



등록특허 10-2383620



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년04월05일
(11) 등록번호 10-2383620
(24) 등록일자 2022년04월01일

- (51) 국제특허분류(Int. Cl.)
H04N 19/31 (2014.01) *H04N 19/105* (2014.01)
H04N 19/136 (2014.01) *H04N 19/184* (2014.01)
H04N 19/187 (2014.01) *H04N 19/423* (2014.01)
H04N 19/44 (2014.01) *H04N 19/463* (2014.01)
H04N 19/503 (2014.01) *H04N 19/70* (2014.01)
H04N 19/85 (2014.01)
- (52) CPC특허분류
H04N 19/31 (2015.01)
H04N 19/105 (2015.01)
- (21) 출원번호 10-2016-7034888
- (22) 출원일자(국제) 2015년06월17일
심사청구일자 2020년06월02일
- (85) 번역문제출일자 2016년12월13일
- (65) 공개번호 10-2017-0021781
- (43) 공개일자 2017년02월28일
- (86) 국제출원번호 PCT/US2015/036175
- (87) 국제공개번호 WO 2015/195763
국제공개일자 2015년12월23일
- (30) 우선권주장
62/013,965 2014년06월18일 미국(US)
14/741,291 2015년06월16일 미국(US)

(56) 선행기술조사문헌

Y-K. Wang, et al. AHG10 output text. JCT-VC of ITU-T and ISO/IEC. JCTVC-R0010 Ver.1, Jun. 14, 2014, pp.1-151

Y-K. Wang, et al. MV-HEVC/SVVC HLS: Signalling and use of HRD parameters for bitstream partitions. JCT-VC of ITU-T and ISO/IEC. JCTVC-R0231 Ver.1, Jun. 22, 2014, pp.1-3

전체 청구항 수 : 총 11 항

심사관 : 이상래

(54) 발명의 명칭 **비트스트림 파티션들에 대한 HRD 파라미터들의 시그널링****(57) 요약**

일 예에서, 비디오 데이터를 코딩 (예를 들어, 인코딩 또는 디코딩) 하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리, 및 비디오 비트스트림의 각각의 비트스트림 파티션의 각각의 서브-계층에 대한 가상 참조 디코더 (HRD) 파라미터들을 코딩하고; HRD 파라미터들을 이용하여 비트스트림을 프로세싱하도록 구성된 비디오

(뒷면에 계속)

대 표 도

코더를 포함한다. 비디오 코더는 서브-계층들의 각각에 대해 반복되는 HRD 파라미터들에 대해 루프의 형태로 HRD 파라미터들, 및/또는 서브-계층들의 각각에 대한 HRD 파라미터들의 개개의 세트들에 대한 인덱스들을 코딩할 수도 있다.

(52) CPC특허분류

H04N 19/136 (2015.01)

H04N 19/184 (2015.01)

H04N 19/187 (2015.01)

H04N 19/423 (2015.01)

H04N 19/44 (2015.01)

H04N 19/463 (2015.01)

H04N 19/503 (2015.01)

H04N 19/70 (2015.01)

H04N 19/85 (2015.01)

명세서

청구범위

청구항 1

HEVC (High Efficiency Video Coding) 표준, 그 멀티뷰 확장, 또는 그 스케일러블 확장에 따라 비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터는 비디오 비트스트림을 포함하고, 상기 비디오 비트스트림은 계층들을 포함하며, 상기 비디오 비트스트림의 계층들은 1 초과의 비트스트림 파티션으로 분할되며,

상기 방법은,

복수의 HRD 파라미터들의 세트들을 디코딩하는 단계;

각각의 출력 계층 세트 및 각각의 비트스트림 파티션의 각각의 서브-계층에 대해, 상기 복수의 HRD 파라미터들의 세트들 중의 HRD 파라미터들의 세트를 나타내는 인덱스를 디코딩하는 단계로서, 상기 비디오 비트스트림으로부터 렌더링될 수 있는 플레이백 프레임 레이트들의 각각은 하나 이상의 서브-계층들의 세트에 대응하고, 각각의 점진적으로 더 높은 서브-계층은 그 서브-계층에서 그리고 그 서브-계층 아래에서 상기 비디오 데이터의 모든 프레임들을 포함하는, 상기 인덱스를 디코딩하는 단계; 및

상기 비트스트림의 인코딩된 픽처들을 디코딩하는 단계,

디코딩된 상기 픽처들을 DPB (decoded picture buffer) 에 저장하는 단계, 및

상기 HRD 파라미터들에 따라 상기 DPB로부터 디코딩된 상기 픽처들을 삭제하는 단계

에 의해 상기 HRD 파라미터들을 사용하여 상기 비디오 비트스트림을 프로세싱하는 단계

를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

삭제

청구항 3

제 1 항에 있어서,

상기 인덱스(들)를 디코딩하는 단계는, 출력 순서에서의 연속적인 픽처들의 HRD 출력 시간들 사이의 시간적 거리들이 상기 비트스트림의 서브-계층에 대하여 제약되는지 여부를 적어도 나타내는 신택스 엘리먼트들의, 상기 HRD 파라미터들이 코딩되는 서브-계층들의 수와 동일한, 수를 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

삭제

청구항 5

제 3 항에 있어서,

상기 출력 순서에서의 연속적인 픽처들의 HRD 출력 시간들 사이의 시간적 거리들이 제약되는지 여부를 적어도 나타내는 상기 신택스 엘리먼트들은, fixed_pic_rate_general_flag[] 신택스 엘리먼트들 또는 fixed_pic_rate_within_CVS_flag[] 신택스 엘리먼트들 중 적어도 하나를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

삭제

청구항 7

제 1 항에 있어서,

상기 복수의 HRD 파라미터들의 세트들을 디코딩하는 단계는, 대응하는 상기 서브-계층들에 대한 연속적인 픽처들의 HRD 출력 시간들을 특정하는 기본 유닛들 사이의, 클록 틱 (clock tick) 들에 있어서의, 시간적 거리를 나타내는 신팩스 엘리먼트들을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

삭제

청구항 9

제 1 항에 있어서,

상기 복수의 HRD 파라미터들의 세트들을 디코딩하는 단계는, 대응하는 상기 서브-계층들의 HRD 동작 모드를 나타내는 신팩스 엘리먼트들을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 10

삭제

청구항 11

제 1 항에 있어서,

상기 방법은 무선 통신 디바이스 상에서 실행가능하고,

상기 무선 통신 디바이스는,

상기 HRD 파라미터들을 저장하도록 구성된 메모리;

상기 메모리에 저장된 상기 HRD 파라미터들 및 상기 비디오 비트스트림을 프로세싱하기 위해 명령들을 실행하도록 구성된 프로세서; 및

상기 비디오 비트스트림 및 상기 HRD 파라미터들을 포함하는 신호를 수신하고 상기 HRD 파라미터들을 상기 메모리에 저장하기 위한 수신기

를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 12

제 11 항에 있어서,

상기 무선 통신 디바이스는 셀룰러 전화기이고,

상기 신호는 상기 수신기에 의해 수신되고 셀룰러 통신 표준에 따라 변조되는, 비디오 데이터를 디코딩하는 방법.

청구항 13

HEVC (High Efficiency Video Coding) 표준, 그 멀티뷰 확장, 또는 그 스케일러블 확장에 따라 비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 비디오 데이터는 비디오 비트스트림을 포함하고, 상기 비디오 비트스트림은 계층들을 포함하며, 상기 비디오 비트스트림의 계층들은 1 초과의 비트스트림 파티션으로 분할되며,

상기 디바이스는,

비디오 데이터를 저장하도록 구성된 메모리; 및

비디오 디코더

를 포함하고,

상기 비디오 디코더는,

복수의 HRD 파라미터들의 세트들을 디코딩하고;

각각의 출력 계층 세트 및 각각의 비트스트림 파티션의 각각의 서브-계층에 대해, 상기 복수의 HRD 파라미터들의 세트들 중의 HRD 파라미터들의 세트를 나타내는 인덱스를 디코딩하는 것으로서, 상기 비디오 비트스트림으로부터 렌더링될 수 있는 플레이백 프레임 레이트들의 각각은 하나 이상의 서브-계층들의 세트에 대응하고, 각각의 점진적으로 더 높은 서브-계층은 그 서브-계층에서 그리고 그 서브-계층 아래에서 상기 비디오 데이터의 모든 프레임들을 포함하는, 상기 인덱스를 디코딩하며; 그리고

상기 비트스트림의 인코딩된 픽처들을 디코딩하는 것,

디코딩된 상기 픽처들을 DPB (decoded picture buffer)에 저장하는 것, 및

상기 HRD 파라미터들에 따라 상기 DPB로부터 디코딩된 상기 픽처들을 삭제하는 것

에 의해 상기 HRD 파라미터들을 사용하여 상기 비디오 비트스트림을 프로세싱하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 14

삭제

청구항 15

제 13 항에 있어서,

상기 디바이스는 무선 통신 디바이스이고,

상기 비디오 비트스트림 및 상기 HRD 파라미터들을 포함하는 신호를 수신하고 상기 HRD 파라미터들을 상기 메모리에 저장하도록 구성된 수신기를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 16

제 15 항에 있어서,

상기 무선 통신 디바이스는 셀룰러 전화기이고,

상기 신호는 상기 수신기에 의해 수신되고 셀룰러 통신 표준에 따라 변조되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 17

명령들이 저장된 컴퓨터 관독가능 저장 매체로서,

상기 명령들은, 프로세서에 의해 실행되는 경우, 상기 프로세서로 하여금, 제 1 항, 제 3 항, 제 5 항, 제 7 항, 제 9 항 및 제 11 항 내지 제 12 항 중 어느 한 항에 기재된 방법을 수행하게 하는, 컴퓨터 관독가능 저장 매체.

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2014년 6월 18일자로 출원된 미국 가출원 제62/013,965호의 이익을 주장하고, 이 미국 가출원은 이로써 그 전체가 참조로 포함된다.

기술분야

[0003] 본 개시물은 비디오 코딩에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대 정보 단말기 (personal digital assistant; PDA) 들, 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 (e-book) 판독기들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰들", 원격화상회의 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, 어드밴스드 비디오 코딩 (Advanced Video Coding; AVC) 에 의해 정의된 표준들, 현재 개발 중인 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, 및 이러한 표준들의 확장들에서 설명되는 것들과 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005]

비디오 코딩 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거하기 위해 공간적 (인트라-픽처) 예측 및/또는 시간적 (인터-픽처) 예측을 포함한다. 블록-기반 비디오 코딩을 위해, 비디오 슬라이스 (예를 들어, 비디오 프레임 또는 비디오 프레임의 일부) 는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 또한 트리블록들, 코딩 유닛 (CU) 들 및/또는 코딩 노드들이라고 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간적 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간적 예측 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들이라고 지칭될 수도 있으며, 참조 픽처들은 참조 프레임들이라고 지칭될 수도 있다.

[0006]

공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록을 발생시킨다. 잔차 데이터 (residual data) 는 코딩될 오리지널 블록과 예측 블록 사이의 픽셀 차이들을 표현한다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터, 및 코딩된 블록과 예측 블록 사이의 차이를 나타내는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가적인 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 변환 계수들을 발생시킬 수도 있으며, 이 잔차 변환 계수들은 그 후에 양자화될 수도 있다. 초기에 2 차원 어레이로 배열되는, 양자화된 변환 계수들은 변환 계수들의 1 차원 벡터를 생성하기 위해 스캐닝될 수도 있으며, 엔트로피 코딩이 적용되어 훨씬 더 많은 압축을 달성할 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0007]

일반적으로, 본 개시물은 비트스트림 파티션들에 대한 가상 참조 디코더 (hypothetical reference decoder; HRD) 파라미터들을 시그널링하기 위한 기법들을 설명한다. 즉, 본 개시물의 기법들은, 예를 들어, 다중-계층 비디오 코딩에서, 비트스트림 파티션들에 대한 HRD 파라미터들의 시그널링을 개선시킬 수도 있다. 비디오 비트스트림은 시간적 차원 (temporal dimension), 뷰 차원 (예를 들어, 멀티-뷰 비디오 데이터용), 스케일러빌리티 (scalability) 차원 (예를 들어, 스케일러블 비디오 코딩용) 등과 같은 다양한 차원들에서의 다양한 계층들을 포함할 수도 있다. 비트스트림 파티션들에 대한 HRD 파라미터 시그널링을 개선시킬 수도 있는, 단독으로 또는 임의의 조합으로 이용될 수도 있는 다양한 기법들이 설명되고, 이 비트스트림 파티션들 중 임의의 것 또는 전부는 비디오 디코더에 의한 후속 디코딩을 위해 개별적으로 추출될 수도 있다.

[0008]

하나의 예에서, 비디오 데이터를 코딩 (예를 들어, 인코딩 또는 디코딩) 하는 방법은 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 신팩스 엘리먼트에 대한 값을 코딩하는 단계로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트 (video parameter set; VPS) 에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 신팩스 엘리먼트에 대한 값을 코딩하는 단계, 신팩스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩하는 단계, 및 HRD 파라미터들을 이용하여 비트스트림을 프로세싱하는 단계를 포함한다.

[0009]

다른 예에서, 비디오 데이터를 코딩 (예를 들어, 인코딩 또는 디코딩) 하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리, 및 비디오 코더를 포함하고, 그 비디오 코더는 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 신팩스 엘리먼트에 대한 값을 코딩하는 것으로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트 (VPS) 에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 신팩스 엘리먼트에 대한 값을 코딩하고, 신팩스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩하며, HRD 파라미터들을 이용하여 비트스트림을 프로세싱하도록 구성된다.

[0010]

다른 예에서, 비디오 데이터를 코딩 (예를 들어, 인코딩 또는 디코딩) 하기 위한 디바이스는 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 신팩스 엘리먼트에 대한 값을 코딩하는 수단으로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트 (VPS) 에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 신팩스 엘리먼트에 대한 값을

코딩하는 수단, 선택스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩하는 수단, 및 HRD 파라미터들을 이용하여 비트스트림을 프로세싱하는 수단을 포함한다.

[0011] 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행될 때, 하나 이상의 프로세서들로 하여금, 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 선택스 엘리먼트에 대한 값을 코딩하게 하는 것으로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트 (VPS) 에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 선택스 엘리먼트에 대한 값을 코딩하게 하고, 선택스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩하게 하며, HRD 파라미터들을 이용하여 비트스트림을 프로세싱하게 하는 명령들로 인코딩된다.

[0012] 하나 이상의 예들의 상세들은 아래의 설명 및 첨부 도면들에 제시된다. 다른 피처들, 목적들, 및 이점들은 이 설명 및 도면들, 그리고 청구항들로부터 명백해질 것이다.

도면의 간단한 설명

[0013] 도 1 은 가상 참조 디코더 (HRD) 파라미터 시그널링을 개선시키기 위한 기법들을 활용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 가상 참조 디코더 (HRD) 파라미터 시그널링을 개선시키기 위한 기법들을 구현할 수도 있는 비디오 인코더의 예를 예시하는 블록도이다.

도 3 은 가상 참조 디코더 (HRD) 파라미터 시그널링을 개선시키기 위한 기법들을 구현할 수도 있는 비디오 디코더의 예를 예시하는 블록도이다.

도 4 는 본 개시물의 기법들에 따른 비디오 데이터를 인코딩하기 위한 예시적인 방법을 예시하는 플로우차트이다.

도 5 는 본 개시물의 기법들에 따른 비디오 데이터를 디코딩하기 위한 예시적인 방법을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0014] 일반적으로, 본 개시물은 가상 참조 디코더 (hypothetical reference decoder; HRD) 파라미터들을 코딩 (예를 들어, 인코딩 또는 디코딩) 하는 것에 관련된 기법들을 설명한다. 일반적으로, 비디오 코딩 프로세스를 위해 코딩된 픽처의 사이즈들을 제어할 뿐만 아니라 타임라인들을 관리하기 위해 HRD 파라미터들이 이용된다.

예를 들어, 비디오 코더는 픽처를 디코딩할 목적들을 위해 코딩된 픽처 버퍼 (coded picture buffer; CPB)로부터 인코딩된 픽처를 추출할 때를 결정하거나, 및/또는 디코딩된 픽처 버퍼 (decoded picture buffer; DPB)로부터 디코딩된 픽처를 추출, 출력, 및/또는 제거할 때를 결정하기 위해 HRD 파라미터들을 이용할 수도 있다.

[0015] 비디오 비트스트림은 다양한 상이한 디코딩 및 렌더링 디바이스들에 의해 이용될 수 있는 코딩된 비디오 데이터를 포함할 수도 있다. 예를 들어, 비디오 디코더는 다양한 디코딩 툴들을 구현할 수도 있는 다양한 비디오 코딩 표준 프로파일들 및 레벨들을 지원할 수도 있다. 이와 유사하게, 비디오 렌더링 디바이스 (예를 들어, 디스플레이)는 다양한 렌더링 능력들 (예를 들어, 리프레시 레이트/프레임 레이트, 동시에 플레이될 수 있는 뷰들의 개수, 인터레이싱된 또는 점진적인 스캔 플레이백 (playback) 등)을 지원할 수도 있다. 이러한 방식으로, 단일 비디오 비트스트림은 다수의 상이한 비디오 디코더들 및 렌더링 디바이스들에 의해 이용가능할 수도 있다.

[0016] 하나의 예로서, 비디오 비트스트림은 다양한 프레임 레이트들이 비디오 비트스트림으로부터 렌더링될 수 있을 때 시간적 스케일러빌리티 (scalability)를 지원한다고 말할 수도 있다. 예를 들어, 동일한 비디오 비트스트림은 15 FPS (frames per second), 30 FPS, 60 FPS, 120 FPS, 및 240 FPS 의 프레임 레이트들을 갖는 비디오를 렌더링하는데 이용될 수도 있다. 일반적으로, 이들 다양한 플레이백 프레임 레이트들 각각은 비트스트림의 하나 이상의 "서브-계층들"의 세트에 대응한다. 각각의 점진적으로 보다 높은 계층은 그 서브-계층에서 와 그 서브-계층 아래에서의 모든 프레임들을 포함한다. 따라서, 15 FPS 플레이백을 위한 픽처들은 서브-계층 0 픽처들을 포함할 수도 있고, 30 FPS 플레이백을 위한 픽처들은 서브-계층들 0 및 서브-계층 1 픽처들을 포함할 수도 있고, 60 FPS 플레이백을 위한 픽처들은 서브-계층들 0, 1, 및 2 의 픽처들을 포함할 수도 있다는 것 등이 있다.

[0017] 이러한 방식으로, 디바이스가 비디오 비트스트림에 의해 지원되는 최대 프레임 레이트보다 더 낮은 프레임 레이

트에서 플레이백을 수행하도록 구성될 때, 디바이스는 플레이백을 위해 필요한 픽처들만을 추출 및 디코딩하기 위해, 비트스트림으로부터의 서브-비트스트림 추출을 수행할 수도 있다. 위의 예를 계속하면, 디바이스가 60 FPS 플레이백을 수행하는 것으로 결정한 경우, 디바이스는 서브-계층들 0, 1, 및 2 의 픽처들을 추출하고, 이들 픽처들만을 (즉, 서브-계층들 3 및 4 의 픽처들을 디코딩하는 일 없이) 디코딩할 수도 있다.

[0018] 비디오 파라미터 세트 (video parameter set; VPS) 신택스 구조는 비트스트림에 포함될 수 있는 서브-계층들의 최대 개수를 나타내는 데이터를 포함할 수도 있다. 따라서, HRD 파라미터들은 최대 개수의 서브-계층들 각각에 대해 시그널링될 수도 있다. 그러나, (예를 들어, 시간적 스케일러빌리티의 목적들을 위한) 서브-비트스트림 추출은 서브-계층들의 최대 개수보다 더 적은 서브-계층을 갖는 추출된 서브-비트스트림을 발생시킬 수도 있다. 최대 개수의 서브-계층들 각각에 대한 정보를 시그널링하기보다는, 본 개시물은 (VPS 에 의해 나타낸 서브-계층들의 최대 개수 이하일 수도 있는) 비트스트림에 실제로 포함되는 개수의 서브-계층들에 대해서만 HRD 파라미터들을 시그널링하기 위한 기법들을 설명한다. 이러한 방법으로, 이들 기법들은 HRD 파라미터들이 최대 개수의 서브-계층들 각각에 대해 시그널링되는 기법들에 의해 비트 절약들을 달성할 수도 있다.

[0019] 이와 유사하게, 본 개시물은 비트스트림의 각각의 파티션의 각각의 서브-계층에 대한 HRD 파라미터들을 시그널링하기 위한 기법들을 설명한다. 예를 들어, VPS 는, 다수의 가능한 출력 계층 세트들 각각을 통해 반복되고, 각각의 가능한 출력 계층 세트들에 대해, 대응하는 출력 계층 세트에 포함된 서브-계층들에 대한 HRD 파라미터들을 시그널링하는 파라미터들의 루프를 포함할 수도 있다.

[0020] 게다가, 본 개시물은 비트스트림 파티션 초기 도달 시간 보충 향상 정보 (supplemental enhancement information; SEI) 메시지에서 비디오 코딩 계층 (video coding layer; VCL) HRD 파라미터들을 조건부로 시그널링하기 위한 기법들을 설명한다. 이것은 이러한 파라미터들이 소정의 조건들에서 불필요하게 시그널링될 수도 있는 기존 기법들의 소정의 잠재적인 결점들을 극복할 수도 있다.

[0021] 본 개시물의 기법들은 일반적으로, 『"SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services-Coding of moving video", 고효율 비디오 코딩, ITU-T H.265, 2013년 4월』에 설명되는, 고효율 비디오 코딩 (HEVC) 이라고도 또한 지칭되는 ITU-T H.265 에 대해 설명된다. 그러나, 이들 기법들은 다른 비디오 코딩 표준들에도 또한 적용될 수도 있다. 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 및 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC 라고도 알려짐) 를, 그의 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장들을 포함하여, 포함한다.

[0022] H.265 표준은 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 및 ISO/IEC 동화상 전문가 그룹 (Motion Picture Experts Group; MPEG) 의 비디오 코딩에 관한 합동 협력 팀 (Joint Collaboration Team on Video Coding; JCT-VC) 에 의해 최근에 마무리되었다. 이하 HEVC WD 라고 지칭되는 최근의 HEVC 초안 명세는, phenix.it-sudparis.eu/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1003-v1.zip 으로부터 입수가능하다. HEVC 에 대한 멀티뷰 확장, 즉, MV-HEVC 는 또한 JCT-3V 에 의해 개발되고 있다. 이하 MV-HEVC WD8 이라고 지칭되는 MV-HEVC 의 최근 작업 초안 (WD) 은, phenix.it-sudparis.eu/jct2/doc_end_user/documents/8_Valencia/wg11/JCT3V-H1002-v5.zip 으로부터 입수가능하다. HEVC 에 대한 스케일러블 확장, 즉, SHVC 는 또한 JCT-VC 에 의해 개발되고 있다. 이하 SHVC WD6 이라고 지칭되는 SHVC 의 최근 작업 초안 (WD) 은, phenix.it-sudparis.eu/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1008-v2.zip 으로부터 입수가능하다.

[0023] MV-HEVC WD8 및 SHVC WD6 은 비트스트림-파티션-특정 HRD 동작이라고 불리는 비트스트림 파티션 기반 HRD 동작의 명세를 포함하고, 여기서 비트스트림의 계층들은 하나보다 더 많은 비트스트림 파티션들로 분할될 수 있고, HRD 는 비트스트림-파티션-특정 HRD 파라미터들에 기초하여 동작할 수도 있다.

[0024] 『JCTVC-R0043v5 (phenix.int-evry.fr/jct/doc_end_user/documents/18_Sapporo/wg11/JCTVC-R0043-v5.zip 에서 입수가능) 및 Sullivan, "Ad hoc group report: Layered coding constraint specifications and capability indications (AHG10)" 에 대한 첨부물들에서의 AHG10 출력 텍스트, ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 비디오 코딩에 관한 합동 협력 팀 (JCT-VC), 18 차 회의: 일본 삿포로, 2014년 6월 30일 - 7월 9일, JCTVC-R0010v2, (이하, "JCTVC-R0010v2"), phenix.int-evry.fr/jct/doc_end_user/documents/18_Sapporo/wg11/JCTVC-R0010-v2.zip 에서 입수가능』은 적합성 (conformance) 정의 및 프로파일/티어/레벨의 비트스트림 파티션 기반 시그널링의 명세들을 포함한다. 접근법은 다음과 같이 요약된다:

- [0025]
 - 각각의 출력 계층 세트에 대해, 파티션들로의 계층들의 하나 이상의 파티셔닝 스킴들이 시그널링된다. 각각의 비트스트림 파티션은 하나 이상의 계층들을 포함할 수 있다.
- [0026]
 - 각각의 비트스트림 파티션에 대해 프로파일, 티어, 및 레벨 (profile, tier, and level; PTL) 의 세트가 시그널링된다.
- [0027]
 - 자연적으로 계층 특정되는 픽쳐 폭, 픽쳐 높이, 및 서브-DPB 사이즈에 대한 세 가지를 제외하고는, 모든 레벨 제한들 및 제약들이 비트스트림 파티션 특정되는 것으로 특정된다.
- [0028]
 - 디코더의 디코딩 능력은 PTL 트리플렛 (triplet) 들의 리스트를 준수하는 것으로서 표현되고, 여기서 리스트에서의 엔트리들의 개수는 다중-계층 디코더를 구축하는데 이용되는 단일-계층 디코더들의 개수를 나타내며, 각각의 PTL 트리플렛은 단일-계층 디코더들 중 하나의 단일-계층 디코더의 PTL 을 나타낸다.
- [0029]
 - PTL 트리플렛들의 리스트를 준수하는 디코더는 다음 조건을 만족시키는 적어도 하나의 파티셔닝 스킴이 존재하는 임의의 출력 계층 세트를 디코딩할 수 있는 것이 요구된다: 파티셔닝 스킴의 각각의 비트스트림 파티션에 대해, 다중-계층 디코더의 단일-계층 디코더들 중 하나는 비트스트림 파티션을 디코딩하기 위해 배타적으로 할당될 수 있다.
- [0030]
 - MV-HEVC WD8 및 SHVC WD6 에서의 비트스트림 파티션 기반 HRD 동작은 변경된 상태로 이용되어 다수의 파티셔닝 스킴들로 더 양호하게 작동하게 한다.
- [0031] MV-HEVC WD8, SHVC WD6, 및 JCTVC-R0010v2 에서의 비트스트림 파티션 기반 HRD 동작에서, HRD 파라미터들은 각각의 비트스트림 파티션에 대해 시그널링된다. 비트스트림 파티션들에 대한 HRD 파라미터들의 시그널링을 위한 기준 방법들은 다음의 단점들에 직면할 수도 있다:
 - 1) 각각의 hrd_parameters() 신택스 구조는 vps_max_sub_layer_minus1 + 1 개의 서브-계층들에 대한 정보를 포함하지만, 신택스 구조는 vps_max_sub_layer_minus1 + 1 보다 더 적은 개수의 서브-계층들을 갖는 비트스트림에 적용된다. 이 경우, 일부 비트들이 순전히 낭비된다.
 - 2) 각각의 비트스트림 파티션에 대해, 가장 높은 서브-계층에 대해서만 HRD 파라미터들이 시그널링되어, 그에 따라 비트스트림 파티션의 시간적 서브세트의 적합성이 정의될 수 없고, 상호동작 가능한 방식으로 출력 계층 세트의 시간적 서브세트만을 소비할 어떠한 방법도 존재하지 않는다.
 - 3) 비트스트림 파티션 초기 도달 시간 SEI 메시지에서, 결코 발생하지 않아야 하는 다음 2 개의 경우들 양쪽 모두가 발생할 수 있다:
 - a. NalHrdBpPresentFlag 가 1 일 때, vcl_initial_arrival_delay[i] 신택스 엘리먼트들을 통한 VCL HRD 파라미터들에 대한 초기 도달 딜레이는 VclHrdBpPresentFlag 가 1 과 동일한 경우라도 시그널링되지 않는다. 이 경우, VCL HRD 적합성이 정의될 수 없다.
 - b. NalHrdBpPresentFlag 가 0 일 때, vcl_initial_arrival_delay[i] 신택스 엘리먼트들을 통한 VCL HRD 파라미터들에 대한 초기 도달 딜레이는 VclHrdBpPresentFlag 가 0 과 동일한 경우라도 시그널링된다. 이 경우, 이를 시그널링은 순전히 비트들을 낭비하고 있다.
- [0033] 따라서, 위에서 언급된 바와 같이, 본 개시물은, 단독으로 또는 임의의 조합으로 이용될 수도 있고 위에서 논의된 단점들 중 임의의 것 또는 전부를 극복할 수도 있는 다양한 기법들을 설명한다. 본 개시물의 기법들의 요약은 추후의 섹션들에서 제공되는 일부 방법들의 상세한 구현과 함께 아래에 제공된다. 일반적으로, 아래에 넘버링되는 아이템들은 위에서 논의된 넘버링된 단점들을 해결할 수도 있다:
 - 1) 각각의 hrd_parameters() 신택스 구조는, 예를 들어 num_sub_layer_hrd_minus1[i] 라고 지칭되는 신택스 엘리먼트에 의해 시그널링되는 바와 같이, 필요한 서브-계층들의 개수에 대한 정보를 포함한다.
 - 2) 각각의 비트스트림 파티션에 대해, 각각의 서브-계층에 대한 HRD 파라미터들이 시그널링된다. 이것은 전달 스케줄들의 개수를 나타내는 신택스 엘리먼트들에 대한 출력 계층 세트에서의 서브-계층들의 개수와 동일한 개수 엔트리들, hrd_parameters() 신택스 구조들의 리스트에 대한 인덱스, 및 나타낸 hrd_parameters() 신택스 구조에서의 전달 스케줄들의 리스트에 대한 인덱스를 갖는 루프를 부가하는 것에 의해 달성될 수 있고, 또는 hrd_parameters() 신택스 구조의 리스트에 대한 인덱스만을 단순히 시그널링하고 나타낸 hrd_parameters() 신

택스 구조에서의 모든 전달 스케줄들을 이용한다.

[0040] 3) 비트스트림 파티션에서 VCL HRD 파라미터들에 대한 초기 도달 딜레이가 VclHrdBpPresentFlag 가 1 과 동일한 경우에만 존재하도록 초기 도달 시간 SEI 메시지 선택스가 변화된다.

[0041] 도 1 은 가상 참조 디코더 (HRD) 파라미터 시그널링을 개선시키기 위한 기법들을 활용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 추후의 시간에 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 비디오 데이터를 목적지 디바이스 (14) 에 컴퓨터 판독가능 매체 (16) 를 통해 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 테스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 전화 핸드셋들 예컨대 소위 "스마트" 폰들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는, 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 구비될 수도 있다.

[0042] 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 이동시키는 것이 가능한 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 컴퓨터 판독가능 매체 (16) 는 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 에 실시간으로 송신할 수 있게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 에 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 예컨대, 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 예컨대 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0043] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스로 출력될 수도 있다. 이와 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 로컬로 액세스되는 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다. 추가 예에서, 저장 디바이스는 소스 디바이스 (12) 에 의해 생성되는 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 저장 디바이스로부터의 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 에 송신하는 것이 가능한 임의의 타입의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예를 들어, 웹사이트용), FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 연결을 포함한 임의의 표준 데이터 연결을 통해, 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한, 무선 채널 (예를 들어, Wi-Fi 연결), 유선 연결 (예를 들어, DSL, 케이블 모뎀 등), 또는 이들 양쪽의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들의 조합일 수도 있다.

[0044] 본 개시물의 기법들은 반드시 무선 애플리케이션들 또는 설정들로 제한되지는 않는다. 이 기법들은 다양한 멀티미디어 애플리케이션들, 예컨대 공중경유 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 예컨대 HTTP 동적 적응 스트리밍 (dynamic adaptive streaming over HTTP; DASH), 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들 중 임의의 것의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화통신과 같은 애플리케이션들을 지원하기 위한 1-방향 또는 2-방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0045] 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 가상 참조 디코더 (HRD) 파라미터 시그널링을 개선시키기 위한 기법들을 적용하도록 구성될 수도 있다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 외부 카메라와 같은 외부 비디오 소스 (18) 로부터 비디오 데이터를 수신할 수도 있다. 이와 마찬가지로, 목적지 디바이스 (14) 는, 통합된 디스플레이 디바이스를 포함하기보다는, 외부 디스플레이 디바이스 와 인터페이싱할 수도 있다.

[0046] 위에서 언급된 바와 같이, 소스 디바이스 (12) 는 출력 인터페이스 (22) 를 포함하고 목적지 디바이스 (14) 는 입력 인터페이스 (28) 를 포함한다. 일부 예들에서, 출력 인터페이스 (22) 는 송신기를 표현하고 입력 인터페이스 (28) 는 수신기를 표현한다. 다른 예들에서, 출력 인터페이스 (22) 및 입력 인터페이스 (28) 는 트랜시버들 (즉, 데이터 신호들을 무선으로 송신하는 것과 수신하는 것 양쪽이 가능한 인터페이스들) 의 예들을 표현한다. 트랜시버들은 무선 신호들로 비디오 데이터를 전송 및 수신하도록 구성될 수도 있다. 예를 들어, 출력 인터페이스 (22) 는, 트랜시버로서 구현될 때, 인코딩된 비디오 데이터를 포함하는 데이터 신호 (예를 들어, 컴퓨터 판독가능 매체 (16)) 를 전송할 수도 있는 한편, 입력 인터페이스 (28) 는, 트랜시버로서 구현될 때, 인코딩된 비디오 데이터를 포함하는 데이터 신호 (예를 들어, 컴퓨터 판독가능 매체 (16)) 를 수신할 수도 있다. 위에서 논의된 바와 같이, 비디오 인코더 (20) 는 인코딩된 비디오 데이터를 출력 인터페이스 (22) 에 제공할 수도 있는 한편, 입력 인터페이스 (28) 는 인코딩된 비디오 데이터를 비디오 디코더 (30) 에 제공할 수도 있다.

[0047] 도 1 의 예시된 시스템 (10) 은 단순히 하나의 예일 뿐이다. 가상 참조 디코더 (HRD) 파라미터 시그널링을 개선시키기 위한 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로 본 개시물의 기법들이 비디오 인코딩 디바이스에 의해 수행되지만, 이 기법들은 또한 "코덱 (CODEC)" 이라고 통상적으로 지칭되는 비디오 인코더/디코더에 의해 수행될 수도 있다. 더욱이, 본 개시물의 기법들은 또한 비디오 프리프로세서 (video preprocessor) 에 의해 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 단순히, 소스 디바이스 (12) 가 목적지 디바이스 (14) 로의 송신을 위해 코딩된 비디오 데이터를 생성하는 이러한 코딩 디바이스들의 예들일 뿐이다. 일부 예들에서, 디바이스들 (12, 14) 은 디바이스들 (12, 14) 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 그에 따라, 시스템 (10) 은, 예를 들어, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오 전화통신을 위해, 비디오 디바이스들 (12, 14) 사이의 1-방향 또는 2-방향 비디오 송신을 지원할 수도 있다.

[0048] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 퍼드 인터페이스를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (18) 는 소스 비디오로서 컴퓨터 그래픽-기반 데이터, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합을 생성할 수도 있다. 일부 경우들에서, 비디오 소스 (18) 가 비디오 카메라인 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 설명되는 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에서, 캡처된, 미리 캡처된, 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 그 후에 출력 인터페이스 (22) 에 의해 컴퓨터 판독가능 매체 (16) 상으로 출력될 수도 있다.

[0049] 컴퓨터 판독가능 매체 (16) 는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 순시적 매체들, 또는 하드 디스크, 플래시 드라이브, 콤팩트 디스크, 디지털 비디오 디스크, 블루레이 디스크, 또는 다른 컴퓨터 판독가능 매체들과 같은 저장 매체들 (즉, 비일시적 저장 매체들) 을 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (미도시) 는 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신하고, 예를 들어, 네트워크 송신을 통해, 인코딩된 비디오 데이터를 목적지 디바이스 (14) 에 제공할 수도 있다. 이와 유사하게, 디스크 스템핑 서비스와 같은 매체 생산 서비스의 컴퓨팅 디바이스는 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신하고 인코딩된 비디오 데이터를 포함하는 디스크를 생성할 수도 있다. 그에 따라, 컴퓨터 판독가능 매체 (16) 는, 다양한 예들에서, 다양한 형태들의 하나 이상의 컴퓨터 판독가능 매체들을 포함하는 것으로 이해될 수도 있다.

[0050] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 판독가능 매체 (16) 로부터 정보를 수신한다. 컴퓨터 판독가능 매체 (16) 의 정보는, 블록들 및 다른 코딩된 유닛들, 예를 들어, GOP들의 특성을 및/또는 프로

세성을 설명하는 선택스 엘리먼트들을 포함하는, 비디오 디코더 (30)에 의해 또한 이용되는, 비디오 인코더 (20)에 의해 정의된 선택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 다양한 디스플레이 디바이스들, 예컨대, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스 중 임의의 것을 포함할 수도 있다.

[0051] 비디오 인코더 (20) 및 비디오 디코더 (30)는, ITU-T H.265라고도 또한 지칭되는, 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 코딩 표준에 따라 동작할 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30)는, MPEG-4, Part 10, 어드밴스드 비디오 코딩 (AVC)이라고 대안적으로 지칭되는 ITU-T H.264 표준과 같은 다른 사유 (proprietary) 또는 산업 표준들, 또는 이러한 표준들의 확장들에 따라 동작할 수도 있다. 그러나, 본 개시물의 기법들은 임의의 특정 코딩 표준으로 제한되지 않는다. 비디오 코딩 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263을 포함한다. 도 1에 도시되지 않았지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각 오디오 인코더 및 디코더와 통합될 수도 있으며, 공통 데이터 스트림 또는 별개의 데이터 스트림들에서 오디오와 비디오 양쪽의 인코딩을 핸들링하기에 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용 가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대, 사용자 데이터그램 프로토콜 (UDP)을 준수할 수도 있다.

[0052] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP)들, 주문형 집적 회로 (ASIC)들, 필드 프로그래밍가능 게이트 어레이 (FPGA)들, 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합과 같은 다양한 적합한 인코더 회로부 중 임의의 것으로서 구현될 수도 있다. 이 기법들이 부분적으로 소프트웨어로 구현될 때, 디바이스는 그 소프트웨어에 대한 명령들을 적합한 비일시적 컴퓨터 판독 가능 매체에 저장할 수도 있고 그 명령들을 하드웨어로 하나 이상의 프로세서들을 이용하여 실행함으로써 본 개시물의 기법들을 수행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이들 중 어느 하나는 조합된 인코더/디코더 (코덱)의 부분으로서 각각의 디바이스에 통합될 수도 있다.

[0053] 일반적으로, 비디오 프레임 또는 픽처는, 루마 및 크로마 샘플들 양쪽을 포함할 수도 있는, 최대 코딩 유닛 (LCU)들이라고도 또한 알려져 있는 트리블록들의 시퀀스로 분할될 수도 있다. 비트스트림 내의 선택스 데이터는, 픽셀들의 개수의 관점에서 최대 코딩 유닛인 LCU에 대한 사이즈를 정의할 수도 있다. 슬라이스는 코딩 순서로 다수의 연속적인 트리블록들을 포함한다. 비디오 프레임 또는 픽처는 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 큐드트리 데이터 구조에 따라 코딩 유닛 (CU)들로 분할될 수도 있다. 일반적으로, 큐드트리 데이터 구조는 CU당 하나의 노드를 포함하는데, 루트 노드는 트리블록에 대응한다. CU가 4개의 서브-CU들로 분할되는 경우, 그 CU에 대응하는 노드는 4개의 리프 (leaf) 노드들을 포함하고, 그 리프 노드들 각각은 서브-CU들 중 하나에 대응한다.

[0054] 큐드트리 데이터 구조의 각각의 노드는 대응하는 CU에 대한 선택스 데이터를 제공할 수도 있다. 예를 들어, 큐드트리에서의 노드는, 그 노드에 대응하는 CU가 서브-CU들로 분할되는지 여부를 나타내는 분할 플래그를 포함할 수도 있다. CU에 대한 선택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU가 서브-CU들로 분할되는지 여부에 의존할 수도 있다. CU가 추가로 분할되지 않는 경우, 그것은 리프-CU라고 지칭된다. 본 개시물에서, 리프-CU의 4개의 서브-CU들은 또한, 오리지널 리프-CU의 명시적인 분할이 없는 경우라도 리프-CU들이라고 지칭된다. 예를 들어, 16x16 사이즈의 CU가 추가로 분할되지 않는 경우, 4개의 8x8 서브-CU들은 또한, 16x16 CU가 전혀 분할되지 않았더라도 리프-CU들이라고 지칭된다.

[0055] CU는, CU가 사이즈 구별을 갖지 않는다는 것을 제외하고는, H.264 표준의 매크로블록과 유사한 목적을 갖는다. 예를 들어, 트리블록은 (서브-CU들이라고도 또한 지칭되는) 4개의 자식 노드들로 분할될 수도 있고, 각각의 자식 노드는 차례로 부모 노드일 수도 있고 다른 4개의 자식 노드들로 분할될 수도 있다. 큐드트리의 리프 노드라고 지칭되는 최종의 미분할된 자식 노드는 리프-CU라고도 또한 지칭되는 코딩 노드를 포함한다. 코딩된 비트스트림과 연관된 선택스 데이터는, 최대 CU 심도라고 지칭되는, 트리블록이 분할될 수도 있는 최대 횟수를 정의할 수도 있으며, 또한 코딩 노드들의 최소 사이즈를 정의할 수도 있다. 이에 따라, 비트스트림은 또한 최소 코딩 유닛 (SCU)을 정의할 수도 있다. 본 개시물은 HEVC의 맥락에서 CU, 예측 유닛 (PU), 또는 변환 유닛 (TU) 중 임의의 것, 또는 다른 표준들의 맥락에서 유사한 데이터 구조들 (예를 들어, H.264/AVC에서의 매크로블록들 및 그것의 서브-블록들)을 지칭하기 위해 용어 "블록"을 사용한다.

[0056] CU는 코딩 노드, 및 이 코딩 노드와 연관된 변환 유닛 (TU)들 및 예측 유닛 (PU)들을 포함한다. CU의

사이즈는 코딩 노드의 사이즈에 대응하고 일반적으로 형상이 정사각형이다. CU의 사이즈는, 8x8 픽셀들로부터, 예를 들어, 64x64 픽셀들 이상의 최대 사이즈를 갖는 트리블록의 사이즈까지의 범위에 있을 수도 있다.

각각의 CU는 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다. CU와 연관된 신택스 데이터는, 예를 들어, 하나 이상의 PU들로의 CU의 파티셔닝을 설명할 수도 있다. 파티셔닝 모드들은 CU가 스킵 또는 다이렉트 모드 인코딩되는지, 인트라-예측 모드 인코딩되는지, 또는 인터-예측 모드 인코딩되는지 간에 다를 수도 있다. PU들은 비-정사각형 형상으로 파티셔닝될 수도 있다. CU와 연관된 신택스 데이터는 또한, 예를 들어, 큐드트리에 따른 하나 이상의 TU들로의 CU의 파티셔닝을 설명할 수도 있다. TU는 형상이 정사각형 또는 비-정사각형 (예를 들어, 직사각형) 일 수 있다.

[0057] HEVC 표준은, 상이한 CU들에 대해 상이할 수도 있는 TU들에 따른 변환들을 허용한다. TU들은 통상적으로 파티셔닝된 LCU에 대해 정의된 주어진 CU 내의 PU들의 사이즈에 기초하여 사이징되지만, 이것이 항상 그러한 것은 아닐 수도 있다. TU들은 통상적으로 PU들과 동일한 사이즈이거나 또는 더 작다. 일부 예들에서, CU에 대응하는 잔차 샘플들은 "잔차 큐드 트리" (RQT)로서 알려진 큐드트리 구조를 이용하여 더 작은 유닛들로 세분될 수도 있다. RQT의 리프 노드들은 변환 유닛 (TU) 들이라고 지칭될 수도 있다. TU들과 연관된 픽셀 차이 값들은 변환 계수들을 생성하기 위해 변환될 수도 있고, 그 변환 계수들은 양자화될 수도 있다.

[0058] 리프-CU는 하나 이상의 PU들을 포함할 수도 있다. 일반적으로, PU는 대응하는 CU의 전부 또는 일부에 대응하는 공간적 영역을 표현하고, 그 PU에 대한 참조 샘플을 취출 및/또는 생성하기 위한 데이터를 포함할 수도 있다. 더욱이, PU는 예측에 관련된 데이터를 포함한다. 예를 들어, PU가 인트라-모드 인코딩될 때, PU에 대한 데이터는 PU에 대응하는 TU에 대한 인트라-예측 모드를 설명하는 데이터를 포함할 수도 있는 잔차 큐드트리 (RQT)에 포함될 수도 있다. RQT는 또한 변환 트리라고도 지칭될 수도 있다. 일부 예들에서, 인트라-예측 모드는, RQT 대신에, 리프-CU 신택스에서 시그널링될 수도 있다. 다른 예로서, PU가 인터-모드 인코딩될 때, PU는, PU에 대한, 하나 이상의 모션 벡터들과 같은 모션 정보를 정의하는 데이터를 포함할 수도 있다. PU에 대한 모션 벡터를 정의하는 데이터는, 예를 들어, 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터에 대한 해상도 (예를 들어, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 참조 픽처, 및/또는 모션 벡터에 대한 참조 픽처 리스트 (예를 들어, 리스트 0, 리스트 1, 또는 리스트 C)를 설명할 수도 있다.

[0059] 하나 이상의 PU들을 갖는 리프-CU는 또한 하나 이상의 TU들을 포함할 수도 있다. 변환 유닛들은, 위에서 논의된 바와 같이, (TU 큐드트리 구조라고도 또한 지칭되는) RQT를 이용하여 특정될 수도 있다. 예를 들어, 분할 플래그가 리프-CU가 4개의 변환 유닛들로 분할되는지 여부를 나타낼 수도 있다. 그 후에, 각각의 변환 유닛은 추가의 서브-TU들로 추가로 분할될 수도 있다. TU가 추가로 분할되지 않는 경우, 그것은 리프-TU라고 지칭될 수도 있다. 일반적으로, 인트라 코딩의 경우, 리프-CU에 속하는 모든 리프-TU들은 동일한 인트라 예측 모드를 공유한다. 즉, 동일한 인트라-예측 모드는 일반적으로 리프-CU의 모든 TU들에 대한 예측된 값들을 계산하기 위해 적용된다. 인트라 코딩의 경우, 비디오 인코더는, TU에 대응하는 CU의 부분과 오리지널 블록 사이의 차이로서, 인트라 예측 모드를 이용하여 각각의 리프-TU에 대한 잔차 값을 계산할 수도 있다. TU는 PU의 사이즈로 반드시 제한되지는 않는다. 따라서, TU들은 PU 보다 더 크거나 또는 더 작을 수도 있다. 인트라 코딩의 경우, PU는 동일한 CU에 대한 대응하는 리프-TU와 병치될 수도 있다. 일부 예들에서, 리프-TU의 최대 사이즈는 대응하는 리프-CU의 사이즈에 대응할 수도 있다.

[0060] 더욱이, 리프-CU들의 TU들은 또한, 위에서 언급된 바와 같이 잔차 큐드트리 (RQT)들 또는 변환 트리들이라고 지칭되는 각각의 큐드트리 데이터 구조들과 연관될 수도 있다. 즉, 리프-CU는 그 리프-CU가 어떻게 TU들로 파티셔닝되는지를 나타내는 큐드트리를 포함할 수도 있다. TU 큐드트리의 루트 노드는 일반적으로 리프-CU에 대응하는 한편, CU 큐드트리의 루트 노드는 일반적으로 트리블록 (또는 LCU)에 대응한다. 분할되지 않은 RQT의 TU들은 리프-TU들이라고 지칭된다. 일반적으로, 본 개시물은, 달리 언급되지 않는 한, 리프-CU 및 리프-TU를 각각 지칭하기 위해 용어들 CU 및 TU를 사용한다.

[0061] 비디오 시퀀스는 통상적으로 비디오 프레임들 또는 픽처들의 시리즈를 포함한다. 픽처들의 그룹 (group of pictures; GOP)은 일반적으로 비디오 픽처들 중 하나 이상의 비디오 픽처의 시리즈를 포함한다. GOP는 GOP의 헤더, 픽처들 중 하나 이상의 픽처의 헤더, 또는 다른 곳에 신택스 데이터를 포함할 수도 있고, 이 신택스 데이터는 GOP에 포함된 다수의 픽처들을 설명한다. 픽처의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 설명하는 슬라이스 신택스 데이터를 포함할 수도 있다. 비디오 인코더 (20)는 통상적으로 비디오 데이터를 인코딩하기 위해 개개의 비디오 슬라이스를 내의 비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정된 또는 변동되는 사이즈들을 가질 수

도 있고, 특정된 코딩 표준에 따라 사이즈가 다를 수도 있다.

[0062] 일 예로서, 다양한 사이즈들의 PU들에 대해 예측이 수행될 수도 있다. 특정 CU의 사이즈가 $2Nx2N$ 이라고 가정하면, 인트라-예측은 $2Nx2N$ 또는 NxN 의 PU 사이즈들에 대해 수행될 수도 있고, 인터-예측은 $2Nx2N$, $2NxN$, $Nx2N$, 또는 NxN 의 대칭적인 PU 사이즈들에 대해 수행될 수도 있다. 인터-예측을 위한 비대칭적인 파티셔닝은 또한 $2NxN_U$, $2NxN_D$, $nLx2N$, 및 $nRx2N$ 의 PU 사이즈들에 대해 수행될 수도 있다. 비대칭적인 파티셔닝에서는, CU의 하나의 방향이 파티셔닝되지 않는 한편, 다른 방향은 25% 및 75%로 파티셔닝된다. 25% 파티션에 대응하는 CU의 일부는 "n" 다음에 "상 (Up)", "하 (Down)", "좌 (Left)", 또는 "우 (Right)"의 표시로 나타낸다. 따라서, 예를 들어, " $2NxN_U$ "는 상부의 $2Nx0.5N$ PU 및 하부의 $2Nx1.5N$ PU로 수평으로 파티셔닝되는 $2Nx2N$ CU를 지칭한다.

[0063] 본 개시물에서, " NxN " 및 " $N \times N$ "은 수직 및 수평 치수들의 관점에서 비디오 블록의 픽셀 치수들, 예를 들어, 16×16 픽셀들 또는 16×16 픽셀들을 지칭하기 위해 상호교환가능하게 사용될 수도 있다. 일반적으로, 16×16 블록은 수직 방향에서의 16 개의 픽셀들 ($y = 16$) 및 수평 방향에서의 16 개의 픽셀들 ($x = 16$)을 갖는다. 이와 마찬가지로, NxN 블록은 일반적으로 수직 방향에서의 N 개의 픽셀들 및 수평 방향에서의 N 개의 픽셀들을 가지며, 여기서 N 은 음이 아닌 정수 값을 표현한다. 블록 내의 픽셀들은 로우들 및 컬럼들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향에서와 동일한 개수의 픽셀들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, 블록들은 $N \times M$ 픽셀들을 포함할 수도 있고, 여기서 M 은 반드시 N 과 동일하지는 않다.

[0064] CU의 PU들을 이용한 인트라-예측 또는 인터-예측 코딩에 후속하여, 비디오 인코더 (20)는 CU의 TU들에 대한 잔차 데이터를 계산할 수도 있다. PU들은 공간 도메인 (또한 픽셀 도메인이라고도 지칭됨)에서의 예측 픽셀 데이터를 생성하는 방법 또는 모드를 설명하는 신팩스 데이터를 포함할 수도 있고, TU들은 잔차 비디오 데이터에 대한 변환, 예를 들어, 이산 코사인 변환 (discrete cosine transform; DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환의 적용에 후속하는 변환 도메인에서의 계수들을 포함할 수도 있다. 잔차 데이터는 인코딩되지 않은 픽처의 픽셀들과 PU들에 대응하는 예측 값을 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20)는 CU에 대한 잔차 데이터를 표현하는 양자화된 변환 계수들을 포함하도록 TU들을 형성할 수도 있다. 즉, 비디오 인코더 (20)는 (잔차 블록의 형태로) 잔차 데이터를 계산하고, 잔차 블록을 변환하여 변환 계수들의 블록을 생성하며, 그 후에 변환 계수들을 양자화하여 양자화된 변환 계수들을 형성할 수도 있다. 비디오 인코더 (20)는 다른 신팩스 정보 (예를 들어, TU에 대한 분할 정보) 뿐만 아니라, 양자화된 변환 계수들을 포함하는 TU를 형성할 수도 있다.

[0065] 위에서 언급된 바와 같이, 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (20)는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로, 계수들을 표현하는데 이용된 데이터의 양을 가능하다면 감소시키기 위해 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, n -비트 값은 양자화 동안에 m -비트 값으로 버림 (round down) 될 수도 있고, 여기서 n 은 m 보다 더 크다.

[0066] 양자화에 후속하여, 비디오 인코더 (20)는 변환 계수들을 스캐닝하여, 양자화된 변환 계수들을 포함하는 2 차원 매트릭스로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 어레이의 전방에 보다 높은 에너지 (그리고 그에 따라 보다 낮은 주파수) 계수들을 배치시키고 어레이의 후방에 보다 낮은 에너지 (그리고 그에 따라 보다 높은 주파수) 계수들을 배치시키도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해 양자화된 변환 계수들을 스캐닝하기 위한 미리 정의된 스캔 순서를 활용할 수도 있다. 다른 예들에서, 비디오 인코더 (20)는 적응 스캔 (adaptive scan)을 수행할 수도 있다. 1 차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캐닝한 후에, 비디오 인코더 (20)는, 예를 들어, 콘텍스트 적응 가변 길이 코딩 (ontext-adaptive variable length coding; CAVLC), 콘텍스트 적응 이진 산술 코딩 (context-adaptive binary arithmetic coding; CABAC), 신팩스 기반 콘텍스트 적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 간격 파티셔닝 엔트로피 (Probability Interval Partitioning Entropy; PIPE) 코딩 또는 다른 엔트로피 인코딩 방법론에 따라 1 차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20)는 또한 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30)에 의한 이용을 위해 인코딩된 비디오 데이터와 연관된 신팩스 엘리먼트들을 엔트로피 인코딩 할 수도 있다.

[0067] CABAC를 수행하기 위해, 비디오 인코더 (20)는 콘텍스트 모델 내의 콘텍스트를 송신될 심볼에 할당할 수도 있다. 콘텍스트는, 예를 들어, 심볼의 이웃하는 값들이 논-제로 (non-zero)인지 아닌지의 여부에 관련될 수

도 있다. CAVLC 를 수행하기 위해, 비디오 인코더 (20) 는 송신될 심볼에 대한 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드 (codeword) 들은, 상대적으로 더 짧은 코드들이 더 가능성 있는 심볼들에 대응하는 한편, 더 긴 코드들이 덜 가능성 있는 심볼들에 대응하도록 구성될 수도 있다. 이러한 방법으로, VLC 의 이용은, 예를 들어, 송신될 각각의 심볼에 대한 동일-길이 코드워드들을 이용하는 것에 비해 비트 절약들을 달성할 수도 있다. 확률 결정은 심볼에 할당된 콘텍스트에 기초할 수도 있다.

[0068] 일반적으로, 비디오 디코더 (30) 는 인코딩된 데이터를 디코딩하기 위해 비디오 인코더 (20) 에 의해 수행된 것과 실질적으로 유사하지만 상반된 프로세스를 수행한다. 예를 들어, 비디오 디코더 (30) 는 수신된 TU 의 변환 계수들을 역 양자화 및 역 변환하여 잔차 블록을 재생성한다. 비디오 디코더 (30) 는 시그널링된 예측 모드 (인트라- 또는 인터-예측) 를 이용하여 예측된 블록을 형성한다. 그 후에 비디오 디코더 (30) 는 (픽셀 단위로) 예측된 블록과 잔차 블록을 결합하여 오리지널 블록을 재생성한다. 블록 경계들을 따라 시각적 아티팩트 (visual artifact) 들을 감소시키기 위해 디블록킹 프로세스를 수행하는 것과 같은 부가적인 프로세싱이 수행될 수도 있다. 게다가, 비디오 디코더 (30) 는 비디오 인코더 (20) 의 CABAC 인코딩 프로세스와 실질적으로 유사하지만 상반된 방식으로 CABAC 를 이용하여 신박스 엘리먼트들을 디코딩할 수도 있다.

[0069] 본 개시물의 하나의 예에 따르면, 비디오 코더 (예컨대 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는, 가상 참조 디코더 (HRD) 파라미터들을 포함하는, 비디오 파라미터 세트 (VPS) 에서의 비디오 유용성 정보 (video usability information; VUI) 를 코딩하도록 구성될 수도 있다. 아래의 테이블 1 은 본 개시물의 소정의 기법들에 따른 HRD 파라미터들에 대한 예시적인 VPS VUI 바이트 시퀀스 페이로드 (byte sequence payload; BSP) 를 설명한다. 테이블 1 에서, 이탈릭체의 텍스트는 JCTVC-R0010v2 에 대한 변형 2 첨부물에서 설명되는 데이터 구조에 관련된 부가들을 나타내는 한편, 팔호들 및 "제거됨:" (예를 들어, [제거됨: "예시적인 제거된 텍스트"]) 을 사용하여 식별된 텍스트는 JCTVC-R0010v2 에 대한 변형 2 첨부물에서 설명되는 데이터 구조로부터의 제거들을 표현한다.

[0070]

테이블 1

| | |
|--|-------|
| vps_vui_bsp_hrd_params [제거됨: "parameters"]() | 디스크립터 |
| vps_num_add_hrd_params [제거됨: "parameters"] | ue(v) |
| for(i = vps_num_hrd_parameters; i < vps_num_hrd_parameters + vps_num_add_hrd_params [제거됨: "parameters"]; i++) { | |
| if(i > 0) { | |
| cprms_add_present_flag[i] | u(1) |
| num_sub_layer_hrd_minus1[i] | ue(v) |
| } | |
| hrd_parameters(cprms_add_present_flag[i], num [제거됨: "vps_max"]_sub_hrd [제거됨: "layers"]_minus1[i]) | |
| } | |
| for(h = 1; h < NumOutputLayerSets; h++) | |
| for(i = 0; i < NumPartitioningSchemes[h]; i++) { | |
| bsp_hrd_params [제거됨: "parameters"]_present_flag[h][i] | u(1) |
| if(bsp_hrd_params [제거됨: "parameters"]_present_flag[h][i]) | |
| { | |
| for(t = 0; t <= | |
| MaxSubLayersInLayerSetMinus1[OlsIdxToLsIdx[i]]; t++) { | |
| num_bsp_schedules [제거됨: "sched_combinations"]_minus1[h][i][t] | ue(v) |
| for(j = 0; j <= num_bsp_schedules [제거됨: "sched_combinations"]_minus1[h][i][t]; j++) | |
| for(k = 0; k <= num_partitions_in_scheme_minus1[h][i]; | |
| k++) { | |
| bsp[제거됨: "_comb"]_hrd_idx[h][i][t][j][k] | u(v) |
| bsp[제거됨: "_comb"]_sched_idx[h][i][t][j][k] | ue(v) |
| } | |
| }[제거됨: "}] | |
| } | |

[0071]

| | |
|---|--|
| } | |
| } | |

[0072]

[0073]

테이블 1 의 신택스 엘리먼트들에 대한 예시적인 시맨틱스 (semantics) 가 아래에 설명된다. 이와 달리 아래에 논의되지 않은 미변화된 신택스 엘리먼트들은 JCTVC-R0010v2 에 대한 변형 2 첨부물에서 설명된 바와 같이 시맨틱스를 유지할 수도 있다. 다시, 이탈릭체의 텍스트는 부가들을 표현하는 한편, [제거됨: ""] 은 삭제들을 표현한다.

[0074]

vps_num_add_hrd_params [제거됨: "parameters"] 는 VPS 에 존재하는 부가적인 hrd_parameters() 신택스 구조들의 개수를 특정한다. vps_num_add_hrd_params [제거됨: "parameters"] 의 값은 0 내지 1024 - vps_num_hrd_parameters 의 범위에 있어야 한다.

[0075]

1 과 동일한 cprms_add_present_flag[i] 는 모든 서브-계층들에 대해 공통인 HRD 파라미터들이 i 번째

`hrd_parameters()` 신택스 구조에 존재한다는 것을 특정한다. 0 과 동일한 `cprms_add_present_flag[i]` 는 모든 서브-계층들에 대해 공통인 HRD 파라미터들이 i 번째 `hrd_parameters()` 신택스 구조에 존재하지 않으면 ($i - 1$) 번째 `hrd_parameters()` 신택스 구조와 동일하게 도출된다는 것을 특정한다.

[0076] `num_sub_layer_hrd_minus1[i]` 플러스 1 은 i 번째 `hrd_parameters()` 신택스 구조에서 `fixed_pic_rate_general_flag[]` 신택스 엘리먼트들의 개수를 특정한다. `num_sub_layer_hrd_minus1[i]` 의 값은 0 내지 `vps_max_sub_layers_minus1` 의 범위에 있어야 한다.

[0077] 1 과 동일한 `bsp_hrd_params` [제거됨: "parameters"]_present_flag[h][i] 는 HRD 파라미터들이 h 번째 OLS 의 i 번째 파티셔닝 스킴들의 모든 비트스트림 파티션들에 대해 존재한다는 것을 특정한다. 0 과 동일한 `bsp_hrd_params` [제거됨: "parameters"]_present_flag[h][i] 는 HRD 파라미터들이 h 번째 OLS 의 i 번째 파티셔닝 스킴의 어떠한 비트스트림 파티션에 대해서도 존재하지 않는다는 것을 특정한다.

[0078] `num_bsp_schedules` [제거됨: "sched_combinations"]_minus1[h][i][t] 플러스 1 은 `HighestTid` 가 t 와 동일할 때 h 번째 OLS 의 i 번째 파티셔닝 스킴의 비트스트림 파티션들에 대해 특정된 전달 스케줄들 [제거됨: "및 `hrd_parameters()`"] [제거됨: "의 조합들"] 의 개수를 특정한다. `num_bsp_schedules_minus1[h][i][t]` 의 값은 0 내지 31 의 범위에 있어야 한다.

[0079] 변수 [제거됨: "SchedCombCnt"] `BspSchedCnt` [h][i][t] 는 `num_bsp_schedules` [제거됨: "sched_combinations"]_minus1[h][i][t] + 1 과 동일하게 설정된다.

[0080] `bsp` [제거됨: "_comb"]_hrd_idx[h][i][t][j][k] 는 `HighestTid` 가 t 와 동일할 때 h 번째 OLS 의 i 번째 파티셔닝 스킴의 k 번째 비트스트림 파티션에 대해 특정된 전달 스케줄 [제거됨: "및 `hrd_parameters()`"] 의 j 번째 [제거됨: "조합"] 에 대한 VPS 에서의 `hrd_parameters()` 신택스 구조의 인덱스를 특정한다. `bsp` [제거됨: "_comb"]_hrd_idx[h][i][t][j][k] 신택스 엘리먼트의 길이는 `Ceil(Log2(vps_num_hrd_parameters + vps_num_add_hrd_params [제거됨: "parameters"]))` 비트들이다. `bsp` [제거됨: "_comb"]_hrd_idx[h][i][t][j][k] 의 값은 0 내지 `vps_num_hrd_parameters + vps_num_add_hrd_params [제거됨: "parameters"] - 1` 의 범위에 있어야 한다.

[0081] `bsp` [제거됨: "_comb"]_sched_idx[h][i][t][j][k] 는 `HighestTid` 가 t 와 동일할 때 h 번째 OLS 의 i 번째 파티셔닝 스킴의 k 번째 비트스트림 파티션에 대해 특정된 전달 스케줄 [제거됨: "및 `hrd_parameters()`"] 의 j 번째 [제거됨: "조합"] [제거됨: "에서"] 로서 이용되는, 인덱스 `bsp` [제거됨: "_comb"]_hrd_idx[h][i][t][j][k] 를 갖는 `hrd_parameters(t)` 신택스 구조의 `sub_layer_hrd_parameters(t)` [제거됨: "hrd_parameters()"] 신택스 구조 내의 그 [제거됨: "한"] 전달 스케줄의 인덱스를 특정한다. `bsp` [제거됨: "_comb"]_sched_idx[h][i][t][j][k] 의 값은 0 내지 `cpb_cnt_minus1[t]` [제거됨: "HighestTid"]] 의 범위에 있어야 하고, 여기서 `cpb_cnt_minus1[t]` [제거됨: "HighestTid"]] 는 인덱스 `bsp` [제거됨: "_comb"]_hrd_idx[h][i][t][j][k] 에 대응하는 `hrd_parameters()` 신택스 구조로부터의 `sub_layer_hrd_parameters(t)` [제거됨: "HighestTid"]] 신택스 구조에서 발견된다.

[0082] HEVC 에 따르면, 다른 종래의 HRD 파라미터들은 또한 HRD 파라미터들 신택스 구조에서 시그널링될 수도 있지만, 반드시 위의 테이블 1 에 나타나 있지는 않다. 예를 들어, HRD 파라미터들은 `fixed_pic_rate_within_cvs_flag[i]` 를 포함할 수도 있는데, 이에 대해 HEVC 는 다음과 같이 시맨틱스를 정의한다:

[0083] 1 과 동일한 `fixed_pic_rate_within_cvs_flag[i]` 는, `HighestTid` 가 i 와 동일할 때, 출력 순서에서의 연속적인 픽쳐들의 HRD 출력 시간들 사이의 시간적 거리가 아래에 특정된 바와 같이 제약됨을 나타낸다. 0 과 동일한 `fixed_pic_rate_within_cvs_flag[i]` 는 이 제약이 적용되지 않을 수도 있음을 나타낸다.

[0084] HRD 파라미터들은 또한 `elemental_duration_in_tc_minus1[i]` 신택스 엘리먼트를 포함할 수도 있는데, 이에 대해 HEVC 는 다음과 같이 시맨틱스를 정의한다:

[0085] `elemental_duration_in_tc_minus1[i]` 플러스 1 (존재한다면) 은, `HighestTid` 가 i 와 동일할 때, 아래에 특정된 바와 같이 출력 순서에서의 연속적인 픽쳐들의 HRD 출력 시간들을 특정하는 기본 유닛들 사이의, 클록 틱 (clock tick) 들에 있어서의, 시간적 거리를 특정한다. `elemental_duration_in_tc_minus1[i]` 의 값은 0 내지 2047 의 범위에 있어야 한다.

[0086] HRD 파라미터들은 또한 `low_delay_hrd_flag[i]` 신택스 엘리먼트를 포함할 수도 있는데, 이에 대해 HEVC 는 다음

과 같이 시맨틱스를 정의한다:

[0087] **low_delay_hrd_flag[i]** 는, 부록 C 에 특정된 바와 같이, HighestTid 가 i 와 동일할 때, HRD 동작 모드를 특정한다. 존재하지 않을 때, low_delay_hrd_flag[i] 의 값은 0 과 동일한 것으로 추론된다.

[0088] 테이블 1 의 예에서, num_sub_layer_hrd_minus1[i] 는 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 십액스 엘리먼트의 예를 표현한다. HRD 파라미터들이 코딩되는 서브-계층들의 개수는 비트스트림의 비디오 파라미터 세트 (VPS) 에 의해 나타낸 서브-계층들의 최대 개수 이하일 수도 있다. 따라서, 비디오 코더는 십액스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩한 후에 HRD 파라미터들을 이용하여 비트스트림을 프로세싱할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HRD 파라미터들에 따라 디코딩된 픽처 베퍼로부터 픽처들을 제거할 수도 있다. 더욱이, 목적지 디바이스 (14) 는 디스플레이 디바이스 (32) 를 이용하여 디코딩된 픽처 베퍼로부터 제거된 픽처들을 디스플레이할 수도 있다.

[0089] 또한 테이블 1 의 예에 나타낸 바와 같이, 비디오 코더는 다중-계층 비트스트림에 포함된 각각의 서브-계층에 대한 HRD 파라미터들의 세트를 코딩할 수도 있다. 테이블 1 에서, "for(t = 0; t <= MaxSubLayersInLayerSetMinus1[0]sIdxToLsIdx[i]]; t++)" 에 의해 나타낸 루프는 특정 계층 세트에 포함된 서브-계층들의 개수에 걸친 루프를 표현하고, 이 루프는 이용가능한 출력 계층 세트들 각각에 대해 수행된다. 이 루프 내에서, HRD 파라미터들에 대한 인덱스들 (bsp_hrd_idx) 이 시그널링된다. 따라서, 이것은 비트스트림의 서브-계층들의 개수와 동일한 개수의 HRD 파라미터들을 코딩하기 위한 하나의 예시적인 기법이다. 특히, 각각의 비트스트림 파티션 (즉, 각각의 출력 계층 세트) 에 대한 서브-계층들의 개수와 HRD 파라미터들 사이의 일 대 일 맵핑이 존재한다.

[0090] 계다가, 비디오 코더 (예컨대 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 아래의 테이블 2 의 예시적인 데이터 구조에 따라 비트스트림 파티션 초기 도달 시간을 나타내는 정보를 코딩 (각각 인코딩 또는 디코딩) 하도록 구성될 수도 있다. 테이블 2 는, JCTVC-R0010v2 의 변형 2 첨부물에 대해 변화된 비트스트림 파티션 초기 도달 시간 SEI 메시지의 예를 표현한다. 다시, 이텔릭체의 텍스트는 부가들을 표현하고 [제거됨: ""] 은 삭제들을 표현한다.

테이블 2

| | |
|---|-------|
| bsp_initial_arrival_time(payloadSize) { | 디스크립터 |
| psIdx = sei_partitioning_scheme_idx | |
| if(NalHrdBpPresentFlag) | |
| for(i = 0; i < | |
| [제거됨: "SchedCombCnt"] BspSchedCnt[sei_ols_idx][psIdx [제거됨: "sei_partitioning_scheme_idx"]][maxTemporalId[0]]; i++) | |
| nal_initial_arrival_delay[i] | u(v) |
| if(VclHrdBpPresentFlag) [제거됨: "else"] | |
| for(i = 0; i < [제거됨: "SchedCombCnt"] BspSchedCnt[sei_ols_idx][psIdx [제거됨: "sei_partitioning_scheme_idx"]][maxTemporalId[0]]; i++) | |
| vcl_initial_arrival_delay[i] | u(v) |
| } | |

[0092] 테이블 2 의 십액스 엘리먼트들에 대한 예시적인 시맨틱스가 아래에 설명된다.

이와 달리 아래에 논의되지 않은 미변화된 십액스 엘리먼트들은 JCTVC-R0010v2 에 대한 변형 2 첨부물에서 설명된 바와 같이 시맨틱스를 유지할 수도 있다. 다시, 이텔릭체의 텍스트는 부가들을 표현하는 한편, [제거됨: ""] 은 삭제들을 표현한다.

[0093] 비트스트림 파티션 초기 도달 시간 SEI 메시지는 비트스트림-파티션-특정 CPB 동작에서 이용될 초기 도달 시간 들을 특정한다.

- [0095] 존재한다면, 이 SEI 메시지는 스케일러블 네스팅 (nesting) SEI 메시지에 포함되는 비트스트림 파티션 네스팅 SEI 메시지 내에 포함되어야 하고, 동일한 비트스트림 파티션 네스팅 SEI 메시지는 또한 베퍼링 주기 SEI 메시지를 포함해야 한다.
- [0096] 다음은 비트스트림 파티션 네스팅 SEI 메시지 선택스 및 시멘틱스에 대해 적용된다:
- [0097] - 선택스 엘리먼트 `initial_cpb_removal_delay_length_minus1` 및 변수들 `NalHrdBpPresentFlag` 및 `VclHrdBpPresentFlag`는, 비트스트림 파티션 네스팅 SEI 메시지가 적용되는 동작 포인트들 중 적어도 하나에 적용 가능한 `hrd_parameters()` 선택스 구조에서 발견된 선택스 엘리먼트들에서 발견되거나 또는 그 선택스 엘리먼트들로부터 도출된다.
- [0098] [제거됨]:
- [0099] 0 내지 `SchedCombCnt[sei_ols_idx][sei_partitioning_scheme_idx]`의 범위에서의 i에 대한 `hrdParamIdx[i]` 가 `bsp_comb_hrd_idx[olsIdx][partitioningSchemeIdx][i][bspIdx]`의 값과 동일하다고 하고, 여기서 `olsIdx`, `partitioningSchemeIdx`, 및 `bspIdx`는 이 비트스트림 파티션 초기 도달 시간 SEI 메시지를 포함하는 비트스트림 파티션 네스팅 SEI 메시지의 `sei_ols_idx`, `sei_partitioning_scheme_idx`, 및 `bsp_idx` 각각과 동일하다. `initialCpbRemovalDelayLength[i]` 가 `initial_cpb_removal_delay_length_minus1 + 1`과 동일하다고 하고, 여기서 `initial_cpb_removal_delay_length_minus1` 은 액티브 VPS에서의 `hrdParamIdx[i]` 번째 `hrd_parameters()` 선택스 구조에서 발견된다.]
- [0100] `nal_initial_arrival_delay[i]` 는, NAL HRD 파라미터들이 이용되고 있을 때, 이 SEI 메시지가 적용되는 비트스트림 파티션의 i 번째 전달 스케줄 [제거됨: "조합"]에 대한 초기 도달 시간을 특정한다. `nal_initial_arrival_delay[i]` 선택스 엘리먼트의 비트들에 있어서의 길이는 `initial_cpb_removal_delay_length_minus1 + 1` [제거됨: "initialCpbRemovalDelayLength[i]"] 와 동일하다.
- [0101] `vcl_initial_arrival_delay[i]` 는, VCL HRD 파라미터들이 이용되고 있을 때, 이 SEI 메시지가 적용되는 비트스트림 파티션의 i 번째 전달 스케줄 [제거됨: "조합"]에 대한 초기 도달 시간을 특정한다. `vcl_initial_arrival_delay[i]` 선택스 엘리먼트의 비트들에 있어서의 길이는 `initial_cpb_removal_delay_length_minus1 + 1` [제거됨: "initialCpbRemovalDelayLength[i]"] 와 동일하다.
- [0102] 이를 기법들의 부가적인 예들이, 예를 들어, 테이블들 3 및 4에 대해 아래에 논의된다. 테이블 3은 테이블 1의 예에 대한 대안을 표현하는 한편, 테이블 4는 테이블 4의 예에 대한 대안을 표현한다. 다시, CTVC-R0010v2의 변형 2 첨부물에 대해 차이들이 나타나 있고, 여기서 이탈릭체의 텍스트는 부가들을 표현하고 [제거됨: ""]은 삭제들을 표현한다.

[0103]

테이블 3

| | |
|---|----------------|
| vps_vui_bsp_hrd_params [제거됨: "parameters"] () { | 디스크립터 |
| vps_num_add_hrd_params [제거됨: "parameters"] | ue(v) |
| for(i = vps_num_hrd_parameters; i < vps_num_hrd_parameters + | |
| vps_num_add_hrd_params [제거됨: "parameters"]; i++) { { | |
| if(i > 0) { | |
| cprms_add_present_flag[i] | u(l) |
| num_sub_layer_hrd_minus1[i] | ue(v) |
| } | |
| hrd_parameters(cprms_add_present_flag[i], num [제거됨: "vps_max"] sub_hrd [제거됨: "layers"] minus1[i]) | |
| } | |
| for(h = 1; h < NumOutputLayerSets; h++) | |
| for(i = 0; i < NumPartitioningSchemes[h]; i++) { | |
| bsp_hrd_params [제거됨: "parameters"]_present_flag[h][i] | u(l) |
| if(bsp_hrd_params [제거됨: "parameters"]_present_flag[h][i] | |
|)[제거됨: {" | |
| [제거됨: "num_bsp_sched_combinations_minus1[h][i]"] | [제거됨: "ue(v)"] |
| [제거됨: "for(j = 0; j <= num_bsp_schedules_combinations_minus1[h][i][t]; j++)"] | |
| for(k = 0; k <= num_partitions_in_scheme_minus1[h][i]; | |
| k++)[제거됨: {""] | |
| bsp[제거됨: "_comb"]_hrd_idx[h][i][제거됨: "[j]"][k] | u(v) |
| [제거됨: "bsp_comb_sched_idx[h][i][j][k]"] | [제거됨: "ue(v)"] |
| [제거됨: "}"] | |
| } | |
| } | |

[0104]

[0105]

테이블 3 의 신택스 엘리먼트들에 대한 예시적인 시맨틱스가 아래에 설명된다. 이와 달리 아래에 논의되지 않은 미변화된 신택스 엘리먼트들은 JCTVC-R0010v2 에 대한 변형 2 첨부물에서 설명된 바와 같이 시맨틱스를 유지할 수도 있다. 다시, 이탈릭체의 텍스트는 부가들을 표현하는 한편, [제거됨: ""] 은 삭제들을 표현한다.

[0106]

vps_num_add_hrd_params [제거됨: "parameters"] 는 VPS 에 존재하는 부가적인 hrd_parameters() 신택스 구조들의 개수를 특정한다. vps_num_add_hrd_params [제거됨: "parameters"] 의 값은 0 내지 1024 - vps_num_hrd_parameters 의 범위에 있어야 한다.

[0107]

1 과 동일한 cprms_add_present_flag[i] 는 모든 서브-계층들에 대해 공통인 HRD 파라미터들이 i 번째 hrd_parameters() 신택스 구조에 존재한다는 것을 특정한다. 0 과 동일한 cprms_add_present_flag[i] 는 모든 서브-계층들에 대해 공통인 HRD 파라미터들이 i 번째 hrd_parameters() 신택스 구조에 존재하지 않으며 (i - 1) 번째 hrd_parameters() 신택스 구조와 동일하게 도출된다는 것을 특정한다.

[0108]

num_sub_layer_hrd_minus1[i] 플러스 1 은 i 번째 hrd_parameters() 신택스 구조에서 fixed_pic_rate_general_flag[] 신택스 엘리먼트들의 개수를 특정한다. num_sub_layer_hrd_minus1[i] 의 값은 0 내지 vps_max_sub_layers_minus1 의 범위에 있어야 한다.

[0109]

1 과 동일한 bsp_hrd_params [제거됨: "parameters"]_present_flag[h][i] 는 HRD 파라미터들이 h 번째 OLS 의 i 번째 파티셔닝 스킴들의 모든 비트스트림 파티션들에 대해 존재한다는 것을 특정한다. 0 과 동일한 bsp_hrd_params [제거됨: "parameters"]_present_flag[h][i] 는 HRD 파라미터들이 h 번째 OLS 의 i 번째 파티셔닝 스킴의 어떠한 비트스트림 파티션에 대해서도 존재하지 않는다는 것을 특정한다.

[0110]

[제거됨: "num_bsp_sched_combinations_minus1[h][i]" 플러스 1 은 h 번째 OLS 의 i 번째 파티셔닝 스킴의 비트스트림 파티션들에 대해 특정된 전달 스케줄들 및 hrd_parameters() 의 조합들의 개수를 특정한다.

[편집됨 (MH): 이 신택스 엘리먼트에 대한 허용된 값 범위를 부가한다.]"]

[0111] [제거됨: "변수 SchedCombCnt[h][i] 는 num_bsp_sched_combinations_minus1[h][i] + 1 과 동일하게 설정된다."]

[0112] bsp [제거됨: "_comb"]_hrd_idx[h][i][제거됨: "[j]"][k] 는 h 번째 OLS [제거됨: "") 의 i 번째 파티셔닝 스킴의 k 번째 비트스트림 파티션 [제거됨: "에 대해 특정된 전달 스케줄 및 hrd_parameters() 의 j 번째 조합"]에 대한 VPS에서의 hrd_parameters() 신택스 구조의 인덱스를 특정한다. bsp [제거됨: "_comb"]_hrd_idx[h][i][제거됨: "[j]"][k] 신택스 엘리먼트의 길이는 Ceil(Log2(vps_num_hrd_parameters + vps_num_add_hrd_params [제거됨: "parameters"])) 비트들이다. bsp [제거됨: "_comb"]_hrd_idx[h][i][제거됨: "[j]"][k]의 값은 0 내지 vps_num_hrd_parameters + vps_num_add_hrd_params [제거됨: "parameters"] - 1의 범위에 있어야 한다.

[0113] [제거됨: "bsp_comb_sched_idx[h][i][j][k] 는 h 번째 OLS 의 i 번째 파티셔닝 스킴의 k 번째 비트스트림 파티션에 대해 특정된 전달 스케줄 및 hrd_parameters() 의 j 번째 조합에서 이용되는 인덱스 bsp_comb_hrd_idx[h][i][j][k] 를 갖는 hrd_parameters() 신택스 구조 내의 전달 스케줄의 인덱스를 특정한다. bsp_comb_sched_idx[h][i][j][k]의 값은 0 내지 cpb_cnt_minus1[HighestTid]의 범위에 있어야 하고, 여기서 cpb_cnt_minus1[HighestTid]는 인덱스 bsp_comb_hrd_idx[h][i][j][k]에 대응하는 hrd_parameters() 신택스 구조로부터의 sub_layer_hrd_parameters(HighestTid) 신택스 구조에서 발견된다. [편집됨 (YK): "sub_layer_hrd_parameters(HighestTid)" 및 "sub_layer_hrd_parameters()"의 양쪽 형태들은 신택스 구조의 참조를 위해 본 문서에서 사용된다. 그들 중 하나만을 일관되게 사용하는 것이 더 나은지 여부를 체크한다.]"]

[0114] HEVC 는 fixed_pic_rate_general_flag[i] 에 대한 다음 시맨틱스를 특정한다: 1 과 동일한 fixed_pic_rate_general_flag[i] 는, HighestTid 가 i 와 동일할 때, 출력 순서에서의 연속적인 픽처들의 HRD 출력 시간들 사이의 시간적 거리가 아래에 특정된 바와 같이 제약됨을 나타낸다. 0 과 동일한 fixed_pic_rate_general_flag[i] 는 이 제약이 적용되지 않을 수도 있음을 나타낸다.

[0115] 테이블 3 의 예에서, num_sub_layer_hrd_minus1[i] 는 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 신택스 엘리먼트의 예를 표현한다. HRD 파라미터들이 코딩되는 서브-계층들의 개수는 비트스트림의 비디오 파라미터 세트 (VPS)에 의해 나타낸 서브-계층들의 최대 개수 이하일 수도 있다. 따라서, 비디오 코더는 신택스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩한 후에 HRD 파라미터들을 이용하여 비트스트림을 프로세싱할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HRD 파라미터들에 따라 디코딩된 픽처 베퍼로부터 픽처들을 제거할 수도 있다. 더욱이, 목적지 디바이스 (14) 는 디스플레이 디바이스 (32) 를 이용하여 디코딩된 픽처 베퍼로부터 제거된 픽처들을 디스플레이할 수도 있다.

[0116] 테이블 3 은 또한 비디오 코더가 다중-계층 비트스트림에 포함된 각각의 서브-계층에 대한 HRD 파라미터들의 세트를 코딩할 수도 있게 하는 다른 예시적인 기법을 표현한다. 테이블 1 에 나타낸 기법들과 대조하면, 테이블 3 의 예는 비트스트림 파티션에 포함된 서브-계층들의 세트에 대한 hrd_parameters() 신택스 구조의 리스트에 대한 인덱스를 단순히 시그널링하는 것을 포함한다.

[0117]

테이블 4

| | |
|--|-------|
| bsp_initial_arrival_time(payloadSize) { | 디스크립터 |
| if(NalHrdBpPresentFlag) | |
| for(i = 0; i <= CpbCnt [제거됨: "SchedCombCnt[sei_ols_idx][sei_partitioning_scheme_idx]"]; i++) | |
| nal_initial_arrival_delay[i] | u(v) |
| if(VclHrdBpPresentFlag)[제거됨: "else"] | |
| for(i = 0; i <= CpbCnt [제거됨: "SchedCombCnt[sei_ols_idx][sei_partitioning_scheme_idx]"]; i++) | |
| vcl_initial_arrival_delay[i] | u(v) |
| } | |

[0118]

테이블 4 의 신택스 엘리먼트들에 대한 예시적인 시맨틱스가 아래에 설명된다. 이와 달리 아래에 논의되지 않은 미변화된 신택스 엘리먼트들은 JCTVC-R0010v2 에 대한 변형 2 첨부물에서 설명된 바와 같이 시맨틱스를 유지할 수도 있다. 다시, 이탈릭체의 텍스트는 부가들을 표현하는 한편, [제거됨: ""] 은 삭제들을 표현한다.

[0120]

비트스트림 파티션 초기 도달 시간 SEI 메시지는 비트스트림-파티션-특정 CPB 동작에서 이용될 초기 도달 시간들을 특정한다.

[0121]

존재한다면, 이 SEI 메시지는 스케일러블 네스팅 SEI 메시지에 포함되는 비트스트림 파티션 네스팅 SEI 메시지 내에 포함되어야 하고, 동일한 비트스트림 파티션 네스팅 SEI 메시지는 또한 버퍼링 주기 SEI 메시지를 포함해야 한다.

[0122]

다음은 비트스트림 파티션 네스팅 SEI 메시지 신택스 및 시맨틱스에 대해 적용된다:

[0123]

- 신택스 엘리먼트 *initial_cpb_removal_delay_length_minus1* 및 변수들 *NalHrdBpPresentFlag* 및 *VclHrdBpPresentFlag* 는, 비트스트림 파티션 네스팅 SEI 메시지가 적용되는 동작 포인트들 중 적어도 하나에 적용 가능한 *hrd_parameters()* 신택스 구조에서 발견된 신택스 엘리먼트들에서 발견되거나 또는 그 신택스 엘리먼트들로부터 도출된다.

[0124]

[제거됨:]

[0125]

0 내지 *SchedCombCnt[sei_ols_idx][sei_partitioning_scheme_idx]* 의 범위에서의 *i* 에 대한 *hrdParamIdx[i]* 가 *bsp_comb_hrd_idx[olsIdx][partitioningSchemeIdx][i][bspIdx]* 의 값과 동일하다고 하고, 여기서 *olsIdx*, *partitioningSchemeIdx*, 및 *bspIdx* 는 이 비트스트림 파티션 초기 도달 시간 SEI 메시지를 포함하는 비트스트림 파티션 네스팅 SEI 메시지의 *sei_ols_idx*, *sei_partitioning_scheme_idx*, 및 *bsp_idx* 각각과 동일하다. *initialCpbRemovalDelayLength[i]* 가 *initial_cpb_removal_delay_length_minus1 + 1* 과 동일하다고 하고, 여기서 *initial_cpb_removal_delay_length_minus1* 은 액티브 VPS 에서의 *hrdParamIdx[i]* 번째 *hrd_parameters()* 신택스 구조에서 발견된다.]

[0126]

nal_initial_arrival_delay[i] 는, NAL HRD 파라미터들이 이용되고 있을 때, 이 SEI 메시지가 적용되는 비트스트림 파티션의 *i* 번째 전달 스캐줄 [제거됨: "조합"] 에 대한 초기 도달 시간을 특정한다. *nal_initial_arrival_delay[i]* 신택스 엘리먼트의 비트들에 있어서의 길이는 *initial_cpb_removal_delay_length_minus1 + 1* [제거됨: "initialCpbRemovalDelayLength[i]"] 와 동일하다.

[0127]

vcl_initial_arrival_delay[i] 는, VCL HRD 파라미터들이 이용되고 있을 때, 이 SEI 메시지가 적용되는 비트스트림 파티션의 *i* 번째 전달 스캐줄 [제거됨: "조합"] 에 대한 초기 도달 시간을 특정한다. *vcl_initial_arrival_delay[i]* 신택스 엘리먼트의 비트들에 있어서의 길이는 *initial_cpb_removal_delay_length_minus1 + 1* [제거됨: "initialCpbRemovalDelayLength[i]"] 와 동일하다.

[0128]

비디오 인코더 (20) 는 추가로, 예를 들어, 프레임 헤더, 블록 헤더, 슬라이스 헤더, 또는 GOP 헤더에서, 블록-기반 신택스 데이터, 프레임-기반 신택스 데이터, 및 GOP-기반 신택스 데이터와 같은 신택스 데이터를 비디오

디코더 (30)에 전송할 수도 있다. GOP 싱크스 데이터는 각각의 GOP에서의 프레임들의 개수를 설명할 수도 있고, 프레임 싱크스 데이터는 대응하는 프레임을 인코딩하기 위해 이용되는 인코딩/예측 모드를 나타낼 수도 있다.

[0129] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은, 적용가능하다면, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적 회로 (ASIC) 들, 필드 프로그래밍가능 게이트 어레이 (FPGA) 들, 이산 로직 회로부, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합과 같은 다양한 적합한 인코더 또는 디코더 회로부 중 임의의 것으로서 구현될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이들 중 어느 하나는 조합된 비디오 인코더/디코더 (코덱)의 부분으로서 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 셀룰러 전화기와 같은 무선 통신 디바이스를 포함할 수도 있다.

[0130] 도 2는 가상 참조 디코더 (HRD) 파라미터 시그널링을 개선시키기 위한 기법들을 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 주어진 비디오 프레임 또는 픽처 내의 비디오에서 공간적 리던던시를 감소시키거나 또는 제거하기 위해 공간적 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접 프레임들 또는 픽처들 내의 비디오에서 시간적 리던던시를 감소시키거나 또는 제거하기 위해 시간적 예측에 의존한다. 인트라-모드 (I 모드)는 수 개의 공간 기반 코딩 모드들 중 임의의 것을 지칭할 수도 있다. 단방향성 예측 (P 모드) 또는 양-예측 (B 모드)과 같은 인터-모드들은 수 개의 시간 기반 코딩 모드들 중 임의의 것을 지칭할 수도 있다.

[0131] 도 2에 도시된 바와 같이, 비디오 인코더 (20)는 인코딩될 비디오 프레임 내의 현재 비디오 블록을 수신한다. 도 2의 예에서, 비디오 인코더 (20)는 모드 선택 유닛 (40), 참조 픽처 메모리 (64) (디코딩된 픽처 버퍼 (DPB)라고도 또한 지칭될 수도 있음), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56)을 포함한다. 모드 선택 유닛 (40)은, 차례로, 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라 예측 유닛 (46), 및 파티션 유닛 (48)을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20)는 또한 역 양자화 유닛 (58), 역 변환 유닛 (60), 및 합산기 (62)를 포함한다. 디블록킹 필터 (도 2에 미도시)가 또한 블록 경계들을 필터링하여 재구성된 비디오로부터 블록키니스 아티팩트들을 제거하기 위해 포함될 수도 있다. 원한다면, 디블록킹 필터는 통상적으로 합산기 (62)의 출력을 필터링할 것이다.

부가적인 필터들 (인 루프 또는 포스트 루프)이 또한 디블록킹 필터에 부가적으로 이용될 수도 있다. 이러한 필터들은 간결성을 위해 도시되지 않지만, 원한다면, (인-루프 필터로서) 합산기 (50)의 출력을 필터링 할 수도 있다.

[0132] 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 압축을 제공하기 위해 하나 이상의 참조 프레임들에서의 하나 이상의 블록들에 관련된 수신된 비디오 블록의 인터-예측 인코딩을 수행한다. 인트라 예측 유닛 (46)은 대안적으로 공간적 압축을 제공하기 위해 코딩될 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 관련된 수신된 비디오 블록의 인트라-예측 인코딩을 수행할 수도 있다. 비디오 인코더 (20)는, 예를 들어, 비디오 데이터의 각각의 블록에 대한 적절한 코딩 모드를 선택하기 위해 다수의 코딩 패스 (coding pass)들을 수행할 수도 있다.

[0133] 더욱이, 파티션 유닛 (48)은 이전 코딩 패스들에서의 이전 파티셔닝 스키ム들의 평가에 기초하여 비디오 데이터의 블록들을 서브-블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛 (48)은 초기에는 프레임 또는 슬라이스를 LCU들로 파티셔닝하고, 레이트-웨곡 분석 (예를 들어, 레이트-웨곡 최적화)에 기초하여 LCU들 각각을 서브-CU들로 파티셔닝할 수도 있다. 모드 선택 유닛 (40)은 서브-CU들로의 LCU의 파티셔닝을 나타내는 큐드트리 데이터 구조를 추가로 생성할 수도 있다. 큐드트리의 리프-노드 CU들은 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다.

[0134] 모드 선택 유닛 (40)은, 예를 들어, 여러 결과들에 기초하여 예측 모드들 중 하나, 인트라 또는 인터를 선택할 수도 있고, 결과적인 예측된 블록을 합산기 (50)에 제공하여 잔차 데이터를 생성하고 합산기 (62)에 제공하여 참조 프레임으로서의 이용을 위해 인코딩된 블록을 재구성한다. 모드 선택 유닛 (40)은 또한 모션 벡터들, 인트라-모드 표시자들, 파티션 정보, 및 다른 이러한 싱크스 정보와 같은 싱크스 엘리먼트들을 엔트로피 인코딩 유닛 (56)에 제공한다.

[0135] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 고도로 통합될 수도 있지만, 개념적인 목적들을 위해 개별적으

로 예시된다. 모션 추정 유닛 (42)에 의해 수행된 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어, 현재 프레임 (또는 다른 코딩된 유닛) 내에서 코딩되고 있는 현재 블록에 관련된 참조 프레임 (또는 다른 코딩된 유닛) 내의 예측 블록에 관련된 현재 비디오 프레임 또는 픽처 내의 비디오 블록의 PU의 변위 (displacement)를 나타낼 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 관점에서 코딩될 블록과 가깝게 매칭시키기 위해 구해지는 블록이다. 일부 예들에서, 비디오 인코더 (20)는 참조 픽처 메모리 (64)에 저장된 참조 픽처들의 서브-정수 픽셀 포지션들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20)는 참조 픽처의 1/4 픽셀 포지션들, 1/8 픽셀 포지션들, 또는 다른 분수 픽셀 포지션들의 값들을 보간할 수도 있다. 따라서, 모션 추정 유닛 (42)은 전체 픽셀 포지션들 및 분수 픽셀 포지션들에 관련된 모션 탐색을 수행할 수도 있고, 분수 픽셀 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0136] 모션 추정 유닛 (42)은 PU의 포지션을 참조 픽처의 예측 블록의 포지션과 비교함으로써, 인터-코딩된 슬라이스에서의 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, 이들 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)에 전송한다.

[0137] 모션 보상 유닛 (44)에 의해 수행된 모션 보상은 모션 추정 유닛 (42)에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페칭 (fetching) 또는 생성하는 것을 수반할 수도 있다. 다시, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은, 일부 예들에서, 기능적으로 통합될 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터의 수신시, 모션 보상 유닛 (44)은 참조 픽처 리스트들 중 하나에서 모션 벡터가 가리키는 예측 블록을 로케이팅시킬 수도 있다. 합산기 (50)는, 아래에 논의되는 바와 같이, 코딩되고 있는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다.

일반적으로, 모션 추정 유닛 (42)은 루마 컴포넌트들에 관하여 모션 추정을 수행하고, 모션 보상 유닛 (44)은 크로마 컴포넌트들 및 루마 컴포넌트들 양쪽에 대해 루마 컴포넌트들에 기초하여 계산된 모션 벡터들을 이용한다. 모드 선택 유닛 (40)은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30)에 의한 이용을 위해 비디오 슬라이스 및 비디오 블록들과 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0138] 인트라 예측 유닛 (46)은, 상술된 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 수행된 인터-예측에 대한 대안으로서, 현재 블록을 인트라-예측할 수도 있다. 특히, 인트라 예측 유닛 (46)은 현재 블록을 인코딩하는데 이용하기 위한 인트라-예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 유닛 (46)은, 예를 들어, 별개의 인코딩 패스들 동안, 다양한 인트라-예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 유닛 (46) (또는 일부 예들에서는, 모드 선택 유닛 (40))은 테스팅된 모드들로부터 이용하기에 적절한 인트라-예측 모드를 선택할 수도 있다.

[0139] 예를 들어, 인트라 예측 유닛 (46)은 다양한 테스팅된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 이용하여 레이트-왜곡 값들을 계산할 수도 있고, 테스팅된 모드들 중에서 최상의 레이트-왜곡 특성을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로, 인코딩된 블록을 생성하기 위해 이용된 비트레이트 (즉, 비트들의 개수) 뿐만 아니라, 인코딩된 블록과, 인코딩된 블록을 생성하기 위해 인코딩되었던 오리지널의 인코딩되지 않은 블록과의 사이의 왜곡 (또는 에러)의 양을 결정한다. 인트라 예측 유닛 (46)은 어떠한 인트라-예측 모드가 블록에 대한 최상의 레이트-왜곡 값을 나타내는지를 결정하기 위해 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 계산할 수도 있다.

[0140] 블록에 대한 인트라-예측 모드를 선택한 후에, 인트라 예측 유닛 (46)은 블록에 대한 선택된 인트라-예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (56)에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56)은 선택된 인트라-예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20)는, 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 변경된 인트라-예측 모드 인덱스 테이블들 (또한 코드워드 맵핑 테이블들이라고도 지칭됨)을 포함할 수도 있는 송신된 비트스트림 구성 데이터에, 다양한 블록들에 대한 인코딩 콘텍스트들의 정의들과, 콘텍스트들의 각각에 대해 이용하기 위한 가장 가능성있는 인트라-예측 모드, 인트라-예측 모드 인덱스 테이블, 및 변경된 인트라-예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.

[0141] 비디오 인코더 (20)는 코딩되고 있는 오리지널 비디오 블록으로부터 모드 선택 유닛 (40)으로부터의 예측 데이터를 감산하는 것에 의해 잔차 비디오 블록을 형성한다. 합산기 (50)는 이 감산 동작을 수행하는 컴포넌

트 또는 컴포넌트들을 표현한다. 변환 프로세싱 유닛 (52)은 잔차 블록에 대해 변환, 예컨대 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환을 적용하여, 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 웨이블릿 변환들, 정수 변환들, 서브-대역 변환들, 이산 사인 변환 (discrete sine transform; DST) 들, 또는 다른 타입들의 변환들이 DCT 대신에 이용될 수 있다. 어느 경우든, 변환 프로세싱 유닛 (52)은 잔차 블록에 변환을 적용하여 변환 계수들의 블록을 생성한다. 변환은 잔차 정보를 픽셀 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 커버팅할 수도 있다. 변환 프로세싱 유닛 (52)은 결과적인 변환 계수들을 양자화 유닛 (54)에 전송할 수도 있다. 양자화 유닛 (54)은 비트 레이트를 추가로 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다.

양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다.

[0142] 양자화에 후속하여, 엔트로피 인코딩 유닛 (56)은 양자화된 변환 계수들을 스캐닝 및 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56)은 콘텍스트 적응 가변 길이 코딩 (CAVLC), 콘텍스트 적응 이진 산술 코딩 (CABAC), 신택스 기반 콘텍스트 적응 이진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 다른 엔트로피 코딩 기법을 수행할 수도 있다. 콘텍스트 기반 엔트로피 코딩의 경우, 콘텍스트는 이웃 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56)에 의한 엔트로피 코딩에 후속하여, 인코딩된 비트 스트림은 다른 디바이스 (예를 들어, 비디오 디코더 (30))로 송신될 수도 있거나, 또는 추후의 송신 또는 취출을 위해 아카이브될 수도 있다.

[0143] 역 양자화 유닛 (58) 및 역 변환 유닛 (60)은 픽셀 도메인에서 잔차 블록을 재구성하기 위해 역 양자화 및 역 변환을 각각 적용한다. 특히, 합산기 (62)는 재구성된 잔차 블록을 모션 보상 유닛 (44) 또는 인트라 예측 유닛 (46)에 의해 앞서 생성되는 모션 보상된 예측 블록에 가산하여, 참조 픽처 메모리 (64)에의 저장을 위한 재구성된 비디오 블록을 생성한다. 재구성된 비디오 블록은 후속 비디오 프레임에서의 블록을 인터-코딩하기 위한 참조 블록으로서 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 이용될 수도 있다.

[0144] 비디오 인코더 (20)는 일반적으로, 코딩된 비디오 시퀀스에서 각각의 픽처의 각각의 블록을 인코딩하기 위해 위에서 논의된 프로세스를 이용한다. 또한, 일부 예들에서, 비디오 인코더 (20)는 픽처들 각각을 할당할 시간적 계층들을 결정할 수도 있다. 게다가, 비디오 인코더 (20)는 다른 계층들, 예를 들어, 다른 뷰들, 스케일러블 비디오 코딩 계층들 등의 픽처들을 인코딩하도록 구성될 수도 있다. 어느 경우든, 비디오 인코더 (20)는, (예를 들어, 다양한 비디오 차원들의) 하나 이상의 계층들에 대해, 각각의 픽처가 속하는 계층을 나타내는 데이터를 추가로 인코딩할 수도 있다.

[0145] 본 개시물의 기법들에 따르면, 비디오 인코더 (20)는 또한, 예를 들어, 비디오 파라미터 세트 (VPS) 들, 시퀀스 파라미터 세트 (SPS) 들, 픽처 파라미터 세트 (PPS) 들, 보충 향상 정보 (SEI) 메시지들 등을 포함하는 파라미터 세트들과 같은 다른 데이터 구조들을 인코딩할 수도 있다. 본 개시물의 기법들에 따르면, 비디오 인코더 (20)는 위의 테이블들 1 또는 3에 대해 설명된 정보를 포함하는 VPS, 및/또는 위의 테이블들 2 또는 4에 대해 설명된 정보를 포함하는 SEI 메시지를 인코딩할 수도 있다.

[0146] 예를 들어, 비디오 인코더 (20)는 (예를 들어, VPS에 포함된) 가상 참조 디코더 (HRD) 파라미터들이 인코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 신택스 엘리먼트에 대한 값을 인코딩할 수도 있다. 본 개시물의 기법들에 따르면, 비디오 인코더 (20)는 비트스트림의 파티션의 각각의 서브-계층에 대한 HRD 파라미터들을 인코딩하지만, 파티션의 서브-계층들보다 더 많은 HRD 파라미터들을 코딩하는 것을 피하게 할 수도 있다.

따라서, 파티션에 대한 HRD 파라미터 데이터 구조들의 개수는, VPS에 나타낸 바와 같이, 서브-계층들의 최대 개수보다 더 적을 수도 있다. 게다가, 비디오 인코더 (20)는 HRD 파라미터들을 이용하여 비트스트림의 데이터를 프로세싱할 수도 있다. 예를 들어, 비디오 인코더 (20)는 HRD 파라미터들에서 시그널링되는 데이터에 따라 참조 픽처 메모리 (64)로부터 디코딩된 픽처들을 폐기할 수도 있다.

[0147] 위에서 논의된 예들에 부가적일 수도 있거나 또는 그에 대한 대안일 수도 있는 다른 예로서, 비디오 인코더 (20)는 VclHrdBpPresentFlag 가 1 과 동일한 (즉, 참의 값을 갖는) 경우에만 비디오 코딩 계층 HRD 파라미터들에 대한 초기 도달 딜레이를 표현하는 신택스 엘리먼트를 인코딩할 수도 있다. H.265에 따르면, VclHrdBpPresentFlag에 대한 값은 다음과 같이 설정된다:

[0148] - 다음 조건들 중 하나 이상이 참인 경우, VclHrdBpPresentFlag의 값이 1 과 동일하게 설정된다:

[0149] - vcl_hrd_parameters_present_flag 가 비트스트림에 존재하고 1 과 동일하다.

- [0150] - 베퍼링 주기 SEI 메시지들에서 비트스트림에 존재될 VCL HRD 동작에 대한 베퍼링 주기들의 존재에 대한 필요성은 본 명세서에서 특정되지 않은 일부 수단에 의해, 애플리케이션에 의해 결정된다.
- [0151] - 그렇지 않다면, VclHrdBpPresentFlag의 값은 0과 동일하게 설정된다.
- [0152] 따라서, 본 개시물의 기법들에 따르면, 비디오 인코더(20)는 비디오 코딩 계층(VCL) HRD 파라미터들 중 적어도 하나가 비트스트림에서 코딩되는 경우에만 또는 VCL HRD 동작들에 대한 베퍼링 주기 정보가 비트스트림에서 필요할 것으로 결정될 때 비디오 코딩 계층 HRD 파라미터들에 대한 초기 도달 딜레이를 표현하는 신택스 엘리먼트를 인코딩할 수도 있다.
- [0153] 이러한 방식으로, 도 2의 비디오 인코더(20)는 가상 참조 디코더(HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 신택스 엘리먼트에 대한 값을 인코딩하는 것으로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트(VPS)에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 신택스 엘리먼트에 대한 값을 인코딩하고, 신택스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 인코딩하며, HRD 파라미터들을 이용하여 비트스트림을 프로세싱하도록 구성된 비디오 인코더의 예를 표현한 것이다.
- [0154] 더욱이, 비디오 인코더(20)는 비디오 코딩 계층(VCL) HRD 파라미터들 중 적어도 하나가 비트스트림에서 코딩될 때에만 또는 VCL HRD 동작들에 대한 베퍼링 주기 정보가 비트스트림에서 필요할 것으로 결정될 때 비트스트림 파티션 초기 도달 시간 보충 향상 정보(SEI) 메시지의 초기 도달 딜레이 신택스 엘리먼트를 인코딩하도록 구성된 비디오 인코더의 예를 표현한 것이다. 즉, 비디오 인코더(20)는 VclHrdBpPresentFlag가 참의 값을 가질 때에만 비트스트림 파티션 초기 도달 시간 보충 향상 정보(SEI) 메시지의 초기 도달 딜레이 신택스 엘리먼트를 인코딩하도록 구성된 비디오 인코더의 예를 표현한 것이다.
- [0155] 도 3은 가상 참조 디코더(HRD) 파라미터 시그널링을 개선시키기 위한 기법들을 구현할 수도 있는 비디오 디코더(30)의 예를 예시하는 블록도이다. 도 3의 예에서, 비디오 디코더(30)는 엔트로피 디코딩 유닛(70), 모션 보상 유닛(72), 인트라 예측 유닛(74), 역 양자화 유닛(76), 역 변환 유닛(78), 참조 픽처 메모리(82) 및 합산기(80)를 포함한다. 비디오 디코더(30)는, 일부 예들에서, 비디오 인코더(20)(도 2)에 대해 설명된 인코딩 패스와는 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 한편, 인트라 예측 유닛(74)은 엔트로피 디코딩 유닛(70)으로부터 수신된 인트라-예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.
- [0156] 디코딩 프로세스 동안, 비디오 디코더(30)는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 표현하는 인코딩된 비디오 비트스트림을 비디오 인코더(20)로부터 수신한다. 비디오 디코더(30)의 엔트로피 디코딩 유닛(70)은 비트스트림을 엔트로피 디코딩하여, 양자화된 계수들, 모션 벡터들 또는 인트라-예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛(70)은 모션 벡터들 및 다른 신택스 엘리먼트들을 모션 보상 유닛(72)으로 포워딩한다. 비디오 디코더(30)는 신택스 엘리먼트들을 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 수신할 수도 있다.
- [0157] 비디오 슬라이스가 인트라-코딩된(I) 슬라이스로서 코딩될 때, 인트라 예측 유닛(74)은 시그널링된 인트라-예측 모드 및 현재 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재 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 신택스 엘리먼트들에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 픽처 리스트들 중 하나 내의 참조 픽처들 중 하나로부터 생성될 수도 있다. 비디오 디코더(30)는 참조 픽처 메모리(82)에 저장된 참조 픽처들에 기초한 디폴트 구성 기법들을 이용하여, 참조 프레임 리스트들, 즉, 리스트 0 및 리스트 1을 구성할 수도 있다. 모션 보상 유닛(72)은 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그 예측 정보를 이용하여 디코딩 중인 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛(72)은, 그 수신된 신택스 엘리먼트들 중 일부를 이용하여 비디오 슬라이스의 비디오 블록들을 코딩하기 위해 이용되는 예측 모드(예를 들어, 인트라- 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 현재 비디오 슬라이스에서의 비디오 블록들을 디코딩하기 위한 다른 정

보를 결정한다.

[0158] 모션 보상 유닛 (72)은 또한 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (72)은 비디오 블록들의 인코딩 동안 비디오 인코더 (20)에 의해 이용되는 것과 같은 보간 필터들을 이용하여 참조 블록들의 서브-정수 픽셀들에 대해 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (72)은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 이용되는 보간 필터들을 결정할 수도 있고 그 보간 필터들을 이용하여 예측 블록들을 생성할 수도 있다.

[0159] 역 양자화 유닛 (76)은 비트스트림에서 제공되며 엔트로피 디코딩 유닛 (70)에 의해 디코딩되는 양자화된 변환 계수들을 역 양자화한다, 즉, 양자화해제한다. 역 양자화 프로세스는 양자화의 정도, 그리고, 이와 마찬가지로, 적용되어야 하는 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 (QP_Y)의 이용을 포함할 수도 있다.

[0160] 역 변환 유닛 (78)은, 예를 들어, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스와 같은 역 변환을 변환 계수들에 적용하여 픽셀 도메인에서 잔차 블록들을 생성한다.

[0161] 모션 보상 유닛 (72)이 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 블록에 대한 예측 블록을 생성한 후에, 비디오 디코더 (30)는 역 변환 유닛 (78)으로부터의 잔차 블록들을 모션 보상 유닛 (72)에 의해 생성된 대응하는 예측 블록들과 합산함으로써, 디코딩된 비디오 블록을 형성한다. 합산기 (80)는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 표현한다. 원한다면, 디블록킹 필터는 또한 디코딩된 블록들을 필터링하기 위해 적용되어 블록키니스 아티팩트들을 제거할 수도 있다. (코딩 루프 중에 또는 코딩 루프 후에) 다른 루프 필터들이 또한 픽셀 천이들을 평활화하거나, 또는 이와 다르게는 비디오 품질을 개선시키기 위해 이용될 수도 있다. 주어진 프레임 또는 픽처에서 디코딩된 비디오 블록들은 그 후에, 후속 모션 보상을 위해 이용되는 참조 픽처들을 저장하는 참조 픽처 메모리 (82)에 저장된다. 참조 픽처 메모리 (82)는 또한 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상의 추후 제시를 위해 디코딩된 비디오를 저장한다.

[0162] 비디오 디코더 (30)는 일반적으로, 코딩된 비디오 시퀀스에서 각각의 픽처의 각각의 블록을 디코딩하기 위해 위에서 논의된 프로세스를 이용한다. 또한, 일부 예들에서, 비디오 디코더 (30)는 픽처들이 할당되는 시간적 계층들을 나타내는 데이터를 디코딩할 수도 있다. 게다가, 비디오 디코더 (30)는 다른 계층들, 예를 들어, 다른 뷰들, 스케일러블 비디오 코딩 계층들 등의 픽처들을 디코딩하도록 구성될 수도 있다. 어느 경우든, 비디오 디코더 (30)는, (예를 들어, 다양한 비디오 차원들의) 하나 이상의 계층들에 대해, 각각의 픽처가 속하는 계층을 나타내는 데이터를 추가로 디코딩할 수도 있다.

[0163] 본 개시물의 기법들에 따르면, 비디오 디코더 (30)는 또한, 예를 들어, 비디오 파라미터 세트 (VPS)들, 시퀀스 파라미터 세트 (SPS)들, 픽처 파라미터 세트 (PPS)들, 보충 향상 정보 (SEI) 메시지를 등을 포함하는 파라미터 세트들과 같은 다른 데이터 구조들을 디코딩할 수도 있다. 본 개시물의 기법들에 따르면, 비디오 디코더 (30)는 위의 테이블들 1 또는 3에 대해 설명된 정보를 포함하는 VPS, 및/또는 위의 테이블들 2 또는 4에 대해 설명된 정보를 포함하는 SEI 메시지를 디코딩할 수도 있다.

[0164] 예를 들어, 비디오 디코더 (30)는 (예를 들어, VPS에 포함된) 가상 참조 디코더 (HRD) 파라미터들이 디코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 신택스 엘리먼트에 대한 값을 디코딩할 수도 있다. 본 개시물의 기법들에 따르면, 비디오 디코더 (30)는 비트스트림의 파티션의 각각의 서브-계층에 대한 HRD 파라미터들을 디코딩하지만, 파티션의 서브-계층들보다 더 많은 HRD 파라미터들을 코딩하는 것을 피하게 할 수도 있다.

따라서, 파티션에 대한 HRD 파라미터 데이터 구조들의 개수는, VPS에 나타낸 바와 같이, 서브-계층들의 최대 개수보다 더 적을 수도 있다. 게다가, 비디오 디코더 (30)는 HRD 파라미터들을 이용하여 비트스트림의 데이터를 프로세싱할 수도 있다. 예를 들어, 비디오 디코더 (30)는 HRD 파라미터들에서 시그널링되는 데이터에 따라 참조 픽처 메모리 (82)로부터 디코딩된 픽처들을 출력 및/또는 폐기할 수도 있다. 특히, 비디오 디코더 (30)는, 비디오 디스플레이로 하여금 디코딩된 픽처들을 제시하게 하기 위해, 디스플레이 디바이스 (32)와 같은 비디오 디스플레이에 디코딩된 픽처들을 출력할 수도 있다.

[0165] 위에서 논의된 예들에 부가적일 수도 있거나 또는 그에 대한 대안일 수도 있는 다른 예로서, 비디오 디코더 (30)는 $VclHrdBpPresentFlag$ 가 1과 동일한 (즉, 참의 값을 갖는) 경우에만 비디오 코딩 계층 HRD 파라미터들에 대한 초기 도달 딜레이를 표현하는 신택스 엘리먼트를 디코딩할 수도 있다. H.265에 따르면, $VclHrdBpPresentFlag$ 에 대한 값은 다음과 같이 설정된다:

- [0166] - 다음 조건들 중 하나 이상이 참인 경우, VclHrdBpPresentFlag 의 값이 1 과 동일하게 설정된다:
- vcl_hrd_parameters_present_flag 가 비트스트림에 존재하고 1 과 동일하다.
- [0168] - 베퍼링 주기 SEI 메시지들에서 비트스트림에 존재될 VCL HRD 동작에 대한 베퍼링 주기들의 존재에 대한 필요성은 본 명세서에서 특정되지 않은 일부 수단에 의해 결정된다.
- [0169] - 그렇지 않다면, VclHrdBpPresentFlag 의 값은 0 과 동일하게 설정된다.
- [0170] 따라서, 본 개시물의 기법들에 따르면, 비디오 디코더 (30) 는 비디오 코딩 계층 (VCL) HRD 파라미터들 중 적어도 하나가 비트스트림에서 코딩되는 경우에만 또는 VCL HRD 동작들에 대한 베퍼링 주기 정보가 비트스트림에서 필요할 것으로 결정될 때 비디오 코딩 계층 HRD 파라미터들에 대한 초기 도달 딜레이를 표현하는 십백스 엘리먼트를 디코딩할 수도 있다.
- [0171] 이러한 방식으로, 도 3 의 비디오 디코더 (30) 는 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 십백스 엘리먼트에 대한 값을 디코딩하는 것으로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트 (VPS) 에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 십백스 엘리먼트에 대한 값을 디코딩하고, 십백스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 디코딩하며, HRD 파라미터들을 이용하여 비트스트림을 프로세싱하도록 구성된 비디오 디코더의 예를 표현한 것이다.
- [0172] 더욱이, 비디오 디코더 (30) 는 비디오 코딩 계층 (VCL) HRD 파라미터들 중 적어도 하나가 비트스트림에서 코딩될 때에만 또는 VCL HRD 동작들에 대한 베퍼링 주기 정보가 비트스트림에서 필요할 것으로 결정될 때 비트스트림 파티션 초기 도달 시간 보충 향상 정보 (SEI) 메시지의 초기 도달 딜레이 십백스 엘리먼트를 디코딩하도록 구성된 비디오 디코더의 예를 표현한 것이다. 즉, 비디오 디코더 (30) 는 VclHrdBpPresentFlag 가 참의 값을 가질 때에만 비트스트림 파티션 초기 도달 시간 보충 향상 정보 (SEI) 메시지의 초기 도달 딜레이 십백스 엘리먼트를 디코딩하도록 구성된 비디오 디코더의 예를 표현한 것이다. 비디오 디코더 (30) 는, 이들 기법들에 기초하여, 비트스트림의 비트들이 비트스트림 파티션 초기 도달 시간 SEI 메시지, 또는 상이한 데이터 구조에 대응하는지 여부를 결정하고, 그에 의해 비트스트림을 올바르게 파싱할 수도 있다.
- [0173] 도 4 는 본 개시물의 기법들에 따른 비디오 데이터를 인코딩하기 위한 예시적인 방법을 예시하는 플로우차트이다. 비디오 인코더 (20) (도 1 및 도 2) 에 대해 설명되었지만, 다른 디바이스들이 도 4 의 방법과 유사한 방법을 수행하도록 구성될 수도 있다는 것을 이해해야 한다.
- [0174] 이 예에서, 비디오 인코더 (20) 는 초기에는 비트스트림의 서브-계층들의 최대 개수를 결정한다 (150). 비디오 인코더 (20) 는 또한 비트스트림에 대한 비디오 파라미터 세트 (VPS) 에서의 서브-계층들의 최대 개수를 시그널링한다 (152). 비트스트림은 궁극적으로 다양한 파티션들로 파티셔닝되고, 이들 각각은 서브-계층들의 특정 서브세트를 포함한다. 따라서, 소정의 파티션들은 서브-계층들의 최대 개수보다 더 적은 서브-계층을 포함할 것이다.
- [0175] 비디오 인코더 (20) 는 그 후에 비트스트림 파티션에서의 서브-계층들을 결정할 수도 있다 (154). 비디오 인코더 (20) 는 그 후에 파티션에서의 각각의 서브-계층에 대한 HRD 파라미터들을 시그널링할 수도 있다 (156). 예를 들어, 테이블들 1 및 3 에 나타낸 바와 같이, 비디오 인코더 (20) 는 bsp_hrd_idx 십백스 엘리먼트들에 대한 값을 인코딩할 수도 있다. 특히, 테이블 1 에서, 비디오 인코더 (20) 는 bsp_hrd_idx[h][i][t][j][k] 에 대한 값을 인코딩할 수도 있는 한편, 테이블 3 에서, 비디오 인코더 (20) 는 bsp_hrd_idx[h][i][j][k] 에 대한 값을 인코딩할 수도 있다. 테이블 1 에서, 이들 값을 출력 계층 세트들, 파티셔닝 스킴들, 및 계층 세트에서의 서브-계층들의 개수에 걸친 네스팅된 루프들 내에서 발생하는 반면, 테이블 3 에서, 이들 값을 출력 계층 세트들 및 파티셔닝 스킴들의 개수에 걸친 네스팅된 루프들 내에서 발생한다.
- [0176] 비디오 인코더 (20) 는 또한 서브-계층들의 픽처들을 인코딩하고 (158), 서브-계층들의 인코딩된 픽처들을 디코딩하며 (160), 디코딩된 픽처들을 참조 픽처 메모리 (64) (도 2) 와 같은 디코딩된 픽처 버퍼 (DPB) 에 저장한다 (162). 비디오 인코더 (20) 는 참조 픽처들로서의 후속 이용을 위해 인코딩된 픽처들의 디코딩된 버전들을 저장하여, 참조 픽처들의 이들 버전들로부터의 후속 예측이 비디오 디코더 (30) 와 같은 디코더에 의해 디코딩된 궁극적인 버전들과 동일하게 되도록 한다. 게다가, 비디오 인코더 (20) 는 HRD 파라미터들에 따라 DPB

로부터 디코딩된 픽처들을 제거한다 (164).

[0177] 더욱이, 본 개시물의 소정의 기법들에 따르면, 비디오 인코더 (20) 는 비트스트림 파티션 초기 도달 시간 SEI 메시지의 데이터를 조건부로 인코딩할 수도 있다 (166). 특히, 비디오 인코더 (20) 는, 예를 들어, 비디오 코딩 계층 (VCL) HRD 파라미터들 중 적어도 하나가 비트스트림에서 코딩되는 경우에만 또는 VCL HRD 동작들에 대한 베퍼링 주기 정보가 비트스트림에서 필요할 것으로 결정될 때, VclHrdBpPresentFlag 가 참의 값 (즉, 1) 을 갖는다는 결정 후에만 비트스트림 파티션 초기 도달 시간 SEI 메시지의 초기 도달 딜레이 싱크스 엘리먼트를 인코딩할 수도 있다.

[0178] 이러한 방식으로, 도 4 의 방법은 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의 개수를 나타내는 싱크스 엘리먼트에 대한 값을 코딩 (이 예에서는, 인코딩) 하는 단계로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트 (VPS) 에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 싱크스 엘리먼트에 대한 값을 코딩하는 단계, 싱크스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩 (이 예에서는, 인코딩) 하는 단계, 및 HRD 파라미터들을 이용하여 비트스트림을 프로세싱하는 단계를 포함하는 방법의 예를 표현한다.

[0179] 도 5 는 본 개시물의 기법들에 따른 비디오 데이터를 디코딩하기 위한 예시적인 방법을 예시하는 플로우차트이다. 비디오 디코더 (30) (도 1 및 도 3) 에 대해 설명되었지만, 다른 디바이스들이 도 5 의 방법과 유사한 방법을 수행하도록 구성될 수도 있다는 것을 이해해야 한다.

[0180] 이 예에서, 비디오 디코더 (30) 는 초기에는 비트스트림의 서브-계층들의 최대 개수를 나타내는 비디오 파라미터 세트 (VPS) 를 디코딩한다 (200). 비트스트림은 궁극적으로 다양한 파티션들로 파티셔닝되고, 이를 각각은 서브-계층들의 특정 서브세트를 포함한다. 따라서, 소정의 파티션들은 서브-계층들의 최대 개수보다 더 적은 서브-계층을 포함한다.

[0181] 비디오 디코더 (30) 는 그 후에 비트스트림 파티션에서의 서브-계층들을 결정할 수도 있다 (202). 비디오 디코더 (30) 는 그 후에 파티션에서의 각각의 서브-계층에 대한 HRD 파라미터들을 디코딩할 수도 있다 (204). 예를 들어, 테이블들 1 및 3 에 나타낸 바와 같이, 비디오 디코더 (30) 는 bsp_hrd_idx 싱크스 엘리먼트들에 대한 값을 디코딩할 수도 있다. 특히, 테이블 1 에서, 비디오 디코더 (30) 는 bsp_hrd_idx[h][i][t][j][k] 에 대한 값을 디코딩할 수도 있는 한편, 테이블 3 에서, 비디오 디코더 (30) 는 bsp_hrd_idx[h][i][j][k] 에 대한 값을 디코딩할 수도 있다. 테이블 1 에서, 이들 값을 출력 계층 세트들, 파티셔닝 스킴들, 및 계층 세트에서의 서브-계층들의 개수에 걸친 네스팅된 루프들 내에서 발생하는 반면, 테이블 3 에서, 이들 값을 출력 계층 세트들 및 파티셔닝 스킴들의 개수에 걸친 네스팅된 루프들 내에서 발생 한다.

[0182] 비디오 디코더 (30) 는 또한 서브-계층들의 인코딩된 픽처들을 디코딩하고 (206), 디코딩된 픽처들을 참조 픽처 메모리 (82) (도 3) 와 같은 디코딩된 픽처 베퍼 (DPB) 에 저장한다 (208). 비디오 디코더 (30) 는 참조 픽처들로서의 후속 이용을 위해 디코딩된 픽처들을 저장하여, 참조 픽처들의 이들 베풀들로부터의 후속 예측이 비디오 디코더 (30) 와 같은 디코더에 의해 디코딩된 궁극적인 베풀들과 동일하게 되도록 한다. 더욱이, 비디오 디코더 (30) 는 디코딩된 픽처들을 저장하여 비디오 디코더 (30) 가 디코딩된 픽처들을 적절한 때에 출력할 수 있다. 따라서, 비디오 디코더 (30) 는 HRD 파라미터들에 따라 DPB 로부터 디코딩된 픽처들을 제거 및 출력한다 (210).

[0183] 더욱이, 본 개시물의 소정의 기법들에 따르면, 비디오 디코더 (30) 는 비트스트림 파티션 초기 도달 시간 SEI 메시지의 데이터를 조건부로 디코딩할 수도 있다 (212). 특히, 비디오 디코더 (30) 는, 예를 들어, 비디오 코딩 계층 (VCL) HRD 파라미터들 중 적어도 하나가 비트스트림에서 코딩되는 경우에만 또는 VCL HRD 동작들에 대한 베퍼링 주기 정보가 비트스트림에서 필요할 것으로 결정될 때, VclHrdBpPresentFlag 가 참의 값 (즉, 1) 을 갖는다는 결정 후에만 비트스트림 파티션 초기 도달 시간 SEI 메시지의 초기 도달 딜레이 싱크스 엘리먼트를 디코딩할 수도 있다. 즉, 비디오 디코더 (30) 와 연관된 파서 (parser) (미도시) 는, 비트스트림의 소정의 비트들을, 비트스트림 파티션 초기 도달 시간 SEI 메시지의 싱크스 엘리먼트, 또는 별개의 싱크스 엘리먼트에 속하는 것으로서 해석할 수도 있다. 다시 말해, 파서는, HRD 파라미터들에 대응하는 비트스트림의 비트들과의 사이를 구별할 수도 있다.

[0184] 이러한 방식으로, 도 5 의 방법은 가상 참조 디코더 (HRD) 파라미터들이 코딩되는 비트스트림의 서브-계층들의

개수를 나타내는 십택스 엘리먼트에 대한 값을 코딩 (이 예에서는, 디코딩) 하는 단계로서, 여기서 값은 HRD 파라미터들이 코딩되는 서브-계층들의 개수가 비트스트림의 비디오 파라미터 세트 (VPS)에 의해 나타낸 서브-계층들의 최대 개수보다 더 적음을 나타내는, 그 십택스 엘리먼트에 대한 값을 코딩하는 단계, 십택스 엘리먼트에 대한 값으로 나타낸 개수의 서브-계층들에 대한 HRD 파라미터들을 코딩 (이 예에서는, 디코딩) 하는 단계, 및 HRD 파라미터들을 이용하여 비트스트림을 프로세싱하는 단계를 포함하는 방법의 예를 표현한다.

[0185] 예에 따라서는, 본 명세서에서 설명되는 기법들 중의 임의의 기법의 소정의 액트 (act) 들 또는 이벤트들이 상이한 시퀀스에서 수행될 수 있거나, 부가될 수도 있거나, 병합될 수도 있거나, 또는 모두 배제 (예를 들어, 모든 설명된 액트들 또는 이벤트들이 기법들의 실시를 위해 필요한 것은 아님) 될 수도 있다는 것이 인식되어야 한다. 더욱이, 소정의 예들에서, 액트들 또는 이벤트들이 순차적으로보다는, 예를 들어, 멀티-스레딩된 (multi-threaded) 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해 동시에 수행될 수도 있다.

[0186] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 그 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, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은, 본 개시물에서 설명되는 기법들의 구현을 위해 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0187] 제한이 아닌 예로서, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 원하는 프로그램 코드를 저장하는데 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체들을 포함할 수 있다. 또한, 임의의 맥락이 컴퓨터 판독가능 매체로 적절히 지정된다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털가입자 회선 (DSL), 또는 무선 기술들, 예컨대, 적외선, 무선, 및 마이크로파를 이용하여, 웹사이트, 서버 또는 다른 원격 소스로부터 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 그 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 연결들, 캐리어 파들, 신호들, 또는 다른 일시적 매체들을 포함하는 것이 아니라, 그 대신에 비일시적, 유형의 저장 매체들에 관한 것이라는 것을 이해해야 한다. 디스크 (disk) 및 디스크 (disc) 는, 본 명세서에서 사용되는 바와 같이, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 데이터를 자기적으로 보통 재생하지만, 디스크 (disc) 들은 레이저들로 데이터를 광학적으로 재생한다. 또한, 상기의 조합들도 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

[0188] 명령들은 하나 이상의 디지털 신호 프로세서 (DSP) 들, 범용 마이크로프로세서들, 주문형 집적 회로 (ASIC) 들, 필드 프로그래밍 가능한 게이트 어레이 (FPGA) 들, 또는 다른 등가의 집적 또는 이산 로직 회로부와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 이에 따라, 본 명세서에서 사용되는 바와 같은 용어 "프로세서"는 전술한 구조, 또는 본 명세서에서 설명되는 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭 할 수도 있다. 또한, 일부 양태들에서, 본 명세서에서 설명되는 기능성은, 인코딩 및 디코딩을 위해 구성되거나 또는 조합된 코덱 내에 포함되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

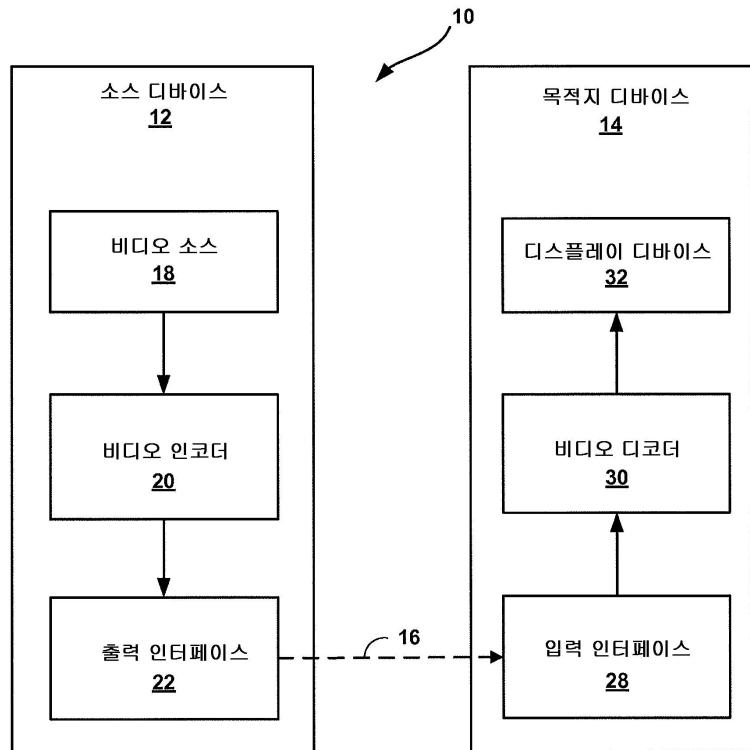
[0189] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩셋) 를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 상술된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 조합될 수도 있거나, 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상술된 하나 이상의 프로세서들을 포함하여, 상호작용하는 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

[0190]

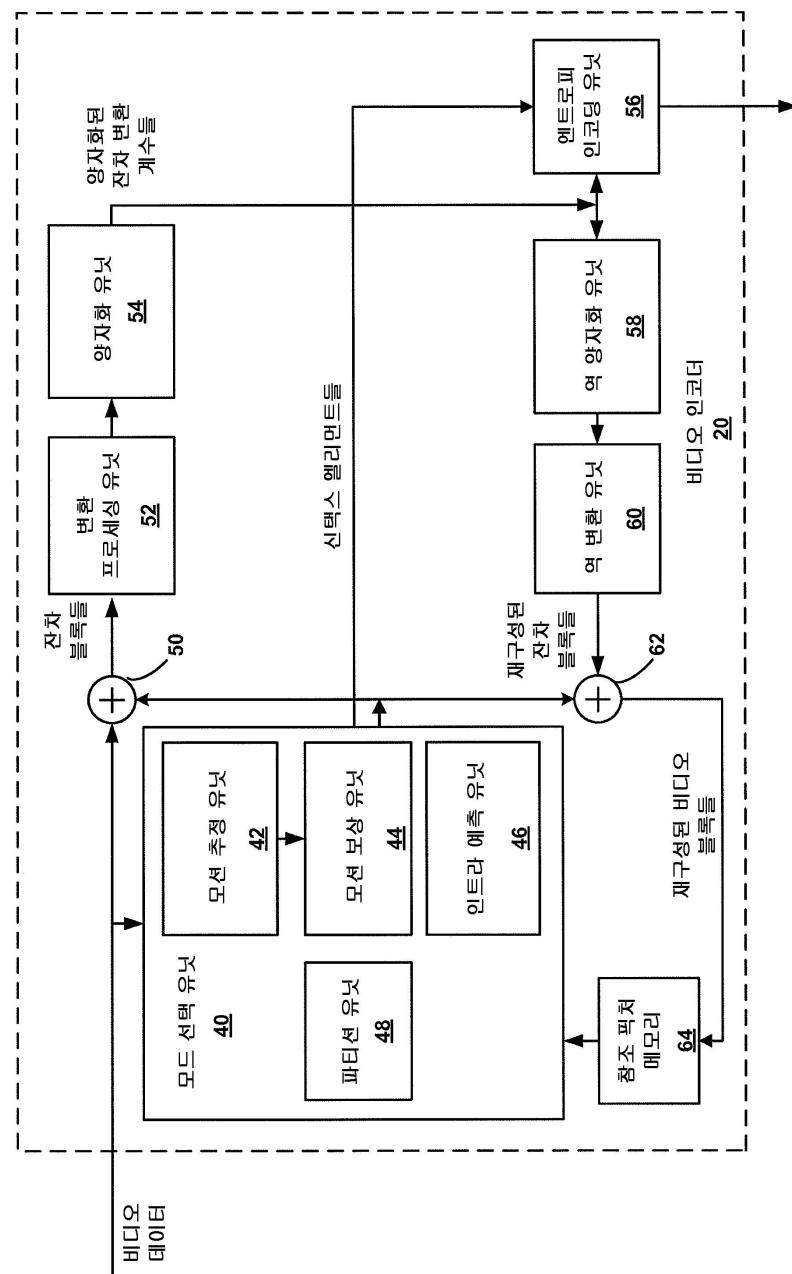
다양한 예들이 설명되었다. 이러한 그리고 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

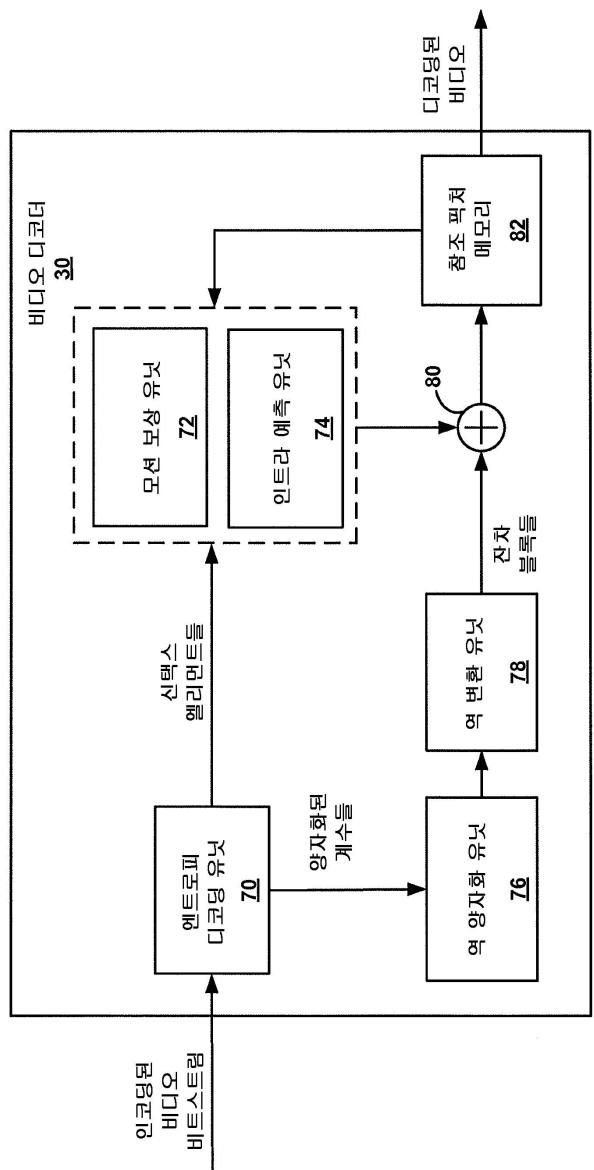
도면1



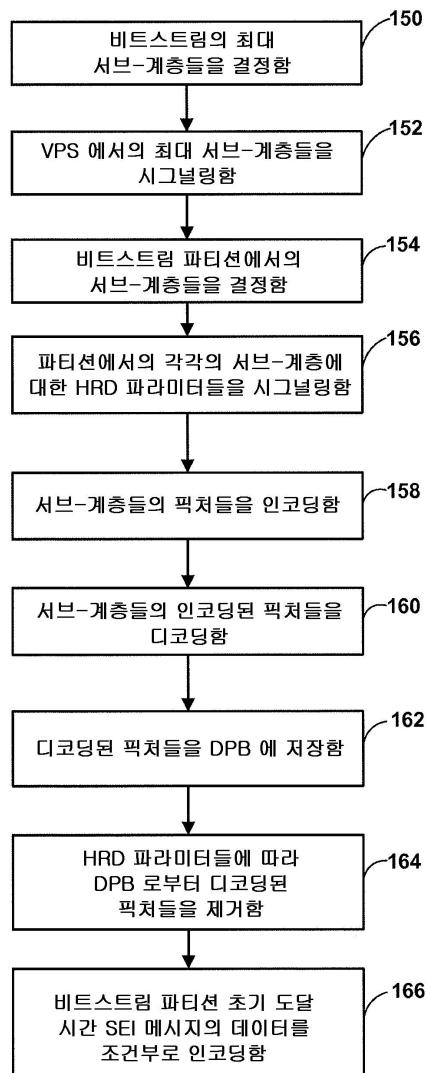
도면2



도면3



도면4



도면5

