



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2017년10월18일

(11) 등록번호 10-1787864

(24) 등록일자 2017년10월12일

(51) 국제특허분류(Int. Cl.)

H04N 19/30 (2014.01) H04N 19/29 (2014.01)

H04N 19/44 (2014.01) H04N 19/46 (2014.01)

H04N 19/513 (2014.01) H04N 19/597 (2014.01)

H04N 19/70 (2014.01) H04N 19/91 (2014.01)

(52) CPC특허분류

H04N 19/30 (2015.01)

H04N 19/29 (2015.01)

(21) 출원번호 10-2015-7031868

(22) 출원일자(국제) 2014년04월08일

심사청구일자 2017년02월23일

(85) 번역문제출일자 2015년11월05일

(65) 공개번호 10-2015-0139945

(43) 공개일자 2015년12월14일

(86) 국제출원번호 PCT/US2014/033367

(87) 국제공개번호 WO 2014/168965

국제공개일자 2014년10월16일

(30) 우선권주장

61/809,858 2013년04월08일 미국(US)

14/247,110 2014년04월07일 미국(US)

(56) 선행기술조사문헌

Y-K. Wang, et al. AHG9: On VPS and SPS in HEVC 3DV and scalable extensions. JCT-VC of ITU-T and ISO/IEC. JCTVC-M0268 Ver.1, Apr. 9, 2013, pp.1-22

Y. Chen, et al. AHG7: Target output views for MV-HEVC. JCT on 3DV of ITU-T and ISO/IEC. JCT3V-C0059 Ver.1, Jan. 10, 2013, pp.1-2

J. Boyce, et al. High level syntax hooks for future extensions. JCT-VC of ITU-T and ISO/IEC. JCTVC-H0388 Ver.4, Feb. 9, 2012, pp.1-6

B. Bross, et al. High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call). JCT-VC of ITU-T and ISO/IEC. JCTVC-L1003 Ver.34, Mar. 2013, pp.1-298

전체 청구항 수 : 총 26 항

심사관 : 이상래

(54) 발명의 명칭 출력 계층 세트에 대한 비디오 데이터 코딩

(57) 요 약

다층 비디오 데이터의 인코딩, 디코딩, 송신, 또는 수신을 포함할 수도 있는, 다층 비디오 데이터를 코딩하는 시스템들, 방법들, 및 디바이스들이 개시된다. 그 시스템들, 방법들, 및 디바이스들은 계층 세트에 대한 제 1 출력 계층 세트를 수신 또는 송신하고 그 계층 세트에 대한 제 2 출력 계층 세트를 수신 또는 송신한다. 그 시스템들, 방법들, 및 디바이스들은 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나의 출력 계층 세트에 대한 비디오 데이터를 코딩 (인코딩 또는 디코딩) 할 수도 있다.

대 표 도



(52) CPC특허분류

HO4N 19/44 (2015.01)

HO4N 19/46 (2015.01)

HO4N 19/513 (2015.01)

HO4N 19/597 (2015.01)

HO4N 19/70 (2015.01)

HO4N 19/91 (2015.01)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

비디오 디코더에 의해, 인코딩된 비디오 비트스트림으로부터, 다층 비디오 데이터를 수신하는 단계로서, 상기 다층 비디오 데이터는,

상기 다층 비디오 데이터의 계층 세트에 포함된 제 1 출력 계층 세트로서, 상기 제 1 출력 계층 세트가 타겟 출력 계층 세트들의 제 1 리스트와 연관되도록, 상기 제 1 출력 계층 세트는 상기 다층 비디오 데이터의 상기 계층 세트의 서브세트를 나타내는, 상기 제 1 출력 계층 세트;

상기 다층 비디오 데이터의 상기 계층 세트에 포함된 제 2 출력 계층 세트로서, 상기 제 2 출력 계층 세트가 타겟 출력 계층 세트들의 제 2 리스트와 연관되도록, 상기 제 2 출력 계층 세트는 상기 다층 비디오 데이터의 상기 계층 세트의 서브세트를 나타내는, 상기 제 2 출력 계층 세트;

상기 다층 비디오 데이터에 대한 시퀀스 파라미터 세트 (SPS);

상기 다층 비디오 데이터의 제 1 계층으로서, 상기 제 1 계층은 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 1 계층;

상기 다층 비디오 데이터의 제 2 계층으로서, 상기 제 2 계층은, 상기 제 1 계층과는 상이한 값의 공간적 해상도, 비트 깊이, 또는 컬러 포맷 중 적어도 하나를 가지고 그리고 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 2 계층

을 포함하는, 상기 다층 비디오 데이터를 수신하는 단계;

상기 비디오 디코더에 의해, 상기 타겟 출력 계층 세트들의 제 1 리스트 또는 상기 타겟 출력 계층 세트들의 제 2 리스트 중 적어도 하나를 이용하여 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해, 인코딩된 비디오 데이터를 디코딩하는 단계; 및

상기 비디오 디코더에 의해, 상기 SPS 에 포함된 정보를 이용하여 상기 제 1 계층 및 상기 제 2 계층에 대해, 인코딩된 비디오 데이터를 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 비디오 디코더에 의해, 상기 비디오 데이터에 대한 제 1 신택스 엘리먼트를 상기 비디오 데이터에 대한 제 2 신택스 엘리먼트로 변경하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 제 1 신택스 엘리먼트는 output_layer_flag[lsIdx][j] 를 포함하고 상기 제 2 신택스 엘리먼트는 output_layer_flag[i][j] 를 포함하고,

상기 i 는 출력 계층 세트의 인덱스를 나타내고, 상기 j 와 상기 lsIdx 는 계층 세트의 인덱스를 나타내는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 SPS 는 비디오 유용성 정보 (Video Usability Information; VUI) 를 포함하며, 그리고

타이밍 정보는 0 보다 더 큰 계층 ID 를 갖는 SPS 의 VUI 내에 포함되지 않는, 비디오 데이터를 디코딩하는 방

법.

청구항 5

제 1 항에 있어서,

상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트 중 적어도 하나와 연관된 타겟 출력 계층 세트는 디폴트 출력 계층 세트이고, 그리고

상기 디폴트 출력 계층 세트에 대한 상기 타겟 출력 계층은 명시적으로 시그널링되지 않는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 SPS 에 대한 계층 ID 는 제로 (0) 보다 더 크고, 그리고

sps_max_sub_layers_minus1 및 sps_temporal_id_nesting_flag 는 상기 SPS 에 포함되지 않는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 비디오 디코더에 의해, 0 과 동일한 output_layer_set_idx[i] 를 위해 사용된 상기 비디오 데이터에 대한 신택스 엘리먼트 layer_id_included_flag[][] 는 계층 세트 0 을 위한 것임을 결정하는 단계; 및

상기 비디오 디코더에 의해, 상기 결정에 기초하여 output_layer_set_idx[i] 를 output_layer_set_idx_minus1[i] 신택스 엘리먼트로 변경하는 단계를 더 포함하고,

상기 i 는 출력 계층 세트의 인덱스를 나타내는, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 비디오 디코더에 의해, 상기 인코딩된 비디오 비트스트림으로부터, 비디오 파라미터 세트 (video parameter set; VPS) 확장물에서의 프로파일, 티어, 및 레벨 신택스 구조들의 비-엔트로피 인코딩된 세트를 수신하는 단계; 및

상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트를 포함하는 복수의 출력 계층 세트들의 각각에 대한 상기 프로파일, 티어, 및 레벨 신택스 구조들 중 하나를 참조하는 단계를 더 포함하고,

상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해 상기 인코딩된 비디오 데이터를 디코딩하는 단계는, 상기 비디오 디코더에 의해, 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트의 각각의 출력 계층 세트에 대해 참조되는 상기 프로파일, 티어, 또는 레벨 신택스 구조 중 하나 이상으로부터의 정보에 기초하여 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해 상기 인코딩된 비디오 데이터를 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 9

다중 비디오 데이터를 인코딩하는 방법으로서,

비디오 인코더에 의해, 제 1 출력 계층 세트와 연관된 타겟 출력 계층 세트들의 제 1 리스트 또는 제 2 출력 계층 세트와 연관된 타겟 출력 계층 세트들의 제 2 리스트 중 적어도 하나를 이용하여 상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트 중 적어도 하나에 대한 비디오 데이터를 인코딩하는 단계로서, 상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트 양자는 상기 다중 비디오 데이터의 단일 계층 세트에 포함되는, 상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트 중 적어도 하나에 대한 비디오 데이터를 인코딩하는 단계;

상기 비디오 인코더에 의해, 상기 다중 비디오 데이터와 연관된 시퀀스 파라미터 세트 (SPS) 에서 상기 다중 비디오 데이터의 제 1 계층 및 상기 다중 비디오 데이터의 제 2 계층에 대한 비디오 데이터를 인코딩하는 단계;

및

상기 비디오 인코더에 의해,

상기 제 1 출력 계층 세트;

상기 제 2 출력 계층 세트;

상기 다층 비디오 데이터와 연관된 상기 SPS;

상기 다층 비디오 데이터의 제 1 계층으로서, 상기 제 1 계층은 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 1 계층; 및

상기 다층 비디오 데이터의 제 2 계층으로서, 상기 제 2 계층은, 상기 제 1 계층과는 상이한 값의 공간적 해상도, 비트 깊이, 또는 컬러 포맷 중 적어도 하나를 가지고 그리고 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 2 계층

을 인코딩된 비디오 비트스트림으로 인코딩하는 단계를 포함하는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 10

제 9 항에 있어서,

상기 비디오 인코더에 의해, 상기 비디오 데이터에 대한 제 1 신택스 엘리먼트를 상기 비디오 데이터에 대한 제 2 신택스 엘리먼트로 변경하는 단계를 더 포함하는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 11

제 10 항에 있어서,

상기 제 1 신택스 엘리먼트는 output_layer_flag[lsIdx][j] 를 포함하고 상기 제 2 신택스 엘리먼트는 output_layer_flag[i][j] 를 포함하고,

상기 i 는 출력 계층 세트의 인덱스를 나타내고, 상기 j 와 상기 lsIdx 는 계층 세트의 인덱스를 나타내는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 12

제 9 항에 있어서,

상기 SPS 는 비디오 유용성 정보 (Video Usability Information; VUI) 를 포함하며, 그리고

타이밍 정보는 0 보다 더 큰 계층 ID 를 갖는 SPS 의 VUI 내에 포함되지 않는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 13

제 9 항에 있어서,

상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나와 연관된 타겟 출력 계층 세트는 디폴트 출력 계층 세트이고, 그리고

상기 디폴트 출력 계층 세트에 대한 상기 타겟 출력 계층은 명시적으로 시그널링되지 않는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 14

제 9 항에 있어서,

상기 SPS 에 대한 계층 ID 는 제로 (0) 보다 더 크고, 그리고

sps_max_sub_layers_minus1 및 sps_temporal_id_nesting_flag 는 상기 SPS 에 포함되지 않는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 15

제 9 항에 있어서,

상기 비디오 인코더에 의해, 0 과 동일한 `output_layer_set_idx[i]` 를 위해 사용된 상기 비디오 데이터에 대한
신택스 엘리먼트 `layer_id_included_flag[][]` 는 계층 세트 0 을 위한 것임을 결정하는 단계; 및

상기 비디오 인코더에 의해, 상기 결정에 기초하여 `output_layer_set_idx[i]` 를
`output_layer_set_idx_minus1[i]` 신택스 엘리먼트로 변경하는 단계를 더 포함하고,

상기 `i` 는 출력 계층 세트의 인덱스를 나타내는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 16

제 9 항에 있어서,

상기 비디오 인코더에 의해, 비디오 파라미터 세트 (video parameter set; VPS) 확장물에서의 프로파일, 티어,
및 레벨 신택스 구조들의 비-엔트로피 인코딩된 세트를 상기 인코딩된 비디오 비트스트림으로 인코딩하는 단계;
및

상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트를 포함하는 복수의 출력 계층 세트들의 각각에 대한 상
기 프로파일, 티어, 및 레벨 신택스 구조들 중 하나를 참조하는 단계를 더 포함하고,

상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대한 상기 비디오 데이터를 인코딩
하는 단계는, 상기 비디오 인코더에 의해, 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트의 각각의
출력 계층 세트에 대해 참조되는 상기 프로파일, 티어, 또는 레벨 신택스 구조 중 하나 이상으로부터의 정보에
기초하여 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대한 상기 비디오 데이터
를 인코딩하는 단계를 포함하는, 다층 비디오 데이터를 인코딩하는 방법.

청구항 17

비디오 데이터를 디코딩하기 위한 장치로서,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

인코딩된 비디오 비트스트림으로부터, 다층 비디오 데이터를 수신하는 것으로서, 상기 다층 비디오 데이터는,

상기 다층 비디오 데이터의 계층 세트에 포함된 제 1 출력 계층 세트로서, 상기 제 1 출력 계층 세트가
타겟 출력 계층 세트들의 제 1 리스트와 연관되도록, 상기 제 1 출력 계층 세트는 상기 다층 비디오 데이터의
상기 계층 세트의 서브세트를 나타내는, 상기 제 1 출력 계층 세트;

상기 다층 비디오 데이터의 상기 계층 세트에 포함된 제 2 출력 계층 세트로서, 상기 제 2 출력 계층
세트가 타겟 출력 계층 세트들의 제 2 리스트와 연관되도록, 상기 제 2 출력 계층 세트는 상기 다층 비디오 데
이터의 상기 계층 세트의 서브세트를 나타내는, 상기 제 2 출력 계층 세트;

상기 다층 비디오 데이터에 대한 시퀀스 파라미터 세트 (SPS);

상기 다층 비디오 데이터의 제 1 계층으로서, 상기 제 1 계층은 상기 SPS 와 연관되는, 상기 다층 비디
오 데이터의 제 1 계층;

상기 다층 비디오 데이터의 제 2 계층으로서, 상기 제 2 계층은, 상기 제 1 계층과는 상이한 값의 공간
적 해상도, 비트 깊이, 또는 컬러 포맷 중 적어도 하나를 가지고 그리고 상기 SPS 와 연관되는, 상기 다층 비디
오 데이터의 제 2 계층

을 포함하는, 상기 다층 비디오 데이터를 수신하는 것을 행하고;

상기 타겟 출력 계층 세트들의 제 1 리스트 또는 상기 타겟 출력 계층 세트들의 제 2 리스트 중 적어도 하나를
이용하여 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해, 인코딩된 비디오

데이터를 디코딩하며; 그리고

상기 SPS 에 포함된 정보를 이용하여 상기 제 1 계층 및 상기 제 2 계층에 대해, 인코딩된 비디오 데이터를 디코딩하도록 구성되는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 18

제 17 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 비디오 데이터에 대한 제 1 선택스 엘리먼트를 상기 비디오 데이터에 대한 제 2 선택스 엘리먼트로 변경하도록 더 구성되는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 19

제 18 항에 있어서,

상기 제 1 선택스 엘리먼트는 output_layer_flag[lsIdx][j] 를 포함하고 상기 제 2 선택스 엘리먼트는 output_layer_flag[i][j] 를 포함하고,

상기 i 는 출력 계층 세트의 인덱스를 나타내고, 상기 j 와 상기 lsIdx 는 계층 세트의 인덱스를 나타내는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 20

제 17 항에 있어서,

상기 SPS 는 비디오 유용성 정보 (Video Usability Information; VUI) 를 포함하며, 그리고

타이밍 정보는 0 보다 더 큰 계층 ID 를 갖는 SPS 의 VUI 내에 포함되지 않는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 21

제 17 항에 있어서,

상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트 중 적어도 하나와 연관된 타겟 출력 계층 세트는 디폴트 출력 계층 세트이고, 그리고

상기 디폴트 출력 계층 세트에 대한 상기 타겟 출력 계층은 명시적으로 시그널링되지 않는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 22

제 17 항에 있어서,

상기 SPS 에 대한 계층 ID 는 제로 (0) 보다 더 크고, 그리고

sps_max_sub_layers_minus1 및 sps_temporal_id_nesting_flag 는 상기 SPS 에 포함되지 않는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 23

제 17 항에 있어서,

상기 하나 이상의 프로세서들은,

0 과 동일한 output_layer_set_idx[i] 를 위해 사용된 상기 비디오 데이터에 대한 선택스 엘리먼트 layer_id_included_flag[][] 는 계층 세트 0 을 위한 것임을 결정하고; 그리고

상기 결정에 기초하여 output_layer_set_idx[i] 를 output_layer_set_idx_minus1[i] 선택스 엘리먼트로 변경하도록 더 구성되고,

상기 i 는 출력 계층 세트의 인덱스를 나타내는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 24

제 17 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 인코딩된 비디오 비트스트림으로부터, 비디오 파라미터 세트 (video parameter set; VPS) 확장물에서의 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트를 수신하고; 그리고

상기 제 1 출력 계층 세트 및 상기 제 2 출력 계층 세트를 포함하는 복수의 출력 계층 세트들의 각각에 대한 상기 프로파일, 티어, 및 레벨 선택스 구조들 중 하나를 참조하도록 더 구성되고,

상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해 상기 인코딩된 비디오 데이터를 디코딩하기 위해, 상기 하나 이상의 프로세서들은, 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트의 각각의 출력 계층 세트에 대해 참조되는 상기 프로파일, 티어, 또는 레벨 선택스 구조 중 하나 이상으로부터의 정보에 기초하여 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해 상기 인코딩된 비디오 데이터를 디코딩하도록 구성되는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 25

비디오 데이터를 디코딩하기 위한 장치로서,

인코딩된 비디오 비트스트림으로부터, 다층 비디오 데이터를 수신하는 수단으로서, 상기 다층 비디오 데이터는,

상기 다층 비디오 데이터의 계층 세트에 포함된 제 1 출력 계층 세트로서, 상기 제 1 출력 계층 세트가 타겟 출력 계층 세트들의 제 1 리스트와 연관되도록, 상기 제 1 출력 계층 세트는 상기 다층 비디오 데이터의 상기 계층 세트의 서브세트를 나타내는, 상기 제 1 출력 계층 세트;

상기 다층 비디오 데이터의 계층 세트에 포함된 제 2 출력 계층 세트로서, 상기 제 2 출력 계층 세트가 타겟 출력 계층 세트들의 제 2 리스트와 연관되도록, 상기 제 2 출력 계층 세트는 상기 다층 비디오 데이터의 상기 계층 세트의 서브세트를 나타내는, 상기 제 2 출력 계층 세트;

상기 다층 비디오 데이터에 대한 시퀀스 파라미터 세트 (SPS);

상기 다층 비디오 데이터의 제 1 계층으로서, 상기 제 1 계층은 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 1 계층;

상기 다층 비디오 데이터의 제 2 계층으로서, 상기 제 2 계층은, 상기 제 1 계층과는 상이한 값의 공간적 해상도, 비트 깊이, 또는 컬러 포맷 중 적어도 하나를 가지고 그리고 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 2 계층

을 포함하는, 상기 다층 비디오 데이터를 수신하는 수단;

상기 타겟 출력 계층 세트들의 제 1 리스트 또는 상기 타겟 출력 계층 세트들의 제 2 리스트 중 적어도 하나를 이용하여 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해, 인코딩된 비디오 데이터를 디코딩하는 수단; 및

상기 SPS 에 포함된 정보를 이용하여 상기 제 1 계층 및 상기 제 2 계층에 대해, 인코딩된 비디오 데이터를 디코딩하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 26

명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 하나 이상의 프로세서들에 의한 실행 시, 상기 하나 이상의 프로세서들로 하여금,

인코딩된 비디오 비트스트림으로부터, 다층 비디오 데이터를 수신하는 것으로서, 상기 다층 비디오 데이터는,

상기 다층 비디오 데이터의 계층 세트에 포함된 제 1 출력 계층 세트로서, 상기 제 1 출력 계층 세트가 타겟 출력 계층 세트들의 제 1 리스트와 연관되도록, 상기 제 1 출력 계층 세트는 상기 다층 비디오 데이터의 상기 계층 세트의 서브세트를 나타내는, 상기 제 1 출력 계층 세트;

상기 다층 비디오 데이터의 계층 세트에 포함된 제 2 출력 계층 세트로서, 상기 제 2 출력 계층 세트가

타겟 출력 계층 세트들의 제 2 리스트와 연관되도록, 상기 제 2 출력 계층 세트는 상기 다층 비디오 데이터의 상기 계층 세트의 서브세트를 나타내는, 상기 제 2 출력 계층 세트;

상기 다층 비디오 데이터에 대한 시퀀스 파라미터 세트 (SPS);

상기 다층 비디오 데이터의 제 1 계층으로서, 상기 제 1 계층은 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 1 계층;

상기 다층 비디오 데이터의 제 2 계층으로서, 상기 제 2 계층은, 상기 제 1 계층과는 상이한 값의 공간적 해상도, 비트 깊이, 또는 컬러 포맷 중 적어도 하나를 가지고 그리고 상기 SPS 와 연관되는, 상기 다층 비디오 데이터의 제 2 계층

을 포함하는, 상기 다층 비디오 데이터를 수신하는 것을 행하게 하고;

상기 타겟 출력 계층 세트들의 제 1 리스트 또는 상기 타겟 출력 계층 세트들의 제 2 리스트 중 적어도 하나를 이용하여 상기 제 1 출력 계층 세트 또는 상기 제 2 출력 계층 세트 중 적어도 하나에 대해, 인코딩된 비디오 데이터를 디코딩하게 하며; 그리고

상기 SPS 에 포함된 정보를 이용하여 상기 제 1 계층 및 상기 제 2 계층에 대해, 인코딩된 비디오 데이터를 디코딩하게 하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2013년 4월 8일자로 출원된 미국 가출원 제61/809,858호를 우선권 주장하며, 본원에 그 전부가 참조로 통합된다.

기술 분야

[0003] 본 개시물은 비디오 코딩에서의 파라미터 세트들의 코딩에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말기들 (PDA들), 랩톱 또는 테스크톱 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (Advanced Video Coding, AVC)에 의해 규정된 표준들, 고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장본들에 기재된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현하는 것에 의해 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 공간적 (화상 내) 예측 및/또는 시간적 (화상 간) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분)는 비디오 블록들로 구획될 수도 있으며, 그 비디오 블록들은 또한 트리블록들 (treeblocks), 코딩 단위들 (CU들) 및/또는 코딩 노드들이라고 지칭될 수도 있다. 화상의 인트라

코딩식 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측을 이용하여 인코딩된다. 화상의 인터 코딩식 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 화상들에서의 참조 샘플들에 관한 시간적 예측을 이용하여 코딩될 수도 있다. 화상들은 프레임들이라고 지칭될 수도 있고, 참조 화상들은 참조 프레임들이라고 지칭될 수도 있다.

[0006] 공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록이 생겨나게 한다. 잔차 데이터는 코딩될 원본 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터 코딩식 블록이 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터와, 코딩된 블록 및 예측 블록 사이의 차이를 나타내는 잔차 데이터에 따라 인코딩된다. 인트라 코딩식 블록이 인트라 코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 결과적으로 잔차 변환 계수들이 생겨나며, 그 계수들은 그 다음에 양자화될 수도 있다. 처음에는 2차원 어레이로 배열된 양자화된 변환 계수들은, 변환 계수들의 1차원 벡터를 생성하기 위하여 스캐닝될 수도 있고, 엔트로피 코딩이 추가의 압축을 달성하기 위해 변환계수들에 적용될 수도 있다.

발명의 내용

과제의 해결 수단

[0007] 본 개시물은 하나를 초과하는 출력 계층 세트가 하나의 계층 세트에 대해 코딩될 수도 있도록 파라미터 세트들을 코딩하는 기법들에 관련한다. 하나의 예에서, 계층 세트에 대한 제 1 출력 계층 세트와 그 계층 세트에 대한 제 2 출력 계층 세트가 비디오 데이터를 코딩하는데 사용될 수도 있다. 예를 들어, 비디오 데이터는 제 1 출력 계층 세트를 위해 그리고 제 2 출력 계층 세트를 위해 코딩될 수도 있다.

[0008] 본 개시물은 파라미터 세트들을 코딩하는 기법들에 관련한다. 예를 들어, 비디오 코더가 비디오 파라미터 세트 (video parameter set, VPS) 에서의 모든 프로파일, 티어 (tier), 및/또는 레벨 정보를 코딩할 수도 있고, 프로파일, 티어, 레벨 정보는 몇몇 예들에서 그 정보가 엔트로피 디코딩 없이 액세스 가능한 방식으로, 하나의 프로파일, 티어, 레벨 신택스 (syntax) 구조에 의해 시그널링된 각각의 출력 계층 세트와 연관될 수도 있다. 하나의 예에서, 본 개시물은 파라미터 세트 확장 신택스 (parameter set extension syntax), 이를테면 비디오 파라미터 세트 (VPS) 확장 신택스에서의 profile_tier_level 신택스 구조들을, 몇몇 예들에서는 엔트로피 코딩 없이 액세스 가능한 (즉, 파라미터 세트 확장물에서의 임의의 엔트로피 코딩된 신택스 엘리먼트들보다 앞선) 파라미터 세트 확장 신택스 내의 포지션에서 코딩하는 기법들을 기술한다. 일부 예들에서 VPS에서의 모든 프로파일, 티어, 및/또는 레벨 정보는 엔트로피 코딩 없이 액세스 가능할 수도 있지만, 다른 예들에서, 감소된 양의 엔트로피 코딩 또는 일부 제한된 엔트로피 코딩이 VPS에서의 일부 또는 전체 프로파일, 티어, 및/또는 레벨 정보에 대해 요구될 수도 있다.

[0009] 다른 예에서, 본 개시물은 VPS에서 다층 (multilayer) 비디오 데이터에서의 계층 의존성 정보를 엔트로피 디코딩 없이 액세스 가능하도록 코딩하는 기법들에 관련한다. 몇몇 예들은, 엔트로피 인코딩되는 파라미터 세트 확장 신택스의 신택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 송신 또는 수신하는 것을 포함하여, 다층 비디오 데이터를 코딩할 수도 있다. 그 기법은 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 코딩하는 것을 포함할 수도 있다. 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타낸다.

[0010] 다른 예에서, 본 개시물은 파라미터 세트들을 코딩하는 기법들에 관련한다. 예를 들어, 비디오 코더가 VPS에서의 모든 프로파일, 티어, 및/또는 레벨 정보를 코딩할 수도 있고, 프로파일, 티어, 레벨 정보는 몇몇 경우들에서 그 정보가 엔트로피 디코딩 없이 액세스 가능한 방식으로, 하나의 프로파일, 티어, 레벨 신택스 구조에 의해 시그널링된 각각의 출력 계층 세트와 연관될 수도 있다. 일부 예들에서 VPS에서의 모든 프로파일, 티어, 및/또는 레벨 정보는 엔트로피 코딩 없이 액세스 가능할 수도 있지만, 다른 예들에서, 감소된 양의 엔트로피 코딩 또는 일부 제한된 엔트로피 코딩이 VPS에서의 일부 또는 전체 프로파일, 티어, 및/또는 레벨 정보에 대해 요구될 수도 있다.

[0011] 다른 예에서, 본 개시물은 VPS에서의 계층마다 시각적 신호 정보, 이를테면 video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs를 코딩하는 기법들에

관련한다.

[0012] 다른 예에서, 본 개시물은 상이한 공간적 분해능들, 비트 깊이, 또는 컬러 포맷들의 계층들에 의한 SPS 공유를 위한 기법들에 관련한다.

[0013] 다른 예에서, 본 개시물은 타이밍 정보가 0보다 더 큰 계층 ID (nuh_layer_id) 를 갖는 시퀀스 파라미터 세트 (sequence parameter set, SPS) 의 비디오 유용성 정보 (video usability information, VUI) 에서 제공되지 않도록 비디오 코딩하는 기법들에 관련한다.

[0014] 다른 예에서, 본 개시물은 타겟 출력 계층들의 명시적 시그널링 (explicit signaling) 이 디폴트 (default) 출력 계층 세트들에 대해 제공되지 않도록 비디오 코딩하는 기법들에 관련한다.

[0015] 다른 예에서, 본 개시물은 각각의 코딩된 비디오 시퀀스 (coded video sequence, CVS) 에서 존재할 수도 있는 시간적 서브-계층들의 최대 수 (sps_max_sub_layers_minus1) 또는 인터 예측이 0보다 더 큰 계층 ID를 갖는 SPS들에서의 CVS들에 대해 추가로 제한되는지의 여부 (sps_temporal_id_nesting_flag) 의 시그널링을 피하는 기법들에 관련한다.

[0016] 다른 예에서, 본 개시물은 본 개시물의 기법들에 따라 output_layer_set_idx_minus1[i]로 변경될 수도 있는 신 턱스 엘리먼트 output_layer_set_idx[i]를 위한 기법들에 관련한다.

[0017] 하나의 예에서, 본 개시물은 다층 비디오 데이터를 디코딩하는 방법을 기술하는데, 그 방법은 계층 세트에 대한 제 1 출력 계층 세트를 수신하는 단계, 그 계층 세트에 대한 제 2 출력 계층 세트를 수신하는 단계, 및 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나에 대한 비디오 데이터를 디코딩하는 단계를 포함한다.

[0018] 다른 예에서, 본 개시물은 다층 비디오 데이터를 인코딩하는 방법을 기술하는데, 그 방법은, 제 1 출력 계층 세트 및 제 2 출력 계층 세트 양쪽 모두가 하나의 계층 세트에 대한 것인, 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나에 대한 비디오 데이터를 인코딩하는 단계, 그 계층 세트에 대한 제 1 출력 계층 세트를 송신하는 단계, 및 그 계층 세트에 대한 제 2 출력 계층 세트를 송신하는 단계를 포함한다.

[0019] 다른 예에서, 본 개시물은 비디오 데이터를 디코딩하는 장치를 기술하는데, 그 장치는 비디오 데이터를 저장하도록 구성된 메모리와, 계층 세트에 대한 제 1 출력 계층 세트를 수신하며, 그 계층 세트에 대한 제 2 출력 계층 세트를 수신하고, 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나에 대한 비디오 데이터를 디코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0020] 하나의 예에서, 본 개시물은 비디오 데이터를 디코딩하는 장치를 기술하는데, 그 장치는 계층 세트에 대한 제 1 출력 계층 세트를 수신하는 수단, 그 계층 세트에 대한 제 2 출력 계층 세트를 수신하는 수단, 및 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나에 대한 비디오 데이터를 디코딩하는 수단을 포함한다.

[0021] 다른 예에서, 본 개시물은 컴퓨터 판독가능 저장 매체를 기술한다. 컴퓨터 판독가능 저장 매체는 실행 시 하나 이상의 프로세서들로 하여금 계층 세트에 대한 제 1 출력 계층 세트를 수신하게 하며, 그 계층 세트에 대한 제 2 출력 계층 세트를 수신하게 하고, 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나에 대한 비디오 데이터를 디코딩하게 하는 명령들을 저장하고 있다.

[0022] 하나 이상의 예들의 세부사항들은 첨부 도면들 및 다음의 설명에서 전개된다. 다른 특징들, 목적들, 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

[0023] 도 1은 본 개시물에서 설명되는 기법들을 이용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 도시하는 블록도이다.

도 2는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 도시하는 블록도이다.

도 3은 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 도시하는 블록도이다.

도 4는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 5는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 다른 예의 방법을 도시하는 흐름도이다.

도 6은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 7은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 8은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 9는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 10은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 11은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 12는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 13은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0024]

본 개시물은 대체로 코딩 및 압축의 분야에 관련된다. 하나의 예로서, 본 개시물은 현재 개발 중인 고 효율 비디오 코딩 (HEVC) 표준에 관련된다. "코딩"이란 용어는 인코딩 및 디코딩을 지칭하고, 그 기법들은 인코딩, 디코딩 또는 인코딩 및 디코딩 양쪽 모두에 적용될 수도 있다. 더 상세히 설명되는 바와 같이, 그 기법들은 화상 내의 블록이 동일한 화상에서의 다른 블록 또는 블록들에 대하여 예측되는 (즉, 공간 예측) 인트라 코딩 (예컨대, 인트라 예측)에 관련될 수도 있다.

[0025]

이전의 비디오 코딩 시스템들에서, 계층 세트들에 대한 프로파일, 티어 및 레벨의 액세스는 엔트로피 디코딩을 요구한다. 마찬가지로, 계층 의존성 정보는 엔트로피 디코딩 없이 액세스가능하지 않다. 출력 계층 세트의 시그널링은 또한 문제가 있고 비효율적일 수도 있다.

[0026]

본원에서 설명되는 시스템들, 방법들, 및 디바이스들 중 하나의 예가, 파라미터 세트 확장 선택스, 이를테면 비디오 파라미터 세트 (VPS) 확장 선택스에서 시그널링될 profile_tier_level() 선택스 구조들의 세트를, 잠재적으로는 엔트로피 코딩 없이 액세스가능한 (예컨대, 확장 선택스에서 엔트로피 코딩된 엘리먼트들보다 앞선) 확장 선택스 내의 포지션에서, 그리고/또는 얼마간의 감소된 또는 제한된 엔트로피 코딩으로 제공한다. VPS 확장 선택스는 예를 목적으로 본 개시물 전체에 걸쳐 설명될 것이다. 일 예에서, VPS는 각각의 슬라이스 세그먼트 헤더에서 발견된 선택스 엘리먼트에 의해 참조되는 PPS에서 발견된 선택스 엘리먼트에 의해 참조되는 SPS에서 발견된 선택스 엘리먼트의 내용에 의해 결정되는 바와 같은, 영 이상의 전체 코딩된 비디오 시퀀스들 (CVS들)에 적용하는 선택스 엘리먼트들을 포함하는 선택스 구조이다. 일 예에서, CVS가, 디코딩 순서에서, 1과 동일한 NoRas1OutputFlag를 갖는 IRAP 액세스 단위와, 뒤따르는, 1과 동일한 NoRas1OutputFlag를 갖는 IRAP 액세스 단위인 임의의 후속 액세스 단위 전까지를 포함하는 모든 후속 액세스 단위들을 포함하는 1과 동일한 NoRas1OutputFlag를 갖는 IRAP 액세스 단위들이 아닌 영 이상의 액세스 단위들을 포함하는 액세스 단위들의 시퀀스이다. 더욱이, profile_tier_level() 선택스 구조는 계층 세트 대신에 출력 계층 세트와 연관될 수도 있는데, 계층 세트가 하나를 초과하는 출력 계층 세트에 대응할 수도 있고, 하나의 계층 세트에 대응하는 상이한 출력 계층 세트들이 상이한 레벨들에 따를 수도 있기 때문이다.

[0027]

profile_tier_level() 선택스 구조는 프로파일들, 티어들, 및 레벨들에 관련된 선택스 정보를 제공하는 선택스 구조이다. 프로파일들, 티어들, 및 레벨들은 HEVC 표준 또는 HEVC 확장 표준이 유사한 기능적 요건들을 가질 수도 있는 다양한 애플리케이션들에 걸쳐 상호운용적이도록 그 HEVC 표준 또는 HEVC 확장 표준을 구현하기 위한 상이한 적합성 (conformance) 포인트들을 특정한다. 몇몇 예들에서, 프로파일이 부합하는 (conforming) 비트스트림을 생성함에 있어서 사용될 수도 있는 코딩 도구들 또는 알고리즘들의 세트를 정의할 수도 있다. 몇몇 예들에서, 티어와 레벨이 비트스트림의 파라미터들의 일부에 대해 제약조건들을 둘 수도

있다. 몇몇 예들에서, 이를 파라미터들은 디코더 프로세싱 부하 및 디코더 메모리 능력들에 대응할 수도 있다. 레벨 정보는 데이터의 프로세싱 상의 제약을 확립할 수도 있다. 예를 들어, 레벨 정보는 최대 샘플레이트, 최대 화상 사이즈, 최대 비트 레이트, 최소 압축 비율을 포함할 수도 있다. 레벨 정보는 디코딩된 화상 버퍼 (DPB) 및 코딩된 화상 버퍼 (CPB)의 용량들을 또한 포함할 수도 있다. CPB는 압축된 데이터를 그것의 디코딩 전에 데이터 흐름 관리 목적으로 보유한다. DPB는 디코딩된 화상 데이터를 보유한다.

[0028] 본원에서 설명되는 시스템들, 방법들, 및 디바이스들의 다른 예가 비디오 데이터의 계층들을 포함하는 다층 비디오 데이터의 코딩을 제공한다. 이를 시스템들, 방법들, 및 디바이스들은 파라미터 세트 확장 선택스, 이를테면 VPS 확장 선택스 내의 엔트로피 인코딩되는 확장본의 선택스 엘리먼트들보다 앞선 포지션에서, 비-엔트로피 인코딩된 계층 의존성 정보를 송신하는 것 또는 수신하는 것을 포함할 수도 있다. 덧붙여, 시스템들, 방법들, 및 디바이스들은 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 디코딩할 수도 있다. 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타낸다. 다시, VPS 확장 선택스는 예를 목적으로 본 개시물 전체에 걸쳐 설명될 것이다.

[0029] 아래에서 더 상세히 설명되는 바와 같이, 하나의 예에서, 하나를 초과하는 출력 계층 세트가 하나의 계층 세트에 대해 시그널링될 수도 있다. 예를 들어, 본원에서 설명된 바와 같이, 선택스 엘리먼트 output_layer_flag[lsIdx][j]는 output_layer_flag[i][j]로 변경될 수도 있고, 관련된 시맨틱스는 그에 따라 변경된다.

[0030] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, 프로파일, 티어, 및 레벨 정보는 그 정보가 엔트로피 디코딩 없이 액세스 가능한 방식으로 VPS에서 시그널링될 수도 있다. 이는 아래의 표들 중 하나 이상의 표에서, ue(v) (ue(v)는 엔트로피 코딩을 나타냄) 외의 기술자들을 갖는 파라미터들로 예시되어 있다. 예를 들어, 그 기술자들은 고정 길이 코딩 (fixed length coding), 예컨대, u(n)을 나타낼 수도 있으며, 여기서 n은 1 이상의 정수이다.

[0031] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, 계층 의존성 정보는 그 정보가 엔트로피 없이 액세스 가능한 방식으로 VPS에서 시그널링될 수도 있다. 이는 아래의 표들 중 하나 이상의 표에서 ue(v) 외의 기술자들을 갖는 파라미터들로 예시되어 있다. 예를 들어, 그 기술자들은 고정 길이 코딩, 예컨대, u(n)을 나타낼 수도 있으며, 여기서 n은 1 이상의 정수이다.

[0032] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, 표현 포맷이 그 포맷이 엔트로피 디코딩 또는 얼마간의 감소된 또는 제한된 엔트로피 코딩 없이 액세스 가능한 방식으로 VPS에서 시그널링될 수도 있다. 예를 들어, chroma_format_vps_idc, separate_colour_plane_vps_flag, pic_width_vps_in_luma_samples, pic_height_vps_in_luma_samples, bit_depth_vps_luma_minus8, 및 bit_depth_vps_chroma_minus8과 같은 표현 포맷 파라미터들은 VPS에서 시그널링되고, 몇몇 예들에서 엔트로피 코딩되지 않는다. 이는 아래의 표들 중 하나 이상의 표에서 ue(v) 외의 기술자들을 갖는 이를 파라미터들로 예시되어 있다. 표현 포맷 파라미터들은 특정 파라미터가 고정 길이 코딩, 예컨대, u(n)이며, 여기서 n이 1 이상의 정수임을 나타내는 기술자들을 가질 수도 있다. "_vps"가 다양한 파라미터 이름들로부터 떼어질 수 있는 몇몇 경우들에서, 예컨대, chroma_format_vps_idc는 chroma_format_idc라고, separate_colour_plane_vps_flag는 separate_colour_plane_flag라고, pic_width_vps_in_luma_samples는 pic_width_in_luma_samples라고, pic_height_vps_in_luma_samples는 pic_height_in_luma_samples라고, bit_depth_vps_luma_minus8는 bit_depth_luma_minus8라고, 그리고 bit_depth_vps_chroma_minus8는 bit_depth_vps_chroma_minus8라고 또한 지칭될 수도 있고, 뿐만 아니라 다른 파라미터들에 대해서도 그려할 수 있다.

[0033] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, video_format, video_full_range_flag, colour_primaries, transfer_characteristics, 또는 matrix_coeffs와 같은 시각적 신호 정보는 VPS에서의 계층마다 시그널링될 수도 있다. 본원에서 설명되는 바와 같이, video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs와 같은 시각적 신호 정보는, 예를 들어, HEVC 표준의 부록 E에서 특정된 바와 같이, 비디오 유용성 정보 (VUI) 파라미터들의 서브세트일 수도 있다. VUI와, 그래서 시각적 신호 정보는 VPS 내에 있고 VPS에서의 계층마다 시그널링될 수도 있다.

[0034] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, 시퀀스 파라미터 세트 (SPS) 공유가 상이한 공간적 분해 능들, 비트 깊이, 또는 컬러 포맷들을 갖는 계층들에 의해 수행될 수도 있다. 일 예에서, SPS는 각각의 슬라이스 세그먼트 헤더에서 발견된 선택스 엘리먼트에 의해 참조되는 PPS에서 발견된 선택스 엘리먼트의 내용에

의해 결정된 바와 같은 영 이상의 전체 CVS들에서의 계층에 적용하는 선택스 엘리먼트들을 포함하는 선택스 구조이다. 몇몇 예들에서, 다층 비디오 데이터를 디코딩하는 방법이 비디오 파라미터 세트 및 제 1 시퀀스 파라미터 세트를 수신하는 단계를 포함할 수도 있다. 그 방법은 제 1 계층을 참조하게 하는 제 1 시퀀스 파라미터 세트를 수신하는 단계, 제 1 계층과는 상이한 값의 공간적 해상도, 비트 깊이, 및 컬러 포맷 중 적어도 하나를 갖고 제 1 시퀀스 파라미터 세트를 참조하는 제 2 계층을 수신하는 단계를 또한 포함할 수도 있다. 더욱이, 예의 방법은 비디오 파라미터 세트 및 제 1 시퀀스 파라미터 세트로부터의 정보에 기초하여 제 1 계층 및 제 2 계층의 비디오 데이터를 디코딩하는 단계를 포함할 수도 있다.

[0035] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, 타이밍 정보가 0보다 더 큰 계층 ID (nuh_layer_id) 를 갖는 SPS들의 VUI에서 코딩되지 않는다. 몇몇 예들에서, 0보다 더 큰 계층 ID를 갖는 SPS를 참조하는 계층의 경우, VPS에서 시그널링된 타이밍 정보는 사용된다.

[0036] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, 타겟 출력 계층들의 명시적 시그널링이 디폴트 출력 계층 세트들에 대해 사용되지 않는다. 예를 들어, 0과 동일한 multiple_output_layer_sets_in_layer_set_flag를 갖는 것은 단지 하나의 출력 계층 세트가 각각의 계층 세트에 대해 VPS에 의해 특정되고 최고 계층이 유일한 타겟 출력 계층임을 특정하기 때문에, 디폴트 출력 계층 세트들에 대한 타겟 출력 계층들의 명시적 시그널링은 필요하지 않다. 예를 들어, multiple_output_layer_sets_in_layer_set_flag가 0과 동일한 경우 계층들이 출력될 것임을 시그널링할 필요가 없는데, 각각의 계층 세트에 대해 하나의 출력 계층 세트만 있고 최고 계층이 유일한 타겟 출력 계층이기 때문이다.

[0037] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, sps_max_sub_layers_minus1 및 sps_temporal_id_nesting_flag는 SPS에서 시그널링되지 않는다. 몇몇 경우들에서, 이 시그널링은 nuh_layer_id > 0인 경우 SPS에서 발생하지 않는다.

[0038] 아래에서 더 상세히 설명되는 바와 같이, 다른 예에서, 선택스 엘리먼트 output_layer_set_idx[i]는 output_layer_set_idx_minus1[i]로 변경될 수도 있다. 이는 0과 동일한 output_layer_set_idx[i]를 위해 사용된 선택스 엘리먼트 layer_id_included_flag[i][j]가 계층 세트 0을 위한 것인 반면, 계층 세트 0의 경우 layer_id_included_flag[i][j]가 정의되지 않기 때문이다. 덧붙여, 계층 세트 0에 대한 타겟 출력 계층은 항상 계층 0 (기본 계층) 이다.

[0039] 위의 다양한 예들은 엔트로피 코딩되지 않는 다양한 데이터, 파라미터들 등을 갖는 양태들을 포함한다. 이 때문에, 데이터, 파라미터들 등을 엔트로피 코딩을 수행하는 엔트로피 코더 없이 디바이스들에 대해 액세스 가능할 수도 있다. 예를 들어, 몇몇 경우들에서 MANEs (Media Aware Network Entities) 가 엔트로피 코딩 디바이스를 갖지 않을 수도 있지만, 데이터, 파라미터들 등과 같은 비-엔트로피 코딩된 정보를 일반적으로 코딩 (예컨대, 디코딩) 할 수 있었을 것이다.

[0040] "HEVC 규격 초안 10" 또는 "WD10"이라고 지칭되는 HEVC 표준의 최근 초안이, 문서 JCTVC-L1003v34, 『Bross et al., "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, CH, 14-23 January, 2013』에 기재되어 있으며, 이는 2013년 10월 15일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip으로부터 다운로드 가능하다.

[0041] "HEVC 규격 초안 6" 또는 "WD6"이라고 지칭되는 HEVC의 다른 최근 규격 초안 (WD) 이, 문서 JCTVC-H1003, 『Bross et al., "High-Efficiency Video Coding (HEVC) text specification draft 6," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 8th Meeting: San Jose, California, USA, Feb. 2012』에 기재되어 있으며, 이는 2013년 10월 15일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H1003-v22.zip으로부터 다운로드 가능하다.

[0042] "HEVC 규격 초안 8" 또는 "WD8"이라고 지칭되는 HEVC의 다른 최근 규격 초안 (WD) 이, 문서 JCTVC-J1003, 『Bross et al., "High-Efficiency Video Coding (HEVC) text specification draft 8," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting: Stockholm, SE, 11-20 July 2012』에 기재되어 있으며, 이는 2013년 10월 15일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip으로부터 다운로드 가능하다.

[0043] HEVC 범위 확장본들이 문서 『JCTVC-N1005_v3, Flynn et al., "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 4," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 13th Meeting: Incheon, KR, 18-26 April 2013』에 기재되어 있으며, 이는 2013년 10월 15일 현재, http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=8139로부터 다운로드 가능하다.

[0044] 도 1은 본 개시물에서 설명되는 기법들을 이용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템 (10) 을 도시하는 블록도이다. 도 1에 예시된 바와 같이, 시스템 (10) 은 비디오 디코더 (30) 에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 생성하는 비디오 인코더 (20) 를 구비한다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 노트북 (즉, 랙톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화기 핸드셋들, 이른바 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 매우 다양한 디바이스들 중 임의의 것을 포함할 수도 있다. 몇몇 경우들에서, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 무선 통신을 위해 장비될 수도 있다.

[0045] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 링크 (16) 를 통해 수신할 수도 있다. 링크 (16) 는 비디오 인코더 (20) 로부터 비디오 디코더 (30) 로 인코딩된 비디오 데이터를 이동시킬 수 있는 임의의 유형의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 링크 (16) 는 비디오 인코더 (20) 가 인코딩된 비디오 데이터를 비디오 디코더 (30) 로 직접 실시간으로 송신하는 것을 가능하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 통신 표준, 이를테면 무선 통신 프로토콜에 따라 변조되고 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 이를테면 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0046] 대안으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (34) 로 출력될 수도 있다. 마찬가지로, 입력 인터페이스 (28) 는 저장 디바이스 (34) 로부터의 인코딩된 데이터에 액세스할 수도 있다. 저장 디바이스 (34) 는 하드 드라이브, 블루 레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산형 또는 국소적으로 액세스되는 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다. 추가의 예에서, 저장 디바이스 (34) 는 비디오 인코더 (20) 에 의해 생성된 인코딩된 비디오를 보유할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 해당할 수도 있다. 목적지 디바이스 (14) 는 저장 디바이스 (34) 로부터의 저장된 비디오 데이터에 스트리밍 또는 다운로드를 통해 또는 입력 인터페이스 (28) 를 통해 액세스할 수도 있다. 그 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있고 그 인코딩된 비디오 데이터를 비디오 디코더 (30) 로 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버 (예컨대, 웹 사이트용), FTP (File Transfer Protocol) 서버, 네트워크 부속 스토리지 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 비디오 디코더 (30) 는 인터넷 접속을 포함한 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이는 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 양쪽 모두의 조합을 포함할 수도 있다. 저장 디바이스 (34) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.

[0047] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 반드시 제한되지는 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 중 임의의 것의 지원 하의 비디오 코딩, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 몇몇 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0048] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22) 를 구비한다. 몇몇 경우들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 구비할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오를 담고

있는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 피드 인터페이스, 및/또는 컴퓨터 그래픽스 데이터를 소스 비디오로서 생성하는 컴퓨터 그래픽 시스템과 같은 소스, 또는 그런 소스들의 조합을 포함할 수도 있다. 하나의 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 이른바 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명되는 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0049] 비디오 인코더 (20) 는 비디오 소스 (18)로부터 수신된 캡처된, 사전 캡처된, 또는 컴퓨터 생성된 비디오를 인코딩한다. 캡처된, 사전 캡처된, 또는 컴퓨터 생성된 비디오는 4:2:0, 4:2:2 또는 4:4:4 샘플 포맷들을 포함하는 위에서 설명된 샘플 포맷들 중 임의의 것에 따라 포맷될 수도 있다. 비디오 인코더 (20) 는 4:2:0, 4:2:2 또는 4:4:4 샘플 포맷들 중 임의의 것에 따라 포맷된 비디오에 대해 비디오 코딩을 수행할 수도 있다.

일부 경우들에서, 비디오 인코더 (20) 는 캡처된, 사전 캡처된, 또는 컴퓨터 생성된 비디오를 코딩 프로세스의 일부로서 업 샘플링 또는 다운 샘플링할 수도 있다. 예를 들어, 캡처된 비디오는 4:4:4 샘플 포맷에 따라 포맷될 수도 있으며, 비디오 인코더 (20) 는 캡처된 비디오를 4:2:2 포맷으로 다운 샘플링하고 다운 샘플링된 비디오에 대해 비디오 인코딩을 수행할 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안으로) 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 디코딩 및/또는 플레이백을 위한 나중의 액세스를 위해 저장 디바이스 (34) 상에 저장될 수도 있다.

[0050] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 구비한다. 몇몇 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 구비할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16) 를 통해 통신되거나 또는 저장 디바이스 (34) 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩함에 있어서, 비디오 디코더, 이를테면 비디오 디코더 (30) 에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 선택스 엘리먼트들을 포함할 수도 있다. 이러한 선택스 엘리먼트들은 통신 매체 상에서 송신되는, 저장 매체 상에 저장되는, 또는 파일 서버 상에 저장되는 인코딩된 비디오 데이터에 포함될 수도 있다.

[0051] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나, 또는 그것 외부에 있을 수도 있다. 몇몇 예들에서, 목적지 디바이스 (14) 는 통합형 디스플레이 디바이스를 포함할 수도 있고 또한 외부 디스플레이 디바이스와 인터페이싱하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0052] 비디오 인코더 (20) 와 비디오 디코더 (30) 는 비디오 압축 표준, 이를테면 현재 개발중인 고 효율 비디오 코딩 (HEVC) 표준에 따라 동작할 수도 있고 일반적으로 현재의 HEVC 테스트 모델 (HM) 또는 미래의 HM을 준수할 수도 있다.

[0053] 대안으로, 비디오 인코더 (20) 와 비디오 디코더 (30) 는, 다르게는 MPEG-4, 파트 10, 고급 비디오 코딩 (AVC) 이라고 지칭되는 ITU-T H.264 표준과 같은 다른 독점 또는 산업 표준들, 또는 그런 표준들의 개정들 또는 확장들에 따라 동작할 수도 있다. 본 개시물의 기법들은 HEVC 확장본들에 적용가능한 것으로서 설명되지만, 임의의 특정 코딩 표준으로 반드시 제한되지는 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2와 ITU-T H.263을 포함한다.

[0054] 비록 도 1에 도시되지 않았지만, 몇몇 양태들에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함하여, 공통 데이터 스트림 또는 개별 데이터 스트림들에서의 오디오 및 비디오 양쪽 모두의 인코딩을 핸들링할 수도 있다. 적용가능하다면, 몇몇 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜들에 부합할 수도 있다.

[0055] 비디오 인코더 (20) 와 비디오 디코더 (30) 각각은 다양한 적합한 인코더 회로, 이를테면 하나 이상의 마이크로 프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 개별 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그것들의 임의의 조합 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 부분적으로 구현되는 경우, 디바이스가 본 개시물의 기법들을 수행하기 위해, 적합한 비일시적 컴퓨터 판독가능 매체 내에 소프트웨어에 대한 명령들을 저장하고 하나 이상의 프

로 세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC)의 일부로서 개별 디바이스 내에 통합될 수도 있다.

[0056] 도 1의 도시된 예에서, 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)는 프로파일, 티어, 및 레벨 신택스 구조들의 비-엔트로피 인코딩된 세트를 송신 또는 수신할 수도 있다. 예를 들어, 비디오 인코더 (20)는 프로파일, 티어, 및 레벨 신택스 구조들의 비-엔트로피 인코딩된 세트를 송신할 수도 있고 비디오 디코더 (30)는, 예컨대 비디오 인코더 (20)로부터 전송된, 프로파일, 티어, 및 레벨 신택스 구조들의 비-엔트로피 인코딩된 세트를 수신할 수도 있다.

[0057] 프로파일, 티어, 및 레벨 신택스 구조들의 비-엔트로피 인코딩된 세트는 엔트로피 인코딩되는 VPS 확장물의 다른 신택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에 있을 수도 있다. 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))는 복수의 출력 계층 세트들의 각각에 대해 프로파일, 티어, 및 레벨 신택스 구조들 중 하나를 참조하고, 출력 계층 세트에 대해 참조된 프로파일, 티어, 및 레벨 신택스 구조로부터의 정보에 기초하여 출력 계층 세트들 중 하나의 출력 계층 세트의 비디오 데이터를 디코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 복수의 출력 계층 세트들의 각각에 대해 프로파일, 티어, 및 레벨 신택스 구조들 중 하나를 참조하고, 출력 계층 세트에 대해 참조된 프로파일, 티어, 및 레벨 신택스 구조로부터의 정보에 기초하여 출력 계층 세트들 중 하나의 출력 계층 세트의 비디오 데이터를 인코딩할 수도 있다. 비디오 디코더 (30)는 복수의 출력 계층 세트들의 각각에 대해 프로파일, 티어, 및 레벨 신택스 구조들 중 하나를 참조하고, 출력 계층 세트에 대해 참조된 프로파일, 티어, 및 레벨 신택스 구조로부터의 정보에 기초하여 출력 계층 세트들 중 하나의 출력 계층 세트의 비디오 데이터를 디코딩할 수도 있다.

[0058] 몇몇 예들에서, 비디오 코더는 0과 동일한 nuh_layer_id를 갖는 시퀀스 파라미터 세트 (SPS)를 또한 송신 또는 수신할 수도 있는데, SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 신택스 구조를 포함한다. nuh_layer_id는 특정 계층을 나타내는데 사용될 수도 있다. 계층이, nuh_layer_id의 특정 값을 모든 VLC NAL 단위들이 갖는 비디오 코딩 계층 (VCL) 네트워크 추상화 계층 (NAL) 단위들과, 연관된 비-VCL NAL 단위들의 세트, 또는 계층적 관계를 갖는 신택스적 구조들의 세트 중 하나의 신택스적 구조일 수도 있다. nuh_layer_id는 하나의 계층만을 포함하는 비트스트림에서 0과 동일할 것이다. nuh_layer_id의 다른 값들이 다수의 계층들을 포함하는 비트스트림들에서 나타날 수도 있다. 일 예에서, NAL 단위가, 추종할 데이터 유형의 표시 (indication) 와 에뮬레이션 방지 바이트들로 필요한대로 점재된 (interspersed) RBSP (raw byte sequence payload) 형태로 당해 데이터를 포함한 바이트들을 포함하는 신택스 구조이다. 일 예에서, RBSP가, NAL 단위에 캡슐화된 그리고 신택스 엘리먼트들과 뒤따르는 RBSP 정지 비트 및 영 개 이상의 0과 동일한 후속 비트들을 포함하는 데이터 비트들의 스트링 형태를 갖거나 또는 비어 있는 정수 수의 바이트들을 포함하는 신택스 구조이다.

[0059] 비디오 인코더 (20)는 0과 동일한 nuh_layer_id를 갖는 SPS를 송신할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 신택스 구조를 포함한다. 비디오 디코더 (30)는 0과 동일한 nuh_layer_id를 갖는 SPS를 수신할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 신택스 구조를 포함한다.

[0060] 몇몇 예들에서, 0보다 더 큰 nuh_layer_id를 갖는 계층이 SPS를 참조하는 경우, 비디오 디코더는 SPS의 프로파일, 티어, 및 레벨 신택스 구조를 사용하여 0보다 더 큰 nuh_layer_id를 갖는 계층의 비디오 데이터를 디코딩하지 않을 수도 있다. 몇몇 예들은 프로파일, 티어, 및 레벨 신택스 구조들을 출력 계층 세트들 중 개별 세트들에 링크할 수도 있다. 따라서, 하나 이상의 특정 프로파일, 티어, 및 레벨 신택스 구조들은 하나 이상의 특정 프로파일, 티어, 및 레벨 신택스 구조들이 하나 이상의 특정 출력 계층 세트들을 인코딩 및/또는 디코딩하는데 사용될 수도 있도록 하나 이상의 특정 출력 계층 세트들에 연결될 수도 있다. 링킹은 프로파일, 티어, 및 레벨 신택스 구조들에 대한 인덱스를 나타내는 신택스 엘리먼트에 기초하여 프로파일, 티어 및 레벨 신택스 구조들의 각각이 링크되는 출력 계층 세트들의 각각을 식별하는 것을 포함할 수도 있다. 비디오 데이터의 디코딩은 개별 출력 계층 세트들에 링크된 프로파일, 티어, 및 레벨 신택스 구조들로부터의 정보에 기초하여 출력 계층 세트들 중 하나 이상의 출력 계층 세트의 비디오 데이터를 디코딩하는 것을 포함할 수도 있다.

[0061] 다수의 계층들을 포함하는 일 예에서, 모든 계층들의 비트스트림은 다수의 준수 (compliant) 비트스트림들로, 예컨대, 각각의 계층 세트마다 하나의 비트스트림으로 분리될 수도 있다. 이를 다수의 준수 비트스트림들의 각각은 계층 세트라고 지칭될 수도 있다. 계층 세트는 특정 계층에 대한 참조와 당해 특정 계층이 디코딩을

위해 의존하는 임의의 참조 계층들을 포함할 수도 있다. 이는 특정 계층이 의존하는 참조 계층들이 있다고 가정한다. 따라서, 계층 세트가 특정 계층에 연관된 NAL 단위들과 그 특정 계층을 디코딩하는데 요구된 임의의 참조 계층들에 대한 NAL 단위들을 포함할 수도 있는 준수 비트스트림이다.

[0062] 출력 계층 세트가 타겟 출력 계층들의 리스트가 특정되는 계층 세트이다. 계층 세트의 경우, 타겟 출력 계층들의 리스트는 특정되지 않는다. 출력 계층 세트는 디스플레이를 위해 출력될 것이 의도되는 하나 이상의 특정 계층들을 위한 것일 수도 있다. 몇몇 예들에서, 하나를 초과하는 출력 계층 세트는 하나의 계층 세트를 위해 시그널링될 수도 있다.

[0063] 몇몇 예들에서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는, 해당되는 경우, 1과 동일할 때, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일할 때, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신 또는 수신할 수도 있다. 예들의 경우, 비디오 인코더 (20) 는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신할 수도 있다. 비디오 디코더 (30) 는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신할 수도 있다.

[0064] 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 1과 동일한 출력 계층 플래그 [i][j]에 대응하는 계층들을 타겟 출력 계층들의 연관된 리스트로서 특정함으로써 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 또한 생성할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 인코딩할 수도 있다. 비디오 디코더 (30) 는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 디코딩할 수도 있다. 도 1의 도시된 예에서, 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 비디오 데이터의 계층들을 포함하는 다층 비디오 데이터를 코딩할 수도 있다.

[0065] 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 송신 또는 수신할 수도 있다. 이는 계층 의존성 정보가, 예컨대, MANE와 같은 엔트로피 코더를 포함하지 않는 디바이스들에 의해 인코딩 또는 디코딩되는 것을 허용할 수 있다. 덧붙여, 그것은 계층 의존성 정보가 일찍 코딩되는 것을 또한 허용할 수도 있다. 예를 들어, 계층 의존성 정보는 디코딩 프로세스에서 더 일찍 디코딩될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 송신할 수도 있다. 비디오 디코더 (30) 는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 수신할 수도 있다.

[0066] 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 인코딩할 수도 있다. 비디오 디코더 (30) 는 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 디코딩할 수도 있다. 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타낸다.

[0067] 몇몇 예들에서, 계층 의존성 정보는 0과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 I를 갖는 계층에 대한 직접 참조 계층이 아님을 특정하고, 1과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층일 수도 있음을 특정하는 direct_dependency_flag[i][j]를 포함한다.

[0068] 본원에서 설명되는 시스템들, 방법들, 및 디바이스들 중 하나의 예는 VPS 확장 선택스에서 시그널링될 profile_tier_level() 선택스 구조들의 세트를 제공한다. 몇몇 예들에서, profile_tier_level() 선택스 구조들의 세트는 엔트로피 코딩 없이 액세스 가능한, 즉, VPS 확장 선택스에서 엔트로피 코딩 엘리먼트들보다 앞선 VPS 확장 선택스 내의 포지션에서 VPS 확장 선택스로 시그널링된다.

[0069] 몇몇 예들은 VPS 확장 선택스의 엔트로피 코딩된 엘리먼트들 앞에 프로파일, 티어, 및 레벨 선택스 구조들을 제공한다. 이들 선택스 구조들은 출력 계층 세트들에 링크될 수 있다. 몇몇 예들에서, 선택스 엘리먼트

vps_num_layer_sets_minus1 더하기 1은 VPS에 의해 특정되는 계층 세트들의 수를 특정한다. 계층 세트가 한 계층과 그 계층이 의존하는 임의의 (영 이상의) 다른 계층들을 포함하는 세트일 수도 있다. 예를 들어, 스케일러를 비디오 코딩에서의 계층 2에 대한 계층 세트가 계층 0, 계층 1 및 계층 2를 포함할 수도 있다. 각각의 계층 세트에 대해, 하나 이상의 출력 계층 세트들이 특정될 수도 있다. 각각의 출력 계층 세트는 프로파일, 레벨 및 티어를 갖도록 지정된다.

[0070] 본 개시물의 하나의 예에서, VPS 신택스의 부분이 표 1에서 다음과 같을 수도 있다:

표 1 - VPS 신택스의 일 예의 부분

video_parameter_set_rbsp() {	기술자
vps_video_parameter_set_id	u(4)
vps_reserved_three_2bits	u(2)
vps_max_layers_minus1	u(6)
vps_max_sub_layers_minus1	u(3)
vps_temporal_id_nesting_flag	u(1)
vps_extension_offset //vps_reserved_0xffff_16bits	u(16)
profile_tier_level(1, vps_max_sub_layers_minus1)	
vps_sub_layer_ordering_info_present_flag	u(1)
for(i = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1); i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering_minus1[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase_plus1[i]	ue(v)
}	

[0072]

vps_max_layer_id	u(6)
vps_num_layer_sets_minus1	ue(v)
for(i = 1; i <= vps_num_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layer_id; j++)	
layer_id_included_flag[i][j]	u(1)
vps_timing_info_present_flag	u(1)
if(vps_timing_info_present_flag) {	
vps_num_units_in_tick	u(32)
vps_time_scale	u(32)
vps_poc_proportional_to_timing_flag	u(1)
if(vps_poc_proportional_to_timing_flag)	
vps_num_ticks_poc_diff_one_minus1	ue(v)
vps_num_hrd_parameters	ue(v)

[0073]

[0074] 위의 표 1의 예에서 도시된 바와 같이, vps_num_layer_sets_minus1은 VPS에서, ue(v) 기술자에 의해 나타내어진 바와 같이, 엔트로피 코딩되고, VPS에서 vps_num_layer_sets_minus1 앞에 다른 엔트로피 코딩된 신택스 엘리먼트들이 있다. 몇몇 예들에서, 엔트로피 디코딩이 바람직하지 않을 수도 있다. 이는 잘라이음 (splicing) 또는 스트림 적응 (예컨대, 시간적 레이트, 품질, 공간적 적응)을 수행할 수도 있는 중간 네트워크 디바이스들에 대해 특히 진실일 수도 있다. 디코더 (예컨대, 비디오 디코더 (30))가 엔트로피 디코더를 일 반적으로 가질 것이다. 많은 경우들에서, 그러나, 중간 네트워크 디바이스가 엔트로피 디코더를 갖지 않는 것이 바람직할 수도 있다. 이는 중간 디바이스를 단순화시키며, 이는 비용 및 소비 전력을 일부 경우들에서 감소시킬 수도 있다. 덧붙여서, 어느 경우에나, 엔트로피 디코딩을 수행하는 일 없이 프로파일, 티어, 및 레벨 정보를 빠르게 액세스하는 것이 바람직할 수도 있다.

[0075]

프로파일, 레벨 및 티어 선택스 구조들 (profile_tier_level() 선택스)의 엔트로피 디코딩을 피하기 위해, 몇몇 예들에서, 본 개시물은 VPS 확장 선택스에서의 임의의 엔트로피 코딩된 엘리먼트들 앞에, 그리고 파싱 엔트로피 코딩된 vps_num_layer_sets_minus1 선택스 엘리먼트 없이 액세스 가능한 프로파일, 레벨 및 티어 선택스 구조들을 제시하는 것을 제안한다. 몇몇 경우들에서, 확장본들은 비디오 코딩 표준들, 이를테면 HEVC 비디오 코딩 표준들을 위해 개발될 수도 있다. 이들 확장본들은 비디오 코딩 표준들에 의해 제공되지 않거나 또는 요구되지 않는 부가적인 기능을 제공할 수도 있다. 확장 선택스가 비디오 코딩 표준에 대한 이들 확장본들 중 하나를 위한 선택스이다. 예를 들어, VPS 확장 선택스는 HEVC 비디오 코딩 표준들에 대한 확장을 위해 사용되는 메시지들의 선택스를 포함할 수도 있다. 이전에, 이것들은 각각의 계층이 액세스됨에 따라 액세스된 듯하다. 지금, 본원에서 설명되는 개념들 중 하나 이상의 개념을 구현하는 시스템, 방법, 또는 디바이스가 그것들 모두를 미리 액세스한 다음, 그것들을 나중에 링크할 수도 있다.

[0076]

표 2 - VPS 확장본에서의 예의 선택스

vps_num_profile_tier_level_minus1	u(6)
for(i = 1; i <= vps_num_profile_tier_level_minus1; i++) {	
vps_profile_present_flag[i]	u(1)
if(!vps_profile_present_flag[i])	
profile_ref_minus1[i]	u(6)
profile_tier_level(vps_profile_present_flag[i],	
vps_max_sub_layers_minus1)	
}	

[0077]

표 2에서 사용되는 바와 같이, vps_num_profile_tier_level_minus1 더하기 1은 VPS에서의 profile_tier_level() 선택스 구조들의 수를 특정한다. 1과 동일한 vps_profile_present_flag[i]는 프로파일 및 티어 정보가 i-번째 profile_tier_level() 선택스 구조에 존재함을 특정한다. 0과 동일한 vps_profile_present_flag[1sIdx]는 프로파일 및 티어 정보가 i-번째 profile_tier_level() 선택스 구조에 존재하지 않고 i-번째 profile_tier_level() 선택스 구조에 대해 유추될 수도 있다는 것을 특정한다.

[0079]

선택스 엘리먼트 profile_ref_minus1[i]는 i-번째 profile_tier_level() 선택스 구조에 대한 프로파일 및 티어 정보가 (profile_ref_minus1[i] + 1)-번째 profile_tier_level() 선택스 구조에 대한 프로파일 및 티어 정보와 동일한 것으로 유추될 수도 있다는 것을 특정한다. profile_ref_minus1[i] + 1의 값은 i 미만일 수도 있다.

[0080]

표 2에 예시된 바와 같이, 위의 선택스에서, 디코더 또는 중간 네트워크 디바이스는 엔트로피 코딩이 (VPS 확장 선택스에서 나중에) 요구되는 선택스의 엘리먼트들을 파싱하기 전에 그것들을 액세스하기 위해 profile_tier_level() 선택스 구조들의 세트를 통해 루프형 실행을 한다. 이를 선택스 구조들은 대응하는 계층 세트들에 다른 루프 프로세스로 또한 링크될 수도 있다. 특히, profile_level_tier_idx[i]는 i-번째 출력 계층 세트에 적용되는 profile_tier_level() 선택스 구조의, VPS에서의 profile_tier_level() 선택스 구조들의 세트로의 인덱스를 특정한다.

[0081]

다른 예의 양태의 경우, 본원에서 설명되는 하나 이상의 예들을 구현하는 시스템들, 방법들, 및 디바이스들은 계층 0 (기본 계층) 위의 계층들에 대한 계층 세트의 인덱스를 나타내기 위해 output_layer_set_idx_minus1[i]를 사용할 수도 있다.

[0082]

output_layer_set_idx_minus1[i] 더하기 1은 i-번째 출력 계층 세트에 대한 계층 세트의 인덱스 (1sIdx)를 특정한다. 일 예에서, output_layer_set_idx_minus1[i]의 값은 0 내지 vps_num_layer_sets_minus1 - 1의 범위에 있을 수도 있다. output_layer_set_idx_minus1[i] 선택스 엘리먼트의 길이는 Ceil(Log2(vps_num_layer_sets_minus1)) 비트이다.

[0083]

본원에서 설명되는 시스템들, 방법들, 및 디바이스들의 다른 예가 비디오 데이터의 계층들을 포함하는 다층 비디오 데이터의 코딩을 제공한다. 이를 시스템들, 방법들, 및 디바이스들은 엔트로피 인코딩되는 VPS 확장본

의 선택스 엘리먼트들보다 앞선 VPS 확장본 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 송신하는 것 또는 수신하는 것을 포함하는 동작들을 수행하도록 구성될 수도 있다. 덧붙여, 시스템들, 방법들, 및 디바이스들은 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 디코딩할 수도 있다. 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타낸다.

[0084] 하나의 예에 따르면, 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30) 가 본 개시물의 기법들에 따라 하나의 계층 세트에 대해 하나를 초과하는 출력 계층 세트를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 하나의 계층 세트에 대해 하나를 초과하는 출력 계층 세트를 인코딩할 수도 있다. 인코딩된 출력 계층 세트들은 출력 인터페이스 (22) 에 의해 링크 (16) 를 통해 또는 저장 디바이스 (34) 를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28) 에 의해 수신될 수도 있다. 인코딩된 출력 계층 세트들은 비디오 디코더 (30) 에 의해 디코딩될 수도 있다.

[0085] 다른 예에 따르면, 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30) 가, 잠재적으로는 본 개시물의 기법들에 따라 엔트로피 디코딩 없이 액세스가능한 방식으로, VPS에서의 모든 프로파일, 티어, 및/또는 레벨 정보를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 VPS에서의 모든 프로파일, 티어, 레벨 정보를, 잠재적으로는 그 정보가 엔트로피 디코딩 없이 잠재적으로 액세스가능한 방식으로, 즉, 프로파일, 티어, 및/또는 레벨 정보의 엔트로피 인코딩 없이 인코딩할 수도 있다. VPS에서의 인코딩된 프로파일, 티어, 및/또는 레벨 정보는 출력 인터페이스 (22) 에 의해 링크 (16) 를 통해 또는 저장 디바이스 (34) 를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28) 에 의해 수신될 수도 있다. VPS에서의 인코딩된 프로파일, 티어, 및/또는 레벨 정보는 비디오 디코더 (30) 에 의해 디코딩될 수도 있는데, 이 비디오 디코더는 엔트로피 디코더, 이를테면 MANE를 포함하지 않는 디바이스일 수도 있다. 몇몇 예들은 VPS에서의 프로파일, 티어, 레벨 정보에 대해 감소된 또는 제한된 엔트로피 인코딩을 사용할 수도 있다. 예를 들어, 아마도 VPS에서의 일부 프로파일, 티어, 레벨 정보는 엔트로피 인코딩되는 반면 VPS에서의 다른 프로파일, 티어, 레벨 정보는 엔트로피 인코딩되지 않는다.

[0086] 다른 예에 따르면, 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 VPS에서의 계층 의존성 정보를 그 계층 의존성 정보가 본 개시물의 기법들에 따라 엔트로피 디코딩 없이 액세스가능하도록 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 VPS에서의 계층 의존성 정보를 그 계층 의존성 정보가 엔트로피 디코딩 없이 액세스가능하도록, 즉, 계층 의존성 정보의 엔트로피 인코딩 없이 인코딩할 수도 있다. VPS에서의 인코딩된 계층 의존성 정보는 출력 인터페이스 (22) 에 의해 링크 (16) 를 통해 또는 저장 디바이스 (34) 를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28) 에 의해 수신될 수도 있다. VPS에서의 인코딩된 계층 의존성 정보는 비디오 디코더 (30) 에 의해 디코딩될 수도 있는데, 이 비디오 디코더는 엔트로피 디코더, 이를테면 MANE를 포함하지 않는 디바이스일 수도 있다.

[0087] 다른 예에 따르면, 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 VPS에서의 표현 포맷을, 잠재적으로는 그 표현 포맷이 본 개시물의 기법들에 따라 엔트로피 디코딩 없이 액세스될 수 있고, 각각의 계층이 특정 표현 포맷과 연관될 수도 있는 방식으로 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 VPS에서의 표현 포맷을, 잠재적으로는 그 표현 포맷이 엔트로피 디코딩 없이 액세스가능한 방식으로, 즉, VPS에서의 표현 포맷의 엔트로피 인코딩 없이 인코딩할 수도 있다. VPS에서의 인코딩된 표현 포맷은 출력 인터페이스 (22) 에 의해 링크 (16) 를 통해 또는 저장 디바이스 (34) 를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28) 에 의해 수신될 수도 있다. VPS에서의 인코딩된 표현 포맷은 비디오 디코더 (30) 에 의해 디코딩될 수도 있는데, 이 비디오 디코더는 엔트로피 디코더, 이를테면 MANE를 포함하지 않는 디바이스일 수도 있다. 몇몇 예들은 VPS에서의 프로파일, 티어, 레벨 정보에 대해 감소된 또는 제한된 엔트로피 디코딩을 사용할 수도 있다. 예를 들어, 아마도 VPS에서의 일부 프로파일, 티어, 레벨 정보는 엔트로피 디코딩되는 반면 VPS에서의 다른 프로파일, 티어, 레벨 정보는 엔트로피 디코딩되지 않는다 (예컨대, VPS에서의 일부 프로파일, 티어, 레벨 정보가 엔트로피 인코딩되었지만 VPS에서의 다른 프로파일, 티어, 레벨 정보는 엔트로피 인코딩되지 않았기 때문임).

[0088] 다른 예에 따르면, 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 본 개시물의 기법들에 따라 VPS에서의 계층마다 시각적 신호 정보 (video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs) 를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 VPS에서의 계층마다 시각적 신호 정보를 인코딩할 수도 있다. 인코딩된 시각적 신호 정보는 출력 인터페이스 (22) 에 의해 링크 (16) 를 통해 또는 저장 디바이스 (34) 를 사용하여 시그널링, 송신 또는 그렇지 않으면 전

송되고 입력 인터페이스 (28)에 의해 수신될 수도 있다. VPS에서의 계층마다의 인코딩된 시각적 신호 정보 비디오 디코더 (30)에 의해 디코딩될 수도 있다. 몇몇 예들에서, video_format 파라미터는 코딩될 화상들의 표현의 포맷을 나타낸다. video_full_range_flag 파라미터는 루마 (luma) 및 크로마 (chroma) 신호들의 흑색 레벨 및 범위를 나타낸다. colour_primaries 파라미터는 소스 원색들의 색도 좌표들을 나타낸다. transfer_characteristics 파라미터는 소스 화상의 광-전자 전송 특성 (opto-electronic transfer characteristic)을 나타낸다. matrix_coeffs 파라미터는 녹색, 청색, 및 적색 원색들로부터 루마 및 크로마 신호들을 도출함에 있어서 사용되는 매트릭스 계수들을 기술한다.

[0089] 다른 예에 따르면, SPS가 본 개시물의 기법들에 따라 상이한 공간적 해상도들, 비트 깊이, 또는 컬러 포맷들을 갖는 계층들에 의해 공유될 수도 있다. 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)는, SPS를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 SPS를 인코딩할 수도 있다. 인코딩된 SPS는 출력 인터페이스 (22)에 의해 링크 (16)를 통해 또는 저장 디바이스 (34)를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28)에 의해 수신될 수도 있다. 인코딩된 SPS는 비디오 디코더 (30)에 의해 디코딩될 수도 있다.

[0090] 다른 예에 따르면, 본 개시물의 기법들에 따라 0보다 더 큰 계층 ID를 갖는 SPS들의 VUI에는 타이밍 정보가 제공되지 않는다. 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)는 0보다 더 큰 계층 ID를 갖는 SPS를 코딩할 수도 있는데, 그 SPS는 VUI에서의 타이밍 정보를 포함하지 않는다. 예를 들어, 비디오 인코더 (20)는 SPS를 인코딩할 수도 있다. 인코딩된 SPS는 출력 인터페이스 (22)에 의해 링크 (16)를 통해 또는 저장 디바이스 (34)를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28)에 의해 수신될 수도 있다. 인코딩된 SPS는 비디오 디코더 (30)에 의해 디코딩될 수도 있다.

[0091] 다른 예에 따르면, 타겟 출력 계층들의 명시적 시그널링이 본 개시물의 기법들에 따라 디폴트 출력 계층 세트들에는 제공되지 않는다. 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)가 디폴트 출력 계층 세트들을 코딩할 수도 있는데, 그 디폴트 출력 계층 세트는 디폴트 출력 계층 세트들에 대한 타겟 출력 계층들의 임의의 명시적 시그널링을 포함하지 않는다. 예를 들어, 비디오 인코더 (20)는 디폴트 출력 계층 세트들을 인코딩한다. 인코딩된 디폴트 출력 계층 세트들은 출력 인터페이스 (22)에 의해 링크 (16)를 통해 또는 저장 디바이스 (34)를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28)에 의해 수신될 수도 있다. 인코딩된 디폴트 출력 계층 세트들은 비디오 디코더 (30)에 의해 디코딩될 수도 있다.

[0092] 다른 예에 따르면, 제시될 수도 있는 시간적 서브계층들의 최대 수 (sps_max_sub_layers_minus1) 및 인터 예측이 추가로 제한되는지의 여부 (sps_temporal_id_nesting_flag)의 시그널링이 0보다 더 큰 계층 ID (nuh_layer_id)를 갖는 SPS들에는 없다. 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)가, 0보다 더 큰 계층 ID를 갖는 SPS들에서, SPS를 참조하는 각각의 CVS에서 존재할 수도 있는 시간적 서브-계층들의 최대 수의 코딩 없이 그리고 인터 예측이 SPS를 참조하는 CVS들에 대해 추가로 제한되는지의 여부의 코딩 없이, 비디오 데이터를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는, 0보다 더 큰 계층 ID를 갖는 SPS들에서, SPS를 참조하는 각각의 CVS에서 존재할 수도 있는 시간적 서브-계층들의 최대 수와 인터 예측이 SPS를 참조하는 CVS들에 대해 추가로 제한되는지의 여부의 코딩 없이, 비디오 데이터를 인코딩할 수도 있다. 인코딩된 정보는 출력 인터페이스 (22)에 의해 링크 (16)를 통해 또는 저장 디바이스 (34)를 사용하여 시그널링, 송신 또는 그렇지 않으면 전송되고 입력 인터페이스 (28)에 의해 수신될 수도 있다. 인코딩된 정보는 비디오 디코더 (30)에 의해 디코딩될 수도 있다.

[0093] 다른 예에 따르면, 선택스 엘리먼트 output_layer_set_idx[i]가 본 개시물의 기법들에 따라 output_layer_set_idx_minus1[i]로 변경될 수도 있다. 비디오 코더, 이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)가 출력 계층 세트에 대응하는 계층 세트의 인덱스를 나타내기 위해 output_layer_set_idx_minus1[i]를 코딩할 수도 있다.

[0094] 도 2는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20)를 도시하는 블록도이다. 비디오 인코더 (20)는 비디오 슬라이스들 내의 비디오 블록들의 인트라 코딩 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 공간적 예측에 의존하여, 주어진 비디오 프레임 또는 화상 내의 비디오에서 공간적 리던던시를 감소시키거나 또는 제거한다. 인터 코딩은 비디오 시퀀스의 동일한 계층에서 상이한 계층들에 걸친 인접한 프레임들 또는 화상들 내의 비디오에서 총 간 리던던시 또는 시간적 리던던시를 감소 또는 제거하기 위해 총 간 또는 시간적 예측에 의존한다. 인트라 모드 (I 모드)는 여러 공간 기반 압축 모드들 중

임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양-예측 (B 모드) 과 같은 인터 모드들은 여러 시간 기반 압축 모드들 중 임의의 것을 말할 수도 있다.

[0095] 도 2의 예에서, 비디오 인코더 (20)는 비디오 데이터 메모리 (40), 예측 프로세싱 유닛 (41), 참조 화상 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 프로세싱 유닛 (54), 및 엔트로피 인코딩 프로세싱 유닛 (56)을 구비한다. 예측 프로세싱 유닛 (41)은 모션 추정 프로세싱 유닛 (42), 모션 보상 프로세싱 유닛 (44), 및 인트라 예측 프로세싱 유닛 (46)을 구비한다. 비디오 블록 복원을 위해, 비디오 인코더 (20)는 역 양자화 프로세싱 유닛 (58), 역 변환 프로세싱 유닛 (60), 및 합산기 (62)를 또한 구비한다. 블록화제거 필터 (도 2에서 도시되지 않음) 가 블록 경계들을 필터링하여 복원된 비디오로부터 블록현상 (blockiness) 아티팩트들을 제거하기 위해 또한 포함될 수도 있다. 원한다면, 블록화제거 필터는 통상 합산기 (62)의 출력을 필터링할 것이다. 부가적인 루프 필터들 (인 루프 또는 포스트 루프) 이 또한 블록화제거 필터에 부가적으로 사용될 수도 있다.

[0096] 비디오 데이터 메모리 (40)는 비디오 인코더 (20)의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (40)에 저장된 비디오 데이터는, 예를 들어, 비디오 소스 (18)로부터 획득될 수도 있다. 디코딩된 화상 버퍼 (DPB) 가, 예컨대, 인트라 코딩 또는 인터 코딩 모드들에서 비디오 인코더 (20)에 의해 비디오 데이터를 인코딩함에 있어서의 사용을 위해 참조 비디오 데이터를 저장하는 참조 화상 메모리 (64) 일 수도 있다. 비디오 데이터 메모리 (40)와 DPB (예컨대, 참조 화상 메모리 (64)는 동기식 DRAM (SDRAM)을 포함한 다이나믹 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (40)와 DPB (예컨대, 참조 화상 메모리 (64)는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (101)는 비디오 인코더 (20)의 다른 컴포넌트들과 온-칩, 또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0097] 도 2에 도시된 바와 같이, 비디오 인코더 (20)는 비디오 데이터를 수신하고 예측 프로세싱 유닛 (41)은 그 데이터를 비디오 블록들로 구획화한다. 몇몇 경우들에서, 예측 프로세싱 유닛 (41)은 레이트 왜곡 분석에 기초하여 비디오 데이터를 구획화할 수도 있다. 수신된 비디오 데이터는 위에서 설명된 샘플 포맷들 중 임의의 것에 따라 포맷될 수도 있다. 예를 들어, 비디오 데이터는 4:2:2 샘플 포맷에 따라 포맷될 수도 있다. 구획화는 또한 비디오 데이터를 슬라이스들, 타일들, 또는 다른 큰 단위들로의 구획화, 뿐만 아니라, 예컨대, LCU들 및 CU들의 큐드트리 구조에 따른 비디오 블록 구획화를 포함할 수도 있다.

[0098] 비디오 인코더 (20)는 일반적으로 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 슬라이스는 다수의 비디오 블록들로 (및 타일들이라고 지칭된 비디오 블록들의 세트들로 아마도) 나누어질 수도 있다. 예측 프로세싱 유닛 (41)은 복수의 가능한 코딩 모드들 중 하나, 이를테면 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나를 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨)에 기초하여 현재 비디오 블록에 대해 선택할 수도 있다. 예측 프로세싱 유닛 (41)은 결과적인 인트라 또는 인터 코딩된 블록을 잔차 블록 데이터를 생성하는 합산기 (50) 및 참조 화상으로서 사용하기 위해 인코딩된 블록을 복원하는 합산기 (62)에 제공할 수도 있다.

[0099] 예측 프로세싱 유닛 (41) 내의 인트라 예측 프로세싱 유닛 (46)은 코딩될 현재 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들을 기준으로 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 프로세싱 유닛 (42) 및 모션 보상 프로세싱 유닛 (44)은 하나 이상의 참조 화상들에서의 하나 이상의 예측 블록들을 기준으로 현재 비디오 블록의 인터 예측 코딩을 수행하여 시간적 압축을 제공한다.

[0100] 모션 추정 프로세싱 유닛 (42)은 비디오 시퀀스에 대한 미리 결정된 패턴에 따라 비디오 슬라이스에 대한 인터 예측 모드를 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들 또는 B 슬라이스들이라고 지정할 수도 있다. 모션 추정 프로세싱 유닛 (42)과 모션 보상 프로세싱 유닛 (44)은 고도로 통합될 수도 있지만 개념적 목적들을 위해 별개로 예시된다. 모션 추정 프로세싱 유닛 (42)에 의해 수행되는 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터가, 예를 들어, 참조 화상 내의 예측 블록에 대한 현재 비디오 프레임 또는 화상 내의 비디오 블록의 예측 단위 (PU)의 변위 (displacement)를 나타낼 수도 있다.

[0101] 예측 블록이, 인터 코딩의 경우, 차의 절대값 합 (SAD), 차의 제곱 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 화소 차이의 관점에서 코딩될 비디오 블록의 PU에 밀접하게 매칭된다고 생각되는 블록일 수

도 있다. 대안으로, 예측 블록이, 인트라 코딩의 경우, 하나 이상의 이웃 블록들로부터의 화소 값들에 관해 공간 예측에 기초하여 형성되는 블록일 수도 있다. 몇몇 예들에서, 인트라 예측을 위해, 비디오 인코더 (20)는 참조 화상 메모리 (64)에 저장된 참조 화상들의 부 정수 (sub-integer) 화소 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20)는 참조 화상의 1/4 화소 위치들, 1/8 화소 위치들, 또는 다른 분수 (fractional) 화소 위치들의 값들을 보간할 수도 있다. 그러므로, 모션 추정 프로세싱 유닛 (42)은 풀 (full) 화소 위치들 및 분수 화소 위치들에 대한 모션 검색을 수행하여 분수 화소 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0102] 모션 추정 프로세싱 유닛 (42)은 PU의 위치와 참조 화상의 예측 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 화상은 참조 화상 메모리 (64)에 저장된 하나 이상의 참조 화상들을 각각이 식별하는 제 1 참조 화상 리스트 (리스트 0) 또는 제 2 참조 화상 리스트 (리스트 1)로부터 선택될 수도 있다. 모션 추정 프로세싱 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 프로세싱 유닛 (56)과 모션 보상 프로세싱 유닛 (44)으로 전송한다.

[0103] 모션 보상 프로세싱 유닛 (44)에 의해 수행되는 모션 보상은 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐치하는 것 또는 생성하는 것, 아마도 부 화소 (sub-pixel) 정밀도로 보간들을 수행하는 것을 수반할 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터를 수신 시, 모션 보상 프로세싱 유닛 (44)은 참조 화상 리스트들 중 하나에서 모션 벡터가 가리키는 예측 블록을 찾을 수도 있다.

[0104] 비디오 인코더 (20)는, 코딩되고 있는 현재 비디오 블록의 화소 값들로부터 예측 블록의 화소 값을 감산하여 화소 차이 값을 형성함으로써, 인터 또는 인트라 코딩을 위해, 잔차 비디오 블록을 형성한다. 화소 차이 값을 블록에 대한 잔차 데이터를 형성하고, 루마 및 크로마 차이 성분들 양쪽 모두를 포함할 수도 있다. 합산기 (50)는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 모션 보상 프로세싱 유닛 (44)은 또한 비디오 슬라이스의 비디오 블록들의 디코딩 시에 비디오 디코더 (30)에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스에 연관된 신박스 엘리먼트들을 생성할 수도 있다.

[0105] 인트라 예측 프로세싱 유닛 (46)은, 위에서 설명된 바와 같이, 모션 추정 프로세싱 유닛 (42) 및 모션 보상 프로세싱 유닛 (44)에 의해 수행된 인터 예측에 대한 대안으로서 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 프로세싱 유닛 (46)은 현재 블록을 인코딩하는데 사용하기 위한 인트라 예측 모드를 결정할 수도 있다. 몇몇 예들에서, 인트라 예측 프로세싱 유닛 (46)은 예컨대, 개별 인코딩 과정들 동안에 다양한 인트라 예측 모드들을 사용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 프로세싱 프로세싱 유닛 (46) (또는 일부 예들에서, 모드 선택 프로세싱 유닛)은 테스트된 모드들로부터 사용할 적절한 인트라 예측 모드를 선택할 수도 있다.

[0106] 예를 들어, 인트라 예측 프로세싱 유닛 (46)은 다양한 테스트된 인트라 예측 모드들에 대한 레이트 왜곡 분석을 사용하여 레이트 왜곡 값을 계산하고, 테스트된 모드들 중에서 최상의 레이트 왜곡 특성들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록과 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러)의 양, 뿐만 아니라 인코딩된 블록을 생성하는데 사용된 비트 레이트 (다시 말하면, 비트들의 수)를 결정한다. 인트라 예측 프로세싱 유닛 (46)은 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 계산하여 어떤 인트라 예측 모드가 그 블록에 대한 최상의 레이트 왜곡 값을 나타내는지를 결정할 수도 있다. 레이트 왜곡 분석은 컬러 성분들의 조합에 대해 수행될 수도 있다는 것에 주의해야 한다.

[0107] 어느 경우에나, 블록에 대한 인트라 예측 모드를 선택한 후, 인트라 예측 프로세싱 유닛 (46)은 그 블록에 대한 선택된 인트라 예측 모드를 나타내는 정보를 엔트로피 인코딩 프로세싱 유닛 (56)에 제공할 수도 있다. 엔트로피 인코딩 프로세싱 유닛 (56)은 본 개시물의 기법들에 따라 선택된 인트라 예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20)는 송신되는 비트스트림에 구성 데이터를 포함시킬 수도 있는데, 이 구성 데이터는 복수의 인트라 예측 모드 인덱스 테이블들 및 복수의 수정된 인트라 예측 모드 인덱스 테이블들 (코드워드 매핑 테이블들이라고 또한 지칭됨), 다양한 블록들에 대한 콘텍스트들을 인코딩하는 정의들, 그리고 그 콘텍스트들의 각각에 대한 사용을 위한 가장 가능성 있는 인트라 예측 모드, 인트라 예측 모드 인덱스 테이블, 및 수정된 인트라 예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다. 그 비트스트림은 또한 (또는 대안으로) 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 디코딩 및/또는 플레이백을 위한 나중의 액세스를 위해 저장 디바이스 (34) 상에 저장될 수도 있다.

[0108] 예측 프로세싱 유닛 (41)이 인터 예측 또는 인트라 예측 중 어느 하나를 통해 현재 비디오 블록에 대한 예측

블록을 생성한 후, 비디오 인코더 (20)는 현재 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 잔차 블록에서의 잔차 비디오 데이터는, 하나 이상의 변환 단위들 (TU들)에 포함되고 변환 프로세싱 유닛 (52)에 인가될 수도 있다. 변환 프로세싱 모듈 (52)은 비디오 인코더 (20)의 컴포넌트, 모듈, 프로세서 또는 프로세서들, 또는 기능 유닛을 지칭하고, 변환 및 양자화 프로세스를 위한 기본 데이터 단위인 TU와 혼동되지 않아야 한다는 것에 주의한다. 변환 프로세싱 유닛 (52)은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 사용하여 잔차 비디오 데이터를 잔차 변환 계수들로 변환시킨다. 변환 프로세싱 유닛 (52)은 잔차 비디오 데이터를 화소 도메인으로부터 변환 도메인, 이를테면 주파수 도메인으로 변환할 수도 있다. 변환 프로세싱 유닛 (52)은 결과적인 변환 계수들을 양자화 프로세싱 유닛 (54)으로 전송할 수도 있다.

[0109] 도 2의 도시된 예에서, 비디오 인코더 (20)가 복수의 출력 계층 세트들의 각각에 대해 프로파일, 티어, 및 레벨 선택스 구조들 중 하나를 참조하고, 출력 계층 세트에 대해 참조된 프로파일, 티어, 및 레벨 선택스 구조로부터의 정보에 기초하여 출력 계층 세트들 중 하나의 출력 계층 세트의 비디오 데이터를 인코딩할 수도 있다.

[0110] 비디오 인코더 (20)는 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트를 송신할 수도 있다. 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트는, 중간 네트워킹 디바이스, 잘라이음 엔진, 미디어 인식 네트워크 엘리먼트, 또는 디코더가 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트를 엔트로피 코딩에 대한 필요 없이 파싱할 수 있도록, 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에 있을 수도 있다.

[0111] 몇몇 예들에서, 비디오 인코더 (20)는 0과 동일한 nuh_layer_id를 갖는 SPS를 송신 또는 수신할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 선택스 구조를 포함한다. 예를 들어, 비디오 인코더 (20)는 0과 동일한 nuh_layer_id를 갖는 SPS를 전송할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 선택스 구조를 포함한다.

[0112] 비디오 인코더 (20)는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 인코딩할 수도 있다. 비디오 인코더 (20)는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 또한 송신할 수도 있다.

[0113] 비디오 인코더 (20)는 비디오 데이터의 계층들 중 하나 이상의 계층의 일부 비디오 데이터를 비-엔트로피 인코딩할 수도 있다. 비-엔트로피 인코딩된 비디오 데이터는 계층 의존성 정보를 포함할 수도 있다. 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타낼 수도 있다. 몇몇 예들에서, 계층 의존성 정보는 0과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 I를 갖는 계층에 대한 직접 참조 계층이 아님을 특정하고, 1과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층일 수도 있음을 특정하는 direct_dependency_flag[i][j]를 포함한다.

[0114] 비디오 인코더 (20)는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 송신할 수도 있다.

[0115] 하나의 예에 따르면, 비디오 인코더 (20)는 하나의 계층 세트에 대해 하나를 초과하는 출력 계층 세트를 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해 수행될 수도 있다.

[0116] 다른 예에 따르면, 비디오 인코더 (20)는 VPS에서의 모든 프로파일, 티어, 레벨 정보를, 잠재적으로는 그 정보가 엔트로피 디코딩 없이 액세스 가능한 방식으로, 즉, 프로파일, 티어, 및/또는 레벨 정보의 엔트로피 인코딩 없이 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해, 그러나, 엔트로피 인코딩 없이 수행될 수도 있다. 따라서, 엔트로피 인코딩 프로세싱 유닛은 엔트로피 인코딩에 더하여 다른 유형들의 인코딩을 수행할 수도 있다.

[0117] 다른 예에 따르면, 비디오 인코더 (20)는 VPS에서의 계층 의존성 정보를 그 계층 의존성 정보가 엔트로피 디코딩 없이 액세스 가능하도록, 즉, 계층 의존성 정보의 엔트로피 인코딩 없이 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해, 그러나, 엔트로피 인코딩 없이 수행될 수도 있다. 따라서, 엔트로피 인코딩 프로세싱 유닛은 엔트로피 인코딩에 더하여 다른 유형들의 인코딩을 수행할 수도 있다.

[0118] 다른 예에 따르면, 비디오 인코더 (20)는 VPS에서의 표현 포맷을, 잠재적으로는 그 표현 포맷이 엔트로피 디코딩 없이 액세스 가능한 방식으로, 즉, VPS에서의 인코딩 표현 포맷의 엔트로피 인코딩 없이 인코딩할 수도 있다.

몇몇 예들에서, 각각의 계층은 특정 표현 포맷과 연관될 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해, 그러나, 엔트로피 인코딩 없이 수행될 수도 있다. 따라서, 엔트로피 인코딩 프로세싱 유닛은 엔트로피 인코딩에 더하여 다른 유형들의 인코딩을 수행할 수도 있다. 비디오 인코더 (20)는 VPS에서의 표현 포맷을, 잠재적으로는 그 표현 포맷이 본 개시물의 기법들에 따라 엔트로피 디코딩 없이 액세스될 수 있고, 각각의 계층이 특정 표현 포맷과 연관될 수도 있는 방식으로 인코딩할 수도 있다.

[0119] 다른 예에 따르면, 비디오 인코더 (20)는 본 개시물의 기법들에 따라 VPS에서의 계층마다 시각적 신호 정보 (video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs)를 코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해 수행될 수도 있다.

[0120] 다른 예에 따르면, SPS가 본 개시물의 기법들에 따라 상이한 공간적 해상도들, 비트 깊이, 또는 컬러 포맷들을 갖는 계층들에 의해 공유될 수도 있다. 비디오 인코더 (20)는 SPS를 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해 수행될 수도 있다.

[0121] 다른 예에 따르면, 본 개시물의 기법들에 따라 0보다 더 큰 계층 ID를 갖는 SPS의 VUI에는 타이밍 정보가 제공되지 않는다. 비디오 인코더 (20)는 SPS를 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해 수행될 수도 있다.

[0122] 다른 예에 따르면, 타겟 출력 계층들의 명시적 시그널링이 본 개시물의 기법들에 따라 디폴트 출력 계층 세트들에는 제공되지 않는다. 비디오 인코더 (20)는 디폴트 출력 계층 세트들을 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해 수행될 수도 있다.

[0123] 다른 예에 따르면, 제시될 수도 있는 시간적 서브 계층들의 최대 수 (sps_max_sub_layers_minus1) 및 인터 예측이 추가로 제한되는지의 여부 (sps_temporal_id_nesting_flag)의 시그널링이 nuh_layer_id가 0과 동일한 경우에만 SPS에서 발생하며, 즉 0보다 더 큰 계층 ID를 갖는 SPS들에서 시그널링되지 않는다. 비디오 인코더 (20)는, 0보다 더 큰 계층 ID를 갖는 SPS들에서, SPS를 참조하는 각각의 CVS에서 존재할 수도 있는 시간적 서브-계층들의 최대 수와 인터 예측이 SPS를 참조하는 CVS들에 대해 추가로 제한되는지의 여부의 코딩 없이, 비디오 데이터를 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해 수행될 수도 있다.

[0124] 다른 예에 따르면, 선택스 엘리먼트 output_layer_set_idx[i]가 본 개시물의 기법들에 따라 output_layer_set_idx_minus1[i]로 변경될 수도 있다. 비디오 인코더 (20)는 VPS에 의해 특정된 i-번째 출력 계층 세트에 대응하는 계층 세트의 인덱스를 나타내기 위해 선택스 엘리먼트 output_layer_set_idx_minus1[i]를 인코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (56)에 의해 수행될 수도 있다.

[0125] 도 3은 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더 (30)를 도시하는 블록도이다. 도 3의 예에서, 비디오 디코더 (30)는 비디오 데이터 메모리 (78), 엔트로피 디코딩 프로세싱 유닛 (80), 예측 프로세싱 유닛 (81), 역 양자화 프로세싱 유닛 (86), 역 변환 프로세싱 유닛 (88), 합산기 (90), 및 참조 화상 메모리 (92)를 구비한다. 예측 프로세싱 유닛 (81)은 모션 보상 프로세싱 유닛 (82)과 인트라 예측 프로세싱 유닛 (84)을 구비한다. 비디오 디코더 (30)는, 몇몇 예들에서, 도 2로부터 비디오 인코더 (20)에 관해 설명된 인코딩 과정에 일반적으로 역인 디코딩 과정을 수행할 수도 있다.

[0126] 비디오 데이터 메모리 (78)는, 비디오 디코더 (30)의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (78)에 저장된 비디오 데이터는, 예를 들어, 컴퓨터 판독가능 매체로부터, 예컨대, 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체들에 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (78)는 인코딩된 비디오 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 화상 베퍼 (CPB)를 형성할 수도 있다. 디코딩된 화상 베퍼 (DPB)가, 예컨대, 인트라 코딩 또는 인터 코딩 모드들에서 비디오 디코더 (30)에 의해 비디오 데이터를 디코딩함에 있어서의 사용을 위해 참조 비디오 데이터를 저장하는 참조 화상 메모리 (92) 일 수도 있다. 비디오 데이터 메모리 (78)와 DPB는 동기식 DRAM (SDRAM)을 포함한 다이나믹 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (78)와 DPB는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (78)는 비디오 디코더 (30)의 다른 컴포넌트들과 온-칩,

또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0127] 디코딩 프로세스 동안, 비디오 디코더 (30)는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20)로부터 수신한다. 비디오 디코더 (30)의 엔트로피 디코딩 프로세싱 유닛 (80)은 그 비트스트림을 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 프로세싱 유닛 (80)은 모션 벡터들 및 다른 신택스 엘리먼트들을 예측 프로세싱 유닛 (81)에 포워딩한다. 비디오 디코더 (30)는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.

[0128] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩되는 경우, 예측 프로세싱 유닛 (81)의 인트라 예측 프로세싱 유닛 (84)은 현재 비디오 슬라이스의 비디오 블록을 위한 예측 데이터를 현재 프레임 또는 화상의 이전에 디코딩된 블록들로부터의 시그널링된 인트라 예측 모드 및 데이터에 기초하여 생성할 수도 있다. 비디오 프레임이 인터 코딩된 (즉, B 또는 P) 슬라이스로서 코딩되는 경우, 예측 프로세싱 유닛 (81)의 모션 보상 프로세싱 유닛 (82)은 엔트로피 디코딩 프로세싱 유닛 (80)으로부터 수신된 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여, 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 그 예측 블록들은은 참조 화상 리스트들 중 하나의 참조 화상 리스트 내의 참조 화상을 중 하나의 참조 화상으로부터 생성될 수도 있다. 비디오 디코더 (30)는 참조 화상 메모리 (92)에 저장된 참조 화상들에 기초하여 디플트 구축 기법들을 이용하여, 참조 프레임 리스트들, 리스트 0 및 리스트 1을 구축할 수도 있다.

[0129] 모션 보상 프로세싱 유닛 (82)은 모션 벡터들 및 다른 신택스 엘리먼트들을 과정함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 프로세싱 유닛 (82)은 수신된 신택스 엘리먼트들의 일부를 이용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예컨대, 복수의 예측 모드들 중 하나), 인터 예측 슬라이스 유형 (예컨대, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 참조 화상 리스트들 중 하나 이상에 대한 구축 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터 코딩된 비디오 블록에 대한 인터 예측 스테이터스, 및 현재 비디오 슬라이스 내의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0130] 모션 보상 프로세싱 유닛 (82)은 보간 필터들에 기초하여 보간을 또한 수행할 수도 있다. 모션 보상 프로세싱 유닛 (82)은 비디오 블록들의 인코딩 동안에 비디오 인코더 (20)에 의해 사용된 것과 같은 보간 필터들을 사용하여 참조 블록들의 부 정수 화소들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 프로세싱 유닛 (82)은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 사용된 보간 필터들을 결정하고 그 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.

[0131] 역 양자화 프로세싱 유닛 (86)은 비트스트림으로 제공되고 엔트로피 디코딩 프로세싱 유닛 (80)에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉, 탈양자화 (dequantize) 한다. 역 양자화 프로세스는 양자화 정도와, 마찬가지로 적용되어야 할 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20)에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다.

[0132] 역 변환 프로세싱 유닛 (88)은 변환 계수들을 수신하고 잔차 블록들을 화소 도메인에서 생성하기 위하여 역 변환, 예컨대, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 변환 계수들에 적용한다. 몇몇 예들에서, 역 변환 프로세싱 유닛 (88)은 변환 단위 구획화 기법들에 기초하여 비디오 인코더에 의해 생성되었던 변환 계수들을 수신할 수도 있다.

[0133] 모션 보상 프로세싱 유닛 (82)이 현재 비디오 블록에 대한 예측 블록을 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 생성한 후, 비디오 디코더 (30)는 역 변환 프로세싱 유닛 (88)으로부터의 잔차 블록들을 모션 보상 프로세싱 유닛 (82)에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (90)는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0134] 원한다면, 블록화제거 필터가 블록현상 아티팩트들을 제거하기 위하여 디코딩된 블록들을 필터링하는데 또한 적용될 수도 있다. 다른 루프 필터들이 (코딩 루프 내 또는 코딩 루프 후 중 어느 하나에서) 화소 전이 (transition)들을 부드럽게 하기 위해, 또는 그렇지 않으면 비디오 품질을 개선하기 위해 또한 사용될 수도 있다. 주어진 프레임 또는 화상에서의 디코딩된 비디오 블록들은 그 다음에 참조 화상 메모리 (92)에 저장되는데, 그 참조 화상 메모리는 후속 모션 보상을 위해 사용되는 참조 화상들을 저장한다. 참조 화상 메모리 (92)는 디스플레이 디바이스, 이를테면 도 1의 디스플레이 디바이스 (32) 상의 나중의 프레젠테이션을 위해 디

코딩된 비디오를 또한 저장한다.

[0135] 도 3의 도시된 예에서 비디오 디코더 (30) 는 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트를 수신할 수도 있다.

[0136] 비디오 디코더 (30) 는 복수의 출력 계층 세트들의 각각에 대해 프로파일, 티어, 및 레벨 선택스 구조들 중 하나를 참조하고, 출력 계층 세트에 대해 참조된 프로파일, 티어, 및 레벨 선택스 구조로부터의 정보에 기초하여 출력 계층 세트들 중 하나의 출력 계층 세트의 비디오 데이터를 디코딩할 수도 있다.

[0137] 몇몇 예들에서, 비디오 코더는 0과 동일한 nuh_layer_id를 갖는 SPS를 송신 또는 수신할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 선택스 구조를 포함한다. 비디오 디코더 (30) 는 0과 동일한 nuh_layer_id를 갖는 SPS를 수신할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 선택스 구조를 포함한다.

[0138] 몇몇 예들에서, 0보다 더 큰 nuh_layer_id를 갖는 계층이 SPS를 참조하는 경우, 비디오 디코더 (30) 는 SPS의 프로파일, 티어, 및 레벨 선택스 구조를 사용하여 0보다 더 큰 nuh_layer_id를 갖는 계층의 비디오 데이터를 디코딩하지 않을 수도 있다.

[0139] 비디오 디코더 (30) 는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신할 수도 있다.

[0140] 비디오 디코더 (30) 는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 또한 생성할 수도 있다.

[0141] 도 3의 도시된 예에서, 비디오 디코더 (30) 는 비디오 데이터의 계층들을 포함하는 다층 비디오 데이터를 디코딩할 수도 있다.

[0142] 비디오 디코더 (30) 는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 수신할 수도 있다.

[0143] 비디오 디코더 (30) 는 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 인코딩할 수도 있다. 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층일 수도 있는지의 여부를 나타낸다.

[0144] 몇몇 예들에서, 계층 의존성 정보는 0과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층이 아님을 특정하고, 1과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층일 수도 있음을 특정하는 direct_dependency_flag[i][j]를 포함한다.

[0145] 몇몇 예들에서, 본 개시물은, 출력 계층 세트들에 대한 프로파일, 티어, 및 레벨 정보의 시그널링, 출력 계층 세트들의 시그널링, 및 계층 의존성의 시그널링에 대한 변경들을 포함하여, HEVC 확장본들에 대한 VPS 및 SPS의 향상된 설계들을 위한 방법들을 설명한다. 몇몇 예들에서, 본 개시물은 상이한 카테고리들의 개개의 제어를 이용한 SPS 파라미터들의 효율적인 시그널링뿐만 아니라 세션 협상을 위해 VPS (확장본)에서의 표현 포맷 (공간적 해상도, 컬러 포맷, 및 비트 깊이)에 대한 정보의 시그널링에 대한 변경들을 포함하여, HEVC 확장본들에 대한 VPS 및 SPS의 향상된 설계들을 위한 방법들을 또한 설명한다.

[0146] 몇몇 예들에서, 현재 VPS 및 SPS 설계들은 적어도 다음의 문제들과 연관될 수도 있다: (1) 선택스 엘리먼트 vps_num_layer_sets_minus1이 ue(v) 코딩되고 그것 앞에 또한 다른 ue(v) 코딩된 선택스 엘리먼트들이 있기 때문에, 계층 세트들에 대한 프로파일, 티어, 및 레벨의 현재의 액세싱은 엔트로피 디코딩을 필요로 하며; (2) 마찬가지로, 계층 의존성 정보는 엔트로피 디코딩 없이 액세스가능하지 않고; 및 (3) 출력 계층 세트의 시그널링이 문제가 있고 효율적이지 않은데, 이는, (a) 0과 동일한 output_layer_set_idx[i]를 위해 사용된 선택스 엘리먼트 layer_id_included_flag[][]는 계층 세트 0에 대한 것인 반면, 계층 세트 0의 경우 layer_id_included_flag[][]는 HEVC 버전 1 (즉, HEVC WD10)에서 정의되지 않고, (b) 계층 세트 0에 대한 타겟 출력 계층이 어쨌든 계층 0 (기본 계층) 임이 분명하고, (c) 현재 기껏해야 하나의 출력 계층 세트가 각각의 계층 세트에 대해 시그널링될 수 있고 그리고 (d) 현재, 0부터 VPS에서의 가능한 최대 계층 ID까지의 계층 ID들의 루프를 사용하는 선택스는 복잡하기 때문이다.

[0147] 멀티뷰 시나리오들에서, 그 경우는 종종 하나의 계층 세트가 상이한 타겟 출력 계층들의 다수의 세트들과 연관 된다는 것이다. 따라서, 선택스 엘리먼트 output_layer_flag[lvIdx][j]는 output_layer_flag[i][j]로 변경

되고, 관련된 시맨틱스는 그에 따라 변경되고, 위의 항목 (d)에서 언급했듯이, 현재, 0부터 VPS에서의 가능한 최대 계층 ID까지의 계층 ID들의 루프를 사용하는 선택스는 복잡하다. 계층 세트에서의 계층들의 루프를 사용하는 것이, 당해 계층이 확실히 타겟 출력 계층일 때의 계층 세트에서의 최고 계층을 제외하면, 훨씬 더 간단 할 것이다.

[0148] 하나의 예에 따르면, 비디오 디코더 (30)는 하나의 계층 세트에 대해 하나를 초과하는 출력 계층 세트를 디코딩할 수도 있다. 몇몇 예들에서, 디코딩은 엔트로피 디코딩 프로세싱 유닛 (80)에 의해 수행될 수도 있다.

[0149] 다른 예에 따르면, 비디오 디코더 (30)는, 잠재적으로는 그 정보가 엔트로피 디코딩하는 일 없이 액세스 가능한 방식으로, VPS에서의 모든 프로파일, 티어, 레벨 정보를 디코딩할 수도 있다. 몇몇 예들에서, 디코딩은 엔트로피 디코딩 프로세싱 유닛 (80)에 의해, 그러나, 엔트로피 디코딩 없이 수행될 수도 있다. 따라서, 디코딩은 엔트로피 디코더를 갖지 않는 디바이스들에 의해 또한 수행될 수도 있다.

[0150] 다른 예에 따르면, 비디오 디코더 (30)는 VPS에서의 계층 의존성 정보를 그 정보가 엔트로피 디코딩 없이 액세스 가능하도록 디코딩할 수도 있다. 몇몇 예들에서, 디코딩은 엔트로피 디코딩 프로세싱 유닛 (80)에 의해, 그러나, 엔트로피 인코딩 없이 수행될 수도 있다. 따라서, 디코딩은 엔트로피 디코더를 갖지 않는 디바이스들에 의해 또한 수행될 수도 있다.

[0151] 다른 예에 따르면, 비디오 디코더 (30)는 VPS에서의 표현 포맷을, 잠재적으로는 그 표현 포맷이 엔트로피 디코딩 없이 액세스 가능한 방식으로, 즉, VPS에서의 표현 포맷의 엔트로피 인코딩 없이 디코딩할 수도 있다. 몇몇 예들에서, 디코딩은 엔트로피 디코딩 프로세싱 유닛 (80)에 의해, 그러나, 엔트로피 디코딩 없이 수행될 수도 있다. 따라서, 디코딩은 엔트로피 디코더를 갖지 않는 디바이스들에 의해 또한 수행될 수도 있다. 비디오 디코더 (30)는, 잠재적으로는 본 개시물의 기법들에 따라 엔트로피 디코딩 없이 VPS에서의 표현 포맷을 디코딩할 수도 있고, 각각의 계층은 특정 표현 포맷에 연관될 수도 있다.

[0152] 다른 예에 따르면, 비디오 디코더 (30)는 본 개시물의 기법들에 따라 VPS에서의 계층마다 시작적 신호 정보 (video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs)를 디코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (80)에 의해 수행될 수도 있다. 몇몇 예들에서, video_format 파라미터는 코딩될 화상들의 표현의 포맷을 나타낸다. video_full_range_flag 파라미터는 루마 및 크로마 신호들의 흑색 레벨 및 범위를 나타낸다. colour_primaries 파라미터는 소스 원색들의 색도 좌표들을 나타낸다. transfer_characteristics 파라미터는 소스 화상의 광-전자 전송 특성을 나타낸다. matrix_coeffs 파라미터는 녹색, 청색, 및 적색 원색들로부터 루마 및 크로마 신호들을 도출함에 있어서 사용되는 매트릭스 계수들을 기술한다.

[0153] 다른 예에 따르면, SPS가 본 개시물의 기법들에 따라 상이한 공간적 해상도들, 비트 깊이, 또는 컬러 포맷들을 갖는 계층들에 의해 공유될 수도 있다. 비디오 디코더 (30)는 SPS를 디코딩할 수도 있다. 몇몇 예들에서, 인코딩은 엔트로피 인코딩 프로세싱 유닛 (80)에 의해 수행될 수도 있다.

[0154] 다른 예에 따르면, 본 개시물의 기법들에 따라 0보다 더 큰 계층 ID를 갖는 SPS들의 VUI에는 타이밍 정보가 제공되지 않는다. 비디오 디코더 (30)는 SPS를 디코딩할 수도 있다. 몇몇 예들에서, 디코딩은 엔트로피 디코딩 프로세싱 유닛 (80)에 의해 수행될 수도 있다.

[0155] 다른 예에 따르면, 타겟 출력 계층들의 명시적 시그널링이 본 개시물의 기법들에 따라 디폴트 출력 계층 세트들에는 제공되지 않는다. 비디오 디코더 (30)는 디폴트 출력 계층 세트들을 디코딩할 수도 있다. 몇몇 예들에서, 디코딩은 엔트로피 디코딩 프로세싱 유닛 (80)에 의해 수행될 수도 있다.

[0156] 다른 예에 따르면, 제시될 수도 있는 시간적 서브-계층들의 최대 수 (sps_max_sub_layers_minus1) 및 인터 예측이 추가로 제한되는지의 여부 (sps_temporal_id_nesting_flag)의 시그널링은 nuh_layer_id가 0과 동일한 경우에만 SPS에서 발생한다. 비디오 디코더 (30)는, 0보다 더 큰 계층 ID를 갖는 SPS들에서, SPS를 참조하는 각각의 CVS에서 존재할 수도 있는 시간적 서브-계층들의 최대 수와 인터 예측이 SPS를 참조하는 CVS들에 대해 추가로 제한되는지의 여부의 코딩 없이, 비디오 데이터를 디코딩할 수도 있다. 몇몇 예들에서, 디코딩은 엔트로피 디코딩 프로세싱 유닛 (80)에 의해 수행될 수도 있다.

[0157] 다른 예에 따르면, 선택스 엘리먼트 output_layer_set_idx[i]가 본 개시물의 기법들에 따라 output_layer_set_idx_minus1[i]로 변경될 수도 있다. 비디오 디코더 (30)는 VPS에 의해 특정된 i-번째 출력 계층 세트에 대응하는 계층 세트의 인덱스를 나타내기 위해 선택스 엘리먼트

output_layer_set_idx_minus1[i]를 디코딩할 수도 있다.

[0158] 아래의 예들에서, 꺾쇠괄호로 묶은 레이블들 (예컨대, [START EXAMPLE A] 및 [END EXAMPLE A]) 이 나타낸 예에 관련된 텍스트를 나타내는데 사용될 것이다. 변경들이 2013년 4월 2일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1008-v1.zip으로부터 다운로드가능한 SHVC, 또는 2013년 4월 2일 현재, http://phenix.itt-sudparis.eu/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1004-v4.zip으로부터 다운로드가능한 MV-HEVC 표준의 현재 버전에 대해 나타내어질 수도 있다.

[0159] 계층 세트들에 대한 프로파일, 티어, 및 레벨 정보의 시그널링, 출력 계층 세트들의 시그널링, 및 계층 의존성의 시그널링에 대한 여러 변경들은, 본 개시물의 양태들에 따라 본원에 포함되는 예 A를 나타내기 위해 이 제안에서 사용될 [START EXAMPLE A] 및 [END EXAMPLE A]에 의해 나타내어진다.

[0160] 선택스 엘리먼트 vps_num_layer_sets_minus1이 ue(v) 코딩 (즉, 엔트로피 인코딩) 되고 그것 전에 다른 ue(v) 코딩된 선택스 엘리먼트들이 또한 있다는 사실로 인해, 현재, 계층 세트들에 대한 프로파일, 티어, 및 레벨의 액세스는 엔트로피 디코딩을 필요로 한다. 엔트로피 디코딩 없이 액세스가능하게 될 이 정보의 경우, profile_tier_level() 선택스 구조들의 세트가 엔트로피 디코딩 없이 액세스가능한 포지션에서 시그널링될 수도 있다. profile_tier_level() 선택스 구조들의 세트는 그 다음에 출력 계층 세트들에 대한 링킹을 위해 참조될 수도 있다. 동일한 계층 세트의 다수의 출력 계층 세트들은 상이한 DPB 사이즈들을 필요로 할 수도 있고, 그래서 상이한 레벨들에 부합할 수 있다.

[0161] 몇몇 예들에서, 선택스 엘리먼트 output_layer_set_idx[i]는, 아래에서 설명되는 바와 같이, output_layer_set_idx_minus1[i]로 변경될 수도 있다. 0과 동일한 nuh_layer_id를 갖는 SPS가 0보다 더 큰 nuh_layer_id를 갖는 계층에 의해 참조되는 경우, SPS에서의 profile_tier_level() 선택스 구조는 당해 계층에 대해 적용되지 않는다. 위와 유사한 이유로, 계층 의존성의 시그널링은 엔트로피 디코딩 없이 액세스가능하게 되도록 이동된다. 선택스 엘리먼트 output_layer_set_idx[i]의 output_layer_set_idx_minus1[i]로의 변경은 다음의 이유들 때문이다. 0과 동일한 output_layer_set_idx[i]를 위해 사용된 선택스 엘리먼트 layer_id_included_flag[][]는 계층 세트 0에 대한 것인 반면, 계층 세트 0의 경우 layer_id_included_flag[][]는 정의되지 않는다. 계층 세트 0에 대한 타겟 출력 계층이 계층 0 (기본 계층)임이 분명하다.

[0162] 따라서, 위에서 알 수 있듯이, 각각의 계층은 SPS를 참조하는 것을 필요로 한다. 기준에, 상이한 값들의 공간적 해상도들, 비트 깊이들, 또는 컬러 포맷들로 된 임의의 두 개의 계층들은 이들 표현 포맷 파라미터들이 SPS에서 시그널링될 때 두 개의 상이한 SPS들을 참조해야 한다. 그러나, 0과 동일한 nuh_layer_id를 갖는 것들을 제외한 모든 SPS들에 대한 이들 파라미터들이 VPS로 이동되는 경우, 및 0보다 더 큰 nuh_layer_id를 갖는 계층에 의해 참조되는, 0과 동일한 nuh_layer_id를 갖는 SPS에서의 표현 포맷 파라미터들이 무시된다고 특정되는 경우, 상이한 값들의 공간적 해상도들, 비트 깊이들, 또는 컬러 포맷들을 갖는 계층들이 동일한 SPS들을 참조하는 것이 가능하다. 다르게 말하면, 본 개시물의 몇몇 실시형태들에 따르면, 상이한 값들의 공간적 해상도들, 비트 깊이들, 또는 컬러 포맷들을 갖는 계층들은, 다른 SPS인 한, 동일한 SPS들을 공유할 수 있다.

[0163] 본원에서 설명되는 몇몇 예들은 출력 계층 세트 시그널링에 대한 선택스 및 시맨틱스를 변경할 수도 있다. 몇몇 경우들에서, 다음의 이슈들 중 하나 이상을 해결할 수도 있는데, 첫째, 현재 기껏해야 하나의 출력 계층 세트가 각각의 계층 세트를 위해 시그널링될 수 있다. 멀티뷰 시나리오들에서, 그 경우는 종종 하나의 계층 세트가 상이한 타겟 출력 계층들의 다수의 세트들과 연관된다는 것이다. 따라서, 선택스 엘리먼트 output_layer_flag[lsIdx][j]는 output_layer_flag[i][j]로 변경될 수도 있고, 관련된 시맨틱스는 그에 따라 변경된다. 따라서, 하나를 초과하는 출력 계층 세트는 하나의 계층 세트를 위해 시그널링될 수도 있다. 둘째, 현재, 선택스는 0부터 VPS에서의 가능한 최대 계층 ID까지의 계층 ID들의 루프를 사용하는데, 이 루프는 복잡하다. 오히려, 계층 세트에서의 계층들의 루프를 사용하는 것이, 당해 계층이 확실히 타겟 출력 계층일 때의 계층 세트에서의 최고 계층을 제외하면, 훨씬 더 간단할 것이다.

[0164] 세션 협상에 대한 VPS (확장본)에서의 표현 포맷 (공간적 해상도, 컬러 포맷, 및 비트 깊이에 대한 정보의 시그널링을 위해, 그 변경들이 예 B를 나타내기 위해 이 제안에서 사용될 꺾쇠괄호로 묶은 레이블들 [START EXAMPLE B] 및 [END EXAMPLE B]를 사용하여 나타내어진다. 그 표현 포맷은 기본 계층을 포함하는 각각의 계층에 대해, 이러한 정보의 리스트에 대한 인덱스를 통해 시그널링되고, profile_tier_level() 선택스 구조에 대해 동일한, 0보다 더 큰 nuh_layer_id를 갖는 SPS들에서 시그널링되지 않는다. 결과적으로, 이는 0보다 더 큰 nuh_layer_id를 갖는 상이한 SPS들에서 동일한 표현 포맷 정보를 용장적으로 (redundantly) 전송하는 것을 또한 감소시킨다. 0과 동일한 nuh_layer_id를 갖는 SPS가 0보다 더 큰 nuh_layer_id를 갖는 계층에 의

해 참조되는 경우, 선택스 엘리먼트들 (chroma_format_idc, separate_colour_plane_flag, pic_width_in_luma_samples, pic_height_in_luma_samples, bit_depth_luma_minus8, 및 bit_depth_chroma_minus8) 의 값들은 당해 계층에 적용되지 않는다.

[0165] 상이한 카테고리들의 개개의 제어를 갖는 SPS 파라미터들의 효율적인 시그널링의 경우, SPS들에 포함될 수도 있는 파라미터들은 다음 6 개의 카테고리 (Cat) 들로 분류될 수도 있다: (1) Cat1: 스케일러블 HEVC 비디오 코딩 (SHVC) WD1 및 멀티-뷰 HEVC (MV-HEVC) WD3와 위에서 이미 다루어진 프로파일-티어-레벨 정보, (2) Cat2: 위에서 다루어진 표현 포맷 정보, (3) Cat3: 서브-계층 정보 (sps_max_sub_layers_minus1, sps_temporal_id_nesting_flag, sps_max_dec_pic_buffering_minus1[i], sps_max_num_reordered_pics[i], 및 sps_max_latency_increase_plus1[i]) (이들 변경들은, Cat3라고 또한 지칭되는 예 C를 나타내기 위해 이 제안에서 사용될 꺾쇠괄호로 묶인 레이블들 [START EXAMPLE C] 및 [END EXAMPLE C]에 의해 나타내어짐), (4) Cat4: 단기 참조 화상 세트 (RPS) 후보들 (이에 대한 변경들은 Cat4라고 또한 지칭되는 예 D를 나타내기 위해 이 제안에서 사용될 꺾쇠괄호로 묶인 레이블들 [START EXAMPLE D] 및 [END EXAMPLE D]에 의해 나타내어짐), (5) Cat5: 비디오 유용성 정보 (VUI) 파라미터들 (이에 대한 변경들은 Cat 5라고 또한 지칭되는 예 E를 나타내기 위해 이 제안에서 사용될 꺾쇠괄호로 묶인 레이블들 [START EXAMPLE E] 및 [END EXAMPLE E]에 의해 나타내어짐), 및 (6) Cat6: 다른 SPS 파라미터들, 이에 대한 변경들 및 공통 변경들은 Cat 6라고 또한 지칭되는 예 F를 나타내기 위해 이 제안에서 사용될 꺾쇠괄호로 묶인 레이블들 [START EXAMPLE F] 및 [END EXAMPLE F]에 의해 나타내어짐.

[0166] 향상 계층들 또는 향상 계층들을 수반하는 계층 세트들의 경우, 카테고리들 (3 내지 6) 의 정보는 액티브 VPS로부터 상속되거나 또는 SPS에서 직접 시그널링될 수도 있는 반면, 카테고리들 (1 및 2) 의 정보는 세션 협상에 대한 중요성으로 인해 VPS에서 시그널링되어야 한다. 0보다 더 큰 nuh_layer_id를 갖는 SPS에서, 최소 세 개까지만, VPS ID, SPS ID 및 플래그 (카테고리 3 내지 카테고리 6 (Cat3 내지 Cat6) 의 데이터가 SPS에 존재하는지 또는 액티브 VPS로부터 상속되는지를 나타내기 위한 것) 를 시그널링하기 위한 선택스 엘리먼트들이 존재한다. 이러한 더미 SPS들의 역할은 액티브 VPS ID를 화상 파라미터 세트들 (PPS) 에게 그리고 그 다음에 비디오 코딩 계층 (VCL) 네트워크 추상화 계층 (NAL) 단위들에게 간접적으로 전달하는 것뿐이다.

[0167] 아래의 변경들은 MV-HEVC WD3 및 SHVC WD1에 기초하고 상이한 컬러 코드들에 의해 식별되며, 즉, 위에서 언급했듯이 상이한 컬러들로 강조표시되고, 제거는 취소선 (strikethrough) 또는 이중 꺾쇠괄호들 [[]] 로 묶인 텍스트들로서 도시된다. 언급되지 않은 부분들은 MV-HEVC WD3 및 SHVC WD1에서와 동일하다.

[0168] 아래의 표 3에서 예시된 것은 원시 바이트 시퀀스 패이로드 (RBSP) 선택스 및 시맨틱스를 위한 비디오 파라미터 세트 (VPS) 에 대한 예의 선택스이다. 이는 SHVC WD1 및 MV-HEVC WD3에서와 동일하다. 그 선택스는 편 이를 위해 아래에서 (표 3에서) 복사된다.

[0169] 표 3 - RBSP 신택스 및 시맨틱스를 위한 VPS에 대한 예의 신택스

video_parameter_set_rbsp() {	기술자
vps_video_parameter_set_id	u(4)
vps_reserved_three_2bits	u(2)
vps_max_layers_minus1	u(6)
vps_max_sub_layers_minus1	u(3)
vps_temporal_id_nesting_flag	u(1)
vps_extension_offset //vps_reserved_0xffff_16bits	u(16)
profile_tier_level(1, vps_max_sub_layers_minus1)	
vps_sub_layer_ordering_info_present_flag	u(1)
for(i = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1);	
i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering_minus1[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase_plus1[i]	ue(v)
}	
vps_max_layer_id	u(6)
vps_num_layer_sets_minus1	ue(v)
for(i = 1; i <= vps_num_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layer_id; j++)	
layer_id_included_flag[i][j]	u(1)
vps_timing_info_present_flag	u(1)
if(vps_timing_info_present_flag) {	
vps_num_units_in_tick	u(32)
vps_time_scale	u(32)
vps_poc_proportional_to_timing_flag	u(1)

[0170]

if(vps_poc_proportional_to_timing_flag)	
vps_num_ticks_poc_diff_one_minus1	ue(v)
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
hrd_layer_set_idx[i]	ue(v)
if(i > 0)	
cprms_present_flag[i]	u(1)
hrd_parameters(cprms_present_flag[i],	
vps_max_sub_layers_minus1)	
}	
}	
vps_extension_flag	u(1)
if(vps_extension_flag) {	
vps_extension()	
vps_extension2_flag	u(1)
if(vps_extension2_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

[0171]

표 4는 비디오 파라미터 세트 확장본 및 시맨틱스의 일 예를 도시한다.

[0172]

[0173]

표 4 - 비디오 파라미터 세트 확장 신택스 및 시맨틱스

기술자	
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
vps_nuh_layer_id_present_flag	u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
[START EXAMPLE A] vps_num_profile_tier_level_minus1	u(6)
for(i = 1; i <= vps_num_profile_tier_level_minus1; i++) {	
vps_profile_present_flag[i]	u(1)
if(!vps_profile_present_flag[i])	
profile_ref_minus1[i]	u(6)
profile_tier_level(vps_profile_present_flag[i],	
vps_max_sub_layers_minus1)	
}	

[0174]

for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1) [END EXAMPLE E A]
[START EXAMPLE B] vps_num_rep_fromats	u(4)
for(i = 0; i < vps_num_rep_fromats; i++)	
rep_format()	
for(i = 1; i <= vps_max_layers_minus1; i++)	
if(vps_num_rep_fromats > 1)	
vps_rep_format_idx[i]	u(4) [END EXAMPLE E B]
[START EXAMPLE C] for(i = 1; i <= vps_max_layers_minus1; i++)	
{	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
} // 위까지의 고정 길이 코딩된 정보 [END EXAMPLE C]	
[START EXAMPLE A] multiple_output_layer_sets_in_layer_set_flag	u(1)
if(!multiple_output_layer_sets_in_layer_set_flag)	
numOutputLayerSets = vps_num_layer_sets_minus1 + 1	
else {	
num_output_layer_sets_minus1	ue(v)
numOutputLayerSets = num_output_layer_sets_minus1 + 1	
}	

for(i = 1; i < numOutputLayerSets; i++) {	
if(i > vps_num_layer_sets_minus1) {	
output_layer_set_idx_minus1[i]	u(v)
lsIdx = output_layer_set_idx_minus1[i] + 1	
for(j = 0 ; j < NumLayersInIdList[lsIdx] - 1; j++)	
output_layer_flag[i][j]	u(1)
}	
profile_level_tier_idx[i]	u(v)
}	
for(lsIdx = 1; lsIdx <= vps_num_layer_sets_minus1; lsIdx ++) {	
vps_profile_present_flag[lsIdx]	u(1)
if(!vps_profile_present_flag[lsIdx])	
profile_layer_set_ref_minus1[lsIdx]	ue(v)
profile_tier_level(vps_profile_present_flag[lsIdx],	
vps_max_sub_layers_minus1)	
}	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0 ; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1) [END EXAMPLE EA]

[START EXAMPLE F]	vps_num_other_sps_params	ue(v)
for(i = 0; i < vps_num_other_sps_params; i++)		
other_sps_parameters()	[END EXAMPLE F]	
[START EXAMPLE D]	vps_num_st_rps_candidates	ue(v)
for(i = 0; i < vps_num_st_rps_candidates; i++)		
short_term_rps_candidates((i == 0) ? 0 : 1)	[END EXAMPLE D]	
[START EXAMPLE E]	vps_num_vui_params	ue(v)
for(i = 0; i < vps_num_vui_params; i++)		
vui_parameters(0)	[END EXAMPLE E]	
[START EXAMPLE F]	for(i = 1; i <= vps_max_layers_minus1; i++)	
{		
if(vps_num_other_sps_params > 1)		
vps_other_sps_params_idx[i]	u(v) [END EXAMPL E F]	
[START EXAMPLE D]	if(vps_num_st_rps_candidates > 1)	
vps_st_rps_idx[i]	u(v) [END EXAMPL E D]	
[START EXAMPLE E]	if(vps_num_vui_params > 1)	
vps_vui_params_idx[i]	u(v)	
}	[END EXAMPLE E]	

[0177]

[START EXAMPLE C]	for(i = 1; i < numOutputLayerSets;	
i++) {		
sub_layer_vps_buf_info_predict_flag[i]	u(1)	
if(!sub_layer_vps_buf_info_predict_flag[i]) {		
sub_layer_vps_buf_info_present_flag[i]	u(1)	
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 :		
MaxSubLayers[i] - 1);		
j <= MaxSubLayers[i] - 1;		
j++)		
max_vps_dec_pic_buffering_minus1[i][j]	ue(v)	
}		
}		
for(i = 1; i <= vps_max_layers_minus1; i++) {		
if(max_sub_layers_vps_predict_flag[i])		
sub_layer_vps_ordering_info_predict_flag[i]	u(1)	
if(!sub_layer_ordering_info_predict_flag[i]) {		
sub_layer_vps_ordering_info_present_flag[i]	u(1)	
for(j = (sub_layer_vps_ordering_info_present_flag[i] ? 0 :		
max_sub_layers_vps_minus1[i]); j		
<= max_sub_layers_vps_minus1[i]; j++)		
max_vps_num_reorder_pics[i][j]	ue(v)	
max_vps_latency_increase_plus1[i][j]	ue(v)	
}		
}		
}		
[END EXAMPLE C]		
}		

[0178]

[START EXAMPLE A]

[0179]

파라미터 vps_num_profile_tier_level_minus1 더하기 1은 VPS에서의 profile_tier_level() 선택스 구조들의 수를 특정한다. 1과 동일한 vps_profile_present_flag[i]는 프로파일 및 티어 정보가 i-번째 profile_tier_level() 선택스 구조에 존재함을 특정한다. 0과 동일한 vps_profile_present_flag[lsIdx]는 프로파일 및 티어 정보가 i-번째 profile_tier_level() 선택스 구조에 존재하지 않고 i-번째 profile_tier_level() 선택스 구조에 대해 유추될 수도 있다는 것을 특정한다.

[0180]

파라미터 profile_ref_minus1[i]는 i-번째 profile_tier_level() 선택스 구조에 대한 프로파일 및 티어 정보가 (profile_ref_minus1[i] + 1)-번째 profile_tier_level() 선택스 구조에 대한 프로파일 및 티어 정보와 동일한 것으로 유추될 수도 있다는 것을 특정한다. profile_ref_minus1[i] + 1의 값은 i 미만일 수도 있다.

[0181]

0과 동일한 파라미터 direct_dependency_flag[i][j]는 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층이 아님을 특정한다. 1과 동일한 파라미터 direct_dependency_flag[i][j]는 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층일 수도 있음을 특정한다. direct_dependency_flag[i][j]가 0 내지 vps_max_layers_minus1의 범위의 i 및 j에 대해 존재하지 않는 경우, 그것은 0과 동일한 것으로 유추될 수도 있다.

[0182]

변수들 NumDirectRefLayers[i] 및 RefLayerId[i][j]는 다음과 같이 도출될 수도 있다:

[0184] `for(i = 1; i <= vps_max_layers_minus1; i++)`

[0185] `for(j = 0, NumDirectRefLayers[i] = 0; j < i; j++)`

[0186] `if(direct_dependency_flag[i][j] == 1)`

[0187] `RefLayerId[i][NumDirectRefLayers[i]++] = layer_id_in_nuh[j]` **[END EXAMPLE A]**

[0188] **[START EXAMPLE B]** 파라미터 `vps_num_rep_formats`은 VPS에서의 다음의 `rep_format()` 신택스 구조들의 수를 특정한다. `vps_rep_format_idx[i]`는 `layer_id_in_nuh[i]`와 동일한 `nuh_layer_id`를 갖는 계층에 적용하는 `rep_format()` 신택스 구조의, VPS에서의 `rep_format()` 신택스 구조들의 세트로의 인덱스를 특정한다. `i`가 0과 동일하거나 또는 `vps_num_rep_formats`이 1과 동일한 경우, `vps_rep_format_idx[i]`의 값은 0과 동일한 것으로 유추될 수도 있다. `vps_rep_format_idx[i]`의 값은 0 내지 `vps_num_rep_formats - 1`의 범위에 있을 수도 있다. **[END EXAMPLE B]**

[0189] **[START EXAMPLE C]** 1과 동일한 `max_sub_layers_vps_predict_flag[i]`는 `max_sub_layers_vps_minus1[i] ~ max_sub_layers_vps_minus1[i - 1]`과 동일한 것으로 유추될 수도 있다는 것과 `sub_layer_vps_ordering_predict_flag[i]`가 존재한다는 것을 특정한다. 0과 동일한 `max_sub_layers_vps_predict_flag[i]`는 `max_sub_layers_vps_minus1[i]`이 명시적으로 시그널링된다는 것을 특정한다. `max_sub_layers_vps_predict_flag[0]`의 값은 0과 동일한 것으로 유추될 수도 있다.

[0190] 파라미터 `max_sub_layers_vps_minus1[i]`는 SPS 신택스 엘리먼트 `sps_max_sub_layers_minus1`의 유추를 위해 사용된다. `max_sub_layers_vps_predict_flag[i]`가 1과 동일한 경우, `max_sub_layers_vps_minus1[i]`는 `max_sub_layers_vps_minus1[i - 1]`과 동일한 것으로 유추될 수도 있다. `max_sub_layers_vps_minus1[i]`의 값은 `vps_max_sub_layers_minus1`과 동일한 것으로 유추될 수도 있다. **[END EXAMPLE C]** **[START EXAMPLE A]** 1과 동일한 `multiple_output_layer_sets_in_layer_set_flag`는 하나를 초과하는 출력 계층 세트가 각각의 계층 세트에 대해 VPS에 의해 특정될 수도 있다. 0과 동일한 `multiple_output_layer_sets_in_layer_set_flag`는 하나의 출력 계층 세트만이 각각의 계층 세트에 대해 VPS에 의해 특정되며, 최고 계층이 유일한 타겟 출력 계층임을 특정한다. 따라서, 0과 동일한 `multiple_output_layer_sets_in_layer_set_flag`가 단지 하나의 출력 계층 세트가 각각의 계층 세트에 대해 VPS에 의해 특정되고 최고 계층이 유일한 타겟 출력 계층임을 특정하기 때문에, 디폴트 출력 계층 세트들에 대한 타겟 출력 계층들의 명시적 시그널링은 필요하지 않다. 예를 들어, 각각의 계층 세트에 대해 하나의 출력 계층 세트만이 있기 때문에 어떤 계층들이 출력될 것인지를 시그널링할 필요가 없다.

[0191] 파라미터 `num_output_layer_sets_minus1` 더하기 1은 VPS에 의해 특정되는 출력 계층 세트들의 수를 특정한다. `num_output_layer_sets_minus1`의 값은 0 내지 1023의 범위에 있을 수도 있다. **[END EXAMPLE A]**

[0192] 대안으로, `num_output_layer_sets_minus1` 대신, `vps_num_layers_sets_minus1 + 1`에 더하여 출력 계층 세트들의 수를 나타내는 `num_addn_output_layer_sets`가 시그널링될 수도 있다.

[0193] **[START EXAMPLE A]** 파라미터 `output_layer_set_idx_minus1[i]` 더하기 1은 `i`-번째 출력 계층 세트에 대한 계층 세트의 인덱스를 특정한다. `output_layer_set_idx_minus1[i]`의 값은 0 내지 `vps_num_layer_sets_minus1 - 1`의 범위에 있을 수도 있다. `output_layer_set_idx_minus1[i]` 신택스 엘리먼트의 길이는 $\text{Ceil}(\text{Log2}(\text{vps_num_layer_sets_minus1}))$ 개 비트이다.

[0194] 1과 동일한 `output_layer_flag[i][j]`는 계층 세트에서의 `j`-번째 계층이 `i`-번째 출력 계층 세트의 타겟 출력 계층임을 특정한다. 0과 동일한 `output_layer_flag[i][j]`는 계층 세트에서의 `j`-번째 계층이 `i`-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정한다.

[0195] `output_layer_flag[i][NumLayersInIdList[lsIdx] - 1]`의 값은 1과 동일한 것으로 유추될 수도 있는데, 여기서 `lsIdx`는 `output_layer_set_idx_minus1[i] + 1`과 동일하다. **[END EXAMPLE A]**

[0196] 대안으로, `output_layer_flag[i][NumLayersInIdList[lsIdx] - 1]`의 값은 1과 동일한 것으로 유추될 수도 있는데, 여기서 `lsIdx`는 `output_layer_set_idx_minus1[i] + 1`과 동일하고 `i`는 0 내지 `vps_num_layer_sets_minus1`의 범위에 있고, `output_layer_flag[i][j]`의 값은 0 내지 `vps_num_layer_sets_minus1`의 범위에서의 `i` 및 0 내지 `NumLayerIdInList[lsIdx] - 2`의 범위에서의 `j`에 대해 0과 동일한 것으로 유추될 수도 있고, `lsIdx`는 `output_layer_set_idx_minus1[i] + 1`과 동일하다.

[0197] **[START EXAMPLE A]** 파라미터 profile_level_tier_idx[i]는 i-번째 출력 계층 세트에 적용되는 profile_tier_level() 신택스 구조의, VPS에서의 profile_tier_level() 신택스 구조들의 세트로의 인덱스를 특정한다. profile_level_tier_idx[i] 신택스 엘리먼트의 길이는 Ceil(Log2(vps_num_profile_tier_level_minus1 + 1)) 개 비트일 수도 있다. profile_level_tier_idx[0]의 값은 0과 동일한 것으로 유추될 수도 있다. profile_level_tier_idx[i]의 값은 0 내지 vps_num_profile_tier_level_minus1의 범위에 있을 수도 있다. **[END EXAMPLE A]**

[0198] **[START EXAMPLE F]** 파라미터 vps_num_other_sps_params는 VPS에서의 다음의 other_sps_parameters() 신택스 구조들의 수를 특정한다. vps_num_other_sps_params의 값은 0 내지 15의 범위에 있을 수도 있다. **[END EXAMPLE F]**

[0199] **[START EXAMPLE D]** 파라미터 vps_num_st_rps_candidates는 VPS에서의 다음의 short_term_rps_candidates() 신택스 구조들의 수를 특정한다. vps_num_st_rps_candidates의 값들은 0 내지 15의 범위에 있을 수도 있다. **[END EXAMPLE D]**

[0200] **[START EXAMPLE E]** 파라미터 vps_num_vui_params는 VPS에서의 다음의 vui_parameters() 신택스 구조들의 수를 특정한다. vps_num_vui_params의 값은 0 내지 15의 범위에 있을 수도 있다. **[END EXAMPLE E]**

[0201] **[START EXAMPLE F]** vps_other_sps_params_idx[i]는 layer_id_in_nuh[i]과 동일한 nuh_layer_id를 갖는 계층에 적용되는 other_sps_parameters() 신택스 구조의, VPS에서의 other_sps_parameters() 신택스 구조들의 세트로의 인덱스를 특정한다. vps_other_sps_params_idx[i] 신택스 엘리먼트의 길이는 Ceil(Log2(vps_num_other_sps_params)) 개 비트일 수도 있다. vps_num_other_sps_params가 1과 동일한 경우, vps_other_sps_params_idx[i]의 값은 0과 동일한 것으로 유추될 수도 있다. vps_other_sps_params_idx[i]의 값은 0 내지 vps_num_other_sps_params - 1의 범위에 있을 수도 있다. **[END EXAMPLE F]**

[0202] **[START EXAMPLE D]** 파라미터 vps_st_rps_idx[i]는 layer_id_in_nuh[i]과 동일한 nuh_layer_id를 갖는 계층에 적용되는 short_term_rps_candidates() 신택스 구조의, VPS에서의 short_term_rps_candidates() 신택스 구조들의 세트로의 인덱스를 특정한다. the vps_st_rps_idx[i] 신택스 엘리먼트의 길이는 Ceil(Log2(vps_num_st_rps_candidates)) 개 비트일 수도 있다. vps_num_st_rps_candidates가 1과 동일한 경우, vps_st_rps_idx[i]의 값은 0과 동일한 것으로 유추될 수도 있다. vps_st_rps_idx[i]의 값은 0 내지 vps_num_st_rps_candidates - 1의 범위에 있을 수도 있다. **[END EXAMPLE D]**

[0203] **[START EXAMPLE E]** 파라미터 vps_vui_params_idx[i]는 layer_id_in_nuh[i]와 동일한 nuh_layer_id를 갖는 계층에 적용되는 vui_parameters() 신택스 구조의, VPS에서의 vui_parameters() 신택스 구조들의 세트로의 인덱스를 특정한다. vps_vui_params_idx[i] 신택스 엘리먼트의 길이는 Ceil(Log2(vps_num_vui_params)) 개 비트일 수도 있다. vps_num_vui_params가 1과 동일한 경우, vps_vui_params_idx[i]의 값은 0과 동일한 것으로 유추될 수도 있다. vps_vui_params_idx[i]의 값은 0 내지 vps_num_vui_params - 1의 범위에 있을 수도 있다. **[END EXAMPLE E]**

[0204] **[START EXAMPLE C]** 0 내지 num_output_layer_sets - 1의 범위에서의 setId에 대한 변수 MaxSubLayers[setId]는 다음과 같이 도출된다:

```

[0205] for(setId = 0; setId < num_output_layer_sets; setId++){
[0206]     lsIdx = output_layer_set_idx_minus1[setId] + 1 //계층 세트 인덱스
[0207]     highestLayerId =
[0208]         LayerSetLayerIdList[lsIdx][NumLayersInIdList[lsIdx] - 1]
[0209]     MaxSubLayers[setId] =
[0210]         (max_sub_layers_vps_minus1[highestLayerId] + 1)
[0211] }
[0212] 1과 동일한 파라미터 max_sub_layers_vps_predict_flag[i]는 max_sub_layers_vps_minus1[i]가 max_sub_layers_vps_minus1[i] - 1]과 동일한 것으로 유추될 수도 있다는 것과

```

sub_layer_vps_ordering_predict_flag[i]가 존재한다는 것을 특정한다. 0과 동일한 max_sub_layers_vps_predict_flag[i]는 max_sub_layers_vps_minus1[i]가 명시적으로 시그널링된다는 것을 특정한다. max_sub_layers_vps_predict_flag[0]의 값은 0과 동일한 것으로 유추될 수도 있다.

[0213] 1과 동일한 파라미터 sub_layer_vps_buf_info_predict_flag[i]는 max_vps_dec_pic_buffering_minus1[i][j]가 j의 각각의 값에 대해 max_vps_dec_pic_buffering_minus1[i - 1][j]와 동일한 것으로 유추될 수도 있다는 것을 특정한다. 0과 동일한 sub_layer_vps_buf_info_predict_flag[i]는 j의 적어도 하나의 값에 대한 max_vps_dec_pic_buffering_minus1[i][j]가 명시적으로 시그널링된다는 것을 특정한다.

[0214] 1과 동일한 파라미터 sub_layer_vps_buf_info_present_flag[i]는 max_vps_dec_pic_buffering_minus1[i][j]가 MaxSubLayers[i] 서브-계층들에 대해 존재한다는 것을 특정한다. 0과 동일한 sub_layer_vps_buf_info_present_flag[i]는 max_vps_dec_pic_buffering_minus1[i][MaxSubLayers[i] - 1]의 값들이 모든 서브-계층들에 적용된다는 것을 특정한다.

[0215] 파라미터 max_vps_dec_pic_buffering_minus1[i][j]는 SPS 신택스 엘리먼트들 sps_max_dec_pic_buffering_minus1[j]의 값들의 추론을 위해 사용된다. max_vps_dec_pic_buffering_minus1[i][j]가 0 내지 MaxSubLayers[i] - 2의 범위의 i에 대해 존재하지 않는 경우, 0과 동일한 sub_layer_vps_buf_info_present_flag[i]로 인해, 그것은 max_vps_dec_pic_buffering_minus1[i][MaxSubLayers[i] - 1]와 동일한 것으로 유추될 수도 있다.

[0216] j의 각각의 값에 대한 max_vps_dec_pic_buffering_minus1[0][j]의 값은 vps_max_dec_pic_buffering_minus1[j]와 동일한 것으로 유추될 수도 있다.

[0217] 1과 동일한 파라미터 sub_layer_vps_ordering_info_predict_flag[i]는, 신택스 엘리먼트들 (sub_layer_vps_ordering_info_present_flag[i], max_vps_num_reorder_pics[i][j], 및 max_vps_latency_increase_plus1[i][j]) 이 sub_layer_vps_ordering_info_present_flag[i - 1], max_vps_num_reorder_pics[i - 1][j], 및 max_vps_latency_increase_plus1[i - 1][j]와 각각 동일한 것으로 유추된다는 것을 특정한다. 0과 동일한 sub_layer_vps_ordering_info_predict_flag[i]는 신택스 엘리먼트들 (sub_layer_vps_ordering_info_present_flag[i], max_vps_num_reorder_pics[i][j], 및 max_vps_latency_increase_plus1[i][j]) 이 명시적으로 시그널링된다는 것을 나타낸다. 존재하지 않는 경우, sub_layer_vps_ordering_info_predict_flag[i]의 값은 0과 동일하게 설정된다.

[0218] 1과 동일한 파라미터 sub_layer_vps_ordering_info_present_flag[i]는 max_vps_num_reorder_pics[i][j] 및 max_vps_latency_increase_plus1[i][j]가 max_sub_layers_vps_minus1 + 1 서브-계층들에 대해 존재한다는 것을 특정한다. 0과 동일한 sub_layer_vps_ordering_info_present_flag[i]는 max_vps_num_reorder_pics[i][vps_max_sub_layers_minus1] 및 max_vps_latency_increase_plus1[i][Max_sub_layers_vps_minus1]의 값들이 모든 서브-계층들에 적용된다는 것을 특정한다.

[0219] 파라미터 max_vps_num_reorder_pics[i][j]는 SPS 신택스 엘리먼트 sps_max_num_reorder_pics[j]의 값들의 추론을 위해 사용된다. max_vps_num_reorder_pics[i][j]가 0 내지 max_sub_layers_vps_minus1[i] - 1의 범위에서의 i에 대해 존재하지 않는 경우, 0과 동일한 sub_layer_vps_ordering_info_present_flag[i]로 인해, 그것은 max_vps_num_reorder_pics[i][Max_sub_layers_vps_minus1[i]]와 동일한 것으로 유추될 수도 있다.

[0220] 파라미터 max_vps_latency_increase_plus1[i][j]가 SPS 신택스 엘리먼트들 sps_max_latency_increase_plus1[j]의 값들의 추론을 위해 사용된다. max_vps_latency_increase_plus1[i][j]가 0 내지 max_sub_layers_vps_minus1[i] - 1의 범위에서의 i에 대해 존재하는 않는 경우, 0과 동일한 sub_layer_vps_ordering_info_present_flag[i]로 인해, 그것은 max_vps_latency_increase_plus1[i][Max_sub_layers_vps_minus1[i]]와 동일한 것으로 유추될 수도 있다.

[END EXAMPLE C]

[START EXAMPLE B]

표 5는 대표적인 포맷 및 시맨틱스의 일 예를 도시한다.

[0223]

표 5 - 표현 포맷 선택스 및 시멘틱스

rep_format() {	기술자
chroma_format_vps_idc	u(2)
if(chroma_format_vps_idc == 3)	
separate_colour_plane_vps_flag	u(1)
pic_width_vps_in_luma_samples	u(16)
pic_height_vps_in_luma_samples	u(16)
bit_depth_vps_luma_minus8	u(3)
bit_depth_vps_chroma_minus8	u(3)
}	

[0224]

[0225]

파라미터 chroma_format_vps_idc, separate_colour_plane_vps_flag, pic_width_vps_in_luma_samples, pic_height_vps_in_luma_samples, bit_depth_vps_luma_minus8, 및 bit_depth_vps_chroma_minus8는, VPS를 참조하는 각각의 SPS에 대해, 각각 SPS 선택스 엘리먼트들 (chroma_format_idc, separate_colour_plane_flag, pic_width_in_luma_samples, pic_height_in_luma_samples, bit_depth_luma_minus8, 및 bit_depth_chroma_minus8)의 값들의 추론을 위해 사용될 수도 있다. 이들 선택스 엘리먼트들의 각각에 대해, 만약 있다면 대응하는 SPS 선택스 엘리먼트의 값에 적용되는 모든 제약조건들이 또한 적용된다. [END]

EXAMPLE B] 표현 포맷 정보는, 예를 들어, 시퀀스들의 비트 깊이, 크로마 샘플링 포맷, 해상도를 일반적으로 포함할 수도 있다. 표 5의 예에서 도시된 바와 같이, 표현 포맷은 chroma_format_vps_idc, separate_colour_plane_vps_flag, pic_width_vps_in_luma_samples, pic_height_vps_in_luma_samples, bit_depth_vps_luma_minus8, 및 bit_depth_vps_chroma_minus8을 포함할 수도 있다. 더욱이, 표 5에 예시된 바와 같이, VPS에서의 표현 포맷의 시그널링은, 표현 포맷, 예컨대, chroma_format_vps_idc, separate_colour_plane_vps_flag, pic_width_vps_in_luma_samples, pic_height_vps_in_luma_samples, bit_depth_vps_luma_minus8, 및 bit_depth_vps_chroma_minus8이 엔트로피 디코딩 없이 액세스 가능도록 수행될 수도 있다. 다르게 말하면, 표 5에 예시된 바와 같이, chroma_format_vps_idc, separate_colour_plane_vps_flag, pic_width_vps_in_luma_samples, pic_height_vps_in_luma_samples, bit_depth_vps_luma_minus8, 및 bit_depth_vps_chroma_minus8은 엔트로피 코딩되지 않으며, 즉, 기술자들은 ue(v) 가 아니다. 몇몇 예들에서, 각각의 계층은 특정 표현 포맷과 연관될 수도 있다.

[0226]

[START EXAMPLE F] 표 6은 다른 SPS 파라미터들의 선택스 및 시멘틱스의 일 예를 도시한다.

[0227]

표 6 - 다른 SPS 파라미터들 선택스 및 시멘틱스

기술자	
conformance_window_vps_flag	u(1)
if(conformance_window_vps_flag) {	
conf_win_vps_left_offset	ue(v)
conf_win_vps_right_offset	ue(v)
conf_win_vps_top_offset	ue(v)
conf_win_vps_bottom_offset	ue(v)
}	
log2_vps_max_pic_order_cnt_lsb_minus4	ue(v)
log2_vps_min_luma_coding_block_size_minus3	ue(v)
log2_vps_diff_max_min_luma_coding_block_size	ue(v)
log2_vps_min_transform_block_size_minus2	ue(v)
log2_vps_diff_max_min_transform_block_size	ue(v)
max_vps_transform_hierarchy_depth_inter	ue(v)
max_vps_transform_hierarchy_depth_intra	ue(v)
scaling_list_enabled_vps_flag	u(1)
if(scaling_list_enabled_vps_flag)	
sps_scaling_list_data_present_vps_flag	u(1)
if(sps_scaling_list_data_present_vps_flag)	
scaling_list_data()	
}	
amp_enabled_vps_flag	u(1)
sample_adaptive_offset_enabled_vps_flag	u(1)
pcm_enabled_vps_flag	u(1)

[0228]

if(pcm_enabled_vps_flag) {	
pcm_vps_sample_bit_depth_luma_minus1	u(4)
pcm_vps_sample_bit_depth_chroma_minus1	u(4)
log2_vps_min_pcm_luma_coding_block_size_minus3	ue(v)
log2_vps_diff_max_min_pcm_luma_coding_block_size	ue(v)
pcm_vps_loop_filter_disabled_flag	u(1)
}	
long_term_ref_pics_present_vps_flag	u(1)
if(long_term_ref_pics_present_vps_flag) {	
num_long_term_ref_pics_vps	ue(v)
for(i = 0; i < num_long_term_ref_pics_vps; i++) {	
lt_ref_pic_poc_lsb_vps[i]	u(v)
used_by_curr_pic_lt_vps_flag[i]	u(1)
}	
}	
temporal_mvp_enabled_vps_flag	u(1)
strong_intra_smoothing_enabled_vps_flag	u(1)
}	

[0229]

[0230]

그 선택스 엘리먼트들의 각각에 대해, 만약 있다면 대응하는 SPS 선택스 엘리먼트의 값에 적용되는 모든 제약조건들이 또한 적용된다: conformance_window_vps_flag, conf_win_vps_left_offset, conf_win_vps_right_offset, conf_win_vps_top_offset, conf_win_vps_bottom_offset, log2_vps_max_pic_order_cnt_lsb_minus4, log2_vps_min_luma_coding_block_size_minus3, log2_vps_diff_max_min_luma_coding_block_size, log2_vps_min_transform_block_size_minus2, log2_vps_diff_max_min_transform_block_size, max_vps_transform_hierarchy_depth_inter, max_vps_transform_hierarchy_depth_intra, scaling_list_enabled_vps_flag, sps_scaling_list_data_present_vps_flag, amp_enabled_vps_flag, sample_adaptive_offset_enabled_vps_flag, pcm_enabled_vps_flag, pcm_vps_sample_bit_depth_luma_minus1, pcm_vps_sample_bit_depth_chroma_minus1, log2_vps_min_pcm_luma_coding_block_size_minus3, log2_vps_diff_max_min_pcm_luma_coding_block_size, lt_ref_pic_poc_lsb_vps[i], used_by_curr_pic_lt_vps_flag[i], temporal_mvp_enabled_vps_flag, 및 strong_intra_smoothing_enabled_vps_flag는 각각, conformance_window_flag, conf_win_left_offset, conf_win_right_offset, conf_win_top_offset, conf_win_bottom_offset, log2_max_pic_order_cnt_lsb_minus4, log2_min_luma_coding_block_size_minus3, log2_diff_max_min_luma_coding_block_size, log2_min_transform_block_size_minus2, log2_diff_max_min_transform_block_size, max_transform_hierarchy_depth_inter, max_transform_hierarchy_depth_intra, scaling_list_enabled_flag, sps_scaling_list_data_present_flag, amp_enabled_flag, sample_adaptive_offset_enabled_flag, pcm_enabled_flag, pcm_sample_bit_depth_luma_minus1, pcm_sample_bit_depth_chroma_minus1, log2_min_pcm_luma_coding_block_size_minus3, log2_diff_max_min_pcm_luma_coding_block_size, pcm_loop_filter_disabled_flag, long_term_ref_pics_present_flag, num_long_term_ref_pics_sps, lt_ref_pic_poc_lsb_sps[i], used_by_curr_pic_lt_flag[i], sps_temporal_mvp_enabled_flag, 및 strong_intra_smoothing_enabled_flag의 값들의 추론을 위해 사용될 수도 있다. [END EXAMPLE F]

[0231]

[START EXAMPLE D] 표 7 은 단기 RPS 후보들의 선택스 및 시멘틱스의 일 예를 도시한다.

[0232]

표 7 - 단기 RPS 후보들의 선택스 및 시맨틱스

short_term_rps_candidates(inferenceEnabledFlag) {	기술자
num_short_term_ref_pic_sets	ue(v)
if(inferenceEnabledFlag)	
pred_st_rps_cand_idx_plus1	u(v)
for(i = 0; i < num_short_term_ref_pic_sets; i++) {	
if(pred_st_rps_cand_idx_plus1 > 0)	
pred_from_rps_cand_list_flag[i]	u(1)
if(!pred_from_rps_cand_list_flag[i])	
short_term_ref_pic_set(i)	
else	
idx_in_rps_cand[i]	ue(v)
}	
}	

[0233]

파라미터 num_short_term_ref_pic_sets는 다음의 short_term_ref_pic_set() 선택스 구조들의 수를 특정한다. num_short_term_ref_pic_sets의 값은 0 내지 64의 범위에 있을 수도 있다.

[0234]

주의 1 - 디코더가 num_short_term_ref_pic_sets + 1 short_term_ref_pic_set() 선택스 구조들의 총 수에 대해 메모리를 할당해야 하는데, 현재 화상의 슬라이스 헤더들에서 직접적으로 시그널링된 short_term_ref_pic_set() 선택스 구조가 있을 수도 있어서이다. 현재 화상의 슬라이스 헤더들에서 직접적으로 시그널링된 short_term_ref_pic_set() 선택스 구조는 num_short_term_ref_pic_sets와 동일한 인덱스를 갖는다.

[0235]

파라미터 pred_st_rps_cand_idx_plus1 빼기 1은 현재 short_term_rps_candidates() 선택스 구조의 적어도 하나의 short_term_ref_pic_set() 선택스 구조로 유추하는데 사용되는 short_term_rps_candidates() 선택스 구조의, VPS에서의 short_term_rps_candidates() 선택스 구조들의 세트로의 인덱스를 특정한다. 존재하지 않는 경우, pred_st_rps_cand_idx_plus1은 0과 동일한 것으로 유추될 수도 있다. pred_st_rps_cand_idx_plus1의 값은 1 내지 vps_num_st_rps_candidates - 1의 범위에 있을 수도 있다.

[0236]

1과 동일한 pred_from_rps_cand_list_flag[i]를 갖는 것은, 현재 short_term_rps_candidates() 선택스 구조의 i-번째 short_term_ref_pic_set() 선택스 구조가 존재하지 않고 다른 short_term_rps_candidates() 선택스 구조에 존재하는 short_term_ref_pic_set() 선택스 구조들 중 하나인 것으로 설정된다는 것을 특정한다. 0과 동일한 pred_from_rps_cand_list_flag[i]는 현재 short_term_rps_candidates() 선택스 구조의 i-번째 short_term_ref_pic_set() 선택스 구조가 존재한다는 것을 특정한다. 존재하지 않는 경우, pred_from_rps_cand_list_flag[i]의 값은 0과 동일한 것으로 유추될 수도 있다. [END EXAMPLE D]

[0237]

대안으로, 1과 동일한 pred_from_rps_cand_list_flag[i]는, i-번째 short_term_ref_pic_set()에 대응하는 변수들 (DeltaPocS0, DeltaPocS1, UsedByCurrPicS1, UsedByCurrPicS0, NumPositivePics, NumNegativePics, 및 NumDeltaPocs) 이 다른 short_term_rps_candidates() 구조에 대응하는 변수들 (DeltaPocS0, DeltaPocS1, UsedByCurrPicS1, UsedByCurrPicS0, NumPositivePics, NumNegativePics, 및 NumDeltaPocs)과 각각 동일한 것으로 유추된다는 것을 특정한다.

[0238]

[START EXAMPLE D] 파라미터 idx_in_rps_cand[i]는 현재 short_term_rps_candidates()의 i-번째 short_term_ref_pic_set() 선택스 구조와 동일한 short_term_ref_pic_set() 선택스 구조의, VPS에서의 (pred_st_rps_cand_idx_plus1 - 1)-번째 short_term_rps_candidates() 선택스 구조의 short_term_ref_pic_set() 선택스 구조들의 세트로의 인덱스를 특정한다.

[0239]

pred_from_rps_cand_list_flag[i]가 1과 동일한 경우, 현재 short_term_rps_candidates() 선택스 구조에서의 i-번째 short_term_ref_pic_set() 선택스 구조는, VPS에서의 (pred_st_rps_cand_idx_plus1 - 1)-번째

short_term_rps_candidates() 선택스 구조의 idx_in_rps_cand[i]-번째 short_term_ref_pic_set() 선택스 구조와 동일하게 설정된다. [END EXAMPLE D]

[0241] 대안으로, pred_from_rps_cand_list_flag[i]가 1과 동일한 경우, 현재 short_term_rps_candidates() 선택스 구조에서의 i-번째 short_term_ref_pic_set() 선택스 구조에 대응하는 변수들 (DeltaPocS0, DeltaPocS1, UsedByCurrPicS1, UsedByCurrPicS0, NumPositivePics, NumNegativePics, 및 NumDeltaPocs) 은, VPS에서의 (pred_st_rps_cand_idx_plus1 - 1)-번째 short_term_rps_candidates() 선택스 구조의 idx_in_rps_cand[i]-번째 short_term_ref_pic_set() 선택스 구조에 대응하는 변수들 (DeltaPocS0, DeltaPocS1, UsedByCurrPicS1, UsedByCurrPicS0, NumPositivePics, NumNegativePics, 및 NumDeltaPocs) 과 각각 동일하게 설정된다. 표 8 은 VUI 파라미터들의 선택스 및 시맨틱스의 일 예를 도시한다.

표 8 - VUI 파라미터들의 선택스 및 시맨틱스

vui_parameters([START EXAMPLE E] timingParamsPresentFlag) {	기술자
[START EXAMPLE E] vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag) { [END EXAMPLE E]	
aspect_ratio_info_present_flag	u(1)
if(aspect_ratio_info_present_flag) {	
aspect_ratio_idc	u(8)
if(aspect_ratio_idc == EXTENDED_SAR) {	
sar_width	u(16)
sar_height	u(16)
}	
}	
overscan_info_present_flag	u(1)
if(overscan_info_present_flag)	
overscan_appropriate_flag	u(1)
video_signal_type_present_flag	u(1)
if(video_signal_type_present_flag) {	
video_format	u(3)
video_full_range_flag	u(1)
colour_description_present_flag	u(1)
if(colour_description_present_flag) {	
colour_primaries	u(8)
transfer_characteristics	u(8)
matrix_coeffs	u(8)
}	
}	
chroma_loc_info_present_flag	u(1)

[0243]

if(chroma_loc_info_present_flag) {	
chroma_sample_loc_type_top_field	ue(v)
chroma_sample_loc_type_bottom_field	ue(v)
}	
neutral_chroma_indication_flag	u(1)
field_seq_flag	u(1)
frame_field_info_present_flag	u(1)
default_display_window_flag	u(1)
if(default_display_window_flag) {	
def_disp_win_left_offset	ue(v)
def_disp_win_right_offset	ue(v)
def_disp_win_top_offset	ue(v)
def_disp_win_bottom_offset	ue(v)
}	
[START EXAMPLE E] if(timingParamsPresentFlag) { [END EXAMPLE E]	
vui_timing_info_present_flag	u(1)
if(vui_timing_info_present_flag) {	
vui_num_units_in_tick	u(32)
vui_time_scale	u(32)
vui_poc_proportional_to_timing_flag	u(1)
if(vui_poc_proportional_to_timing_flag)	
vui_num_ticks_poc_diff_one_minus1	ue(v)
vui_hrd_parameters_present_flag	u(1)
if(vui_hrd_parameters_present_flag)	
hrd_parameters(1,	
sps_max_sub_layers_minus1)	
}	

[START EXAMPLE E]	}	[END EXAMPLE E]	
bitstream_restriction_flag		u(1)	
if(bitstream_restriction_flag) {			
tiles_fixed_structure_flag		u(1)	
motion_vectors_over_pic_boundaries_flag		u(1)	
restricted_ref_pic_lists_flag		u(1)	
min_spatial_segmentation_idc		ue(v)	
max_bytes_per_pic_denom		ue(v)	
max_bits_per_min_cu_denom		ue(v)	
log2_max_mv_length_horizontal		ue(v)	
log2_max_mv_length_vertical		ue(v)	
}			
[START EXAMPLE E]	}	[END EXAMPLE E]	
}			

[0245]

[0246] [START EXAMPLE E] timingParamsPresentFlag가 0과 동일한 경우, 다음이 적용된다:
 vui_timing_info_present_flag, vui_num_units_in_tick, vui_time_scale,
 vui_poc_proportional_to_timing_flag, 및 vui_num_ticks_poc_diff_one_minus1의 값들은 각각
 vps_timing_info_present_flag, vps_num_units_in_tick, vps_time_scale,
 vps_poc_proportional_to_timing_flag, 및 vps_num_ticks_poc_diff_one_minus1과 동일한 것으로 유추된다.
 vui_hrd_parameters_present_flag의 값은 0과 동일한 것으로 유추될 수도 있다.

[0247] 1과 동일한 파라미터 vui_parameters_present_flag는 vui_parameters() 선택스 구조에 더 많은 선택스 엘리먼트들이 있음을 특정한다. 0과 동일한 vui_parameters_present_flag는 vui_parameters() 선택스 구조에 선택스 엘리먼트들이 더 이상 없음을 특정한다.... [END EXAMPLE E]

[0248] 하나의 예에서, 표 8에서 알 수 있듯이, 시각적 신호 정보 (예컨대, video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs) 는 VUI 파라미터들의 서브세트이다. 게다가, 표 4의 예에서 도시된 바와 같이, VUI와, 그래서 시각적 신호 정보는, VPS에 있다. 표 4 및 표 8로 부터의 정보를 결합함으로써 시각적 신호 정보는 VPS에서의 계층마다 시그널링된다는 것이 보여진다.

[0249] 표 9는 일 예의 시퀀스 파라미터 세트 RBSP 선택스 및 시맨틱스를 도시한다.

[0250]

표 9 - 시퀀스 파라미터 세트 RBSP 신택스 및 시맨틱스

seq_parameter_set_rbsp() {	기술자
sps_video_parameter_set_id	u(4)
[START EXAMPLE F] if(nuh_layer_id > 0)	
inherit_sps_params_from_vps_flag	u(1)
if(!inherit_sps_params_from_vps_flag) { [END EXAMPLE F]	
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
[START EXAMPLE F] } [END EXAMPLE F]	
if(nuh_layer_id == 0)	
profile_tier_level(1, sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
[START EXAMPLE B] if(nuh_layer_id == 0) { [END EXAMPLE B]	
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
[START EXAMPLE B] } [END EXAMPLE B]	
[START EXAMPLE F] if(!inherit_sps_params_from_vps_flag) { [END EXAMPLE F]	
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	

[0251]

[START EXAMPLE F]	}	[END EXAMPLE F]	
[START EXAMPLE B]	if(nuh_layer_id == 0) {	[END EXAMPLE B]	
bit_depth_luma_minus8		ue(v)	
bit_depth_chroma_minus8		ue(v)	
[START EXAMPLE B]	}	[END EXAMPLE B]	
[START EXAMPLE F]	if(!inherit_sps_params_from_vps_flag)		
{ [END EXAMPLE F]			
log2_max_pic_order_cnt_lsb_minus4		ue(v)	
sps_sub_layer_ordering_info_present_flag		u(1)	
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1);			
i <= sps_max_sub_layers_minus1; i++) {			
sps_max_dec_pic_buffering_minus1[i]		ue(v)	
sps_max_num_reorder_pics[i]		ue(v)	
sps_max_latency_increase_plus1[i]		ue(v)	
}			
log2_min_luma_coding_block_size_minus3		ue(v)	
log2_diff_max_min_luma_coding_block_size		ue(v)	
log2_min_transform_block_size_minus2		ue(v)	
log2_diff_max_min_transform_block_size		ue(v)	
max_transform_hierarchy_depth_inter		ue(v)	
max_transform_hierarchy_depth_intra		ue(v)	
scaling_list_enabled_flag		u(1)	
if(scaling_list_enabled_flag) {			
sps_scaling_list_data_present_flag		u(1)	
if(sps_scaling_list_data_present_flag)			
scaling_list_data()			
}			

[0252]

amp_enabled_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)
log2_min_pcm_luma_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_luma_coding_block_size	ue(v)
pcm_loop_filter_disabled_flag	u(1)
}	
[START EXAMPLE D] short_term_rps_candidates(0)	
num_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i) [END EXAMPLE D]	
long_term_ref_pics_present_flag	u(1)
if(long_term_ref_pics_present_flag) {	
num_long_term_ref_pics_sps	ue(v)
for(i = 0; i < num_long_term_ref_pics_sps; i++) {	
lt_ref_pic_poc_lsb_sps[i]	u(v)
used_by_curr_pic_lt_sps_flag[i]	u(1)
}	
}	
sps_temporal_mvp_enabled_flag	u(1)
strong_intra_smoothing_enabled_flag	u(1)
[START EXAMPLE E] vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag) [END EXAMPLE E]	
vui_parameters([START EXAMPLE E] nuh_layer_id == 0 [END EXAMPLE E])	

[START EXAMPLE F]	} [END EXAMPLE F]	
sps_extension_flag		u(1)
if(sps_extension_flag) {		
sps_extension()		
sps_extension2_flag		u(1)
if(sps_extension2_flag)		
while(more_rbsp_data())		
sps_extension_data_flag		u(1)
}		
rbsp_trailing_bits()		
}		

[0254]

[START EXAMPLE B]

[0256]

SPS의 nuh_layer_id가 0과 동일한 경우, chroma_format_idc, separate_colour_plane_flag, pic_width_in_luma_samples, pic_height_in_luma_samples, bit_depth_luma_minus8, 및 bit_depth_chroma_minus8의 값들은, 액티브 VPS에서의 0-번째 rep_format() 신택스 구조의 chroma_format_vps_idc, separate_colour_plane_vps_flag, pic_width_vps_in_luma_samples, pic_height_vps_in_luma_samples, bit_depth_vps_luma_minus8, 및 bit_depth_vps_chroma_minus8와 각각 동일 할 수도 있다.

[0257]

SPS를 참조하는 0보다 더 큰 nuh_layer_id를 갖는 각각의 계층의 경우, layer_id_in_nuh[layerIdx]가 그 계층의 nuh_layer_id와 동일하게 되는 값과 동일하게 layerIdx가 설정된다고 하면, 다음이 적용된다: **[END EXAMPLE B]**

[0258]

[START EXAMPLE A] SPS의 nuh_layer_id가 0과 동일한 경우, SPS에서의 profile_tier_level() 신택스 구조는 무시된다. **[END EXAMPLE A]**

[0259]

표 8 및 표 9에 예시된 바와 같이, 타이밍 정보가, 예컨대, 계층 0보다 더 큰 임의의 계층에 대해, SPS VUI에서 시그널링되지 않는다.

[0260]

-**[START EXAMPLE B]** chroma_format_idc, separate_colour_plane_flag, pic_width_in_luma_samples, pic_height_in_luma_samples, bit_depth_luma_minus8, 및 bit_depth_chroma_minus8의 값들은, 이들 신택스 엘리먼트들이 VPS에서 존재하는지의 여부에 상관 없이, 액티브 VPS에서의 vps_rep_format_idx[layerIdx]-번째 rep_format() 신택스 구조의 chroma_format_vps_idc, separate_colour_plane_vps_flag, pic_width_vps_in_luma_samples, pic_height_vps_in_luma_samples, bit_depth_vps_luma_minus8, 및 bit_depth_vps_chroma_minus8와 각각 동일하게 되도록 유추된다. **[END EXAMPLE B]**

[0261]

[START EXAMPLE F] 1과 동일한 파라미터 inherit_sps_params_from_vps_flag는, SPS RBSP의 경우, 신택스 엘리먼트 sps_extension_flag까지의 모든 신택스 엘리먼트들 및 신택스 구조들의 값들이, 신택스 엘리먼트들 (sps_video_parameter_set_id 및 sps_seq_parameter_set_id) 과는 달리, 액티브 VPS로부터 상속될 수도 있다는 것을 특정한다. 0과 동일한 inherit_sps_params_from_vps_flag는 이들 값들이 액티브 VPS로부터 상속되지 않는다는 것을 특정한다. 존재하지 않는 경우, inherit_sps_params_from_vps_flag의 값은 0과 동일한 것으로 유추될 수도 있다. vps_num_rep_formats, vps_num_other_sps_params, vps_num_st_rps_candidates, 또는 vps_num_vui_params가 0과 동일한 경우, inherit_sps_params_from_vps_flag의 값은 0과 동일할 수도 있다. inherit_sps_params_from_vps_flag가 1과 동일한 경우, 다음이 적용된다: **[END EXAMPLE F]**

[0262]

[START EXAMPLE C] SPS를 참조하는 각각의 계층에 대한 sps_max_sub_layers_minus1의 값은, layer_id_in_nuh[layerIdx]가 계층의 nuh_layer_id와 동일하게 되는 값과 layerIdx가 동일한 경우, max_sub_layers_vps_minus1[layerIdx]와 동일한 것으로 유추될 수도 있다.

[0263]

SPS를 참조하는 각각의 계층에 대한 sps_max_dec_pic_buffering_minus1[i]의 값들은 0 내지

MaxSubLayers[1sIdx] - 1의 범위에서의 i에 대해 max_vps_dec_pic_buffering_minus1[1sIdx][i]와 동일한 것으로 유추될 수도 있는데, 여기서 1sIdx는 0 내지 vps_num_layer_sets_minus1의 범위에 있고, 그 계층이 최고 계층 및 유일한 타겟 출력 계층이 되는 출력 계층 세트의, 출력 계층 세트들 중 VPS에 의해 특정된 세트로의 인덱스를 나타낸다.

[0264] SPS를 참조하는 각각의 계층에 대한 sps_max_num_reordered_pics[i] 및 sps_max_latency_increase_plus1[i]의 값들은 max_vps_num_reordered_pics[layerIdx][i] 및 max_vps_latency_increase_plus1[layerIdx][i]와 동일한 것으로 유추되는데, 여기서 layerIdx는, 0 내지 sps_max_sub_layers_minus1의 범위에서의 i에 대해, layer_id_in_nuh[layerIdx]가 그 계층의 nuh_layer_id와 동일하게 되는 값과 동일하다. SPS를 참조하는 각각의 계층에 대한 sps_temporal_id_nesting_flag의 값은 vps_temporal_id_nesting_flag와 동일한 것으로 유추될 수도 있다. [END EXAMPLE C]

[0265] 위의 논의 및 표 9로부터 알 수 있듯이, 몇몇 예들은 nuh_layer_id > 0인 경우 sps_max_sub_layers_minus1 및 sps_temporal_id_nesting_flag를 시그널링하지 않는다.

[START EXAMPLE F] SPS를 참조하는 각각의 계층에 대한 conformance_window_flag, conf_win_left_offset, conf_win_right_offset, conf_win_top_offset, conf_win_bottom_offset, log2_max_pic_order_cnt_lsb_minus4, log2_min_luma_coding_block_size_minus3, log2_diff_max_min_luma_coding_block_size, log2_min_transform_block_size_minus2, log2_diff_max_min_transform_block_size, max_transform_hierarchy_depth_inter, max_transform_hierarchy_depth_intra, scaling_list_enabled_flag, sps_scaling_list_data_present_flag, amp_enabled_flag, sample_adaptive_offset_enabled_flag, pcm_enabled_flag, pcm_sample_bit_depth_luma_minus1, pcm_sample_bit_depth_chroma_minus1, log2_min_pcm_luma_coding_block_size_minus3, log2_diff_max_min_pcm_luma_coding_block_size, pcm_loop_filter_disabled_flag, long_term_ref_pics_present_flag, num_long_term_ref_pics_sps, lt_ref_pic_poc_lsb_sps[i], used_by_curr_pic_lt_flag[i], sps_temporal_mvp_enabled_flag, 및 strong_intra_smoothing_enabled_flag의 값들은, 액티브 VPS에서의 vps_other_sps_params_idx[layerIdx]-번 째 other_sps_parameters() 선택스 구조의 conformance_window_vps_flag, conf_win_vps_left_offset, conf_win_vps_right_offset, conf_win_vps_top_offset, conf_win_vps_bottom_offset, log2_vps_max_pic_order_cnt_lsb_minus4, log2_vps_min_luma_coding_block_size_minus3, log2_vps_diff_max_min_luma_coding_block_size, log2_vps_min_transform_block_size_minus2, log2_vps_diff_max_min_transform_block_size, max_vps_transform_hierarchy_depth_inter, max_vps_transform_hierarchy_depth_intra, scaling_list_enabled_vps_flag, sps_scaling_list_data_present_vps_flag, amp_enabled_vps_flag, sample_adaptive_offset_enabled_vps_flag, pcm_enabled_vps_flag, pcm_vps_sample_bit_depth_luma_minus1, pcm_vps_sample_bit_depth_chroma_minus1, log2_vps_min_pcm_luma_coding_block_size_minus3, log2_vps_diff_max_min_pcm_luma_coding_block_size, pcm_vps_loop_filter_disabled_flag, long_term_ref_pics_present_vps_flag, num_long_term_ref_pics_vps, lt_ref_pic_poc_lsb_vps[i], used_by_curr_pic_lt_vps_flag[i], temporal_mvp_enabled_vps_flag, 및 strong_intra_smoothing_enabled_vps_flag와 각각 동일한 것으로 유추되는데, 여기서 layerIdx는 layer_id_in_nuh[layerIdx]가 그 계층의 nuh_layer_id와 동일하게 되는 값과 동일하다.

[0267] sps_scaling_list_data_present_vps_flag가 1과 동일한 경우, SPS를 유추하는 각각의 계층에 대한 선택스 구조 scaling_list_data()의 값은 액티브 VPS에서의 vps_other_sps_params_idx[layerIdx]-번 째 other_sps_parameters() 선택스 구조의 선택스 구조 scaling_list_data()와 동일한 것으로 유추될 수도 있는데, 여기서 layerIdx는 layer_id_in_nuh[layerIdx]가 그 계층의 nuh_layer_id와 동일하게 될 수도 있는 값과 동일하다. [END EXAMPLE F]

[START EXAMPLE D] SPS를 유추하는 각각의 계층에 대한 선택스 구조 short_term_rps_candidates()의 값은 액티브 VPS에서의 vps_st_rps_idx[layerIdx]-번 째 short_term_rps_candidates() 선택스 구조와 동일한 것으로 유추될 수도 있는데, 여기서 layerIdx는 layer_id_in_nuh[layerIdx]가 그 계층의 nuh_layer_id와 동일하게 되는 값과 동일하다. [END EXAMPLE D]

[START EXAMPLE E] 몇몇 예들에서, SPS를 유추하는 각각의 계층에 대한 선택스 구조 vui_parameters()의 값은 액티브 VPS에서의 vps_vui_params_idx[layerIdx]-번 째 vui_parameters() 선택스 구조와 동일한 것으로 유추될 수도 있는데, 여기서 layerIdx는 layer_id_in_nuh[layerIdx]가 그 계층의 nuh_layer_id와 동일하게 되는

값과 동일하다. [END EXAMPLE E]

[0270] ...

[0271] 파라미터 conf_win_left_offset, conf_win_right_offset, conf_win_top_offset, 및 conf_win_bottom_offset은 출력을 위한 좌표들에서 특정된 직사각형 지역의 관점에서, 디코딩 프로세스로부터 출력되는, 코딩된 비디오 스트림 (coded video stream, CVS)에서의 화상들의 샘플들을 특정한다. [START EXAMPLE F]

inherit_sps_params_from_vps_flag가 0과 동일하고 [END EXAMPLE F] conformance_window_flag가 0과 동일한 경우, conf_win_left_offset, conf_win_right_offset, conf_win_top_offset, 및 conf_win_bottom_offset의 값들은 0과 동일한 것으로 유추될 수도 있다.

[0272] 적합성 크로핑 윈도우는 SubWidthC * conf_win_left_offset부터 pic_width_in_luma_samples - (SubWidthC * conf_win_right_offset + 1) 까지의 수평 화상 좌표들 및 SubHeightC * conf_win_top_offset 내지 pic_height_in_luma_samples - (SubHeightC * conf_win_bottom_offset + 1) 의 수직 화상 좌표들을 갖는 루마 샘플들을 포함한다.

[0273] SubWidthC * (conf_win_left_offset + conf_win_right_offset) 의 값은 pic_width_in_luma_samples 미만일 수도 있고, SubHeightC * (conf_win_top_offset + conf_win_bottom_offset) 의 값은 pic_height_in_luma_samples 미만일 수도 있다.

[0274] ChromaArrayType이 0과 동일하지 않은 경우, 두 개의 크로마 어레이들의 대응하는 특정된 샘플들은 화상 좌표들 (x/SubWidthC, y/SubHeightC) 를 갖는 샘플들이며, 여기서 (x, y)는 특정 루마 샘플들의 화상 좌표들이다.

[0275] 주의 3 - 적합성 크로핑 윈도우 오프셋 파라미터들은 출력에서만 적용된다. 모든 내부 디코딩 프로세스들이 비크로핑된 화상 사이즈에 적용된다.

[0276] ...

[0277] 1과 동일한 파라미터 sps_sub_layer_ordering_info_present_flag는 [START EXAMPLE C], inherit_sps_params_from_vps_flag가 0과 동일한 경우, [END EXAMPLE C] sps_max_dec_pic_buffering_minus1[i], sps_max_num_reordered_pics[i], 및 sps_max_latency_increase_plus1[i]가 sps_max_sub_layers_minus1 + 1 서브-계층들에 대해 존재한다는 것을 특정한다. 0과 동일한 sps_sub_layer_ordering_info_present_flag는 [START EXAMPLE C], inherit_sps_params_from_vps_flag가 0과 동일한 경우, [END EXAMPLE C] sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1], sps_max_num_reordered_pics[sps_max_sub_layers_minus1], 및 sps_max_latency_increase_plus1[sps_max_sub_layers_minus1]의 값들이 모든 서브-계층들에 적용된다는 것을 특정한다.

[0278] 파라미터 sps_max_dec_pic_buffering_minus1[i] 더하기 1은 HighestTid가 i와 동일한 경우 화상 저장 버퍼들의 단위에서 CVS에 대한 디코딩된 화상 버퍼의 최대 요구된 사이즈를 특정한다. 몇몇 예들에서, sps_max_dec_pic_buffering_minus1[i]의 값은 0 내지 MaxDpbSize - 1 (하위절 A.4에서 특정된 바와 같음) 의 범위에 있을 수도 있다. i가 0보다 더 큰 경우, sps_max_dec_pic_buffering_minus1[i]는 sps_max_dec_pic_buffering_minus1[i - 1] 이상일 수도 있다. sps_max_dec_pic_buffering_minus1[i]의 값은 i의 각각의 값에 대해 vps_max_dec_pic_buffering_minus1[i] 이하일 수도 있다. [START EXAMPLE C] inherit_sps_params_from_vps_flag가 0과 동일하고 [END EXAMPLE C] sps_max_dec_pic_buffering_minus1[i]가 0 내지 sps_max_sub_layers_minus1 - 1의 범위에서의 i에 대해 존재하는 않는 경우, 0과 동일한 sps_sub_layer_ordering_info_present_flag로 인해, 그것은 sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1]과 동일한 것으로 유추될 수도 있다.

[0279] 파라미터 sps_max_num_reordered_pics[i]는 HighestTid가 i와 동일한 경우 디코딩 순서에서 CVS에서의 임의의 화상에 선행하고 출력 순서에서 당해 화상을 뒤따를 수 있는 화상들의 최대 허용된 수를 나타낸다. sps_max_num_reordered_pics[i]의 값은 0 내지 sps_max_dec_pic_buffering_minus1[i]의 범위에 있을 수도 있다. i가 0보다 더 큰 경우, sps_max_num_reordered_pics[i]는 sps_max_num_reordered_pics[i - 1] 이상일 수도 있다. sps_max_num_reordered_pics[i]의 값은 i의 각각의 값에 대해 vps_max_num_reordered_pics[i] 이하일 수도 있다. [START EXAMPLE C] inherit_sps_params_from_vps_flag가 0과 동일하고 [END EXAMPLE C] sps_max_num_reordered_pics[i]가 0 내지 sps_max_sub_layers_minus1 - 1의 범위에서의 i에 대해 존재하지 않는 경우, 0과 동일한 sps_sub_layer_ordering_info_present_flag로 인해, 그것은

`sps_max_num_reordered_pics[sps_max_sub_layers_minus1]`과 동일한 것으로 유추될 수도 있다.

[0280] 0과 동일하지 않는 파라미터 `sps_max_latency_increase_plus1[i]`는 `SpsMaxLatencyPictures[i]`의 값을 컴퓨팅하는데 사용되는데, 그 값은 `HighestTid`가 `i`와 동일한 경우 출력 순서에서 CVS에서의 임의의 화상에 선행하고 디코딩 순서에서 당해 화상을 뒤따를 수 있는 화상들의 최대 수를 특정한다.

[0281] `sps_max_latency_increase_plus1[i]`가 0과 동일하지 않는 경우, `SpsMaxLatencyPictures[i]`의 값은 다음과 같이 특정된다: $SpsMaxLatencyPictures[i] = sps_max_num_reordered_pics[i] + (7-9)$

[0282] `sps_max_latency_increase_plus1[i] - 1`

[0283] `sps_max_latency_increase_plus1[i]`가 0과 동일한 경우, 대응하는 제한은 표현되지 않는다. 몇몇 예들에서, `sps_max_latency_increase_plus1[i]`의 값은 0 내지 232 - 2의 범위에 있을 수도 있다. 몇몇 예들에서, `vps_max_latency_increase_plus1[i]`가 0과 동일하지 않는 경우, `sps_max_latency_increase_plus1[i]`의 값은 0과 동일할 수도 있고 `i`의 각각의 값에 대해 `vps_max_latency_increase_plus1[i]` 이하일 수도 있다.

[START EXAMPLE C] `inherit_sps_params_from_vps_flag`가 0과 동일하고 [END EXAMPLE C] `sps_max_latency_increase_plus1[i]`가 0 내지 `sps_max_sub_layers_minus1 - 1`의 범위에서의 `i`에 대해 존재하는 않는 경우, 0과 동일한 `sps_sub_layer_ordering_info_present_flag`로 인해, 그것은 `sps_max_latency_increase_plus1[sps_max_sub_layers_minus1]`과 동일한 것으로 유추될 수도 있다.

[0284] ...

[0285] 1과 동일한 파라미터 `sps_scaling_list_data_present_flag`는 스케일링 리스트 데이터가 SPS에서 존재하거나 [START EXAMPLE F] 또는 액티브 VPS로부터 상속된다는 것을 특정한다 [END EXAMPLE F]. 몇몇 예들에서, 0과 동일한 `sps_scaling_list_data_present_flag`는 스케일링 리스트 데이터가 SPS에 존재하지 않고 [START EXAMPLE F] 액티브 VPS로부터 상속되지 않는다는 것을 특정한다 [END EXAMPLE F]. [START EXAMPLE F] `inherit_sps_params_from_vps_flag`가 0과 동일하고 `sps_scaling_list_data_present_flag` [END EXAMPLE F]가 존재하지 않는 경우, `sps_scaling_list_data_present_flag`의 값은 0과 동일한 것으로 유추될 수도 있다. `scaling_list_enabled_flag`가 1과 동일하고 `sps_scaling_list_data_present_flag`가 0과 동일한 경우, 디폴트 스케일링 리스트 데이터는 HEVC 표준의 하위절 7.4.5에서 특정된 스케일링 리스트 데이터 시맨틱스에서 기술된 바와 같이 어레이 `ScalingFactor`를 도출하는데 사용된다.

[0286] 파라미터 `pcm_loop_filter_disabled_flag`는 루프 필터 프로세스가 1과 동일한 `pcm_flag`를 갖는 코딩 유닛에서의 복원된 샘플들에 대해 불가능인지의 여부를 다음과 같이 특정한다: `pcm_loop_filter_disabled_flag`가 1과 동일하면, 1과 동일한 `pcm_flag`를 갖는 코딩 유닛에서의 복원된 샘플들에 대해 프로세싱하는 샘플 적응 오프셋 필터와 블록화제거 필터는 불가능이 된다. 그렇지 않으면 (`pcm_loop_filter_disabled_flag` 값이 0과 동일하면), 1과 동일한 `pcm_flag`를 갖는 코딩 유닛에서의 복원된 샘플들에 대해 프로세싱하는 샘플 적응 오프셋 필터와 블록화제거 필터는 불가능이 되지 않는다.

[0287] [START EXAMPLE F] `inherit_sps_params_from_vps_flag`가 0과 동일하고 [END EXAMPLE F] `pcm_loop_filter_disabled_flag`가 존재하지 않는 경우, 그것은 0과 동일한 것으로 유추될 수도 있다.

[0288] [START EXAMPLE D] [[`num_short_term_ref_pic_sets`는 SPS에 포함된 `short_term_ref_pic_set()` 신택스 구조들의 수를 특정한다. `num_short_term_ref_pic_sets`의 값은 0 내지 64의 범위에 있을 것이다.]

[0289] 주의 5 - 디코더가 `num_short_term_ref_pic_sets + 1` `short_term_ref_pic_set()` 신택스 구조들의 총 수에 대해 메모리를 할당해야 하는데, 현재 화상의 슬라이스 헤더들에서 직접적으로 시그널링된 `short_term_ref_pic_set()` 신택스 구조가 있을 수도 있어서이다. 현재 화상의 슬라이스 헤더들에서 직접적으로 시그널링된 `short_term_ref_pic_set()` 신택스 구조는 `num_short_term_ref_pic_sets`와 동일한 인덱스를 갖는다.][END EXAMPLE D]

[0290] ...

[0291] 파라미터 `lt_ref_pic_poc_lsb_sps[i]`는 SPS[START EXAMPLE F]에 의해[[에서]][END EXAMPLE F] 특정된 `i`-번째 후보 장기 참조 화상의 화상 순서 카운트 모듈로 `MaxPicOrderCntLsb`를 특정한다. `lt_ref_pic_poc_lsb_sps[i]`를 표현하는데 사용된 비트들의 수는 $\log_2 \max_pic_order_cnt_lsb_minus4 + 4$ 와 동일하다.

[0292] 0과 동일한 파라미터 used_by_curr_pic_lt_sps_flag[i]는 SPS[START EXAMPLE F]에 의해 [[에서]][END EXAMPLE F] 특정된 i-번째 후보 장기 참조 화상이 [START EXAMPLE F]에 의해[[에서]][END EXAMPLE F] 특정된 i-번째 후보 장기 참조 화상을 자신의 장기 RPS 내에 포함하는 화상에 의한 참조를 위해 사용되지 않는다는 것을 특정한다.

[0293] ...

[0294] [START EXAMPLE E][[1과 동일한 vui_parameters_present_flag는 부록 E에서 특정된 바와 같은 vui_parameters() 선택스 구조가 존재한다는 것을 특정한다. 0과 동일한 vui_parameters_present_flag는 부록 E에서 특정된 바와 같은 vui_parameters() 선택스 구조가 존재하지 않음을 특정한다.]] [END EXAMPLE E]

[0295] 도 4는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 4의 도시된 예에서, 비디오 디코더 (30)는 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트를 수신할 수도 있다 (402). 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에 있을 수도 있다.

[0296] 비디오 디코더 (30)는 복수의 출력 계층 세트들의 각각에 대해 프로파일, 티어, 및 레벨 선택스 구조들 중 하나를 참조하고 (404), 출력 계층 세트에 대해 참조된 프로파일, 티어, 및 레벨 선택스 구조로부터의 정보에 기초하여 출력 계층 세트들 중 하나의 출력 계층 세트의 비디오 데이터를 디코딩할 수도 있다 (406). 예를 들어, 비디오 디코더 (30)는, 비디오 디코더 (30)의 디코딩 능력보다 더 높은 능력이, 디코더의 프로파일, 티어, 및 레벨에 의해 나타내어진 바와 같이, 비트스트림의 디코딩을 위해 필요하다는 것을 출력 계층 세트의 비트스트림의 프로파일, 티어, 및 레벨이 나타낸다면, 출력 계층 세트를 무시할 수도 있다. 요청된 디코딩 능력이, 출력 계층 세트의 비트스트림의 프로파일, 티어, 및 레벨에 의해 나타내어진 바와 같이, 비디오 디코더 (30)의 디코딩 능력보다 더 높지 않다면, 비디오 디코더 (30)는 그 비트스트림의 프로파일, 티어, 및 레벨에 의해 나타내어진 바와 같은 요구된 디코딩 프로세스들을 사용하여 출력 계층 세트의 비트스트림을 디코딩한다.

[0297] 몇몇 예들에서, 비디오 디코더 (30)는 0과 동일한 nuh_layer_id를 갖는 SPS를 수신할 수도 있는데, SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 선택스 구조를 포함한다.

[0298] 몇몇 예들에서, 비디오 디코더 (30)는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 수신할 수도 있다. 예를 들어, 비디오 인코더 (20)는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신할 수도 있다. 비디오 디코더 (30)는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신할 수도 있다.

[0299] 비디오 디코더 (30)는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 또한 생성할 수도 있다. 예를 들어, 비디오 인코더 (20)는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 또한 인코딩할 수도 있다.

[0300] 도 5는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 5의 도시된 예에서, 비디오 인코더 (20)는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 생성할 수도 있다. 예를 들어, 비디오 인코더 (20)는 복수의 출력 계층 세트들의 각각에 대해 프로파일, 티어, 및 레벨 선택스 구조들 중 하나를 참조할 수도 있다 (502). 비디오 인코더 (20)는 출력 계층 플래그 [i][j]에 기초하여 출력 계층 세트를 또한 인코딩할 수도 있다.

[0301] 비디오 인코더 (20)는 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트를 갖는 VPS 확장물과 함께 VPS를 인코딩할 수도 있다 (506). 몇몇 예들에서, 비디오 인코더 (20)는 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트를 갖는 VPS 확장물과 VPS를 송신, 저장, 또는 저장되게 할 수도 있다. 프로파일, 티어, 및 레벨 선택스 구조들의 비-엔트로피 인코딩된 세트는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에 있을 수도 있다. 예를 들어, 비디오 인코더 (20)는 선택되는 프로파일, 티어, 레벨에 따라 비디오 데이터를 인코딩할 수도 있다. 비디오 인코더 (20)

는 VPS에서의 프로파일, 티어, 레벨 선택스를 디코더에 의한 사용을 위해 또한 인코딩할 수도 있다.

[0302] 몇몇 예들에서, 비디오 인코더 (20)는 0과 동일한 nuh_layer_id를 갖는 SPS를 인코딩할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 선택스 구조를 포함한다. 예를 들어, 비디오 인코더 (20)는 0과 동일한 nuh_layer_id를 갖는 SPS를 전송할 수도 있는데, 그 SPS는 비디오 데이터의 계층에 대한 프로파일, 티어, 및 레벨 선택스 구조를 포함한다. 0과 동일한 nuh_layer_id를 갖는 SPS가 0보다 더 큰 nuh_layer_id를 갖는 계층에 의해 참조되는 경우, SPS에서의 profile_tier_level() 선택스 구조는 당해 계층에 대해 적용되지 않는다.

[0303] 따라서, 위에서 알 수 있듯이, 각각의 계층은 SPS를 참조하는 것을 필요로 한다. 기준에, 상이한 값들의 공간적 해상도들, 비트 깊이들, 또는 컬러 포맷들로 된 임의의 두 개의 계층들은 이를 표현 포맷 파라미터들이 SPS에서 시그널링될 때 두 개의 상이한 SPS들을 참조해야 한다. 그러나, 0과 동일한 nuh_layer_id를 갖는 것들을 제외한 모든 SPS들에 대한 이들 파라미터들이 VPS로 이동되는 경우, 및 0보다 더 큰 nuh_layer_id를 갖는 계층에 의해 참조되는, 0과 동일한 nuh_layer_id를 갖는 SPS에서의 표현 포맷 파라미터들이 무시된다고 특정되는 경우, 상이한 값들의 공간적 해상도들, 비트 깊이들, 또는 컬러 포맷들을 갖는 계층들이 동일한 SPS들을 참조하는 것이 가능하다. 다르게 말하면, 본 개시물의 몇몇 실시형태들에 따르면, 상이한 값들의 공간적 해상도들, 비트 깊이들, 또는 컬러 포맷들을 갖는 계층들은, 그 계층들을 위해 요구된 다른 SPS 파라미터들이 동일한 한, 동일한 SPS를 공유할 수 있다.

[0304] 몇몇 예들에서, 비디오 인코더 (20)는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신할 수도 있다. 예들의 경우, 비디오 인코더 (20)는, 1과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층임을 특정하고, 0과 동일한 경우, 계층 세트에서의 j-번째 계층이 i-번째 출력 계층 세트의 타겟 출력 계층이 아님을 특정하는 출력 계층 플래그 [i][j]를 송신할 수도 있다.

[0305] 도 6은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 6의 도시된 예에서, 비디오 디코더 (30)는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 VPS 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 수신할 수도 있다 (602).

[0306] 비디오 디코더 (30)는 엔트로피 인코딩된 선택스 엘리먼트에 앞서 비-엔트로피 인코딩된 계층 의존성 정보를 디코딩할 수도 있다 (604). 일 예에서, 비디오 디코더 (30)는 임의의 엔트로피 인코딩된 선택스 엘리먼트에 앞서 비-엔트로피 인코딩된 계층 의존성 정보를 디코딩할 수도 있다.

[0307] 비디오 디코더 (30)는 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 디코딩할 수도 있다 (606). 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타낼 수도 있다. 몇몇 예들에서, 계층 의존성 정보가 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타내는 경우, 계층 의존성 정보는 그 계층들 중 어떤 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지를 또한 나타낸다. 다르게 말하면, 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타내고 계층들 중 다른 계층에 대한 직접 참조 계층인 그 계층들 중 하나의 계층을 식별할 수도 있다. 몇몇 예들에서, 계층 의존성 정보는 0과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층이 아님을 특정하고, 1과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층일 수도 있음을 특정하는 direct_dependency_flag[i][j]를 포함한다. 화상을 디코딩하는 경우, 비디오 디코더 (30)는, 다른 정보도 있지만 무엇보다도, 계층 의존성 정보에 기초하여 층 간 참조 화상 세트를 도출할 수도 있고, 다른 정보도 있지만 무엇보다도, 층 간 참조 화상 세트에 기초하여 참조 화상 리스트를 추가로 도출할 수도 있고, 그 다음에 직접 참조 계층에서의 화상으로부터 층 간 예측을 사용하여 화상을 디코딩할 수도 있다.

[0308] 도 7은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다. 비디오 인코더 (20)는 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터를 인코딩할 수도 있다 (702). 비디오 인코더 (20)는 엔트로피 인코딩되는 VPS 확장물의 선택스 엘리먼트들보다 앞선 비디오 파라미터 세트 (VPS) 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 인코딩할 수도 있다 (704). 예를 들어, 비디오 인코더 (20)는 비-엔트로피 인코딩된 계층 의존성 정보에 기초하여 비디오 데이터의 계층들 중 하나 이상의 계층의 비디오 데이터

를 인코딩할 수도 있다. 계층 의존성 정보는 계층들 중 하나의 계층이 그 계층들 중 다른 계층에 대한 직접 참조 계층인지의 여부를 나타낼 수도 있다. 몇몇 예들에서, 비디오 인코더 (20)는 엔트로피 인코딩되는 VPS 확장물의 신택스 엘리먼트들보다 앞선 비디오 파라미터 세트 (VPS) 확장물 내의 포지션에서 비-엔트로피 인코딩된 계층 의존성 정보를 송신, 저장, 또는 저장되게 할 수도 있다.

[0309] 몇몇 예들에서, 계층 의존성 정보는 0과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층이 아님을 특정하고, 1과 동일한 경우, 인덱스 j를 갖는 계층이 인덱스 i를 갖는 계층에 대한 직접 참조 계층일 수도 있음을 특정하는 direct_dependency_flag[i][j]를 포함한다.

[0310] 도 8은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 8의 예에서, 하나를 초과하는 출력 계층 세트는 하나의 계층 세트에 대해 시그널링될 수도 있다. 따라서, 하나를 초과하는 출력 계층 세트가 하나의 계층 세트에 대해 시그널링되면, 비디오 디코더 (30) 가, 예컨대, 입력 인터페이스 (28)에 의해, 계층 세트에 대한 제 1 출력 계층 세트를 수신할 수도 있다 (802). 비디오 디코더 (30)는 그 계층 세트에 대한 제 2 출력 계층 세트를 또한 수신할 수도 있다 (804).

더욱이, 도 8의 예가 하나의 계층 세트에 대해 두 개의 출력 계층 세트를 도시하지만, 세 개의, 네 개의, 또는 심지어 더 많은 출력 계층 세트들이 계층 세트에 대해 시그널링될 수도 있다는 것이 이해될 것이다. 비디오 디코더 (30)는 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나의 출력 계층 세트에 대한 비디오 데이터를 디코딩할 수도 있다 (806).

[0311] 도 9는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 9의 예에서, 하나를 초과하는 출력 계층 세트는 하나의 계층 세트에 대해 시그널링될 수도 있다. 따라서, 비디오 인코더 (20)는 제 1 출력 계층 세트 및 제 2 출력 계층 세트 중 적어도 하나의 출력 계층 세트에 대한 비디오 데이터를 인코딩할 수도 있다 (902). 따라서, 출력 인터페이스 (22)는 비디오 인코더 (20)로부터의 인코딩된 데이터를 송신할 수도 있다. 송신되는 데이터는 계층 세트에 대한 제 1 출력 계층 세트 (904) 및 그 계층 세트에 대한 제 2 출력 계층 세트 (906)를 포함할 수도 있다. 몇몇 예들에서, 출력 인터페이스 (22)는 그 데이터를 입력 인터페이스 (28)로 송신할 수도 있다. 다른 예들에서, 출력 인터페이스 (22)는 그 데이터를 저장을 위해 저장 디바이스 (34)로 송신할 수도 있다.

[0312] 도 10은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 10의 예는, 잠재적으로는 표현 포맷이 엔트로피 디코딩 없이 액세스 가능한 방식으로 VPS에서의 표현 포맷의 시그널링을 도시한다. 다르게 말하면, VPS에서의 표현 포맷은 엔트로피 인코딩되지 않는다. 그것은, 예를 들어, 고정 길이 코딩될 수도 있다. 따라서, 비디오 디코더 (30)는 VPS 내의 비-엔트로피 인코딩된 표현 포맷을, 예컨대, 입력 인터페이스 (28)를 통해 수신할 수도 있다 (1002). 표현 포맷은 크로마 포맷, 상이한 컬러 평면들 (colour planes)이 따로따로 코딩되는지의 여부, 화상 폭, 화상 높이, 루마 비트 깊이, 및 크로마 비트 깊이 중 하나 이상을 포함할 수도 있다. 비디오 디코더 (30)는 VPS 내의 비-엔트로피 인코딩된 표현 포맷에 기초하여 비디오 데이터를 또한 디코딩할 수도 있다 (1004). 그 표현 포맷이 잠재적으로는 엔트로피 디코딩 없이 액세스 가능한 방식으로 VPS 내에 있기 때문에, 엔트로피 코딩을 수행하지 않는 디바이스들은 그 표현 포맷에 액세스할 수도 있으며, 예를 들어, 몇몇 경우들에서 미디어 인식 네트워크 엔티티들 (MANE들)은 엔트로피 코딩 디바이스를 갖지 않을 수도 있다. 비디오 디코더 (30)는, 잠재적으로는 본 개시물의 기법들에 따라 엔트로피 디코딩 없이 VPS에서의 표현 포맷을 디코딩할 수도 있고, 각각의 계층은 특정 표현 포맷에 연관될 수도 있다.

[0313] 도 11은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 11의 예는, 잠재적으로는 표현 포맷이 엔트로피 디코딩 없이 액세스 가능한 방식으로 VPS에서의 표현 포맷의 시그널링을 도시한다. 다르게 말하면, VPS에서의 표현 포맷은 엔트로피 인코딩되지 않는다. 그것은, 예를 들어, 고정 길이 코딩될 수도 있다. 따라서, 비디오 인코더 (20)는 VPS 내의 비-엔트로피 인코딩된 표현 포맷에 기초하여 비디오 데이터를 인코딩할 수도 있다 (1102). 표현 포맷은 크로마 포맷, 상이한 컬러 평면들이 따로따로 코딩되는지의 여부, 화상 폭, 화상 높이, 루마 비트 깊이, 및 크로마 비트 깊이 중 하나 이상을 포함할 수도 있다. 비디오 인코더 (20)는 VPS 내의 비-엔트로피 인코딩된 표현 포맷을 송신할 수도 있다 (1104). 비디오 인코더 (20)는 VPS에서의 표현 포맷을, 그 표현 포맷이 본 개시물의 기법들에 따라 엔트로피 디코딩 없이 액세스될 수 있고, 각각의 계층이 특정 표현 포맷과 연관될 수도 있는 방식으로 인코딩할 수도 있다.

[0314] 도 12는 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는

흐름도이다. 도 12의 예는 VPS에서의 계층마다의 시각적 신호 정보, 예컨대, video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs의 시그널링을 도시한다. 따라서, 비디오 디코더 (30)는, 각각의 계층이 시각적 신호 정보를 포함하는, 일련의 계층들을 포함하는 VPS를 수신할 수도 있다 (1002). 비디오 디코더 (30)는 VPS에서의 계층마다 시그널링된 수신된 시각적 신호 정보에 기초하여 비디오 데이터를 또한 디코딩할 수도 있다.

[0315] 도 13은 본원에서 설명되는 시스템들 및 방법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 도시하는 흐름도이다. 도 13의 예는 VPS에서의 계층마다의 시각적 신호 정보, 예컨대, video_format, video_full_range_flag, colour_primaries, transfer_characteristics, matrix_coeffs의 시그널링을 도시한다. 따라서, 비디오 인코더 (20)는 VPS에서의 계층마다 시그널링된 수신된 시각적 신호 정보에 기초하여 비디오 데이터를 인코딩할 수도 있다 (1302). 비디오 인코더 (20)는 각각의 계층이 일련의 계층들 중 각각 계층에 대한 시각적 신호 정보를 포함하는, 일련의 계층들을 포함하는 VPS를 송신할 수도 있다 (1304).

[0316] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 관독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 관독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 관독가능 저장 매체들, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방식으로, 컴퓨터 관독가능 매체들은 일반적으로 (1) 비일시적 (non-transitory) 인 유형의 컴퓨터 관독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 관독가능 매체를 포함할 수도 있다.

[0317] 비제한적인 예로, 이러한 컴퓨터 관독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 관독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털가입자 회선 (DSL), 또는 무선 기술들 이를테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다. 그러나, 컴퓨터 관독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비일시적 (non-transient), 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc)는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들로 써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 관독가능 매체들의 범위 내에 포함되어야 한다.

[0318] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 접적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 다른 동등한 접적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서"는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 말할 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec)으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0319] 본 개시물의 기법들은 무선 핸드셋, 접적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋)를 포함한 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어에 연계하여, 위에서 설명된 바와 같은 하

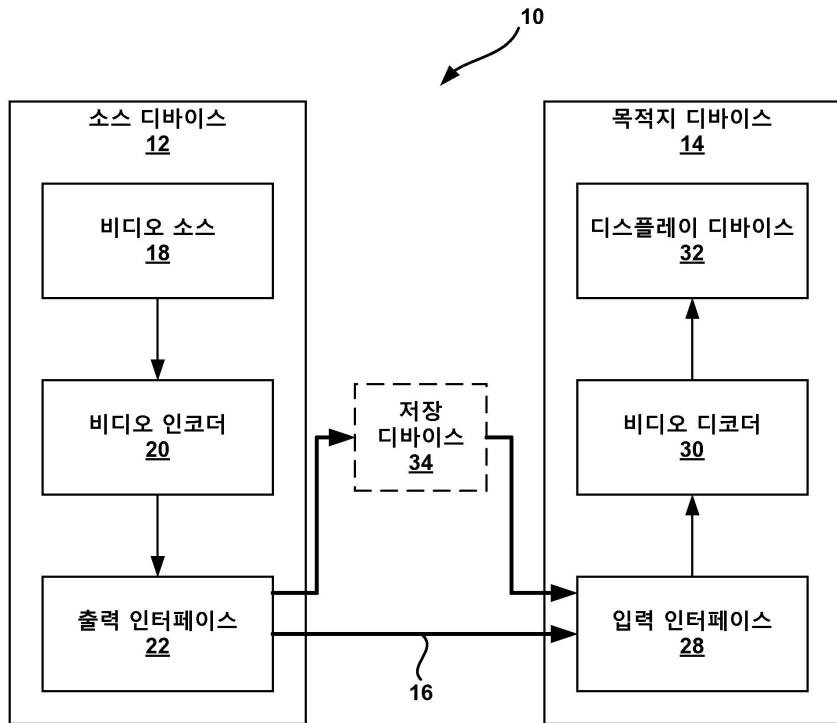
나 이상의 프로세서들을 포함한 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

[0320]

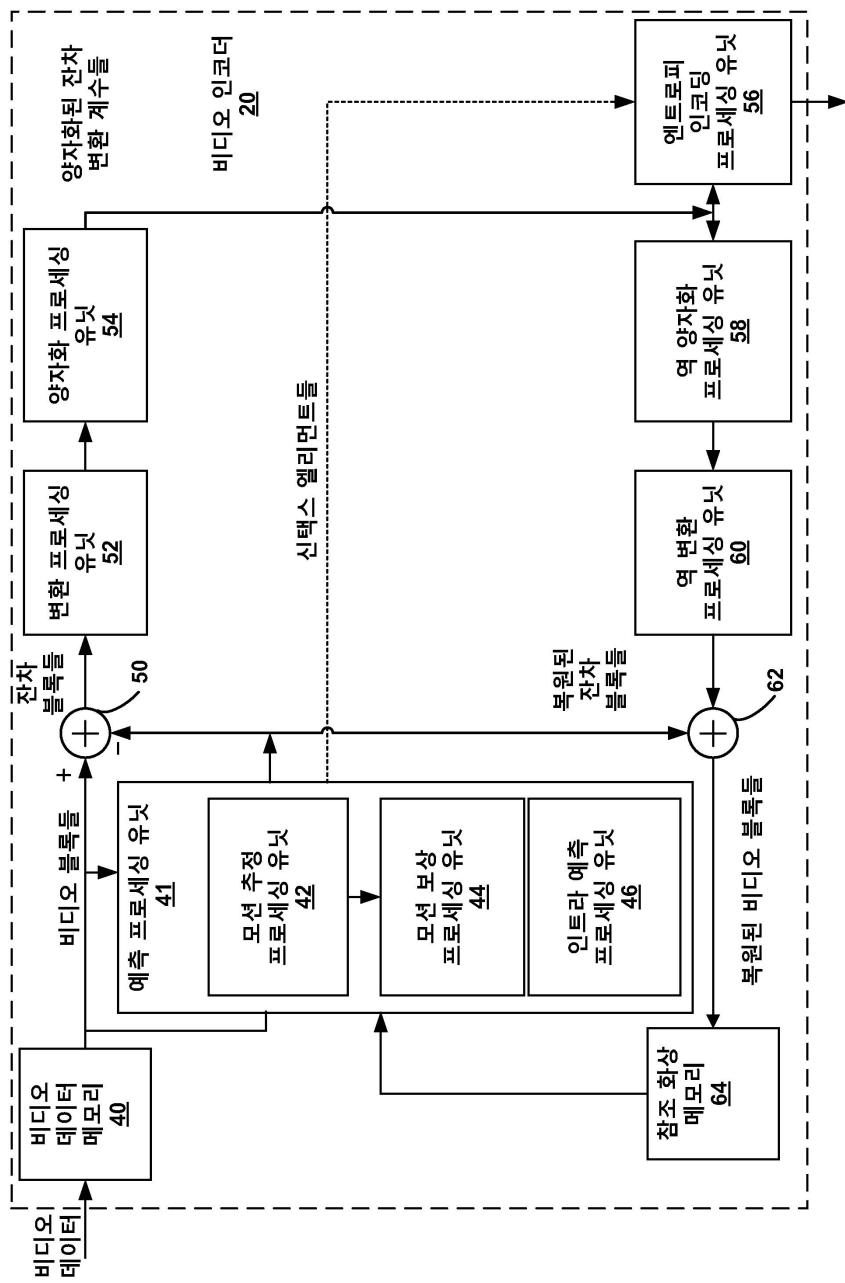
다양한 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

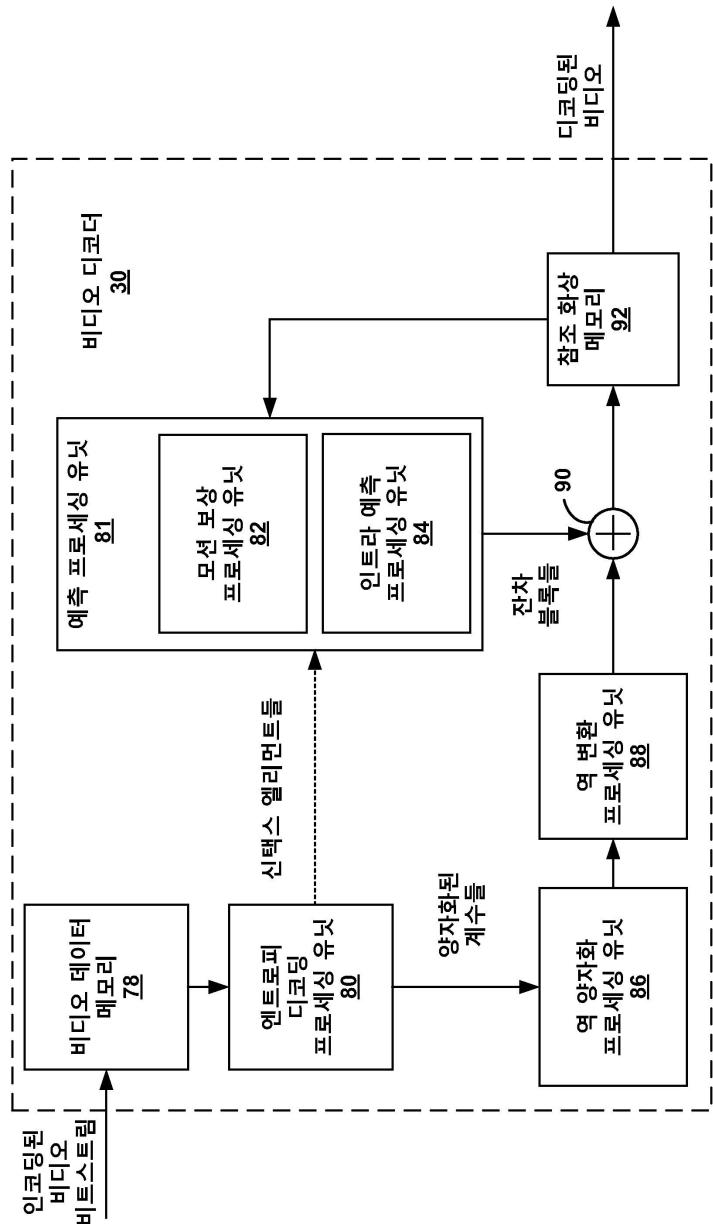
도면1



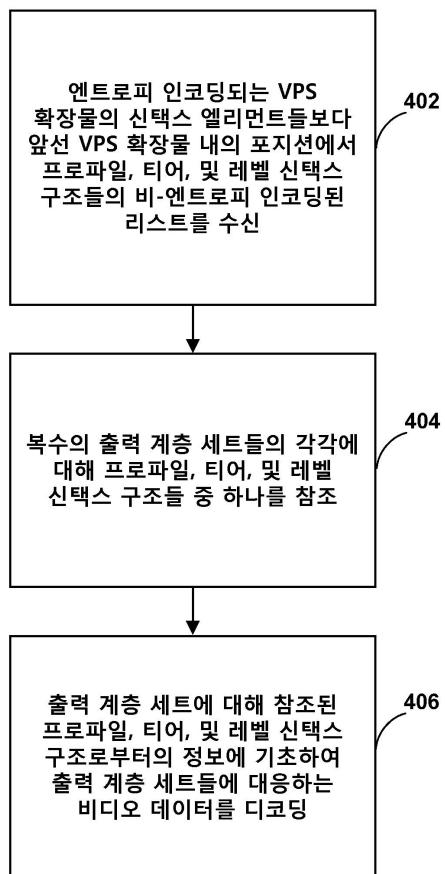
도면2



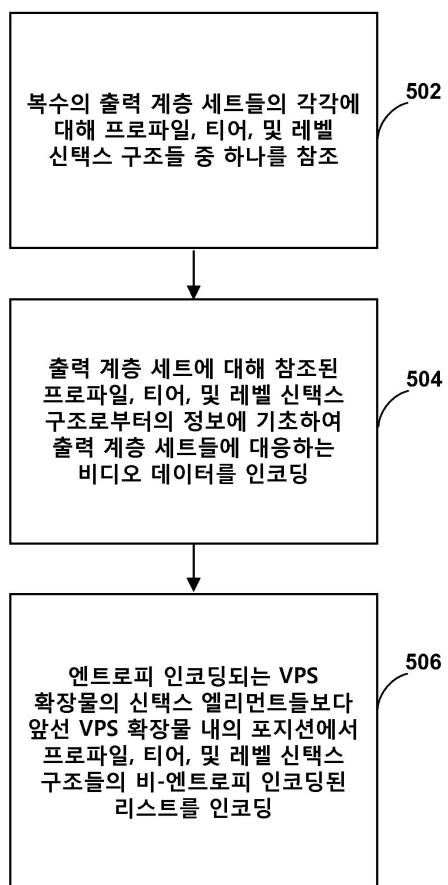
도면3



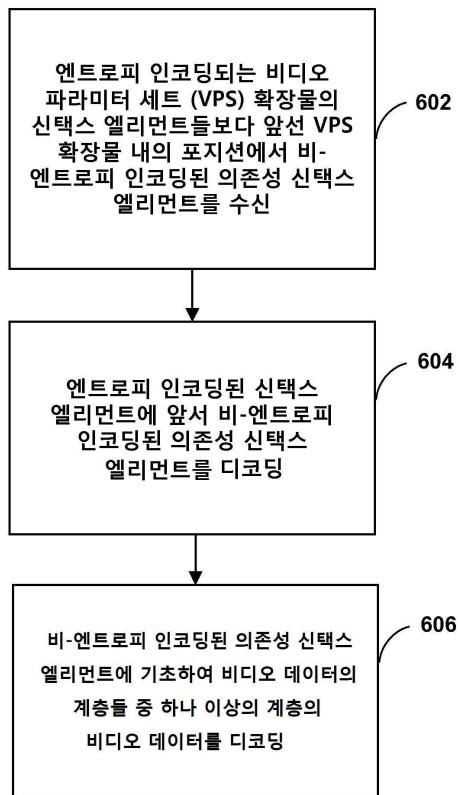
도면4



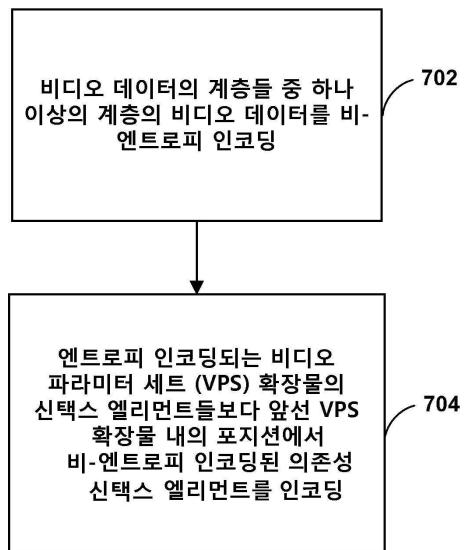
도면5



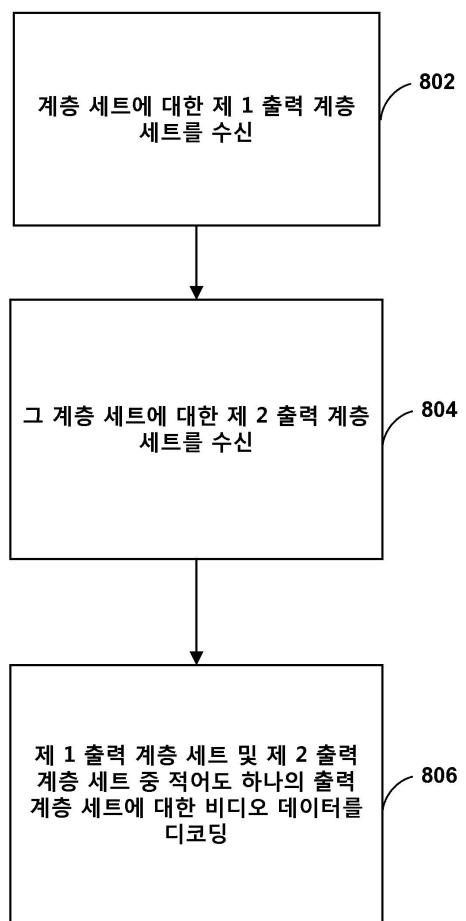
도면6



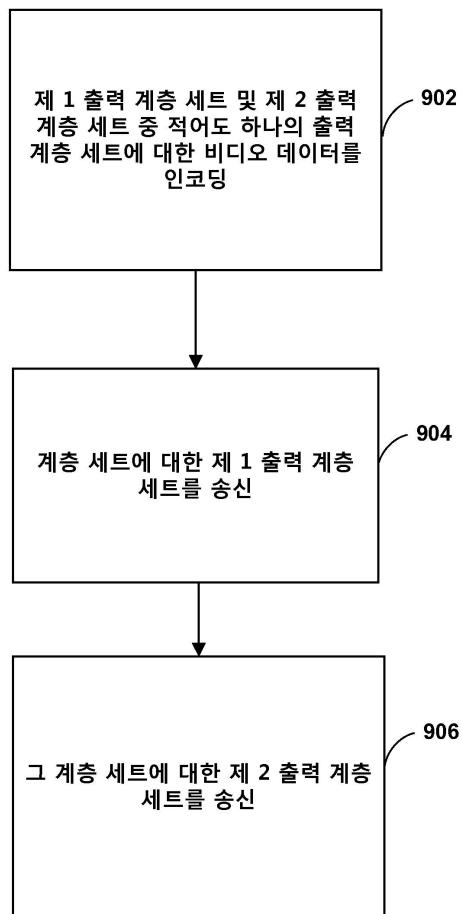
도면7



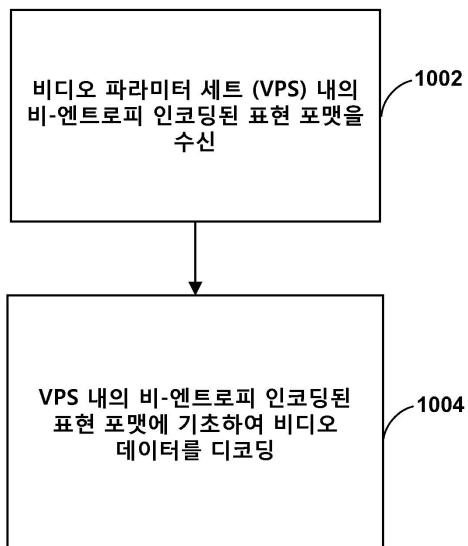
도면8



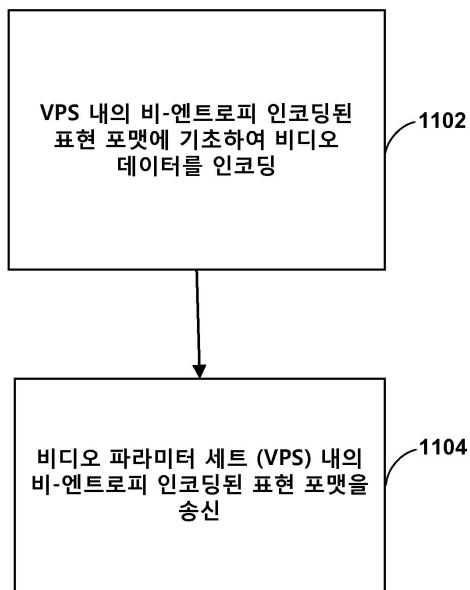
도면9



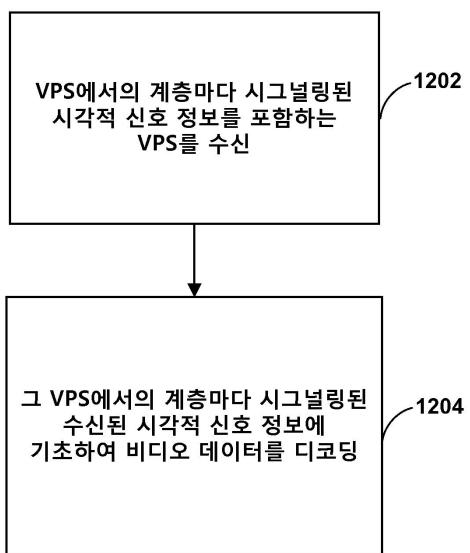
도면10



도면11



도면12



도면13

