기본 뷰, 예컨대, 종속 뷰의 픽처를 코딩할 때, 동일한 액세스 유닛이지만 상이한 뷰로부터, 예컨대, 기본 뷰와 같은 참조 뷰로부터의 픽처는 참조 픽처 리스트 내에 추가될 수도 있다. 인터-뷰 참조 픽처는 임의의 인터 예측 (예컨대, 시간적 또는 인터-뷰) 참조 픽처에서와 마찬가지로, 참조 픽처 리스트의 임의의 위치에 놓여질 수 있다.

[0076] 현재의 픽처의 블록을 예측하기 위하여 이용된 참조 픽처의 블록은 모션 벡터에 의해 식별된다. 멀티뷰 코딩에서는, 적어도 2 개의 종류들의 모션 벡터들이 있다. 시간적 모션 벡터 (temporal motion vector; TMV) 는, 코딩되고 있는 블록 (예컨대, 위에서 설명된 바와 같은 인터 예측의 제 1 예) 과 동일한 뷰에 있지만, 코딩되고 있는 블록과는 상이한 시간 인스턴스 또는 액세스 유닛인 시간적 참조 픽처에서의 블록을 지시하는 모션 벡터이고, 대응하는 인터 예측은 모션-보상된 예측 (motion-compensated prediction; MCP) 으로서 지칭된다. 또 다른 타입의 모션 벡터는, 현재의 픽처와 동일한 액세스 유닛에 있지만, 상이한 뷰인 블록을 지시하는 디스페리티 모션 벡터 (disparity motion vector; DMV) 이다. DMV 에 있어서, 대응하는 인터 예측은 디스페리티-보상된 예측 (disparity-compensated prediction; DCP) 또는 인터-뷰 예측으로서 지칭된다.

[0077] 다음 섹션에서는, (예컨대, H.264/MVC 에서와 같은) 멀티뷰 및 (예컨대, 3D-HEVC 에서와 같은) 멀티뷰 플러스 심도 코딩 기법들이 논의될 것이다. 먼저, MVC 기법들이 논의될 것이다. 위에서 언급된 바와 같이, MVC 는 ITU-T H.264/AVC 의 멀티뷰 코딩 확장이다. MVC 에서, 복수의 뷰들에 대한 데이터는 시간-우선 순서 (time-first order) 로 코딩되고, 따라서, 디코딩 순서 배열은 시간-우선 코딩으로서 지칭된다. 특히, 공통의 시간 인스턴스에서의 복수의 뷰들의 각각에 대한 뷰 컴포넌트들 (즉, 픽처들) 이 코딩될 수도 있고, 그 다음으로, 상이한 시간 인스턴스에 대한 뷰 컴포넌트들의 또 다른 세트가 코딩될 수도 있는 등등과 같다. 액세스 유닛은 하나의 출력 시간 인스턴스에 대한 뷰들의 전부의 코딩된 픽처들을 포함할 수도 있다. 액세스 유닛들의 디코딩 순서는 출력 (또는 디스플레이) 순서와 반드시 동일한 것은 아니라는 것을 이해해야 한다.

[0078] 전형적인 MVC 디코딩 순서 (즉, 비트스트림 순서) 는 도 2 에서 도시되어 있다. 디코딩 순서 배열은 시간-우선 코딩으로서 지칭된다. 액세스 유닛들의 디코딩 순서는 출력 또는 디스플레이 순서와 동일하지 않을 수도 있다는 것에 주목해야 한다. 도 2 에서, S0 내지 S7 은 각각 멀티뷰 비디오의 상이한 뷰들을 지칭한다. T0 내지 T8 은 각각 하나의 출력 시간 인스턴스 (output time instance) 를 나타낸다. 액세스 유닛은

하나의 출력 시간 인스턴스에 대한 모든 뷰들의 코딩된 픽처들을 포함할 수도 있다. 예를 들어, 제 1 액세스 유닛은 시간 인스턴스 T0 에 대한 뷰들 S0 내지 S7 의 모두를 포함할 수도 있고, 제 2 액세스 유닛은 시간 인스턴스 T1 에 대한 뷰들 S0 내지 S7 의 모두를 포함할 수도 있는 등등과 같다.

[0079] 간결함의 목적들을 위하여, 개시물은 다음의 정의들을 이용할 수도 있다:

[0080] **뷰 컴포넌트:** 단일 액세스 유닛에서의 뷰의 코딩된 표현. 뷰가 코딩된 텍스처 및 심도 표현들의 양자를 포함할 때, 뷰 컴포넌트는 텍스처 뷰 컴포넌트 및 심도 뷰 컴포넌트로 구성된다.

[0081] **텍스처 뷰 컴포넌트:** 단일 액세스 유닛에서의 뷰의 텍스처의 코딩된 표현.

[0082] **심도 뷰 컴포넌트:** 단일 액세스 유닛에서의 뷰의 심도의 코딩된 표현.

[0083] 도 2 에서, 뷰들의 각각은 픽처들의 세트들을 포함한다. 예를 들어, 뷰 S0 는 픽처들 0, 8, 16, 24, 32, 40, 48, 56, 및 64 의 세트를 포함하고, 뷰 S1 은 픽처들 1, 9, 17, 25, 33, 41, 49, 57, 및 65 의 세트를 포함하는 등등과 같다. 3D 비디오 코딩, 예컨대, 3D-HEVC 를 위하여, 각각의 픽처는 2 개의 컴포넌트 픽처들을 포함한다: 하나의 컴포넌트 픽처는 텍스처 뷰 컴포넌트로서 지칭되고, 다른 컴포넌트 픽처는 심도 뷰 컴포넌트로서 지칭된다. 뷰의 픽처들의 세트 내의 텍스처 뷰 컴포넌트 및 심도 뷰 컴포넌트는 서로에 대응하는 것으로서 고려될 수도 있다. 예를 들어, 뷰의 픽처들의 세트 내의 텍스처 뷰 컴포넌트는 뷰의 픽처들의 세트 내의 심도 뷰 컴포넌트에 대응하는 것으로서 고려되고, 그 반대로 마찬가지다 (즉, 심도 뷰 컴포넌트는 세트에서 그 텍스처 뷰 컴포넌트에 대응하고, 그 반대로 마찬가지다). 이 개시물에서 이용된 바와 같이, 심도 뷰 컴포넌트에 대응하는 텍스처 뷰 컴포넌트는 텍스처 뷰 컴포넌트 및 심도 뷰 컴포넌트가 단일 액세스 유닛의 동일한 뷰의 일부인 것으로서 고려될 수도 있다.

[0084] 텍스처 뷰 컴포넌트는 디스플레이되는 실제적인 이미지 컨텐츠를 포함한다. 예를 들어, 텍스처 뷰 컴포넌트는 루마 (Y) 및 크로마 (Cb 및 Cr) 컴포넌트들을 포함할 수도 있다. 심도 뷰 컴포넌트는 그 대응하는 텍스처 뷰 컴포넌트에서의 픽셀들의 상대적인 심도들을 표시할 수도 있다. 하나의 예로서, 심도 뷰 컴포넌트는 루마 값들만을 포함하는 그레이 스케일 이미지 (gray scale image) 이다. 다시 말해서, 심도 뷰 컴포넌트는 임의의 이미지 컨텐츠를 전달할 수도 있는 것이 아니라, 오히려, 텍스처 뷰 컴포넌트에서의 픽셀들의 상대적인 심도들의 척도를 제공할 수도 있다.

[0085] 예를 들어, 심도 뷰 컴포넌트에서의 순수하게 백색인 픽셀은 대응하는 텍스처 뷰 컴포넌트에서의 그 대응하는 픽셀 또는 픽셀들이 뷰어의 관점으로부터 더 근접하다는 것을 표시하고, 심도 뷰 컴포넌트에서의 순수하게 흑색인 픽셀은 대응하는 텍스처 뷰 컴포넌트에서의 그 대응하는 픽셀 또는 픽셀들이 뷰어의 관점으로부터 더욱 멀어져 있다는 것을 표시한다. 흑색과 백색 사이의 회색의 다양한 음영 (shade) 들은 상이한 심도 레벨들을 표시한다. 예를 들어, 심도 뷰 컴포넌트에서의 매우 회색인 픽셀은 텍스처 뷚 컴포넌트에서의 그 대응하는 픽셀이 심도 뷚 컴포넌트에서의 약하게 회색인 픽셀보다 더욱 멀어져 있다는 것을 표시한다. 픽셀들의 심도를 식별하기 위하여 그레이 스케일만이 필요하게 되므로, 심도 뷚 컴포넌트에 대한 컬러 값들이 임의의 목적으로 작용하지 않을 수도 있어서, 심도 뷚 컴포넌트는 크로마 컴포넌트들을 포함할 필요가 없다.

[0086] 심도를 식별하기 위하여 루마 값들 (예컨대, 강도 값 (intensity value) 들) 만을 이용하는 심도 뷚 컴포넌트는 예시의 목적들을 위해 제공되고, 제한하는 것으로 고려되지 않아야 한다. 다른 예들에서, 임의의 기법은 텍스처 뷚 컴포넌트에서의 픽셀들의 상대적인 심도들을 표시하기 위하여 사용될 수도 있다.

[0087] 멀티-뷰 비디오 코딩을 위한 (각각의 뷰 내의 인터-픽처 예측 및 인터-뷰 예측의 양자를 포함하는) 전형적인 MVC 예측 구조가 도 3 에서 도시되어 있다. 예측 방향들은 화살표들에 의해 표시되고, 지시-대상 (pointed-to) 객체는 지시-원천 (pointed-from) 객체를 예측 참조로서 이용한다. MVC 에서, 인터-뷰 예측은, H.264/AVC 모션 보상의 신택스를 이용하지만, 상이한 뷰에서의 픽처가 참조 픽처로서 이용되는 것을 허용하는 디스패리티 모션 보상에 의해 지원된다.

[0088] 도 3 의 예에서, (뷰 ID 들 "S0" 내지 "S7" 을 가지는) 8 개의 뷰들이 예시되어 있고, 12 개의 시간적 로케이션 들 ("T0" 내지 "T11") 이 각각의 뷰에 대해 예시되어 있다. 즉, 도 3 에서의 각각의 행 (row) 은 뷰에 대응하는 반면, 각각의 열 (column) 은 시간적 로케이션을 표시한다.

[0089] MVC 는 H.264/AVC 디코더들에 의해 디코딩 가능한 소위 기본 뷰를 가지고 스테레오 뷰 쌍 (stereo view pair) 들은 MVC 에 의해 또한 지원될 수 있지만, MVC 의 장점은, 그것이 3D 비디오 입력으로서 2 개를 초과하는 뷰들을 이용하여 다수의 뷰들에 의해 표현된 이 3D 비디오를 디코딩하는 예를 지원할 수 있다는 점이다. MVC 디코

더를 가지는 클라이언트의 렌더러 (renderer) 는 다수의 뷰들을 갖는 3D 비디오 컨텐츠를 예상할 수도 있다.

[0090] 도 3 에서의 픽처들은 각각의 행 및 각각의 열의 교차점에서 표시된다. H.264/AVC 표준은 비디오의 부분을 나타내기 위하여 용어 프레임을 이용할 수도 있다. 이 개시물은 용어 픽처 및 프레임을 상호 교환가능하게 이용할 수도 있다.

[0091] 도 3 에서의 픽처들은, 대응하는 픽처가 인트라-코딩되는지 (즉, I-픽처), 또는 하나의 방향에서 (즉, P-픽처로서) 또는 다수의 방향들에서 (즉, B-픽처로서) 인터-코딩되는지 여부를 지정하는 글자를 포함하는 블록을 이용하여 예시된다. 일반적으로, 예측들은 화살표들에 의해 표시되고, 여기서, 지시-대상 픽처들은 예측 참조를 위한 지시-원천 픽처를 이용한다. 예를 들어, 시간적 로케이션 T0 에서의 뷰 S2 의 P-픽처는 시간적 로케이션 T0 에서의 뷰 S0 의 I-픽처로부터 예측된다.

[0092] 단일 뷰 비디오 인코딩에서와 같이, 멀티뷰 비디오 코딩 비디오 시퀀스의 픽처들은 상이한 시간적 로케이션들에서의 픽처들에 대하여 예측방식으로 인코딩될 수도 있다. 예를 들어, 시간적 로케이션 T1 에서의 뷰 S0 의 b-픽처는, b-픽처가 I-픽처로부터 예측되는 것을 표시하는, 시간적 로케이션 T0 에서의 뷰 S0 의 I-픽처로부터 그것으로 지시된 화살표를 가진다. 그러나, 추가적으로, 멀티뷰 비디오 인코딩의 맥락에서는, 픽처들이 인터-뷰 예측될 수도 있다. 즉, 뷰 컴포넌트는 참조를 위한 다른 뷰들에서의 뷰 컴포넌트들을 이용할 수 있다. MVC 에서는, 예를 들어, 인터-뷰 예측이 또 다른 뷰에서의 뷰 컴포넌트가 인터-예측 참조인 것처럼 실현된다. 잠재적인 인터-뷰 참조들은 시퀀스 파라미터 세트 (Sequence Parameter Set; SPS) MVC 확장에서 시그널링되고, 인터-예측 또는 인터-뷰 예측 참조들의 신축적 순서화 (flexible ordering) 를 가능하게 하는 참조 픽처 리스트 구성 프로세스에 의해 수정될 수 있다. 인터-뷰 예측은 또한, 3D-HEVC (멀티뷰 플러스 심도) 를 포함하는 HEVC 의 제안된 멀티뷰 확장의 특징이다.

[0093] 도 3 은 인터-뷰 예측의 다양한 예들을 제공한다. 도 3 의 예에서, 뷰 S1 의 픽처들은 동일한 시간적 로케이션들에서의 뷰들 S0 및 S2 의 픽처들로부터 인터-뷰 예측될 뿐만 아니라, 뷰 S1 의 상이한 시간적 로케이션들에서의 픽처들로부터 예측되는 것으로서 예시된다. 예를 들어, 시간적 로케이션 T1 에서의 뷰 S1 의 b-픽처는 시간적 로케이션 T1 에서의 뷰들 S0 및 S2 의 b-픽처들뿐만 아니라, 시간적 로케이션들 T0 및 T2 에서의 뷰 S1 의 B-픽처들의 각각으로부터 예측된다.

[0094] 일부의 예들에서, 도 3 은 텍스처 뷰 컴포넌트들을 예시하는 것으로서 도시될 수도 있다. 예를 들어, 도 2 에서 예시된 I-픽처, P-픽처, B-픽처, 및 b-픽처들은 뷰들의 각각에 대한 텍스처 뷰 컴포넌트들로서 고려될 수도 있다. 이 개시물에서 설명된 기법들에 따르면, 도 3 에서 예시된 텍스처 뷰 컴포넌트들의 각각에 대하여, 대응하는 심도 뷰 컴포넌트가 있다. 일부의 예들에서, 심도 뷰 컴포넌트들은 대응하는 텍스처 뷰 컴포넌트들에 대해 도 3 에서 예시된 것과 유사한 방식으로 예측될 수도 있다.

[0095] 2 개의 뷰들의 코딩은 또한, MVC 에서 지원될 수 있다. MVC 의 장점들 중의 하나는, MVC 인코더가 2 개를 초과하는 뷰들을 3D 비디오 입력으로서 취할 수 있고 MVC 디코더가 이러한 멀티뷰 표현을 디코딩할 수 있다는 점이다. 이와 같이, MVC 디코더를 갖는 임의의 렌더러는 2 개를 초과하는 뷰들을 갖는 3D 비디오 컨텐츠를 예상할 수도 있다.

[0096] MVC 에서는, 인터-뷰 예측이 (즉, 동일한 시간 인스턴스를 갖는) 동일한 액세스 유닛에서의 픽처들 사이에서 허용된다. 비-기본 뷰들 중의 하나에서의 픽처를 코딩할 때, 픽처가 상이한 뷰에 있지만 동일한 시간 인스턴스 내에 있을 경우, 픽처는 참조 픽처 리스트 새로 추가될 수도 있다. 인터-뷰 참조 픽처는 임의의 인터 예측 참조 픽처처럼, 참조 픽처 리스트의 임의의 위치에 놓여질 수 있다. 도 3 에서 도시된 바와 같이, 뷰 컴포넌트는 참조를 위한 다른 뷰들에서의 뷰 컴포넌트들을 이용할 수 있다. MVC 에서는, 인터-뷰 예측이 또 다른 뷰에서의 뷰 컴포넌트가 인터-예측 참조였던 것처럼 실현된다.

[0097] 멀티뷰 비디오 코딩의 맥락에서는, 2 개의 종류들의 모션 벡터들이 있다. 하나는 정상적인 모션 벡터로서 지칭된다. 정상적인 모션 벡터는 시간적 참조 픽처들을 지시하고, 대응하는 시간적 인터 예측은 모션-보상된 예측 (MCP) 이다. 다른 모션 벡터는 디스패리티 모션 벡터 (DMV) 이다. DMV 는 상이한 뷰에서의 픽처들 (즉, 인터-뷰 참조 픽처들) 을 지시하고, 대응하는 인터 예측은 디스패리티-보상된 예측 (DCP) 이다.

[0098] 또 다른 타입의 멀티뷰 비디오 코딩 포맷은 (예컨대, 3D-HEVC 에서와 같은) 심도 맵들의 이용을 도입한다. 3D 텔리비전 및 자유로운 뷰포인트 비디오들을 위해 인기있는 멀티뷰-비디오-플러스-심도 (multiview-video-plus-depth; MVD) 데이터 포맷에 대하여, 텍스처 이미지들 및 심도 맵들은 멀티뷰 텍스처 픽처들로 독립적으로 코딩될 수 있다. 도 4 는 텍스처 이미지 및 그 연관된 샘플당 심도 맵을 갖는 MVD 데이터 포맷을 예시한다.

심도 범위는 대응하는 3D 포인트들에 대한 카메라로부터 최소 znear 및 최대 zfar 거리의 범위에 있는 것으로 한정될 수도 있다.

[0099] HEVC 에서, 모션 벡터 예측을 위한 기법들은 병합 모드, 스킵 모드, 및 진보된 모션 벡터 예측 (AMVP) 모드를 포함할 수도 있다. 일반적으로, 병합 모드 및/또는 스kip 모드에 따르면, 현재의 비디오 블록 (예컨대, PU) 은 또 다른 이전에 코딩된 이웃하는 블록, 예컨대, 동일한 픽처에서의 공간적으로 이웃하는 블록, 또는 시간적 또는 인터뷰 참조 픽처에서의 블록으로부터, 모션 정보, 예컨대, 모션 벡터, 예측 방향, 및 참조 픽처 인덱스를 승계한다. 병합/스킵 모드를 구현할 때, 비디오 인코더 (20) 는 정의된 사안에서 참조 블록들의 모션 정보인 병합 후보들의 리스트를 구성하고, 병합 후보들 중의 하나를 선택하고, 선택된 병합 후보를 식별하는 후보 리스트 인덱스를 비트스트림에서 비디오 디코더 (30) 로 시그널링한다.

[0100] 병합/스킵 모드를 구현할 시에, 비디오 디코더 (30) 는 정의된 방식에 따라 병합 후보 리스트를 재구성하고, 인덱스에 의해 표시된 후보 리스트에서 병합 후보들 중의 하나를 선택한다. 다음으로, 비디오 디코더 (30) 는 병합 후보들 중의 선택된 하나를, 병합 후보들 중의 선택된 하나의 모션 벡터와 동일한 해상도에서, 그리고 병합 후보들 중의 선택된 하나에 대한 모션 벡터와 동일한 참조 픽처를 지시하는, 현재의 PU 에 대한 모션 벡터로서 이용할 수도 있다. 병합 모드 및 스kip 모드는 비디오 인코더 (20) 가 현재의 비디오 블록의 인터-예측을 위한 모션 정보의 전부가 아니라, 인덱스를 병합 후보 리스트로 시그널링하는 것을 허용함으로써 비트스트림 효율을 촉진시킨다.

[0101] AMVP 를 구현할 때, 비디오 인코더 (20) 는 정의된 사안에서 후보 모션 벡터 예측자 (MVP) 들의 리스트를 구성하고, 후보 MVP 들 중의 하나를 선택하고, 선택된 MVP 를 식별하는 후보 리스트 인덱스를 비트스트림에서 비디오 디코더 (30) 로 시그널링한다. 병합 모드와 유사하게, AMVP 를 구현할 때, 비디오 디코더 (30) 는 정의된 사안에서 후보 MVP 들의 리스트를 재구성하고, 후보 리스트 인덱스에 기초하여 MVP 들 중의 하나를 선택한다.

[0102] 그러나, 병합/스킵 모드와 반대로, AMVP 를 구현할 때, 비디오 인코더 (20) 는 또한, 참조 픽처 인덱스 및 예측 방향을 시그널링하고, 이에 따라, 후보 리스트 인덱스에 의해 특정된 MVP 가 지시하는 참조 픽처를 특정한다. 또한, 비디오 인코더 (20) 는 현재의 블록에 대한 모션 벡터 차이 (MVD) 를 결정하고, 여기서, MVD 는 MVP 와, 이와 다르게 현재의 블록에 대해 이용될 실제적인 모션 벡터 사이의 차이이다. AMVP 에 대하여, 참조 픽처 인덱스, 참조 픽처 방향, 및 후보 리스트 인덱스에 추가하여, 비디오 인코더 (20) 는 비트스트림에서 현재의 블록에 대한 MVD 를 시그널링한다. 소정의 블록에 대한 참조 픽처 인덱스 및 예측 벡터 차이의 시그널링으로 인해, AMVP 는 병합/스킵 모드만큼 효율적이지 않을 수도 있지만, 코딩된 비디오 데이터의 개선된 충실도 (fidelity) 를 제공할 수도 있다.

[0103] 도 5 는 현재의 비디오 블록 (47), 5 개의 공간적으로 이웃하는 블록들 (41, 42, 43, 44, 및 45), 및 또 다른 픽처로부터, 그러나 현재의 픽처와 동일한 뷰에서의 시간적 참조 블록 (46) 의 예를 도시한다. 시간적 참조 블록 (46) 은 예를 들어, 현재의 비디오 블록 (47) 과는 상이한 시간적 인스턴스이지만 동일한 뷰인 픽처에서의 공동-위치된 블록일 수도 있다. 일부의 예들에서, 현재의 비디오 블록 (47) 및 참조 비디오 블록들 (41 내지 46) 은 일반적으로 현재 개발 중인 HEVC 표준에서 정의될 수도 있다. 참조 비디오 블록들 (41 내지 46) 은 HEVC 표준에 따라 A0, A1, B0, B1, B2, 및 T 로 라벨표시된다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 모션 정보 예측 모드, 예컨대, 병합/스킵 모드 또는 AMVP 모드에 따라, 참조 비디오 블록들 (41 내지 46) 의 모션 정보에 기초하여 현재의 비디오 블록 (47) 의, TMV 를 포함하는 모션 정보를 예측할 수도 있다. 이하에서 더욱 상세하게 설명된 바와 같이, 비디오 블록들의 TMV 들은 이 개시물의 기법들에 따라 진보된 잔차 예측을 위한 DMV 들과 함께 이용될 수도 있다.

[0104] 도 5 에서 예시된 바와 같이, 비디오 블록들 (42, 44, 43, 41, 및 45) 은 현재의 비디오 블록 (47)에 관하여, 각각 좌측, 상부, 상부-우측, 하부-좌측, 및 상부-좌측일 수도 있다. 그러나, 도 5 에서 예시된 현재의 비디오 블록 (47) 에 관한 이웃하는 블록들 (41 내지 45) 의 수 및 로케이션들은 예들에 불과하다. 상이한 수의 이웃하는 블록들 및/또는 상이한 로케이션들에서의 블록들의 모션 정보는 현재의 비디오 블록 (47) 에 대한 모션 정보 예측 후보 리스트에서의 포함을 위하여 고려될 수도 있다.

[0105] 현재의 비디오 블록 (47) 에 대한 공간적으로 이웃하는 블록들 (42, 44, 43, 41, 및 45) 의 각각의 공간적 관계는 다음과 같이 설명될 수도 있다. 루마 로케이션 ( $xP$ ,  $yP$ ) 은 현재의 픽처의 상부-좌측 샘플에 관하여 현재의 블록의 상부-좌측 루마 샘플을 특정하기 위하여 이용된다. 변수들 nPSW 및 nPSH 는 루마에 대한 현재의 블록의 폭 및 높이를 나타낸다. 공간적으로 이웃하는 블록 (42) 의 상부-좌측 루마 샘플은  $xP - 1$ ,  $yP +$

$nPSH - 1$  이다. 공간적으로 이웃하는 블록 (44)의 상부-좌측 루마 샘플은  $xP + nPSW - 1$ ,  $yP - 1$  이다. 공간적으로 이웃하는 블록 (43)의 상부-좌측 루마 샘플은  $xP + nPSW$ ,  $yP - 1$  이다. 공간적으로 이웃하는 블록 (41)의 상부-좌측 루마 샘플은  $xP - 1$ ,  $yP + nPSH$  이다. 공간적으로 이웃하는 블록 (45)의 상부-좌측 루마 샘플은  $xP - 1$ ,  $yP - 1$  이다. 루마 로케이션들에 대하여 설명되었지만, 현재 및 참조 블록들은 크로마 컴포넌트들을 포함할 수도 있다.

[0106]

공간적으로 이웃하는 블록들 (41 내지 45)의 각각은 현재의 비디오 블록 (47)의 모션 정보, 예컨대, TMV를 예측하기 위한 공간적 모션 정보 후보를 제공할 수도 있다. 비디오 코더, 예컨대, 비디오 인코더 (20)(도 1) 및/또는 비디오 디코더 (30)(도 1)는 미리 결정된 순서, 예컨대, 스캔 순서에서 공간적으로 이웃하는 참조 블록들의 모션 정보를 고려할 수도 있다. 예를 들어, 3D-HEVC의 경우, 비디오 디코더는 다음의 순서에서의 병합 모드에 대한 병합 후보 리스트에서의 포함을 위한 참조 블록들의 모션 정보를 고려할 수도 있다: 42, 44, 43, 41, 및 45. 예시된 예에서, 공간적으로 이웃하는 블록들 (41 내지 45)은 현재의 비디오 블록 (47)의 좌측 및/또는 상부이다. 대부분의 비디오 코더들이 비디오 블록들을 래스터 스캔 (raster scan) 순서로 픽처의 상부-좌측으로부터 코딩하므로, 이 배열은 전형적이다. 따라서, 이러한 예들에서, 공간적으로 이웃하는 블록들 (41 내지 45)은 전형적으로 현재의 비디오 블록 (47) 이전에 코딩될 것이다. 그러나, 다른 예들에서, 예컨대, 비디오 코더가 비디오 블록들을 상이한 순서로 코딩할 때, 공간적으로 이웃하는 블록들 (41 내지 45)은 현재의 비디오 블록 (47)의 우측 및/또는 하부에 위치될 수도 있다.

[0107]

시간적 참조 블록 (46)은 코딩 순서에서 반드시 바로 이전은 아니지만, 현재의 비디오 블록 (47)의 현재의 픽처 이전에 코딩된 시간적 참조 픽처 내에 위치된다. 추가적으로, 블록 (46)의 참조 픽처는 반드시, 디스플레이 순서에서 현재의 비디오 블록 (47)의 픽처 이전은 아니다. 참조 비디오 블록 (46)은 일반적으로, 현재의 픽처에서의 현재의 비디오 블록 (47)의 로케이션에 관하여 참조 픽처에서 공동-위치될 수도 있다. 일부의 예들에서, 참조 비디오 블록 (46)은 현재의 픽처에서의 현재의 비디오 블록 (47)의 위치의 우측 및 하부에 위치되거나, 현재의 픽처에서 현재의 비디오 블록 (47)의 중심 위치를 커버한다.

[0108]

도 6은 예컨대, 병합/스킵 모드 또는 AMVP 모드에 따라, 현재의 비디오 블록 (50)의 모션 정보의 예측을 위한 인터-뷰 예측된 모션 벡터 후보 (inter-view predicted motion vector candidate; IPMVC) 및 인터-뷰 디스파리티 모션 벡터 후보 (inter-view disparity motion vector candidate; IDMVC)의 유도의 예를 예시하는 개념도이다. 인터-뷰 예측이 인에이블될 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 새로운 모션 벡터 후보, IPMVC, 또는 IDMVC를 현재의 비디오 블록 (50)에 대한 모션 정보 후보 리스트에 추가할 수도 있다.

IPMVC는 이하에서 더욱 상세하게 설명된 바와 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)가 이 개시물의 기법들에 따라 현재의 비디오 블록 (50) 또는 또 다른 비디오 블록의 ARP를 위하여 이용할 수도 있는 현재의 비디오 블록 (50)에 대한 TMV를 예측할 수도 있다. IDMVC는, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)가 현재의 비디오 블록 (50)의 ARP를 위하여 이용할 수도 있는 현재의 비디오 블록 (50)에 대한 DMV를 예측할 수도 있다.

[0109]

도 6의 예에서, 현재의 블록 (50)은 현재의 뷰  $V_m$  내에 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 디스파리티 벡터 (DV)(51)를 이용하여 참조 뷰  $V_0$ 에서 대응 또는 참조 블록 (52)을 위치시킬 수도 있다. 비디오 코더는 카메라 파라미터들에 기초하여, 또는 본원에서 설명된 기법들 중의 임의의 것에 따라 DV (51)를 결정할 수도 있다. 예를 들어, 비디오 코더는 이웃하는 블록의 DMV 또는 DV에 기초하는 것을 포함하여, 예컨대, 이웃하는 블록 기반 디스파리티 벡터 (NBDV) 유도를 이용하여, 현재의 비디오 블록 (50)에 대한 DV (51)를 결정할 수도 있다.

[0110]

참조 블록 (52)이 인트라-코딩되지 않고 인터-뷰 예측되지 않고, 그 참조 픽처, 예컨대, 참조 픽처 (58) 또는 참조 픽처 (60)가 현재의 비디오 블록 (50)의 동일한 참조 픽처 리스트에서의 하나의 엔트리의 것과 동일한 픽처 순서 카운트 (POC) 값을 가질 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 POC에 기초한 참조 인덱스를 현재의 비디오 블록 (50)에 대한 IPMVC가 되도록 변환한 후에, 그 모션 정보 (예측 방향, 참조 픽처들, 및 모션 벡터들)를 유도할 수도 있다.

[0111]

도 6의 예에서, 참조 비디오 블록 (52)은 제 1 참조 픽처 리스트 (RefPicList0)에서 특정되는 참조 뷰  $V_0$ 에서의 제 1 참조 픽처 (58)를 지시하는 TMV (54)와, 제 2 참조 픽처 리스트 (RefPicList1)에서 특정되는 참조 뷰  $V_0$ 에서의 제 2 픽처 (60)를 지시하는 TMV (56)와 연관된다. 현재의 비디오 블록 (50)에 의한 TMV들 (54 및 56)의 승계는 도 6에서의 점선 화살표들에 의해 예시된다. 참조 비디오 블록 (52)의 모션 정보에 기초하여, 비디오 코더는 현재의 비디오 블록 (50)에 대한 IPMVC를, 예컨대, 제 1 참조 픽처 리스트에

서 참조 픽처 (58) 와 동일한 POC 를 갖는, 제 1 참조 픽처 리스트 (RefPicList0) 에서 특정되는 현재의 뷰  $V_m$  에서의 제 1 참조 픽처 (66) 를 지시하는 TMV (62) 와, 예컨대, 참조 픽처 (60) 와 동일한 POC 를 갖는, 제 2 참조 픽처 리스트 (RefPicList1) 에서 특정되는 현재의 뷰  $V_m$  에서의 제 2 픽처 (68) 를 지시하는 TMV (64) 중의 적어도 하나인 것으로 유도한다.

[0112] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (50) 의 ARP 를 위하여 TMV (62) 및/또는 TMV (64) 를 이용할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 또한, DV (51) 를 현재의 비디오 블록 (50) 에 대한 IDMVC 로 변환할 수도 있고, IDMVC 를, IPMVC 와는 상이한 위치에서 현재의 비디오 블록 (50) 에 대한 모션 정보 후보 리스트에 추가한다. IPMVC 또는 IDMVC 의 각각은 이 맥락에서 '인터-뷰 후보' 로서 지칭될 수도 있다.

[0113] 병합/스킵 모드에서, 비디오 코더는 IPMVC 를 이용가능할 경우, 병합 후보 리스트에 대한 모든 공간적 및 시간적 병합 후보들 이전에 삽입한다. 병합/스킵 모드에서, 비디오 코더는 IDMVC 를, AO (도 5 의 블록 (41)) 으로부터 유도된 공간적 병합 후보 이전에 삽입한다. IDMVC 로의 DV (51) 의 변환은 현재의 비디오 블록 (50) 에 대한 DMV 로의 DV (51) 의 변환으로 고려될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (50) 의 ARP 를 위하여 DMV 를 이용할 수도 있다.

[0114] 일부의 상황들에서, 비디오 코더는 현재의 비디오 블록에 대한 DV 를 유도할 수도 있다. 예를 들어, 도 6 을 참조하여 위에서 설명된 바와 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (50) 에 대한 DV (51) 를 유도할 수도 있다. 일부의 예들에서, 비디오 코더는 현재의 비디오 블록에 대한 DV 를 유도하기 위하여 NBDV 유도를 이용할 수도 있다. NBDV 유도는 3D-HEVC 에서의 디스패리티 벡터 유도 방법으로서 이용된다.

[0115] 3D-HEVC 에 대한 제안들은 뷰들의 전부에 대한 텍스처-우선 코딩 순서를 이용한다. 다시 말해서, 비트스트림에서의 복수의 뷰들의 각각에 대하여, 텍스처 컴포넌트는 뷰에 대한 임의의 심도 컴포넌트 이전에 코딩, 예컨대, 인코딩 또는 디코딩된다. 일부의 경우에는, 예컨대, 인터-뷰 예측을 위하여, DV 는 특정한 액세스 유닛에서의 뷰의 텍스처 컴포넌트에서 비디오 블록을 코딩하기 위하여 필요하게 된다. 그러나, 텍스처-우선 코딩에서는, 현재의 비디오 블록의 대응하는 심도 컴포넌트가 현재의 비디오 블록에 대한 DV 를 결정하기 위하여 이용가능하지 않다. NBDV 유도는 비디오 코더에 의해 채용될 수도 있고, 이러한 상황들에서 현재의 비디오 블록에 대한 DV 를 유도하기 위하여 3D-HEVC 에 대해 제안된다. 현재의 3D-HEVC 설계에서, NBDV 유도로부터 유도된 DV 는 NBDV 프로세스로부터의 DV 에 의해 지시된 참조 뷰의 심도 맵으로부터 심도 데이터를 취출함으로써 추가로 세분화될 수 있다.

[0116] DV 는 2 개의 뷰들 사이의 변위 (displacement) 의 추정자 (estimator) 를 위하여 이용된다. 이웃하는 블록들이 비디오 코딩에서 거의 동일한 모션/디스패리티 정보를 공유하므로, 현재의 비디오 블록은 이웃하는 블록들에서의 모션 벡터 정보를 그 모션/디스패리티 정보의 양호한 예측자로서 이용할 수 있다. 이 사상에 후속하여, NBDV 유도는 상이한 뷰들에서 DV 들을 추정하기 위한 이웃하는 디스패리티 정보를 이용한다.

[0117] NBDV 유도에 따르면, 비디오 코더는 몇몇 공간적 및 시간적으로 이웃하는 블록들을 식별한다. 이웃하는 블록들의 2 개의 세트들이 사용된다. 하나의 세트는 공간적으로 이웃하는 블록들로부터 나온 것이고, 다른 세트는 시간적으로 이웃하는 블록들로부터 나온 것이다. 다음으로, 비디오 코더는 현재의 블록과 후보 (이웃하는) 블록 사이의 상관 (correlation) 의 우선순위에 의해 결정된 미리 정의된 순서로 공간적 및 시간적으로 이웃하는 블록들의 각각을 검사한다. 비디오 코더가 DMV, 즉, 후보들의 모션 정보에서, 이웃하는 후보 블록으로부터 (동일한 액세스 유닛이지만, 상이한 뷰에서의) 인터-뷰 참조 픽처로 지시하는 모션 벡터를 식별할 때, 비디오 코더는 DMV 를 DV 로 변환하고, 연관된 뷰 순서 인덱스를 반환한다. 예를 들어, 비디오 코더는 현재의 블록에 대한 DV 의 수평 컴포넌트를 DMV 의 수평 컴포넌트와 동일하게 설정할 수도 있고, DV 의 수직 컴포넌트를 0 으로 설정할 수도 있다.

[0118] 3D-HEVC 는 초기에, Zhang 등, "3D-CE5.h: Disparity vector generation results (디스패리티 벡터 생성 결과들)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 비디오 코딩 확장 개발에 관한 합동 협력 팀, 1 차 회의: Stockholm, SE, 2012 년 7 월 16 - 20, 문서 JCT3V-A0097 (MPEG 번호 m26052), 이하 "JCT3V-A0097" 에서 제안된 NBDV 유도 기법들을 채택하였다. JCT3V-A0097 은 다음의 링크로부터 다운로드될 수도 있다: [http://phenix.int-evry.fr/jct2/doc\\_end\\_user/current\\_document.php?id=89](http://phenix.int-evry.fr/jct2/doc_end_user/current_document.php?id=89). JCT3V-A0097 의 전체 내용은 참조로 본원에 편입된다.

- [0119] 3D-HEVC 에 대한 일부의 제안들에서, 비디오 코더가 NBDV 유도 프로세스를 수행할 때, 비디오 코더는 시간적으로 이웃하는 블록들에서의 디스파리티 모션 벡터들, 공간적으로 이웃하는 블록들에서의 디스파리티 모션 벡터들, 그 다음으로, 묵시적 디스파리티 벡터 (implicit disparity vector; IDV) 들을 순서대로 검사한다. IDV 는 인터-뷰 예측을 이용하여 코딩되는 공간적 또는 시간적으로 이웃하는 PU 의 디스파리티 벡터일 수도 있다. IDV 들은 또한, 유도된 디스파리티 벡터들로서 지정될 수도 있다. IDV 는, PU 가 인터-뷰 예측을 채용할 때에 생성될 수도 있고, 즉, AMVP 또는 병합 모드들에 대한 후보는 디스파리티 벡터의 도움으로 다른 뷰에서의 참조 블록으로부터 유도된다. 이러한 디스파리티 벡터는 IDV 로 칭해진다. IDV 는 DV 유도의 목적을 위하여 PU 에 저장될 수도 있다. 예를 들어, 블록이 모션 예측으로 코딩되더라도, 블록에 대한 유도된 DV 는 다음의 비디오 블록을 코딩할 목적으로 폐기되지 않는다. 이에 따라, 비디오 코더가 DMV 또는 IDV 를 식별할 때, 비디오 코더는 식별된 DMV 또는 IDV 를 반환할 수도 있다.
- [0120] 묵시적 디스파리티 벡터 (IDV) 들은, Sung 등, "3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding (HEVC-기반 3D 비디오 코딩을 위한 디스파리티 벡터 유도의 간략화)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장 개발에 관한 합동 협력 팀, 1 차 회의: Stockholm, SE, 2012 년 7 월 16 - 20 일, 문서 JCT3V-A0126 (MPEG 번호 m26079), 이하 "JCT3V-A0126"에서 설명된 간략화된 NBDV 유도 프로세스와 함께 포함되었다. JCT3V-A0126 는 다음의 링크로부터 다운로드 될 수도 있다: [http://phenix.int-evry.fr/jct2/doc\\_end\\_user/current\\_document.php?id=142](http://phenix.int-evry.fr/jct2/doc_end_user/current_document.php?id=142).
- [0121] 3D-HEVC 를 위한 NBDV 유도 프로세스의 추가의 개발은 Kang 등, "3D-CE5.h: Improvement for disparity vector derivation (디스파리티 벡터 유도를 위한 개선)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 비디오 코딩 확장 개발에 관한 합동 협력 팀, 2차 회의: Shanghai, CN, 2012 년 10 월 13-19 일, 문서 JCT3V-B0047 (MPEG 번호 m26736), 이하 "JCT3V-B0047"에서 설명되었다. JCT3V-B0047 는 다음의 링크로부터 다운로드 될 수도 있다: [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=236](http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=236).
- [0122] JCT3V-B0047 에서, 3D-HEVC 를 위한 NBDV 유도 프로세스는 디코딩된 픽처 버퍼에서 저장된 IDV 들을 제거함으로써 추가로 간략화되었다. 코딩 이득은 또한, 랜덤 액세스 포인트 (random access point; RAP) 픽처 선택으로 개선되었다. 비디오 코더는 반환된 디스파리티 모션 벡터 또는 IDV 를 디스파리티 벡터로 변환할 수도 있고, 인터-뷰 예측 및 인터-뷰 잔차 예측을 위하여 디스파리티 벡터를 이용할 수도 있다. 랜덤 액세스는 비트스트림에서 최초 코딩된 픽처가 아닌 코딩된 픽처로부터 시작하는 비트스트림의 디코딩을 지칭한다. 규칙적인 간격들에서의 비트스트림으로의 랜덤 액세스 픽처들 또는 랜덤 액세스 포인트들의 삽입은 랜덤 액세스를 가능하게 할 수도 있다. 예시적인 타입들의 랜덤 액세스 픽처들은 순간적 디코더 리프레시 (Instantaneous Decoder Refresh; IDR) 픽처들, 클린 랜덤 액세스 (Clean Random Access; CRA) 픽처들, 및 파손된 링크 액세스 (Broken Link Access; BLA) 픽처들을 포함한다. 이 때문에, IDR 픽처들, CRA 픽처들, 및 BLA 픽처들은 집합적으로 RAP 픽처들로서 지정된다. 일부의 예들에서, RAP 픽처들은 BLA\_W\_LP, BLA\_W\_RADL, BLA\_N\_LP, IDR\_W\_RADL, IDR\_N\_LP, RSV\_IRAP\_VCL22, RSV\_IRAP\_VCL23, 또는 CRA\_NUT 과 동일한 NAL 유닛 타입들을 가질 수도 있다.
- [0123] 3D-HEVC 에 대한 CU-기반 DV 유도를 위한 기법들은 Kang 등, "CE2.h: CU-based disparity vector derivation in 3D-HEVC (3D-HEVC 에서의 CU-기반 디스파리티 벡터 유도)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 비디오 코딩 확장 개발에 관한 합동 협력 팀, 4차 회의: Incheon, KR, 2013 년 4 월 20-26 일, 문서 JCT3V-D0181 (MPEG 번호 m29012), 이하 "JCT3V-D0181"에서 제안되었다. JCT3V-D0181 은 다음의 링크로부터 다운로드 될 수도 있다: [http://phenix.it-sudparis.eu/jct3v/doc\\_end\\_user/current\\_document.php?id=866](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=866).
- [0124] 비디오 코더가 DMV 또는 IDV 를 식별할 때, 비디오 코더는 검사 프로세스를 종결시킬 수도 있다. 이에 따라, 일단 비디오 코더가 현재의 블록에 대한 DV 를 구하면, 비디오 코더는 NBDV 유도 프로세스를 종결시킬 수도 있다. 비디오 코더가 NBDV 유도 프로세스를 수행함으로써 현재의 블록에 대한 DV 를 결정할 수 없을 때 (즉, NBDV 유도 프로세스 동안에 구해진 DMV 또는 IDV 가 없을 때), NBDV 는 이용불가능한 것으로서 표기된다. 다시 말해서, NBDV 유도 프로세스는 이용불가능한 디스파리티 벡터를 반환하는 것으로 고려될 수 있다.
- [0125] 비디오 코더가 NBDV 유도 프로세스를 수행함으로써 현재의 블록에 대한 DV 를 유도할 수 없을 경우 (즉, 디스파리티 벡터가 구해지지 않을 경우), 비디오 코더는 제로 DV 를 현재의 PU 에 대한 DV 로서 이용할 수도 있다. 제로 DV 는 0 과 동일한 수평 및 수직 컴포넌트들의 양자를 가지는 DV 이다. 이에 따라, NBDV 유도 프로세스가 이용불가능한 결과를 반환하더라도, DV 를 요구하는 비디오 코더의 다른 코딩 프로세스들은 현재의 블록에 대한 제로 디스파리티 벡터를 이용할 수도 있다. 일부의 예들에서, 비디오 코더가 NBDV 유도 프로세스를

수행함으로써 현재의 블록에 대한 DV 를 유도할 수 없을 경우, 비디오 코더는 현재의 블록에 대한 인터-뷰 잔차 예측을 디스에이블할 수도 있다. 그러나, 비디오 코더가 NBDV 유도 프로세스를 수행함으로써 현재의 블록에 대한 DV 를 유도할 수 있는지 여부에 관계없이, 비디오 코더는 현재의 블록에 대한 인터-뷰 예측을 이용할 수도 있다. 즉, 모든 미리 정의된 이웃하는 블록들을 검사한 후에 DV 가 구해지지 않을 경우, 제로 디스패리티 벡터는 인터-뷰 예측을 위해 이용될 수도 있는 반면, 인터-뷰 잔차 예측은 대응하는 CU 에 대해 디스에이블될 수도 있다.

[0126] 도 7 은 공간적으로 이웃하는 블록들로부터, 현재의 비디오 블록에 대한 DV 가 현재의 비디오 블록 (90) 에 관하여 NBDV 유도를 이용하여 유도될 수도 있는, 예시적인 공간적으로 이웃하는 블록들을 예시하는 개념도이다.

도 7 에서 예시된 5 개의 공간적으로 이웃하는 블록들은 현재의 비디오 블록에 관하여 하부-좌측 블록 (96), 좌측 블록 (95), 상부-우측 블록 (92), 상부 블록 (93), 및 상부-좌측 블록 (94) 이다. 공간적으로 이웃하는 블록들은 현재의 비디오 블록을 커버하는 CU 의 하부-좌측, 좌측, 상부-우측, 상부, 및 상부-좌측 블록들일 수도 있다. NBDV 에 대한 이 공간적으로 이웃하는 블록들은 예컨대, HEVC 에서의 MERGE/AMVP 모드들에 따른 현재의 비디오 블록에 대한 모션 정보 예측을 위한 비디오 코더에 의해 이용된 공간적으로 이웃하는 블록들과 동일할 수도 있다는 것에 주목해야 한다. 이러한 경우들에는, 공간적으로 이웃하는 블록들의 모션 정보가 현재의 비디오 블록에 대한 모션 정보 예측을 위해 이미 고려되므로, 비디오 코더에 의한 추가적인 메모리 액세스가 NBDV 에 대해 요구되지 않을 수도 있다.

[0127] 시간적으로 이웃하는 블록들을 검사하기 위하여, 비디오 코더는 후보 픽처 리스트를 구성한다. 일부의 예들에서, 비디오 코더는 현재의 뷰, 즉, 현재의 비디오 블록과 동일한 뷰로부터의 2 개까지의 참조 픽처들을 후보 픽처들로서 취급할 수도 있다. 비디오 코더는 먼저, 공동-위치된 참조 픽처를 후보 픽처 리스트 내에 삽입 할 수도 있고, 그 다음으로, 후보 픽처들의 나머지를 참조 픽처 인덱스의 상승 순서로 삽입할 수도 있다. 양자의 참조 픽처 리스트들에서 동일한 참조 인덱스를 갖는 참조 픽처들이 이용가능할 때, 비디오 코더는 공동-위치된 픽처와 동일한 참조 픽처 리스트에서의 하나를, 다른 참조 픽처 리스트로부터의 다른 참조 픽처 이전에 삽입 할 수도 있다. 일부의 예들에서, 비디오 코더는 후보 픽처 리스트에서의 후보 픽처들의 각각으로부터 시간적으로 이웃하는 블록들을 유도하기 위한 3 개의 후보 영역들을 식별할 수도 있다. 3 개의 후보 영역들은 다음과 같이 정의될 수도 있다:

[0128] · CPU: 현재의 PU 또는 현재의 CU 의 공동-위치된 영역.

[0129] · CLCU: 현재의 블록의 공동-위치된 영역을 커버하는 최대 코딩 유닛 (LCU).

[0130] · BR: CPU 의 하부-우측 4x4 블록.

[0131] 후보 영역을 커버하는 PU 가 DMV 를 특정할 경우, 비디오 코더는 PU 의 디스패리티 모션 벡터에 기초하여 현재의 비디오 유닛의 DV 를 결정할 수도 있다.

[0132] 위에서 논의된 바와 같이, 공간적 및 시간적으로 이웃하는 블록들로부터 유도된 DMV 들에 추가하여, 비디오 코더는 IDV 들에 대해 검사할 수도 있다. 3D-HTM 7.0 및 더 이후의 버전들에 대한 제안된 NBDV 유도 프로세스에서, 비디오 코더는 시간적으로 이웃하는 블록들에서의 DMV 들, 그 다음으로 공간적으로 이웃하는 블록들에서의 DMV 들, 및 그 다음으로 IDV 들을 순서대로 검사한다. 일단 DMV 또는 IDV 가 구해지면, 프로세스가 종결 된다. 게다가, NBDV 유도 프로세스에서 검사된 공간적으로 이웃하는 블록들의 수는 2 로 추가로 감소된다.

[0133] 비디오 코더가 이웃하는 PU (즉, 공간적으로 또는 시간적으로 이웃하는 PU) 를 검사할 때, 비디오 코더는 먼저, 이웃하는 PU 가 디스패리티 모션 벡터를 가지는지 여부를 검사할 수도 있다. 이웃하는 PU 들의 어느 것도 디스패리티 모션 벡터를 가지지 않을 경우, 비디오 코더는 공간적으로 이웃하는 PU 들 중의 임의의 것이 IDV 를 가지는지 여부를 결정할 수도 있다. 공간적으로 이웃하는 PU 들 중의 하나가 IDV 를 가지고 IDV 가 병합/스킵 모드로서 코딩될 경우, 비디오 코더는 검사 프로세스를 종결시킬 수도 있고, IDV 를 현재의 PU 에 대한 최종 디스패리티 벡터로서 이용할 수도 있다.

[0134] 위에서 표시된 바와 같이, 비디오 코더는 현재의 블록 (예컨대, CU, PU 등) 에 대한 DV 를 유도하기 위하여 NBDV 유도 프로세스를 적용할 수도 있다. 현재의 블록에 대한 디스패리티 벡터는 참조 뷰에서의 참조 픽처 (즉, 참조 컴포넌트) 에서 로케이션을 표시할 수 있다. 일부의 3D-HEVC 설계들에서, 비디오 코더는 참조 뷰에 대한 심도 정보를 액세스하도록 허용된다. 일부의 이러한 3D-HEVC 설계들에서, 비디오 코더가 현재의 블록에 대한 DV 를 유도하기 위하여 NBDV 유도 프로세스를 이용할 때, 비디오 코더는 현재의 블록에 대한 디스패리티 벡터를 추가로 세분화하기 위하여 세분화 프로세스를 적용할 수도 있다. 비디오 코더는 참조 픽처의

심도 맵에 기초하여 현재의 블록에 대한 DV 를 세분화할 수도 있다. 비디오 코더는 후방 뷰 합성 예측 (backward view synthesis prediction) 을 위한 DMV 를 세분화하기 위하여 유사한 세분화 프로세스를 이용할 수도 있다. 이러한 방법으로, 심도는 후방 뷰 합성 예측을 위하여 이용되어야 할 DV 또는 DMV 를 세분화하기 위하여 이용될 수 있다. 이 세분화 프로세스는 NBDV 세분화 ("NBDV-R"), NBDV 세분화 프로세스, 또는 심도-중심 (depth-oriented) NBDV (Do-NBDV) 로서 본원에서 지칭될 수도 있다.

[0135] NBDV 유도 프로세스가 이용가능한 디스패리티 벡터를 반환할 때 (예컨대, NBDV 유도 프로세스가 NBDV 유도 프로세스가 이웃하는 블록의 디스패리티 모션 벡터 또는 IDV 에 기초하여 현재의 블록에 대한 디스패리티 벡터를 유도할 수 있다는 것을 표시하는 변수를 반환할 때), 비디오 코더는 참조 뷰의 심도 맵으로부터 심도 데이터를 추출함으로써 디스패리티 벡터를 추가로 세분화할 수도 있다. 일부의 예들에서, 세분화 프로세스는 다음의 2 개의 단계들을 포함한다:

[0136] 1) 기본 뷰와 같은 이전에 코딩된 참조 심도 뷰에서의 유도된 디스패리티 벡터에 의해 대응하는 심도 블록을 위치시키고; 대응하는 심도 블록의 사이즈는 현재의 PU 의 사이즈와 동일하다.

[0137] 2) 대응하는 심도 블록의 4 개의 코너 픽셀들로부터 하나의 심도 값을 선택하고, 그것을 세분화된 DV 의 수평 컴포넌트로 변환한다. DV 의 수직 컴포넌트는 변경되지 않는다.

[0138] 세분화된 DV 는 현재의 비디오 블록에 대한 인터-뷰 예측을 위하여 이용될 수도 있는 반면, 세분화되지 않은 DV 는 현재의 비디오 블록에 대한 인터-뷰 잔차 예측을 위하여 이용될 수도 있다. 게다가, 세분화된 DV 는 그것이 이하에서 더욱 상세하게 설명되는 후방 뷰 합성 예측 (backward view synthesis prediction; BVSP) 로 코딩될 경우에, 하나의 PU 의 모션 벡터로서 저장된다. 3D-HTM 7.0 및 더 이후의 버전들에 대한 제안된 NBDV 프로세스에서, 기본 뷰의 심도 뷰 컴포넌트는 NBDV 프로세스로부터 유도된 뷰 순서 인덱스의 값에 관계 없이 액세스된다.

[0139] 새로운 병합 후보를 생성하기 위한 서브-PU 레벨 인터-뷰 모션 예측 기법들은 An 등, "3D-CE3: Sub-PU level inter-view motion prediction (서브-PU 레벨 인터-뷰 모션 예측)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장 개발에 관한 합동 협력 팀, 6차 회의: Geneva, CH, 2013 년 10 월 25 일 - 11 월 1일, 문서 JCT3V-F0110, 이하 "JCT3V-F0110" 에서 제안되었다. JCT3V-F0110 는 다음의 링크로부터 다운로드될 수도 있다: [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=1447](http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1447). 새로운 후보는 병합 후보 리스트에 추가된다.

[0140] 도 8 은 서브-예측 유닛 (PU) 인터-뷰 모션 예측을 예시하는 개념도이다. 도 8 에서 도시된 바와 같이, 현재의 뷰 V1 에서의 현재의 PU (98) 는 다수의 서브-PU 들 (예컨대, 4 개의 서브-PU 들) 로 분할될 수도 있다. 각각의 서브-PU 에 대한 디스패리티 벡터는 참조 뷰 V0 에서 대응하는 참조 블록들을 위치시키기 위하여 이용될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 PU (8) 의 대응하는 서브-PU 들과의 이용을 위한 참조 블록들의 각각과 연관된 모션 벡터들을 복사 (즉, 재이용) 하도록 구성될 수도 있다.

[0141] 일 예에서, 서브-PU 병합 후보라는 명칭의 새로운 후보는 다음의 기법들을 이용하여 유도된다. 먼저, nPSW x nPSH 에 의한 현재의 PU 의 사이즈, NxN 에 의한 시그널링된 서브-PU 사이즈, 및 subWxsubH 에 의한 최종 서브-PU 사이즈를 나타낸다. 현재의 PU 는 PU 사이즈 및 시그널링된 서브-PU 사이즈에 따라 다음과 같이 하나 또는 다수의 서브-PU 들로 분할될 수도 있다:

$$\text{subW} = \max ( N, \text{nPSW} ) != N ? N : \text{nPSW};$$

$$\text{subH} = \max ( N, \text{nPSH} ) != N ? N : \text{nPSH};$$

[0144] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 (참조 픽처 리스트 0 또는 참조 픽처 리스트 1 의 어느 하나를 나타내는 X 를 갖는) 각각의 참조 픽처 리스트에 대하여 디폴트 모션 벡터 tmvLX 를 (0, 0) 로, 그리고 참조 인덱스 refLX 를 -1 로 설정할 수도 있다. 래스터 스캔 순서에서의 각각의 서브-PU 에 대하여, 다음이 적용된다:

[0145] - 다음에 의해 참조 샘플 로케이션 (xRefSub, yRefSub) 을 획득하기 위하여, DoNBDV 유도 프로세스 또는 NBDV 프로세스로부터 획득된 DV 를 현재의 서브-PU 의 중간 위치에 추가한다:

$$\text{xRefSub} = \text{Clip3}( 0, \text{PicWidthInSamplesL} - 1, \text{xPSub} + \text{nPSWsub}/2 + ( ( \text{mvDisp}[ 0 ] + 2 )$$

>> 2 ) )

[0147]  $yRefSub = Clip3( 0, PicHeightInSamplesL - 1, yPSub + nPSHSub /2 + ( ( mvDisp[ 1 ] + 2 ) >> 2 ) )$

[0148] ( $xRefSub$ ,  $yRefSub$ ) 를 커버하는 참조 뷰에서의 블록은 현재의 서브-PU 에 대한 참조 블록으로서 이용된다.

[0149] - 식별된 참조 블록에 대하여:

[0150] 1) 식별된 참조 블록이 시간적 모션 벡터들을 이용하여 코딩될 경우, 다음이 적용된다:

[0151] - 연관된 모션 파라미터들은 현재의 서브-PU 에 대한 후보 모션 파라미터들로서 이용될 수 있다.

[0152] -  $tmvLX$  및  $refLX$  는 현재의 서브-PU 의 모션 정보로 업데이트된다.

[0153] - 현재의 서브-PU 가 레스터 스캔 순서에서 최초의 것이 아닐 경우, 모션 정보 ( $tmvLX$  및  $refLX$ ) 는 모든 이전의 서브-PU 들에 의해 승계된다.

[0154] 2) 이와 다를 경우 (참조 블록이 인트라 코딩됨), 현재의 서브-PU 의 모션 정보는  $tmvLX$  및  $refLX$  로 설정된다.

[0155] 상이한 서브-PU 블록 사이즈들은  $4x4$ ,  $8x8$ , 및  $16x16$  을 포함하여, 서브-PU 레벨 인터-뷰 모션 예측을 위한 상기 설명된 기법들에서 이용될 수도 있다. 서브-PU 블록에 대한 사이즈는 뷰 파라미터 세트 (view parameter set; VPS) 와 같은 파라미터 세트에서 시그널링될 수도 있다.

[0156] 진보된 잔차 예측 (ARP) 은 추가적인 코딩 효율을 제공하기 위하여 뷰들 사이의 잔차 상관을 활용하는 것을 추구하는 코딩 틀이다. ARP 에서, 잔차 예측자는 참조 뷰에서의 모션 보상을 위한 현재의 뷰에서 모션 정보를 정렬함으로써 생성된다. 게다가, 가중화 인자들은 뷰들 사이의 품질 차이들을 보상하기 위하여 도입된다.

ARP 가 하나의 블록에 대하여 인에이블될 때, 현재의 잔차와 잔차 예측자 사이의 차이가 시그널링된다. 즉, 현재의 블록에 대한 예측자는 잔차 예측자의 잔차로부터 감산되고, 결과적인 차이가 시그널링된다. 3D-HEVC 에 대한 일부의 제안들에서, ARP 는 Part\_2Nx2N 와 동일한 파티션 모드를 갖는 인터-코딩된 CU 들에 적용되지만 한다.

[0157] 도 9 는 시간적으로-예측된 비디오 블록의 시간적 ARP 를 위한 예시적인 제안들에 대한 예시적인 예측 구조를 예시하는 개념도이다. Part\_2Nx2N 과 동일한 파티션 모드를 갖는 CU 들에 적용된 ARP 는 Zhang 등, "CE4: Advanced residual prediction for multiview coding (멀티뷰 코딩을 위한 진보된 잔차 예측)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 비디오 코딩 확장 개발에 관한 합동 협력 팀, 4차 회의: Incheon, KR, 2013 년 4 월 20-26 일, 문서 JCT3V-D0177 (MPEG 번호 m29008), 이하 "JCT3V-D0177" 에서 제안된 바와 같이, 4 차 JCT3V 회의에서 채택되었다. JCT3V-D0177 는 다음의 링크로부터 다운로드될 수도 있다:

[0158] [http://phenix.it-sudparis.eu/jct3v/doc\\_end\\_user/current\\_document.php?id=862](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=862).

[0159] 도 9 에서 도시된 바와 같이, 비디오 코더는 현재의, 예컨대, 종속적인 뷰  $V_m$  의 현재의 픽처 (102) 에서의 현재의 비디오 블록 (100) 에 대한 잔차의 예측에서 다음의 블록들을 호출하거나 식별한다.

[0160] 1) (뷰  $V_m$  에서의) 현재의 비디오 블록 (100): *Curr*

[0161] 2) 참조/기본 뷰 (도 9 에서의  $V_0$ ) 의 인터-뷰 참조 픽처 (108) 에서의 인터-뷰 참조 비디오 블록 (106): *Base*. 비디오 코더는 현재의 비디오 블록 (100) (*Curr*)의 DV (104) 에 기초하여 인터-뷰 참조 비디오 블록 (106) 을 유도한다. 비디오 코더는 위에서 설명된 바와 같이, NBDV 유도를 이용하여 DV (104) 를 결정할 수도 있다.

[0162] 3) 현재의 비디오 블록 (100) (*Curr*) 과 동일한 뷰 ( $V_m$ ) 에서의 시간적 참조 픽처 (114) 에서의 시간적 참조 비디오 블록 (112). 비디오 코더는 현재의 비디오 블록 (100) 의 TMV (110) 에 기초하여 시간적 참조 비디오 블록 (112) 을 유도한다. 비디오 코더는 본원에서 설명된 기법들 중의 임의의 것을 이용하여 TMV (110) 를 결정할 수도 있다.

[0163] 4) 참조 뷰, 즉, 인터-뷰 참조 비디오 블록 (106) (*Base*) 과 동일한 뷰에서의 시간적 참조 픽처 (118) 에서의 시간적 참조 비디오 블록 (116): *BaseTRef*. 비디오 코더는 현재의 비디오 블록 (100) (*Curr*) 의 TMV (110) 를 이용하여 참조 뷰에서의 시간적 참조 비디오 블록 (116) 을 유도한다. TMV+DV 의 벡터 (121)

는 현재의 비디오 블록 (100) (*Curr*) 에 관하여 시간적 참조 비디오 블록 (116) (*BaseTRef*) 을 식별할 수도 있다.

[0164] 비디오 인코더 (20) 가 TMV (110) 를 이용하여 식별하는 시간적 참조 비디오 블록 (112) 에 기초하여, 비디오 인코더 (20) 가 현재의 비디오 블록 (100) 을 시간적으로 인터 예측할 때, 비디오 인코더 (20) 는 현재의 비디오 블록 (100) 과 시간적 참조 비디오 블록 (112) 사이의 픽셀-대-픽셀 (pixel-by-pixel) 차이들을 잔차 블록으로서 결정한다. ARP 가 존재하지 않으면, 비디오 인코더 (20) 는 잔차 블록을 변환하고, 양자화하고, 엔트로피 인코딩할 것이다. 비디오 디코더 (30) 는 인코딩된 비디오 비트스트림을 엔트로피 디코딩할 것이고, 잔차 블록을 유도하기 위하여 역 양자화 및 변환을 수행할 것이고, 현재의 비디오 블록 (100) 을 재구성하기 위하여 잔차 블록을 참조 비디오 블록 (112) 의 재구성에 적용할 것이다.

[0165] ARP 를 이용하면, 비디오 코더들은, 잔차 블록의 값들을 예측하는, 즉, 현재의 비디오 블록 (100) (*Curr*) 과 시간적 참조 비디오 블록 (112) (*CurrTRef*) 사이의 차이를 예측하는 잔차 예측자 블록을 결정한다. 다음으로, 비디오 인코더 (20) 는 잔차 블록과 잔차 예측자 블록 사이의 차이를 단지 인코딩할 필요가 있을 수도 있어서, 현재의 비디오 블록 (100) 을 인코딩하기 위하여 인코딩된 비디오 비트스트림 내에 포함된 정보의 양을 감소시킬 수도 있다. 도 9 의 시간적 ARP 예에서, 현재의 비디오 블록 (100) 의 잔차에 대한 예측자는, 현재의 비디오 블록 (100) (*Curr*) 및 시간적 참조 비디오 블록 (112) (*CurrTRef*) 에 대응하며, DV (104) 에 의해 식별되는 참조/기본 뷰 ( $V_0$ ) 에서의 블록들에 기초하여 결정된다. 참조 뷰에서의 이 대응하는 블록들 사이의 차이는 잔차, 즉, 현재의 비디오 블록 (100) (*Curr*) 과 시간적 참조 비디오 블록 (112) (*CurrTRef*) 사이의 차이의 양호한 예측자일 수도 있다. 특히, 비디오 코더들은 참조 뷰에서의 인터-뷰 참조 비디오 블록 (106) (*Base*) 및 시간적 참조 비디오 블록 (116) (*BaseTRef*) 을 식별하고, 인터-뷰 참조 비디오 블록 (106) 과 시간적 참조 비디오 블록 (116) 사이의 차이 (*BaseTRef-Base*) 에 기초하여 잔차 예측자 블록을 결정하고, 감산 동작은 나타낸 픽셀 어레이들의 각각의 픽셀에 적용한다. 일부의 예들에서, 비디오 코더는 가중화 인자  $w$  를 잔차 예측자에 적용할 수도 있다. 이러한 예들에서, 현재의 블록, 즉, 잔차 예측자 블록과 합산된 참조 블록의 최종 예측자는 다음으로서 나타낼 수도 있다: *CurrTRef+ w\*(BaseTRef-Base)*.

[0166] 도 10 은 현재의 뷰 ( $V_m$ ) 에서의 현재의 비디오 블록 (120) 의 시간적 ARP 를 위한 예시적인 양방향 예측 구조를 예시하는 개념도이다. 상기 설명들 및 도 9 는 단방향 예측을 예시한다. ARP 를 양방향 예측의 경우로 확장할 때, 비디오 코더는 현재의 비디오 블록 (120) 에 대한 잔차 예측자 블록 식별하기 위하여 상기 기법들을 잔차 픽처 리스트들 중의 하나 또는 양자에 적용할 수도 있다. 특히, 비디오 코더는 이들 중의 하나가 시간적 ARP 를 위하여 이용가능한 TMV 를 포함하는지를 결정하기 위하여 현재의 비디오 블록 (120) 에 대한 참조 리스트들 중의 하나 또는 양자를 검사할 수도 있다. 도 10 에 의해 예시된 예에서, 현재의 비디오 블록 (120) 은 제 1 참조 픽처 리스트 (RefPicList0) 에서 제 1 시간적 참조 픽처 (134) 를 지시하는 TMV (130), 및 제 2 참조 픽처 리스트 (RefPicList1) 에서 제 2 시간적 참조 픽처 (136) 를 지시하는 TMV (132) 와 연관된다.

[0167] 일부의 예들에서, 비디오 코더는 이들 중의 하나가 시간적 ARP 를 위하여 이용가능한 TMV 를 포함하는지 여부를 결정하기 위하여 검사 순서에 따라 참조 픽처 리스트들을 검사할 것이고, 제 1 리스트가 이러한 TMV 를 포함할 경우에 검사 순서에 따라 제 2 리스트를 검사할 필요가 없을 것이다. 일부의 예들에서, 비디오 코더는 양자의 참조 픽처 리스트들을 검사할 것이고, 양자의 리스트들이 TMV 를 포함할 경우, 예컨대, 현재의 비디오 블록의 잔차에 관한 TMV 들을 이용하여 생성된 결과적인 잔차 예측자들의 비교에 기초하여 어느 TMV 를 이용할 것인지를 결정할 것이다. 특히, ARP 를 위한 현재의 제안들, 예컨대, JCT3VC-D0177 에 따르면, 현재의 블록이 하나의 참조 픽처 리스트에 대한 (상이한 뷰에서의) 인터-뷰 참조 픽처를 이용할 때, 잔차 예측 프로세스가 디스에이블된다.

[0168] 도 10 에서 예시된 바와 같이, 비디오 코더는, 현재의 픽처 (122) 와는 상이한 참조 뷰 ( $V_0$ ) 내에 있지만, 동일한 액세스 유닛 내에 있는 인터-뷰 참조 픽처 (128) 내의 대응하는 인터-뷰 참조 비디오 블록 (126) (*Base*) 을 식별하기 위하여, 예컨대, NBDV 유도 프로세스에 따라 현재의 비디오 블록 (120) 에 대해 식별된 DV (124) 를 이용할 수도 있다. 비디오 코더는 또한, 양자의 참조 픽처 리스트들, 예컨대, RefPicList0 및 RefPicList1 에서의 참조 뷰의 다양한 시간적 참조 픽처들 내의 인터-뷰 참조 비디오 블록 (126) (*Base*) 에 대한 시간적 참조 블록들 (*BaseTRef*) 을 식별하기 위하여, 현재의 비디오 블록 (120) 에 대한 TMV 들 (130 및 132) 을 이용할 수도 있다. 도 10 의 예에서, 비디오 코더는 현재의 비디오 블록 (120) 의 TMV 들 (130 및 132) 에 기초하여, 제 1 참조 픽처 리스트, 예컨대, RefPicList0 에서의 시간적 참조 픽처 (142) 내의 시간적 참조 비디오 블록 (*BaseTRef*) (140) 과, 제 2 참조 픽처 리스트, 예컨대, RefPicList1 에서의 시간적 참조 픽처 (146) 내의

시간적 참조 비디오 블록 (*BaseTRef*) (144) 을 식별한다.

[0169] 참조 뷰에서의 현재의 비디오 블록 (120) 의 TMV 들 (130 및 132) 의 이용은 도 10 에서의 점선 화살표들에 의해 예시된다. 도 10 에서, 참조 뷰에서의 시간적 참조 비디오 블록들 (140 및 144) 은 TMV 들 (130 및 132) 에 기초한 그 식별로 인해 모션 보상된 참조 블록들로서 지칭된다. 비디오 코더는 시간적 참조 비디오 블록 (140) 과 인터-뷰 참조 비디오 블록 (126) 사이의 차이에 기초하여, 또는 시간적 참조 비디오 블록 (144) 과 인터-뷰 참조 비디오 블록 (126) 사이의 차이에 기초하여, 현재의 비디오 블록 (120) 에 대한 잔차 예측자 블록을 결정할 수도 있다.

[0170] 재반복하기 위하여, 디코더 측에서의 제안된 시간적 ARP 프로세스는 다음과 같이 (도 10 을 참조하여) 설명될 수 있다:

1. 비디오 디코더 (30) 는 타겟 참조 뷰 ( $V_0$ ) 를 지시하는, 예컨대, NBDV 유도를 이용하여 현재의 3D-HEVC 에서 특정된 바와 같은 DV 를 획득한다. 다음으로, 동일한 액세스 유닛 내의 참조 뷰의 픽처 (128) 에서, 비디오 디코더 (30) 는 DV (124) 에 의해 대응하는 인터-뷰 참조 비디오 블록 (126) (*Base*) 을 식별한다.

2. 비디오 디코더 (30) 는 대응하는 인터-뷰 참조 비디오 블록 (126) 에 대한 모션 정보를 유도하기 위하여 현재의 비디오 블록 (120) 의 모션 정보, 예컨대, TMV (130, 132) 를 재이용한다. 비디오 디코더 (30) 는, 모션 보상된 시간적 참조 비디오 블록 (140, 144) (*BaseTRef*) 을 식별하기 위하여, 그리고 *BaseTRef-Base* 를 결정함으로써 잔차 예측자 블록을 결정하기 위하여, 현재의 비디오 블록 (120) 의 TMV (130, 132) 와, 참조 비디오 블록 (126) 에 대한 참조 뷰에서의 유도된 참조 픽처 (142, 146) 에 기초하여, 대응하는 인터-뷰 참조 비디오 블록 (126) 에 대한 모션 보상을 적용할 수도 있다. 현재의 블록, 대응하는 블록 (*Base*), 및 모션 보상된 블록 (*BaseTRef*) 사이의 관계는 도 9 및 도 10 에서 도시되어 있다. 일부의 예들에서, 현재의 뷰 ( $V_m$ ) 의 참조 픽처와 동일한 POC (픽처 순서 카운트) 값을 가지는 참조 뷰 ( $V_0$ ) 에서의 참조 픽처는 대응하는 블록의 참조 픽처로서 선택된다.

3. 비디오 디코더 (30) 는 가중화된 잔차 예측자 블록을 얻기 위하여 가중화 인자  $w$  를 잔차 예측자 블록에 적용할 수도 있고, 현재의 비디오 블록 (120) 을 재구성하기 위하여 가중화된 잔차 블록의 값들을 예측된 샘플들에 추가할 수도 있다.

[0174] 도 11 은 이 개시물에서 설명된 기법들에 따라 인터-뷰 예측된 비디오 블록의 인터-뷰 ARP 를 위한 예시적인 예측 구조의 개념도이다. 도 11 과 관련된 기법들은 Zhang 등, "CE4: Further improvements on advanced residual prediction (진보된 잔차 예측에 관한 추가의 개선들)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장 개발에 관한 합동 협력 팀, 6차 회의: Geneva, CH, 2013 년 10 월 25 일 - 11 월 1 일, 이하 "JCT3V-F0123" 에서 제안되었다. JCT3V-F0123 는 다음의 링크로부터 다운로드될 수도 있다: [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=1460](http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1460).

[0175] 도 11 에서 예시된 예시적인 기법에 따르면, 비디오 코더, 예컨대, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, 인터-뷰 예측되는 현재의 블록의 잔차를 예측하기 위하여, 상이한 액세스 유닛으로부터의 인터-뷰 잔차를 이용할 수도 있다. ARP 가 현재의 블록의 모션 벡터가 DMV 일 때에 수용되지 않고, 현재의 비디오 블록에 대한 모션 벡터가 TMV 일 때에 수행되지만 하는 ARP 에 대한 제안들과 대조적으로, 도 11 의 예시적인 기법은 ARP 를 수행하기 위하여 DMV 를 이용한다.

[0176] 특히, 현재의 픽처 (152) 에서의 현재의 비디오 블록 (150) (Curr) 의 모션 벡터가 DMV (154) 이고, 참조 뷰 ( $V_0$ ) 에서의 인터-뷰 참조 픽처 (158) 내의 인터-뷰 참조 비디오 블록 (156) (*Base*) 이 적어도 하나의 TMV (160) 를 포함할 때, 도 11 의 예시적인 기법은 비디오 코더, 예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30) 에 의해 수행될 수도 있다. 일부의 예들에서, DMV (154) 는 현재의 비디오 블록 (150) 의 모션 정보 예측을 위한 IDMVC 로서 작동하기 위하여 DMV 로 변환되었던 DV 일 수도 있다.

[0177] 비디오 코더는 현재의 비디오 블록 (150) 에 대한 DMV (154) 를 이용하여 인터-뷰 참조 픽처 (158) 에서 인터-뷰 참조 비디오 블록 (156) (*Base*) 을 식별한다. 비디오 코더는 참조 뷰 ( $V_0$ ) 에서의 시간적 참조 픽처 (164) 내의 시간적 참조 비디오 블록 (162) (*BaseTRef*) 을 식별하기 위하여, DMV 와 함께, 인터-뷰 참조 비디오 블록 (156) 의 TMV (160) 및 연관된 참조 픽처, 예컨대, 참조 뷰 ( $V_0$ ) 에서의 시간적 참조 픽처 (164) 를 이용한다. TMV (160) 및 DMV (154) 에 기초한 시간적 참조 비디오 블록 (162) (*BaseTRef*) 의 식별은 점선 벡터

(170) (TMV + DMV)에 의해 표현된다. 비디오 코더는 또한, 현재의 뷰 ( $V_m$ )에서의 시간적 참조 픽처 (168) 내의 시간적 참조 비디오 블록 (166) (*CurrTRef*)을 식별하기 위하여 TMV (160)를 이용한다. 참조 뷰 ( $V_0$ )에서의 시간적 참조 비디오 블록 (162) (*BaseTRef*) 및 현재의 뷰 ( $V_m$ )에서의 시간적 참조 비디오 블록 (166) (*CurrTRef*)은 동일한 액세스 유닛 내에 있을 수도 있고, 즉, 참조 뷰 ( $V_0$ )에서의 시간적 참조 픽처 (164) 및 현재의 뷰 ( $V_m$ )에서의 시간적 참조 픽처 (168)는 동일한 액세스 유닛 내에 있을 수도 있다.

[0178] 다음으로, 비디오 코더, 예컨대, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 이 2개의 후자의 블록들 사이의 픽셀-대-픽셀 차이, 즉, 현재의 뷰에서의 시간적 참조 비디오 블록 (166)과 참조 뷰에서의 시간적 참조 비디오 블록 (162) 사이의 차이, 또는 *CurrTRef-BaseTRef*에 기초하여, 현재의 비디오 블록 (150)과는 상이한 액세스 유닛에서의 인터-뷰 잔차 예측자 블록을 계산할 수도 있다. 인터-뷰 잔차 예측자로서 나타낸 차이 신호는 현재의 비디오 블록 (150)의 잔차를 예측하기 위하여 이용될 수 있다. 현재의 비디오 블록 (150)의 예측 신호는 인터-뷰 예측자, 즉, 인터-뷰 참조 비디오 블록 (156) (*Base*)과, 현재의 뷰에서의 시간적 참조 비디오 블록 (166) 및 참조 뷰에서의 시간적 참조 비디오 블록 (162) 사이의 차이에 기초하여 결정된 상이한 액세스 유닛에서의 예측된 인터-뷰 잔차와의 합일 수도 있다. 일부의 예들에서, 가중화 인자  $w$ 는 상이한 액세스 유닛에서의 예측된 인터-뷰 잔차에 적용된다. 이러한 예들에서, 현재의 비디오 블록 (150)의 예측 신호는 다음과 같을 수도 있다: *Base + w\*(CurrTRef-BaseTRef)*.

[0179] 일부의 예들에서, 비디오 코더는 위에서 논의된 바와 같이, 예컨대, 시간적 ARP를 위한 타겟 참조 픽처의 결정과 유사하게, 인터-뷰 ARP를 위한 타겟 액세스 유닛에서의 타겟 참조 픽처를 결정할 수도 있다. 일부의 예들에서, JCT3V-D0177을 참조하여 위에서 논의된 바와 같이, 각각의 참조 픽처 리스트에 대한 타겟 참조 픽처는 참조 픽처 리스트에서의 제1 참조 픽처이다. 다른 예들에서, 하나 또는 양자의 참조 픽처 리스트들에 대한 타겟 참조 픽처, 예컨대, 타겟 POC는 예컨대, PU, CU, 슬라이스, 픽처, 또는 다른 것에 기초하여 비디오 인코더 (20)로부터 비디오 디코더 (30)로 시그널링될 수도 있다. 다른 예들에서, 각각의 참조 픽처 리스트에 대한 타겟 참조 픽처는 현재의 블록 및 더 작은 참조 픽처 인덱스와 비교하여 최소 POC 차이를 가지는 참조 픽처 리스트에서의 시간적 참조 픽처이다. 다른 예들에서, 2개의 참조 픽처 리스트들에 대한 타겟 참조 픽처는 동일하다.

[0180] TMV (160)에 의해 표시된 참조 뷰에서의 시간적 참조 비디오 블록을 포함하는 픽처가 타겟 ARP 참조 픽처와는 상이한 액세스 유닛 (시간 인스턴스)내에 있을 경우, 비디오 코더는 인터-뷰 ARP를 위한 참조 뷰에서의 시간적 참조 비디오 블록 (162) (*BaseTRef*)을 식별하기 위하여, TMV (160)를 타겟 참조 픽처, 예컨대, 타겟 참조 픽처 (164)로 스케일링할 수도 있다. 이러한 예들에서, 비디오 코더는 타겟 ARP 참조 픽처를 포함하는 액세스 유닛에서 시간적 참조 비디오 블록 (162)을 위치시킨다. 비디오 코더는 POC 스케일링에 의해 TMV (160)를 스케일링할 수도 있다. 게다가, 스케일링된 TMV는 타겟 ARP 참조 픽처에서 위치되는 현재의 뷰에서의 시간적 참조 비디오 블록 (*CurrTRef*) (166)을 식별하기 위하여 이용된다.

[0181] 일부의 예들에서, 비디오 코더는 TMV (160)를 LX (X는 0 또는 1임) 타겟 참조 픽처로 스케일링하고, 여기서, LX는 TMV를 포함하는 PU의 RefPicListX에 대응한다. 일부의 예들에서, 비디오 코더는 TMV들을 RefPicList0 또는 RefPicList1 중의 어느 하나 또는 양자로부터 L0 또는 L1 타겟 참조 픽처로 각각 스케일링할 수도 있다. 일부의 예들에서, 비디오 코더는 TMV (160)를 LX 타겟 참조 픽처로 스케일링하고, X는 현재의 비디오 블록 (150), 예컨대, 현재의 PU의 DMV (154)가 RefPicListX에 대응한다는 조건을 충족시킨다.

[0182] 유사하게, 일부의 예들에서, 비디오 코더는 타겟 참조 뷰에서의 참조 픽처 (158)내의 인터-뷰 참조 비디오 블록 (156)을 식별하기 전에, DMV (154)를 ARP를 위한 타겟 참조 뷰로 스케일링한다. 비디오 코더는 뷰 순서 차이 스케일링에 의해 DMV (154)를 스케일링할 수도 있다. 타겟 참조 뷰는 비디오 인코더 (20) 및 비디오 디코더 (30)에 의해 미리 결정되고 알려질 수도 있거나, 예컨대, PU, CU, 슬라이스, 픽처, 또는 다른 것에 기초하여 비디오 인코더 (20)로부터 비디오 디코더 (30)로 시그널링될 수도 있다.

[0183] 인터-뷰 ARP의 일부의 예들에서, 비디오 코더, 예컨대, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 동일한 예측 구조 및 도 11에서 예시된 식별된 참조 비디오 블록들 (156, 162, 및 166)을 이용하여 현재의 블록 (150)에 대한 예측 신호를 유도할 수도 있지만, 상이한 액세스 유닛에서의 참조 블록들 (162 및 166)이 아니라, 참조 뷰에서의 참조 블록들 (156 및 162) 사이의 차이에 기초하여 잔차 예측자 블록을 결정할 수도 있다. 이러한 예들에서, 비디오 코더는 가중화 인자를 다른 샘플 어레이들, 예컨대, 참조 뷰에서의 참조 블록들 (156 및 162) 사이의 차이에 적용할 수도 있고, 따라서, 다음과 같이 현재의 비디오 블록 (150)에 대한 예측

신호를 유도할 수도 있다:  $CurrTRef + w * (Base - BaseTRef)$ . 인터-뷰 ARP 의 일부의 예들에서, 비디오 코더는 참조 비디오 블록들 (156, 162, 및 166) 을 유도하기 위하여, 이들이 분수 픽셀 위치들과 정렬될 경우에, 이 중선형 필터 (bilinear filter) 들을 포함하는 다양한 보간 필터들을 이용할 수도 있다.

[0184] 도 11 은 현재 및 참조 뷰들에서의 시간적 참조 비디오 블록들이 인터-뷰 참조 블록과 연관된 TMV 및 인터-뷰 참조 비디오 블록의 연관된 참조 픽처를 이용하여 식별되는 인터-뷰 ARP 예들을 예시하지만, 다른 예들에서, 다른 TMV 들 및 연관된 참조 픽처들은 현재 및 참조 뷰들에서의 시간적 참조 비디오 블록들을 식별하기 위하여 이용될 수도 있다. 예를 들어, 현재의 비디오 블록의 DMV 가 현재의 비디오 블록의 제 1 참조 픽처 리스트 (예컨대, RefPicList0 또는 RefPicList1)로부터 나온 것일 경우, 비디오 코더는 현재의 블록의 제 2 참조 픽처 리스트에 대응하는 TMV 와, 현재의 비디오 블록의 제 2 참조 픽처 리스트 (예컨대, RefPicList0 또는 RefPicList1 의 다른 것)로부터의 연관된 참조 픽처를 이용할 수도 있다. 이러한 예들에서, 비디오 코더는 TMV 와 연관된 참조 픽처에서의 현재의 뷰에서의 시간적 참조 비디오 블록을 식별할 수도 있거나, 현재의 뷰에서의 시간적 참조 비디오 블록을 식별하기 위하여 TMV 를 ARP 를 위한 타겟 액세스 유닛 및 타겟 참조 픽처로 스케일링 할 수도 있다. 이러한 예들에서, 비디오 코더는, 현재의 뷰에서의 시간적 참조 비디오 블록이 위치되었던 참조 픽처와 동일한 액세스 유닛 내에 있는 참조 픽처에서의 참조 뷰 내의 시간적 참조 비디오 블록을 식별할 수도 있다. 다른 예들에서는, 인터-뷰 참조 비디오 블록의 TMV, 또는 현재의 비디오 블록의 다른 참조 픽처 리스트의 TMV 가 아니라, 비디오 코더는 ARP 를 위한 현재 및 참조 뷰들에서의 시간적 참조 비디오 블록들을 식별하기 위하여, 현재의 비디오 블록의 공간적 또는 시간적으로 이웃하는 비디오 블록들의 모션 정보로부터 유도된 TMV 및 연관된 참조 픽처를 유사하게 이용할 수도 있다.

[0185] 다음의 설명에서는, 하나의 참조 픽처 리스트에 대한 대응하는 참조가 시간적 참조 픽처이고 ARP 가 적용될 경우, ARP 프로세스는 시간적 ARP 로서 나타낸다. 이와 다르게, 하나의 참조 픽처 리스트에 대한 대응하는 참조가 인터-뷰 참조 픽처이고 ARP 가 적용될 경우, ARP 프로세스는 인터-뷰 ARP 로서 나타낸다.

[0186] ARP 를 위한 일부의 제안들에서는, 3 개의 가중화 인자들, 즉, 0, 0.5, 및 1 이 이용될 수도 있다. 현재의 CU 에 대한 최소 레이트-왜곡 비용 (rate-distortion cost)으로 이어지는 가중화 인자는 최종 가중화 인자로서 선택되고, 대응하는 가중화 인자 인덱스 (가중화 인자 0, 1, 및 0.5 에 각각 대응하는 0, 1, 및 2) 는 CU 레벨에서 비트스트림에서 송신된다. 하나의 CU 에서의 모든 PU 예측들은 동일한 가중화 인자를 공유한다. 가중화 인자가 0 과 동일할 때, ARP 는 현재의 CU 에 대하여 이용되지 않는다.

[0187] 3D-HEVC 를 위한 ARP 에 관련된 추가적인 양태들은 Zhang 등, "3D-CE4: Advanced residual prediction for multiview coding (멀티뷰 코딩을 위한 진보된 잔차 예측)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 비디오 코딩 확장 개발에 관한 합동 협력 팀, 3차 회의: Geneva, CH, 2013 년 1 월 17 - 23 일, 문서 JCT3V-C0049 (MPEG 번호 m27784), 이하 "JCT3V-C0049" 에서 설명되어 있다. JCT3V-C0049 는 다음의 링크로부터 다운로드될 수도 있다:

[0188] [http://phenix.int-evry.fr/jct3v/doc\\_end\\_user/current\\_document.php?id=487](http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=487).

[0189] JCT3V-C0049 에서, 비-제로 (non-zero) 가중화 인자들로 코딩된 상이한 PU 들의 참조 픽처들은 PU 마다 (또는 현재의 비디오 블록마다) 상이할 수도 있다. 그러므로, 참조 뷰로부터의 상이한 픽처들은 모션-보상된 블록 (BaseTRef), 예컨대, 도 9 및 도 10 에서의 시간적 참조 비디오 블록들 (116, 140, 및 144), 또는 참조 뷰 (Base) 에서의 대응하는 인터-뷰 참조 비디오 블록, 예컨대, 도 9 및 도 10 에서의 인터-뷰 참조 비디오 블록들 (106 및 126) 을 생성하기 위하여 액세스될 필요가 있을 수도 있다.

[0190] 시간적 잔차에 대하여, 가중화 인자가 0 과 동일하지 않을 때, 현재의 PU 의 모션 벡터들은 잔차 및 및 잔차 예측자 생성 프로세스들 양자에 대한 모션 보상을 수행하기 전에 고정된 픽처를 향해 스케일링된다. ARP 가 인터-뷰 잔차에 적용될 때, 참조 블록 (예컨대, 도 11 에서의 블록 (156)) 의 시간적 모션 벡터들은 잔차 및 잔차 예측자 생성 프로세스들의 양자에 대한 모션 보상을 수행하기 전에 고정된 픽처를 향해 스케일링된다. 양자의 경우들 (즉, 시간적 잔차 또는 인터-뷰 잔차) 에 대하여, 고정된 픽처는 각각의 참조 픽처 리스트의 제 1 이용가능한 시간적 참조 픽처로서 정의된다. 디코딩된 모션 벡터가 고정된 픽처를 지시하지 않을 때, 그 것은 먼저 스케일링되고, 그 다음으로, CurrTRef 및 BaseTRef 를 식별하기 위하여 이용된다.

[0191] ARP 를 위하여 이용된 이러한 참조 픽처는 타겟 ARP 참조 픽처로 칭해진다. 현재의 슬라이스가 B 슬라이스 일 때, 타겟 ARP 참조 픽처는 하나의 특정한 참조 픽처 리스트와 연관되는 것에 주목해야 한다. 그러므로, 2 개의 타겟 ARP 참조 픽처들이 사용될 수도 있다.

- [0192] 타겟 ARP 참조 픽처들의 이용가능한 검사는 다음과 같이 수행될 수도 있다: 먼저, RpRefPicLX 에 의해 하나의 참조 픽처 리스트 X (X 는 0 또는 1 임) 와 연관된 타겟 ARP 참조 픽처를 나타내고, RefPicInRefViewLX 에 의해, NBDV 유도 프로세스로부터 유도된 것과 동일한 뷰 순서 인덱스를 가지며, RpRefPicLX 의 동일한 POC 값을 갖는 뷰에서의 픽처를 나타낸다.
- [0193] 다음의 조건 중의 하나가 거짓 (false) 일 때, ARP 는 참조 픽처 리스트 X 에 대하여 디스에이블된다:
- RpRefPicLX 가 이용불가능함
  - RefPicInRefViewLX 가 디코딩된 픽처 버퍼 내에 저장되지 않음
- [0194]
- [0195]
- [0196] - RefPicInRefViewLX 가 NBDV 유도 프로세스로부터의 DV 또는 현재의 블록과 연관된 DMV 에 의해 위치된 대응하는 블록 (예컨대, 도 9 에서의 블록 (106) 또는 도 11 에서의 블록 (156)) 의 참조 픽처 리스트들 중의 임의의 것 내에 포함되지 않고, ARP 는 이 참조 픽처 리스트에 대하여 디스에이블됨.
- [0197] ARP 가 적용될 때, 이중선형 필터는 잔차 및 잔차 예측자를 생성할 때에 이용될 수도 있다. 즉, 예컨대, 도 9 에서의 블록들 (106, 112, 및 116) 은 이중선형 필터를 이용하여 생성된다.
- [0198] 도 12 는 하나의 참조 픽처 리스트에 대한 인터-뷰 예측과, 또 다른 참조 픽처 리스트에 대한 시간적 예측을 이용하는 양방향 ARP 를 위한 예시적인 예측 구조를 예시하는 개념도이다. 도 12 의 예시적인 기법은, 현재의 비디오 블록 (250) 의 양방향 예측을 위한 하나의 예측 방향이 (예컨대, 참조 픽처 리스트 X 에 대한) 시간적 예측을 이용하고, 현재의 비디오 블록 (250) 의 다른 예측 방향이 (예컨대, 참조 픽처 리스트 Y ( $Y=1-X$ ) 에 대한) 인터-뷰 예측을 이용할 때에 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있다.
- [0199] 도 12 의 예에서, 현재의 비디오 블록 (250) 은 TMV (210) 및 DMV (254) 와 연관될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 도 9 를 참조하여 위에서 설명된 것과 유사한 방식으로 참조 픽처 리스트 X (즉, 제 1 예측 방향) 에 대한 참조 블록들을 식별하고 액세스하도록 구성될 수도 있다.
- [0200] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의, 예컨대, 종속적인 뷰  $V_m$  의 현재의 픽처 (253) 에서의 현재의 비디오 블록 (250) 에 대한 잔차의 예측에서 다음의 블록들을 식별한다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 참조/기본 뷰 (도 12 에서의  $V_0$ ) 의 인터-뷰 참조 픽처 (258) 에서 인터-뷰 참조 비디오 블록 (206) (*BaseX*) 을 식별한다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (250) (*Curr*) 의 DV (204) 에 기초하여 인터-뷰 참조 비디오 블록 (206) 을 식별한다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 위에서 설명된 바와 같이, NBDV 유도를 이용하여 DV (204) 를 결정할 수도 있다.
- [0201] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (250) (*Curr*) 과 동일한 뷰 ( $V_m$ ) 에서의 시간적 참조 픽처 (270) 내의 시간적 참조 비디오 블록 (212) (*CurrTRefX*) 을 추가로 식별할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (250) 의 TMV (210) 를 이용하여 시간적 참조 비디오 블록 (212) 을 식별한다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 본원에서 설명된 기법들 중의 임의의 것을 이용하여 TMV (210) 를 결정할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 참조 뷰, 즉, 인터-뷰 참조 비디오 블록 (206) (*BaseX*) 과 동일한 뷰에서의 시간적 참조 픽처 (272) 내의 시간적 참조 비디오 블록 (216) (*BaseTRefX*) 을 추가로 식별할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (250) (*Curr*) 의 TMV (210) 를 이용하여 참조 뷰에서의 시간적 참조 비디오 블록 (216) 을 식별할 수도 있다. TMV (210) + DV (204) 의 벡터 (220) 는 현재의 비디오 블록 (250) (*Curr*) 에 관하여 시간적 참조 비디오 블록 (216) (*BaseTRefX*) 을 식별할 수도 있다. 도 12 에서 알 수 있는 바와 같이, 참조 픽처 리스트 X (즉, 제 1 예측 방향) 에 대하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 3 개의 참조 블록들 (즉, 참조 블록들 (206, 212, 및 216)) 을 식별하고 액세스하도록 구성된다.
- [0202] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 도 11 을 참조하여 위에서 설명된 것과 유사한 방식으로 참조 픽처 리스트 Y (즉, 제 1 예측 방향) 에 대한 참조 블록들을 식별하고 액세스하도록 구성될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 참조/기본 뷰 (도 12 에서의  $V_0$ ) 의 인터-뷰 참조 픽처 (258) 에서 인터-뷰 참조 비디오 블록 (256) (*BaseY*) 을 식별한다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (250) (*Curr*) 의 DMV (254) 에 기초하여 인터-뷰 참조 비디오 블록 (256) 을

식별한다.

- [0203] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (250) (*Curr*) 과 동일한 뷰 ( $V_m$ ) 에서의 시간적 참조 픽처 (268) 내의 시간적 참조 비디오 블록 (273) (*CurrTrefY*) 을 추가로 식별할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 비디오 블록 (250) 의 TMV' (285) 를 이용하여 시간적 참조 비디오 블록 (273) 을 식별할 수도 있다. 비디오 코더는 참조 뷰 ( $V_0$ ) 에서의 시간적 참조 픽처 (265) 내의 시간적 참조 비디오 블록 (271) (*BaseTrefY*) 을 식별하기 위하여, DMV 와 함께, 인터-뷰 참조 비디오 블록 (256) 의 TMV' (285) 및 연관된 참조 픽처, 예컨대, 참조 뷰 ( $V_0$ ) 에서의 시간적 참조 픽처 (265) 를 이용한다. TMV' (285) 및 DMV (254) 에 기초한 시간적 참조 비디오 블록 (271) (*BaseTrefY*) 의 식별은 점선 벡터 ( $TMV' + DMV$ ) 에 의해 표현된다. 참조 뷰 ( $V_0$ ) 에서의 시간적 참조 비디오 블록 (271) (*BaseTrefY*) 및 현재의 뷰 ( $V_m$ ) 에서의 시간적 참조 비디오 블록 (273) (*CurrTrefY*) 은 동일한 액세스 유닛 내에 있을 수도 있고, 즉, 참조 뷰 ( $V_0$ ) 에서의 시간적 참조 픽처 (265) 및 현재의 뷰 ( $V_m$ ) 에서의 시간적 참조 픽처 (268) 는 동일한 액세스 유닛 내에 있을 수도 있다.
- [0204] 도 12 에서 알 수 있는 바와 같이, 참조 픽처 리스트 Y (즉, 제 2 예측 방향) 에 대하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 추가적인 3 개의 참조 블록들 (즉, 참조 블록들 (256, 271, 및 273)) 을 식별하고 액세스하도록 구성된다.
- [0205] 3D-HEVC 에서의 ARP 를 위한 상기 언급된 기법들은 몇몇 단점들을 나타낸다. 일 예로서, 양방향 예측은 2 개의 상이한 참조 픽처 리스트들에 대한 모션 정보의 이용을 본질적으로 포함하므로, 모션 정보에 대한 메모리 액세스들의 수는 양방향 예측과 함께 블록-레벨 ARP 또는 PU-레벨 ARP 를 수행할 때에 증가된다. 게다가, 식별되고 액세스되는 참조 블록들의 수는 높다. 이에 따라, ARP 와 조합한 양방향 예측은 디코더 복잡도를 증가시킨다.
- [0206] 이 개시물은 비디오 디코더 복잡도를 감소시키기 위하여 ARP 를 위한 상기 언급된 문제를 해결하기 위한 다양한 예시적인 기법들을 제안한다. 이하에서 열거된 기법들의 각각은 ARP 와, ARP 를 위한 현재의 제안들에 관한 다른 연관된 비디오 코딩 기법들을 수행하기 위하여 필요하게 된 메모리 액세스들의 수를 감소시킨다.
- [0207] 도 13 은 이 개시물의 기법들에 따라, 하나의 참조 픽처 리스트에 대한 인터-뷰 예측과, 또 다른 참조 픽처 리스트에 대한 시간적 예측을 이용하는 양방향 ARP 를 위한 예시적인 예측 구조를 예시하는 개념도이다. 도 13 의 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 양방향 예측 및 ARP 를 이용하여 현재의 비디오 블록 (250) 을 코딩하도록 구성된다. 양방향 예측은 참조 픽처 리스트 X (예컨대, 제 1 예측 방향) 에 대한 시간적 예측 및 참조 픽처 리스트 Y (예컨대, 제 2 예측 방향) 에 대한 인터-뷰 예측을 포함한다.
- [0208] 도 13 의 기법들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 도 12 를 참조하여 위에서 설명된 것과 동일한 방식으로 참조 픽처 리스트 X (예컨대, 제 1 예측 방향) 에 대한 참조 블록 (206) (*BaseX*), 참조 블록 (216) (*BaseTrefX*), 및 참조 블록 (212) (*CurrTrefX*) 을 식별하도록 구성된다. 즉, TMV (210) 는 참조 블록 (206) (*BaseX*) 및 현재의 비디오 블록 (250) 에 관하여 참조 블록 (216) (*BaseTrefX*) 및 참조 블록 (212) (*CurrTrefX*) 을 각각 식별하기 위하여 이용된다. 게다가, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 (즉, 도 12 를 참조하여 위에서 설명된 것과 동일한 방식으로) DMV (254) 를 이용하여 참조 픽처 리스트 Y (예컨대, 제 2 예측 방향) 에 대한 참조 블록 (256) (*BaseY*) 을 식별하도록 구성된다.
- [0209] 그러나, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 *CurrTrefY* 및 *BaseTrefY* 를 식별하기 위하여 참조 블록 (256) (*BaseY*) 과 연관된 시간적 모션 정보를 이용하지 않는다. 그 대신에, 이 개시물의 기법들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 *CurrTrefY* 및 *BaseTrefY* 를 식별하기 위하여 참조 리스트 X 에 대한 시간적 모션 정보 (즉, TMV (210)) 를 이용하도록 구성될 수도 있다. 도 13 에서 도시된 바와 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 참조 블록 (256) (*BaseY*) 에 관한 TMV (210) 를 이용하여 뷰  $V_0$  에서의 참조 블록 (290) (*BaseTrefY*) 을 식별하도록 구성된다. 즉, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 양자의 DMV (254) 및 TMV (210) 를 이용하여 참조 블록 (290) (*BaseTrefY*) 을 식별하도록 구성된다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 TMV (210) 를 이용하여 현재의 비디오 블록 (250) 과 동일한 뷰 ( $V_m$ ) 에서의 CurrTrefY 를 식별하도록 추가로 구성된다. 이와 같이, 참조 블록 (212) 은 *CurrTrefX* 및 *CurrTrefY* 양자로서 작용한다. 이에 따라, 이 개시물의 기법들을 이용하면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 양방향 예측으로 ARP 를 수행할 때, 6 개 대신에, 5 개의 참조

블록들을 식별하고 액세스하기만 한다.

[0210] 요약하면, 참조 팩쳐 리스트 Y (예컨대, 제 2 예측 방향)에 대응하는 인터-뷰 ARP를 위한 참조 블록들을 식별할 시에, 비디오 인코더(20) 및 비디오 디코더(30)는 상이한 액세스 유닛에서의 참조 블록들(즉, 참조 블록들(290 및 212))을 식별하기 위하여 참조 팩쳐 리스트 X에 대한 시간적 예측과 연관된 시간적 모션 정보(예컨대, 도 13에서의 TMV(210))를 이용할 수도 있다. 이 경우, 인터-뷰 ARP를 수행할 때, 현재의 뷰와는 상이한 액세스 유닛에서의 참조 블록(즉, 참조 블록(212))은 생성될 필요가 없는데, 이것은 이러한 참조 블록이 참조 팩쳐 리스트 X에 대한 시간적 ARP를 위하여 식별된 참조 블록과 동일하기 때문이다. 즉, 참조 블록(212)은 시간적 ARP 및 인터-뷰 ARP 양자를 위하여 이용된다.

[0211] 이러한 방법으로, 제 1 예측 방향에 대한 시간적 모션 정보는 제 2 예측 방향에 대하여 재이용된다. 이와 같이, 제 2 예측 방향에 대응하는 제 1 인코딩된 블록의 모션 벡터에 의해 식별된 블록의 시간적 모션 정보가 액세스될 필요가 없으므로, 시간적 모션 정보의 더 적은 메모리 액세스들이 행해질 필요가 있고, 이에 따라, 더욱 신속한 비디오 디코딩을 허용한다. 게다가, ARP를 수행할 때에 이용된 참조 블록들의 총 수는 6으로부터 5로 감소될 수도 있고, 이것은 승산 및 가산 동작들을 이용한 보간의 측면에서 더 적은 연산 복잡도로 귀착된다. 마찬가지로, 양방향 인터-예측을 수행할 때, 비디오 인코더(20)는 제 2 예측 방향을 인코딩할 때에 제 1 예측 방향에 대한 시간적 모션 정보를 재이용하도록 구성될 수도 있다.

[0212] 개시물의 또 다른 예에서, 비디오 인코더(20) 및 비디오 디코더(30)는 양방향 예측의(예컨대, 참조 팩쳐 리스트 X에 대응하는) 하나의 예측 방향이 시간적 참조 팩쳐에 대응하고(예컨대, 참조 팩쳐 리스트 Y에 대응하는) 다른 예측 방향이 인터-뷰 참조 팩쳐에 대응할 때에 간략화된 ARP 프로세스를 수행하도록 구성될 수도 있다. 이 경우, 참조 팩쳐 리스트 X에 대응하는 시간적 ARP를 위하여, 비디오 인코더(20) 및 비디오 디코더(30)는 NBDV/DoNBDV 유도 프로세스로부터 유도된 디스패리티 벡터를 이용하는 대신에, 참조 뷰에서의 참조 블록들(예컨대, 도 12에서의 참조 블록(273))을 식별하기 위하여 인터-뷰 참조 팩쳐와 연관된 디스패리티 모션 벡터(MVY)를 이용하도록 구성될 수도 있다. 한편, NBDV 또는 DoNBDV 프로세스로부터의 디스패리티 벡터는 변경된 상태로 유지되고, 이것은 IPMVC 또는 IDMVC를 생성하기 위하여 인터-뷰 모션 예측에서 여전히 이용될 수도 있다.

[0213] 상기 방법들은 PU-레벨 ARP 또는 블록-레벨 ARP 양자에 적용될 수도 있다는 것에 주목해야 한다. PU-레벨 및 블록-레벨 ARP는 이하에서 더욱 상세하게 설명될 것이다.

[0214] 블록-레벨 ARP를 위한 기법들이 지금부터 논의될 것이다. 하나의 PU 내의 모든 블록들이 ARP(때때로, PU-레벨 ARP로 칭해짐)를 위한 동일한 모션 정보를 공유하는 상기 설명과는 상이하게, 블록-레벨 ARP에서는, ARP를 수행하기 위하여, 하나의 PU가 몇몇 서브-블록들(예컨대, 8x8 서브-블록들)로 분할되고 각각의 서브-블록은 그 자신의 유도된 모션 정보와 연관된다. 즉, 각각의 서브-블록은 현재의 PU와 동일한 모션 정보를 공유한다. 그러나, 유도된 모션 정보(즉, 시간적 ARP에서의 디스패리티 벡터 또는 인터-뷰 ARP에서의 시간적 모션 벡터)는 각각의 서브-블록에 대해 결정될 수도 있다.

[0215] 도 14는 블록-기반 시간적 ARP를 예시하는 개념도이다. 도 14에서 도시된 바와 같이, 현재의 팩쳐(302)는 4개의 서브-블록들(300a, 300b, 300c, 및 300d)로 분할된 현재의 블록(300)(Curr)을 포함한다. 모션 벡터(310)(mvLX)는 현재의 블록(300)에 대해 인터-예측을 수행하기 위하여 이용된 모션 벡터이다. 모션 벡터(310)는 참조 팩쳐(314)에서, 서브-블록들(312a 내지 312d)을 포함하는 참조 블록(312)(CurrTref)을 지시한다. 현재의 팩쳐(302) 및 참조 팩쳐(314)는 동일한 뷰(Vm)내에 있다.

[0216] 블록-기반 시간적 ARP를 위하여, 디폴트(default) 유도된 모션 벡터는 서브-블록들(300a 내지 300d)의 각각에 대해 이용된다. 시간적 ARP를 위하여, 디폴트 유도된 모션 벡터는 도 14에서의 i 번째 서브-블록에 대한 DV[i]에 의해 나타낸 디스패리티 벡터이고, 현재의 ARP에서와 같이, NBDV 유도 프로세스를 이용하여 유도될 수도 있다. 즉, NBDV 유도 프로세스는 서브-블록들(300a 내지 300d)의 각각에 대한 DV를 유도하기 위하여 서브-블록들(300a 내지 300d)의 각각에 대해 수행될 수도 있다. 유도된 DV들의 각각은 참조 뷰(308)에서의 특정한 참조 블록(306a 내지 306d)(Base)을 지시한다. 예를 들어, DV(304)(DV[0])는 참조 블록(306a)을 지시하고, DV(305)(DV[1])는 참조 블록(306b)을 지시한다.

[0217] 참조 뷰(308)는 현재의 팩쳐(302)와 동일한 시간 인스턴스(time instance)에 있지만, 또 다른 뷰 내에 있다. 참조 블록(312)내의 서브-블록들(312a 내지 312d)중의 하나의 서브-블록의 중심 위치가 디스패리티 모션 벡터를 포함할 때, 현재의 서브-블록(300a 내지 300d)의 대응하는 하나에 대한 디스패리티 벡터 DV[i]

는 그 디스패리티 모션 벡터를 이용하기 위하여 업데이트된다. 즉, 예를 들어, 현재의 서브-블록 (300a)에 대응하는 참조 서브-블록 (312a)의 중심 위치가 연관된 디스패리티 모션 벡터를 가질 경우, 참조 서브-블록 (312a)과 연관된 디스패리티 모션 벡터는 서브-블록 (300a)에 대한 디스패리티 벡터로서 이용된다.

[0218] 일단 참조 블록들 (306a 내지 306d)의 각각이 식별되었으면, 모션 벡터 (310)는 참조 픽처 (318)에서의 참조 블록들 (316a 내지 316d) (BaseTRef)을 구하기 위하여 이용될 수도 있다. 참조 픽처 (318)는 현재의 픽처 (302)와는, 상이한 뷰뿐만 아니라 상이한 시간 인스턴스 내에 있다. 다음으로, 잔차 예측자들은 대응하는 참조 블록들 (316a 내지 316d) (BaseTRef)로부터 참조 블록들 (306a 내지 306d) (Base)을 감산함으로써 결정될 수도 있다. 다음으로, ARP는 서브-블록들 (300a 내지 300d)의 각각에 대해 수행될 수도 있다.

[0219] 도 15는 블록-기반 인터-뷰 ARP를 예시하는 개념도이다. 도 15에서 도시된 바와 같이, 현재의 픽처 (352)는 4개의 서브-블록들 (350, 350b, 350c, 및 350d)로 분할된 현재의 블록 (350) (Curr)을 포함한다. 디스패리티 모션 벡터 (360) (DMV)는 현재의 블록 (350)에 대해 인터-뷰 예측을 수행하기 위하여 이용된 디스패리티 모션 벡터이다. 디스패리티 모션 벡터 (360)는 참조 픽처 (358)에서, 서브-블록들 (356a 내지 356d)을 포함하는 참조 블록 (356) (Base)을 지시한다. 현재의 픽처 (352) 및 참조 픽처 (358)는 동일한 시간 인스턴스 내에 있지만, 상이한 뷰들 내에 있다.

[0220] 블록-기반 인터-뷰 ARP를 위하여, 디폴트 유도된 모션 벡터는 서브-블록들 (350a 내지 350d)의 각각을 위하여 이용된다. 인터-뷰 ARP를 위하여, 디폴트 유도된 모션 벡터는 도 15에서의 i 번째 서브-블록에 대한 mvLX[ i ]에 의해 나타낸 모션 벡터이고, 현재의 ARP에서와 같이, 서브-블록들 (356a 내지 356d)의 각각의 서브-블록의 중심 위치를 커버하는 시간적 모션 벡터로서 설정될 수도 있다. 즉, 서브-블록들 (356)내의 i 번째 8x8 블록의 중심 위치를 커버하는 블록은 시간적 모션 벡터를 포함하고, mvLX[ i ]는 그 시간적 모션 벡터가 되도록 업데이트된다.

[0221] 유도된 모션 벡터들의 각각은 참조 뷰 (368)에서의 특정한 참조 블록 (366a 내지 366d) (BaseTref)을 지시한다. 예를 들어, 모션 벡터 (354) (mvLX[0])는 참조 블록 (368a)을 지시하고, 모션 벡터 (355) (mvLX[3])는 참조 블록 (366d)을 지시한다.

[0222] 일단 참조 블록들 (366a 내지 366d)의 각각이 식별되었으면, 디스패리티 모션 벡터 (360)는 참조 픽처 (364)에서의 참조 블록들 (362a 내지 362d) (CurrTRef)을 구하기 위하여 이용될 수도 있다. 참조 픽처 (364)는 현재의 픽처 (352)와는 상이한 시간 인스턴스 내에 있다. 다음으로, 잔차 예측자들은 대응하는 참조 블록들 (366a 내지 366d) (BaseTref)로부터 참조 블록들 (362a 내지 362d) (CurrTref)을 감산함으로써 결정될 수도 있다. 다음으로, ARP는 서브-블록들 (350a 내지 350d)의 각각에 대해 수행될 수도 있다.

[0223] 위에서 설명된 바와 같이, 블록-기반 시간적 ARP를 위하여, 모션 벡터 (310)는 참조 블록 (312) (CurrTref)을 위치시키기 위하여 액세스되고 이용된다. 마찬가지로, 블록-기반 인터-뷰 ARP를 위하여, 디스패리티 모션 벡터 (360)는 참조 블록 (356) (Base)을 위치시키기 위하여 액세스되고 이용된다.

[0224] 도 16은 서브-PU 병합 후보를 갖는 블록-기반 ARP를 예시하는 개념도이다. 서브-PU 인터-뷰 모션 예측이 인에이블될 때, NBDV/DoNBDV 유도 프로세스로부터의 유도된 디스패리티 벡터 (410)에 의해 식별된 하나의 참조 블록 (406)의 모션 정보는 서브-PU 병합 후보를 유도하기 위하여 이용된다. 즉, 블록 (400) (Curr)내의 각각의 서브-PU에 대하여, 서브-PU 병합 후보가 결정된 후, 그것은 도 14에서 도시된 바와 같이, 모션 벡터 (404) (mvLX[0]) 및 모션 벡터 (405) (mvLX[1])에 의해 나타낸 바와 같은 그 시간적 모션 정보를 가질 것이다. 모션 벡터들 (404 및 405)은 참조 블록들 (412) (CurrTref) 및 참조 블록들 (416) (BaseTref)을 식별하기 위하여 이용될 수도 있다.

[0225] ARP 프로세스가 소환될 때, 참조 블록 (412) (CurrTRef)내의 각각의 서브-블록 (예컨대, 8x8 블록)의 모션 정보가 또한 액세스된다. 대응하는 서브-블록 (412a 내지 412d) (CurrRef)이 디스패리티 모션 벡터와 연관될 때, 디스패리티 모션 벡터는 참조 뷰에서의 참조 블록 (예컨대, 블록 (406))을 위치시키기 위하여 이용될 수도 있다.

[0226] 그러므로, 2개의 블록들의 모션 정보가 액세스될 필요가 있을 수도 있다. 즉, NBDV/DoNBDV 프로세스로부터 DV에 의해 식별된 하나의 블록에 대한 모션 정보는 서브-PU 인터-뷰 병합 후보에 대해 액세스된다. 게다가, 모션 정보는 임의의 유도된 시간적 모션 정보에 의해 식별된 블록에 대해 액세스된다.

[0227] 3D-HEVC에서의 ARP를 위한 상기 언급된 기법들은 몇몇 단점들을 나타낸다. 일 예로서, 양자의 서브-PU 인터-뷰 병합 예측 및 블록-레벨 시간적 ARP가 하나의 PU를 코딩하기 위하여 이용될 때, 2개의 참조 블록들의

모션 정보가 액세스된다. 하나는 DoNBDV/NBDV 유도 프로세스로부터 유도된 디스패리티 벡터에 의해 식별되는 참조 뷰에서의 참조 블록이다. 게다가, 대응하는 모션 정보는 서브-PU 인터-뷰 병합 후보를 유도하기 위하여 액세스된다. 서브-PU 인터-뷰 병합 후보가 유도된 후, 시간적 참조 픽처에서의 또 다른 블록은 시간적 참조 픽처에서의 블록이 디스패리티 모션 벡터를 포함하는지 여부를 검사하기 위하여 액세스된다. 상이한 블록들과 연관된 모션 정보의 이중 액세스는 비디오 디코더 설계의 복잡도를 상당히 증가시키고, 디코더 스루풋 (throughput) 을 감소시킬 수도 있다.

[0228] 또 다른 단점으로서, 서브-PU (즉, 블록-레벨) ARP 가 이용될 때, 현재의 블록의 시간적 모션 벡터에 의해 지시된 참조 블록과 연관된 디스패리티 모션 벡터들은 디폴트 디스패리티 벡터를 업데이트하기 위하여 이용된다.

하나의 서브-블록 블록에 대하여, 서브-블록이 그 이웃하는 블록들 (좌측, 상부, 하부, 또는 우측의 어느 하나) 과 동일한 디스패리티 모션 벡터를 가지더라도, ARP 프로세스는 각각의 서브-블록에서 여전히 수행되고, 이에 따라, 비디오 디코더 복잡도를 증가시킨다.

[0229] 이 개시물은 비디오 디코더 복잡도를 감소시키기 위하여 ARP 를 위한 상기 언급된 문제를 해결하기 위한 다양한 예시적인 기법들을 제안한다. 이하에서 열거된 기법들의 각각은 ARP 와, ARP 를 위한 현재의 제안들에 관한 다른 연관된 비디오 코딩 기법들을 수행하기 위하여 필요하게 된 메모리 액세스들의 수를 감소시킨다.

[0230] 개시물의 일 예에서, 서브-PU 인터-뷰 모션 예측이 인에이블되고, (시간적 모션 정보에 대응하는) 서브-PU 인터-뷰 병합 후보가 현재의 PU 에 적용될 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록-레벨 ARP 를 디스에이블하도록 구성될 수도 있다. 그 대신에, PU-레벨 ARP 가 인에이블될 수도 있다.

[0231] 서브-PU 인터-뷰 모션 예측이 인에이블되고, PU-레벨 ARP 가 적용될 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각의 서브-PU 에 대한 시간적 모션 정보를 결정할 수도 있다. 즉, 각각의 서브-PU 는 그 자신의 시간적 모션 정보를 가진다. 그러나, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 모든 서브-PU 들에 대해 동일한 디스패리티 벡터를 결정한다. 시간적 모션 정보 및 디스패리티 벡터는 위에서 설명된 바와 같이, 잔차 및 잔차 예측자를 유도하기 위하여 이용된다. 서브-PU 인터-뷰 모션 예측이 적용될 때, 이용된 ARP 프로세스는 시간적 ARP 인 것에 주목해야 한다.

[0232] 서브-PU 인터-뷰 모션 예측이 이용되지 않을 때, 다음의 예시적인 기법들이 이용된다. 일 예에서, 시간적 ARP 를 위하여 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각의 서브-PU 에 대한 디스패리티 벡터를 결정할 수도 있다. 일 예에서, 결정된 디스패리티 벡터는 시간적 참조 픽처에서의 현재의 서브-PU 의 시간적 모션 정보에 의해 식별된 현재의 서브-PU 의 참조 블록으로부터 유도된 디스패리티 모션 정보일 수도 있다. 인터-뷰 ARP 를 위하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각의 서브-PU 에 대한 시간적 모션 정보를 결정할 수도 있다. 일 예에서, 시간적 모션 정보는 인터-뷰 참조 픽처에서의 현재의 서브-PU 의 디스패리티 모션 정보에 의해 식별된 현재의 서브-PU 의 참조 블록으로부터 유도될 수도 있다.

[0233] 개시물의 또 다른 예에서, 서브-PU 인터-뷰 모션 예측이 인에이블될 때, 비디오 인코더 (20) 및 비디오 디코더 는 연관된 참조 픽처가 시간적 참조 픽처일 경우에 특정한 참조 픽처 리스트에 대응하는 하나의 예측 방향에 대한 블록-레벨 ARP 를 디스에이블하도록 구성될 수도 있다. 이 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이 예측 방향에 대한 PU-레벨 ARP 를 인에이블하기만 하도록 구성될 수도 있다.

[0234] 일 예에서는, 다음의 프로세스가 적용된다. 현재의 PU 가 인터-뷰 병합 후보를 이용할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각의 서브-PU 에 대한 시간적 모션 정보를 결정한다. 그러나, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 모든 서브-PU 들에 대해 동일한 디스패리티 벡터를 결정한다. 시간적 모션 정보 및 디스패리티 벡터는 위에서 설명된 바와 같이, 잔차 및 잔차 예측자를 유도하기 위하여 이용된다.

[0235] 이와 다르게, 현재의 PU 가 다른 이용가능한 병합 후보들 중의 하나 (즉, 인터-뷰 병합 후보가 아님) 를 이용할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 PU-레벨 시간적 ARP 를 적용하고, 현재의 PU 내의 모든 블록들은, 대응하는 참조 픽처가 시간적 참조 픽처일 경우에 하나의 예측 방향에 대하여, 동일한 모션 정보를 공유한다. 하나의 예측 방향에 대하여, 대응하는 참조 픽처가 인터-뷰 참조 픽처일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 PU-레벨 인터-뷰 ARP 를 이용하고, 현재의 PU 내의 모든 블록들은 동일한 모션 정보를 공유한다. 이 경우, 블록-레벨 ARP 가 또한 적용될 수도 있고, 현재의 PU 내의 블록들은 동일한 디스패리티 모션 정보 및 상이한 시간적 모션 정보를 공유할 수도 있다.

[0236] 개시물의 또 다른 예에서, 블록-레벨 ARP 가 인에이블될 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 모션 정보에 기초하여 ARP 를 수행하기 위한 블록 사이즈를 결정할 수도 있다. 일 예에서, 특정한 참조 픽처

리스트에 대응하는 하나의 예측 방향에 대하여, 대응하는 참조 픽처가 시간적 참조 픽처일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30)는 블록-레벨 ARP를 이용할 수도 있다. 이 경우, 현재의 블록은 그 이웃하는 블록 (예컨대, 좌측, 상부, 하부, 및/또는 우측 이웃하는 블록들)과 동일한 디스패리티 모션 정보를 가진다.

또한, 현재의 블록 및 이웃하는 블록들은 함께 병합되고, ARP는 병합된 블록들에 대해 한 번 수행된다.

[0237] 개시물의 또 다른 예에서, 참조 픽처 리스트에 대응하는 하나의 예측 방향에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30)는 대응하는 참조 픽처가 인터-뷰 참조 픽처일 경우에 블록-레벨 ARP를 이용할 수도 있다.

이 경우, 현재의 블록은 그 이웃하는 블록들 (예컨대, 좌측, 상부, 하부, 및/또는 우측 이웃하는 블록들)과 동일한 시간적 모션 정보를 가진다. 또한, 현재의 블록 및 이웃하는 블록들은 함께 병합되고, ARP는 병합된 블록들에 대해 한 번 수행된다.

[0238] 도 17은 이 개시물에서 설명된 기법들을 구현하도록 구성될 수도 있는 예시적인 비디오 인코더 (20)를 예시하는 블록도이다. 비디오 인코더 (20)는 비디오 슬라이스들 내의 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라 코딩은 소정의 비디오 프레임 또는 픽처 내의 비디오에 있어서의 공간적 중복성을 감소시키거나 제거하기 위하여 공간적 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내의 비디오에 있어서의 중복성을 감소시키거나 제거하기 위하여 시간적 또는 인터-뷰 예측에 의존한다. 인트라-모드 (I 모드)는 몇몇 공간 기반 압축 모드들 중의 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향-예측 (B 모드)과 같은 인터-모드들은 몇몇 시간 기반 압축 모드들 중의 임의의 것을 포함할 수도 있다.

[0239] 도 17의 예에서, 비디오 인코더 (20)는 비디오 데이터 메모리 (235), 예측 프로세싱 유닛 (241), 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) (264), 합산기 (251), 변환 프로세싱 유닛 (252), 양자화 프로세싱 유닛 (255), 및 엔트로피 인코딩 유닛 (257)을 포함한다. 예측 프로세싱 유닛 (241)은 모션 추정 유닛 (242), 모션 및 디스패리티 보상 유닛 (244), 진보된 잔차 예측 (ARP) 유닛 (245), 및 인트라 예측 프로세싱 유닛 (246)을 포함한다. 비디오 블록 재구성을 위하여, 비디오 인코더 (20)는 또한, 역양자화 프로세싱 유닛 (259), 역변환 프로세싱 유닛 (260), 및 합산기 (262)를 포함한다. 디블록킹 필터 (deblocking filter; 도 17에서 도시되지 않음)는 또한, 블록 경계들을 필터링하여, 재구성된 비디오로부터 블록화 아티팩트 (blockiness artifact)들을 제거하기 위하여 포함될 수도 있다. 희망하는 경우, 디블록킹 필터는 전형적으로 합산기 (262)의 출력을 필터링할 것이다. (루프 내의 또는 루프 이후의) 추가적인 루프 필터들은 또한, 디블록킹 필터에 추가하여 이용될 수도 있다.

[0240] 다양한 예들에서, 비디오 인코더 (20)의 하나 이상의 하드웨어 유닛들은 이 개시물의 기법들을 수행하도록 구성될 수도 있다. 예를 들어, 모션 및 디스패리티 보상 유닛 (244) 및 ARP 유닛 (245)은 이 개시물의 기법들을 단독으로, 또는 비디오 인코더 (20)의 다른 유닛들과 조합하여 수행할 수도 있다.

[0241] 도 17에서 도시된 바와 같이, 비디오 인코더 (20)는 인코딩되어야 할 비디오 프레임 (예컨대, 텍스처 이미지 또는 심도 맵) 내의 비디오 데이터 (예컨대, 휘도 블록, 크로미넌스 블록, 또는 심도 블록과 같은 비디오 데이터의 블록)를 수신한다. 비디오 데이터 메모리 (235)는 비디오 인코더 (20)의 컴포넌트들에 의해 인코딩되어야 할 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (40)에서 저장된 비디오 데이터는 예를 들어, 비디오 소스 (18)로부터 획득될 수도 있다. DPB (264)는 (예컨대, 인트라 또는 인터 예측 코딩 모드들로서 또한 지정된, 인트라 또는 인터 코딩 모드들에서) 비디오 인코더 (20)에 의해 비디오 데이터를 인코딩함에 있어서 이용하기 위한 참조 비디오 데이터를 저장하는 메모리 버퍼이다. 비디오 데이터 메모리 (235) 및 DPB (264)는 동기식 DRAM (synchronous DRAM; SDRAM), 자기저항성 RAM (magnetoresistive RAM; MRAM), 저항성 RAM (resistive RAM; RRAM), 또는 다른 타입들의 메모리 디바이스들을 포함하는 동적 랜덤 액세스 메모리 (dynamic random access memory; DRAM)와 같은 다양한 메모리 디바이스들 중의 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (235) 및 DPB (264)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (235)는 비디오 인코더 (20)의 다른 컴포넌트들과 함께 온-칩 (on-chip) 일 수도 있거나, 그러한 컴포넌트들에 관하여 오프-칩 (off-chip) 일 수도 있다.

[0242] 도 17에서 도시된 바와 같이, 비디오 인코더 (20)는 비디오 데이터를 수신하고, 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한, 예컨대, LCU 들 및 CU 들의 큐드트리 구조에 따른, 비디오 블록 파티셔닝 뿐만 아니라, 슬라이스 (slice) 들, 타일 (tile) 들, 또는 다른 더 큰 유닛들로의 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20)는 인코딩되어야 할 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트

들을 일반적으로 예시한다. 슬라이스는 다수의 비디오 블록들로 (그리고 아마도 타일들로서 지칭된 비디오 블록들의 세트들로) 분할될 수도 있다.

[0243] 예측 프로세싱 유닛 (241)은 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨)에 기초하여, 현재의 비디오 블록에 대하여, 복수의 인트라 코딩 모드들 중의 하나 또는 복수의 인터 코딩 모드들 중의 하나와 같은 복수의 가능한 코딩 모드들 중의 하나를 선택할 수도 있다. 예측 프로세싱 유닛 (241)은 결과적인 인트라-코딩된 또는 인터-코딩된 블록을, 잔차 블록 데이터를 생성하기 위하여 합산기 (251)에, 그리고 참조 픽처로서의 이용을 위한 인코딩된 블록을 재구성하기 위하여 합산기 (262)에 제공할 수도 있다.

[0244] 예측 프로세싱 유닛 (241) 내의 인트라 예측 프로세싱 유닛 (246)은 공간적 압축을 제공하기 위하여, 코딩되어야 할 현재의 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 관하여 현재의 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다. 예측 프로세싱 유닛 (241) 내의 모션 추정 유닛 (242) 및 모션 및 디스패리티 보상 유닛 (244)은 예컨대, 시간적 및/또는 인터-뷰 압축을 제공하기 위하여, (인터-뷰 참조 픽처들을 포함하는) 하나 이상의 참조 픽처들 내의 하나 이상의 예측 블록들에 관하여 현재의 비디오 블록의 (인터-뷰 코딩을 포함하는) 인터-예측 코딩을 수행한다.

[0245] 모션 추정 유닛 (242)은 비디오 시퀀스에 대한 미리 결정된 패턴에 따라 비디오 슬라이스에 대한 인터-예측 모드를 결정하도록 구성될 수도 있다. 모션 추정 유닛 (242) 및 모션 및 디스패리티 보상 유닛 (244)은 고도로 통합될 수도 있지만, 개념적인 목적들을 위하여 별도로 예시되어 있다. 모션 추정 유닛 (242)에 의해 수행된 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 예를 들어, 모션 벡터는 참조 픽처 내에서의 예측 블록에 관하여 현재의 비디오 프레임 또는 픽처 내에서의 비디오 블록의 PU의 변위를 표시할 수도 있다.

[0246] 예측 블록은, 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는 픽셀 차이의 측면에서, 코딩되어야 할 비디오 블록의 PU와 근접하게 일치시키기 위하여 구해지는 블록이다. 일부의 예들에서, 비디오 인코더 (20)는 DPB (264) 내에 저장된 참조 픽처들의 정수 미만 (sub-integer) 픽셀 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20)는 참조 픽처의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 보간할 수도 있다. 그러므로, 모션 추정 유닛 (242)은 전체 픽셀 위치들 및 분수 픽셀 위치들에 관하여 모션 검색을 수행할 수도 있고, 분수 픽셀 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0247] 모션 추정 유닛 (242)은 PU의 위치를 (시간적 또는 인터-뷰 참조 픽처를 포함하는) 참조 픽처의 예측 블록의 위치와 비교함으로써, 인터-코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터 및/또는 디스패리티 모션 벡터를 계산한다. 위에서 설명된 바와 같이, 모션 벡터는 모션-보상된 예측을 위하여 이용될 수도 있는 반면, 디스패리티 모션 벡터는 디스패리티-보상된 예측을 위하여 이용될 수도 있다. 참조 픽처는 제 1 참조 픽처 리스트 (List 0 또는 RefPicList0) 또는 제 2 참조 픽처 리스트 (List 1 또는 RefPicList1)로부터 선택될 수도 있고, 이들의 각각은 DPB (264) 내에 저장된 하나 이상의 참조 픽처들을 식별한다. 모션 추정 유닛 (242)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (257) 및 모션 및 디스패리티 보상 유닛 (244)으로 전송한다.

[0248] 모션 및/또는 디스패리티 보상 유닛 (244)에 의해 수행된 모션 및/또는 디스패리티 보상은 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페치 (fetch)하거나 생성하여, 서브-픽셀 정밀도 (sub-pixel precision)로의 보간들을 아마도 수행하는 것을 수반할 수도 있다. 현재의 비디오 블록의 PU에 대한 모션 벡터를 수신할 시에, 모션 및 디스패리티 보상 유닛 (244)은 모션 벡터가 참조 픽처 리스트들 중의 하나에서 지시하는 예측 블록을 위치시킬 수도 있다. 비디오 인코더 (20)는 코딩되고 있는 현재의 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 픽셀 차이 값들은 블록에 대한 잔차 데이터를 형성하고, 루마 및 크로마 차이 컴포넌트들의 양자화를 포함할 수도 있다. 합산기 (251)는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

모션 및 디스패리티 보상 유닛 (244)은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩할 시에 비디오 디코더 (30)에 의한 이용을 위하여 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0249] ARP 유닛 (245) 및 모션 및 디스패리티 보상 유닛 (244)을 포함하는 비디오 인코더 (20)는 양방향 예측 및 ARP 기법들, 예컨대, 본원에서 설명된 인터-뷰 또는 시간적 ARP 기법들 중의 임의의 것을 수행할 수도 있다. 특히, 개시물의 일 예에서, 비디오 인코더는 양방향 예측 및 인터-뷰 ARP를 이용하여 비디오 데이터의 현재

의 블록을 인코딩하도록 구성될 수도 있다. 비디오 데이터의 현재의 블록에 대하여, 모션 및 디스패리티 보상 유닛 (244)은 비디오 데이터의 현재의 블록을 위한 제 1 예측 방향 (예컨대, 참조 픽처 리스트 X)에 대한 시간적 모션 정보를 결정하고, 제 1 예측 방향에 대해 결정된 시간적 모션 정보를 이용하여 제 2 예측 방향 (예컨대, 참조 픽처 리스트 Y)에 대한 참조 블록들을 식별하도록 구성될 수도 있고, 제 2 예측 방향에 대한 참조 블록들은 비디오 데이터의 현재의 블록과는 상이한 액세스 유닛 내에 있다. 이러한 방법으로, 현재의 비디오 블록을 인코딩하기 위하여, 모션 정보 및 참조 블록들의 더 적은 메모리 액세스들이 필요하게 된다.

[0250] 인트라 예측 프로세싱 유닛 (246)은 위에서 설명된 바와 같이, 모션 추정 유닛 (242) 및 모션 및 디스패리티 보상 유닛 (244)에 의해 수행된 인터-예측에 대한 대안으로서, 현재의 블록에 대해 인트라-예측을 수행할 수도 있다. 특히, 인트라 예측 프로세싱 유닛 (246)은 현재의 블록을 인코딩하기 위하여 이용하기 위한 인트라-예측 모드를 결정할 수도 있다. 일부의 예들에서, 인트라 예측 프로세싱 유닛 (246)은 예컨대, 별도의 인코딩 패스 (encoding pass)를 동안에 다양한 인트라-예측 모드들을 이용하여 현재의 비디오 블록을 인코딩할 수도 있고, 인트라 예측 프로세싱 유닛 (246) (또는 일부의 예들에서, 예측 프로세싱 유닛 (241))은 테스팅된 모드들로부터 이용하기 위한 적절한 인트라-예측 모드를 선택할 수도 있다.

[0251] 예를 들어, 인트라 예측 프로세싱 유닛 (246)은 다양한 테스팅된 인트라-예측 모드들에 대한 레이트-왜곡 분석 (rate-distortion analysis)을 이용하여 레이트-왜곡 값을 계산할 수도 있고, 테스팅된 모드들 중에서 최상의 레이트-왜곡 특성을 가지는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로, 인코딩된 블록을 생성하기 위해 이용된 비트 레이트 (즉, 비트들의 수) 뿐만 아니라, 인코딩된 블록과, 인코딩된 블록을 생성하기 위하여 인코딩되었던 원래의 인코딩되지 않은 블록과의 사이의 왜곡 (또는 에러)의 양을 결정한다. 인트라 예측 프로세싱 유닛 (246)은 어느 인트라-예측 모드가 블록에 대한 최상의 레이트-왜곡 값을 나타내는지를 결정하기 위하여 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율 (ratio)들을 계산할 수도 있다.

[0252] 어떤 경우에도, 블록에 대한 인트라-예측 모드를 선택한 후, 인트라 예측 프로세싱 유닛 (246)은 블록에 대한 선택된 인트라-예측 모드를 표시하는 정보를 엔트로피 인코딩 유닛 (257)에 제공할 수도 있다. 엔트로피 인코딩 유닛 (257)은 이 개시물의 기법들에 따라, 선택된 인트라-예측 모드를 표시하는 정보를 인코딩할 수도 있다. 비디오 인코더 (20)는 복수의 인트라-예측 모드 인덱스 표들 및 복수의 수정된 인트라-예측 모드 인덱스 표들 (또한 코드워드 맵핑 표 (codeword mapping table)들로서 지칭됨)을 포함할 수도 있는 송신된 비트 스트림 구성 데이터에서, 다양한 블록들에 대한 인코딩 컨텍스트들의 정의들과, 컨텍스트들의 각각에 대해 이용하기 위한 가장 가능성 있는 인트라-예측 모드, 인트라-예측 모드 인덱스 표, 및 수정된 인트라-예측 모드 인덱스 표의 표시들을 포함할 수도 있다.

[0253] 예측 프로세싱 유닛 (241)이 인터-예측 또는 인트라-예측의 어느 하나를 통해 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 인코더 (20)는 현재의 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 잔차 블록에서의 잔차 비디오 데이터는 하나 이상의 TU 들에 포함될 수도 있고 변환 프로세싱 유닛 (252)에 적용될 수도 있다. 변환 프로세싱 유닛 (252)은 이산 코사인 변환 (discrete cosine transform; DCT) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 잔차 비디오 데이터를 잔차 변환 계수들로 변환한다. 변환 프로세싱 유닛 (252)은 잔차 비디오 데이터를 픽셀 도메인으로부터, 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다.

[0254] 변환 프로세싱 유닛 (252)은 결과적인 변환 계수들을 양자화 프로세싱 유닛 (255)으로 전송할 수도 있다. 양자화 프로세싱 유닛 (255)은 비트 레이트를 추가로 감소시키기 위하여 변환 계수들을 양자화한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화도 (degree of quantization)는 양자화 파라미터를 조절함으로써 수정될 수도 있다. 일부의 예들에서, 다음으로, 양자화 프로세싱 유닛 (255)은 양자화된 변환 계수들을 포함하는 행렬 (matrix)의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (257)은 스캔을 수행할 수도 있다.

[0255] 양자화에 후속하여, 엔트로피 인코딩 유닛 (257)은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (257)은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 2 진 산술 코딩 (CABAC), 십액스-기반 컨텍스트-적응 2 진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (257)에 의한 엔트로피 인코딩에 후속하여, 인코딩된 비트스트림은 비디오 디코더 (30)로 송신될 수도 있거나, 비디오 디코더 (30)에 의한 더 이후의 송신 또는 취출을 위해 아카이빙 (archiving) 될 수도 있다. 엔트로피 인코딩 유닛

(257) 은 또한, 코딩되고 있는 현재의 비디오 슬라이스에 대한 모션 벡터들 및 다른 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0256] 역양자화 프로세싱 유닛 (259) 및 역변환 프로세싱 유닛 (260)은 참조 픽처의 참조 블록으로서의 더 이후의 이용을 위해 픽셀 도메인에서 잔차 블록을 재구성하기 위하여, 역양자화 및 역변환을 각각 적용한다. 모션 및 디스패리티 보상 유닛 (244)은 잔차 블록을 참조 픽처 리스트들 중의 하나의 참조 픽처 리스트 내의 참조 픽처들 중의 하나의 참조 픽처의 예측 블록에 추가함으로써 참조 블록을 계산할 수도 있다. 모션 및 디스패리티 보상 유닛 (244)은 또한, 모션 추정 시에 이용하기 위한 정수-미만 픽셀 값들을 계산하기 위하여 하나 이상의 보간 필터들을 재구성된 잔차 블록에 적용할 수도 있다. 합산기 (262)는 DPB (264)에서의 저장을 위한 참조 블록을 생성하기 위하여, 재구성된 잔차 블록을 모션 및 디스패리티 보상 유닛 (244)에 의해 생성된 모션 보상된 예측 블록에 추가한다. 참조 블록은 후속 비디오 프레임 또는 픽처에서 블록을 인터-예측하기 위하여, 모션 추정 유닛 (242) 및 모션 및 디스패리티 보상 유닛 (244)에 의해 참조 블록으로서 이용될 수도 있다.

[0257] 도 18은 이 개시물에서 설명된 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30)를 예시하는 블록도이다. 도 18의 예에서, 비디오 디코더 (30)는 비디오 데이터 메모리 (278), 엔트로피 디코딩 유닛 (280), 예측 프로세싱 유닛 (281), 역양자화 프로세싱 유닛 (286), 역변환 프로세싱 유닛 (288), 합산기 (291), 및 디코딩된 픽처 버퍼 (DPB; 292)를 포함한다. 예측 프로세싱 유닛 (281)은 모션 및 디스패리티 보상 유닛 (282), ARP 유닛 (283), 및 인트라 예측 프로세싱 유닛 (284)을 포함한다. 일부의 예들에서, 비디오 디코더 (30)는 도 17로부터의 비디오 인코더 (20)에 대하여 설명된 인코딩 패스와 일반적으로 상반되는 디코딩 패스를 수행할 수도 있다.

[0258] 다양한 예들에서, 비디오 디코더 (30)의 하나 이상의 하드웨어 유닛들은 이 개시물의 기법들을 수행하도록 구성될 수도 있다. 예를 들어, ARP 유닛 (283) 및 모션 및 디스패리티 보상 유닛 (282)은 이 개시물의 기법들을 단독으로, 또는 비디오 인코더의 다른 유닛들과 조합하여 수행할 수도 있다.

[0259] 비디오 데이터 메모리 (278)는 비디오 디코더 (30)의 컴포넌트들에 의해 디코딩되어야 할, 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (278) 내에 저장된 비디오 데이터는 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 카메라와 같은 로컬 비디오 소스로부터, 또는 물리적 데이터 저장 매체들을 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (278)는 인코딩된 비디오 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (coded picture buffer; CPB)를 형성할 수도 있다. DPB (292)는 (예컨대, 인트라 또는 인터 예측 코딩 모드들로서 또한 지정된, 인트라 또는 인터 코딩 모드들에서) 비디오 디코더 (30)에 의해 비디오 데이터를 디코딩함에 있어서 이용하기 위한 참조 비디오 데이터를 저장하는 DPB의 하나의 예이다. 비디오 데이터 메모리 (278) 및 DPB (292)는 동기식 DRAM (SDRAM), 자기저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들을 포함하는 동적 랜덤 액세스 메모리 (DRAM)와 같은 다양한 메모리 디바이스들 중의 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (278) 및 DPB (292)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (278)는 비디오 디코더 (30)의 다른 컴포넌트들과 함께 온-칩일 수도 있거나, 그러한 컴포넌트들에 관하여 오프-칩일 수도 있다.

[0260] 디코딩 프로세스 동안, 비디오 디코더 (30)는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 선택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20)로부터 수신한다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (280)은 양자화된 계수들, 모션 벡터들, 및 다른 선택스 엘리먼트들을 생성하기 위하여 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (280)은 모션 벡터들 및 다른 선택스 엘리먼트들을 예측 프로세싱 유닛 (281)으로 포워딩 (forwarding) 한다. 비디오 디코더 (30)는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 선택스 엘리먼트들을 수신할 수도 있다.

[0261] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (281)의 인트라 예측 프로세싱 유닛 (284)은 시그널링된 인트라 예측 모드와, 현재의 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B, 또는 P) 슬라이스 또는 인터-뷰 코딩된 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (281)의 모션 및 디스패리티 보상 유닛 (282)은 엔트로피 디코딩 유닛 (280)으로부터 수신된 모션 벡터들, 디스패리티 모션 벡터들, 및 다른 선택스 엘리먼트들에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 인터-뷰 참조 픽처들을 포함하는 참조 픽처 리스트들 중의 하나 내의 참조 픽처들 중의 하나로부터 생성될 수도 있다. 비디오 디코더 (30)는 DPB (292) 내에 저장

된 참조 픽처들에 기초하여, 디폴트 구성 기법들 또는 임의의 다른 기법을 이용하여 참조 프레임 리스트들, RefPicList0 및 RefPicList1 을 구성할 수도 있다.

[0262] 모션 및 디스패리티 보상 유닛 (282) 은 모션 벡터들 및 다른 신팩스 엘리먼트들을 파싱 (parsing) 함으로써 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 디코딩되고 있는 현재의 비디오 블록에 대한 예측 블록들을 생성하기 위하여 예측 정보를 이용한다. 예를 들어, 모션 및 디스패리티 보상 유닛 (282) 은 비디오 슬라이스의 비디오 블록들을 코딩하기 위하여 이용된 예측 모드 (예컨대, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 타입 (예컨대, B 슬라이스, P 슬라이스, 및/또는 인터-뷰 예측된 슬라이스), 슬라이스에 대한 참조 픽처 리스트들 중의 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들 및/또는 디스패리티 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재의 비디오 슬라이스에서의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위하여, 수신된 신팩스 엘리먼트들의 일부를 이용한다.

[0263] 모션 및 디스패리티 보상 유닛 (282) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 및 디스패리티 보상 유닛 (282) 은 참조 블록들의 정수-미만 픽셀들에 대한 보간된 값들을 계산하기 위하여, 비디오 블록들의 인코딩 동안에 비디오 인코더 (20) 에 의해 이용된 바와 같은 보간 필터들을 이용할 수도 있다.

이 경우, 모션 및 디스패리티 보상 유닛 (282) 은 수신된 신팩스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 이용된 보간 필터들을 결정할 수도 있고, 예측 블록들을 생성하기 위하여 보간 필터들을 이용할 수도 있다.

[0264] 역양자화 프로세싱 유닛 (286) 은, 비트스트림에서 제공되며 엔트로피 디코딩 유닛 (280) 에 의해 디코딩되는 양자화된 변환 계수들을 역양자화, 즉, 탈양자화(dequantize) 한다. 역양자화 프로세스는 적용되어야 할 양자화도 및, 마찬가지로, 역양자화도를 결정하기 위하여 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 이용을 포함할 수도 있다. 역변환 프로세싱 유닛 (288) 은 픽셀 도메인 (pixel domain) 에서 잔차 블록들을 생성하기 위하여, 역변환, 예컨대, 역 DCT, 역정수 변환 (inverse integer transform), 또는 개념적으로 유사한 역변환 프로세스를 변환 계수들에 적용한다.

[0265] ARP 유닛 (283) 및 모션 및 디스패리티 보상 유닛 (282) 을 포함하는 비디오 디코더 (30) 는 양방향 예측 및/또는 ARP 기법들, 예컨대, 본원에서 설명된 인터-뷰 또는 시간적 ARP 기법들 중의 임의의 것을 수행할 수도 있다.

특히, 개시물의 일 예에서, 비디오 디코더 (30) 는 양방향 예측 및 인터-뷰 ARP 를 이용하여 인코딩된 비디오 데이터의 현재의 블록을 수신할 수도 있다. 비디오 데이터의 인코딩된 블록은 비디오 데이터 메모리 (278) 내에 저장될 수도 있다. 비디오 데이터의 인코딩된 블록에 대하여, 모션 및 디스패리티 보상 유닛 (282) 은 비디오 데이터의 인코딩된 블록을 위한 제 1 예측 방향 (예컨대, 참조 픽처 리스트 X) 에 대한 시간적 모션 정보를 결정하고, 제 1 예측 방향에 대해 결정된 시간적 모션 정보를 이용하여 제 2 예측 방향 (예컨대, 참조 픽처 리스트 Y) 에 대한 참조 블록들을 식별하도록 구성될 수도 있고, 제 2 예측 방향에 대한 참조 블록들은 비디오 데이터의 현재의 블록과는 상이한 액세스 유닛 내에 있다. 이러한 방법으로, 인코딩된 비디오 블록을 디코딩하기 위하여, 모션 정보 및 참조 블록들의 더 적은 메모리 액세스들이 필요하게 된다.

[0266] 모션 및 디스패리티 보상 유닛 (282) 이 모션 벡터들 및 다른 신팩스 엘리먼트들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30) 는 역변환 프로세싱 유닛 (288) 으로부터의 잔차 블록들을, 모션 및 디스패리티 보상 유닛 (282) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (291) 는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

희망하는 경우, 디블록킹 필터는 또한, 블록화 아티팩트들을 제거하기 위하여 디코딩된 블록들을 필터링하도록 적용될 수도 있다. (코딩 루프 내에 또는 코딩 루프 이후 중의 어느 하나에서의) 다른 루프 필터들은 또한, 픽셀 천이 (pixel transitioni) 들을 평탄화하거나, 또는 이와 다르게 비디오 품질을 개선시키기 위하여 이용될 수도 있다. 다음으로, 소정의 프레임 또는 픽처에서의 디코딩된 비디오 블록들은, 후속 모션 보상을 위하여 이용되는 참조 픽처들을 저장하는 DPB (292) 내에 저장된다. DPB (292) 는 또한, 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에서의 더 이후의 프리젠테이션 (presentation) 을 위하여 디코딩된 비디오를 저장한다.

[0267] 도 19 는 이 개시물에서 설명된 기법들에 따라 비디오 블록을 인코딩하기 위한 예시적인 ARP 방법을 예시하는 흐름도이다. 도 19 의 기법들은 모션 및 디스패리티 보상 유닛 (244) 및 ARP 유닛 (245) 을 포함하는 비디오 인코더 (20) 의 하드웨어 구조들의 임의의 조합에 의해 수행될 수도 있다.

[0268] 개시물의 일 예에서, 비디오 인코더 (20) 는 ARP 및 양방향 예측을 이용하여 비디오 데이터의 블록을 인코딩하도록 구성될 수도 있다. 이 예에서, 양방향 예측은 제 1 예측 방향에 대한 (예컨대, 참조 픽처 리스트 X) 에

대한) 시간적 예측과, 제 2 예측 방향에 대한 (예컨대, 참조 블록 리스트 Y에 대한) 인터-뷰 예측을 포함한다.

모션 및 디스패리티 보상 유닛 (244)은 비디오 데이터의 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하도록 구성될 수도 있다 (1900). ARP 유닛 (245)은 결정된 시간적 모션 정보를 이용하여 제 1 예측 방향에 대한 참조 블록들을 식별하고 (1910), 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하도록 (1920) 구성될 수도 있다. 참조 블록들은 비디오 데이터의 블록에 대한 액세스 유닛과는 상이한 액세스 유닛 내에 있을 수도 있다. ARP 유닛 (245)은 제 1 예측 방향 및 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 블록에 대해 진보된 잔차 예측을 수행하도록 추가로 구성될 수도 있다 (1930).

[0269] 개시물의 다른 예들에서, 모션 및 디스패리티 보상 유닛 (244)은 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보를 결정하도록 구성될 수도 있다. 게다가, ARP 유닛 (245)은 결정된 시간적 모션 정보를 이용하여 제 1 예측 방향에 대한 제 1 참조 블록을 식별하도록 구성될 수도 있고, 제 1 참조 블록은 제 1 뷰의 제 2 액세스 유닛 내에 있다. ARP 유닛 (245)은 결정된 시간적 모션 정보를 이용하여 제 2 예측 방향에 대한 제 2 참조 블록을 식별하고, 결정된 시간적 모션 정보 및 결정된 디스패리티 모션 정보를 이용하여 제 2 예측 방향에 대한 제 3 참조 블록을 식별하도록 추가로 구성될 수도 있고, 제 3 참조 블록은 제 2 뷰의 제 3 액세스 유닛 내에 있다.

[0270] 도 20은 이 개시물에서 설명된 기법들에 따라 비디오 블록을 디코딩하기 위한 예시적인 ARP 방법을 예시하는 흐름도이다. 도 20의 기법들은 비디오 디코더, ARP 유닛 (283), 및 모션 및 디스패리티 보상 유닛 (282)의 하드웨어 구조들의 임의의 조합에 의해 수행될 수도 있다.

[0271] 개시물의 일 예에서, 비디오 디코더 (30)는 제 1 뷰의 제 1 액세스 유닛에서의 비디오 데이터의 제 1 인코딩된 블록을 저장하도록 구성될 수도 있고, 비디오 데이터의 제 1 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었다 (2000). 양방향 예측은 제 1 예측 방향에 대한 시간적 예측 및 제 2 예측 방향에 대한 인터-뷰 예측을 포함할 수도 있다.

[0272] 모션 및 디스패리티 보상 유닛 (282)은 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하도록 구성될 수도 있다 (2010). ARP 유닛 (283)은 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보를 결정하고 (2020), 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 식별하도록 (2030) 구성될 수도 있다. 참조 블록들은 제 1 액세스 유닛과는 상이한 액세스 유닛 내에 있을 수도 있다. ARP 유닛 (283)은 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행하도록 추가로 구성될 수도 있다 (2040).

[0273] 개시물의 또 다른 예에서, ARP 유닛 (238)은 결정된 시간적 모션 정보를 이용하여 제 1 예측 방향에 대한 참조 블록들을 식별하고, 제 1 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행하도록 구성될 수도 있다. ARP 유닛 (238)은 결정된 시간적 모션 정보를 이용하여 제 2 예측 방향에 대한 제 2 참조 블록을 식별하고, 결정된 시간적 모션 정보 및 결정된 디스패리티 모션 정보를 이용하여 제 2 예측 방향에 대한 제 3 참조 블록을 식별하도록 추가로 구성될 수도 있고, 제 3 참조 블록은 제 2 뷰의 제 3 액세스 유닛 내에 있다. 제 1 예측 방향의 제 1 참조 블록은 제 2 예측 방향의 제 2 참조 블록과 동일하다.

[0274] 개시물의 또 다른 예에서, 비디오 디코더 (30)는 제 1 예측 방향에 대한 식별된 참조 블록들 및 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록을 디코딩하도록 구성될 수도 있다.

[0275] 개시물의 또 다른 예에서, 비디오 디코더 (30)는 잔차 비디오 데이터를 생성하기 위하여, 블록-레벨 진보된 잔차 예측 또는 예측 유닛-레벨 진보된 잔차 예측 중의 하나를 이용하여 비디오 데이터의 제 1 인코딩된 블록을 디코딩하고, 비디오 데이터의 디코딩된 블록을 생성하기 위하여, 양방향 예측, 제 1 예측 방향에 대한 식별된 참조 블록, 및 제 2 예측 방향에 대한 식별된 참조 블록을 이용하여 잔차 데이터를 디코딩하도록 구성될 수도 있다.

[0276] 개시물의 또 다른 예에서, 비디오 디코더 (30)는 제 3 뷰의 제 4 액세스 유닛에서의 비디오 데이터의 제 2 인코딩된 블록을 저장하도록 추가로 구성될 수도 있고, 비디오 데이터의 제 2 인코딩된 블록은 진보된 잔차 예측 및 양방향 예측을 이용하여 인코딩되었다. 양방향 예측은 제 3 예측 방향에 대한 시간적 예측 및 제 4 예측

방향에 대한 인터-뷰 예측을 포함할 수도 있다.

[0277] 모션 및 디스패리티 보상 유닛 (282)은 비디오 데이터의 제 1 인코딩된 블록을 위한 제 1 예측 방향에 대한 시간적 모션 정보를 결정하도록 구성될 수도 있다. ARP 유닛 (283)은 결정된 시간적 모션 정보를 이용하여 제 1 예측 방향에 대한 참조 블록들을 식별하도록 구성될 수도 있다. ARP 유닛 (283)은 제 1 예측 방향에 대한 결정된 시간적 모션 정보를 이용하여, 제 1 예측 방향과는 상이한 제 2 예측 방향에 대한 참조 블록들을 추가로 식별할 수도 있고, 참조 블록들은 제 1 액세스 유닛과는 상이한 액세스 유닛 내에 있다. ARP 유닛 (283)은 또한, 제 1 예측 방향 및 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록에 대해 진보된 잔차 예측을 수행할 수도 있다.

[0278] 개시물의 또 다른 예에서, 모션 및 디스패리티 보상 유닛 (282)은 비디오 데이터의 제 1 인코딩된 블록을 위한 제 2 예측 방향에 대한 디스패리티 모션 정보를 결정하도록 구성될 수도 있다. ARP 유닛 (283)은 결정된 시간적 모션 정보를 이용하여 제 1 예측 방향에 대한 제 1 참조 블록을 식별하도록 구성될 수도 있고, 제 1 참조 블록은 제 1 뷰의 제 2 액세스 유닛 내에 있다. ARP 유닛 (283)은 결정된 시간적 모션 정보 및 결정된 디스패리티 모션 정보를 이용하여 제 2 예측 방향에 대한 제 2 참조 블록을 식별하도록 추가로 구성될 수도 있고, 제 2 참조 블록은 제 2 뷰의 제 3 액세스 유닛 내에 있다.

[0279] 개시물의 또 다른 예에서, 비디오 디코더 (30)는 제 1 예측 방향에 대한 식별된 참조 블록들 및 제 2 예측 방향에 대한 식별된 참조 블록들을 이용하여 비디오 데이터의 제 1 인코딩된 블록을 디코딩하도록 구성될 수도 있다. 비디오 디코더 (30)는 잔차 비디오 데이터를 생성하기 위하여, 블록-레벨 진보된 잔차 예측 또는 예측 유닛-레벨 진보된 잔차 예측 중의 하나를 이용하여 비디오 데이터의 제 1 인코딩된 블록을 디코딩하고, 비디오 데이터의 디코딩된 블록을 생성하기 위하여, 양방향 예측, 제 1 예측 방향에 대한 식별된 참조 블록, 및 제 2 예측 방향에 대한 식별된 참조 블록을 이용하여 잔차 데이터를 디코딩하도록 추가로 구성될 수도 있다.

[0280] 일부의 예들에서, 이 개시물에서 설명된 기법들 중의 하나 이상의 양태들은 미디어 인지 네트워크 엘리먼트 (media aware network element; MANE), 스트림 적응 프로세서, 스플라이싱 프로세서 (splicing processor), 또는 편집 프로세서와 같은 중간 네트워크 디바이스에 의해 수행될 수도 있다. 예를 들어, 이러한 중간 디바이스는 이 개시물에서 설명된 바와 같은 다양한 시그널링 중의 임의의 것을 생성하거나 수신하도록 구성될 수도 있다.

[0281] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현될 경우, 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 컴퓨터-판독가능 매체 상에서 송신될 수도 있고, 하드웨어-기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터-판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터-판독가능 저장 매체들, 또는 예전대, 통신 프로토콜에 따라 하나의 장소로부터 또 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비-일시적 유형의 컴퓨터-판독가능 저장 매체들, 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스 될 수 있는 임의의 이용가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0282] 제한이 아닌 예로서, 이러한 컴퓨터-판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 희망하는 프로그램 코드를 저장하기 위해 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속은 컴퓨터-판독가능 매체로 적절하게 칭해진다. 예를 들어, 동축 케이블, 광섬유 케이블, 트위스트 페어 (twisted pair), 디지털 가입자 회선 (digital subscriber line; DSL), 또는 적외선, 라디오 (radio), 및 마이크로파 (microwave)와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신될 경우, 동축 케이블, 광섬유 케이블, 트위스트 페어, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 반송파들, 신호들, 또는 다른 순시적 매체 (transient medium)들을 포함하는 것이 아니라, 그 대신에, 비-순시적인, 유형의 저장 매체들에 관한 것이라는 것을 이해해야 한다. 본원에서 이용된 바와 같은 디스크 (disk) 및 디스크 (disc)는 컴팩트 디스크 (compact disc; CD), 레이저 디스크 (laser disc), 광학 디스크 (optical disc), 디지털 다기능 디스크

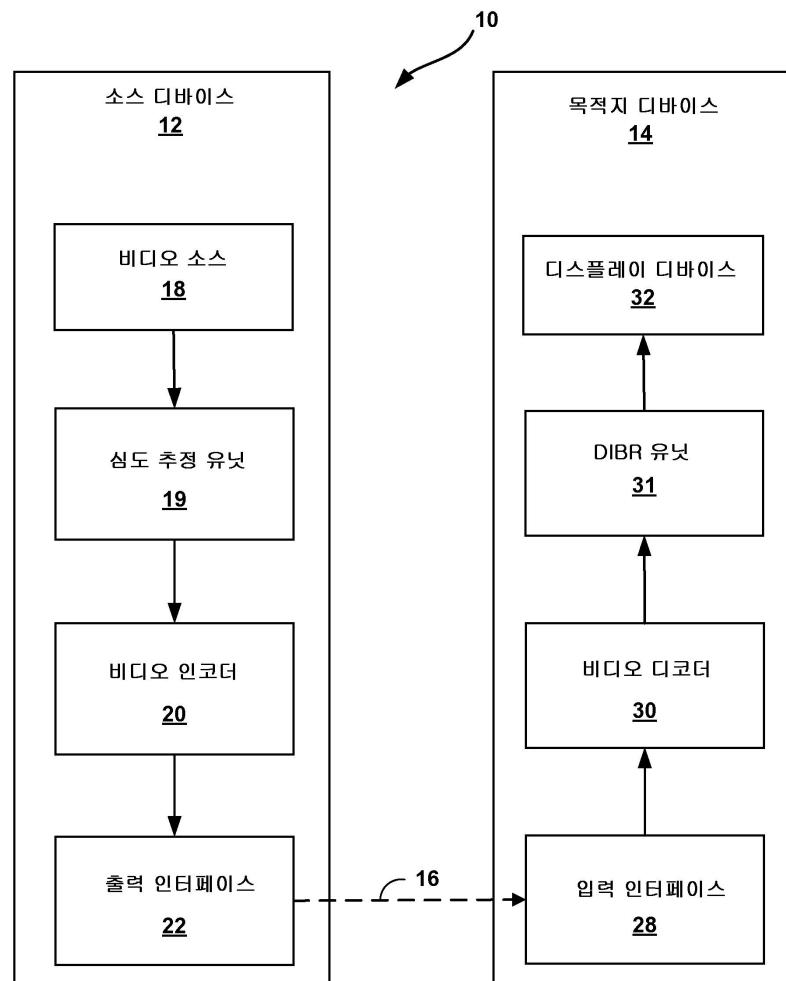
(digital versatile disc; DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (Blu-ray disc) 를 포함하고, 여기서, 디스크 (disk) 들은 통상 데이터를 자기적으로 재생하는 반면, 디스크 (disc) 들은 데이터를 레이저들로 광학적으로 재생한다. 상기의 조합들은 또한 컴퓨터-판독가능 매체들의 범위 내에 포함되어야 한다.

[0283] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍 가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적된 또는 별도의 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 이용된 바와 같은 용어 "프로세서"는 상기한 구조, 또는 본원에서 설명된 기법들의 구현을 위해 적당한 임의의 다른 구조 중의 임의의 것을 지칭할 수도 있다. 게다가, 일부의 양태들에서는, 본원에서 설명된 기능성이 인코딩 및 디코딩을 위해 구성되거나 조합된 코덱 (codec) 내에 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에서 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

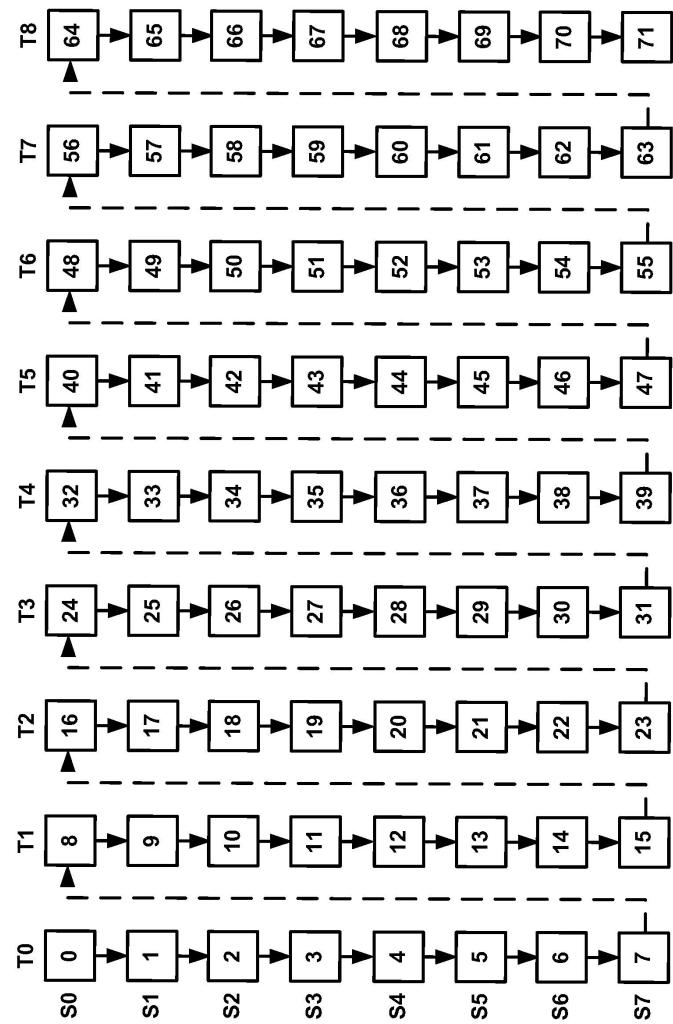
[0284] 이 개시물의 기법들은 무선 핸드셋 (wireless handset), 집적 회로 (integrated circuit; IC) 또는 IC 들의 세트 (예를 들어, 칩셋) 를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위하여 이 개시물에서 설명되어 있지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛 내에 조합될 수도 있거나, 적당한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는 상호동작하는 하드웨어 유닛들의 집합에 의해 제공될 수도 있다. 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

## 도면

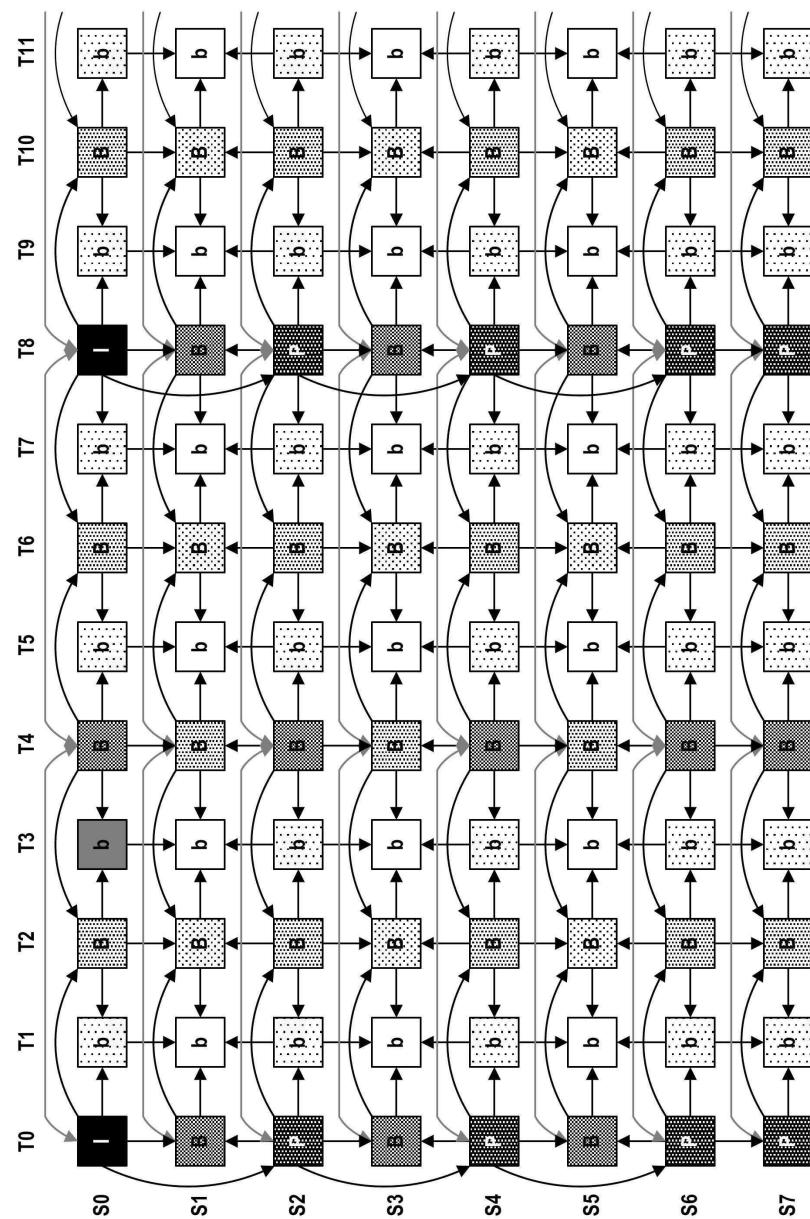
### 도면1



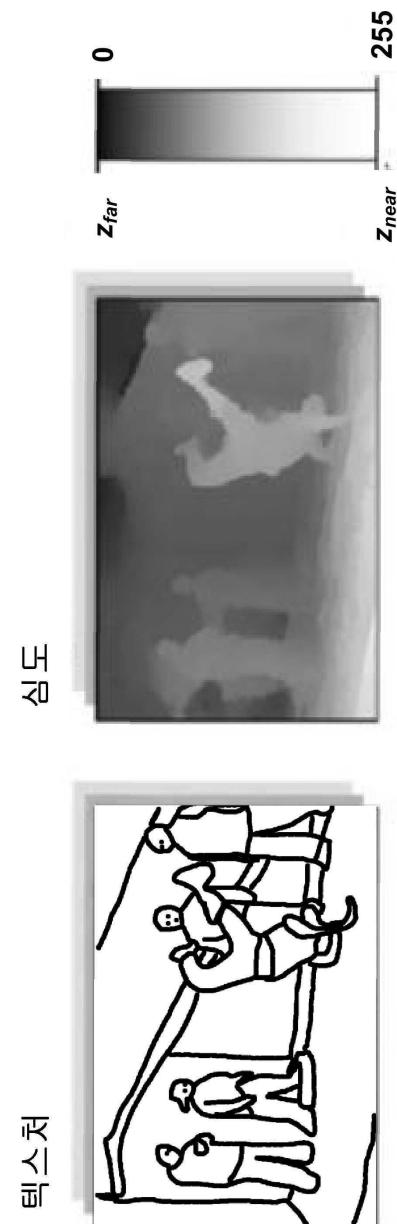
## 도면2



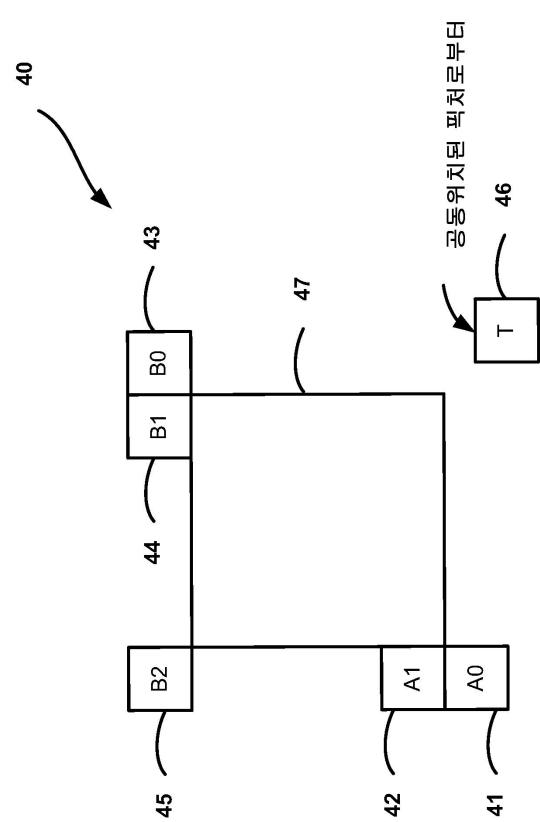
도면3



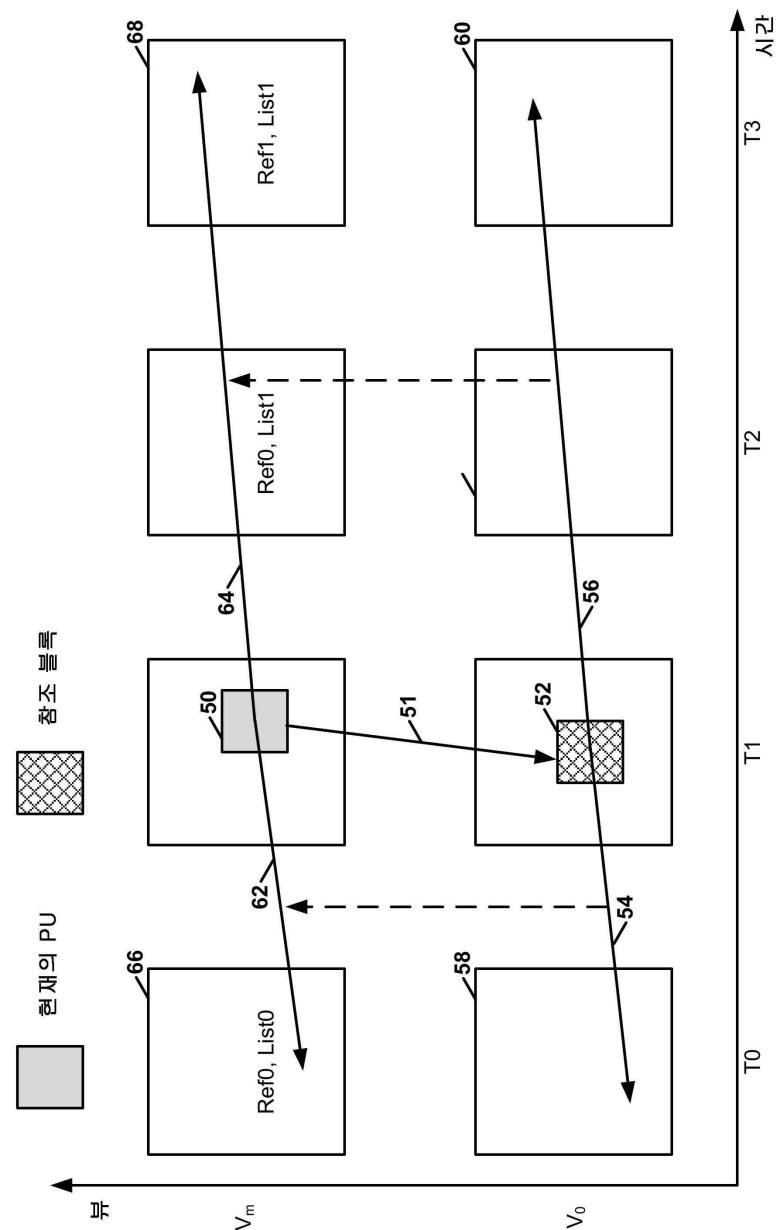
도면4



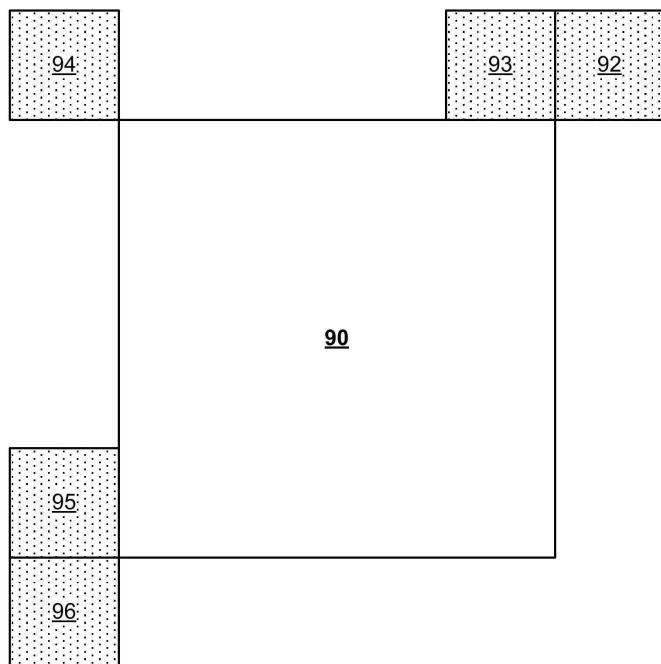
도면5



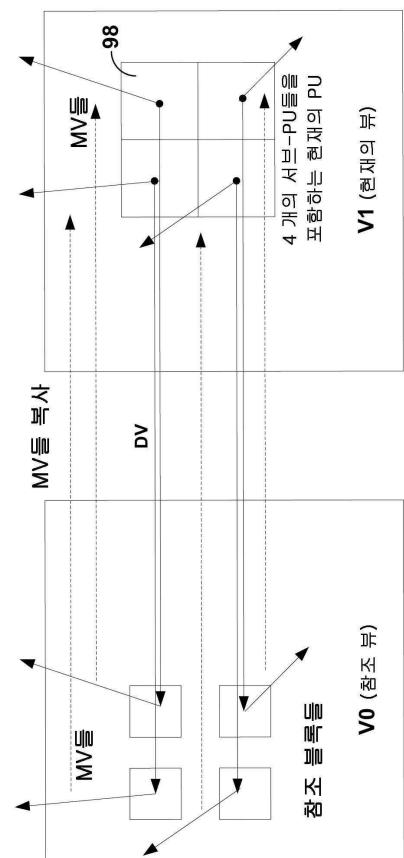
도면6



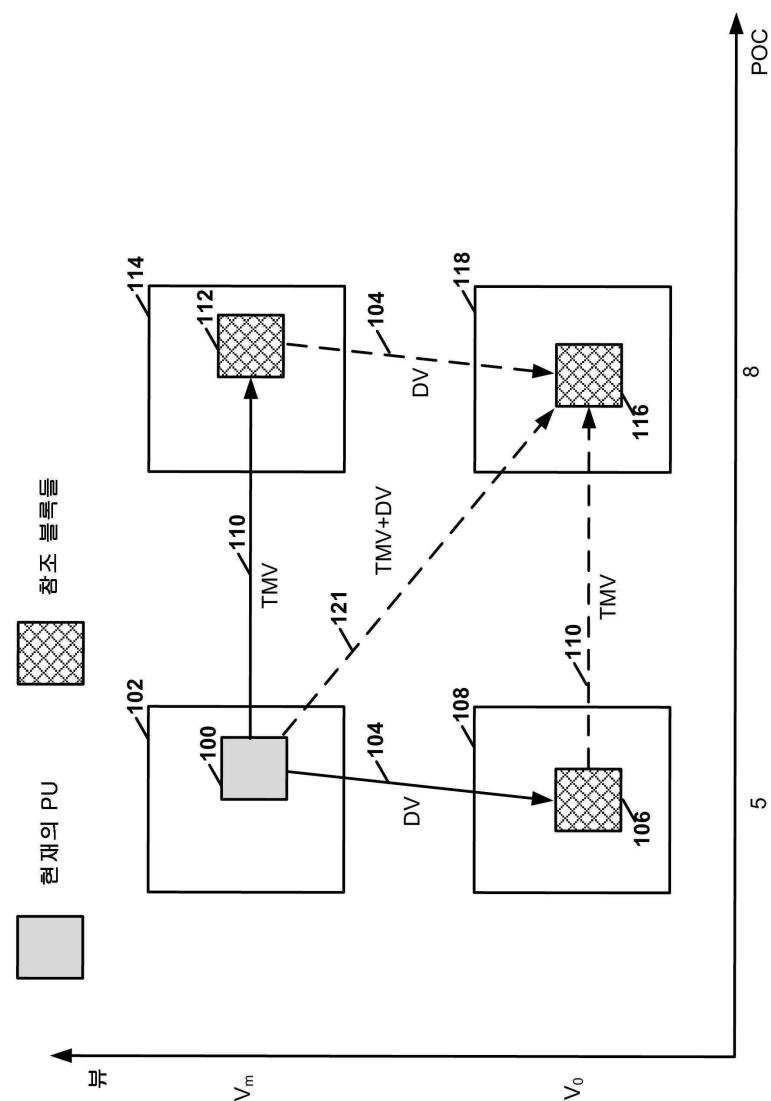
도면7



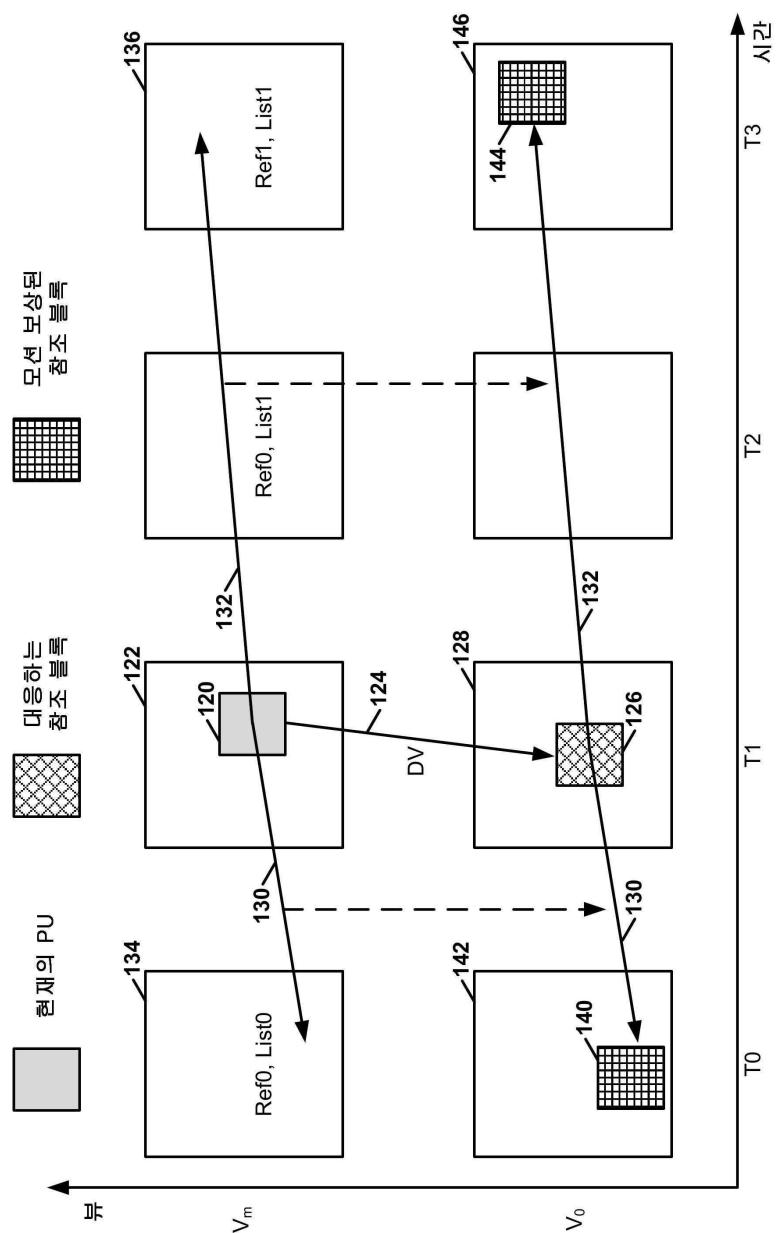
도면8



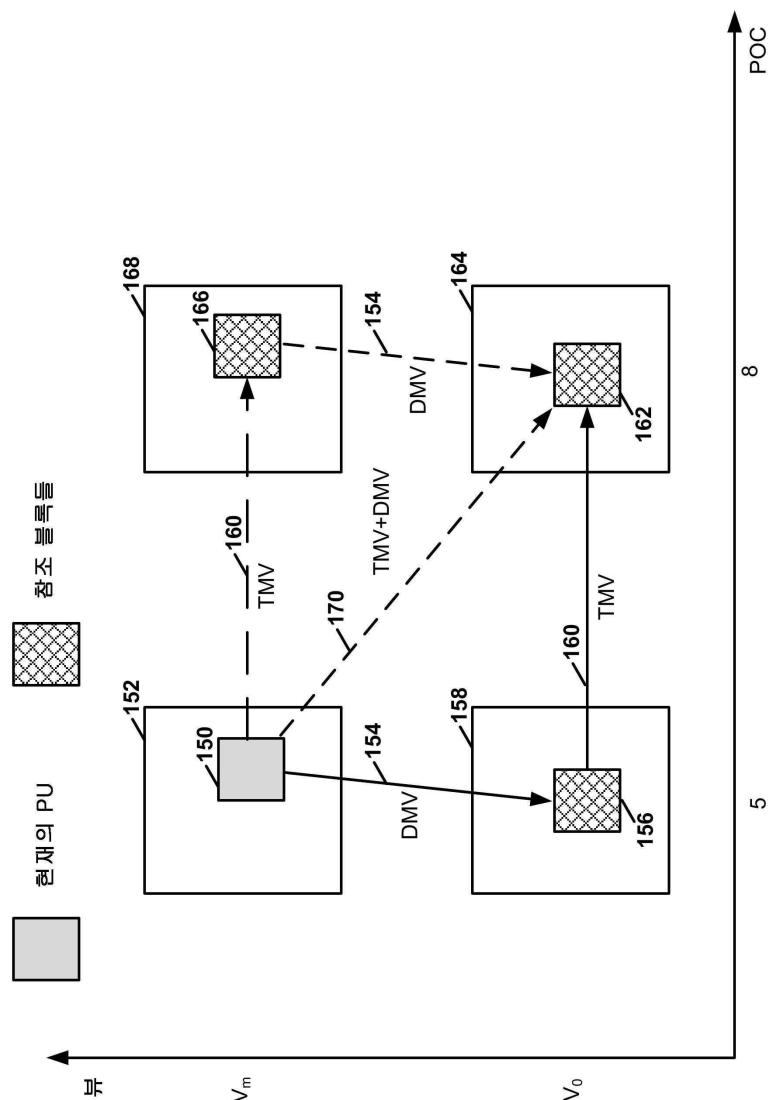
도면9



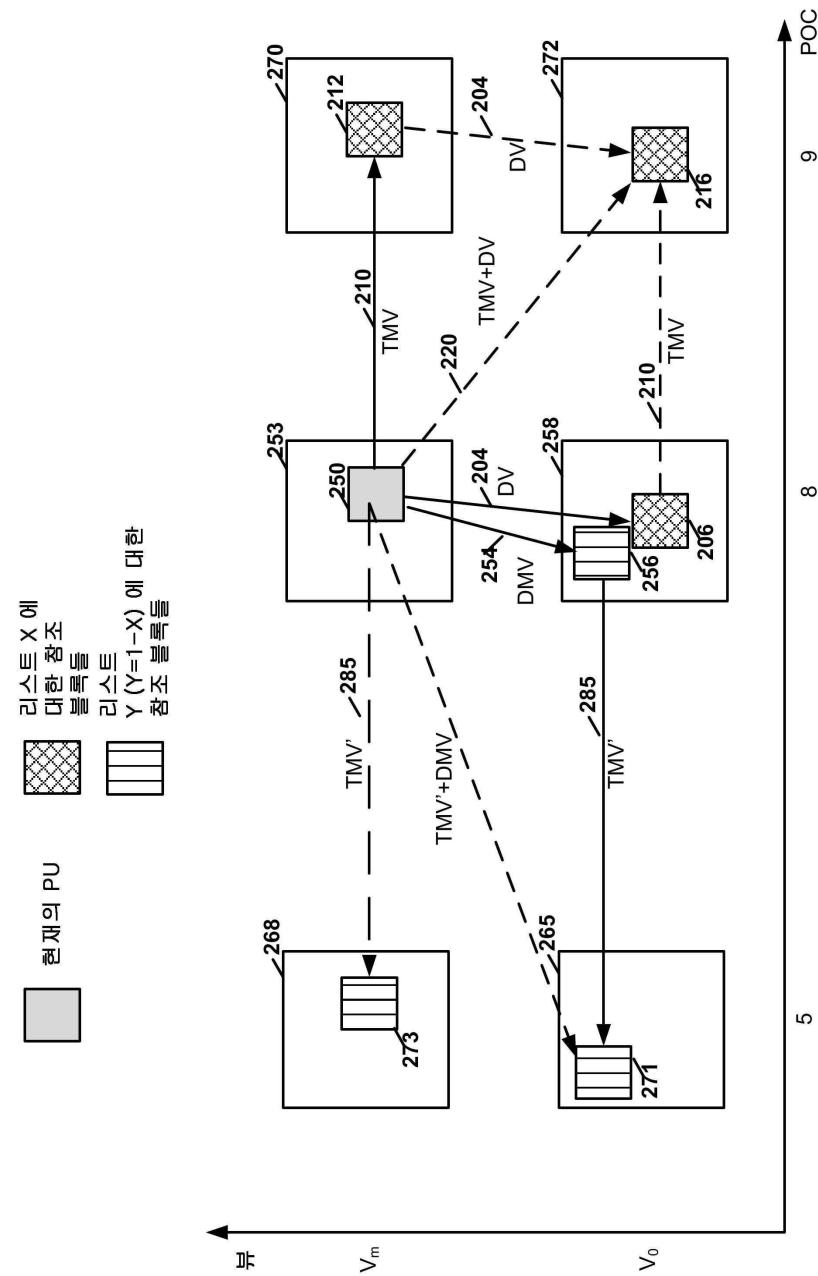
도면 10



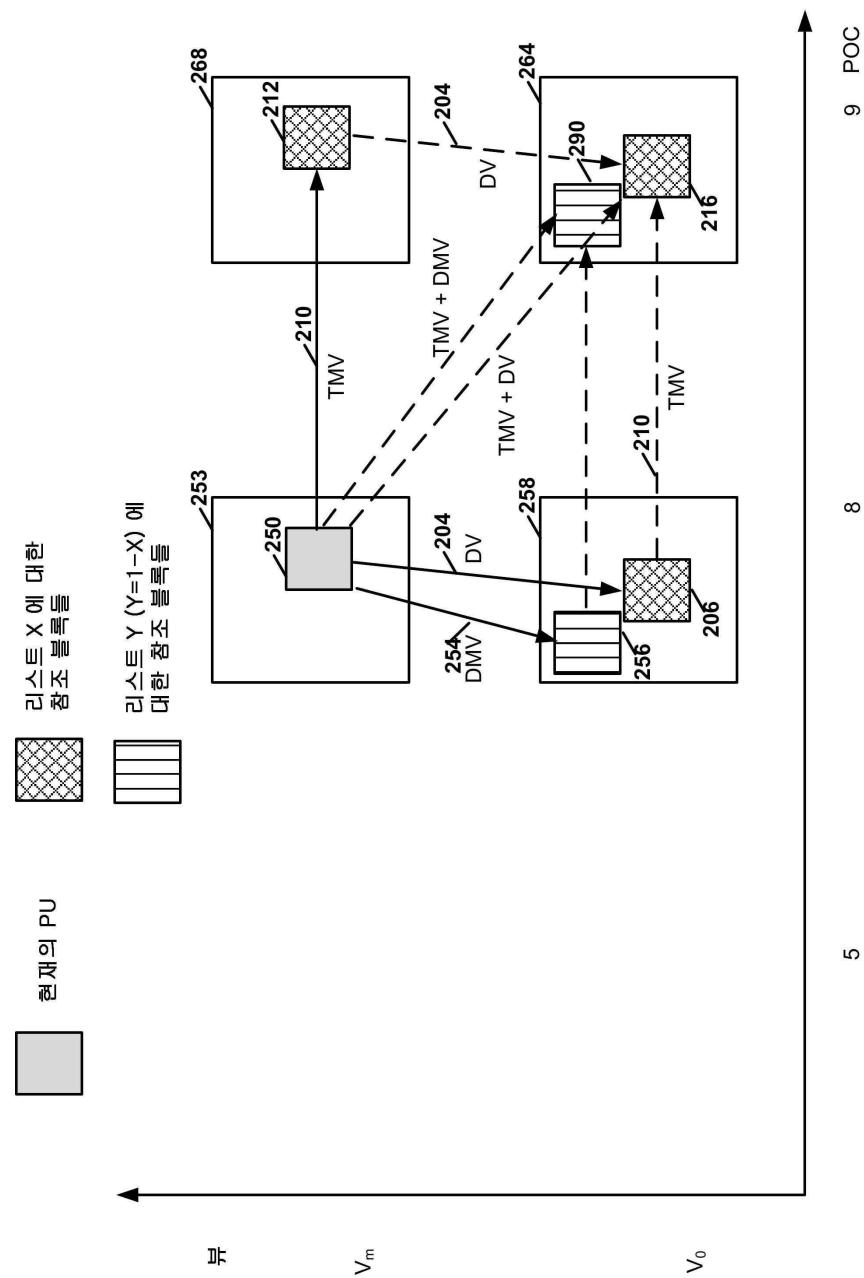
도면 11



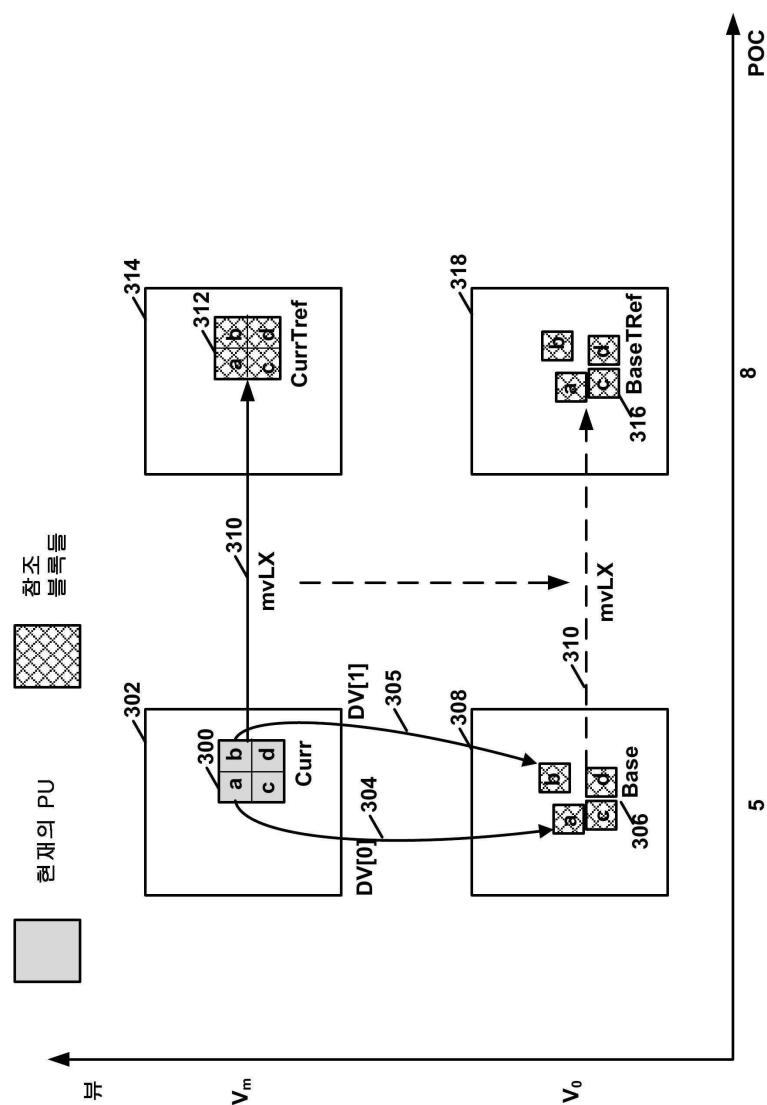
## 도면 12



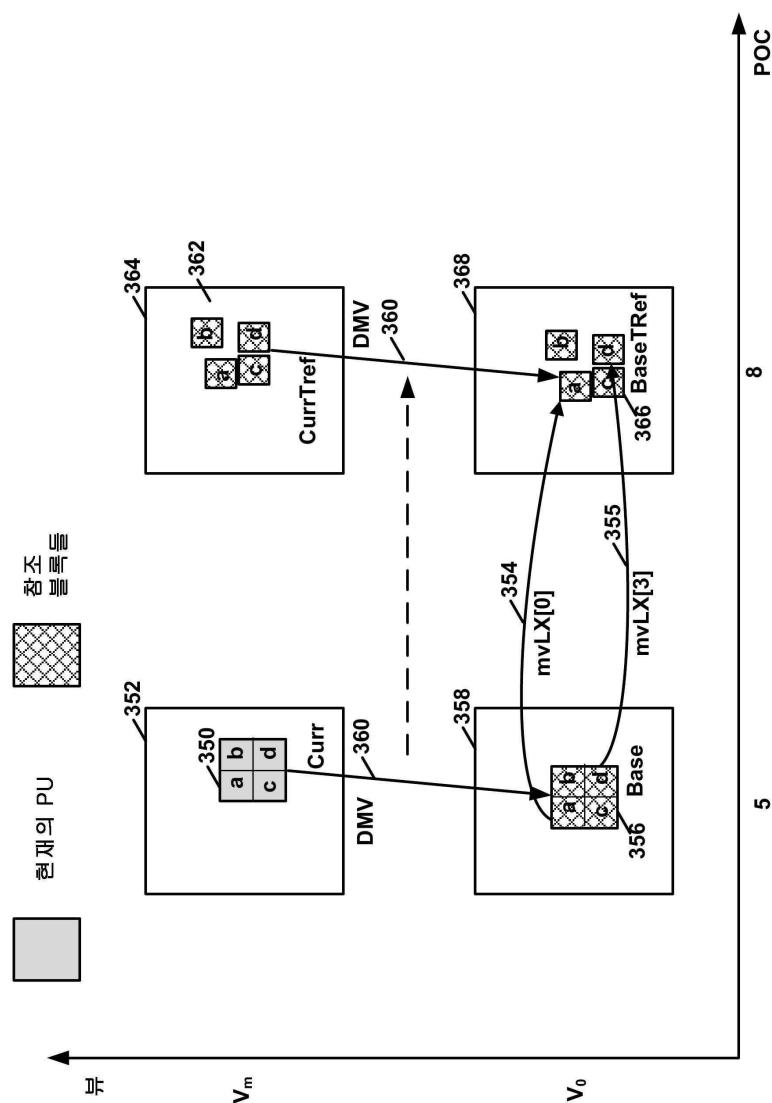
## 도면 13



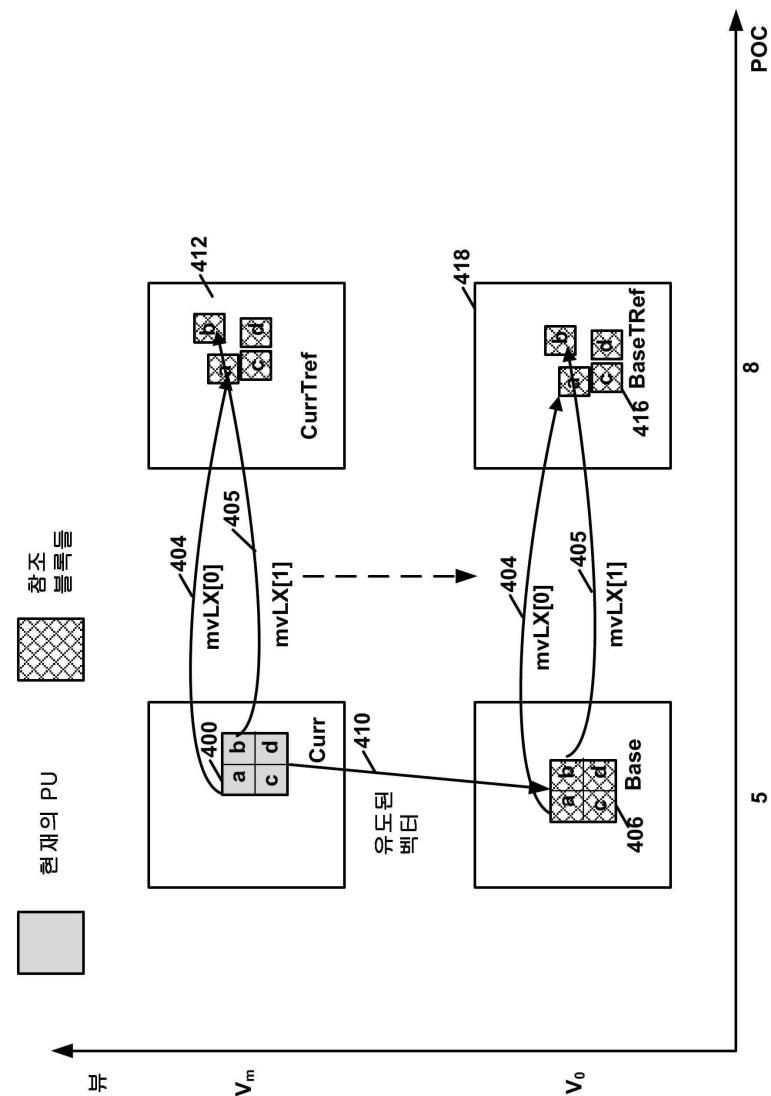
도면14



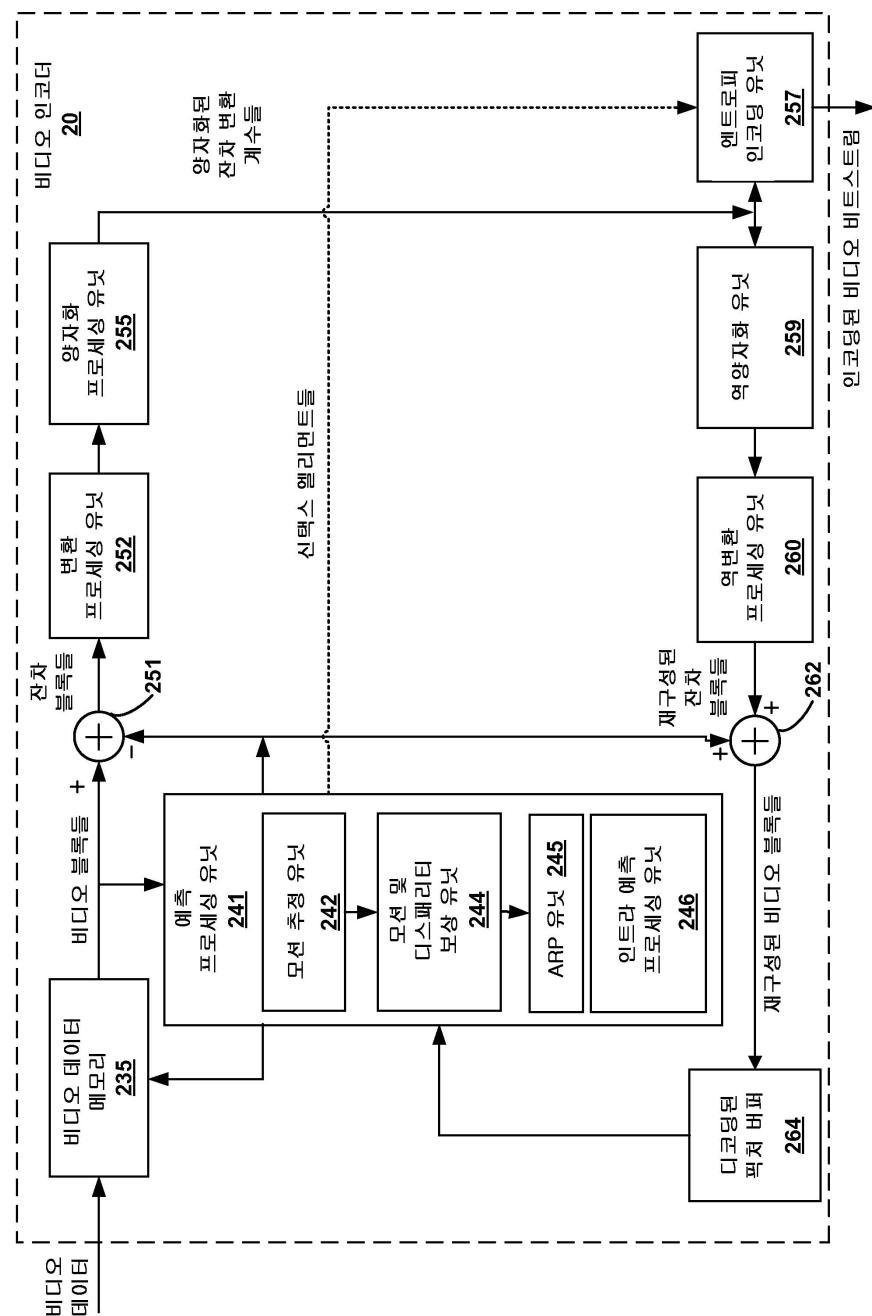
도면 15



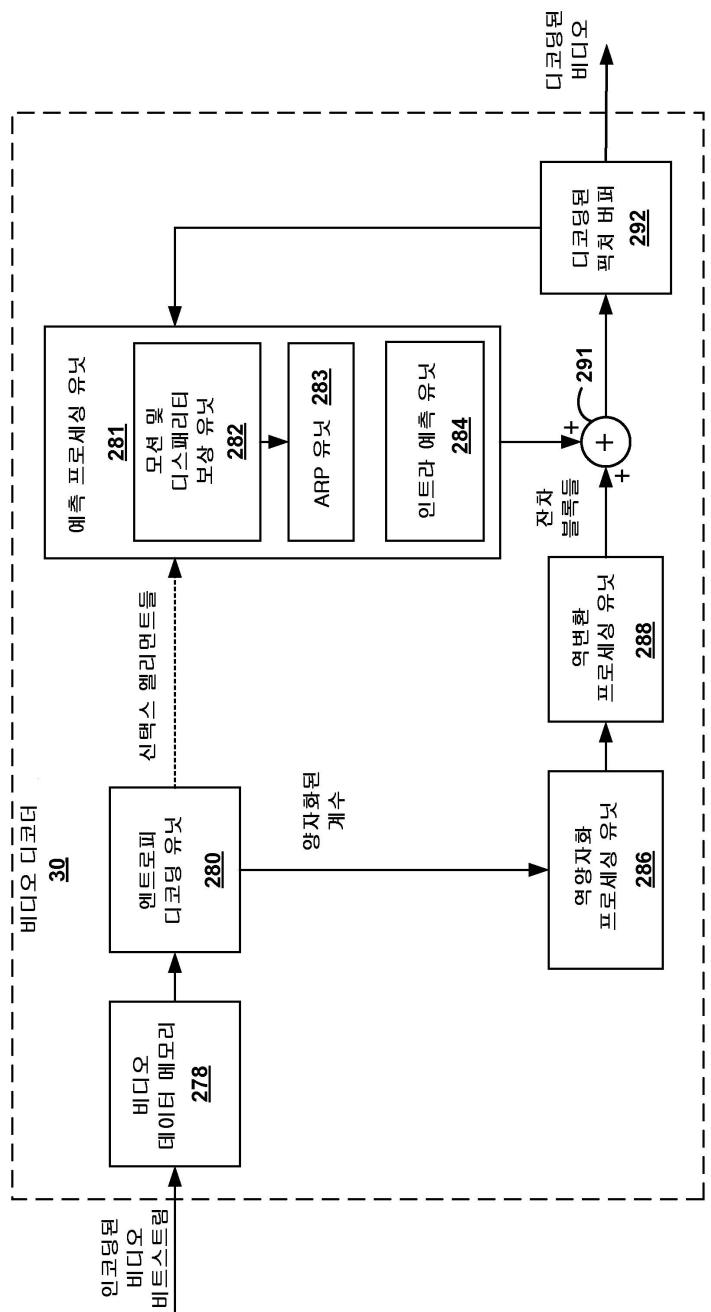
## 도면16



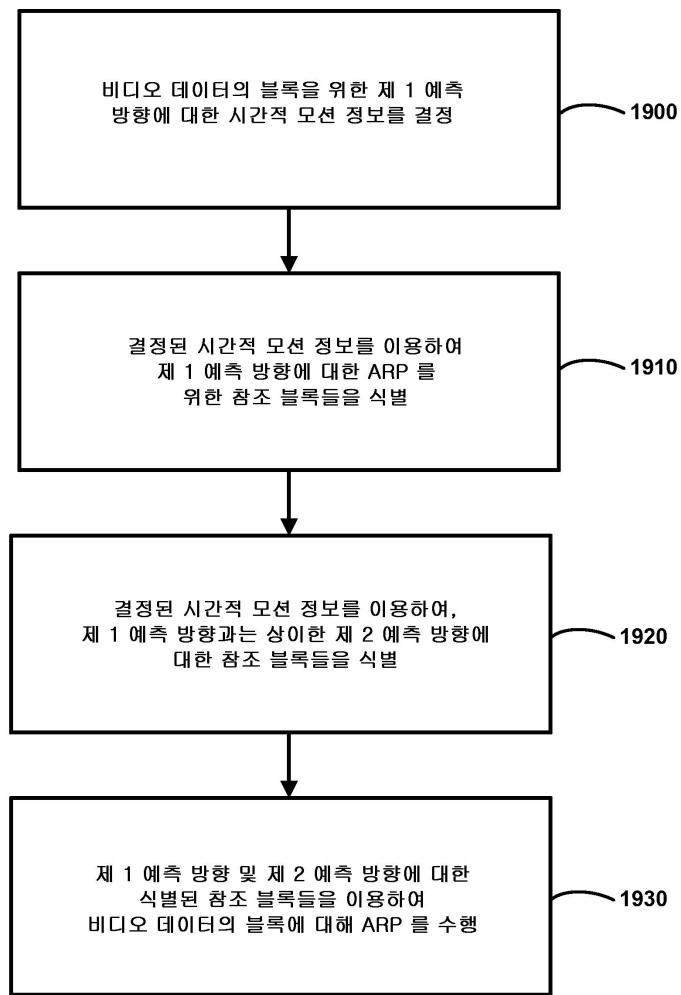
## 도면17



도면 18



## 도면19



**도면20**