



등록특허 10-2127549



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2020년06월26일  
(11) 등록번호 10-2127549  
(24) 등록일자 2020년06월22일

- (51) 국제특허분류(Int. Cl.)  
*HO4N 19/70* (2014.01) *HO4N 19/33* (2014.01)  
*HO4N 19/52* (2014.01) *HO4N 19/58* (2014.01)
- (52) CPC특허분류  
*HO4N 19/70* (2015.01)  
*HO4N 19/33* (2015.01)
- (21) 출원번호 10-2016-7020905
- (22) 출원일자(국제) 2014년12월30일  
심사청구일자 2019년12월06일
- (85) 번역문제출일자 2016년07월29일
- (65) 공개번호 10-2016-0102074
- (43) 공개일자 2016년08월26일
- (86) 국제출원번호 PCT/US2014/072698
- (87) 국제공개번호 WO 2015/103224  
국제공개일자 2015년07월09일

(30) 우선권주장  
61/923,529 2014년01월03일 미국(US)  
14/584,473 2014년12월29일 미국(US)

(56) 선행기술조사문현  
JCTVC-P0184-v4  
JCTVC-00223  
US20130195186 A1

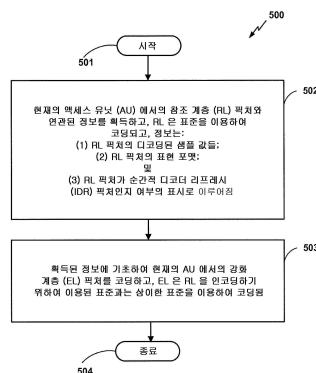
전체 청구항 수 : 총 11 항

심사관 : 조우연

(54) 발명의 명칭 멀티-계층 비디오 코딩에서의 상이한 코덱의 기본 계층의 지원

**(57) 요 약**

특정 양태들에 따라 비디오 정보를 코딩하기 위한 장치는 메모리 및 프로세서를 포함한다. 메모리 유닛은 강화 계층 (EL) 및 대응하는 참조 계층 (RL) 과 연관된 비디오 정보를 저장하도록 구성된다. 프로세서는 현재의 액세스 유닛 (AU) 에서의 EL 픽처를 코딩하도록 구성되고, 상기 EL 은 상기 RL 을 코딩하기 위하여 이용되는 제 2 표준과는 상이한 제 1 표준을 이용하여 코딩되고, 상기 EL 픽처의 코딩은 상기 현재의 액세스 유닛에서의 RL 픽처와 연관된 정보에 기초하고, 상기 RL 픽처와 연관된 정보는, 외부 수단에 의해 제공되며, (1) RL 픽처의 디코딩된 샘플 값들; (2) RL 픽처의 표현 포맷; 및 (3) RL 픽처가 순간적 디코딩 리프레시 (IDR) 픽처인지 아닌지 여부의 표시로 이루어진다.

**대 표 도** - 도5

(52) CPC특허분류

*HO4N 19/52* (2015.01)

*HO4N 19/58* (2015.01)

(72) 발명자

천 잉

미국 92121-1714 캘리포니아주 샌디에고 모어하우

스 드라이브 5775

헨드리 프누

미국 92121-1714 캘리포니아주 샌디에고 모어하우  
스 드라이브 5775

---

## 명세서

### 청구범위

#### 청구항 1

비디오 정보를 코딩하기 위한 장치로서,

상기 계층 (EL) 및 대응하는 참조 계층 (RL) 과 연관된 비디오 정보를 저장하도록 구성된 메모리; 및

상기 메모리에 동작가능하게 결합된 프로세서를 포함하며,

상기 프로세서는,

현재의 액세스 유닛 (AU)에서의 EL 블록을 코딩하는 것으로서, 상기 EL은 H.265/HEVC 비디오 코딩 표준을 이용하여 코딩되고 상기 RL은 H.264/AVC 비디오 코딩 표준을 이용하여 코딩되며, 상기 EL 블록의 코딩은 상기 현재의 액세스 유닛에서의 RL 블록과 연관된 정보에 기초하는, 상기 EL 블록을 코딩하고, 그리고

상기 현재의 AU에서의 상기 EL 블록을 코딩하도록 구성된 상기 프로세서 외부의 엔티티로부터 상기 RL 블록과 연관된 상기 정보를 수신하는 것으로서, 상기 정보는:

- (1) 상기 RL 블록의 디코딩된 샘플 값들;
- (2) 상기 RL 블록의 표현 포맷; 및
- (3) 상기 RL 블록이 순간적 디코딩 리프레시 (IDR) 블록인지 여부의 표시

로 이루어지는, 상기 정보를 수신하도록

구성되며,

상기 프로세서는 추가로, 상기 RL 블록의 상기 디코딩된 샘플 값들을 출력하지 않도록 구성되고;

상기 메모리는 디코딩된 블록 버퍼 (DPB)를 포함하고, 상기 프로세서는 추가로, 상기 EL을 저장하기 위해 이 용도로 상기 디코딩된 블록 버퍼의 일부가 아닌 디코딩된 블록 저장소 내에 상기 RL 블록의 상기 디코딩된 샘플 값들을 저장하도록 구성되고;

상기 프로세서는 추가로, 상기 RL 블록이 속하는 액세스 유닛에 관련된 정보를 상기 외부의 엔티티로부터 획득하도록 구성되고;

상기 프로세서는 추가로, 상기 RL 블록의 계층 식별자를 상기 EL 블록의 계층 식별자 또는 0과 동일하게 설정하도록 구성되고;

상기 프로세서는 추가로, 상기 RL 블록의 블록 순서 카운트 (POC)를 상기 EL 블록의 POC와 동일하게 설정하도록 구성되고;

상기 프로세서는 추가로, 상기 RL 블록을 장기 참조 블록으로서 표기하도록 구성되고;

상기 메모리는 코딩된 블록 버퍼 (CPB)를 포함하고, 상기 프로세서는 추가로, 상기 RL 블록을 CPB 동작들에서 0 비트들을 가지게 설정하도록 구성되며; 그리고

상기 프로세서는 추가로, 상기 EL에 대한 DPB 동작들에서 상기 RL 블록의 상기 디코딩된 샘플 값들을 참조하지 않도록 구성되는, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 2

제 1 항에 있어서,

상기 외부의 엔티티는 제 1 표준을 이용하여 비디오 정보를 코딩하도록 구성되는 코더인, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 3

제 1 항에 있어서,

상기 프로세서는 추가로,

상기 현재의 AU에서의 상기 RL 픽처와 연관된 제 2 정보를 수신하는 것으로서, 상기 제 2 정보는: (4) 상기 RL 픽처가 프레임 또는 필드인지 여부의 표시를 포함하는, 상기 제 2 정보를 수신하고; 그리고

상기 현재의 AU에서의 상기 RL 픽처와 연관된 상기 정보 및 상기 제 2 정보에 기초하여 상기 EL 픽처를 코딩하도록

구성되는, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 4

제 3 항에 있어서,

상기 RL 픽처가 필드일 경우, 상기 RL 픽처와 연관된 상기 제 2 정보는: (5) 필드 패리티를 더 포함하는, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 5

제 1 항에 있어서,

상기 RL 픽처의 표현 포맷은: 상기 RL 픽처의 폭, 상기 RL 픽처의 높이, 상기 RL 픽처의 컬러 포맷, 상기 RL 픽처의 루마 컴포넌트들의 비트 십도, 또는 상기 RL 픽처의 크로마 컴포넌트들의 비트 십도 중의 적어도 하나를 포함하는, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 6

제 5 항에 있어서,

상기 RL 픽처의 상기 폭 및 상기 RL 픽처의 상기 높이는 루마 샘플들의 유닛들로 특정되는, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 7

제 1 항에 있어서,

상기 RL 픽처와 연관된 상기 정보는 하나 이상의 애플리케이션 프로그래밍 인터페이스들 (API들) 을 이용하여 제공되는, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 8

제 1 항에 있어서,

상기 장치는: 테스크톱 컴퓨터, 노트북 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터, 셋톱 박스, 전화 핸드셋, 스마트 폰, 무선 통신 디바이스, 스마트 패드, 텔레비전, 카메라, 디스플레이 디바이스, 디지털 미디어 플레이어, 비디오 게임용 콘솔, 또는 비디오 스트리밍 디바이스 중의 하나 이상을 포함하는, 비디오 정보를 코딩하기 위한 장치.

#### 청구항 9

비디오 정보를 코딩하는 방법으로서,

컴퓨터 하드웨어를 포함하는 프로세서를 이용하여,

강화 계층 (EL) 및 대응하는 참조 계층 (RL) 과 연관된 비디오 정보를 메모리 내에 저장하고,

현재의 액세스 유닛 (AU)에서의 EL 픽처를 코딩하는 것으로서, 상기 EL은 H.265/HEVC 비디오 코딩 표준을 이용하여 코딩되고 상기 RL은 H.264/AVC 비디오 코딩 표준을 이용하여 코딩되며, 상기 EL 픽처의 코딩은 상기 현재의 액세스 유닛에서의 RL 픽처와 연관된 정보에 기초하는, 상기 EL 픽처를 코딩하고, 그리고

상기 프로세서 외부의 엔티티로부터 상기 RL 픽처와 연관된 상기 정보를 수신하는 것으로서, 상기 정보

는:

- (1) 상기 RL 픽처의 디코딩된 샘플 값들;
- (2) 상기 RL 픽처의 표현 포맷; 및
- (3) 상기 RL 픽처가 순간적 디코딩 리프레시 (IDR) 픽처인지 여부의 표시

로 이루어지는, 상기 정보를 수신하는 단계를 포함하며,

상기 프로세서는 추가로, 상기 RL 픽처의 상기 디코딩된 샘플 값들을 출력하지 않도록 구성되고;

상기 메모리는 디코딩된 픽처 버퍼 (DPB) 를 포함하고, 상기 방법은 상기 프로세서를 이용하여 상기 EL 을 저장하기 위해 이용된 상기 디코딩된 픽처 버퍼의 일부가 아닌 디코딩된 픽처 저장소 내에 상기 RL 픽처의 상기 디코딩된 샘플 값들을 저장하는 단계를 더 포함하고;

상기 방법은 상기 프로세서를 이용하여 상기 RL 픽처가 속하는 액세스 유닛에 관련된 정보를 상기 외부의 엔티티로부터 획득하는 단계를 더 포함하고;

상기 방법은 상기 프로세서를 이용하여 상기 RL 픽처의 계층 식별자를 상기 EL 픽처의 계층 식별자 또는 0 과 동일하게 설정하는 단계를 더 포함하고;

상기 방법은 상기 프로세서를 이용하여 상기 RL 픽처의 픽처 순서 카운트 (POC) 를 상기 EL 픽처의 POC 와 동일하게 설정하는 단계를 더 포함하고;

상기 방법은 상기 프로세서를 이용하여 상기 RL 픽처를 장기 참조 픽처로서 표기하는 단계를 더 포함하고;

상기 메모리는 코딩된 픽처 버퍼 (CPB) 를 포함하고, 상기 방법은 상기 프로세서를 이용하여 상기 RL 을 CPB 동작들에서 0 비트들을 가지게 설정하는 단계를 더 포함하고; 그리고

상기 프로세서는 추가로, 상기 EL 에 대한 DPB 동작들에서 상기 RL 픽처의 상기 디코딩된 샘플 값들을 참조하지 않도록 구성되는, 비디오 정보를 코딩하는 방법.

## 청구항 10

제 9 항에 있어서,

상기 RL 픽처의 표현 포맷은: 상기 RL 픽처의 폭, 상기 RL 픽처의 높이, 상기 RL 픽처의 컬러 포맷, 상기 RL 픽처의 루마 컴포넌트들의 비트 심도, 또는 상기 RL 픽처의 크로마 컴포넌트들의 비트 심도 중의 적어도 하나를 포함하는, 비디오 정보를 코딩하는 방법.

## 청구항 11

컴퓨터 하드웨어를 포함하는 프로세서 상에서 실행될 경우, 상기 프로세서로 하여금, 제 9 항 또는 제 10 항에 기재된 방법을 수행하게 하는 명령들을 포함하는 비-일시적 컴퓨터 판독가능 저장 매체.

## 청구항 12

삭제

## 청구항 13

삭제

## 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 개시물은 비디오 코딩 및 압축의 분야에 관한 것으로, 특히, 스케일러블 비디오 코딩 (scalable video coding; SVC), 멀티뷰 비디오 코딩 (multiview video coding; MVC), 또는 3D 비디오 코딩 (3D video coding;

3DV) 의 분야에 관한 것이다.

## 배경기술

[0002] 디지털 비디오 기능들은, 디지털 텔레비전들, 디지털 디렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말 (PDA) 들, 랩톱 또는 데스크톱 컴퓨터들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화들, 화상 원격회의 디바이스들 등을 포함하는 광범위한 디바이스들 내로 편입될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, 어드밴스드 비디오 코딩 (AVC) 에 의해 정의된 표준들, 현재 개발 중인 고효율 비디오 코딩 (HEVC) 표준, 및 이러한 표준들의 확장들에서 설명된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신하고, 수신하고, 인코딩하고, 및/또는 저장할 수도 있다.

[0003] 비디오 압축 기법들은 비디오 시퀀스들에 내재된 중복성을 감소시키거나 제거하기 위해 공간적 (인트라-픽처) 예측 및/또는 시간적 (인터-픽처) 예측을 수행한다. 블록-기반 비디오 코딩을 위하여, 비디오 슬라이스 (예컨대, 비디오 프레임, 비디오 프레임의 부분 등) 는 비디오 블록들로 파티셔닝 (partitioning) 될 수도 있으며, 이 비디오 블록들은 또한, 트리블록 (treeblock) 들, 코딩 유닛 (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.

[0004] 공간적 또는 시간적 예측은 코딩되어야 할 블록에 대한 예측 블록으로 귀착된다. 잔차 데이터는 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 블록 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 후 이들은 양자화될 수도 있다. 초기에 2 차원 어레이로 배치된 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

### 해결하려는 과제

### 과제의 해결 수단

[0005] 스케일러블 비디오 코딩 (SVC) 은, 참조 계층 (RL) 으로서 때때로 지정된 기본 계층 (BL) 및 하나 이상의 스케일러블 강화 계층 (EL) 들이 이용되는 비디오 코딩을 지정한다. SVC 에서, 기본 계층을 품질의 기본 레벨을 갖는 비디오 데이터를 반송 (carry) 할 수 있다. 하나 이상의 강화 계층들은 예를 들어, 더 높은 공간적, 시간적, 및/또는 신호-대-잡음 (SNR) 레벨들을 지원하기 위하여 추가적인 비디오 데이터를 반송할 수 있다. 강화 계층들은 이전에 인코딩된 계층에 대하여 정의될 수도 있다. 예를 들어, 하부 계층은 BL 로서 작용할 수도 있는 반면, 상부 계층은 EL 로서 작용할 수도 있다. 중간 계층들은 EL 들 또는 RL 들의 어느 하나, 또는 양자로서 작용할 수도 있다. 예를 들어, 중간 계층 (예컨대, 가장 낮은 계층도 아니고 가장 높은 계층도 아닌 계층) 은 기본 계층 또는 임의의 중간에 삽입된 강화 계층들과 같이, 중간 계층 하부의 계층들에 대하여 EL 일 수도 있고, 이와 동시에, 중간 계층 상부의 하나 이상의 강화 계층들에 대하여 RL 로서 작용할 수도 있다. 유사하게, HEVC 표준의 멀티뷰 또는 3D 확장에서는, 다수의 뷰들이 있을 수도 있고, 하나의 뷰의 정보는 또 다른 뷰의 정보 (예컨대, 모션 추정, 모션 벡터 예측, 및/또는 다른 중복성들) 를 코딩 (예컨대, 인코딩 또는 디코딩) 하기 위하여 사용될 수도 있다.

[0006] 특정 양태들에 따라 비디오 정보를 코딩하기 위한 장치는 메모리 및 프로세서를 포함한다. 메모리 유닛은 강화 계층 (EL) 및 대응하는 참조 계층 (RL) 과 연관된 비디오 정보를 저장하도록 구성된다. 프로세서는 현

제의 액세스 유닛 (AU)에서의 EL 픽처를 코딩하도록 구성되고, EL은 RL을 코딩하기 위하여 이용되는 제 2 표준과는 상이한 제 1 표준을 이용하여 코딩되고, EL 픽처의 코딩은 현재의 액세스 유닛에서의 RL 픽처와 연관된 정보에 기초하고, RL 픽처와 연관된 정보는 외부 수단에 의해 제공되고, (1) RL 픽처의 디코딩된 샘플 값들; (2) RL 픽처의 표현 포맷; 및 (3) RL 픽처가 순간적 디코딩 리프레시 (IDR) 픽처인지 여부의 표시로 이루어진다.

[0007] 이 개시물의 시스템들, 방법들, 및 디바이스들은 각각 몇몇 혁신적인 양태들을 가지며, 그 단 하나가 본원에서 개시된 바람직한 속성들을 전적으로 담당하지는 않는다. 하나 이상의 예들의 세부사항들은 첨부한 도면들 및 이하의 설명에서 기재되어 있고, 이들은 본원에서 설명된 발명 개념들의 전체 범위를 제한하도록 의도된 것이 아니다. 다른 특징들, 목적들, 및 장점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백할 것이다.

### 도면의 간단한 설명

[0008] 도 1a는 이 개시물에서 설명된 양태들에 따라 기법들을 사용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 1b는 이 개시물에서 설명된 양태들에 따라 기법들을 수행할 수도 있는 또 다른 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2a는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 인코더의 예를 예시하는 블록도이다.

도 2b는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 인코더의 예를 예시하는 블록도이다.

도 3a는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 디코더의 예를 예시하는 블록도이다.

도 3b는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 디코더의 예를 예시하는 블록도이다.

도 4는 이 개시물의 하나의 실시형태에 따라, 상이한 계층들에서의 픽처들의 일 예의 구성을 예시하는 블록도이다.

도 5는 이 개시물의 하나의 실시형태에 따라, 비디오 정보를 코딩하는 방법을 예시하는 플로우차트이다.

### 발명을 실시하기 위한 구체적인 내용

[0009] 일반적으로, 이 개시물은 HEVC (고효율 비디오 코딩)와 같은, 진보된 비디오 코덱들의 맥락에서의 스케일러를 비디오 코딩을 위한 인터-계층 예측에 관한 것이다. 더욱 구체적으로, 본 개시물은, SHVC로서 지정될 수도 있는 HEVC의 스케일러를 비디오 코딩 확장에서의 인터-계층 예측의 개선된 성능을 위한 시스템들 및 방법들에 관한 것이다.

[0010] 이하의 설명에서는, 어떤 실시형태들에 관련된 H.264/진보된 비디오 코딩 (AVC) 기법들이 설명되고; HEVC 표준 및 관련된 기법들이 또한 논의된다. 어떤 실시형태들은 HEVC 및/또는 H.264 표준들의 맥락에서 본원에서 설명되지만, 당해 분야의 당업자는 본원에서 개시된 시스템들 및 방법들이 임의의 적당한 비디오 코딩 표준에 적용가능할 수도 있다는 것을 인식할 수도 있다. 예를 들어, 본원에서 개시된 실시형태들은 다음의 표준들 중의 하나 이상에 적용가능할 수도 있다: 그 스케일러를 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장들을 포함하는, 국제 전기통신 연합 (International Telecommunication Union; ITU) 전기통신 표준화 섹터 (Telecommunication Standardization Sector) (ITU-T) H.261, 표준화를 위한 국제 기구 (International Organization for Standardization; ISO) 및 국제 전기기술 위원회 (International Electrotechnical Commission; IEC) 동화상 전문가 그룹 (Moving Picture Experts Group; MPEG) 1 (MPEG-1) 비주얼 (Visual), ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 및 (또한, ISO/IEC MPEG-4 AVC로서 알려진) ITU-T H.264.

[0011] HEVC는 일반적으로 많은 점들에서 이전의 비디오 코딩 표준들의 프레임워크 (framework)를 따른다. HEVC에서의 예측의 유닛은 어떤 이전의 비디오 코딩 표준들에서의 예측의 유닛들 (예컨대, 매크로블록)과는 상이하

다. 실제로, 매크로블록의 개념은 어떤 이전의 비디오 코딩 표준들에서 이해된 바와 같이 HEVC에서 존재하지는 않는다. 매크로블록은 다른 가능한 장점들 중에서도, 높은 유연성 (flexibility)을 제공할 수도 있는, 큐드트리 방식에 기초한 계층적 구조에 의해 대체된다. 예를 들어, HEVC 방식 내에서는, 3 개의 타입들의 블록들, 코딩 유닛 (CU), 예측 유닛 (PU), 및 변환 유닛 (TU)이 정의된다. CU는 영역 분할의 기본 유닛을 지정할 수도 있다. CU는 매크로블록의 개념과 유사할 수도 있지만, HEVC는 CU들의 최대 사이즈를 한정하지 않고, 컨텐츠 적응성 (content adaptivity)을 개선시키기 위하여 4 개의 동일한 사이즈의 CU들로의 재귀적 분할을 허용할 수도 있다. PU는 인터/인트라 예측의 기본 유닛으로 고려될 수도 있고, 단일 PU는 불규칙적인 이미지 패턴들을 효과적으로 코딩하기 위하여 다수의 임의적인 형상 파티션들을 포함할 수도 있다.

TU는 변환의 기본 유닛으로 고려될 수도 있다. TU는 PU로부터 독립적으로 정의될 수 있지만; 그러나, TU의 사이즈는 TU가 속하는 CU의 사이즈로 제한될 수도 있다. 3 개의 상이한 개념들로의 블록 구조의 이 분리는 각각의 유닛이 유닛의 개개의 역할에 따라 최적화되도록 할 수도 있고, 이것은 개선된 코딩 효율로 귀착될 수도 있다.

[0012] 예시만의 목적들을 위하여, 본원에서 개시된 어떤 실시형태들은 비디오 데이터의 2 개의 계층들 (예컨대, 기본 계층과 같은 더 낮은 계층과, 강화 계층과 같은 더 높은 계층) 만을 포함하는 예들로 설명된다. 비디오 데이터의 "계층"은 뷰, 프레임 레이트, 해상도등과 같은 적어도 하나의 공통의 특성을 가지는 픽셀들의 시퀀스를 일반적으로 지정할 수도 있다. 예를 들어, 계층은 멀티-뷰 비디오 데이터의 특정한 뷰 (예컨대, 관점)와 연관된 비디오 데이터를 포함할 수도 있다. 또 다른 예로서, 계층은 스케일러블 비디오 데이터의 특정한 계층과 연관된 비디오 데이터를 포함할 수도 있다. 따라서, 이 개시물은 비디오 데이터의 계층 및 뷰를 상호 교환 가능하게 지정할 수도 있다. 즉, 비디오 데이터의 뷰는 비디오 데이터의 계층으로서 지정될 수도 있고, 비디오 데이터의 계층은 비디오 데이터의 뷰로서 지정될 수도 있다. 게다가, (또한, 멀티-계층 비디오 코더 또는 멀티-계층 인코더-디코더로서 지정된) 멀티-계층 코덱은 멀티뷰 코덱 또는 스케일러블 코덱 (예컨대, MV-HEVC, 3D-HEVC, SHVC, 또는 또 다른 멀티-계층 코딩 기법을 이용하여 비디오 데이터를 인코딩 및/또는 디코딩하도록 구성된 코덱)을 공동으로 지정할 수도 있다. 비디오 인코딩 및 비디오 디코딩의 양자는 비디오 코딩으로서 일반적으로 지정될 수도 있다. 이러한 예들은 다수의 기본 및/또는 강화 계층들을 포함하는 구성들에 적용 가능할 수도 있다는 것을 이해해야 한다. 게다가, 설명의 용이함을 위하여, 다음의 개시물은 어떤 실시형태들을 참조한 용어들 "프레임들" 또는 "블록들"을 포함한다. 그러나, 이 용어들은 제한하는 것으로 의도된 것은 아니다. 예를 들어, 이하에서 설명된 기법들은 블록들 (예컨대, CU, PU, TU, 매크로블록들 등), 슬라이스들, 프레임들 등과 같은 임의의 적당한 비디오 유닛들과 함께 이용될 수 있다.

### 비디오 코딩 표준들

[0014] 비디오 이미지, TV 이미지, 스틸 이미지 (still image), 또는 비디오 레코더 또는 컴퓨터에 의해 생성된 이미지와 같은 디지털 이미지는 수평 및 수직 라인들로 배열된 픽셀들 또는 샘플들로 구성될 수도 있다. 단일 이미지에서의 픽셀들의 수는 전형적으로 수만 개이다. 각각의 픽셀은 전형적으로 휘도 (luminance) 및 색차 (chrominance) 정보를 포함한다. 압축이 없다면, 이미지 인코더로부터 이미지 디코더로 전달되어야 할 정보의 순전한 분량은 실시간 이미지 송신을 불가능하게 할 것이다. 송신되어야 할 정보의 양을 감소시키기 위하여, JPEG, MPEG, 및 H.263 표준들과 같은 다수의 상이한 압축 방법들이 개발되었다.

[0015] 비디오 코딩 표준들은 그 SVC 및 MVC 확장들을 포함하는, ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼, 및 (또한, ISO/IEC MPEG-4 AVC로서 알려진) ITU-T H.264를 포함한다.

[0016] 게다가, 새로운 비디오 코딩 표준, 즉, HEVC는 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 및 ISO/IEC 모션 픽처 전문가 그룹 (Motion Picture Experts Group; MPEG)의 비디오 코딩에 관한 공동 협력 팀 (Joint Collaboration Team on Video Coding; JCT-VC)에 의해 개발되고 있다. HEVC 초안 10에 대한 전체 인용은 문서 JCTVC-L1003, Bross 등, "High Efficiency Video Coding (HEVC) Text Specification Draft 10 (고효율 비디오 코딩 (HEVC) 텍스트 사양 초안 10)", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11의 비디오 코딩에 관한 공동 협력 팀 (JCT-VC), 12 차 회의: Geneva, Switzerland, 2013년 1월 14일 내지 2013년 1월 23일이다. HEVC에 대한 멀티뷰 확장, 즉, MV-HEVC와, SHVC로 명명된 HEVC에 대한 스케일러블 확장은 또한, JCT-3V (3D 비디오 코딩 확장 개발에 관한 ITU-T/ISO/IEC 공동 협력 팀) 및 JCT-VC에 의해 각각 개발되고 있다.

### 개요

[0018]

멀티-계층 코딩의 일부의 경우들에는, 참조 계층 (RL) 이 하나의 표준을 이용하여 코딩될 수도 있고, 강화 계층 (EL) 은 또 다른 표준을 이용하여 코딩될 수도 있다. 예를 들어, RL 은 H.264/AVC 에 따라 코딩될 수도 있고, EL 은 H.265/HEVC 에 따라 코딩될 수도 있다. 더 이전의 표준 (또는 코딩 방식) 또는 표준의 더 이전의 버전 (version) (또는 코딩 방식) 을 이용하여 코딩되는 RL 에 대한 지원을 제공하는 것은, 더 이전의 표준 또는 소정의 표준의 더 이전의 버전을 이용하여 코딩된 비디오 데이터가 현재의 표준 또는 더 이전의 버전을 가지는 표준의 현재의 버전을 이용하는 멀티-계층 비디오 코딩에서 사용될 수도 있으므로 유익할 수 있다. 그러나, 상이한 표준 또는 표준의 상이한 버전에서 코딩된 RL 을 프로세싱하는 많은 양태들을 관리하는 것은 멀티-계층 코딩을 위한 지원이 복잡하게 되게 할 수 있다. 예를 들어, 인코더 또는 디코더는 RL 픽처들의 출력을 처리하고, RL 픽처들에 대해 저장된 디코딩된 픽처를 유지하는 등을 행할 필요가 있을 수도 있다.

[0019]

이러한, 그리고 다른 과제들을 해결하기 위하여, 어떤 양태들에 따른 기법들은 간략화된 방식으로 상이한 표준들을 이용하거나 표준의 상이한 버전들을 이용하는 멀티-계층 비디오 코딩을 지원할 수 있다. 예를 들어, HEVC 디코더는 HEVC 이외의 표준을 이용하여 코딩된 RL 픽처들의 관리 및 프로세싱을 최소화할 수도 있다. 예시적인 목적들을 위하여, RL 은 H.264/AVC 를 이용하여 코딩되는 것으로서 설명될 것이고, EL 은 H.265/HEVC 를 이용하여 코딩되는 것으로서 설명될 것이다. 그러나, 상이한 표준들의 임의의 조합은 RL 및 EL 을 코딩하기 위하여 이용될 수 있다. 어떤 양태들에 따르면, 기법들은 다음과 같이 RL 픽처들의 관리 및 프로세싱을 최소화한다: (1) 디코딩된 RL 픽처들은 외부 수단에 의해 제공되고, 그리고 (2) EL 픽처들의 출력과의 동기화를 포함하여, RL 픽처들의 출력은 외부 수단에 의해 제어된다. 외부 수단은 RL 을 코딩하기 위하여 이용된 표준을 지원하는 코더 (예컨대, 인코더 또는 디코더) 를 지칭할 수도 있다. HEVC 디코더는 상이한 표준들을 이용하는 멀티-계층 비디오 코딩을 지원하기 위하여 어떤 규칙들을 구현할 수 있다. 규칙들 및 규칙들에 관련된 세부사항들은 이하에서 더욱 설명된다.

[0020]

### 비디오 코딩 시스템

[0021]

신규한 시스템들, 장치들, 및 방법들의 다양한 양태들은 첨부한 도면들을 참조하여 이하에서 더욱 완전하게 설명된다. 그러나, 이 개시물은 많은 상이한 형태들로 구체화될 수도 있고, 이 개시물의 전반에 걸쳐 제시된 임의의 특정 구조 또는 기능으로 제한된 것으로서 해석되지 않아야 한다. 오히려, 이 양태들은, 이 개시물이 철저하고 완전할 것이며, 개시물의 범위를 당해 분야의 당업자들에게 완전히 전달하도록 제공된다. 본원에서의 교시 사항들에 기초하여, 당해 분야의 당업자는 개시물의 범위가, 본 개시물의 임의의 다른 양태에 독립적으로 또는 이와 조합하여 구현되든지 간에, 본원에서 개시된 신규한 시스템들, 장치들, 및 방법들의 임의의 양태를 커버 (cover) 하도록 의도되는 것을 인식해야 한다. 예를 들어, 본원에서 기재된 임의의 수의 양태들을 이용하여 장치가 구현될 수도 있거나 방법이 실시될 수도 있다. 게다가, 본 개시물의 범위는 본원에서 기재된 본 개시물의 다양한 양태들에 추가하거나 이 다양한 양태들 이외에, 다른 구조, 기능성, 또는 구조 및 기능성을 이용하여 실시되는 이러한 장치 또는 방법을 커버하도록 의도된다. 본원에서 개시된 임의의 양태는 청구항의 하나 이상의 구성요소들에 의해 구체화될 수도 있다는 것을 이해해야 한다.

[0022]

특정한 양태들이 본원에서 설명되지만, 이 양태들의 많은 변형들 및 치환들은 개시물의 범위 내에 속한다. 바람직한 양태들의 일부의 이익들 및 장점들이 언급되지만, 개시물의 범위는 특정한 이익들, 용도들, 또는 목적들에 제한되도록 의도된 것이 아니다. 오히려, 개시물의 양태들은 상이한 무선 기술들, 시스템 구성들, 네트워크들, 및 송신 프로토콜들에 폭넓게 적용가능하도록 의도되며, 이들의 일부는 바람직한 양태들의 도면들 및 다음의 설명에서 예로서 예시되어 있다. 상세한 설명 및 도면들은 제한하는 것이 아니라 개시물의 예시에 불과하고, 개시물의 범위는 첨부된 청구항들 및 그 등가물들에 의해 정의된다.

[0023]

첨부된 도면들은 예들을 예시한다. 첨부된 도면들에서 참조 번호들에 의해 표시된 구성요소들은 다음의 설명에서 유사한 참조 번호들에 의해 표시된 구성요소들에 대응한다. 이 개시물에서, 서수 단어들 (예컨대, "제 1", "제 2", "제 3" 및 등등) 로 시작하는 명칭들을 가지는 구성요소들은 구성요소들이 특정한 순서를 가지는 것을 반드시 암시하지는 않는다. 오히려, 이러한 서수 단어들은 동일하거나 유사한 타입의 상이한 구성요소들을 지칭하기 위하여 이용되는 것에 불과하다.

[0024]

도 1a 는 이 개시물에서 설명된 양태들에 따라 기법들을 사용할 수도 있는 일 예의 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 설명된 바와 같이, 용어 "비디오 코더" 는 비디오 인코더들 및 비디오 디코더들의 양자를 일반적으로 지칭한다. 이 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 비디오 인코딩 및 비디오 디코딩을 일반적으로 지칭할 수도 있다. 비디오 인코더들 및 비디오 디코더들에 추가하여, 본 출원에서 설명된 양태들은 트랜스코더 (transcoder) 들 (예컨대, 비트스트림을 디코딩할 수 있고 또 다른 비트스

트림을 재-인코딩 (re-encode) 할 수 있는 디바이스들) 및 미들박스 (middlebox) 들 (예컨대, 비트스트림을 수정하고, 변환하고, 및/또는 이와 다르게 조작할 수 있는 디바이스들) 과 같은 다른 관련된 디바이스들로 확장될 수도 있다.

[0025] **도 1a** 에서 도시된 바와 같이, 비디오 코딩 시스템 (10) 은 목적지 디바이스 (14) 에 의해 더 이후의 시간에 디코딩되어야 할 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12) 를 포함한다. **도 1a** 의 예에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 별도의 디바이스들을 구성한다. 그러나, 소스 및 목적지 디바이스들 (12, 14) 은 **도 1b** 의 예에서 도시된 바와 같이, 동일한 디바이스 상에 있을 수도 있거나 동일한 디바이스의 일부일 수도 있다는 것에 주목한다.

[0026] **도 1a** 를 다시 한번 참조하면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크톱 컴퓨터들, 노트북 (예컨대, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스 (set-top box) 들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 광범위한 디바이스들 중의 임의의 것을 포함할 수도 있다. 일부의 경우들에는, 소스 디바이스 (12) 및 목적지 디바이스 (14) 가 무선 통신을 위해 구비될 수도 있다.

[0027] 목적지 디바이스 (14) 는 링크 (16) 를 통해, 디코딩되어야 할 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. **도 1a** 의 예에서, 링크 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 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 무선 주파수 (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.

[0028] 대안적으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 선택적인 저장 디바이스 (31) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 목적지 디바이스 (14) 의, 예를 들어, 입력 인터페이스 (28) 에 의해 저장 디바이스 (31) 로부터 액세스될 수도 있다. 저장 디바이스 (31) 는 하드 드라이브, 플래시 메모리, 휴발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적당한 디지털 저장 매체들과 같은, 분산되거나 국소적으로 액세스된 다양한 데이터 저장 매체들 중의 임의의 것을 포함할 수도 있다.

추가의 예에서, 저장 디바이스 (31) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 유지할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 저장 디바이스 (31) 로부터 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. 일 예의 파일 서버들은 (예컨대, 웹사이트를 위한) 웹 서버, 파일 전송 프로토콜 (File Transfer Protocol; 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, 무선 로컬 영역 네트워크 (WLAN) 접속), 유선 접속 (예컨대, 디지털 가입자 회선 (DSL), 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 저장 디바이스 (31) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양자의 조합일 수도 있다.

[0029] 이 개시물의 기법들은 무선 애플리케이션들 또는 설정들로 제한되지는 않는다. 기법들은 오버-디-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 (예컨대, 하이퍼텍스트 전송 프로토콜 (HTTP) 을 통한 동적 적응적 스트리밍 등), 데이터 저장 매체 상에서의 저장을 위한 디지털 비디오의 인코딩, 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, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 및/또는 영상 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위하여

일방향 (one-way) 또는 양방향 (two-way) 비디오 송신을 지원하도록 구성될 수도 있다.

[0030] **도 1a** 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 일부의 경우들에는, 출력 인터페이스 (22) 가 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 비디오 컨텐츠 제공자로부터 비디오를 수신하기 위한 비디오 공급 인터페이스, 및/또는 소스 비디오로서 컴퓨터 그래픽 데이터를 생성하기 위한 컴퓨터 그래픽 시스템과 같은 소스, 또는 이러한 소스들의 조합을 포함할 수도 있다. 하나의 예로서, 비디오 소스 (18) 가 비디오 카메라일 경우, 소스 모디바이스 (12) 및 목적지 디바이스 (14) 는 **도 1b** 의 예에서 예시된 바와 같이, 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 이 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0031] 캡처된 (captured), 프리-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안적으로), 디코딩 및/또는 재생을 위하여, 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 더 이후의 액세스를 위해 저장 디바이스 (31) 상으로 저장될 수도 있다. **도 1a** 및 **도 1b** 에서 예시된 비디오 인코더 (20) 는 **도 2a** 에서 예시된 비디오 인코더 (20), **도 2b** 에서 예시된 비디오 인코더 (23), 또는 본원에서 설명된 임의의 다른 비디오 인코더를 포함할 수도 있다.

[0032] **도 1a** 의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부의 경우들에는, 입력 인터페이스 (28) 가 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해, 및/또는 저장 디바이스 (31) 로부터 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 를 통해 통신되거나 저장 디바이스 (31) 상에서 제공된 인코딩된 비디오 데이터는 비디오 데이터를 디코딩함에 있어서, 비디오 디코더 (30) 와 같은 비디오 디코더에 의한 이용을 위하여 비디오 인코더 (20) 에 의해 생성된 다양한 신택스 엘리먼트 (syntax element) 들을 포함할 수도 있다. 이러한 신택스 엘리먼트들은 통신 매체 상에서 송신되거나, 저장 매체 상에 저장되거나, 파일 서버에 저장된 인코딩된 비디오 데이터와 함께 포함될 수도 있다. **도 1a** 및 **도 1b** 에서 예시된 비디오 디코더 (30) 는 **도 3a** 에서 예시된 비디오 디코더 (30), **도 3b** 에서 예시된 비디오 디코더 (33), 또는 본원에서 설명된 임의의 다른 비디오 디코더를 포함할 수도 있다.

[0033] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합될 수도 있거나, 목적지 디바이스 (14) 의 외부에 있을 수도 있다. 일부의 예들에서, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함할 수도 있고, 또한, 외부의 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중의 임의의 것을 포함할 수도 있다.

[0034] 관련된 양태들에서, **도 1b** 는 일 예의 비디오 인코딩 및 디코딩 시스템 (10') 을 도시하고, 여기서, 소스 및 목적지 디바이스들 (12, 14) 은 디바이스 (11) 상에 있거나 그 일부이다. 디바이스 (11) 는 "스마트" 폰 등과 같은 전화 핸드셋일 수도 있다. 디바이스 (11) 는 소스 및 목적지 디바이스들 (12, 14) 과 동작가능하게 통신하는 선택적인 제어기/프로세서 디바이스 (13) 를 포함할 수도 있다. **도 1b** 의 시스템 (10') 은 비디오 인코더 (20) 와 출력 인터페이스 (22) 사이에 비디오 프로세싱 유닛 (21) 을 더 포함할 수도 있다. 일부의 구현예들에서, 비디오 프로세싱 유닛 (21) 은 **도 1b** 에서 예시된 바와 같이, 별도의 유닛이지만, 그러나, 다른 구현예들에서는, 비디오 프로세싱 유닛 (21) 이 비디오 인코더 (20) 및/또는 프로세서/제어기 디바이스 (13) 의 부분으로서 구현될 수 있다. 시스템 (10') 은 또한, 비디오 시퀀스에서 관심있는 객체를 추적할 수 있는 선택적인 추적기 (29) 를 포함할 수도 있다. 추적되어야 할 관심있는 객체는 본 개시물의 하나 이상의 양태들과 관련하여 설명된 기법에 의해 세그먼트화될 수도 있다. 관련된 양태들에서, 추적하는 것은 디스플레이 디바이스 (32) 에 의해, 단독으로 또는 추적기 (29) 와 함께 수행될 수도 있다. **도 1b** 의 시스템 (10') 및 그 컴포넌트들은 그 외에는, **도 1a** 의 시스템 (10) 및 그 컴포넌트들과 유사하다.

[0035] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC 와 같은 비디오 압축 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HEVC Test Model; HM) 을 준수할 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코

더 (30) 는 MPEG-4, Part 10, AVC, 또는 이러한 표준들의 확장들로서 대안적으로 지정된 ITU-T H.264 표준과 같은 다른 독점적 또는 산업 표준들에 따라 동작할 수도 있다. 그러나, 이 개시물의 기법들은 임의의 특정한 코딩 표준으로 제한되지는 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 을 포함한다.

[0036] **도 1a 및 도 1a** 의 예들에서 도시되지 않았지만, 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. 적용가능한 경우, 일부의 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol; UDP) 과 같은 다른 프로토콜들을 준수할 수도 있다.

[0037] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (digital signal processor; DSP) 들, 주문형 집적 회로 (application specific integrated circuit; ASIC) 들, 필드 프로그래밍가능 게이트 어레이 (field programmable gate array; FPGA) 들, 개별 로직, 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 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, 그 어느 하나는 조합된 인코더/디코더 (combined encoder/decoder; CODEC) 의 일부로서 각각의 디바이스 내에 통합될 수도 있다.

#### 비디오 코딩 프로세스

[0039] 위에서 간단히 언급된 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 인코딩한다. 비디오 데이터는 하나 이상의 픽처들을 포함할 수도 있다. 픽처들의 각각은 비디오의 일부를 형성하는 스크린 이미지이다. 일부의 사례들에서, 픽처는 비디오 "프레임" 으로서 지정될 수도 있다. 비디오 인코더 (20) 가 비디오 데이터를 인코딩할 때, 비디오 인코더 (20) 는 비트스트림을 생성할 수도 있다. 비트스트림은 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 비트스트림은 코딩된 픽처들 및 연관된 데이터를 포함할 수도 있다. 코딩된 픽처는 픽처의 코딩된 표현이다.

[0040] 비트스트림을 생성하기 위하여, 비디오 인코더 (20) 는 비디오 데이터에서의 각각의 픽처에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 픽처들에 대해 인코딩 동작들을 수행할 때, 비디오 인코더 (20) 는 일련의 코딩된 픽처들 및 연관된 데이터를 생성할 수도 있다. 연관된 데이터는 비디오 파라미터 세트 (VPS) 들, 시퀀스 파라미터 세트 (SPS) 들, 픽처 파라미터 세트 (PPS) 들, 적응 파라미터 세트 (APS) 들, 및 다른 선택 구조들을 포함할 수도 있다. SPS 는 픽처들의 제로 (zero) 이상의 시퀀스들에 적용가능한 파라미터들을 포함할 수도 있다. PPS 는 제로 이상의 픽처들에 적용가능한 파라미터들을 포함할 수도 있다. APS 는 제로 이상의 픽처들에 적용가능한 파라미터들을 포함할 수도 있다. APS 에서의 파라미터들은 PPS 에서의 파라미터들보다 변경될 가능성이 더 많은 파라미터들일 수도 있다.

[0041] 코딩된 픽처를 생성하기 위하여, 비디오 인코더 (20) 는 픽처를 동일한 사이즈의 비디오 블록들로 파티셔닝할 수도 있다. 비디오 블록은 샘플들의 2 차원 어레이일 수도 있다. 비디오 블록들의 각각은 트리블록과 연관된다. 일부의 사례들에서, 트리블록은 최대 코딩 유닛 (LCU) 으로서 지정될 수도 있다. HEVC 의 트리블록들은 H.264/AVC 와 같은 이전의 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, 트리블록은 반드시 특정한 사이즈로 제한되는 것은 아니고, 하나 이상의 코딩 유닛 (CU) 들을 포함할 수도 있다. 비디오 인코더 (20) 는 트리블록들의 비디오 블록들을 CU 들과 연관된 비디오 블록들, 이 때문에, 명칭 "트리블록들" 로 파티셔닝하기 위하여 쿼드트리 파티셔닝을 이용할 수도 있다.

[0042] 일부의 예들에서, 비디오 인코더 (20) 는 픽처를 복수의 슬라이스들로 파티셔닝할 수도 있다. 슬라이스들의 각각은 정수 (integer number) 의 CU 들을 포함할 수도 있다. 일부의 사례들에서, 슬라이스는 정수의 트리블록들을 포함한다. 다른 사례들에서, 슬라이스의 경계는 트리블록 내에 있을 수도 있다.

[0043] 픽처에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 픽처의 각각의 슬라이스에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 슬라이스에 대해 인코딩 동작을 수행할 때, 비디오 인코더 (20) 는 슬라이스와 연관된 인코딩된 데이터를 생성할 수도 있다. 슬라이스와 연관된 인코딩된 데이터는 "코딩된 슬라이스" 로서 지정될 수도 있다.

- [0044] 코딩된 슬라이스를 생성하기 위하여, 비디오 인코더 (20)는 슬라이스에서의 각각의 트리블록에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20)가 트리블록에 대해 인코딩 동작을 수행할 때, 비디오 인코더 (20)는 코딩된 트리블록을 생성할 수도 있다. 코딩된 트리블록은 트리블록의 인코딩된 버전을 나타내는 데이터를 포함할 수도 있다.
- [0045] 비디오 인코더 (20)가 코딩된 슬라이스를 생성할 때, 비디오 인코더 (20)는 래스터 스캔 순서 (raster scan order)에 따라 슬라이스에서의 트리블록들에 대해 인코딩 동작들을 수행 (예컨대, 인코딩) 할 수도 있다. 예를 들어, 비디오 인코더 (20)가 슬라이스에서의 트리블록들의 각각을 인코딩하였을 때까지, 비디오 인코더 (20)는 슬라이스에서의 트리블록들의 최상부로우를 가로질러 좌측으로부터 우측으로, 그 다음으로, 트리블록들의 다음의 더 낮은 로우를 가로질러 좌측으로부터 우측으로, 등등으로 진행하는 순서로 슬라이스의 트리블록들을 인코딩할 수도 있다.
- [0046] 래스터 스캔 순서에 따라 트리블록들을 인코딩한 결과로서, 소정의 트리블록의 상부 및 좌측의 트리블록들이 인코딩되었을 수도 있지만, 소정의 트리블록의 하부 및 우측의 트리블록들은 인코딩되지 않았을 수도 있다. 결과적으로, 비디오 인코더 (20)는 소정의 트리블록을 인코딩할 때에 소정의 트리블록의 상부 및 좌측의 트리블록들을 인코딩함으로써 생성된 정보를 액세스할 수 있을 수도 있다. 그러나, 비디오 인코더 (20)는 소정의 트리블록을 인코딩할 때에 소정의 트리블록의 하부 및 우측의 트리블록들을 인코딩함으로써 생성된 정보를 액세스할 수 없을 수도 있다.
- [0047] 코딩된 트리블록을 생성하기 위하여, 비디오 인코더 (20)는 비디오 블록을 점진적으로 더 작은 비디오 블록들을 분할하기 위하여 트리블록의 비디오 블록에 대해 큐드트리 파티셔닝을 재귀적으로 수행할 수도 있다. 더 작은 비디오 블록들의 각각은 상이한 CU와 연관될 수도 있다. 예를 들어, 비디오 인코더 (20)는 트리블록의 비디오 블록을 4개의 동일한 사이즈의 서브-블록들로 파티셔닝할 수도 있고, 서브-블록들 중의 하나 이상을 4개의 동일한 사이즈의 서브-서브 블록 (sub-sub-block)들로 파티셔닝할 수도 있는 등등과 같다. 파티셔닝된 CU는, 그 비디오 블록이 다른 CU들과 연관된 비디오 블록들로 파티셔닝되는 CU일 수도 있다. 비-파티셔닝된 (non-partitioned) CU는, 그 비디오 블록이 다른 CU들과 연관된 비디오 블록들로 파티셔닝되지 않는 CU일 수도 있다.
- [0048] 비트스트림에서의 하나 이상의 신택스 엘리먼트들은 비디오 인코더 (20)가 트리블록의 비디오 블록을 파티셔닝할 수도 있는 최대 횟수를 표시할 수도 있다. CU의 비디오 블록은 형상에 있어서 정사각형일 수도 있다. CU의 비디오 블록의 사이즈 (예컨대, CU의 사이즈)는 8x8 픽셀들로부터, 64x64 픽셀들 이상의 최대치를 갖는 트리블록의 비디오 블록의 사이즈 (예컨대, 트리블록의 사이즈)까지의 범위일 수도 있다.
- [0049] 비디오 인코더 (20)는 z-스캔 순서에 따라 트리블록의 각각의 CU에 대해 인코딩 동작들을 수행 (예컨대, 인코딩) 할 수도 있다. 다시 말해서, 비디오 인코더 (20)는 상부-좌측 CU, 상부-우측 CU, 하부-좌측 CU, 및 그 다음으로, 하부-우측 CU를 그 순서로 인코딩할 수도 있다. 비디오 인코더 (20)가 파티셔닝된 CU에 대해 인코딩 동작을 수행할 때, 비디오 인코더 (20)는 z-스캔 순서에 따라 파티셔닝된 CU의 비디오 블록의 서브-블록들과 연관된 CU들을 인코딩할 수도 있다. 다시 말해서, 비디오 인코더 (20)는 상부-좌측 서브-블록과 연관된 CU, 상부-우측 서브-블록과 연관된 CU, 하부-좌측 서브-블록과 연관된 CU, 및 그 다음으로, 하부-우측 서브-블록과 연관된 CU를 그 순서로 인코딩할 수도 있다.
- [0050] z-스캔 순서에 따라 트리블록의 CU들을 인코딩한 결과로서, 소정의 CU의 상부, 상부-좌측, 상부-우측, 좌측, 및 하부-좌측의 CU들이 인코딩되었을 수도 있다. 소정의 CU의 하부 및 우측의 CU들은 아직 인코딩되지 않았다. 결과적으로, 비디오 인코더 (20)는 소정의 CU를 인코딩할 때에 소정의 CU와 이웃하는 일부의 CU들을 인코딩함으로써 생성된 정보를 액세스할 수 있을 수도 있다. 그러나, 비디오 인코더 (20)는 소정의 CU를 인코딩할 때에 소정의 CU와 이웃하는 다른 CU들을 인코딩함으로써 생성된 정보를 액세스할 수 없을 수도 있다.
- [0051] 비디오 인코더 (20)가 비-파티셔닝된 CU를 인코딩할 때, 비디오 인코더 (20)는 CU에 대해 하나 이상의 예측 유닛 (PU)들을 생성할 수도 있다. CU의 PU들의 각각은 CU의 비디오 블록 내의 상이한 비디오 블록과 연관될 수도 있다. 비디오 인코더 (20)는 CU의 각각의 PU에 대한 예측된 비디오 블록을 생성할 수도 있다. PU의 예측된 비디오 블록은 샘플들의 블록일 수도 있다. 비디오 인코더 (20)는 PU에 대한 예측된 비디오 블록을 생성하기 위하여 인트라 예측 또는 인터 예측을 이용할 수도 있다.
- [0052] 비디오 인코더 (20)가 PU의 예측된 비디오 블록을 생성하기 위하여 인트라 예측을 이용할 때, 비디오 인코더

(20) 는 PU 와 연관된 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측된 비디오 블록을 생성할 수도 있다. 비디오 인코더 (20) 가 CU 의 PU 들의 예측된 비디오 블록들을 생성하기 위하여 인트라 예측을 이용할 경우, CU 는 인트라-예측된 CU 이다. 비디오 인코더 (20) 가 PU 의 예측된 비디오 블록을 생성하기 위하여 인터 예측을 이용할 때, 비디오 인코더 (20) 는 PU 와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU 의 예측된 비디오 블록을 생성할 수도 있다. 비디오 인코더 (20) 가 CU 의 PU 들의 예측된 비디오 블록들을 생성하기 위하여 인터 예측을 이용할 경우, CU 는 인터-예측된 CU 이다.

[0053] 또한, 비디오 인코더 (20) 가 PU 에 대한 예측된 비디오 블록을 생성하기 위하여 인터 예측을 이용할 때, 비디오 인코더 (20) 는 PU 에 대한 모션 정보를 생성할 수도 있다. PU 에 대한 모션 정보는 PU 의 하나 이상의 참조 블록들을 표시할 수도 있다. PU 의 각각의 참조 블록은 참조 픽처 내의 비디오 블록일 수도 있다. 참조 픽처는 PU 와 연관된 픽처 이외의 픽처일 수도 있다. 일부의 사례들에서, PU 의 참조 블록은 또한, PU 의 "참조 샘플" 로서 지칭될 수도 있다. 비디오 인코더 (20) 는 PU 의 참조 블록들에 기초하여 PU 에 대한 예측된 비디오 블록을 생성할 수도 있다.

[0054] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU 들의 예측된 비디오 블록들을 생성한 후, 비디오 인코더 (20) 는 CU 의 PU 들에 대한 예측된 비디오 블록들에 기초하여 CU 에 대한 잔차 데이터를 생성할 수도 있다. CU 에 대한 잔차 데이터는 CU 의 PU 들에 대한 예측된 비디오 블록들에서의 샘플들과, CU 의 원래의 비디오 블록과의 사이의 차이들을 표시할 수도 있다.

[0055] 또한, 비-파티셔닝된 CU 에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 CU 의 잔차 데이터를 CU 의 변환 유닛 (TU) 들과 연관된 잔차 데이터의 하나 이상의 블록들 (예컨대, 잔차 비디오 블록들) 로 파티셔닝하기 위하여 CU 의 잔차 데이터에 대해 재귀적 쿼드트리 파티셔닝 (recursive quadtree partitioning) 을 수행할 수도 있다. CU 의 각각의 TU 는 상이한 잔차 비디오 블록과 연관될 수도 있다.

[0056] 비디오 인코더 (20) 는 TU 들과 연관된 변환 계수 블록들 (예컨대, 변환 계수들의 블록들) 을 생성하기 위하여, 하나 이상의 변환들을 TU 들과 연관된 잔차 비디오 블록들에 적용할 수도 있다. 결과적으로, 변환 계수 블록은 변환 계수들의 2 차원 (2D) 행렬일 수도 있다.

[0057] 변환 계수 블록을 생성한 후, 비디오 인코더 (20) 는 변환 계수 블록에 대해 양자화 프로세스를 수행할 수도 있다. 양자화는 일반적으로, 변환 계수들을 나타내기 위해 이용된 데이터의 양을 가능한 대로 감소시키기 위하여 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 변환 계수들의 일부 또는 전부와 연관된 비트 심도 (bit depth) 를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안에 m-비트 변환 계수로 라운드 (round down) 될 수도 있고, 여기서, n 은 m 보다 더 크다.

[0058] 비디오 인코더 (20) 는 각각의 CU 를 양자화 파라미터 (quantization parameter; QP) 값과 연관시킬 수도 있다. CU 와 연관된 QP 값은 비디오 인코더 (20) 가 CU 와 연관된 변환 계수 블록들을 어떻게 양자화하는지를 결정할 수도 있다. 비디오 인코더 (20) 는 CU 와 연관된 QP 값을 조절함으로써 CU 와 연관된 변환 계수 블록들에 적용된 양자화도 (degree of quantization) 를 조절할 수도 있다.

[0059] 비디오 인코더 (20) 가 변환 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수 블록에서 변환 계수들을 나타내는 신택스 엘리먼트들의 세트들을 생성할 수도 있다. 비디오 인코더 (20) 는 컨텍스트 적응 이진 산술 코딩 (Context Adaptive Binary Arithmetic Coding; CABAC) 동작들과 같은 엔트로피 인코딩 동작들을 이 신택스 엘리먼트들의 일부에 적용할 수도 있다. 컨텐츠 적응 가변 길이 코딩 (content adaptive variable length coding; CAVLC), 확률 간격 파티셔닝 엔트로피 (probability interval partitioning entropy; PIPE) 코딩, 또는 다른 이진 산술 코딩 (binary arithmetic coding) 과 같은 다른 엔트로피 코딩 기법들이 또한 이용될 수 있다.

[0060] 비디오 인코더 (20) 에 의해 생성된 비트스트림은 일련의 네트워크 추상화 계층 (Network Abstraction Layer; NAL) 유닛들을 포함할 수도 있다. NAL 유닛들의 각각은 NAL 유닛에서의 데이터의 탑입의 표시를 포함하는 신택스 구조와, 데이터를 포함하는 바이트들일 수도 있다. 예를 들어, NAL 유닛은 비디오 파라미터 세트, 시퀀스 파라미터 세트, 픽처 파라미터 세트, 코딩된 슬라이스, 보충 강화 정보 (supplemental enhancement information; SEI), 액세스 유닛 구분자 (access unit delimiter), 필러 데이터 (filler data), 또는 또 다른 탑입의 데이터를 나타내는 데이터를 포함할 수도 있다. NAL 유닛에서의 데이터는 다양한 신택스 구조들을 포함할 수도 있다.

[0061] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 비트스트림은

비디오 인코더 (20)에 의해 인코딩된 비디오 데이터의 코딩된 표현을 포함할 수도 있다. 비디오 디코더 (30)가 비트스트림을 수신할 때, 비디오 디코더 (30)는 비트스트림에 대해 파싱 동작 (parsing operation)을 수행할 수도 있다. 비디오 디코더 (30)가 파싱 동작을 수행할 때, 비디오 디코더 (30)는 비트스트림으로부터 신택스 엘리먼트들을 추출할 수도 있다. 비디오 디코더 (30)는 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 비디오 데이터의 블록들을 재구성할 수도 있다. 신택스 엘리먼트들에 기초하여 비디오 데이터를 재구성하기 위한 프로세스는 일반적으로, 신택스 엘리먼트들을 생성하기 위하여 비디오 인코더 (20)에 의해 수행된 프로세스와 상반적일 수도 있다.

[0062] 비디오 디코더 (30)가 CU와 연관된 신택스 엘리먼트들을 추출한 후, 비디오 디코더 (30)는 신택스 엘리먼트들에 기초하여 CU의 PU들에 대한 예측된 비디오 블록들을 생성할 수도 있다. 계다가, 비디오 디코더 (30)는 CU의 TU들과 연관된 변환 계수 블록들을 역자화할 수도 있다. 비디오 디코더 (30)는 CU의 TU들과 연관된 잔차 비디오 블록들을 재구성하기 위하여 변환 계수 블록들에 대해 역변환들을 수행할 수도 있다. 예측된 비디오 블록들을 생성하고 잔차 비디오 블록들을 재구성한 후, 비디오 디코더 (30)는 예측된 비디오 블록들 및 잔차 비디오 블록들에 기초하여 CU의 비디오 블록을 재구성할 수도 있다. 이러한 방법으로, 비디오 디코더 (30)는 비트스트림에서의 신택스 엘리먼트들에 기초하여 CU들의 비디오 블록들을 재구성할 수도 있다.

### 비디오 인코더

[0064] 도 2a는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 인코더의 예를 예시하는 블록도이다. 비디오 인코더 (20)는 HEVC에 대한 것과 같은 비디오 프레임의 단일 계층을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 인코더 (20)는, 도 4 및 도 5에 대하여 위에서 그리고 이하에서 더욱 상세하게 설명된 상이한 표준들 및 관련된 프로세스들을 이용하여 멀티-계층 코딩을 위한 지원의 방법들을 포함하지만, 이것으로 제한되지는 않는, 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 하나의 예로서, 예측 프로세싱 유닛 (100)은 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 또 다른 실시형태에서, 비디오 인코더 (20)는 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성되는 선택적인 인터-계층 예측 유닛 (128)을 포함한다.

다른 실시형태들에서, 인터-계층 예측은 예측 프로세싱 유닛 (100) (예컨대, 인터 예측 유닛 (121) 및/또는 인트라 예측 유닛 (126))에 의해 수행될 수 있고, 이 경우, 인터-계층 예측 유닛 (128)은 생략될 수도 있다.

그러나, 이 개시물의 양태들은 그렇게 제한되지는 않는다. 일부의 예들에서, 이 개시물에서 설명된 기법들은 비디오 인코더 (20)의 다양한 컴포넌트들 사이에서 공유될 수도 있다. 일부의 예들에서, 추가적으로 또는 대안적으로, 프로세서 (도시되지 않음)는 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0065] 설명의 목적들을 위하여, 이 개시물은 HEVC 코딩의 맥락에서 비디오 인코더 (20)를 설명한다. 그러나, 이 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 2a에서 도시된 예는 단일 계층 코덱에 대한 것이다. 그러나, 도 2b에 대하여 추가로 설명되는 바와 같이, 비디오 인코더 (20)의 일부 또는 전부는 멀티-계층 코덱의 프로세싱에 대해 중복될 수도 있다.

[0066] 비디오 인코더 (20)는 비디오 슬라이스들 내의 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라 코딩은 소정의 비디오 프레임 또는 블록 내의 비디오에 있어서의 공간적 중복성을 감소시키거나 제거하기 위하여 공간적 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 블록들 내의 비디오에 있어서의 시간적 중복성을 감소시키거나 제거하기 위하여 시간적 예측에 의존한다. 인트라-모드 (I 모드)는 몇몇 공간 기반 코딩 모드들 중의 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드)과 같은 인터-모드들은 몇몇 시간 기반 코딩 모드들 중의 임의의 것을 지칭할 수도 있다.

[0067] 도 2a의 예에서, 비디오 인코더 (20)는 복수의 기능적 컴포넌트들을 포함한다. 비디오 인코더 (20)의 기능적 컴포넌트들은 예측 프로세싱 유닛 (100), 잔차 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 유닛 (110), 재구성 유닛 (112), 필터 유닛 (113), 디코딩된 블록 버퍼 (114), 및 엔트로피 인코딩 유닛 (116)을 포함한다. 예측 프로세싱 유닛 (100)은 인터 예측 유닛 (121), 모션 추정 유닛 (122), 모션 보상 유닛 (124), 인트라 예측 유닛 (126), 및 인터-계층 예측 유닛 (128)을 포함한다. 다른 예들에서, 비디오 인코더 (20)는 더 많거나, 더 적거나, 또는 상이한 기능적 컴포넌트들을 포함할 수도 있다. 또한, 모션 추정 유닛 (122) 및 모션 보상 유닛 (124)은 고도로 통합될 수도 있지만, 설

명의 목적들을 위하여 별도로 **도 2a** 의 예에서 표현된다.

[0068] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 다양한 소스들로부터 비디오 데이터를 수신할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 (예컨대, **도 1a** 또는 **도 1b** 에서 도시된) 비디오 소스 (18) 또는 또 다른 소스로부터 비디오 데이터를 수신할 수도 있다. 비디오 데이터는 일련의 팩쳐들을 나타낼 수도 있다. 비디오 데이터를 인코딩하기 위하여, 비디오 인코더 (20) 는 팩쳐들의 각각에 대해 인코딩 동작을 수행할 수도 있다. 팩쳐에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 팩쳐의 각각의 슬라이스에 대해 인코딩 동작들을 수행할 수도 있다. 슬라이스에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 슬라이스에서의 트리블록들에 대해 인코딩 동작들을 수행할 수도 있다.

[0069] 트리블록에 대해 인코딩 동작을 수행하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 비디오 블록을 점진적으로 더 작은 비디오 블록들로 분할하기 위하여 트리블록의 비디오 블록에 대해 큐드트리 파티셔닝을 수행할 수도 있다. 더 작은 비디오 블록들의 각각은 상이한 CU 와 연관될 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 트리블록의 비디오 블록을 4 개의 동일한 사이즈의 서브-블록들로 파티셔닝할 수도 있고, 서브-블록들 중의 하나 이상을 4 개의 동일한 사이즈의 서브-서브-블록들로 파티셔닝할 수도 있는 등등과 같다.

[0070] CU 들과 연관된 비디오 블록들의 사이즈들은  $8 \times 8$  샘플들로부터,  $64 \times 64$  픽셀들 이상의 최대치를 갖는 트리블록의 사이즈까지의 범위일 수도 있다. 이 개시물에서, "NxN" 및 "N 대 (by) N" 은 수직 및 수평 차원들의 측면에서의 비디오 블록의 샘플 차원들, 예컨대,  $16 \times 16$  샘플들 또는 16 대 16 샘플들을 지칭하기 위하여 상호 교환 가능하게 이용될 수도 있다. 일반적으로,  $16 \times 16$  비디오 블록은 수직 방향에서의 16 개의 샘플들 ( $y = 16$ ) 및 수평 방향에서의 16 개의 샘플들 ( $x = 16$ ) 을 가진다. 마찬가지로, NxN 블록은 일반적으로 수직 방향에서의 N 개의 샘플들 및 수평 방향에서의 N 개의 샘플들을 가지며, 여기서, N 은 음이 아닌 정수 값을 나타낸다.

[0071] 또한, 트리블록에 대해 인코딩 동작을 수행하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 트리블록에 대한 계층적 큐드트리 데이터 구조를 생성할 수도 있다. 예를 들어, 트리블록은 큐드트리 데이터 구조의 루트 노드 (root node) 에 대응할 수도 있다. 예측 프로세싱 유닛 (100) 이 트리블록의 비디오 블록을 4 개의 서브-블록들로 파티셔닝할 경우, 루트 노드는 큐드트리 데이터 구조에서 4 개의 자식 노드 (child node) 들을 가진다. 자식 노드들의 각각은 서브-블록들 중의 하나와 연관된 CU 에 대응한다. 예측 프로세싱 유닛 (100) 이 서브-블록들 중의 하나를 4 개의 서브-서브-블록들로 파티셔닝할 경우, 서브-블록과 연관된 CU 에 대응하는 노드는 4 개의 자식 노드들을 가질 수도 있고, 그 각각은 서브-서브-블록들 중의 하나와 연관된 CU 에 대응한다.

[0072] 큐드트리 데이터 구조의 각각의 노드는 대응하는 트리블록 또는 CU 에 대한 신팩스 데이터 (예컨대, 신팩스 엘리먼트들) 를 포함할 수도 있다. 예를 들어, 큐드트리에서의 노드는, 노드에 대응하는 CU 의 비디오 블록이 4 개의 서브-블록들로 파티셔닝 (예컨대, 분할) 되는지 여부를 표시하는 분할 플래그 (split flag) 를 포함할 수도 있다. CU 에 대한 신팩스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU 의 비디오 블록이 서브-블록들로 분할되는지 여부에 종속될 수도 있다. 그 비디오 블록이 파티셔닝되지 않는 CU 는 큐드트리 데이터 구조에서의 리프 노드 (leaf node) 에 대응할 수도 있다. 코딩된 트리블록은 대응하는 트리블록에 대한 큐드트리 데이터 구조에 기초한 데이터를 포함할 수도 있다.

[0073] 비디오 인코더 (20) 는 트리블록의 각각의 비-파티셔닝된 CU 에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 비-파티셔닝된 CU 에 대해 인코딩 동작을 수행할 때, 비디오 인코더 (20) 는 비-파티셔닝된 CU 의 인코딩된 표현을 나타내는 데이터를 생성한다.

[0074] CU 에 대해 인코딩 동작을 수행하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU 들 중에서 CU 의 비디오 블록을 파티셔닝할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다양한 PU 사이즈들을 지원할 수도 있다. 특정한 CU 의 사이즈가  $2N \times 2N$  인 것으로 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는  $2N \times 2N$  또는  $N \times N$  의 PU 사이즈들과,  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$ ,  $2N \times nU$ ,  $nL \times 2N$ ,  $nR \times 2N$ , 또는 유사한 것의 대칭적인 PU 사이즈들에서의 인터-예측을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한,  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$ , 및  $nR \times 2N$  의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다. 일부의 예들에서, 예측 프로세싱 유닛 (100) 은, CU 의 비디오 블록의 측면들과 직각으로 만나지 않는 경계를 따라 CU 의 PU 들 중에서 CU 의 비디오 블록을 파티셔닝하기 위하여 기하학적 파티셔닝 (geometric partitioning) 을 수행할 수도 있다.

- [0075] 인터 예측 유닛 (121)은 CU의 각각의 PU에 대해 인터 예측을 수행할 수도 있다. 인터 예측은 시간적 압축을 제공할 수도 있다. PU에 대해 인터 예측을 수행하기 위하여, 모션 추정 유닛 (122)은 PU에 대한 모션 정보를 생성할 수도 있다. 모션 보상 유닛 (124)은 CU와 연관된 픽처 이외의 픽처들(예컨대, 참조 픽처들)의 디코딩된 샘플들 및 모션 정보에 기초하여 PU에 대한 예측된 비디오 블록을 생성할 수도 있다. 이 개시물에서, 모션 보상 유닛 (124)에 의해 생성된 예측된 비디오 블록은 인터-예측된 비디오 블록으로서 지칭될 수도 있다.
- [0076] 슬라이스들은 I 슬라이스들, P 슬라이스들, 또는 B 슬라이스들일 수도 있다. 모션 추정 유닛 (122) 및 모션 보상 유닛 (124)은 PU가 I 슬라이스, P 슬라이스, 또는 B 슬라이스에 있는지 여부에 따라 CU의 PU에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들은 인트라 예측된다. 이 때문에, PU가 I 슬라이스에 있을 경우, 모션 추정 유닛 (122) 및 모션 보상 유닛 (124)은 PU에 대해 인터 예측을 수행하지 않는다.
- [0077] PU가 P 슬라이스에 있을 경우, PU를 포함하는 픽처는 "리스트 0"으로서 지칭된 참조 픽처들의 리스트와 연관된다. 리스트 0에서의 참조 픽처들의 각각은 다른 픽처들의 인터 예측을 위하여 이용될 수도 있는 샘플들을 포함한다. 모션 추정 유닛 (122)이 P 슬라이스에서의 PU에 대하여 모션 추정 동작을 수행할 때, 모션 추정 유닛 (122)은 PU에 대한 참조 블록을 위하여 리스트 0에서의 참조 픽처들을 검색할 수도 있다. PU의 참조 블록은, PU의 비디오 블록에서의 샘플들에 가장 근접하게 대응하는 샘플들의 세트, 예컨대, 샘플들의 블록일 수도 있다. 모션 추정 유닛 (122)은 참조 픽처에서의 샘플들의 세트가 PU의 비디오 블록에서의 샘플들에 얼마나 근접하게 대응하는지를 결정하기 위하여 다양한 메트릭 (metric)들을 이용할 수도 있다. 예를 들어, 모션 추정 유닛 (122)은 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해, 참조 픽처에서의 샘플들의 세트가 PU의 비디오 블록에서의 샘플들에 얼마나 근접하게 대응하는지를 결정할 수도 있다.
- [0078] P 슬라이스에서의 PU의 참조 블록을 식별한 후, 모션 추정 유닛 (122)은 참조 블록을 포함하는 리스트 0에서 참조 픽처를 표시하는 참조 인덱스와, PU와 참조 블록 사이의 공간적 변위를 표시하는 모션 벡터를 생성할 수도 있다. 다양한 예들에서, 모션 추정 유닛 (122)은 변동되는 정밀도의 정도들에 대한 모션 벡터들을 생성할 수도 있다. 예를 들어, 모션 추정 유닛 (122)은 1/4 샘플 정밀도, 1/8 샘플 정밀도, 또는 다른 분수 샘플 정밀도로 모션 벡터들을 생성할 수도 있다. 분수 샘플 정밀도의 경우, 참조 블록 값들은 참조 픽처에서의 정수-위치 샘플 값들로부터 보간 (interpolate) 될 수도 있다. 모션 추정 유닛 (122)은 참조 인덱스 및 모션 벡터를 PU의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124)은 PU의 모션 정보에 의해 식별된 참조 블록에 기초하여 PU의 예측된 비디오 블록을 생성할 수도 있다.
- [0079] PU가 B 슬라이스에 있을 경우, PU를 포함하는 픽처는 "리스트 0" 및 "리스트 1"로서 지칭된, 참조 픽처들의 2개의 리스트들과 연관될 수도 있다. 일부의 예들에서, B 슬라이스를 포함하는 픽처는 리스트 0 및 리스트 1의 조합인 리스트 조합과 연관될 수도 있다.
- [0080] 또한, PU가 B 슬라이스에 있을 경우, 모션 추정 유닛 (122)은 PU에 대한 단방향 예측 또는 양방향 예측을 수행할 수도 있다. 모션 추정 유닛 (122)이 PU에 대한 단방향 예측을 수행할 때, 모션 추정 유닛 (122)은 PU에 대한 참조 블록을 위하여 리스트 0 또는 리스트 1의 참조 픽처들을 검색할 수도 있다. 다음으로, 모션 추정 유닛 (122)은 참조 블록을 포함하는 리스트 0 또는 리스트 1에서 참조 픽처를 표시하는 참조 인덱스와, PU와 참조 블록 사이의 공간적 변위를 표시하는 모션 벡터를 생성할 수도 있다. 모션 추정 유닛 (122)은 참조 인덱스, 예측 방향 표시자, 및 모션 벡터를 PU의 모션 정보로서 출력할 수도 있다. 예측 방향 표시자는 참조 인덱스가 리스트 0 또는 리스트 1에서 참조 픽처를 표시하는지 여부를 표시할 수도 있다. 모션 보상 유닛 (124)은 PU의 모션 정보에 의해 표시된 참조 블록에 기초하여 PU의 예측된 비디오 블록을 생성할 수도 있다.
- [0081] 모션 추정 유닛 (122)이 PU에 대한 양방향 예측을 수행할 때, 모션 추정 유닛 (122)은 PU에 대한 참조 블록을 위하여 리스트 0에서 참조 픽처들을 검색할 수도 있고, 또한, PU에 대한 또 다른 참조 블록을 위하여 리스트 1에서 참조 픽처들을 검색할 수도 있다. 다음으로, 모션 추정 유닛 (122)은 참조 블록들을 포함하는 리스트 0 또는 리스트 1에서 참조 픽처들을 표시하는 참조 인덱스들과, 참조 블록들과 PU 사이의 공간적 변위들을 표시하는 모션 벡터들을 생성할 수도 있다. 모션 추정 유닛 (122)은 PU의 참조 인덱스들 및 모션 벡터들을 PU의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124)은 PU의 모션 정보에 의해 표시된 참조 블록들에 기초하여 PU의 예측된 비디오 블록을 생성할 수도 있다.

- [0082] 일부의 사례들에서, 모션 추정 유닛 (122) 은 PU 에 대한 모션 정보의 전체 세트를 엔트로피 인코딩 유닛 (116) 으로 출력하지는 않는다. 오히려, 모션 추정 유닛 (122) 은 또 다른 PU 의 모션 정보를 참조하여 PU 의 모션 정보를 시그널링할 수도 있다. 예를 들어, 모션 추정 유닛 (122) 은 PU 의 모션 정보가 이웃하는 PU 의 모션 정보와 충분히 유사한 것으로 결정할 수도 있다. 이 예에서, 모션 추정 유닛 (122) 은 PU 와 연관된 십택스 구조에서, PU 가 이웃하는 PU 와 동일한 모션 정보를 가지는 것을 비디오 디코더 (30) 에 표시하는 값을 표시할 수도 있다. 또 다른 예에서, 모션 추정 유닛 (122) 은 PU 와 연관된 십택스 구조에서, 이웃하는 PU 및 모션 벡터 차이 (motion vector difference; MVD) 를 식별할 수도 있다. 모션 벡터 차이는 PU 의 모션 벡터와, 표시된 이웃하는 PU 의 모션 벡터와의 사이의 차이를 표시한다. 비디오 디코더 (30) 는 PU 의 모션 벡터를 결정하기 위하여 표시된 이웃하는 PU 의 모션 벡터 및 모션 벡터 차이를 이용할 수도 있다. 제 2 PU 의 모션 정보를 시그널링할 때에 제 1 PU 의 모션 정보를 참조함으로써, 비디오 인코더 (20) 는 더 적은 비트들을 이용하여 제 2 PU 의 모션 정보를 시그널링할 수 있을 수도 있다.
- [0083] 도 5 를 참조하여 이하에서 추가로 논의된 바와 같이, 예측 프로세싱 유닛 (100) 은 도 5 에서 예시된 방법들을 수행함으로써 PU (또는 임의의 다른 참조 계층 및/또는 강화 계층 블록들 또는 비디오 유닛들) 를 코딩 (예컨대, 인코딩 또는 디코딩) 하도록 구성될 수도 있다. 예를 들어, (예컨대, 모션 추정 유닛 (122) 및/또는 모션 보상 유닛 (124) 을 통한) 인터 예측 유닛 (121), 인트라 예측 유닛 (126), 또는 인터-계층 예측 유닛 (128) 은 도 5 에서 예시된 방법들을 모두 함께 또는 별도로 수행하도록 구성될 수도 있다.
- [0084] CU 에 대해 인코딩 동작을 수행하는 것의 일부로서, 인트라 예측 유닛 (126) 은 CU 의 PU 들에 대해 인트라 예측을 수행할 수도 있다. 인트라 예측은 공간적 압축을 제공할 수도 있다. 인트라 예측 유닛 (126) 이 PU 에 대해 인트라 예측을 수행할 때, 인트라 예측 유닛 (126) 은 동일한 픽처에서의 다른 PU 들의 디코딩된 샘플들에 기초하여 PU 에 대한 예측 데이터를 생성할 수도 있다. PU 에 대한 예측 데이터는 예측된 비디오 블록 및 다양한 십택스 엘리먼트들을 포함할 수도 있다. 인트라 예측 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU 들에 대해 인트라 예측을 수행할 수도 있다.
- [0085] PU 에 대해 인트라 예측을 수행하기 위하여, 인트라-예측 프로세싱 유닛 (126) 은 PU 에 대한 예측 데이터의 다수의 세트들을 생성하기 위하여 다수의 인트라 예측 모드들을 이용할 수도 있다. 인트라 예측 유닛 (126) PU 에 대한 예측 데이터의 세트를 생성하기 위하여 인트라 예측 모드를 이용할 때, 인트라 예측 유닛 (126) 은 인트라 예측 모드와 연관된 방향 및/또는 경도 (gradient) 로 PU 의 비디오 블록을 가로질러 이웃하는 PU 들의 비디오 블록들로부터 샘플들을 확장할 수도 있다. PU 들, CU 들, 및 트리블록들에 대한 좌측에서 우측, 상부에서 하부의 인코딩 순서를 가정하면, 이웃하는 PU 들은 PU 의 상부, 상부 및 우측, 상부 및 좌측, 또는 좌측 일 수도 있다. 인트라 예측 유닛 (126) 은 PU 의 사이즈에 따라, 다양한 개수들의 인트라 예측 모드들, 예컨대, 33 개의 방향성 인트라 예측 모드 (directional intra prediction mode) 들을 이용할 수도 있다.
- [0086] 예측 프로세싱 유닛 (100) 은 PU 에 대한 모션 보상 유닛 (124) 에 의해 생성된 예측 데이터, 또는 PU 에 대한 인트라 예측 유닛 (126) 에 의해 생성된 예측 데이터 중으로부터 PU 에 대한 예측 데이터를 선택할 수도 있다. 일부의 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 PU 에 대한 예측 데이터를 선택한다.
- [0087] 예측 프로세싱 유닛 (100) 이 인트라 예측 유닛 (126) 에 의해 생성된 예측 데이터를 선택할 경우, 예측 프로세싱 유닛 (100) 은 PU 들에 대한 예측 데이터를 생성하기 위하여 이용되었던 인트라 예측 모드, 예컨대, 선택된 인트라 예측 모드를 시그널링할 수도 있다. 예측 프로세싱 유닛 (100) 은 선택된 인트라 예측 모드를 다양한 방법들로 시그널링할 수도 있다. 예를 들어, 선택된 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일할 가능성이 있을 수도 있다. 다시 말해서, 이웃하는 PU 의 인트라 예측 모드는 현재의 PU 에 대한 가장 가능성 있는 모드일 수도 있다. 이에 따라, 예측 프로세싱 유닛 (100) 은 선택된 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일하다는 것을 표시하기 위한 십택스 엘리먼트를 생성할 수도 있다.
- [0088] 위에서 논의된 바와 같이, 비디오 인코더 (20) 는 인터-계층 예측 유닛 (128) 을 포함할 수도 있다. 인터-계층 예측 유닛 (128) 은 SVC 에서 이용가능한 하나 이상의 상이한 계층들 (예컨대, 기본 또는 참조 계층) 을 이용하여 현재의 블록 (예컨대, EL 에서의 현재의 블록) 을 예측하도록 구성된다. 이러한 예측은 인터-계층 예측으로서 지칭될 수도 있다. 인터-계층 예측 유닛 (128) 은 인터-계층 중복성을 감소시키기 위한 예측 방법들을 사용함으로써, 코딩 효율을 개선시키고 연산 자원 요건들을 감소시킨다. 인터-계층 예측의 일부의 예들은 인터-계층 인트라 예측, 인터-계층 모션 예측, 및 인터-계층 잔차 예측을 포함한다. 인터-계층 인트라 예측은 강화 계층에서 현재의 블록을 예측하기 위하여 기본 계층에서의 공동-위치된 블록들의 재구성을 이용

한다. 인터-계층 모션 예측은 강화 계층에서 모션을 예측하기 위하여 기본 계층의 모션 정보를 이용한다. 인터-계층 잔차 예측은 강화 계층의 잔차를 예측하기 위하여 기본 계층의 잔차를 이용한다. 인터-계층 예측 방식들의 각각은 이하에서 더욱 상세하게 논의된다.

[0089] 예측 프로세싱 유닛 (100) 이 CU 의 PU 들에 대한 예측 데이터를 선택한 후, 잔차 생성 유닛 (102) 은 CU 의 비디오 블록으로부터 CU 의 PU 들의 예측된 비디오 블록들을 감산함으로써 (예컨대, 마이너스 부호에 의해 표시됨) CU 에 대한 잔차 데이터를 생성할 수도 있다. CU 의 잔차 데이터는 CU 의 비디오 블록에서 샘플들 의 상이한 샘플 컴포넌트들에 대응하는 2D 잔차 비디오 블록들을 포함할 수도 있다. 예를 들어, 잔차 데이터는 CU 의 PU 들의 예측된 비디오 블록들에서의 샘플들의 휘도 컴포넌트들과, CU 의 원래의 비디오 블록에서의 샘플들의 휘도 컴포넌트들과의 사이의 차이들에 대응하는 잔차 비디오 블록을 포함할 수도 있다. 게다가, CU 의 잔차 데이터는 CU 의 PU 들의 예측된 비디오 블록들에서의 샘플들의 색차 컴포넌트들과, CU 의 원래의 비디오 블록에서의 샘플들의 색차 컴포넌트들과의 사이의 차이들에 대응하는 잔차 비디오 블록들을 포함할 수도 있다.

[0090] 예측 프로세싱 유닛 (100) 은 CU 의 잔차 비디오 블록들을 서브-블록들로 파티셔닝하기 위하여 큐드트리 파티셔닝을 수행할 수도 있다. 각각의 분할되지 않은 잔차 비디오 블록은 CU 의 상이한 TU 와 연관될 수도 있다.

CU 의 TU들과 연관된 잔차 비디오 블록들의 사이즈들 및 위치들은 CU 의 PU 들과 연관된 비디오 블록들의 사이즈들 및 위치들에 기초할 수도 있거나 기초하지 않을 수도 있다. "잔차 큐드 트리" (residual quad tree; RQT) 로서 알려진 큐드트리 구조는 잔차 비디오 블록들의 각각과 연관된 노드들을 포함할 수도 있다. CU 의 TU 들은 RQT 의 리프 노드들에 대응할 수도 있다.

[0091] 변환 프로세싱 유닛 (104) 은 하나 이상의 변환들을 TU 와 연관된 잔차 비디오 블록에 적용함으로써 CU 의 각각의 TU 에 대한 하나 이상의 변환 계수 블록들을 생성할 수도 있다. 변환 계수 블록들의 각각은 변환 계수들 의 2D 행렬일 수도 있다. 변환 프로세싱 유닛 (104) 은 다양한 변환들을 TU 와 연관된 잔차 비디오 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 이산 코사인 변환 (discrete cosine transform; DCT), 방향성 변환 (directional transform), 또는 개념적으로 유사한 변환을 TU 와 연관된 잔차 비디오 블록에 적용할 수도 있다.

[0092] 변환 프로세싱 유닛 (104) 이 TU 와 연관된 변환 계수 블록을 생성한 후, 양자화 유닛 (106) 은 변환 계수 블록에서 변환 계수들을 양자화할 수도 있다. 양자화 유닛 (106) 은 CU 와 연관된 QP 값에 기초하여 CU 의 TU 와 연관된 변환 계수 블록을 양자화할 수도 있다.

[0093] 비디오 인코더 (20) 는 QP 값을 CU 와 다양한 방법들로 연관시킬 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 와 연관된 트리블록에 대해 레이트-왜곡 분석을 수행할 수도 있다. 레이트-왜곡 분석에서, 비디오 인코더 (20) 는 트리블록에 대해 인코딩 동작을 다수 회 수행함으로써 트리블록의 다수의 코딩된 표현들을 생성할 수도 있다. 비디오 인코더 (20) 가 트리블록의 상이한 인코딩된 표현들을 생성할 때, 비디오 인코더 (20) 는 상이한 QP 값을 CU 와 연관시킬 수도 있다. 비디오 인코더 (20) 는, 소정의 QP 값이 최저 비트레이트 및 왜곡 메트릭을 가지는 트리블록의 코딩된 표현에서 CU 와 연관될 때에 소정의 QP 값이 CU 와 연관되는 것을 시그널링할 수도 있다.

[0094] 역양자화 유닛 (108) 및 역변환 유닛 (110) 은 변환 계수 블록으로부터 잔차 비디오 블록을 재구성하기 위하여, 역양자화 및 역변환들을 변환 계수 블록에 각각 적용할 수도 있다. 재구성 유닛 (112) 은 TU 와 연관된 재구성된 비디오 블록을 생성하기 위하여, 재구성된 잔차 비디오 블록을, 예측 프로세싱 유닛 (100) 에 의해 생성된 하나 이상의 예측된 비디오 블록들로부터의 대응하는 샘플들에 추가할 수도 있다. CU 의 각각의 TU 에 대한 비디오 블록들을 이러한 방법으로 재구성함으로써, 비디오 인코더 (20) 는 CU 의 비디오 블록을 재구성할 수도 있다.

[0095] 재구성 유닛 (112) 이 CU 의 비디오 블록을 재구성한 후, 필터 유닛 (113) 은 CU 와 연관된 비디오 블록에서 블록킹 아티팩트들을 감소시키기 위하여 디블록킹 동작을 수행할 수도 있다. 하나 이상의 디블록킹 동작들을 수행한 후, 필터 유닛 (113) 은 CU 의 재구성된 비디오 블록을 디코딩된 픽처 버퍼 (114) 내에 저장할 수도 있다. 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 후속 픽처들의 PU 들에 대해 인터 예측을 수행하기 위하여 재구성된 비디오 블록을 포함하는 참조 픽처를 이용할 수도 있다. 게다가, 인트라 예측 유닛 (126) 은 CU 와 동일한 픽처에서의 다른 PU 들에 대해 인트라 예측을 수행하기 위하여, 디코딩된 픽처 버퍼 (114) 에서의 재구성된 비디오 블록들을 이용할 수도 있다.

[0096]

엔트로피 인코딩 유닛 (116) 은 비디오 인코더 (20) 의 다른 기능적 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (116) 은 양자화 유닛 (106) 으로부터 변환 계수 블록들을 수신할 수도 있고, 예측 프로세싱 유닛 (100) 으로부터 신팩스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (116) 이 데이터를 수신할 때, 엔트로피 인코딩 유닛 (116) 은 엔트로피 인코딩된 데이터를 생성하기 위하여 하나 이상의 엔트로피 인코딩 동작들을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 데이터에 대해, 컨텍스트 적응 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, 가변-대-가변 (variable-to-variable; V2V) 길이 코딩 동작, 신팩스-기반 컨텍스트-적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 또는 또 다른 타입의 엔트로피 인코딩 동작을 수행할 수도 있다. 엔트로피 인코딩 유닛 (116) 은 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다.

[0097]

데이터에 대해 엔트로피 인코딩 동작을 수행하는 것의 일부로서, 엔트로피 인코딩 유닛 (116) 은 컨텍스트 모델을 선택할 수도 있다. 엔트로피 인코딩 유닛 (116) 이 CABAC 동작을 수행하고 있을 경우, 컨텍스트 모델은 특정 값들을 가지는 특정 빈 (bin) 들의 확률들의 추정치들을 표시할 수도 있다. CABAC 의 맥락에서, 용어 "빈" 은 신팩스 엘리먼트의 이진화된 버전의 비트를 지칭하기 위하여 이용된다.

[0098]

### 멀티-계층 비디오 인코더

[0099]

도 2b 는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 (또한, 간단하게 비디오 인코더 (23) 로서 지칭된) 멀티-계층 비디오 인코더 (23) 의 예를 예시하는 블록도이다. 비디오 인코더 (23) 는 SHVC 및 멀티뷰 코딩을 위한 것과 같은 멀티-계층 비디오 프레임들을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 인코더 (23) 는 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0100]

비디오 인코더 (23) 는 비디오 인코더 (20A) 및 비디오 인코더 (20B) 를 포함하고, 그 각각은 비디오 인코더 (20) 로서 구성될 수도 있으며 비디오 인코더 (20) 에 대하여 위에서 설명된 기능들을 수행할 수도 있다. 또한, 참조 번호들의 재이용에 의해 표시된 바와 같이, 비디오 인코더들 (20A 및 20B) 은 비디오 인코더 (20) 로서 시스템들 및 서브시스템들의 적어도 일부를 포함할 수도 있다. 비디오 인코더 (23) 는 2 개의 비디오 인코더들 (20A 및 20B) 을 포함하는 것으로서 예시되어 있지만, 비디오 인코더 (23) 는 이와 같이 제한되지 않으며 임의의 수의 비디오 인코더 (20) 계층들을 포함할 수도 있다. 일부의 실시형태들에서, 비디오 인코더 (23) 는 액세스 유닛에서의 각각의 픽처 또는 프레임에 대한 비디오 인코더 (20) 를 포함할 수도 있다. 예를 들어, 5 개의 픽처들을 포함하는 액세스 유닛은 5 개의 인코더 계층들을 포함하는 비디오 인코더에 의해 프로세싱되거나 인코딩될 수도 있다. 일부의 실시형태들에서, 비디오 인코더 (23) 는 액세스 유닛에서의 프레임들보다 더 많은 인코더 계층들을 포함할 수도 있다. 일부의 이러한 경우들에 있어서, 비디오 인코더 계층들의 일부는 일부의 액세스 유닛들을 프로세싱할 때에 비활성 (inactive) 일 수도 있다.

[0101]

비디오 인코더들 (20A 및 20B) 에 추가하여, 비디오 인코더 (23) 는 리샘플링 유닛 (resampling unit; 90) 을 포함할 수도 있다. 리샘플링 유닛 (90) 은 일부의 경우들에 있어서, 예를 들어, 강화 계층을 생성하기 위하여, 수신된 비디오 프레임의 기본 계층을 업샘플링할 수도 있다. 리샘플링 유닛 (90) 은 프레임의 수신된 기본 계층과 연관된 특정 정보를 업샘플링할 수도 있지만, 다른 정보는 그러하지 않다. 예를 들어, 리샘플링 유닛 (90) 은 기본 계층의 픽셀들의 공간적 사이즈 또는 수를 업샘플링할 수도 있지만, 슬라이스들의 수 또는 픽처 순서 카운트는 일정하게 유지될 수도 있다. 일부의 경우들에는, 리샘플링 유닛 (90) 이 수신된 비디오를 프로세싱하지 않을 수도 있고 및/또는 선택적일 수도 있다. 예를 들어, 일부의 경우들에는, 예측 프로세싱 유닛 (100) 이 업샘플링 (upsampling) 을 수행할 수도 있다. 일부의 실시형태들에서, 리샘플링 유닛 (90) 은 계층을 업샘플링하며, 슬라이스 경계 규칙들 및/또는 래스터 스캔 (raster scan) 규칙들의 세트를 따르기 위하여 하나 이상의 슬라이스들을 재편성하거나, 재정의하거나, 수정하거나, 조절하도록 구성된다. 액세스 유닛에서의 기본 계층 또는 더 낮은 계층을 업샘플링하는 것으로서 주로 설명되지만, 일부의 경우들에는, 리샘플링 유닛 (90) 이 계층을 다운샘플링 (downsampling) 할 수도 있다. 예를 들어, 비디오의 스트리밍 동안에 대역폭이 감소될 경우, 프레임은 업샘플링되는 대신에 다운샘플링될 수도 있다.

[0102]

리샘플링 유닛 (90) 은 더 낮은 계층의 인코더 (예컨대, 비디오 인코더 (20A)) 의 디코딩된 픽처 버퍼 (114) 로부터 픽처 또는 프레임 (또는 픽처와 연관된 픽처 정보) 을 수신하고 픽처 (또는 수신된 픽처 정보) 를 업샘플링하도록 구성될 수도 있다. 다음으로, 이 업샘플링된 픽처는 더 낮은 계층의 인코더와 동일한 액세스 유닛에서의 픽처를 인코딩하도록 구성된 더 높은 계층의 인코더 (예컨대, 비디오 인코더 (20B)) 의 예측 프로세싱 유닛 (100) 에 제공될 수도 있다. 일부의 경우들에는, 더 높은 계층의 인코더가 더 낮은 계층의 인코더로부터

터 제거된 하나의 계층이다. 다른 경우에는, 도 2b 의 계층 0 비디오 인코더와 계층 1 인코더 사이에 하나 이상의 더 높은 계층의 인코더들이 있을 수도 있다.

[0103] 일부의 경우에는, 리샘플링 유닛 (90) 이 생략되거나 우회될 수도 있다. 이러한 경우에는, 비디오 인코더 (20A) 의 디코딩된 픽처 버퍼 (114)로부터의 픽처가 직접적으로, 또는 적어도 리샘플링 유닛 (90) 에 제공되지 않으면서, 비디오 인코더 (20B) 의 예측 프로세싱 유닛 (100) 에 제공될 수도 있다. 예를 들어, 비디오 인코더 (20B) 에 제공된 비디오 데이터와, 비디오 인코더 (20A) 의 디코딩된 픽처 버퍼 (114)로부터의 참조 픽처가 동일한 사이즈 또는 해상도일 경우, 참조 픽처는 임의의 리샘플링 없이 비디오 인코더 (20B) 에 제공될 수도 있다.

[0104] 일부의 실시형태들에서, 비디오 인코더 (23) 는 비디오 데이터를 비디오 인코더 (20A) 에 제공하기 전에, 다운샘플링 유닛 (94) 을 이용하여 더 낮은 계층의 인코더에 제공되어야 할 비디오 데이터를 다운샘플링한다. 대안적으로, 다운샘플링 유닛 (94) 은 비디오 데이터를 업샘플링하거나 다운샘플링할 수 있는 리샘플링 유닛 (90) 일 수도 있다. 또 다른 실시형태들에서는, 다운샘플링 유닛 (94) 이 생략될 수도 있다.

[0105] 도 2b 에서 예시된 바와 같이, 비디오 인코더 (23) 는 멀티플렉서 (98) 또는 멀티 (mux) 를 더 포함할 수도 있다. 멀티 (98) 는 비디오 인코더 (23)로부터의 조합된 비트스트림을 출력할 수 있다. 조합된 비트스트림은 비디오 인코더들 (20A 및 20B) 의 각각으로부터의 비트스트림을 취합으로써, 그리고 어느 비트스트림이 소정의 시간에 출력되는지를 교대시킴으로써 생성될 수도 있다. 일부의 경우에는, 2 개의 (또는 2 개를 초과하는 비디오 인코더 계층들의 경우에는 더 많음) 비트스트림들로부터의 비트들이 한 번에 하나의 비트씩 교대될 수도 있지만, 많은 경우에는 비트스트림들이 상이하게 조합된다. 예를 들어, 출력 비트스트림은 선택된 비트스트림을 한 번에 하나의 블록씩 교대시킴으로써 생성될 수도 있다. 또 다른 예에서, 출력 비트스트림은 비디오 인코더들 (20A 및 20B) 의 각각으로부터의 1:1 이 아닌 (non-1:1) 비율의 블록들을 출력함으로써 생성될 수도 있다. 예를 들어, 2 개의 블록들은 비디오 인코더 (20A)로부터 출력된 각각의 블록에 대하여 비디오 인코더 (20B)로부터 출력될 수도 있다. 일부의 실시형태들에서, 멀티 (98)로부터의 출력 스트림은 미리 프로그래밍될 수도 있다. 다른 실시형태들에서, 멀티 (98) 는 소스 디바이스 (12) 를 포함하는 소스 디바이스 상의 프로세서로부터와 같이, 비디오 인코더 (23) 의 외부의 시스템으로부터 수신된 제어 신호에 기초하여 비디오 인코더들 (20A, 20B)로부터의 비트스트림들을 조합할 수도 있다. 제어 신호는 비디오 소스 (18)로부터의 비디오의 해상도 또는 비트레이트에 기초하여, 링크 (16) 의 대역폭에 기초하여, 사용자와 연관된 가입 (예컨대, 유료 가입 대 무료 가입) 에 기초하여, 또는 비디오 인코더 (23)로부터 희망된 해상도 출력을 결정하기 위한 임의의 다른 인자에 기초하여 생성될 수도 있다.

## 비디오 디코더

[0107] 도 3a 는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 디코더의 예를 예시하는 블록도이다. 비디오 디코더 (30) 는 HEVC 에 대한 것과 같은 비디오 프레임의 단일 계층을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 디코더 (30) 는, 도 4 및 도 5 에 대하여 위에서 그리고 이하에서 더욱 상세하게 설명된 상이한 표준들 및 관련된 프로세스들을 이용하여 멀티-계층 코딩을 위한 지원의 방법들을 포함하지만, 이것으로 제한되지는 않는, 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 하나의 예로서, 모션 보상 유닛 (162) 및/또는 인트라 예측 유닛 (164) 은 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 하나의 실시형태에서, 비디오 디코더 (30) 는 선택적으로, 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성되는 인터-계층 예측 유닛 (166) 을 포함할 수도 있다. 다른 실시형태들에서, 인터-계층 예측은 예측 프로세싱 유닛 (152) (예컨대, 모션 보상 유닛 (162) 및/또는 인트라 예측 유닛 (164)) 에 의해 수행될 수 있고, 이 경우, 인터-계층 예측 유닛 (166) 은 생략될 수도 있다. 그러나, 이 개시물의 양태들은 그렇게 제한되지는 않는다. 일부의 예들에서, 이 개시물에서 설명된 기법들은 비디오 디코더 (30) 의 다양한 컴포넌트들 사이에서 공유될 수도 있다. 일부의 예들에서, 추가적으로 또는 대안적으로, 프로세서 (도시되지 않음) 는 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0108] 설명의 목적들을 위하여, 이 개시물은 HEVC 코딩의 맥락에서 비디오 디코더 (30) 를 설명한다. 그러나, 이 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 3a 에서 도시된 예는 단일 계층 코덱에 대한 것이다. 그러나, 도 3b 에 대하여 추가로 설명되는 바와 같이, 비디오 디코더 (30) 의 일부 또는 전부는 멀티-계층 코덱의 프로세싱에 대해 중복될 수도 있다.

[0109] 도 3a 의 예에서, 비디오 디코더 (30) 는 복수의 기능적 컴포넌트들을 포함한다. 비디오 디코더 (30) 의 기

능적 컴포넌트들은 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 유닛 (156), 재구성 유닛 (158), 필터 유닛 (159), 및 디코딩된 픽처 버퍼 (160)를 포함한다. 예측 프로세싱 유닛 (152)은 모션 보상 유닛 (162), 인트라 예측 유닛 (164), 및 인터-계층 예측 유닛 (166)을 포함한다. 일부의 예들에서, 비디오 디코더 (30)는 **도 2a**의 비디오 인코더 (20)에 대하여 설명된 인코딩 패스 (encoding pass)와 일반적으로 상반되는 디코딩 패스 (decoding pass)를 수행할 수도 있다. 다른 예들에서, 비디오 디코더 (30)는 더 많거나, 더 적거나, 또는 상이한 기능적 컴포넌트들을 포함할 수도 있다.

[0110] 비디오 디코더 (30)는 인코딩된 비디오 데이터를 포함하는 비트스트림을 수신할 수도 있다. 비트스트림은 복수의 신팩스 엘리먼트들을 포함할 수도 있다. 비디오 디코더 (30)가 비트스트림을 수신할 때, 엔트로피 디코딩 유닛 (150)은 비트스트림에 대해 파싱 동작을 수행할 수도 있다. 비트스트림에 대해 파싱 동작을 수행한 결과로서, 엔트로피 디코딩 유닛 (150)은 비트스트림으로부터 신팩스 엘리먼트들을 추출할 수도 있다.

파싱 동작을 수행하는 것의 일부로서, 엔트로피 디코딩 유닛 (150)은 비트스트림에서의 엔트로피 인코딩된 신팩스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 유닛 (156), 재구성 유닛 (158), 및 필터 유닛 (159)은 비트스트림으로부터 추출된 신팩스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성하는 재구성 동작을 수행할 수도 있다.

[0111] 위에서 논의된 바와 같이, 비트스트림은 일련의 NAL 유닛들을 포함할 수도 있다. 비트스트림의 NAL 유닛들은 비디오 파라미터 세트 NAL 유닛들, 시퀀스 파라미터 세트 NAL 유닛들, 픽처 파라미터 세트 NAL 유닛들, SEI NAL 유닛들 등을 포함할 수도 있다. 비트스트림에 대해 파싱 동작을 수행하는 것의 일부로서, 엔트로피 디코딩 유닛 (150)은 시퀀스 파라미터 세트 NAL 유닛들로부터 시퀀스 파라미터 세트들, 픽처 파라미터 세트 NAL 유닛들로부터 픽처 파라미터 세트들, SEI NAL 유닛들로부터 SEI 데이터, 등등과 같이 추출하고 이를 엔트로피 디코딩하는 파싱 동작들을 수행할 수도 있다.

[0112] 게다가, 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림에 대해 파싱 동작을 수행하는 것의 일부로서, 엔트로피 디코딩 유닛 (150)은 코딩된 슬라이스 NAL 유닛들로부터 코딩된 슬라이스들을 추출하고 이를 엔트로피 디코딩하는 파싱 동작들을 수행할 수도 있다. 코딩된 슬라이스들의 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 속하는 신팩스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 신팩스 엘리먼트들은 슬라이스를 포함하는 픽처와 연관된 픽처 파라미터 세트를 식별하는 신팩스 엘리먼트를 포함할 수도 있다. 엔트로피 디코딩 유닛 (150)은 슬라이스 헤더를 복원하기 위하여 코딩된 슬라이스 헤더에서의 신팩스 엘리먼트들에 대해, CABAC 디코딩 동작들과 같은 엔트로피 디코딩 동작들을 수행할 수도 있다.

[0113] 코딩된 슬라이스 NAL 유닛들로부터 슬라이스 데이터를 추출하는 것의 일부로서, 엔트로피 디코딩 유닛 (150)은 슬라이스 데이터에서의 코딩된 CU 들로부터 신팩스 엘리먼트들을 추출하는 파싱 동작들을 수행할 수도 있다. 추출된 신팩스 엘리먼트들은 변환 계수 블록들과 연관된 신팩스 엘리먼트들을 포함할 수도 있다. 다음으로, 엔트로피 디코딩 유닛 (150)은 신팩스 엘리먼트들의 일부에 대해 CABAC 디코딩 동작들을 수행할 수도 있다.

[0114] 엔트로피 디코딩 유닛 (150)이 비-파티셔닝된 CU에 대해 파싱 동작을 수행한 후, 비디오 디코더 (30)는 비-파티셔닝된 CU에 대해 재구성 동작을 수행할 수도 있다. 비-파티셔닝된 CU에 대해 재구성 동작을 수행하기 위하여, 비디오 디코더 (30)는 CU의 각각의 TU에 대해 재구성 동작을 수행할 수도 있다. CU의 각각의 TU에 대한 재구성 동작을 수행함으로써, 비디오 디코더 (30)는 CU와 연관된 잔차 비디오 블록을 재구성할 수도 있다.

[0115] TU에 대해 재구성 동작을 수행하는 것의 일부로서, 역양자화 유닛 (154)은 TU와 연관된 변환 계수 블록을 역양자화, 예컨대, 탈양자화 (de-quantize) 할 수도 있다. 역양자화 유닛 (154)은 HEVC에 대해 제안되거나 H.264 디코딩 표준에 의해 정의된 역양자화 프로세스들과 유사한 방식으로 변환 계수 블록을 역양자화할 수도 있다. 역양자화 유닛 (154)은 양자화도와, 마찬가지로, 적용하기 위한 역양자화 유닛 (154)에 대한 역양자화도를 결정하기 위하여, 변환 계수 블록의 CU에 대하여 비디오 인코더 (20)에 의해 계산된 양자화 파라미터 QP를 이용할 수도 있다.

[0116] 역양자화 유닛 (154)이 변환 계수 블록을 역양자화한 후, 역변환 유닛 (156)은 변환 계수 블록과 연관된 TU에 대한 잔차 비디오 블록을 생성할 수도 있다. 역변환 유닛 (156)은 TU에 대한 잔차 비디오 블록을 생성하기 위하여 역변환을 변환 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 유닛 (156)은 역 DCT, 역정수 변환, 역 카루넨-루베 변환 (Karhunen-Loeve transform; KLT), 역회전 변환, 역 방향성 변환, 또는 또 다른

역변환을 변환 계수 블록에 적용할 수도 있다. 일부의 예들에서, 역변환 유닛 (156)은 비디오 인코더 (20)로부터의 시그널링에 기초하여 변환 계수 블록에 적용하기 위한 역변환을 결정할 수도 있다. 이러한 예들에서, 역변환 유닛 (156)은 변환 계수 블록과 연관된 트리블록에 대한 쿼드트리의 루트 노드에서의 시그널링된 변환에 기초하여 역변환을 결정할 수도 있다. 다른 예들에서, 역변환 유닛 (156)은 블록 사이즈, 코딩 모드 등과 같은 하나 이상의 코딩 특성들로부터 역변환을 추론할 수도 있다. 일부의 예들에서, 역변환 유닛 (156)은 캐스케이드 역변환 (cascaded inverse transform)을 적용할 수도 있다.

[0117] 일부의 예들에서, 모션 보상 유닛 (162)은 보간 필터들에 기초하여 보간을 수행함으로써 PU의 예측된 비디오 블록을 세분화할 수도 있다. 서브-샘플 정밀도를 갖는 모션 보상을 위해 이용되어야 할 보간 필터들에 대한 식별자들은 신팩스 엘리먼트들 내에 포함될 수도 있다. 모션 보상 유닛 (162)은 참조 블록의 정수-미만 샘플들에 대한 보간된 값들을 계산하기 위하여, PU의 예측된 비디오 블록의 생성 동안에 비디오 인코더 (20)에 의해 이용된 동일한 보간 필터들을 이용할 수도 있다. 모션 보상 유닛 (162)은 수신된 신팩스 정보에 따라 비디오 인코더 (20)에 의해 이용된 보간 필터들을 결정할 수도 있고, 예측된 비디오 블록을 생성하기 위하여 보간 필터들을 이용할 수도 있다.

[0118] 도 5를 참조하여 이하에서 추가로 논의된 바와 같이, 예측 프로세싱 유닛 (152)은 도 5에서 예시된 방법들을 수행함으로써 PU(또는 임의의 다른 참조 계층 및/또는 강화 계층 블록들 또는 비디오 유닛들)를 코딩(예컨대, 인코딩 또는 디코딩) 할 수도 있다. 예를 들어, 모션 보상 유닛 (162), 인트라 예측 유닛 (164), 또는 인터-계층 예측 유닛 (166)은 도 5에서 예시된 방법들을 모두 함께 또는 별도로 수행하도록 구성될 수도 있다.

[0119] PU가 인트라 예측을 이용하여 인코딩될 경우, 인트라 예측 유닛 (164)은 PU에 대한 예측된 비디오 블록을 생성하기 위하여 인트라 예측을 수행할 수도 있다. 예를 들어, 인트라 예측 유닛 (164)은 비트스트림에서의 신팩스 엘리먼트들에 기초하여 PU에 대한 인트라 예측 모드를 결정할 수도 있다. 비트스트림은, 인트라 예측 유닛 (164)이 PU의 인트라 예측 모드를 결정하기 위하여 이용할 수도 있는 신팩스 엘리먼트들을 포함할 수도 있다.

[0120] 일부의 사례들에서, 신팩스 엘리먼트들은 인트라 예측 유닛 (164)이 현재의 PU의 인트라 예측 모드를 결정하기 위하여 또 다른 PU의 인트라 예측 모드를 이용해야 하는 것을 표시할 수도 있다. 예를 들어, 현재의 PU의 인트라 예측 모드가 이웃하는 PU의 인트라 예측 모드와 동일할 가능성이 있을 수도 있다. 다시 말해서, 이웃하는 PU의 인트라 예측 모드는 현재의 PU에 대한 가장 가능성 있는 모드일 수도 있다. 이 때문에, 이 예에서, 비트스트림은, PU의 인트라 예측 모드가 이웃하는 PU의 인트라 예측 모드와 동일한 것을 표시하는 작은 신팩스 엘리먼트를 포함할 수도 있다. 다음으로, 인트라 예측 유닛 (164)은 공간적으로 이웃하는 PU들의 비디오 블록들에 기초하여 PU에 대한 예측 데이터(예컨대, 예측된 샘플들)를 생성하기 위하여 인트라 예측 모드를 이용할 수도 있다.

[0121] 위에서 논의된 바와 같이, 비디오 디코더 (30)는 또한, 인터-계층 예측 유닛 (166)을 포함할 수도 있다. 인터-계층 예측 유닛 (166)은 SVC에서 이용가능한 하나 이상의 상이한 계층들(예컨대, 기본 또는 참조 계층)을 이용하여 현재의 블록(예컨대, EL에서의 현재의 블록)을 예측하도록 구성된다. 이러한 예측은 인터-계층 예측으로서 지칭될 수도 있다. 인터-계층 예측 유닛 (166)은 인터-계층 중복성을 감소시키기 위한 예측 방법들을 사용함으로써, 코딩 효율을 개선시키고 연산 자원 요건들을 감소시킨다. 인터-계층 예측의 일부의 예들은 인터-계층 인트라 예측, 인터-계층 모션 예측, 및 인터-계층 잔차 예측을 포함한다. 인터-계층 인트라 예측은 강화 계층에서 현재의 블록을 예측하기 위하여 기본 계층에서의 공동-위치된 블록들의 재구성을 이용한다. 인터-계층 모션 예측은 강화 계층에서 모션을 예측하기 위하여 기본 계층의 모션 정보를 이용한다. 인터-계층 잔차 예측은 강화 계층의 잔차를 예측하기 위하여 기본 계층의 잔차를 이용한다. 인터-계층 예측 방식들의 각각은 이하에서 더욱 상세하게 논의된다.

[0122] 재구성 유닛 (158)은 CU의 비디오 블록을 재구성하기 위하여, CU의 TU들과 연관된 잔차 비디오 블록들 및 CU의 PU들의 예측된 비디오 블록들, 즉, 적용 가능하다면, 인트라-예측 데이터 또는 인터-예측 데이터의 어느 하나를 이용할 수도 있다. 이에 따라, 비디오 디코더 (30)는 비트스트림에서의 신팩스 엘리먼트들에 기초하여 예측된 비디오 블록 및 잔차 비디오 블록을 생성할 수도 있고, 예측된 비디오 블록 및 잔차 비디오 블록에 기초하여 비디오 블록을 생성할 수도 있다.

[0123] 재구성 유닛 (158)이 CU의 비디오 블록을 재구성한 후, 필터 유닛 (159)은 CU와 연관된 블록킹 아티팩트들을 감소시키기 위하여 디블록킹 동작을 수행할 수도 있다. 필터 유닛 (159)이 CU와 연관된 블록킹 아티팩트들을 감소시키기 위하여 디블록킹 동작을 수행할 수도 있다.

트들을 감소시키기 위하여 디블록킹 동작을 수행한 후, 비디오 디코더 (30)는 CU의 비디오 블록을 디코딩된 픽처 버퍼 (160) 내에 저장할 수도 있다. 디코딩된 픽처 버퍼 (160)는 후속 모션 보상, 인트라 예측, 및 도 1a 또는 도 1b의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에서의 제시를 위하여 참조 픽처들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30)는, 디코딩된 픽처 버퍼 (160)에서의 비디오 블록들에 기초하여, 다른 CU들의 PU들에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다.

#### [0124] 멀티-계층 디코더

도 3b는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 (또한, 간단하게 비디오 디코더 (33)로서 지칭된) 멀티-계층 비디오 디코더 (33)의 예를 예시하는 블록도이다. 비디오 디코더 (33)는 SHVC 및 멀티뷰 코딩을 위한 것과 같은 멀티-계층 비디오 프레임들을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 디코더 (33)는 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

비디오 디코더 (33)는 비디오 디코더 (30A) 및 비디오 디코더 (30B)를 포함하고, 그 각각은 비디오 디코더 (30)로서 구성될 수도 있으며 비디오 디코더 (30)에 대하여 위에서 설명된 기능들을 수행할 수도 있다. 또한, 참조 번호들의 재이용에 의해 표시된 바와 같이, 비디오 디코더들 (30A 및 30B)은 비디오 디코더 (30)로서 시스템들 및 서브시스템들의 적어도 일부를 포함할 수도 있다. 비디오 디코더 (33)는 2개의 비디오 디코더들 (30A 및 30B)을 포함하는 것으로서 예시되어 있지만, 비디오 디코더 (33)는 이와 같이 제한되지 않으며 임의의 수의 비디오 디코더 (30) 계층들을 포함할 수도 있다. 일부의 실시형태들에서, 비디오 디코더 (33)는 액세스 유닛에서의 각각의 픽처 또는 프레임에 대한 비디오 디코더 (30)를 포함할 수도 있다. 예를 들어, 5개의 픽처들을 포함하는 액세스 유닛은 5개의 디코더 계층들을 포함하는 비디오 디코더에 의해 프로세싱되거나 디코딩될 수도 있다. 일부의 실시형태들에서, 비디오 디코더 (33)는 액세스 유닛에서의 프레임들보다 더 많은 디코더 계층들을 포함할 수도 있다. 일부의 이러한 경우들에 있어서, 비디오 인코더 계층들의 일부는 일부의 액세스 유닛들을 프로세싱할 때에 비활성일 수도 있다.

비디오 디코더들 (30A 및 30B)에 추가하여, 비디오 디코더 (33)는 업샘플링 유닛 (92)을 포함할 수도 있다. 일부의 실시형태들에서, 업샘플링 유닛 (92)은 프레임 또는 액세스 유닛에 대한 참조 픽처 리스트에 추가되어야 할 강화된 계층을 생성하기 위하여, 수신된 비디오 프레임의 기본 계층을 업샘플링할 수도 있다. 이 강화된 계층은 디코딩된 픽처 버퍼 (160) 내에 저장될 수 있다. 일부의 실시형태들에서, 업샘플링 유닛 (92)은 도 2a의 리샘플링 유닛 (90)에 대하여 설명된 실시형태들의 일부 또는 전부를 포함할 수 있다. 일부의 실시형태들에서, 업샘플링 유닛 (92)은 계층을 업샘플링하며, 슬라이스 경계 규칙들 및/또는 래스터 스캔 규칙들의 세트를 준수하기 위하여 하나 이상의 슬라이스들을 재편성하거나, 재정의하거나, 수정하거나, 조절하도록 구성된다. 일부의 경우들에는, 업샘플링 유닛 (92)은 수신된 비디오 프레임의 계층을 업샘플링 및/또는 다운샘플링하도록 구성된 리샘플링 유닛일 수도 있다.

업샘플링 유닛 (92)은 더 낮은 계층의 디코더 (예컨대, 비디오 디코더 (30A))의 디코딩된 픽처 버퍼 (160)로부터 픽처 또는 프레임 (또는 픽처와 연관된 픽처 정보)을 수신하고 픽처 (또는 수신된 픽처 정보)를 업샘플링하도록 구성될 수도 있다. 다음으로, 이 업샘플링된 픽처는 더 낮은 계층의 디코더와 동일한 액세스 유닛에서의 픽처를 디코딩하도록 구성된 더 높은 계층의 디코더 (예컨대, 비디오 디코더 (30B))의 예측 프로세싱 유닛 (152)에 제공될 수도 있다. 일부의 경우들에는, 더 높은 계층의 디코더가 더 낮은 계층의 디코더로부터 제거된 하나의 계층이다. 다른 경우들에는, 도 3b의 계층 0 디코더 및 계층 1 디코더 사이에 하나 이상의 더 높은 계층의 디코더들이 있을 수도 있다.

일부의 경우들에는, 업샘플링 유닛 (92)이 생략되거나 우회될 수도 있다. 이러한 경우들에는, 비디오 디코더 (30A)의 디코딩된 픽처 버퍼 (160)로부터의 픽처가 직접적으로, 또는 적어도 업샘플링 유닛 (92)에 제공되지 않으면서, 비디오 디코더 (30B)의 예측 프로세싱 유닛 (152)에 제공될 수도 있다. 예를 들어, 비디오 디코더 (30B)에 제공된 비디오 데이터와, 비디오 디코더 (30A)의 디코딩된 픽처 버퍼 (160)로부터의 참조 픽처가 동일한 사이즈 또는 해상도일 경우, 참조 픽처는 업샘플링 없이 비디오 디코더 (30B)에 제공될 수도 있다. 또한, 일부의 실시형태들에서, 업샘플링 유닛 (92)은 비디오 디코더 (30A)의 디코딩된 픽처 버퍼 (160)로부터 수신된 참조 픽처를 업샘플링하거나 다운샘플링하도록 구성된 리샘플링 유닛 (90)일 수도 있다.

도 3b에서 예시된 바와 같이, 비디오 디코더 (33)는 디멀티플렉서 (99) 또는 디믹스 (demux)를 더 포함할 수도 있다. 디믹스 (99)는 인코딩된 비디오 비트스트림을 다수의 비트스트림들로 분할할 수 있고, 디믹스 (99)에 의해 출력된 각각의 비트스트림은 상이한 비디오 디코더 (30A 및 30B)에 제공될 수 있다. 다수의 비트스트림들은 비트스트림을 수신함으로써 생성될 수도 있고, 비디오 디코더들 (30A 및 30B)의 각각은 소정의

시간에 비트스트림의 부분을 수신한다. 일부의 경우에는, 디엑스 (99) 에서 수신된 비트스트림으로부터의 비트들이 비디오 디코더들 (예컨대, 도 3b 의 예에서의 비디오 디코더들 (30A 및 30B)) 의 각각 사이에서 한 번에 하나의 비트씩 교대될 수도 있지만, 많은 경우에는, 비트스트림이 상이하게 분할된다. 예를 들어, 비트스트림은 어느 비디오 디코더가 비트스트림을 한 번에 하나의 블록씩 수신하는지를 교대시킴으로써 분할될 수도 있다. 또 다른 예에서, 비트스트림은 1:1 이 아닌 비율의 블록들에 의해 비디오 디코더들 (30A 및 30B) 의 각각으로 분할될 수도 있다. 예를 들어, 2 개의 블록들은 비디오 디코더 (30A) 에 제공된 각각의 블록에 대하여 비디오 디코더 (30B) 에 제공될 수도 있다. 일부의 실시형태들에서, 디엑스 (99) 에 의한 비트스트림의 분할은 미리 프로그래밍될 수도 있다. 다른 실시형태들에서, 디엑스 (99) 는 목적지 모듈 (14) 을 포함하는 목적지 디바이스 상의 프로세서로부터와 같이, 비디오 디코더 (33) 의 외부의 시스템으로부터 수신된 제어 신호에 기초하여 비트스트림을 분할할 수도 있다. 제어 신호는 입력 인터페이스 (28) 로부터의 비디오의 해상도 또는 비트레이트에 기초하여, 링크 (16) 의 대역폭에 기초하여, 사용자와 연관된 가입 (예컨대, 유료 가입 대 무료 가입) 에 기초하여, 또는 비디오 디코더 (33) 에 의해 획득가능한 해상도를 결정하기 위한 임의의 다른 인자에 기초하여 생성될 수도 있다.

**[0131] 인트라 랜덤 액세스 포인트 (Intra Random Access Point; IRAP) 핵처들:**

일부의 비디오 코딩 방식들은 비트스트림의 전반에 걸쳐 다양한 랜덤 액세스 포인트들을 제공할 수도 있어서, 비트스트림에서 그 랜덤 액세스 포인트들을 선행하는 임의의 핵처들을 디코딩할 필요 없이, 비트스트림은 그 랜덤 액세스 포인트들 중의 임의의 것으로부터 시작하여 디코딩될 수도 있다. 이러한 비디오 코딩 방식들에서는, (예컨대, 랜덤 액세스 포인트를 제공하는 핵처와 동일한 액세스 유닛 내에 있는 그러한 핵처들을 포함하는) 출력 순서에서 랜덤 액세스 포인트를 후행하는 모든 핵처들은 랜덤 액세스 포인트를 선행하는 임의의 핵처들을 이용하지 않으면서 정확하게 디코딩될 수 있다. 예를 들어, 비트스트림의 부분이 송신 동안, 또는 디코딩 동안에 손실되더라도, 디코더는 다음의 랜덤 액세스 포인트로부터 시작하여 비트스트림을 디코딩하는 것을 재개 할 수 있다. 랜덤 액세스에 대한 지원은 예를 들어, 동적 스트리밍 서비스들, 탐색 동작들, 채널 스위칭 등을 용이하게 할 수도 있다.

일부의 코딩 방식들에서, 이러한 랜덤 액세스 포인트들은 인트라 랜덤 액세스 포인트 (IRAP) 핵처들로서 지칭되는 핵처들에 의해 제공될 수도 있다. 예를 들어, 액세스 유닛 "auA" 내에 포함된 강화 계층 ("layerA") 에서의 (예컨대, 강화 계층 IRAP 핵처에 의해 제공된) 랜덤 액세스 포인트는 계층-특정 랜덤 액세스를 제공할 수도 있어서, layerB 내에 있으며 디코딩 순서에서 auA 를 선행하는 액세스 유닛 ("auB") 내에 포함된 랜덤 액세스 포인트 (또는 auA 내에 포함된 랜덤 액세스 포인트) 를 가지는 layerA 의 각각의 참조 계층 ("layerB") (예컨대, layerA 를 예측하기 위하여 이용되는 계층인 참조 계층) 에 대하여, (auB 에서 위치된 그러한 핵처들을 포함하는) 출력 순서에서 auB 를 후행하는 layerA 에서의 핵처들은 auB 를 선행하는 layerA 에서의 임의의 핵처들을 디코딩할 필요 없이 정확하게 디코딩 가능하다.

IRAP 핵처들은 인트라 예측을 이용하여 코딩될 수도 있고 (예컨대, 다른 핵처들을 참조하지 않으면서 코딩됨), 예를 들어, 순간적 디코딩 리프레시 (IDR) 핵처들, 클린 랜덤 액세스 (CRA) 핵처들, 및 파손 링크 액세스 (broken link access; BLA) 핵처들을 포함할 수도 있다. 비트스트림에서 IDR 핵처가 있을 때, 디코딩 순서에서 IDR 핵처를 선행하는 모든 핵처들은 디코딩 순서에서 IDR 핵처를 후행하는 핵처들에 의한 예측을 위하여 이용되지 않는다. 비트스트림에서 CRA 핵처가 있을 때, CRA 핵처를 후행하는 핵처들은 예측을 위하여 디코딩 순서에서 CRA 핵처를 선행하는 핵처들을 이용할 수도 있거나 이용하지 않을 수도 있다. 디코딩 순서에서 CRA 핵처를 후행하지만, 디코딩 순서에서 CRA 핵처를 선행하는 핵처들을 이용하는 그러한 핵처들은 랜덤 액세스 스킵된 선두 (random access skipped leading; RASL) 핵처들로서 지칭될 수도 있다. 디코딩 순서에서 IRAP 핵처를 후행하며 출력 순서에서 IRAP 핵처를 선행하는 또 다른 타입의 핵처는, 디코딩 순서에서 IRAP 핵처를 선행하는 임의의 핵처들에 대한 참조들을 포함하지 않을 수도 있는 랜덤 액세스 디코딩 가능 선두 (random access decodable leading; RADL) 핵처이다. RASL 핵처들은 CRA 핵처를 선행하는 핵처들이 이용 가능하지 않을 경우에 디코더에 의해 폐기될 수도 있다. BLA 핵처는, (예컨대, 2 개의 비트스트림들이 함께 스플라이싱 (splice) 되고, BLA 핵처는 디코딩 순서에서 제 2 비트스트림의 최초 핵처이기 때문에) BLA 핵처를 선행하는 핵처들이 디코더에 의해 이용 가능하지 않을 수도 있다는 것을 디코더에 표시한다. IRAP 핵처인 기본 계층 핵처 (예컨대, 0 의 계층 ID 값을 가지는 핵처) 를 포함하는 액세스 유닛 (예컨대, 다수의 계층들에 가로지른 동일한 출력 시간과 연관된 모든 코딩된 핵처들로 구성되는 핵처들의 그룹) 은 IRAP 액세스 유닛으로서 지칭될 수도 있다.

### [0135] IRAP 픽처들의 교차-계층 정렬

SVC에서, IRAP 픽처들은 상이한 계층들을 가로질러 정렬되도록 (예컨대, 동일한 액세스 유닛에서 포함됨) 요구되지 않을 수도 있다. 예를 들어, IRAP 픽처들이 정렬되도록 요구되었을 경우, 적어도 하나의 IRAP 픽처를 포함하는 임의의 액세스 유닛은 IRAP 픽처들을 포함하기만 할 것이다. 다른 한편으로, IRAP 픽처들이 정렬되도록 요구되지 않았을 경우, 단일 액세스 유닛에서는, (예컨대, 제 1 계층에서의) 하나의 픽처는 IRAP 픽처일 수도 있고, (예컨대, 제 2 계층에서의) 또 다른 픽처는 비-IRAP 픽처일 수도 있다. 비트스트림에서 이러한 비-정렬된 (non-aligned) IRAP 픽처들을 가지는 것은 일부의 장점들을 제공할 수도 있다. 예를 들어, 2-계층 비트스트림에서, 강화 계층에서보다 기본 계층에서 더 많은 IRAP 픽처들이 있을 경우, 브로드캐스트 및 멀티캐스트 애플리케이션들에서는, 낮은 투-인 (tune-in) 지연 및 높은 코딩 효율이 달성될 수 있다.

[0137] 일부의 비디오 코딩 방식들에서, 픽처 순서 카운트 (; POC)는 디코딩된 픽처들이 디스플레이되는 상대적인 순서를 추적하기 위하여 이용될 수도 있다. 이러한 코딩 방식들 중의 일부는 POC 값들이 어떤 타입들의 픽처들이 비트스트림에서 나타날 때마다 재설정 (예컨대, 제로로 설정되거나 비트스트림에서 시그널링된 일부의 값으로 설정됨) 되게 할 수도 있다. 예를 들어, 어떤 IRAP 픽처들의 POC 값들이 재설정될 수도 있어서, 디코딩 순서에서 그러한 IRAP 픽처들을 선행하는 다른 픽처들의 POC 값들이 또한 재설정되게 할 수도 있다. 이것은 IRAP 픽처들이 상이한 계층들을 가로질러 정렬되도록 요구되지 않을 때에 문제가 될 수도 있다. 예를 들어, 하나의 픽처 ("picA")가 IRAP 픽처이고 동일한 액세스 유닛에서의 또 다른 픽처 ("picB")가 IRAP 픽처가 아닐 때, picA를 포함하는 계층에서의 picA가 IRAP 픽처인 것으로 인해 재설정되는 픽처 ("picC")의 POC 값은 picB를 포함하는 계층에서의, 재설정되지 않는 픽처 ("picD")의 POC 값과는 상이할 수도 있고, 여기서, picC 및 picD는 동일한 액세스 유닛 내에 있다. 이것은 picC 및 picD가 동일한 액세스 유닛 (예컨대, 동일한 출력 시간)에 속하더라도 picC 및 picD가 상이한 POC 값들을 가지게 한다. 이에 따라, 이 예에서는, picC 및 picD의 POC 값을 유도하기 위한 유도 프로세스가 POC 값을 및 액세스 유닛들의 정의와 부합하는 POC 값을 생성하도록 수정될 수 있다.

### 픽처 순서 카운트 (POC)

[0139] 위에서 논의된 바와 같이, 특정한 코딩된 픽처에 대한 픽처 순서 카운트 (POC)의 값 (예컨대, HEVC에서의 PicOrderCntVal)은 동일한 코딩된 비디오 시퀀스에서의 다른 픽처들에 대하여 픽처 출력 프로세스에서의 특정한 코딩된 픽처의 상대적인 순서를 나타낸다. 일부의 실시형태들에서, POC는 최하위 비트 (LSB)들 및 최상위 비트 (MSB)들을 포함하고, POC는 MSB 및 LSB를 연결함으로써 획득될 수도 있다. 다른 실시형태들에서, POC는 MSB 값 및 LSB 값을 가산 (add) 함으로써 획득될 수도 있다. LSB는 슬라이스 헤더 (slice header)에서 시그널링될 수도 있고, MSB는 현재의 픽처의 NAL 유닛 타입과, (1) RASL 또는 RADL 픽처들이 아니고, (2) 폐기 가능 (예컨대, 다른 픽처가 이들에 종속되지 않음으로써, 이들이 대역폭 제약들을 충족시키기 위하여 누락되도록 한다는 것을 표시하는, "폐기 가능"으로서 표기된 픽처들) 하지 않고, (3) 서브-계층 비-참조 픽처들 (예컨대, 동일한 시간적 서브-계층 또는 동일한 계층에서의 다른 픽처들에 의해 참조를 위하여 이용되지 않는 픽처들)이 아니고, (4) 0과 동일한 시간적 ID (예컨대, 시간적 서브-계층 ID)를 가지는, 디코딩 순서에서의 하나 이상의 이전의 픽처들의 MSB 및 LSB에 기초하여 인코더 또는 디코더에 의해 연산될 수도 있다. (1) 내지 (4)에서 설명된 이러한 픽처들은 POC-앵커 (POC-anchor) 픽처들로서 본원에서 지칭될 수도 있다. 유사하게, 0보다 더 큰 시간적 ID 값을 가지는 픽처들, RASL 또는 RADL 픽처들, 폐기 가능한 픽처들, 또는 서브-계층 비-참조 픽처들은 비-POC-앵커 (non-POC-anchor) 픽처들로서 지칭될 수도 있다. POC-앵커 픽처들은, 인코더 및/또는 디코더가 (예컨대, 대역폭 제약을 충족시키기 위하여) 비트스트림으로부터 제거하도록 선택하지 않을 수도 있는 픽처들을 더 포함할 수도 있다. POC-앵커 픽처들은, 인코더 및/또는 디코더가 (예컨대, 대역폭 제약을 충족시키기 위하여) 비트스트림으로부터 제거하도록 구성될 수도 있는 픽처들의 타입들 이외의 임의의 픽처를 더 포함할 수도 있다. 비-POC-앵커 픽처들은 POC-앵커 픽처가 아닌 임의의 픽처를 포함할 수도 있다.

[0140] 현재의 픽처가 (1) 1과 동일한 NoRaslOutputFlag (예컨대, 1로 설정될 경우에 RASL 픽처들이 출력되지 않아야 하는 것을 표시하고, 0으로 설정될 경우에 RASL 픽처들이 출력되어야 하는 것을 표시하는 플래그)를 갖는 IRAP 픽처, 또는 (2) 비트스트림의 최초 픽처인 CRA 픽처일 때, POC MSB의 값은 0과 동일하도록 추론된다. 위에서 설명된 바와 같이, 멀티-계층 비트스트림 (예컨대, 하나를 초과하는 계층을 갖는 SHVC 또는 MV-HEVC 비트스트림)에서는, 하나 이상의 픽처들이 IRAP 픽처들이고 하나 이상의 다른 픽처들이 비-IRAP 픽처들인 액세스 유닛 (AU)들이 존재할 수도 있고, 이러한 AU들은 "비-정렬된 IRAP AU 들"로서 지칭될 수도 있다. 비-정렬된 IRAP AU 들을 포함하는 비트스트림들을 디코딩할 때, 비트스트림에서 시그널링된 POC LSB 값들에 기초하

여 유도된 POC 들은 액세스 유닛에서의 모든 픽처들이 동일한 POC 값을 가져야 한다는 비트스트림 준수성 요건을 위반할 것이라는 것이라는 가능하다.

#### [0141] 계층 초기화 픽처 (Layer Initialization Picture; LIP)

일부의 코딩 방식들에서, 계층 초기화 픽처 ("LIP picture")는 1로 설정된 NoRas1OutputFlag 플래그 (예컨대, 1로 설정될 경우에 RASL 픽처들이 출력되지 않아야 하는 것을 표시하고, 0으로 설정될 경우에 RASL 픽처들이 출력되어야 하는 것을 표시하는 플래그)를 가지는 IRAP 픽처인 픽처, 또는 기본 계층 픽처 (예컨대, 0의 계층 ID 또는 비트스트림에서 정의된 가장 작은 계층 ID를 가지는 픽처)가 1로 설정된 NoRas1OutputFlag를 가지는 IRAP 액세스 유닛인 초기 IRAP 액세스 유닛에서 포함되는 픽처로서 정의될 수도 있다.

일부의 실시형태들에서, SPS는 각각의 LIP에서 활성화될 수 있다. 예를 들어, 1로 설정된 NoRas1OutputFlag 플래그를 가지는 각각의 IRAP 픽처, 또는 초기 IRAP 액세스 유닛 내에 포함되는 각각의 픽처, 이전에 활성화되었던 SPS와는 상이할 수도 있는 (예컨대, 상이한 픽처 해상도들 등을 특정하는) 새로운 SPS.

그러나, LIP 픽처가 IRAP 픽처 (예컨대, 초기 IRAP 액세스 유닛 내에 포함된 임의의 픽처)가 아니고 초기 IRAP 액세스 유닛에서의 기본 계층 픽처가 0으로 설정된 플래그 NoClrasOutputFlag 플래그 (예컨대, 1로 설정될 경우에 교차-계층 랜덤 액세스 스킁 픽처들이 출력되지 않아야 하는 것을 표시하고, 0으로 설정될 경우에 교차-계층 랜덤 액세스 스킁 픽처들이 출력되어야 하는 것을 표시하는 플래그)를 갖는 IDR 픽처일 경우, LIP 픽처는 새로운 SPS를 활성화하도록 허용되지 않아야 한다. 새로운 SPS가 이러한 경우에 이러한 LIP 픽처에서 활성화될 경우, 특히, 새로운 SPS의 SPS RBSP의 내용들이 초기 IRAP 액세스 유닛 전에 미리 활성이었던 SPS의 내용과는 상이할 때, 상이한 픽처 해상도들 및 오류 내성 (error resilience)에 있어서 문제들이 있을 수 있었다. 예를 들어, 새로운 SPS는 해상도를 업데이트할 수도 있고, 상이한 사이즈들의 픽처들을 참조하기 위하여 시간적 예측을 이용할 수도 있다.

#### [0144] 픽처들의 범핑 및 플러싱

(예컨대, 다른 픽처들을 예측하기 위하여 디스플레이되거나 출력될 수 있도록) 디코딩되는 픽처들은 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 내에 저장된다. 출력되어야 하는 픽처들은 "출력을 위하여 필요함"으로서 표기될 수도 있고, 다른 픽처들을 예측하기 위하여 이용되어야 하는 픽처들은 "참조를 위하여 이용됨"으로서 표기될 수도 있다. "출력을 위하여 필요함"으로서 표기되지도 않고 "참조를 위하여 이용됨"으로서 표기되지도 않는 디코딩된 픽처들 (예컨대, "참조를 위하여 이용됨" 또는 "출력을 위하여 필요함"으로서 초기에 표기되었지만, "참조를 위하여 이용되지 않음" 또는 "출력을 위하여 필요하지 않음"으로서 추후에 표기되었던 픽처들)은, 이들이 디코딩 프로세스에 의해 제거될 때까지 DPB에서 존재할 수도 있다. 출력 순서 준수 디코더들에서는, DPB로부터 픽처들을 제거하는 프로세스가 종종 "출력을 위하여 필요함"으로서 표기되는 픽처들을 출력을 바로 후행한다. 출력 및 추후의 제거의 이 프로세스는 "범핑 (bumping)"으로서 지칭될 수도 있다.

또한, 이 픽처들이 "출력을 위하여 필요함"으로서 표기될 수도 있더라도, 디코더가 출력 없이 DPB에서 픽처들을 제거할 수도 있는 상황들이 있다. 본원에서의 설명의 용이함을 위하여, (디코딩된 픽처들이 "출력을 위하여 필요함" 또는 "참조를 위하여 이용됨"으로서 표기되는지 여부에 관계 없이) IRAP 픽처를 디코딩할 시에 DPB에서 존재하는 디코딩된 픽처들은 IRAP 픽처와 연관된 "지연 DPB 픽처들" 또는 IRAP 픽처의 "연관된 지연 DPB 픽처들"로서 지칭된다. HEVC 맥락에서, 이러한 상황들의 일부의 예들이 이하에서 설명된다.

하나의 예에서, "1"의 값과 동일한 NoRas1OutputFlag를 갖는 CRA 픽처가 비트스트림의 중간 (예컨대, 비트스트림에서 최초 픽처가 아님)에 존재할 때, CRA 픽처와 연관된 지연 DPB 픽처들은 출력되지 않을 것이고, DPB로부터 제거될 것이다. 이러한 상황들은 스플라이스 포인트들에서 발생할 가능성이 있고, 여기서, 2개의 비트스트림들은 함께 합쳐지고, 후자의 비트스트림의 최초 픽처는 "1"의 값과 동일한 NoRas1OutputFlag를 갖는 CRA 픽처이다. 또 다른 예에서, "1"의 값과 동일한 NoRas1OutputFlag를 가지며 CRA 픽처가 아닌 IRAP 픽처 picA가 비트스트림의 중간에 존재하고, 픽처의 해상도가 (예컨대, 새로운 SPS의 활성화로) picA에서 변경될 때, 연관된 지연 DPB 픽처들이 DPB를 점유하는 것을 계속할 경우, picA로부터 시작하는 픽처들의 디코딩은 예를 들어, 버퍼 오버플로우 (buffer overflow)로 인해 문제가 될 수도 있으므로, picA의 연관된 지연 DPB 픽처들은 이들이 출력될 수 있기 전에 DPB로부터 제거될 수도 있다. 이 경우, DPB로부터의 출력 없이 지연 픽처들을 플러싱 (flush) 하기 위하여, picA와 연관된 no\_output\_of\_prior\_pics\_flag (예컨대, 1로 설정될 경우, 이전에 디코딩되었고 DPB에서 저장되었던 픽처들이 출력되지 않으면서 DPB로부터 제거되어야 한다는

것을 표시하고, 0 으로 설정될 경우, 이전에 디코딩되었고 DPB 에서 저장되었던 픽처들이 출력되지 않으면서 DPB 로부터 제거되지 않아야 한다는 것을 표시하는 플래그) 의 값은 인코더 또는 스플라이서 (splicer) 에 의해 "1" 의 값과 동일하게 설정되어야 하거나, NoOutputOfPriorPicsFlag (예컨대, 비트스트림에서 포함된 정보에 기초하여 결정될 수도 있는 유도된 값) 는 디코더에 의해 "1" 의 값과 동일하도록 유도될 수도 있다. 스플라이싱 동작은 도 4 에 대하여 이하에서 추가로 설명된다.

[0148] 출력 없이 DPB 로부터 연관된 지연 DPB 픽처들을 제거하는 이 프로세스는 "플러싱 (flushing)" 으로서 지칭될 수도 있다. 위에서 설명되지 않은 상황들에서도, IRAP 픽처는 "1" 의 값과 동일한 no\_output\_of\_prior\_pics\_flag 의 값을 특정할 수도 있어서, 디코더는 IRAP 픽처의 연관된 DPB 지연 픽처들을 플러싱할 것이다.

#### 스플라이스 포인트를 포함하는 비트스트림

[0149] **도 4** 를 참조하여, 스플라이스 포인트를 가지는 일 예의 비트스트림이 설명될 것이다. **도 4** 는 스플라이싱 비트스트림들 (410 및 420) 에 의해 생성된 멀티-계층 비트스트림 (400) 을 도시한다. 비트스트림 (410) 은 강화 계층 (EL) (410A) 및 기본 계층 (BL) (410B) 을 포함하고, 비트스트림 (420) 은 EL (420A) 및 BL (420B) 을 포함한다. EL (410A) 은 EL 픽처 (412A) 를 포함하고, BL (410B) 은 BL 픽처 (412B) 를 포함한다. EL (420A) 은 EL 픽처들 (422A, 424A, 및 426A) 을 포함하고, BL (420B) 은 BL 픽처들 (422B, 424B, 및 426B) 을 포함한다. 멀티-계층 비트스트림 (400) 은 액세스 유닛 (AU) 들 (430 내지 460) 을 더 포함한다. AU (430) 는 EL 픽처 (412A) 및 BL 픽처 (412B) 를 포함하고, AU (440) 는 EL 픽처 (422A) 및 BL 픽처 (422B) 를 포함하고, AU (450) 는 EL 픽처 (424A) 및 BL 픽처 (424B) 를 포함하고, AU (460) 는 EL 픽처 (426A) 및 BL 픽처 (426B) 를 포함한다. **도 4** 의 예에서, BL 픽처 (422B) 는 IRAP 픽처이고, AU (440) 에서의 대응하는 EL 픽처 (422A) 는 트레일링 픽처 (예컨대, 비-IRAP 픽처) 이고, 결과적으로, AU (440) 는 비-정렬된 IRAP AU 이다. 또한, AU (440) 는 스플라이스 포인트 (470) 를 바로 후행하는 액세스 유닛인 것에 주목해야 한다.

[0150] **도 4** 의 예는 2 개의 상이한 비트들이 함께 합쳐지는 경우를 예시하지만, 일부의 실시형태들에서, 스플라이스 포인트는 비트스트림의 부분이 제거될 때에 존재할 수도 있다. 예를 들어, 비트스트림은 부분들 A, B, 및 C 를 가질 수도 있고, 부분 B 는 부분들 A 및 C 사이에 있다. 부분 B 가 비트스트림으로부터 제거될 경우, 나머지 부분들 A 및 C 는 함께 합쳐질 수도 있고, 이들이 함께 합쳐지는 포인트는 스플라이스 포인트로서 지칭될 수도 있다. 더욱 일반적으로, 본 출원에서 논의된 바와 같은 스플라이스 포인트는 하나 이상의 시그널링되거나 유도된 파라미터들 또는 플래그들이 미리 결정된 값들을 가질 때에 존재하는 것으로 간주될 수도 있다.

예를 들어, 스플라이스 포인트가 특정한 로케이션에서 존재한다는 특정 표시를 수신하지 않고도, 디코더는 플래그 (예컨대, NoClrasOutputFlag) 의 값을 결정할 수도 있고, 플래그의 값에 기초하여 이 출원에서 설명된 하나 이상의 기법들을 수행할 수도 있다.

#### 상이한 표준들을 이용하여 코딩된 참조 계층 및 강화 계층에 대한 지원

[0151] 멀티-계층 코딩의 일부의 경우에는, 참조 계층 (RL) 이 하나의 표준을 이용하여 코딩될 수도 있고, 강화 계층 (EL) 은 또 다른 표준을 이용하여 코딩될 수도 있다. 예를 들어, RL 은 H.264/AVC 에 따라 코딩될 수도 있고, EL 은 H.265/HEVC 에 따라 코딩될 수도 있다. 표준의 더 이전의 버전을 이용하여 코딩되는 RL 에 대한 지원을 제공하는 것은, 더 이전의 버전을 이용하여 코딩된 비디오 데이터가 현재의 표준을 이용한 멀티-계층 비디오 코딩에서 사용될 수도 있으므로 유리할 수 있다. 그러나, 상이한 표준에서 코딩된 RL 을 프로세싱하는 많은 양태들을 관리하는 것은 멀티-계층 코딩을 위한 지원이 복잡하게 되게 할 수 있다. 예를 들어, 인코더 또는 디코더는 RL 픽처들의 출력을 처리하고, RL 픽처들에 대해 저장된 디코딩된 픽처를 유지하는 등을 행할 필요가 있을 수도 있다. 논의를 용이하게 하기 위하여, 상이한 표준들을 이용하여 RL 및 EL 을 코딩하는 것은 상이한 표준들을 이용하는 멀티-계층 비디오 코딩으로서 지칭될 수도 있다. 예시적인 목적들을 위하여, 설명은 상이한 표준들을 이용하는 것을 지칭하지만, 본원에서 설명된 기법들은 또한, 동일한 표준의 상이한 버전들을 이용하는 것에 적용할 수 있다. 기법들은 또한, 상이한 코딩 방식들을 이용하는 것, 또는 코딩 방식의 상이한 버전들을 이용하는 것에 적용할 수도 있다.

[0152] 이러한, 그리고 다른 과제들을 해결하기 위하여, 어떤 양태들에 따른 기법들은 간략화된 방식으로 상이한 표준들을 이용하는 멀티-계층 비디오 코딩을 지원할 수 있다. 예를 들어, HEVC 디코더는 HEVC 이외의 표준을 이용하여 코딩된 RL 픽처들의 관리 및 프로세싱을 최소화할 수도 있다. 예시적인 목적들을 위하여, RL 은 H.264/AVC 를 이용하여 코딩되는 것으로서 설명될 것이고, EL 은 H.265/HEVC 를 이용하여 코딩되는 것으로서 설명될 것이다. 그러나, 상이한 표준들의 임의의 조합은 RL 및 EL 을 코딩하기 위하여 이용될 수 있다.

어떤 양태들에 따르면, 기법들은 다음과 같이 RL 팩처들의 관리 및 프로세싱을 최소화한다: (1) 디코딩된 RL 팩처들은 외부 수단에 의해 제공되고 (예컨대, 캡슐화 (encapsulation) 가 제공되지 않음), 그리고 (2) EL 팩처들의 출력과의 동기화를 포함하여, RL 팩처들의 출력은 외부 수단에 의해 제어된다. 캡슐화는, RL 팩처들이 특정 NAL 유닛 헤더 선택스를 이용하여 비트스트림에서 제공되도록, RL 팩처들에 대한 어떤 NAL 유닛 헤더 선택스를 정의하는 것을 지정할 수도 있다. 외부 수단은 RL 을 코딩하기 위하여 이용된 표준을 지원하는 코더 (예컨대, 인코더 또는 디코더) 를 지정할 수도 있다. HEVC 디코더는 상이한 표준들을 이용하는 멀티-계층 비디오 코딩을 지원하기 위하여, 이하에서 상세하게 설명된 어떤 규칙들을 구현할 수 있다. 위에서 설명된 바와 같이, 기법들은 또한, 상이한 표준들을 간단하게 이용하는 것이 아니라, 표준의 상이한 버전들을 이용하는 것에 적용할 수 있고, 또한, 상이한 코딩 방식들을 이용하는 것, 또는 코딩 방식의 상이한 버전들을 이용하는 것에 적용할 수 있다.

[0155] 이 개시물의 전반에 걸쳐 이용된 다양한 용어들은 그 정상적인 의미를 가지는 넓은 용어들이다. 게다가, 일부의 실시형태들에서, 어떤 용어들은 다음의 비디오 개념들과 관련된다. 표준은 H.264/AVC, H.265/HEVC 등과 같은 비디오 코딩 표준을 지정할 수 있다. 일부의 실시형태들에서, 표준은 H.265/HEVC, SHVC, MV-HEVC 등과 같이, 동일한 표준의 상이한 확장들을 지정할 수 있다. 예를 들어, 참조 계층은 HEVC 를 이용하여 인코딩되고, 강화 계층은 SHVC 또는 MV-HEVC 를 이용하여 인코딩된다. 위에서 설명된 바와 같이, 이 맥락에서의 외부 수단은, EL 디코더의 일부가 아니지만, 예컨대, 애플리케이션 프로그래밍 인터페이스 (application programming interface; API) 를 통해 EL 디코더와 상호작용하는 임의의 장치 또는 엔티티를 지정할 수도 있다. 하나의 예에서, 외부 수단은 RL 을 코딩하기 위하여 이용된 표준을 지원하는 코더 (예컨대, 인코더 또는 디코더) 를 지정할 수도 있다. 또 다른 예에서, 외부 수단은 EL 디코더를 포함하는 수신기의 일부를 지정할 수도 있다. 예를 들어, 동일한 수신기는 EL 디코더 및 외부 수단의 양자를 포함할 수도 있다. 이러한 경우, 외부 수단은 EL 디코더의 일부가 아니라는 의미에서, EL 디코더의 여전히 외부에 있다. 어떤 실시형태들에서, 외부 수단은 또한, 외부 장치로서 지정될 수도 있다.

#### 일 예의 실시형태

[0157] 하나의 실시형태에서, 다음의 규칙들은 상이한 표준들을 이용하는 멀티-계층 코딩을 지원하기 위하여 구현된다:

[0158] 1) 현재의 액세스 유닛에 대하여, 정보가 제공되지 않거나 (현재의 액세스 유닛에 대해 기본 계층 팩처가 존재하지 않는다는 것을 의미함), 기본 계층 팩처의 다음의 정보가 외부 수단에 의해 제공된다:

[0159] - 기본 계층 디코딩된 팩처의 디코딩된 샘플 값들

[0160] - 루마 샘플들에서의 폭 및 높이, 컬러 포맷, 루마 비트 심도, 및 크로마 비트 심도를 포함하는, 기본 계층 디코딩된 팩처의 표현 포맷.

[0161] - 기본 계층 팩처가 IDR 팩처인지 아닌지 여부.

[0162] - 선택적으로, 팩처가 프레임 또는 필드이든지 간에, 그리고 필드일 때, (필드가 상부 필드 또는 하부 필드인지 여부를 표시하는) 필드 패리티. 제공되지 않을 경우, 디코딩된 팩처는 프레임 팩처인 것으로 추론된다.

[0163] 2) 기본 계층 팩처들의 출력은 기본 계층 디코더의 책임이다. 선택적으로, 오직 하나의 비-기본 계층이 존재하고, 그것은 유일한 타겟 출력 계층이다.

[0164] 3) (멀티-표준) SHVC 디코더는 기본 계층 디코딩된 팩처에 대하여 메모리의 하나의 디코딩된 팩처 저장소를 유지할 필요만 있을 것이고, 이 메모리는 DPB 의 일부로서 고려되지 않는다.

[0165] 4) 액세스 유닛에 대한 기본 계층 디코딩된 팩처의 연관성은 외부 수단 (예컨대, 기본 계층 디코더 또는 다른 외부 수단) 의 책임이다.

[0166] 5) 기본 계층 디코딩된 팩처의 nuh\_layer\_id 는 0 과 동일하게 설정된다. 대안적으로, 기본 계층 디코딩된 팩처의 nuh\_layer\_id 는 액세스 유닛에서 모든 강화 계층 팩처들 중에서, nuh\_layer\_id 의 최저 값을 갖는 강화 계층 팩처의 인터-계층 참조 팩처 세트에서의 엔트리의 nuh\_layer\_id 와 동일하게 설정된다.

[0167] 6) 기본 계층 디코딩된 팩처의 팩처 순서 카운트 (POC) 는 강화 계층 팩처들의 팩처 순서 카운트와 동일하게 설정된다. 이 경우, 이러한 스케일러블 또는 멀티뷰 코덱에서의 기본 계층 디코더에 의해 디코딩된 기본 계층 팩처의 실제적인 팩처 순서 카운트는, 그것이 AVC 디코더에 의해 디코딩될 때의 동일한 팩처의 팩처 순서

카운트 값과는 상이할 수도 있다.

[0168] 7) 기본 계층 디코딩된 픽처는 "장기 참조 (long-term reference) 를 위하여 이용됨" 으로서 표기된다.

[0169] 8) 가상적 참조 디코더 또는 버퍼링 모델의 코딩된 픽처 버퍼 동작들에 대하여, 기본 계층은 제로 비트 (zero bit) 들을 가지는 것으로서 고려된다.

[0170] 9) 가상적 참조 디코더 또는 버퍼링 모델의 디코딩된 픽처 버퍼 동작들에 대하여, 강화 계층들의 디코딩된 픽처들만이 고려된다.

[0171] 규칙들은 EL 을 코딩하는 코더 (예컨대, 인코더 및/또는 디코더) 에 의해 구현될 수 있다. 규칙들의 일부 또는 전부는 또한, RL 을 코딩하는 코더에 의해 구현될 수도 있다. 예를 들어, RL 코더는 RL 픽처들에 대한 정보를 EL 코더에 제공하도록 수정될 수 있다. RL 코더에 의해 수행되는 것으로서 표시된 임의의 기능성은 임의의 다른 외부 수단에 의해 수행될 수 있고, 외부 수단에 의해 수행되는 것으로서 표시된 임의의 기능성은 RL 코더에 의해 수행될 수 있다. 규칙들은 위에서 기본 계층 및 강화 계층의 측면에서 설명되지만, 규칙들은 임의의 강화 계층과, 기본 계층일 수도 있거나 계층이 아닐 수도 있는 그 대응하는 참조 계층에 적용할 수 있다. 각각의 규칙은 이하에서 차례로 설명된다.

#### 규칙 1 - RL 픽처들의 정보

[0173] 현재의 AU 가 RL 픽처를 포함하지 않을 경우, 외부 수단은 RL 픽처들에 대한 임의의 정보를 제공하지 않는다. 현재의 AU 가 RL 픽처를 포함할 경우, 외부 수단은 다음의 3 개의 타입들의 정보를 제공한다:

[0174] (1) RL 디코딩된 픽처의 디코딩된 샘플 값들;

[0175] (2) 루마 샘플들에서의 폭 및 높이, 컬러 포맷, 루마 비트 심도, 및 크로마 비트 심도를 포함하는, RL 디코딩된 픽처의 표현 포맷;

[0176] (3) RL 픽처가 IDR 픽처인지 여부.

[0177] 외부 수단은 상이한 표준들을 이용하는 멀티-계층 코딩을 지원하기 위하여 RL 픽처에 대하여 (1) 내지 (3) 만을 제공할 수 있다. EL 픽처들의 디코딩 및 EL 픽처들의 출력이 일부의 경우들에 있어서 (3) 에 종속될 수도 있으므로, 정보 (3) 은 중요할 수 있다.

[0178] 외부 수단은 선택적으로 다음의 타입의 정보를 제공할 수도 있다:

[0179] (4) 픽처가 프레임 또는 필드이든지 간에, 그리고 픽처가 필드일 때, (필드가 상부 필드 또는 하부 필드인지 여부를 표시하는) 필드 패리티.

[0180] 프레임은 프로그래시브 프레임 (progressive frame) 인 픽처를 지칭할 수 있다. 필드는, 프레임의 샘플로 우들의 절반, 홀수 또는 짝수 로우들을 포함하는, 인터레이스 픽처 (interlace picture) 인 픽처를 지칭할 수 있다. 필드 패리티는 필드가 상부 필드 또는 하부 필드인지 여부를 표시할 수 있다. 상부 필드는 짝수 번호의 샘플들로 우들을 포함하는 필드를 지칭할 수 있고, 로우 인덱싱 번호는 0 으로부터 시작한다. 하부 필드는 홀수 번호의 샘플들로 우들을 포함하는 필드를 지칭할 수 있고, 로우 인덱싱 번호는 0 으로부터 시작한다. (4) 가 제공되지 않을 경우, 디코딩된 RL 픽처는 프레임 픽처인 것으로 추론된다.

[0181] 정보 (1) 내지 (4) 중의 임의의 것은 애플리케이션 프로그래밍 인터페이스 (API) 들 또는 다른 적절한 방법들을 이용하여 제공될 수 있다. 하나의 예에서, RL 코더는 디코딩된 RL 픽처들을 전달하기 위한 API 를 가질 수 있고, EL 코더는 디코딩된 RL 픽처들을 받아들이기 위한 API 를 가질 수 있다. 2 개의 API 들은 디코딩된 RL 픽처 값들을 RL 코더로부터 EL 코더로 전송하기 위하여 접속될 수 있다. 또 다른 예에서, RL 코더는 디코딩된 RL 픽처들을 파일 또는 버퍼 내에 저장할 수 있고, EL 코더는 파일 또는 버퍼로부터 디코딩된 RL 픽처들을 액세스할 수 있다.

#### 규칙 2 - RL 픽처들의 출력

[0183] RL 코더 또는 또 다른 외부 수단은 RL 픽처들의 출력을 처리한다. 선택적으로, 오직 하나의 비-참조 계층 (예컨대, 하나의 EL) 이 존재하고, 유일한 타겟 출력 계층이 있는 것으로 특정된다. 따라서, EL 코더는 임의의 RL 픽처들을 출력할 필요가 없다.

#### 규칙 3 - RL 디코딩된 픽처들에 대한 디코딩된 픽처 저장소

[0185] EL 코더는 RL 디코딩된 픽처에 대하여 메모리의 단지 하나의 디코딩된 픽처 저장소를 유지할 수 있고, RL 디코딩된 픽처들에 대한 디코딩된 픽처 저장소는 DPB 의 일부로서 고려되지 않는다. 디코딩된 RL 픽처들은 일반적으로 동일한 AU 내의 인터-계층 예측 (inter-layer prediction; ILP) 에서 이용되기만 하므로, 일단 EL 코더가 현재의 AU 의 디코딩을 완료하면, 현재의 AU 에 대한 디코딩된 RL 픽처들은 디코딩된 픽처 저장소로부터 제거될 수 있다. 그러므로, 디코딩된 픽처 저장소의 메모리 사이즈는 고정된 상태로 유지될 수 있고, RL 픽처들에 대한 디코딩된 픽처 저장소를 위하여 하나의 프레임 버퍼 사이즈가 있을 수 있다.

#### 규칙 4 - 액세스 유닛에 대한 RL 디코딩된 픽처의 연관성

[0187] RL 코더 또는 또 다른 외부 수단은 AU 에 대한 RL 디코딩된 픽처의 연관성을 처리한다. 예를 들어, 각각의 EL 픽처는 특정한 AU 에 속하고, RL 코더 또는 또 다른 외부 수단은 RL 디코딩된 픽처를 EL 에서의 AU 와 어떻게 일치시킬 것인지를 결정한다. 하나의 실시형태에서, RL 디코딩된 픽처는 타임스탬프 (timestamp) 들에 기초하여 AU 와 일치될 수 있다. EL 에 대한 비트스트림 및 BL 에 대한 비트스트림은 양자 모두 타임스탬프들을 포함하고, RL 코더 또는 또 다른 외부 수단은 타임스탬프들에 기초하여 RL 디코딩된 픽처를 EL 에서의 AU 와 일치시킨다.

#### 규칙 5 - RL 디코딩된 픽처의 계층 ID 의 설정

[0189] EL 코더는 RL 디코딩된 픽처의 nuh\_layer\_id 를 0 으로 설정한다. 변수 nuh\_layer\_id 는 멀티-계층 코딩에서 계층의 식별자를 지칭할 수 있다. 대안적으로, EL 코더는 RL 디코딩된 픽처의 nuh\_layer\_id 를, AU 에서의 모든 EL 픽처들 중에서 nuh\_layer\_id 의 최저 값을 가지는 EL 픽처의 인터-계층 참조 픽처 세트에서의 엔트리의 nuh\_layer\_id 로 설정한다. 예를 들어, ILP 를 위하여 이용될 수 있는 EL 이외의 하나 이상의 더 낮은 계층들이 있을 경우, EL 코더는 RL 디코딩된 픽처의 nuh\_layer\_id 를 이 계층들로부터의 최저 nuh\_layer\_id 값으로 설정한다.

#### 규칙 6 - RL 디코딩된 픽처의 픽처 순서 카운트의 설정

[0191] EL 코더는 RL 디코딩된 픽처의 POC 를, 강화 계층 픽처들의 POC 와 동일하게 설정한다. RL 디코딩된 픽처의 POC 는 EL 코더에 의해 디코딩될 때와, RL 코더에 의해 디코딩될 때에 상이할 수도 있다. EL 코더가 RL 디코딩된 픽처의 디코딩된 샘플 값들을 수신하기만 하므로, EL 코더는 RL 디코딩된 픽처의 POC 를 액세스할 수도 있거나 액세스하지 않을 수도 있다. EL 코더는 RL 코더에 의해 획득된 픽처의 원래의 POC 값을 반드시 알아야 하지 않을 수도 있다. EL 코더는 RL 디코딩된 픽처의 POC 를 EL 픽처의 POC 와 동일하도록 설정할 수 있다.

#### 규칙 7 - RL 디코딩된 픽처의 표기

[0193] EL 코더는 RL 디코딩된 픽처를 "장기 참조를 위하여 이용됨" 으로서 표기한다. RL 디코딩된 픽처가 "장기 참조를 위하여 이용됨" 으로서 표기되지 않을 경우, 어떤 스케일링 연산 (scaling operation) 들은 ILP 를 위하여 수행될 수도 있고, 이것은 RL 픽처 및 EL 픽처의 POC 값들의 델타 (delta) 에 의한 제산 연산 (division operation) 을 수반할 수 있다. EL 코더가 RL 디코딩된 픽처의 POC 를 EL 픽처의 POC 와 동일하도록 설정하므로, 이것은 0 에 의한 제산을 수반할 것이다. 따라서, 0 에 의한 제산을 방지하기 위하여, EL 코더는 RL 디코딩된 픽처를 "장기 참조를 위하여 이용됨" 으로서 표기할 수 있다.

#### 규칙 8 - 코딩된 픽처 버퍼 동작들

[0195] 가상적 참조 디코더 (hypothetical reference decoder; HRD) 의 코딩된 픽처 버퍼 (coded picture buffer; CPB) 동작들에 대하여, EL 코더는 RL 을 제로 비트들을 가지는 것으로 고려한다. HRD 는 네트워크로부터 비트스트림을 수신하고 비트스트림을 디코딩하기 위한 버퍼링 모델을 지칭할 수 있다. 예를 들어, 비트스트림은 압축된 픽처들인 코딩된 픽처들을 포함하고, 코딩된 픽처들은 디코딩을 위하여 CPB 내에 저장된다. EL 코더가 외부 수단으로부터 RL 디코딩된 픽처의 디코딩된 샘플 값들을 수신하므로, EL 코더는 CPB 동작들에 대하여 RL 픽처에 대한 코딩된 픽처들을 참작할 필요가 없다.

#### 규칙 9 - 디코딩된 픽처 버퍼 동작들

[0197] HRD 또는 버퍼링 모델의 디코딩된 픽처 버퍼 동작들에 대하여, EL 코더는 강화 계층들의 디코딩된 픽처들을 고려하기만 한다. 위에서 언급된 바와 같이, RL 디코딩된 픽처들은 EL 코더에 의해 디코딩되지 않고, DPB 의 일부인 것으로 고려되지 않는 별도의 디코딩된 픽처 저장소 내에 저장된다. 그러므로, EL 코더는 DPB 동작

들에 대하여 디코딩된 RL 픽처들을 참작하지 않는다.

[0198] 상이한 표준들을 이용하는 멀티-계층 코딩을 지원하는 방법

도 5 는 이 개시물의 하나의 실시형태에 따라, 비디오 정보를 코딩하는 방법을 예시하는 플로우차트이다. 방법은 상이한 표준들을 이용하는 멀티-계층 코딩을 위한 지원을 제공하는 것에 관한 것이다. 프로세스 (500) 는 실시형태에 따라, 인코더 (예컨대, 도 2a, 도 2b 등에서 도시된 바와 같은 인코더), 디코더 (예컨대, 도 3a, 도 3b 등에서 도시된 바와 같은 디코더), 또는 임의의 다른 컴포넌트에 의해 수행될 수도 있다. 프로세스 (500) 의 블록들은 도 3b 에서의 디코더 (33) 에 대하여 설명되지만, 프로세스 (500) 는 위에서 언급된 바와 같이, 인코더와 같은 다른 컴포넌트들에 의해 수행될 수도 있다. 디코더 (33) 의 계층 1 비디오 디코더 (30B) 및/또는 디코더 (33) 의 계층 0 디코더 (30A) 는 실시형태에 따라 프로세스 (500) 를 수행할 수도 있다. 도 5 에 대하여 설명된 모든 실시형태들은 별도로, 또는 서로 조합하여 구현될 수도 있다. 프로세스 (500) 에 관련된 어떤 세부사항들은 예컨대, 도 4 에 대하여, 위에서 그리고 이하에서 설명된다.

프로세스 (500) 는 블록 (501) 에서 시작된다. 디코더 (33) 는 강화 계층 (EL) 및 대응하는 참조 계층 (RL) 과 연관된 비디오 정보를 저장하기 위한 메모리를 포함할 수 있다.

블록 (502) 에서, 디코더 (33) 는 현재의 액세스 유닛 (AU) 에서의 참조 계층 픽처와 연관된 정보를 획득한다.

RL 은 표준 (예컨대, H.264/AVC) 을 이용하여 코딩된다. RL 을 코딩하기 위하여 이용된 표준은 또한, "RL 표준" 으로서 지칭될 수도 있다. RL 픽처와 연관된 정보는: (1) RL 픽처의 디코딩된 샘플 값들; (2) RL 픽처의 표현 포맷; 및 (3) RL 픽처가 순간적 디코딩 리프레시 (IDR) 픽처인지 여부의 표시로 이루어질 수 있다.

RL 픽처와 연관된 정보는 예를 들어, 디코더 (33) 의 외부인, 외부 수단 또는 장치에 의해 제공될 수 있다. 하나의 실시형태에서, 외부 수단은 RL 표준을 이용하여 비디오 정보를 코딩하도록 구성되는 코더이다. 예를 들어, 코더는 RL 또는 임의의 다른 외부 수단을 코딩하기 위하여 이용되었던 코더일 수 있다. RL 픽처의 표현 포맷은: RL 픽처의 폭, RL 픽처의 높이, RL 픽처의 컬러 포맷, RL 픽처의 루마 컴포넌트들의 비트 심도, 또는 RL 픽처의 크로마 컴포넌트들의 비트 심도 중의 적어도 하나를 포함할 수 있다. RL 픽처의 폭 및 RL 픽처의 높이는 루마 샘플들의 유닛들로 특정될 수도 있다. 하나의 실시형태에서, 정보는 하나 이상의 애플리케이션 프로그래밍 인터페이스 (API) 들을 이용하여 제공된다. 다른 실시형태들에서, 정보는 파일, 버퍼, 또는 다른 적절한 수단을 이용하여 제공된다.

일부의 실시형태들에서, 디코더 (33) 는 현재의 AU 에서의 RL 픽처와 연관된 추가적인 정보를 수신한다. 추가적인 정보는 (4) RL 픽처가 프레임 또는 필드인지 여부의 표시를 포함할 수 있다. RL 픽처가 필드일 경우, 추가적인 정보는: (5) 필드 패리티를 더 포함할 수 있다. 추가적인 정보는 선택적일 수 있다.

블록 (503) 에서, 디코더 (33) 는 획득된 정보에 기초하여 현재의 AU 에서의 EL 픽처를 코딩한다. EL 픽처는 RL 을 코딩하기 위하여 이용된 표준 (예컨대, H.265/HEVC 또는 SHVC) 과는 상이한 표준을 이용하여 코딩된다. 하나의 실시형태에서, RL 을 코딩하기 위하여 이용된 표준은 H.264/AVC 이고, EL 을 코딩하기 위하여 이용된 표준은 H.265/HEVC 또는 SHVC 이다. EL 을 코딩하기 위하여 이용된 표준은 "EL 표준" 으로서 지칭될 수도 있다.

어떤 실시형태들에서, 디코더 (33) 는 다음의 특징들 중의 하나 이상을 구현할 수도 있다:

• 디코더 (33) 는 RL 디코딩된 픽처들의 출력을 처리하지 않는다. 예를 들어, 디코더 (33) 는 RL 픽처의 디코딩된 샘플 값들을 출력하지 않도록 구성된다.

• 디코더 (33) 의 메모리는 디코딩된 픽처 버퍼 (DPB) 를 포함하고, 디코더 (33) 는 DPB 의 일부가 아닌 디코딩된 픽처 저장소 내에 RL 픽처의 디코딩된 샘플 값들을 저장한다.

• 디코더 (33) 는 EL 에서의 AU 에 대한 RL 픽처들의 연관성을 처리하지 않는다. 예를 들어, 디코더 (33) 는 RL 표준을 이용하여 비디오 정보를 코딩하도록 구성되는 코더로부터 RL 픽처의 AU 를 획득한다. 코더는 EL 비트스트림 및 BL 비트스트림에서의 타임스탬프들에 기초하여 RL 픽처를 EL 에서의 AU 에 대해 연관시킬 수도 있고, 디코더 (33) 에 의해 요청될 때에 RL 픽처의 AU 를 제공할 수도 있다.

• 디코더 (33) 는 RL 픽처의 계층 식별자 (예컨대, nuh\_layer\_ID) 를 EL 픽처의 계층 식별자와 동일하게 설정한다.

• 디코더 (33) 는 RL 픽처의 픽처 순서 카운트 (POC) 를 EL 픽처의 POC 와 동일하게 설정한다.

- [0210] · 디코더 (33) 는 RL 핵처를 장기 참조 핵처 (예컨대, "장기 참조를 위하여 이용됨") 로서 표기한다.
- [0211] · 디코더 (33) 의 메모리는 코딩된 핵처 버퍼 (CPB) 를 포함하고, 디코더 (33) 는 RL 을 CPB 동작들을 위하여 0 비트들을 가지는 것으로 고려한다. 예를 들어, 디코더 (33) 는 CPB 동작들에서 RL 을 0 비트들을 가지는 것으로 설정한다.
- [0212] · 디코더 (33) 는 DPB 동작들을 위하여 EL 의 디코딩된 핵처들만을 고려한다. 예를 들어, 디코더 (33) 는 DPB 동작들에서 RL 핵처의 디코딩된 샘플 값들을 참조하지 않도록 구성된다.
- [0213] 프로세스 (500) 는 블록 (504) 에서 종료된다. 블록들은 실시형태에 따라 프로세스 (500) 에서 추가 및/또는 생략될 수도 있고, 프로세스 (500) 의 블록들은 실시형태에 따라 상이한 순서들로 수행될 수도 있다.
- [0214] 이 개시물에서 상이한 표준들을 이용하는 멀티-계층 코딩을 위한 지원에 대하여 설명된 임의의 특징들 및/또는 실시형태들은 별도로 또는 그 임의의 조합으로 구현될 수도 있다. 예를 들어, 도 1 내지 도 4 및 개시물의 다른 부분들과 관련하여 설명된 임의의 특징들 및/또는 실시형태들은 도 5 와 관련하여 설명된 임의의 특징들 및/또는 실시형태들과의 임의의 조합으로 구현될 수도 있고, 그 반대도 마찬가지이다.
- [0215] 본원에서 개시된 정보 및 신호들은 다양한 상이한 기술들 및 기법들 중의 임의의 것을 이용하여 표현될 수도 있다. 예를 들어, 상기 설명의 전반에 걸쳐 참조될 수도 있는 데이터, 명령들, 커맨드 (command) 들, 정보, 신호들, 비트들, 심볼들, 및 칩들은 전입들, 전류들, 전자기파들, 자기장들 또는 입자들, 광학 필드들 또는 입자들, 또는 그 임의의 조합에 의해 표현될 수도 있다.
- [0216] 본원에서 개시된 실시형태들과 관련하여 설명된 다양한 예시적인 논리적 블록들, 회로들, 및 알고리즘 단계들은 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양자의 조합들로서 구현될 수도 있다. 하드웨어 및 소프트웨어의 이 교환가능성을 명확하게 예시하기 위하여, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 단계들은 일반적으로 그 기능성의 측면에서 위에서 설명되었다. 이러한 기능성이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 애플리케이션과, 전체적인 시스템에 부과된 설계 제약들에 종속된다. 숙련된 기술자들은 각각의 특별한 애플리케이션을 위한 다양한 방법들로 설명된 기능성을 구현할 수도 있지만, 이러한 구현 규정들은 본 발명의 범위로부터의 이탈을 야기시키는 것으로 해석되지 않아야 한다.
- [0217] 본원에서 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 이러한 기법들은 범용 컴퓨터들, 무선 통신 디바이스 핸드셋들, 또는 무선 통신 디바이스 핸드셋들 및 다른 디바이스들에서의 애플리케이션을 포함하는 다수의 용도들을 가지는 집적 회로 디바이스들과 같은 다양한 디바이스들 중의 임의의 것에서 구현될 수도 있다. 모듈들 또는 컴포넌트들로서 설명된 임의의 특징들은 집적된 로직 디바이스에서 함께, 또는 개별적이지만 상호 동작가능한 로직 디바이스들로서 별도로 구현될 수도 있다. 소프트웨어로 구현될 경우, 기법들은, 실행될 때, 위에서 설명된 방법들 중의 하나 이상을 수행하는 명령들을 포함하는 프로그램 코드를 포함하는 컴퓨터-판독가능 데이터 저장 매체에 의해 적어도 부분적으로 실현될 수도 있다. 컴퓨터-판독가능 데이터 저장 매체는 패키징 재료들을 포함할 수도 있는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있다. 컴퓨터-판독가능 매체는 동기식 랜덤 액세스 메모리 (synchronous dynamic random access memory; SDRAM) 와 같은 랜덤 액세스 메모리 (random access memory; RAM), 판독-전용 메모리 (read-only memory; ROM), 비-휘발성 랜덤 액세스 메모리 (non-volatile random access memory; NVRAM), 전기적 소거 가능 프로그래밍 가능 판독-전용 메모리 (electrically erasable programmable read-only memory; EEPROM), 플래시 메모리 (FLASH memory), 자기 또는 광학 데이터 저장 매체들 등과 같은 메모리 또는 데이터 저장 매체들을 포함할 수도 있다. 추가적으로 또는 대안적으로, 기법들은 전파된 신호들 또는 파 (wave) 들과 같이, 명령들 또는 데이터 구조들의 형태로 프로그램 코드를 반송하거나 통신하며 컴퓨터에 의해 액세스, 판독, 및/또는 실행될 수 있는 컴퓨터-판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.
- [0218] 프로그램 코드는, 하나 이상의 디지털 신호 프로세서 (digital signal processor; DSP) 들, 범용 마이크로프로세서들, 애플리케이션 특정 집적 회로 (application specific integrated circuit; ASIC) 들, 필드 프로그래밍 가능한 로직 어레이 (field programmable logic array; FPGA) 들, 또는 다른 등가의 집적 또는 개별 로직 회로부와 같은 하나 이상의 프로세서들을 포함할 수도 있는 프로세서에 의해 실행될 수도 있다. 이러한 프로세서는 이 개시물에서 설명된 기법들 중의 임의의 것을 수행하도록 구성될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안적으로, 프로세서는 임의의 기존의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한, 컴퓨팅 디바이스들의 조합, 예컨대, DSP 및 마이크로프로세서, 복수의 마이크로프로세서들, DSP 코어와 함께 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성

의 조합으로서 구현될 수도 있다. 따라서, 본원에서 이용된 바와 같은 용어 "프로세서"는 상기한 구조, 상기 구조의 임의의 조합, 또는 본원에서 설명된 기법들의 구현을 위해 적당한 임의의 다른 구조 또는 장치 중의 임의의 것을 지칭할 수도 있다. 게다가, 일부의 양태들에서, 본원에서 설명된 기능성은, 인코딩 및 디코딩을 위해 구성되거나, 조합된 비디오 인코더-디코더 (combined video encoder-decoder; CODEC) 내에 편입된 전용 소프트웨어 모듈들 또는 하드웨어 모듈들 내에서 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0219]

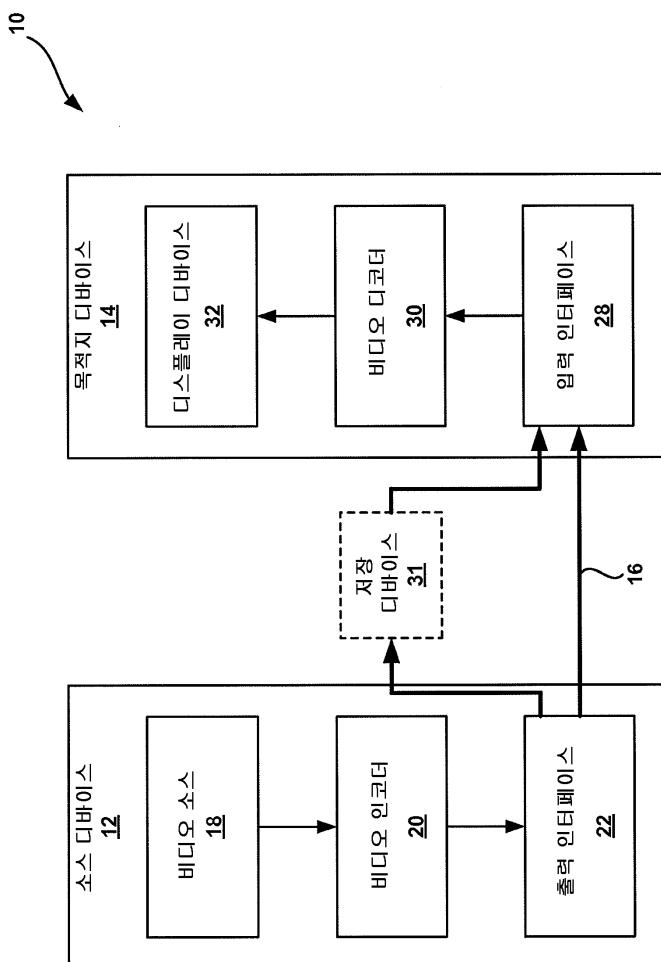
이 개시물의 기법들은 무선 핸드셋 (wireless handset), 집적 회로 (integrated circuit; IC) 또는 IC들의 세트 (예를 들어, 칩셋)를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위하여 이 개시물에서 설명되어 있지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛 내에 조합될 수도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

[0220]

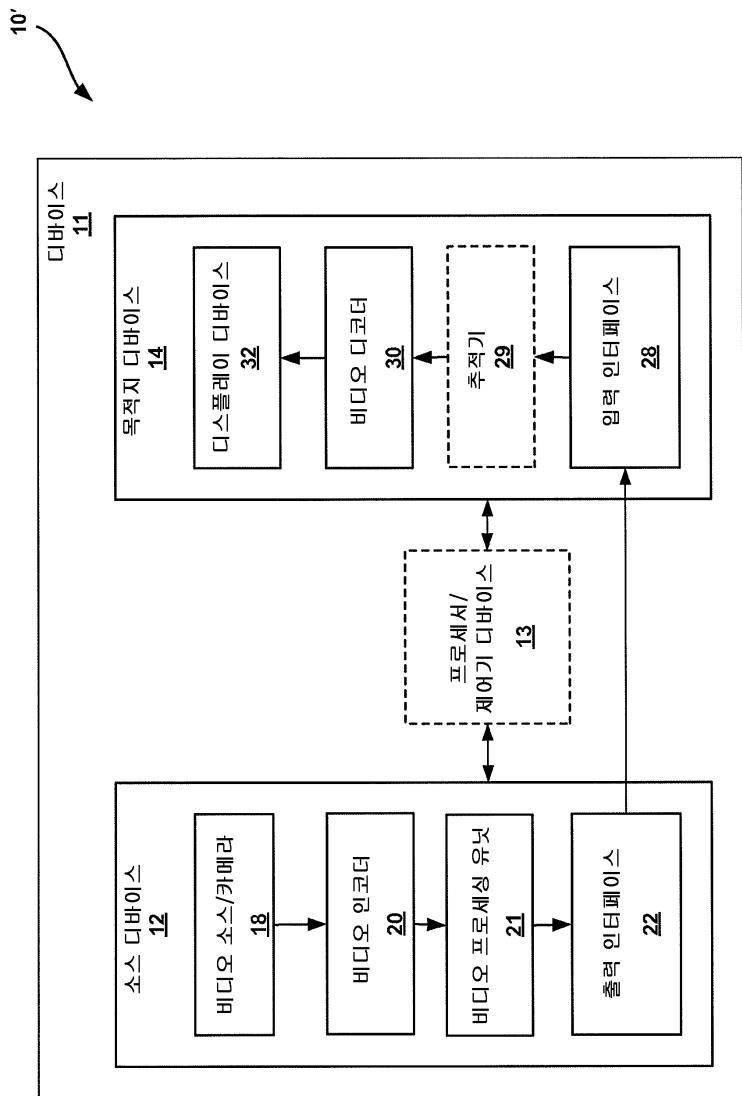
개시물의 다양한 실시형태들이 설명되었다. 이러한 그리고 다른 양태들은 다음의 청구항들의 범위 내에 있다.

## 도면

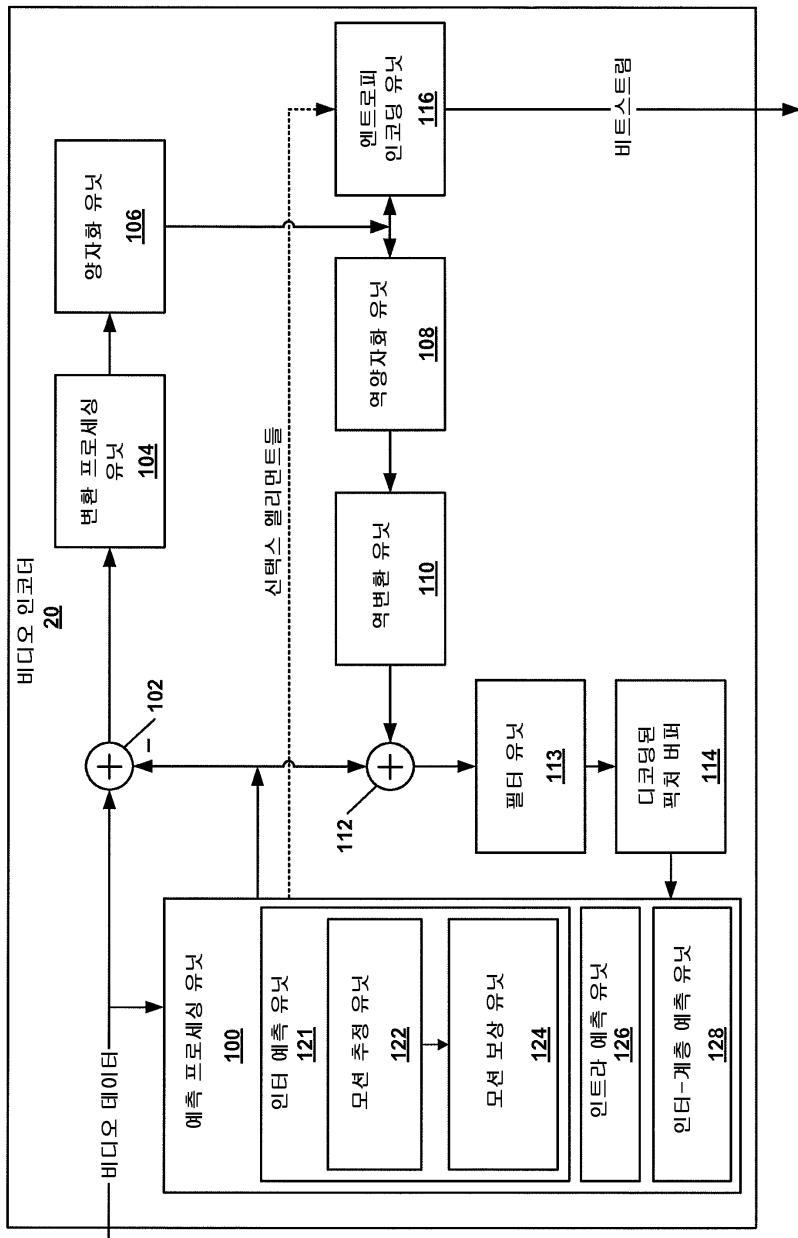
### 도면 1a



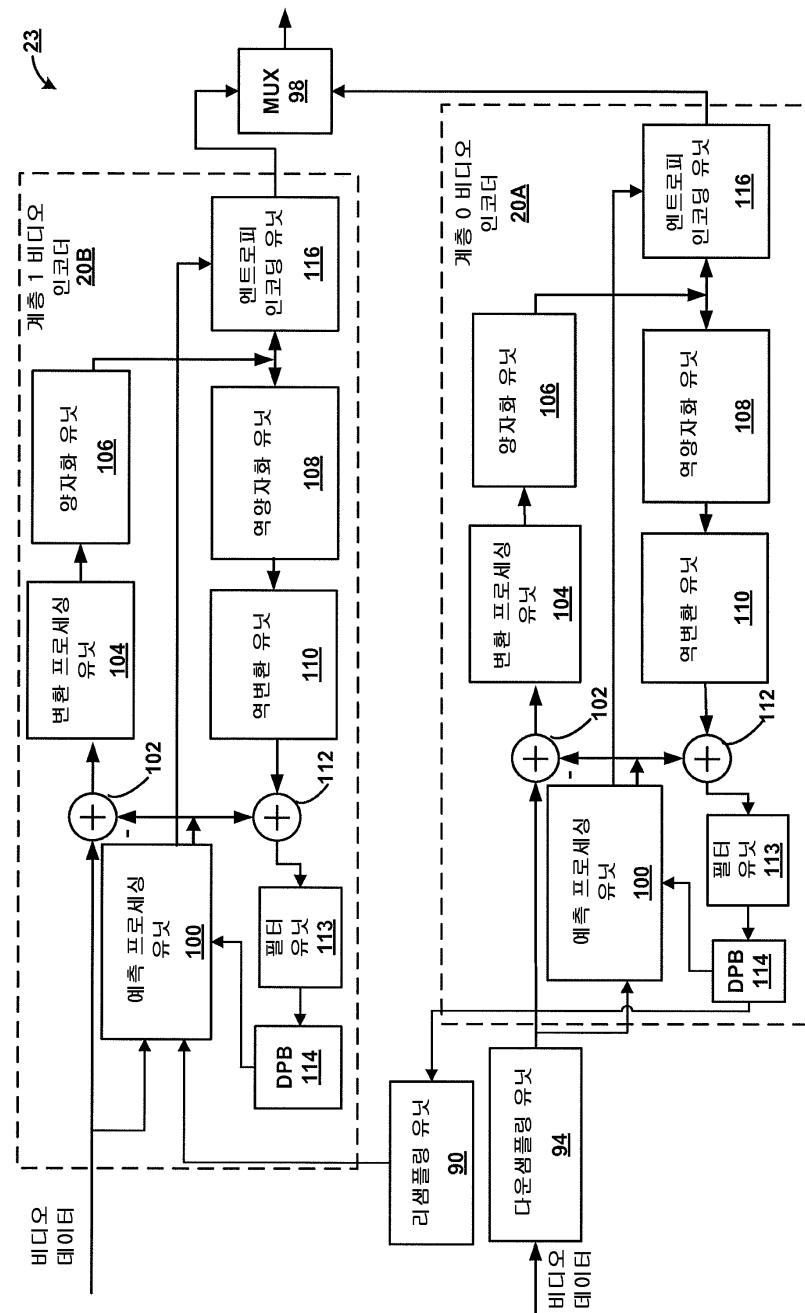
## 도면 1b



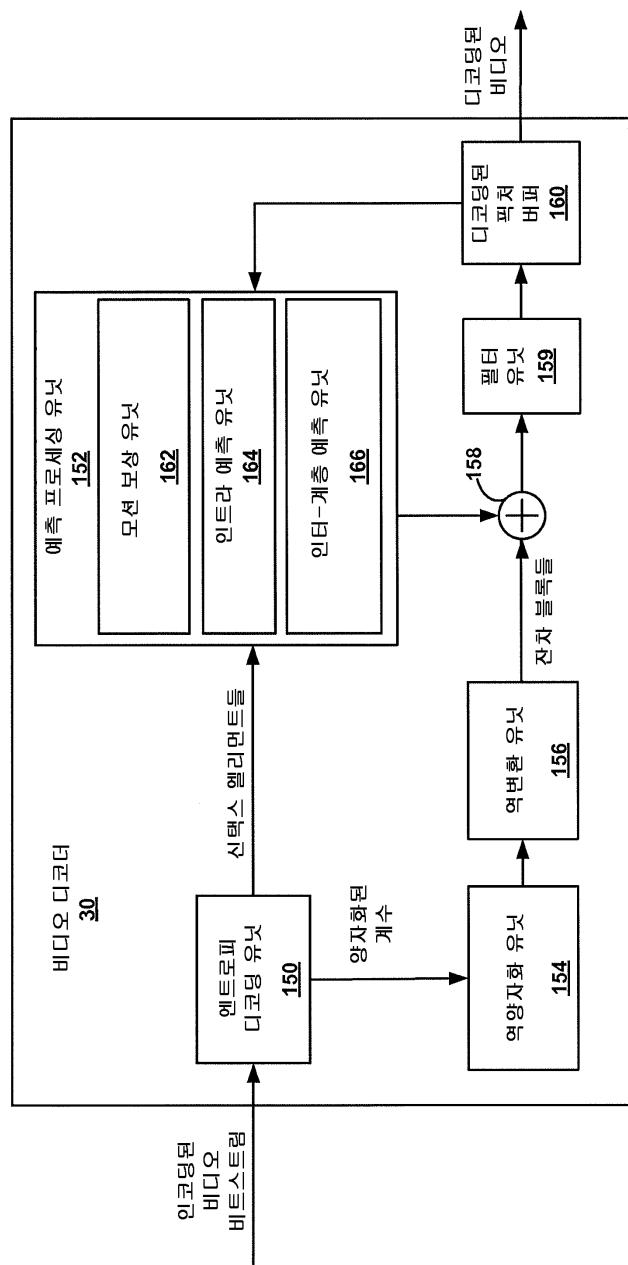
도면2a



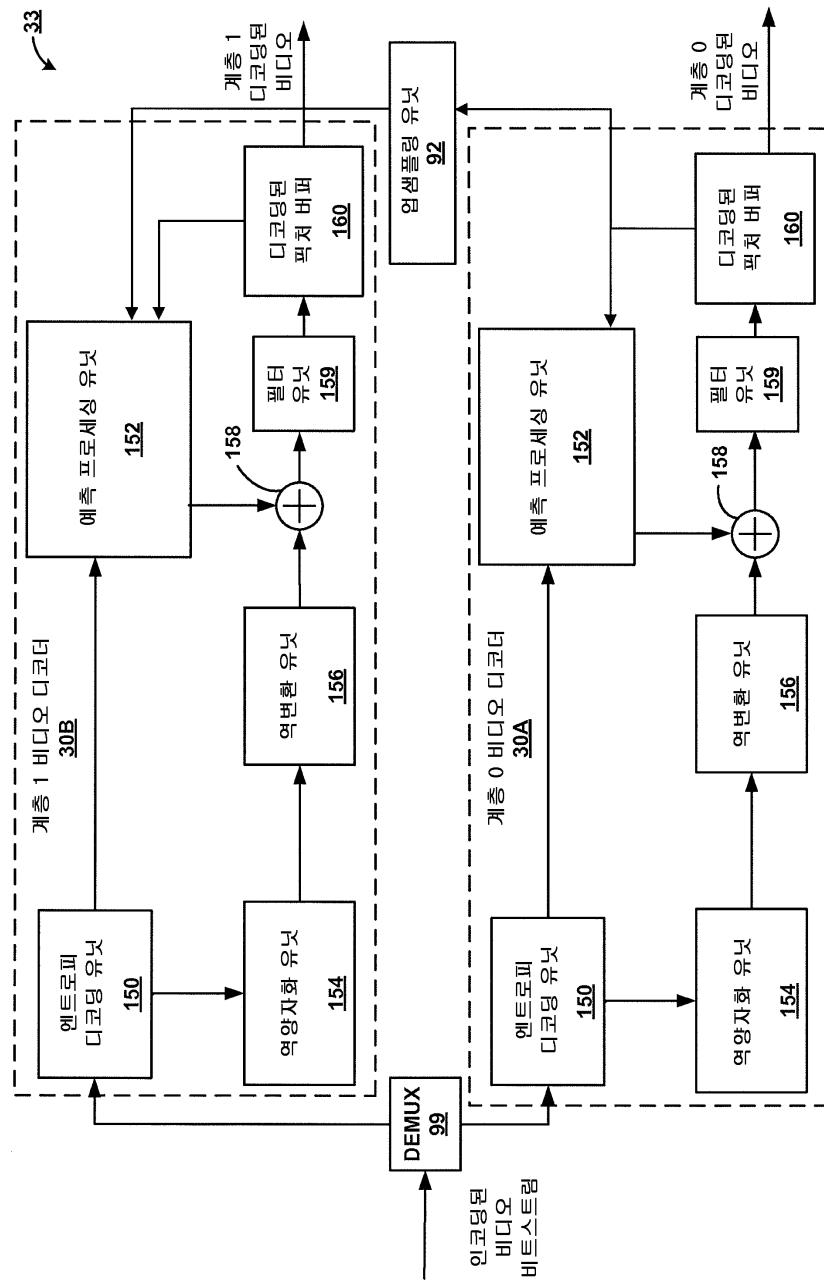
## 도면2b



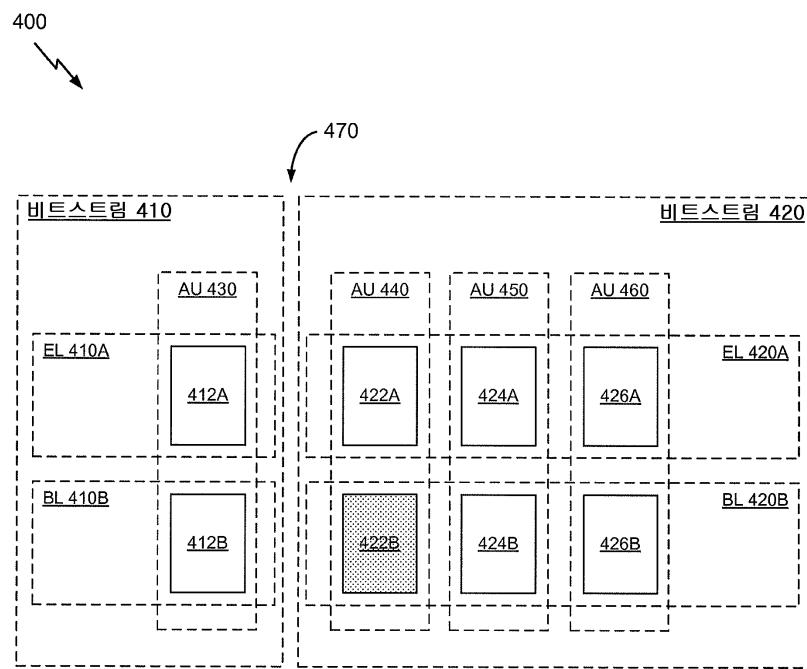
### 도면3a



### 도면3b



도면4



도면5

