



등록특허 10-2295128



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년08월27일
 (11) 등록번호 10-2295128
 (24) 등록일자 2021년08월24일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/30 (2014.01) *HO4N 19/463* (2014.01)
HO4N 19/573 (2014.01) *HO4N 19/70* (2014.01)
- (52) CPC특허분류
HO4N 19/30 (2015.01)
HO4N 19/463 (2015.01)
- (21) 출원번호 10-2015-7031208
- (22) 출원일자(국제) 2014년04월02일
 심사청구일자 2019년03월15일
- (85) 번역문제출일자 2015년10월29일
- (65) 공개번호 10-2015-0139894
- (43) 공개일자 2015년12월14일
- (86) 국제출원번호 PCT/US2014/032718
- (87) 국제공개번호 WO 2014/168799
 국제공개일자 2014년10월16일
- (30) 우선권주장
 61/809,813 2013년04월08일 미국(US)
 14/242,526 2014년04월01일 미국(US)

(56) 선행기술조사문헌

KR1020120013966 A*

Y-K. Wang, et al. AHG9: Signalling and derivation of inter-layer RPS for HEVC 3DV and scalable extensions. JCT-VC of ITU-T and ISO/IEC. JCTVC-M0269 Ver.1, Apr. 9, 2013, pp.1-5

Y-K. Wang, et al. MV-HEVC/SHVC HLS: Signalling and derivation of inter-layer RPS for HEVC 3DV and scalable extensions. JCT-3V of ITU-T and ISO/IEC. JCT3V-D0048 Ver.1, Apr. 10, 2013, pp.1-5

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 41 항

심사관 : 이상래

(54) 발명의 명칭 계층간 픽처 시그널링 및 관련된 프로세스들

(57) 요 약

하나의 구현에서는, 비디오 정보를 인코딩하거나 디코딩하기 위한 장치가 제공된다. 장치는 코딩되고 있는 현재 픽처와 연관된 계층간 참조 픽처들을 저장하도록 구성된 메모리를 포함한다. 장치는 메모리에 동작 가능하게 결합된 프로세서를 더 포함한다. 하나의 실시형태에서, 프로세서는 계층간 예측을 이용하여 현재 픽처

(뒷면에 계속)

대 표 도

를 예측하기 위해 이용할 계층간 참조 팩처들의 수를 표시하도록 구성된다. 프로세서는 또한, 계층간 예측을 이용하여 현재 팩처를 예측하기 위해 상기 계층간 참조 팩처들 중 어느 것을 이용할 것인지를 표시하도록 구성된다. 프로세서는 또한, 계층간 참조 팩처들의 수의 표시, 및 계층간 예측을 이용하여 현재 팩처를 예측하기 위해 계층간 참조 팩처들 중 어느 것을 이용할 것인지의 표시를 이용하여, 현재 팩처와 연관된 계층간 참조 팩처 세트를 결정하도록 구성된다.

(52) CPC특허분류

HO4N 19/573 (2015.01)

HO4N 19/70 (2015.01)

(72) 발명자

라파카 크리쉬나칸트

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

천 잉

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

비디오 정보를 코딩하도록 구성된 장치로서,

현재 계층에서의 하나 이상의 픽처들을 예측하는데 이용되는 계층간 참조 픽처들을 저장하도록 구성된 메모리; 및

상기 메모리에 동작가능하게 결합되고,

계층간 예측을 이용하여 상기 현재 계층에서의 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처들의 수를 결정하고;

결정된 상기 수에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 사용되는 하나 이상의 계층간 참조 픽처들의 각각에 대응하는 참조 계층의 식별을 포함하는 참조 계층들의 리스트를 결정하며;

결정된 상기 참조 계층들의 리스트에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처 세트를 결정하도록 구성된

프로세서를 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 2

제 1 항에 있어서,

상기 프로세서는 또한, 상기 계층간 참조 픽처들의 수, 또는 상기 참조 계층들의 리스트, 또는 양자 모두에 기초하여, 상기 계층간 참조 픽처 세트에서의 참조된 하나 이상의 픽처들의 순서를 결정하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 3

제 1 항에 있어서,

상기 프로세서는 또한, 상기 현재 픽처와 연관된 슬라이스 헤더에서 상기 현재 픽처와 연관된 0 또는 1 개의 직접 종속 계층을 시그널링하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 4

제 1 항에 있어서,

상기 프로세서는 또한, 상기 계층간 참조 픽처 세트를 결정하는 것에 앞서, 상기 계층간 참조 픽처 세트를 비우도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 5

제 1 항에 있어서,

계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 상기 계층간 참조 픽처들의 수는 0 과 상기 현재 픽처와 연관된 직접 참조 계층들의 수와의 사이인, 비디오 정보를 코딩하도록 구성된 장치.

청구항 6

제 1 항에 있어서,

상기 프로세서는 또한, 상기 현재 픽처를 예측하는데 이용되는 상기 계층간 참조 픽처들의 결정된 상기 수와 같은 수의 계층간 예측을 위한 계층간 참조 픽처들을 이용하기 위하여 상기 현재 픽처의 모든 슬라이스들을 요구하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 7

제 1 항에 있어서,

상기 프로세서는 또한, 상기 현재 픽처를 예측하는데 사용되는 상기 계층간 참조 픽처들의 수가 0 과 동일하다는 결정에 기초하여, 상기 현재 픽처가 계층간 예측을 이용하지 않으면서 예측된다는 것을 결정하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 8

제 1 항에 있어서,

상기 프로세서는 또한, 상기 수를 0 또는 1 중 어느 하나이도록 제한하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 9

제 1 항에 있어서,

상기 프로세서는 또한, 슬라이스 헤더에서 상기 현재 픽처를 예측하기 위해 이용되는 상기 계층간 참조 픽처들의 수를 시그널링하거나 또는 수신하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 10

제 1 항에 있어서,

상기 프로세서는 또한, 슬라이스 헤더에서 상기 참조 계층들의 리스트에서 하나 이상의 참조 계층들을 표시하는 표시를 시그널링하거나 수신하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 11

제 1 항에 있어서,

상기 프로세서는 상기 계층간 참조 픽처 세트에 기초하여 계층간 예측을 이용하여 상기 현재 픽처를 인코딩하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 12

제 1 항에 있어서,

상기 프로세서는 상기 계층간 참조 픽처 세트에 기초하여 계층간 예측을 이용하여 상기 현재 픽처를 디코딩하도록 구성되는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 13

제 1 항에 있어서,

디지털 텔레비전, 디지털 직접 방송 시스템, 무선 방송 시스템, 개인 정보 단말 (PDA), 램프 컴퓨터, 데스크톱 컴퓨터, 태블릿 컴퓨터, 전자책 리더, 디지털 카메라, 디지털 레코딩 디바이스, 디지털 미디어 플레이어, 비디오 게임용 디바이스, 비디오 게임 콘솔, 셀룰러 전화, 위성 라디오 전화, 스마트폰, 화상 원격회의 디바이스, 및 비디오 스트리밍 디바이스로 이루어지는 그룹으로부터 선택된 디바이스를 더 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 14

비디오 정보를 디코딩하는 방법으로서,

현재 계층에서의 하나 이상의 픽처들을 예측하는데 이용되는 계층간 참조 픽처들을 저장하는 단계;

계층간 예측을 이용하여 상기 계층에서의 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처들의 수를 결정하는 단계;

결정된 상기 수에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 사용되는 하나 이상의 계층

간 참조 픽처들의 각각에 대응하는 참조 계층의 식별을 포함하는 참조 계층들의 리스트를 결정하는 단계;
 결정된 상기 참조 계층들의 리스트에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처 세트를 결정하는 단계; 및
 상기 계층간 참조 픽처 세트에 기초하여 계층간 예측을 이용하여 상기 현재 픽처를 디코딩하는 단계를 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 15

제 14 항에 있어서,

상기 계층간 참조 픽처들의 수, 또는 상기 참조 계층들의 리스트, 또는 양자 모두에 기초하여, 상기 계층간 참조 픽처 세트에서의 참조된 하나 이상의 픽처들의 순서를 결정하는 단계를 더 포함하는, 비디오 정보를 디코딩 하는 방법.

청구항 16

제 14 항에 있어서,

상기 현재 픽처와 연관된 슬라이스 헤더에서 상기 현재 픽처와 연관된 0 또는 1 개의 직접 종속 계층을 시그널 하는 단계를 더 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 17

제 14 항에 있어서,

상기 계층간 참조 픽처 세트를 결정하는 단계에 앞서, 상기 계층간 참조 픽처 세트를 비우는 단계를 더 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 18

제 14 항에 있어서,

계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 상기 계층간 참조 픽처들의 수는 0 과 상기 현재 픽처와 연관된 직접 참조 계층들의 수와의 사이인, 비디오 정보를 디코딩하는 방법.

청구항 19

제 14 항에 있어서,

상기 현재 픽처를 예측하는데 이용되는 상기 계층간 참조 픽처들의 결정된 상기 수와 같은 수의 계층간 예측을 위한 계층간 참조 픽처들을 이용하기 위하여 상기 현재 픽처의 모든 슬라이스들을 요구하는 단계를 더 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 20

제 14 항에 있어서,

상기 현재 픽처를 예측하는데 사용되는 상기 계층간 참조 픽처들의 수가 0 과 동일하다는 결정에 기초하여, 상기 현재 픽처가 계층간 예측을 이용하지 않으면서 예측된다는 것을 결정하는 단계를 더 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 21

제 14 항에 있어서,

상기 수를 0 또는 1 중 어느 하나이도록 제한하는 단계를 더 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 22

제 14 항에 있어서,

슬라이스 헤더에서 상기 현재 픽처를 예측하기 위해 이용되는 상기 계층간 참조 픽처들의 수를 수신하는 단계를

더 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 23

제 14 항에 있어서,

슬라이스 헤더에서 상기 참조 계층들의 리스트에서 하나 이상의 참조 계층들을 표시하는 표시를 수신하는 단계를 더 포함하는, 비디오 정보를 디코딩하는 방법.

청구항 24

비디오 정보를 인코딩하는 방법으로서,

현재 계층에서의 하나 이상의 픽처들을 예측하는데 이용되는 계층간 참조 픽처들을 저장하는 단계;

계층간 예측을 이용하여 상기 현재 계층에서의 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처들의 수를 결정하는 단계;

결정된 상기 수에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 사용되는 하나 이상의 계층간 참조 픽처들의 각각에 대응하는 참조 계층의 식별을 포함하는 참조 계층들의 리스트를 결정하는 단계;

결정된 상기 참조 계층들의 리스트에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처 세트를 결정하는 단계; 및

상기 계층간 참조 픽처 세트에 기초하여 계층간 예측을 이용하여 상기 현재 픽처를 인코딩하는 단계를 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 25

제 24 항에 있어서,

상기 계층간 참조 픽처들의 수, 또는 상기 참조 계층들의 리스트, 또는 양자 모두에 기초하여, 상기 계층간 참조 픽처 세트에서의 참조된 하나 이상의 픽처들의 순서를 결정하는 단계를 더 포함하는, 비디오 정보를 인코딩 하는 방법.

청구항 26

제 24 항에 있어서,

상기 현재 픽처와 연관된 슬라이스 헤더에서 상기 현재 픽처와 연관된 0 또는 1 개의 직접 종속 계층을 시그널 하는 단계를 더 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 27

제 24 항에 있어서,

상기 계층간 참조 픽처 세트를 결정하는 단계에 앞서, 상기 계층간 참조 픽처 세트를 비우는 단계를 더 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 28

제 24 항에 있어서,

계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 상기 계층간 참조 픽처들의 수는 0 과 상기 현재 픽처와 연관된 직접 참조 계층들의 수와의 사이인, 비디오 정보를 인코딩하는 방법.

청구항 29

제 24 항에 있어서,

상기 현재 픽처를 예측하는데 이용되는 상기 계층간 참조 픽처들의 결정된 상기 수와 같은 수의 계층간 예측을 위한 계층간 참조 픽처들을 이용하기 위하여 상기 현재 픽처의 모든 슬라이스들을 요구하는 단계를 더 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 30

제 24 항에 있어서,

상기 현재 픽처를 예측하는데 사용되는 상기 계층간 참조 픽처들의 수가 0 과 동일하다는 결정에 기초하여, 상기 현재 픽처가 계층간 예측을 이용하지 않으면서 예측된다는 것을 결정하는 단계를 더 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 31

제 24 항에 있어서,

상기 수를 0 또는 1 중 어느 하나이도록 제한하는 단계를 더 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 32

제 24 항에 있어서,

슬라이스 헤더에서 상기 현재 픽처를 예측하기 위해 이용되는 상기 계층간 참조 픽처들의 수를 시그널링하는 단계를 더 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 33

제 24 항에 있어서,

슬라이스 헤더에서 상기 참조 계층들의 리스트에서 하나 이상의 참조 계층들을 표시하는 표시를 시그널링하는 단계를 더 포함하는, 비디오 정보를 인코딩하는 방법.

청구항 34

비디오 정보를 코딩하도록 구성된 장치로서,

현재 계층에서의 하나 이상의 픽처들을 예측하는데 이용되는 계층간 참조 픽처들을 저장하기 위한 수단;

계층간 예측을 이용하여 상기 현재 계층에서의 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처들의 수를 결정하기 위한 수단;

결정된 상기 수에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 사용되는 하나 이상의 계층간 참조 픽처들의 각각에 대응하는 참조 계층의 식별을 포함하는 참조 계층들의 리스트를 결정하기 위한 수단;

결정된 상기 참조 계층들의 리스트에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처 세트를 결정하기 위한 수단; 및

상기 계층간 참조 픽처 세트에 기초하여 계층간 예측을 이용하여 상기 현재 픽처를 코딩하기 위한 수단을 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 35

제 34 항에 있어서,

상기 계층간 참조 픽처들의 수, 또는 상기 참조 계층들의 리스트, 또는 양자 모두에 기초하여, 상기 계층간 참조 픽처 세트에서의 참조된 하나 이상의 픽처들의 손실을 결정하기 위한 수단을 더 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 36

제 34 항에 있어서,

상기 현재 픽처와 연관된 슬라이스 헤더에서 상기 현재 픽처와 연관된 0 또는 1 개의 직접 종속 계층을 시그널링하기 위한 수단을 더 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 37

제 34 항에 있어서,

상기 계층간 참조 픽처 세트를 결정하는 것에 앞서, 상기 계층간 참조 픽처 세트를 비우기 위한 수단을 더 포함하는, 비디오 정보를 코딩하도록 구성된 장치.

청구항 38

특정 명령을 포함하는 비일시적 컴퓨터 판독가능 매체로서,

상기 명령은, 컴퓨팅 하드웨어를 포함하는 프로세서 상에서 실행되는 경우, 상기 프로세서로 하여금,

현재 계층에서의 하나 이상의 픽처들을 예측하는데 이용되는 계층간 참조 픽처들을 저장하게 하며;

계층간 예측을 이용하여 상기 현재 계층에서의 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처들의 수를 결정하게 하고;

결정된 상기 수에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 사용되는 하나 이상의 계층간 참조 픽처들의 각각에 대응하는 참조 계층의 식별을 포함하는 참조 계층들의 리스트를 결정하게 하고;

결정된 상기 참조 계층들의 리스트에 기초하여, 계층간 예측을 이용하여 상기 현재 픽처를 예측하는데 이용되는 계층간 참조 픽처 세트를 결정하게 하며;

상기 계층간 참조 픽처 세트에 기초하여 계층간 예측을 이용하여 상기 현재 픽처를 코딩하게 하는, 비일시적 컴퓨터 판독가능 매체.

청구항 39

제 38 항에 있어서,

상기 프로세서로 하여금, 상기 계층간 참조 픽처들의 수, 또는 상기 참조 계층들의 리스트, 또는 양자 모두에 기초하여, 상기 계층간 참조 픽처 세트에서의 참조된 하나 이상의 픽처들의 손실을 결정하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 매체.

청구항 40

제 38 항에 있어서,

상기 프로세서로 하여금, 상기 현재 픽처와 연관된 슬라이스 헤더에서 상기 현재 픽처와 연관된 0 또는 1 개의 직접 종속 계층을 시그널링하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 매체.

청구항 41

제 38 항에 있어서,

상기 프로세서로 하여금, 상기 계층간 참조 픽처 세트를 결정하는 것에 앞서, 상기 계층간 참조 픽처 세트를 비우게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 매체.

발명의 설명

기술 분야

[0001]

분야

이 개시물은 비디오 코딩 및 압축의 분야에 관련된다. 특히, 그것은 진보된 비디오 코딩 (Advanced Video Coding; AVC) 을 위한 SVC 를 포함하는 스케일러블 비디오 코딩 (scalable video coding; SVC) 뿐만 아니라, 스케일러블 HEVC (Scalable HEVC; SHVC) 로서 또한 지칭되는 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 을 위한 SVC 에 관련된다. 그것은 또한, MV-HEVC 로서 지칭된, HEVC 의 멀티뷰 확장과 같은 3D 비디오 코딩에 관련된다. 다양한 실시형태들은 개선된 계층간 예측 시그널링 및 관련된 프로세스들 (예컨대, 계층간 참조 픽처 세트들의 유도, 참조 픽처 리스트들의 유도 등) 을 위한 시스템들 및 방법들과 관련 있다.

배경 기술

[0003]

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

[0004]

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

[0005]

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

[0006]

어떤 비디오 코딩 기법들은 또 다른 계층에서의 비디오 정보의 값을 예측하기 위해 하나 이상의 계층들로부터의 비디오 정보를 사용한다. 이러한 예측은 일반적으로 계층간 예측 (inter-layer prediction; ILP) 으로서 지칭된다. 일반적으로, 계층들은 동일한 액세스 유닛에서 상주한다. 일부의 경우들에는, 예측되고 있는 비디오 블록이 예측을 수행하기 위해 이용된 비디오 정보를 포함하는 계층보다 더 높다. 예를 들어, 일부의 경우들에는, 예측되고 있는 비디오 블록 (예컨대, 현재 블록) 이 (이하에서 논의된 바와 같은) 강화 계층 (enhancement layer) 에서 상주하고, 현재 블록을 예측하기 위해 이용된 비디오 정보는 더 낮은 강화 계층 또는 기본 계층에서 상주한다.

[0007]

고효율 비디오 코딩 (HEVC) 은 이러한 계층간 예측을 위한 기법들을 제공한다. 그러나, 현재의 기법들은 코딩 성능을 제한하는 다양한 비효율성들을 겪고 있다. 예를 들어, 현재의 기법들을 이용하면, ILP 가 픽처에 대해 이용되지 않을 경우, 또는 ILP 가 이용되지만 하나의 계층간 참조 픽처 (inter-layer reference picture; ILRP) 만이 ILP 동안에 이용되도록 허용될 경우에는, 코딩 디바이스들 (예컨대, 인코더들, 디코더들) 이 참조 픽처 리스트 수정 선택스 엘리먼트들을 이용해야 할 것이다. 이러한 선택스 엘리먼트들은 비트들을 희생시키고, 그러므로, 프로세싱 및 코딩 효율에 부정적으로 영향을 줄 수 있다.

[0008]

게다가, 현재의 기법들을 이용하면, 일부의 상황들에서, 참조 픽처 서브세트의 유도가 계층간 예측이 이용될 때에 적당하게 발생하지 않을 수도 있다. 예를 들어, 하나의 액세스 유닛에서, 현재의 강화 계층의 픽처가 현재의 강화 계층의 직접 종속 계층인 계층에 대한 픽처를 가지지 않을 경우, 계층간 참조 픽처 세트 (reference picture set; RPS) 서브세트 (예컨대, RefPicSetInterLayer) 의 현재의 유도 프로세스가 마찬가지로 작동하지 않는다. 이것은 현재의 유도 프로세스가 모든 직접 종속 계층들의 모든 픽처들이 존재하는 것으로 가정하기 때문이다. 특히, 디코더들이 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 에서의 어떤 픽처에도 대

응하지 않는 계층간 RPS 서브세트 (예컨대, RefPicSet InterLayer)에서의 엔트리가 송신 동안에 손실되는지 여부, 또는 그것이 원래의 비트스트림에서 없었는지를 알기 위한 방법이 현재에는 없다.

[0009] 본원에서 설명된 기법들은 이러한 기법들과 관련 있는 이러한 그리고 다른 쟁점들을 다룬다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0010] 일반적으로, 이 개시물은 스케일러블 비디오 코딩 (SVC)과 관련된 기법들을 설명한다. 이하에서 설명된 다양한 기법들은 계층간 예측 시그널링 및 관련된 프로세스들을 위한 방법들 및 디바이스들을 설명한다.

[0011] 하나의 구현에서는, 비디오 정보를 인코딩하거나 디코딩하기 위한 장치가 제공된다. 장치는 기본 계층, 강화 계층, 또는 양자 모두와 연관된 비디오 정보 및/또는 참조 계층 픽처들을 저장하도록 구성된 메모리를 포함한다. 장치는 메모리에 동작가능하게 결합된 프로세서를 더 포함한다. 프로세서는 강화 계층에서 비디오 유닛의 값을 결정하기 위해 계층간 참조 픽처로서의 최대로 하나의 참조 계층 픽처의 사용을 제한하도록 구성된다.

[0012] 하나의 실시형태에서는, 비디오 정보를 인코딩하거나 디코딩하기 위한 장치가 제공된다. 장치는 코딩되고 있는 현재 픽처와 연관된 계층간 참조 픽처들을 저장하도록 구성된 메모리를 포함한다. 장치는 또한 메모리에 동작가능하게 결합된 프로세서를 포함한다. 하나의 실시형태에서, 프로세서는 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 이용할 계층간 참조 픽처들의 수를 표시하도록 구성된다. 프로세서는 또한, 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시하도록 구성된다. 프로세서는 또한, 계층간 참조 픽처들의 수의 표시, 및 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시를 이용하여, 현재 픽처와 연관된 계층간 참조 픽처 세트를 결정하도록 구성된다.

[0013] 또 다른 실시형태에서는, 비디오 정보를 디코딩하는 방법이 제공된다. 방법은, 코딩되고 있는 현재 픽처와 연관된 계층간 참조 픽처들을 저장하는 단계; 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 이용할 계층간 참조 픽처들의 수를 표시하는 단계; 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시하는 단계; 계층간 참조 픽처들의 수의 표시, 및 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시를 이용하여, 현재 픽처와 연관된 계층간 참조 픽처 세트를 결정하는 단계; 및 계층간 참조 픽처 세트 및 계층간 예측을 이용하여 현재 픽처를 디코딩하는 단계를 포함한다.

[0014] 또 다른 실시형태에서는, 비디오 정보를 인코딩하는 방법이 제공된다. 방법은, 코딩되고 있는 현재 픽처와 연관된 계층간 참조 픽처들을 저장하는 단계; 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 이용할 계층간 참조 픽처들의 수를 표시하는 단계; 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시하는 단계; 계층간 참조 픽처들의 수의 표시, 및 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시를 이용하여, 현재 픽처와 연관된 계층간 참조 픽처 세트를 결정하는 단계; 및 계층간 참조 픽처 세트 및 계층간 예측을 이용하여 현재 픽처를 인코딩하는 단계를 포함한다.

[0015] 또 다른 실시형태에서는, 비디오 정보를 코딩하도록 구성된 장치가 제공된다. 장치는, 코딩되고 있는 현재 픽처와 연관된 계층간 참조 픽처들을 저장하기 위한 수단; 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 이용할 계층간 참조 픽처들의 수를 표시하기 위한 수단; 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시하기 위한 수단; 계층간 참조 픽처들의 수의 표시, 및 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시를 이용하여, 현재 픽처와 연관된 계층간 참조 픽처 세트를 결정하기 위한 수단; 및 계층간 참조 픽처 세트 및 계층간 예측을 이용하여 현재 픽처를 코딩하기 위한 수단을 포함한다.

[0016] 또 다른 실시형태에서는, 비일시적 컴퓨터 판독가능 매체가 제공된다. 비일시적 컴퓨터 판독가능 매체는 특정 명령을 포함하고, 상기 명령은, 컴퓨팅 하드웨어를 포함하는 프로세서 상에서 실행되는 경우, 상기 프로세서

로 하여금, 코딩되고 있는 현재 픽처와 연관된 계층간 참조 픽처들을 저장하게 하고; 계층간 예측을 이용하여 상기 현재 픽처를 예측하기 위해 이용할 계층간 참조 픽처들의 수를 표시하게 하고; 계층간 예측을 이용하여 상기 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시하게 하고; 상기 계층간 참조 픽처들의 수의 상기 표시, 및 계층간 예측을 이용하여 상기 현재 픽처를 예측하기 위해 상기 계층간 참조 픽처들 중 어느 것을 이용할 것인지를 표시하는 상기 표시를 이용하여, 상기 현재 픽처와 연관된 계층간 참조 픽처 세트를 결정하게 하며; 상기 계층간 참조 픽처 세트 및 계층간 예측을 이용하여 상기 현재 픽처를 코딩하게 한다.

[0017] 하나 이상의 예들의 세부사항들은 첨부한 도면들 및 이하의 설명에서 기재되어 있고, 이들은 본원에서 설명된 발명 개념들의 전체 범위를 제한하도록 의도된 것이 아니다. 다른 특징들, 목적들, 및 장점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0018] 도면들 전반에 걸쳐, 참조된 엘리먼트들 사이의 대응관계를 표시하기 위해 참조 번호들이 재이용될 수도 있다. 도면들은 본원에서 설명된 일 예의 실시형태들을 예시하도록 제공되고, 개시물의 범위를 제한하도록 의도된 것은 아니다.

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

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

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

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

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

도 4 내지 도 6 은 이 개시물의 양태들에 따라 계층간 예측 시그널링의 방법들의 실시형태들을 예시하는 플로우 차트들이다.

발명을 실시하기 위한 구체적인 내용

[0019] 이 개시물에서 설명된 기법들은 일반적으로 스케일러블 비디오 코딩 (SHVC, SVC) 및 멀티뷰/3D 비디오 코딩 (예컨대, 멀티뷰 코딩 플러스 심도, MVC+D) 과 관련 있다. 예를 들어, 기법들은 고효율 비디오 코딩 (HEVC) 스케일러블 비디오 코딩 (SVC, 때때로 SHVC로서 지칭됨) 확장과 관련될 수도 있고, 이 확장과 함께 또는 이 확장 내에서 이용될 수도 있다. SHVC, SVC 확장에서는, 비디오 정보의 다수의 계층들이 있을 수 있다. 바로 하부 레벨에서의 계층은 기본 계층 (base layer; BL) 으로서 작용할 수도 있고, 바로 상부에서의 계층 (또는 최고 계층) 은 강화된 계층 (enhanced layer; EL) 으로서 작용할 수도 있다. "강화된 계층" 은 때때로 "강화 계층" 으로서 지칭되고, 이 용어들은 상호 교환가능하게 이용될 수도 있다. 기본 계층은 때때로 "참조 계층" (reference layer; RL) 으로서 지칭되고, 이 용어들도 상호 교환가능하게 이용될 수도 있다. 기본 계층 및 상부 계층 사이의 모든 계층들은 EL 들 또는 참조 계층 (RL) 들의 어느 하나 또는 양자 모두로서 작용할 수도 있다. 예를 들어, 중간의 계층은 기본 계층 또는 임의의 중간에 삽입된 강화 계층들과 같은, 그것 아래의 계층들에 대한 EL 일 수도 있고, 이와 동시에, 그것 위의 강화 계층들에 대한 RL 으로서 작용할 수도 있다. 기본 계층 및 상부 계층 (또는 최고 계층) 사이의 각각의 계층은 더 높은 계층에 의한 계층간 예측을 위한 참조로서 이용될 수도 있고, 계층간 예측을 위한 참조로서 더 낮은 계층을 이용할 수도 있다.

[0020] 간략함을 위해, 예들은 단지 2 개의 계층들: BL 및 EL 의 측면에서 제시되어 있지만, 이하에서 설명된 사상들 및 실시형태들은 다수의 계층들을 갖는 경우들에도 마찬가지로 적용가능한 것을 양호하게 이해해야 한다. 게다가, 설명의 용이함을 위해, 용어들 "프레임들" 또는 "블록들" 이 종종 이용된다. 그러나, 이 용어들은 제한하는 것으로 의도된 것은 아니다. 예를 들어, 이하에서 설명된 기법들은 픽셀들, 블록들 (예컨대, CU, PU, TU, 매크로블록들 등), 슬라이스들, 프레임들, 픽처 등을 포함하는 다양한 비디오 유닛들 중의 임의의 것과

함께 이용될 수 있다.

[0021] 비디오 코딩

비디오 코딩 표준들은 그 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (Multiview Video Coding; MVC) 및 멀티뷰 코딩 플러스 심도 (MVC+D) 확장들을 포함하는, ITU-T H.261, ISO/IEC MPEG-1 비주얼 (Visual), 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 AVC로서 알려짐)를 포함한다. 이하에서 HEVC WD10 으로서 지정된 최신 HEVC 초안 사양은 http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip로부터 입수가능하다. HEVC에 대한 멀티뷰 확장, 즉, MV-HEVC는 또한, JCT-3V에 의해 개발되고 있다. 이하의 MV-HEVC WD3의 최근의 작업 초안 (Working Draft; WD)은 http://phenix.int-sudparis.eu/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1004-v4.zip로부터 입수가능하다. SHVC로 명명된, HEVC에 대한 스케일러블 확장은 또한, JCT-VC에 의해 개발되고 있다. SHVC WD1로서 지정되는 SHVC의 최근의 작업 초안 (WD)은 http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1008-v1.zip로부터 입수가능하다.

SVC에서는, 비디오 정보가 다수의 계층들로서 제공될 수도 있다. 바로 하부 레벨에서의 계층은 단지 기본 계층 (BL)으로서 작용할 수 있고, 바로 상부 레벨에서의 계층은 강화 계층 (EL)으로서 작용할 수 있다. 상부 및 하부 계층들 사이의 모든 계층들은 강화 계층들 및 기본 계층들의 양자 모두로서 작용할 수도 있다. 예를 들어, 중간의 계층은 그것 아래의 계층들에 대한 EL, 그리고 이와 동시에, 그것 위의 계층들에 대한 BL로서 작용할 수 있다. 설명의 간략함을 위해, 우리는 이하에서 설명된 기법들을 예시함에 있어서, 2개의 계층들, BL 및 EL이 있는 것으로 간주할 수 있다. 그러나, 본원에서 설명된 모든 기법들은 다수의 (2개를 초과하는) 계층들을 갖는 경우들에 대해 마찬가지로 적용가능하다.

스케일러블 비디오 코딩 (SVC)은 품질 (또한, 신호-대-잡음 (signal-to-noise; SNR)) 스케일러블리티, 공간적 스케일러블리티 및/또는 시간적 스케일러블리티를 제공하기 위해 이용될 수도 있다. 예를 들어, 하나의 실 시형태에서는, 참조 계층 및 강화 계층이 제 1 레벨보다 더 높은 제 2 품질 레벨 (예컨대, 더 적은 잡음, 더 큰 해상도, 더 양호한 프레임 레이트 등)에서 비디오를 디스플레이하기 위해 충분한 비디오 정보를 함께 포함하도록, 참조 계층 (예컨대, 기본 계층)은 제 1 품질 레벨에서 비디오를 디스플레이하기 위해 충분한 비디오 정보를 포함하고 강화 계층은 참조 계층과 관련된 추가적인 비디오 정보를 포함한다. 강화된 계층은 기본 계층과는 상이한 공간적 해상도를 가질 수도 있다. 예를 들어, EL 및 BL 사이의 공간적 종횡비 (aspect ratio)는 1.0, 1.5, 2.0 또는 다른 상이한 비율들일 수 있다. 다시 말해서, EL의 공간적 종횡은 BL의 공간적 종횡의 1.0, 1.5, 또는 2.0 배와 동일할 수도 있다. 일부의 예들에서, EL의 스케일링 인자 (scaling factor)는 BL 보다 더 클 수도 있다. 예를 들어, EL에서의 핵심들의 사이즈는 BL에서의 핵심들의 사이즈 보다 더 클 수도 있다. 이러한 방법으로, 제한은 아니지만, EL의 공간적 해상도가 BL의 공간적 해상도보다 더 크다는 것이 가능할 수도 있다.

H.264에 대한 SVC 확장에서는, 현재 블록의 예측이 SVC에 대해 제공되는 상이한 계층들을 이용하여 수행될 수도 있다. 이러한 예측은 계층간 예측으로서 지정될 수도 있다. 계층간 예측은 계층간 리던던시를 감소시키기 위해 SVC에서 사용될 수도 있다. 계층간 예측의 일부의 예들은 계층간 인트라 예측, 계층간 모션 예측, 및 계층간 잔여 예측을 포함할 수도 있다. 계층간 인트라 예측은 강화 계층에서 현재 블록을 예측하기 위해 기본 계층에서의 공동-위치된 (co-located) 블록들의 복원을 이용한다. 계층간 모션 예측은 강화 계층에서 모션을 예측하기 위해 기본 계층의 모션 정보 (모션 벡터들을 포함함)를 이용한다. 계층간 잔여 예측은 강화 계층의 잔여를 예측하기 위해 기본 계층의 잔여를 이용한다.

계층간 모션 예측의 일부의 실시형태들에서는, 강화 계층에서 현재 블록을 예측하기 위해, (예컨대, 공동-위치된 블록에 대한) 기본 계층의 모션 데이터 (모션 벡터들을 포함함)가 이용될 수도 있다. 예를 들어, 강화 계층에서 비디오 유닛을 코딩하는 동안, 비디오 코더들은 추가적인 가정 (hypothesis)들을 식별하기 위해 이용될 수 있는 추가적인 모션 보상 데이터를 얻기 위해 참조 계층으로부터의 정보를 이용할 수 있다. 이 추가적인 가정들은 비디오 비트스트림에서 이미 존재하는 데이터로부터 목적으로 유도되므로, 비디오 코딩에 있어서의 추가적인 성능은 비트스트림 사이즈에 있어서의 추가적인 비용이 거의 또는 전혀 없이 얻어질 수 있다.

또 다른 예에서, 공간적으로 이웃하는 비디오 유닛들로부터의 모션 정보는 추가적인 가정들을 위치시키기 위해 이용될 수 있다. 다음으로, 유도된 가정은 비디오 유닛의 값의 더 양호한 예측을 생성하기 위해, 평균화될 수 있거나, 이와 다르게 명시적으로 인코딩된 가정과 조합될 수 있다. 기본 (또는 참조) 계층의 공간적

해상도가 현재 블록의 계층의 공간적 해상도와는 상이할 때와 같은 어떤 상황들에서는, 현재 블록을 인코딩하거나 디코딩하기 위해 이용되기 이전에, 기본 계층 모션 정보가 공간적으로 스케일링된다. 마찬가지로, 계층이 현재 블록의 계층과는 상이한 공간적 해상도를 가질 때에는, 기본 (또는 참조) 계층 내의 블록의 위치가 이하에서 설명된 바와 같이 계층간 위치 맵핑에 의해 결정될 수도 있다.

[0027] 비디오 용어

이 개시물의 전반에 걸쳐 이용된 다양한 용어는 그 통상의 의미를 갖는 광범위한 용어들이다. 게다가, 일부의 실시형태들에서, 어떤 용어들은 다음의 비디오 개념들과 관련 있다. 꾹처는 그 용어가 현재의 표준들 (예컨대, HEVC)에서 이용될 때에 비디오 꾹처를 지칭할 수 있다. 코딩된 꾹처는 SVC에서의 계층 표면, MVC에서의 뷰 컴포넌트, 및 MVC+D에서의 텍스처 또는 심도 뷰 컴포넌트를 지칭할 수 있다. SVC 및 MVC에서 이용된 바와 같은 용어와 유사한 액세스 유닛 (AU)은 동일한 출력 시간과 연관된 모든 코딩된 꾹처들 및 그 연관된 비-비디오 코딩 계층 (non-video coding layer; non-VCL) 네트워크 추상 계층 (network abstraction layer; NAL) 유닛들을 지칭할 수 있다. 인트라 랜덤 액세스 포인트 (intra random access point; IRAP) 액세스 유닛은 코딩된 꾹처들의 전부가 IRAP 꾹처들인 액세스 유닛을 지칭할 수 있다. 코딩된 비디오 시퀀스 (coded video sequence; CVS)는, 디코딩 순서에서, 1과 동일한 플래그 NoRas1OutputFlag를 갖는 IRAP 액세스 유닛을 포함하며, 1과 동일한 NoRas1OutputFlag를 갖는 RAP 액세스 유닛인 임의의 후속 액세스 유닛까지의 모든 후속 액세스 유닛들을 포함하지만 이 임의의 후속 액세스 유닛을 포함하지는 않는, 액세스 유닛들의 시퀀스를 지칭할 수 있다. 게다가, 비트스트림은 하나 이상의 CVS들의 표현을 형성하는 NAL 유닛 스트림 또는 바이트 스트림의 형태인 비트들의 시퀀스를 지칭할 수도 있다. 비트스트림에서의 최초 액세스 유닛은 위에서 설명된 바와 같이 IRAP 액세스 유닛이다.

[0029] 개요

본원에서 설명된 실시형태들은 위에서 논의된 것들을 포함하는, 현재의 HEVC 확장 (예컨대, SHVC로서 알려진 HEVC에 대한 스케일러블 확장)으로 ILP를 구현하는 것과 연관된 문제들을 다룬다. 예를 들어, 인코딩 및 디코딩 디바이스들 및 방법들의 다양한 실시형태들은 다음의 하나 이상을 포함한다: (1) 계층간 꾹처 참조로서 이용되는 꾹처들의 손실을 검출하기 위해 이용될 수 있도록, 계층간 RPS 서브세트의 시그널링 및 유도; (2) 최고 nuh_layer_id를 갖는 종속 계층이 계층간 예측을 위해 이용된 고유의 계층인지 여부를 표시하는 것; (3) 슬라이스 (또는 세그먼트) 헤더에서의 전혀 없거나 (즉, 제로) 또는 단지 하나인 직접 종속 계층의 시그널링; (4) 초기 참조 꾹처 리스트에서 하나의 계층간 참조를 포함하기만 하는 옵션을 가능하게 하기 위해 참조 꾹처 리스트 초기화를 변화시키는 것; 및/또는 (5) 계층간 참조 꾹처가 비워지도록 하고, 그것이 최종 참조 꾹처 리스트에 있지 않기만 하면, 초기 참조 꾹처 리스트 새로 추가되도록 하는 것. 비워질 때, 꾹처는 "참조 꾹처 아님"으로서 계층간 RPS 새로 추가된다.

본원에서 설명된 기법들의 다수는 다양한 표준들 (또는 그 초안들)에서 현재 이용된 코드, 신택스, 및/또는 시맨틱들에 대한 변화들로서 구현될 수도 있다. 이러한 표준들 (또는 그 초안들)은 멀티-뷰 고효율 비디오 코딩, 작업 초안 3 (MV-HEVC WD3) 및 HEVC에 대한 스케일러블 비디오 코딩 확장 작업 초안 1 (SHVC WD1) 뿐만 아니라, 유사한 기법들을 채용하는 미래의 표준들 및 초안들을 포함한다. 이러한 코드, 신택스, 및/또는 시맨틱들의 일 예의 실시형태들은 개시물의 전반에 걸쳐 제공된다.

현재의 기법들을 이용하면, 현재 꾹처를 코딩할 때, 현재 꾹처와 연관된 계층간 참조 꾹처 세트가 초기에 생성된다. 계층간 참조 꾹처 세트는 일반적으로, 현재 꾹처가 속하는 계층의 직접 종속 계층들에 기초하여 생성된다. 현재 꾹처가 속하는 계층의 직접 참조 계층들은 또한, 현재 꾹처와 연관된 직접 참조 계층들로서 지칭된다. 다음으로, 참조 꾹처 리스트는 그에 대한 계층간 참조 꾹처 세트가 서브세트인 참조 꾹처 세트에 기초하여 생성된다. 현재 꾹처가 계층간 예측을 이용하여 코딩되지 않아야 할 경우, 또는 하나의 꾹처만이 현재 꾹처를 코딩하기 위해 이용되도록 허용될 경우에는, 신택스 엘리먼트들이 참조 꾹처 리스트를 수정하도록 제공되어야 한다. 예를 들어, 이러한 추가적인 신택스는 하나 이상의 꾹처들이 참조 꾹처 리스트에서 포함되지 않아야 한다는 것을 표시할 수도 있다. 참조 꾹처 리스트는 이러한 추가적인 신택스를 이용하여 수정될 수도 있다.

이하에서 설명된 기법들은 참조 꾹처 리스트가 이 비효율을 회피하는 방식으로 구성되도록 한다. 예를 들어, 이하의 실시형태들에 따르면, 참조 꾹처 리스트는 초기에, 참조 꾹처 리스트 수정 및 연관된 신택스를 이용하는 것을 회피하는 방법으로 구성될 수도 있다. 게다가, 이하에서 설명된 참조 꾹처 세트 및 참조 꾹처 리스트 구성의 기법들은 코딩 디바이스가 가능한 한 초기에, 누락된 계층간 참조된 꾹처가 (예컨대, 송신 동안

예) 손실되었는지 또는 (예컨대, 원래의 비트스트림에서) 제공되지 않았는지 여부를 결정하도록 한다.

[0034] 실제로, 이러한 누락 (missing) 픽처들을 가능한 한 조기에 알게 되어, 코딩 디바이스가 적절한 액션을 취할 수 있는 것이 유리하다. 예를 들어, 코딩 디바이스는 그것이 누락되고 있음을 알게 되는 경우에 누락 픽처의 송신을 재요청할 수도 있다. 누락 픽처는 계층간 참조 픽처 세트 또는 참조 픽처 리스트에서 참조되는 픽처를 지칭할 수 있지만, 그것은 (도 2a 내지 도 3b에서 예시되며 이하에서 더욱 상세하게 논의된 바와 같이, 코딩 디바이스의 참조 프레임 메모리 내에 포함될 수 있는) 코딩 디바이스의 디코딩된 픽처 버퍼에서 존재하지 않는다.

[0035] 하나의 실시형태에서, 계층간 참조 픽처 세트 서브세트는 계층간 픽처 참조를 위해 이용된 픽처들의 손실을 검출하기 위해 이용될 수도 있도록 하는 방식으로 시그널링되고 유도된다. 예를 들어, (이하에 설명된 바와 같이) 도 4 및 도 5에서 예시된 바와 같이, 표시가 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 이용된 계층간 참조 픽처들의 수를 표시하는 표시가 시그널링될 수도 있다. 수는 0 과, 현재 픽처와 연관된 직접 참조 계층들의 수의 사이일 수 있다. 게다가, 현재 픽처의 모든 슬라이스들이 계층간 예측을 위해 현재 픽처에 의해 이용된 동일한 수의 픽처들을 갖게 요구되도록, 제한이 제공될 수도 있다. 또한, 디바이스는 수가 0 과 동일한 경우, 이것이 현재 픽처가 계층간 예측을 이용하지 않으면서 코딩되어야 한다는 것을 표시하도록 구성될 수도 있다. 일부의 실시형태들에서, 수는 1에 불과한 것 (예컨대, 0 또는 1)으로 제한될 수도 있다. 이러한 제한은 효과적으로 하나에 불과한 참조 픽처가 계층간 예측 동안에 이용되도록 할 것이다.

[0036] 게다가, 표시가 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 어느 특별한 계층간 참조 픽처들이 이용되는지를 표시하는 표시가 시그널링될 수도 있다. 현재 픽처의 모든 슬라이스들이 동일한 계층간 참조 픽처들을 이용하게 요구되도록, 제한이 제공될 수도 있다.

[0037] 일부의 실시형태들에서는, 전혀 없거나 (즉, 제로) 단지 하나인 직접 종속 계층이 현재 픽처의 슬라이스 헤더에서 시그널링되도록, 제한이 추가로 제공된다. 또한, 일부의 실시형태들에서, 초기 참조 픽처 리스트 생성은 초기 참조 픽처 리스트에서 하나의 계층간 참조 픽처만을 포함할 가능성을 허용한다.

[0038] 이러한 기법들의 실시형태들은 이하의 일 예의 신팩스 및 방법들에 대하여 더욱 상세하게 추가로 설명된다.

"이용된 최고 종속성 계층" 플래그에 대한 비디오 파라미터 세트 신팩스 및 시맨틱들

[0039] 하나의 실시형태에서는, 이용된 최고 종속성 계층과 연관된 플래그 (예컨대, highest_dep_layer_used_flag) 를 포함하는 신팩스 엘리먼트가 제공된다. 플래그는 최고 종속 계층이 계층간 예측을 이용하는 각각의 픽처에 대한 계층간 예측을 위해 이용되는지 여부를 특정한다. 이러한 신팩스 및 시맨틱들의 하나의 실시형태는 다음과 같다:

표 1

vps_extension() {	디스크립터
...	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
highest_dep_layer_used_flag	u(1)
}	

표 1: 비디오 파라미터 세트 신팩스

표 2

1 과 동일한 `highest_dep_layer_used_flag` 는 최고 종속 계층이 인터-계층 예측을 이용하는 각각의 끝에 대한 인터-계층 예측을 위해 이용됨을 특정한다. 0 과 동일한 `highest_dep_layer_used_flag` 는 최고 종속 계층이 인터-계층 예측을 이용하는 각각의 끝에 대한 인터-계층 예측을 위해 이용될 수도 있거나 이용되지 않을 수도 있음을 특정한다.

표 2 비디오 파라미터 세트 시멘틱들

[0042]

일반적인 슬라이스 세그먼트 헤더 신택스 및 시멘틱들

[0044]

하나의 실시형태에서는, 계층간 예측을 위해 현재 끝에 의해 이용된 끝들의 수와 연관된 변수 (예컨대, `num_inter_layer_ref_pics`) 를 포함하는 신택스 엘리먼트가 제공된다. 신택스 엘리먼트는 또한, 현재 끝에 의해 지정된 i-번째 계층간 참조 끝의 `nuh_layer_id` 를 나타내는 변수 (예컨대, `ref_layer_idx_delta[i]`) 를 포함한다. 이러한 신택스 및 시멘틱들의 하나의 실시형태는 다음과 같다:

표 3

slice_segment_header() {	디스크립터
...	
if(<code>nuh_layer_id</code> > 0 && <code>NumDirectRefLayers[LayerIdInVps[<code>nuh_layer_id</code>]]</code> > 0)	
<code>num_inter_layer_ref_pics</code>	ue(v)
for(<code>i</code> = 0; <code>i</code> < <code>num_inter_layer_ref_pics</code> ; <code>i</code> ++)	
if(<code>i</code> > 0 ! <code>highest_dep_layer_used_flag</code>)	
<code>ref_layer_idx_delta[i]</code>	ue(v)
}	
if(<code>slice_segment_header_extension_present_flag</code>) {	
<code>slice_segment_header_extension_length</code>	ue(v)
for(<code>i</code> = 0; <code>i</code> < <code>slice_segment_header_extension_length</code> ; <code>i</code> ++)	
<code>slice_segment_header_extension_data_byte[i]</code>	u(8)
}	
<code>byte_alignment()</code>	
}	

표 3: 슬라이스 세그먼트 헤더 신택스

[0045]

표 4

`num_inter_layer_ref_pics` 는 인터-계층 예측을 위하여 현재의 픽처에 의해 이용된 픽처들의 수를 특정한다. 존재하지 않을 때, `num_inter_layer_ref_pics` 의 값은 0 과 동일한 것으로 추론된다. `num_inter_layer_ref_pics` 의 값은 0 내지 `NumDirectRefLayers[LayerIdInVps[nuh_layer_id]]` 까지의 범위에 있을 것이다.

`ref_layer_idx_delta[i]` 는 현재의 픽처에 의해 참조된 i -번째 인터-계층 참조 픽처의 `nuh_layer_id` 를 나타내는 변수 `RefPicLayerId[i]` 를 특정한다. `highest_dep_layer_used_flag` 가 1 과 동일하고 `num_inter_layer_ref_pics` 가 0 보다 더 클 때, `ref_layer_idx_delta[0]` 의 값은 0 과 동일한 것으로 추론된다.

표 4: 슬라이스 세그먼트 헤더 시멘틱들

[0046]

픽처의 모든 슬라이스들은 `num_inter_layer_ref_pics` 의 동일한 값을 가질 것이다. 게다가, 0 과 동일한 `num_inter_layer_ref_pics` 는 현재 픽처에 대해 계층간 예측이 이용되지 않음을 표시한다. 값은 1 이하인 것으로 제한될 수도 있다. 예를 들어, SHVC 프로파일은 이러한 제한을 제공할 수도 있어서, 하나에 이르는 픽처가 계층간 참조를 위해 이용된다.

[0048]

하나의 실시형태에서는, 참조 픽처 계층 식별 변수 `RefPicLayerId[i]` 가 다음과 같이 유도된다:

표 5

```

for( i = 0; i < num_inter_layer_ref_pics; i++ ) {
    currLayerIdx = LayerIdInVps[ nuh_layer_id ]
    if( i == 0 )
        if( highest_dep_layer_used_flag )
            refLayerIdx[ 0 ] = NumDirectRefLayers[ currLayerIdx ] - 1
        else
            refLayerIdx[ 0 ] = NumDirectRefLayers[ currLayerIdx ] -
                1 - ref_layer_idx_delta[ 0 ]
        else
            refLayerIdx[ i ] = refLayerIdx[ i - 1 ] +
                ref_layer_idx_delta[ i ] + 1
            RefPicLayerId[ i ] =
                RefLayerId[ currLayerIdx ][ refLayerIdx[ i ] ]
    }
}

```

표 5: `RefPicLayerId[i]` 유도의 예

[0049]

이러한 실시형태에서는, 픽처의 모든 슬라이스들이 동일한 참조 계층 인덱스 델타 값 (예컨대, `ref_layer_idx_delta`) 을 가질 것이다.

[0051]

참조 픽처 리스트 수정 시멘틱들

[0052]

하나의 실시형태에서는, SHVC 또는 MV-HEVC 사양의 하위조항 F.7.4.7.2 이 변수 `NumPocTotalCurr` 가 어떻게 유

도되는지에 대한 것으로 수정될 수도 있다. 예를 들어, 하나의 실시형태에서는, NumPocTotalCurr 의 유도를 특정하는 수학식 7-43 이 다음으로 대체된다:

표 6

```

NumPocTotalCurr = 0
for( i = 0; i < NumNegativePics[ CurrRpsIdx ]; i++)
    if(UsedByCurrPicS0[ CurrRpsIdx ][ i ])
        NumPocTotalCurr++
for( i = 0; i < NumPositivePics[ CurrRpsIdx ]; i++)
    if(UsedByCurrPicS1[ CurrRpsIdx ][ i ])
        NumPocTotalCurr++
for( i = 0; i < num_long_term_sps + num_long_term_pics;
i++ )
    if( UsedByCurrPicLt[ i ])
        NumPocTotalCurr++
NumPocTotalCurr += num_inter_layer_ref_pics

```

표 6: NumPocTotalCurr 유도의 예

[0053]

계층간 참조 픽처 세트에 대한 디코딩 프로세스

하나의 실시형태에서는, 계층간 참조 픽처 세트에 대한 디코딩 프로세스가 제공된다. 프로세스의 출력은 계층간 참조 픽처들의 업데이트된 리스트 RefPicSetInterLayer 이다. 하나의 실시형태에서는, 리스트 RefPicSetInterLayer 가 먼저 비워지고 다음으로, 다음과 같이 유도된다:

표 7

```

for( i = 0; i < num_inter_layer_ref_pics; i++ ) {
    if( 현재의 픽처와 동일한 PicOrderCntVal 및 RefPicLayerId[ i ] 와 동일한
        nuh_layer_id 를 가지는 픽처 picX 가 디코딩된 픽처 베퍼 (DPB) 에서 있음 ) {
        RefPicSetInterLayer[ i ] = picX
        RefPicSetInterLayer[ i ] 는 "장기 참조를 위해 이용됨" 으로 마킹됨
    } else
        RefPicSetInterLayer[ i ] = "참조 픽처 없음"
}

```

표 7: RefPicSetInterLayer 유도의 예

[0056]

이러한 실시형태에서는, 업데이트된 리스트, RefPicSetInterLayer 에서 "참조 픽처 없음" 과 동일한 엔트리가 없을 것이다.

0 보다 더 큰 nuh_layer_id 를 갖는 코딩된 픽처의 디코딩을 종료하기 위한 마킹 프로세스

하나의 실시형태에서는, 0 보다 더 큰 nuh_layer_id 를 갖는 코딩된 픽처를 디코딩하는 종점을 마킹하기 위한 프로세스가 제공된다. 이 프로세스의 출력은 일부의 디코딩된 픽처들에 대한 "단기 참조를 위해 이용됨" 으로 잠재적으로 업데이트된 마킹이다. 이러한 마킹을 위한 하나의 예의 프로세스는 다음과 같다:

표 8

```

for( i = 0; i < num_inter_layer_ref_pics; i++ )
    RefPicSetInterLayer[ i ] 는 "단기 참조를 위해 이용됨" 으로서 마킹됨

```

[0060] 표 8: "단기 참조를 위해 이용됨" 으로서 RefPicSetInterLayer 를 마킹하는 예

[0061] 참조 픽처 리스트 구성을 위한 디코딩 프로세스

[0062] 하나의 실시형태에서는, 참조 픽처 리스트 구성을 위한 디코딩 프로세스가 제공된다. 프로세스는 각각의 P 또는 B 슬라이스에 대한 디코딩 프로세스의 시작 시에 호출될 수도 있다. 참조 픽처들은 현존하는 표준들에서 특정된 바와 같은 참조 인덱스들을 통해 어드레싱될 수도 있다. 예를 들어, 참조 픽처들은 HEVC 표준의 하위조항 8.5.3.3.2 에서 특정된 바와 같이 어드레싱될 수도 있다. 참조 인덱스는 참조 픽처 리스트로의 인덱스이다. P 슬라이스를 디코딩할 때, 단일의 참조 픽처 리스트 RefPicList0 가 있다. B 슬라이스를 디코딩할 때에는, RefPicList0 에 추가하여, 제 2 독립 참조 픽처 리스트 RefPicList1 가 있다.

[0063] 이하에서 유도된 최종 RefPicList0 및 RefPicList1 에서의 각각의 엔트리가 디코딩된 픽처 베퍼 (DPB) 에서 존재하는 픽처에 대응할 것이라는 것은 비트스트림 준수의 요건이다. 각각의 슬라이스에 대한 디코딩 프로세스의 시작 시에, 참조 픽처 리스트 RefPicList0 와, B 슬라이스들에 대하여, RefPicList1 은 다음과 같이 유도된다: 일시적인 참조 픽처 리스트 0 변수 NumRpsCurrTempList0 는 Max(num_ref_idx_10_active_minus1 + 1, NumPocTotalCurr) 과 동일하게 설정되고, 리스트 RefPicListTemp0 는 다음과 같이 구성된다:

표 9

```

rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ 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 < num_inter_layer_ref_pics; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetInterLayer[ i ]
}

```

[0064] 표 9: 일시적인 참조 픽처 리스트 0 을 구성하는 예

[0065] 다음으로, 참조 픽처 리스트 0 (RefPicList0) 은 다음과 같이 구성될 수도 있다:

표 10

```

for( rIdx = 0; rIdx <= num_ref_idx_10_active_minus1; rIdx++)
    RefPicList0[rIdx]=ref_pic_list_modification_flag_10      ?
RefPicListTemp0[list_entry_10[rIdx]] : RefPicListTemp0[ rIdx ]

```

[0066] 표 10: 참조 픽처 리스트 0 을 구성하는 예

[0067] 슬라이스가 B 슬라이스일 때, 변수 NumRpsCurrTempList1 는 Max(num_ref_idx_11_active_minus1 + 1, NumPocTotalCurr) 과 동일하게 설정되고, 리스트 RefPicListTemp1 은 다음과 같이 구성된다:

표 11

```

rIdx = 0
while( rIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrAfter[ 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 < num_inter_layer_ref_pics; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetInterLayer[ i ]
}

```

[0068] 표 11: 일시적인 참조 픽처 리스트 1 을 구성하는 예

[0069] 슬라이스가 B 슬라이스일 때, 참조 픽처 리스트 RefPicList1 는 다음과 같이 구성된다:

표 12

```

for( rIdx = 0; rIdx <= num_ref_idx_11_active_minus1; rIdx++)
    RefPicList1[     rIdx     ] =     ref_pic_list_modification_flag_11      ?
RefPicListTemp1[ list_entry_11 [ rIdx ] ] : RefPicListTemp1[ rIdx ]

```

[0070] 표 12: 참조 픽처 리스트 1 을 구성하는 예

[0071] "하나의 ILP 참조 픽처 단독" 플래그에 대한 비디오 파라미터 세트 확장 신택스 및 시멘틱들

[0072] 또 다른 실시형태에서는, 하나의 계층간 예측 (ILP) 참조 픽처만이 ILP 를 위해 이용되는지 여부를 표시하는 플래그가 제공된다. 신택스 및 시멘틱들은 다음과 같이 제공될 수도 있다:

표 13

vps_extension() {	디스크립터
...	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
one_ilp_ref_pic_only_flag	u(1)
}	

표 13: 비디오 파라미터 세트 신택스

[0073]

표 14

1 과 동일한 one_ilp_ref_pic_only_flag 는 각각의 픽처가 최대로 하나의 인터-계층 참조 픽처를 지칭하는 것을 특정한다. 0 과 동일한 one_ilp_ref_pic_only_flag 는 각각의 픽처가 하나를 초과하는 인터-계층 참조 픽처를 지정할 수도 있는 것을 특정한다.

표 14: 비디오 파라미터 세트 시멘틱들

[0074]

[0075] 이 실시형태로 제공된 하나의 장점은, HEVC 확장들의 SHVC 및 3DV 계열의 양자 모두에 대한 하이-레벨 (high-level) 신택스는 여전히 동일할 수 있다는 것이다. 각각의 픽처가 최대로 하나의 계층간 참조 픽처를 지칭하는 것이 요구될 경우, 이 플래그는 프로파일 정의들에서 1 과 동일하도록 요구될 수 있다. 이와 다르게, 이 실시형태는 각각의 픽처가 최대로 하나의 계층간 참조 픽처를 지칭하는 비트스트림들에 대한 참조 픽처 리스트 수정 커맨드들에 대해 이용된 비트들을 절감하기 위해 여전히 유용할 수도 있다.

[0076] 또 다른 실시형태에서는, 신택스가 다음과 같이 제공될 수도 있다:

표 15

vps_extension() {	디스크립터
...	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
one_ilp_ref_pic_only_flag	u(1)
if(one_ilp_ref_pic_only_flag)	
ilp_ref_pic_present_in_slice_flag	u(1)
}	

표 15: 비디오 파라미터 세트 신택스

[0077]

[0078] 이러한 실시형태에서의 시멘틱들은 다음을 포함할 수도 있다:

표 16

1 과 동일한 `one_ilp_ref_pic_only_flag` 는 각각의 픽처가 최대로 하나의 인터-계층 참조 픽처를 지칭하는 것을 특정한다. 0 과 동일한 `one_ilp_ref_pic_only_flag` 는 각각의 픽처가 하나를 초과하는 인터-계층 참조 픽처를 지정할 수도 있는 것을 특정한다.

표 16: 비디오 파라미터 세트 시맨틱들

[0079]

[0080] 이 실시형태에서, HEVC 확장들의 SHVC 및 3DV 계열의 양자 모두에 대한 하이-레벨 신택스는 여전히 동일할 수 있다. 각각의 픽처가 최대로 하나의 계층간 참조 픽처를 지칭하는 것이 요구될 경우, 이 `one_ilp_ref_pic_only_flag` 플래그는 프로파일 정의들에서 1 과 동일하도록 요구될 수 있다. 이와 다르게, 이 실시형태는 각각의 픽처가 최대로 하나의 계층간 참조 픽처를 지칭하는 비트스트림들에 대한 참조 픽처 리스트 수정 커맨드들에 대해 이용된 비트들을 절감하기 위해 여전히 유용할 수도 있다. 시맨틱들은 또한 다음을 포함할 수도 있다:

표 17

1 과 동일한 `ilp_ref_pic_present_in_slice_flag` 는 `ref_layer_idx_delta` 가 슬라이스 헤더에서 존재할 수도 있는 것을 특정한다. 0 과 동일한 `ilp_ref_pic_present_in_slice_flag` 는 `ref_layer_idx_delta` 가 슬라이스 헤더에서 존재하지 않는 것을 특정한다. 존재하지 않을 때, `ilp_ref_pic_present_in_slice_flag` 의 값은 1 과 동일한 것으로 추론된다.

표 17: 비디오 파라미터 세트 시맨틱들

[0081]

[0082] 또 다른 실시형태에서는, 신택스 및 시맨틱들이 다음과 같이 제공될 수도 있다:

표 18

vps_extension() {	디스크립터
...	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
<code>direct_dependency_flag</code> [i][j]	u(1)
if(scalability_mask[1])	
<code>ilp_ref_pic_present_in_slice_flag</code>	u(1)
}	

표 18: 비디오 파라미터 세트 신택스

[0083]

표 19

1 과 동일한 `ilp_ref_pic_present_in_slice_flag` 는 `ref_layer_idx_delta` 가 슬라이스 헤더에서 존재할 수도 있는 것을 특정한다. 0 과 동일한 `ilp_ref_pic_present_in_slice_flag` 는 `ref_layer_idx_delta` 가 슬라이스 헤더에서 존재하지 않는 것을 특정한다. 존재하지 않을 때, `ilp_ref_pic_present_in_slice_flag` 의 값은 1 과 동일한 것으로 추론된다.

표 19: 비디오 파라미터 세트 시멘틱들

[0084]

[0085] 이 실시형태에서는, `scalability_mask[1]` 가 1 과 동일한 경우, `one_ilp_ref_pic_only_flag` 의 값이 1 과 동일한 것으로 추론된다.

일반적인 슬라이스 세그먼트 헤더 싱택스 및 시맨틱들

[0087] 또 다른 실시형태에서, 슬라이스 세그먼트 헤더 싱택스 및 시맨틱들은 다음과 같이 제공된다:

표 20

slice_segment_header() {	디스크립터
...	
if(<code>nuh_layer_id</code> > 0) {	
<code>no_inter_layer_pred_flag</code>	<code>u(1)</code>
if(<code>one_ilp_ref_pic_only_flag</code> && ! <code>no_inter_layer_pred_flag</code>)	
<code>ref_layer_idx_delta</code>	<code>ue(v)</code>
}	
}	
if(<code>slice_segment_header_extension_present_flag</code>) {	
<code>slice_segment_header_extension_length</code>	<code>ue(v)</code>
for(i = 0; i < <code>slice_segment_header_extension_length</code> ; i++)	
<code>slice_segment_header_extension_data_byte[i]</code>	<code>u(8)</code>
}	
<code>byte_alignment()</code>	
}	

표 20: 슬라이스 세그먼트 헤더 싱택스

[0088]

[0089] 또 다른 실시형태에서, 슬라이스 세그먼트 헤더 싱택스는 다음과 같이 제공된다:

표 21

<pre> slice_segment_header() { ... if(nuh_layer_id > 0) { no_inter_layer_pred_flag if(ilp_ref_pic_present_in_slice_flag && !no_inter_layer_pred_flag) ref_layer_idx_delta } if(slice_segment_header_extension_present_flag) { slice_segment_header_extension_length for(i = 0; i < slice_segment_header_extension_length; i++) slice_segment_header_extension_data_byte[i] } byte_alignment() } </pre>	디스크립터 u(1) ue(v) ue(v) ue(v) u(8)
--	--

표 21: 슬라이스 세그먼트 헤더 선택스

[0090]

[0091] 슬라이스 세그먼트 헤더 시멘틱들은 다음을 포함할 수도 있다:

표 22

1 과 동일한 no_inter_layer_pred_flag 는 현재의 픽처가 인터-계층 예측을 이용하지 않는 것을 특정한다. 0 과 동일한 no_inter_layer_pred_flag 는 현재의 픽처가 인터-계층 예측을 이용할 수도 있는 것을 특정한다.

표 22: 슬라이스 세그먼트 헤더 시멘틱들

[0092]

[0093] 이러한 실시형태에서, 픽처의 모든 슬라이스들은 no_inter_layer_pred_flag 의 동일한 값을 가질 것이다. 변수 NumEnt InRefPicSetInterLayer 는 다음과 같이 유도될 수도 있다:

표 23

```

if( no_inter_layer_pred_flag || NumDirectRefLayers[ LayerIdInVps[ nuh_layer_id ] ] == 0 )
    NumEnt InRefPicSetInterLayer = 0
else if( one_ilp_ref_pic_only_flag )
    NumEnt InRefPicSetInterLayer = 1
else
    NumEnt InRefPicSetInterLayer = NumDirectRefLayers[ LayerIdInVps[ nuh_layer_id ] ]

```

표 23: NumEnt InRefPicSetInterLayer 유도

[0094]

[0095] **ref_layer_idx_delta** 는 현재 픽처에 의해 참조된 계층간 참조 픽처의 **nuh_layer_id** 를 나타내는 변수 **RefPicLayerId** 를 특정한다. **ref_layer_idx_delta** 가 존재하지 않을 때, 그것은 0 과 동일한 것으로 추론된다. 변수 **RefPicLayerId** 는 다음과 같이 유도될 수도 있다:

표 24

```

currLayerIdx = LayerIdInVps[ nuh_layer_id ]
refLayerIdx = NumDirectRefLayers[ currLayerIdx ] - 1 - ref_layer_idx_delta
RefPicLayerId = RefLayerId[ currLayerIdx ][ refLayerIdx ]

```

표 24: RefPicLayerId 유도

[0096]

[0097] 이러한 실시형태에서, 픽처의 모든 슬라이스들은 **ref_layer_idx_delta** 의 동일한 값을 가질 것이다.

[0098]

참조 픽처 리스트 수정 시멘틱들

[0099]

또 다른 실시형태에서는, SHVC 또는 MV-HEVC 사양의 하위조항 F.7.4.7.2 이 이하에서 도시된 바와 같이 수정될 수도 있다. **NumPocTotalCurr** 의 유도를 특정하는 수학식 (7-43) 은 다음에 의해 대체된다:

표 25

```

NumPocTotalCurr = 0
for( i = 0; i < NumNegativePics[ CurrRpsIdx ]; i++ )
    if(UsedByCurrPicS0[ CurrRpsIdx ][ i ])
        NumPocTotalCurr++
for( i = 0; i < NumPositivePics[ CurrRpsIdx ]; i++ )
    if(UsedByCurrPicS1[ CurrRpsIdx ][ i ])
        NumPocTotalCurr++
for( i = 0; i < num_long_term_sps + num_long_term_pics; i++ )
    if( UsedByCurrPicLt[ i ] )
        NumPocTotalCurr++
NumPocTotalCurr += NumEntInRefPicSetInterLayer

```

표 25: NumPocTotalCurr 유도

[0100]

계층간 참조 픽처 세트에 대한 디코딩 프로세스

[0102]

이 프로세스의 출력은 계층간 참조 픽처들의 업데이트된 리스트 **RefPicSetInterLayer** 이다. 하나의 실시형태에서는, 리스트 **RefPicSetInterLayer** 가 먼저 비워지고 다음으로, 다음과 같이 유도된다:

표 26

```

for( i = 0; i < NumEntInRefPicSetInterLayer; i++ ) {
    if( one_ilp_ref_pic_only_flag )
        tmpLayerId = RefPicLayerId
    else
        tmpLayerId = RefLayerId[ LayerIdInVps[ nuh_layer_id ][ i ] ]
    if( 현재의 픽처와 동일한 PicOrderCntVal 를 가지는 픽처 picX 가 DPB 내에 있고
        nuh_layer_id 가 tmpLayerId 와 동일함 ) {
        RefPicSetInterLayer[ i ] = picX
        RefPicSetInterLayer[ i ] 는 "장기 참조를 위해 이용됨" 으로 표기됨
    } else
        RefPicSetInterLayer[ i ] = "참조 픽처 없음"
}

```

표 26: RefPicSetInterLayer 유도

[0103]

[0104] 이러한 실시형태에서는, one_ilp_ref_pic_only_flag 가 1 과 동일한 경우, RefPicSetInterLayer 에서 "참조 픽처 없음" 과 동일한 엔트리가 없을 것이다.

[0105] 0 보다 더 큰 nuh_layer_id 를 갖는 코딩된 픽처의 디코딩을 종료하기 위한 마킹 프로세스

[0106] 또 다른 실시형태에서는, 0 보다 더 큰 nuh_layer_id 를 갖는 코딩된 픽처의 디코딩을 종료하기 위한 마킹 프로세스가 제공된다. 이 프로세스의 출력은, DPB 에서의 일부의 디코딩된 픽처들의 마킹 상태가 "단기 참조를 위해 이용됨" 으로 마킹되는 것으로 변화될 수도 있다는 것이다.

표 27

```

for( i = 0; i < NumEntInRefPicSetInterLayer; i++ )
    if( 픽처 RefPicSetInterLayer[ i ] 가 DPB 내에 있음 )
        RefPicSetInterLayer[ i ] 는 "단기 참조를 위해 이용됨" 으로 마킹됨

```

표 27: 마킹 프로세스

[0107]

[0108] 참조 픽처 리스트들 구성을 위한 디코딩 프로세스

[0109] 또 다른 실시형태에서는, 참조 픽처 리스트 구성을 위한 디코딩 프로세스가 제공된다. 프로세스는 각각의 P 또는 B 슬라이스에 대한 디코딩 프로세스의 시작 시에 이용될 수도 있다. 참조 픽처들은 HEVC 사양의 하위 조항 8.5.3.3.2 에서 특정된 바와 같이 참조 인덱스들을 통해 어드레싱된다. 참조 인덱스는 참조 픽처 리스트의 인덱스이다. P 슬라이스를 디코딩할 때, 단일의 참조 픽처 리스트 RefPicList0 가 있다. B 슬라이스를 디코딩할 때에는, RefPicList0 에 추가하여, 제 2 독립 참조 픽처 리스트 RefPicList1 가 있다.

[0110] 이하에서 유도된 최종 RefPicList0 및 RefPicList1 에서의 각각의 엔트리가 DPB 에서 존재하는 픽처에 대응할 것이라는 것은 비트스트림 준수의 요건이다. 하나의 실시형태에서는, 각각의 슬라이스에 대한 디코딩 프로세스의 시작 시에, 참조 픽처 리스트 RefPicList0 와, B 슬라이스들에 대하여, RefPicList1 는 다음과 같이 유도된다. 변수 NumRpsCurrTempList0 는 Max(num_ref_idx_10_active_minus1 + 1, NumPocTotalCurr) 와 동

일하게 설정되고, 일시적인 참조 퍽처 리스트 RefPicListTemp0 는 다음과 같이 구성된다:

표 28

```
rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ 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 < NumEntInRefPicSetInterLayer; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetInterLayer[ i ]
}
```

표 28: 일시적인 참조 퍽처 리스트 0 (RefPicListTemp0) 구성 프로세스

[0111] 최종 참조 퍽처 리스트 RefPicList0 는 다음과 같이 구성될 수도 있다:

표 29

```
for( rIdx = 0; rIdx <= num_ref_idx_10_active_minus1; rIdx++)
    RefPicList0[ rIdx ] = ref_pic_list_modification_flag_10 ?
        RefPicListTemp0[ list_entry_10[ rIdx ] ] : RefPicListTemp0[ rIdx ]
```

표 29: 참조 퍽처 리스트 0 (RefPicList0) 구성 프로세스

[0113] 슬라이스가 B 슬라이스일 때, 변수 NumRpsCurrTempList1 은 Max(num_ref_idx_11_active_minus1 + 1, NumPocTotalCurr) 와 동일하게 설정되고, 리스트 RefPicListTemp1 은 다음과 같이 구성된다:

표 30

```

rIdx = 0
while( rIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrAfter[ 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 < NumEntInRefPicSetInterLayer; rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetInterLayer[ i ]
}

```

표 30: 일시적인 참조 픽처 리스트 1 (RefPicListTemp1) 구성 프로세스

[0115]

슬라이스가 B 슬라이스일 때, 최종 참조 픽처 리스트 1 RefPicList1 은 다음과 같이 구성된다:

표 31

```

for( rIdx = 0; rIdx <= num_ref_idx_11_active_minus1; rIdx++ )
    RefPicList1[ rIdx ] = ref_pic_list_modification_flag_11 ?
        RefPicListTemp1[ list_entry_11[ rIdx ] ] :
        RefPicListTemp1[ rIdx ]

```

표 31: 참조 픽처 리스트 1 (RefPicListTemp1) 구성 프로세스

[0117]

또 다른 실시형태에서는, 계층간 픽처 세트에 대한 디코딩 프로세스가 제공된다. 프로세스의 출력은 계층간 참조 픽처들의 업데이트된 리스트 RefPicSetInterLayer 이다. 리스트 RefPicSetInterLayer 가 먼저 비워지고 다음으로, 다음과 같이 유도된다:

표 32

```

for( i = 0; i < NumDirectRefLayers[ LayerIdInVps[ nuh_layer_id ] ]; i++ ) {
    if( 현재의 픽처와 동일한 PicOrderCntVal 및 RefLayerId[ LayerIdInVps[ nuh_layer_id ][ i ] ]
        와 동일한 nuh_layer_id 를 가지는 픽처 picX 가 DPB 에서 있음 ) {
        RefPicSetInterLayer[ i ] = picX
        RefPicSetInterLayer[ i ] 는 "장기 참조를 위해 이용됨" 으로 마킹됨
    } else
        RefPicSetInterLayer[ i ] = "참조 픽처 없음"
}

```

표 32: 인터-계층 참조 픽처들의 생성된 업데이트된 리스트에 대한 프로세스

[0119]

[0120] 0 보다 더 큰 nuh_layer_id 를 갖는 코딩된 픽처의 디코딩을 종료하기 위한 마킹 프로세스

또 다른 실시형태에서는, 0 보다 더 큰 nuh_layer_id 를 갖는 코딩된 픽처의 디코딩을 종료하기 위한 마킹 프로세스가 제공된다. 프로세스의 출력은 일부의 디코딩된 픽처들에 대한 "단기 참조를 위해 이용됨" 으로 잠재적으로 업데이트된 마킹이다.

표 33

```
for( i = 0; i < NumDirectRefLayers[ LayerIdInVps[ nuh_layer_id ] ]; i++ )
    if( 픽처 RefPicSetInterLayer[ i ] 가 DPB 내에 있음 )
        RefPicSetInterLayer[ i ] 는 "단기 참조를 위해 이용됨" 으로 마킹됨
```

[0122] 표 33: 마킹 프로세스

[0123] 참조 픽처 리스트들 구성을 위한 디코딩 프로세스

또 다른 실시형태에서는, 참조 픽처 리스트 구성을 위한 디코딩 프로세스가 제공된다. 프로세스는 각각의 P 또는 B 슬라이스에 대한 디코딩 프로세스의 시작 시에 이용될 수도 있다. 참조 픽처들은 HEVC 사양의 하위 조항 8.5.3.3.2 에서 특정된 바와 같이 참조 인덱스들을 통해 어드레싱된다. 참조 인덱스는 참조 픽처 리스트로의 인덱스이다. P 슬라이스를 디코딩할 때, 단일의 참조 픽처 리스트 RefPicList0 가 있다. B 슬라이스를 디코딩할 때에는, RefPicList0 에 추가하여, 제 2 독립 참조 픽처 리스트 RefPicList1 가 있다.

[0125] 본원에서 유도된 최종 RefPicList0 및 RefPicList1 에서의 각각의 엔트리가 DPB 에서 존재하는 픽처에 대응할 것이라는 것이 비트스트림 준수의 요건이고, 디코딩된 비디오 시퀀스가 SHVC 에 대해 특정된 하나 이상의 프로파일들을 준수할 때, RefPicList0 에서의 모든 픽처들 및 RefPicList1 에서의 모든 픽처들의 결합에서 포함된 하나에 불과한 계층간 참조 픽처가 있을 것이다. 하나의 실시형태에서는, 각각의 슬라이스에 대한 디코딩 프로세스의 시작 시에, 참조 픽처 리스트 RefPicList0 와, B 슬라이스들에 대하여, RefPicList1 는 본원에서 논의된 바와 같이 유도된다.

[0126] 신규한 시스템들, 장치들, 및 방법들의 다양한 양태들은 첨부한 도면들을 참조하여 이하에서 더욱 완전하게 설명된다. 그러나, 이 개시물은 많은 상이한 형태들로 구체화될 수도 있지만, 이 개시물의 전반에 걸쳐 제시된 임의의 특정 구조 또는 기능으로 제한되는 것으로 해석되지 않아야 한다. 오히려, 이 양태들은, 이 개시물이 철저하고 완벽할 것이며, 개시물의 범위를 당업자들에게 완전히 전달하도록 제공된다. 본원에서의 교시 사항들에 기초하여, 당업자는 개시물의 범위가 발명의 임의의 다른 양태에 독립적으로 또는 이와 조합하여 구현되든지 간에, 본원에서 개시된 신규한 시스템들, 장치들, 및 방법들의 임의의 양태를 전달하도록 의도된 것임을 인식해야 한다. 예를 들어, 본원에서 기재된 임의의 수의 양태들을 이용하여 장치가 구현될 수도 있거나 방법이 실시될 수도 있다. 게다가, 발명의 범위는 다른 구조, 기능성, 또는 구조와, 본원에서 기재된 발명의 다양한 양태들에 추가하여, 또는 이 다양한 양태들 이외의 기능성을 이용하여 실시되는 이러한 장치 또는 방법을 커버하도록 의도된 것이다. 본원에서 개시된 임의의 양태는 청구항의 하나 이상의 엘리먼트들에 의해 구체화될 수도 있다는 것을 이해해야 한다.

[0127] 특별한 양태들이 본원에서 설명되지만, 이 양태들의 많은 변형들 및 치환들은 개시물의 범위 내에 속한다. 바람직한 양태들의 일부의 이익들 및 장점들이 언급되지만, 개시물의 범위는 특별한 이익들, 용도들, 또는 목적들에 제한되도록 의도된 것이 아니다. 오히려, 개시물의 양태들은 상이한 무선 기술들, 시스템 구성들, 네트워크들, 및 송신 프로토콜들에 폭넓게 적용가능하도록 의도된 것이며, 이들의 일부는 바람직한 양태들의 도면들 및 다음의 설명에서 예로서 예시되어 있다. 상세한 설명 및 도면들은 제한하는 것이 아니라 개시물의 예시에 불과하고, 개시물의 범위는 첨부된 청구항들 및 그 등가물들에 의해 정의된다.

[0128] 비디오 코딩 시스템

[0129] 도 1 은 이 개시물에서 설명된 양태들에 따라 기법들을 사용할 수도 있는 일 예의 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 설명된 바와 같이, 용어 "비디오 코더" 는 일반적으로 비디오 인코더들 및 비디오 디코더들의 양자 모두를 지칭한다. 이 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 일반적으

로 비디오 인코딩 및 비디오 디코딩을 지칭할 수도 있다.

[0130] 도 1 에서 도시된 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 및 목적지 디바이스 (14) 를 포함한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 생성한다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 소스 디바이스 (12) 는, 컴퓨터-판독가능한 저장 매체 또는 다른 통신 채널을 포함할 수도 있는 통신 채널 (16) 을 통해 비디오 데이터를 목적지 디바이스 (14) 에 제공할 수 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크톱 컴퓨터들, 노트북 (예컨대, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 (set-top) 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 차량내 컴퓨터들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들을 포함할 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 구비될 수도 있다.

[0131] 목적지 디바이스 (14) 는 통신 채널 (16) 을 통해 디코딩되어야 할 인코딩된 비디오 데이터를 수신할 수도 있다. 통신 채널 (16) 은 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 일 타입의 매체 또는 디바이스를 포함할 수도 있다. 예를 들어, 통신 채널 (16) 은 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 직접 송신하는 것을 가능하게 하기 위한 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조될 수도 있고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 라디오 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 가능하게 하기 위해 유용할 수도 있는 다른 장비를 포함할 수도 있다.

[0132] 일부의 실시형태들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스로 출력될 수도 있다. 이러한 예들에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 저장하는 저장 디바이스 또는 컴퓨터-판독가능한 저장 매체에 대응할 수도 있다. 예를 들어, 목적지 디바이스 (14) 는 디스크 액세스 또는 카드 액세스를 통해 컴퓨터-판독가능한 저장 매체를 액세스할 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스 (28) 에 의해 컴퓨터-판독가능한 저장 매체로부터 액세스될 수도 있다. 컴퓨터-판독가능한 저장 매체는 하드 드라이브, 블루-레이 (Blu-ray) 디스크들, DVD 들, CD-ROM 들, 플래시 메모리, 휴발성 또는 비-휴발성 메모리, 또는 비디오 데이터를 저장하기 위한 다른 디지털 저장 매체들과 같은, 분산되거나 국소적으로 액세스된 다양한 데이터 저장 매체들 중의 임의의 것을 포함할 수도 있다. 컴퓨터-판독가능한 저장 매체는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 컴퓨터-판독가능한 저장 매체로부터의 저장된 비디오 데이터를 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있으며 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 일 타입의 서버일 수도 있다. 일 예의 파일 서버들은 (예컨대, 웹사이트를 위한) 웹 서버, FTP 서버, 네트워크 연결 저장 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함하는 표준 데이터 접속을 통해 인코딩된 비디오 데이터를 액세스할 수도 있다. 이것은, 파일 서버 상에 저장된 인코딩된 비디오 데이터를 액세스하기 위해 적당한 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양자 모두의 조합을 포함할 수도 있다. 컴퓨터-판독가능한 저장 매체로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양자 모두의 조합일 수도 있다.

[0133] 이 개시물의 기법들은 무선 애플리케이션들 또는 세팅들에 추가하여, 애플리케이션들 또는 세팅들을 적용할 수 있다. 기법들은 오버-디-에어 (over-the-air) 텔레비전 방송들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, HTTP 를 통한 동적 적응 스트리밍 (dynamic adaptive streaming over HTTP; DASH) 과 같은 인터넷 스트리밍 비디오 송신들, 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부의 실시형태들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 및/또는 화상 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위해 일방향 (one-way) 또는 양방향 (two-way) 비디오 송신을 지원하도록 구성될 수도 있다.

[0134]

도 1에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 소스 디바이스 (12)의 비디오 인코더 (20)는 다수의 표준들 또는 표준 확장들을 준수하는 비디오 데이터를 포함하는 비트스트림을 코딩하기 위한 기법들을 적용하도록 구성될 수도 있다. 다른 실시형태들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배치들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12)는 외부 카메라와 같은 외부 비디오 소스 (18)로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14)는 통합된 디스플레이 디바이스를 포함하는 것이 아니라, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0135]

소스 디바이스 (12)의 비디오 소스 (18)는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 및/또는 비디오 컨텐츠 제공자로부터 비디오를 수신하기 위한 비디오 공급 인터페이스를 포함할 수도 있다. 비디오 소스 (18)는 소스 비디오로서, 또는 라이브 비디오 (live video), 아카이빙된 비디오 (archived video), 및 컴퓨터-생성된 비디오의 조합으로서, 컴퓨터 그래픽-기반 데이터를 생성할 수도 있다. 일부의 실시형태들에서는, 비디오 소스 (18)가 비디오 카메라일 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14)가 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 캡처된 (captured), 프리-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20)에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 출력 인터페이스 (22)에 의해 통신 채널 (16)로 출력될 수도 있고, 이 통신 채널 (16)은 위에서 논의된 바와 같이, 컴퓨터-판독가능한 저장 매체를 포함할 수도 있다.

[0136]

컴퓨터-판독가능한 매체는 무선 방송 또는 유선 네트워크 송신과 같은 순시적 매체 (transient medium)들, 또는 하드 디스크, 플래시 드라이브, 컴팩트 디스크, 디지털 비디오 디스크, 블루-레이 디스크, 또는 다른 컴퓨터-판독가능한 매체들과 같은 저장 매체들 (예컨대, 비일시적 저장 매체들)을 포함할 수도 있다. 네트워크 서버 (도시되지 않음)는 (예컨대, 네트워크 송신을 통해) 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신할 수도 있으며 인코딩된 비디오 데이터를 목적지 디바이스 (14)에 제공할 수도 있다. 디스크 스탬핑 (disc stamping) 서비스와 같은 매체 생산 설비의 컴퓨팅 디바이스는 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신할 수도 있고, 인코딩된 비디오 데이터를 포함하는 디스크를 생산할 수도 있다. 그러므로, 통신 채널 (16)은 다양한 형태들의 하나 이상의 컴퓨터-판독가능한 저장 매체들을 포함하는 것으로 이해될 수도 있다.

[0137]

목적지 디바이스 (14)의 입력 인터페이스 (28)는 통신 채널 (16)로부터 정보를 수신할 수 있다. 통신 채널 (16)의 정보는, 블록들 및 다른 코딩된 유닛들, 예컨대, GOP 들의 특성들 및/또는 프로세싱을 설명하는 신택스 엘리먼트들을 포함하는 신택스 정보로서, 비디오 인코더 (20)에 의해 정의되며 또한, 비디오 디코더 (30)에 의해 이용될 수 있는 상기 신택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관 (cathode ray tube; CRT), 액정 디스플레이 (liquid crystal display; LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode; OLED) 디스플레이, 또는 또 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중의 임의의 것을 포함할 수도 있다.

[0138]

비디오 인코더 (20) 및 비디오 디코더 (30)는 현재 개발 중에 있는 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 코딩 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HEVC Test Model; HM)을 준수할 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30)는 MPEG-4, Part 10, 진보된 비디오 코딩 (AVC), 또는 이러한 표준들의 확장들로서 대안적으로 지정된 ITU-T H.264 표준과 같은 다른 독점적인 또는 산업적인 표준들에 따라 동작할 수도 있다. 그러나, 이 개시물의 기법들은 임의의 특별한 코딩 표준으로 제한되지 않는다.

비디오 코딩 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263을 포함한다. 도 1에서 도시되지 않았지만, 일부의 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통의 데이터 스트림 또는 별도의 데이터 스트림들에서 오디오 및 비디오 양자 모두의 인코딩을 처리하기 위한 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용 가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol; UDP)과 같은 다른 프로토콜들을 준수할 수도 있다.

[0139]

도 1은 예에 불과하고, 이 개시물의 기법들은 인코딩 및 디코딩 디바이스들 사이의 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 세팅들 (예컨대, 비디오 인코딩 또는 비디오 디코딩)에 적용할 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 취출 (retrieve) 될 수 있고, 네트워크를 통해 스트리밍될 수 있는 등등과 같다. 인코딩 디바이스는 데이터를 인코딩하고 데이터를 메모리에 저장할 수도 있고, 및/또는 디

코딩 디바이스는 메모리로부터 데이터를 취출하고 데이터를 디코딩할 수도 있다. 다수의 예들에서, 인코딩 및 디코딩은, 서로 통신하지 않지만, 간단하게 데이터를 메모리로 인코딩하고 및/또는 메모리로부터 데이터를 취출하고 데이터를 디코딩하는 디바이스들에 의해 수행된다.

[0140] 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (digital signal processor; DSP)들, 애플리케이션 특정 집적 회로 (application specific integrated circuit; ASIC)들, 필드 프로그래밍가능한 게이트 어레이 (field programmable gate array; FPGA)들, 개별로직, 소프트웨어, 하드웨어, 펌웨어 또는 그 임의의 조합들과 같은 다양한 적당한 인코더 회로부 중의 임의의 것으로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현될 때, 디바이스는 소프트웨어를 위한 명령들을 비일시적 (non-transitory) 컴퓨터-판독가능한 매체에서 저장할 수도 있고, 이 개시물의 기법들을 수행하기 위해 하나 이상의 프로세서들을 이용하여 명령들을 하드웨어로 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 그 어느 하나는 조합된 인코더/디코더 (combined encoder/decoder; CODEC)의 일부로서 각각의 디바이스 내에 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 셀룰러 전화와 같은 무선 통신 디바이스를 포함할 수도 있다.

[0141] JCT-VC 는 HEVC 표준의 개발에 노력을 들이고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HEVC Test Model; HM)로서 지정된 비디오 코딩 디바이스의 전화형 모델에 기초한다. HM은 예컨대, ITU-T H.264/AVC에 따라 현존하는 디바이스들과 관련된 비디오 코딩 디바이스들의 몇몇 추가적인 기능들을 추정한다. 예를 들어, H.264 가 9 개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM은 33 개만큼 많은 인트라-예측 인코딩 모드들을 제공할 수도 있다.

[0142] 일반적으로, HM의 작업 모델은, 비디오 프레임 또는 픽처가 루마 (luma) 및 크로마 (chroma) 샘플들의 양자 모두를 포함하는 최대 코딩 유닛 (largest coding unit; LCU)들 또는 트리블록들의 시퀀스 (sequence)로 분할될 수도 있다는 것을 설명한다. 비트스트림 내의 싱택스 데이터는 픽셀들의 수의 측면에서 최대 코딩 유닛인 LCU에 대한 사이즈를 정의할 수도 있다. 슬라이스는 코딩 순서에서 다수의 연속적인 트리블록들을 포함한다. 비디오 프레임 또는 픽처는 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 큐드트리에 따라 코딩 유닛 (CU)들로 분할될 수도 있다. 일반적으로, 큐드트리 데이터 구조는 CU 당 하나의 노드를 포함하고, 루트 노드 (root node)는 트리블록에 대응한다. CU가 4 개의 서브-CU 들로 분할될 경우, CU에 대응하는 노드는 4 개의 리프 노드 (leaf node)들을 포함하고, 4 개의 리프 노드들의 각각은 서브-CU 들 중 하나에 대응한다.

[0143] 큐드트리 데이터 구조의 각각의 노드는 대응하는 CU에 대한 싱택스 데이터를 제공할 수도 있다. 예를 들어, 큐드트리에서의 노드는, 노드에 대응하는 CU가 서브-CU 들로 분할되는지 여부를 표시하는 분할 플래그 (split flag)를 포함할 수도 있다. CU에 대한 싱택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU가 서브-CU 들로 분할되는지 여부에 종속될 수도 있다. CU가 추가로 분할되지 않을 경우, 그것은 리프-CU로서 지정된다. 이 개시물에서는, 원래의 리프-CU의 명시적 분할이 없더라도, 리프-CU의 4 개의 서브-CU들은 또한 리프-CU들로서 지정될 것이다. 예를 들어, 16x16 사이즈에서의 CU가 추가로 분할되지 않을 경우, 16x16 CU가 결코 분할되지 않았지만, 4 개의 8x8 서브-CU 들이 또한 리프-CU들로서 지정될 것이다.

[0144] CU가 사이즈 구분 (size distinction)을 가지지 않는다는 것을 제외하고는, CU는 H.264 표준의 매크로블록 (macroblock)과 유사한 목적을 가진다. 예를 들어, 트리블록은 4 개의 자식 노드 (child node)들 (또한, 서브-CU들로서 지정됨)로 분할될 수도 있고, 각각의 자식 노드는 궁극적으로 부모 노드 (parent node)일 수도 있고, 또 다른 4 개의 자식 노드들로 분할될 수도 있다. 큐드트리의 리프 노드로서 지정된, 최종적인 분할되지 않은 자식 노드는 리프-CU로서 또한 지정된 코딩 노드를 포함한다. 코딩된 비트스트림과 연관된 싱택스 데이터는 최대 CU 심도로서 지정된, 트리블록이 분할될 수도 있는 최대 횟수를 정의할 수도 있고, 또한, 코딩 노드들의 최소 사이즈를 정의할 수도 있다. 따라서, 비트스트림은 또한, 최소 코딩 유닛 (smallest coding unit; SCU)을 정의할 수도 있다. 이 개시물은 HEVC의 문맥에서 CU, PU, 또는 TU 중의 임의의 것, 또는 다른 표준들의 문맥에서 유사한 데이터 구조들 (예컨대, H.264/AVC에서 매크로블록들 및 그 서브-블록들)을 지정하기 위해 용어 "블록"을 이용한다.

[0145] CU는 코딩 노드와, 코딩 노드와 연관된 예측 유닛 (PU)들 및 변환 유닛 (TU)들을 포함한다. CU의 사이즈는 코딩 노드의 사이즈에 대응하고, 형상에 있어서 정사각형이어야 한다. CU의 사이즈는 8x8 픽셀들로부터 64x64 픽셀들 이상의 최대치를 갖는 트리블록의 사이즈까지의 범위일 수도 있다. 각각의 CU는 하나 이

상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다. CU 와 연관된 선택스 데이터는 예를 들어, 하나 이상의 PU 들로의 CU 의 파티셔닝을 설명할 수도 있다. 파티셔닝 모드들은 CU 가 스kip (skip) 또는 직접 모드 (direct mode) 인코딩되는지, 인트라-예측 모드 인코딩되는지, 또는 인터-예측 모드 인코딩되는지 여부의 사이에서 상이할 수도 있다. PU 들은 형상에 있어서 비-정사각형 (non-square) 이 되도록 파티셔닝될 수도 있다. CU 와 연관된 선택스 데이터는 또한, 예를 들어, 쿼드트리에 따른 하나 이상의 TU 들로의 CU 의 파티셔닝을 설명할 수도 있다. TU 는 형상에 있어서 정사각형 또는 비-정사각형 (예컨대, 직사각형) 일 수 있다.

[0146] HEVC 표준은 상이한 CU 들에 대해 상이할 수도 있는, TU 들에 따른 변환들을 허용한다. 이것은 항상 그러하지는 않을 수도 있지만, TU 들은 전형적으로, 파티셔닝된 LCU 에 대해 정의된 주어진 CU 내에서의 PU 들의 사이즈에 기초하여 사이즈가 정해진다. TU 들은 전형적으로 동일한 사이즈이거나 PU 들보다 더 작다. 일부의 예들에서, CU 에 대응하는 잔여 샘플들은 "잔여 쿼드 트리" (residual quad tree; RQT) 로서 알려진 쿼드트리 구조를 이용하여 더 작은 유닛들로 재분할될 수도 있다. RQT 의 리프 노드들은 변환 유닛 (TU) 들로서 지칭될 수도 있다. TU 들과 연관된 픽셀 차이 값들은 양자화될 수도 있는 변환 계수들을 생성하기 위해 변환될 수도 있다.

[0147] 리프-CU 는 하나 이상의 예측 유닛 (PU) 들을 포함할 수도 있다. 일반적으로, PU 는 대응하는 CU 의 전부 또는 부분에 대응하는 공간적 영역 (area) 을 나타내고, PU 에 대한 참조 샘플을 취출하기 위한 데이터를 포함할 수도 있다. 또한, PU 는 예측과 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라-모드 인코딩될 때, PU 에 대한 데이터는 PU 에 대응하는 TU 에 대한 인트라-예측 모드를 설명하는 데이터를 포함할 수도 있는 잔여 쿼드트리 (RQT) 내에 포함될 수도 있다. 또 다른 예로서, PU 가 인터-모드 인코딩될 때, PU 는 PU 에 대한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. PU 에 대한 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터에 대한 해상도 (예컨대, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 지시하는 참조 픽처, 및/또는 모션 벡터에 대한 참조 픽처 리스트 (예컨대, List 0, List 1, 또는 List C) 를 설명할 수도 있다.

[0148] 하나 이상의 PU 들을 갖는 리프-CU 는 또한, 하나 이상의 변환 유닛 (TU) 들을 포함할 수도 있다. 위에서 논의된 바와 같이, 변환 유닛들은 RQT (또한, TU 쿼드트리 구조로서 지칭됨) 를 이용하여 특정될 수도 있다. 예를 들어, 분할 플래그는 리프-CU 가 4 개의 변환 유닛들로 분할되는지 여부를 표시할 수도 있다. 다음으로, 각각의 변환 유닛은 추가의 서브-TU 들로 추가로 분할될 수도 있다. TU 가 추가로 분할되지 않을 때, 그것은 리프-TU 로서 지칭될 수도 있다. 일반적으로, 인트라 코딩을 위해, 리프-CU 에 속하는 모든 리프-TU 들은 동일한 인트라 예측 모드를 공유한다. 즉, 동일한 인트라-예측 모드는 리프-CU 의 모든 TU 들에 대한 예측된 값들을 계산하기 위해 일반적으로 적용된다. 인트라 코딩을 위해, 비디오 인코더는 TU 에 대응하는 CU 의 부분과 원래의 블록과의 사이의 차이로서, 인트라 예측 모드를 이용하여 각각의 리프-TU 에 대한 잔여 값을 계산할 수도 있다. TU 는 반드시 PU 의 사이즈로 제한되는 것은 아니다. 이에 따라, TU 들은 PU 보다 더 크거나 더 작을 수도 있다. 인트라 코딩을 위해, PU 는 동일한 CU 에 대한 대응하는 리프-TU 와 콜로케이팅 (collocate) 될 수도 있다. 일부의 예들에서, 리프-TU 의 최대 사이즈는 대응하는 리프-CU 의 사이즈에 대응할 수도 있다.

[0149] 또한, 리프-CU 들의 TU 들은 또한, 잔여 쿼드트리 (RQT) 들로서 지칭된 각각의 쿼드트리 데이터 구조들과 연관될 수도 있다. 즉, 리프-CU 는 리프-CU 가 어떻게 TU 들로 파티셔닝되는지를 표시하는 쿼드트리를 포함할 수도 있다. TU 쿼드트리의 루트 노드는 일반적으로 리프-CU 에 대응하는 반면, CU 쿼드트리의 루트 노드는 일반적으로 트리블록 (또는 LCU) 에 대응한다. 분할되지 않은 RQT 의 TU 들은 리프-TU 들로서 지칭된다. 일반적으로, 이와 다르게 언급되지 않으면, 이 개시물은 리프-CU 및 리프-TU 를 각각 지칭하기 위해 용어들 CU 및 TU 를 이용한다.

[0150] 비디오 시퀀스는 전형적으로 일련의 비디오 프레임들 또는 픽처들을 포함한다. 픽처들의 그룹 (group of pictures; GOP) 은 일반적으로 비디오 픽처들 중의 일련의 하나 이상을 포함한다. GOP 는 GOP 의 헤더, 픽처들 중 하나 이상의 픽처의 헤더, 또는 다른 곳에서의 선택스 데이터로서, GOP 에 포함된 픽처들의 수를 설명하는 상기 선택스 데이터를 포함할 수도 있다. 픽처의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 설명하는 슬라이스 선택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 전형적으로 비디오 데이터를 인코딩하기 위해 개별적인 비디오 슬라이스들 내의 비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정된 또는 변동되는 사이즈들을 가질 수도

있고, 특정된 코딩 표준에 따라 사이즈에 있어서 상이할 수도 있다.

[0151] 일 예로서, HM은 다양한 PU 사이즈들에 있어서의 예측을 지원한다. 특별한 CU의 사이즈가 $2Nx2N$ 인 것으로 가정하면, HM은 $2Nx2N$ 또는 NxN 의 PU 사이즈들에 있어서의 인트라-예측과, $2Nx2N$, $2NxN$, $Nx2N$, 또는 NxN 의 대칭적인 PU 사이즈들에 있어서의 인터-예측을 지원한다. HM은 또한, $2NxN$, $2NxN$, $nLx2N$, 및 $nRx2N$ 의 PU 사이즈들에 있어서의 인터-예측을 위한 비대칭적 파티셔닝을 지원한다. 비대칭적 파티셔닝에서는, CU의 하나의 방향이 파티셔닝되지 않는 반면, 다른 방향은 25 % 및 75 %로 파티셔닝된다. 25 % 파티션에 대응하는 CU의 부분은 "n"과, 그 다음으로, "상부", "하부", "좌측", 또는 "우측"의 표시에 의해 표시된다. 이에 따라, 예를 들어, " $2NxN$ "는 상부의 $2Nx0.5N$ PU 및 하부의 $2Nx1.5N$ PU로 수평으로 파티셔닝되는 $2Nx2N$ CU를 지칭한다.

[0152] 이 개시물에서, " NxN " 및 "N 바이 (by) N"은 수직 및 수평 차원들의 측면에서의 비디오 블록의 픽셀 차원들, 예컨대, $16x16$ 픽셀들 또는 16 대 16 픽셀들을 지칭하기 위해 상호 교환가능하게 이용될 수도 있다. 일반적으로, $16x16$ 블록은 수직 방향에서의 16 개의 픽셀들 ($y = 16$) 및 수평 방향에서의 16 개의 픽셀들 ($x = 16$)을 가질 것이다. 마찬가지로, NxN 블록은 일반적으로 수직 방향에서의 N 개의 픽셀들 및 수평 방향에서의 N 개의 픽셀들을 가지며, 여기서, N은 음이 아닌 정수 값을 나타낸다. 블록 내의 픽셀들은 행 (row) 들 및 열 (column) 들로 배치될 수도 있다. 또한, 블록들은 수직 방향에서와 동일한 수의 픽셀들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, 블록들은 NxM 픽셀들을 포함할 수도 있으며, 여기서, M은 반드시 N과 동일하지는 않다.

[0153] CU의 PU들을 이용한 인트라-예측 또는 인터-예측 코딩에 후속하여, 비디오 인코더 (20)는 CU의 TU들에 대한 잔여 데이터를 계산할 수도 있다. PU들은 공간 도메인 (또한 픽셀 도메인으로서 지칭됨)에서 예측 픽셀 데이터를 생성하는 방법 또는 모드를 설명하는 신팩스 데이터를 포함할 수도 있고, TU들은 변환, 예컨대, 이산 사인 변환 (discrete sine transform; DST), 이산 코사인 변환 (discrete cosine transform; DCT), 정수 변환, 웨이블렛 변환 (wavelet transform), 또는 개념적으로 유사한 변환의 잔여 비디오 데이터에 대한 적용에 후속하는 변환 도메인에서의 계수들을 포함할 수도 있다. 잔여 데이터는 인코딩되지 않은 픽처의 픽셀들과 PU들에 대응하는 예측 값들과의 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20)는 CU에 대한 잔여 데이터를 포함하는 TU들을 형성할 수도 있고, 그 다음으로, CU에 대한 변환 계수들을 생성하기 위해 TU들을 변환할 수도 있다.

[0154] 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (20)는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 그 가장 광범위한 통상의 의미를 갖도록 의도된 광범위한 용어가다. 하나의 실시형태에서, 양자화는 계수들을 나타내기 위해 이용된 데이터의 양을 가능한 대로 감소시키기 위해 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 값은 양자화 동안에 m-비트 값으로 버림 (round down) 될 수도 있으며, 여기서, n은 m 보다 더 크다.

[0155] 양자화에 후속하여, 비디오 인코더는 변환 계수들을 스캔 (scan) 하여, 양자화된 변환 계수들을 포함하는 2 차원 행렬로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 더 높은 에너지 (및 이에 따라, 더 낮은 주파수) 계수들을 어레이의 전방에 배치하고 더 낮은 에너지 (및 이에 따라, 더 높은 주파수) 계수들을 어레이의 후방에 배치하도록 설계될 수도 있다. 일부의 예들에서, 비디오 인코더 (20)는 양자화된 변환 계수들을 스캔하여 엔트로피 인코딩될 수 있는 직렬화된 벡터 (serialized vector)를 생성하기 위해, 미리 정의된 스캔 순서를 사용할 수도 있다. 다른 예들에서, 비디오 인코더 (20)는 적응적 스캔 (adaptive scan)을 수행할 수도 있다. 1 차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캔한 후, 비디오 인코더 (20)는 예컨대, 컨텍스트-적용 가변 길이 코딩 (context-adaptive variable length coding; CAVLC), 컨텍스트-적용 2 진 산술 코딩 (context-adaptive binary arithmetic coding; CABAC), 신팩스-기반 컨텍스트-적용 2 진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 간격 파티셔닝 엔트로피 (Probability Interval Partitioning Entropy; PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라 1 차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20)는 또한, 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30)에 의한 이용을 위한 인코딩된 비디오 데이터와 연관된 신팩스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0156] CABAC를 수행하기 위해, 비디오 인코더 (20)는 컨텍스트 모델 내의 컨텍스트를 송신되어야 할 심볼에 배정할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 비-제로 (non-zero)인지 아닌지의 여부와 관련

있을 수도 있다. CAVLC 를 수행하기 위해, 비디오 인코더 (20) 는 송신되어야 할 심볼에 대한 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드 (codeword) 들은, 상대적으로 더 짧은 코드들이 더 확률이 큰 심볼들에 대응하는 반면, 더 긴 코드들은 더 확률이 작은 심볼들에 대응하도록 구성될 수도 있다. 이러한 방법으로, VLC 의 이용은 예를 들어, 송신되어야 할 각각의 심볼에 대한 동일-길이 코드워드들을 이용하는 것에 비해 비트 절감을 달성할 수도 있다. 확률 결정은 심볼에 배정된 컨텍스트에 기초할 수도 있다.

[0157] 비디오 인코더 (20) 는 예컨대, 프레임 헤더, 블록 헤더, 슬라이스 헤더, 또는 GOP 헤더에서, 블록-기반 십екс 데이터, 프레임-기반 십екс 데이터, 및 GOP-기반 십екс 데이터와 같은 십екс 데이터를 비디오 디코더 (30) 로 추가로 전송할 수도 있다. GOP 십екс 데이터는 각각의 GOP 에서의 다수의 프레임들을 설명할 수도 있고, 프레임 십екс 데이터는 대응하는 프레임을 인코딩하기 위해 이용된 인코딩/예측 모드를 표시할 수도 있다.

비디오 인코더

[0159] 도 2a 는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 인코더의 예를 예시하는 블록도이다. 비디오 인코더 (20) 는 HEVC 에 대한 것과 같은 비디오 비트스트림의 단일 계층을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 인코더 (20) 는, 도 4 내지 도 6 에 대하여 상기에서 그리고 이하에서 더 옥상세하게 설명된 계층간 예측 시그널링 및 관련된 프로세스들을 수행하는 방법들을 포함하지만 이것으로 제한되지는 않는 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 하나의 예로서, (제공될 때) 계층간 예측 유닛 (66) 은 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 그러나, 이 개시물의 양태들은 그렇게 제한되지 않는다. 일부의 예들에서, 이 개시물에서 설명된 기법들은 비디오 인코더 (20) 의 다양한 컴포넌트들 사이에서 공유될 수도 있다. 일부의 예들에서, 추가적으로 또는 대안적으로, 프로세서 (도시되지 않음) 는 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0160] 설명의 목적들을 위해, 이 개시물은 HEVC 코딩의 문맥에서 비디오 인코더 (20) 를 설명한다. 그러나, 이 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 2a 의 인코더 (20) 는 코덱의 단일 계층을 예시한다. 그러나, 도 2b 에 대하여 추가로 설명되는 바와 같이, 비디오 인코더 (20) 의 일부 또는 전부는 멀티-계층 코덱에 따른 프로세싱에 대해 중복될 수도 있다.

[0161] 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라-예측, 인터-예측, 및 계층간 예측 (때로 인트라-코딩, 인터-코딩 또는 계층간 코딩으로 지칭됨) 을 수행할 수도 있다. 인트라 코딩은 주어진 비디오 프레임 또는 픽처 내에서의 비디오에 있어서의 공간적 리던던시를 감소시키거나 제거하기 위해 공간적 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내의 비디오에 있어서의 시간적 리던던시를 감소시키거나 제거하기 위해 시간적 예측에 의존한다. 계층간 코딩은 동일한 비디오 코딩 시퀀스 내의 상이한 계층 (들) 내의 비디오에 기초한 예측에 의존한다. 인트라-모드 (I 모드) 는 몇몇 공간기반 코딩 모드들 중의 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 과 같은 인터-모드들은 몇몇 시간 기반 코딩 모드들 중의 임의의 것을 지칭할 수도 있다.

[0162] 도 2a 에서 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩되어야 할 비디오 프레임 내의 현재의 비디오 블록을 수신한다. 도 2a 의 예에서, 비디오 인코더 (20) 는 모드 선택 유닛 (40), 참조 프레임 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 모드 선택 유닛 (40) 은 궁극적으로, 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라 예측 유닛 (46), 계층간 예측 유닛 (66) 및 파티션 유닛 (48) 을 포함한다. 참조 프레임 메모리 (64) 는 디코딩된 픽처 베파를 포함할 수도 있다. 디코딩된 픽처 베파는 그 통상의 의미를 갖는 광범위한 용어이고, 일부의 실시형태들에서, 참조 프레임들의 비디오 코덱-관리된 데이터 구조를 지칭한다.

[0163] 비디오 블록 복원을 위해, 비디오 인코더 (20) 는 또한, 역양자화 유닛 (58), 역변환 유닛 (60), 및 합산기 (62) 를 포함한다. 디블록킹 필터 (deblocking filter; 도 2 에서 도시되지 않음) 는 또한, 블록 경계들을 필터링하여 복원된 비디오로부터 블록화 아티팩트 (blockiness artifact) 들을 제거하기 위해 포함될 수도 있다. 희망하는 경우, 디블록킹 필터는 전형적으로 합산기 (62) 의 출력을 필터링할 것이다. (루프 내의 또는 루프 이후의) 추가적인 필터들은 또한, 디블록킹 필터에 추가하여 이용될 수도 있다. 이러한 필터들은 간결함을 위해 도시되어 있지 않지만, 희망하는 경우, (루프내 (in-loop) 필터로서) 합산기 (50) 의 출력

을 필터링 할 수도 있다.

[0164] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩되어야 할 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 다수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 시간적 예측을 제공하기 위해, 하나 이상의 참조 프레임들 내의 하나 이상의 블록들과 관련하여 수신된 비디오 블록의 인터-예측 코딩을 수행한다. 인트라 예측 유닛 (46) 은 대안적으로, 공간적 예측을 제공하기 위해, 코딩되어야 할 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들과 관련하여 수신된 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다. 비디오 인코더 (20) 는 예컨대, 비디오 데이터의 각각의 블록에 대한 적절한 코딩 모드를 선택하기 위해, 다수의 코딩 과정 (coding pass) 들을 수행할 수도 있다.

[0165] 또한, 파티션 유닛 (48) 은 이전의 코딩 과정들에서의 이전의 파티셔닝 방식들의 평가에 기초하여, 비디오 데이터의 블록들을 서브-블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛 (48) 은 초기에, 프레임 또는 슬라이스를 LCU 들로 파티셔닝할 수도 있고, 레이트-왜곡 (rate-distortion) 분석 (예컨대, 레이트-왜곡 최적화 등) 에 기초하여 LCU 들의 각각을 서브-CU 들로 파티셔닝할 수도 있다. 모드 선택 유닛 (40) 은 서브-CU 들로의 LCU 의 파티셔닝을 표시하는 큐드트리 데이터 구조를 추가로 생성할 수도 있다. 큐드트리의 리프-노드 (leaf-node) CU 들은 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다.

[0166] 모드 선택 유닛 (40) 은 예컨대, 여러 결과들에 기초하여 코딩 모드들 중 하나, 인트라, 인터, 또는 계층간 예측 모드를 선택할 수도 있고, 결과적인 인트라-코딩된, 인터-코딩된, 또는 계층간 코딩된 블록을, 잔여 블록 데이터를 생성하기 위해 합산기 (50) 에, 그리고 참조 프레임으로서의 이용을 위한 인코딩된 블록을 복원하기 위해 합산기 (62) 에 제공할 수도 있다. 모드 선택 유닛 (40) 은 또한, 모션 벡터들, 인트라-모드 표시자들, 파티션 정보, 및 다른 이러한 신택스 정보와 같은 신택스 엘리먼트들을 엔트로피 인코딩 유닛 (56) 에 제공한다.

[0167] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적인 목적들을 위해 별도로 예시되어 있다. 모션 추정 유닛 (42) 에 의해 수행된 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는 예를 들어, 현재의 프레임 (또는 다른 코딩된 유닛) 내에서 코딩되고 있는 현재 블록과 관련된 참조 프레임 (또는 다른 코딩된 유닛) 내의 예측 블록과 관련된 현재의 비디오 프레임 또는 픽처 내의 비디오 블록의 PU 의 변위를 표시할 수도 있다. 예측 블록은, 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는 픽셀 차이의 측면에서, 코딩되어야 할 블록과 근접하게 일치하는 것으로 판명되는 블록이다. 일부의 예들에서, 비디오 인코더 (20) 는 참조 프레임 메모리 (64) 내에 저장된 참조 픽처들의 정수 미만 (sub-integer) 픽셀 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 픽처의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 보간할 수도 있다. 그러므로, 모션 추정 유닛 (42) 은 전체 픽셀 위치들 및 분수 픽셀 위치들과 관련하여 모션 검색을 수행할 수도 있고, 분수 픽셀 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0168] 모션 추정 유닛 (42) 은 PU 의 위치를 참조 픽처의 예측 블록의 위치와 비교함으로써, 인터-코딩된 슬라이스에서의 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 참조 픽처는 제 1 참조 픽처 리스트 (List 0) 또는 제 2 참조 픽처 리스트 (List 1) 로부터 선택될 수도 있고, 이들의 각각은 참조 프레임 메모리 (64) 내에 저장된 하나 이상의 참조 픽처들을 식별한다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0169] 모션 보상 유닛 (44) 에 의해 수행된 모션 보상은 모션 추정 유닛 (42) 에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페치 (fetch) 하거나 생성하는 것을 포함할 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 일부의 예들에서 기능적으로 통합될 수도 있다. 현재의 비디오 블록의 PU 에 대한 모션 벡터를 수신할 시에, 모션 보상 유닛 (44) 은 모션 벡터가 참조 픽처 리스트들 중 하나에서 지시하는 예측 블록을 위치시킬 수도 있다. 합산기 (50) 는 이하에서 논의된 바와 같이, 코딩되고 있는 현재의 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔여 비디오 블록을 형성한다. 일부의 실시형태들에서, 모션 추정 유닛 (42) 은 루마 컴포넌트들과 관련하여 모션 추정을 수행할 수 있고, 모션 보상 유닛 (44) 은 크로마 컴포넌트들 및 루마 컴포넌트들의 양자 모두에 대한 루마 컴포넌트들에 기초하여 계산된 모션 벡터들을 이용할 수 있다. 모드 선택 유닛 (40) 은 비디오 슬라이스의 비디오 블록들을 디코딩 함에 있어서 비디오 디코더 (30) 에 의해 이용하기 위한 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘

리먼트들을 생성할 수도 있다.

[0170] 인트라 예측 유닛 (46)은 위에서 설명된 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 수행된 인터-예측에 대한 대안으로서, 현재 블록을 인트라-예측하거나 계산할 수도 있다. 특히, 인트라 예측 유닛 (46)은 현재 블록을 인코딩하기 위해 이용할 인트라-예측 모드를 결정할 수도 있다. 일부의 예들에서, 인트라 예측 유닛 (46)은 예컨대, 별도의 인코딩 과정들 동안에 다양한 인트라-예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 유닛 (46) (또는 일부의 예들에서, 모드 선택 유닛 (40))은 테스팅된 모드들로부터 이용하기 위한 적절한 인트라-예측 모드를 선택할 수도 있다.

[0171] 예를 들어, 인트라 예측 유닛 (46)은 다양한 테스팅된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 이용하여 레이트-왜곡 값들을 계산할 수도 있고, 테스팅된 모드들 중에서 최상의 레이트-왜곡 특성을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로, 인코딩된 블록과, 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록과의 사이의 왜곡 (또는 에러)의 양뿐만 아니라, 인코딩된 블록을 생성하기 위해 이용된 비트레이트 (즉, 비트들의 수)를 결정한다. 인트라 예측 유닛 (46)은 어느 인트라-예측 모드가 블록에 대한 최상의 레이트-왜곡 값을 나타내는지를 결정하기 위해 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율 (ratio)들을 계산할 수도 있다.

[0172] 블록에 대한 인트라-예측 모드를 선택한 후, 인트라 예측 유닛 (46)은 블록에 대한 선택된 인트라-예측 모드를 표시하는 정보를 엔트로피 인코딩 유닛 (56)에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56)은 선택된 인트라-예측 모드를 표시하는 정보를 인코딩할 수도 있다. 비디오 인코더 (20)는 복수의 인트라-예측 모드 인덱스 표들 및 복수의 수정된 인트라-예측 모드 인덱스 표들 (또한 코드워드 맵핑 표 (codeword mapping table) 들로서 지칭됨)을 포함할 수도 있는 송신된 비트스트림 구성 데이터에서, 다양한 블록들에 대한 인코딩 컨텍스트들의 정의들과, 컨텍스트들의 각각에 대해 이용하기 위한 가장 가능성 있는 인트라-예측 모드, 인트라-예측 모드 인덱스 표, 및 수정된 인트라-예측 모드 인덱스 표의 표시들을 포함할 수도 있다.

[0173] 비디오 인코더 (20)는 계층간 예측 유닛 (66)을 포함할 수도 있다. 계층간 예측 유닛 (66)은 SVC에서 이용 가능한 하나 이상의 상이한 계층들 (예컨대, 기본 또는 참조 계층)을 이용하여 현재 블록 (예컨대, EL에서의 현재 블록)을 예측하도록 구성된다. 이러한 예측은 계층간 예측으로서 지칭될 수도 있다. 계층간 예측 유닛 (66)은 계층간 리던던시를 감소시키기 위한 예측 방법들을 사용함으로써, 코딩 효율을 개선시키고 연산 자원 요건들을 감소시킨다. 계층간 예측의 일부의 예들은 계층간 인트라 예측, 계층간 모션 예측, 및 계층간 잔여 예측을 포함한다. 계층간 인트라 예측은 강화 계층에서 현재 블록을 예측하기 위해 기본 계층에서의 공동-위치된 블록들의 복원을 이용한다. 계층간 모션 예측은 강화 계층에서 모션을 예측하기 위해 기본 계층의 모션 정보를 이용한다. 계층간 잔여 예측은 강화 계층의 잔여를 예측하기 위해 기본 계층의 잔여를 이용한다. 기본 및 강화 계층들이 상이한 공간적 해상도들을 가질 때, 이하에서 더욱 상세하게 설명된 바와 같이, 시간적 스케일링 함수를 이용한 공간적 모션 벡터 스케일링 및/또는 계층간 위치 맵핑은 계층간 예측 유닛 (66)에 의해 수행될 수도 있다.

[0174] 비디오 인코더 (20)는 코딩되고 있는 원래의 비디오 블록으로부터 모드 선택 유닛 (40)으로부터의 예측 데이터를 감산함으로써 잔여 비디오 블록을 형성한다. 합산기 (50)는 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 프로세싱 유닛 (52)은 이산 코사인 변환 (discrete cosine transform; DCT) 또는 개념적으로 유사한 변환과 같은 변환을 잔여 블록에 적용하여, 잔여 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 유닛 (52)은 DCT 와 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 예를 들어, 이산 사인 변환 (discrete sine transform; DST) 들, 웨이블렛 변환들, 정수 변환들, 서브-대역 변환 (sub-band transform) 들 또는 다른 타입들의 변환들이 또한 이용될 수 있다.

[0175] 변환 프로세싱 유닛 (52)은 변환을 잔여 블록에 적용하여, 잔여 변환 계수들의 블록을 생성할 수 있다. 변환은 잔여 정보를 픽셀 값 도메인으로부터, 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다. 변환 프로세싱 유닛 (52)은 결과적인 변환 계수들을 양자화 유닛 (54)으로 전송할 수도 있다. 양자화 유닛 (54)은 비트 레이트를 추가로 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화도 (degree of quantization)는 양자화 파라미터를 조절함으로써 수정될 수도 있다. 일부의 예들에서, 다음으로, 양자화 유닛 (54)은 양자화된 변환 계수들을 포함하는 행렬 (matrix)의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56)은 스캔을 수행할 수도 있다.

[0176] 양자화에 후속하여, 엔트로피 인코딩 유닛 (56)은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들

어, 엔트로피 인코딩 유닛 (56)은 컨텍스트-적응 가변 길이 코딩 (context adaptive variable length coding; CAVLC), 컨텍스트 적응 2 진 산술 코딩 (CABAC), 신택스-기반 컨텍스트-적응 2 진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 간격 파티셔닝 엔트로피 (probability interval partitioning entropy; PIPE) 코딩 또는 또 다른 엔트로피 코딩 기법을 수행할 수도 있다. 컨텍스트-기반 엔트로피 코딩의 경우, 컨텍스트는 이웃하는 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56)에 의한 엔트로피 코딩에 후속하여, 인코딩된 비트스트림은 또 다른 디바이스 (예컨대, 비디오 디코더 (30))에 송신될 수도 있거나, 더 이후의 송신 또는 취출을 위해 아카이빙될 수도 있다.

[0177] 역양자화 유닛 (58) 및 역변환 유닛 (60)은 (예컨대, 참조 블록으로서의 더 이후의 이용을 위한) 픽셀 도메인에서의 잔여 블록을 복원하기 위해 역양자화 및 역변환을 각각 적용한다. 모션 보상 유닛 (44)은 잔여 블록을 참조 프레임 메모리 (64)의 프레임들 중 하나의 프레임의 예측 블록에 추가함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44)은 또한, 모션 추정 시에 이용하기 위한 정수 미만 픽셀 값들을 계산하기 위해 하나 이상의 보간 필터들을 복원된 잔여 블록에 적용할 수도 있다. 합산기 (62)는 참조 프레임 메모리 (64)에서의 저장을 위한 복원된 비디오 블록을 생성하기 위해, 복원된 잔여 블록을, 모션 보상 유닛 (44)에 의해 생성된 모션 보상된 예측 블록에 추가한다. 복원된 비디오 블록은 추후의 비디오 프레임에서 블록을 인터-코딩하기 위해, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 참조 블록으로서 이용될 수도 있다.

멀티-계층 비디오 인코더

[0179] 도 2b 는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 멀티-계층 비디오 인코더 (21)의 예를 예시하는 블록도이다. 비디오 인코더 (21)는 SHVC 및 멀티뷰 코딩을 위한 것과 같은 멀티-계층 비디오 프레임들을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 인코더 (21)는 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0180] 비디오 인코더 (21)는 비디오 인코더 (20A) 및 비디오 인코더 (20B)를 포함하고, 그 각각은 도 2a 의 비디오 인코더 (20)로서 구성될 수도 있으며 비디오 인코더 (20)에 대하여 위에서 설명된 기능들을 수행할 수도 있다. 또한, 참조 번호들의 제이용에 의해 표시된 바와 같이, 비디오 인코더들 (20A 및 20B)은 비디오 인코더 (20)로서 시스템들 및 서브시스템들의 적어도 일부를 포함할 수도 있다. 비디오 인코더 (21)는 2 개의 비디오 인코더들 (20A 및 20B)을 포함하는 것으로서 예시되어 있지만, 비디오 인코더 (21)는 이와 같이 제한되지 않으며 임의의 수의 비디오 인코더 (20) 계층들을 포함할 수도 있다. 일부의 실시형태들에서, 비디오 인코더 (21)는 액세스 유닛에서의 각각의 픽처 또는 프레임에 대한 비디오 인코더 (20)를 포함할 수도 있다. 예를 들어, 5 개의 픽처들을 포함하는 액세스 유닛은 5 개의 인코더 계층들을 포함하는 비디오 인코더에 의해 프로세싱되거나 인코딩될 수도 있다. 일부의 실시형태들에서, 비디오 인코더 (21)는 액세스 유닛에서의 프레임들보다 더 많은 인코더 계층들을 포함할 수도 있다. 일부의 이러한 경우들에 있어서, 비디오 인코더 계층들의 일부는 일부의 액세스 유닛들을 프로세싱할 때에 비활성 (inactive) 일 수도 있다.

[0181] 비디오 인코더들 (20A 및 20B)에 추가하여, 비디오 인코더 (21)는 리샘플링 유닛 (resampling unit; 90)을 포함할 수도 있다. 리샘플링 유닛 (90)은 일부의 경우들에 있어서, 예를 들어, 강화 계층을 생성하기 위해 수신된 비디오 프레임의 기본 계층을 업샘플링 (upsampling) 할 수도 있다. 리샘플링 유닛 (90)은 프레임의 수신된 기본 계층과 연관된 특별한 정보를 업샘플링할 수도 있지만, 다른 정보는 그러하지 않다. 예를 들어, 리샘플링 유닛 (90)은 기본 계층의 픽셀들의 공간적 사이즈 또는 수를 업샘플링할 수도 있지만, 슬라이스들의 수 또는 픽처 순서 카운트는 일정하게 남아 있을 수도 있다. 일부의 경우들에는, 리샘플링 유닛 (90)이 수신된 비디오를 프로세싱하지 않을 수도 있고 및/또는 선택적일 수도 있다. 예를 들어, 일부의 경우에는, 모드 선택 유닛 (40)이 업샘플링을 수행할 수도 있다. 일부의 실시형태들에서, 리샘플링 유닛 (90)은 계층을 업샘플링하며, 슬라이스 경계 규칙들 및/또는 레스터 스캔 (raster scan) 규칙들의 세트를 준수하기 위해 하나 이상의 슬라이스들을 인식하거나, 재정의하거나, 수정하거나, 조절하도록 구성된다. 액세스 유닛에서의 기본 계층 또는 더 낮은 계층을 업샘플링하는 것으로 주로 설명되지만, 일부의 경우들에는, 리샘플링 유닛 (90)이 계층을 다운샘플링 (downsampling) 할 수도 있다. 예를 들어, 비디오의 스트리밍 동안에 대역폭이 감소될 경우, 프레임은 업샘플링되는 대신에 다운샘플링될 수도 있다. 리샘플링 유닛 (90)은 마찬가지로, 크롭핑 (cropping) 및/또는 패딩 (padding) 연산들을 수행하도록 추가로 구성될 수도 있다.

[0182] 리샘플링 유닛 (90)은 더 낮은 계층의 인코더 (예컨대, 비디오 인코더 (20A))의 디코딩된 픽처 버퍼 (114)로

부터 픽처 또는 프레임 (또는 픽처와 연관된 픽처 정보) 을 수신하고 픽처 (또는 수신된 픽처 정보) 를 업샘플링하도록 구성될 수도 있다. 다음으로, 이 업샘플링된 픽처는 더 낮은 계층의 인코더와 동일한 액세스 유닛에서의 픽처를 인코딩하도록 구성된 더 높은 계층의 인코더 (예컨대, 비디오 인코더 (20B)) 의 모드 선택 유닛 (40) 에 제공될 수도 있다. 일부의 경우들에 있어서, 더 높은 계층의 인코더는 더 낮은 계층의 인코더로부터 제거된 하나의 계층이다. 다른 경우에는, 도 2b 의 계층 0 비디오 인코더 및 계층 1 인코더 사이에 하나 이상의 더 높은 계층의 인코더들이 있을 수도 있다.

[0183] 일부의 경우에는, 리샘플링 유닛 (90) 이 생략되거나 바이패스될 수도 있다. 이러한 경우에는, 비디오 인코더 (20A) 의 디코딩된 픽처 버퍼 (64) 로부터의 픽처가 직접적으로, 또는 적어도 리샘플링 유닛 (90) 에 제공되지 않으면서, 비디오 인코더 (20B) 의 모드 선택 유닛 (40) 에 제공될 수도 있다. 예를 들어, 비디오 인코더 (20B) 에 제공된 비디오 데이터와, 비디오 인코더 (20A) 의 디코딩된 픽처 버퍼 (64) 로부터의 참조 픽처가 동일한 사이즈 또는 해상도일 경우, 참조 픽처는 임의의 리샘플링 없이 비디오 인코더 (20B) 에 제공될 수도 있다.

[0184] 일부의 실시형태들에서, 비디오 인코더 (21) 는 비디오 데이터를 비디오 인코더 (20A) 에 제공하기 전에, 다운샘플링 유닛 (94) 을 이용하여 더 낮은 계층의 인코더에 제공되어야 할 비디오 데이터를 다운샘플링한다. 대안적으로, 다운샘플링 유닛 (94) 은 비디오 데이터를 업샘플링하거나 다운샘플링할 수 있는 리샘플링 유닛 (90) 일 수도 있다. 또 다른 실시형태들에서는, 다운샘플링 유닛 (94) 이 생략될 수도 있다.

[0185] 도 2b 에서 예시된 바와 같이, 비디오 인코더 (21) 는 멀티플렉서 (98) 또는 mux 를 더 포함할 수도 있다. mux (98) 는 비디오 인코더 (21) 로부터의 조합된 비트스트림을 출력할 수 있다. 조합된 비트스트림은 비디오 인코더들 (20A 및 20B) 의 각각으로부터의 비트스트림을 취합으로써, 그리고 어느 비트스트림이 주어진 시간에 출력되는지를 교번함으로써 생성될 수도 있다. 일부의 경우에는, 2 개의 (또는 2 개를 초과하는 비디오 인코더 계층들의 경우에는 더 많음) 비트스트림들로부터의 비트들이 한 번에 하나의 비트씩 교번될 수도 있지만, 많은 경우에는 비트스트림들이 상이하게 조합된다. 예를 들어, 출력 비트스트림은 선택된 비트스트림을 한 번에 하나의 블록씩 교번함으로써 생성될 수도 있다. 또 다른 예에서, 출력 비트스트림은 비디오 인코더들 (20A 및 20B) 의 각각으로부터의 비-1:1 (non-1:1) 비율의 블록들을 출력함으로써 생성될 수도 있다.

예를 들어, 2 개의 블록들은 비디오 인코더 (20A) 로부터 출력된 각각의 블록에 대하여 비디오 인코더 (20B) 로부터 출력될 수도 있다. 일부의 실시형태들에서, mux (98) 로부터의 출력 스트림은 미리 프로그래밍될 수도 있다. 다른 실시형태들에서, mux (98) 는 소스 디바이스 (12) 상의 프로세서로부터와 같이, 비디오 인코더 (21) 의 외부의 시스템으로부터 수신된 제어 신호에 기초하여 비디오 인코더들 (20A, 20B) 로부터의 비트스트림들을 조합할 수도 있다. 제어 신호는 비디오 소스 (18) 로부터의 비디오의 해상도 또는 비트레이트에 기초하여, 채널 (16) 의 대역폭에 기초하여, 사용자와 연관된 가입 (예컨대, 유료 가입 대 무료 가입) 에 기초하여, 또는 비디오 인코더 (21) 로부터 희망된 해상도 출력을 결정하기 위한 임의의 다른 인자에 기초하여 생성될 수도 있다.

비디오 디코더

[0186] 도 3a 는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 비디오 디코더의 예를 예시하는 블록도이다. 비디오 디코더 (30) 는 HEVC 에 대한 것과 같은 비디오 비트스트림의 단일 계층을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 디코더 (30) 는, 도 4 내지 도 6 에 대하여 상기에서 그리고 이하에서 더 옥 상세하게 설명된 계층간 예측 시그널링 및 관련된 프로세스들을 수행하는 방법들을 포함하지만 이것으로 제한되지는 않는 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 하나의 예로서, 계층간 예측 유닛 (75) 은 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다. 그러나, 이 개시물의 양태들은 그렇게 제한되지 않는다. 일부의 예들에서, 이 개시물에서 설명된 기법들은 비디오 디코더 (30) 의 다양한 컴포넌트들 사이에서 공유될 수도 있다. 일부의 예들에서, 추가적으로 또는 대안적으로, 프로세서 (도시되지 않음) 는 이 개시물에서 설명된 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0187] 설명의 목적들을 위해, 이 개시물은 HEVC 코딩의 문맥에서 비디오 디코더 (30) 를 설명한다. 그러나, 이 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 3a 의 디코더 (30) 는 코덱의 단일 계층을 예시한다. 그러나, 도 3b 에 대하여 추가로 설명되는 바와 같이, 비디오 디코더 (30) 의 일부 또는 전부는 멀티-계층 코덱에 따른 프로세싱에 대해 중복될 수도 있다.

- [0189] 도 3a의 예에서, 비디오 디코더 (30)는 엔트로피 디코딩 유닛 (70), 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 계층간 예측 유닛 (75), 역양자화 유닛 (76), 역변환 유닛 (78), 참조 프레임 메모리 (82) 및 합산기 (80)를 포함한다. 일부의 실시형태들에서, 모션 보상 유닛 (72) 및/또는 인트라 예측 유닛 (74)은 계층간 예측을 수행하도록 구성될 수도 있고, 이 경우, 계층간 예측 유닛 (75)은 생략될 수도 있다. 일부의 예들에서, 비디오 디코더 (30)는 비디오 인코더 (20) (도 2a)에 대하여 설명된 인코딩 과정과 일반적으로 상반되는 디코딩 과정을 수행할 수도 있다. 모션 보상 유닛 (72)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있는 반면, 인트라 예측 유닛 (74)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 인트라-예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다. 참조 프레임 메모리 (82)는 디코딩된 픽처 버퍼를 포함할 수도 있다. 디코딩된 픽처 버퍼는 그 통상의 의미를 갖는 광범위한 용어이고, 일부의 실시형태들에서, 참조 프레임들의 비디오 코덱-관리된 데이터 구조를 지칭한다.
- [0190] 디코딩 프로세스 동안, 비디오 디코더 (30)는 비디오 인코더 (20)로부터, 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 양자화된 계수들, 모션 벡터들 또는 인트라-예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 생성하기 위해 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (70)은 모션 벡터들 및 다른 신택스 엘리먼트들을 모션 보상 유닛 (72)으로 포워딩한다. 비디오 디코더 (30)는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.
- [0191] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 인트라 예측 유닛 (74)은 시그널링된 인트라 예측 모드와, 현재의 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (예컨대, 즉, B, P 또는 GPB) 슬라이스로서 코딩될 때, 모션 보상 유닛 (72)은 모션 벡터들과, 엔트로피 디코딩 유닛 (70)으로부터 수신된 다른 신택스 엘리먼트들에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 픽처 리스트들 중 하나 내의 참조 픽처들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30)는 참조 프레임 메모리 (82) 내에 저장된 참조 픽처들에 기초하여, 디폴트 구성 (default construction) 기법들을 이용하여 참조 프레임 리스트들, List 0 및 List 1을 구성할 수도 있다. 모션 보상 유닛 (72)은 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱함으로써 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 디코딩되고 있는 현재의 비디오 블록에 대한 예측 블록들을 생성하기 위해 예측 정보를 이용한다. 예를 들어, 모션 보상 유닛 (72)은 비디오 슬라이스의 비디오 블록들을 코딩하기 위해 이용된 예측 모드 (예컨대, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 타입 (예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 픽처 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재의 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위해, 수신된 신택스 엘리먼트들의 일부를 이용한다.
- [0192] 모션 보상 유닛 (72)은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (72)은 참조 블록들의 정수 미만 픽셀들에 대한 보간된 값들을 계산하기 위해, 비디오 블록들의 인코딩 동안에 비디오 인코더 (20)에 의해 이용된 바와 같은 보간 필터들을 이용할 수도 있다. 이 경우, 모션 보상 유닛 (72)은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 이용된 보간 필터들을 결정할 수도 있고, 예측 블록들을 생성하기 위해 보간 필터들을 이용할 수도 있다.
- [0193] 비디오 디코더 (30)는 또한, 계층간 예측 유닛 (75)을 포함할 수도 있다. 계층간 예측 유닛 (75)은 SVC에서 이용가능한 하나 이상의 상이한 계층들 (예컨대, 기본 또는 참조 계층)을 이용하여 현재 블록 (예컨대, EL에서의 현재 블록)을 예측하도록 구성된다. 이러한 예측은 계층간 예측으로서 지칭될 수도 있다. 계층간 예측 유닛 (75)은 계층간 리던던시를 감소시키기 위한 예측 방법들을 사용함으로써, 코딩 효율을 개선시키고 연산 자원 요건들을 감소시킨다. 계층간 예측의 일부의 예들은 계층간 인트라 예측, 계층간 모션 예측, 및 계층간 잔여 예측을 포함한다. 계층간 인트라 예측은 강화 계층에서 현재 블록을 예측하기 위해 기본 계층에서의 공동-위치된 블록들의 복원을 이용한다. 계층간 모션 예측은 강화 계층에서 모션을 예측하기 위해 기본 계층의 모션 정보를 이용한다. 계층간 잔여 예측은 강화 계층의 잔여를 예측하기 위해 기본 계층의 잔여를 이용한다. 기본 및 강화 계층들이 상이한 공간적 해상도들을 가질 때, 이하에서 더욱 상세하게 설명된 바와 같이, 공간적 모션 벡터 스케일링 및/또는 계층간 위치 맵핑은 시간적 스케일링 함수를 이용하여 계층간 예측 유닛 (75)에 의해 수행될 수도 있다.
- [0194] 역양자화 유닛 (76)은, 비트스트림에서 제공되며 엔트로피 디코딩 유닛 (70)에 의해 디코딩된 양자화된 변환

계수들을 역양자화 (inverse quantize), 예컨대, 양자화 해제 (de-quantize) 한다. 역양자화 프로세스는 적용되어야 할 양자화도 및 마찬가지로 역양자화도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 QPY의 이용을 포함할 수도 있다.

[0195] 역변환 유닛 (78)은 픽셀 도메인에서 잔여 블록들을 생성하기 위해, 역변환, 예컨대, 역 DCT, 역 DST, 역정수 변환, 또는 개념적으로 유사한 역변환 프로세스를 변환 계수들에 적용한다.

[0196] 모션 보상 유닛 (72)이 모션 벡터들 및 다른 선택스 엘리먼트들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30)는 역변환 유닛 (78)으로부터의 잔여 블록들을 모션 보상 유닛 (72)에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (90)는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 희망하는 경우, 디블록킹 필터는 또한, 블록화 아티팩트들을 제거하기 위해 디코딩된 블록들을 필터링하도록 적용될 수도 있다. (코딩 루프 내 또는 코딩 루프 이후 중 어느 하나에서의) 다른 루프 필터들은 또한, 픽셀 천이 (pixel transition)들을 평탄화하거나, 또는 이와 다르게 비디오 품질을 개선시키기 위해 이용될 수도 있다. 다음으로, 주어진 프레임 또는 픽처에서의 디코딩된 비디오 블록들은, 후속 모션 보상을 위해 이용된 참조 픽처들을 저장하는 참조 픽처 메모리 (92) 내에 저장된다. 참조 프레임 메모리 (82)는 또한, 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에서의 더 이후의 제시를 위한 디코딩된 비디오를 저장한다.

멀티-계층 디코더

[0198] 도 3b는 이 개시물에서 설명된 양태들에 따라 기법들을 구현할 수도 있는 멀티-계층 비디오 디코더 (31)의 예를 예시하는 블록도이다. 비디오 디코더 (31)는 SHVC 및 멀티뷰 코딩을 위한 것과 같은 멀티-계층 비디오 프레임들을 프로세싱하도록 구성될 수도 있다. 또한, 비디오 디코더 (31)는 이 개시물의 기법들 중의 임의의 것 또는 전부를 수행하도록 구성될 수도 있다.

[0199] 비디오 디코더 (31)는 비디오 디코더 (30A) 및 비디오 디코더 (30B)를 포함하고, 그 각각은 도 3a의 비디오 디코더 (30)로서 구성될 수도 있으며 비디오 디코더 (30)에 대하여 위에서 설명된 기능들을 수행할 수도 있다. 또한, 참조 번호들의 재이용에 의해 표시된 바와 같이, 비디오 디코더들 (30A 및 30B)은 비디오 디코더 (30)로서 시스템들 및 서브시스템들의 적어도 일부를 포함할 수도 있다. 비디오 디코더 (31)는 2개의 비디오 디코더들 (30A 및 30B)을 포함하는 것으로서 예시되어 있지만, 비디오 디코더 (31)는 이와 같이 제한되지 않으며 임의의 수의 비디오 디코더 (30) 계층들을 포함할 수도 있다. 일부의 실시형태들에서, 비디오 디코더 (31)는 액세스 유닛에서의 각각의 픽처 또는 프레임에 대한 비디오 디코더 (30)를 포함할 수도 있다. 예를 들어, 5개의 픽처들을 포함하는 액세스 유닛은 5개의 디코더 계층들을 포함하는 비디오 디코더에 의해 프로세싱되거나 디코딩될 수도 있다. 일부의 실시형태들에서, 비디오 디코더 (31)는 액세스 유닛에서의 프레임들보다 더 많은 디코더 계층들을 포함할 수도 있다. 일부의 이러한 경우들에 있어서, 비디오 인코더 계층들의 일부는 일부의 액세스 유닛들을 프로세싱할 때에 비활성일 수도 있다.

[0200] 비디오 디코더들 (30A 및 30B)에 추가하여, 비디오 디코더 (31)는 업샘플링 유닛 (92)을 포함할 수도 있다. 일부의 실시형태들에서, 업샘플링 유닛 (92)은 프레임 또는 액세스 유닛에 대한 참조 픽처 리스트에 추가되어야 할 강화된 계층을 생성하기 위해 수신된 비디오 프레임의 기본 계층을 업샘플링할 수도 있다. 이 강화된 계층은 참조 프레임 메모리 (82) 내에 (예컨대, 그 디코딩된 픽처 버퍼 등 내에) 저장될 수 있다. 일부의 실시형태들에서, 업샘플링 유닛 (92)은 도 2a의 리샘플링 유닛 (90)에 대하여 설명된 실시형태들의 일부 또는 전부를 포함할 수 있다. 일부의 실시형태들에서, 업샘플링 유닛 (92)은 계층을 업샘플링하며, 슬라이스 경계 규칙들 및/또는 래스터 스캔 규칙들의 세트를 준수하기 위해 하나 이상의 슬라이스들을 인식하거나, 재정의하거나, 수정하거나, 조절하도록 구성된다. 일부의 경우에는, 업샘플링 유닛 (92)은 수신된 비디오 프레임의 계층을 업샘플링 및/또는 다운샘플링하도록 구성된 리샘플링 유닛일 수도 있다.

[0201] 업샘플링 유닛 (92)은 더 낮은 계층의 디코더 (예컨대, 비디오 디코더 (30A))의 디코딩된 픽처 버퍼 (82)로부터 픽처 또는 프레임 (또는 픽처와 연관된 픽처 정보)을 수신하고 픽처 (또는 수신된 픽처 정보)를 업샘플링하도록 구성될 수도 있다. 다음으로, 이 업샘플링된 픽처는 더 낮은 계층의 디코더와 동일한 액세스 유닛에서의 픽처를 디코딩하도록 구성된 더 높은 계층의 디코더 (예컨대, 비디오 인코더 (30B))의 모드 선택 유닛 (71)에 제공될 수도 있다. 일부의 경우들에 있어서, 더 높은 계층의 디코더는 더 낮은 계층의 디코더로부터 제거된 하나의 계층이다. 다른 경우에는, 도 3b의 계층 0 디코더 및 계층 1 디코더 사이에 하나 이상

의 더 높은 계층의 디코더들이 있을 수도 있다.

[0202] 일부의 경우에는, 업샘플링 유닛 (92) 이 생략되거나 바이패스될 수도 있다. 이러한 경우에는, 비디오 디코더 (30A) 의 디코딩된 픽처 버퍼 (82)로부터의 픽처가 직접적으로, 또는 적어도 업샘플링 유닛 (92) 에 제공되지 않으면서, 비디오 디코더 (30B) 의 모드 선택 유닛 (71) 에 제공될 수도 있다. 예를 들어, 비디오 디코더 (30B) 에 제공된 비디오 데이터와, 비디오 디코더 (30A) 의 디코딩된 픽처 버퍼 (82)로부터의 참조 픽처가 동일한 사이즈 또는 해상도일 경우, 참조 픽처는 업샘플링 없이 비디오 디코더 (30B) 에 제공될 수도 있다. 또한, 일부의 실시형태들에서, 업샘플링 유닛 (92) 은 비디오 디코더 (30A) 의 디코딩된 픽처 버퍼 (82)로부터 수신된 참조 픽처를 업샘플링하거나 다운샘플링하도록 구성된 리샘플링 유닛 (90) 일 수도 있다.

[0203] 도 3b 에서 예시된 바와 같이, 비디오 디코더 (31) 는 디멀티플렉서 (99) 또는 demux 를 더 포함할 수도 있다. demux (99) 는 인코딩된 비디오 비트스트림을 다수의 비트스트림들로 분할할 수 있고, demux (99) 에 의해 출력된 각각의 비트스트림은 상이한 비디오 디코더 (30A 및 30B) 에 제공된다. 다수의 비트스트림들은 비트스트림을 수신함으로써 생성될 수도 있고, 비디오 디코더들 (30A 및 30B) 의 각각은 주어진 시간에 비트스트림의 부분을 수신한다. 일부의 경우에는, demux (99) 에서 수신된 비트스트림으로부터의 비트들이 각각의 비디오 디코더들 (예컨대, 도 3b 의 예에서의 비디오 디코더들 (30A 및 30B)) 사이에서 한 번에 하나의 비트씩 교번될 수도 있지만, 많은 경우에는, 비트스트림이 상이하게 분할된다. 예를 들어, 비트스트림은 어느 비디오 디코더가 비트스트림을 한 번에 하나의 블록씩 수신하는지를 교번함으로써 분할될 수도 있다. 또 다른 예에서, 비트스트림은 비-1:1 (non-1:1) 비율의 블록들에 의해 비디오 디코더들 (30A 및 30B) 의 각각으로 분할될 수도 있다. 예를 들어, 2 개의 블록들은 비디오 디코더 (30A) 에 제공된 각각의 블록에 대하여 비디오 디코더 (30B) 에 제공될 수도 있다. 일부의 실시형태들에서, demux (99) 에 의한 비트스트림의 분할은 미리 프로그래밍될 수도 있다. 다른 실시형태들에서, demux (99) 는 목적지 디바이스 (14) 상의 프로세서로부터와 같이, 비디오 디코더 (31) 의 외부의 시스템으로부터 수신된 제어 신호에 기초하여 비트스트림을 분할할 수도 있다. 제어 신호는 입력 인터페이스 (28)로부터의 비디오의 해상도 또는 비트레이트에 기초하여, 채널 (16) 의 대역폭에 기초하여, 사용자와 연관된 가입 (예컨대, 유료 가입 대 무료 가입) 에 기초하여, 또는 비디오 디코더 (31) 에 의해 얻어질 수 있는 해상도를 결정하기 위한 임의의 다른 인자에 기초하여 생성될 수도 있다.

계층간 예측 시그널링 및 관련된 프로세스들

[0204] 도 4 는 도 2 의 비디오 인코더 (20) 또는 도 3 의 비디오 디코더 (30) 에 의해 수행될 수도 있는, 계층간 예측 시그널링 및 관련된 프로세스들을 수행하는 방법의 하나의 실시형태를 예시한다. 방법 (400) 은 도 2 의 비디오 인코더 (20) 의 모션 추정 유닛 (42), 모션 보상 유닛 (44), 인트라 예측 유닛 (46), 및 계층간 예측 유닛 (66) 중의 임의의 하나 이상에 의해 수행될 수도 있다. 또 다른 실시형태에서, 방법 (400) 은 도 3 의 디코더의 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 및 계층간 예측 유닛 (75) 중의 임의의 하나 이상에 의해 수행될 수도 있다.

[0205] [0206] 방법 (400) 은 블록 (410) 에서 시작된다. 블록 (420) 에서는, 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 이용할 픽처들의 수의 표시가 제공된다. 표시는 위에서 더욱 상세하게 논의된 num_inter_layer_ref_pics 신택스에 대응할 수 있다. 표시는 계층간 예측을 위해 현재 픽처에 의해 이용된 픽처들의 수를 특정한다. 하나의 실시형태에서는, 존재하지 않을 때, 표시의 값이 0 과 동일한 것으로 추론된다. 표시의 값은 0 내지 NumDirectRefLayers[LayerIdInVps [nuh_layer_id]] 까지의 범위에 있다. 블록 (430) 에서는, 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 어느 특별한 계층간 참조 픽처들이 이용되어야 하는지의 표시가 제공된다. 표시는 위에서 더욱 상세하게 논의된 ref_layer_idx_delta[i] 신택스에 대응할 수 있다. 또 다른 실시형태에서, 표시는 위에서 논의된 바와 같이, RefPicLayerId[i] 에 대응할 수 있다. 블록 (440) 에서는, 현재 픽처에 대한 계층간 참조 픽처 세트가 결정된다. 계층간 참조 픽처 세트는 계층간 참조 픽처들의 수와, 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 어느 계층간 참조 픽처들을 이용할 것인지의 표시를 이용하여 결정된다. 방법 (400) 은 블록 (450) 에서 종료된다.

[0207] 도 5 는 도 2 의 비디오 인코더 (20) 또는 도 3 의 비디오 디코더 (30) 에 의해 수행될 수도 있는, 계층간 예측 시그널링 및 관련된 프로세스들을 수행하는 방법의 또 다른 실시형태를 예시한다. 방법 (500) 은 도 2 의 비디오 인코더 (20) 의 모션 추정 유닛 (42), 모션 보상 유닛 (44), 인트라 예측 유닛 (46), 및 계층간 예측 유닛 (66) 중의 임의의 하나 이상에 의해 수행될 수도 있다. 또 다른 실시형태에서, 방법 (500) 은 도 3 의

디코더의 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 및 계층간 예측 유닛 (75) 중의 임의의 하나 이상에 의해 수행될 수도 있다.

[0208] 방법 (500)은 블록 (510)에서 시작된다. 블록 (520)에서는, 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 이용할 픽처들의 수의 표시가 제공된다. 표시는 위에서 더욱 상세하게 논의된 num_inter_layer_ref_pics 선택에 대응할 수 있다. 표시는 계층간 예측을 위해 현재 픽처에 의해 이용된 픽처들의 수를 특정한다. 하나의 실시형태에서는, 존재하지 않을 때, 표시의 값이 0과 동일한 것으로 추론된다. 표시의 값은 0 내지 NumDirectRefLayers[LayerIdInVps[nuh_layer_id]] 까지의 범위에 있다. 블록 (530)에서는, 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 어느 특별한 계층간 참조 픽처들이 이용되어야 하는지의 표시가 제공된다. 표시는 위에서 더욱 상세하게 논의된 ref_layer_idx_delta[i] 선택에 대응할 수 있다. 또 다른 실시형태에서, 표시는 위에서 논의된 바와 같이, RefPicLayerId[i]에 대응할 수 있다. 블록 (540)에서는, 디코딩된 픽처 버퍼 (예컨대, 도 2 및 도 3의 참조 프레임 메모리의 디코딩된 픽처 버퍼)에 있지 않은 현재 픽처에 대한 계층간 참조 픽처 세트에서의 픽처가 디코딩된 픽처 버퍼로의 송신 동안에 손실되었거나 제공되지 않았는지 여부가 결정된다. 방법 (500)은 블록 (550)에서 종료된다.

[0209] 도 6은 도 2의 비디오 인코더 (20) 또는 도 3의 비디오 디코더 (30)에 의해 수행될 수도 있는, 계층간 예측 시그널링 및 관련된 프로세스들을 수행하는 방법의 또 다른 실시형태를 예시한다. 방법 (600)은 도 2의 비디오 인코더 (20)의 모션 추정 유닛 (42), 모션 보상 유닛 (44), 인트라 예측 유닛 (46), 및 계층간 예측 유닛 (66) 중의 임의의 하나 이상에 의해 수행될 수도 있다. 또 다른 실시형태에서, 방법 (600)은 도 3의 디코더의 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 및 계층간 예측 유닛 (75) 중의 임의의 하나 이상에 의해 수행될 수도 있다.

[0210] 방법 (600)은 블록 (610)에서 시작된다. 블록 (620)에서는, 코딩 (예컨대, 예측) 하기 위한 각각의 픽처가 각각의 픽처를 코딩하기 위해 이용되어야 할 최대로 하나의 계층간 참조 픽처를 지칭하는지 여부를 표시하는 표시가 제공된다. 표시는 위에서 더욱 상세하게 논의된 one_ilp_ref_pic_only_flag에 대응할 수 있다. 블록 (630)에서는, 표시가 선택적으로 제공된다. 선택적인 표시는 계층간 예측을 이용하여 현재 픽처를 예측하기 위해 어느 계층간 참조 픽처들이 이용되어야 하는지의 표시가 슬라이스 헤더 내에 존재하는지 여부에 대응한다. 선택적인 표시는 위에서 더욱 상세하게 논의된 ilp_ref_pic_present_in_slice_flag에 대응할 수 있다. 방법 (600)은 블록 (640)에서 종료된다.

용어

[0211] [0212] 상기 개시물은 특별한 실시형태들을 설명하였지만, 많은 변형들이 가능하다. 예를 들어, 위에서 언급된 바와 같이, 상기 기법들은 3D 비디오 인코딩에 적용될 수도 있다. 3D 비디오의 일부의 실시형태들에서는, 참조 계층 및 강화 계층이 비디오의 제 2 뷰를 디스플레이하기 위해 충분한 비디오 정보를 함께 포함하도록, 참조 계층 (예컨대, 기본 계층)은 비디오의 제 1 뷰를 디스플레이하기 위해 충분한 비디오 정보를 포함하고 강화 계층은 참조 계층과 관련된 추가적인 비디오 정보를 포함한다. 이 2 개의 뷰들은 입체 이미지를 생성하기 위해 이용될 수 있다. 위에서 논의된 바와 같이, 참조 계층으로부터의 모션 정보는 개시물의 양태들에 따라, 강화 계층에서의 비디오 유닛을 인코딩하거나 디코딩할 때에 추가적인 묵시적 가정을 식별하기 위해 이용될 수 있다. 이것은 3D 비디오 비트스트림에 대해 더 큰 코딩 효율을 제공할 수 있다.

[0213] 예에 따라서는, 본원에서 설명된 기법들 중의 임의의 것의 어떤 액트 (act) 들 또는 이벤트 (event) 들이 상이한 시퀀스에서 수행될 수 있거나, 추가될 수도 있거나, 병합될 수도 있거나, 또는 모두 배제 (예를 들어, 모든 설명된 액트들 또는 이벤트들이 기법들의 실시를 위해 필요한 것은 아님) 될 수도 있다는 것을 인식해야 한다. 또한, 어떤 예들에서는, 액트들 또는 이벤트들이 순차적인 것이 아니라, 예컨대, 멀티-스레딩된 (multi-threaded) 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해 동시에 수행될 수도 있다.

[0214] 본원에서 개시된 정보 및 신호들은 다양한 상이한 기술들 및 기법들 중의 임의의 것을 이용하여 표현될 수도 있다. 예를 들어, 상기 설명의 전반에 걸쳐 참조될 수도 있는 데이터, 명령들, 커맨드들, 정보, 신호들, 비트들, 심볼들, 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 입자들, 광학 필드들 또는 입자들, 또는 그 임의의 조합에 의해 표현될 수도 있다.

[0215] 본원에서 개시된 실시형태들과 관련하여 설명된 다양한 예시적인 논리적 블록들, 모듈들, 회로들, 및 알고리즘 단계들은 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양자의 조합들로서 구현될 수도 있다. 하드웨어 및 소프

트웨어의 이 교환가능성을 명확하게 예시하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 단계들은 일반적으로 그 기능성의 측면에서 위에서 설명되었다. 이러한 기능성이 하드웨어 또는 소프트웨어로 구현되는지 여부는 특별한 애플리케이션 및 전체 시스템에 부과된 설계 제약들에 종속된다. 당업자들은 각각의 특별한 애플리케이션을 위한 다양한 방법들로 설명된 기능성을 구현할 수도 있지만, 이러한 구현 판정들은 본 발명의 범위로부터의 이탈을 야기시키는 것으로 해석되지 않아야 한다.

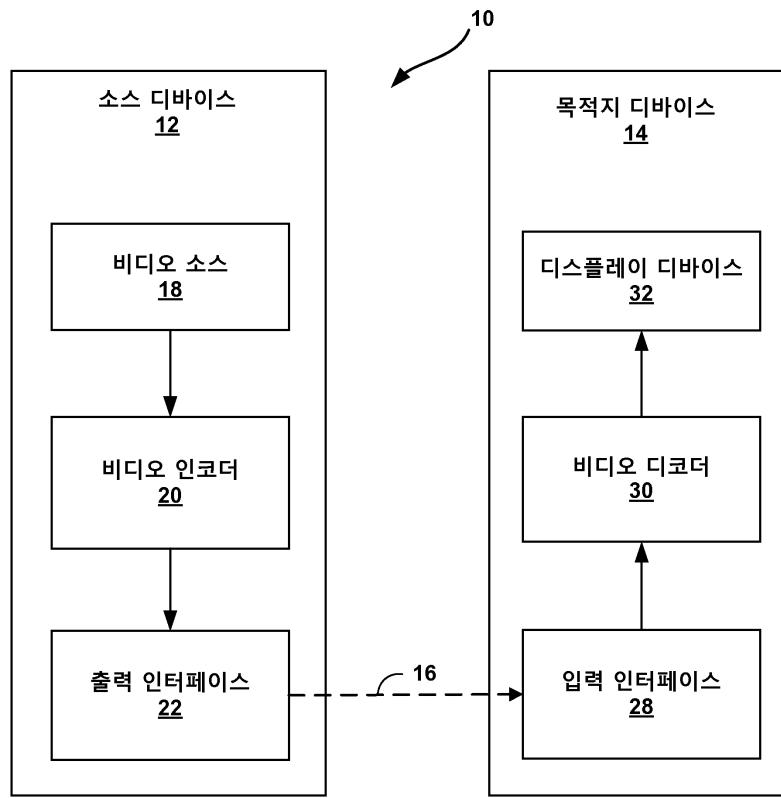
[0216] 본원에서 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 이러한 기법들은 범용 컴퓨터들, 무선 통신 디바이스 핸드셋들, 또는 무선 통신 디바이스 핸드셋들 및 다른 디바이스들에서의 애플리케이션을 포함하는 다수의 용도들을 갖는 집적 회로 디바이스들과 같은 다양한 디바이스들 중의 임의의 것에서 구현될 수도 있다. 모듈들 또는 컴포넌트들로서 설명된 임의의 특징들은 집적된 로직 디바이스에서 함께, 또는 개별적이지만 상호 동작가능한 로직 디바이스들로서 별도로 구현될 수도 있다. 소프트웨어로 구현될 경우, 기법들은, 실행될 때, 위에서 설명된 방법들 중 하나 이상을 수행하는 명령들을 포함하는 프로그램 코드를 포함하는 컴퓨터-판독가능한 데이터 저장 매체에 의해 적어도 부분적으로 실현될 수도 있다. 컴퓨터-판독가능한 데이터 저장 매체는 패키징 재료들을 포함할 수도 있는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있다. 컴퓨터-판독가능한 매체는 동기식 랜덤 액세스 메모리 (synchronous dynamic random access memory; SDRAM) 와 같은 랜덤 액세스 메모리 (random access memory; RAM), 판독-전용 메모리 (read-only memory; ROM), 비-휘발성 랜덤 액세스 메모리 (non-volatile random access memory; NVRAM), 전기적 소거가능 프로그래밍가능 판독-전용 메모리 (electrically erasable programmable read-only memory; EEPROM), 플래시 메모리 (FLASH memory), 자기 또는 광학 데이터 저장 매체들 등과 같은 메모리 또는 데이터 저장 매체들을 포함할 수도 있다. 추가적으로 또는 대안적으로, 기법들은 전파된 신호들 또는 파동 (wave) 들과 같이, 명령들 또는 데이터 구조들의 형태로 프로그램 코드를 반송하거나 통신하며 컴퓨터에 의해 액세스, 판독, 및/또는 실행될 수 있는 컴퓨터-판독가능한 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

[0217] 프로그램 코드는, 하나 이상의 디지털 신호 프로세서 (digital signal processor; DSP) 들, 범용 마이크로프로세서들, 애플리케이션 특정 집적 회로 (application specific integrated circuit; ASIC) 들, 필드 프로그래밍 가능한 로직 어레이 (field programmable logic array; FPGA) 들, 또는 다른 등가의 집적 또는 개별 로직 회로 부와 같은 하나 이상의 프로세서들을 포함할 수도 있는 프로세서에 의해 실행될 수도 있다. 이러한 프로세서는 이 개시물에서 설명된 기법들 중의 임의의 것을 수행하도록 구성될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안적으로, 프로세서는 임의의 기존의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한, 컴퓨팅 디바이스들의 조합, 예컨대, DSP 및 마이크로프로세서, 복수의 마이크로프로세서들, DSP 코어와 함께 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성의 조합으로서 구현될 수도 있다. 따라서, 본원에서 이용된 바와 같은 용어 "프로세서" 는 상기한 구조, 상기 구조의 임의의 조합, 또는 본원에서 설명된 기법들의 구현을 위해 적당한 임의의 다른 구조 또는 장치 중의 임의의 것을 지칭할 수도 있다. 게다가, 일부의 양태들에서, 본원에서 설명된 기능성은, 인코딩 및 디코딩을 위해 구성되거나, 조합된 비디오 인코더-디코더 (combined video encoder-decoder; CODEC) 내에 편입된 전용 소프트웨어 모듈들 또는 하드웨어 모듈들 내에서 제공될 수도 있다.

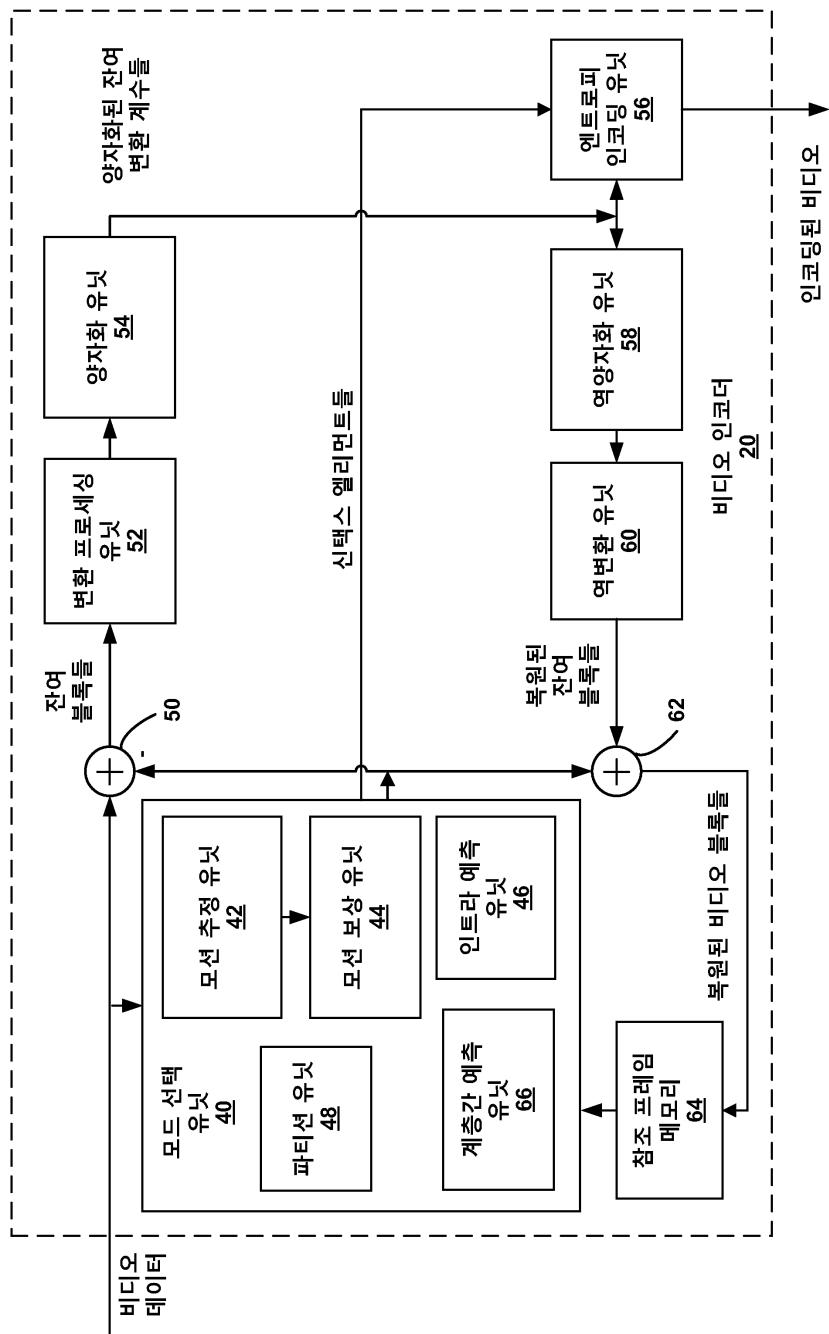
[0218] 발명의 다양한 실시형태들이 설명되었다. 이러한 그리고 다른 실시형태들은 다음의 청구항들의 범위 내에 있다.

도면

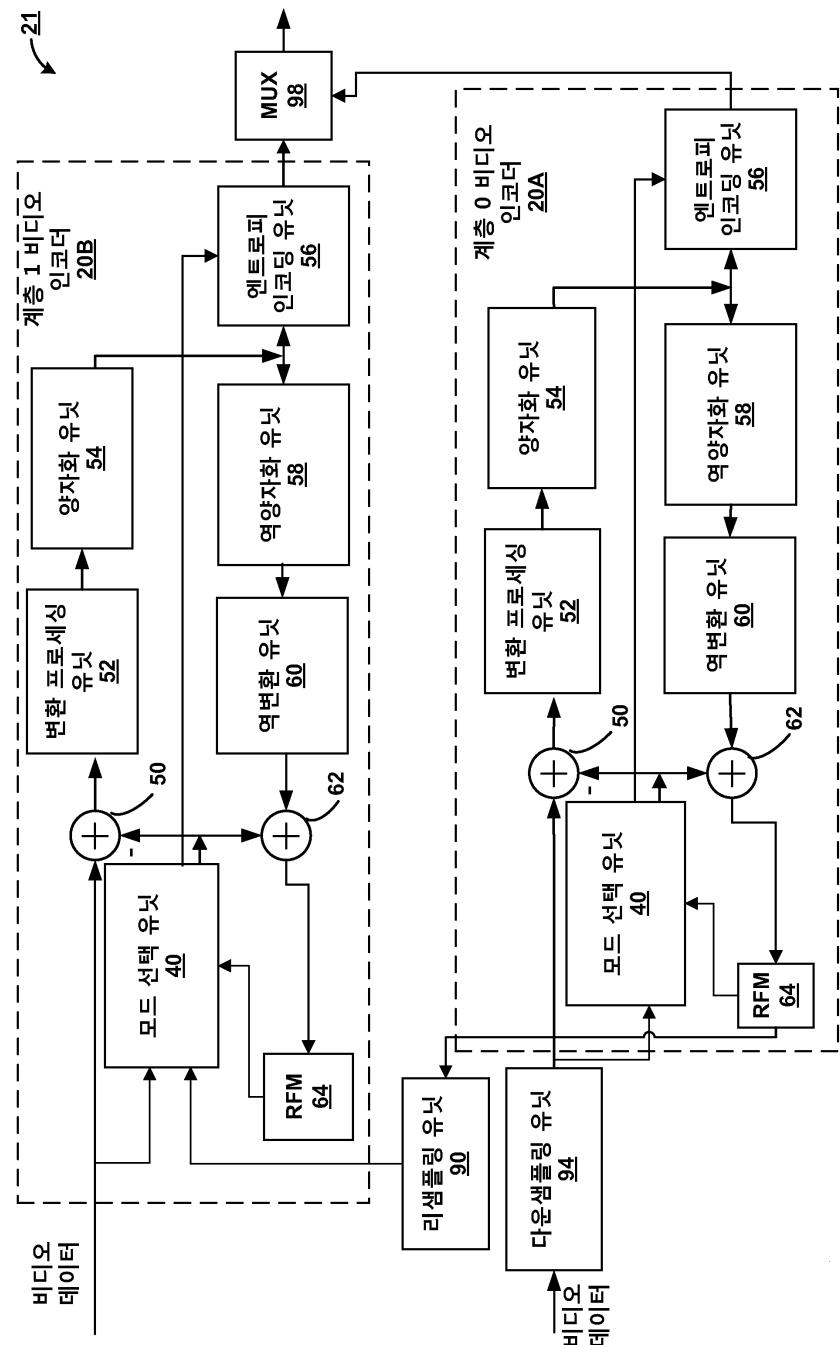
도면1



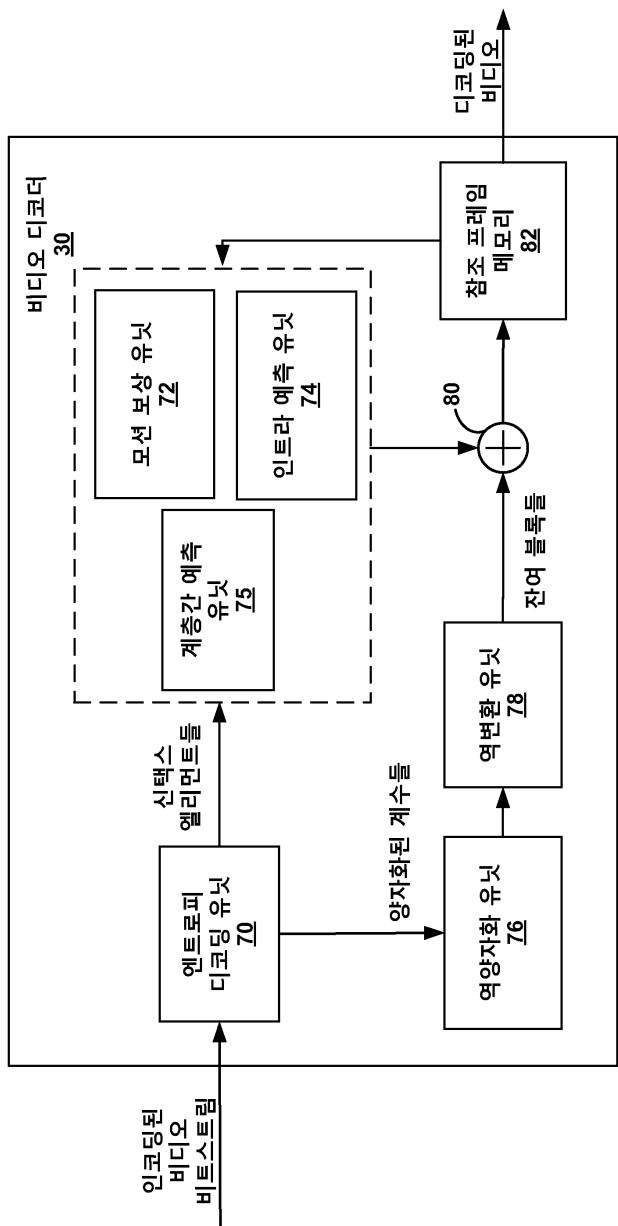
도면2a



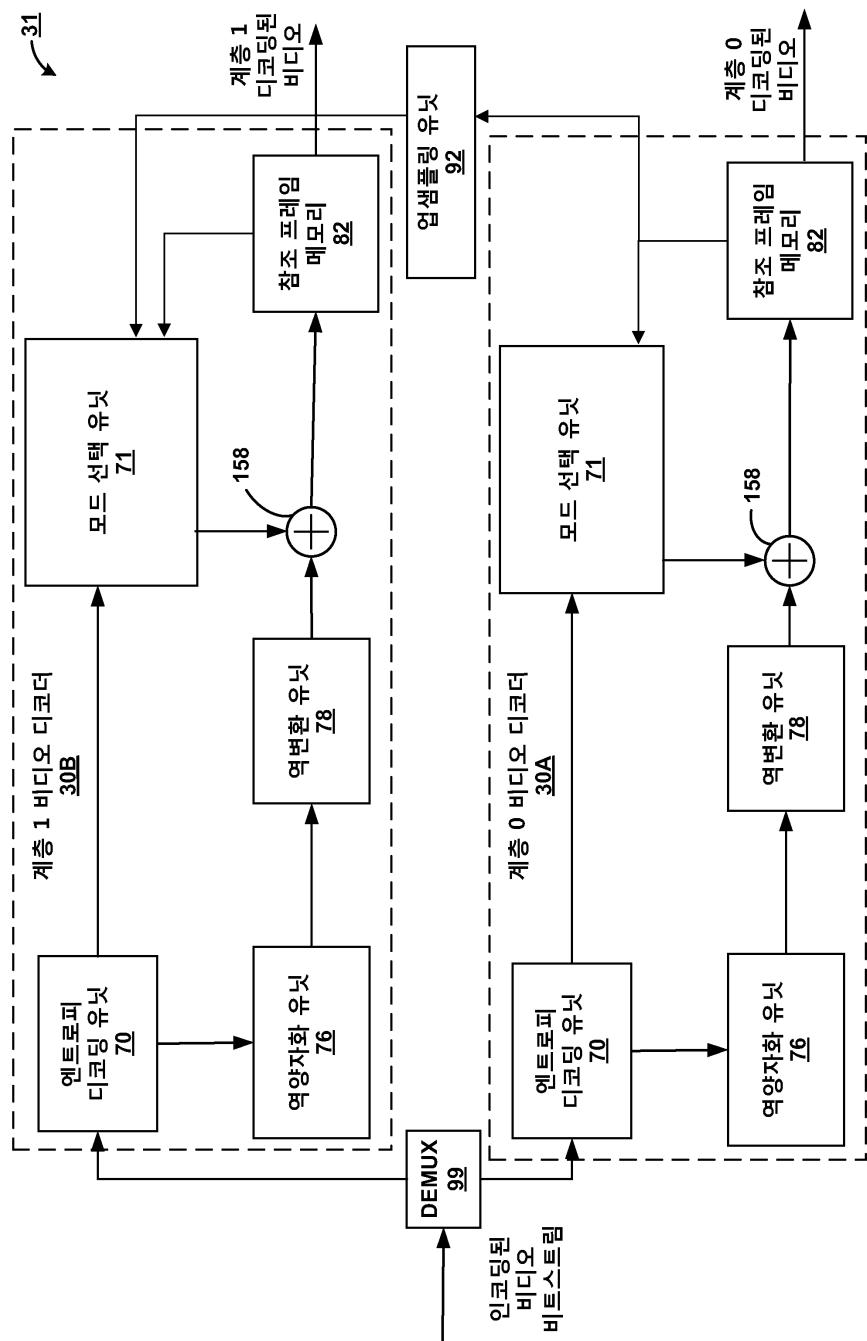
도면2b



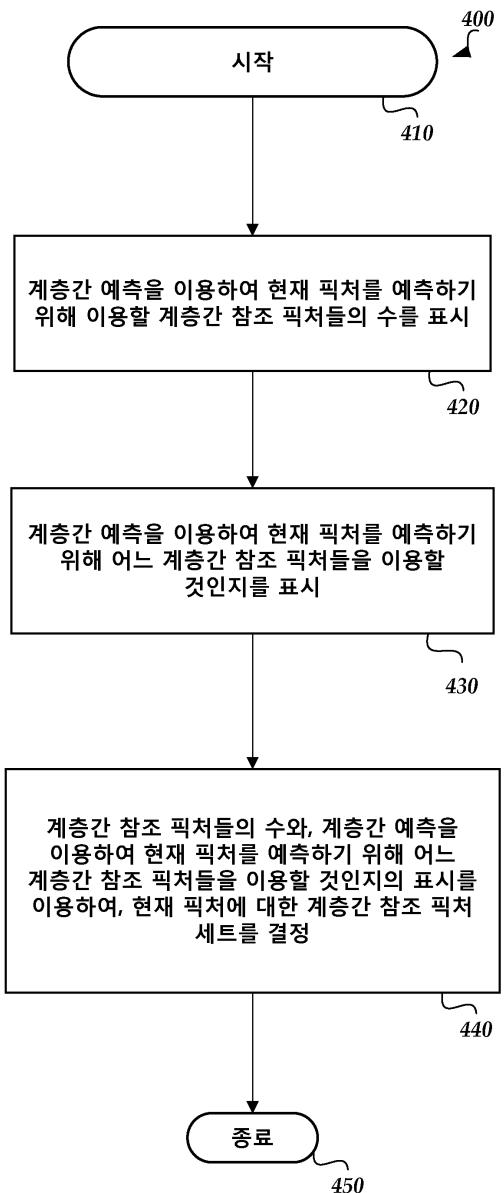
도면3a



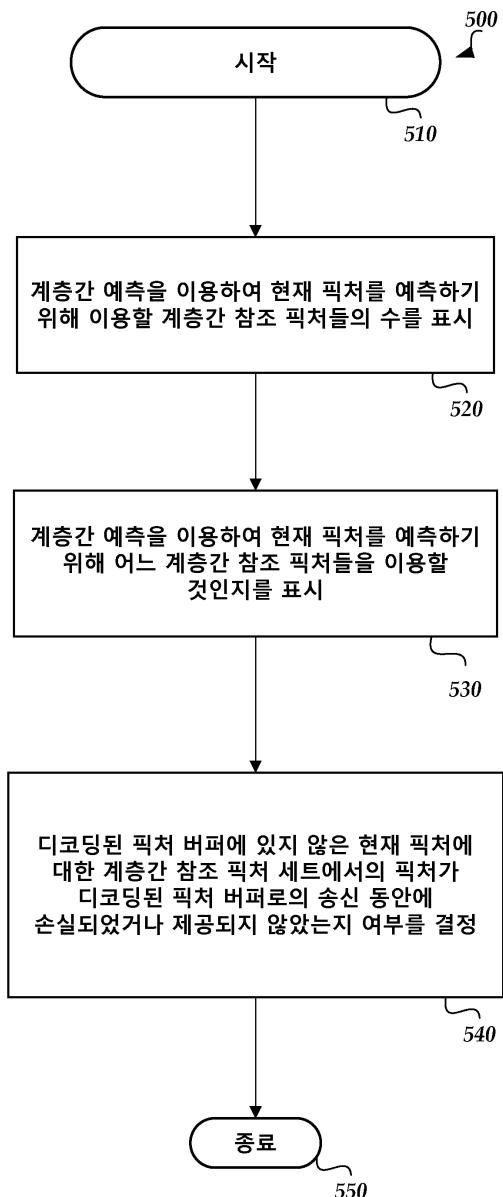
도면3b



도면4



도면5



도면6

