



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0034319

(43) 공개일자 2016년03월29일

(51) 국제특허분류(Int. Cl.)

H04N 19/33 (2014.01) H04N 19/105 (2014.01)

H04N 19/187 (2014.01) H04N 19/29 (2014.01)

H04N 19/70 (2014.01)

(52) CPC특허분류

H04N 19/33 (2015.01)

H04N 19/105 (2015.01)

(21) 출원번호 10-2016-7002288

(22) 출원일자(국제) 2014년07월14일

심사청구일자 없음

(85) 번역문제출일자 2016년01월26일

(86) 국제출원번호 PCT/US2014/046546

(87) 국제공개번호 WO 2015/009629

국제공개일자 2015년01월22일

(30) 우선권주장

61/846,509 2013년07월15일 미국(US)

(뒷면에 계속)

(71) 출원인

켈컴 인코퍼레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

라파카 크리스티나칸트

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

세레진 바딤

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(뒷면에 계속)

(74) 대리인

특허법인코리아나

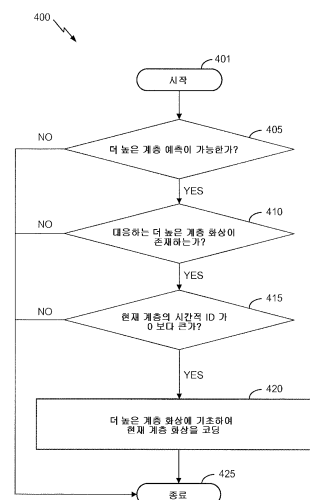
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 비디오 정보의 스케일러블 코딩을 위한 디바이스 및 방법

(57) 요약

비디오 정보를 코딩하도록 구성된 장치는 메모리 유닛 및 메모리 유닛과 통신하는 프로세서를 포함한다. 메모리 유닛은 현재의 계층 및 향상 계층과 연관된 비디오 정보를 저장하도록 구성되며, 현재의 계층은 현재의 화상을 갖는다. 프로세서는 현재의 계층이 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하고, 향상 계층이 현재의 화상에 대응하는 향상 계층 화상을 가지는지 여부를 결정하며, 현재의 계층이 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 향상 계층이 현재의 화상에 대응하는 향상 계층 화상을 가진다고 결정하는 것에 응답하여, 향상 계층 화상에 기초하여 현재의 화상을 코딩하도록 구성된다. 프로세서는 비디오 정보를 인코딩하거나 디코딩할 수도 있다.

대표도 - 도4



(52) CPC특허분류

H04N 19/187 (2015.01)

H04N 19/29 (2015.01)

H04N 19/70 (2015.01)

(72) 발명자

천 지안레

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

왕 예-쿠이

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장

61/847,931 2013년07월18일 미국(US)

61/884,978 2013년09월30일 미국(US)

14/329,804 2014년07월11일 미국(US)

명세서

청구범위

청구항 1

비디오 정보를 코딩하도록 구성된 장치로서,

현재 계층 및 향상 계층과 연관된 비디오 정보를 저장하도록 구성된 메모리 유닛으로서, 상기 현재 계층은 현재 화상을 갖는, 상기 메모리 유닛; 및

상기 메모리 유닛과 통신하는 프로세서를 포함하고,

상기 프로세서는,

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하고,

상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지는지 여부를 결정하며,

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하도록 구성된, 비디오 정보를 코딩하도록 구성된 장치.

청구항 2

제 1 항에 있어서,

상기 프로세서는 또한 상기 현재 화상이 0 보다 큰 시간적 ID 를 갖는지 여부를 결정하도록 구성되고,

상기 현재 화상을 코딩하는 것은, 상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고, 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지며, 상기 현재 화상이 0 보다 큰 시간적 ID 를 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 것을 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 3

제 1 항에 있어서,

상기 프로세서는 또한 상기 비디오 정보가 신호대 잡음비 (SNR) 또는 공간적 확장성을 나타내는지 여부를 결정하도록 구성되고,

상기 현재 화상을 코딩하는 것은, 상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고, 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지며, 상기 비디오 정보가 신호대 잡음비 (SNR) 또는 공간적 확장성을 나타낸다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 것을 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 4

제 1 항에 있어서,

상기 향상 계층은 상기 현재 계층의 계층 ID 보다 큰 계층 ID 를 갖는 하나 이상의 더 높은 계층들을 포함하고,

상기 향상 계층 화상은 상기 하나 이상의 더 높은 계층들 각각으로부터의 화상을 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 5

제 1 항에 있어서,

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부의 결정은 동일한 코딩된 비디오 시퀀스 (coded video sequence: CVS) 내의 상기 현재 계층에서 0 보다 큰 시간적 ID 를 갖는 각 화상에

대해 동일한, 비디오 정보를 코딩하도록 구성된 장치.

청구항 6

제 1 항에 있어서,

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부의 결정은 동일한 코딩된 비디오 시퀀스 (CVS) 내의 상기 현재 계층에서 0 과 동일한 시간적 ID 를 갖는 각 화상에 대해 동일한, 비디오 정보를 코딩하도록 구성된 장치.

청구항 7

제 1 항에 있어서,

상기 프로세서는 또한, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 것에 응답하여, 상기 코딩된 향상 계층 화상과 연관된 모션 정보를 상기 코딩된 현재 화상의 모션 정보로 대체하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 8

제 1 항에 있어서,

상기 프로세서는 또한, 상기 현재 화상을 포함하는 액세스 유닛 내의 각 화상을 코딩한 후, 0 보다 큰 계층 ID 를 갖는 각 계층에서의 상기 액세스 유닛 내의 화상과 연관된 모션 정보를 상기 각 계층 바로 아래에 있는 계층에서의 다른 화상의 모션 정보로 대체하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 9

제 1 항에 있어서,

상기 프로세서는 또한:

상기 현재 계층에서의 화상들에 대해 디블록킹 필터 및 샘플 적응 오프셋 (SAO) 을 디스에이블하고;

상기 현재 계층에서의 화상들에 대한 제약된 인트라 예측을 인에이블하며;

상기 현재 계층에서의 넌-제로 모션 정보를 사용하는 모션 예측을 디스에이블하고;

상기 향상 계층에서의 향상 계층 블록과 연관된 하나의 참조 화상 인덱스만이 상기 현재 화상에 대응하고, 상기 현재 화상 내의 동일장소에 배치된 현재 계층 블록이 양방향 예측을 사용하는 경우 상기 향상 계층에서의 양방향 예측을 디스에이블하며;

상기 디블록킹 필터 및 SAO 를 디스에이블하는 것, 상기 제약된 인트라 예측을 인에이블하는 것, 상기 모션 예측을 디스에이블하는 것, 및 상기 양방향 예측을 디스에이블하는 것에 응답하여, 상기 비디오 정보의 단일 루프 코딩을 수행하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 10

제 1 항에 있어서,

상기 프로세서는 적어도 상기 향상 계층 화상과 연관된 텍스처 정보 및 상기 현재 계층에서의 하나 이상의 화상들과 연관된 모션 정보를 사용하여 상기 현재 화상을 코딩함으로써 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 11

제 10 항에 있어서,

상기 프로세서는 또한:

상기 향상 계층에서의 다른 향상 계층 화상의 모션 정보를, 상기 다른 향상 계층 화상이 코딩된 후에 상기 다른 향상 계층 화상에 대응하는 다른 현재 계층 화상의 모션 정보로 대체하고;

상기 다른 향상 계층 화상의 상기 모션 정보를 사용하여 상기 현재 화상을 코딩하도록 구성되는, 비디오 정보를

코딩하도록 구성된 장치.

청구항 12

제 10 항에 있어서,

상기 프로세서는 또한:

상기 현재 계층에서의 다른 현재 계층 화상의 텍스처 정보를 상기 다른 현재 계층 화상에 대응하는 다른 향상 계층 화상의 텍스처 정보로, 상기 다른 향상 계층 화상이 코딩된 후에 대체하고;

상기 다른 현재 계층 화상의 상기 텍스처 정보를 사용하여 상기 현재 화상을 코딩하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 13

제 1 항에 있어서,

상기 장치는 인코더를 포함하고,

상기 프로세서는 또한 비트스트림에서 상기 비디오 정보를 인코딩하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 14

제 1 항에 있어서,

상기 장치는 디코더를 포함하고,

상기 프로세서는 또한 비트스트림에서 상기 비디오 정보를 디코딩하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 15

제 1 항에 있어서,

상기 장치는 컴퓨터들, 노트북들, 랩톱들, 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 전화기 핸드셋들, 스마트폰들, 스마트패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 및 차량내 컴퓨터들 중 하나 이상으로 이루어진 그룹으로부터 선택된 디바이스를 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 16

비디오 정보를 코딩하는 방법으로서,

현재 계층이 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하는 단계;

상기 향상 계층이 상기 현재 계층에서의 현재 화상에 대응하는 향상 계층 화상을 가지는지 여부를 결정하는 단계; 및

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 단계를 포함하는, 비디오 정보를 코딩하는 방법.

청구항 17

제 16 항에 있어서,

상기 현재 화상이 0 보다 큰 시간적 ID 를 갖는지 여부를 결정하는 단계를 더 포함하고,

상기 현재 화상을 코딩하는 단계는, 상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고, 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지며, 상기 현재 화상이 0 보다 큰 시간적 ID 를 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 단

계를 포함하는, 비디오 정보를 코딩하는 방법.

청구항 18

제 16 항에 있어서,

상기 비디오 정보가 신호대 잡음비 (SNR) 또는 공간적 확장성을 나타내는지 여부를 결정하는 단계를 더 포함하고,

상기 현재 화상을 코딩하는 단계는, 상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고, 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지며, 상기 비디오 정보가 신호대 잡음비 (SNR) 또는 공간적 확장성을 나타낸다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 단계를 포함하는, 비디오 정보를 코딩하는 방법.

청구항 19

제 16 항에 있어서,

상기 향상 계층 화상의 추가적인 표현이 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 단계 전에 필요한지 여부를 나타내는 플래그 또는 선택스 엘리먼트를 송신하거나 수신하는 단계를 더 포함하는, 비디오 정보를 코딩하는 방법.

청구항 20

제 16 항에 있어서,

상기 향상 계층은 상기 현재 계층의 계층 ID 보다 큰 계층 ID 를 갖는 하나 이상의 더 높은 계층들을 포함하고, 상기 향상 계층 화상은 상기 하나 이상의 더 높은 계층들 각각으로부터의 화상을 포함하는, 비디오 정보를 코딩하는 방법.

청구항 21

제 16 항에 있어서,

상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 단계에 응답하여, 상기 코딩된 향상 계층 화상과 연관된 모션 정보를 상기 코딩된 현재 화상의 모션 정보로 대체하는 단계를 더 포함하는, 비디오 정보를 코딩하는 방법.

청구항 22

제 16 항에 있어서,

상기 현재 화상을 포함하는 액세스 유닛 내의 각 화상을 코딩한 후, 0 보다 큰 계층 ID 를 갖는 각 계층에서의 상기 액세스 유닛 내의 화상과 연관된 모션 정보를 상기 각 계층 바로 아래에 있는 계층에서의 다른 화상의 모션 정보로 대체하는 단계를 더 포함하는, 비디오 정보를 코딩하는 방법.

청구항 23

제 16 항에 있어서,

상기 현재 계층에서의 화상들에 대해 디블록킹 필터 및 샘플 적응 오프셋 (SAO) 을 디스에이블하는 단계;

상기 현재 계층에서의 화상들에 대한 제약된 인트라 예측을 인에이블하는 단계;

상기 현재 계층에서의 n -제로 모션 정보를 사용하는 모션 예측을 디스에이블하는 단계;

상기 향상 계층에서의 향상 계층 블록과 연관된 하나의 참조 화상 인덱스만이 상기 현재 화상에 대응하고, 상기 현재 화상 내의 동일장소에 배치된 현재 계층 블록이 양방향 예측을 사용하는 경우 상기 향상 계층에서의 양방향 예측을 디스에이블하는 단계;

상기 디블록킹 필터 및 SAO 를 디스에이블하는 단계, 상기 제약된 인트라 예측을 인에이블하는 단계, 상기 모션 예측을 디스에이블하는 단계, 및 상기 양방향 예측을 디스에이블하는 단계에 응답하여, 상기 비디오 정보의 단

일 루프 코딩을 수행하는 단계를 더 포함하는, 비디오 정보를 코딩하는 방법.

청구항 24

제 16 항에 있어서,

상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 단계는, 상기 향상 계층 화상과 연관된 텍스트 정보 및 상기 현재 계층에서의 하나 이상의 화상들과 연관된 모션 정보를 사용하여 상기 현재 화상을 코딩하는 단계를 포함하는, 비디오 정보를 코딩하는 방법.

청구항 25

제 24 항에 있어서,

상기 향상 계층에서의 다른 향상 계층 화상의 모션 정보를, 상기 다른 향상 계층 화상이 코딩된 후에 상기 다른 향상 계층 화상에 대응하는 다른 현재 계층 화상의 모션 정보로 대체하는 단계; 및

상기 다른 향상 계층 화상의 상기 모션 정보를 사용하여 상기 현재 화상을 코딩하는 단계를 더 포함하는, 비디오 정보를 코딩하는 방법.

청구항 26

제 24 항에 있어서,

상기 현재 계층에서의 다른 현재 계층 화상의 텍스트 정보를 상기 다른 현재 계층 화상에 대응하는 다른 향상 계층 화상의 텍스트 정보로, 상기 다른 향상 계층 화상이 코딩된 후에 대체하는 단계; 및

상기 다른 현재 계층 화상의 상기 텍스트 정보를 사용하여 상기 현재 화상을 코딩하는 단계를 더 포함하는, 비디오 정보를 코딩하는 방법.

청구항 27

실행될 때, 장치로 하여금 프로세스를 수행하게 하는 코드를 포함하는 비일시적 컴퓨터 판독가능 매체로서,

상기 프로세스는,

현재 계층 및 향상 계층과 연관된 비디오 정보를 저장하는 것으로서, 상기 현재 계층은 현재 화상을 갖는, 상기 비디오 정보를 저장하는 것;

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하는 것;

상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지는지 여부를 결정하는 것; 및

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 것을 포함하는, 비일시적 컴퓨터 판독가능 매체.

청구항 28

제 27 항에 있어서,

상기 프로세스는 상기 현재 화상이 0 보다 큰 시간적 ID 를 갖는지 여부를 결정하는 것을 더 포함하고,

상기 현재 화상을 코딩하는 것은, 상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고, 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지며, 상기 현재 화상이 0 보다 큰 시간적 ID 를 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 것을 포함하는, 비일시적 컴퓨터 판독가능 매체.

청구항 29

비디오 정보를 코딩하도록 구성된 비디오 코딩 디바이스로서,

현재 계층 및 향상 계층과 연관된 비디오 정보를 저장하는 수단으로서, 상기 현재 계층은 현재 화상을 갖는, 상

기 비디오 정보를 저장하는 수단;

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하는 수단;

상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지는지 여부를 결정하는 수단; 및

상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 수단을 포함하는, 비디오 코딩 디바이스.

청구항 30

제 29 항에 있어서,

상기 현재 화상이 0 보다 큰 시간적 ID 를 갖는지 여부를 결정하는 수단을 더 포함하고,

상기 현재 화상을 코딩하는 것은, 상기 현재 계층이 상기 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고, 상기 향상 계층이 상기 현재 화상에 대응하는 향상 계층 화상을 가지며, 상기 현재 화상이 0 보다 큰 시간적 ID 를 가진다고 결정하는 것에 응답하여, 상기 향상 계층 화상에 기초하여 상기 현재 화상을 코딩하는 것을 포함하는, 비디오 코딩 디바이스.

발명의 설명

기술 분야

[0001] 본 개시는 비디오 코딩 및 압축의 분야에 관한 것이고, 특히 스케일러블 비디오 코딩 (SVC), 멀티뷰 비디오 코딩 (MVC), 또는 3D 비디오 코딩 (3DV) 에 관한 것이다.

배경 기술

[0002] 디지털 비디오 능력들은, 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDAs), 랩탑 또는 데스크탑 컴퓨터들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 화상회의 디바이스들 등 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), 현재 개발중인 HEVC (High Efficiency Video Coding) 표준, 및 이런 표준들의 확장판들에 의해 정의된 표준들에서 설명되는 것들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이런 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 보다 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0003] 비디오 압축 기법들은 비디오 시퀀스들에 고유한 리던던시를 감소시키거나 또는 제거하기 위해 공간 (인트라-화상) 예측 및/또는 시간 (인터-화상) 예측을 수행한다. 블록-기반 비디오 코딩에 있어서, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 일부 등) 는 트리블록들, 코딩 유닛들 (CUs), 및/또는 코딩 노드들로서도 지칭될 수도 있는 비디오 블록들로 파티셔닝될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 화상 내의 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 화상 내의 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수도 있으며, 참조 화상들은 참조 프레임들로서 지칭될 수도 있다.

[0004] 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 초래한다. 레지듀얼 데이터는 코딩될 원래 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라서 인코딩되며, 레지듀얼 데이터는 코딩된 블록과 예측 블록 사이의 차이를 나타낸다. 인트라-코딩된 블록은 인트라-코딩 모드 및 레지듀얼 데이터에 따라서 인코딩된다. 추가적인 압축을 위해, 레지듀얼 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있어, 레지듀얼 변환 계수들을 야기하고, 이들은 그 후 양자화될 수도 있다. 2차원 어레이로 처음에 배열된 양자화된 변환 계수들은 변환 계수들의 1차원 벡터를 발생시키기 위해 스캐닝될 수도 있으며, 훨씬 더 많은 압축을 달성하기 위해 엔트로피 코딩이 적용될 수도 있다.

발명의 내용

해결하려는 과제

[0005]

스케일러블 비디오 코딩 (SVC) 은 때때로 참조 계층 (RL) 으로서 지칭되는 베이스 계층 (BL), 및 하나 이상의 스케일러블 향상 계층들 (ELs) 이 사용되는 비디오 코딩을 지칭한다. SVC 에서, 베이스 계층은 품질의 베이스 레벨을 갖는 비디오 데이터를 반송할 수 있다. 하나 이상의 향상 계층들은 예를 들어 더 높은 공간, 시간, 및/또는 신호-대-잡음 (SNR) 레벨들을 지원하도록 추가적인 비디오 데이터를 반송할 수 있다. 향상 계층들은 이전에 인코딩된 계층에 대해 정의될 수도 있다. 예를 들어, 하위 계층 (bottom layer) 은 BL 로서 작용할 수도 있는 반면, 상위 계층 (top layer) 은 EL 로서 작용할 수도 있다. 중간 계층들 (middle layers) 은 EL 들 또는 RL 들, 또는 양자 모두로서 작용할 수도 있다. 예를 들어, 중간에 있는 계층은 베이스 계층 또는 임의의 개재하는 향상 계층들과 같은, 그것 아래의 계층들에 대해 EL 일 수도 있고, 동시에 그것 위의 하나 이상의 향상 계층들에 대해 RL 으로서 작용할 수도 있다. 유사하게, 멀티뷰 또는 HEVC 표준의 3D 확장에서, 다수의 뷰들이 존재할 수도 있고, 하나의 뷰에 대한 정보가 다른 뷰에 대한 정보를 코딩 (예를 들어, 인코딩 또는 디코딩) 하는데 이용될 수도 있다 (예를 들어, 모션 추정, 모션 벡터 예측 및/또는 다른 리던던시들).

[0006]

SVC 에서, 송신된 비트스트림은 다수의 계층들을 포함하고, 디코더는 디스플레이 디바이스의 비트레이트 제약들에 따라 그 다수의 계층들 중 하나 이상을 디코딩하기로 선택할 수도 있다. 예를 들어, 비트스트림은 2 개의 계층들, BL 및 EL 을 포함할 수도 있다. BL 을 디코딩하는 것은 3 mbps 를 요구할 수도 있고, BL 및 EL 양자 모두를 디코딩하는 것은 6 mbps 를 요구할 수도 있다. 4.5 mbps 의 용량을 갖는 디바이스의 경우, 디코더는 3 mbps 에서 BL 만을, 또는 디코딩되는 추가적인 EL 패킷들로부터 야기되는 화상 품질 향상을 이용하기 위해 4.5 mbps 아래에 유지되기에 충분한 EL 패킷들만을 포기하면서, BL 및 EL 의 조합을 디코딩하기로 선택할 수도 있다.

[0007]

그러나, 일부 구현들에서, EL 은 일반적으로 더 높은 품질 화상들을 가지기 때문에, EL 화상들은 더 큰 코딩 효율을 달성하기 위해 BL 화상들을 코딩하는데 사용될 수도 있다. 그러한 구현들에서, EL 화상들은 BL 화상들을 정확하게 디코딩하는데 필수적일 수도 있다. 이러한 제약은 상술된 바와 같이 디코더가 비트레이트 염려로 인해 BL 만을 (또는 EL 패킷들의 일부를 포기하면서 BL 및 EL 의 조합을) 디코딩하기로 선택할 수도 있는 경우에 문제를 제기한다. BL 을 코딩하는데 사용되는 EL 의 임의의 부분이 미싱 (missing) 된 경우, 디코더는 그 미싱된 부분에 대응하는 BL 의 부분을 대신 사용할 수도 있다. 그러한 경우에, 드리프트로서 알려진 현상이 도입된다. 드리프트는 EL 화상들을 사용하여 최적화되는, BL 화상들의 텍스처 정보 (예를 들어, 샘플들) 또는 모션 정보 (예를 들어, 모션 벡터들) 가 BL 화상들에 적용되는 경우 발생한다. 드리프트는 비디오 품질을 열화시킬 수도 있다.

[0008]

더 낮은 계층 (예를 들어, BL) 이 드리프트를 최소화하면서 더 높은 계층 (예를 들어, EL) 에 기초하여 코딩되는 것을 허용하는 것으로부터 야기되는 코딩 효율 이득을 활용하는 코딩 스킴이 원해진다.

과제의 해결 수단

[0009]

본 개시의 시스템들, 방법들 및 디바이스들 각각은 수개의 혁신적인 양태들을 가지며, 이들 양태들의 어느 단일의 양태가 여기에 개시된 바람직한 속성들에 대해 책임이 있는 것은 아니다.

[0010]

하나의 양태에서, 비디오 정보를 코딩 (예를 들어, 인코딩 또는 디코딩) 하도록 구성된 장치는 메모리 유닛 및 메모리 유닛과 통신하는 프로세서를 포함한다. 메모리 유닛은 현재의 계층 및 향상 계층과 연관된 비디오 정보를 저장하도록 구성되며, 현재의 계층은 현재의 화상을 갖는다. 프로세서는 현재의 계층이 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하고, 향상 계층이 현재의 화상에 대응하는 향상 계층 화상을 가지는지 여부를 결정하며, 현재의 계층이 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 향상 계층이 현재의 화상에 대응하는 향상 계층 화상을 가진다고 결정하는 것에 응답하여, 향상 계층 화상에 기초하여 현재의 화상을 코딩하도록 구성된다. 프로세서는 비디오 정보를 인코딩하거나 디코딩할 수도 있다.

[0011]

하나의 양태에서, 비디오 정보를 코딩 (예를 들어, 인코딩 또는 디코딩) 하는 방법은 현재의 계층이 향상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하는 단계; 향상 계층이 현재의 계층에서의 현재의 화상에 대응하는 향상 계층 화상을 가지는지 여부를 결정하는 단계; 및 현재의 계층이 향상 계층으로부터의 정

보를 사용하여 코딩될 수도 있고 항상 계층이 현재의 화상에 대응하는 항상 계층 화상을 가진다고 결정하는 것에 응답하여, 항상 계층 화상에 기초하여 현재의 화상을 코딩하는 단계를 포함한다.

[0012]

하나의 양태에서, 비밀시적 컴퓨터 판독가능 매체는, 실행될 때, 장치로 하여금 프로세스를 수행하게 하는 코드를 포함한다. 그 프로세스는 현재의 계층 및 항상 계층과 연관된 비디오 정보를 저장하는 것으로서, 현재의 계층은 현재의 화상을 갖는, 상기 저장하는 것; 현재의 계층이 항상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하는 것; 항상 계층이 현재의 화상에 대응하는 항상 계층 화상을 가지는지 여부를 결정하는 것; 및 현재의 계층이 항상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 항상 계층이 현재의 화상에 대응하는 항상 계층 화상을 가진다고 결정하는 것에 응답하여, 항상 계층 화상에 기초하여 현재의 화상을 코딩하는 것을 포함한다.

[0013]

하나의 양태에서, 비디오 정보를 코딩하도록 구성된 비디오 코딩 디바이스는 현재의 계층 및 항상 계층과 연관된 비디오 정보를 저장하는 수단으로서, 현재의 계층은 현재의 화상을 갖는, 상기 저장하는 수단; 현재의 계층이 항상 계층으로부터의 정보를 사용하여 코딩될 수도 있는지 여부를 결정하는 수단; 항상 계층이 현재의 화상에 대응하는 항상 계층 화상을 가지는지 여부를 결정하는 수단; 및 현재의 계층이 항상 계층으로부터의 정보를 사용하여 코딩될 수도 있고 항상 계층이 현재의 화상에 대응하는 항상 계층 화상을 가진다고 결정하는 것에 응답하여, 항상 계층 화상에 기초하여 현재의 화상을 코딩하는 수단을 포함한다.

도면의 간단한 설명

[0014]

도 1a 는 본 개시에 기술된 양태들에 따른 기법들을 이용할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템을 도시하는 블록도이다.

도 1b 는 본 개시에 기술된 양태들에 따른 기법들을 수행할 수도 있는 다른 예시의 비디오 인코딩 및 디코딩 시스템을 도시하는 블록도이다.

도 2a 는 본 개시에 기술된 양태들에 따른 기법들을 구현할 수도 있는 비디오 인코더의 예를 도시하는 블록도이다.

도 2b 는 본 개시에 기술된 양태들에 따른 기법들을 구현할 수도 있는 비디오 인코더의 예를 도시하는 블록도이다.

도 3a 는 본 개시에 기술된 양태들에 따른 기법들을 구현할 수도 있는 비디오 디코더의 예를 도시하는 블록도이다.

도 3b 는 본 개시에 기술된 양태들에 따른 기법들을 구현할 수도 있는 비디오 디코더의 예를 도시하는 블록도이다.

도 4 는 본 개시의 하나의 실시형태에 따라, 비디오 정보를 코딩하는 방법을 도시하는 플로우 차트를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0015]

여기에 기술된 소정의 실시형태들은 HEVC (고효율 비디오 코딩) 와 같은 진보된 비디오 코덱들의 컨텍스트에서 스케일러블 비디오 코딩을 위한 인터-계층 예측에 관련된다. 더욱 상세하게는, 본 개시는 HEVC 의 스케일러블 비디오 코딩 (SVC) 확장에서의 인터-계층 예측의 개선된 성능을 위한 시스템들 및 방법들에 관련된다.

[0016]

이하의 설명에서, 소정의 실시형태들에 관련된 H.264/AVC 기법들이 기술된다; HEVC 표준 및 관련된 기법들이 또한 논의된다. 소정의 실시형태들이 HEVC 및/또는 H.264 표준들의 컨텍스트에서 여기에 기술되지만, 본 기술 분야에 통상의 지식을 가진 자는 여기에 개시된 시스템들 및 방법들이 임의의 적합한 비디오 코딩 표준에 적용 가능할 수도 있다는 것을 인정할 수도 있다. 예를 들어, 여기에 개시된 실시형태들은 다음의 표준들 중 하나 이상에 적용 가능할 수도 있다: 그의 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장들을 포함하여, U-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 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC 로서 알려짐) 을 포함한다.

[0017]

HEVC 는 일반적으로 많은 점들에서 이전의 비디오 코딩 표준들의 프레임워크를 따른다. HEVC 에서의 예측의 유닛은 소정의 이전의 비디오 코딩 표준들에서의 그것 (예를 들어, 매크로블록) 과 상이하다. 사실, 매크로블록의 개념은 소정의 이전의 비디오 코딩 표준들에서 이해되는 바와 같이 HEVC 에 존재하지 않는다. 매크로블록은 다른 가능한 이익들 중에서 높은 유연성을 제공할 수도 있는 쿼드트리 스킵에 기초한 계층적 구조에

의해 대체된다. 예를 들어, HEVC 스킵 내에서, 3 개의 타입들의 블록들, 코딩 유닛 (CU), 예측 유닛 (PU), 및 변환 유닛 (TU) 이 정의된다. CU 는 영역 분할의 기본 유닛을 지칭할 수도 있다. CU 는 매크로블록의 개념과 유사한 것으로 고려될 수도 있지만, 그것은 최대 사이즈를 제한하지 않고 4 개의 동일한 사이즈 CU 들로의 반복적 분할을 허용하여 콘텐츠 적응성을 향상시킬 수도 있다. PU 는 인터/인트라 예측의 기본 유닛으로 고려될 수도 있고, 그것은 불규칙적인 이미지 패턴들을 효과적으로 코딩하기 위해 단일의 PU 에 다수의 임의적 형상 파티션들을 포함할 수도 있다. TU 는 변환의 기본 유닛으로 고려될 수도 있다. 그것은 PU 와는 독립적으로 정의될 수 있다; 그러나, 그것의 사이즈는 TU 가 속하는 CU 에 제한될 수도 있다. 3 개의 상이한 개념들로의 블록 구조의 이러한 분리는 각각이 그의 역할에 따라 최적화되는 것을 허용할 수도 있고, 이는 향상된 코딩 효율을 야기할 수도 있다.

[0018]

단지 설명의 목적으로, 여기에 개시된 소정의 실시형태들은 2 개의 계층들 (예를 들어, 베이스 계층과 같은 더 낮은 계층, 및 향상 계층과 같은 더 높은 계층) 만을 포함하는 예들로 기술된다. 그러한 예들은 다수의 베이스 및/또는 향상 계층들을 포함하는 구성들에 적용가능할 수도 있다는 것이 이해되어야 한다. 또, 설명의 용이를 위해, 다음의 개시는 소정의 실시형태들과 관련하여 용어들 "프레임들" 또는 "블록들" 을 포함한다. 그러나, 이러한 용어들은 제한하는 것을 의미하지 않는다. 예를 들어, 아래에 기술된 기법들은 블록들 (예를 들어, CU, PU, TU, 매크로블록들 등), 슬라이스들, 프레임들 등과 같은 임의의 적합 비디오 유닛들과 함께 사용될 수 있다.

[0019]

비디오 코딩 표준들

[0020]

비디오 이미지, TV 이미지, 스틸 이미지, 또는 비디오 리코더 또는 컴퓨터에 의해 생성된 이미지와 같은 디지털 이미지는 수평 및 수직 라인들로 배열된 화소들 또는 샘플들로 이루어질 수도 있다. 단일의 이미지 내의 화소들의 수는 통상적으로 수만개나 된다. 각각의 화소는 통상 루미넌스 및 크로미넌스 정보를 포함한다. 압축 없이, 이미지 인코더로부터 이미지 디코더로 전달될 정보의 양은 매우 거대하여 그것은 실시간 이미지 송신을 불가능하게 한다. 송신될 정보의 양을 감소시키기 위해, JPEG, MPEG 및 H.263 표준들과 같은 다수의 상이한 압축 방법들이 개발되어왔다.

[0021]

비디오 코딩 표준들은 그의 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장들을 포함하여, 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 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC 로서 알려짐) 을 포함한다.

[0022]

또, 새로운 비디오 코딩 표준, 즉 고효율 비디오 코딩 (HEVC) 이 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 및 ISO/IEC 동화상 전문가 그룹 (MPEG) 의 비디오 코딩에 관한 조인트 콜라보레이션 팀 (JCT-VC) 에 의해 개발되고 있는 중이다. HEVC 드래프트 10 에 대한 완전한 인용은 문서 JCVT-L1003, Bross et al., "High Efficiency Video Coding (HEVC) Text Specification Draft 10," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, Switzerland, January 14, 2013 to January 23, 2013 이다. HEVC 에 대한 멀티뷰 확장, 즉 MV-HEVC, 및 HEVC 에 대한 스케일러블 확장, 즉 SHVC 가 또한 각각 JCT-3V (3D 비디오 코딩 확장 개발에 관한 ITU-T/ISO/IEC 조인트 콜라보레이션 팀) 및 JCT-VC 에 의해 개발되고 있는 중이다.

[0023]

신규한 시스템들, 장치들, 및 방법들의 여러 양태들은 첨부하는 도면들을 참조하여 이하에 더욱 완전히 기술된다. 그러나, 본 개시는 많은 상이한 형태들로 구현될 수도 있고 본 개시에 걸쳐 제시된 임의의 특징의 구조 또는 기능에 제한되는 것으로 해석되지 않아야 한다. 오히려, 이들 양태들은 본 개시가 철저하고 완전하도록, 그리고 통상의 기술자들에게 본 개시의 범위를 완전히 전달하도록 제공된다. 여기의 교시들에 기초하여, 통상의 기술자는 본 개시의 임의의 다른 양태와 독립하여, 또는 결합하여 구현되는지 여부에 관계없이, 본 개시의 범위는 여기에 개시된 신규한 시스템들, 장치들 및 방법들의 임의의 양태를 커버하는 것으로 의도된다는 것을 인정해야 한다. 예를 들어, 여기에 진술된 임의의 수의 양태들을 사용하여 장치가 구현될 수도 있거나 방법이 실시될 수도 있다. 또, 본 개시의 범위는 여기에 진술된 본 개시의 여러 양태들에 더하여 또는 그 여러 양태들 이외에 다른 구조, 기능성, 또는 구조 및 기능성을 사용하여 실시되는 그러한 장치 또는 방법을 커버하도록 의도된다. 여기에 개시된 임의의 양태는 청구항의 하나 이상의 엘리먼트들에 의해 구현될 수도 있다는 것이 이해되어야 한다.

[0024]

특정의 양태들이 여기에 기술되지만, 이들 양태들의 많은 변형들 및 치환들은 본 개시의 범위 내에 있다. 바람직한 양태들의 일부 이익들 및 이점들이 언급되지만, 본 개시의 범위는 특정의 이익들, 사용들, 또는 목적들에 제한되지 않는다. 오히려, 본 개시의 양태들은 상이한 무선 기술들, 시스템 구성들, 네트워크들 및 송

신 프로토콜들에 넓게 적용가능한 것으로 의도되며, 이들 중 일부는 바람직한 양태들의 다음의 설명에서 그리고 도면들에서 예로써 도시된다. 상세한 설명 및 도면들은 제한하는 것보다는 오히려 본 개시의 단순한 설명이고, 본 개시의 범위는 첨부된 청구범위 및 이들의 균등물에 의해 정의된다.

[0025] 첨부된 도면들은 예들을 도시한다. 첨부된 도면들 내의 참조 번호들에 의해 나타난 엘리먼트들은 다음의 설명에서 유사한 참조 번호들에 의해 나타난 엘리먼트들에 대응한다. 본 개시에서, 순서적 단어들 (예를 들어, "제 1", "제 2", "제 3" 등) 로 시작하는 이름들을 갖는 엘리먼트들은 그 엘리먼트들이 특정의 순서를 갖는 것을 반드시 암시하는 것은 아니다. 오히려, 그러한 순서적 단어들은 단지 동일하거나 유사한 타입의 상이한 엘리먼트들을 지칭하기 위해 사용된다.

[0026] 비디오 코딩 시스템

[0027] 도 1a 는 본 개시에 기술된 양태들에 따른 기법들을 이용할 수도 있는 예시적인 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 사용될 때, 용어 "비디오 코더" 는 비디오 인코더들 및 비디오 디코더들 양쪽을 포괄적으로 지칭한다. 본 개시에서, 용어들 "비디오 코딩" 또는 "코딩" 은 비디오 인코딩 또는 비디오 디코딩을 포괄적으로 지칭할 수도 있다.

[0028] 도 1a 에 나타난 바와 같이, 비디오 코딩 시스템 (10) 은 목적지 모듈 (14)에 의해 이후의 시간에 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 모듈 (12) 을 포함한다. 도 1a 의 예에서, 소스 모듈 (12) 및 목적지 모듈 (14) 은 별개의 디바이스들 상에 있다 - 구체적으로는, 소스 모듈 (12) 은 소스 디바이스의 일부이고, 목적지 모듈 (14) 은 목적지 디바이스의 일부이다. 그러나, 소스 및 목적지 모듈들 (12, 14) 은 도 1b 의 예에서 도시되는 바와 같이 동일한 디바이스 상에 있거나 그 디바이스의 일부일 수도 있다.

[0029] 다시 한번 도 1a 를 참조하면, 소스 모듈 (12) 및 목적지 모듈 (14) 은 데스크탑 컴퓨터들, 노트북 (예컨대, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전자기 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 임의의 광범위한 디바이스들을 포함할 수도 있다. 일부 경우들에서, 소스 모듈 (12) 및 목적지 모듈 (14) 는 무선 통신을 위해 구비될 수도 있다.

[0030] 목적지 모듈 (14) 는 링크 (16) 를 통해서 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 는 인코딩된 비디오 데이터를 소스 모듈 (12) 로부터 목적지 모듈 (14) 로 이동시키는 것이 가능한 소정 타입의 매체 또는 디바이스를 포함할 수도 있다. 도 1a 의 예에서, 링크 (16) 는 소스 모듈 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 모듈 (14) 로 실시간으로 송신가능하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 목적지 모듈 (14) 로 송신될 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 근거리 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은, 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 모듈 (12) 로부터 목적지 모듈 (14) 로의 통신을 용이하는 데 유용할 수도 있는 다른 장비를 포함할 수도 있다.

[0031] 대안적으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 선택적 저장 디바이스 (31) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스 (28) 에 의해 저장 디바이스 (31) 로부터 액세스될 수도 있다. 저장 디바이스 (31) 는 하드 드라이브, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 임의의 다양한 분포된 또는 국부적으로 액세스되는 데이터 저장 매체들을 포함할 수도 있다. 다른 예에서, 저장 디바이스 (31) 는 소스 모듈 (12) 에 의해 생성된 인코딩된 비디오를 유지할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 모듈 (14) 는 스트리밍 또는 다운로드를 통해서 저장 디바이스 (31) 로부터 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 모듈 (14) 로 송신가능한 서버의 임의의 형태일 수도 있다. 예시적인 파일 서버들은 (예컨대, 웹사이트용) 웹 서버, FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 모듈 (14) 는 인터넷 접속을 포함하여, 임의의 표준 데이터 접속을 통해서 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀, 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 양자의 조합을 포함할 수도 있다. 저장 디바이스 (31) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다.

- [0032] 본 개시의 기법들은 무선 애플리케이션들 또는 설정들에 제한되지 않는다. 기법들은 지상파 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예를 들어 인터넷을 통한 스트리밍 비디오 송신들 (예를 들어, HTTP 를 통한 동적 적응형 스트리밍 (DASH) 등), 데이터 저장 매체 상에의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은 임의의 다양한 멀티미디어 애플리케이션들을 지원하는 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.
- [0033] 도 1a 의 예에서, 소스 모듈 (12) 은 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 일부 경우들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 모듈 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스, 예를 들어, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 공급 인터페이스, 및/또는 소스 비디오로서 컴퓨터 그래픽 데이터를 생성하는 컴퓨터 그래픽 시스템, 또는 그러한 소스들의 조합과 같은 소스를 포함할 수도 있다. 하나의 예로서, 비디오 소스 (18) 가 비디오 카메라인 경우, 소스 모듈 (12) 및 목적지 모듈 (14) 는 도 1b 의 예에서 도시된 바와 같이 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시에 기술된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.
- [0034] 캡처된, 미리 캡처된, 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 모듈 (12) 의 출력 인터페이스 (22) 를 통해 목적지 모듈 (14) 로 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안적으로) 디코딩 및/또는 플레이백을 위해, 목적지 모듈 (14) 또는 다른 디바이스들에 의한 나중의 액세스를 위해 저장 디바이스 (31) 상으로 저장될 수도 있다.
- [0035] 도 1a 의 예에서, 목적지 모듈 (14) 은 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 모듈 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 를 통해 통신되거나, 저장 디바이스 (31) 상에서 제공되는 인코딩된 비디오 데이터는 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30) 와 같은 비디오 디코더에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 신택스 엘리먼트들을 포함할 수도 있다. 그러한 신택스 엘리먼트들은 통신 매체 상에서 통신되거나, 저장 매체 상에 저장되거나, 파일 서버에 저장된 인코딩된 비디오 데이터와 함께 포함될 수도 있다.
- [0036] 디스플레이 디바이스 (32) 는 목적지 모듈 (14) 과 통합되거나 목적지 모듈 (14) 의 외부에 있을 수도 있다. 일부 예들에서, 목적지 모듈 (14) 은 통합된 디스플레이 디바이스를 포함하고 또한 외부 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 모듈 (14) 은 디스플레이 디바이스일 수도 있다. 일반적으로 디스플레이 디바이스 (32) 는 사용자에게 디코딩된 비디오 데이터를 디스플레이하고, 액정 디스플레이 (LCD), 플라스마 디스플레이, 유기 발광다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 임의의 다양한 디스플레이 디바이스들을 포함할 수도 있다.
- [0037] 관련된 양태들에서, 도 1b 는 예시의 비디오 인코딩 및 디코딩 시스템 (10') 을 도시하며, 여기서 소스 및 목적지 모듈들 (12, 14) 은 디바이스 또는 사용자 디바이스 (11) 상에 있거나 또는 그것의 일부이다. 디바이스 (11) 는 "스마트" 폰 등과 같은 전화 핸드셋일 수도 있다. 디바이스 (11) 는 소스 및 목적지 모듈들 (12, 14) 과 동작적 통신하는 선택적 제어기/프로세서 모듈 (13) 을 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. 일부 구현들에서, 비디오 프로세싱 유닛 (21) 은 도 1b 에 도시된 바와 같이 별개의 유닛이다; 그러나 다른 구현들에서는, 비디오 프로세싱 유닛 (21) 은 비디오 인코더 (20) 및/또는 프로세서/제어기 모듈 (13) 의 일부로서 구현될 수 있다. 시스템 (10') 은 또한 비디오 시퀀스에서 관심의 오브젝트를 추적할 수 있는 선택적 추적기 (29) 를 포함할 수도 있다. 추적될 오브젝트 또는 관심은 본 개시의 하나 이상의 양태들과 관련하여 기술된 기법에 의해 세그먼트화될 수도 있다. 관련된 양태들에서, 추적하는 것은 디스플레이 디바이스 (32) 에 의해, 단독으로 또는 추적기 (29) 와 협력하여 수행될 수도 있다. 도 1b 의 시스템 (10') 및 그것의 컴포넌트들은 도 1a 의 시스템 (10) 및 그것의 컴포넌트들과 그 외에는 유사하다.
- [0038] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 개발 중인 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오

압축 표준에 따라 동작할 수도 있고, HEVC Test Model (HM) 에 따를 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 대안적으로 MPEG-4, Part 10, 진보된 비디오 코딩 (AVC) 로서 지칭되는 ITU-T H.264 표준과 같은 다른 사유의 또는 산업상 표준들, 또는 그러한 표준들의 확장들에 따라 동작할 수도 있다.

그러나, 본 개시의 기법들은 임의의 특정의 코딩 표준에 제한되지 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 을 포함한다.

[0039]

도 1a 및 도 1b 의 예들에 도시되지 않지만, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통의 데이터 스트림 또는 별개의 데이터 스트림들에서 오디오 및 비디오 양자의 인코딩을 핸들링하기 위해 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜들에 따를 수도 있다.

[0040]

비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 반도체들 (ASICs), 필드 프로그래머블 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들과 같은 임의의 다양한 적합한 인코더 회로로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어로 구현되는 경우, 디바이스는 적합한 비일시적 컴퓨터 판독가능 매체에 그 소프트웨어에 대한 명령들을 저장하고 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하여 하드웨어로 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어떤 것은 각각의 디바이스에서 결합된 인코더/디코더 (코덱) 의 부분으로서 통합될 수도 있다.

[0041]

비디오 코딩 프로세스

[0042]

위에서 간략하게 언급된 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 인코딩한다. 비디오 데이터는 하나 이상의 화상들을 포함할 수도 있다. 화상들 각각은 비디오의 부분을 형성하는 스틸 이미지이다. 일부 예들에서, 화상은 비디오 "프레임" 으로서 지칭될 수도 있다. 비디오 인코더 (20) 가 비디오 데이터를 인코딩하는 경우, 비디오 인코더 (20) 는 비트스트림을 생성할 수도 있다. 비트스트림은 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 비트스트림은 코딩된 화상들 및 연관된 데이터를 포함할 수도 있다. 코딩된 화상은 화상의 코딩된 표현이다.

[0043]

비트스트림을 생성하기 위해, 비디오 인코더 (20) 는 비디오 데이터 내의 각 화상에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 화상들에 대해 인코딩 동작들을 수행할 때, 비디오 인코더 (20) 는 일련의 코딩된 화상들 및 연관된 데이터를 생성할 수도 있다. 연관된 데이터는 비디오 파라미터 세트들 (VPS), 시퀀스 파라미터 세트들, 화상 파라미터 세트들, 적응 파라미터 세트들, 및 다른 선택 구조들을 포함할 수도 있다. 시퀀스 파라미터 세트 (SPS) 는 화상들의 제로 이상의 시퀀스들에 적용가능한 파라미터들을 포함할 수도 있다. 화상 파라미터 세트 (PPS) 는 제로 이상의 화상들에 적용가능한 파라미터들을 포함할 수도 있다. 적응 파라미터 세트 (APS) 는 제로 이상의 화상들에 적용가능한 파라미터들을 포함할 수도 있다. APS 내의 파라미터들은 PPS 내의 파라미터들보다 더 변화하기 쉬운 파라미터들일 수도 있다.

[0044]

코딩된 화상을 생성하기 위해, 비디오 인코더 (20) 는 화상을 동일하게 사이징된 비디오 블록들로 파티셔닝할 수도 있다. 비디오 블록은 샘플들의 2차원 어레이일 수도 있다. 비디오 블록들 각각은 트리블록과 연관된다. 일부 예들에서, 트리블록은 최대 코딩 유닛 (LCU) 으로서 지칭될 수도 있다. HEVC 의 트리블록들은 H.264/AVC 와 같은 이전의 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, 트리블록은 특정의 사이즈로 반드시 제한되지는 않으며, 하나 이상의 코딩 유닛들 (CUs) 을 포함할 수도 있다. 비디오 인코더 (20) 는 트리블록들의 비디오 블록들을 CU 들과 연관된 비디오 블록들, 이리하여 명칭 "트리블록들" 로 파티셔닝하기 위해 쿼드트리 파티셔닝을 사용할 수도 있다.

[0045]

일부 예들에서, 비디오 인코더 (20) 는 화상을 복수의 슬라이스들로 파티셔닝할 수도 있다. 슬라이스들 각각은 CU 들의 정수 개수를 포함할 수도 있다. 일부 예들에서, 슬라이스는 정수 개수의 트리블록들을 포함한다. 다른 예들에서, 슬라이스의 경계는 트리블록 내에 있을 수도 있다.

[0046]

화상에 대한 인코딩 동작을 수행하는 것의 부분으로서, 비디오 인코더 (20) 는 화상의 각 슬라이스에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 슬라이스에 대해 인코딩 동작을 수행하는 경우, 비디오 인코더 (20) 는 그 슬라이스와 연관된 인코딩된 데이터를 생성할 수도 있다. 슬라이스와 연관된 인코딩된 데이터는 "코딩된 슬라이스" 로서 지칭될 수도 있다.

- [0047] 코딩된 슬라이스를 생성하기 위해, 비디오 인코더 (20) 는 슬라이스 내의 각 트리블록에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 트리블록에 대해 인코딩 동작을 수행하는 경우, 비디오 인코더 (20) 는 코딩된 트리블록을 생성할 수도 있다. 코딩된 트리블록은 트리블록의 인코딩된 버전을 표현하는 데이터들을 포함할 수도 있다.
- [0048] 비디오 인코더 (20) 가 코딩된 슬라이스를 생성하는 경우, 비디오 인코더 (20) 는 래스터 스캔 순서에 따라 슬라이스 내의 트리블록들에 대한 인코딩 동작들을 수행 (예를 들어, 인코딩) 할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 비디오 인코더 (20) 가 슬라이스 내의 트리블록들 각각을 인코딩할 때까지 슬라이스 내의 트리블록들의 가장 상위의 행을 가로질러 좌측에서 우측으로, 그 후 트리블록들의 다음의 하위 행을 가로질러 좌측에서 우측으로 등으로 진행되는 순서로 슬라이스의 트리블록들을 인코딩할 수도 있다.
- [0049] 래스터 스캔 순서에 따라 트리블록들을 인코딩하는 결과로서, 주어진 트리블록의 상측 및 좌측의 트리블록들은 인코딩되었을 수도 있지만, 주어진 트리블록의 하측 및 우측의 트리블록들은 아직 인코딩되지 않았다. 결과적으로, 비디오 인코더 (20) 는 주어진 트리블록을 인코딩할 때 주어진 트리블록의 상측 및 좌측의 트리블록들을 인코딩함으로써 생성된 정보에 액세스할 수 있을 수도 있다. 그러나, 비디오 인코더 (20) 는 주어진 트리블록을 인코딩할 때 주어진 트리블록의 하측 및 우측의 트리블록들을 인코딩함으로써 생성된 정보에 액세스할 수 없을 수도 있다.
- [0050] 코딩된 트리블록을 생성하기 위해, 비디오 인코더 (20) 는 비디오 블록을 점진적으로 더 작은 비디오 블록들로 분할하기 위해 트리블록의 비디오 블록에 대해 쿼드트리 파티셔닝을 반복적으로 수행할 수도 있다. 더 작은 비디오 블록들 각각은 상이한 CU 와 연관될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 트리블록의 비디오 블록을 4 개의 동일하게 사이징된 서브 블록들로 파티셔닝하고, 그 서브 블록들 중 하나 이상을 4 개의 동일하게 사이징된 서브 서브 블록들로 파티셔닝할 수도 있는 등등이다. 파티셔닝된 CU 는 그의 비디오 블록이 다른 CU 들과 연관된 비디오 블록들로 파티셔닝되는 CU 일 수도 있다. 난-파티셔닝된 CU 는 그의 비디오 블록이 다른 CU 들과 연관된 비디오 블록들로 파티셔닝되지 않는 CU 일 수도 있다.
- [0051] 비트스트림 내의 하나 이상의 선택스 엘리먼트들은 비디오 인코더 (20) 가 트리블록의 비디오 블록을 파티셔닝할 수도 있는 최대 회수를 나타낼 수도 있다. CU 의 비디오 블록은 형상이 정사각형일 수도 있다. CU 의 비디오 블록의 사이즈 (예를 들어, CU 의 사이즈) 는 8 x 8 화소들로부터 최대 64 x 64 화소들 이상의 최대 값을 갖는 트리블록의 비디오 블록의 사이즈 (예를 들어, 트리블록의 사이즈) 까지의 범위에 있을 수도 있다.
- [0052] 비디오 인코더 (20) 는 z-스캔 순서에 따라 트리블록의 각 CU 에 대해 인코딩 동작들을 수행 (예를 들어, 인코딩) 할 수도 있다. 즉, 비디오 인코더 (20) 는 그 순서로 좌상측 CU, 우상측 CU, 좌하측 CU, 및 그 후 우하측 CU 를 인코딩할 수도 있다. 비디오 인코더 (20) 가 파티셔닝된 CU 에 대해 인코딩 동작을 수행하는 경우, 비디오 인코더 (20) 는 z-스캔 순서에 따라 파티셔닝된 CU 의 비디오 블록의 서브 블록들과 연관된 CU 들을 인코딩할 수도 있다. 즉, 비디오 인코더 (20) 는 그 순서로 좌상측 서브 블록과 연관된 CU, 우상측 서브 블록과 연관된 CU, 좌하측 서브 블록과 연관된 CU, 및 그 후 우하측 서브 블록과 연관된 CU 를 인코딩할 수도 있다.
- [0053] z-스캔 순서에 따라 트리블록의 CU 들을 인코딩하는 것의 결과로서, 주어진 CU 의 상측, 좌상측, 우상측, 및 좌하측 CU 들이 인코딩되었을 수도 있다. 주어진 CU 의 우하측 CU 들은 아직 인코딩되지 않았다. 결과적으로, 비디오 인코더 (20) 는 주어진 CU 를 인코딩할 때 주어진 CU 와 이웃하는 일부 CU 들을 인코딩함으로써 생성된 정보에 액세스할 수 있을 수도 있다. 그러나, 비디오 인코더 (20) 는 주어진 CU 를 인코딩할 때 주어진 CU 와 이웃하는 다른 CU 들을 인코딩함으로써 생성된 정보에 액세스할 수 없을 수도 있다.
- [0054] 비디오 인코더 (20) 가 난-파티셔닝된 CU 를 인코딩할 때, 비디오 인코더 (20) 는 CU 에 대한 하나 이상의 예측 유닛들 (PUs) 을 생성할 수도 있다. CU 의 PU 들 각각은 CU 의 비디오 블록 내의 상이한 비디오 블록과 연관될 수도 있다. 비디오 인코더 (20) 는 CU 의 각 PU 에 대해 예측된 비디오 블록을 생성할 수도 있다. PU 의 예측된 비디오 블록은 샘플들의 블록일 수도 있다. 비디오 인코더 (20) 는 PU 에 대한 예측된 비디오 블록을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수도 있다.
- [0055] 비디오 인코더 (20) 가 PU 의 예측된 비디오 블록을 생성하기 위해 인트라 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 화상의 디코딩된 샘플들에 기초하여 PU 의 예측된 비디오 블록을 생성할 수도 있다. 비디오 인코더 (20) 가 CU 의 PU 들의 예측된 비디오 블록들을 생성하기 위해 인트라 예측을 사용한다면, CU

는 인트라 예측된 CU 이다. 비디오 인코더 (20) 가 PU 의 예측된 비디오 블록을 생성하기 위해 인터 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 화상 이외의 하나 이상의 화상들의 디코딩된 샘플들에 기초하여 PU 의 예측된 비디오 블록을 생성할 수도 있다. 비디오 인코더 (20) 가 CU 의 PU 들의 예측된 비디오 블록들을 생성하기 위해 인터 예측을 사용한다면, CU 는 인터 예측된 CU 이다.

[0056] 더욱이, 비디오 인코더 (20) 가 PU 에 대한 예측된 비디오 블록을 생성하기 위해 인터 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 에 대한 모션 정보를 생성할 수도 있다. PU 에 대한 모션 정보는 PU 의 하나 이상의 참조 블록들을 나타낼 수도 있다. PU 의 각각의 참조 블록은 참조 화상 내의 비디오 블록일 수도 있다. 참조 화상은 PU 와 연관된 화상 이외의 화상일 수도 있다. 일부 예들에서, 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.

[0057] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU 들에 대한 예측된 비디오 블록들을 생성한 후, 비디오 인코더 (20) 는 CU 의 PU 들에 대한 예측된 비디오 블록들에 기초하여 CU 에 대한 레지듀얼 데이터를 생성할 수도 있다. CU 에 대한 레지듀얼 데이터는 CU 의 PU 들에 대한 예측된 비디오 블록들에서의 샘플들과 CU 의 오리지널 비디오 블록 사이의 차이들을 나타낼 수도 있다.

[0058] 더욱이, 난-파티셔닝된 CU 에 대한 인코딩 동작을 수행하는 것의 부분으로서, 비디오 인코더 (20) 는 CU 의 레지듀얼 데이터를 CU 의 변환 유닛들 (TUs) 과 연관된 레지듀얼 데이터의 하나 이상의 블록들 (예를 들어, 레지듀얼 비디오 블록들) 로 파티셔닝하기 위해 CU 의 레지듀얼 데이터에 대해 반복적인 쿼드트리 파티셔닝을 수행할 수도 있다. CU 의 각 TU 는 상이한 레지듀얼 비디오 블록과 연관될 수도 있다.

[0059] 비디오 인코더 (20) 는 TU 들과 연관된 변환 계수 블록들 (예를 들어, 변환 계수들의 블록들) 을 생성하기 위해 TU 들과 연관된 레지듀얼 비디오 블록들에 하나 이상의 변환들을 적용할 수도 있다. 개념적으로, 변환 계수 블록은 변환 계수들의 2차원 (2D) 행렬일 수도 있다.

[0060] 변환 계수 블록을 생성한 후, 비디오 인코더 (20) 는 변환 계수 블록에 양자화 프로세스를 수행할 수도 있다. 양자화는 일반적으로 변환 계수들이 그 변환 계수들을 표현하기 위해 사용되는 데이터의 양을 감소시키기 위해 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 변환 계수들의 일부 또는 전부와 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안 m-비트 변환 계수로 라운드 다운될 수도 있으며, 여기서 n 은 m 보다 크다.

[0061] 비디오 인코더 (20) 는 각각의 CU 를 양자화 파라미터 (QP) 값과 연관시킬 수도 있다. CU 와 연관된 QP 값은 비디오 인코더 (20) 가 CU 와 연관된 변환 계수 블록들을 양자화하는 방법을 결정할 수도 있다. 비디오 인코더 (20) 는 CU 와 연관된 QP 값을 조정함으로써 CU 와 연관된 변환 계수 블록들에 적용된 양자화의 정도를 조정할 수도 있다.

[0062] 비디오 인코더 (20) 가 변환 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수 블록 내의 변환 계수들을 표현하는 선택스 엘리먼트들의 세트들을 생성할 수도 있다. 비디오 인코더 (20) 는 이들 선택스 엘리먼트들의 일부에 컨텍스트 적응형 이진 산술 코딩 (CABAC) 동작들과 같은 엔트로피 인코딩 동작들을 적용할 수도 있다. 컨텍스트 적응형 가변 길이 코딩 (CAVLC), 확률 구간 파티셔닝 엔트로피 (PIPE) 코딩, 또는 다른 이진 산술 코딩과 같은 다른 엔트로피 코딩 기법들이 또한 사용될 수 있을 것이다.

[0063] 비디오 인코더 (20) 에 의해 생성된 비트스트림은 네트워크 추상화 계층 (NAL) 유닛들의 시리즈를 포함할 수도 있다. NAL 유닛들 각각은 NAL 유닛에서의 데이터의 타입의 표시를 포함하는 선택스 구조 및 그 데이터를 포함하는 바이트들일 수도 있다. 예를 들어, NAL 유닛은 비디오 파라미터 세트, 시퀀스 파라미터 세트, 화상 파라미터 세트, 코딩된 슬라이스, 보충 강화 정보 (SEI), 액세스 유닛 구분 문자 (delimiter), 필러 데이터, 또는 다른 타입의 데이터를 표현하는 데이터를 포함할 수도 있다. NAL 유닛에서의 데이터는 여러 선택스 구조들을 포함할 수도 있다.

[0064] 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 수신하는 경우, 비디오 디코더 (30) 는 그 비트스트림에 대해 파싱 동작을 수행할 수도 있다. 비디오 디코더 (30) 가 파싱 동작을 수행할 때, 비디오 디코더 (30) 는 비트스트림으로부터 선택스 엘리먼트들을 추출할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 비디오 데이터의 화상들을 재구성할 수도 있다. 선택스 엘리먼트들에 기초하여 비디오 데이터를

재구성하는 프로세스는 일반적으로 신텍스 엘리먼트를 생성하기 위해 비디오 인코더 (20) 에 의해 수행된 프로세스의 역일 수도 있다.

[0065] 비디오 디코더 (30) 가 CU 와 연관된 신텍스 엘리먼트들을 추출한 후, 비디오 디코더 (30) 는 신텍스 엘리먼트들에 기초하여 CU 의 PU 들에 대한 예측된 비디오 블록들을 생성할 수도 있다. 또, 비디오 디코더 (30) 는 CU 의 TU 들과 연관된 변환 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 CU 의 TU 들과 연관된 레지듀얼 비디오 블록들을 재구성하기 위해 변환 계수 블록들에 대해 역변환들을 수행할 수도 있다. 예측된 비디오 블록들을 생성하고 레지듀얼 비디오 블록들을 재구성한 후, 비디오 디코더 (30) 는 그 예측된 비디오 블록들 및 레지듀얼 비디오 블록들에 기초하여 CU 의 비디오 블록을 재구성할 수도 있다. 이러한 방식으로, 비디오 디코더 (30) 는 비트스트림 내의 신텍스 엘리먼트들에 기초하여 CU 들의 비디오 블록들을 재구성할 수도 있다.

[0066] **비디오 인코더**

[0067] 도 2a 는 본 개시에서 기술된 양태들에 따른 기법들을 구현할 수도 있는 비디오 인코더의 예를 도시하는 블록도이다. 비디오 인코더 (20) 는 예를 들어 HEVC 에 대한 비디오 프레임의 단일의 계층을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 인코더 (20) 는 본 개시의 임의의 또는 모든 기법들을 수행하도록 구성될 수도 있다. 하나의 예로서, 예측 프로세싱 유닛 (100) 은 본 개시에 기술된 기법들의 임의의 것 또는 모두를 수행하도록 구성될 수도 있다. 다른 실시형태에서는, 비디오 인코더 (20) 는 본 개시에 기술된 기법들의 임의의 것 또는 모두를 수행하도록 구성되는 선택적 인터-계층 예측 유닛 (128) 을 포함한다. 다른 실시형태들에서, 인터-계층 예측은 예측 프로세싱 유닛 (100) (예를 들어, 인터 예측 유닛 (121) 및/또는 인트라 예측 유닛 (126)) 에 의해 수행될 수 있으며, 이러한 경우에 인터-계층 예측 유닛 (128) 은 생략될 수도 있다. 그러나, 본 개시의 양태들은 그렇게 제한되지 않는다. 일부 예들에서, 본 개시에 기술된 기법들은 비디오 인코더 (20) 의 여러 컴포넌트들 사이에 공유될 수도 있다. 일부 예들에서, 추가적으로 또는 대안적으로, 프로세서 (도시하지 않음) 는 본 개시에 기술된 임의의 또는 모든 기법들을 수행하도록 구성될 수도 있다.

[0068] 설명의 목적으로, 본 개시는 HEVC 코딩의 콘텍스트에서 비디오 인코더 (20) 를 기술한다. 그러나, 본 개시의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 2a 에 도시된 예는 단일 계층 코덱에 대한 것이다. 그러나, 도 2b 에 대해 더욱 기술되는 바와 같이, 비디오 인코더 (20) 의 일부 또는 전부가 다중 계층 코덱의 프로세싱을 위해 중복될 수도 있다.

[0069] 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라 코딩 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 주어진 비디오 프레임 또는 화상 내의 비디오에서의 공간적 리던던시를 감소시키거나 제거하기 위해 공간 예측에 의존한다. 인터 코딩은 비디오 시퀀스의 인접한 프레임들 또는 화상들 내의 비디오에서의 시간적 리던던시를 감소시키거나 제거하기 위해 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 임의의 수 개의 공간 기반 코딩 모드들을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 과 같은 인터-모드들은 임의의 수 개의 시간 기반 코딩 모드들을 지칭할 수도 있다.

[0070] 도 2a 의 예에서, 비디오 인코더 (20) 는 복수의 기능적 컴포넌트들을 포함한다. 비디오 인코더 (20) 의 기능적 컴포넌트들은 예측 프로세싱 유닛 (100), 레지듀얼 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 유닛 (110), 재구성 유닛 (112), 필터 유닛 (113), 디코딩된 화상 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 많은, 더 적은, 또는 상이한 기능적 컴포넌트들을 포함할 수도 있다. 더욱이, 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 고도로 집적될 수도 있지만, 설명의 목적으로 도 2a 의 예에서는 별개로 표현된다.

[0071] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 여러 소스들로부터 비디오 데이터를 수신할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 (예를 들어, 도 1a 또는 도 1b 에 도시된) 비디오 소스 (18) 또는 다른 소스로부터 비디오 데이터를 수신할 수도 있다. 비디오 데이터는 화상들의 시리즈를 표현한다. 비디오 데이터를 인코딩하기 위해, 비디오 인코더 (20) 는 화상들 각각에 대해 인코딩 동작을 수행할 수도 있다. 화상에 대해 인코딩 동작을 수행하는 것의 부분으로서, 비디오 인코더 (20) 는 화상의 각 슬라이스에 대해 인코딩 동작들을 수행할 수도 있다. 슬라이스에 대해 인코딩 동작을 수행하는 것의 부분으로서, 비디오 인코더 (20) 는 슬라이스 내의 트리블록들에 대해 인코딩 동작들을 수행할 수도 있다.

- [0072] 트리블록에 대해 인코딩 동작을 수행하는 것의 부분으로서, 예측 프로세싱 유닛 (100) 은 비디오 블록을 점진적으로 더 작은 비디오 블록들로 분할하기 위해 트리블록의 비디오 블록에 대해 쿼드트리 파티셔닝을 수행할 수도 있다. 더 작은 비디오 블록들 각각은 상이한 CU 와 연관될 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 트리블록의 비디오 블록을 4 개의 동일하게 사이징된 서브 블록들로 파티셔닝하고, 그 서브 블록들 중 하나 이상을 4 개의 동일하게 사이징된 서브 서브 블록들로 파티셔닝할 수도 있는 등등이다.
- [0073] CU 들과 연관된 비디오 블록들의 사이즈들은 8×8 샘플들로부터 최대 64×64 샘플들 이상의 최대값을 갖는 트리블록의 사이즈까지의 범위에 있을 수도 있다. 본 개시에서, " $N \times N$ " 및 " N 바이 N " 은 수직 및 수평 차원들 예를 들어, 16×16 샘플들 또는 16 바이 16 샘플들의 면에서 비디오 블록의 샘플 차원들을 지칭하기 위해 교환가능하게 사용될 수도 있다. 일반적으로 16×16 비디오 블록은 수직방향으로 16 개의 샘플들 ($y=16$) 및 수평 방향으로 16 개의 샘플들 ($x=16$) 을 가진다. 마찬가지로, $N \times N$ 블록은 일반적으로 수직 방향으로 N 개의 샘플들 및 수평 방향으로 N 개의 샘플들을 가지며, 여기서 N 은 음이 아닌 정수 값을 나타낸다.
- [0074] 더욱이, 트리블록에 대해 인코딩 동작을 수행하는 것의 부분으로서, 예측 프로세싱 유닛 (100) 은 트리블록에 대해 계층적 쿼드트리 데이터 구조를 생성할 수도 있다. 예를 들어, 트리블록은 쿼드트리 데이터 구조의 루트 노드에 대응할 수도 있다. 예측 프로세싱 유닛 (100) 이 트리블록의 비디오 블록을 4 개의 서브 블록들로 파티셔닝하는 경우, 루트 노드는 쿼드트리 데이터 구조에서 4 개의 차일드 노드들을 갖는다. 차일드 노드들 각각은 서브 블록들 중 하나와 연관된 CU 에 대응한다. 예측 프로세싱 유닛 (100) 이 서브 블록들 중 하나를 4 개의 서브 서브 블록들로 파티셔닝하는 경우, 서브 블록과 연관된 CU 에 대응하는 노드는 4 개의 차일드 노드들을 가질 수도 있으며, 이들 각각은 서브 서브 블록들 중 하나와 연관된 CU 에 대응한다.
- [0075] 쿼드트리 데이터 구조의 각 노드는 대응하는 트리블록 또는 CU 에 대한 선택스 데이터 (예를 들어, 선택스 엘리먼트들) 를 포함할 수도 있다. 예를 들어, 쿼드트리 내의 노드는 그 노드에 대응하는 CU 의 비디오 블록이 4 개의 서브 블록들로 파티셔닝 (예를 들어, 분할) 되는지 여부를 나타내는 분할 플래그를 포함할 수도 있다. CU 에 대한 선택스 엘리먼트들은 반복적으로 정의될 수도 있고, CU 의 비디오 블록이 서브 블록들로 분할되는지 여부에 의존할 수도 있다. 비디오 블록이 파티셔닝되지 않는 CU 는 쿼드트리 데이터 구조에서 리프 노드에 대응할 수도 있다. 코딩된 트리블록은 대응하는 트리블록에 대한 쿼드트리 데이터 구조에 기초한 데이터를 포함할 수도 있다.
- [0076] 비디오 인코더 (20) 는 트리블록의 각각의 년-파티셔닝된 CU 에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 년-파티셔닝된 CU 에 대해 인코딩 동작을 수행하는 경우, 비디오 인코더 (20) 는 년-파티셔닝된 CU 의 인코딩된 표현을 표현하는 데이터를 생성한다.
- [0077] CU 에 대해 인코딩 동작을 수행하는 것의 부분으로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU 들 사이에 CU 의 비디오 블록을 파티셔닝할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 여러 PU 사이즈들을 지원할 수도 있다. 특정의 CU 의 사이즈가 $2N \times 2N$ 이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들을, 및 $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $nL \times 2N$, $nR \times 2N$ 등의 대칭적 PU 사이즈들에서의 인터 예측을 지원한다. 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 비대칭 파티셔닝을 지원할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 직각인 CU 의 비디오 블록의 변들을 충족하지 않는 경계를 따라 CU 의 PU 들 사이에 CU 의 비디오 블록을 파티셔닝하기 위해 지오메트릭 파티셔닝을 수행할 수도 있다.
- [0078] 인터 예측 유닛 (121) 은 CU 의 각 PU 에 대해 인터 예측을 수행할 수도 있다. 인터 예측은 시간적 압축을 제공할 수도 있다. PU 에 대해 인터 예측을 수행하기 위해, 모션 추정 유닛 (122) 은 PU 에 대한 모션 정보를 생성할 수도 있다. 모션 보상 유닛 (124) 은 모션 정보 및 CU 와 연관된 화상 이외의 화상들 (예를 들어, 참조 화상들) 의 디코딩된 샘플들에 기초하여 PU 에 대한 예측된 비디오 블록을 생성할 수도 있다. 본 개시에서, 모션 보상 유닛 (124) 에 의해 생성된 예측된 비디오 블록은 인터 예측된 비디오 블록으로서 지칭될 수도 있다.
- [0079] 슬라이스들은 I 슬라이스들, P 슬라이스들, 또는 B 슬라이스들일 수도 있다. 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 PU 가 I 슬라이스, P 슬라이스, 또는 B 슬라이스에 있는지 여부에 따라 CU 의 PU 에 대한 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU 들은 인트라 예측된다. 이리하여, PU 가 I 슬라이스 내에 있는 경우, 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 PU 에 대해 인터 예측을 수행하

지 않는다.

- [0080] PU 가 P 슬라이스 내에 있는 경우, PU 를 포함하는 화상은 "리스트 0" 으로서 지칭된 참조 화상들의 리스트와 연관된다. 리스트 0 내의 참조 화상들 각각은 다른 화상들의 인터 예측을 위해 사용될 수도 있는 샘플들을 포함한다. 모션 추정 유닛 (122) 이 P 슬라이스 내의 PU 에 대해 모션 추정 동작을 수행하는 경우, 모션 추정 유닛 (122) 은 PU 에 대한 참조 블록을 위해 리스트 0 내의 참조 화상들을 검색할 수도 있다. PU 의 참조 블록은 PU 의 비디오 블록 내의 샘플들에 가장 근접하게 대응하는 샘플들의 세트, 예를 들어 샘플들의 블록 일 수도 있다. 모션 추정 유닛 (122) 은 참조 화상 내의 샘플들의 세트가 PU 의 비디오 블록 내의 샘플들에 얼마나 근접하게 대응하는지를 결정하는 다양한 메트릭들을 사용할 수도 있다. 예를 들어, 모션 추정 유닛 (122) 은 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 차이 메트릭들에 의해 참조 화상 내의 샘플들의 세트가 PU 의 비디오 블록 내의 샘플들에 얼마나 근접하게 대응하는지를 결정할 수도 있다.
- [0081] P 슬라이스 내의 PU 의 참조 블록을 식별한 후, 모션 추정 유닛 (122) 은 PU 와 참조 블록 사이의 공간적 변위를 나타내는 모션 벡터 및 참조 블록을 포함하는 리스트 0 내의 참조 화상을 나타내는 참조 인덱스를 생성할 수도 있다. 여러 예들에서, 모션 추정 유닛 (122) 은 여러 정밀도로 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 경우에, 참조 블록 값들은 참조 화상 내의 정수-위치 샘플 값들로부터 보간될 수도 있다. 모션 추정 유닛 (122) 은 PU 의 모션 정보로서 모션 벡터 및 참조 인덱스를 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 정보에 의해 식별된 참조 블록에 기초하여 PU 의 예측된 비디오 블록을 생성할 수도 있다.
- [0082] PU 가 B 슬라이스 내에 있는 경우, PU 를 포함하는 화상은 "리스트 0" 및 "리스트 1" 로서 지칭되는 참조 화상들의 2 개의 리스트들과 연관될 수도 있다. 일부 예들에서, B 슬라이스를 포함하는 화상은 리스트 0 및 리스트 1 의 조합인 리스트 조합과 연관될 수도 있다.
- [0083] 더욱이, PU 가 B 슬라이스 내에 있는 경우, 모션 추정 유닛 (122) 은 PU 에 대한 단방향 예측 또는 양방향 예측을 수행할 수도 있다. 모션 추정 유닛 (122) 이 PU 에 대한 단방향 예측을 수행하는 경우, 모션 추정 유닛 (122) 은 PU 에 대한 참조 블록을 위해 리스트 0 또는 리스트 1 의 참조 화상들을 검색할 수도 있다. 모션 추정 유닛 (122) 은 그 후 PU 와 참조 블록 사이의 공간적 변위를 나타내는 모션 벡터 및 참조 블록을 포함하는 리스트 0 또는 리스트 1 내의 참조 화상을 나타내는 참조 인덱스를 생성할 수도 있다. 모션 추정 유닛 (122) 은 PU 에 대한 모션 정보로서 참조 인덱스, 예측 방향 표시자, 및 모션 벡터를 출력할 수도 있다. 예측 방향 표시자는 참조 인덱스가 리스트 0 또는 리스트 1 내의 참조 화상을 나타내는지 여부를 나타낼 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 정보에 의해 표시된 참조 블록에 기초하여 PU 의 예측된 비디오 블록을 생성할 수도 있다.
- [0084] 모션 추정 유닛 (122) 이 PU 에 대한 양방향 예측을 수행하는 경우, 모션 추정 유닛 (122) 은 PU 에 대한 참조 블록을 위해 리스트 0 내의 참조 화상들을 검색하고 PU 에 대한 다른 참조 블록을 위해 리스트 1 내의 참조 화상들을 또한 검색할 수도 있다. 모션 추정 유닛 (122) 은 그 후 PU 와 참조 블록들 사이의 공간적 변위들을 나타내는 모션 벡터들 및 참조 블록들을 포함하는 리스트 0 및 리스트 1 내의 참조 화상들을 나타내는 참조 인덱스들을 생성할 수도 있다. 모션 추정 유닛 (122) 은 PU 의 모션 정보로서 PU 의 모션 벡터들 및 참조 인덱스들을 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 정보에 의해 표시된 참조 블록들에 기초하여 PU 의 예측된 비디오 블록을 생성할 수도 있다.
- [0085] 일부 예들에서, 모션 추정 유닛 (122) 은 엔트로피 인코딩 유닛 (116) 으로 PU 에 대한 모션 정보의 풀 세트를 출력하지 않는다. 오히려, 모션 추정 유닛 (122) 은 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 유사하다고 결정할 수도 있다. 이러한 예에서, 모션 추정 유닛 (122) 은, PU 와 연관된 신덱스 구조에서, PU 가 이웃하는 PU 와 동일한 모션 정보를 갖는다는 것을 비디오 디코더 (30) 에게 표시하는 값을 표시할 수도 있다. 다른 예에서, 모션 추정 유닛 (122) 은, PU 와 연관된 신덱스 구조에서, 이웃하는 PU 및 모션 벡터 차이 (MVD) 를 식별할 수도 있다. 모션 벡터 차이는 PU 의 모션 벡터와 표시된 이웃하는 PU 의 모션 벡터 사이의 차이를 나타낸다. 비디오 디코더 (30) 는 PU 의 모션 벡터를 결정하기 위해 표시된 이웃하는 PU 의 모션 벡터 및 모션 벡터 차이를 사용할 수도 있다. 제 2 PU 의 모션 정보를 시그널링할 때 제 1 PU 의 모션 정보를 참조함으로써, 비디오 인코더 (20) 는 더 적은 수의 비트들을 사용하여 제 2 PU 의 모션 정보를 시그널링할 수 있을 수도 있다.

- [0086] 도 4 를 참조하여 이하에 더욱 논의되는 바와 같이, 예측 프로세싱 유닛 (100) 은 도 4 에 도시된 방법들을 수행함으로써 PU (또는 임의의 다른 참조 계층 및/또는 항상 계층 블록들 또는 비디오 유닛들) 를 코딩 (예를 들어, 인코딩 또는 디코딩) 하도록 구성될 수도 있다. 예를 들어, (예를 들어, 모션 추정 유닛 (122) 및/또는 모션 보상 유닛 (124) 을 통해) 인터 예측 유닛 (121), 인트라 예측 유닛 (126), 또는 인터-계층 예측 유닛 (128) 은 함께 또는 별개로 도 4 에 도시된 방법들을 수행하도록 구성될 수도 있다.
- [0087] CU 에 대해 인코딩 동작을 수행하는 것의 부분으로서, 인트라 예측 유닛 (126) 은 CU 의 PU 들에 대해 인트라 예측을 수행할 수도 있다. 인트라 예측은 공간 압축을 제공할 수도 있다. 인트라 예측 유닛 (126) 이 PU 에 대해 인트라 예측을 수행하는 경우, 인트라 예측 유닛 (126) 은 동일한 화상 내의 다른 PU 들의 디코딩된 샘플들에 기초하여 PU 에 대한 예측 데이터를 생성할 수도 있다. PU 에 대한 예측 데이터는 예측된 비디오 블록 및 여러 선택스 엘리먼트들을 포함할 수도 있다. 인트라 예측 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들 내의 PU 들에 대해 인트라 예측을 수행할 수도 있다.
- [0088] PU 에 대해 인트라 예측을 수행하기 위해, 인트라 예측 유닛 (126) 은 PU 에 대한 예측 데이터의 다수의 세트들을 생성하기 위해 다수의 인트라 예측 모드들을 사용할 수도 있다. 인트라 예측 유닛 (126) 이 PU 에 대한 예측 데이터의 세트를 생성하기 위해 인트라 예측 모드를 사용하는 경우, 인트라 예측 유닛 (126) 은 인트라 예측 모드와 연관된 방향 및/또는 기울기로 PU 의 비디오 블록을 가로질러 이웃하는 PU 들의 비디오 블록들로부터 샘플들을 확장할 수도 있다. 이웃하는 PU 들은 PU 들, CU 들, 및 트리블록들에 대해 좌측에서 우측으로, 상측에서 하측으로의 인코딩 순서를 가정할 때 PU 의 상측, 우상측, 좌상측, 또는 좌측에 있을 수도 있다. 인트라 예측 유닛 (126) 은 PU 의 사이즈에 따라 여러 개수들의 인트라 예측 모드들, 예를 들어 33 개의 방향성 인트라 예측 모드들을 사용할 수도 있다.
- [0089] 예측 프로세싱 유닛 (100) 은 PU 에 대해 모션 보상 유닛 (124) 에 의해 생성된 예측 데이터 또는 PU 에 대해 인트라 예측 유닛 (126) 에 의해 생성된 예측 데이터 중에서 PU 에 대한 예측 데이터트 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 PU 에 대한 예측 데이터를 선택한다.
- [0090] 예측 프로세싱 유닛 (100) 이 인트라 예측 유닛 (126) 에 의해 생성된 예측 데이터를 선택하는 경우, 예측 프로세싱 유닛 (100) 은 PU 들에 대한 예측 데이터를 생성하는데 사용되었던 인트라 예측 모드, 예를 들어 선택된 인트라 예측 모드를 시그널링할 수도 있다. 예측 프로세싱 유닛 (100) 은 여러 방법들로 그 선택된 인트라 예측 모드를 시그널링할 수도 있다. 예를 들어, 선택된 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일한 개연성이 있다. 즉, 이웃하는 PU 의 인트라 예측 모드는 현재의 PU 에 대한 가장 개연성 있는 모드일 수도 있다. 따라서, 예측 프로세싱 유닛 (100) 은 선택된 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일하다는 것을 나타내는 선택스 엘리먼트를 생성할 수도 있다.
- [0091] 상술된 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 (예를 들어, 베이스 또는 참조 계층) 을 사용하여 현재의 블록 (예를 들어, EL 내의 현재의 블록) 을 예측하도록 구성된다. 그러한 예측은 인터-계층 예측으로서 지칭될 수도 있다. 인터-계층 예측 유닛 (128) 은 인터-계층 리던던시를 감소시키는 예측 방법들을 이용하여, 코딩 효율을 향상시키고 계산 자원 요건들을 감소시킨다. 인터-계층 예측의 일부 예들은 인터-계층 인트라 예측, 인터-계층 모션 예측, 및 인터-계층 레지듀얼 예측을 포함한다. 인터-계층 인트라 예측은 항상 계층에서의 현재의 블록을 예측하기 위해 베이스 계층에서의 동일 장소에 배치된 블록들의 재구성을 사용한다. 인터-계층 모션 예측은 항상 계층에서의 모션을 예측하기 위해 베이스 계층의 모션 정보를 사용한다. 인터-계층 레지듀얼 예측은 항상 계층의 레지듀얼을 예측하기 위해 베이스 계층의 레지듀얼을 사용한다. 인터-계층 예측 스킴들 각각은 이하에 더욱 상세히 논의된다.
- [0092] 예측 프로세싱 유닛 (100) 이 CU 의 PU 들에 대한 예측 데이터를 선택한 후, 레지듀얼 생성 유닛 (102) 은 CU 의 비디오 블록으로부터 CU 의 PU 들의 예측된 비디오 블록들을 감산 (예를 들어, 마이너스 부호에 의해 표시됨) 함으로써 CU 에 대한 레지듀얼 데이터를 생성할 수도 있다. CU 의 레지듀얼 데이터는 CU 의 비디오 블록 내의 샘플들의 상이한 샘플 컴포넌트들에 대응하는 2D 레지듀얼 비디오 블록들을 포함할 수도 있다. 예를 들어, 레지듀얼 데이터는 CU 의 PU 들의 예측된 비디오 블록들 내의 샘플들의 루미넌스 컴포넌트들과 CU 의 오리지널 비디오 블록 내의 샘플들의 루미넌스 컴포넌트들 사이의 차이들에 대응하는 레지듀얼 비디오 블록을 포함할 수도 있다. 또, CU 의 레지듀얼 데이터는 CU 의 PU 들의 예측된 비디오 블록들 내의 샘플들의 크로미넌스 컴포넌트들과 CU 의 오리지널 비디오 블록 내의 샘플들의 크로미넌스 컴포넌트들 사이의 차이들에 대

응하는 레지듀얼 비디오 블록을 포함할 수도 있다.

- [0093] 예측 프로세싱 유닛 (100) 은 CU 의 레지듀얼 비디오 블록들을 서브 블록들로 파티셔닝하기 위해 쿼드트리 파티셔닝을 수행할 수도 있다. 각각의 비분할된 레지듀얼 비디오 블록은 CU 의 상이한 TU 와 연관될 수도 있다. CU 의 TU 들과 연관된 레지듀얼 비디오 블록들의 사이즈들 및 위치들은 CU 의 PU 들과 연관된 비디오 블록들의 사이즈들 및 위치들에 기초할 수도 있거나 기초하지 않을 수도 있다. "레지듀얼 쿼드 트리" (RQT) 로서 알려진 쿼드트리 구조는 레지듀얼 비디오 블록들 각각과 연관된 노드들을 포함할 수도 있다. CU 의 TU 들은 RQT 의 리프 노드들에 대응할 수도 있다.
- [0094] 변환 프로세싱 유닛 (104) 은 TU 와 연관된 레지듀얼 비디오 블록에 하나 이상의 변환들을 적용함으로써 CU 의 각 TU 에 대해 하나 이상의 변환 계수 블록들을 생성할 수도 있다. 변환 계수 블록들 각각은 변환 계수들의 2D 행렬일 수도 있다. 변환 프로세싱 유닛 (104) 은 TU 와 연관된 레지듀얼 비디오 블록에 여러 변환들을 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 TU 와 연관된 레지듀얼 비디오 블록에 이산 코사인 변환 (DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 적용할 수도 있다.
- [0095] 변환 프로세싱 유닛 (104) 이 TU 와 연관된 변환 계수 블록을 생성한 후, 양자화 유닛 (106) 은 변환 계수 블록 내의 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.
- [0096] 비디오 인코더 (20) 는 여러 방법들로 QP 값을 CU 와 연관시킬 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 와 연관된 트리블록에 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) 는 비디오 인코더 (20) 가 트리블록의 상이한 인코딩된 표현들을 생성하는 경우 상이한 QP 값들을 CU 와 연관시킬 수도 있다. 비디오 인코더 (20) 는 주어진 QP 값이 최저 비트레이트 및 왜곡 메트릭을 갖는 트리블록의 코딩된 표현에서 CU 와 연관되는 경우 주어진 QP 값이 CU 와 연관된다는 것을 시그널링할 수도 있다.
- [0097] 역양자화 유닛 (108) 및 역변환 유닛 (110) 은 변환 계수 블록으로부터 레지듀얼 비디오 블록을 재구성하기 위해, 각각 변환 계수 블록에 역양자화 및 역변환들을 적용할 수도 있다. 재구성 유닛 (112) 은 TU 와 연관된 재구성된 비디오 블록을 생성하기 위해 예측 프로세싱 유닛 (100) 에 의해 생성된 하나 이상의 예측된 비디오 블록들로부터의 대응하는 샘플들에 재구성된 레지듀얼 비디오 블록을 가산할 수도 있다. 이러한 방식으로 CU 의 각 TU 에 대한 비디오 블록들을 재구성함으로써, 비디오 인코더 (20) 는 CU 의 비디오 블록을 재구성할 수도 있다.
- [0098] 재구성 유닛 (112) 이 CU 의 비디오 블록을 재구성한 후, 필터 유닛 (113) 은 CU 와 연관된 비디오 블록 내의 블록킹 아티팩트들을 감소시키기 위한 디블록킹 동작을 수행할 수도 있다. 하나 이상의 디블록킹 동작들을 수행한 후, 필터 유닛 (113) 은 디코딩된 화상 버퍼 (114) 에 CU 의 재구성된 비디오 블록을 저장할 수도 있다. 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 후속하는 화상들의 PU 들에 대한 인터 예측을 수행하기 위해 재구성된 비디오 블록을 포함하는 참조 화상을 사용할 수도 있다. 또, 인트라 예측 유닛 (126) 은 CU 와 동일한 화상 내의 다른 PU 들에 대한 인트라 예측을 수행하기 위해 디코딩된 화상 버퍼 (114) 내의 재구성된 비디오 블록들을 사용할 수도 있다.
- [0099] 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 선택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (116) 이 데이터를 수신하는 경우, 엔트로피 인코딩 유닛 (116) 은 엔트로피 인코딩된 데이터를 생성하기 위해 하나 이상의 엔트로피 인코딩 동작들을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 데이터에 대해 컨텍스트 적응형 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 선택스 기반 컨텍스트 적응형 이진 산술 코딩 (SBAC) 동작, 확률 구간 파티셔닝 엔트로피 (PIPE) 코딩 동작, 또는 다른 타입의 엔트로피 코딩 동작을 수행할 수도 있다. 엔트로피 인코딩 유닛 (116) 은 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다.
- [0100] 데이터에 대해 엔트로피 인코딩 동작을 수행하는 것의 부분으로서, 엔트로피 인코딩 유닛 (116) 은 컨텍스트 모델을 선택할 수도 있다. 엔트로피 인코딩 유닛 (116) 이 CABAC 동작을 수행하고 있는 경우, 컨텍스트 모델은 특정의 값들을 갖는 특정의 bin들의 가능성들의 추정들을 나타낼 수도 있다. CABAC 의 컨텍스트에서, 용

어 "빈" 은 선택스 엘리먼트의 이진화된 버전의 비트를 지칭하는데 사용된다.

[0101] **다중 계층 비디오 인코더**

[0102] 도 2b 는 본 개시에 기술된 양태들에 따른 기법들을 구현할 수도 있는 다중 계층 비디오 인코더 (23) 의 예를 도시하는 블록 다이어그램이다. 비디오 인코더 (23) 는 예를 들어 SHVC 및 멀티뷰 코딩을 위해 다중 계층 비디오 프레임들을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 인코더 (23) 는 본 개시의 기법들의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0103] 비디오 인코더 (23) 는 비디오 인코더 (20A) 및 비디오 인코더 (20B) 를 포함하며, 이들 각각은 비디오 인코더 (20) 로서 구성될 수도 있고 비디오 인코더 (20) 에 대해 상술된 기능들을 수행할 수도 있다. 또한, 참조 번호들의 재사용에 의해 표시되는 바와 같이, 비디오 인코더들 (20A 및 20B) 는 비디오 인코더 (20) 로서 적어도 일부의 시스템들 및 서브시스템들을 포함할 수도 있다. 비디오 인코더 (23) 는 2 개의 비디오 인코더들 (20A 및 20B) 을 포함하는 것으로서 도시되지만, 비디오 인코더 (23) 는 그와 같이 제한되지 않고, 임의의 수의 비디오 인코더 (20) 계층들을 포함할 수도 있다. 일부 실시형태들에서, 비디오 인코더 (23) 는 액세스 유닛 내의 각 화상 또는 프레임에 대해 비디오 인코더 (20) 를 포함할 수도 있다. 예를 들어, 5 개의 화상들을 포함하는 액세스 유닛은 5 개의 인코더 계층들을 포함하는 비디오 인코더에 의해 프로세싱되거나 인코딩될 수도 있다. 일부 실시형태들에서, 비디오 인코더 (23) 는 액세스 유닛 내의 프레임들보다 더 많은 인코더 계층들을 포함할 수도 있다. 일부 그러한 케이스들에서, 비디오 인코더 계층들의 일부는 일부 액세스 유닛들을 프로세싱할 때 비활성일 수도 있다.

[0104] 비디오 인코더들 (20A 및 20B) 에 더하여, 비디오 인코더 (23) 는 리샘플링 유닛 (90) 을 포함할 수도 있다. 리샘플링 유닛 (90) 은 일부 케이스들에서 예를 들어 향상 계층을 생성하기 위해 수신된 비디오 프레임의 베이스 계층을 업샘플링할 수도 있다. 리샘플링 유닛 (90) 은, 다른 정보는 아니고, 프레임의 수신된 베이스 계층과 연관된 특성의 정보를 업샘플링할 수도 있다. 예를 들어, 리샘플링 유닛 (90) 은 베이스 계층의 화소들의 수 또는 공간 사이즈를 업샘플링할 수도 있지만, 슬라이스들의 수 또는 화상 순서 카운트는 일정하게 유지할 수도 있다. 일부 케이스들에서, 리샘플링 유닛 (90) 은 수신된 비디오를 프로세싱하지 않을 수도 있고 및/또는 선택적일 수도 있다. 예를 들어, 일부 케이스들에서, 예측 프로세싱 유닛 (100) 이 업샘플링을 수행할 수도 있다. 일부 실시형태들에서, 리샘플링 유닛 (90) 은 계층을 업샘플링하고 슬라이스 경계 규칙들 및/또는 래스터 스캔 규칙들의 세트에 따르기 위해 하나 이상의 슬라이스들을 재조직화, 재정의, 변경, 또는 조정하도록 구성된다. 액세스 유닛 내의 베이스 계층, 또는 더 낮은 계층을 업샘플링하는 것으로서 주로 기술되지만, 일부 경우들에서, 리샘플링 유닛 (90) 은 계층을 다운샘플링할 수도 있다. 예를 들어, 비디오의 스트리밍 중에 대역폭이 감소되는 경우, 프레임은 업샘플링되는 대신에 다운샘플링될 수도 있다.

[0105] 리샘플링 유닛 (90) 은 더 낮은 계층 인코더 (예를 들어, 비디오 인코더 (20A)) 의 디코딩된 화상 버퍼 (114) 로부터 화상 또는 프레임 (또는 화상과 연관된 화상 정보) 을 수신하고 그 화상 (또는 그 수신된 화상 정보) 를 업샘플링하도록 구성될 수도 있다. 이러한 업샘플링된 화상은 그 후 더 낮은 계층 인코더와 동일한 액세스 유닛 내의 화상을 인코딩하도록 구성된 더 높은 계층 인코더 (예를 들어, 비디오 인코더 (20B)) 의 예측 프로세싱 유닛 (100) 에 제공될 수도 있다. 일부 경우들에서, 더 높은 계층 인코더는 더 낮은 계층 인코더로부터 제거된 하나의 계층이다. 다른 경우들에서, 도 2b 의 계층 0 비디오 인코더와 계층 1 인코더 사이의 하나 이상의 더 높은 계층 인코더들이 존재할 수도 있다.

[0106] 일부 경우들에서, 리샘플링 유닛 (90) 은 생략되거나 우회될 수도 있다. 그러한 경우들에서, 비디오 인코더 (20A) 의 디코딩된 화상 버퍼 (114) 로부터의 화상은 비디오 인코더 (20B) 의 예측 프로세싱 유닛 (100) 으로 직접 또는 적어도 리샘플링 유닛 (90) 으로 제공되지 않고 제공될 수도 있다. 예를 들어, 비디오 인코더 (20B) 로 제공된 비디오 데이터 및 비디오 인코더 (20A) 의 디코딩된 화상 버퍼 (114) 로부터의 참조 화상이 동일한 사이즈 또는 해상도인 경우, 참조 화상은 임의의 리샘플링 없이 비디오 인코더 (20B) 에 제공될 수도 있다.

[0107] 일부 실시형태들에서, 비디오 인코더 (23) 는 비디오 인코더 (20A) 에 비디오 데이터를 제공하기 전에 다운샘플링 유닛 (94) 을 사용하여 더 낮은 계층 인코더로 제공되도록 비디오 데이터를 다운샘플링한다. 대안적으로, 다운샘플링 유닛 (94) 은 비디오 데이터를 업샘플링 또는 다운샘플링할 수 있는 리샘플링 유닛 (90) 일 수도 있다. 또 다른 실시형태들에서, 다운샘플링 유닛 (94) 은 생략될 수도 있다.

[0108] 도 2b 에 도시된 바와 같이, 비디오 인코더 (23) 는 멀티플렉서 (98) 또는 MUX 를 더 포함할 수도 있다. 멀

터플렉서 (98) 는 비디오 인코더 (23) 로부터 결합된 비트스트림을 출력할 수 있다. 결합된 비트스트림은 비디오 인코더들 (20A 및 20B) 각각으로부터 비트스트림을 취하고 주어진 시간에 어느 비트스트림이 출력되는지를 교번함으로써 생성될 수도 있다. 일부 경우들에서 2 개의 (3 개 이상의 비디오 인코더 계층들의 경우에는 더 많은) 비트스트림들로부터의 비트들은 한번에 하나의 비트씩 교번될 수도 있지만, 많은 경우들에서 비트스트림들은 상이하게 결합된다. 예를 들어, 출력 비트스트림은 한번에 하나의 블록씩 선택된 비트스트림을 교번함으로써 생성될 수도 있다. 다른 예에서, 출력 비트스트림은 비디오 인코더들 (20A 및 20B) 각각으로부터의 블록들의 $n-1:1$ 비율을 출력함으로써 생성될 수도 있다. 예를 들어, 2 개의 블록들이 비디오 인코더 (20A) 로부터 출력된 각 블록에 대해 비디오 인코더 (20B) 로부터 출력될 수도 있다. 일부 실시형태들에서, 멀티플렉서 (98) 로부터의 출력 스트림은 사전 프로그래밍될 수도 있다. 다른 실시형태들에서, 멀티플렉서 (98) 는 소스 모듈 (12) 을 포함하는 소스 디바이스상의 프로세서로부터와 같이 비디오 인코더 (23) 의 외부의 시스템으로부터 수신된 제어 신호에 기초하여 비디오 인코더들 (20A 및 20B) 로부터의 비트스트림들을 결합할 수도 있다. 제어 신호는 비디오 소스 (18) 로부터의 비디오의 해상도 또는 비트레이트에 기초하여, 링크 (16) 의 대역폭에 기초하여, 사용자와 연관된 서브스크립션 (유료 서브스크립션 대 무료 서브스크립션) 에 기초하여, 또는 비디오 인코더 (23) 로부터 원하는 해상도 출력을 결정하기 위한 임의의 다른 팩터에 기초하여 생성될 수도 있다.

[0109] **비디오 디코더**

[0110] 도 3a 는 본 개시에 기술된 양태들에 따른 기법들을 구현할 수도 있는 비디오 디코더의 예를 도시하는 블록도이다. 비디오 디코더 (30) 는 예를 들어 HEVC 에 대한 비디오 비트스트림의 단일 계층을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 디코더 (30) 는 본 개시의 기법들의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 하나의 예로서, 모션 보상 유닛 (162) 및/또는 인트라 예측 유닛 (164) 이 본 개시에 기술된 기법들 중 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 하나의 실시형태에서, 비디오 디코더 (30) 는 본 개시에 기술된 기법들의 임의의 것 또는 전부를 수행하도록 구성되는 인터-계층 예측 유닛 (166) 을 선택적으로 포함할 수도 있다. 다른 실시형태들에서, 인터-계층 예측은 예측 프로세싱 유닛 (152) (예를 들어, 모션 보상 유닛 (162) 및/또는 인트라 예측 유닛 (164)) 에 의해 수행될 수 있고, 그 경우에 인터-계층 예측 유닛 (166) 은 생략될 수도 있다. 그러나, 본 개시의 양태들은 그렇게 제한되지 않는다. 일부 예들에서, 본 개시에 기술된 기법들은 비디오 디코더 (30) 의 여러 컴포넌트들 사이에서 공유될 수도 있다. 일부 예들에서, 추가적으로, 또는 대안적으로, 프로세서 (도시하지 않음) 는 본 개시에 기술된 기법들 중 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0111] 설명의 목적으로, 본 개시는 HEVC 코딩의 콘텍스트에서 비디오 디코더 (30) 를 기술한다. 그러나, 본 개시의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 3a 의 도시된 예는 단일 계층 코덱에 대한 것이다. 그러나, 도 3b 에 대해 더욱 기술되는 바와 같이, 비디오 디코더 (30) 의 일부 또는 전부가 다중 계층 코덱의 프로세싱을 위해 중복될 수도 있다.

[0112] 도 3a 의 예에서, 비디오 디코더 (30) 는 복수의 기능적 컴포넌트들을 포함한다. 비디오 디코더 (30) 의 기능적 컴포넌트들은 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 유닛 (156), 재구성 유닛 (158), 필터 유닛 (159), 및 디코딩된 화상 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. 일부 예들에서, 비디오 디코더 (30) 는 도 2a 의 비디오 인코더 (20) 에 대해 기술된 인코딩 패스 (pass) 에 대해 일반적으로 역인 디코딩 패스를 수행할 수도 있다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능적 컴포넌트들을 포함할 수도 있다.

[0113] 비디오 디코더 (30) 는 인코딩된 비디오 데이터를 포함하는 비트스트림을 수신할 수도 있다. 비트스트림은 복수의 선택스 엘리먼트들을 포함할 수도 있다. 비디오 디코더 (30) 가 비트스트림을 수신하는 경우, 엔트로피 디코딩 유닛 (150) 은 비트스트림에 대해 파싱 동작을 수행할 수도 있다. 비트스트림에 대해 파싱 동작을 수행하는 것의 결과로서, 엔트로피 디코딩 유닛 (150) 은 비트스트림으로부터 선택스 엘리먼트들을 추출할 수도 있다. 파싱 동작을 수행하는 것의 부분으로서, 엔트로피 디코딩 유닛 (150) 은 비트스트림 내의 엔트로피 인코딩된 선택스 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 선택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성하는 재구성 동작을 수행할 수도 있다.

[0114] 상술된 바와 같이, 비트스트림은 NAL 유닛들의 시리즈를 포함할 수도 있다. 비트스트림의 NAL 유닛들은 비

디오 파라미터 세트 NAL 유닛들, 시퀀스 파라미터 세트 NAL 유닛들, 화상 파라미터 세트 NAL 유닛들, SEI NAL 유닛들 등을 포함할 수도 있다. 비트스트림에 대해 파싱 동작을 수행하는 것의 부분으로서, 엔트로피 디코딩 유닛 (150)은 시퀀스 파라미터 세트 NAL 유닛들로부터의 시퀀스 파라미터 세트들, 화상 파라미터 세트 NAL 유닛들로부터의 화상 파라미터 세트들, SEI NAL 유닛들로부터의 SEI 데이터 등을 추출하고 엔트로피 디코딩하는 파싱 동작들을 수행할 수도 있다.

[0115] 또, 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림에 대해 파싱 동작을 수행하는 것의 부분으로서, 엔트로피 디코딩 유닛 (150)은 코딩된 슬라이스 NAL 유닛들로부터의 코딩된 슬라이스들을 추출하고 엔트로피 디코딩하는 파싱 동작들을 수행할 수도 있다. 코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 속하는 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더 내의 신택스 엘리먼트들은 슬라이스를 포함하는 화상과 연관된 화상 파라미터 세트들을 식별하는 신택스 엘리먼트를 포함할 수도 있다. 엔트로피 디코딩 유닛 (150)은 슬라이스 헤더를 복원하기 위해 코딩된 슬라이스 헤더 내의 신택스 엘리먼트들에 대해, CABAC 디코딩 동작들과 같은 엔트로피 디코딩 동작들을 수행할 수도 있다.

[0116] 코딩된 슬라이스 NAL 유닛들로부터 슬라이스를 추출하는 것의 부분으로서, 엔트로피 디코딩 유닛 (150)은 슬라이스 데이터 내의 코딩된 CU 들로부터 신택스 엘리먼트들을 추출하는 파싱 동작들을 수행할 수도 있다. 추출된 신택스 엘리먼트들은 변환 계수 블록들과 연관된 신택스 엘리먼트들을 포함할 수도 있다. 엔트로피 디코딩 유닛 (150)은 그 후 신택스 엘리먼트들 중 일부에 대해 CABAC 디코딩 동작들을 수행할 수도 있다.

[0117] 엔트로피 디코딩 유닛 (150)이 난-파티셔닝된 CU에 대해 파싱 동작을 수행한 후, 비디오 디코더 (30)는 난-파티셔닝된 CU에 대해 재구성 동작을 수행할 수도 있다. 난-파티셔닝된 CU에 대해 재구성 동작을 수행하기 위해, 비디오 디코더 (30)는 CU의 각 TU에 대해 재구성 동작을 수행할 수도 있다. CU의 각 TU에 대해 재구성 동작을 수행함으로써, 비디오 디코더 (30)는 CU와 연관된 레지듀얼 비디오 블록을 재구성할 수도 있다.

[0118] TU에 대해 재구성 동작을 수행하는 것의 부분으로서, 역양자화 유닛 (154)은 TU와 연관된 변환 계수 블록을 역양자화, 예를 들어 탈양자화할 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 양자화의 정도 및, 마찬가지로, 역양자화 유닛 (154)이 적용할 역양자화의 정도를 결정하기 위해 변환 계수 블록의 CU에 대해 비디오 인코더 (20)에 의해 계산된 양자화 파라미터 (QP)를 사용할 수도 있다.

[0119] 역양자화 유닛 (154)이 변환 계수 블록을 역양자화한 후, 역변환 유닛 (156)은 변환 계수 블록과 연관된 TU에 대한 레지듀얼 비디오 블록을 생성할 수도 있다. 역변환 유닛 (156)은 TU에 대한 레지듀얼 비디오 블록을 생성하기 위해 변환 계수 블록에 역변환을 적용할 수도 있다. 예를 들어, 역변환 유닛 (156)은 변환 계수 블록에 역 DCT, 역 정수 변화, 역 KLT (Karhunen-Loeve) 변환, 역 회전변환, 역 방향성 변화, 또는 다른 역변환을 적용할 수도 있다. 일부 예들에서, 역변환 유닛 (156)은 비디오 인코더 (20)로부터의 시그널링에 기초하여 변환 계수 블록에 적용할 역변환을 결정할 수도 있다. 그러한 예들에서, 역변환 유닛 (156)은 변환 계수 블록과 연관된 트리블록에 대한 쿼드트리의 루트 노드에서 시그널링된 변환에 기초하여 역변환을 결정할 수도 있다. 다른 예들에서, 역변환 유닛 (156)은 블록 사이즈, 코딩 모드 등과 같은 하나 이상의 코딩 특징들로부터 역변환을 추론할 수도 있다. 일부 예들에서, 역변환 유닛 (156)은 캐스캐이딩된 역변환을 적용할 수도 있다.

[0120] 일부 예들에서, 모션 보상 유닛 (162)은 보간 필터들에 기초하여 보간을 수행함으로써 PU의 예측된 비디오 블록을 정제할 수도 있다. 서브 샘플 정밀도를 갖는 모션 보상에 사용될 보간 필터들에 대한 식별자들이 신택스 엘리먼트들에 포함될 수도 있다. 모션 보상 유닛 (162)은 참조 블록의 서브 정수 샘플들에 대해 보간된 값들을 계산하기 위해 PU의 예측된 비디오 블록의 생성 동안 비디오 인코더 (20)에 의해 사용된 동일한 보간 필터들을 사용할 수도 있다. 모션 보상 유닛 (162)은 수신된 신택스 정보에 따라 비디오 인코더 (20)에 의해 사용된 보간 필터들을 결정하고 예측된 비디오 블록을 생성하기 위해 그 보간 필터들을 사용할 수도 있다.

[0121] 도 4를 참조하여 이하에 더욱 논의되는 바와 같이, 예측 프로세싱 유닛 (152)은 도 4에 도시된 방법들을 수행함으로써 PU (또는 임의의 다른 참조 계층 및/또는 항상 계층 블록 또는 비디오 유닛들)을 코딩 (예를 들어, 인코딩 또는 디코딩)할 수도 있다. 예를 들어, 모션 보상 유닛 (162), 인트라 예측 유닛 (164), 또는 인트라

-계층 예측 유닛 (166) 은 함께 또는 별개로 도 4 에 도시된 방법들을 수행하도록 구성될 수도 있다.

[0122] PU 가 인트라 예측을 사용하여 인코딩되는 경우, 인트라 예측 유닛 (164) 은 PU 에 대한 예측된 비디오 블록을 생성하기 위해 인트라 예측을 수행할 수도 있다. 예를 들어, 인트라 예측 유닛 (164) 은 비트스트림 내의 선택스 엘리먼트들에 기초하여 PU 에 대한 인트라 예측 모드를 결정할 수도 있다. 비트스트림은 PU 의 인트라 예측 모드를 결정하기 위해 인트라 예측 유닛 (164) 이 사용할 수도 있는 선택스 엘리먼트들을 포함할 수도 있다.

[0123] 일부 예들에서, 선택스 엘리먼트들은 현재의 PU 의 인트라 예측 모드를 결정하기 위해 인트라 예측 유닛 (164) 이 다른 PU 의 인트라 예측 모드를 사용해야 한다는 것을 나타낼 수도 있다. 예를 들어, 현재의 PU 의 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일하다는 것은 개연성이 있을 수도 있다. 즉, 이웃하는 PU 의 인트라 예측 모드는 현재의 PU 에 대한 가장 개연성 있는 모드일 수도 있다. 이리하여, 이러한 예에서, 비트스트림은 PU 의 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일하다는 것을 나타내는 소형 선택스 엘리먼트를 포함할 수도 있다. 인트라 예측 유닛 (164) 은 그 후 공간적으로 이웃하는 PU 들의 비디오 블록들에 기초하여 PU 에 대한 예측 데이터 (예를 들어, 예측된 샘플들) 를 생성하기 위해 그 인트라 예측 모드를 사용할 수도 있다.

[0124] 상술된 바와 같이, 비디오 디코더 (30) 는 또한 인터-계층 예측 유닛 (166) 을 포함할 수도 있다. 인터-계층 예측 유닛 (166) 은 SVC 에서 이용가능한 하나 이상의 상이한 계층들 (예를 들어, 베이스 또는 참조 계층) 을 사용하여 현재의 블록 (예를 들어, EL 에서의 현재의 블록) 을 예측하도록 구성된다. 그러한 예측은 인터-계층 예측으로서 지칭될 수도 있다. 인터-계층 예측 유닛 (166) 은 인터-계층 리던던시를 감소시키는 예측 방법들을 이용함으로써, 코딩 효율을 향상시키고 계산 자원 조건들을 감소시킨다. 인터-계층 예측의 일부 예들은 인터-계층 인트라 예측, 인터-계층 모션 예측, 및 인터-계층 레지듀얼 예측을 포함한다. 인터-계층 인트라 예측은 항상 계층에서의 현재의 블록을 예측하기 위해 베이스 계층에서의 동일 장소에 배치된 블록들의 재구성을 사용한다. 인터-계층 모션 예측은 항상 계층에서의 모션을 예측하기 위해 베이스 계층의 모션 정보를 사용한다. 인터-계층 레지듀얼 예측은 항상 계층의 레지듀를 예측하기 위해 베이스 계층의 레지듀를 사용한다. 인터-계층 예측 스킴들 각각은 이하에 더욱 상세히 논의된다.

[0125] 재구성 유닛 (158) 은 CU 의 TU 들과 연관된 레지듀얼 비디오 블록들 및 CU 의 PU 들의 예측된 비디오 블록들, 적용가능한 대로 예를 들어 인트라 예측 데이터 또는 인터 예측 데이터를 사용하여, CU 의 비디오 블록을 재구성할 수도 있다. 따라서, 비디오 디코더 (30) 는 비트스트림 내의 선택스 엘리먼트들에 기초하여 예측된 비디오 블록 및 레지듀얼 비디오 블록을 생성할 수도 있고 그 예측된 비디오 블록 및 레지듀얼 비디오 블록에 기초하여 비디오 블록을 생성할 수도 있다.

[0126] 재구성 유닛 (158) 이 CU 의 비디오 블록을 재구성한 후, 필터 유닛 (159) 은 CU 와 연관된 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행할 수도 있다. 필터 유닛 (159) 이 CU 와 연관된 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행한 후, 비디오 디코더 (30) 는 디코딩된 화상 버퍼 (160) 내에 CU 의 비디오 블록을 저장할 수도 있다. 디코딩된 화상 버퍼 (160) 는 후속하는 모션 보상, 인트라 예측, 및 도 1a 또는 도 1b 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상의 제시를 위해 참조 화상들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 디코딩된 화상 버퍼 (160) 내의 비디오 블록들에 기초하여 다른 CU 들의 PU 들에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다.

[0127] 다중 계층 디코더

[0128] 도 3b 는 본 개시에 기술된 양태들에 따른 기법들을 구현할 수도 있는 다중 계층 비디오 디코더 (33) 의 예를 도시하는 블록도이다. 비디오 디코더 (33) 는 예를 들어 SHVC 및 멀티뷰 코딩에 대한 다중 계층 비디오 프레임들을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 디코더 (33) 는 본 개시의 기법들의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0129] 비디오 디코더 (33) 는 비디오 디코더 (30A) 및 비디오 디코더 (30B) 를 포함하고, 이들 각각은 비디오 디코더 (30) 로서 구성될 수도 있고 비디오 디코더 (30) 에 대한 상술된 기능들을 수행할 수도 있다. 또한, 참조 번호들이 재사용에 의해 나타낸 바와 같이, 비디오 디코더들 (30A 및 30B) 은 비디오 디코더 (30) 로서 시스템들 및 서브시스템들의 적어도 일부를 포함할 수도 있다. 비디오 디코더 (33) 가 2 개의 비디오 디코더들 (30A 및 30B) 을 포함하는 것으로서 도시되지만, 비디오 디코더 (31) 는 그렇게 제한되지 않고 임의의 수의 비디오 디코더 (30) 계층들을 포함할 수도 있다. 일부 실시형태들에서, 비디오 디코더 (33) 는 액세스 유닛

내의 각 화상 또는 프레임에 대해 비디오 디코더 (30) 를 포함할 수도 있다. 예를 들어, 5 개의 화상들을 포함하는 액세스 유닛은 5 개의 디코더 계층들을 포함하는 비디오 디코더에 의해 프로세싱되거나 디코딩될 수도 있다. 일부 실시형태들에서, 비디오 디코더 (33) 는 액세스 유닛 내의 프레임들보다 더 많은 디코더 계층들을 포함할 수도 있다. 일부 그러한 경우들에서, 비디오 디코더 계층들의 일부는 일부 액세스 유닛들을 프로세싱할 때 비활성일 수도 있다.

[0130]

비디오 디코더들 (30A 및 30B) 에 더하여, 비디오 디코더 (33) 는 업샘플링 유닛 (92) 을 포함할 수도 있다. 일부 실시형태들에서, 업샘플링 유닛 (92) 은 프레임 또는 액세스 유닛에 대한 참조 화상 리스트에 추가될 향상된 계층을 생성하기 위해 수신된 비디오 프레임의 베이스 계층을 업샘플링할 수도 있다. 이러한 향상된 계층은 디코딩된 화상 버퍼 (160) 에 저장될 수 있다. 일부 실시형태들에서, 업샘플링 유닛 (92) 은 도 2a 의 리샘플링 유닛 (90) 에 대해 기술된 실시형태들의 일부 또는 전부를 포함할 수 있다. 일부 실시형태들에서, 업샘플링 유닛 (92) 은 계층을 업샘플링하고 하나 이상의 슬라이스들을 재조직화, 재정의, 변경 또는 조정하여 슬라이스 경계 규칙들 및/또는 래스터 스캔 규칙들의 세트에 부응하도록 구성된다. 일부 경우들에서, 업샘플링 유닛 (92) 은 수신된 비디오 프레임의 계층을 업샘플링 및/또는 다운샘플링하도록 구성된 리샘플링 유닛일 수도 있다.

[0131]

업샘플링 유닛 (92) 은 더 낮은 계층 디코더 (예를 들어, 비디오 디코더 (30A)) 의 디코딩된 화상 버퍼 (160) 로부터 화상 또는 프레임 (또는 그 화상과 연관된 화상 정보) 을 수신하고 그 화상 (또는 그 수신된 화상 정보) 을 업샘플링하도록 구성될 수도 있다. 이러한 업샘플링된 화상은 그 후 더 낮은 계층 디코더와 동일한 액세스 유닛 내의 화상을 디코딩하도록 구성된 더 높은 계층 디코더 (예를 들어, 비디오 디코더 (30B)) 의 예측 프로세싱 유닛 (152) 에 제공될 수도 있다. 일부 경우들에서, 더 높은 계층 디코더는 더 낮은 계층 디코더로부터 제거된 하나의 계층이다. 다른 경우들에서, 도 3b 의 계층 0 디코더와 계층 1 디코더 사이에 하나 이상의 더 높은 계층 디코더들이 존재할 수도 있다.

[0132]

일부 경우들에서, 업샘플링 유닛 (92) 은 생략되거나 우회될 수도 있다. 그러한 경우들에서, 비디오 디코더 (30A) 의 디코딩된 화상 버퍼 (160) 로부터의 화상은 비디오 디코더 (30B) 의 예측 프로세싱 유닛 (152) 으로 직접 또는 적어도 업샘플링 유닛 (92) 으로 제공되지 않고 제공될 수도 있다. 예를 들어, 비디오 디코더 (30B) 로 제공된 비디오 데이터 및 비디오 디코더 (30A) 의 디코딩된 화상 버퍼 (160) 로부터의 참조 화상이 동일한 사이즈 또는 해상도인 경우, 참조 화상은 업샘플링 없이 비디오 디코더 (30B) 에 제공될 수도 있다. 또한, 일부 실시형태들에서, 업샘플링 유닛 (92) 은 비디오 디코더 (30A) 의 디코딩된 화상 버퍼 (160) 로부터 수신된 참조 화상을 업샘플링 또는 다운샘플링하도록 구성된 리샘플링 유닛 (90) 일 수도 있다.

[0133]

도 3b 에 도시된 바와 같이, 비디오 디코더 (33) 는 디멀티플렉서 (99), 또는 DEMUX 를 더 포함할 수도 있다. 디멀티플렉서 (99) 는 상이한 비디오 디코더들 (30A 및 30B) 로 제공되는 디멀티플렉서 (99) 에 의해 출력되는 각 비트스트림을 갖는 다수의 비트스트림들로 인코딩된 비디오 비트스트림을 분할할 수 있다. 다수의 비트스트림들은 비트스트림을 수신함으로써 생성될 수도 있고, 비디오 디코더들 (30A 및 30B) 각각은 주어진 시간에 비트스트림의 일부를 수신한다. 일부 경우들에서 디멀티플렉서 (99) 에서 수신된 비트스트림으로부터의 비트들은 비디오 디코더들 (예를 들어, 도 3b 의 예에서 비디오 디코더들 (30A 및 30B)) 각각 사이에서 한번에 1 비트씩 교번될 수도 있지만, 많은 경우들에서, 비트스트림은 상이하게 분할된다. 예를 들어, 비트스트림은 어느 비디오 디코더가 한번에 하나의 블록씩 비트스트림을 수신하는지를 교번함으로써 분할될 수도 있다. 다른 예에서, 비트스트림은 비디오 디코더들 (30A 및 30B) 각각으로 블록들의 년-1:1 비율에 의해 분할될 수도 있다. 예를 들어, 2 개의 블록들이 비디오 디코더 (30A) 로 제공된 각 블록에 대해 비디오 디코더 (30B) 로 제공될 수도 있다. 일부 실시형태들에서, 디멀티플렉서 (99) 에 의한 비트스트림의 분할은 사전 프로그래밍될 수도 있다. 다른 실시형태들에서, 디멀티플렉서 (99) 는 목적지 모듈 (14) 을 포함하는 목적지 디바이스상의 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 서브스크립션 (유료 서브스크립션 대 무료 서브스크립션) 에 기초하여, 또는 비디오 디코더 (33) 에 의해 획득가능한 해상도를 결정하기 위한 임의의 다른 팩터에 기초하여 생성될 수도 있다.

[0134]

코딩 효율 대 드리프트

[0135]

상술된 바와 같이, BL 을 코딩하기 위해 사용되는 EL 의 임의의 부분이 미싱되는 경우 드리프트가 발생한다. 예를 들어, 디코더가 2 개의 계층들, BL 및 EL 을 포함하는 비트스트림을 프로세싱하며, 여기서 BL 은 EL 에

포함된 정보를 사용하여 코딩되고, 디코더가 비트스트림의 BL 부분만을 디코딩하기로 선택하는 경우, BL 을 코딩하기 위해 사용되는 정보가 더 이상 이용가능하지 않기 때문에 드리프트가 발생할 것이다.

[0136] 드리프트의 최소화

[0137] 하나의 구현에서, EL 화상들은 BL 내의 정보를 사용하여 코딩될 수도 있지만, BL 화상들은 EL 내의 정보를 사용하여 코딩되지 않을 수도 있다. 그러한 예에서, EL 의 일부가 상실되더라도, BL 이 EL 에 기초하여 코딩되지 않기 때문에 BL 의 디코딩은 영향을 받지 않는다.

[0138] 다른 구현에서, "중요 화상들" 이 비트스트림 전체에 걸쳐 지정되고, 그러한 중요 화상들은 BL 내의 정보만을 사용할 수 있다. 따라서, EL 의 일부가 상실되더라도, 적어도 이들 중요 화상들은 드리프트에 의해 영향을 받지 않는다. 이러한 구현에서, BL 화상들이 EL 화상들에 기초하여 코딩되는 것을 허용함으로써 코딩 효율이 향상될 수도 있지만, 리프레시 화상들로서도 지칭될 수도 있는 이들 중요 화상들을 가짐으로써, 드리프트의 역효과들이 상당히 감소될 수도 있다.

[0139] 현존하는 코딩 스킴들

[0140] 일부 구현들 (예를 들어, HEVC) 은 참조 화상들로서 더 높은 계층 디코딩된 화상들을 사용하여 더 낮은 계층들이 코딩되는 것을 허용하지 않을 수도 있다. 또한, 일부 구현들은 더 높은 계층 디코딩된 화상이 더 낮은 계층에서의 현재의 화상의 참조 화상이라는 것을 나타내기 위한 임의의 메커니즘을 갖지 않을 수도 있다. 그러한 구현들에서, 본 개시에 기술된 기법들은 드리프트와 연관된 역효과들을 최소화하면서 더 높은 계층 (예를 들어, EL) 에 기초하여 더 낮은 계층 (예를 들어, BL) 이 코딩되는 것을 허용하는 것으로부터 야기되는 코딩 효율 이득을 활용하도록 이용될 수도 있다.

[0141] 예시의 실시형태들

[0142] 본 개시에서, 더 높은 계층 디코딩된 화상들이 더 낮은 계층 화상들을 코딩하기 위한 참조 화상들로서 사용될 수도 있는지 여부에 대한 표시들을 시그널링하고 프로세싱하기 위한 여러 예시의 실시형태들이 기술된다. 하나 이상의 그러한 실시형태들은 현존하는 구현 (예를 들어, HEVC 확장들) 과 관련하여 기술될 수도 있다. 본 개시의 실시형태들은 서로 독립적으로 또는 결합하여 적용될 수 있고, 스케일러블 코딩, 깊이를 갖거나 갖지 않는 멀티-뷰 코딩, 및 HEVC 및 다른 비디오 코덱들의 다른 확장들에 적용가능하거나 확장될 수도 있다.

[0143] BL 및 EL 의 예가 일부 실시형태들을 기술하기 위해 사용되지만, 여기에 기술된 기법들은 RL 및 EL, BL 및 다수의 EL 들, RL 및 다수의 EL 들 등과 같은 계층들의 임의의 쌍 또는 그룹에 적용되거나 확장될 수도 있다.

[0144] 더 높은 계층 디코딩된 화상들을 사용하는 것에 대한 VPS 레벨 신호 표시

[0145] 하나의 실시형태에서, 비디오 파라미터 세트 (VPS) 에서 제공된 플래그 또는 선택스 엘리먼트는 더 높은 계층 디코딩된 화상들이 더 낮은 계층 화상들을 코딩하기 위한 참조 화상들로서 사용될 수도 있는지 여부를 나타낸다. 플래그 또는 선택스 엘리먼트가 VPS 에서 제공되기 때문에, 플래그 또는 선택스 엘리먼트에 의해 제공된 임의의 표시는 동일한 코딩된 비디오 시퀀스 (CVS) 내의 모든 계층들에 적용될 것이다. 그러한 플래그 또는 선택스 엘리먼트의 구현을 도시하는 예시의 선택스가 아래에 있다. 관련 부분들은 이탤릭체로 도시된다.

vps_extension() {	디스크립터
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
.....	
for(i = 0; i <= vps_max_layers_minus1 - 1; i++)	
enable_higher_layer_ref_pic_pred[i]	u(1)
.....	
}	

표 1. enable_higher_layer_ref_pic_pred 를 설명하는 예시의 선택스

[0147]

예시의 시맨틱스 #1

[0148]

예를 들어, 다음의 시맨틱스가 플래그 또는 선택스 엘리먼트를 정의하기 위해 사용될 수도 있다: 0 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i]` 보다 큰 `nuh_layer_id` 를 갖는 디코딩된 화상들은 `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 화상들에 대한 참조로서 사용되지 않는다는 것을 특징한다. 1 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i]` 보다 큰 `nuh_layer_id` 를 갖는 디코딩된 화상들은, 이용가능한 경우, `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 및 0 보다 큰 시간적 ID 를 갖는 화상들에 대한 참조로서 사용될 수도 있다는 것을 특징한다. 존재하지 않는 경우, `enable_higher_layer_ref_pic_pred[i]` 는 0 인 것으로 추론된다.

[0149]

이러한 예에서, 임의의 더 높은 계층은 참조 계층일 수도 있고, 더 높은 계층 예측이 시간적 ID 가 0 보다 큰 시간적 계층들에 대해 이용가능하다. 여기서, 디코딩된 화상들의 이용가능성은 현재의 화상과 동일한 액세스 유닛 내에 임의의 디코딩된 화상들이 존재하는지 여부에 의해 결정될 수도 있다. 예를 들어, 1 의 `enable_higher_layer_ref_pic_pred[i]` 값은 더 높은 계층 디코딩된 화상들은, 존재하는 경우, 현재의 계층 내의 현재의 화상을 코딩하기 위해 사용될 수도 있다. 다른 실시형태에서, 이용가능성은 현재의 화상의 액세스 유닛에 제한되지 않고, 다른 시간적으로 이웃하는 액세스 유닛들을 포함할 수도 있다.

[0150]

예시의 시맨틱스 #2

[0151]

다른 예에서, 다음의 시맨틱스가 플래그 또는 선택스 엘리먼트를 정의하기 위해 사용될 수도 있다: 0 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i]` 보다 큰 `nuh_layer_id` 를 갖는 디코딩된 화상들은 `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 화상들에 대한 참조로서 사용되지 않는다는 것을 특징한다. 1 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i]` 보다 큰 `nuh_layer_id` 를 갖는 디코딩된 화상들은, 이용가능한 경우, `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 화상들에 대한 참조로서 사용될 수도 있다는 것을 특징한다. 존재하지 않는 경우, `enable_higher_layer_ref_pic_pred[i]` 는 0 과 동일한 것으로 추론된다.

[0152]

이러한 예에서, 임의의 더 높은 계층은 참조 계층일 수도 있고, 더 높은 계층 예측이 시간적 ID 가 0 보다 큰 그러한 계층들에 대해서 뿐만 아니라, 모든 시간적 계층들에 대해 이용가능하다.

[0153]

예시의 시맨틱스 #3

[0154]

또 다른 예에서, 다음의 시맨틱스가 플래그 또는 선택스 엘리먼트를 정의하기 위해 사용될 수도 있다: 0 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i]` 보다 큰 `nuh_layer_id` 를 갖는 디코딩된 화상들은 `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 화상들에 대한 참조로서 사용되지 않는다는 것을 특징한다. 1 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i+1]` 과 동일한 `nuh_layer_id` 를 갖는 디코딩된 화상들은, 이용가능한 경우, `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 및 0 보다 큰 시간적 ID 를 갖는 화상들에 대한 참조로서 사용될 수도 있다는 것을 특징한다. 존재하지 않는 경우, `enable_higher_layer_ref_pic_pred[i]` 는 0 과 동일한 것으로 추론된다.

[0155]

이러한 예에서, 바로 (immediately) 더 높은 계층은 참조 계층일 수도 있고, 더 높은 계층 예측이 시간적 ID 가 0 보다 큰 시간적 계층들에 대해 이용가능하다.

[0156]

[0157]

예시의 시맨틱스 #4

[0158]

또 다른 예에서, 다음의 시맨틱스가 플래그 또는 선택스 엘리먼트를 정의하기 위해 사용될 수도 있다: 0 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i]` 보다 큰 `nuh_layer_id` 를 갖는 디코딩된 화상들은 `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 화상들에 대한 참조로서 사용되지 않는다는 것을 특징한다. 1 과 동일한 `enable_higher_layer_ref_pic_pred[i]` 는 CVS 내에서 `layer_id_in_nuh[i+1]` 과 동일한 `nuh_layer_id` 를 갖는 디코딩된 화상들은, 이용가능한 경우, `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 화상들에 대한 참조로서 사용될 수도 있다는 것을 특징한다. 존재하지 않는 경우, `enable_higher_layer_ref_pic_pred[i]` 는 0 과 동일한 것으로 추론된다.

[0159]

이러한 예에서, 바로 더 높은 계층은 참조 계층일 수도 있고, 더 높은 계층 예측이 시간적 ID 가 0 보다 큰 그

러한 계층들에 대해서 뿐만 아니라, 모든 시간적 계층들에 대해 이용가능하다.

[0160] **플래그 또는 선택스 엘리먼트의 로케이션**

[0161] 상술된 `enable_higher_layer_ref_pic_pred[i]` 플래그 또는 선택스 엘리먼트는 VPS, SPS, PPS, 슬라이스 헤더, 및 그것의 확장들에서 시그널링될 수도 있다. 그것은 또한 보충 강화 정보 (SEI) 메시지 또는 비디오 가용성 정보 (VUI) 메시지로써 시그널링될 수도 있다.

[0162] **예시의 플로우차트**

[0163] 도 4 는 본 개시의 실시형태에 따라, 비디오 정보를 코딩하는 방법 (400) 을 설명하는 플로우차트를 도시한다. 도 4 에 도시된 단계들은 인코더 (예를 들어, 도 2a 또는 도 2b 에 도시된 바와 같은 비디오 인코더), 디코더 (예를 들어, 도 3a 또는 도 3b 에 도시된 바와 같은 비디오 디코더), 또는 임의의 다른 컴포넌트에 의해 수행될 수도 있다. 편의상, 방법 (400) 은 인코더, 디코더, 또는 다른 컴포넌트일 수도 있는 코더에 의해 수행되는 것으로 기술된다.

[0164] 방법 (400) 은 블록 (401) 에서 시작한다. 블록 (405) 에서, 코더는 더 높은 계층 디코딩된 화상들이 현재의 계층 화상들을 코딩하기 위해 사용되는 것이 허용되는지 여부를 결정한다. 블록 (410) 에서, 코더는 현재의 계층 내의 현재 계층 화상이 더 높은 계층 내의 대응하는 더 높은 계층 화상을 가지는지 여부를 결정한다. 블록 (415) 에서, 코더는 현재 계층 화상의 시간적 ID 가 0 보다 큰 지 여부를 결정한다. 예를 들어, 더 높은 계층 화상들의 사용을 0 보다 큰 시간적 ID 를 갖는 현재 계층 화상들로 제한하는 것은 드리프트의 역효과들이 감소되도록 현재의 계층 내에 적어도 일부 중요 화상들이 존재할 것을 보장한다. 더 높은 계층 디코딩된 화상들이 현재 계층 화상들을 코딩하기 위해 사용되는 것이 허용된다는 것, 현재 계층 내의 현재 계층 화상이 더 높은 계층 내의 대응하는 더 높은 계층 화상을 갖는다는 것, 및 현재 계층 화상의 시간적 ID 가 0 보다 크다는 것을 결정하는 것에 응답하여, 코더는 대응하는 더 높은 계층 화상에 기초하여 현재 계층 화상을 코딩한다. 방법 (400) 은 425 에서 종료한다.

[0165] 상술된 바와 같이, 도 2a 의 비디오 인코더 (20), 도 2b 의 비디오 인코더 (23), 도 3a 의 비디오 디코더 (30), 또는 도 3b 의 비디오 디코더 (33) 의 하나 이상의 컴포넌트들 (예를 들어, 인터-계층 예측 유닛 (128) 및/또는 인터-계층 예측 유닛 (166)) 은, 더 높은 계층 디코딩된 화상들이 현재 계층 화상들을 코딩하기 위해 사용되는 것이 허용되는지 여부를 결정하는 것, 현재 계층 내의 현재 화상이 더 높은 계층 내의 대응하는 더 높은 계층 화상을 가지는지 여부를 결정하는 것, 현재 화상의 시간적 ID 가 0 보다 큰지 여부를 결정하는 것, 및 대응하는 더 높은 계층 화상에 기초하여 현재 화상을 코딩하는 것과 같은, 본 개시에서 논의된 임의의 기법들을 구현하기 위해 사용될 수도 있다.

[0166] 방법 (400) 에서, 도 4 에 도시된 하나 이상의 블록들은 제거될 (예를 들어, 수행되지 않을) 수도 있고, 및/또는 방법이 수행되는 순서가 바뀔 수도 있다. 예를 들어, 블록 (415) 가 도 4 에 도시되지만, 그것은 현재 계층 화상의 시간적 ID 가 0 보다 크다는 제한을 제거하기 위해 제거될 수도 있다. 다른 예로서, 블록 (420) 이 도 4 에 도시되지만, 실제로 현재 계층 화상을 코딩하는 것은 방법 (400) 의 부분일 필요가 없고 따라서 방법 (400) 으로부터 생략될 필요가 있다. 따라서, 본 개시의 실시형태들은 도 4 에 도시된 예에 또는 그 예에 의해 제한되지 않고, 다른 변형들이 본 개시의 사상으로부터 이탈하지 않고 구현될 수도 있다.

[0167] **더 높은 계층 디코딩된 화상들의 사용에 대한 무 (no) 명시적 시그널링**

[0168] 이러한 실시형태에서, 각 화상에 대해, 화상이 더 높은 계층 참조 화상을 사용하는지 여부가 이하에 기술된 프로세스를 사용하여 결정된다.

[0169] 현재 화상이 예측을 위해 더 높은 계층 디코딩된 화상을 사용할 수도 있는지 여부를 결정하기 위해, 예시의 변수 `enableHigherLayerRefpicforCurrPicFlag` 가 도입된다. `i` 와 동일한 계층 인덱스를 갖는 현재의 계층 내의 현재의 화상에 대한 변수 `enableHigherLayerRefpicforCurrPicFlag` 는 다음과 같이 정의될 수도 있다: 0 과 동일한 `enableHigherLayerRefpicforCurrPicFlag` 는 `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 현재 화상에 대해, `layer_id_in_nuh[i]` 보다 큰 `nuh_layer_id` 를 갖는 디코딩된 화상들은 현재 화상에 대해 참조로서 사용되지 않는다는 것을 특징한다. 1 과 동일한 `enableHigherLayerRefpicforCurrPicFlag` 는 `layer_id_in_nuh[i]` 와 동일한 `nuh_layer_id` 를 갖는 현재 화상에 대해, `layer_id_in_nuh[i+1]` 과 동일한 `nuh_layer_id` 를 갖는 디코딩된 화상들은, 이용가능한 경우, 현재 화상에 대한 참조로서 사용될 수도 있다는 것을 특징한다.

[0170] i 의 계층 인덱스를 갖는 현재 계층 내의 현재 화상의 경우, 다음의 조건들 모두가 만족된다면 변수 enableHigherLayerRefpicforCurrPicFlag 의 값은 1 로 세팅된다:

[0171] a) 현재 화상의 시간적 ID 가 0 과 동일하다;

[0172] b) scalability_mask[i] 가 1 과 동일하여, SNR 또는 공간 확장성을 나타낸다;

[0173] c) (예를 들어, 상술된) VPS 플래그 enable_higher_layer_ref_pic_pred[i] 가 1 과 동일하여, 더 높은 계층 예측이 허용된다는 것을 나타낸다; 및

[0174] d) layer_id_in_nuh[i+1] 과 동일한 nuh_layer_id 를 갖는 대응하는 디코딩된 화상들이 이용가능하다 (예를 들어, 현재 화상에 대응하는 동일장소에 배치된 화상이 동일한 액세스 유닛 내에 존재한다).

[0175] 이들 조건들 모두가 만족되는 경우, 변수 enableHigherLayerRefpicforCurrPicFlag 는 1 로 세팅되어 더 높은 계층 디코딩된 화상들이 현재 화상을 코딩하기 위해 사용될 수도 있다는 것을 나타낸다. 이들 조건들 중 하나 이상이 만족되지 않으면, 변수 enableHigherLayerRefpicforCurrPicFlag 가 제로로 세팅되어 더 높은 계층 디코딩된 화상들이 현재 화상을 코딩하기 위해 사용되지 않을 수도 있다는 것을 나타낸다.

[0176] 더 높은 계층 디코딩된 화상들의 사용에 대한 명시적 시그널링

[0177] 대안적인 실시형태에서, 플래그, enableHigherLayerRefpicforCurrPicFlag 는 현재 계층 내의 현재 화상이 참조로서 더 높은 계층 참조 화상들을 사용하는지 여부를 특정하기 위해 명시적으로 시그널링될 수도 있다. enableHigherLayerRefpicforCurrPicFlag 플래그는 다음과 같이 정의될 수도 있다: 0 과 동일한 enableHigherLayerRefpicforCurrPicFlag 는 layer_id_in_nuh[i] 와 동일한 nuh_layer_id 를 갖는 현재 화상에 대해, layer_id_in_nuh[i] 보다 큰 nuh_layer_id 를 갖는 디코딩된 화상들은 현재 화상에 대해 참조로서 사용되지 않는다는 것을 특정한다. 1 과 동일한 enableHigherLayerRefpicforCurrPicFlag 는 layer_id_in_nuh[i] 와 동일한 nuh_layer_id 를 갖는 현재 화상에 대해, layer_id_in_nuh[i+1] 과 동일한 nuh_layer_id 를 갖는 디코딩된 화상들은, 이용가능한 경우, 현재 화상에 대한 참조로서 사용된다는 것을 특정한다. 예를 들어, enableHigherLayerRefpicforCurrPicFlag 플래그는 PPS, 슬라이스 헤더, 또는 그것의 확장들에서 시그널링될 수도 있다. 그것은 또한 SEI 메시지 또는 VUI 메시지로써 시그널링될 수도 있다.

[0178] 다른 실시형태에서, 플래그, enableHigherLayerRefpicforCurrPicFlag 는 현재 계층 내의 현재 화상이 참조로서 더 높은 계층 참조 화상들을 사용하는지 여부를 특정하기 위해 명시적으로 시그널링된다. enableHigherLayerRefpicforCurrPicFlag 플래그는 다음과 같이 정의될 수도 있다: 0 과 동일한 enableHigherLayerRefpicforCurrPicFlag 는 layer_id_in_nuh[i] 와 동일한 nuh_layer_id 를 갖는 현재 화상에 대해, layer_id_in_nuh[i] 보다 큰 nuh_layer_id 를 갖는 디코딩된 화상들은 현재 화상에 대해 참조로서 사용되지 않는다는 것을 특정한다. 1 과 동일한 enableHigherLayerRefpicforCurrPicFlag 는 layer_id_in_nuh[i] 와 동일한 nuh_layer_id 를 갖는 현재 화상에 대해, layer_id_in_nuh[i+k] 과 동일한 nuh_layer_id 를 갖는 디코딩된 화상들은, 이용가능한 경우, 현재 화상에 대한 참조로서 사용된다는 것을 특정한다.

[0179] 이러한 실시형태에서, 현재 화상의 바로 위에 있는 더 높은 계층 (예를 들어, 이전의 예에서 도시된 바와 같은 layer_id_in_nuh[i+1]) 의 참조 화상들을 사용하는 것 대신에, 현재 화상 위의 k 번째 더 높은 계층의 참조 화상들이 현재 화상을 코딩하기 위해 사용된다 (이러한 예에서 도시된 바와 같은 layer_id_in_nuh[i+k]). 예를 들어, k 의 값은 명시적으로 시그널링되거나 VPS 에서 시그널링된 직접 의존성 플래그로부터 추론될 수도 있다.

[0180] 더 높은 계층 참조 화상의 사용을 나타내는 플래그의 해석

[0181] 하나의 실시형태에서, enableHigherLayerRefpicforCurrPicFlag 의 값은 0 보다 큰 시간적 ID 를 갖는 동일한 CVS 내의 동일한 계층의 모든 화상들에 대해 동일한 값을 갖는다. 그러한 제한은 임의의 컨퍼밍 (conforming) 비트스트림이 그러한 제한을 만족하도록 비트스트림 컨퍼먼스 (conformance) 제약으로서 구현될 수도 있다.

[0182] 다른 실시형태에서, enableHigherLayerRefpicforCurrPicFlag 의 값은 0 과 동일한 시간적 ID 를 갖는 동일한 CVS 내의 동일한 계층의 모든 화상들에 대해 동일한 값을 갖는다. 그러한 제한은 임의의 컨퍼밍 비트스트림이 그러한 제한을 만족하도록 비트스트림 컨퍼먼스 제약으로서 구현될 수도 있다.

[0183] RPS 및 화상 마킹을 위한 도출 프로세스의 예시의 구현

- [0184] 하나의 실시형태에서, RPS 및 화상 마킹을 위한 도출 프로세스는 이하에 설명된 바와 같이 구현될 수도 있다. 예시의 코딩 스킴 (예를 들어, HEVC) 에 대한 임의의 변경들은 이탤릭체로 강조되며, 삭제들은 삭제선에 의해 표시된다. 예시의 구현에서 참조되는 HEVC 스케일러블 확장의 드래프트 사양의 섹션 F.8.1.3 이 또한 이하에 재생된다.
- [0185] **섹션 F.8.1.3 계층 내의 디코딩 순서에서 첫번째인 화상들에 대해 이용가능하지 않은 참조 화상들의 생성**
- [0186] 이러한 프로세서는 `FirstPicInLayerDecodedFlag[layerId]` 가 0 과 동일할 때 `layerId` 와 동일한 `nuh_layer_id` 를 갖는 화상에 대해 호출된다.
- [0187] 주의 - 크로스-계층 랜덤 액세스 스킵된 (CL-RAS) 화상은 0 보다 큰 `nuh_layer_id` 를 갖는 코딩된 화상의 디코딩을 시작하기 위한 디코딩 프로세스가 호출될 때 `LayerInitializedFlag[layerId]` 가 0 과 동일하도록 `layerId` 와 동일한 `nuh_layer_id` 를 갖는 화상이다. CL-RAS 화상들에 대한 디코딩 프로세스의 전체 사양은 그러한 CL-RAS 화상들의 허용된 선택스 콘텐츠에 대한 제약들을 특정할 목적으로만 포함된다. 디코딩 프로세스 동안, 임의의 CL-RAS 화상들은, 이들 화상들이 출력을 위해 특정되지 않고 출력을 위해 특정되는 임의의 다른 화상들의 디코딩 프로세스에 영향을 주지 않기 때문에, 무시될 수도 있다. 그러나, 부록 C 에서 특정된 바와 같은 HRD 동작들에서, CL-RAS 화상들은 CPB 도달 및 제거 시간들의 도출에서 고려될 필요가 있을 수도 있다.
- [0188] 이러한 프로세스가 호출될 때, 다음이 적용된다:
- [0189] - `i` 가 "no-reference picture" 와 동일한 0 내지 `NumPocStCurrBefore - 1` 의 범위에 있는 상태에서, 각각의 `RefPicSetStCurrBefore[i]` 에 대해, 화상이 서브 조항 8.3.3.2 에서 특정된 바와 같이 생성되고, 다음이 적용된다:
- [0190] - 생성된 화상에 대한 `PicOrderCntVal` 의 값이 `PocStCurrBefore[i]` 와 동일하게 세팅된다.
- [0191] - 생성된 화상에 대한 `PicOutputFlag` 의 값이 0 과 동일하게 세팅된다.
- [0192] - 생성된 화상은 "단기 참조를 위해 사용되는" 것으로서 마킹된다.
- [0193] - `RefPicSetStCurrBefore[i]` 는 생성된 참조 화상이도록 세팅된다.
- [0194] - 생성된 화상에 대한 `nuh_layer_id` 의 값은 `nuh_layer_id` 와 동일하게 세팅된다.
- [0195] - `i` 가 "no-reference picture" 와 동일한 0 내지 `NumPocStCurrAfter - 1` 의 범위에 있는 상태에서, 각각의 `RefPicSetStCurrAfter[i]` 에 대해, 화상이 서브 조항 8.3.3.2 에서 특정된 바와 같이 생성되고, 다음이 적용된다:
- [0196] - 생성된 화상에 대한 `PicOrderCntVal` 의 값이 `PocStCurrAfter[i]` 와 동일하게 세팅된다.
- [0197] - 생성된 화상에 대한 `PicOutputFlag` 의 값이 0 과 동일하게 세팅된다.
- [0198] - 생성된 화상은 "단기 참조를 위해 사용되는" 것으로서 마킹된다.
- [0199] - `RefPicSetStCurrAfter[i]` 는 생성된 참조 화상이도록 세팅된다.
- [0200] - 생성된 화상에 대한 `nuh_layer_id` 의 값은 `nuh_layer_id` 와 동일하게 세팅된다.
- [0201] - `i` 가 "no reference picture" 와 동일한 0 내지 `NumPocStFoll - 1` 의 범위에 있는 상태에서, 각각의 `RefPicSetStFoll[i]` 에 대해, 화상이 서브 조항 8.3.3.2 에서 특정된 바와 같이 생성되고, 다음이 적용된다:
- [0202] - 생성된 화상에 대한 `PicOrderCntVal` 의 값이 `PocStFoll[i]` 와 동일하게 세팅된다.
- [0203] - 생성된 화상에 대한 `PicOutputFlag` 의 값이 0 과 동일하게 세팅된다.
- [0204] - 생성된 화상은 "단기 참조를 위해 사용되는" 것으로서 마킹된다.
- [0205] - `RefPicSetStFoll[i]` 는 생성된 참조 화상이도록 세팅된다.
- [0206] - 생성된 화상에 대한 `nuh_layer_id` 의 값은 `nuh_layer_id` 와 동일하게 세팅된다.
- [0207] - `i` 가 "no-reference picture" 와 동일한 0 내지 `NumPocLtCurr - 1` 의 범위에 있는 상태에서, 각각의 `RefPicSetLtCurr[i]` 에 대해, 화상이 서브 조항 8.3.3.2 에서 특정된 바와 같이 생성되고, 다음이 적용된다:

- 생성된 화상에 대한 PicOrderCntVal 의 값이 PocLtCurr[i] 와 동일하게 세팅된다.
- 생성된 화상에 대한 slice_pic_order_cnt_lsb 의 값이 (PocLtCurr[i] & (MaxPicOrderCntLsb -1)) 과 동일한 것으로 추론된다.
- 생성된 화상에 대한 PicOutputFlag 의 값이 0 과 동일하게 세팅된다.
- 생성된 화상은 "장기 참조를 위해 사용되는" 것으로서 마킹된다.
- RefPicSetLtCurr[i] 는 생성된 참조 화상이도록 세팅된다.
- 생성된 화상에 대한 nuh_layer_id 의 값은 nuh_layer_id 와 동일하게 세팅된다.
- i 가 "no-reference picture" 와 동일한 0 내지 NumPocLtFoll - 1 의 범위에 있는 상태에서, 각각의 RefPicSetLtFoll[i] 에 대해, 화상이 서브 조항 8.3.3.2 에서 특정된 바와 같이 생성되고, 다음이 적용된다:
- 생성된 화상에 대한 PicOrderCntVal 의 값이 PocLtFoll[i] 와 동일하게 세팅된다.
- 생성된 화상에 대한 slice_pic_order_cnt_lsb 의 값이 (PocLtFoll[i] & (MaxPicOrderCntLsb -1)) 과 동일한 것으로 추론된다.
- 생성된 화상에 대한 PicOutputFlag 의 값이 0 과 동일하게 세팅된다.
- 생성된 화상은 "장기 참조를 위해 사용되는" 것으로서 마킹된다.
- RefPicSetLtFoll[i] 는 생성된 참조 화상이도록 세팅된다.
- 생성된 화상에 대한 nuh_layer_id 의 값은 nuh_layer_id 와 동일하게 세팅된다.

섹션 F.8.3.2 참조 화상 세트에 대한 디코딩 프로세스

RPS 및 화상 마킹을 위한에 대한 도출 프로세스는 다음의 순서화된 단계들에 따라 수행된다:

1. 다음이 적용된다:

```
for(i=0; i < NumPocLtCurr; i++)
```

```
    if(!CurrDeltaPocMsbPresentFlag[i])
```

```
        if(slice_pic_order_cnt_lsb, PocLtCurr[i] 가 입력들로서 주어진 채로 서브 조항 F.8.1.3 을 호출함으로써
        도출되는, PocLtCurr[i] 와 동일한 slice_pic_order_cnt_lsb 및 currPicLayerId + offsetPicLayerId 와
        동일한 nuh_layer_id 를 갖는 DPB 내의 참조 화상 picX 가 존재한다)
```

```
            RefPicSetLtCurr[i]=picX
```

```
        else
```

```
            RefPicSetLtCurr[i]="no reference picture"
```

```
    else
```

```
        if(PicOrderCntVal, PocLtcurr[i] 가 입력들로서 주어진 채로 서브 조항 F.8.1.3 을 호출함으로써
        도출되는, PocLtCurr[i] 와 동일한 PicOrderCntVal 및 currPicLayerId + offsetPicLayerId 와
        동일한 nuh_layer_id 를 갖는 DPB 내의 참조 화상 picX 가 존재한다)
```

```
            RefPicSetLtCurr[i]=picX
```

```
        else
```

```
            RefPicSetLtCurr[i]="no reference picture"
```

```
        (F-3)
```

```
for(i=0; i < NumPocLtFoll; i++)
```

```
    if(!FollDeltaPocMsbPresentFlag[i])
```


[0239] if(*slice_pic_order_cnt_lsb*, *PocLtFoll[i]* 가 입력들로서 주어진 채로 서브 조항 F.8.1.3 을 호출함으로써 도출되는, *PocLtFoll[i]* 와 동일한 *slice_pic_order_cnt_lsb* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 참조 화상 *picX* 가 존재한다)

[0240] RefPicSetLtFoll[i]=picX

[0241] else

[0242] RefPicSetLtFoll[i]="no reference picture"

[0243] else

[0244] if(*PicOrderCntVal*, *PocLtFoll[i]* 가 입력들로서 주어진 채로 서브 조항 F.8.1.3 을 호출함으로써 도출되는, *PocLtFoll[i]* 와 동일한 *PicOrderCntVal* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 참조 화상 *picX* 가 존재한다)

[0245] RefPicSetLtFoll[i]=picX

[0246] else

[0247] RefPicSetLtFoll[i]="no reference picture"

[0248] 2. [*currPicLayerId* 와 동일한 *nuh_layer_id* 를 갖고] RefPicSetLtCurr 및 RefPicSetLtFoll 에 포함되는 모든 참조 화상들은 "장기 참조를 위해 사용되는" 것으로서 마킹된다.

[0249] 3. 다음이 적용된다:

[0250] for(*i*=0; *i* < NumPocStCurrBefore; *i*++)

[0251] if(*PicOrderCntVal*, *PocStCurrBefore[i]* 가 입력들로서 주어진 채로 서브 조항 F.8.1.3 을 호출함으로써 도출되는, *PocStCurrBefore[i]* 와 동일한 *PicOrderCntVal* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 단기 참조 화상 *picX* 가 존재한다)

[0252] RefPicSetStCurrBefore[i]=picX

[0253] else

[0254] RefPicSetStCurrBefore[i]="no reference picture"

[0255] for(*i*=0; *i* < NumPocStCurrAfter; *i*++)

[0256] if(*PicOrderCntVal*, *PocStCurrAfter[i]* 가 입력들로서 주어진 채로 서브 조항 F.8.1.3 을 호출함으로써 도출되는, *PocStCurrAfter[i]* 와 동일한 *PicOrderCntVal* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 단기 참조 화상 *picX* 가 존재한다)

[0257] RefPicSetStCurrAfter[i]=picX

[0258] else

[0259] RefPicSetStCurrAfter[i]="no reference picture"

[0260] for(*i*=0; *i* < NumPocStFoll; *i*++)

[0261] if(*PicOrderCntVal*, *PocStFoll[i]* 가 입력들로서 주어진 채로 서브 조항 F.8.1.3 을 호출함으로써 도출되는, *PocStFoll[i]* 와 동일한 *PicOrderCntVal* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 단기 참조 화상 *picX* 가 존재한다)

[0262] RefPicSetStFoll[i]=picX

[0263] else

[0264] RefPicSetStFoll[i]="no reference picture"

[0265] 4. [*currPicLayerId* 와 동일한 *nuh_layer_id* 를 갖고] RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter, 또는 RefPicSetStFoll 에 포함되지 않는 DPB 내의 모든 참조

화상들은 "참조를 위해 사용되지 않는" 것으로서 마킹된다.

[0266] offsetPicLayerId 의 도출 프로세스

[0267] 위에서 도입된 offsetPicLayerId 변수의 도출은 다음과 같이 수행될 수도 있다:

[0268] 이러한 프로세스에 대한 입력은

[0269] - 단기 참조 화상들에 대한 PicOrderCntVal 및 장기 참조 화상들에 대한 slice_pic_order_cnt_lsb 에 대응하는 변수 currPocVal.

[0270] - 5 개의 리스트들 PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, 및 PocLtFoll 의 poc 값들에 대응하는 변수 refPocVal.

[0271] 이러한 프로세스에 대한 출력은

[0272] - currPicLayerId 와 동일한 nuh_layer_id 를 갖는 화상에 대응하는 offsetPicLayerId

[0273] 변수 currPicTemporalId 가 현재의 화상의 TemporalId 이도록 세팅된다

[0274] 변수 CurrPicnoResampleFlag 이 enable_non_curr_layer_ref_pic_pred[currPicLayerId] 와 동일하게 세팅된다

[0275] if(refPocVal 과 동일한 currPocVal 및 currPicLayerId + 1 과 동일한 nuh_layer_id 를 갖는 DPB 내의 참조 화상 picX 가 존재하고, CurrPicnoResampleFlag 가 1 과 동일하며 currPicTemporalId 가 0 보다 크다)

offsetPicLayerId = 1

else

offsetPicLayerId = 0

[0279] 섹션 F.13.5.2.2 DPB 로부터의 화상들의 출력 및 제거

[0280] 현재 화상의 디코딩 전 (그러나, 현재 화상의 제 1 슬라이스의 슬라이스 헤더를 파싱한 후) DPB 로부터의 화상들의 출력 및 제거는 현재 화상의 제 1 디코딩 유닛이 CPB 로부터 제거될 때 순간적으로 발생하며 다음과 같이 진행된다:

[0281] 서브 조항 F.8.3.2 에서 특정된 바와 같은 RPS 에 대한 디코딩 프로세스는 nuh_layer_id 의 동일한 값을 갖는 화상들만을 마킹하도록 호출된다.

[0282] 더 높은 계층 화상들에 대한 시간적 모션 벡터들 업데이트

[0283] 상술된 여러 실시형태들은 샘플들과 함께 참조 계층에 대한 향상 계층의 시간적 모션 벡터 예측자 (TMVP) 후보를 사용할 수도 있다. 그렇게 하는 것이 코딩 효율을 향상시킬 수도 있지만, 그것은 동시에 EL 패킷들이 비트스트림에 존재하지 않는 경우 (예를 들어, 그것들이 미성되거나 의도적으로 포기된다면) 모션 벡터 디코딩 동안 드리프트를 야기할 수도 있다.

[0284] 이러한 드리프트를 극복하는 것을 도울 수도 있는 일부 예시의 실시형태들이 이하에 기술된다. 이러한 예시의 실시형태들은 서로 독립적으로 또는 결합하여 적용될 수 있고, 스케일러블 코딩, 깊이를 갖거나 갖지 않는 멀티뷰 코딩, 및 HEVC 및 다른 비디오 코덱들의 다른 확장들에 적용가능하거나 확장될 수도 있다.

[0285] 중요 액세스 유닛

[0286] 용어 "중요 액세스 유닛" 은 중요 화상들만을 포함하는 액세스 유닛을 지칭할 수도 있다. 중요 화상은 0 의 시간적 ID 를 갖는 화상일 수도 있다. 다른 예에서, 중요 화상은 중요 화상으로서 명시적으로 시그널링되는 화상일 수도 있다. 용어 "비중요 (non-key) 액세스 유닛" 은 중요 액세스 유닛이 아닌 액세스 유닛을 지칭할 수도 있다.

[0287] 더 높은 계층 화상에 대한 TMVP 업데이트

[0288] 더 높은 계층 화상들이 더 낮은 계층들에 대한 참조로서 사용되는 경우, 더 높은 계층들에 대한 다음의 시간적 모션 벡터 정보 업데이트가 제안된다...

[0289] 하나의 실시형태에서, 비중요 액세스 유닛의 마지막 디코딩 유닛을 디코딩한 후, 계층 인덱스 $i > 0$ 으로부터 시작하는 모든 계층들에 대해, 시간적 모션 벡터 정보는, 그러한 더 낮은 계층이 존재하는 경우, 인덱스 $j=i-1$

을 갖는 더 낮은 계층 내의 동일장소에 배치된 참조 화상으로부터 계층 인덱스 i 를 갖는 그것의 바로 더 높은 계층으로 카피된다. 중요 액세스 유닛의 경우, 그러한 업데이트는 생략된다.

[0290] 다른 실시형태에서, 비중요 액세스 유닛의 마지막 디코딩 유닛을 디코딩한 후, 계층 인덱스 $i > 0$ 으로부터 시작하는 모든 계층들에 대해, 시간적 모션 벡터 정보는, 그러한 더 낮은 계층이 존재하는 경우, 인덱스 $j=i-1$ 을 갖는 더 낮은 계층 내의 동일장소에 배치된 참조 화상으로부터 계층 인덱스 i 를 갖는 그것의 바로 더 높은 계층으로 카피된다. 이러한 예에서, 계층 인덱스 j 는 명시적으로 시그널링될 수도 있다. 예를 들어, 현재의 계층이 그로부터 정보 (예를 들어, 시간적 모션 벡터 정보) 를 도출하는 2 이상의 향상 계층이 존재하는 경우, 현재 계층을 위해 사용되는 향상 계층의 계층 인덱스 j 는 비트스트림에서 시그널링될 수 있다.

[0291] 또 다른 예에서, 플래그는 위의 문단들에서 정의된 프로세스들을 명시적으로 인에이블 또는 디스에이블하도록 선택적으로 시그널링될 수도 있다. 이러한 플래그는 VPS, SPS, PPS 와 같은 상이한 입도 (granularity) 인덱스 파라미터 세트들에서, 또는 VUI 또는 SEI 메시지로써, 및 슬라이스 헤더에서 또는 그들의 각각의 확장 헤더들에서 시그널링될 수도 있다.

[0292] **중요 화상 프레임워크를 갖는 단일-루프 디코딩 메커니즘**

[0293] 인터-계층 텍스처 예측이 제약된 인트라 예측 (CIP) 를 사용하여 코딩되는 동일장소에 배치된 코딩 유닛들 (CUs) 또는 디코딩 순서에서 조기의 액세스 유닛들로부터의 임의의 정보를 참조하지 않고 코딩되는 동일장소에 배치된 CU 들에 제한되는 경우, 소정의 구현들 (예를 들어, SHVC) 에서 단일-루프 디코딩 구조를 사용하는 것이 가능하고 때때로 바람직하다. 하나의 예에서, 디코딩 순서에서 조기의 액세스 유닛들로부터의 임의의 정보를 참조하지 않고 CU 를 코딩하는 것은 CU 가 인터-계층 텍스처 예측 (예를 들어, 인트라 BL) 을 사용하여 코딩되는 것을 의미할 수도 있다.

[0294] 그러나, 현존하는 코딩 스킴들에서, 단일-루프 디코딩 구조가 인에이블되는지 여부에 대한 이러한 표시는 이용 가능하지 않을 수도 있다. 아래에 기술되는 예시의 실시형태들을 사용함으로써, 단일-루프 디코딩이 더욱 이롭게 이용될 수 있다.

[0295] **단일-루프 디코딩: 중요 액세스 유닛들**

[0296] 이러한 실시형태에서, 더 높은 계층 참조 화상들이 더 낮은 계층들에 대한 참조로서 사용되는 경우, 중요 액세스 유닛들에 대해, 인터-계층 예측이 직접 또는 간접적으로 디코딩 순서에서 조기의 액세스 유닛들로부터의 어떠한 정보도 사용하지 않고 코딩된 샘플들로부터 예측되는 이웃하는 코딩 블록들의 디코딩된 샘플들 및 레지듀얼 데이터만을 사용하여 수행된다고 진술하는 인코더 컨포먼스 제한이 구현된다. 그러한 제한은 플래그를 사용하여 시그널링될 수도 있다. 예시의 플래그 `key_pic_constrained_inter_layer_pred_idc` 는 다음과 같이 정의될 수도 있다: 0 과 동일한 `key_pic_constrained_inter_layer_pred_idc` 는, 중요 액세스 유닛들 (또는 화상들) 에 대해, 인터-계층 예측이 인트라 또는 인터 예측 모드들 중 어느 하나를 사용하여 코딩되는 동일장소에 배치된 코딩 유닛들의 디코딩된 샘플들 및 레지듀얼 데이터를 사용한다는 것을 나타낸다. 1 과 동일한 `constrained_inter_layer_pred_flag` 는 제약된 인터-계층 예측을 나타내며, 그 경우에 인터-계층 예측은, 인트라/인터 예측 또는 인터-계층 예측 또는 그들이 조합을 통해, 직접 또는 간접적으로 디코딩 순서에서 조기의 액세스 유닛들로부터의 어떠한 정보도 사용하지 않고 코딩되는 동일장소에 배치된 코딩 유닛들로부터의 디코딩된 샘플들 및 레지듀얼 데이터만을 사용한다.

[0297] 그 플래그는 VPS, SPS, PPS 와 같은 상이한 입도 인덱스 파라미터 세트들에서, 또는 VUI 또는 SEI 메시지로써, 및 슬라이스 헤더에서 또는 그들의 각각의 확장 헤더들에서 시그널링될 수도 있다.

[0298] **단일-루프 디코딩: 비중요 액세스 유닛들**

[0299] 비중요 액세스 유닛들 (또는 화상들) 의 경우, 단일-루프 디코딩을 허용하기 위해, 다음의 제한들이 적용될 수도 있다:

[0300] 1) 참조 계층 화상들에 대해 디블록킹 필터 및 샘플 적응 오프셋 (SAO) 을 디스에이블;

[0301] 2) 참조 계층 화상들에 대한 제약된 인트라 예측 (CIP) 를 인에이블

[0302] 3) 재구성된 참조 계층 화상들로부터의 년-제로 모션 예측을 디스에이블

[0303] 4) 현재의 블록 내의 각 샘플의 참조 화상 인덱스 `refIdxLX` (X 는 0 또는 1 에 의해 대체됨) 중 하나만이 참조 계층 화상에 대응하고 현재 계층 샘플에 대한 동일장소에 배치된 참조 샘플이 양방향 예측을 사용하는 경우 향

상 계층 블록에 대한 양방향 예측을 디스에이블.

[0304] 대안적으로, 네 번째 제한은 다음에 의해 대체될 수도 있다:

[0305] 4) 현재 계층 샘플들 (xCurr, yCurr) 에 대응하는 참조 화상 인덱스 refIdxLX (X 는 0 또는 1 에 의해 대체됨) 중 하나만이 참조 계층 화상을 가리키고, 동일장소에 배치된 참조 샘플이 양방향 예측을 사용하는 경우 항상 계층 블록에 대한 양방향 예측을 디스에이블.

[0306] 이러한 예에서, 상기 제한들 중 4 개 모두가 만족되는 경우, 단일-루프 디코딩이 비중요 액세스 유닛들에 대해 인에이블될 수도 있다. 예를 들어, 단일-루프 디코딩에서, EL 은 비중요 액세스 유닛들에 대한 참조 계층을 완전히 재구성하지 않고 디코딩될 수도 있다. 단일-루프 디코딩은 BL 및 EL 양자 모두가 인터 예측을 위해 동일한 참조들을 사용하기 때문에 이러한 예에서 인에이블된다. 이러한 예에서, EL 은 그 재구성에 다른 레지듀얼 신호를 가산할 수도 있다. 예를 들어, 인코더는 비트스트림에 추가적인 에러 신호들을 가산할 수도 있다. 그러한 추가적인 에러 신호들은 디코딩된 화상들의 품질을 향상시키고 비디오 품질을 향상시키기 위해 사용될 수도 있다.

[0307] **더 높은 계층 화상의 상이한 표현의 사용**

[0308] 하나의 실시형태에서, 더 높은 계층 화상들의 상이한 표현 (예를 들어, 리샘플링) 이 사용될 수도 있는지 여부가 이하의 도출 프로세스를 사용하여 추론된다. 예를 들어, 현재의 화상을 코딩하기 위해 더 높은 계층 참조 화상을 사용하기 전에, 더 높은 계층 참조 화상은 상이한 표현 (예를 들어, 사이즈, 비트-깊이 등) 으로 변환될 필요가 있을 수도 있다.

[0309] 하나의 예에서, 예시의 변수 additionalHigherLayerRefpicforCurrPicFlag 가 사용될 수도 있다. 계층 id i 를 갖는 현재의 계층 내의 현재 화상에 대한 변수 additionalHigherLayerRefpicforCurrPicFlag 는 다음과 같이 정의될 수도 있다: 0 과 동일한 additionalHigherLayerRefpicforCurrPicFlag 는, layer_id_in_nuh[i] 와 동일한 nuh_layer_id 를 갖는 현재의 화상에 대해, layer_id_in_nuh[i] 보다 큰 nuh_layer_id 를 갖는 디코딩된 화상들이 현재 화상에 대한 참조로서 사용되는 경우, 어떠한 추가적인 참조 화상 표현도 필요하지 않는다는 것을 특징한다. 1 과 동일한 additionalHigherLayerRefpicforCurrPicFlag 는, layer_id_in_nuh[i] 와 동일한 nuh_layer_id 를 갖는 현재 화상에 대하여, layer_id_in_nuh[i] 보다 큰 nuh_layer_id 를 갖는 디코딩된 화상들이 현재 화상에 대한 참조로서 사용되는 경우, 추가적인 참조 화상 표현이 필요하다는 것을 특징한다.

[0310] 하나의 실시형태에서, 계층 ID i 를 갖는 현재 계층 내의 현재 화상에 대해, additionalHigherLayerRefpicforCurrPicFlag 의 값은 SNR 확장성에 대해 0 으로, 및 다른 확장성에 대해 1 로 세팅될 수도 있다.

[0311] 다른 실시형태에서, 변수들 PicWidthInSamplesL 및 PicHeightInSamplesL 은 루마 샘플들의 유닛들에서 현재 화상의 폭 및 높이와 동일하게 세팅될 수도 있고, 변수들 RefLayerPicWidthInSamplesL 및 RefLayerPicHeightInSamplesL 은 각각 루마 샘플들의 유닛들에서 디코딩된 참조 계층 화상의 폭 및 높이와 동일하게 세팅될 수도 있다. 또, 변수들 ScaledRefLayerLeftOffset, ScaledRefLayerTopOffset, ScaledRefLayerRightOffset 및 ScaledRefLayerBottomOffset 은 다음과 같이 도출될 수도 있다:

ScaledRefLayerLeftOffset = scaled_ref_layer_left_offset[dRIIdx] << 1
ScaledRefLayerTopOffset = scaled_ref_layer_top_offset[dRIIdx] << 1
ScaledRefLayerRightOffset = scaled_ref_layer_right_offset[dRIIdx] << 1
ScaledRefLayerBottomOffset = scaled_ref_layer_bottom_offset[dRIIdx] << 1

[0312]

[0313] 현재 계층의 PicWidthInSamplesL 이 RefLayerPicWidthInSamplesL 과 동일하고, 현재 계층의 PicHeightInSamplesL 이 RefLayerPicHeightInSamplesL 과 동일하며, ScaledRefLayerLeftOffset, ScaledRefLayerTopOffset, ScaledRefLayerRightOffset 및 ScaledRefLayerBottomOffset 의 값이 모두 0 과 동일한 경우, additionalHigherLayerRefpicforCurrPicFlag 의 값은 0 으로 세팅될 수도 있다. 그렇지 않으면, additionalHigherLayerRefpicforCurrPicFlag 의 값은 1 로 세팅된다.

[0314] 다른 실시형태에서, 연관된 NAL 유닛에 의해 참조된 시퀀스 파라미터 세트 (SPS) 내에 있을 수도 있는 (사용된 참조 화상들의 수를 나타내는) max_num_ref_frames 가 2 보다 작은 경우, additionalHigherLayerRefpicforCurrPicFlag 는 0 으로 세팅될 수도 있다. 비트스트림 컨포먼스 제한은 현

재의 디코딩된 참조 화상을, 그리고 additionalHigherLayerRefpicforCurrPicFlag 가 1 과 동일한 경우에, 현재 참조 베이스 화상을 마킹한 후에, "참조를 위해 사용되는" 것으로서 마킹된 프레임들의 총수가 max_num_ref_frames 및 1 중 더 큰 것을 초과하지 않는다고 진술한다. 1 과 동일한 additionalHigherLayerRefpicforCurrPicFlag 를 갖는 참조 화상들은 인터 예측을 위해 참조 화상들로서만 사용되고 출력되지 않는다.

[0315] **다시 논의되는 코딩 효율 대 드리프트**

[0316] 상술된 바와 같이, 코딩 효율과 드리프트 효과들 사이에 트레이드-오프가 존재할 수도 있다. 더 높은 계층 화상들에 기초한 더 낮은 계층 화상들의 코딩을 허용하고 동시에 드리프트의 효과들을 최소화하는 여러 실시형태들이 본 개시에서 논의되어 왔다. 하나 이상의 그러한 실시형태들에서, 모션 및 텍스처 정보 양자 모두는 더 높은 계층 디코딩된 화상으로부터 도출될 수도 있다.

[0317] **상이한 계층들로부터의 모션 정보 및 텍스처 정보**

[0318] 다른 실시형태에서, 모션 정보는 현재 계층의 시간적 화상들로부터 도출될 수도 있고, 텍스처 정보는 현재 계층 내의 현재 화상을 코딩하기 위해 더 높은 계층 디코딩된 화상들로부터 도출될 수도 있다. 더 높은 계층으로부터의 텍스처 정보는 더 양호한 품질을 가질 수도 있다는 것이 이해되어야 한다. 그러나, 현재 계층으로부터 모션 정보를 도출하는 것이 더 양호할지도 모를 경우들이 존재할 수도 있다. 또, 더 높은 계층 패킷들이 상실된 경우, 모션 정보에 도입된 에러 (예를 들어, 드리프트)는 텍스처 정보에 도입된 에러보다 더 심각할 수도 있다. 따라서, 현재 계층으로부터 모션 정보를 도출함으로써, 적어도 모션 정보는 더 높은 계층 패킷들이 상실되거나 의도적으로 포기되는 경우에 드리프트-프루프 (drift-proof) 로 만들어질 수도 있다.

[0319] 현재 계층 내의 현재 화상을 코딩할 때 더 높은 계층으로부터 도출된 텍스처 정보 및 현재 계층으로부터 도출된 모션 정보를 사용하는 일부 예시의 구현들이 이하에 기술된다. 이들 방법들은 서로 독립적으로 또는 결합하여 적용될 수 있고, 스케일러블 코딩, 깊이를 갖거나 갖지 않는 멀티뷰 코딩, 및 HEVC 및 다른 비디오 코덱들의 다른 확장들에 적용가능하거나 확장될 수도 있다.

[0320] **실시형태 #1: 코레벨 변경**

[0321] 하나의 실시형태에서, 참조 화상 세트 (RPS) 구성은 RPS 가 EL 및 BL 양자 모두로부터의 화상들을 포함하도록 변경된다. 예를 들어, RPS 내의 엔트리들의 수가 배가되며, 여기서 RPS 내의 EL 화상들의 수는 RPS 내의 BL 화상들의 수와 동일하다. 하나의 실시형태에서, RPS 는 이하의 섹션 F.8.3.2 에서 도시된 바와 같이 변경될 수도 있다. 다른 실시형태에서, RPS 는 본 기술에서 알려진 임의의 방법을 포함하여, 여기서 논의되지 않은 임의의 방법을 사용하여 추가적인 BL 화상들을 포함하도록 변경될 수도 있다.

[0322] RPS 가 구성된 후, 참조 화상 리스트 (RPL) 가 구성된다. 하나의 예에서, RPS 는 현재 화상을 코딩하기 위해 사용될 수도 있는 모든 코딩된 화상을 포함할 수도 있는 반면, RPL 은 현재 화상에 의해 사용될 것 같은 그러한 디코딩된 화상들을 포함할 수도 있다. 인코더는 어느 화상들이 RPL 에 삽입되는지를 선택할 수도 있다. RPL 내의 참조 화상들 각각은 대응하는 참조 인덱스를 사용하여 참조될 수도 있다.

[0323] RPL 이 구성된 후, RPL 은 변경된다. 하나의 실시형태에서, RPL 은 (예를 들어, 동일장소에 배치된 참조 인덱스를 갖는 RPL 내의 마지막 엔트리를 RPS 내에 존재하는 대응하는 베이스 계층 화상으로 대체함으로써) 이하에 섹션 H.8.3.4 에서 도시된 바와 같이 변경된다. 예를 들어, 인코더는 베이스 계층 내의 현재 화상의 RPL 내로 BL 화상 #1 을 삽입하는 것이 바람직할 수도 있다는 것을 결정할 수도 있다. 그러한 경우에, 인코더는 RPL 내의 마지막 화상을 BL 화상 #1 로 대체할 수도 있다. 다른 실시형태에서, BL 화상 #1 은 RPL 내의 (예를 들어, 동일한 액세스 유닛 내의) BL 화상 #1 에 대응하는 EL 참조 화상을 대체한다. 다른 실시형태에서, BL 화상 #1 은 현재 화상의 RPL 내의 임의의 위치에 있는 임의의 EL 화상을 대체할 수도 있다.

[0324] **실시형태 #1의 구현: SHVC 사양에 대한 제안된 변경**

[0325] (이탤릭체로 도시된) 다음의 변경들이 HEVC 스케일러블 확장 (SHVC) 의 드리프트에 대해 행해질 수도 있다.

[0326] **섹션 F.8.3.2 참조 화상 세트에 대한 디코딩 프로세스**

[0327]

[0328] 현재 화상의 RPS 는 5 개의 RPS 리스트들로 이루어진다; RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr 및 RefPicSetLtFoll.

[0329] RefPicSetStCurrBefore, RefPicSetStCurrAfter, 및 RefPicSetStFoll 은 집합적으로 단기 RPS 로서 지칭된다. RefPicSetLtCurr 및 RefPicSetLtFoll 은 집합적으로 장기 RPS 로서 지칭된다.

[0330] 주의 1 - RefPicSetStCurrBefore, RefPicSetStCurrAfter, 및 RefPicSetLtCurr 는 디코딩 순서에서 현재 화상에 후속하는 하나 이상의 화상들 및 현재 화상의 인터 예측을 위해 사용될 수도 있는 모든 참조 화상들을 포함한다. RefPicSetStFoll 및 RefPicSetLtFoll 은 현재 화상의 인터 예측을 위해 사용되지 않지만 디코딩 순서에서 현재 화상에 후속하는 하나 이상의 화상들에 대한 인터 예측에서 사용될 수도 있는 모든 참조 화상들로 이루어진다.

[0331] 변수 *offsetPicLayerId* 는 *enable_higher_layer_ref_pic_pred[currPicLayerId]* 가 0 과 동일하지 않고 *TemporalId* 가 현재의 화상에 대하여 0 과 동일하지 않는 경우 1 과 동일하게 세팅된다.

[0332] RPS 및 화상 마킹에 대한 도출 프로세스는 다음의 순서화된 단계들에 따라 수행된다:

[0333] 1. 다음이 적용된다:

[0334] for(*i*=0; *i* < NumPocLtCurr; *i*++)

[0335] if(!CurrDeltaPocMsbPresentFlag[*i*])

[0336] if(PocLtCurr[*i*] 와 동일한 *slice_pic_order_cnt_lsb* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 참조 화상 *picX* 가 존재한다)

[0337] RefPicSetLtCurr[*i*]=*picX*

[0338] else

[0339] RefPicSetLtCurr[*i*]="no reference picture"

[0340] else

[0341] if(PocLtCurr[*i*] 와 동일한 *PicOrderCntVal* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 참조 화상 *picX* 가 존재한다)

[0342] RefPicSetLtCurr[*i*]=*picX*

[0343] else

[0344] RefPicSetLtCurr[*i*]="no reference picture"

[0345] for(*i*=0; *i* < NumPocLtFoll; *i*++)

[0346] if(!FollDeltaPocMsbPresentFlag[*i*])

[0347] if(PocLtFoll[*i*] 와 동일한 *slice_pic_order_cnt_lsb* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 참조 화상 *picX* 가 존재한다)

[0348] RefPicSetLtFoll[*i*]=*picX*

[0349] else

[0350] RefPicSetLtFoll[*i*]="no reference picture"

[0351] else

[0352] if(PocLtFoll[*i*] 와 동일한 *PicOrderCntVal* 및 *currPicLayerId* + *offsetPicLayerId* 와 동일한 *nuh_layer_id* 를 갖는 DPB 내의 참조 화상 *picX* 가 존재한다)

[0353] RefPicSetLtFoll[*i*]=*picX*

[0354] else

[0355] RefPicSetLtFoll[*i*]="no reference picture"

[0356] if(*offsetLayerId*) {

[0357] for(*i*=0; *i* < NumPocLtCurr; *i*++)

[0358] *if(!CurrDeltaPocMsbPresentFlag[i])*

[0359] *if(PocLtCurr[i] 와 동일한 slice_pic_order_cnt_lsb 및 currPicLayerId 와 동일한 nuh_layer_id 를 갖는 DPB*
내의 참조 화상 picX 가 존재한다)

[0360] *RefPicSetLtCurr[i + NumPocLtCurr]=picX*

[0361] *else*

[0362] *RefPicSetLtCurr[i + NumPocLtCurr]="no reference picture"*

[0363] *else*

[0364] *if(PocLtCurr[i] 와 동일한 PicOrderCntVal 및 currPicLayerId 와 동일한 nuh_layer_id 를 갖는 DPB 내의 참조*
화상 picX 가 존재한다)

[0365] *RefPicSetLtCurr[i + NumPocLtCurr]=picX*

[0366] *else*

[0367] *RefPicSetLtCurr[i + NumPocLtCurr]="no reference picture"*

[0368] *for(i=0; i < NumPocLtFoll; i++)*

[0369] *if(!FollDeltaPocMsbPresentFlag[i])*

[0370] *if(PocLtFoll[i] 와 동일한 slice_pic_order_cnt_lsb 및 currPicLayerId 와 동일한 nuh_layer_id 를 갖는 DPB*
내의 참조 화상 picX 가 존재한다)

[0371] *RefPicSetLtFoll[i + NumPocLtFoll]=picX*

[0372] *else*

[0373] *RefPicSetLtFoll[i + NumPocLtFoll]="no reference picture"*

[0374] *else*

[0375] *if(PocLtFoll[i] 와 동일한 PicOrderCntVal 및 currPicLayerId 와 동일한 nuh_layer_id 를 갖는 DPB 내의 참조*
화상 picX 가 존재한다)

[0376] *RefPicSetLtFoll[i + NumPocLtFoll]=picX*

[0377] *else*

[0378] *RefPicSetLtFoll[i + NumPocLtFoll]="no reference picture"*

[0379] *}*

[0380] 2. currPicLayerId 와 동일한 nuh_layer_id 를 갖고 RefPicSetLtCurr 및 RefPicSetLtFoll 에 포함되는 모든 참
조 화상들은 "장기 참조를 위해 사용되는" 것으로서 마킹된다.

[0381] 3. 다음이 적용된다:

[0382] *for(i=0; i < NumPocStCurrBefore; i++)*

[0383] *if(PocStCurrBefore[i] 와 동일한 PicOrderCntVal 및 currPicLayerId + offsetPicLayerId 와 동일한*
nuh_layer_id 를 갖는 DPB 내의 단기 참조 화상 picX 가 존재한다)

[0384] *RefPicSetStCurrBefore[i]=picX*

[0385] *else*

[0386] *RefPicSetStCurrBefore[i]="no reference picture"*

[0387] *for(i=0; i < NumPocStCurrAfter; i++)*

[0388] *if(PocStCurrAfter[i] 와 동일한 PicOrderCntVal 및 currPicLayerId + offsetPicLayerId 와 동일한*
nuh_layer_id 를 갖는 DPB 내의 단기 참조 화상 picX 가 존재한다)

[0389] RefPicSetStCurrAfter[i]=picX

[0390] else

[0391] RefPicSetStCurrAfter[i]="no reference picture"

[0392] for(i=0; i < NumPocStFoll; i++)

[0393] if(PocStFoll[i] 와 동일한 PicOrderCntVal 및 currPicLayerId + offsetPicLayerId 와 동일한
nuh_layer_id 를 갖는 DPB 내의 단기 참조 화상 picX 가 존재한다)

[0394] RefPicSetStFoll[i]=picX

[0395] else

[0396] RefPicSetStFoll[i]="no reference picture"

[0397] if(offsetPicLayerId) {

[0398] for(i=0; i < NumPocStCurrBefore; i++)

[0399] if(PocStCurrBefore[i] 와 동일한 PicOrderCntVal 및 currPicLayerId 와 동일한 nuh_layer_id 를 갖는 DPB 내
의 단기 참조 화상 picX 가 존재한다)

[0400] RefPicSetStCurrBefore[i + NumPocStCurrBefore]=picX

[0401] else

[0402] RefPicSetStCurrBefore[i + NumPocStCurrBefore]="no reference picture"

[0403] for(i=0; i < NumPocStCurrAfter; i++)

[0404] if(PocStCurrAfter[i] 와 동일한 PicOrderCntVal 및 currPicLayerId 와 동일한 nuh_layer_id 를 갖는 DPB 내의
단기 참조 화상 picX 가 존재한다)

[0405] RefPicSetStCurrAfter[i + NumPocStCurrBefore]=picX

[0406] else

[0407] RefPicSetStCurrAfter[i + NumPocStCurrBefore]="no reference picture"

[0408] for(i=0; i < NumPocStFoll; i++)

[0409] if(PocStFoll[i] 와 동일한 PicOrderCntVal 및 currPicLayerId 와 동일한 nuh_layer_id 를 갖는 DPB 내의 단기
참조 화상 picX 가 존재한다)

[0410] RefPicSetStFoll[i + NumPocStCurrBefore]=picX

[0411] else

[0412] RefPicSetStFoll[i + NumPocStCurrBefore]="no reference picture"

[0413] }

[0414] 4. currPicLayerId 와 동일한 nuh_layer_id 를 갖고 RefPicSetLtCurr, RefPicSetLtFoll,
RefPicSetStCurrBefore, RefPicSetStCurrAfter, 또는 RefPicSetStFoll 에 포함되지 않는 DPB 내의 모든 참조
화상들은 "참조를 위해 사용되지 않는" 것으로서 마킹된다.

[0415] 주의 2 - 대응하는 화상들이 DPB 내에 존재하지 않기 때문에 "no reference picture" 와 동일한 RPS 리스트들
내의 하나 이상의 엔트리들이 존재할 수도 있다. "no reference picture" 와 동일한 RefPicSetStFoll 또는
RefPicSetLtFoll 내의 엔트리들은 무시되어야 한다. 의도하지 않은 화상 상실은 "no reference picture"
와 동일한 RefPicSetStCurrBefore, RefPicSetStCurrAfter, 또는 RefPicSetLtCurr 내의 각각의 엔트리에 대해
추론되어야 한다.

[0416] **섹션 F.13.5.2.2 DPB 로부터의 화상들의 출력 및 제거**

[0417] 현재 화상의 디코딩 전에 (그러나 현재 화상의 제 1 슬라이스의 슬라이스 헤더를 파싱한 후에) DPB 로부터의 화

상들의 출력 및 제거는 현재 화상의 제 1 디코딩 유닛이 CPB 로부터 제거될 때 순간적으로 발생하며 다음과 같이 진행된다:

[0418] 서브 조항 F.8.3.2 에서 특정된 바와 같은 RPS 에 대한 디코딩 프로세스는 *nuh_layer_id* 의 동일한 값을 갖는 화상들만을 마킹하도록 호출된다.

[0419] **섹션 H.8.3.4 참조 화상 리스트들 구성을 위한 디코딩 프로세스**

[0420] 이러한 프로세스는 각각의 P 또는 B 슬라이스에 대해 디코딩 프로세스의 시작에서 호출된다.

[0421] 참조 화상들은 서브조항 8.5.3.3.2 에서 특정된 바와 같은 참조 인덱스들을 통해 어드레싱된다. 참조 인덱스는 참조 화상 리스트로의 인덱스이다. P 슬라이스를 디코딩하는 경우, 단일의 참조 화상 리스트 RefPicList0 가 존재한다. B 슬라이스를 디코딩하는 경우, RefPicList0 에 더하여 제 2 독립적인 참조 화상 리스트 RefPicList1 이 존재한다.

[0422] 각 슬라이스에 대한 디코딩 프로세스의 시작에서, 참조 화상 리스트들 RefPicList0, 및 B 슬라이스들의 경우, RefPicList1 이 다음과 같이 도출된다:

[0423] 변수 *offsetPicLayerId* 는, *enable_higher_layer_ref_pic_pred[currPicLayerId]* 가 1 과 동일하고 *TemporalId* 가 현재의 화상에 대하여 0 보다 큰 경우, 1 과 동일하게 세팅된다.

[0424] 변수 NumRpsCurrTempList0 이 Max(num_ref_idx_l0_active_minus1 + 1, NumPicTotalCurr) 과 동일하게 세팅되고, 리스트 RefPicListTemp0 는 다음과 같이 구성된다:

```

rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++,
i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumActiveRefLayerPics0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetInterLayer0[ i ]
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0; rIdx++,
i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumActiveRefLayerPics1; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetInterLayer1[ i ]
}
while( rIdx < NumRpsCurrTempList0 << offsetPicLayerId ) {

```

[0425]

```

for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++,
i++)
    RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i + NumRpsCurrTempList0]
for( i = 0; i < NumActiveRefLayerPics0; rIdx++, i++)
    RefPicListTemp0[ rIdx ] = RefPicSetInterLayer0[ i + NumRpsCurrTempList0]
for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0; rIdx++,
i++)
    RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i + NumRpsCurrTempList0]
for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++)
    RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i + NumRpsCurrTempList0]
for( i = 0; i < NumActiveRefLayerPics1; rIdx++, i++)
    RefPicListTemp0[ rIdx ] = RefPicSetInterLayer1[ i + NumRpsCurrTempList0]
}

```

리스트 RefPicList0 은 다음과 같이 구성된다:

```

for( rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++)
    RefPicList0[ rIdx ] = ref_pic_list_modification_flag_l0 ?
    RefPicListTemp0[ list_entry_l0[ rIdx ] ] : RefPicListTemp0[ rIdx ]
if(offsetPicLayerId && collocated_from_l0_flag)
    RefPicList0[ rIdx - 1 ] = ref_pic_list_modification_flag_l0 ?
    RefPicListTemp0[ list_entry_l0[ collocated_ref_idx ] + NumRpsCurrTempList0 ] :
    RefPicListTemp0[ collocated_ref_idx + NumRpsCurrTempList0]

```

슬라이스가 B 슬라이스인 경우, 변수 NumRpsCurrTempList1 은 $\text{Max}(\text{num_ref_idx_l1_active_minus1} + 1, \text{NumPicTotalCurr})$ 과 동일하게 세팅되고, 리스트 RefPicListTemp1 는 다음과 같이 구성된다:

```

rIdx = 0
while( rIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++, i++)
    )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumActiveRefLayerPics1; rIdx++, i++)
        RefPicListTemp1[ rIdx ] = RefPicSetInterLayer1[ i ]
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1; rIdx++,
i++)

```

```

    RefPicListTemp1[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumActiveRefLayerPics0; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetInterLayer0[ i ]
}
while( rIdx < NumRpsCurrTempList1 << offsetPicLayerId ) {
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++,
        i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrAfter[ i + NumRpsCurrTempList1 ]
    for( i = 0; i < NumActiveRefLayerPics1; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetInterLayer1[ i + NumRpsCurrTempList1 ]
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1; rIdx++,
        i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrBefore[ i + NumRpsCurrTempList1 ]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetLtCurr[ i + NumRpsCurrTempList1 ]
    for( i = 0; i < NumActiveRefLayerPics0; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetInterLayer0[ i + NumRpsCurrTempList1 ]
}

```

[0431]

[0432]

슬라이스가 B 슬라이스인 경우, 리스트 RefPicList1 은 다음과 같이 구성된다:

```

for( rIdx = 0; rIdx <= num_ref_idx_l1_active_minus1; rIdx++ )
    RefPicList1[ rIdx ] = ref_pic_list_modification_flag_l1 ?
    RefPicListTemp1[ list_entry_l1[ rIdx ] ] : RefPicListTemp1[ rIdx ]
if( offsetPicLayerId && !collocated_from_l0_flag )
    RefPicList1[ rIdx - 1 ] = ref_pic_list_modification_flag_l1 ?
    RefPicListTemp1[ list_entry_l1[ collocated_ref_idx ] + NumRpsCurrTempList1 ] :
    RefPicListTemp1[ collocated_ref_idx + NumRpsCurrTempList1 ]

```

[0433]

[0434]

주의 - 인터 계층 참조 화상들로부터의 모션 벡터들이 단지 제로 모션인 것으로 제약되기 때문에, SHVC 인코더는 단지 인터-계층 참조 화상들만이 현재 화상 내의 모든 슬라이스들의 참조 화상 리스트들에 존재할 때, slice_temporal_mvp_enabled_flag 를 제로로 세팅함으로써 현재 화상에 대한 시간적 모션 벡터 예측을 디스에이블해야 한다. 이것은 collocated_from_l0_flag 및 collocated_ref_idx 와 같은 임의의 추가적인 선택스 엘리먼트들을 전송할 필요를 회피한다.

[0435]

주의 - offsetPicLayerId 가 0 과 동일하지 않는 경우, collocated_ref_idx 는 그의 각각의 리스트에서의 마지막 인덱스 위치와 동일할 것이다.

[0436]

실시형태 #2: 베이스 계층으로부터 향상 계층으로의 모션 정보의 카피

[0437]

하나의 실시형태에서, BL 의 모션 정보는 그의 동일장소에 배치된 향상 계층 화상으로 카피될 수 있다. 예를 들어, 현재 화상의 RPL 은 하나 이상의 EL 화상들을 포함할 수도 있다. 하나 이상의 EL 화상들의 모션 정보는 하나 이상의 BL 화상들의 모션 정보로 대체될 수도 있다. 하나의 예에서, EL 화상의 모션 정보는 EL 화상에 대해 동일장소에 배치되는 BL 화상의 모션 정보로 겹쳐 쓰여진다.

[0438]

하나의 실시형태에서, 모션 정보 카피 프로세스는 4 x 4 서브 블록 레벨에서 구현될 수도 있다. 다른 실시형태에서, 모션 정보 카피 프로세스는 4 x 4 이외의 서브 블록 레벨에서 구현될 수도 있다. 모션 정보 카피 프로세스는 그의 모션 정보가 대체되고/겹쳐 쓰여지고 있는 향상 계층 화상을 디코딩한 후에 수행될 수도 있다.

[0439] **실시형태 #3: 항상 계층으로부터 베이스 계층으로의 텍스처 정보의 카피**

[0440] 하나의 실시형태에서, EL의 텍스처 정보는 그의 동일장소에 배치된 BL 화상으로 카피될 수 있다. 예를 들어, 현재 화상의 RPL은 하나 이상의 BL 화상들을 포함할 수도 있다. 하나 이상의 BL 화상들의 텍스처 정보는 하나 이상의 EL 화상들의 텍스처 정보로 대체될 수도 있다. 하나의 예에서, BL 화상의 텍스처 정보는 BL 화상에 대해 동일장소에 배치되는 EL 화상의 텍스처 정보로 겹쳐 쓰여진다.

[0441] 하나의 실시형태에서, 텍스처 정보 카피 프로세스는 4 x 4 서브 블록 레벨에서 구현될 수도 있다. 다른 실시형태에서, 텍스처 정보 카피 프로세스는 4 x 4 이외의 서브 블록 레벨에서 구현될 수도 있다. 텍스처 정보 카피 프로세스는 그의 텍스처 정보가 카피되고 있는 항상 계층 화상을 디코딩한 후에 수행될 수도 있다. 하나의 실시형태에서, EL 화상은 그것의 텍스처 정보가 그것의 동일장소에 배치된 BL 화상으로 카피되기 전에 리샘플링될 수도 있다. 리샘플링은 BL과 EL 사이의 확장성 비율에 기초할 수도 있다.

[0442] **다른 고려들**

[0443] 여기에 개시된 정보 및 신호들은 임의의 다양한 상이한 기술들 및 기법들을 사용하여 표현될 수도 있다. 예를 들어, 상기 상세한 설명에 걸쳐 참조될 수도 있는 데이터, 명령들, 커맨드들, 정보, 신호들, 비트들, 심볼들, 및 칩들은 전압들, 전류들, 전자기파들, 자기 필드들 또는 입자들, 광학 필드들 또는 입자들, 또는 이들의 임의의 조합에 의해 표현될 수도 있다.

[0444] 여기에 개시된 실시형태들과 관련하여 기술된 여러 예시적인 로직컬 블록들, 모듈들, 회로들 및 알고리즘 단계들은 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양자 모두의 조합들로서 구현될 수도 있다. 하드웨어 및 소프트웨어의 이러한 교환가능성을 명확히 도시하기 위해, 여러 도시된 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들은 일반적으로 그들의 기능성에 의해 상술되었다. 그러한 기능성이 하드웨어로서 또는 소프트웨어로서 구현되는지 여부는 특정의 애플리케이션 및 전체 시스템에 부과된 설계 제약들에 달려 있다. 통상의 기술자들은 각각의 특정의 애플리케이션에 대해 다양한 방식으로 기술된 기능성을 구현할 수도 있지만, 그러한 구현 결정들은 본 개시의 범위로부터 이탈을 야기하는 것으로서 해석되지 않아야 한다.

[0445] 여기에 기술된 기법들은 하드웨어, 소프트웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 그러한 기법들은 범용 컴퓨터들, 무선 통신 디바이스 핸드셋들, 또는 무선 통신 디바이스 핸드셋들 및 다른 디바이스들 내의 애플리케이션을 포함하여 다수의 용도들을 갖는 집적 회로 디바이스들과 같은 임의의 다양한 디바이스들에서 구현될 수도 있다. 모듈들 또는 컴포넌트들로서 기술된 임의의 특징들은 통합된 로직 디바이스에서 함께 또는 이산 그러나 상호동작가능한 로직 디바이스들로서 별개로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기법들은 실행될 때 상술된 방법들의 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 부분을 형성할 수도 있다. 컴퓨터 판독가능 매체는 동적 랜덤 액세스 메모리 (SDRAM)과 같은 랜덤 액세스 메모리 (RAM), 리드 오직 메모리 (ROM), 비휘발성 랜덤 액세스 메모리 (NVRAM), 전기적으로 소거가능한 프로그램가능한 리드 오직 메모리 (EEPROM), 플래시 메모리, 자기 또는 광학 데이터 저장 매체들 등과 같은 메모리 또는 데이터 저장 매체들을 포함할 수도 있다. 기법들은 추가적으로 또는 대안적으로 명령들 또는 데이터 구조들의 형태의 프로그램 코드를 반송하거나 통신하고 전파된 신호들 또는 전파들과 같은 컴퓨터에 의해 액세스, 판독 및/또는 실행될 수 있는 컴퓨터 판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

[0446] 프로그램 코드는 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로를 포함할 수도 있는 프로세서에 의해 실행될 수도 있다. 그러한 프로세서는 본 개시에 기술된 임의의 기법들을 수행하도록 구성될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만; 대안적으로, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예를 들어 DSP 및 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 결합한 하나 이상의 마이크로프로세서들, 또는 임의의 다른 그러한 구성으로서 구현될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서"는 임의의 상술된 구조, 상술된 구조의 임의의 조합 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 또는 장치를 지칭할 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 비디

오 인코더-디코더 (코덱) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있을 것이다.

[0447]

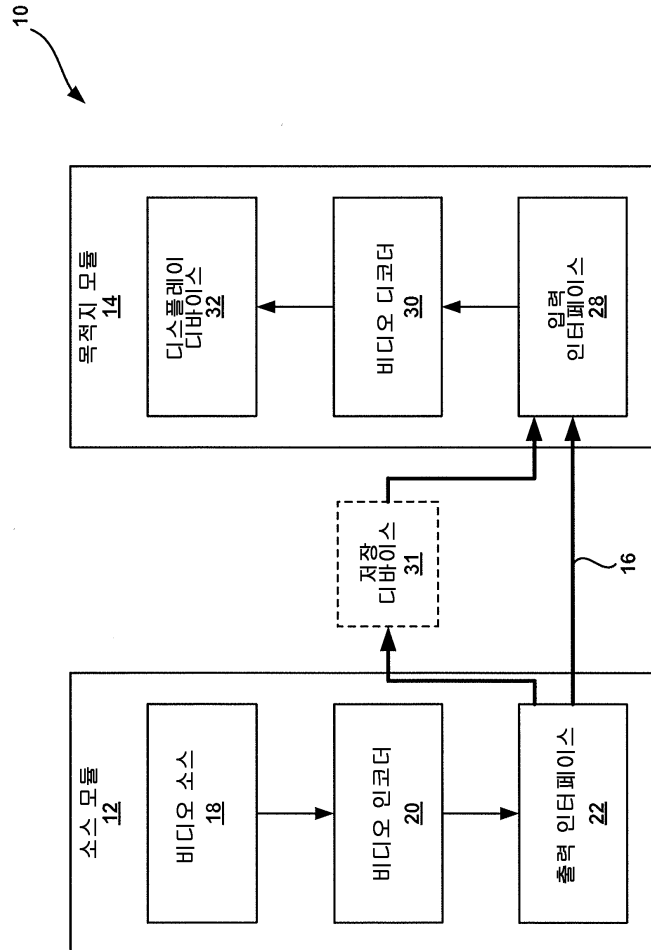
본 개시의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC 들의 세트 (예를 들어, 칩셋) 를 포함하여, 광범위한 디바이스들 장치들에서 구현될 수도 있다. 여러 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시에서 기술되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 상술된 바와 같이, 여러 유닛들은 코덱 하드웨어 유닛에 결합되거나 적합한 소프트웨어 및/또는 펌웨어와 협력하여 상술된 하나 이상의 프로세서들을 포함하는 상호 동작하는 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

[0448]

본 발명의 여러 실시형태들이 기술되었다. 이들 및 다른 실시형태들은 다음의 청구범위의 범위 내에 있다.

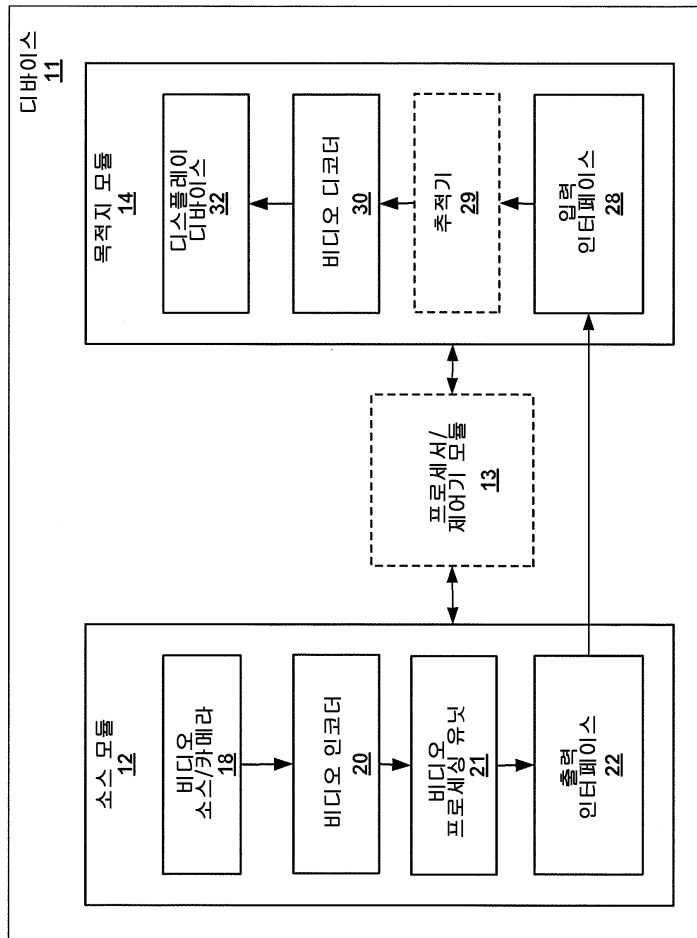
도면

도면1a

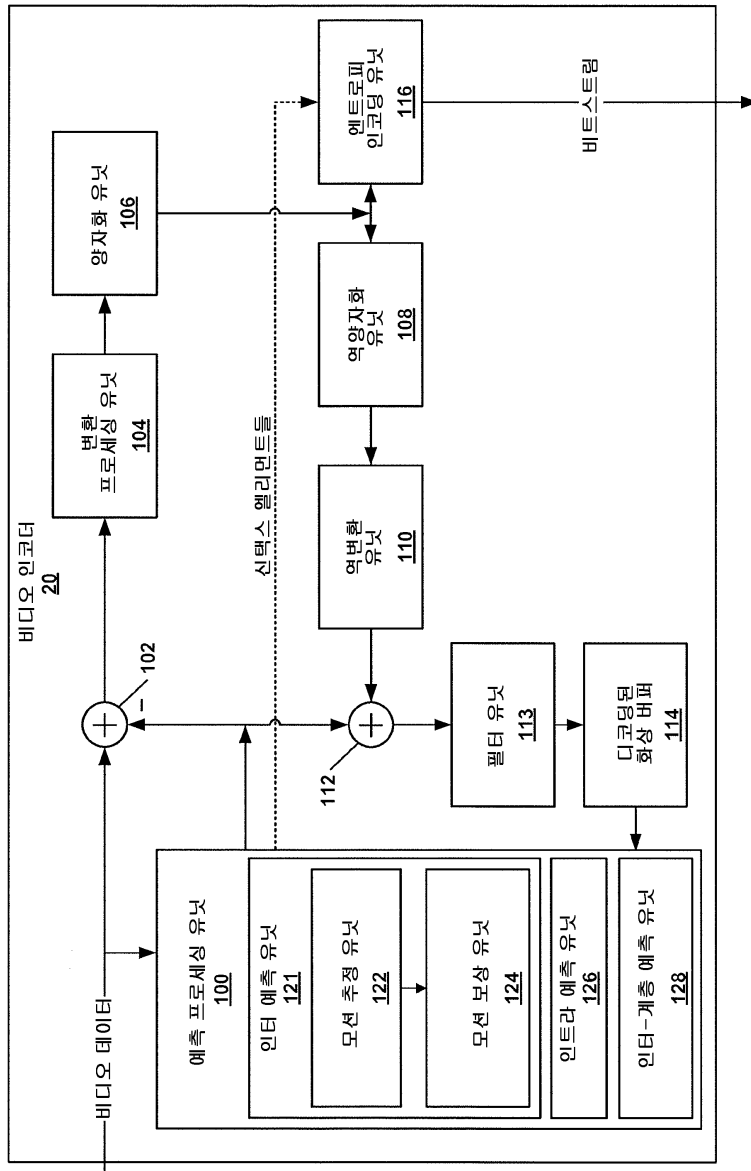


도면1b

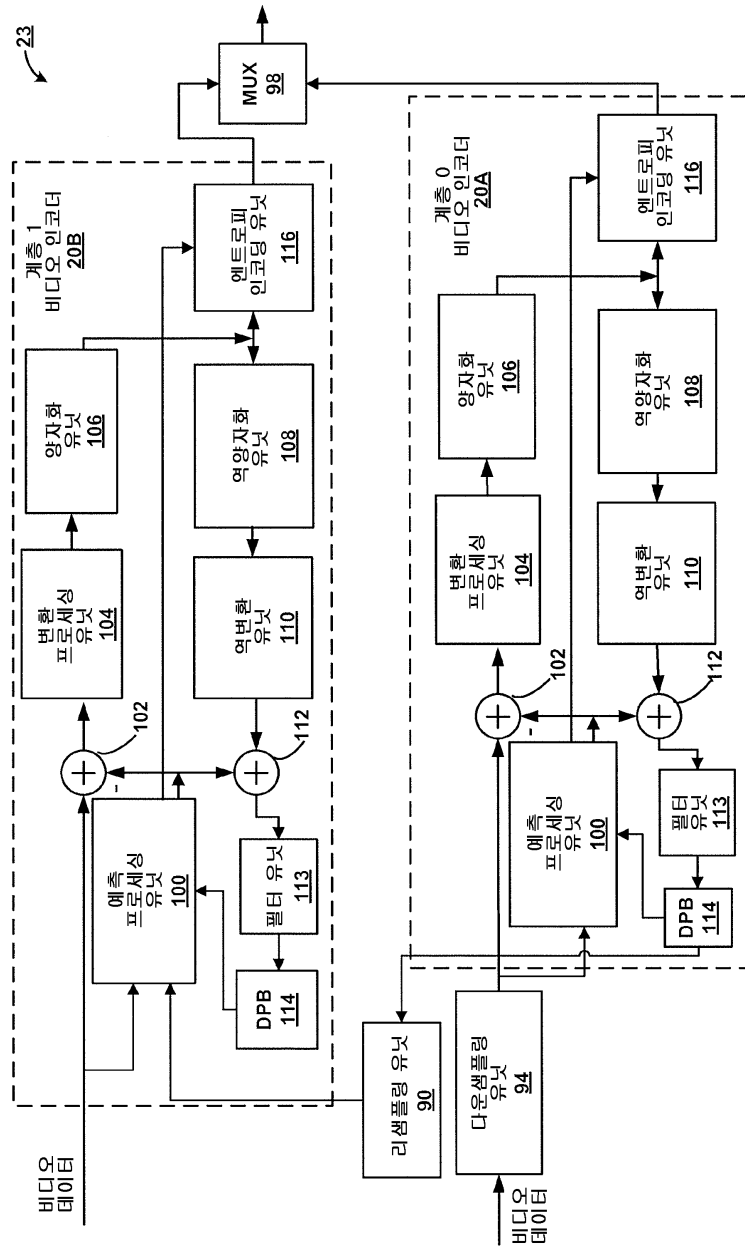
10'



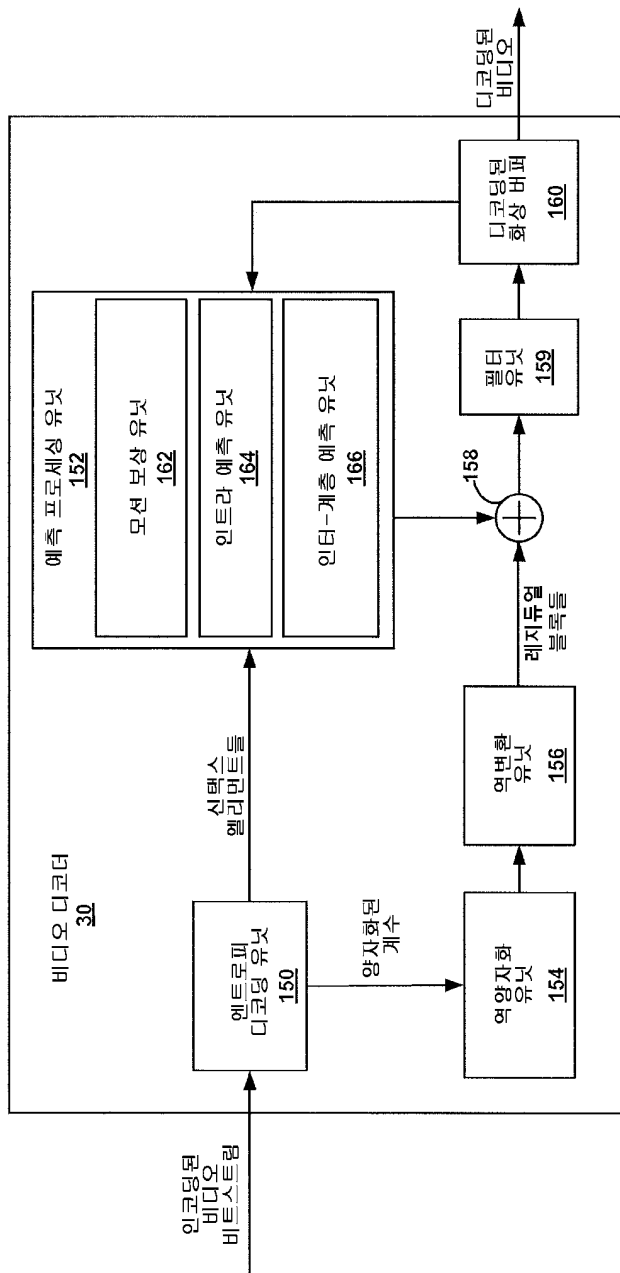
도면2a



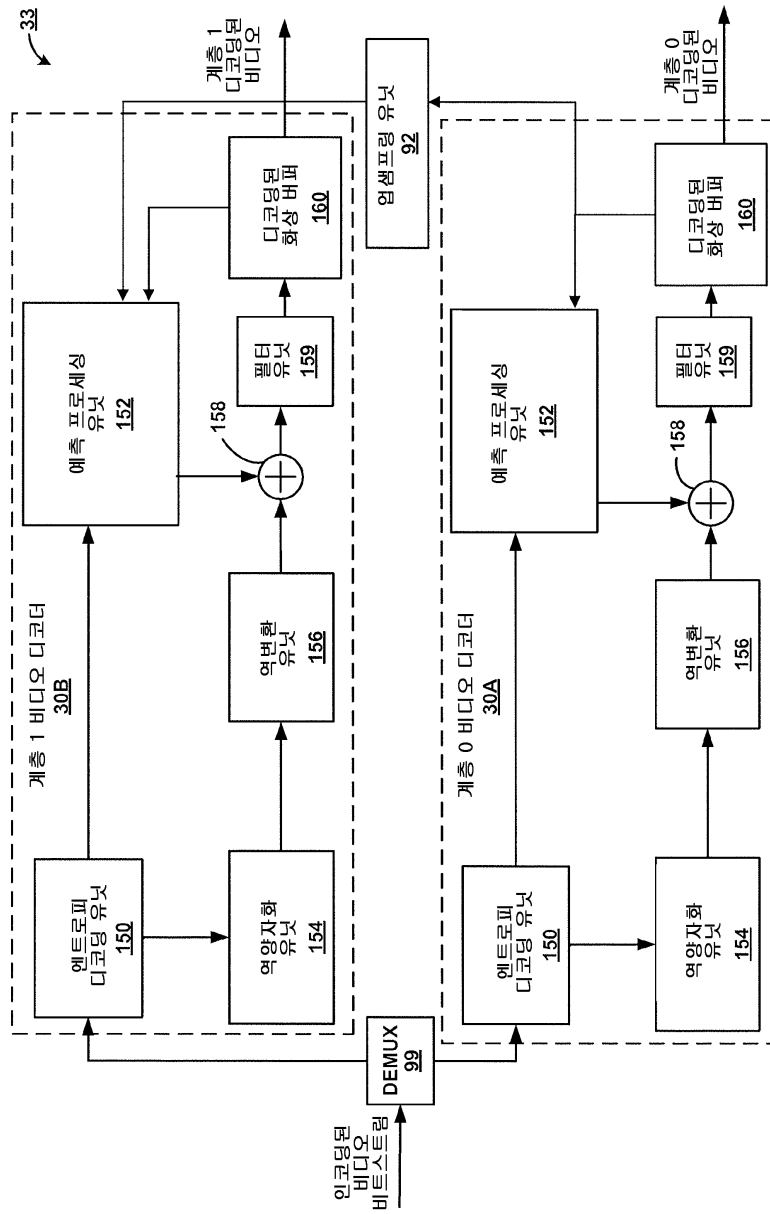
도면2b



도면3a



도면3b



도면4

