



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2017년07월20일

(11) 등록번호 10-1760165

(24) 등록일자 2017년07월14일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/70 (2014.01) *HO4N 19/30* (2014.01)
- (21) 출원번호 10-2014-7022571
- (22) 출원일자(국제) 2013년01월11일
심사청구일자 2017년01월06일
- (85) 번역문제출일자 2014년08월12일
- (65) 공개번호 10-2014-0120336
- (43) 공개일자 2014년10월13일
- (86) 국제출원번호 PCT/US2013/021227
- (87) 국제공개번호 WO 2013/106705
국제공개일자 2013년07월18일
- (30) 우선권주장
13/738,377 2013년01월10일 미국(US)
(뒷면에 계속)

- (56) 선행기술조사문현
US20100208796 A1
US20130094774 A1
KR1020080088633 A

- (73) 특허권자
웰컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
천 잉
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
왕 예-쿠이
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- (74) 대리인
특허법인코리아나

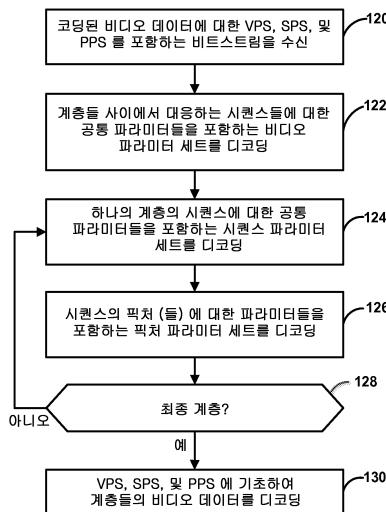
전체 청구항 수 : 총 45 항

심사관 : 조우연

(54) 발명의 명칭 비디오 코딩을 위한 파라미터 세트들 및 NAL 단위 헤더들의 코딩

(57) 요 약

하나의 예에서, 비디오 인코더 또는 비디오 디코더와 같은 비디오 코더는 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하는 것으로서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하는, 상기 비디오 파라미터 세트를 코딩하고, VPS 에 적어도 부분적으로 기초하여 비디오 데이터 하나 이상의 계층들을 코딩하도록 구성된다. 비디오 코더는 고효율 비디오 코딩, 멀티뷰 비디오 코딩, 스케일러블 비디오 코딩, 또는 다른 비디오 코딩 표준들 또는 비디오 코딩 표준들의 확장들을 준수하는 비디오 데이터에 대한 VPS 를 코딩할 수도 있다. VPS 는 다양한 상이한 계층들 (예를 들어, 뷰들, 품질 계층들 등) 내의 비디오 데이터의 대응하는 시퀀스들에 대한 파라미터들을 특정하는 데이터를 포함할 수도 있다. VPS 의 파라미터들은 대응하는 비디오 데이터가 어떻게 코딩되는지에 대한 표시들을 제공할 수도 있다.

대 표 도 - 도8

(30) 우선권주장

61/586,777 2012년01월14일 미국(US)

61/587,070 2012년01월16일 미국(US)

61/588,629 2012년01월19일 미국(US)

61/637,195 2012년04월23일 미국(US)

61/637,774 2012년04월24일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

상기 방법은,

비디오 데이터의 복수의 계층들을 포함하는 비트스트림에 대한 비디오 파라미터 세트 (video parameter set; VPS) 를 코딩하는 단계로서, 상기 비디오 데이터의 복수의 계층들의 각각은 상기 VPS 를 참조하는, 상기 VPS 를 코딩하는 단계를 포함하고,

상기 VPS 를 코딩하는 단계는,

상기 비디오 데이터의 복수의 계층들 중 적어도 하나의 계층에서 재순서화될 프레임들의 수를 표시하는 상기 VPS 의 데이터를 코딩하는 단계,

상기 비디오 데이터의 복수의 계층들의 디코딩 동안에 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 에 저장될 픽처들의 수를 표시하는 상기 VPS 의 데이터를 코딩하는 단계,

상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 시간적 계층들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하는 단계;

상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 뷰들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하는 단계;

특성 표시자에 대한 샘플 인덱스의 맵핑을 정의하는 정보를 코딩하는 단계로서, 상기 맵핑을 정의하는 정보를 코딩하는 것은, 복수의 종속성 인덱스들의 각각에 대한 각각의 공간 해상도, 복수의 시간적 인덱스들의 각각에 대한 프레임 레이트, 또는 복수의 뷰 인덱스들의 각각에 대한 뷰 식별자 중 하나 이상을 코딩하는 것을 포함하는, 상기 특성 표시자에 대한 샘플 인덱스의 맵핑을 정의하는 정보를 코딩하는 단계; 및

상기 VPS 에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 VPS 를 코딩하는 단계는, 가상 참조 디코더 (hypothetical reference decoder; HRD) 파라미터들의 하나 이상의 세트들을 표시하는 상기 VPS 의 데이터를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 VPS 를 코딩하는 단계는, 상기 VPS 가 대응하는 표준을 넘어서는 확장을 포함하는지 여부를 표시하는 상기 VPS 의 데이터와, 상기 VPS 가 상기 확장을 포함할 때, 상기 확장에 대한 데이터를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 비디오 데이터의 복수의 계층들을 코딩하는 단계는, 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 에 따라 상기 비디오 데이터의 복수의 계층들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 비디오 데이터의 복수의 계층들을 코딩하는 단계는, 멀티뷰 비디오 코딩 (Multiview Video Coding; MVC) 또는 스케일러블 비디오 코딩 (Scalable Video Coding; SVC) 중 적어도 하나에 따라 상기 비디오 데이터의 복수의 계층들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 VPS 를 코딩하는 단계는,

상기 비디오 데이터의 복수의 계층들의 하나 이상의 차원들에 대하여,

상기 비디오 데이터의 복수의 계층들에서의 우선순위 계층들의 수,

상기 비디오 데이터의 복수의 계층들에서의 종속성 계층들의 수,

상기 비디오 데이터의 복수의 계층들에서의 시간적 계층들의 수, 또는

상기 비디오 데이터의 복수의 계층들에서의 상기 종속성 계층들 중 임의의 종속성 계층에 대한 품질 계층들의 최대 수

중 하나 이상을 특정하는 정보를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 7

제 6 항에 있어서,

상기 비디오 데이터의 복수의 계층들의 서브세트가 동일한 공간 해상도 및 동일한 비트 심도를 가질 때, 상기 서브세트의 상기 계층들의 각각은 상기 종속성 계층들 중 상이한 하나에 대응하는, 비디오 데이터를 코딩하는 방법.

청구항 8

제 7 항에 있어서,

상기 맵핑을 정의하는 정보를 코딩하는 단계는, 상기 비디오 데이터의 복수의 계층들의 차원의 특성을 정의하는 특성 표시자가 제로로부터 샘플 차원 카운터 마이너스 1 까지의 인덱스 범위 내에 있지 않을 때, 복수의 특성 인덱스들의 각각에 대한 각각의 특성 표시자를 특정하는 정보를 코딩하는 단계를 포함하고, 상기 카운터는 인덱스에 의해 정의되는, 비디오 데이터를 코딩하는 방법.

청구항 9

제 7 항에 있어서,

상기 맵핑을 정의하는 정보를 코딩하는 단계는,

복수의 비트 심도 인덱스들의 각각에 대한 휘도 및 색도에 대한 특정 심도 값들의 쌍, 또는

복수의 색도 샘플링 포맷들의 각각에 대한 특정 색도 샘플링 포맷 표시자

중 하나 이상을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 10

제 1 항에 있어서,

상기 VPS 를 코딩하는 단계는, 제어 파라미터들 및 하나 이상의 툴 인에이블/디스에이블 플래그들을 정의하는 정보를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 11

제 10 항에 있어서,

상기 제어 파라미터들 및 상기 하나 이상의 툴 인레이블/디스레이블 플래그들은,

pcm_bit_depth_luma_minus1,

pcm_bit_depth_chroma_minus1,

loop_filter_across_slice_flag,

pcm_loop_filter_disable_flag,

temporal_id_nesting_flag,

하나 이상의 타일 관련 싱크스 엘리먼트들,

chroma_pred_from_luma_enabled_flag,

sample_adaptive_offset_enabled_flag,

adaptive_loop_filter_enabled_flag, 또는

inter_4x4_enabled_flag

중 하나 이상을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 12

제 1 항에 있어서,

상기 VPS 를 코딩하는 단계는, 하나 이상의 동작점 디스크립터 (descriptor) 들을 정의하는 정보를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 13

제 12 항에 있어서,

상기 하나 이상의 동작점 디스크립터들을 정의하는 상기 정보를 코딩하는 단계는,

최대 동작점들의 수,

상이한 계층들 또는 뷰들 사이의 종속성,

동작점들의 각각에 대한 프로파일 및 레벨, 또는

상기 동작점들의 각각에 대한 비트 레이트

중 하나 이상을 정의하는 정보를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 14

제 1 항에 있어서,

상기 비디오 데이터의 복수의 계층들의 각각에 대한 각각의 계층화된 시퀀스 파라미터 세트 (layered sequence parameter set; LPS) 를 코딩하는 단계를 더 포함하고,

상기 VPS 에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하는 단계는, 상기 VPS 및 각각의 LPS 에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 15

제 14 항에 있어서,

상기 비디오 데이터의 복수의 계층들의 각각에 대한 각각의 LPS 들을 코딩하는 단계는, 각각의 차원에 대하여, 각각의 차원에 대한 인덱스를 표시하는 샘플 차원 표시를 정의하는 정보를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 16

제 14 항에 있어서,

상기 비디오 데이터의 복수의 계층들의 각각에 대한 각각의 LPS 들을 코딩하는 단계는, 제어 파라미터들 및 툴 인에이블/디스에이블 플래그들을 정의하는 정보를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 17

제 16 항에 있어서,

상기 제어 파라미터들 및 상기 하나 이상의 툴 인에이블/디스에이블 플래그들은,

```
pcm_bit_depth_luma_minus1,  
pcm_bit_depth_chroma_minus1,  
loop_filter_across_slice_flag,  
pcm_loop_filter_disable_flag,
```

하나 이상의 타일 관련 선택스 엘리먼트들,

```
chroma_pred_from_luma_enabled_flag,  
sample_adaptive_offset_enabled_flag,
```

adaptive_loop_filter_enabled_flag, 또는

코딩 단위 (coding unit; CU) 계층구조

중 하나 이상을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 18

제 14 항에 있어서,

상기 비디오 데이터의 복수의 계층들의 각각에 대한 각각의 LPS 들을 코딩하는 단계는, 공통 픽처 파라미터 세트 (picture parameter set; PPS) 를 참조하는 슬라이스, 슬라이스들의 그룹, 픽처, 또는 수개의 픽처들 중 적어도 하나에 적용하는 하나 이상의 다른 파라미터 세트들의 정보를 정의하는 정보를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 19

제 1 항에 있어서,

하나 이상의 픽처 파라미터 세트 (PPS) 들이 상기 VPS 를 참조하지 않고, 상기 비디오 데이터의 복수의 계층들의 계층화된 시퀀스 파라미터 세트 (LPS) 들을 참조하지 않도록, 상기 PPS 들을 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 20

제 19 항에 있어서,

상기 VPS 에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하는 단계는, 상기 PPS 들 중 하나의 PPS 의 선택스 엘리먼트가 상기 VPS 또는 상기 LPS 들 중 각각의 하나의 LPS 와 충돌할 때, 상기 PPS 들 중 하나의 PPS 의 선택스 엘리먼트에 기초하여 상기 비디오 데이터의 복수의 계층들 중 대응하는 하나의 계층을 코딩하도록, 상기 VPS, 상기 PPS 들, 및 상기 LPS 들에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 21

제 1 항에 있어서,

상기 비디오 데이터의 복수의 계층들에 대하여 상기 VPS 를 포함하는 모든 파라미터 세트들을 함께 그룹화하는

그룹화 파라미터 세트 (grouping parameter set; GPS) 를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 22

제 21 항에 있어서,

상기 GPS 를 코딩하는 단계는, 상기 GPS 의 식별자를 정의하는 정보를 코딩하는 단계를 포함하고,

상기 방법은, 상기 GPS 의 상기 식별자에 대응하는 슬라이스 헤더의 정보를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 23

제 1 항에 있어서,

상기 비디오 데이터의 복수의 계층들을 코딩하는 단계는, 상기 비디오 데이터의 복수의 계층들을 디코딩하는 단계를 포함하고,

상기 VPS 를 코딩하는 단계는, 상기 VPS 를 파싱하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 24

제 1 항에 있어서,

상기 비디오 데이터의 복수의 계층들을 코딩하는 단계는, 상기 비디오 데이터의 복수의 계층들을 인코딩하는 단계를 포함하고,

상기 VPS 를 코딩하는 단계는, 상기 VPS 를 구성하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 25

비디오 데이터를 코딩하기 위한 디바이스로서,

상기 디바이스는 비디오 코더를 포함하고,

상기 비디오 코더는,

비디오 데이터의 복수의 계층들을 포함하는 비트스트림에 대한 비디오 파라미터 세트 (video parameter set; VPS) 를 코딩하는 것으로서, 상기 비디오 데이터의 복수의 계층들의 각각은 상기 VPS 를 참조하는, 상기 VPS 를 코딩하는 것을 행하도록 구성되고,

상기 VPS 를 코딩하는 것을 행하도록 구성된 상기 비디오 코더는,

상기 비디오 데이터의 복수의 계층들 중 적어도 하나의 계층에서 재순서화될 프레임들의 수를 표시하는 상기 VPS 의 데이터를 코딩하고,

상기 비디오 데이터의 복수의 계층들의 디코딩 동안에 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 에 저장될 픽처들의 수를 표시하는 상기 VPS 의 데이터를 코딩하며,

상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 시간적 계층들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하고;

상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 뷰들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하고;

특성 표시자에 대한 샘플 인덱스의 맵핑을 정의하는 정보를 코딩하고; 그리고

상기 VPS 에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하도록 구성되며,

상기 맵핑을 정의하는 정보를 코딩하기 위해, 상기 비디오 코더는, 복수의 종속성 인덱스들의 각각에 대한 각각의 공간 해상도, 복수의 시간적 인덱스들의 각각에 대한 프레임 레이트, 또는 복수의 뷰 인덱스들의 각각에 대한 뷰 식별자 중 하나 이상을 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 26

제 25 항에 있어서,

상기 VPS 를 코딩하는 것을 행하도록 구성된 상기 비디오 코더는, 가상 참조 디코더 (HRD) 파라미터들의 하나 이상의 세트들을 표시하는 상기 VPS 의 데이터를 코딩하도록 더 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 27

제 25 항에 있어서,

상기 VPS 를 코딩하는 것을 행하도록 구성된 상기 비디오 코더는, 상기 VPS 가 대응하는 표준을 넘어서는 확장을 포함하는지 여부를 표시하는 상기 VPS 의 데이터와, 상기 VPS 가 상기 확장을 포함할 때, 상기 확장에 대한 데이터를 코딩하도록 더 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 28

제 25 항에 있어서,

상기 비디오 코더는, 고효율 비디오 코딩 (HEVC), 멀티뷰 비디오 코딩 (MVC), 및 스케일러블 비디오 코딩 (SVC) 중 하나에 따라 상기 비디오 데이터의 복수의 계층들을 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 29

제 25 항에 있어서,

상기 비디오 코더는 비디오 디코더를 포함하고, 상기 디바이스는 상기 비디오 데이터를 표시하도록 구성된 디스플레이를 더 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 30

제 25 항에 있어서,

상기 비디오 코더는 비디오 인코더를 포함하고, 상기 디바이스는 상기 비디오 데이터를 생성하도록 구성된 카메라를 더 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 31

제 25 항에 있어서,

상기 디바이스는,

집적 회로;

마이크로프로세서; 또는

상기 비디오 코더를 포함하는 모바일 무선 통신 디바이스

중 적어도 하나를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 32

제 25 항에 있어서,

상기 VPS 를 코딩하는 것을 행하도록 구성된 상기 비디오 코더는, 하나 이상의 동작점 디스크립터 (descriptor) 들을 정의하는 정보를 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 33

제 32 항에 있어서,

하나 이상의 동작점 디스크립터들을 정의하는 정보를 코딩하기 위해, 상기 비디오 코더는,

최대 동작점들의 수,
 상이한 계층들 또는 뷰들 사이의 종속성,
 동작점들의 각각에 대한 프로파일 및 레벨, 또는
 상기 동작점들의 각각에 대한 비트 레이트
 중 하나 이상을 정의하는 정보를 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 34

비디오 데이터를 코딩하기 위한 디바이스로서,
 상기 디바이스는,
 비디오 데이터의 복수의 계층들을 포함하는 비트스트림에 대한 비디오 파라미터 세트 (video parameter set; VPS) 를 코딩하는 수단으로서, 상기 비디오 데이터의 복수의 계층들의 각각은 상기 VPS 를 참조하는, 상기 VPS 를 코딩하는 수단을 포함하고,
 상기 VPS 를 코딩하는 수단은,
 상기 비디오 데이터의 복수의 계층들 중 적어도 하나의 계층에서 재순서화될 프레임들의 수를 표시하는 상기 VPS 의 데이터를 코딩하는 수단,
 상기 비디오 데이터의 복수의 계층들의 디코딩 동안에 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 에 저장될 픽처들의 수를 표시하는 상기 VPS 의 데이터를 코딩하는 수단,
 상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 시간적 계층들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하는 수단;
 상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 뷰들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하는 수단;
 특성 표시자에 대한 샘플 인덱스의 맵핑을 정의하는 정보를 코딩하는 수단; 및
 상기 VPS 에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하는 수단을 포함하며,
 상기 맵핑을 정의하는 정보를 코딩하는 수단은, 복수의 종속성 인덱스들의 각각에 대한 각각의 공간 해상도, 복수의 시간적 인덱스들의 각각에 대한 프레임 레이트, 또는 복수의 뷰 인덱스들의 각각에 대한 뷰 식별자 중 하나 이상을 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 35

제 34 항에 있어서,
 상기 VPS 를 코딩하는 수단은, 가상 참조 디코더 (HRD) 파라미터들의 하나 이상의 세트들을 표시하는 상기 VPS 의 데이터를 코딩하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 36

제 34 항에 있어서,
 상기 VPS 를 코딩하는 수단은, 상기 VPS 가 대응하는 표준을 넘어서는 확장을 포함하는지 여부를 표시하는 상기 VPS 의 데이터와, 상기 VPS 가 상기 확장을 포함할 때, 상기 확장에 대한 데이터를 코딩하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 37

제 34 항에 있어서,
 상기 VPS 를 코딩하는 수단은, 고효율 비디오 코딩 (HEVC), 멀티뷰 비디오 코딩 (MVC), 및 스케일러블 비디오 코딩 (SVC) 중 하나에 따라 상기 비디오 데이터의 복수의 계층들을 코딩하는 수단을 포함하는, 비디오 데이터를

코딩하기 위한 디바이스.

청구항 38

제 34 항에 있어서,

상기 VPS 를 코딩하는 수단은, 하나 이상의 동작점 디스크립터 (descriptor) 를 정의하는 정보를 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 39

제 38 항에 있어서,

상기 하나 이상의 동작점 디스크립터들을 정의하는 정보를 코딩하는 수단은,

최대 동작점들의 수,

상이한 계층들 또는 뷰들 사이의 종속성,

동작점들의 각각에 대한 프로파일 및 레벨, 또는

상기 동작점들의 각각에 대한 비트 레이트

중 하나 이상을 정의하는 정보를 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 40

명령들을 저장한 비일시적 컴퓨터 관독가능 저장 매체로서,

상기 명령들은, 실행되는 경우, 프로세서로 하여금:

비디오 데이터의 복수의 계층들을 포함하는 비트스트림에 대한 비디오 파라미터 세트 (video parameter set; VPS) 를 코딩하는 것으로서, 상기 비디오 데이터의 복수의 계층들의 각각은 상기 VPS 를 참조하는, 상기 VPS 를 코딩하는 것을 행하게 하고,

상기 프로세서로 하여금 상기 VPS 를 코딩하는 것을 행하게 하는 상기 명령들은,

상기 프로세서로 하여금,

상기 비디오 데이터의 복수의 계층들 중 적어도 하나의 계층에서 재순서화될 프레임들의 수를 표시하는 상기 VPS 의 데이터를 코딩하게 하고,

상기 비디오 데이터의 복수의 계층들의 디코딩 동안에 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 에 저장될 픽처들의 수를 표시하는 상기 VPS 의 데이터를 코딩하게 하며,

상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 시간적 계층들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하게 하고;

상기 비디오 데이터의 복수의 계층들을 포함하는 상기 비트스트림에서 뷰들의 최대 수를 표시하는 상기 VPS 의 데이터를 코딩하게 하고;

특성 표시자에 대한 샘플 인덱스의 맵핑을 정의하는 정보를 코딩하게 하고; 그리고

상기 VPS 에 적어도 부분적으로 기초하여 상기 비디오 데이터의 복수의 계층들을 코딩하게 하는 명령들을 포함하며,

상기 프로세서로 하여금 상기 맵핑을 정의하는 정보를 코딩하게 하는 상기 명령들은, 실행되는 경우 상기 프로세서로 하여금, 복수의 종속성 인덱스들의 각각에 대한 각각의 공간 해상도, 복수의 시간적 인덱스들의 각각에 대한 프레임 레이트, 또는 복수의 뷰 인덱스들의 각각에 대한 뷰 식별자 중 하나 이상을 코딩하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 관독가능 저장 매체.

청구항 41

제 40 항에 있어서,

상기 프로세서로 하여금 상기 VPS 를 코딩하는 것을 행하게 하는 상기 명령들은, 실행되는 경우 상기 프로세서로 하여금, 가상 참조 디코더 (HRD) 파라미터들의 하나 이상의 세트들을 표시하는 상기 VPS 의 데이터를 코딩하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 42

제 40 항에 있어서,

상기 프로세서로 하여금 상기 VPS 를 코딩하는 것을 행하게 하는 상기 명령들은, 실행되는 경우 상기 프로세서로 하여금, 상기 VPS 가 대응하는 표준을 넘어서는 확장을 포함하는지 여부를 표시하는 상기 VPS 의 데이터와, 상기 VPS 가 상기 확장을 포함할 때, 상기 확장에 대한 데이터를 코딩하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 43

제 40 항에 있어서,

상기 프로세서로 하여금 상기 비디오 데이터의 복수의 계층들을 코딩하게 하는 상기 명령들은, 실행되는 경우 상기 프로세서로 하여금, 고효율 비디오 코딩 (HEVC), 멀티뷰 비디오 코딩 (MVC), 및 스케일러블 비디오 코딩 (SVC) 중 하나에 따라 상기 비디오 데이터의 복수의 계층들을 코딩하게 하는 명령들을 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 44

제 40 항에 있어서,

상기 프로세서로 하여금 상기 VPS 를 코딩하는 것을 행하게 하는 상기 명령들은, 실행되는 경우 상기 프로세서로 하여금, 하나 이상의 동작점 디스크립터 (descriptor) 들을 정의하는 정보를 코딩하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 45

제 44 항에 있어서,

상기 프로세서로 하여금 상기 하나 이상의 동작점 디스크립터들을 정의하는 정보를 코딩하게 하는 상기 명령들은,

실행되는 경우 상기 프로세서로 하여금,

최대 동작점들의 수,

상이한 계층들 또는 뷰들 사이의 종속성,

동작점들의 각각에 대한 프로파일 및 레벨, 또는

상기 동작점들의 각각에 대한 비트 레이트

중 하나 이상을 정의하는 정보를 코딩하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 46

삭제

청구항 47

삭제

청구항 48

삭제

청구항 49

삭제

청구항 50

삭제

청구항 51

삭제

발명의 설명

기술 분야

[0001]

본 출원은:

[0002]

2012년 1월 14일자로 출원된 미국 가출원 제 61/586,777 호;

[0003]

2012년 1월 16일자로 출원된 미국 가출원 제 61/587,070 호;

[0004]

2012년 1월 19일자로 출원된 미국 가출원 제 61/588,629 호;

[0005]

2012년 4월 23일자로 출원된 미국 가출원 제 61/637,195 호; 및

[0006]

2012년 4월 24일자로 출원된 미국 가출원 제 61/637,774 호의 이익을 주장한 것이고, 이 출원들 각각은 그 각각의 전체에 있어서 참조를 위해 본원에 포함된다.

[0007]

본 개시물은 비디오 코딩에 관한 것이다.

배경 기술

[0008]

디지털 비디오 기능들은, 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말 (personal digital assistant; PDA) 들, 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, 진보된 비디오 코딩 (Advanced Video Coding; AVC), 현재 개발 중에 있는 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, 및 이러한 표준들의 확장들에 의해 정의된 표준들에 설명된 것들과 같은 비디오 코딩 기술들을 구현한다. 당면한 HEVC 표준의 최근의 초안은 http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G1103-v3.zip에서 입수 가능하다. 비디오 디바이스들은 이러한 비디오 코딩 기술들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신하고, 수신하고, 인코딩하고, 디코딩하고, 및/또는 저장할 수도 있다.

[0009]

비디오 코딩 기술들은 비디오 시퀀스들에 고유한 중복성을 감소시키거나 제거하기 위해 공간적 (인트라-픽처) 예측 및/또는 시간간 (인터-픽처) 예측을 포함한다. 블록-기반 비디오 코딩을 위하여, 비디오 슬라이스 (예를 들어, 비디오 프레임 또는 비디오 프레임의 일부분) 는 비디오 블록들로 파티셔닝 (partitioning) 될 수도 있으며, 비디오 블록들은 트리블록들, 코딩 단위 (coding unit; CU) 들 및/또는 코딩 노드들이라고 또한 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스 내의 비디오 블록들은 동일한 픽처 내의 이웃하는 블록들 내의 참조 샘플들에 대한 공간적 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽처 내의 이웃하는 블록들 내의 참조 샘플들에 대한 공간적 예측, 또는 다른 참조 픽처들 내의 참조 샘플들에 대한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들이라고 지칭될 수도 있고, 참조 픽처들은 참조 프레임들이라고 지칭될 수도 있다.

[0010]

공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록으로 귀착된다. 잔차 데이터는 코딩될 원래의 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 지시하는 모션 벡터와, 코딩된 블록 및 예측 블록 사이의 차이를 표시하는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드와 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위하여, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어 잔차 변환 계수들로 귀착될 수도 있고, 그 후 이

들은 양자화될 수도 있다. 초기에 2 차원 어레이로 배치된 양자화된 변환 계수들은 변환 계수들의 1 차원 벡터를 생성하기 위해 스캐닝될 수도 있고, 엔트로피 코딩은 훨씬 더 많은 압축을 달성하기 위해 적용될 수도 있다.

발명의 내용

과제의 해결 수단

- [0011] 대체로, 본 개시물은 비디오 코딩을 위한 파라미터 세트들 및 네트워크 추상화 계층 (network abstraction layer; NAL) 단위들을 코딩하기 위한 기술들을 설명한다. 이 기술들은 스케일러블 비디오 코딩 (scalable video coding; SVC) 비디오 데이터 및 멀티뷰 비디오 코딩 (multiview video coding; MVC) 비디오 데이터뿐만 아니라, 2 차원 비디오 데이터와 같은 단일-계층 코딩된 데이터에 적용될 수도 있다. 따라서, 파라미터 세트들 및 NAL 단위들은 다양한 타입들의 비디오 데이터 사이에서 상호 호환가능할 수도 있다. 예를 들어, 비디오 인코더 또는 비디오 디코더와 같은 비디오 코더는 비디오 데이터의 하나 이상의 계층들에 대한 파라미터들을 정의하는 비디오 파라미터 세트 (video parameter set; VPS) 를 코딩할 수도 있다. 계층들은 예를 들어, SVC 계층들 (다양한 프레임 레이트들, 공간 해상도들, 및/또는 품질 레벨들을 가짐) 및/또는 MVC 데이터의 뷰들 (예를 들어, 수평축 주위의 다양한 카메라 관점들로부터 캡처된 장면의 이미지들의 시퀀스들) 에 대응할 수도 있다.
- [0012] 하나의 예에서, 비디오 데이터를 코딩하는 방법은, 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하는 단계로서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하는, 그 비디오 파라미터 세트를 코딩하는 단계, 및 VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩하는 단계를 포함한다.
- [0013] 또 다른 예에서, 비디오 데이터를 코딩하는 디바이스는 비디오 인코더 또는 비디오 디코더와 같은 비디오 코더를 포함하고, 그 비디오 코더는, 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하는 것으로서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하는, 그 비디오 파라미터 세트를 코딩하고, VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩하도록 구성된다.
- [0014] 또 다른 예에서, 비디오 데이터를 코딩하는 디바이스는, 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하는 수단으로서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하는, 그 비디오 파라미터 세트를 코딩하는 수단; 및 VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩하는 수단을 포함한다.
- [0015] 또 다른 예에서, 컴퓨터 관독가능 저장 매체는 명령들을 저장하고, 그 명령들은, 실행될 경우, 프로세서로 하여금, 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하게 하는 것으로서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하는, 그 비디오 파라미터 세트를 코딩하게 하고, VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩하게 한다.
- [0016] 하나 이상의 예들의 세부 사항들은 첨부한 도면들 및 아래의 설명에서 기재된다. 다른 특징들, 목적들, 및 장점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백할 것이다.

도면의 간단한 설명

- [0017] 도 1 은 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 네트워크 추상화 계층 (NAL) 단위들을 코딩하기 위한 기술들을 사용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.
- 도 2 는 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 NAL 단위들을 코딩하기 위한 기술들을 구현할 수도 있는 비디오 인코더 (20) 의 예를 예시하는 블록도이다.
- 도 3 은 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 NAL 단위들을 코딩하기 위한 기술들을 구현할 수도 있는 비디오 디코더 (30) 의 예를 예시하는 블록도이다.
- 도 4 는 일 예의 MVC 예측 패턴을 예시하는 개념적인 도면이다.
- 도 5 는 비디오 파라미터 세트 (video parameter set; VPS) 및 다양한 계층 파라미터 세트 (layer parameter set; LPS) 들을 예시하는 개념적인 도면이다.

도 6 은 일 예의 그룹화 파라미터 세트 (grouping parameter set; GPS) 와, 다른 파라미터 세트들을 갖는 GPS 및 슬라이스 헤더들의 관계들을 예시하는 개념적인 도면이다.

도 7 은 본 개시물의 기술들에 따라 비디오 데이터를 인코딩하기 위한 일 예의 방법을 예시하는 플로우차트이다.

도 8 은 본 개시물의 기술들에 따라 비디오 데이터를 디코딩하기 위한 일 예의 방법을 예시하는 플로우차트이다.

도 9 는 VPS 에서 시그널링되는 바와 같은 시간적 계층들의 수에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다.

도 10 은 하나 이상의 계층들에서 재순서화될 픽처들 및 디코딩된 픽처 버퍼에 저장될 픽처들의 수에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다.

도 11 은 VPS 에서 시그널링되는 가상 참조 디코더 (hypothetical reference decoder; HRD) 파라미터들에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다.

도 12 는 VPS 에서 시그널링되는 확장 데이터에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0018]

대체로, 본 개시물은 비디오 파라미터 세트 (VPS) 를 이용하여 비디오 데이터를 코딩하는 것을 설명한다. 비디오 데이터는 복수의 계층들, 소정의 계층 내의 픽처들의 시퀀스, 시퀀스 내의 픽처, 픽처 내의 슬라이스들, 및 슬라이스 내의 블록들 (예를 들어, 매크로블록들 또는 코딩 트리 단위들) 을 포함하는 것으로서 계층적으로 범주화될 수도 있다. 시퀀스 파라미터 세트 (sequence parameter set; SPS) 들은 픽처들의 시퀀스에 대한 덜 빈번하게 변화하는 파라미터들을 시그널링하기 위해 이용될 수도 있고, 픽처 파라미터 세트 (picture parameter set; PPS) 들은 개별적인 픽처들에 대한 덜 빈번하게 변화하는 파라미터들을 시그널링하기 위해 이용될 수도 있다.

[0019]

본 개시물의 기술들에 따르면, VPS 는 각각의 계층들에 걸쳐 복수의 시퀀스들에 대한 덜 빈번하게 변화하는 파라미터들을 시그널링할 수도 있다. 즉, VPS 는 상이한 계층들의 시간적으로 함께-위치된 (co-located) 시퀀스들의 세트에 대한 파라미터들을 포함할 수도 있다. 상이한 계층들은 예를 들어, 멀티-뷰 비디오 데이터에 대한 상이한 뷰들, 상이한 품질 계층들, 상이한 공간 해상도 계층들, 시간적 스케일러블 계층들 (즉, 상이한 프레임 레이트들을 허용하는 계층들), 등을 포함할 수도 있다. 이러한 방식으로, 하나의 VPS 는 복수의 상이한 계층들에 대해 제공될 수도 있어서, VPS 는 각각의 계층들의 각각 (예를 들어, 각각의 계층들 내의 각각의 시퀀스들) 에 대해 공통적인 파라미터들을 시그널링한다. 비트스트림은 복수의 계층들의 각각을 포함하는 것으로 말해질 수도 있고, 각각의 계층들은 각각의 서브-비트스트림들을 형성할 수도 있다. 또한, 서브-비트스트림은 2 개 이상의 계층들의 조합에 대응할 수도 있다.

[0020]

본 개시물은 VPS 내에 포함될 수도 있는 데이터의 다양한 예들을 설명한다. 이러한 데이터는 일부의 예들에서, 대응하는 계층들 내에서 서브 계층들의 수 (예를 들어, 서브-계층들의 최대 수) 의 표시를 포함할 수도 있다. 예를 들어, VPS 는 시간적인 계층들의 수 및/또는 시간적인 계층들의 최대 수 (예를 들어, 최고 시간적 계층 식별자) 를 시그널링하는 데이터를 포함할 수도 있다.

[0021]

또 다른 예로서, VPS 는 추가적으로 또는 대안적으로, SPS 에서 이전에 시그널링된 (즉, 기존의 SPS 들에서 시그널링된) 임의의 데이터와 실질적으로 유사한 데이터를 포함할 수도 있다. 이러한 방식으로, 비트스트림의 2 개 이상의 계층들의 시퀀스들이 실질적으로 유사하거나 동일한 파라미터들을 포함할 때, 비디오 코더는 상이한 계층들 사이의 다양한 시퀀스들에 대한 각각의 SPS 들 내의 이러한 데이터를 중복적으로 코딩하는 것이 아니라, 계층들의 시퀀스들에 대한 파라미터들을 시그널링하기 위하여 VPS 를 코딩할 수도 있다.

[0022]

VPS 는 추가적으로 또는 대안적으로, 비디오 표현 정보, 가상 참조 디코더 (HRD) 파라미터들, 및/또는 비트스트림 한정 정보 (bitstream restriction information) 와 같은 비디오 유용성 정보 (video usability information; VUI) 를 정의하는 데이터를 포함할 수도 있다. 비트스트림 한정 정보는 모션 벡터 범위, (예를 들어, DPB 에 의해 유지될 픽처들의 수의 측면에서) 디코딩된 픽처 버퍼 (decoded picture buffer; DPB) 사이즈, 재순서화 프레임들의 수 (즉, 디코딩 순서로부터 디스플레이 순서로 재순서화될 프레임들의 수의 표시),

블록들의 코딩된 사이즈들 (예를 들어, 매크로블록 (MB) 들 또는 코딩 트리 단위들), 및 꾹처들의 코딩된 사이즈들에 대한 한정들을 포함할 수도 있다. VPS 는 하나 이상의 VPS 확장부들을 위한 데이터를 더 제공할 수도 있어서, VPS 는 추가의 표준들, 또는 당면한 HEVC 표준에 대한 확장들에 의해 확장될 수 있다.

[0023]

도 1 은 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 네트워크 추상화 계층 (NAL) 단위들을 코딩하기 위한 기술들을 사용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (destination device; 14) 에 의해 추후의 시간에 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (source device; 12) 를 포함한다. 특히, 소스 디바이스 (12) 는 컴퓨터 판독가능 매체 (16) 를 통해 비디오 데이터를 목적지 디바이스 (14) 에 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 노트북 (즉, 랙톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스, 등을 포함하는 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부의 경우들에 있어서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 구비될 수도 있다.

[0024]

목적지 디바이스 (14) 는 컴퓨터 판독가능 매체 (16) 를 통해 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 컴퓨터 판독가능 매체 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 컴퓨터 판독가능 매체 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 직접적으로 송신하는 것을 가능하게 하기 위한 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조될 수도 있고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 무선 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷-기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하기 위하여 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0025]

일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, 블루레이 (Blu-ray) 디스크들, DVD 들, CD-ROM 들, 플래쉬 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적당한 디지털 저장 매체들과 같은 다양한 분산된 또는 국소적으로 액세스되는 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다. 추가의 예에서, 저장 디바이스는 소스 디바이스 (12) 에 의해 생성되는 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 저장 디바이스로부터의 저장된 비디오 데이터를 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있으며 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 일 예의 파일 서버들은 (예를 들어, 웹사이트를 위한) 웹 서버, FTP 서버, 네트워크 연결 저장 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터를 액세스할 수도 있다. 이것은 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장되는 인코딩된 비디오 데이터를 액세스하기에 적당한 이들의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 그 조합일 수도 있다.

[0026]

본 개시물의 기술들은 무선 애플리케이션들 또는 설정들에 반드시 제한되지는 않는다. 기술들은 다양한 멀티미디어 애플리케이션들 중 임의의 것, 예컨대, 오버-디-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, HTTP 를 통한 동적 적응 스트리밍 (dynamic adaptive streaming over HTTP; DASH) 과 같은 인터넷 스트리밍 비디오 송신들, 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 및/또는 화상통화와 같은 애플리케이션들을 지원하기 위하여 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0027]

도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를

포함한다. 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 본 개시물에 따르면, 소스 디바이스 (12)의 비디오 인코더 (20)는 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 NAL 단위들을 코딩하기 위한 기술들을 적용하도록 구성될 수도 있다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 및 배치들을 포함할 수도 있다.

예를 들어, 소스 디바이스 (12)는 외부 카메라와 같은 외부 비디오 소스 (18)로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14)는 통합된 디스플레이 디바이스를 포함하는 것이 아니라, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0028] 도 1의 예시된 시스템 (10)은 하나의 예에 불과하다. 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 NAL 단위들을 코딩하기 위한 기술들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로, 본 개시물의 기술들은 비디오 인코딩 디바이스에 의해 수행되지만, 기술들은 "CODEC"이라고 전형적으로 지칭되는 비디오 인코더/디코더에 의해 또한 수행될 수도 있다. 또한, 본 개시물의 기술들은 비디오 프리프로세서 (video preprocessor)에 의해 또한 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는, 소스 디바이스 (12)가 목적지 디바이스 (14)로의 송신을 위한 코딩된 비디오 데이터를 생성하는 이러한 코딩 디바이스들의 예들에 불과하다. 일부 예들에서, 디바이스들 (12, 14)은, 디바이스들 (12, 14)의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 이에 따라, 시스템 (10)은 예를 들어, 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 또는 화상전화를 위하여, 비디오 디바이스들 (12, 14) 사이에서 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0029] 소스 디바이스 (12)의 비디오 소스 (18)는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 및/또는 비디오 컨텐츠 제공자로부터 비디오를 수신하기 위한 비디오 공급 인터페이스를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (18)는 컴퓨터 그래픽-기반 데이터를 소스 비디오로서, 또는 라이브 (live) 비디오, 아카이브된 비디오, 및 컴퓨터-생성된 비디오의 조합으로서 생성할 수도 있다. 일부의 경우들에 있어서, 비디오 소스 (18)가 비디오 카메라일 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14)는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 위에서 언급된 바와 같이, 본 개시물에서 설명된 기술들은 일반적으로 비디오 코딩에 적용 가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에 있어서, 캡처된, 프리-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20)에 의해 인코딩될 수도 있다. 다음으로, 인코딩된 비디오 정보는 출력 인터페이스 (22)에 의해 컴퓨터 관독 가능 매체 (16) 상으로 출력될 수도 있다.

[0030] 컴퓨터 관독 가능 매체 (16)는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 일시적인 매체들, 또는 하드 디스크, 플래쉬 드라이브, 컴팩트 디스크, 디지털 비디오 디스크, 블루레이 디스크, 또는 다른 컴퓨터 관독 가능 매체들과 같은 저장 매체들 (즉, 비-일시적인 저장 매체들)을 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (도시되지 않음)는 예를 들어, 네트워크 송신을 통해, 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신할 수도 있고 인코딩된 비디오 데이터를 목적지 디바이스 (14)에 제공할 수도 있다. 유사하게, 디스크 스탬핑 설비 (disc stamping facility)와 같은 매체 생산 설비의 컴퓨팅 디바이스는 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신할 수도 있고, 인코딩된 비디오 데이터를 포함하는 디스크를 생산할 수도 있다. 그러므로, 컴퓨터 관독 가능 매체 (16)는 다양한 예들에서, 다양한 형태들의 하나 이상의 컴퓨터 관독 가능 매체들을 포함하는 것으로 이해될 수도 있다.

[0031] 목적지 디바이스 (14)의 입력 인터페이스 (28)는 컴퓨터 관독 가능 매체 (16)로부터 정보를 수신한다. 컴퓨터 관독 가능 매체 (16)의 정보는, 블록들 및 다른 코딩된 단위들, 예를 들어, GOP들의 특성 및/또는 프로세싱을 설명하는 신택스 엘리먼트 (syntax element)들을 포함하는 신택스 정보로서, 비디오 인코더 (20)에 의해 정의되며 비디오 디코더 (30)에 의해 또한 이용되는 상기 신택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0032] 비디오 인코더 (20) 및 비디오 디코더 (30)는 현재 개발 중에 있는 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 코딩 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HEVC Test Model; HM)을 준수할 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30)는 MPEG-4, Part 10, 진보된 비디오 코딩 (AVC)이라고 대안적으로 지칭되는 ITU-T H.264 표준, 또는 이러한 표준들의 확장들과 같은 다른 전용 또는 산업 표준들에 따라 동작할 수도 있다. 그러나, 본 개시물의 기술들은 임의의 특정한 코딩 표준에 제한되지 않는다. 비

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

[0033] ITU-T H.264/MPEG-4 (AVC) 표준은 합동 비디오 팀 (Joint Video Team; JVT) 이라고 알려진 집단 파트너쉽의 산물로서 ISO/IEC 동화상 전문가 그룹 (Moving Picture Experts Group; MPEG) 과 함께 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 에 의해 공식화되었다. 일부 양태들에서, 본 개시물에서 설명된 기술들은 H.264 표준을 일반적으로 준수하는 디바이스들에 적용될 수도 있다. H.264 표준은, 본원에서 H.264 표준 또는 H.264 사양, 또는 H.264/AVC 표준 또는 사양이라고 지칭될 수도 있는, 2005년 3월자의 ITU-T Study Group 에 의한 ITU-T 추천안 H.264, Advanced Video Coding for generic audiovisual services (일반적인 시청각 서비스들을 위한 진보된 비디오 코딩) 에 설명되어 있다. 합동 비디오 팀 (JVT) 은 H.264/MPEG-4 AVC 에 대한 확장들에 계속 노력을 들인다.

[0034] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (digital signal processor; DSP) 들, 주문형 접적회로 (application specific integrated circuit; ASIC) 들, 필드 프로그래밍 가능 게이트 어레이 (field programmable gate array; FPGA) 들, 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들과 같은 다양한 적당한 인코더 회로부 중 임의의 것으로서 구현될 수도 있다. 기술들이 부분적으로 소프트웨어로 구현될 때, 디바이스는 소프트웨어에 대한 명령들을 적당한 비일시적인 컴퓨터 판독가능 매체에 저장할 수도 있고, 본 개시물의 기술들을 수행하기 위해 하나 이상의 프로세서들을 이용하는 하드웨어에서 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 이들 중 어느 하나는 각각의 디바이스에서 조합된 인코더/디코더 ("CODEC") 의 부분으로서 통합될 수도 있다.

[0035] JCT-VC 는 HEVC 표준의 개발에 노력을 들이고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 이라고 지칭되는 비디오 코딩 디바이스의 진화하는 모델에 기초한다. HM 은, 예를 들어, ITU-T H.264/AVC 에 따른 기준 디바이스들에 대하여 비디오 코딩 디바이스들의 몇몇 추가적인 능력들을 가정한다. 예를 들어, H.264 가 9 개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM 은 33 개만큼 많은 인트라-예측 인코딩 모드들을 제공할 수도 있다.

[0036] 일반적으로, HM의 작업 모델은, 비디오 프레임 또는 픽처가 루마 (luma) 및 크로마 (chroma) 샘플들을 둘 모두 포함하는 최대 코딩 단위 (largest coding unit; LCU) 들 또는 트리블록들의 시퀀스로 분할될 수도 있다는 것을 설명한다. 비트스트림 내에서의 신택스 데이터는 픽셀들의 수의 측면에서 최대 코딩 단위인 LCU 에 대한 사이즈를 정의할 수도 있다. 슬라이스는 코딩 순서로 된 다수의 연속 트리블록들을 포함한다. 비디오 프레임 또는 픽처는 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 쿼드트리 (quadtree) 에 따라 코딩 단위 (CU) 들로 분할될 수도 있다. 일반적으로, 쿼드트리 데이터 구조는 CU 당 하나의 노드를 포함하고, 루트 노드 (root node) 는 트리블록에 대응한다. CU 가 4 개의 서브-CU 들로 분할될 경우, CU 에 대응하는 노드는 4 개의 리프 노드 (leaf node) 들을 포함하고, 리프 노드들의 각각은 서브-CU 들 중 하나에 대응한다.

[0037] 쿼드트리 데이터 구조의 각각의 노드는 대응하는 CU 에 대한 신택스 데이터를 제공할 수도 있다. 예를 들어, 쿼드트리 내의 노드는 그 노드에 대응하는 CU 가 서브-CU 들로 분할되는지 여부를 표시하는 분할 플래그 (split flag) 를 포함할 수도 있다. CU 에 대한 신택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU 가 서브-CU 들로 분할되는지 여부에 종속될 수도 있다. CU 가 더 분할되지 않을 경우, 그것은 리프-CU 라고 지칭된다. 본 개시물에서는, 원래의 리프-CU 의 명시적인 분할이 없더라도, 리프-CU 의 4 개의 서브-CU 들은 리프-CU 들이라고 또한 지칭될 것이다. 예를 들어, 16x16 사이즈의 CU 가 더 분할되지 않을 경우에는, 16x16 CU 가 전혀 분할되지 않았지만, 4 개의 8x8 서브-CU 들은 리프-CU 들이라고 또한 지칭될 것이다.

[0038] CU 가 사이즈 구분을 가지지 않는다는 점을 제외하고는, CU 는 H.264 표준의 매크로블록과 유사한 목적을 가진다. 예를 들어, 트리블록은 4 개의 자식 (child) 노드들 (서브-CU 들이라고 또한 지칭됨) 로 분할될 수도 있고, 각각의 자식 노드는 궁극적으로 부모 (parent) 노드일 수도 있고, 또 다른 4 개의 자식 노드들로 분할될 수도 있다. 쿼드트리의 리프 노드라고 지칭되는 최종적인 분할되지 않은 자식 노드는 리프-CU 라고 또한 지

칭되는 코딩 노드를 포함한다. 코딩된 비트스트림과 연관된 십екс 데이터는 최대 CU 심도 (depth) 라고 지칭되는, 트리블록이 분할될 수도 있는 최대 횟수를 정의할 수도 있고, 코딩 노드들의 최대 사이즈를 또한 정의할 수도 있다. 따라서, 비트스트림은 최소 코딩 단위 (smallest coding unit; SCU) 를 또한 정의할 수도 있다. 본 개시물은 HEVC 의 상황에서의 CU, PU, 또는 TU 중 임의의 것, 또는 다른 표준들의 상황에서의 유사한 데이터 구조들 (예를 들어, H.264/AVC 에서 매크로블록들 및 그 서브-블록들) 을 지칭하기 위하여 용어 "블록" 을 이용한다.

[0039] CU 는 코딩 노드, 및 코딩 노드와 연관된 예측 단위 (prediction unit; PU) 들 및 변환 단위 (transform unit; TU) 들을 포함한다. CU 의 사이즈는 코딩 노드의 사이즈에 대응하고 형상에 있어서 정사각형이어야 한다.

CU 의 사이즈는 8x8 픽셀들로부터 최대 64x64 픽셀들 이상의 픽셀들을 갖는 트리블록의 사이즈까지의 범위일 수도 있다. 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다. CU 와 연관된 십екс 데이터는, 예를 들어, CU 를 하나 이상의 PU 들로 파티셔닝하는 것을 설명할 수도 있다. 파티셔닝 모드들은, CU 가 스kip (skip) 또는 직접 모드 인코딩되는지, 인트라-예측 모드 인코딩되는지, 또는 인터-예측 모드 인코딩되는지 여부 사이에서 상이할 수도 있다. PU 들은 형상에 있어서 비-정사각형 (non-square) 인 것으로 파티셔닝될 수도 있다. CU 와 연관된 십екс 데이터는 예를 들어, 큐드트리에 따라 CU 를 하나 이상의 TU 들로 파티셔닝하는 것을 또한 설명할 수도 있다. TU 는 형상에 있어서 정사각형 또는 비-정사각형 (예를 들어, 직사각형) 일 수 있다.

[0040] HEVC 표준은 TU 들에 따른 변환들을 허용하고, 이것은 상이한 CU 들에 대해 상이할 수도 있다. TU 들은 파티셔닝된 LCU 에 대해 정의된 소정의 CU 내에서의 PU 들의 사이즈에 기초하여 전형적으로 사이즈가 정해지지만, 이것은 항상 그런 것은 아닐 수도 있다. TU 들은 전형적으로 PU 들과 동일한 사이즈이거나 또는 그보다 더 작다. 일부 예들에서, CU 에 대응하는 잔차 샘플들은, "잔차 큐드 트리" (residual quad tree; RQT) 로서 알려진 큐드트리 구조를 이용하여 더 작은 단위들로 서브분할될 수도 있다. RQT 의 리프 노드들은 변환 단위 (TU) 들이라고 지칭될 수도 있다. TU 들과 연관된 픽셀 차이 값들은 변환 계수들을 생성하도록 변환될 수도 있고, 변환 계수들은 양자화될 수도 있다.

[0041] 리프-CU 는 하나 이상의 예측 단위 (PU) 들을 포함할 수도 있다. 일반적으로, PU 는 대응하는 CU 의 전부 또는 일부분에 대응하는 공간적인 영역을 나타내고, PU 에 대한 참조 샘플을 취출하기 위한 데이터를 포함할 수도 있다. 또한, PU 는 예측과 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라-모드 인코딩될 때, PU 에 대한 데이터는 잔차 큐드트리 (RQT) 내에 포함될 수도 있고, 잔차 큐드트리는 PU 에 대응하는 TU 에 대한 인트라-예측 모드를 설명하는 데이터를 포함할 수도 있다. 또 다른 예로서, PU 가 인터-모드 인코딩될 때, PU 는 PU 에 대한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. PU 에 대한 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터의 해상도 (예를 들어, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 지시하는 참조 픽처, 및/또는 모션 벡터에 대한 참조 픽처 리스트 (예를 들어, 리스트 0, 리스트 1, 또는 리스트 C) 를 설명할 수도 있다.

[0042] 하나 이상의 PU 들을 갖는 리프-CU 는 하나 이상의 변환 단위 (TU) 들을 또한 포함할 수도 있다. 변환 단위 들은 위에서 논의된 바와 같이, RQT (TU 큐드트리 구조라고 또한 지칭됨) 를 이용하여 특정될 수도 있다. 예를 들어, 분할 플래그는 리프-CU 가 4 개의 변환 단위들로 분할되는지 여부를 표시할 수도 있다. 다음으로, 각각의 변환 단위는 추가의 서브-TU 들로 더 분할될 수도 있다. TU 가 더 분할되지 않을 때, 그것은 리프-TU 라고 지칭될 수도 있다. 일반적으로, 인트라 코딩을 위하여, 리프-CU 에 속하는 모든 리프-TU 들은 동일한 인트라 예측 모드를 공유한다. 즉, 동일한 인트라-예측 모드는 일반적으로 리프-CU 의 모든 TU 들에 대한 예측된 값들을 계산하기 위하여 적용된다. 인트라 코딩을 위하여, 비디오 인코더는 TU 에 대응하는 CU 의 일부분과 원래의 블록 사이의 차이로서, 인트라 예측 모드를 이용하는 각각의 리프-TU에 대한 잔차 값을 계산할 수도 있다. TU 는 반드시 PU 의 사이즈에 제한되지는 않는다. 따라서, TU 들은 PU 보다 더 크거나 더 작을 수도 있다. 인트라 코딩을 위하여, PU 는 동일한 CU 에 대한 대응하는 리프-TU 와 함께 위치될 수도 있다. 일부 예들에서, 리프-TU 의 최대 사이즈는 대응하는 리프-CU 의 사이즈에 대응할 수도 있다.

[0043] 또한, 리프-CU 들의 TU 들은 잔차 큐드트리 (RQT) 들이라고 지칭되는 각각의 큐드트리 데이터 구조들과 또한 연관될 수도 있다. 즉, 리프-CU 는 리프-CU 가 어떻게 TU 들로 파티셔닝 되는지를 표시하는 큐드트리를 포함할 수도 있다. TU 큐드트리의 루트 노드는 일반적으로 리프-CU 에 대응하는 반면, CU 큐드트리의 루트 노드는 일반적으로 트리블록 (또는 LCU) 에 대응한다. 분할되지 않은 RQT 의 TU 들은 리프-TU 들이라고 지칭된다. 대체로, 본 개시물은 달리 언급되지 않으면, 리프-CU 및 리프-TU 를 지칭하기 위하여 용어들 CU 및 TU

를 이용한다.

[0044] 비디오 시퀀스는 비디오 프레임들 또는 픽처들의 시리즈 (series) 를 전형적으로 포함한다. 픽처들의 그룹 (group of pictures; GOP) 은 비디오 픽처들의 중 하나 이상의 시리즈를 일반적으로 포함한다. GOP 는 GOP 의 헤더, 픽처들의 하나 이상의 헤더, 또는 다른 장소에서, GOP 내에 포함된 픽처들의 수를 설명하는 신택스 데 이터를 포함할 수도 있다. 픽처의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 설명하는 슬라이스 신택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터를 인코딩하기 위하여 개별적인 비디오 슬라이스들 내의 비디오 블록들에 대해 전형적으로 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정된 또는 변동되는 사이즈들을 가질 수도 있고, 특정된 코딩 표준에 따라 사이즈에 있어서 상이할 수도 있다.

[0045] 일 예로서, HM 은 다양한 PU 사이즈들에서의 예측을 지원한다. 특정 CU 의 사이즈가 2Nx2N 이라고 가정하면, HM 은 2Nx2N 또는 NxN 의 PU 사이즈들에서의 인트라-예측, 및 2Nx2N, 2NxN, Nx2N, 또는 NxN 의 대칭적 PU 사이즈들에서의 인터-예측을 지원한다. HM 은 2NxnU, 2NxnD, nLx2N, 및 nRx2N 의 PU 사이즈들에서의 인터-예측에 대한 비대칭적 파티셔닝을 또한 지원한다. 비대칭적 파티셔닝에서, CU 의 하나의 방향은 파티셔닝되지 않는 반면, 다른 방향은 25% 및 75% 로 파티셔닝된다. 25% 파티션에 대응하는 CU 의 일부분은 "상부", "하부", "좌측", 또는 "우측" 의 표시가 후속하는 "n" 에 의해 표시된다. 따라서, 예를 들어, "2NxnU" 는 수평적으로 상부 2Nx0.5N PU 와 하부 2Nx1.5N PU 로 파티셔닝되는 2Nx2N CU 를 지칭한다.

[0046] 본 개시물에서, "NxN" 및 "N 바이 (by) N" 은, 예를 들어, 16x16 픽셀들 또는 16 바이 16 픽셀들과 같이 수직 및 수평 차원들의 측면에서 비디오 블록의 픽셀 차원들을 지칭하기 위해 상호 교환 가능하게 이용될 수도 있다. 일반적으로, 16x16 블록은 수직 방향으로 16 픽셀들 ($y=16$) 및 수평 방향으로 16 픽셀들 ($x=16$) 을 가질 것이다. 마찬가지로, NxN 블록은 수직 방향으로 N 픽셀들 및 수평 방향으로 N 픽셀들을 일반적으로 가지며, 여기서 N은 음이 아닌 정수 값을 나타낸다. 블록 내의 픽셀들은 행 (row) 들 및 열 (column) 들로 배치될 수도 있다. 또한, 블록들은 수직 방향에서의 것과 동일한 수의 픽셀들을 수평 방향에서 가질 필요가 없다. 예를 들어, 블록들은 NxM 픽셀들을 포함할 수도 있으며, 여기서 M 은 N 과 반드시 동일하지는 않다.

[0047] CU 의 PU 들을 이용한 인트라-예측 또는 인터-예측 코딩에 후속하여, 비디오 인코더 (20) 는 CU 의 TU 들에 대한 잔차 데이터를 계산할 수도 있다. PU 들은 공간 도메인 (또한 픽셀 도메인이라고 지칭됨) 에서 예측 픽셀 데이터를 생성하는 방법 또는 모드를 설명하는 신택스 데이터를 포함할 수도 있고, TU 들은 변환, 예를 들어, 이산 코사인 변환 (discrete cosine transform; DCT), 정수 변환, 웨이블렛 변환 (wavelet transform), 또는 개념적으로 유사한 변환의 잔차 비디오 데이터로의 적용 이후의 변환 도메인에서의 계수들을 포함할 수도 있다. 잔차 데이터는 PU 들에 대응하는 예측 값들과 인코딩되지 않은 픽처의 픽셀들 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔차 데이터를 포함하는 TU 들을 형성할 수도 있고, 그 다음으로, CU 에 대한 변환 계수들을 생성하기 위하여 TU 들을 변환할 수도 있다.

[0048] 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행 할 수도 있다. 양자화는, 계수들을 표현하기 위해 이용되는 데이터의 양을 가능한 대로 감소시키기 위해 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 일반적으로 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 값은 양자화 동안 m-비트 값으로 내림 (round down) 될 수도 있으며, 여기서 n 은 m 보다 더 크다.

[0049] 양자화에 후속하여, 비디오 인코더는 양자화된 변환 계수들을 포함하는 2 차원 행렬로부터 1 차원 벡터를 생성 하는 변환 계수들을 스캐닝할 수도 있다. 스캔은 더 높은 에너지 (및 그러므로 더 낮은 주파수) 의 계수들을 어레이의 전방에 두고 더 낮은 에너지 (및 그러므로 더 높은 주파수) 의 계수들을 어레이의 후방에 두도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 양자화된 변환 계수들을 스캐닝하여 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위하여 미리 정의된 스캔 순서를 사용할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응 스캔을 수행할 수도 있다. 1 차원 벡터를 형성하기 위하여 양자화된 변환 계수들을 스캐닝한 후, 비디오 인코더 (20) 는 예를 들어, 컨텍스트 적응 가변 길이 코딩 (context adaptive variable length coding; CAVLC), 컨텍스트 적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; CABAC), 신택스-기반 컨텍스트-적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 간격 파티셔닝 엔트로피 (Probability Interval Partitioning Entropy; PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라 1 차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30) 에 의해 이용하기 위한 인코딩된 비디오

데이터와 연관된 신택스 엘리먼트들을 또한 엔트로피 인코딩할 수도 있다.

[0050] CABAC 를 수행하기 위하여, 비디오 인코더 (20) 는 컨텍스트 모델 내의 컨텍스트를 송신될 심볼에 배정할 수도 있다. 컨텍스트는, 예를 들어, 심볼의 이웃하는 값들이 비-제로 (non-zero) 인지 아닌지의 여부에 관련될 수도 있다. CAVLC 를 수행하기 위하여, 비디오 인코더 (20) 는 송신될 심볼에 대한 가변 길이 코드를 선택 할 수도 있다. VLC 에서의 코드워드들은, 상대적으로 더 짧은 코드들이 더 큰 확률의 심볼들에 대응하는 반면, 더 긴 코드들이 더 작은 확률의 심볼들에 대응하도록 구성될 수도 있다. 이러한 방식으로, VLC 의 이용은, 예를 들어, 송신될 각각의 심볼에 대해 동일-길이의 코드워드들을 이용하는 것에 비해 비트 절약을 달성할 수도 있다. 확률 결정은 심볼에 배정된 컨텍스트에 기초할 수도 있다.

[0051] 본 개시물의 기술들에 따르면, 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은 비디오 코더는 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하고 VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩하도록 구성될 수도 있다. 아래에서 더욱 상세하게 설명된 표 2 및 표 5 는 VPS 의 신택스 엘리먼트들의 일 예의 세트들을 포함한다. 비디오 데이터의 하나 이상의 계층들의 각각은 VPS, 즉, 동일한 VPS 를 참조할 수도 있다. 다시 말해서, VPS 는 비디오 데이터의 공통의 세트의 모든 계층들, 예를 들어, MVC 비디오 데이터의 모든 SVC 계층들 및/또는 모든 뷰들에 적용될 수도 있다.

[0052] VPS 는 정보의 다양한 범주들을 포함할 수도 있다. 예를 들어, VPS 는 샘플 차원 카운터 설명 (sample dimension counter description; SDCD) 을 포함할 수도 있다. 즉, 각각의 차원에 대하여, 비디오 코더는 인덱스들의 세트를 시그널링할 수도 있다. 가능한 차원들은, cnt_p: 코딩된 비디오 시퀀스에 포함된 우선순위 계층들의 수; cnt_d: 비트스트림에서 상이한 종속성 계층들이 얼마나 많은지, 동일한 공간 해상도 및 비트 심도를 갖는 다수의 계층들은 상이한 종속성 계층들에 속할 수도 있음; cnt_t: 비트스트림에서 시간적 계층들이 얼마나 많은지; cnt_q: 비트스트림에서 임의의 종속성 계층에 대한 품질 계층들의 최대 수; 및 cnt_v: 뷰들의 최대 수를 포함한다. 비트 심도 설정들은 8-비트 또는 12-비트를 포함할 수도 있고, 상이한 컬러 컴포넌트에 대해 상이할 수도 있다. 크로마 샘플링 포맷들은 4:0:0, 4:2:0 및 4:4:4 를 포함할 수도 있다.

[0053] VPS 는 특성 맵핑에 대한 샘플 인덱스를 또한 포함할 수도 있다. 각각의 차원에 대하여, 특성 표시자가 0 으로부터 샘플 차원 카운터 마이너스 (minus) 1 까지의 범위인 인덱스와 동일하지 않을 경우, 각각의 특성 인덱스에 대한 특성 표시자를 특정하기 위하여 루프가 도입될 수도 있다. 맵핑은 각각의 종속성 인덱스 (dependency index) 에 대하여, 특정 비트 심도 값 및 특정 크로마 샘플 포맷을 갖는 특정 공간 해상도를 포함할 수도 있다. 디코더에서 항상 고정된 룩업 (look-up) 테이블이 있을 경우에는 이것이 생략될 수도 있고, 예를 들어, 0 이 4:2:0 에 대응할 수도 있고, 1 이 4:4:4 에 대응할 수도 있고, 2 가 4:0:0 에 대응할 수도 있다는 것에 주목해야 한다. 맵핑은 추가적으로 또는 대안적으로, 각각의 시간적 인덱스/id 에 대하여, 특정 프레임 레이트 또는 평균 프레임 레이트를; 각각의 뷰 인덱스에 대하여, 특정 뷰 id를; 각각의 비트 심도 인덱스에 대하여, 루마 및 크로마에 대한 한 쌍의 특정 비트 심도 값들을; 그리고 각각의 크로마 샘플링 포맷에 대하여, 특정 크로마 샘플링 포맷 표시자를 포함할 수도 있다.

[0054] VPS 는 다음과 같은 제어 파라미터들 및 툴 인레이블/디스에이블 플래그들을 또한 포함할 수도 있다: pcm_bit_depth_luma_minus1, pcm_bit_depth_chroma_minus1, loop_filter_across_slice_flag, pcm_loop_filter_disable_flag, temporal_id_nesting_flag, 하나 이상의 타일 관련된 신택스 엘리먼트들, chroma_pred_from_luma_enabled_flag, sample_adaptive_offset_enabled_flag, adaptive_loop_filter_enabled_flag, 및 inter_4x4_enabled_flag.

[0055] VPS 는 하나 이상의 동작점 설명들을 또한 포함할 수도 있다. 동작점들은 비트스트림에 포함된 비디오 데이터의 뷰들의 총 수의 서브세트를 일반적으로 설명한다. 동작점은 디코딩하거나, 출력하거나, 또는 둘 모두를 행할 때에 참조를 위해 이용될 수도 있는 다른 뷰들 뿐만 아니라, 출력을 위해 타겟으로 정해진 특정한 수의 뷰들도 포함할 수도 있다. 비트스트림은 동작점 설명들에 의해 설명되는 하나 이상의 동작점들을 포함할 수도 있다. 동작점 설명들은 최대 동작점들의 수, 상이한 계층들 또는 뷰들 사이의 종속성, 각각의 동작점에 대한 프로파일 및 레벨, 각각의 동작점에 대한 비트 레이트, 각각의 동작점에 대한 동작점들 사이의 종속성, 각각의 동작점에 대한 다른 한정들, 비디오 유용성 정보 (VUI) 또는 VUI 의 일부, 및/또는 각각의 계층 또는 뷰에 대한 VUI 또는 VUI 의 일부를 정의하는 정보를 포함할 수도 있다. 추가적으로 또는 대안적으로, 동작점 설명들은 각각의 동작점에 대하여, 동작점 비디오 코딩 계층 (video coding layer; VCL) 네트워크 추상화 계층 (NAL) 단위 표현을 포함할 수도 있다. 일부 예들에서, 동작점 VCL NAL 단위 표현은 각각의 차원에 대하여, 3 개의 가능한 선택들을 포함할 수도 있다: (1) 특정 인덱스 값: 예를 들어, 공간 해상도에 대하여, 크로마 샘

플링 포맷에 대한 비트 심도에 대한 것; (2) 인덱스 값의 범위: 예를 들어, 시간적 계층들에 대하여, 0 내지 최고 시간적 계층 id, 품질 계층들에 대하여, 0 내지 최고 품질 계층 id; 또는 (3) 인덱스 값들의 리스트, 예를 들어, 뷰들에 대하여, 뷰 인덱스 값들의 리스트.

[0056] 일부 예들에서, VPS 는 비트스트림의 계층들 사이에서 시간적 계층들의 최대 수를 표시하는 데이터를 포함할 수도 있다. 즉, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 대응하는 비트스트림에 대한 시간적 계층들의 최대 수를 표시하는 데이터를 포함하는 VPS 를 코딩하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 시간적 계층들의 최대 수를 결정할 수도 있고, 시간적 계층들의 결정된 최대 수를 나타내는 데이터를 포함하기 위하여 VPS 를 인코딩할 수도 있는 반면, 비디오 디코더 (30) 는 시간적 계층들의 최대 수를 결정하기 위하여 VPS 를 디코딩할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 시간적 계층들의 결정된 최대 수에 기초하여 비트스트림의 비디오 데이터를 또한 코딩할 수도 있다. 예를 들어, 시간적 계층들의 최대 수는 다양한 시간적 계층들을 나타내기 위하여 필요한 시간적 식별자들의 수에 영향을 줄 수도 있다.

또 다른 예로서, 시간적 계층들의 최대 수는, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 예를 들어, 꽂쳐 순서 카운트 (picture order count; POC) 값들을 이용하여 참조 꽂쳐 식별자들을 코딩하는 방식에 영향을 줄 수도 있다.

[0057] 또 다른 예로서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 동일한 시간적 계층까지의 그리고 동일한 시간적 계층을 포함하는 참조 데이터만을 이용하여 특정한 시간적 계층의 데이터를 코딩하도록 구성될 수도 있다.

다시 말해서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 더 상위 시간적 계층의 참조 데이터를 이용하여 특정한 시간적 계층의 데이터를 코딩하는 것을 회피하도록 구성될 수도 있다. 이러한 방식으로, 비디오 디코더 (30) 는 서브-비트스트림 추출 이후에도 시간적 계층들의 소정의 세트의 비디오 데이터를 정확하게 디코딩하는 것을 보장할 수 있다. 즉, 서브-비트스트림 추출이 수행될 경우, 추출된 서브-비트스트림의 최상위 계층 위의 어떤 시간적 계층들은 참조를 위하여 이용가능하지 않을 것이다. 현재의 계층의 또는 현재의 계층 아래의 계층들의 데이터를 참조하기만 하여 각각의 시간적 계층의 데이터를 코딩함으로써, 그렇지 않을 경우에 특정한 계층에서의 데이터가 더 상위 계층으로부터의 데이터에 종속되도록 하는 것으로부터 기인할 수도 있는 예들로서, 서브-비트스트림 추출의 결과로 손실될 예들이 회피될 수 있다.

[0058] 일부 예들에서, VPS 는 추가적으로 또는 대안적으로, 비트스트림의 하나 이상의 계층들에서 재순서화될 꽂쳐들의 수와, 디코딩된 꽂쳐 버퍼 (DPB) 에 저장될 꽂쳐들의 수 중 어느 하나 또는 둘 모두를 표시하는 데이터를 포함한다. 위에서 언급된 바와 같이, 이러한 데이터는 비트스트림 한정 정보라고 지칭될 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코더 (30) 의 기능들을 결정할 수도 있고, 대응하는 비트스트림이 비디오 디코더 (30) 에 의해 디코딩되기에 적절한 것인지 여부, 또는 목적지 디바이스 (14) 가 (예를 들어, 컨텐츠의 다수의 버전들이 이용가능한 것으로 가정하여, 네트워크-기반 컨텐츠 제공자로부터) 대안적인 컨텐츠를 선택해야 하는지 여부를 결정하기 위하여 비트스트림 한정 정보를 이용할 수도 있다.

[0059] 또한, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 데이터의 코딩 동안에 비트스트림 한정 정보를 이용할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 비트스트림 한정 정보가 위반되지 않은 것을 보장할 수도 있다. 즉, 비트스트림 한정 정보가 최대 N 꽂쳐들이 DPB 에 저장되어야 함을 표시한다고 가정하면, 비디오 인코더 (20) 는 불과 N 개의 꽂쳐들이 임의의 소정의 시간에 하나 이상의 참조 꽂쳐 리스트들의 임의의 조합으로 포함된다는 것을 보장할 수도 있다. 또 다른 예로서, 꽂쳐 재순서화 정보가 최대 M 개의 꽂쳐들만큼 꽂쳐가 시프트 (shift) 되어야 함을 표시한다고 가정하면, 비디오 인코더 (20) 는 M 개를 초과하는 꽂쳐들만큼 꽂쳐가 시프트되지 않는다는 것을 보장할 수도 있다. 이러한 방식의 꽂쳐들의 시프트는 디코딩 순서와 꽂쳐의 디스플레이 순서 사이의 차이에 일반적으로 대응한다. 마찬가지로, 비디오 디코더 (30) 는 예를 들어, DPB 플러싱 (flushing) 과 같은 DPB 관리를 수행하기 위하여, 코딩 동안에 이러한 정보를 이용할 수도 있다.

비디오 인코더 (20) 및 비디오 디코더 (30) 는 참조 꽂쳐 식별자 값들을 코딩할 때, DPB 에 저장될 꽂쳐들의 최대 수 및/또는 재순서화될 꽂쳐들의 수와 같은 비트스트림 한정 정보를 또한 이용할 수도 있다.

[0060] 일부 예들에서, VPS 는 추가적으로 또는 대안적으로, 가상 참조 디코더 (HRD) 파라미터들을 표시하는 데이터를 포함한다. HRD 파라미터들은 예를 들어, 데이터가 코딩된 꽂쳐 버퍼 (coded picture buffer; CPB) 로부터 제거되어야 할 시간들을 설명하는 데이터를 포함한다. 비디오 디코더 (30) 와 같은 디코더들에서는, CPB 가 데이터가 디코딩을 준비할 때까지 코딩된 비디오 데이터가 저장되는 버퍼를 나타낸다. 비디오 디코더 (30) 와 같은 디코더들은, 예를 들어, 인터-예측된 데이터를 위한 그리고 디코딩 순서로부터 디스플레이 순서로의 꽂쳐들의 재순서화를 위한 참조 데이터로서 이용될 디코딩된 비디오 데이터가 저장되는 디코딩된 꽂쳐 버퍼 (DPB)

를 또한 포함할 수도 있다.

[0061] HRD 파라미터들은 특정한 픽처들이 CPB로부터 언제 제거되고 디코딩되어야 하는지를 표시하는 데이터를 포함할 수도 있다. 따라서, 비디오 인코더 (20)는 픽처들이 CPB로부터 언제 제거될 수 있고 디코딩될 수 있는지를 표시하기 위하여 VPS의 HRD 파라미터들을 인코딩할 수도 있는 반면, 비디오 디코더 (30)는 CPB로부터 픽처들을 언제 제거할 것인지를 결정하기 위하여 VPS의 HRD 파라미터들을 디코딩할 수도 있다. 마찬가지로, 비디오 인코더 (20) 및 비디오 디코더 (30)는 예를 들어, HRD 파라미터들에 의해 표시된 코딩 순서로, HRD 파라미터들에 따라 픽처들을 코딩할 수도 있다. 이러한 방식으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 HRD 파라미터들을 포함하는 VPS를 코딩하고, HRD 파라미터들에 적어도 부분적으로 기초하여 VPS에 대응하는 비디오 데이터를 코딩하도록 구성될 수도 있다.

[0062] VPS는 예를 들어, 하나 이상의 추가적인 코딩 툴들에 대한 데이터를 제공하기 위하여, VPS가 확장되었는지 여부를 표시하는 확장 데이터를 또한 포함할 수도 있다. 이러한 코딩 툴들은 예를 들어, ITU-T H.264/AVC 또는 당면한 HEVC 표준과 같은 대응하는 비디오 코딩 표준의 툴들과는 상이한 툴들일 수도 있다. 또한, 이러한 코딩 툴들은 구성 데이터를 요구할 수도 있다. 이 구성 데이터는 VPS의 확장 데이터에서 제공될 수도 있다. 이러한 방식으로, 이러한 코딩 툴들을 이용하여 비디오 데이터를 코딩할 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 확장 데이터가 존재하는지 여부를 표시하는 VPS와, 존재할 경우, VPS의 확장 데이터를 코딩할 수도 있다. 또한, 이러한 확장 데이터가 존재할 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 확장 데이터를 이용하여 비디오 데이터를 코딩하기 위하여 대응하는 코딩 툴들을 실행할 수도 있다.

[0063] 다양한 비디오 코딩 표준들은 에러 없는 (error-free) 비트스트림들에 대한 대응하는 신택스, 시멘틱들 (semantics), 및 디코딩 프로세스를 정의하고, 이를 중 임의의 것은 어떤 프로파일 또는 레벨을 준수한다. 비디오 코딩 표준들은 일반적으로 인코더를 특정하지 않지만, 인코더는 생성된 비트스트림들이 디코더에 대하여 표준을 따른다는 것을 보장하는 작업을 부여받는다. 비디오 코딩 표준들의 상황에서, "프로파일"은 알고리즘들, 특징들, 또는 툴들 및 이들에 적용되는 제약들의 서브세트에 대응한다. H.264 표준에 의해 정의된 바와 같이, 예를 들어, "프로파일"은 H.264 표준에 의해 특정되는 전체 비트스트림 신택스의 서브세트이다. "레벨"은 예를 들어, 디코더 메모리 및 연산과 같은 디코더 차원 소비의 제한들에 대응하고, 이 제한들은 픽처들의 해상도, 비트 레이트, 및 블록 프로세싱 레이트와 관련된다. 프로파일은 profile_idc (프로파일 표시자) 값으로 시그널링될 수도 있는 반면, 레벨은 level_idc (레벨 표시자) 값으로 시그널링될 수도 있다. 본 개시물의 기술들에 따르면, 프로파일 및 레벨 정보는 위에서 논의된 바와 같이, 동작점 설명들에서 특정될 수도 있다.

[0064] 일부 예들에서, 비트스트림의 각각의 계층 또는 뷔는 비디오 파라미터 세트 (VPS)를 참조하고, 계층화된 시퀀스 파라미터 세트 (Layered sequence Parameter Set; LPS)는 각각의 계층에 대하여 활성일 수도 있다. LPS는 설계에서 VPS를 참조함으로써 가능한 한 경량 (lightweight)으로서 유지될 수도 있다. LPS는 아래에서 논의된 정보 중 임의의 것 또는 전부를 포함할 수도 있다. LPS는 각각의 차원에 대하여, 각각의 차원에 대한 인덱스를 표시하는 샘플 차원 표시를 포함할 수도 있다. 예를 들어, VPS에서, 공간 해상도 0에 대한 인덱스가 320x240의 공간 특성에 배정되고 공간 해상도 1의 인덱스가 640x480에 배정되고, 현재의 계층에 640x480의 해상도가 배정될 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 현재의 계층에 대한 1의 값으로 신택스 엘리먼트를 코딩할 수도 있다. 즉, 비디오 인코더 (20)는 640x480의 해상도를 특정하기 위하여 신택스 엘리먼트에 대한 1의 값을 시그널링할 수도 있는 반면, 비디오 디코더 (30)는 신택스 엘리먼트에 대한 1의 값에 기초하여, 1의 값을 가지는 신택스 엘리먼트를 갖는 현재의 계층이 640x480의 해상도를 가지는 것으로 결정할 수도 있다.

[0065] LPS는 제어 파라미터들 및 툴 인에이블/디스에이블 플래그들을 또한 포함할 수도 있다. 예를 들어, 제어 파라미터들 및 툴 인에이블/디스에이블 플래그들은 pcm_bit_depth_luma_minus1, pcm_bit_depth_chroma_minus1, loop_filter_across_slice_flag, pcm_loop_filter_disable_flag, 하나 이상의 타일 관련 신택스 엘리먼트들, chroma_pred_from_luma_enabled_flag, sample_adaptive_offset_enabled_flag, adaptive_loop_filter_enabled_flag, 및 코딩 단위 (CU) 계층구조 (hierarchy)를 포함할 수도 있다.

[0066] LPS는 슬라이스, 슬라이스들의 그룹, 픽처, 또는 몇몇 픽처들에 적용하는 다른 타입들의 파라미터 세트들의 정보를 더 포함할 수도 있다. 이 파라미터 세트들의 각각은 특정 픽처 파라미터 세트 (PPS)를 참조할 수도 있다.

- [0067] 비디오 인코더 (20) 및 비디오 디코더 (30) 와 같은 비디오 코더는 PPS 가 LPS 또는 VPS 를 참조하지 않는 것으로 보장하고 및/또는 결정하도록 구성될 수도 있다. 따라서, 비디오 코더는 비트스트림의 각각의 PPS 가 LPS 또는 VPS 를 참조하지 않는 것을 보장할 수도 있다. PPS 의 파싱 (parsing) 은 독립적일 수도 있다. PPS 가 VPS 또는 LPS 의 신팩스 엘리먼트들과 동일한 신팩스 엘리먼트들 중 하나 이상을 포함할 때, PPS 의 신팩스 엘리먼트들은 VPS 또는 LPS 의 신팩스 엘리먼트들을 겹쳐 쓸 수도 있다.
- [0068] 비디오 코더는 모든 파라미터 세트들을 함께 그룹화하는 그룹화 파라미터 세트 (GPS) 를 코딩하도록 더 구성될 수도 있다. 비디오 코더는 GPS 내의 복수의 상이한 그룹들을 코딩할 수도 있고, 그 각각은 개별적인 GPS 식별자 (id) 들을 가진다. GPS 내의 그룹들의 각각은 파라미터 세트들의 상이한 조합을 포함할 수도 있다. 이러한 방식으로, 슬라이스 헤더는 대응하는 GPS id 에 대한 참조를 포함하기만 할 필요가 있고, 파라미터 세트의 타입의 표시를 포함할 필요는 없다. 2012년 1월 25일자로 출원된 미국 특허 출원 제 61/590,702 호는 또한, 파라미터 세트들의 상이한 타입들이 함께 그룹화되고 파라미터 세트 그룹화 RBSP 의 ID 만이 슬라이스 헤더에서 시그널링되는 기술들을 더욱 상세하게 설명한다.
- [0069] 위에서 논의된 바와 같이, 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은 코더는 비디오 파라미터 세트 및/또는 그룹화 파라미터 세트를 코딩하도록 구성될 수도 있다. 비디오 파라미터 세트의 예들은 도 5 에 대하여 더욱 상세하게 논의되는 한편, 그룹화 파라미터 세트의 예들은 도 6 에 대하여 더욱 상세하게 논의된다.
- [0070] 비디오 인코더 (20) 는 블록-기반 신팩스 데이터, 프레임-기반 신팩스 데이터, 및 GOP-기반 신팩스 데이터와 같은 신팩스 데이터를, 예를 들어, 프레임 헤더, 블록 헤더, 슬라이스 헤더, 또는 GOP 헤더에서 비디오 디코더 (30) 로 더 전송할 수도 있다. GOP 신팩스 데이터는 각각의 GOP 에서의 다수의 프레임들을 설명할 수도 있고, 프레임 신팩스 데이터는 대응하는 프레임을 인코딩하기 위하여 이용되는 인코딩/예측 모드를 표시할 수도 있다.
- [0071] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각, 적용 가능한 경우, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적 회로 (ASIC) 들, 필드 프로그래밍 가능 게이트 어레이 (FPGA) 들, 이산 로직 회로부, 소프트웨어, 하드웨어, 펌웨어 또는 그 임의의 조합들과 같은 다양한 적당한 인코더 또는 디코더 회로부 중 임의의 것으로서 구현될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 그 중 어느 하나는 조합된 비디오 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 셀룰러 전화와 같은 무선 통신 디바이스를 포함할 수도 있다.
- [0072] 도 2 는 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 NAL 단위들을 코딩하기 위한 기술들을 구현할 수도 있는 비디오 인코더 (20) 의 예를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 소정의 비디오 프레임 또는 꾹쳐 내에서 비디오에 있어서의 공간적 중복성을 감소시키거나 제거하기 위한 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 꾹쳐들 내에서 비디오에 있어서의 시간적 중복성을 감소시키거나 제거하기 위한 시간적 예측에 의존한다. 인트라-모드 (I-모드) 는 몇몇 공간 기반 코딩 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 (uni-directional) 예측 (P 모드) 또는 양-예측 (bi-prediction) (B 모드) 과 같은 인터-모드들은 몇몇 시간 기반 코딩 모드들 중 임의의 것을 지칭할 수도 있다.
- [0073] 도 2 에 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩될 비디오 프레임 내의 현재의 비디오 블록을 수신한다. 도 2 의 예에서, 비디오 인코더 (20) 는 모드 선택 유닛 (40), 참조 꾹처 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 모드 선택 유닛 (40) 은 궁극적으로, 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라 예측 유닛 (46), 및 파티션 유닛 (48) 을 포함한다. 비디오 블록 재구성을 위하여, 비디오 인코더 (20) 는 역양자화 유닛 (58), 역변환 유닛 (60), 및 합산기 (62) 를 또한 포함한다. 디블록킹 필터 (도 2 에 도시되지 않음) 는 블록 경계들을 필터링하여 재구성된 비디오로부터 블록화 아티팩트 (blockiness artifact) 들을 제거하기 위하여 또한 포함될 수도 있다. 희망하는 경우, 디블록킹 필터는 합산기 (62) 의 출력을 전형적으로 필터링할 것이다. (루프 내의 또는 루프 이후의) 추가적인 필터들은 디블록킹 필터에 추가하여 또한 이용될 수도 있다. 이러한 필터들은 간결함을 위하여 도시되어 있지 않지만, 희망하는 경우, (인-루프 (in-loop) 필터로서) 합산기 (50) 의 출력을 필터링할 수도 있다.
- [0074] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 다수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은

시간적 예측을 제공하기 위하여 하나 이상의 참조 프레임들 내의 하나 이상의 블록들에 대한 수신된 비디오 블록의 인터-예측 코딩을 수행한다. 인트라 예측 유닛 (46)은 공간적 예측을 제공하기 위하여 코딩될 블록과 동일한 프레임 또는 슬라이스 내의 하나 이상의 이웃하는 블록들에 대한 수신된 비디오 블록의 인트라-예측 코딩을 대안적으로 수행할 수도 있다. 비디오 인코더 (20)는 예를 들어, 비디오 데이터의 각각의 블록에 대한 적절한 코딩 모드를 선택하기 위하여 다수의 코딩 패스 (coding pass) 들을 수행할 수도 있다.

[0075] 또한, 파티션 유닛 (48)은 이전의 코딩 패스들에서의 이전의 파티셔닝 방식들의 평가에 기초하여 비디오 데이터의 블록들을 서브-블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛 (48)은 초기에 프레임 또는 슬라이스를 LCU 들로 파티셔닝할 수도 있고, 레이트-왜곡 분석 (예를 들어, 레이트-왜곡 최적화)에 기초하여 LCU 들의 각각을 서브-CU 들로 파티셔닝할 수도 있다. 모드 선택 유닛 (40)은 LCU 의 서브-CU 들로의 파티셔닝을 표시하는 큐드트리 데이터 구조를 더 생성할 수도 있다. 큐드트리의 리프-노드 CU 들은 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다.

[0076] 모드 선택 유닛 (40)은 예를 들어, 예의 결과들에 기초하여 코딩 모드들, 인트라 또는 인터 중 하나를 선택할 수도 있고, 인트라-코딩된 또는 인터-코딩된 블록을, 잔차 블록 데이터를 생성하기 위하여 합산기 (50)에, 그리고 참조 프레임으로서의 이용을 위한 인코딩된 블록을 재구성하기 위하여 합산기 (62)에 제공한다. 모드 선택 유닛 (40)은 모션 벡터들, 인트라-모드 표시자들, 파티션 정보, 및 다른 이러한 신택스 정보와 같은 신택스 엘리먼트들을 엔트로피 인코딩 유닛 (56)에 또한 제공한다.

[0077] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 고도로 통합될 수도 있지만, 개념적인 목적들을 위하여 별도로 예시되어 있다. 모션 추정 유닛 (42)에 의해 수행되는 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는 예를 들어, 현재의 프레임 내에서 코딩되고 있는 현재의 블록 (또는 다른 코딩된 단위)에 대한 참조 프레임 내의 예측 블록 (또는 다른 코딩된 단위)에 대하여 현재의 비디오 프레임 또는 픽처 내의 비디오 블록의 PU의 변위를 표시할 수도 있다. 예측 블록은, 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는 픽셀 차이의 측면에서, 코딩될 블록과 근접하게 정합하는 것으로 판명되는 블록이다. 일부 예들에서, 비디오 인코더 (20)는 참조 픽처 메모리 (64)에 저장된 참조 픽처들의 정수-미만 (sub-integer) 픽셀 위치들에 대한 값을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20)는 참조 픽처의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값을 보간할 수도 있다. 그러므로, 모션 추정 유닛 (42)은 전체 픽셀 위치들 및 분수 픽셀 위치들에 대한 모션 검색을 수행할 수도 있고, 분수 픽셀 정밀도 (fractional pixel precision)를 갖는 모션 벡터를 출력할 수도 있다.

[0078] 모션 추정 유닛 (42)은 PU의 위치를 참조 픽처의 예측 블록의 위치와 비교함으로써 인터-코딩된 슬라이스 내의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽처는 제 1 참조 픽처 리스트 (리스트 0) 또는 제 2 참조 픽처 리스트 (리스트 1)로부터 선택될 수도 있고, 이 리스트들의 각각은 참조 픽처 메모리 (64)에 저장된 하나 이상의 참조 픽처들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44)으로 전송한다.

[0079] 모션 보상 유닛 (44)에 의해 수행되는 모션 보상은 모션 추정 유닛 (42)에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페치 (fetch) 또는 생성하는 것을 포함할 수도 있다. 다시, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 일부 예들에서 기능적으로 통합될 수도 있다. 현재의 비디오 블록의 PU에 대한 모션 벡터를 수신할 시에는, 모션 보상 유닛 (44)은 모션 벡터가 참조 픽처 리스트들 중 하나에서 지시하는 예측 블록을 위치시킬 수도 있다. 합산기 (50)는 아래에서 논의되는 바와 같이, 코딩되고 있는 현재의 비디오 블록의 픽셀 값을로부터 예측 블록의 픽셀 값을 감산하여 픽셀 차이 값을 형성함으로써 잔차 비디오 블록을 형성한다. 일반적으로, 모션 추정 유닛 (42)은 루마 컴포넌트들에 대한 모션 추정을 수행하고, 모션 보상 유닛 (44)은 크로마 컴포넌트들 및 루마 컴포넌트들 둘 모두에 대한 루마 컴포넌트들에 기초하여 계산된 모션 벡터들을 이용한다. 모드 선택 유닛 (40)은 비디오 슬라이스의 비디오 블록들을 디코딩할 시에 비디오 디코더 (30)에 의한 이용을 위한 비디오 슬라이스 및 비디오 블록들과 연관된 신택스 엘리먼트들을 또한 생성할 수도 있다.

[0080] 인트라 예측 유닛 (46)은 위에서 설명된 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 수행되는 인터-예측에 대한 대안으로서, 현재의 블록을 인트라-예측할 수도 있다. 특히, 인트라 예측 유닛 (46)은 현재의 블록을 인코딩하기 위하여 이용하기 위한 인트라-예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 유닛 (46)은 예를 들어, 별도의 인코딩 패스들 동안에 다양한 인트라-예측 모드들을 이용하

여 현재의 블록을 인코딩할 수도 있고, 인트라 예측 유닛 (46) (또는 일부 예들에서, 모드 선택 유닛 (40)) 은 테스팅된 모드들로부터 이용하기 위한 적절한 인트라-예측 모드를 선택할 수도 있다.

[0081] 예를 들어, 인트라 예측 유닛 (46)은 다양한 테스팅된 인트라-예측 모드들에 대한 레이트-왜곡 (rate-distortion) 분석을 이용하여 레이트-왜곡 값들을 계산할 수도 있고, 테스팅된 모드들 사이에서 최적의 레이트-왜곡 특성을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로, 인코딩된 블록을 생성하기 위해 이용되는 비트레이트 (즉, 비트들의 수) 뿐만 아니라, 인코딩된 블록과, 인코딩된 블록을 생성하기 위하여 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러) 의 양을 결정한다. 인트라 예측 유닛 (46)은 어느 인트라-예측 모드가 블록에 대한 최적의 레이트-왜곡 값을 나타내는지를 결정하기 위하여, 다양한 인코딩된 블록들에 대한 레이트들 및 왜곡들로부터 비율들을 계산할 수도 있다.

[0082] 블록에 대한 인트라-예측 모드를 선택한 후, 인트라 예측 유닛 (46)은 블록에 대한 선택된 인트라-예측 모드를 표시하는 정보를 엔트로피 인코딩 유닛 (56)에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56)은 선택된 인트라-예측 모드를 표시하는 정보를 인코딩할 수도 있다. 비디오 인코더 (20)는, 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 수정된 인트라-예측 모드 인덱스 테이블들 (또한 코드워드 맵핑 테이블들이라고 지칭됨)을 포함할 수도 있는 송신된 비트스트림 구성 데이터에서, 다양한 블록들에 대한 인코딩 컨텍스트들의 정의들과, 가장 확률이 높은 인트라-예측 모드, 인트라-예측 모드 인덱스 테이블, 및 컨텍스트들의 각각에 대해 이용하기 위한 수정된 인트라-예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.

[0083] 비디오 인코더 (20)는 코딩되고 있는 원래의 비디오 블록으로부터 모드 선택 유닛 (40)으로부터의 예측 데이터를 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50)는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 프로세싱 유닛 (52)은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 유닛 (52)은 DCT 와 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 웨이블렛 변환들, 정수 변환들, 서브-대역 (sub-band) 변환들 또는 다른 타입들의 변환들이 또한 이용될 수 있다.

[0084] 어떤 경우에나, 변환 프로세싱 유닛 (52)은 변환을 잔차 블록에 적용하여, 잔차 변환 계수들의 블록을 생성한다. 변환은 잔차 정보를 픽셀 값 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다. 변환 프로세싱 유닛 (52)은 결과적인 변환 계수들을 양자화 유닛 (54)으로 전송할 수도 있다. 양자화 유닛 (54)은 비트 레이트를 더욱 감소시키기 위하여 변환 계수들을 양자화한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화 정도는 양자화 파라미터를 조절함으로써 수정될 수도 있다. 일부 예들에서는, 다음으로, 양자화 유닛 (54)이 양자화된 변환 계수들을 포함하는 행렬의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56)은 스캔을 수행할 수도 있다.

[0085] 양자화에 후속하여, 엔트로피 인코딩 유닛 (56)은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56)은 컨텍스트 적용 가변 길이 코딩 (CAVLC), 컨텍스트 적용 이진 산술 코딩 (CABAC), 신팩스-기반 컨텍스트-적용 이진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 코딩 기술을 수행할 수도 있다. 컨텍스트-기반 엔트로피 코딩의 경우에는, 컨텍스트가 이웃하는 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56)에 의한 엔트로피 코딩에 후속하여, 인코딩된 비트스트림은 또 다른 디바이스 (예를 들어, 비디오 디코더 (30))로 송신될 수도 있거나, 이후의 송신 또는 취출을 위하여 아카이브될 수도 있다.

[0086] 역양자화 유닛 (58) 및 역변환 유닛 (60)은 예를 들어, 참조 블록으로서의 이후의 이용을 위해 픽셀 도메인에서 잔차 블록을 재구성하기 위하여 역양자화 및 역변환을 각각 적용한다. 모션 보상 유닛 (44)은 잔차 블록을 참조 픽처 메모리 (64)의 프레임들 중 하나의 프레임의 예측 블록에 추가함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44)은 모션 추정 시에 이용하기 위한 정수-미만 픽셀 값을 계산하기 위하여 하나 이상의 보간 필터들을 재구성된 잔차 블록에 또한 적용할 수도 있다. 합산기 (62)는 참조 픽처 메모리 (64)에 저장하기 위한 재구성된 비디오 블록을 생성하기 위하여, 재구성된 잔차 블록을 모션 보상 유닛 (44)에 의해 생성되는 모션 보상된 예측 블록에 추가한다. 재구성된 비디오 블록은 후속 비디오 프레임에서 블록을 인터-코딩하기 위한 참조 블록으로서, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 이용될 수도 있다.

[0087] 비디오 인코더 (20)는 시퀀스 파라미터 세트 (SPS), 픽처 파라미터 세트 (PPS), 적응 파라미터 세트 (APS), 또는 다른 이러한 시그널링 데이터 구조들뿐만 아니라, 본 개시물의 기술들에 따른, 비디오 파라미터 세트 (VPS), 계층 파라미터 세트 (LPS), 및/또는 그룹화 파라미터 세트를 코딩하도록 더 구성될 수도 있다. 더 상세하게

는, 엔트로피 인코딩 유닛 (56) 은 이 데이터 구조들 중 임의의 것 또는 전부를 코딩하도록 구성될 수도 있다.

이 다양한 데이터 구조들의 파라미터들이 코딩 성능에 영향을 줄 수도 있는 한도까지, 모드 선택 유닛 (40) 은 적절한 파라미터들을 선택할 수도 있고, 예를 들어, VPS 내에서의 포함을 위하여 파라미터들을 엔트로피 인코딩 유닛 (56) 에 전달할 수도 있다. 시간적 계층들의 수, 재순서화될 픽처들의 수, 및 디코딩된 픽처 버퍼에 저장될 픽처들의 수와 같은 다른 파라미터들이 사용자, 예를 들어, 관리자에 의해 선택될 수도 있다. 다른 예들에서, HRD 파라미터들과 같은 어떤 파라미터들은 인코딩 프로세스를 통해 발생할 수도 있다.

[0088] 엔트로피 인코딩 유닛 (56) 은 본 개시물에 의해 설명된 다양한 타입들의 데이터 중 임의의 것 또는 전부를 포함하기 위하여 VPS 를 코딩할 수도 있다. 비디오 인코더 (20) 는 VPS 의 파라미터들에 따라 데이터를 또한 인코딩할 수도 있다. 더욱 상세하게, 비디오 인코더 (20) 는 VPS 의 파라미터들에 따라 VPS 가 대응하는 비디오 데이터의 하나 이상의 계층들 사이에서 픽처들의 시퀀스들을 코딩할 수도 있다.

[0089] 이러한 방식으로, 도 2 의 비디오 인코더 (20) 는 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하도록 구성된 비디오 인코더의 예를 나타내고, 여기서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하고, VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩 한다.

[0090] 비디오 인코더에 대하여 일반적으로 설명되었지만, VPS 의 인코딩은 다른 디바이스들, 예를 들어, 미디어-인지 네트워크 엘리먼트 (media-aware network element; MANE) 에 의해 수행될 수도 있다. MANE 는 (도 1 의 소스 디바이스 (12) 와 같은) 소스 디바이스와 (목적지 디바이스 (14) 와 같은) 목적지 디바이스 사이의 네트워크 엘리먼트에 대응할 수도 있다. MANE 는 본 개시물의 기술들에 따라 VPS 를 인코딩하도록 구성될 수도 있다.

MANE 는 MANE 에 의해 수신된 다른 데이터 구조들, 예를 들어, 시퀀스 파라미터 세트들의 데이터를 이용하여 VPS 를 생성할 수도 있다.

[0091] 도 3 은 비디오 데이터의 하나 이상의 계층들에 대한 파라미터 세트들 및 NAL 단위들을 코딩하기 위한 기술들을 구현할 수도 있는 비디오 디코더 (30) 의 예를 예시하는 블록도이다. 도 3 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (70), 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 역양자화 유닛 (76), 역변환 유닛 (78), 참조 픽처 메모리 (82) 및 합산기 (80) 를 포함한다. 참조 픽처 메모리 (82) 는 "디코딩된 픽처 버퍼" 또는 DPB 라고 또한 지칭될 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 비디오 인코더 (20; 도 2) 에 대하여 설명된 인코딩 패스와 일반적으로 상반되는 디코딩 패스를 수행할 수도 있다. 모션 보상 유닛 (72) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있는 반면, 인트라 예측 유닛 (74) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 인트라-예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.

[0092] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들을 나타내는 인코딩된 비디오 비트스트림 및 연관된 선택스 엘리먼트들을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 양자화된 계수들, 모션 벡터들 또는 인트라-예측 모드 표시자들, 및 다른 선택스 엘리먼트들을 생성하기 위하여 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (70) 은 모션 벡터들 및 다른 선택스 엘리먼트들을 모션 보상 유닛 (72) 으로 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 선택스 엘리먼트들을 수신할 수도 있다.

[0093] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 인트라 예측 유닛 (74) 은 현재의 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터 및 시그널링된 인트라 예측 모드에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B, P 또는 GPB) 슬라이스로서 코딩될 때, 모션 보상 유닛 (72) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 모션 벡터들 및 다른 선택스 엘리먼트들에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 픽처 리스트들 중 하나 내에서 참조 픽처들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는 참조 픽처 메모리 (82) 에 저장된 참조 픽처들에 기초하여 디폴트 구성 기술들을 이용하여 참조 프레임 리스트들, 리스트 0 및 리스트 1 을 구성할 수도 있다.

[0094] 모션 보상 유닛 (72) 은 모션 벡터들 및 다른 선택스 엘리먼트들을 과정함으로써 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 디코딩되고 있는 현재의 비디오 블록에 대한 예측 블록들을 생성하기 위하여 예측 정보를 이용한다. 예를 들어, 모션 보상 유닛 (72) 은 비디오 슬라이스의 비디오 블록들을 코딩하기 위해 이용되는 예측 모드 (예를 들어, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 타입 (예를 들어, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 픽처 리스트들 중 하나 이상에 대한

구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재의 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위하여 수신된 신택스 엘리먼트들 중 일부를 이용한다.

[0095] 모션 보상 유닛 (72)은 보간 필터들에 기초하여 보간을 또한 수행할 수도 있다. 모션 보상 유닛 (72)은 참조 블록들의 정수-미만 픽셀들에 대한 보간된 값들을 계산하기 위하여 비디오 블록들의 인코딩 동안에 비디오 인코더 (20)에 의해 이용되는 바와 보간 필터들을 이용할 수도 있다. 이 경우, 모션 보상 유닛 (72)은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 이용되는 보간 필터들을 결정할 수도 있고, 예측 블록들을 생성하기 위하여 보간 필터들을 이용할 수도 있다.

[0096] 역양자화 유닛 (76)은 비트스트림에서 제공되며 엔트로피 디코딩 유닛 (80)에 의해 디코딩되는 양자화된 변환 계수들을 역양자화, 즉, 양자화해제 (de-quantize) 한다. 역양자화 프로세스는 양자화의 정도와, 마찬가지로, 적용되어야 할 역양자화의 정도를 결정하기 위하여 비디오 슬라이스 내의 각각의 비디오 블록에 대한 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 QPY의 이용을 포함할 수도 있다. 역변환 유닛 (78)은 픽셀 도메인에서 잔차 블록들을 생성하기 위하여, 역변환, 예를 들어, 역DCT, 역정수 변환, 또는 개념적으로 유사한 역변환 프로세스를 변환 계수들에 적용한다.

[0097] 모션 보상 유닛 (72)이 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30)는 역변환 유닛 (78)으로부터의 잔차 블록들을 모션 보상 유닛 (72)에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (90)는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 희망하는 경우, 디블록킹 필터는 블록화 아티팩트들을 제거하기 위하여 디코딩된 블록들을 필터링하도록 또한 적용될 수도 있다. (코딩 루프 내의 또는 코딩 루프 이후의) 다른 루프 필터들은 픽셀 전이 (transition)들을 평활화하기 위하여, 또는 이와 달리 비디오 품질을 개선시키기 위하여 또한 이용될 수도 있다. 다음으로, 소정의 프레임 또는 픽처 내의 디코딩된 비디오 블록들은 후속 모션 보상을 위해 이용되는 참조 픽처들을 저장하는 참조 픽처 메모리 (82)에 저장된다.

참조 픽처 메모리 (82)는 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에서의 이후의 프리젠테이션을 위한 디코딩된 비디오를 또한 저장한다.

[0098] 본 개시물의 기술들에 따르면, 비디오 디코더 (30)는 시퀀스 파라미터 세트 (SPS), 픽처 파라미터 세트 (PPS), 적응 파라미터 세트 (APS), 또는 다른 이러한 시그널링 데이터 구조들뿐만 아니라, 본 개시물의 기술들에 따른, 비디오 파라미터 세트 (VPS), 계층 파라미터 세트 (LPS), 및/또는 그룹화 파라미터 세트를 디코딩할 수도 있다. 더 상세하게는, 엔트로피 디코딩 유닛 (70)은 이 데이터 구조들 중 임의의 것 또는 전부를 디코딩하도록 구성될 수도 있다. 이 다양한 데이터 구조들을 디코딩함으로써, 엔트로피 디코딩 유닛 (70)은 대응하는 비디오 데이터를 디코딩하기 위하여 이용될 파라미터들을 결정할 수도 있다. 예를 들어, 비디오 디코더 (30)는 디코딩된 VPS의 파라미터들을 이용하여 하나 이상의 계층들의 비디오 데이터의 대응하는 시퀀스들을 디코딩할 수도 있다.

[0099] 도 3에 도시되지 않았지만, 비디오 디코더 (30)는 코딩된 픽처 버퍼 (CPB)를 추가적으로 포함할 수도 있다. CPB는 보통 엔트로피 디코딩 유닛 (70)이전에 제공될 것이다. 대안적으로, CPB는 일시적인 저장을 위하여 엔트로피 디코딩 유닛 (70)에 결합될 수도 있고, 또는 이러한 데이터가 디코딩될 때까지 엔트로피-디코딩된 데이터를 저장하기 위하여 엔트로피 디코딩 유닛 (70)의 출력에서 결합될 수도 있다. 일반적으로, CPB는, 예를 들어, 비디오 디코더 (30)가 디코딩된 VPS로부터 추출할 수도 있는 HRD 파라미터들에 의해 표시된 바와 같이, 코딩된 비디오 데이터가 디코딩되어야 할 때까지 코딩된 비디오 데이터를 저장한다. 마찬가지로, 비디오 디코더 (30)의 다른 엘리먼트들은 예를 들어, VPS를 이용하여 비디오 데이터를 디코딩하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30)는 다양한 시간적 계층들의 픽처들에 대한 시간적 식별자들, 재순서화될 및/또는 참조 픽처 메모리 (82; DPB를 나타냄)에 저장될 픽처들의 수를 표시하는 데이터를 디코딩할 수도 있다.

[0100] 또한, 비디오 디코더 (30)는 비디오 코딩 표준의 확장들에 의해 제공되는 다양한 코딩 툴들에 따라 비디오 데이터를 프로세싱하기 위한 추가적인 프로세싱 유닛들을 포함할 수도 있다. 대안적으로, 도 3에 도시된 비디오 디코더 (30)의 현존하는 엘리먼트들은 이러한 확장들의 코딩 툴들을 실행하도록 구성될 수도 있다. 엔트로피 디코딩 유닛 (70)은 VPS 확장 데이터를 디코딩하고, 이러한 확장 데이터를, 확장들에 의해 제공되는 코딩 툴들을 실행하도록 구성된 유닛들에 제공하도록 구성될 수도 있다.

[0101] 이러한 방식으로, 도 3의 비디오 디코더 (30)는 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터

세트 (VPS) 를 코딩하도록 구성된 비디오 디코더의 예를 나타내고, 여기서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하고, VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩 한다.

[0102] 비디오 디코더에 대하여 일반적으로 설명되었지만, VPS 의 디코딩은 다른 디바이스들, 예를 들어, 미디어-인지 네트워크 엘리먼트 (MANE) 에 의해 수행될 수도 있다. MANE 는 본 개시물의 기술들에 따라 VPS 를 디코딩하도록 구성될 수도 있다. MANE 는 VPS 의 데이터를 이용하여, 하나 이상의 시퀀스 파라미터 세트들과 같은 다른 파라미터 세트 데이터를 더 생성할 수도 있다. 이러한 방식으로, MANE 는 ITU-T H.264/AVC 와 같은 이전의 표준들과의 역-호환성을 제공할 수도 있다.

[0103] 도 4 는 일 예의 MVC 예측 패턴을 예시하는 개념적인 도면이다. 멀티-뷰 비디오 코딩 (Multi-view video coding; MVC) 은 ITU-T H.264/AVC 의 확장이다. 유사한 기술이 HEVC 에 적용될 수도 있다. 도 4 의 예에서는, 8 개의 뷰들 (뷰 ID 들 "S0" 내지 "S7" 을 가짐) 이 예시되어 있고, 12 개의 시간적 위치들 ("T0" 내지 "T11") 이 각각의 뷰에 대해 예시되어 있다. 즉, 도 4 의 각각의 행은 뷰에 대응하는 반면, 각각의 열은 시간적 위치를 표시한다.

[0104] 멀티-뷰 비디오 코딩을 위한 전형적인 MVC 예측 (각각의 뷰 내에서의 인터-픽처 예측 및 인터-뷰 예측의 둘 모두를 포함함) 구조가 도 4 에 도시되어 있고, 여기서, 예측들은 화살표들에 의해 표시되고, 지시-대상인 (pointed-to) 객체는 예측 참조를 위한 지시-원천인 (pointed-from) 객체를 이용한다. MVC 에서, 인터-뷰 예측은 H.264/AVC 모션 보상의 신팩스를 이용할 수도 있는 디스파리티 (disparity) 모션 보상에 의해 지원되지만, 상이한 뷰 내의 픽처가 참조 픽처로서 이용되도록 한다.

[0105] 2 개의 뷰들의 코딩은 MVC 에 의해 또한 지원될 수 있고, MVC 의 장점들 중 하나는, MVC 인코더가 2 개를 초과하는 뷰들을 3D 비디오 입력으로서 취할 수 있고 MVC 디코더는 이러한 멀티뷰 표현을 디코딩할 수 있다는 점이다. 따라서, MVC 디코더를 갖는 임의의 렌더러 (renderer) 는 2 개를 초과하는 뷰들을 갖는 3D 비디오 컨텐츠를 수신하도록 구성될 수도 있다.

[0106] MVC 가 H.264/AVC 디코더들에 의해 디코딩 가능한 소위 기본 뷰 (base view) 를 가지며 스테레오 뷰 쌍은 MVC 에 의해 또한 지원될 수 있지만, MVC 의 하나의 장점은, MVC 가 2 개를 초과하는 뷰들을 3D 비디오 입력으로서 이용하며 다수의 뷰들에 의해 표현된 이 3D 비디오를 디코딩하는 예를 지원할 수 있다는 점이다. MVC 디코더를 갖는 클라이언트의 렌더러는 다수의 뷰들을 갖는 3D 비디오 컨텐츠를 예상할 수도 있다.

[0107] 전형적인 MVC 디코딩 순서는 시간-우선 코딩이라고 지칭된다. 액세스 단위는 하나의 출력 시간 인스턴스 (instance) 에 대한 모든 뷰들의 코딩된 픽처들을 포함할 수도 있다. 예를 들어, 시간 (T0) 의 픽처들의 각각은 공통 액세스 단위에 포함될 수도 있고, 시간 (T1) 의 픽처들의 각각은 제 2 의 공통 액세스 단위에 포함될 수도 있고, 이하 등등과 같다. 디코딩 순서는 출력 또는 디스플레이 순서와 반드시 동일하지는 않다.

[0108] 도 4 의 프레임들은, 대응하는 프레임이 인트라-코딩되는지 (즉, I-프레임), 또는 하나의 방향에서 (즉, P-프레임으로서) 또는 다수의 방향들에서 (즉, B-프레임으로서) 인터-코딩되는지 여부를 지정하는 글자를 포함하는 음영처리된 블록을 이용하여 도 4 의 각각의 행 및 각각의 열의 교차점에서 표시된다. 일반적으로, 예측들은 화살표들에 의해 표시되고, 여기서, 지시-대상인 프레임은 예측 참조를 위한 지시-원천인 객체를 이용한다. 예를 들어, 시간적 위치 (T0) 에서의 뷰 (S2) 의 P-프레임은 시간적 위치 (T0) 에서의 뷰 (S0) 의 I-프레임으로부터 예측된다.

[0109] 단일 뷰 비디오 인코딩에서와 마찬가지로, 멀티뷰 비디오 코딩 비디오 시퀀스의 프레임들은 상이한 시간적 위치들에서 프레임들에 대하여 예측 방식으로 인코딩될 수도 있다. 예를 들어, 시간적 위치 (T1) 에서의 뷰 (S0) 의 b-프레임은 시간적 위치 (T0) 에서 뷰 (S0) 의 I-프레임으로부터 자신에게 지시된 화살표를 가지고, 이것은 b-프레임이 I-프레임으로부터 예측되는 것을 표시한다. 그러나, 추가적으로, 멀티뷰 비디오 인코딩의 상황에서는, 프레임들이 인터-뷰 예측될 수도 있다. 즉, 뷰 컴포넌트는 참조를 위한 다른 뷰들 내의 뷰 컴포넌트들을 이용할 수 있다. 예를 들어, MVC 에서는, 인터-뷰 예측이 또 다른 뷰 내의 뷰 컴포넌트가 인터-예측 참조인 것처럼 실현된다. 잠재적인 인터-뷰 참조들은 시퀀스 파라미터 세트 (SPS) MVC 확장부에서 시그널링될 수도 있으며 참조 픽처 리스트 구성 프로세스에 의해 수정될 수 있고, 이것은 인터-예측 또는 인터-뷰 예측 참조들의 유연성 있는 순서화를 가능하게 한다.

[0110] H.264/AVC 의 MVC 확장부에서는, 일 예로서, 인터-뷰 예측이, H.264/AVC 모션 보상의 신팩스를 이용하는 디스파리티 모션 보상에 의해 지원되지만, 상이한 뷰 내의 픽처가 참조 픽처로서 이용되도록 한다. 2 개의 뷰들의

코딩은 입체적 뷰들이라고 일반적으로 지칭되는 MVC 에 의해 지원될 수 있다. MVC 의 장점들 중 하나는, MVC 인코더가 2 개를 초과하는 뷰들을 3D 비디오 입력으로서 취할 수 있고, MVC 디코더는 이러한 멀티뷰 표현을 디코딩할 수 있다는 점이다. 따라서, MVC 디코더를 갖는 렌더링 디바이스는 2 개를 초과하는 뷰들을 갖는 3D 비디오 컨텐츠들을 예상할 수도 있다.

[0111] MVC 에서, 인터-뷰 예측 (inter-view prediction; IVP) 은 동일한 액세스 단위 (즉, 동일한 시간 인스턴스를 가짐) 에서 픽처들 사이에서 허용된다. 액세스 단위는 일반적으로, 공통의 시간적 인스턴스에 대한 모든 뷰 컴포넌트들 (예를 들어, 모든 NAL 단위들) 을 포함하는 데이터의 단위이다. 따라서, MVC 에서는, 인터-뷰 예측이 동일한 액세스 단위에서 픽처들 사이에서 허용된다. 비-기본 (non-base) 뷰들 중 하나 내의 픽처를 코딩할 때, 픽처가 상이한 뷰 내에 있지만 동일한 시간 인스턴스를 가질 경우 (예를 들어, 동일한 POC 값, 및 이에 따라 동일한 액세스 단위임) 에는, 픽처는 참조 픽처 리스트 내로 추가될 수도 있다. 인터-뷰 예측 참조 픽처는 임의의 인터 예측 참조 픽처처럼, 참조 픽처 리스트의 임의의 위치에 넣어질 수도 있다.

[0112] 멀티-뷰 비디오 코딩의 상황에서는, 2 개의 종류의 모션 벡터들이 있다. 하나는 시간적 참조 픽처들을 지시하는 정상 모션 벡터들이고, 대응하는 인터 예측 모드는 모션 보상된 예측 (motion compensated prediction; MCP) 이라고 지칭된다. 다른 하나는 상이한 뷰 내의 픽처들을 지시하는 디스파리티 모션 벡터들이고, 대응하는 인터-뷰 예측 모드는 디스파리티 보상된 예측 (disparity compensated prediction; DCP) 이라고 지칭된다.

[0113] 기존의 HEVC 에서는, 모션 파라미터들의 예측을 위한 2 개의 모드들이 있고; 하나는 병합 모드 (merge mode) 이고, 다른 하나는 진보된 모션 벡터 예측 (advanced motion vector prediction; AMVP) 이다. 병합 모드에서는, 모션 파라미터들 (참조 픽처들 및 모션 벡터들) 의 후보 리스트가 구성되고, 여기서, 후보는 공간적 또는 시간적으로 이웃하는 블록들로부터 나올 수 있다. 공간적으로 그리고 시간적으로 이웃하는 블록들은 후보 리스트, 즉, 모션 예측 정보가 선택될 수도 있는 후보들의 세트를 형성할 수도 있다. 따라서, 비디오 인코더 (20) 는 후보 리스트로의 인덱스를 코딩함으로써 모션 예측 정보로서 선택된 모션 파라미터들을 코딩할 수도 있다. 비디오 디코더 (30) 가 인덱스를 디코딩한 후, 인덱스가 지시하는 대응하는 블록의 모든 모션 파라미터들은 병합 모드에서 인계될 수도 있다.

[0114] AMVP, 이에 따라 기존의 HEVC 에서는, 각각의 모션 가설에 대한 모션 벡터 예측기들의 후보 리스트가 코딩된 참조 인덱스에 기초하여 유도된다. 이 리스트는 시간적 참조 픽처 내의 함께 위치된 블록의 이웃하는 블록의 모션 파라미터들에 기초하여 유도되는 시간적 모션 벡터 예측기 뿐만 아니라 동일한 참조 인덱스와 연관되는 이웃하는 블록들의 모션 벡터들을 포함한다. 선택된 모션 벡터들은 후보 리스트로의 인덱스를 송신함으로써 시그널링된다. 추가적으로, 참조 인덱스 값들 및 모션 벡터 차이들이 또한 시그널링된다.

[0115] 도 4 는 인터-뷰 예측의 다양한 예들을 제공한다. 도 4 의 예에서, 뷰 (S1) 의 프레임들은 동일한 시간적 위치들에서의 뷰들 (S0 및 S2) 의 프레임들의 프레임들로부터 예측되는 인터-뷰 뿐만 아니라, 뷰 (S1) 의 상이한 시간적 위치들에서의 프레임들로부터 예측되는 것으로 예시되어 있다. 예를 들어, 시간적 위치 (T1) 에서의 뷰 (S1) 의 b-프레임은 시간적 위치 (T1) 에서의 뷰들 (S0 및 S2) 의 b-프레임들 뿐만 아니라, 시간적 위치들 (T0 및 T2) 에서의 뷰 (S1) 의 B-프레임들의 각각으로부터 예측된다.

[0116] 도 4 의 예에서, 대문자 "B" 및 소문자 "b" 는 상이한 인코딩 방법론들이 아니라, 프레임들 사이의 상이한 계측 구조 관계들을 표시하도록 의도된 것이다. 일반적으로, 대문자 "B" 프레임들은 소문자 "b" 프레임들보다 예측 계층구조에서 상대적으로 더 높다. 도 4 는 상이한 레벨들의 음영처리를 이용하여 예측 계층구조에서의 변동들을 또한 예시하고, 여기서, 더 큰 양의 음영처리 (즉, 상대적으로 더 어두운) 프레임들은 더 작은 음영처리 (즉, 상대적으로 더 밝음) 를 갖는 그러한 프레임들보다 예측 계층구조에서 더 높다. 예를 들어, 도 4 의 모든 I-프레임들은 완전 음영처리로 예시되어 있는 반면, P-프레임들은 다소 더 밝은 음영처리를 가지고, B-프레임들 (및 소문자 b-프레임들) 은 서로에 대해 다양한 레벨들의 음영처리를 가지지만, P-프레임들 및 I-프레임들의 음영처리보다 항상 더 밝다.

[0117] 일반적으로, 예측 계층구조에서 상대적으로 더 높은 프레임들은 계층구조에서 상대적으로 더 낮은 프레임들을 디코딩하기 전에 디코딩되어야 하고, 이에 따라, 계층구조에서 상대적으로 더 높은 그러한 프레임들이 계층구조에서 상대적으로 더 낮은 프레임들의 디코딩 동안에 참조 프레임들로서 이용될 수 있다는 점에서, 예측 계층구조는 뷰 순서 인덱스들과 관련된다. 뷰 순서 인덱스는 액세스 단위에서 뷰 컴포넌트들의 디코딩 순서를 표시하는 인덱스이다. 뷰 순서 인덱스들은 H.264/AVC 의 부록 H (MVC 보정) 에서 특정된 바와 같이, SPS MVC 확장부에서 암시되어 있다. SPS 에서는, 각각의 인덱스 i 에 대하여, 대응하는 view_id 가 시그널링된다.

일부 예들에서, 뷰 컴포넌트들의 디코딩은 뷰 순서 인덱스의 상승 순서를 따를 것이다. 모든 뷰들이 제시될 경우, 뷰 순서 인덱스들은 0 으로부터 num_views_minus_1 까지의 연속 순서이다.

[0118] 이러한 방식으로, 참조 프레임들을 참조하여 인코딩되는 프레임들을 디코딩하기 전에, 참조 프레임들로서 이용되는 프레임들이 디코딩될 수도 있다. 뷰 순서 인덱스는 액세스 단위에서 뷰 컴포넌트들의 디코딩 순서를 표시하는 인덱스이다. 각각의 뷰 순서 인덱스 i 에 대하여, 대응하는 view_id 가 시그널링된다. 뷰 컴포넌트들의 디코딩은 뷰 순서 인덱스들의 상승 순서를 따른다. 모든 뷰들이 제시될 경우, 뷰 순서 인덱스들의 세트는 제로로부터 뷰들의 전체 개수보다 하나 적은 것까지의 연속적으로 순서화된 세트를 포함할 수도 있다.

[0119] 계층구조의 동일한 레벨들에서의 어떤 프레임들에 대하여, 디코딩 순서는 서로에 대해 중요하지 않을 수도 있다. 예를 들어, 시간적 위치 (T0) 에서의 뷰 (S0) 의 I-프레임은, 궁극적으로, 시간적 위치 (T0) 에서의 뷰 (S4) 의 P-프레임에 대한 참조 프레임으로서 이용되는, 시간적 위치 (T0) 에서의 뷰 (S2) 의 P-프레임에 대한 참조 프레임으로서 이용된다. 따라서, 시간적 위치 (T0) 에서의 뷰 (S0) 의 I-프레임은, 시간적 위치 (T0) 에서 뷰 (S4) 의 P-프레임 전에 디코딩되어야 하는, 시간적 위치 (T0) 에서의 뷰 (S2) 의 P-프레임 전에 디코딩되어야 한다. 그러나, 뷰들 (S1 및 S3) 사이에서는, 뷰들 (S1 및 S3) 은 예측을 위하여 서로에 대해 의존하지 않지만, 그 대신에, 예측 계층구조에서 더 높은 뷰들로부터만 예측되므로, 디코딩 순서는 중요하지 않다. 또한, 뷰 (S1) 가 뷰들 (S0 및 S2) 이후에 디코딩되기만 하면, 뷰 (S1) 는 뷰 (S4) 이전에 디코딩될 수도 있다.

[0120] 이러한 방식으로, 계층구조 순서화는 뷰들 (S0 내지 S7) 을 설명하기 위하여 이용될 수도 있다. 표기 SA > SB 는 뷰 (SA) 가 뷰 (SB) 이전에 디코딩되어야 함을 의미한다고 하자. 이 표기를 이용하면, 도 4 의 예에서, S0 > S2 > S4 > S6 > S7 이다. 또한, 도 4 의 예에 대하여, S0 > S1, S2 > S1, S2 > S3, S4 > S3, S4 > S5, 및 S6 > S5 이다. 이 요건들을 위반하지 않는 뷰들에 대한 임의의 디코딩 순서가 가능하다. 따라서, 어떤 제한들만을 갖는 많은 상이한 디코딩 순서들이 가능하다.

[0121] 본 개시물의 기술들에 따르면, 뷰들 (S0 - S7) 의 각각은 대응하는 비트스트림의 각각의 계층이라고 간주될 수도 있다. 따라서, VPS 는 뷰들 (S0 - S7) 중 임의의 것 또는 전부에 적용가능한 비트스트림의 파라미터들을 설명할 수도 있는 반면, 개별적인 계층 파라미터 세트들은 뷰들 (S0 - S7) 중 임의의 것 또는 전부에 대해 제공될 수도 있다. 추가적으로, 그룹화 파라미터 세트는 파라미터 세트들의 그룹에 대해 제공될 수도 있어서, 뷰들 (S0 - S7) 의 개별적인 꽂처들 내의 슬라이스들은 그룹화 파라미터 세트의 식별자를 간단하게 참조할 수도 있다.

[0122] 도 4 에 도시된 바와 같이, 뷰 컴포넌트는 참조를 위한 다른 뷰들 내의 뷰 컴포넌트들을 이용할 수 있다. 이것은 인터-뷰 예측이라고 지칭된다. MVC 에서는, 인터-뷰 예측이 또 다른 뷰 내의 뷰 컴포넌트가 인터 예측 참조였던 것처럼 실현된다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 (표 1 의 예에서 도시된 바와 같은) 시퀀스 파라미터 세트 (SPS) MVC 확장부 내의 잠재적인 인터-뷰 참조들을 코딩할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 참조 꽂처 리스트 구성 프로세스를 실행함으로써 잠재적인 인터-뷰 참조들을 더 수정할 수도 있고, 이것은 인터 예측 또는 인터-뷰 예측 참조들의 유연성 있는 순서화를 가능하게 할 수도 있다.

표 1

	C	디스크립터
num_views_minus1	0	ue(v)
for(i = 0; i <= num_views_minus1; i++)		
view_id[i]	0	ue(v)
for(i = 1; i <= num_views_minus1; i++) {		
num_anchor_refs_10[i]	0	ue(v)
for(j = 0; j < num_anchor_refs_10[i]; j++)		
anchor_ref_10[i][j]	0	ue(v)
num_anchor_refs_11[i]	0	ue(v)
for(j = 0; j < num_anchor_refs_11[i]; j++)		
anchor_ref_11[i][j]	0	ue(v)
}		
for(i = 1; i <= num_views_minus1; i++) {		
num_non_anchor_refs_10[i]	0	ue(v)
for(j = 0; j < num_non_anchor_refs_10[i]; j++)		
non_anchor_ref_10[i][j]	0	ue(v)
num_non_anchor_refs_11[i]	0	ue(v)
for(j = 0; j < num_non_anchor_refs_11[i]; j++)		
non_anchor_ref_11[i][j]	0	ue(v)
}		
num_level_values_signaled_minus1	0	ue(v)
for(i = 0; i <= num_level_values_signaled_minus1; i++) {		
level_idc[i]	0	u(8)
num_applicable_ops_minus1[i]	0	ue(v)
for(j = 0; j <= num_applicable_ops_minus1[i]; j++) {		
applicable_op_temporal_id[i][j]	0	u(3)
applicable_op_num_target_views_minus1[i][j]	0	ue(v)
for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++)		
applicable_op_target_view_id[i][j][k]	0	ue(v)
applicable_op_num_views_minus1[i][j]	0	ue(v)
}		
}		
}		

[0123]

[0124] 표 1에 도시된 SPS MVC 확장부에서는, 각각의 뷰에 대하여, 참조 픽처 리스트 0 및 참조 픽처 리스트 1을 형성하기 위하여 이용될 수 있는 뷰들의 수가 시그널링된다. SPS MVC 확장부에서 시그널링되는 바와 같은 앵커 픽처 (anchor picture)에 대한 예측 관계는 동일한 뷰의 (SPS MVC 확장부에서 시그널링된) 비-앵커 픽처 (non-anchor picture)에 대한 예측 관계와는 상이할 수 있다.

[0125]

비디오 코딩 표준들은, 그 스케일러블 비디오 코딩 (Scalable Video Coding; SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장들을 포함하는, ITU-T H.261, ISO/IEC MPEG-1 비쥬얼 (Visual), ITU-T H.262 또는 ISO/IEC MPEG-2 비쥬얼, ITU-T H.263, ISO/IEC MPEG-4 비쥬얼 및 ITU-T H.264 (ISO/IEC MPEG-4 AVC라고 또한 알려짐)를 포함한다.

[0126]

추가적으로, ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 및 ISO/IEC 모션 픽처 전문가 그룹 (Motion Picture Experts Group; MPEG)의 비디오 코딩에 관한 합동 협력 팀 (Joint Collaboration Team on Video Coding; JCT-VC)에 의해 개발되고 있는 새로운 비디오 코딩 표준, 즉, 고효율 비디오 코딩 (HEVC)이 있다. 이후에 HEVC WD4라고 지칭되는 HEVC의 최근의 작업 초안 (Working Draft; WD)은 HEVC WD4d1로서 나타낸 http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F803-v3.zip으로부터 입수가능하다.

[0127]

시퀀스 및 픽처 파라미터 세트 메커니즘은 코딩된 블록 데이터의 송신으로부터 딜 빈번하게 변화하는 정보의 송신을 결합해제한다. 시퀀스 및 픽처 파라미터 세트들은 일부 응용들에서, 신뢰성 있는 전송 메커니즘을 이용하여 "대역외 (out-of-band)"로 전달될 수도 있다. 픽처 파라미터 세트 원시 바이트 시퀀스 페이로드

(raw byte sequence payload; RBSP) 는 하나 이상의 코딩된 픽처들의 코딩된 슬라이스 네트워크 추상화 계층 (NAL) 단위들에 의해 참조될 수 있는 파라미터들을 포함할 수도 있다. 시퀀스 파라미터 세트 RBSP 는 하나 이상의 픽처 파라미터 세트 RBSP 들, 또는 베퍼링 주기 SEI 메시지를 포함하는 하나 이상의 보충 강화 정보 (supplemental enhancement information; SEI) NAL 단위들에 의해 참조될 수 있는 파라미터들을 포함할 수도 있다.

[0128] 시퀀스 파라미터 세트는 비디오 유용성 정보 (VUI) 라고 칭해지는 선택적인 파라미터들의 세트를 포함할 수도 있다. VUI 는 선택적인 정보의 다음의 3 개의 범주들을 포함할 수도 있다: 비디오 표현 정보, 가상 참조 디코더 (HRD) 정보, 및 비트스트림 한정 정보. 비디오 표현 정보는 종횡비, 루마에 대한 컬러 공간 변환 관련 정보 크로마 위상 시프트들 및 프레임레이트를 포함한다. HRD 는 코딩된 비디오 시퀀스들에 대한 비디오 베퍼링 파라미터들을 포함한다. 비트스트림 한정 정보는 모션 벡터 범위, 디코딩된 픽처 베퍼 (DPB) 사이즈, 및 재순서화 프레임들의 수, 및 블록들 (예를 들어, 매크로블록들 또는 코딩 단위 (CU) 들) 및 픽처들의 코딩된 사이즈들을 포함한다.

[0129] HEVC WD5 는 적응 파라미터 세트 (APS) 지원을 포함한다. 적응 파라미터 세트 (APS) 의 개념은 http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F747-v4.zip 으로부터 입수 가능한 JCTVC-F747 에서 또한 발견될 수 있다.

[0130] 통합된 NAL 단위 헤더는 HEVC 의 잠재적인 스케일러를 또는 멀티뷰 확장들을 준수하는 스케일러를 비트스트림들 뿐만 아니라 HEVC 비-스케일러를 비트스트림들의 둘 모두에 대해 이용될 수도 있다. 통합된 NAL 단위 헤더는 다음의 양태들에서 현재의 HEVC NAL 단위 헤더와는 상이할 수도 있다: 하나의 전체적인 코딩된 비디오 시퀀스에 대하여 고정된 NAL 단위 헤더 길이가 있을 수도 있는 반면, 그 길이는 상이한 코딩된 비디오 시퀀스들과, NAL 단위 헤더에서의 스케일러빌리티 신택스 엘리먼트들의 효율적인 코딩에 걸쳐, 그리고 필요하지 않은 특정한 신택스 엘리먼트가 존재할 필요가 없을 때에 변동될 수 있다. 이러한 설계에서는, 상이한 NAL 단위 태입 또는 파라미터 세트가 전체적인 비트스트림에 대해 이용될 수 있다.

[0131] 도 5 는 비디오 파라미터 세트 (VPS) 및 다양한 계층 파라미터 세트 (LPS) 들을 예시하는 개념적인 도면이다. 도 5 의 두 번째 LPS 를 따르는 타원들은 임의의 수 N 개의 VPS 들이 있을 수도 있음을 표시하도록 의도된 것이고, 여기서 N 은 정수이다. 예를 들어, 각각의 계층 (예를 들어, 각각의 SVC 계층 또는 각각의 MVC 뷰) 은 대응하는 LPS 를 가질 수도 있다. 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은 비디오 코더는 도 5 에 예시된 것들과 같은 VPS 및 하나 이상의 LPS 들을 코딩하도록 구성될 수도 있다.

[0132] 아래의 표 2 는 VPS 에 대한 일 예의 원시 바이트 시퀀스 페이로드 (RBPS) 신택스를 제공한다.

표 2

video_parameter_set_rbsp()	디스크립터
video_para_set_id	u(8)
// sample dimension counter description	
cnt_p	u(3)
cnt_d	u(3)
cnt_t	u(3)
cnt_q	u(3)
cnt_v	u(4)
cnt_f	u(4)
// sample index to characteristic mapping	
for(i = 0; i < cnt_d; i++) {	
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
}	
for(i = 0; i < cnt_t; i++)	
average_frame_rate[i]	u(16)
if(cnt_v > 1)	
for(i=0; i < cnt_v; i++)	
view_id[i]	ue(v)
// control parameters and tool enabling/disabling flags	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
chroma_pred_from_luma_enabled_flag	u(1)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
temporal_id_nesting_flag	u(1)
inter_4x4_enabled_flag	u(1)
operation_point_desription()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0133]

[0134] 비디오 코더들은, 코딩된 비디오 시퀀스 (예를 들어, 하나 이상의 계층들을 포함하는 비트스트림) 가 하나의 활성 비디오 파라미터 세트 (VPS) 를 가지기만 할 수 있도록 구성될 수도 있다. VPS 는 특정한 타입의 NAL 단위 내에 캡슐화 (encapsulate) 될 수도 있다. 예를 들어, VPS RBSP 에 대한 nal_unit_type 은 10 일 수도 있다. 표 2 의 VPS 에 대한 일 예의 시멘틱들은 아래에 설명된다:

[0135] 이 예에서, video_para_set_id 는 대응하는 비디오 파라미터 세트 (VPS) 를 식별한다.

[0136] 이 예에서, cnt_p 는 대응하는 코딩된 비디오 시퀀스에서 존재하는 priority_id 값들의 최대 수를 특정한다.

[0137] 이 예에서, cnt_d 는 대응하는 코딩된 비디오 시퀀스에서 존재하는 종속성 계층들의 최대 수를 특정한다. 동일한 해상도를 갖는 다수의 뷰들은 동일한 종속성 계층에 속하는 것으로 간주될 수도 있다. 2 개의 종속성 계층들은 동일한 공간 해상도를 가질 수도 있다.

[0138] 이 예에서, cnt_t 는 코딩 비디오 시퀀스에서 존재하는 시간적 계층들의 최대 수를 특정한다.

[0139] 이 예에서, cnt_q 는 코딩된 비디오 시퀀스 내의 종속성 계층에서 존재하는 품질 계층들의 최대 수를 특정한다.

[0140] 이 예에서, cnt_v 는 코딩된 비디오 시퀀스에서 존재하는 뷰들의 최대 수를 특정한다.

[0141] 이 예에서, cnt_f 는 NAL 단위 헤더에서 reserved_flags 신택스 엘리먼트를 나타내기 위하여 이용된 비트들의 수를 특정한다.

- [0142] 이 예에서, `pic_width_in_luma_samples[i]` 및 `pic_height_in_luma_samples[i]` 는 루마 샘플들의 단위들로 i 번째 종속성 계층 해상도의 폭 및 높이를 각각 특정한다.
- [0143] 이 예에서, `bit_depth_luma_minus8[i]` 플러스 8 및 `bit_depth_chroma_minus8[i]` 플러스 8 은 i 번째 심도 표현의 루마 및 크로마 컴포넌트들의 비트 심도를 특정한다.
- [0144] 이 예에서, `chroma_format_idc[i]` 는 i 번째 크로마 샘플 포맷 표현의 크로마 샘플 포맷을 특정한다. 예를 들어, 0 과 동일한 값은 4:2:0 을 표시할 수도 있고; 1 과 동일한 값은 4:4:4 를 표시할 수도 있고, 2 와 동일한 값은 4:2:2 를 표시할 수도 있고, 3 과 동일한 값은 4:0:0 을 표시할 수도 있다.
- [0145] 이 예에서, `average_frame_rate[i]` 은 256 초당 프레임들의 단위들로, i 번째 시간적 계층 표현의 평균 프레임 레이트를 특정한다.
- [0146] 이 예에서, `view_id[i]` 는 i 와 동일한 뷰 순서 인덱스를 가지는 i 번째 뷰의 뷰 식별자를 특정한다. 존재하지 않을 때, `view_id[0]` 의 값은 0 인 것으로 추론될 수도 있다. 0 과 동일한 `vps_extension_flag` 는 `vps_extension_data_flag` 신택스 엘리먼트들이 비디오 파라미터 세트 RBSP 에서 존재하지 않는 것으로 특정한다. `vps_extension_flag` 는 당면한 HEVC 표준을 준수하는 비트스트림들에서 0 과 동일할 수도 있다. `vps_extension_flag` 에 대한 1 의 값은 예를 들어, ITU-T | ISO/IEC 에 의한 추후의 이용을 위하여 반전될 수도 있다. 비디오 디코더 (30) 와 같은 디코더들은 비디오 파라미터 세트 NAL 단위에서 `vps_extension_flag` 에 대한 값 1 을 따르는 모든 데이터를 무시할 수도 있다.
- [0147] 이 예에서, `vps_extension_data_flag` 는 임의의 값을 가질 수도 있다. 그것은 당면한 HEVC 표준에서 특정된 프로파일들의 준수에 영향을 주지 않지만, 당면한 표준의 추가의 개발을 허용한다.
- [0148] VPS 에서의 다른 신택스 엘리먼트들은 현재의 HEVC 작업 초안의 SPS 에서의 동일한 명칭들을 갖는 신택스 엘리먼트들과 동일한 시맨틱들을 가질 수도 있다. 그러한 신택스 엘리먼트들은, 더 하위 레벨 파라미터 세트들에 의해 겹쳐 쓰이지 않을 경우에, 이 VPS 를 참조하는 코딩된 비디오 시퀀스에 적용될 수도 있다.
- [0149] 일부 예들에서, `3DV_flag` 는 심도가 코딩된 비디오 시퀀스에서 존재하는지를 표시하기 위하여 VPS 에서 더 시그널링될 수도 있다.
- [0150] 일부 예들에서, VUI 파라미터들은 LPS 에서 시그널링된다.
- [0151] 일부 예들에서, 신택스 엘리먼트들 `cnt_p`, `cnt_t`, `cnt_d`, `cnt_q`, 및 `cnt_v` 는 `priority_id`, `temporal_id`, `dependency_id`, `quality_id` 및 `view_idx` 를 각각 코딩하기 위하여 이용된 비트들의 수들을 특정하고, `priority_id` 값들, 시간적 계층들, 종속성 계층들, 품질 계층들, 및 코딩된 비디오 시퀀스에서 존재하는 뷰들의 최대 수들은 VPS 에서 또한 시그널링될 수도 있다.
- [0152] 일부 예들에서는, 신택스 엘리먼트들 `cnt_p`, `cnt_t`, `cnt_d`, `cnt_q`, `cnt_v` 및 `cnt_f` 를 포함하기 위하여 NAL 단위의 또 다른 타입이 도입될 수도 있다. 이 새로운 NAL 단위 타입은 식별자 (ID) 를 또한 포함할 수도 있고, ID는 VPS 에서 참조될 수도 있다.
- [0153] 일부 예들에서, 표 2 의 `log2_max_pic_order_cnt_lsb_minus4` 로부터 `inter_4x4_enabled_flag` 까지의 신택스 엘리먼트들은 VPS 에서 시그널링되지 않지만, 그 대신에, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 LPS 에서의 이 신택스 엘리먼트들을 코딩할 수도 있다.
- [0154] 일부 예들에서, 표 2 의 `operation_point_desription()` 신택스 구조는 VPS 에 포함되지 않고; 그 대신에, 비디오 인코더 (20) 및 비디오 디코더 (30), 또는 다른 엘리먼트들 (예를 들어, 출력 인터페이스 (22) 및/또는 입력 인터페이스 (28)) 은 보충 강화 정보 (SEI) 메시지 내의 `operation_point_desription()` 신택스 구조에서의 컨텐츠를 코딩할 수도 있다.
- [0155] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 VPS 에서의 비디오 유용성 정보 (VUI) 파라미터들을 코딩할 수도 있다. 예를 들어, VPS 는 모션 벡터 범위, DPB 사이즈, 재순서화 프레임들의 수, 및 블록들 (예를 들어, 매크로블록들 또는 CU 들) 및 픽처들의 코딩된 사이즈들에 대한 한정들과 같은, 비트스트림 한정 정보를 특정하는 데이터를 포함할 수도 있다. 이러한 방식으로, VPS 는 (비디오 디코더 (30) 와 같은) 비디오 디코더가 대응하는 비트스트림, 즉, VPS 를 포함하는 비트스트림을 적당하게 디코딩하기 위하여, 요구된 DPB 사이즈를 표시하는 정보를 특정할 수도 있다. 마찬가지로, VPS 는 픽처 재순서화 정보 즉, 디코딩 순서에서 소정의 픽처를 선행할 수도 있으며 출력 순서 (즉, 디스플레이 순서) 에서 소정의 픽처를 후행하는 픽처들

의 수를 특정할 수도 있다.

[0156] 추가적으로 또는 대안적으로, VPS 는 가상 참조 디코더 (HRD) 정보를 특정하는 데이터를 포함할 수도 있다. 위에서 언급된 바와 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 VPS 에서의, HRD 정보를 포함할 수도 있는 VUI 파라미터들을 코딩 (즉, 시그널링) 할 수도 있다. 따라서, VPS 는 예를 들어, 대응하는 비트 스트림의 동작점을 설명하는 데이터를 포함할 수도 있다. 예를 들어, VPS 는 최대 동작점들의 수, 상이한 계층들 또는 뷰들 사이의 종속성들, 각각의 동작점에 대한 프로파일 및 레벨 정보, 각각의 동작점에 대한 동작점 VCL NAL 단위, 각각의 동작점에 대한 비트 레이트, 동작점들 사이의 종속성, 각각의 동작점에 대한 한정들, 각각의 동작점에 대한 VUI 또는 부분 VUI, 및/또는 각각의 계층 또는 뷰에 대한 VUI 또는 부분 VUI 중 하나 이상을 설명하는 데이터를 포함할 수도 있다.

[0157] VPS 는 각각의 차원에 대하여: 특정 인덱스 값, 인덱스 값들의 범위, 또는 인덱스 값들의 리스트를 또한 포함할 수도 있다. 예를 들어, VPS 가 특정 인덱스 값을 설명하는 데이터를 포함할 때, 인덱스 값은 공간 해상도에 대하여, 크로마 샘플링 포맷에 대한 비트 심도에 대응할 수도 있다. 또 다른 예로서, VPS 가 인덱스 값들의 범위를 포함할 때, 시간적 계층들에 대하여, 범위는 제로 (0) 내지 최고 시간적 계층 ID 를 포함할 수도 있고, 품질 계층들에 대하여, 범위는 제로 (0) 내지 최고 품질 계층 ID 를 포함할 수도 있다. 또 다른 예로서, VPS 가 인덱스 값들의 리스트를 설명하는 데이터를 포함할 때, 리스트는 다수의 뷰들에 대한 뷰 인덱스 값들의 리스트를 포함할 수도 있다.

[0158] 일부 예들에서, 비디오 인코더 (20) 는 하나 이상의 표현 파라미터들 (폭, 높이, 비트 심도 등) 을 인코딩 (즉, 시그널링) 할 수도 있고, 비디오 디코더는 이들을 디코딩할 수도 있고, 표현 포맷 파라미터들의 상이한 세트들이 있을 수도 있다. 다음으로, 계층 또는 동작점은 표현 포맷 파라미터들의 이러한 세트의 인덱스를 참조할 수도 있다. 이러한 세트에 대한 선택스 설계의 예는 아래의 표 3 에 도시되어 있다.

표 3

<code>num_rep_formats_minus1</code>	ue(v)
<code>for(i = 0; i <= num_rep_formats_minus1; i++) {</code>	
<code>pic_width_in_luma_samples[i]</code>	ue(v)
<code>pic_height_in_luma_samples[i]</code>	ue(v)
<code>bit_depth_luma_minus8[i]</code>	ue(v)
<code>bit_depth_chroma_minus8[i]</code>	ue(v)
<code>chroma_format_idc[i]</code>	u(2)
<code>}</code>	
<code>for(i = 0; i < cnt_d; i++) {</code>	
<code>rep_format_idx[i]</code>	ue(v)
<code>}</code>	

[0159]

[0160] 일부 예들에서, `ref_format_idx` 는 대신에 계층 파라미터 세트에서 시그널링될 수도 있다.

[0161] 아래의 표 4 는 동작점 설명을 위한 일 예의 선택스를 제공한다.

표 4

operation_points_description() {	디스크립터
num_operation_point_minus1	ue(v)
for(i = 0; i <= num_operation_points_minus1; i++) {	
op_profile_level_idc[i]	u(24)
operation_point_id[i]	ue(v)
priority_id[i]	ue(v)
temporal_id[i]	ue(v)
quality_id[i]	ue(v)
dependency_id[i]	ue(v)
if(cnt_v > 1) {	
num_target_output_views_minus1[i]	ue(v)
for(j = 0; j <= num_target_output_views_minus1[i]; j++)	
view_idx[i][j]	ue(v)
}	
frm_rate_info_present_flag[i]	u(1)
avg_bitrate[i]	u(16)
max_bitrate[i]	u(16)
max_bitrate_calc_window[i]	u(16)
constant_frm_rate_idc[i]	u(2)
if(cnt_v > 1) {	
num_directly_dependent_views[i]	
for(j = 0; j < num_directly_dependent_views[i]; j++) {	
directly_dependent_view_idx[i][j]	
}	
}	
if(cnt_v > 1)	
for(i = 1; i < cnt_v; i++) {	
num_ref_views[i]	ue(v)
for(j = 0; j < num_ref_views[i]; j++)	
ref_view_idx[i][j]	ue(v)
}	
}	

[0162]

[0163] 표 4 의 신택스 엘리먼트들에 대한 시맨틱들의 예들은 아래에서 논의된다:

[0164] 이 예에서, num_operation_point_minus1 플러스 1 은, 코딩된 비디오 시퀀스에 존재하며 그 동작점 정보가 다음의 신택스 엘리먼트들에 의해 시그널링되는 동작점들의 최대 수를 특정한다.

[0165] 이 예에서, op_profile_level_idc[i], operation_point_id[i], priority_id[i], num_target_output_views_minus1[i], frm_rate_info_present_flag[i], avg_bitrate[i], max_bitrate[i], max_bitrate_calc_window[i], constant_frm_rate_idc[i] 및 num_directly_dependent_views[i] 는 H.264 의 뷰 스케일러빌리티 정보 SEI 메시지 내의 동일한 명칭들을 갖는 신택스 엘리먼트들과 동일한 시맨틱들을 가질 수도 있다.

[0166] 이 예에서, quality_id[i] 및 dependency_id[i] 는 H.264 의 스케일러빌리티 정보 SEI 메시지 내의 동일한 명칭들을 갖는 신택스 엘리먼트들과 동일한 시맨틱들을 가질 수도 있다.

[0167] 이 예에서, directly_dependent_view_idx[i][j] 는 현재의 동작점의 타겟 출력 뷰가 현재의 동작점의 표현 내에서 직접 종속되어 있는 j 번째 뷰의 뷰 인덱스를 특정한다.

[0168] 이 예에서, num_ref_views[i] 는 i 와 동일한 뷰 순서 인덱스를 갖는 뷰 컴포넌트들을 디코딩함에 있어서 초기 참조 퍽처 리스트 RefPicList0 및 RefPicList1 에서의 인터-뷰 예측을 위한 뷰 컴포넌트들의 수를 특정한다.
이 예에서, num_ref_views[i] 의 값은 Min(15, num_views_minus1) 보다 더 크지 않을 것이다. 일부 예들에서, num_ref_views[0] 의 값은 0 과 동일하다.

[0169] 이 예에서, ref_view_idx[i][j] 는 i 와 동일한 뷰 순서 인덱스를 갖는 뷰 컴포넌트를 디코딩함에 있어서 초기 참조 퍽처 리스트 RefPicList0 및 RefPicList1 에서의 인터-뷰 예측을 위한 j 번째 뷰 컴포넌트의 뷰 순서

인덱스를 특정한다. 이 예에서, ref_view_idx[i][j] 의 값은 0 내지 31 까지의 범위 내에 있을 것이다.

[0170] 일부 예들에서는, 대안으로서, (예를 들어, H.264 에서 설명된 바와 같은) 스케일러빌러티 정보 SEI 메시지에서의 신택스 엘리먼트들의 일부, 예를 들어, 계층 종속성 정보 관련 신택스 엘리먼트들이 표 4 의 operation_points_description() 신택스 구조에 포함될 수도 있다.

[0171] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 표 4 의 operation_points_description() 신택스 구조에서의 일부 VUI 파라미터들을 코딩 (즉, 시그널링) 할 수도 있다.

[0172] 아래의 표 5 는 비디오 파라미터 세트에 대한 대안적인 신택스를 제공한다:

표 5

video_parameter_set_rbsp()	디스크립터
video_parameter_set_id	ue(v)
num_temporal_layers_minus1	u(3)
for (i = 0; i <= num_temporal_layers_minus1; i++) {	
profile_idc[i]	u(8)
reserved_zero_8bits[i] /* equal to 0 */	u(8)
level_idc[i]	u(8)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
chroma_format_idc	u(2)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pic_cropping_flag	u(1)
if(pic_cropping_flag) {	
pic_crop_left_offset	ue(v)
pic_crop_right_offset	ue(v)
pic_crop_top_offset	ue(v)
pic_crop_bottom_offset	ue(v)
}	
temporal_id_nesting_flag	u(1)
bit_equal_to_one /* equal to 1 */	u(1)
extension_type /* equal to 0 for 3DV */	ue(v)
num_layers_minus2	ue(v)
num_rep_formats_minus1	ue(v)
for(i = 1; i <= num_rep_formats_minus1; i++) {	
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
pic_cropping_flag[i]	u(1)
if(pic_cropping_flag[i]) {	
pic_crop_left_offset[i]	ue(v)
pic_crop_right_offset[i]	ue(v)
pic_crop_top_offset[i]	ue(v)
pic_crop_bottom_offset[i]	ue(v)
}	
}	
for(i = 1; i <= num_layers_minus1; i++) {	
rep_format_idx[i]	ue(v)
if(extension_type == 1) {	
dependency_id[i]	ue(v)
quality_id[i]	ue(v)
num_directly_dependent_layers[i]	ue(v)

[0173]

for(j = 0; j < num_directly_dependency_layers[i]; j++)	
delta_reference_layer_id_minus1[i][j]	ue(v)
}	
}	
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)	
if(extension_type == 0)	
view_dependency()	
num_additional_profiles_levels_minus1	ue(v)
for(i = 0; i <= num_additional_profiles_levels_minus1; i++) {	
additional_profile_idc[i]	u(8)
additional_reserved_zero_8bits[i] /* equal to 0 */	u(8)
additional_level_idc[i]	u(8)
num_applicable_operation_points_minus1[i]	ue(v)
for(j = 0; j <= num_applicable_operation_points[i]; j++) {	
temporal_id[i][j]	ue(v)
layer_id[i][j]	ue(v)
if(extension_type == 0) { /* Always true for 3DV */	
depth_included_flag	u(1)
num_target_output_views_minus1[i][j]	ue(v)
for(k = 0; k < num_target_output_views_minus1[i][j]; k++)	
layer_id[i][j][k]	ue(v)
}	
else (extension_type == 1)	
layer_id[i][j]	ue(v)
}	
}	
}	
vps_vui_parameters_present_flag	u(1)
if(vps_vui_parameters_present_flag)	
vps_vui_parameters()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0174]

표 5 의 비디오 파라미터 세트의 선택스에 대한 시맨틱들의 예들은 아래에서 논의된다. 일반적으로, 아래에서 논의되지 않은 유사하게 명명된 선택스 엘리먼트들은 표 2 에 대하여 위에서 논의된 것과 동일한 시맨틱들을 가질 수도 있다. 다른 선택스 엘리먼트들에 대한 시맨틱들은 다음과 같을 수도 있다:

[0176]

이 예에서, bit_equal_to_one 은 1 과 동일하다 (즉, 2 진수 "1" 값).

[0177]

이 예에서, 0 과 동일한 extention_type 은 다수의 뷔 계층들이 비트스트림에서 존재할 수도 있음을 표시한다. 이 예에서, 1 과 동일한 extension_type 은 다수의 종속성 및/또는 품질 계층들이 비트스트림에서 존재할 수도 있는 것으로 특정한다.

[0178]

이 예에서, num_rep_formats_minus1 플러스 1 은 이 비디오 파라미터 세트에 의해 지원되는 상이한 세트들의 표현 포맷들의 최대 수를 특정하고, 표현 포맷은 비트 심도 및 크로마 포맷 (즉, bit_depth_luma_minus8, bit_depth_chroma_minus8, 및 chroma_format_idc 값들의 세트들), 코딩된 비디오 시퀀스에서의 크롭핑 (cropping) 윈도우 정보 및 꽉쳐 해상도를 포함한다. num_rep_formats_minus1 의 값은 0 내지 X 까지의 범위 내에 있을 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 bit_depth_luma_minus8, bit_depth_chroma_minus8, 및 chroma_format_idc 에 의해 기본 계층 (base layer) 에 대한 비트 심도 및 크로마 포맷의 세트를 코딩할 수도 있고, 비트 심도 및 크로마 포맷의 세트들은 선택스 엘리먼트들의 다음의 세트 bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i], 및 chroma_format_idc[i] 에 의해 강화 계층 (enhancement layer) 들에 대해 시그널링된다.

[0179]

비디오 인코더 (20) 및 비디오 디코더 (30) 는 bit_depth_luma_minus8, bit_depth_chroma_minus8, chroma_format_idc, pic_width_in_luma_samples, pic_height_in_luma_samples, pic_cropping_flag,

pic_crop_left_offset, pic_crop_right_offset, pic_crop_top_offset, 및 pic_crop_bottom_offset 에 의해 표 현 포맷의 제 1 세트를 코딩할 수도 있다.

[0180] 이 예에서, bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i], 및 chroma_format_idc[i] 는 코딩된 비디오 시퀀스에서의 bit_depth_luma_minus8, bit_depth_chroma_minus8, 및 chroma_format_idc 값들의 i 번째 세트를 각각 특정한다.

[0181] 이 예에서, pic_width_in_luma_samples[i] 및 pic_height_in_luma_samples[i] 는 i 번째 표현 포맷을 이용하여 루마 샘플들의 단위들로 각각의 디코딩된 픽처의 폭 및 높이를 각각 특정한다.

[0182] 이 예에서, pic_cropping_flag[i], pic_crop_left_offset[i], pic_crop_right_offset[i], pic_crop_top_offset[i], 및 pic_crop_bottom_offset[i] 는 출력의 픽처 좌표들에서 특정된 직사각형 영역의 측면에서, 표현 포맷의 i 번째 세트에 대하여, 디코딩 프로세스로부터 출력되는 코딩된 비디오 시퀀스에서의 픽처들의 샘플들을 특정한다.

[0183] 이 예에서, rep_format_idx[i] 는 i 와 동일한 layer_id 를 갖는 계층에 적용되는 추가적인 비트 심도 및 크로마 포맷의 세트로 인덱싱되는 값들을 특정한다. i 와 동일한 layer_id 를 갖는 계층에 대한 bit_depth_luma_minus8, bit_depth_chroma_minus8, 및 chroma_format_idc 의 값들은 각각 bit_depth_luma_minus8[rep_format_idx[i]], bit_depth_chroma_minus8[rep_format_idx[i]], 및 chroma_format_idc[rep_format_idx[i]] 와 동일할 수도 있다. rep_format_idx[i] 의 값은 0 내지 X 까지의 범위 내에 있을 것이다.

[0184] 이 예에서, dependency_id[i] 는 i 와 동일한 layer_id 를 갖는 계층에 대한 종속성 식별자를 특정한다. dependency_id[i] 는 0 내지 X 까지의 범위 내에 있을 수도 있다. dependency_id[i] 는 존재하지 않을 때에 0 인 것으로 추론될 수도 있다. num_directly_dependent_layers[i] 가 0 보다 더 클 때, dependency_id[i] 는 i 와 동일한 layer_id 를 갖는 계층이 종속되는 임의의 계층의 종속성 식별자 이상일 수도 있다.

[0185] 이 예에서, quality_id[i] 는 i 와 동일한 layer_id 를 갖는 계층에 대한 품질 식별자를 특정한다. quality_id[i] 는 0 내지 X 까지의 범위 내에 있을 수도 있다. quality_id[i] 는 존재하지 않을 때에, 0 인 것으로 추론될 수도 있다. num_directly_dependent_layers[i] 가 0 보다 더 클 때, quality_id[i] 는, i 와 동일한 layer_id 를 갖는 계층이 종속되어 있으며 dependency_id[i] 와 동일한 종속성 식별자를 가지는 임의의 계층의 품질 식별자 이상일 수도 있다.

[0186] 이 예에서, num_short_term_ref_pic_sets 는 비디오 파라미터 세트에서 특정되는 단기 참조 픽처 세트들의 수를 특정한다. num_short_term_ref_pic_sets 의 값은 0 내지 64 까지의 범위 내에 있을 수도 있다.

[0187] 이 예에서, 1 과 동일한 depth_included_flag 는 현재의 3DV 동작점이 심도를 포함한다는 것을 표시한다. 이 예에서, 0 과 동일한 depth_included_flag 는 현재의 3DV 동작점이 심도를 포함하지 않는다는 것을 표시한다.

[0188] 표 5 의 뷰 종속성 엘리먼트에 대한 일 예의 신택스는 아래의 표 6 에서 제공된다:

표 6

view_dependency() {	
num_views_minus1	ue(v)
for(i = 0; i <= num_views_minus1; i++)	
view_id[i]	ue(v)
for(i = 1; i <= num_views_minus1; i++) {	
num_ref_views[i]	
for(j = 0; j < num_ref_views[i]; j++)	
ref_view_idx[i][j]	ue(v)
inter_view_texture_flag[i][j]	u(l)
}	
}	
}	

[0189]

[0190] 아래의 표 7 은 각각의 비-기본 뷰의 뷰 종속성이 시퀀스 레벨에서 직접 시그널링되는 일 예의 데이터의 세트를 정의한다.

표 7

for(i = 1; i <= num_views_minus1; i++) {	
num_ref_views[i]	ue(v)
for(j = 0; j < num_ref_views[i]; j++)	
ref_view_idx[i][j]	ue(v)
}	

[0191]

[0192] 이 예에서, num_ref_views[i] 는 i 와 동일한 뷰 순서 인덱스를 갖는 뷰 컴포넌트들을 디코딩함에 있어서 초기 참조 픽처 리스트 RefPicList0 및 RefPicList1 에서의 인터-뷰 예측을 위한 뷰 컴포넌트들의 수를 특정한다.

이 예에서, num_ref_views[i] 의 값은 Min(15, num_views_minus1) 보다 더 크지 않다. 이 예에서, num_ref_views[0] 의 값은 0 과 동일하다.

[0193]

이 예에서, ref_view_idx[i][j] 는 i 와 동일한 뷰 순서 인덱스를 갖는 뷰 컴포넌트를 디코딩함에 있어서 초기 참조 픽처 리스트 RefPicList0 및 RefPicList1 에서의 인터-뷰 예측을 위한 j 번째 뷰 컴포넌트의 뷰 순서 인덱스를 특정한다. 이 예에서, ref_view_idx[i][j] 의 값은 0 내지 31 까지의 범위 내에 있다.

[0194]

위에서 언급된 바와 같이, NAL 단위의 특정한 타입 (예를 들어, NAL 단위 타입 10) 은 비디오 파라미터 세트를 캡슐화하기 위해 이용될 수도 있다. NAL 단위 선택지는 아래의 표 8 의 예에 도시된 바와 같이 수정될 수도 있다.

표 8

nal_unit(NumBytesInNALunit) {	디스크립터
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
if(nal_unit_type != 10) { // not VPS NAL unit	
if(cnt_p > 1)	
priority_id	u(v)
if(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
if(cnt_d > 1)	
dependency_id	u(v)
if(cnt_q > 1)	
quality_id	u(v)
reserved_one_bit	u(1)
if(cnt_v > 1)	
view_idx	u(v)
if(cnt_f)	
reserved_flags	u(v)
m = Ceil(log2(cnt_p)) + Ceil(log2(cnt_t)) + Ceil(log2(cnt_d)) + Ceil(log2(cnt_q)) + Ceil(log2(cnt_v)) + cnt_f + 2	
if(((m + 7 >> 3) << 3) - m)	
reserved_bits	u(v)
nalUnitHeaderBytes += ((m + 7) >> 3)	
}	
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte [NumBytesInRBSP++]	b(8)
rbsp_byte [NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte [NumBytesInRBSP++]	b(8)
}	
}	

[0195]

이 예에서, "if(nal_unit_type != 10)" 스테이트먼트 (statement) 내의 엘리먼트들은 기존의 NAL 단위 선택스에 대하여 추가된다. 이 예에서, 선택스 엘리먼트들 priority_id, temporal_id, dependency_id, quality_id 및 view_idx 를 시그널링하기 위하여 이용되는 비트들의 수는 각각 Ceil(log2(cnt_p)), Ceil(log2(cnt_t)), Ceil(log2(cnt_d)), Ceil(log2(cnt_q)) 및 Ceil(log2(cnt_v)) 이다. 마찬가지로, 이 예에서, 선택스 엘리먼트들 priority_id, temporal_id, dependency_id, quality_id 및 view_idx 중 임의의 것이 존재하지 않을 때, 그 선택스 엘리먼트의 값은 0 과 동일한 것으로 추론된다.

[0197]

도출될 수도 있는 추론들 및 비트들의 수에 대하여 위에서 정의된 바를 제외하고, 표 8 의 선택스 엘리먼트들에 대한 시맨틱들은 다음과 같이 정의될 수도 있다. priority_id, dependency_id, 및 quality_id 의 시맨틱들은 ITU-T H.264/AVC 의 SVC 확장부에서 정의된 바와 같을 수도 있다. temporal_id 의 시맨틱들은 HEVC 의 WD4 에서 정의된 바와 같을 수도 있다. 이 예에서, reserved_one_bit 는 1 과 동일하다. reserved_one_bit 에 대한 값 0 은 HEVC 표준의 추후의 확장들에 의해 특정될 수도 있다. 비디오 디코더 (30) 와 같은 디코더들은 reserved_one_bit 의 값을 무시하도록 구성될 수도 있다.

[0198]

이 예에서, view_idx 는 뷰에 대한 뷰 순서 인덱스를 특정한다. view_idx 의 시맨틱들은 ITU-T H.264/AVC 의 MVC 확장부에서 특정된 바와 같은 선택스 엘리먼트 "뷰 순서 인덱스" 와 동일할 수도 있다.

[0199] 이 예에서, reserved_flags 의 각각의 비트는 1 과 동일하다. reserved_flags 의 다른 값들은 당면한 HEVC 표준의 추후의 확장들에 의해 특정될 수도 있다. 비디오 디코더 (30) 와 같은 디코더들은 시맨틱들을 reserved_flags 의 비트들에 배정하는 확장에 따라 동작하도록 구성되지 않으면, reserved_flags 의 값을 무시하도록 구성될 수도 있다. 이 예에서, reserved_flags 를 나타내기 위해 이용된 비트들의 수는 reserved_flags_len 이다.

[0200] 이 예에서, reserved_bits 의 각각의 비트는 1 과 동일하다. reserved_bits 에 대한 다른 값들은 당면한 HEVC 표준의 추후의 표준에 의해 특정될 수도 있다. 비디오 디코더 (30) 와 같은 디코더들은 다시, 이러한 추후의 확장에 따라 구성되지 않으면, reserved_bits 의 값을 무시하도록 구성될 수도 있다. 이 예에서, reserved_bits 를 나타내기 위해 이용된 비트들의 수는 ((m+7 >>3)<<3)-m 이다.

[0201] 아래의 표 9 는 계층 파라미터 세트에 대한 일 예의 선택스를 제공한다. 일부 예들에서, 동일한 선택스는 도 5 의 LPS 들의 각각에 대해 이용될 수도 있다.

표 9

layer_para_set()	디스크립터
depth_flag	u(1)
layer_para_set_id	ue(v)
vps_id	ue(v)
// cu hierarchy {{	
log2_min_coding_block_size_minus3	ue(v)

[0202]

log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
log2_min_pcm_coding_block_size_minus3	ue(v)
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
// cu hierarchy }}	
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
//tiles	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
if (num_tile_columns_minus1 != 0 num_tile_rows_minus1 != 0) {	
tile_boundary_independence_idc	u(1)
uniform_spacing_idc	u(1)
if (uniform_spacing_idc != 1) {	
for (i=0; i<num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for (i=0; i<num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
}	
lps_extension_flag	u(1)
if(lps_extension_flag)	
while(more_rbsp_data())	
lps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0203]

- [0204] 표 9 의 LPS 선택에 대한 시맨틱들의 예들이 아래에서 설명된다. 상이한 계층들 (예를 들어, MVC 내의 상이한 뷰들 또는 SVC 내의 상이한 계층들) 은 상이한 LPS 들을 참조할 수도 있다. 동일한 종속성 계층 내의 상이한 품질 계층들은 동일한 LPS 를 공유할 수도 있다. 동일한 종속성 계층 내의 상이한 시간적 계층들은 동일한 LPS 를 공유할 수도 있다. 대안적으로, 상이한 뷰들은 동일한 LPS 를 참조할 수도 있고, 상이한 종속성 계층들은 동일한 LPS 를 참조할 수도 있다.
- [0205] 이 예에서, 1 과 동일한 depth_flag 는, LPS 가 LPS NAL 단위의 temporal_id, dependency_id, quality_id 및 view_idx 의 값들에 의해 식별된 심도 표현들에 적용되는 것으로 특정한다. 0 과 동일한 Depth_flag 는, LPS 가 LPS NAL 단위의 temporal_id, dependency_id, quality_id 및 view_idx 의 값들에 의해 식별된 텍스처 표현들에 적용되는 것으로 특정한다.
- [0206] 이 예에서, layer_para_set_id 는 현재의 계층 파라미터 세트 (LPS) 의 id 를 특정한다. dependency_id 및 view_idx 의 동일한 값들을 갖는 상이한 계층 파라미터 세트들은 layer_para_set_id 에 대한 하나의 값 공간을 각각 공유하고, 이것은 depencey_id 및 view_idx 의 상이한 조합을 갖는 상이한 LPS 들이 layer_para_set_id 의 동일한 값을 가질 수도 있음을 의미한다.
- [0207] 대안적으로, 모든 LPS 들은 하나의 값 공간을 공유할 수도 있고, 이것은 각각의 LPS 가 layer_para_set_id 의 별개의 값을 가진다는 것을 의미할 수도 있다.
- [0208] 이 예에서, vps_id 는 이 계층 파라미터 세트가 참조하는 비디오 파라미터 세트를 식별한다.
- [0209] 이 예에서, 0 과 동일한 lps_extension_flag 는 lps_extension_data_flag 선택스 엘리먼트들이 계층 파라미터 세트 RBSP 선택 구조에서 존재하지 않는다는 것으로 특정한다. 이 예에서, lps_extension_flag 는 당면한 HEVC 표준을 준수하는 비트스트림들에서 0 과 동일할 수도 있다. lps_extension_flag 에 대한 1 의 값은 ITU-T | ISO/IEC 에 의한 추후의 이용을 위해 예약된다. 비디오 디코더 (30) 와 같은 디코더들은 계층 파라미터 세트 NAL 단위에서 lps_extension_flag 에 대한 값 1 을 따르는 모든 데이터를 무시할 수도 있다.
- [0210] 이 예에서, lps_extension_data_flag 는 임의의 값을 가질 수도 있고, 당면한 HEVC 표준에서 특정된 프로파일들의 준수에 영향을 주지 않는다.
- [0211] 다른 선택스 엘리먼트들은 HEVC WD 의 SPS 에서 동일한 명칭들을 갖는 선택스 엘리먼트들과 동일한 시맨틱들을 가질 수도 있지만, 이 LPS 를 참조하는 픽처들에만 적용될 수도 있다.
- [0212] LPS 는 NAL 단위 내에 포함될 수도 있고, 그 헤더는 상기 표 8 에 따라 정의될 수도 있다. 다음의 선택스 엘리먼트들은 LPS 와 연관될 때에 다음의 약간 수정된 시맨틱들을 가진다.
- [0213] 이 예에서, priority_id 는 이 LPS 를 참조하는 모든 NAL 단위들의 priority_id 값들의 최소값과 동일하다.
- [0214] 이 예에서, temporal_id 는 이 LPS 를 참조하는 모든 NAL 단위들의 temporal_id 의 최소값과 동일하다.
- [0215] 이 예에서, dependency_id 는 이 LPS 를 참조하는 모든 NAL 단위들의 dependency_id 와 동일하다.
- [0216] 이 예에서, quality_id 는 이 LPS 를 참조하는 모든 NAL 단위들의 quality_id 의 최소값과 동일하다.
- [0217] 이 예에서, v_idx 는 현재의 LPS 의 뷰 인덱스이다. 이 LPS 를 참조하는 모든 픽처들은 view_id[v_idx] 의 뷰 id 를 가질 수도 있다.
- [0218] 대안적으로, 상기 선택스 엘리먼트들은 표 10 의 예에서 도시된 바와 같이, 계층 파라미터 세트 선택스 테이블에서 직접 시그널링될 수도 있다. 더욱 상세한 선택스 테이블은 아래의 표 9 에 따라 설계될 수 있다. 이 경우, 그러한 선택스 엘리먼트들은 LPS 의 NAL 단위 헤더에 있지 않고, LPS 의 파싱은 vps_id 와 동일한 ID 를 갖는 VPS 에 종속적일 수도 있다.

표 10

layer_para_set() {	디스크립터
vps_id	u(8)
if(cnt_p > 1)	
priority_id	u(v)
if(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
if(cnt_d > 1)	
dependency_id	u(v)
if(cnt_q > 1)	
quality_id	u(v)
if(cnt_v > 1)	
view_idx	u(v)
depth_flag	u(1)
layer_para_set_id	ue(v)
// cu hierarchy {{	
...	
rbsp_trailing_bits()	
}	

[0219]

[0220] 이 경우의 LPS 는 상기 신택스 엘리먼트들을 복제하는 NAL 단위 헤더를 가질 필요가 없다. LPS 를 캡슐화하는 NAL 단위의 NAL 단위 타입이 예를 들어, 5 라고 가정하면, NAL 단위 헤더 신택스는 표 11 에 도시된 바와 같이 약간 수정될 수도 있고, 이것은 표 8 의 "if" 스테이트먼트에서 예외 "&& nal_unit_type != 5" 를 추가한다:

표 11

nal_unit(NumBytesInNALunit) {	디스크립터
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
if(nal_unit_type != 10 && nal_unit_type != 5) { // not VPS NAL unit	
if(cnt_p > 1)	
priority_id	u(v)
if(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
if(cnt_d > 1)	
dependency_id	u(v)
if(cnt_q > 1)	
quality_id	u(v)
reserved_one_bit	u(1)
if(cnt_v > 1)	
view_idx	u(v)
if(cnt_f)	
reserved_flags	u(v)
m = Ceil(log2(cnt_p)) + Ceil(log2(cnt_t)) + Ceil(log2(cnt_d)) +	
Ceil(log2(cnt_q)) + Ceil(log2(cnt_v)) + cnt_f + 2	
if(((m + 7) >> 3) << 3) - m)	
reserved_bits	u(v)
nalUnitHeaderBytes += ((m + 7) >> 3)	
}	
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte [NumBytesInRBSP++]	b(8)
rbsp_byte [NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte [NumBytesInRBSP++]	b(8)
}	
}	

[0221]

[0222] 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 아래의 표 12의 예에서 도시된 바와 같이, 고정 길이 코딩을 이용하여 스케일러블 특성 관련 신택스 엘리먼트들을 코딩할 수도 있다.

표 12

layer_para_set() {	디스크립터
vps_id	u(8)
priority_id	u(5)
temporal_id	u(3)
dependency_id	u(3)
quality_id	u(3)
view_idx	u(8)
layer_para_set_id	ue(v)
// cu hierarchy {{	

[0223]

[0224]

아래의 표 13 은 본 개시물의 기술들에 따라 꽂쳐 파라미터 세트 (PPS) 에 대한 신택스의 예를 제공한다. 이 예에서, 꽂쳐 파라미터 세트는 기준의 HEVC 의 PPS 와 반대로, "seq_parameter_set_id" 를 시그널링할 필요가 없다.

표 13

pic_parameter_set_rbsp() {	디스크립터
pic_parameter_set_id	ue(v)
...	
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0225]

표 13 의 PPS 에 대한 시맨틱들의 예들은 아래에 설명된다.

[0226]

이 예에서, 0 과 동일한 pps_extension_flag 는, pps_extension_data_flag 신택스 엘리먼트들이 꽂쳐 파라미터 세트 RBSP 신택스 구조에서 존재하지 않는다는 것으로 특정한다. 이 예에서, pps_extension_flag 는 당면한 HEVC 표준을 준수하는 비트스트림들에서 0 과 동일하다. pps_extension_flag 에 대한 1 의 값은 ITU-T | ISO/IEC 에 의한 추후의 이용을 위해 예약될 수도 있다. 비디오 디코더 (30) 와 같은 디코더들은 꽂쳐 파라미터 세트 NAL 단위에서의 pps_extension_flag 에 대한 값 1 을 따르는 모든 데이터를 무시할 수도 있다.

[0227]

이 예에서, pps_extension_data_flag 는 임의의 값을 가질 수도 있다. 그것은 당면한 HEVC 표준에서 특정된 프로파일들의 준수에 반드시 영향을 주지는 않는다. pps_extension_data_flag 의 값들에 대한 시맨틱들은 본 개시물의 기술들과 충돌하지 않으면서, HEVC 표준 또는 그 표준의 확장들의 추가의 개발들에서 배정될 수도 있다.

[0228]

본 개시물의 실시예들에서는, 시퀀스 파라미터 세트 id 또는 계층 파라미터 세트 id 가 PPS 에서 시그널링될 필요가 없다. PPS 에서의 일부 다른 신택스 엘리먼트들은 LPS 로 이동될 수도 있다. 즉, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 표 13 에 도시된 신택스 엘리먼트들과 유사한 데이터를 포함하는 하나 이상의 LPS 들을 코딩하도록 구성될 수도 있다.

[0229]

도 6 은 일 예의 그룹화 파라미터 세트와, 다른 파라미터 세트들을 갖는 GPS 및 슬라이스 헤더들의 관계들을 예시하는 개념적인 도면이다. 이 예에서, 다른 파라미터 세트들은 LPS 들, SPS 들, PPS 들, 타입 0 의 적응 파라미터 세트 (APS) 들 (예를 들어, 적응 루프 필터 (adaptive loop filter; ALF) 파라미터들을 시그널링하는 APS 들), 타입 1 의 APS 들 (예를 들어, 양자화 행렬을 시그널링하는 APS 들) 을 포함한다. 이 예에서, GPS 는 복수의 상이한 그룹들을 포함하고, 그 각각은 고유의 GPS ID (또한 그룹 ID 라고 지칭됨) 를 가지고, 여기서, 각각의 그룹은 파라미터 세트 ID 에 의해 다양한 파라미터 세트들의 각각 중 특정한 하나를 표시한다.

이러한 방식으로, 슬라이스 헤더들은 group_id 를 갖는 그룹에 대응하는 파라미터 세트들의 각각을 특정하기 위하여 group_id 를 특정하기만 할 필요가 있다.

[0230]

아래의 표 14 및 표 15 는 그룹화 파라미터 세트 RBSP 에 대한 신택스의 대안적인 예들을 제공한다.

표 14

group_para_set_rbsp()	디스크립터
number_signaled_para_set_groups_minus1	ue(v)
for(i = 0; i<= number_signaled_para_set_groups_minus1; i++) {	
para_set_group_id[i]	ue(v)
lps_id[i]	ue(v)
pps_id[i]	ue(v)
for (j= 0; j< numParaSetTypes; j++)	
para_set_type_id[i][j]	
}	
gps_extension_flag	u(1)
if(gps_extension_flag)	
while(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0232]

표 15

group_para_set_rbsp()	디스크립터
number_signaled_para_set_groups_minus1	ue(v)
for(i = 0; i<= number_signaled_para_set_groups_minus1; i++) {	
para_set_group_id[i]	ue(v)
lps_id[i]	ue(v)
pps_id[i]	ue(v)
aps_id[i]	ue(v)
}	
gps_extension_flag	u(1)
if(gps_extension_flag)	
while(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0233]

[0234] 비디오 인코더 (20) 및 비디오 디코더 (30) 와 같은 비디오 코더들은 예를 들어, 표 14 또는 표 15 에 따라 그룹화 파라미터 세트를 코딩하도록 구성될 수도 있다. 그룹화 파라미터 세트들의 선택스에 대한 시맨틱들의 예들은 아래에서 제공된다.

[0235] 이 예에서, number_signalled_para_set_groups_minus1 플러스 1 은 시그널링되는 파라미터 그룹들의 수를 특정 한다. 이 값은 0 내지 30 까지의 범위 내에 있을 수도 있다.

[0236] 이 예에서, para_set_group_id[i] 는 i 번째 시그널링된 파라미터 세트 그룹의 ID 를 특정한다. para_set_group_id[i] 의 값은 0 내지 31 까지의 범위 내에 있을 것이다.

[0237] 이 예에서, para_set_type_id[i][j] 는 i 번째 파라미터 세트 그룹에 대한 j 번째 파라미터 세트 타입의 ID 를 특정한다.

[0238] 이 예에서, lps_id[i] 는 para_set_group_id[i] 의 그룹 id 를 갖는 파라미터 세트 그룹에 의해 참조 되는 계층 파라미터 세트의 id 를 표시한다. lps_id[i] 와 동일한 layer_para_set_id 를 갖는 LPS 의 dependency_id 및 view_idx 의 값들은 파라미터 세트 그룹 NAL 단위의 dependency_id 및 view_idx 의 값들과 각각 동일할 수도 있다.

[0239] 파라미터 세트 그룹화 RBSP 의 dependency_id 및 view_idx 의 값들은 표 14 및 표 15 의 예들에서 이 RBSP 의

NAL 단위 헤더에 존재하고, LPS 의 dependency_id 및 view_idx 의 값들은 이 LPS 의 NAL 단위 헤더 또는 LPS 의 신택스 테이블 중 어느 하나에 존재할 수도 있다.

[0240] 대안적으로, lps_id[i] 와 동일한 layer_para_set_id 를 갖는 LPS 의 dependency_id 및 view_idx 의 값들은 파라미터 세트 그룹 NAL 단위의 dependency_id 및 view_idx 의 값들과 각각 동일하지 않을 수도 있다.

[0241] 이 예에서, pps_id[i] 는 para_set_group_id[i] 의 그룹 id 를 갖는 파라미터 세트 그룹에 의해 참조되는 꽉쳐 파라미터 세트의 id 를 표시한다.

[0242] 이 예에서, aps_id[i] 는 para_set_group_id[i] 의 그룹 id 를 갖는 파라미터 세트 그룹에 의해 참조되는 적응 파라미터 세트의 id 를 표시한다.

[0243] 이 예에서, 0 과 동일한 gps_extension_flag 는 gps_extension_data_flag 신택스 엘리먼트들이 파라미터 세트 그룹화 RBSP 신택스 구조에 존재하지 않는다는 것으로 특정한다. gps_extension_flag 는 당면한 HEVC 표준을 준수하는 비트스트림들에서 0 과 동일할 수도 있다. gps_extension_flag 에 대한 1 의 값은 ITU-T | ISO/IEC 에 의한 추후의 이용을 위해 예약될 수도 있다. 비디오 디코더 (30) 와 같은 디코더들은 파라미터 세트 그룹화 NAL 단위에서 gps_extension_flag 에 대한 값 1 을 따르는 모든 데이터를 무시할 수도 있다. 일반적으로, gps_extension_data_flag 는 임의의 값을 가질 수도 있다. 그것은 당면한 HEVC 표준에서 특정된 프로파일들의 준수에 영향을 줄 필요가 없다.

[0244] 일부 예들에서, para_set_type_id[i][j] 는 대신에, 위에서 설명된 바와 같은 aps_id[i] 와 유사한 시맨틱들을 갖는 aps_id[i][j] 일 수도 있다.

[0245] 도 6 에 도시된 바와 같이, 슬라이스 헤더에서 꽉쳐 파라미터 세트 ID 를 참조하는 대신에, 본 개시물의 기술들에 따르면, 슬라이스 헤더는 파라미터 세트 그룹 ID 를 참조할 수도 있고, 이에 따라, LPS, PPS 및 각각의 타입의 APS (예를 들어, ALF 파라미터들 및 양자화 행렬들을 제공하는 APS 들) 를 간접적으로 참조할 수도 있다.

[0246] (코딩된 슬라이스를 포함하는) VCL NAL 단위가 예를 들어, H.264/AVC 설계 원리에 기초하여 간접적으로 파라미터 세트를 참조할 때, 비디오 코더는 비디오 파라미터 세트 또는 계층 파라미터 세트들을 활성화할 수도 있다.

[0247] 일부 예들에서, 파라미터 세트들은 코딩된 슬라이스에 의해서가 아니라, NAL 단위의 특정 타입에 의해 활성화될 수 있다. 예를 들어, 이 특정 타입의 NAL 단위 타입 (파라미터 세트를 활성화 NAL 단위) 는, 비트스트림에서 존재할 경우, 하나, 그리고 정확하게 하나의 VPS 를 활성화할 수도 있다. 추가적으로, 다양한 대안들에서는, NAL 단위의 이러한 타입이 적어도 하나의 LPS 를 활성화할 수도 있다. 추가적으로, NAL 단위의 이러한 타입은 적어도 하나의 PPS 를 활성화할 수도 있다. 추가적으로, NAL 단위의 이러한 타입은 적어도 하나의 APS 를 활성화할 수도 있다. 파라미터 세트들 활성화 NAL 단위는 그룹화 파라미터 세트 RBSP 일 수 있다. 파라미터 세트들 활성화 (parameter sets activation; PSA) NAL 단위는 하나의 코딩된 비디오 시퀀스에 적용가능할 수도 있다. PSA NAL 단위는 비-VCL NAL 단위, 즉, 비디오 코더와 직접적으로 관련되지 않은 것으로 간주될 수도 있다. PSA NAL 단위의 NAL 단위 헤더 신택스는 VPS NAL 단위와 동일할 수도 있다.

[0248] 일부 예들에서, PSA NAL 단위는, 액세스 단위 내에 존재할 경우, 액세스 단위의 최초 VCL NAL 단위를 선행할 수도 있다. 코딩된 비디오 시퀀스의 최초 액세스 단위, 예를 들어, IDR 꽉쳐에는 적어도 하나의 PSA NAL 단위가 있을 수도 있다. 동일한 코딩된 비디오 시퀀스 내의 다수의 PSA NAL 단위들은 동일한 VPS id 를 포함할 수도 있고; 따라서, 상이한 비디오 파라미터 세트들은 동일한 코딩된 비디오 시퀀스 내에서 활성화될 필요가 없다. PSA NAL 단위는, 액세스 단위 내에 존재할 경우, 임의의 LPS, PPS, APS, 또는 SEI NAL 단위가 존재한다면, 이를 선행할 수도 있다. VPS NAL 단위는, 액세스 단위 내에 존재할 경우, 임의의 LPS, PPS, APS, 또는 SEI NAL 단위가 존재한다면, 이를 선행할 수도 있다. 추가적으로, 다양한 대안들에서, PSA NAL 단위는, 액세스 단위 내에 존재할 경우, VPS NAL 단위가 존재한다면, 이를 선행할 수도 있다.

[0249] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 와 같은 비디오 코더들은 예를 들어, HEVC 의 기준의 SPS 신택스와 반대로, 시퀀스 파라미터 세트 (SPS) 에 대한 표 16 의 신택스를 사용하도록 구성될 수도 있다.

표 16

seq_parameter_set_rbsp() {	디스크립터
video_parameter_set_id	ue(v)
rep_format_idx	ue(v)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
}	
qpprime_v_zero_transquant_bypass_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
for(i = 0; i <= max_temporal_layers_minus1; i++) {	
max_dec_pic_buffering[i]	ue(v)
num_reorder_pics[i]	ue(v)
max_latency_increase[i]	ue(v)
}	
restricted_ref_pic_lists_flag	u(1)
if(restricted_ref_pic_lists_flag)	
lists_modification_present_flag	u(1)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
if(pcm_enabled_flag) {	
log2_min_pcm_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_coding_block_size	ue(v)
}	
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
scaling_list_enable_flag	
chroma_pred_from_luma_enabled_flag	u(1)
deblocking_filter_in_aps_enabled_flag	u(1)
seq_loop_filter_across_slices_enabled_flag	u(1)
asymmetric_motion_partitions_enabled_flag	u(1)
non_square_quadtree_enabled_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
if(adaptive_loop_filter_enabled_flag)	
alf_coef_in_slice_flag	u(1)
if(pcm_enabled_flag)	
pcm_loop_filter_disable_flag	u(1)
temporal_id_nesting_flag	u(1)
if(log2_min_coding_block_size_minus3 == 0)	
inter_4x4_enabled_flag	u(1)
long_term_ref_pics_present_flag	u(1)
tiles_or_entropy_coding_sync_idc	u(2)

[0250]

if(tiles_or_entropy_coding_sync_idc == 1) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0251]

표 16 의 일 예의 SPS 는 기존의 SPS 선택스로부터, profile_idc, reserved_zero_8bits, level_idc, chroma_format_idc, separate_colour_plane_flag 및 대응하는 조건문 "if", max_temporal_layers_minus1, pic_width_in_luma_samples, pic_height_in_luma_samples, pic_cropping_flag, pic_crop_left_offset, pic_crop_right_offset, pic_crop_top_offset, 및 pic_crop_bottom_offset 및 대응하는 조건문 "if" 스테이트먼트, bit_depth_luma_minus8, bit_depth_chroma_minus8, num_short_term_ref_pic_sets, 및 short_term_ref_pic_set(i) 및 대응하는 조건문 "if" 스테이트먼트를 제거한다. 또한, 표 16 의 일 예의 SPS 는 video_parameter_set_id 및 rep_format_idx 를 추가한다. 다른 나머지 선택스 엘리먼트들에 대한 시맨틱들은 기존의 HEVC 에서 정의된 것과 동일할 수도 있다. 추가된 엘리먼트들 video_parameter_set_id 및 rep_format_idx 에 대한 시맨틱들은 다음과 같이 정의될 수도 있다:

[0253]

이 예에서, video_parameter_set_id 는 현재의 SPS 에 의해 참조되는 비디오 파라미터 세트 (VPS) 를 식별한다. 대안적으로, video_parameter_set_id 는 시그널링될 필요가 없고, GPS 는 SPS 를 특정 VPS 에 링크시키기 위하여 이용될 수도 있다.

[0254]

이 예에서, rep_format_idx 는 참조된 비디오 파라미터 세트에서 시그널링되는 표현 포맷에 대한 인덱스를 특정 한다.

[0255]

또 다른 대안으로서, 표 17 은 그룹화 파라미터 세트에 대한 선택스의 또 다른 예를 제공한다. 이 예에서는, 위에서 설명된 바와 같이, 비디오 파라미터 세트 ID 선택스 엘리먼트가 SPS 선택스에서 존재하지 않는 것으로 가정된다.

표 17

group_para_set_rbsp() {	디스크립터
gps_id	ue(v)
vps_id	ue(v)
sps_id	ue(v)
pps_id	ue(v)
num_ref_aps_ids	ue(v)
for(i = 0; i < num_ref_aps_ids; i++) {	
ref_aps_id[i]	ue(v)
ref_aps_param_type[i]	ue(v)
}	
gps_extension_flag	u(1)
if(gps_extension_flag)	
while(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0256]

[0257] 표 17 의 선택스 엘리먼트들에 대한 시맨틱들은 다음과 같이 정의될 수도 있다:

[0258] 이 예에서, gps_id 는 그룹 파라미터 세트 (GPS) 의 식별자를 특정한다.

[0259] 이 예에서, vps_id 는 GPS 에 의해 참조되는 비디오 파라미터 세트의 식별자를 특정한다.

[0260] 이 예에서, sps_id 는 GPS 에 의해 참조되는 시퀀스 파라미터 세트의 식별자를 특정한다.

[0261] 이 예에서, pps_id 는 GPS 에 의해 참조되는 픽처 시퀀스 파라미터 세트의 식별자를 특정한다.

[0262] 이 예에서, num_ref_aps_ids 는 다음의 ref_aps_id[i] 선택스 엘리먼트들의 수를 특정한다. num_ref_aps_ids 의 값은 0 내지 4 까지의 범위 내에 있을 것이다.

[0263] 이 예에서, ref_aps_id[i] 는 그룹 파라미터 세트에 의해 참조되는 i 번째 적응 파라미터 세트를 식별한다.

[0264] ref_aps_id[i] 의 동일한 값은 한번을 초과하여 루프에서 존재할 수도 있고, 이에 따라, 동일한 APS 로부터의 하나를 초과하는 타입의 APS 파라미터들은 동일한 GPS 에 의해 참조될 수 있고, GPS 를 참조하는 코딩된 슬라이스들에 적용될 수도 있다.

[0265] 이 예에서, ref_aps_param_type[i] 는 그룹 파라미터 세트에 의해 참조되는 i 번째 적응 파라미터 세트에 포함된 APS 파라미터들의 타입을 특정한다. ref_aps_param_type[i] 의 값은 0 내지 3 까지의 범위 내에 있을 수도 있다. ref_aps_param_type[i] 에 대한 0 내지 3 까지의 값들은 스케일링 리스트, 디블록 킹 필터, 샘플 적응적 오프셋 (sample adaptive offset; SAO), 및 ALF 의 APS 파라미터 타입들에 각각 대응한다. i 의 임의의 2 개의 상이한 값들에 대한 ref_aps_param_type[i] 의 값들은 일부 예들에서 동일하지 않을 것이다.

[0266] 이 예에서, 0 과 동일한 gps_extension_flag 는 gps_extension_data_flag 선택스 엘리먼트들이 파라미터 세트 그룹화 RBSP 선택스 구조에 존재하지 않는 것으로 특정한다. gps_extension_flag 는 당면한 HEVC 표준을 준수하는 비트스트림들에서 0 과 동일할 수도 있다. gps_extension_flag 에 대한 1 의 값은 ITU-T | ISO/IEC 에 의한 추후의 이용을 위하여 예약될 수도 있다. 비디오 디코더 (30) 와 같은 디코더들은 파라미터 세트 그룹화 NAL 단위에서 gps_extension_flag 에 대한 값 1 을 따르는 모든 데이터를 무시할 수도 있다.

[0267] 이 예에서, gps_extension_data_flag 는 임의의 값을 가질 수도 있다. 그것은 당면한 HEVC 표준에서 특정된 프로파일들의 준수에 영향을 줄 필요가 없다.

[0268] 비디오 인코더 (20) 및 비디오 디코더 (30) 와 같은 비디오 코더들은, GPS 가 표 17 에 따라 특정되거나 표 17 의 예를 실질적으로 준수할 때, 단일-계층 또는 단일-뷰 비트스트림들에 대한 파라미터 세트들을 활성화하기 위

하여 다음의 프로세스를 적용할 수도 있다.

[0269] 적응 파라미터 세트 RBSP 는 코딩된 슬라이스 NAL 단위들에 의해 참조되는 하나 이상의 그룹 파라미터 세트들을 통해 간접적으로 하나 이상의 코딩된 픽처들의 코딩된 슬라이스 NAL 단위들에 의해 참조될 수 있는 파라미터들을 포함할 수도 있다. 각각의 적응 파라미터 세트 RBSP 는 초기에 디코딩 프로세스의 동작의 시작 시에 활성이 아닌 것으로 간주될 수도 있다. 최대 하나의 적응 파라미터 세트 RBSP 는 디코딩 프로세스의 동작 동안의 임의의 소정의 순간에 APS 파라미터들의 각각의 타입에 대하여 활성인 것으로 간주될 수도 있고, 특정한 타입의 APS 파라미터들에 대한 임의의 특정한 적응 파라미터 세트 RBSP 의 활성화는 그 특정한 타입의 APS 파라미터들에 대한 (만약 있다면) 이전에 활성인 적응 파라미터 세트 RBSP 의 비활성화로 귀착된다.

[0270] (*aps_id* 의 특정한 값을 갖는) 적응 파라미터 세트 RBSP 가 특정한 타입의 APS 파라미터들에 대해 활성이 아니고 그것은 코딩된 슬라이스 NAL 단위에 의해 참조되는 그룹 파라미터 세트를 통해 간접적으로 (*aps_id* 의 그 값을 이용하여) 그 특정한 타입의 APS 파라미터들에 대한 코딩된 슬라이스 NAL 단위에 의해 참조될 때, 그것은 그 특정한 타입의 APS 파라미터들에 대해 활성화될 수도 있다. 이 적응 파라미터 세트 RBSP 는 그 특정한 타입의 APS 파라미터들에 대한 또 다른 적응 파라미터 세트 RBSP 의 활성화에 의해 비활성화될 때까지 그 특정한 타입의 APS 파라미터들에 대한 활성 적응 파라미터 세트 RBSP 라고 칭해진다. *aps_id* 의 그 특정한 값을 갖는 적응 파라미터 세트 RBSP 는 그 활성화 이전에 디코딩 프로세스에 의해 이용가능할 수도 있다.

[0271] 픽처 파라미터 세트 RBSP 는 코딩된 슬라이스 NAL 단위들에 의해 참조되는 하나 이상의 그룹 파라미터 세트들을 통해 간접적으로 하나 이상의 코딩된 픽처들의 코딩된 슬라이스 NAL 단위들에 의해 참조될 수 있는 파라미터들을 포함할 수도 있다. 각각의 픽처 파라미터 세트 RBSP 는 초기에 디코딩 프로세스의 동작의 시작 시에 활성이 아닌 것으로 간주될 수도 있다. 최대 하나의 픽처 파라미터 세트 RBSP 는 디코딩 프로세스의 동작 동안의 임의의 소정의 순간에 활성인 것으로 간주될 수도 있고, 임의의 특정한 픽처 파라미터 세트 RBSP 의 활성화는 (만약 있다면) 이전에 활성인 픽처 파라미터 세트 RBSP 의 비활성화로 귀착된다.

[0272] (*pic_parameter_set_id* 의 특정한 값을 갖는) 픽처 파라미터 세트 RBSP 가 활성이 아니고, 그것은 코딩된 슬라이스 NAL 단위에 의해 참조되는 그룹 파라미터 세트를 통해 간접적으로 (*pic_parameter_set_id* 의 그 값을 이용하여) 코딩된 슬라이스 NAL 단위에 의해 참조될 때, 그것은 활성화될 수도 있다. 이 픽처 파라미터 세트 RBSP 는 또 다른 픽처 파라미터 세트 RBSP 의 활성화에 의해 비활성화될 때까지 활성 픽처 파라미터 세트 RBSP 라고 칭해진다. *pic_parameter_set_id* 의 그 특정한 값을 갖는 픽처 파라미터 세트 RBSP 는 그 활성화 이전에 디코딩된 프로세스에 의해 이용가능할 수도 있다.

[0273] 코딩된 픽처에 대한 활성 픽처 파라미터 세트 RBSP 에 대한 *pic_parameter_set_id* 의 값을 포함하는 임의의 픽처 파라미터 세트 NAL 단위는, 그것이 코딩된 픽처의 최종 VCL NAL 단위를 후행하고 또 다른 코딩된 픽처의 최초 VCL NAL 단위를 선행하지 않으면, 코딩된 픽처에 대한 활성 픽처 파라미터 세트 RBSP 의 것과 동일한 컨텐츠를 가질 수도 있다.

[0274] 시퀀스 파라미터 세트 RBSP 는, 코딩된 슬라이스 NAL 단위들에 의해 참조되는 하나 이상의 그룹 파라미터 세트들을 통해 간접적으로 하나 이상의 코딩된 픽처들의 코딩된 슬라이스 NAL 단위들에 의해 참조될 수 있거나, 베퍼링 주기 SEI 메시지를 포함하는 하나 이상의 SEI NAL 단위들에 의해 참조될 수 있는 파라미터들을 포함할 수도 있다. 각각의 시퀀스 파라미터 세트 RBSP 는 초기에 디코딩 프로세스의 동작의 시작 시에 활성이 아닌 것으로 간주될 수도 있다. 최대 하나의 시퀀스 파라미터 세트 RBSP 는 디코딩 프로세스의 동작 동안의 임의의 소정의 순간에 활성인 것으로 간주될 수도 있고, 임의의 특정한 시퀀스 파라미터 세트 RBSP 의 활성화는 (만약 있다면) 이전에 활성인 시퀀스 파라미터 세트 RBSP 의 비활성화로 귀착된다.

[0275] (*seq_parameter_set_id* 의 특정한 값을 갖는) 시퀀스 파라미터 세트 RBSP 가 이미 활성이 아니며, (*seq_parameter_set_id* 의 그 값을 이용하여) 코딩된 슬라이스 NAL 단위에 의해 참조되는 그룹 파라미터 세트를 통해 간접적으로, 코딩된 슬라이스 NAL 단위에 의해 참조되거나, (*seq_parameter_set_id* 의 그 값을 이용하여) 베퍼링 주기 SEI 메시지를 포함하는 SEI NAL 단위에 의해 참조될 때, 시퀀스 파라미터 세트 RBSP 는 활성화될 수도 있다. 이 시퀀스 파라미터 세트 RBSP 는 또 다른 시퀀스 파라미터 세트 RBSP 의 활성화에 의해 비활성화될 때까지 활성 시퀀스 파라미터 세트 RBSP 라고 칭해진다. *seq_parameter_set_id* 의 그 특정한 값을 가지며 0 과 동일한 *temporal_id* 를 갖는 액세스 단위 내에 포함된 시퀀스 파라미터 세트 RBSP 는 그 활성화 이전에 디코딩 프로세스에 의해 이용가능할 수도 있다. 활성화된 시퀀스 파라미터 세트 RBSP 는 전체적인 코딩된 비디오 시퀀스에 대해 활성인 상태로 있을 것이다.

- [0276] 비디오 파라미터 세트 RBSP 는, 코딩된 슬라이스 NAL 단위들에 의해 참조되는 하나 이상의 그룹 파라미터 세트들을 통해 간접적으로 하나 이상의 코딩된 픽처들의 코딩된 슬라이스 NAL 단위들에 의해 참조될 수 있거나, 베페링 주기 SEI 메시지를 포함하는 하나 이상의 SEI NAL 단위들에 의해 참조될 수 있는 파라미터들을 포함할 수도 있다. 각각의 비디오 파라미터 세트 RBSP 는 초기에 디코딩 프로세스의 동작의 시작 시에 활성이 아닌 것으로 간주될 수도 있다. 최대 하나의 비디오 파라미터 세트 RBSP 는 디코딩 프로세스의 동작 동안의 임의의 소정의 순간에 활성인 것으로 간주될 수도 있고, 임의의 특정한 비디오 파라미터 세트 RBSP 의 활성화는 (만약 있다면) 이전에 활성인 비디오 파라미터 세트 RBSP 의 비활성화로 귀착된다.
- [0277] (`video_parameter_set_id` 의 특정한 값을 갖는) 비디오 파라미터 세트 RBSP 가 이미 활성이 아니며, (`video_parameter_set_id` 의 그 값을 이용하여) 코딩된 슬라이스 NAL 단위에 의해 참조되는 그룹 파라미터 세트를 통해 간접적으로, 코딩된 슬라이스 NAL 단위에 의해 참조되거나, (`video_parameter_set_id` 의 그 값을 이용하여) 베페링 주기 SEI 메시지를 포함하는 SEI NAL 단위에 의해 참조될 때, 비디오 파라미터 세트 RBSP 는 활성화될 수도 있다. 이 비디오 파라미터 세트 RBSP 는 또 다른 비디오 파라미터 세트 RBSP 의 활성화에 의해 비활성화될 때까지 활성 비디오 파라미터 세트 RBSP 라고 칭해진다. `video_parameter_set_id` 의 그 특정한 값을 가지며 0 과 동일한 `temporal_id` 를 갖는 액세스 단위 내에 포함된 비디오 파라미터 세트 RBSP 는 그 활성화 이전에 디코딩 프로세스에 의해 이용가능할 것이다. 활성화된 비디오 파라미터 세트 RBSP 는 전체적인 코딩된 비디오 시퀀스에 대해 활성인 상태로 있을 것이다.
- [0278] 코딩된 비디오 시퀀스에 대한 활성 시퀀스 파라미터 세트 RBSP 에 대한 `seq_parameter_set_id` 의 값을 포함하는 임의의 시퀀스 파라미터 세트 NAL 단위는, 코딩된 비디오 시퀀스의 최종 액세스 단위를 후행하고 또 다른 코딩된 비디오 시퀀스의 (존재할 경우) 베페링 주기 SEI 메시지를 포함하는 최초 SEI NAL 단위 및 최초 VCL NAL 단위를 선행하지 않으면, 코딩된 비디오 시퀀스에 대한 활성 시퀀스 파라미터 세트 RBSP의 컨텐츠와 동일한 컨텐츠를 가질 수도 있다.
- [0279] 코딩된 비디오 시퀀스에 대한 활성 비디오 파라미터 세트 RBSP 에 대한 `video_parameter_set_id` 의 값을 포함하는 임의의 비디오 파라미터 세트 NAL 단위는, 코딩된 비디오 시퀀스의 최종 액세스 단위를 후행하고 또 다른 코딩된 비디오 시퀀스의 (존재할 경우) 베페링 주기 SEI 메시지를 포함하는 최초 SEI NAL 단위 및 최초 VCL NAL 단위를 선행할 수도 있지 않으면, 코딩된 비디오 시퀀스에 대한 활성 비디오 파라미터 세트 RBSP의 컨텐츠와 동일한 컨텐츠를 가질 수도 있다.
- [0280] 비디오 파라미터 세트들, 시퀀스 파라미터 세트들, 픽처 파라미터 세트들 및 적응 파라미터 세트들에서의 신팩스 엘리먼트들의 값들 (및 그러한 신팩스 엘리먼트들로부터 유도된 변수들의 값들) 과 다른 신팩스 엘리먼트들 사이의 관계에 대해 표현되는 모든 제약들은 각각의 특정한 타입의 APS 파라미터들에 대한 활성 비디오 파라미터 세트들, 활성 시퀀스 파라미터 세트, 활성 픽처 파라미터 세트 및 활성 적응 파라미터 세트에만 적용될 수도 있는 제약들의 표현들이다. 비트스트림에서 활성화되지 않는 임의의 비디오 파라미터 세트 RBSP 가 존재할 경우, 그 신팩스 엘리먼트들은 그것이 다른 준수하는 비트스트림에서 참조를 위해 활성화되었을 경우에 특정된 제약들을 준수할 값을 가질 수도 있다. 비트스트림에서 활성화되지 않는 임의의 시퀀스 파라미터 세트 RBSP 가 존재할 경우, 그 신팩스 엘리먼트들은 그것이 다른 준수하는 비트스트림에서 참조를 위해 활성화되었을 경우에 특정된 제약들을 준수할 값을 가질 수도 있다. 비트스트림에서 결코 활성화되지 않은 임의의 픽처 파라미터 세트 RBSP 가 존재할 경우, 그 신팩스 엘리먼트들은 그것이 다른 준수하는 비트스트림에서 참조를 위해 활성화되었을 경우에 특정된 제약들을 준수할 값을 가질 수도 있다. 비트스트림에서 결코 활성화되지 않은 임의의 적응 파라미터 세트 RBSP 가 존재할 경우, 그 신팩스 엘리먼트들은 그것이 다른 준수하는 비트스트림에서 참조를 위해 활성화되었을 경우에 특정된 제약들을 준수할 값을 가질 수도 있다.
- [0281] 디코딩 프로세스의 동작 동안에는, 각각의 타입의 APS 파라미터들에 대한 활성 비디오 파라미터 세트, 활성 시퀀스 파라미터 세트, 활성 픽처 파라미터 세트 및 활성 적응 파라미터 세트의 파라미터들의 값들이 효과가 있는 것으로 간주될 수도 있다. SEI 메시지들의 해독을 위하여, 동일한 액세스 단위에서의 코딩된 픽처의 VCL NAL 단위들에 대한 디코딩 프로세스의 동작에 대해 활성인 비디오 파라미터 세트, 시퀀스 파라미터 세트, 픽처 파라미터 세트 및 적응 파라미터 세트의 파라미터들의 값들은, SEI 메시지 시맨틱들에서 이와 다르게 특정되지 않을 경우에 효과가 있는 것으로 간주될 수도 있다.
- [0282] 도 7 은 본 개시물의 기술들에 따라 비디오 데이터를 인코딩하기 위한 일 예의 방법을 예시하는 플로우차트이다. 비디오 인코더 (20) 에 대하여 설명되었지만, 다른 비디오 인코딩 디바이스들이 도 7 의 방법을 수행하도록 구성될 수도 있다는 것을 이해해야 한다.

[0283] 먼저, 이 예에서는, 비디오 인코더 (20) 가 원시 비디오 데이터의 하나 이상의 계층들을 포함하는 비트스트림을 수신한다 (100). 예를 들어, 비디오 소스 (18; 도 1) 는 멀티-뷰 비디오 테이터를 비디오 인코더 (20) 에 제공할 수도 있다. 대안적으로, 비디오 인코더 (20) 또는 그 프리프로세서는 원시 비디오 비트스트림을 복수의 다양한 계층들, 예를 들어, 공간 분해도 계층들, 품질 계층들, 시간적 계층들, 등으로 분할할 수도 있다.

또 다른 예들에서, 비트스트림은 다양한 계층들의 조합, 예를 들어, 뷰들, 공간 해상도 계층들, 품질 계층들, 시간적 계층들, 등의 임의의 조합으로 파티셔닝될 수도 있다.

[0284] 비디오 인코더 (20) 는 계층들의 세트 사이에서 대응하는 시퀀스들에 대한 하나 이상의 공통 파라미터들을 결정할 수도 있다 (102). 대응하는 시퀀스들은 상이한 계층들에서 대응하는 시간적 위치들을 가지는 시퀀스들일 수도 있다. 즉, T1 의 (디스플레이 시간의 측면에서의) 시작 시간 및 T2 의 (다시, 디스플레이 시간의 측면에서의) 종료 시간을 가지는 제 1 시퀀스와, T1 의 시작 시간 및 T2 의 종료 시간을 또한 가지는 제 2 시퀀스는 서로 대응한다고 말할 수도 있다. 특히, 제 1 시퀀스는 제 1 계층의 일부를 형성할 수도 있고, 제 2 시퀀스는 제 2 의 상이한 계층의 일부를 형성할 수도 있다. "시퀀스" 는 예를 들어, 순간 디코딩 리프레쉬 (instantaneous decoding refresh; IDR) 꾹처에서 시작하고 디코딩 순서에서 후속 IDR 꾹처 직전에 종료되는, 디코딩 순서에서 연속 꾹처들의 시리즈를 포함할 수도 있다. 일반적으로, 파라미터들은 하나 이상의 계층들, 예를 들어, N 개의 계층들의 대응하는 시퀀스들의 세트에 대응할 수도 있고, 여기서, N 은 정수이다.

다음으로, 비디오 인코더 (20) 는 결정된 파라미터들에 대한 데이터를 포함하는 VPS 를 인코딩할 수도 있다 (104). 예를 들어, 비디오 인코더 (20) 는 표 2 또는 표 5 의 예들 중 하나에 대응하는 VPS 를 코딩할 수도 있다.

[0285] 비디오 인코더 (20) 는 하나의 계층 내의 시퀀스에 대한 공통 파라미터들을 또한 결정할 수도 있다 (106). 시퀀스는 VPS 가 코딩되었던 다른 계층들에서의 다른 시퀀스들에 대응하는 시퀀스들 중 하나를 포함할 수도 있다. 비디오 인코더 (20) 는 시퀀스에 대한 공통 파라미터들을 포함하는 시퀀스 파라미터 세트 (SPS) 를 코딩할 수도 있다 (108). 따라서, VPS 및 SPS 는 별도의 데이터 구조들이며 상이한 타입들의 비디오 데이터에 대응한다는 것을 이해해야 한다. VPS 는 복수의 계층들 사이에서 대응하는 시퀀스들의 세트에 대응할 수도 있는 반면, SPS 는 하나의 계층에서 하나의 시퀀스에 대응한다. SPS 는 H.264/AVC 의 SPS, (위에서 표 1에 예시된) MVC 에 의해 확장된 바와 같은 H.264/AVC 의 SPS, 당면한 HEVC 표준, 또는 위에서 설명된 표 16 의 예를 실질적으로 준수할 수도 있다. 추가적으로, 비디오 인코더 (20) 는 시퀀스에서의 꾹처에 대한 꾹처 파라미터 세트 (PPS) 를 코딩할 수도 있다 (110). PPS 는 H.264/AVC 의 SPS, 당면한 HEVC 표준, 또는 위에서 설명된 표 13 의 예를 실질적으로 준수할 수도 있다. 도 7 의 방법은 오직 하나의 PPS 의 코딩을 도시하지만, 다수의 PPS 들이 코딩될 수도 있다는 것을 이해해야 한다. 하나 이상의 꾹처들은 동일한 PPS 를 참조할 수도 있다.

[0286] 다음으로, 비디오 인코더 (20) 는 SPS 및 PPS 들이 코딩되었던 최근의 계층이 최종 계층인지 여부를 결정할 수도 있다 (112). 최종 계층이 아직 다루어지지 않았을 경우 (112 의 "아니오" 가지), 비디오 인코더 (20) 는 다음 계층을 선택할 수도 있고, 예를 들어, 단계들 (106 - 110) 에 따라 다음 계층에 대한 SPS 및 하나 이상의 PPS 들을 코딩할 수도 있다. 최종 계층이 다루어졌을 경우 (112 의 "예" 가지), 비디오 인코더 (20) 는 VPS, SPS, 및 PPS 의 데이터에 기초하여 다양한 계층들의 비디오 데이터를 인코딩할 수도 있다. VPS 에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 다양한 예들은 도 9 내지 도 12 에 대하여 아래에서 더욱 상세하게 설명된다.

[0287] 도 7 의 예에서는 도시되지 않았지만, 일부 예들에서, 비디오 인코더 (20) 는 위에서 설명된 바와 같이, 하나 이상의 LPS 들 및/또는 하나 이상의 GPS 들을 추가적으로 인코딩할 수도 있다. LPS 들은 표 9, 표 10, 또는 표 12 의 예들을 실질적으로 준수할 수도 있는 반면, GPS 는 표 14, 표 15, 또는 표 17 의 예들을 실질적으로 준수할 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 는 또한, LPS 들 및/또는 GPS 들에 적어도 부분적으로 기초하여 비디오 데이터를 코딩한다.

[0288] 이러한 방식으로, 도 7 의 방법은 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하는 것으로서, 여기서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하는, 상기 비디오 파라미터 세트 (VPS) 를 코딩하는 것과, VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩하는 것을 포함하는 방법의 예를 나타낸다.

[0289] 도 8 은 본 개시물의 기술들에 따라 비디오 데이터를 디코딩하기 위한 일 예의 방법을 예시하는 플로우차트이다. 비디오 디코더 (30) 에 대하여 설명되었지만, 다른 비디오 디코딩 디바이스들은 도 8 의

방법을 수행하도록 구성될 수도 있다는 것을 이해해야 한다.

[0290] 먼저, 비디오 디코더 (30) 는 코딩된 비디오 데이터의 계층들에 대한 VPS, 하나 이상의 SPS 들, 및 하나 이상의 PPS 들을 포함하는 비트스트림을 수신한다 (120). 다음으로, 비디오 디코더 (30) 는 하나 이상의 계층들 사이에서 대응하는 시퀀스들에 대한 공통 파라미터들을 포함하는 VPS 를 디코딩할 수도 있다 (122). 마찬가지로, 비디오 디코더 (30) 는 하나의 계층의 시퀀스에 대한 공통 파라미터들을 포함하는 시퀀스 파라미터 세트를 디코딩할 수도 있다 (124). 또한, 비디오 디코더 (30) 는 시퀀스의 픽처에 대한 파라미터들을 포함하는 픽처 파라미터 세트를 디코딩할 수도 있다 (126). 위에서 논의된 바와 같이, 하나 이상의 픽처들은 동일한 PPS 를 참조할 수도 있고, 그러므로, PPS 의 파라미터들은 하나 이상의 픽처들에 대해 공통인 것으로 간주될 수도 있다. 마찬가지로, 도 8 에 도시되지 않았지만, 비디오 디코더 (30) 는 시퀀스에 대한 복수의 PPS 들을 디코딩할 수도 있다.

[0291] 또한, 비디오 디코더 (30) 는 가장 최근의 계층이 다루어져야 할 최종 계층이었는지 여부를 결정할 수도 있다 (128). 가장 최근의 계층이 최종 계층이 아니었을 경우 (128 의 "아니오" 가지), 비디오 디코더 (30) 는 단계들 (124 및 126) 에 따라 후속 계층에 대한 SPS 및 하나 이상의 PPS 들을 디코딩하도록 진행할 수도 있다. 다른 한편으로, 가장 최근의 계층이 최종 계층이었을 경우 (128 의 "예" 가지), 비디오 디코더 (30) 는 VPS, SPS 들, 및 PPS 들에 기초하여 계층들의 비디오 데이터를 디코딩하도록 진행할 수도 있다 (130). VPS 에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 예들은 도 9 내지 도 12 에 대하여 더욱 상세하게 논의된다.

[0292] 도 8 의 예에서 도시되지 않았지만, 일부 예들에서는, 비디오 디코더 (30) 가 위에서 설명된 바와 같이, 하나 이상의 LPS 를 및/또는 하나 이상의 GPS 를 추가적으로 디코딩할 수도 있다. LPS 들은 표 9, 표 10, 또는 표 12 의 예들을 실질적으로 준수할 수도 있는 반면, GPS 는 표 14, 표 15, 또는 표 17 의 예들을 실질적으로 준수할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 는 또한, LPS 를 및/또는 GPS 를 적어도 부분적으로 기초하여 비디오 데이터를 디코딩한다.

[0293] 이러한 방식으로, 도 8 의 방법은 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 세트 (VPS) 를 코딩하는 것으로서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하는, 상기 비디오 파라미터 세트 (VPS) 를 코딩하는 것과, VPS 에 적어도 부분적으로 기초하여 비디오 데이터의 하나 이상의 계층들을 코딩하는 것을 포함하는 방법의 예를 나타낸다.

[0294] 도 9 는 VPS 에서 시그널링되는 바와 같은 시간적 계층들의 수에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다. 도 9 의 방법은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있다. 예시의 목적들을 위하여, 도 9 의 방법은 비디오 디코더 (30) 에 대하여 설명된다.

[0295] 이 예에서, 비디오 디코더 (30) 는 예를 들어, VPS 가 대응하는 하나 이상의 계층들의 비디오 데이터에서의 시간적 계층들의 수를 표시하는 VPS 를 코딩 (즉, 디코딩) 한다 (150). 예를 들어, 비디오 디코더 (30) 는 상기 표 2 에 대하여 설명된 바와 같이 "cnt_t" 를 디코딩할 수도 있다. 또 다른 예로서, 비디오 디코더 (30) 는 상기 표 5 에 대하여 설명된 바와 같이, num_temporal_layers_minus1 을 디코딩할 수도 있다.

[0296] 이 표시에 기초하여, 이 예에서는, 비디오 디코더 (30) 가 시간적 계층들의 각각에 대한 시간적 식별자들을 디코딩한다 (152). 마찬가지로, 비디오 디코더 (30) 는 시간적 계층들의 수에 기초하여 참조 픽처 식별자 값들을 결정할 수도 있다 (154). 예를 들어, 비디오 디코더 (30) 는, 계층 N 에서의 현재의 픽처에 대하여, 현재의 픽처가 참조를 위하여 계층 N+1 에서의 또는 계층 N+1 위에서의 픽처들을 이용하지 않을 것이라고 결정하도록 구성될 수도 있다. 그러므로, 비디오 디코더 (30) 는 계층 N 에서의 또는 계층 N 아래의 계층들에서의 잠재적인 참조 픽처들에 대한 식별자들을 결정할 수도 있다. 또한, 비디오 디코더 (30) 는 계층 N 까지의 (그리고 이를 포함하는) 계층들의 참조 데이터를 이용하여 시간적 계층 N 에서의 픽처들의 데이터를 디코딩할 수도 있다 (156). 따라서, 도 9 는 비디오 데이터의 하나 이상의 계층들에서의 시간적 계층들의 최대 수를 표시하는 VPS 의 데이터를 코딩하는 것과, VPS 에 적어도 부분적으로 기초하여 하나 이상의 계층들을 코딩하는 것을 포함하는 방법의 예를 나타낸다.

[0297] 도 10 은 하나 이상의 계층들에서 재순서화될 픽처들 및 디코딩된 픽처 버퍼에 저장될 픽처들의 수에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다. 도 10 의 방법은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있다. 예시의 목적들을 위하여, 도

10 의 방법은 비디오 디코더 (30) 에 대하여 설명된다.

[0298] 이 예에서, 비디오 디코더 (30) 는 비디오 데이터의 하나 이상의 계층들에서 재순서화될 픽처들의 수와, 소정의 시간에 디코딩된 픽처 버퍼 (예를 들어, 참조 픽처 메모리 (82) 에 저장될 픽처들의 수를 표시하는 VPS 를 디코딩한다 (160). 예를 들어, 비디오 디코더 (30) 는 상기 표 16 에 대하여 설명된 바와 같은 num_reordered_pics 및/또는 DPB 사이즈를 특정하는 비트스트림 한정 정보에 실질적으로 대응하는 VPS 의 신택스 엘리먼트를 디코딩할 수도 있다. 다른 예들에서, VPS 는 재순서화될 픽처들의 수 및 디코딩된 픽처 버퍼에 저장될 픽처들의 수 중 반드시 둘 모두가 아니라, 오직 하나 또는 다른 하나를 포함할 수 있다. 다음으로, 비디오 디코더 (30) 는 재순서화될 및/또는 저장될 픽처들의 수에 기초하여 디코딩된 픽처 버퍼 (예를 들어, 참조 픽처 메모리 (82)) 를 관리할 수도 있다 (162). 예를 들어, 비디오 디코더 (30) 는, 저장될 픽처들의 수 보다 더 많은 픽처들이 참조 픽처 메모리 (82) 에 저장될 때에 참조 픽처 메모리 (82) 로부터 픽처들을 제거할 수도 있다.

[0299] 비디오 디코더 (30) 는 DPB 에서의 (즉, 참조 픽처 메모리 (82) 에서의) 픽처들의 수에 기초하여 참조 픽처 식별자 값들을 또한 결정할 수도 있다 (164). 또한, 비디오 디코더 (30) 는 참조 픽처 식별자 값들에 기초하여 픽처들의 데이터를 디코딩할 수도 있다 (166). 따라서, 도 10 의 방법은 하나 이상의 계층들의 디코딩 동안에 디코딩된 픽처 버퍼 (DPB) 에 저장될 픽처들의 수를 표시하는 VPS 의 데이터를 코딩하는 것을 포함하는 방법과, 하나 이상의 계층들 중 적어도 하나의 계층에서 재순서화될 프레임들의 수를 표시하는 VPS 의 데이터를 코딩하는 것을 포함하는 방법의 예를 나타낸다.

[0300] 도 11 은 VPS 에서 시그널링되는 가상 참조 디코더 (HRD) 파라미터들에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다. 도 11 의 방법은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있다. 예시의 목적들을 위하여, 도 11 의 방법은 비디오 디코더 (30) 에 대하여 설명된다.

[0301] 이 예에서, 비디오 디코더 (30) 는 HRD 파라미터들을 표시하는 VPS 를 디코딩한다 (170). 비디오 디코더 (30) 는 HRD 파라미터들에 기초하여 코딩된 픽처 버퍼 (CPB) 로부터의 픽처들에 대한 제거 시간들을 더 결정할 수도 있다 (172). 다음으로, 비디오 디코더 (30) 는 결정된 제거 시간들에 기초하여 CPB 로부터 데이터를 제거할 수도 있고 (174), CPB 로부터 제거된 데이터를 디코딩할 수도 있다. 따라서, 도 11 의 방법은 하나 이상의 가상 참조 디코더 (HRD) 파라미터들을 표시하는 VPS 의 데이터를 코딩하는 것과, HDR 파라미터들에 기초하여 하나 이상의 계층들의 데이터를 코딩하는 것을 포함하는 방법의 예를 나타낸다.

[0302] 도 12 는 VPS 에서 시그널링되는 확장 데이터에 적어도 부분적으로 기초하여 비디오 데이터를 코딩하는 일 예의 방법을 예시하는 플로우차트이다. 도 12 의 방법은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 수행될 수도 있다. 예시의 목적들을 위하여, 도 12 의 방법은 비디오 디코더 (30) 에 대하여 설명된다.

[0303] 이 예에서, 비디오 디코더 (30) 는 VPS 가 확장 데이터를 포함하는지 여부를 표시하는 VPS 의 데이터를 디코딩한다 (180). 예를 들어, 비디오 디코더 (30) 는 VPS 의 vps_extension_flag 를 디코딩할 수도 있다. 다음으로, 비디오 디코더 (30) 는 데이터가 VPS 가 확장 데이터를 포함하고 있음을 표시하는지 여부를 결정한다 (182). 데이터가 VPS 가 확장 데이터를 포함하고 있음을 표시할 경우 (182 의 "예" 가지), 비디오 디코더 (30) 는 하나 이상의 확장 코딩 툴들에 대한 VPS 확장 데이터를 코딩하고 (184), 확장 코딩 툴들 및 확장 데이터를 이용하여 비디오 데이터를 디코딩한다 (186). 다른 한편으로, 데이터가 VPS 가 확장 데이터를 포함하고 있지 않음을 표시할 경우 (182 의 "아니오" 가지), 비디오 디코더 (30) 는 기존의 코딩 툴들을 이용하여 비디오 데이터를 디코딩할 수도 있다 (188). 이러한 방식으로, 도 12 의 방법은, VPS 의 확장 데이터에 기초하여 비디오 데이터를 코딩하는 것 뿐만 아니라, VPS 가 대응하는 표준을 넘어서는 확장을 포함하는지 여부를 표시하는 VPS 의 데이터와, VPS 가 확장을 포함할 때, 확장을 위한 데이터를 코딩하는 것을 포함하는 방법의 예를 나타낸다.

[0304] 예에 따라서는, 본원에서 설명된 기술들 중 임의의 것의 어떤 액트 (act) 를 또는 이벤트들이 상이한 시퀀스에서 수행될 수 있거나, 추가될 수도 있거나, 병합될 수도 있거나, 또는 완전히 배제될 수도 있다 (예를 들어, 모든 설명된 액트들 또는 이벤트들이 교시 내용들의 실시를 위하여 필요한 것은 아님) 는 것을 인식해야 한다.

또한, 어떤 예들에서, 액트들 또는 이벤트들은 순차적인 것이 아니라, 예를 들어, 멀티-스레딩된 (multi-threaded) 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해 동시에 수행될 수도 있다.

[0305] 하나 이상의 예들에서는, 설명된 기능들이 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수

도 있다. 소프트웨어로 구현될 경우, 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터 판독가능 매체 상에 저장되거나 컴퓨터 판독가능 매체를 통해 송신될 수도 있고, 하드웨어-기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라 하나의 장소로부터 또 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비-일시적인 유형의 컴퓨터 판독가능 저장 매체들, 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기술들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0306]

제한이 아닌 예로서, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래쉬 메모리, 또는 명령들 또는 데이터 구조들의 형태로 희망하는 프로그램 코드를 저장하기 위해 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속은 컴퓨터 판독가능 매체라고 적절하게 칭해진다. 예를 들어, 동축 케이블, 광섬유 케이블, 트위스트 페어 (twisted pair), 디지털 가입자 회선 (digital subscriber line; DSL), 또는 적외선, 라디오 (radio), 및 마이크로파 (microwave) 와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신되는 경우, 동축 케이블, 광섬유 케이블, 트위스트 페어, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하는 것이 아니라, 그 대신에, 비-일시적인 유형의 저장 매체들에 관한 것이라는 것을 이해해야 한다. 본 원에서 이용된 바와 같은 디스크 (disk) 및 디스크 (disc) 는 컴팩트 디스크 (compact disc; CD), 레이저 디스크 (laser disc), 광학 디스크 (optical disc), 디지털 다기능 디스크 (digital versatile disc; DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (Blu-ray disc) 를 포함하고, 여기서, 디스크 (disk) 들은 통상 데이터를 자기적으로 재생하는 반면, 디스크 (disc) 들은 데이터를 레이저들로 광학적으로 재생한다. 상기의 조합들은 컴퓨터 판독가능 매체들의 범위 내에 또한 포함되어야 한다.

[0307]

명령들은 하나 이상의 디지털 신호 프로세서 (DSP) 들, 범용 마이크로프로세서들, DSP 들, 주문형 집적 회로 (ASIC) 들, 필드 프로그래밍가능 로직 어레이 (FPGA) 들, 또는 다른 등가의 통합된 또는 별개의 로직 회로부와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 이용된 바와 같은 용어 "프로세서" 는 상기한 구조 중 임의의 것 또는 본원에서 설명된 기술들의 구현을 위해 적당한 임의의 다른 구조를 지칭 할 수도 있다. 추가적으로, 일부 양태들에서는, 본원에서 설명된 기능성이 인코딩 및 디코딩을 위해 구성되거나 조합된 코덱 내에 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 기술들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에서 완전히 구현될 수 있다.

[0308]

또 다른 예들에서, 본 개시물은 그 상에 저장된 데이터 구조를 포함하는 컴퓨터 판독가능 매체를 구상하고, 여기서, 데이터 구조는 본 개시물과 부합하는 인코딩된 비트스트림을 포함한다. 특히, 인코딩된 비트스트림은 비디오 데이터의 하나 이상의 계층들과, 비디오 데이터의 하나 이상의 계층들에 대한 비디오 파라미터 (VPS) 를 포함할 수도 있고, 여기서, 비디오 데이터의 하나 이상의 계층들의 각각은 VPS 를 참조하고, 비디오 데이터의 하나 이상의 계층들은 VPS 에 적어도 부분적으로 기초하여 코딩된다.

[0309]

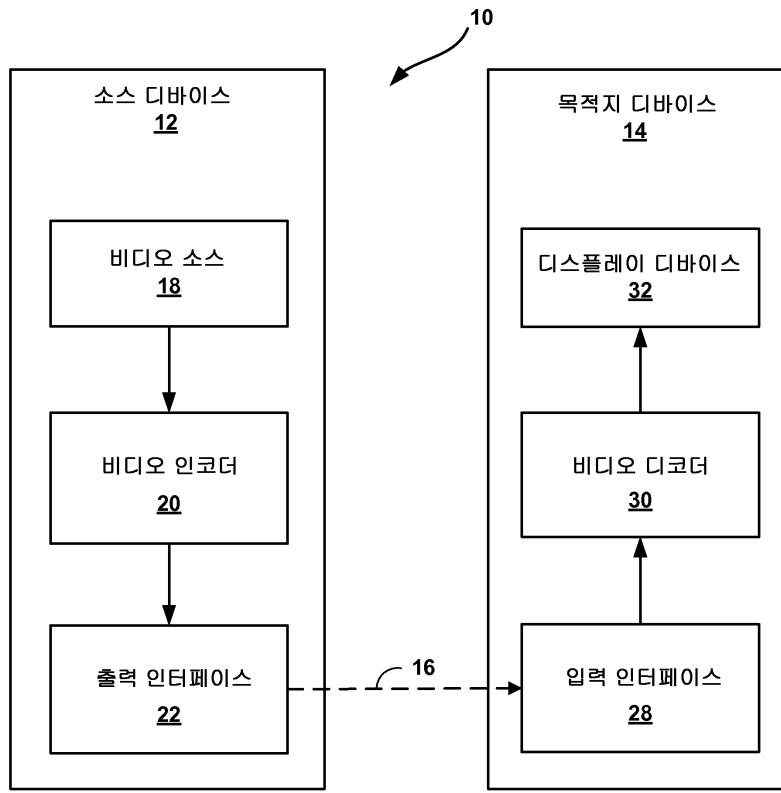
본 개시물의 기술들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들어, 칩셋) 를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위하여 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛 내에 조합될 수도 있거나, 적당한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는 상호동작 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

[0310]

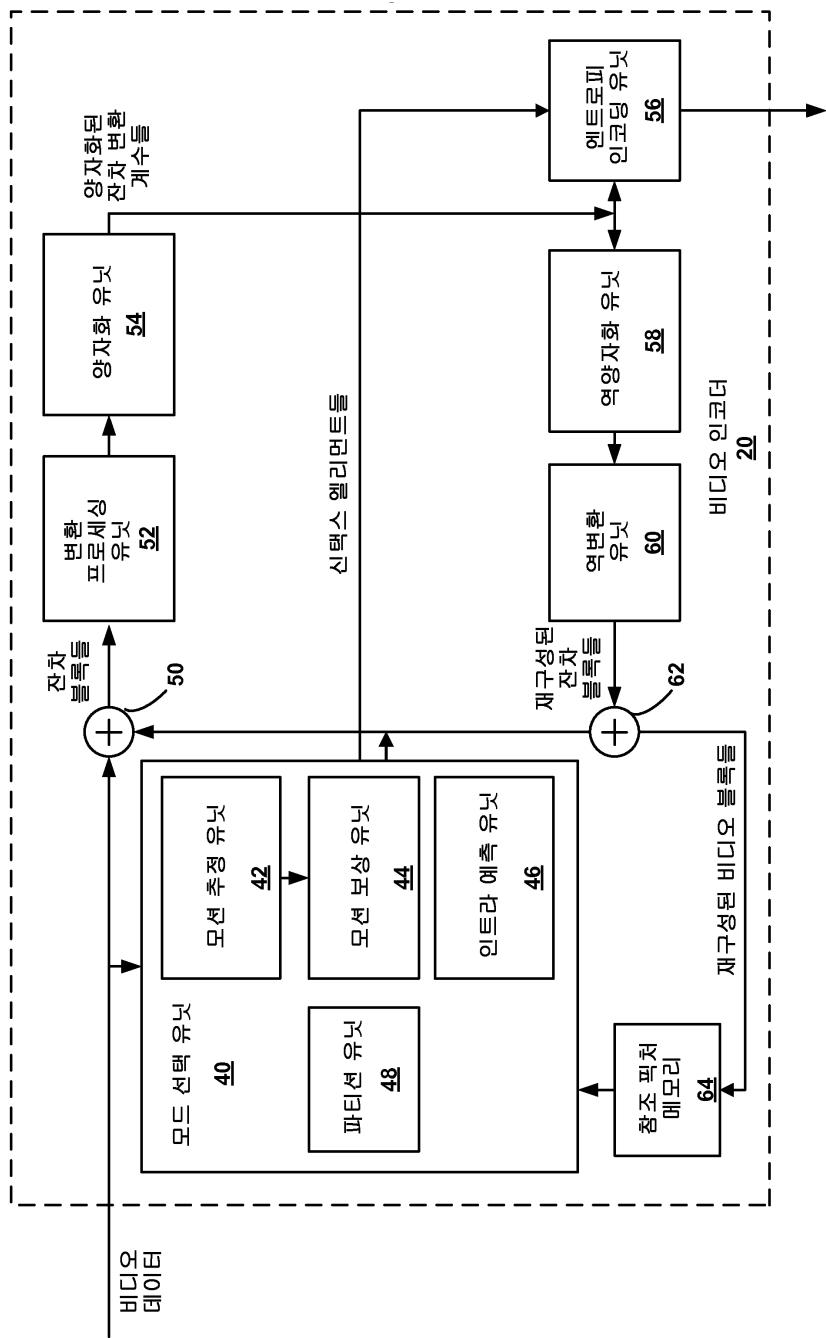
다양한 예들이 설명되었다. 이러한 그리고 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

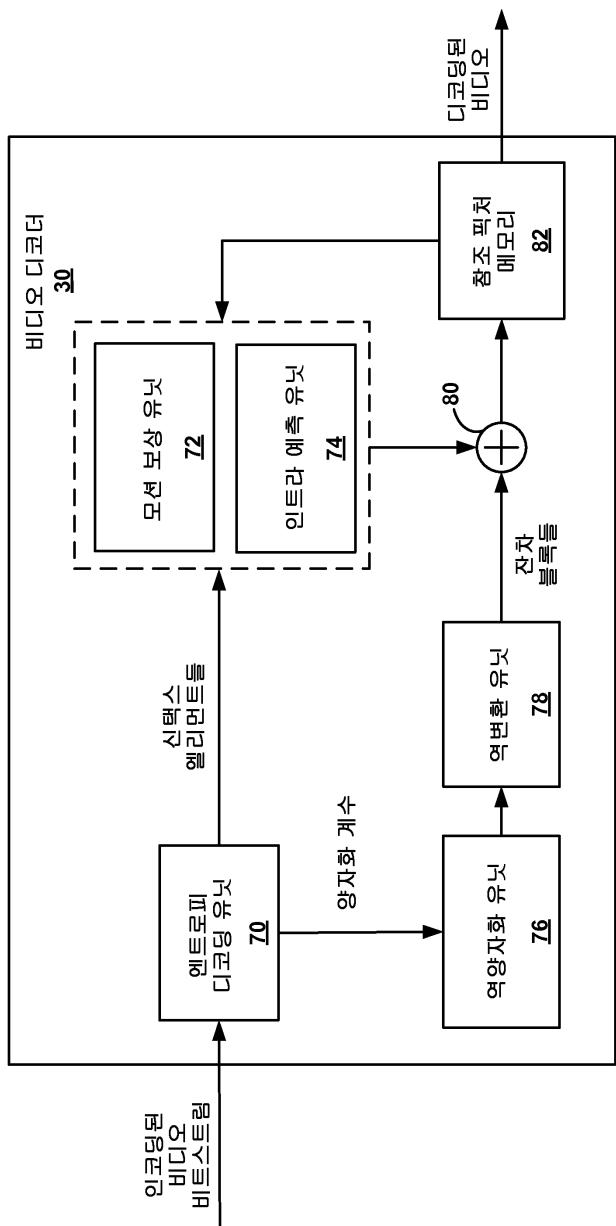
도면1



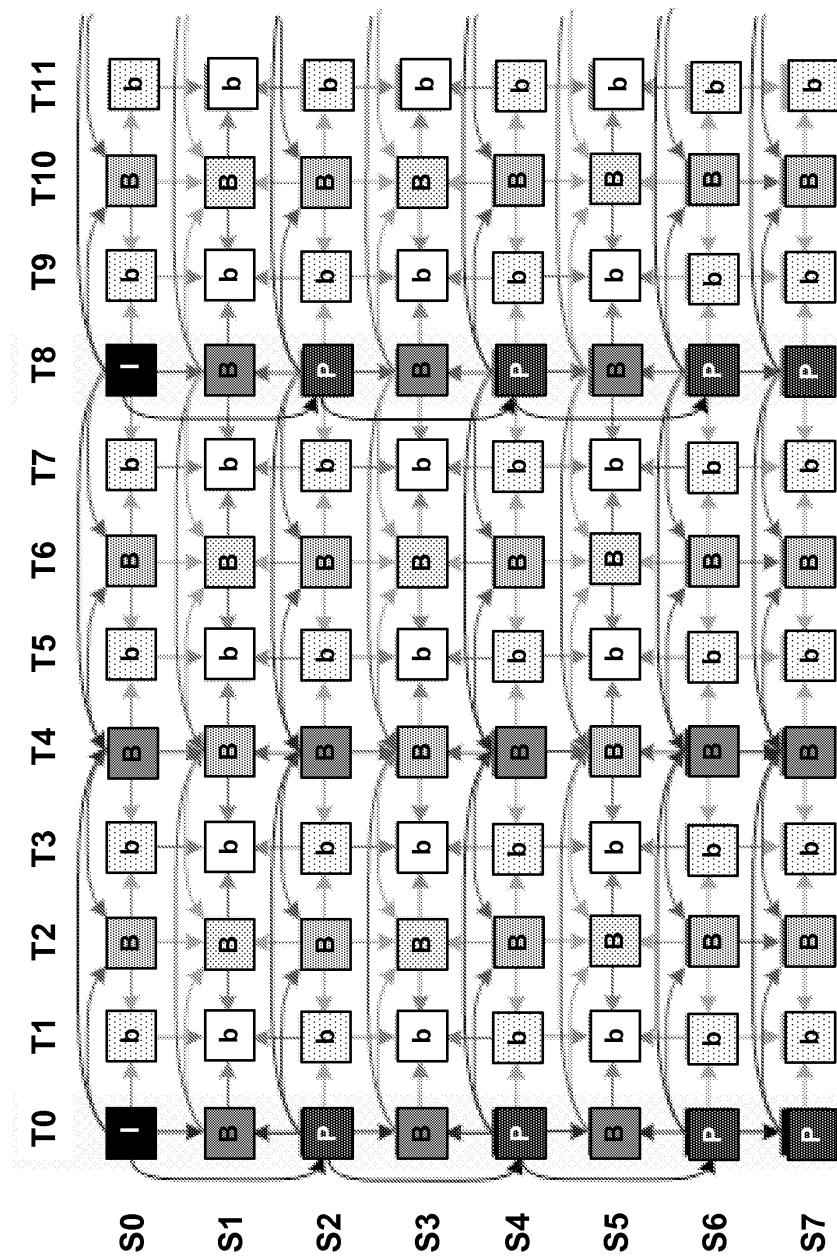
도면2



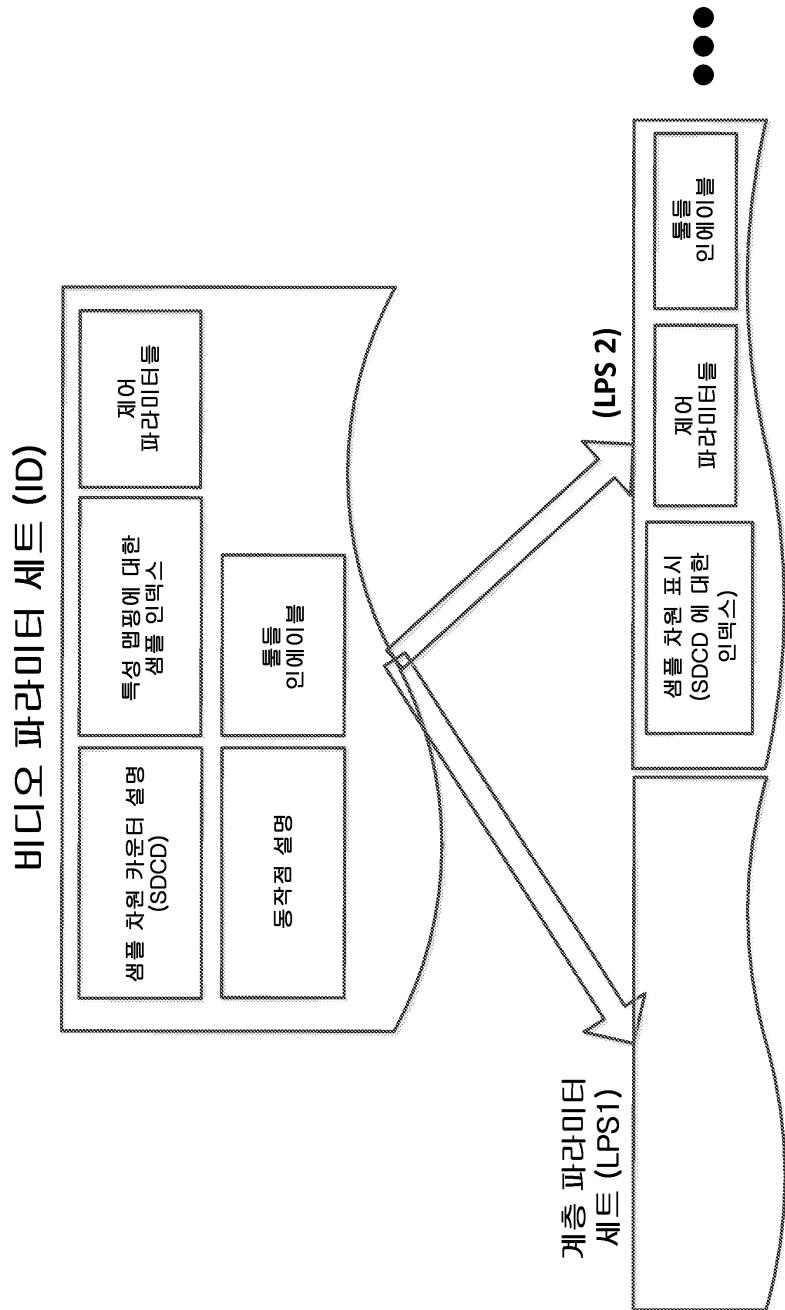
도면3



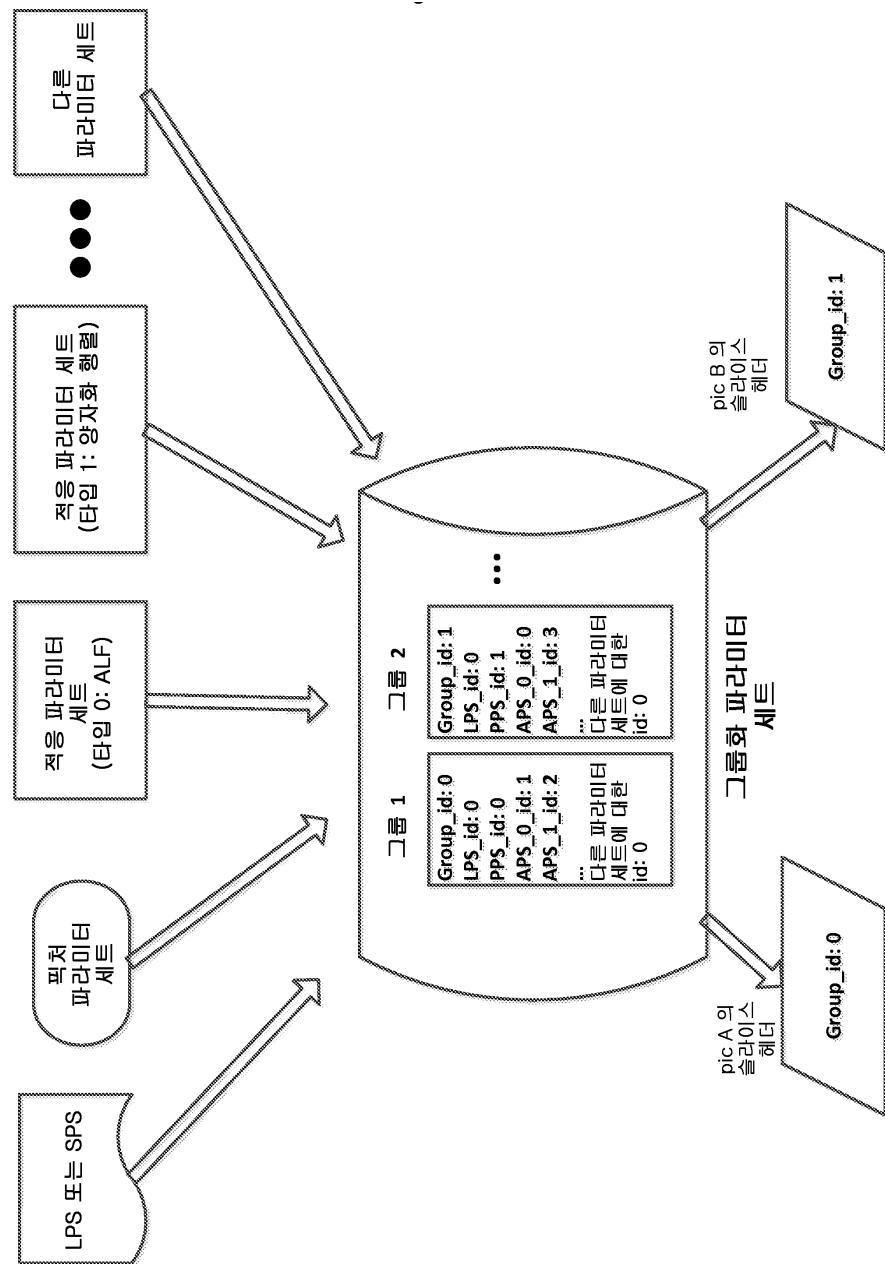
도면4



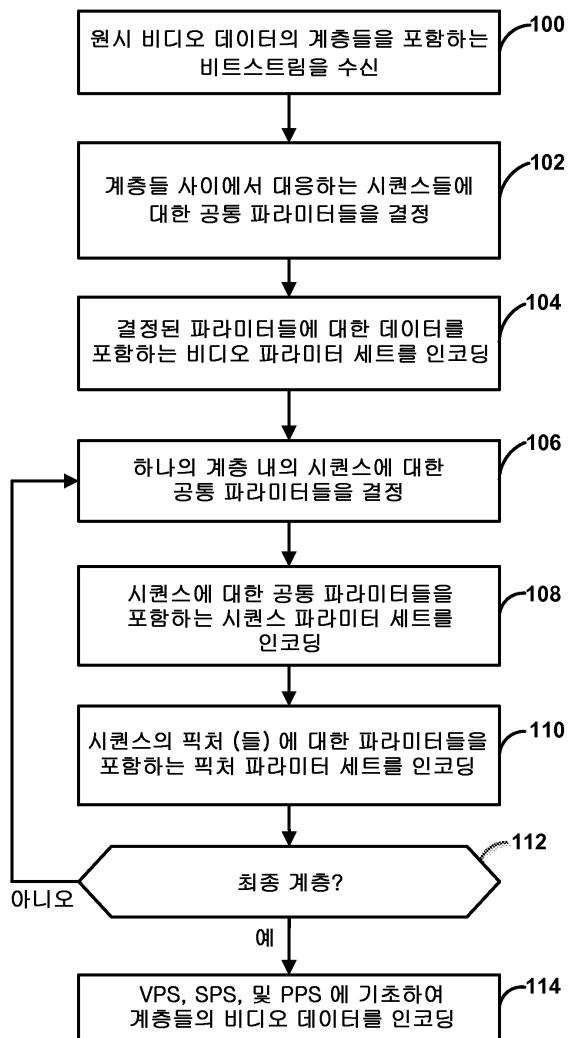
도면5



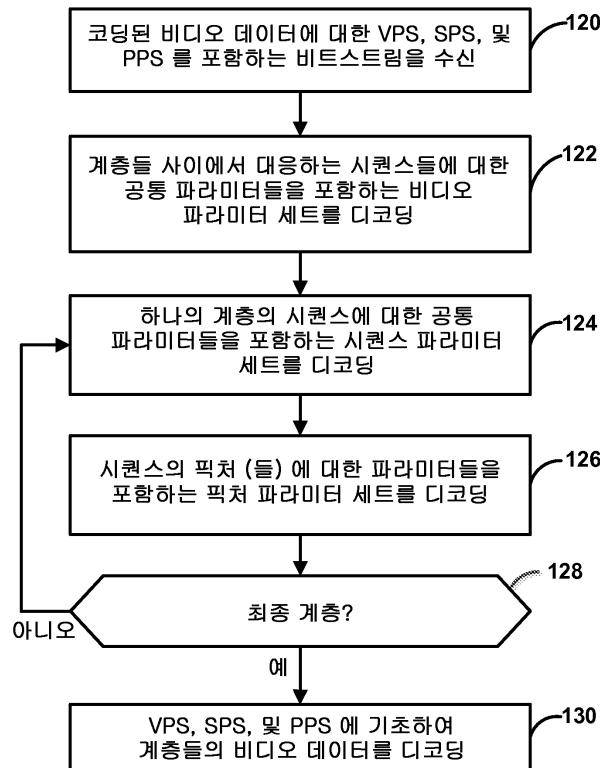
도면6



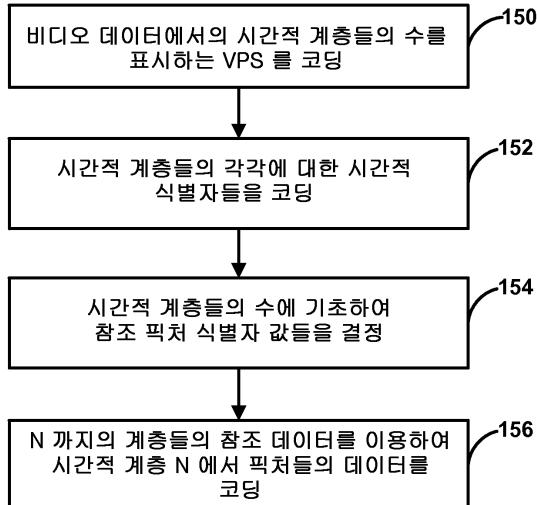
도면7



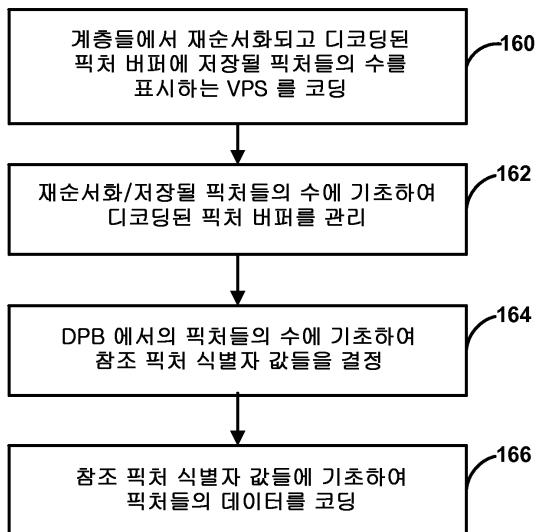
도면8



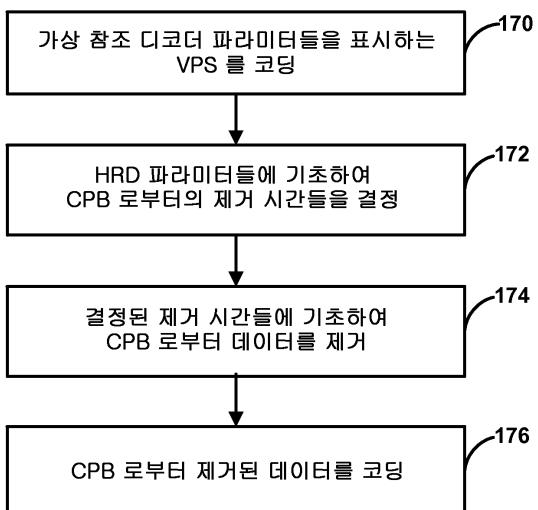
도면9



도면10



도면11



도면12

