



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년11월17일
(11) 등록번호 10-1799165
(24) 등록일자 2017년11월13일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) H04N 19/176 (2014.01)
H04N 19/30 (2014.01) H04N 19/46 (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/176 (2015.01)
(21) 출원번호 10-2015-7002367
(22) 출원일자(국제) 2013년07월02일
심사청구일자 2017년05월15일
(85) 번역문제출일자 2015년01월28일
(65) 공개번호 10-2015-0024934
(43) 공개일자 2015년03월09일
(86) 국제출원번호 PCT/US2013/049126
(87) 국제공개번호 WO 2014/008290
국제공개일자 2014년01월09일
(30) 우선권주장
61/667,387 2012년07월02일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
Y-K Wang et al., 'HRD parameters in VPS',
(JCTVC-J0562), JCT-VC of ITU-T SG16 WP3 and
ISO/IEC JTC1/SC29/WG11 10th Meeting:
STOCKHOLM, SE, 2012.07.14.
(뒷면에 계속)

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하
우스 드라이브 5775
(72) 발명자
천 잉
미국 92121-1714 캘리포니아주 샌디에고 모어하
우스 드라이브 5775
왕 예-쿠이
미국 92121-1714 캘리포니아주 샌디에고 모어하
우스 드라이브 5775
(74) 대리인
특허법인코리어나

전체 청구항 수 : 총 49 항

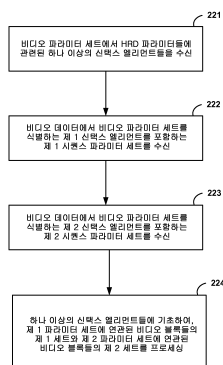
심사관 : 조우연

(54) 발명의 명칭 HEVC 및 확장들에 대한 비디오 파라미터 세트

(57) 요약

비디오 코더는, 비디오 파라미터 세트에서, 가상 참조 디코더 (HRD) 파라미터들에 관련된 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 수신하며; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 1 신택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 수신하며; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 2 신택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 수신하며; 그리고, 하나 이상의 신택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 코딩하도록 구성될 수 있다.

대표도 - 도12



(52) CPC특허분류

H04N 19/30 (2015.01)

H04N 19/46 (2015.01)

(56) 선행기술조사문헌

US20060239563 A1

US20060039480 A1

US20060256851 A1

US20060098001 A1

(30) 우선권주장

61/669,587 2012년07월09일 미국(US)

61/798,135 2013년03월15일 미국(US)

13/932,960 2013년07월01일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터에서, 가상 참조 디코더 (hypothetical reference decoder; HRD) 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 포함하는 비디오 파라미터 세트 (VPS) 를 수신하는 단계;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트 (SPS) 를 수신하는 단계;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 SPS 를 수신하는 단계;

상기 하나 이상의 선택스 엘리먼트들에 기초하여, 상기 제 1 SPS 와 연관된 비디오 블록들의 제 1 세트 및 상기 제 2 SPS 와 연관된 비디오 블록들의 제 2 세트를 디코딩하는 단계; 및

디코딩된 상기 비디오 블록들의 제 1 세트 및 디코딩된 상기 비디오 블록들의 제 2 세트를 포함하는 디코딩된 비디오를 출력하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 SPS 는 상기 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함하고, 상기 제 2 SPS 는 상기 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들에 대한 상기 값들이 디폴트 값들임을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들이 상기 비디오 데이터로 시그널링됨을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 4 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 4 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트가 일정한지 또는 가변적인지를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 4 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 코딩된 화상 버퍼 관리를 위한 하나 이상의 선택스 엘리먼트들을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 4 항에 있어서,

상기 비디오 데이터는 2 개 이상의 시간 계층들을 포함하고,

상기 비디오 데이터를 디코딩하는 방법은,

상기 VPS 에서, 상기 2 개 이상의 시간 계층들의 각 개별의 시간 계층에 대해, 상기 개별의 시간 계층에 대한 HRD 파라미터들의 개별의 세트에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 9

비디오 데이터를 인코딩하는 방법으로서,

상기 비디오 데이터에 대한 비디오 파라미터 세트 (VPS) 내의 포함을 위해, 가상 참조 디코더 (HRD) 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 생성하는 단계;

상기 비디오 데이터 내의 포함을 위해, 상기 VPS 를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트 (SPS) 를 생성하는 단계;

상기 비디오 데이터 내의 포함을 위해, 상기 VPS 를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 SPS 를 생성하는 단계; 및

상기 하나 이상의 선택스 엘리먼트들에 기초하여, 상기 제 1 SPS 와 연관된 비디오 블록들의 제 1 세트 및 상기 제 2 SPS 와 연관된 비디오 블록들의 제 2 세트를 인코딩하는 단계; 및

인코딩된 상기 비디오 블록들의 제 1 세트 및 인코딩된 상기 비디오 블록들의 제 2 세트를 포함하는 비디오 데이터를 출력하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 10

제 9 항에 있어서,

상기 제 1 SPS 는 상기 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함하고, 상기 제 2 SPS 는 상기 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 11

제 9 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들에 대한 상기 값들이 디폴트 값들임을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 12

제 9 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들이 상기 비디오 데이터로 시그널링됨을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 13

제 12 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 14

제 12 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트가 일정한지 또는 가변적인지를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 15

제 12 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 코딩된 화상 버퍼 관리를 위한 하나 이상의 선택스 엘리먼트들을 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 16

제 12 항에 있어서,

상기 비디오 데이터는 2 개 이상의 시간 계층들을 포함하고,

상기 비디오 데이터를 인코딩하는 방법은,

상기 VPS 내의 포함을 위해, 상기 시간 계층들의 각각에 대한 HRD 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 17

비디오 데이터를 프로세싱하는 디바이스로서,

상기 비디오 데이터를 저장하는 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터에서, 가상 참조 디코더 (HRD) 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 포함하는 비디오 파라미터 세트 (VPS) 를 수신하며;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트 (SPS) 를 수신하며;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 SPS 를 수신하며;

상기 하나 이상의 선택스 엘리먼트들에 기초하여, 상기 제 1 SPS 와 연관된 비디오 블록들의 제 1 세트 및 상기 제 2 SPS 와 연관된 비디오 블록들의 제 2 세트를 디코딩하며; 그리고

디코딩된 상기 비디오 블록들의 제 1 세트 및 디코딩된 상기 비디오 블록들의 제 2 세트를 포함하는 디코딩된 비디오를 출력하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 18

제 17 항에 있어서,

상기 제 1 SPS 는 상기 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함하고, 상기 제 2 SPS 는 상기 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 19

제 17 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들에 대한 상기 값들이 디폴트 값들임을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 20

제 17 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들이 상기 비디오 데이터로 시그널링됨을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 21

제 20 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 22

제 20 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트가 일정한지 또는 가변적인지를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 23

제 20 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 코딩된 화상 버퍼 관리를 위한 하나 이상의 선택스 엘리먼트들을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 24

제 20 항에 있어서,

상기 비디오 데이터는 2 개 이상의 시간 계층들을 포함하고,

상기 하나 이상의 프로세서들은 또한,

상기 VPS 에서, 상기 2 개 이상의 시간 계층들의 각 개별의 시간 계층에 대해, 상기 개별의 시간 계층에 대한 HRD 파라미터들의 개별의 세트에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 수신하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 25

비디오 데이터를 프로세싱하는 디바이스로서,

상기 비디오 데이터를 저장하는 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터에 대한 비디오 파라미터 세트 (VPS) 내의 포함을 위해, 가상 참조 디코더 (HRD) 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 생성하며;

상기 비디오 데이터 내의 포함을 위해, 상기 VPS 를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파

라미터 세트 (SPS) 를 생성하며;

상기 비디오 데이터 내의 포함을 위해, 상기 VPS 를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 SPS 를 생성하며;

상기 하나 이상의 선택스 엘리먼트들에 기초하여, 상기 제 1 SPS 와 연관된 비디오 블록들의 제 1 세트 및 상기 제 2 SPS 와 연관된 비디오 블록들의 제 2 세트를 인코딩하며; 그리고

인코딩된 상기 비디오 블록들의 제 1 세트 및 인코딩된 상기 비디오 블록들의 제 2 세트를 포함하는 인코딩된 비디오 데이터를 출력하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 26

제 25 항에 있어서,

상기 제 1 SPS 는 상기 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함하고, 상기 제 2 SPS 는 상기 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 27

제 25 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들에 대한 상기 값들이 디폴트 값들임을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 28

제 25 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들이 상기 비디오 데이터로 시그널링됨을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 29

제 28 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 30

제 28 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트가 일정한지 또는 가변적인지를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 31

제 28 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 코딩된 화상 버퍼 관리를 위한 하나 이상의 선택스 엘리먼트들을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 32

제 28 항에 있어서,

상기 비디오 데이터는 2 개 이상의 시간 계층들을 포함하고,

상기 하나 이상의 프로세서들은 또한,

상기 VPS 내의 포함을 위해, 상기 시간 계층들의 각각에 대한 HRD 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 생성하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 33

제 25 항에 있어서,

상기 디바이스는,

집적회로;

마이크로프로세서; 또는

무선 통신 디바이스

중 적어도 하나를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 34

비디오 데이터를 프로세싱하는 디바이스로서,

상기 비디오 데이터에서, 가상 참조 디코더 (HRD) 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 포함하는 비디오 파라미터 세트 (VPS) 를 수신하는 수단;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트 (SPS) 를 수신하는 수단;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 SPS 를 수신하는 수단;

상기 하나 이상의 선택스 엘리먼트들에 기초하여, 상기 제 1 SPS 와 연관된 비디오 블록들의 제 1 세트 및 상기 제 2 SPS 와 연관된 비디오 블록들의 제 2 세트를 디코딩하는 수단; 및

디코딩된 상기 비디오 블록들의 제 1 세트 및 디코딩된 상기 비디오 블록들의 제 2 세트를 포함하는 디코딩된 비디오를 출력하는 수단을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 35

제 34 항에 있어서,

상기 제 1 SPS 는 상기 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함하고, 상기 제 2 SPS 는 상기 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 36

제 34 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들에 대한 상기 값들이 디폴트 값들임을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 37

제 34 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들이 상기 비디오 데이터로 시그널링됨을 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 38

제 37 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 39

제 37 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트가 일정한지 또는 가변적인지를 나타내는 선택스 엘리먼트를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 40

제 37 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 코딩된 화상 버퍼 관리를 위한 하나 이상의 선택스 엘리먼트들을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 41

제 37 항에 있어서,

상기 비디오 데이터는 2 개 이상의 시간 계층들을 포함하고,

상기 디바이스는,

상기 VPS 에서, 상기 2 개 이상의 시간 계층들의 각 개별의 시간 계층에 대해, 상기 개별의 시간 계층에 대한 HRD 파라미터들의 개별의 세트에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 수신하는 수단을 더 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 42

명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 때, 상기 하나 이상의 프로세서들로 하여금,

비디오 데이터에서, 가상 참조 디코더 (HRD) 파라미터들에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 포함하는 비디오 파라미터 세트 (VPS) 를 수신하며;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트 (SPS) 를 수신하며;

상기 비디오 데이터에서, 상기 VPS 를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 SPS 를 수신하며;

상기 하나 이상의 선택스 엘리먼트들에 기초하여, 상기 제 1 SPS 와 연관된 비디오 블록들의 제 1 세트 및 상기 제 2 SPS 와 연관된 비디오 블록들의 제 2 세트를 디코딩하며; 그리고

디코딩된 상기 비디오 블록들의 제 1 세트 및 디코딩된 상기 비디오 블록들의 제 2 세트를 포함하는 디코딩된 비디오를 출력하도록 하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 43

제 42 항에 있어서,

상기 제 1 SPS 는 상기 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함하고, 상기 제 2 SPS 는 상기 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 44

제 42 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들에 대한 상기 값들이 디폴트 값들임을 나타내는 선택스 엘리먼트를 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 45

제 42 항에 있어서,

상기 HRD 파라미터들에 대한 상기 값들을 식별하는 상기 하나 이상의 선택스 엘리먼트들은, 상기 HRD 파라미터들이 상기 비디오 데이터로 시그널링됨을 나타내는 선택스 엘리먼트를 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 46

제 45 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트를 나타내는 선택스 엘리먼트를 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 47

제 45 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 상기 비디오 데이터에 대한 비트레이트가 일정한지 또는 가변적인지를 나타내는 선택스 엘리먼트를 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 48

제 45 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들은 코딩된 화상 버퍼 관리를 위한 하나 이상의 선택스 엘리먼트들을 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 49

제 45 항에 있어서,

상기 비디오 데이터는 2 개 이상의 시간 계층들을 포함하고,

상기 비일시적 컴퓨터 판독가능 저장 매체는, 실행될 때, 상기 하나 이상의 프로세서들로 하여금,

상기 VPS 에서, 상기 2 개 이상의 시간 계층들의 각 개별의 시간 계층에 대해, 상기 개별의 시간 계층에 대한 HRD 파라미터들의 개별의 세트에 대한 값들을 식별하는 하나 이상의 선택스 엘리먼트들을 수신하도록 하는 추가의 명령들을 저장하는, 비일시적 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2012년 7월 2일자로 출원된 미국 가출원 제61/667,387호; 2012년 7월 9일자로 출원된 미국 가출원 제61/669,587호; 2013년 3월 15일자로 출원된 미국 가출원 제61/798,135호에 관련되며, 그것들의 각각의 전체 내용은 본원에 참조로 통합된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 데이터의 프로세싱에 관한 것이며, 더 상세하게는, 본 개시물은 비디오 데이터에 대한 파라미터 세트들을 생성하고 프로세싱하는 것에 관련된 기법들을 설명한다.

배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인휴대 정보단말들 (PDA들), 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들,

디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 다양한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (AVC) 에 의해 규정된 표준들, 현재 개발중인 고효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장들에 기재된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현하는 것에 의해 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 공간적 (화상 내 (intra-picture)) 예측 및/또는 시간적 (화상 간 (inter-picture)) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 제거한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분) 는 비디오 블록들로 구획될 수도 있으며, 그 비디오 블록들은 또한 트리블록들 (treeblocks), 코딩 단위들 (CU들) 및/또는 코딩 노드들이라고 지칭될 수도 있다. 화상의 인트라 코딩된 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측을 이용하여 인코딩된다. 화상의 인터 코딩된 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 화상들에서의 참조 샘플들에 관한 시간적 예측을 이용할 수도 있다. 화상들은 프레임들이라고 지칭될 수도 있고, 참조 화상들은 참조 프레임들이라고 지칭될 수도 있다.

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

발명의 내용

과제의 해결 수단

[0007] 본 개시물은 비디오 코딩에서 파라미터 세트들에 대한 설계 기법들을 설명하고, 더 상세하게는, 본 개시물은 비디오 파라미터 세트 (video parameter set; VPS) 들에 관련된 기법들을 설명한다.

[0008] 하나의 예에서, 비디오 데이터를 디코딩하는 방법은, 비디오 파라미터 세트에서, 가상 참조 디코더 (hypothetical reference decoder; HRD) 파라미터들에 관련된 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 수신하는 단계; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 1 신택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 수신하는 단계; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 2 신택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 수신하는 단계; 및, 하나 이상의 신택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 코딩하는 단계를 포함한다.

[0009] 다른 예에서, 비디오 데이터를 인코딩하는 방법은, 비디오 파라미터 세트 내의 포함을 위해, 가상 참조 디코더 (HRD) 파라미터들에 관련된 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 생성하는 단계; 비디오 데이터 내의 포함을 위해, 비디오 파라미터 세트를 식별하는 제 1 신택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 생성하는 단계; 비디오 데이터 내의 포함을 위해, 비디오 파라미터 세트를 식별하는 제 2 신택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 생성하는 단계; 및, 하나 이상의 신택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 코딩하는 단계를 포함한다.

[0010] 다른 예에서, 비디오 데이터를 프로세싱하는 디바이스는, 비디오 파라미터 세트에서, 가상 참조 디코더 (HRD) 파라미터들에 관련된 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 수신하며; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 1 신택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 수신하며; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 2 신택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 수신하며; 그리고, 하나 이상의 신택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제

1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 코딩하도록 구성된 비디오 디코더를 포함한다.

[0011] 다른 예에서, 비디오 데이터를 프로세싱하는 디바이스는, 비디오 파라미터 세트 내의 포함을 위해, 가상 참조 디코더 (HRD) 파라미터들에 관련된 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 생성하며; 비디오 데이터 내의 포함을 위해, 비디오 파라미터 세트를 식별하는 제 1 신택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 생성하며; 비디오 데이터 내의 포함을 위해, 비디오 파라미터 세트를 식별하는 제 2 신택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 생성하며; 그리고, 하나 이상의 신택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 코딩하도록 구성된 비디오 인코더를 포함한다.

[0012] 다른 예에서, 비디오 데이터를 프로세싱하는 디바이스는, 비디오 파라미터 세트에서, 가상 참조 디코더 (HRD) 파라미터들에 관련된 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 수신하는 수단; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 1 신택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 수신하는 수단; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 2 신택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 수신하는 수단; 및, 하나 이상의 신택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 코딩하는 수단을 포함한다.

[0013] 다른 예에서, 컴퓨터 판독가능 저장 매체는, 하나 이상의 프로세서들에 의해 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 파라미터 세트에서, 가상 참조 디코더 (HRD) 파라미터들에 관련된 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 수신하며; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 1 신택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 수신하며; 비디오 데이터에서, 비디오 파라미터 세트를 식별하는 제 2 신택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 수신하며; 하나 이상의 신택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 코딩하도록 하는 명령들을 저장한다.

[0014] 하나 이상의 예들의 세부사항들은 첨부 도면들 및 다음의 설명에서 언급된다. 다른 특징들, 목적들, 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

[0015] 도 1은 본 개시물에서 설명되는 기법들을 이용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 도시하는 블록도이다.

도 2는 일 예의 MVC 디코딩 순서를 도시하는 개념도이다.

도 3은 일 예의 MVC 시간적 및 뷰간 예측 구조를 도시하는 개념도이다.

도 4는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 도시하는 블록도이다.

도 5는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 도시하는 블록도이다.

도 6은 네트워크의 일부를 형성하는 디바이스들의 일 예의 세트를 도시하는 블록도이다.

도 7은 본 개시물의 기법들에 따라 파라미터 세트를 프로세싱하는 일 예의 방법을 도시하는 흐름도이다.

도 8은 본 개시물의 기법들에 따라 파라미터 세트를 생성하는 일 예의 방법을 도시하는 흐름도이다.

도 9는 본 개시물의 기법들에 따라 파라미터 세트를 디코딩하는 일 예의 방법을 도시하는 흐름도이다.

도 10은 본 개시물의 기법들에 따라 파라미터 세트를 프로세싱하는 일 예의 방법을 도시하는 흐름도이다.

도 11은 본 개시물의 기법들에 따라 파라미터 세트를 생성하는 일 예의 방법을 도시하는 흐름도이다.

도 12는 본 개시물의 기법들에 따라 파라미터 세트를 프로세싱하는 일 예의 방법을 도시하는 흐름도이다.

도 13은 본 개시물의 기법들에 따라 파라미터 세트를 생성하는 일 예의 방법을 도시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 개시물은 비디오 코딩에서 파라미터 세트들에 대한 설계 기법들을 설명하고, 더 상세하게는, 본 개시물은 비디오 파라미터 세트 (video parameter set; VPS) 들에 관련된 기법들을 설명한다. VPS들에 더하여, 파라미

터 세트들의 다른 예들은, 몇몇 이름을 대자면, 시퀀스 파라미터 세트 (sequence parameter set; SPS) 들, 화상 파라미터 세트 (picture parameter set; PPS) 들, 및 적응 파라미터 세트 (adaptation parameter set; APS) 들을 포함한다.

[0017] 비디오 인코더가 비디오 데이터를 인코딩한다. 비디오 데이터는 하나 이상의 화상들을 포함할 수도 있는데, 화상들의 각각은 비디오의 일부를 형성하는 정지 이미지이다. 비디오 인코더가 비디오 데이터를 인코딩하는 경우, 비디오 인코더는 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 생성한다. 그 비트스트림은 코딩된 화상들과, 연관된 데이터를 포함할 수도 있는데, 코딩된 화상은 화상의 코딩된 표현을 지칭한다. 연관된 데이터는 VPS들, SPS들, PPS들, 및 APS들, 및 잠재적으로 다른 선택스 구조들을 포함하는 다양한 유형들의 파라미터 세트들을 포함할 수도 있다. SPS들은 전체 비디오 시퀀스에 유효한 데이터를 운반하는데 사용되는 반면, PPS들은 화상 단위로 유효한 정보를 운반한다. APS들은, 화상 단위로도 유효하지만 PPS에서의 정보보다는 더 빈번하게 변화할 것으로 예상되는 화상 적응 정보를 운반한다.

[0018] HEVC는 HEVC 규격 초안 (working draft) 이 다음과 같이 기술하는 VPS를 또한 도입했다:

[0019] 비디오 파라미터 세트 (VPS): seq_parameter_set_id 선택스 엘리먼트에 의해 참조되는 시퀀스 파라미터 세트에서 발견된 video_parameter_set_id 선택스 엘리먼트의 콘텐츠에 의해 결정된 바와 같은 0 또는 그 이상의 전체 코딩된 비디오 시퀀스들에 적용하는 선택스 엘리먼트들을 포함하는 선택스 구조인데, seq_parameter_set_id 선택스 엘리먼트는 각각의 슬라이스 세그먼트 헤더에서 발견되는 pic_parameter_set_id 선택스 엘리먼트에 의해 참조되는 화상 파라미터 세트에서 발견된다.

[0020] 따라서, VPS들이 전체 코딩된 비디오 시퀀스들에 적용되므로, VPS는 드물게 변화하는 선택스 엘리먼트들을 포함한다. HEVC의 일부 버전들에서의 VPS, SPS, PPS, 및 APS 메커니즘은 코딩된 비디오 블록 데이터의 송신으로부터 드물게 변화하는 정보의 송신을 분리시킨다. VPS들, SPS들, PPS들, 및 APS들은, 일부 애플리케이션들에서, "대역 외 (out-of-band)" 전달될 수도 있으며 즉, 코딩된 비디오 데이터를 포함하는 단위들과 함께 전송되지 않을 수도 있다. 대역 외 송신은 일반적으로 신뢰할 수 있고, 대역 내 (in-band) 송신에 비하여 개선된 신뢰도를 위해 바람직할 수도 있다. HEVC WD7에서, VPS, SPS, PPS, 또는 APS의 식별자 (ID) 가 각각의 파라미터 세트를 위해 코딩될 수도 있다. 각각의 SPS는 SPS ID와 VPS ID를 포함하며, 각각의 PPS는 PPS ID와 SPS ID를 포함하고, 각각의 슬라이스 헤더는 PPS ID와 가능하다면 APS ID를 포함한다. 이런 식으로, ID 들은 상이한 경우들에서 사용될 적절한 파라미터 세트를 식별하는데 사용될 수 있다.

[0021] 위에서 도입된 바와 같이, 비디오 인코더들은 통상적으로 비디오 데이터를 인코딩하고, 디코더들은 통상적으로 비디오 데이터를 디코딩한다. 인코더들과 디코더들은, 그러나, 비디오 데이터를 프로세싱하는데 사용되는 디바이스들에 불과한 것은 아니다. 비디오가, 예를 들어, 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷 같은 글로벌 네트워크와 같은 패킷 기반 네트워크의 일부로서 전송되는 경우, 라우팅 디바이스들과 다른 이러한 디바이스들은 비디오 데이터를 프로세싱하여, 그 비디오 데이터를 소스로부터 목적지 디바이스로 전달할 수도 있다. 때때로 미디어 인식 네트워크 엘리먼트 (media aware network element; MANE) 들이라고 지칭되는 특수한 라우팅 디바이스들이, 비디오 데이터의 콘텐츠에 기초하여 다양한 라우팅 기능들을 수행할 수도 있다. 비디오 데이터의 콘텐츠를 결정하고 이들 라우팅 기능들을 수행하기 위해, MANE는 인코딩된 비트스트림에서의 정보, 이를테면 VPS 또는 SPS에서의 정보에 액세스할 수도 있다.

[0022] 파라미터 세트에서, 일부 선택스 엘리먼트들은 고정된 수의 비트들을 사용하여 코딩되는 반면, 일부 선택스 엘리먼트들은 가변적인 수의 비트들을 사용하여 코딩된다. 가변 길이의 선택스 엘리먼트들을 프로세싱하기 위하여, 디바이스가 엔트로피 디코딩 능력들을 필요로 할 수도 있다. 엔트로피 디코딩을 수행하는 것은, 그러나, MANE 또는 다른 네트워크 엘리먼트들에 대해 바람직하지 않은 소정 레벨의 복잡도를 도입할 수도 있다. 본 개시물에서 도입된 하나의 기법에 따르면, 임의의 엔트로피 디코딩 없이 디코딩될 수 있는 선택스 엘리먼트들을 식별함에 있어서 네트워크 엘리먼트들을 지원하기 위하여 오프셋 선택스 엘리먼트가 파라미터 세트, 이를테면 VPS에 포함될 수 있다. 오프셋 선택스 엘리먼트에는 고정 길이 선택스 엘리먼트들이 선행할 수도 있다. 오프셋 선택스 엘리먼트는 그러면 가변 길이 선택스 엘리먼트들을 사용하여 코딩될 선택스 엘리먼트들을 파라미터 세트에서 식별할 수도 있다. 오프셋 선택스 엘리먼트를 사용하여, 디바이스, 이를테면 MANE 는 가변 길이 코딩된 선택스 엘리먼트들을 스킵하고 고정 길이 선택스 엘리먼트들의 프로세싱을 재개할 수도 있다. 오프셋 선택스 엘리먼트는 스킵될 다수의 바이트들을 파라미터 세트 내에서 식별함으로써 스킵될 선택스 엘리먼트들을 식별할 수도 있다. 이들 스킵된 바이트들은 스킵된 선택스 엘리먼트들에 대응할 수도 있다. 위에서 언급했듯이, 스킵된 선택스 엘리먼트들은 가변 길이 코딩된 선택스 엘리먼트들을 포함할 수도

있고 또한 고정 길이 코딩된 선택스 엘리먼트들을 포함할 수도 있다.

[0023] 이 맥락에서, 선택스 엘리먼트들을 스킵하는 것은 MANE가 가변 길이들로 코딩되는 선택스 엘리먼트들의 파싱 또는 다른 프로세싱을 회피할 수도 있다는 것을 의미한다. 따라서, MANE는, 그렇지 않으면 엔트로피 디코딩을 필요로 할 수도 있는 일부 선택스 엘리먼트들을 스킵하면서, 엔트로피 디코딩을 수행하지 않고 VPS에서 일부 선택스 엘리먼트들 (예컨대, 고정 길이 엘리먼트들) 을 프로세싱할 수 있다. 일부 고정 길이 선택스 엘리먼트들이 또한 다양한 예들에서 스킵될 수도 있으므로, MANE에 의해 스킵된 선택스 엘리먼트들은 가변 길이 선택스 엘리먼트들로 제한되지 않는다. 비디오 디코더는, 오프셋 선택스 엘리먼트를 수신 시, 선택스 엘리먼트들 중 하나 이상을 기본적으로 무시하도록 구성될 수도 있는데, 이는 비디오 디코더가 MANE에 의해 스킵되었던 선택스 엘리먼트들의 파싱과 프로세싱을 회피할 수도 있다는 것을 의미한다.

[0024] 오프셋 선택스 엘리먼트의 사용은, 예컨대, MANE가 엔트로피 디코딩을 수행할 필요를 없앴으로써 파라미터 세트의 부분들을 프로세싱하기 위해 MANE에 필요한 복잡도를 감소시킬 수도 있다. 덧붙여, 오프셋 선택스 엘리먼트의 사용은, 본 개시물에서 제안한 바와 같이, 파라미터 세트들에 대한 계층 포맷의 사용을 가능하게 할 수도 있다. 계층 포맷의 일 예로서, VPS에서는, VPS 내에서 혼합된 기본 계층 및 향상 계층 (enhancement layer) 에 대한 선택스 엘리먼트들을 가지는 대신, 기본 계층의 모든 또는 실질적으로 모든 선택스 엘리먼트들이 제 1 향상 계층의 모든 또는 실질적으로 모든 선택스 엘리먼트들에 선행할 수도 있으며, 결국 제 1 향상 계층의 모든 또는 실질적으로 모든 선택스 엘리먼트들이 제 2 향상 계층에 대한 모든 또는 실질적으로 모든 선택스 엘리먼트들에 선행할 수도 있으며, 등등이다. 본 개시물에서 도입된 오프셋 선택스 엘리먼트를 사용하여, MANE는 기본 계층에 대한 다수의 고정 길이 선택스 엘리먼트들을 프로세싱하며, 기본 계층에 대한 다수의 가변 길이 선택스 엘리먼트들을 스킵하며, 제 1 향상 계층에 대한 다수의 고정 길이 선택스 엘리먼트들을 프로세싱하며, 제 1 향상 계층에 대한 다수의 가변 길이 선택스 엘리먼트들을 스킵하며, 제 2 향상 계층에 대한 다수의 고정 길이 선택스 엘리먼트들을 스킵하며, 등등을 할 수도 있다. 비디오 디코더는 MANE에 의해 스킵된 선택스 엘리먼트들을 파싱하고 프로세싱하도록 구성될 수도 있다.

[0025] 오프셋 선택스 엘리먼트의 사용은 비디오 코딩 표준에 대한 미래의 확장들을 부가적으로 가능하게 할 수도 있다. 예를 들어, 다른 유형들의 가변 길이 코딩된 정보가 (예컨대, HEVC에 대한 미래의 확장에 따라) 비트스트림에 추가되었다 하더라도, 하나 이상의 오프셋 선택스 엘리먼트들은 이러한 가변 길이 엘리먼트들의 스킵을 용이하게 하기 위해 정의될 수도 있다. 다르게 말하면, 하나 이상의 오프셋 선택스 엘리먼트들은 비트스트림 내에서 고정 길이 선택스 엘리먼트들의 로케이션을 식별하는데 사용될 수 있고, 오프셋 선택스 엘리먼트들은, 예컨대, MANE에 의해 디코딩이 회피될 수도 있는 임의의 다른 엘리먼트들의 비트스트림으로의 추가를 설명하기 위해 수정될 수도 있다.

[0026] 본 개시물은 세션 협상에 관련된 선택스 엘리먼트들을, 다른 파라미터 세트, 이를테면 SPS 내애가 아니라, 비디오 파라미터 세트 내에 포함시키는 것을 추가적으로 제안한다. 세션 협상에 관련된 선택스 엘리먼트들을 VPS에 포함시킴으로써, 특히 VPS가 비디오의 단지 단일 계층에 대한 정보가 아니라 다수의 계층들에 대한 정보를 기술하는 경우에 시그널링 오버헤드가 감소되는 것이 가능할 수도 있다. 더구나, 본 개시물은 세션 협상 선택스 엘리먼트들을 위해 고정 길이 선택스 엘리먼트들을 사용하는 것을 제안하고, 고정 길이 세션 협상 선택스 엘리먼트들은 임의의 가변 길이 선택스 엘리먼트들 앞에 위치될 수 있다. 가변 길이의 선택스 엘리먼트들을 프로세싱하기 위하여, 엔트로피 디코딩을 수행할 수 있는 디바이스가 필요하다. 엔트로피 디코딩을 수행하는 것은, 그러나, MANE에 대해 바람직하지 않은 소정 레벨의 복잡도를 도입할 수도 있다. 따라서, VPS에서 임의의 가변 길이 선택스 엘리먼트들 앞에 존재하는 고정 길이 선택스 엘리먼트들을 사용함으로써, MANE는 엔트로피 디코딩을 수행하지 않고 세션 협상을 위해 선택스 엘리먼트들을 파싱하는 것이 가능할 수도 있다.

[0027] 아래의 표 2는 VPS에 포함될 수도 있는 세션 협상 관련된 선택스 엘리먼트들의 예들을 보여준다. 세션 협상을 위한 정보의 예들은 프로파일들, 티어 (tier) 들 및 레벨들을 식별하는 정보를 포함한다. HEVC 규격 초안은 프로파일들, 티어들, 및 레벨들을 다음과 같이 기술한다:

[0028] "프로파일"은 이 권고안(국제 표준에 의해 특정되는 전체 비트스트림 선택스의 서브세트이다. 주어진 프로파일의 선택스에 의해 부과되는 경계들 내에서, 디코딩된 화상들의 특정된 사이즈와 같이 비트스트림에서 선택스 엘리먼트들에 의해 취해진 값들에 의존하여 인코더들 및 디코더들의 성능에서 매우 큰 변화를 요구하는 것이 여전히 가능하다. 많은 애플리케이션들에서, 특정 프로파일 내에서 선택스의 모든 가정적 사용 (hypothetical use) 들을 다룰 수 있는 디코더를 구현하는 것은 현재 실용적이지도 않고 경제적이지도 않다.

[0029] 이 문제를 다루기 위하여, "티어들"과 "레벨들"이 각각의 프로파일 내에서 특정된다. 티어의 레벨

은 비트스트림에서 선택스 엘리먼트들의 값들에 부과되는 특정된 제약들의 세트이다. 이들 제약들은 값들에 대한 단순한 제한들일 수도 있다. 다르게는, 그것들은 값들의 산술적 결합들 (예컨대, 화상 폭 곱하기 화상 높이 곱하기 초당 디코딩되는 화상들의 수) 에 대한 제약들의 형태를 취할 수도 있다. 더 낮은 티어에 특정된 레벨은 더 높은 티어에 특정된 레벨보다 더 제약된다.

[0030] 클라이언트와 MANE 사이의 세션 협상 동안, 클라이언트는 특정된 프로파일, 레벨, 및/또는 티어에 따라 코딩된 비디오 데이터의 MANE에서의 가용성에 관해 물어볼 수도 있다. MANE는 프로파일, 레벨, 및 티어 정보를 포함하는 VPS의 제 1 부분 (즉 고정 길이 코딩된 부분) 을 파싱하는 것이 가능할 수도 있다. MANE에서 이용가능한 동작 포인트들 중에서, 적절한 것이 클라이언트에 의해 선택될 수 있고, MANE는 세션이 협상된 후에 대응하는 패키지들을 클라이언트로 포워딩할 수 있다.

[0031] 본 개시물은 가상 참조 디코더 (HRD) 를 식별하기 위한 선택스 엘리먼트들을, 다른 파라미터 세트, 이를테면 SPS 내애가 아니라, 비디오 파라미터 세트 내에 포함시키는 것을 부가적으로 제안한다. HRD 파라미터들은 인코딩 프로세스가 생성할 수도 있는 순응하는 NAL 단위 스트림들 또는 순응하는 바이트 스트림들의 변동성 (variability) 에 대한 제약들을 특정하는 가상 디코더 모델을 식별한다. 2 개의 유형들의 HRD 파라미터 세트들 (NAL HRD 파라미터들 및 VCL HRD 파라미터들) 이 VPS에 포함될 수도 있다. NAL HRD 파라미터들은 유형 II 비트스트림 적합성 (conformance) 에 관련되는 반면, VCL HRD 파라미터들은 모든 비트 스트림 적합성에 관련된다. HEVC는 HRD 적합성의 적용을 받는 비트스트림의 2 개의 유형들 사이를 현재 구별했다. 첫째는 유형 I 비트스트림이라 지칭되고 비트스트림에서의 모든 액세스 단위들을 위해 VCL NAL 단위들 및 충전 데이터 NAL 단위들만을 포함하는 NAL 단위 스트림을 말한다. 제 2 유형의 비트스트림은 유형 II 비트스트림이라 지칭되고 비트스트림에서의 모든 액세스 단위들을 위해 VCL NAL 단위들 및 충전 데이터 NAL 단위들에 더하여 다른 유형들의 부가적인 NAL 단위들을 포함한다.

[0032] 본 개시물의 기법들은 단일 계층 코딩에뿐만 아니라 스케일러블 및 멀티뷰 비디오 코딩에 적용될 수 있다. 계층은, 예를 들어, 공간적 스케일러블 계층, 품질 스케일러블 계층, 텍스처 뷰, 또는 깊이 뷰일 수도 있다. HEVC에서, 계층은 일 세트의 비디오 코딩 계층 (video coding layer; VCL) NAL 단위들 및 연관된 비-VCL NAL 단위들을 일반적으로 말하는데, 그것들은 모두가 특정 계층 ID 값을 가진다. 계층들은 제 1 계층이 더 낮은 계층을 포함할 수도 있다는 의미에서 계층적일 수 있다. 계층 세트는 서브-비트스트림 추출 프로세스의 동작에 의해 다른 비트스트림으로부터 생성된 비트스트림 내에서 표현되는 계층들의 세트를 말하기 위해 때때로 사용된다. 동작 포인트는 다른 비트스트림을 이용한 서브-비트스트림 추출 프로세스의 동작에 의해 다른 비트스트림으로부터 생성된 비트스트림을 일반적으로 말한다. 동작 포인트는 계층 세트에서의 모든 계층들을 포함할 수도 있거나 또는 계층 세트의 서브세트로서 형성된 비트스트림일 수도 있다.

[0033] 도 1은 본 개시물에서 설명되는 기법들을 이용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템 (10) 을 도시하는 블록도이다. 도 1에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12) 를 포함한다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 미디어 인식 네트워크 엘리먼트 (MANE) (29) 에 의해 라우팅될 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 전화기 핸드셋들, 이른바 "스마트" 폰들, 이른바 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 다양한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 무선 통신을 위해 장비될 수도 있다.

[0034] 시스템 (10) 은 상이한 비디오 코딩 표준들, 독점 표준, 또는 멀티뷰 코딩의 임의의 다른 방식에 따라 동작할 수도 있다. 예를 들어, 비디오 인코더 (20) 와 비디오 디코더 (30) 는, 그 SVC (Scalable Video Coding) 및 MVC (Multiview Video Coding) 확장들을 포함하는, 이를테면, ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual 및 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC로 알려짐) 를 포함하는 비디오 압축 표준에 따라 동작할 수도 있다. MVC 확장의 최근, 공개적으로 입수가능한 공동 초안이 『"Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264, Mar 2010』 에 기재되어 있다. MVC 확장의 더 최근의, 공개적으로 입수가능한 공동 초안이 『"Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264, June 2011』 에 기재되어 있다. MVC 확장의 현재 공동 초안은 2012년 1월 현재 승인되었다.

[0035] 덧붙여서, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 및 ISO/IEC 동 화상 전문가 그룹 (MPEG) 의 JCT-VC (Joint

Collaboration Team on Video Coding)에 의해 현재 개발 중인 새로운 비디오 코딩 표준, 즉 고 효율 비디오 코딩 (HEVC) 표준이 있다. HEVC의 최근의 규격 초안 (WD) 이고 이후로 HEVC WD7이라고 지칭하는 것은, 2013년 7월 1일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I1003-v6.zip에서 입수가가능하다.

[0036] HEVC 표준의 개발은 진행중이고, HEVC의 더 새로운 규격 초안 (WD) 이고 HEVC WD9라고 지칭되는 것이, 2013년 7월 1일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v10.zip에서 입수가가능하다. 설명의 목적으로, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 HEVC 또는 H.264 표준 및 이러한 표준들의 확장들의 맥락에서 설명되고 있다. 본 개시물의 기법들은, 그러나, 임의의 특정 코딩 표준으로 제한되지 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2와 ITU-T H.263을 포함한다. 독점 코딩 기법들, 이를테면 On2 VP6/VP7/VP8이라고 지칭되는 것들이, 본원에서 설명되는 기법들 중 하나 이상을 또한 구현할 수도 있다. "HEVC 규격 초안 10" 또는 "HEVC WD10"이라고 지칭되는 근간의 HEVC 표준의 더 새로운 초안은 『Bross et al., "Editors' proposed corrections to HEVC version 1," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 13th Meeting, Incheon, KR, April 2013』에 기재되어 있으며, 그것은 2013년 7월 1일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip에서 입수가가능하며, 그 전체 내용은 본원에 참조로 통합된다.

[0037] 본 개시물의 기법들은 HEVC 기반 3D-비디오 코딩 (3D-HEVC) 을 포함하는 여러 MVC 및/또는 3D 비디오 코딩 표준들에 잠재적으로 적용가능하다. 본 개시물의 기법들은 H.264/3D-AVC 및 H.264/MVC+D 비디오 코딩 표준들, 또는 그 확장들, 뿐만 아니라 다른 코딩 표준들에 또한 적용가능할 수도 있다. 본 개시물의 기법들은 가끔은 특정 비디오 코딩 표준의 기술용어를 참조하여 또는 사용하여 설명될 수도 있지만, 이러한 설명은 설명된 기법들이 그 특정 표준에만 제한된다는 의미로 해석되어서는 안된다.

[0038] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 링크 (16) 를 통해 수신할 수도 있다. 링크 (16) 는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 이동시킬 수 있는 임의의 유형의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 링크 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접 실시간으로 송신하는 것을 가능하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 통신 표준, 이를테면 무선 통신 프로토콜에 따라 변조되고 목적지 디바이스 (14) 에 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 이를테면 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다. 링크 (16) 는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 비디오 데이터를 라우팅하는 하나 이상의 MANE들, 이를테면 MANE (29) 를 포함할 수도 있다.

[0039] 대안으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (27) 로 출력될 수도 있다. 마찬가지로, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (27) 로부터 액세스될 수도 있다. 저장 디바이스 (27) 는 하드 드라이브, 블루 레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산형 또는 국소적으로 액세스되는 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다. 추가의 예에서, 저장 디바이스 (27) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 유지할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 저장 디바이스 (27) 로부터의 저장된 비디오 데이터에 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 에 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, 네트워크 부속 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 데이터에 액세스할 수도 있다. 이는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽 모두의 조합을 포함할 수도 있다. 저장 디바이스 (27) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다. 저장 디바이스 (27) 로부터 추출된 비디오 데이터는 하나 이상의 MANE들, 이를테면 MANE (29) 를 사

용하여 목적지 디바이스 (14) 로 라우팅될 수도 있다.

[0040] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 반드시 제한되지는 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들 중 임의의 것의 지원 하의 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0041] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22) 를 구비한다. 비디오 인코더 (20) 는, 예를 들어, 본 개시물에서 설명되는 오프셋 선택스를 생성할 수도 있다. 일부 경우들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 구비할 수도 있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오를 담고 있는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 피드 인터페이스, 및/또는 컴퓨터 그래픽 데이터를 소스 비디오로서 생성하는 컴퓨터 그래픽 시스템과 같은 소스, 또는 그런 소스들의 조합을 포함할 수도 있다. 하나의 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 이른바 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명되는 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0042] 캡처된, 사전-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (12) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (20) 의 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안으로) 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 디코딩 및/또는 플레이백을 위한 나중의 액세스를 위해 저장 디바이스 (27) 에 저장될 수도 있다.

[0043] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 구비한다. 비디오 디코더 (30) 는 본 개시물에서 설명되는 오프셋 선택스 엘리먼트를 파싱할 수도 있다. 위에서 설명된 바와 같이, 비디오 디코더 (30) 는 일부 경우들에서 오프셋 선택스 엘리먼트를 무시하며, 따라서 비디오 디코더 (30) 가 MANE에 의해 스킵된 선택스 엘리먼트들을 파싱하는 것을 가능하게 할 수도 있다. 일부 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 구비할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16) 를 통해 통신되거나 또는 저장 디바이스 (27) 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터의 디코딩 시에, 비디오 디코더, 이를테면 비디오 디코더 (30) 에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 선택스 엘리먼트들을 포함할 수도 있다. 이러한 선택스 엘리먼트들은 통신 매체 상에서 송신되는, 저장 매체 상에 저장되는, 또는 파일 서버에 저장되는 인코딩된 비디오 데이터에 포함될 수도 있다.

[0044] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나, 또는 그것 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는 통합형 디스플레이 디바이스를 포함할 수도 있고 또한 외부 디스플레이 디바이스와 인터페이싱하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0045] 비록 도 1에 도시되지 않았지만, 일부 양태들에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함하여, 공통 데이터 스트림 또는 개별 데이터 스트림들에서의 오디오 및 비디오 양쪽 모두의 인코딩을 핸들링할 수도 있다. 적용가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜들에 순응할 수도 있다.

[0046] 비디오 인코더 (20) 와 비디오 디코더 (30) 각각은 다양한 적합한 인코더 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 개별 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그것들의 임의의 조합 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행

하기 위해, 적합한 비일시적 컴퓨터 판독가능 매체 내에 소프트웨어에 대한 명령들을 저장하고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC) 의 일부로서 개별 디바이스 내에 통합될 수도 있다.

[0047] JCT-VC는 HEVC 표준의 개발에 노력하고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 이라고 지칭되는 비디오 코딩 디바이스의 진화 모델에 기초하고 있다. HM은, 예컨대, ITU-T H.264/AVC에 따른 기존 디바이스들에 비해 비디오 코딩 디바이스들의 몇몇 부가적인 능력들을 상정한다. 예를 들어, H.264가 9 개의 인트라 예측 인코딩 모드들을 제공하는 반면, HM은 33 개 정도의 인트라 예측 인코딩 모드들을 제공할 수도 있다.

[0048] 일반적으로, HM의 작업 모델은 비디오 프레임 또는 화상이 루마 및 크로마 샘플들 양쪽 모두를 포함하는 트리블록들 또는 최대 코딩 단위 (largest coding unit; LCU) 들의 시퀀스로 분할될 수도 있다는 것을 설명한다. 트리블록은 H.264 표준의 매크로블록과 유사한 목적을 가진다. 슬라이스는 코딩 순서에서 연속적인 다수의 트리블록들을 포함한다. 비디오 프레임 또는 화상은 하나 이상의 슬라이스들로 구획될 수도 있다. 각각의 트리블록은 쿼드트리에 따라 코딩 단위들 (CU들) 로 분할될 수도 있다. 예를 들어, 쿼드트리의 루트 노드인 트리블록은 4 개의 자식 노드들로 분할될 수도 있고, 각각의 자식 노드는 다시 부모 노드가 될 수도 있고 다른 4 개의 자식 노드들로 분할될 수도 있다. 쿼드트리의 리프 노드인, 최종의 분할되지 않는 자식 노드는 코딩 노드, 즉 코딩된 비디오 블록을 포함한다. 코딩된 비트스트림에 연관된 신택스 데이터는 트리블록이 분할될 수도 있는 최대 횡수를 정의할 수도 있고, 또한 코딩 노드들의 최소 사이즈를 정의할 수도 있다.

[0049] CU는 코딩 노드와 그 코딩 노드에 연관된 예측 단위들 (PU들) 및 변환 단위들 (TU들) 을 포함한다. CU의 사이즈는 코딩 노드의 사이즈에 대응하고 형상이 정사각형이어야만 한다. CU의 사이즈는 8x8 화소들로부터 최대 64x64 화소들 또는 그 이상을 갖는 트리블록의 사이즈까지의 범위일 수도 있다. 각각의 CU는 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다. CU에 연관된 신택스 데이터는, 예를 들어, 하나 이상의 PU들로의 CU의 구획화를 기술할 수도 있다. 구획화 모드들은 CU가 스킵되는지 또는 직접 모드 인코딩되는지, 인트라 예측 모드 인코딩되는지, 또는 인터 예측 모드 인코딩되는지의 사이에서 상이할 수도 있다. PU들은 형상이 정사각형이 아니도록 구획될 수도 있다. CU에 연관된 신택스 데이터는 또한, 예를 들어 쿼드트리에 따른 하나 이상의 TU들로의 CU의 구획화를 기술할 수도 있다. TU는 형상이 정사각형이거나 또는 정사각형이 아닐 수 있다.

[0050] HEVC 표준은 TU들에 따른 변환들을 허용하는데, 이 변환들은 상이한 CU들에 대해 상이할 수도 있다. TU들은, 항상 그런 것은 아닐 수도 있지만, 구획된 LCU에 대해 정의된 주어진 CU 내의 PU들의 사이즈에 기초하여 통상 사이즈가 정해진다. TU들은 통상 PU들과 동일한 사이즈이거나 또는 그것들보다 작다. 일부 예들에서, CU에 대응하는 잔차 샘플들은 "잔차 쿼드 트리 (residual quad tree)" (RQT) 로서 알려진 쿼드트리 구조를 이용하여 더 작은 단위들로 세분될 수도 있다. RQT의 리프 노드들은 변환 단위 (transform unit; TU) 들이라고 지칭될 수도 있다. TU들에 연관된 화소 차이 값들은 양자화될 수도 있는 변환 계수들을 생성하기 위해 변환될 수도 있다.

[0051] 일반적으로, PU는 예측 프로세스에 관련된 데이터를 포함한다. 예를 들어, PU가 인트라 모드 인코딩되는 경우, PU는 PU에 대한 인트라 예측 모드를 기술하는 데이터를 포함할 수도 있다. 다른 예로서, PU가 인터 모드 인코딩되는 경우, PU는 PU에 대한 모션 벡터를 정의하는 데이터를 포함할 수도 있다. PU에 대한 모션 벡터를 정의하는 데이터는, 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터에 대한 분해능 (예컨대, 1/4 화소 정밀도 또는 1/8 화소 정밀도), 모션 벡터가 가리키는 참조 화상, 및/또는 모션 벡터에 대한 참조 화상 리스트 (예컨대, 리스트 0, 리스트 1, 또는 리스트 C) 를 기술할 수도 있다.

[0052] 대체로, TU는 변환 및 양자화 프로세스들을 위해 사용된다. 하나 이상의 PU들을 갖는 주어진 CU는 또한 하나 이상의 변환 단위들 (TU들) 을 포함할 수도 있다. 예측에 이어, 비디오 인코더 (20) 는 PU에 대응하는 잔차 값들을 계산할 수도 있다. 그 잔차 값들은 엔트로피 코딩을 위한 직렬화된 (serialized) 변환 계수들을 생성하기 위해 TU들을 사용하여 변환 계수들로 변환되며, 양자화되고, 스캐닝될 수도 있는 화소 차이 값들을 포함한다. 본 개시물은 통상적으로 용어 "비디오 블록"을 CU의 코딩 노드를 지칭하기 위해 사용한다. 일부 특정 경우들에서, 본 개시물은 또한 용어 "비디오 블록"을 트리블록, 즉, 코딩 노드와 PU들 및 TU들을 포함하는 CU 또는 LCU를 지칭하기 위해 사용할 수도 있다.

[0053] 비디오 시퀀스는 통상 일련의 비디오 프레임들 또는 화상들을 포함한다. 화상들의 그룹 (GOP) 은 일반적으로 일련의 하나 이상의 비디오 화상들을 포함한다. GOP는 GOP에 포함된 다수의 화상들을 기술하는 신택스

데이터를 GOP의 헤더, 그 화상들 중 하나 이상의 화상들의 헤더, 또는 다른 곳에 포함할 수도 있다. 화상의 각각의 슬라이스는 개별 슬라이스에 대한 인코딩 모드를 기술하는 슬라이스 선택스 데이터를 포함할 수도 있다.

비디오 인코더 (20) 는 비디오 데이터를 인코딩하기 위하여 개개의 비디오 슬라이스들 내의 비디오 블록들에 대해 통상 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정된 또는 가변하는 사이즈들을 가질 수도 있고, 특정된 코딩 표준에 따라 사이즈를 달리할 수도 있다.

[0054] 일 예로서, HM은 다양한 PU 사이즈들에서 예측을 지원한다. 특정 CU의 사이즈가 $2N \times 2N$ 이라고 가정하면, HM은 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들에서의 인트라 예측과, $2N \times 2N$, $2N \times N$, $N \times 2N$, 또는 $N \times N$ 의 대칭적 PU 사이즈들에서의 인터 예측을 지원한다. HM은 또한 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에서의 인터 예측을 위한 비대칭 구획화를 지원한다. 비대칭 구획화에서, CU의 하나의 방향은 구획되지 않는 반면, 다른 방향은 25% 및 75%로 구획된다. 25% 구획에 대응하는 CU의 부분은 "n" 다음의 "상 (Up)", "하 (Down)", "좌 (Left)", 또는 "우 (Right)"의 표시로 나타내어진다. 따라서, 예를 들어, " $2N \times nU$ "는 상단의 $2N \times 0.5N$ PU 및 하단의 $2N \times 1.5N$ PU로 수평으로 구획되는 $2N \times 2N$ CU를 의미한다.

[0055] 본 개시물에서, " $N \times N$ " 및 " N 곱하기 N ", 예컨대, 16×16 화소들 또는 16 곱하기 16 화소들은 수직 및 수평 치수들의 측면에서 비디오 블록의 화소 치수들을 상호교환적으로 말하는데 이용될 수도 있다. 일반적으로, 16×16 블록은 수직 방향의 16 개 화소들 ($y = 16$) 과 수평 방향의 16 개 화소들 ($x = 16$) 을 가질 것이다. 비슷하게, $N \times N$ 블록은 일반적으로 수직 방향의 N 개 화소들 및 수평 방향의 N 개 화소들을 가지며, 여기서 N 은 음이 아닌 정수 값을 나타낸다. 블록에서의 화소들은 행들 및 열들로 배열될 수도 있다. 더구나, 블록들은 수직 방향에서와 동일한 수의 화소들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, 블록들은 $N \times M$ 개 화소들을 포함할 수도 있으며, 여기서 M 은 N 과 반드시 동일하지는 않다.

[0056] CU의 PU들을 이용하는 인트라 예측 또는 인터 예측 코딩에 이어, 비디오 인코더 (20) 는 CU의 TU들에 대한 잔차 데이터를 계산할 수도 있다. PU들은 공간 도메인 (또한 화소 도메인이라고 지칭됨) 에서의 화소 데이터를 포함할 수도 있고 TU들은, 잔차 비디오 데이터에 대한 변환, 예컨대, 이산 코사인 변환 (DCT), 정수 변환, 웨이트 브릿 변환, 또는 개념적으로 유사한 변환의 적용에 뒤따르는 변환 도메인에서의 계수들을 포함할 수도 있다. 잔차 데이터는 인코딩되지 않은 화상의 화소들 및 PU들에 대응하는 예측 값들 사이의 화소 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU에 대한 잔차 데이터를 포함하는 TU들을 형성한 다음 CU에 대한 변환 계수들을 생성하기 위해 TU들을 변환할 수도 있다.

[0057] 변환 계수들을 생성하는 임의의 변환들에 이어, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 변환 계수들이 그 계수들을 표현하는데 사용된 데이터의 양을 가능한 한 줄이도록 양자화되어서, 추가의 압축을 제공하는 프로세스를 일반적으로 말한다. 양자화 프로세스는 그 계수들의 일부 또는 전체에 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n -비트 값은 양자화 동안에 m -비트 값으로 버림될 (rounded down) 수도 있으며, 여기서 n 은 m 보다 크다.

[0058] 일부 예들에서, 비디오 인코더 (20) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해 미리 정의된 스캔 순서를 활용하여 양자화된 변환 계수들을 스캔할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캔을 수행할 수도 있다. 1차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캔한 후, 비디오 인코더 (20) 는, 예컨대, 콘텍스트 적응 가변 길이 코딩 (context adaptive variable length coding; CAVLC), 콘텍스트 적응 이진 산술 코딩 (context adaptive binary arithmetic coding; CABAC), 신택스 기반 콘텍스트 적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 간격 구획화 엔트로피 (Probability Interval Partitioning Entropy; PIPE) 코딩 또는 다른 엔트로피 인코딩 방법론에 따라, 1차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한 비디오 데이터의 디코딩 시에 비디오 디코더 (30) 에 의한 사용을 위해 인코딩된 비디오 데이터에 연관된 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0059] CABAC를 수행하기 위해, 비디오 인코더 (20) 는 콘텍스트 모델 내의 콘텍스트를 송신될 심볼에 할당할 수도 있다. 그 콘텍스트는, 예를 들어, 심볼의 이웃 값들이 영이 아닌지의 여부에 관련될 수도 있다. CAVLC를 수행하기 위해, 비디오 인코더 (20) 는 송신될 심볼에 대한 가변 길이 코드를 선택할 수도 있다. VLC에서의 코드워드들은 상대적으로 더 짧은 코드들이 더 가능성있는 (probable) 심볼들에 대응하는 반면, 더 긴 코드들은 덜 가능성있는 심볼들에 대응하도록 구성될 수도 있다. 이런 식으로, VLC의 사용은, 예를 들어, 송신될 각각의 심볼에 대해 동일 길이 코드워드들을 사용하여, 비트 절약을 달성할 수도 있다. 확률 결정은 심볼에 할당된 콘텍스트에 기초할 수도 있다.

- [0060] 본 개시물은, 단일-계층 코딩 뿐만 아니라 스케일러블 및 멀티뷰 코딩에 상호 호환 방식으로 적용될 수 있는 비디오 파라미터 세트들 및 시퀀스 파라미터 세트들 양쪽 모두를 포함하는 파라미터 세트들에 대한 설계 방법들을 설명한다. 멀티뷰 비디오 코딩 (multiview video coding; MVC) 은 H.264/AVC 의 확장이다. MVC 규격이 아래에서 간략히 논의된다.
- [0061] 도 2는 본 개시물에서 설명되는 하나 이상의 예들에 따른, 일 예의 MVC 인코딩 또는 디코딩 순서를 예시하는 그래픽 도면이다. 예를 들어, 도 2에 예시된 디코딩 순서 배열은 시간 우선 (time-first) 코딩이라고 지칭된다. 도 2에서, S0 내지 S7 각각은 멀티뷰 비디오의 상이한 뷰들을 지칭한다. T0 내지 T8 각각은 하나의 출력 시간 인스턴스를 나타낸다. 액세스 단위가 하나의 출력 시간 인스턴스에 대한 모든 뷰들의 코딩된 화상들을 포함할 수도 있다. 예를 들어, 제 1 액세스 단위는 시간 인스턴스 T0에 대한 뷰들 (S0 내지 S7) 의 모두 (즉, 화상들 (0 내지 7)) 를 포함하며, 제 2 액세스 단위는 시간 인스턴스 T1에 대한 뷰들 (S0 내지 S7) 의 모두 (즉, 화상들 (8 내지 15)) 를 포함하며, 등등이다. 이 예들에서, 화상들 (0 내지 7) 은 동일한 시간 인스턴스 (즉, 시간 인스턴스 T0) 에 있고, 화상들 (8 내지 15) 은 동일한 시간 인스턴스 (즉, 시간 인스턴스 T1) 에 있다. 동일한 시간 인스턴스를 갖는 화상들은 일반적으로 동시에 디스플레이되고, 그것은 관람자로 하여금 3D 볼륨을 포괄하는 이미지를 지각하게 하는 동일한 시간 인스턴스의 화상들 내의 대상들 간의, 수평 디스패리티, 및 어쩌면 얼마간의 수직 디스패리티이다.
- [0062] 도 2에서, 뷰들의 각각은 화상들의 세트들을 포함한다. 예를 들어, 뷰 S0은 화상들 (0, 8, 16, 24, 32, 40, 48, 56, 및 64) 의 세트를 포함하며, 뷰 S1은 화상들 (1, 9, 17, 25, 33, 41, 49, 57, 및 65) 의 세트를 포함하며, 등등이다. 각각의 세트는 2 개의 화상들을 포함하는데, 하나의 화상은 텍스처 뷰 성분이라고 지칭되고, 다른 화상은 깊이 뷰 성분이라고 지칭된다. 뷰의 화상들의 세트 내의 텍스처 뷰 성분 및 깊이 뷰 성분은 서로 대응하는 것으로 간주될 수도 있다. 예를 들어, 뷰의 화상들의 세트 내의 텍스처 뷰 성분은 뷰의 화상들의 세트 내의 깊이 뷰 성분에 대응하는 것으로 간주될 수 있고, 역으로도 대응하는 것으로 간주될 수도 있다 (즉, 깊이 뷰 성분은 세트에서의 그것의 텍스처 뷰 성분에 대응하고, 역으로도 대응한다). 본 개시물에서 사용된 바와 같이, 대응하는 텍스처 뷰 성분 및 깊이 뷰 성분은 단일 액세스 단위의 동일한 뷰의 일부인 것으로 간주될 수도 있다.
- [0063] 텍스처 뷰 성분은 디스플레이되는 실제 이미지 콘텐츠를 포함한다. 예를 들어, 텍스처 뷰 성분은 루마 (Y) 및 크로마 (Cb 및 Cr) 성분들을 포함할 수도 있다. 깊이 뷰 성분은 그것의 대응하는 텍스처 뷰 성분에서 화소들의 상대 깊이들을 나타낼 수도 있다. 하나의 예로서, 깊이 뷰 성분은 루마 값들만을 포함하는 그레이 스케일 이미지와 유사할 수도 있다. 다르게 말하면, 깊이 뷰 성분은 임의의 이미지 콘텐츠를 전달하지 않지만, 대신 텍스처 뷰 성분에서의 화소들의 상대 깊이들의 측정치를 제공할 수도 있다.
- [0064] 예를 들어, 깊이 뷰 성분에서 순수 백색 화소에 대응하는 화소 값은 대응하는 텍스처 뷰 성분에서의 그것의 대응하는 화소 또는 화소들이 관람자의 관점에서 더 가깝다는 것을 나타낼 수도 있고, 깊이 뷰 성분에서의 순수 흑색 화소에 대응하는 화소 값은 대응하는 텍스처 뷰 성분에서의 그것의 대응하는 화소 또는 화소들이 관람자의 관점에서 더 멀리 있다는 것을 나타낼 수도 있다. 흑색 및 백색 사이에서의 회색의 다양한 음영들에 대응하는 화소 값들은 상이한 깊이 레벨들을 나타낸다. 예를 들면, 깊이 뷰 성분에서의 매우 회색 화소는 텍스처 뷰 성분에서의 그것의 대응하는 화소가 깊이 뷰 성분에서의 약간 회색 화소보다 더 멀리 있다는 것을 나타낸다. 오직 하나의 화소 값이, 그레이 스케일과 유사하게, 화소들의 깊이를 식별하는데 필요하기 때문에, 깊이 뷰 성분은 오직 하나의 화소 값을 포함할 수도 있다. 따라서, 크로마 성분들과 유사한 값들은 깊이를 코딩하는 경우에 필요하지 않다.
- [0065] 깊이를 식별하기 위해 루마 값들 (예컨대, 세기 값들) 만을 사용하는 깊이 뷰 성분이 예시 목적으로 제공되고 제한하는 것으로 간주되지 않아야 한다. 다른 예들에서, 임의의 기법이 텍스처 뷰 성분에서 화소들의 상대 깊이들을 나타내기 위해 이용될 수도 있다.
- [0066] MVC에 따라, 텍스처 뷰 성분들은 동일한 뷰에서의 텍스처 뷰 성분들로부터 또는 하나 이상의 상이한 뷰들에서의 텍스처 뷰 성분들로부터 인터 예측된다. 텍스처 뷰 성분들은, "비디오 블록들"이라고 지칭되는 그리고 H.264 컨텍스트에서 "매크로블록들"이라 통상 불리는 비디오 데이터의 블록들 내에 코딩될 수도 있다.
- [0067] MVC에서, 뷰간 예측은 디스패리티 모션 보상에 의해 지원되는데, 이 디스패리티 모션 보상은 H.264/AVC 모션 보상의 신택스를 사용하지만, 상이한 뷰에서의 화상이 코딩되고 있는 화상을 예측하기 위한 참조 화상으로서 사용되는 것을 허용한다. 2 개의 뷰들의 코딩은 MVC에 의해 또한 지원될 수 있다. MVC의 하나의 잠재적인 장점은, MVC 인코더가 2 개를 초과하는 뷰들을 3D 비디오 입력으로서 취할 수 있고 MVC 디코더는 캡처된 비디오

의 이러한 멀티뷰 표현을 디코딩할 수도 있다는 것이다. MVC 디코더들을 갖는 임의의 렌더러 (renderer) 는 2 개를 초과하는 뷰들을 갖는 3D 비디오 콘텐츠를 프로세싱할 수도 있다.

[0068] MVC에서, 뷰간 예측은 동일한 액세스 단위의 (즉, 동일한 시간 인스턴스를 가진) 화상들 사이에서 허용된다. 비-기본 뷰에서의 화상을 인코딩하는 경우, 화상이 상이한 뷰에 있지만 동일한 시간 인스턴스를 가지면, 그 화상은 참조 화상 리스트에 추가될 수도 있다. 뷰간 예측 참조 화상은, 임의의 인터 예측 참조 화상처럼, 참조 화상 리스트의 임의의 위치에 놓일 수 있다.

[0069] 도 3은 일 예의 MVC 예측 패턴을 도시하는 개념도이다. 도 3의 예에서, 8 개의 뷰들 (뷰 ID들인 "S0" 내지 "S7"을 가짐) 이 도시되고, 12 개의 시간적 로케이션들 ("T0" 내지 "T11") 이 각각의 뷰를 위해 도시되어 있다. 다시 말하면, 도 3의 각각의 행은 뷰에 대응하는 한편, 각각의 열은 시간적 로케이션을 나타낸다. 도 3의 예에서, 대문자 "B" 및 소문자 "b"는 상이한 코딩 방법론들보다는 화상들 사이의 상이한 계층적 관계들을 나타내는데 사용된다. 대체로, 대문자 "B"의 화상들은 소문자 "b"의 프레임들보다 예측 계층구조에서 상대적으로 상위에 있다.

[0070] 도 3에서, 뷰 S0은 기본 뷰로서 간주될 수도 있고, 뷰들 (S1 내지 S7) 은 의존성 뷰들로서 간주될 수도 있다. 기본 뷰는 뷰간 예측되지 않은 화상들을 포함한다. 기본 뷰에서의 화상은 동일한 뷰에서의 다른 화상들에 관해 인터 예측될 수 있다. 예를 들면, 뷰 S0에서의 화상들 중 어느 것도 뷰들 (S1 내지 S7) 중 임의의 뷰에서의 화상에 관해 인터 예측될 수 없지만, 뷰 S0에서의 화상들의 일부는 뷰 S0에서의 다른 화상들에 관해 인터 예측될 수 있다.

[0071] 의존성 뷰는 뷰간 예측되는 화상들을 포함한다. 예를 들어, 뷰들 (S1-S7) 중 각각의 하나는 다른 뷰에서의 화상에 관해 인터 예측되는 적어도 하나의 화상을 포함한다. 의존성 뷰에서의 화상들은 기본 뷰에서의 화상들에 관해 인터 예측될 수도 있거나, 또는 다른 의존성 뷰들에서의 화상들에 관해 인터 예측될 수도 있다.

[0072] 기본 뷰 및 하나 이상의 의존성 뷰들 양쪽 모두를 포함하는 비디오 스트림은 상이한 유형들의 비디오 디코더들에 의해 디코딩가능할 수도 있다. 예를 들어, 비디오 디코더의 하나의 기본 유형은 기본 뷰만을 디코딩하도록 구성될 수도 있다. 덧붙여서, 다른 유형의 비디오 디코더는 뷰들 (S0 내지 S7) 의 각각을 디코딩하도록 구성될 수도 있다. 기본 뷰 및 의존성 뷰들 양쪽 모두를 디코딩하도록 구성되는 디코더는, 멀티뷰 코딩을 지원하는 디코더라고 지칭될 수도 있다.

[0073] 도 3의 화상들은 도 3에서 각각의 행 및 각각의 열의 교차부분에 표시된다. MVC 확장들을 갖는 H.264/AVC 표준은 프레임이란 용어를 사용하여 비디오의 부분을 나타낼 수도 있는 반면, HEVC 표준은 화상이란 용어를 사용하여 비디오의 부분을 나타낼 수도 있다. 본 개시물은 용어 화상 및 프레임을 교환적으로 사용한다.

[0074] 도 3의 화상들은 대응 화상이 인트라 코딩된 것 (다시 말하면, I-화상) 인지, 또는 한 방향으로 인터 코딩된 것 (다시 말하면, P-화상) 인지 또는 다수의 방향으로 인터 코딩된 것 (다시 말하면, B-화상) 인지를 지정하는 글자를 포함하는 음영진 블록을 사용하여 예시된다. 대체로, 예측들은 화살표들로 나타내어지며, 가리켜진 (pointed-to) 화상들은 예측 참조를 위해 가리키는 (pointed-from) 화상을 이용한다. 예를 들어, 시간적 로케이션 T0에서의 뷰 S2의 P-화상은 시간적 로케이션 T0에서의 뷰 S0의 I-화상으로부터 예측된다.

[0075] 단일 뷰 비디오 인코딩에서처럼, 멀티뷰 비디오 코딩 비디오 시퀀스의 화상들은 상이한 시간적 로케이션들에서 화상들에 관해 예측하여 인코딩될 수도 있다. 예를 들어, 시간적 로케이션 T1에서의 뷰 S0의 B-화상은 시간적 로케이션 T0에서 자신으로 향하는 뷰 S0의 I-화상으로부터의 화살표를 가져서, b-화상이 I-화상으로부터 예측됨을 나타낸다. 그러나, 부가적으로, 멀티뷰 비디오 인코딩의 맥락에서, 화상들은 뷰간 예측될 수도 있다. 다시 말하면, 뷰 성분 (예컨대, 텍스처 뷰 성분) 은 참조를 위해 다른 뷰들의 뷰 성분들을 이용할 수 있다. MVC에서, 예를 들어, 뷰간 예측은 다른 뷰의 뷰 성분이 인터 예측 참조인 듯이 실현된다. 잠재적 뷰간 참조들은 시퀀스 파라미터 세트 (SPS) MVC 확장으로 시그널링되고 참조 화상 리스트 구성 프로세스에 의해 수정될 수 있는데, 이 구성 프로세스는 인터 예측 또는 뷰간 예측 참조들의 유연한 순서화 (ordering) 를 가능하게 한다.

[0076] 도 3은 뷰간 예측의 다양한 예들을 제공한다. 뷰 S1의 화상들은, 도 3의 예에서, 뷰 S1의 상이한 시간적 로케이션들에 있는 화상들로부터 예측된 것으로서 뿐만 아니라, 동일한 시간적 로케이션들에 있는 뷰들 (S0 및 S2) 의 화상들로부터 뷰간 예측된 것으로서 예시된다. 예를 들어, 시간적 로케이션 T1에 있는 뷰 S1의 B-화상은 시간적 로케이션들 (T0 및 T2) 에 있는 뷰 S1의 B-화상들 뿐만 아니라 시간적 로케이션 T1에 있는 뷰들 (S0 및 S2) 의 B-화상들 각각으로부터 예측된다.

- [0077] 도 3은 또한 예측 계층구조에서의 변화들을 상이한 음영 (shading) 레벨들을 이용하여 도시하는데, 음영의 양이 더 큰 (다시 말하면, 상대적으로 더 어두운) 프레임들일수록 적은 음영을 갖는 (다시 말하면, 상대적으로 더 밝은) 프레임들보다 예측 계층구조에서 더 상위에 있다. 예를 들어, 도 3의 모든 I-화상들은 짙은 음영으로 도시되는 반면, P-화상들은 약간 더 밝은 음영을 가지고, B-화상들 (및 소문자 b-화상들) 은 서로 상대적으로지만 P-화상들 및 I-화상들의 음영보다는 항상 더 밝은 여러 가지 음영 레벨들을 가진다.
- [0078] 대체로, 예측 계층구조는, 예측 계층구조에서 상대적으로 더 상위의 화상들이 그 계층구조에서 상대적으로 더 하위의 화상들을 디코딩하기 전에 디코딩되어야 한다는 점에서, 뷰 순서 인덱스들에 관련될 수도 있다. 계층구조에서 상대적으로 더 상위의 그들 화상들은 계층구조에서 상대적으로 더 하위의 화상들의 디코딩 동안에 참조 화상들로서 사용될 수 있다. 뷰 순서 인덱스는 액세스 단위에서 뷰 성분들의 디코딩 순서를 나타내는 인덱스이다. 뷰 순서 인덱스들은 H.264/AVC (MVC 개정안) 의 부록 A에 명시된 바와 같이, 시퀀스 파라미터 세트 (SPS) MVC 확장에서 암시된다. SPS에서, 각각의 인덱스 i 에 대해, 대응하는 $view_id$ 가 시그널링된다. 뷰 성분들의 디코딩은 뷰 순서 인덱스의 오름 차순을 따를 수도 있다. 모든 뷰들이 제시된다면, 뷰 순서 인덱스들은 0부터 $num_views_minus_1$ 로의 연속하는 순서로 존재한다.
- [0079] 이런 방식으로, 참조 화상들로서 사용되는 화상들은 참조 화상들에 의존하는 화상들보다 먼저 디코딩된다. 뷰 순서 인덱스는 액세스 단위에서 뷰 성분들의 디코딩 순서를 나타내는 인덱스이다. 각각의 뷰 순서 인덱스 i 에 대해, 대응하는 $view_id$ 가 시그널링된다. 뷰 성분들의 디코딩은 뷰 순서 인덱스들의 오름 차순을 따른다. 모든 뷰들이 제시된다면, 뷰 순서 인덱스들의 세트는 영부터 뷰들의 전체 수보다는 하나 작은 수까지의 연속 순서의 세트를 포함할 수도 있다.
- [0080] 계층구조의 동일한 레벨들에 있는 특정한 화상들의 경우, 디코딩 순서는 서로에 대해 중요하지 않을 수도 있다. 예를 들어, 시간적 로케이션 T0에 있는 뷰 S0의 I-화상은 시간적 로케이션 T0에 있는 뷰 S2의 P-화상에 대한 참조 화상으로서 사용될 수도 있으며, 이 P-화상은 결국 시간적 로케이션 T0에 있는 뷰 S4의 P-화상에 대한 참조 화상으로서 사용될 수도 있다. 따라서, 시간적 로케이션 T0에 있는 뷰 S0의 I-화상은 시간적 로케이션 T0에 있는 뷰 S2의 P-화상보다 먼저 디코딩되어야 하고, 이 뷰 S2의 P-화상은 결국 시간적 로케이션 T0에 있는 뷰 S4의 P-화상보다 먼저 디코딩되어야 한다. 그러나, 뷰들 (S1 및 S3) 사이에서, 디코딩 순서가 중요하지 않은데, 뷰들 (S1 및 S3) 이 예측을 위해 서로 의존하지 않기 때문이다. 대신 뷰들 (S1 및 S3) 은 예측 계층구조에서 더 상위인 다른 뷰들로부터만 예측된다. 더군다나, 뷰 S1이 뷰들 (S0 및 S2) 뒤에 디코딩되는 한, 뷰 S1은 뷰 S4보다 먼저 디코딩될 수도 있다.
- [0081] 이런 방식으로, 계층적 순서화가 뷰들 (S0 내지 S7) 을 기술하는데 사용될 수도 있다. 본 개시물에서, 표기 " $SA > SB$ "는 뷰 SA가 뷰 SB보다 먼저 디코딩되어야 한다는 것을 의미한다. 이 표기법을 이용하면, 도 2의 예에서, $S0 > S2 > S4 > S6 > S7$ 이다. 또한, 도 2의 예에 관하여, $S0 > S1$, $S2 > S1$, $S2 > S3$, $S4 > S3$, $S4 > S5$, 그리고 $S6 > S5$ 이다. 이 계층적 순서화를 위반하지 않는 뷰들에 대한 어느 디코딩 순서라도 가능하다. 따라서, 제한들이 계층적 순서화에 기초하는 많은 상이한 디코딩 순서들이 가능하다.
- [0082] SPS MVC 확장이 이제 설명될 것이다. 뷰 성분은 참조를 위해 다른 뷰들에서의 뷰 성분들을 사용할 수 있으며; 이는 뷰간 예측이라 불린다. MVC에서, 뷰간 예측은 다른 뷰의 뷰 성분이 인터 예측 참조인 듯이 실현된다. 그러나, 잠재적 뷰간 참조들은 시퀀스 파라미터 세트 (SPS) MVC 확장 (다음의 신택스 테이블인 표 1에서 도시된 바와 같음) 으로 시그널링되고, 참조 화상 리스트 구축 프로세스에 의해 수정될 수 있는데, 이 구축 프로세스는 인터 예측 또는 뷰간 예측 참조들의 유연한 순서화를 가능하게 한다. 비디오 인코더 (20) 는 표 1에 도시된 바와 같은 신택스를 생성하도록 구성된 비디오 인코더의 일 예를 나타내고, 비디오 디코더 (30) 는 이러한 신택스를 파싱하고 프로세싱하도록 구성된 비디오 디코더의 일 예를 나타낸다.

표 1

seq_parameter_set_mvc_extension() {		기술자
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_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 1; i <= num_views_minus1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
num_level_values_signaled_minus1		ue(v)
for(i = 0; i <= num_level_values_signaled_minus1; i++) {		
level_idc[i]		u(8)
num_applicable_ops_minus1[i]		ue(v)
for(j = 0; j <= num_applicable_ops_minus1[i]; j++) {		
applicable_op_temporal_id[i][j]		u(3)
applicable_op_num_target_views_minus1[i][j]		ue(v)
for(k = 0; k <=		
applicable_op_num_target_views_minus1[i][j]; k++)		
applicable_op_target_view_id[i][j][k]		ue(v)
applicable_op_num_views_minus1[i][j]		ue(v)
}		
}		
}		

[0083]

[0084]

SPS MVC 확장에서, 각각의 뷰에 대해, 참조 화상 리스트 0 및 참조 화상 리스트 1을 형성하는데 사용될 수 있는 뷰들의 수가 시그널링된다. 앵커 화상에 대한 예측 관계는, SPS MVC 확장으로 시그널링된 바와 같이, 동일한 뷰의 (SPS MVC 확장으로 시그널링된) 년-앵커 화상에 대한 예측 관계와 상이할 수 있다.

[0085]

HEVC에 대한 파라미터 세트들이 이제 설명될 것이다. HEVC WD7에서, HEVC에서의 비디오, 시퀀스, 화상 및 적응 파라미터 세트 메커니즘은 드물게 변화하는 정보의 송신을 코딩된 블록 데이터의 송신으로부터 분리한다. 비디오, 시퀀스, 화상 및 적응 파라미터 세트들이, 일부 애플리케이션들에서, "대역 외" 전달될 수도 있으며, 즉, 코딩된 비디오 데이터를 포함하는 단위들과 함께 전송되지 않을 수도 있다. 대역 외 송신은 통상적으로 신뢰할 수 있다.

[0086]

HEVC WD7에서, 비디오 시퀀스 파라미터 세트 (VPS), 시퀀스 파라미터 세트 (SPS), 화상 파라미터 세트 (PPS) 또는 적응 파라미터 세트 (APS)의 식별자가 가변 길이 선택스 엘리먼트 'ue(v)'를 사용하여 코딩된다. 각각의 SPS는 SPS ID와 VPS ID를 포함하며, 각각의 PPS는 PPS ID와 SPS ID를 포함하고, 각각의 슬라이스 헤더는 PPS ID와 가능하다면 APS ID를 포함한다.

[0087]

비록 비디오 파라미터 세트 (VPS)가 HEVC WD7에서 지원되지만, 시퀀스 레벨 정보 파라미터들의 대부분은 여전히 SPS에서만 존재한다. WD7의 VPS 설계의 여러 문제들 또는 잠재적인 단점들이 존재한다. 하나의 예로서, SPS들에 포함된 상당한 양의 정보가 모든 SPS들에 대해 동일할 것이거나 또는 적어도 2개의 SPS들에 대해 동일할 것이다. 이 정보를 SPS에서 중복하는 것은 더 높은 대역폭을 요구한다. 파라미터 세트들 (적어도 VPS, SPS 및 PPS를 포함함)은 대역 외 시그널링될 필요가 있을 수도 있다. 대역 내 시그널링되면, 이러한 비트-레이트 증가는 랜덤 액세스 포인트에서의 각각의 튜닝 (tune-in)에 효과적이다.

[0088]

제 2 예로서, 잠재적인 HEVC 확장들에서, AVC와 유사한 설계 원리들이 추종되면, 대부분의 동작 포인트 기술 정보는 SPS 또는 VPS에 포함되지 않을 수도 있고, 대신, SEI 메시지들이 세션 초기화 및 협상을 위해 사용될 수도

있다. 따라서, MANE는 위에서 언급된 목적들을 위해 SPS, VPS, 및 SEI 메시지들을 파싱하는데 필요하게 될 수도 있다. 제 3 예로서, WD7에서의 SPS에 존재하는 일부 정보가 HEVC 확장들에서 변경되거나 또는 제거될 수도 있다.

[0089] 위에서 논의된 잠재적인 문제들을 해결하기 위해, 본 개시물은 VPS 또는 다른 파라미터 세트들을 포함하는 파라미터 세트들의 설계를 위한 여러 기법들을 제안한다. 예를 들어, 본 개시물에서 설명된 기법들에 따르면, 전체 코딩된 비디오 시퀀스에 대해 통상적으로 동일한 정보가 VPS에 존재할 수도 있지만, SPS 레벨에서 변할 것인 신택스 엘리먼트들만이 SPS에 존재할 수도 있다. 다른 신택스 엘리먼트들은 VPS에 이미 존재한다면 SPS로부터 배제될 수도 있다.

[0090] 본 개시물의 기법들의 다른 예로서, 세션 협상에 관련된 정보가 VPS에 존재할 수도 있다. 세션 협상에 관련된 정보의 예들은 프로파일 정보, 레벨 정보, 프레임 해상도 정보, 프레임 레이트 정보, 및 비트 레이트 정보, 뿐만 아니라 다른 정보들을 포함한다. 본 개시물의 기법들의 다른 예로서, VPS는, 세션 협상에 중요한 동작 포인트 정보의 파싱이 기본 계층 또는 뷰 및 향상 계층들 또는 뷰들 모두를 위한 정보를 잠재적으로 포함하는 가변 길이 코딩을 필요로 하지 않은 방식으로 설계될 수도 있다. VPS에서의 신택스 엘리먼트들은, 각각의 그룹에 대해, HEVC 확장이 0 또는 그 이상의 인스턴스들을 제공할 것이고 HEVC 확장에서의 동작 포인트들이 인텍스만을 참조하도록 그룹화될 수도 있다.

[0091] VPS, SPS, 비디오 유용성 정보 (VUI), 그리고 HRD 파라미터들 및 슬라이스 헤더에 대한 신택스 및 시맨틱스의 다양한 예들이 아래에서 제공된다. 표 2 내지 표 6은 제 1 예를 도시한다. 위에서 언급된 표 1은 VPS 신택스의 일 예를 도시한다. 표 2 내지 표 6에서, 뿐만 아니라 본 개시물에서의 다른 표들에서 "기술자 (descriptor)" 열은, 각각의 신택스 엘리먼트에 대한 비트들의 수를 식별하며, 여기서 비트들의 수를 나타내는 "v"는 가변적일 수도 있다. "기술자" 열에서의 숫자 값들은 신택스 엘리먼트가 고정된 수의 비트들을 사용하여 전달됨을 나타낸다. 예를 들어, "u(8)"은 고정된 수의 8 비트들로 신택스 엘리먼트를 나타내는 반면, "ue(v)"는 가변적인 수의 비트들로 신택스 엘리먼트를 나타낸다. 기술자 ue(v)를 갖는 신택스 엘리먼트들을 파싱하기 위하여, 파싱 디바이스 (이를테면 비디오 디코더 또는 MANE) 는 이러한 신택스 엘리먼트들을 디코딩하고 해석하기 위해 엔트로피 코딩을 구현할 필요가 있을 수도 있다.

표 2

비디오 파라미터 세트 RBSP 선택스

video_parameter_set_rbsp() {	기술자
vps_max_temporal_layers_minus1	u(3)
vps_max_layers_minus1	u(5)
profile_space	u(3)
profile_idc	u(5)
for(j = 0; j < 32; j++)	
profile_compatibility_flag[i]	u(1)
constraint_flags	u(16)
level_idc	u(8)
level_lower_temporal_layers_present_flag	u(1)
if(level_lower_temporal_layers_present_flag)	
for(i = 0; i < vps_max_temporal_layers_minus1; i++)	
level_idc_temporal_subset[i]	u(8)
video_parameter_set_id	u(5)
vps_temporal_id_nesting_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
bit_depth_luma_minus8	u(2)
bit_depth_chroma_minus8	u(2)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
for(i = 0; i <= vps_max_temporal_layers_minus1; i++) {	
bitrate_info_present_flag[i]	u(1)
frm_rate_info_present_flag[i]	u(1)
if(bitrate_info_present_flag[i]) {	
avg_bitrate[i]	u(16)
max_bitrate[i]	u(16)
}	
if(frm_rate_info_present_flag[i]) {	
constant_frm_rate_idc[i]	u(2)
avg_frm_rate[i]	u(16)
}	
}	
next_essential_info_byte_offset	u(12)
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)
}	
for(i = 0, nalHrdPresent = 0, vclHrdPresent = 0;	
i <= vps_max_temporal_layers_minus1; i++) {	
nal_hrd_parameters_present_flag[i]	u(1)
if(nal_hrd_parameters_present_flag[i]) {	

[0092]

hrd_parameters(nalHrdPresent)	
nalHrdPresent++	
}	
vcl_hrd_parameters_present_flag[i]	u(1)
if(vcl_hrd_parameters_present_flag[i]) {	
hrd_parameters(vclHrdPresent)	
vclHrdPresent++	
}	
if(nalHrdPresent + vclHrdPresent == 1) {	
low_delay_hrd_flag	u(1)
sub_pic_cpb_params_present_flag	u(1)
num_units_in_sub_tick	u(32)
}	
vps_max_dec_pic_buffering[i]	ue(v)
vps_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
num_vps_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_vps_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

[0093]

표 3

시퀀스 파라미터 세트 RBSP 선택스

seq_parameter_set_rbsp() {	기술자
seq_parameter_set_id	ue(v)
video_parameter_set_id	ue(v)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)
}	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
restricted_ref_pic_lists_flag	<u>u(1)</u>
if(restricted_ref_pic_lists_flag)	
lists_modification_present_flag	<u>u(1)</u>
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)

[0094]

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	u(1)
if(scaling_list_enable_flag) {	
sps_scaling_list_data_present_flag	u(1)
if(sps_scaling_list_data_present_flag)	
scaling_list_param()	
}	
chroma_pred_from_luma_enabled_flag	u(1)
transform_skip_enabled_flag	u(1)
seq_loop_filter_across_slices_enabled_flag	u(1)
asymmetric_motion_partitions_enabled_flag	u(1)
nsrqt_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)
if(log2_min_coding_block_size_minus3 == 0)	
inter_4x4_enabled_flag	u(1)
num_short_term_ref_pic_sets	ue(v)
use_rps_from_vps_flag	u(1)
for(i = 0; i < num_short_term_ref_pic_sets; i++){	
idx = use_rps_from_vps_flag ? num_vps_short_term_ref_pic_sets + i : i	
short_term_ref_pic_set(idx)	
}	
long_term_ref_pics_present_flag	u(1)
sps_temporal_mvp_enable_flag	u(1)
tiles_fixed_structure_idc	u(2)
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0095]

표 4

VUI 파라미터들 선택스

vui_parameters() {	기술자
aspect_ratio_info_present_flag	u(1)
if(aspect_ratio_info_present_flag) {	
aspect_ratio_idc	u(8)
if(aspect_ratio_idc == Extended_SAR) {	
sar_width	u(16)
sar_height	u(16)
}	
}	
overscan_info_present_flag	u(1)
if(overscan_info_present_flag)	
overscan_appropriate_flag	u(1)
video_signal_type_present_flag	u(1)
if(video_signal_type_present_flag) {	
video_format	u(3)
video_full_range_flag	u(1)
colour_description_present_flag	u(1)
if(colour_description_present_flag) {	
colour_primaries	u(8)
transfer_characteristics	u(8)
matrix_coefficients	u(8)
}	
}	
chroma_loc_info_present_flag	u(1)
if(chroma_loc_info_present_flag) {	
chroma_sample_loc_type_top_field	ue(v)
chroma_sample_loc_type_bottom_field	ue(v)
}	
neutral_chroma_indication_flag	u(1)
field_seq_flag	u(1)
timing_info_present_flag	u(1)
if(timing_info_present_flag) {	
num_units_in_tick	u(32)
time_scale	u(32)
fixed_pic_rate_flag	u(1)
}	
bitstream_restriction_flag	u(1)
if(bitstream_restriction_flag) {	
motion_vectors_over_pic_boundaries_flag	u(1)
max_bytes_per_pic_denom	ue(v)
max_bits_per_minu_denom	ue(v)
log2_max_mv_length_horizontal	ue(v)
log2_max_mv_length_vertical	ue(v)
}	
}	

[0096]

표 5

HRD 파라미터들 선택스

hrd_parameters(i) {	기술자
if(i == 0) {	
cpb_cnt_minus1	ue(v)
bit_rate_scale	u(4)
cpb_size_scale	u(4)
}	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
bit_rate_value_minus1 [i][SchedSelIdx]	ue(v)
cpb_size_value_minus1 [i][SchedSelIdx]	ue(v)
if(i == 0)	
cbr_flag [SchedSelIdx]	u(1)
}	
if(i == 0) {	
initial_cpb_removal_delay_length_minus1	u(5)
cpb_removal_delay_length_minus1	u(5)
dpb_output_delay_length_minus1	u(5)
time_offset_length	u(5)
}	
}	

[0097]

표 6

슬라이스 헤더 선택스

slice_header() {	기술자
first_slice_in_pic_flag	u(1)
pic_parameter_set_id	ue(v)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
if(!dependent_slice_flag) {	
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(RapPicFlag) {	
rap_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
if(!IdrPicFlag) {	
pic_order_cnt_lsb	u(v)
short_term_ref_pic_set_sps_flag	u(1)
if(!short_term_ref_pic_set_sps_flag)	
short_term_ref_pic_set(NumShortTermRefPicSets)	
else	
short_term_ref_pic_set_idx	u(v)

[0098]

if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i = 0; i < num_long_term_pics; i++) {	
poc_lsb_lt[i]	u(v)
delta_poc_msb_present_flag[i]	u(1)
if(delta_poc_msb_present_flag[i])	
delta_poc_msb_cycle_lt[i]	ue(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
}	
if(sample_adaptive_offset_enabled_flag) {	
slice_sample_adaptive_offset_flag[0]	u(1)
if(slice_sample_adaptive_offset_flag[0]) {	
slice_sample_adaptive_offset_flag[1]	u(1)
slice_sample_adaptive_offset_flag[2]	u(1)
}	
if(adaptive_loop_filter_enabled_flag)	
aps_id	ue(v)
if(slice_type == P slice_type == B) {	
if(sps_temporal_mvp_enable_flag)	
pic_temporal_mvp_enable_flag	u(1)
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
if(lists_modification_present_flag)	
ref_pic_list_modification()	
if(slice_type == B)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag && slice_type != I)	
cabac_init_flag	u(1)
slice_qp_delta	se(v)
if(deblocking_filter_control_present_flag) {	
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_header_disable_deblocking_filter_flag	u(1)
if(!slice_header_disable_deblocking_filter_flag) {	
beta_offset_div2	se(v)
tc_offset_div2	se(v)
}	

[0099]

}	
}	
if(pic_temporal_mvp_enable_flag) {	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
if(slice_type != I && ((collocated_from_l0_flag && num_ref_idx_l0_active_minus1 > 0) (!collocated_from_l0_flag && num_ref_idx_l1_active_minus1 > 0))	
collocated_ref_idx	ue(v)
}	
if((weighted_pred_flag && slice_type == P) (weighted_bipred_idc == 1 && slice_type == B))	
pred_weight_table()	
if(slice_type == P slice_type == B)	
five_minus_max_num_merge_cand	ue(v)
if(adaptive_loop_filter_enabled_flag) {	
slice_adaptive_loop_filter_flag	u(1)
if(slice_adaptive_loop_filter_flag && alf_coef_in_slice_flag)	
alf_param()	
if(slice_adaptive_loop_filter_flag && !alf_coef_in_slice_flag)	
alf_cu_control_param()	
}	
if(seq_loop_filter_across_slices_enabled_flag && (slice_adaptive_loop_filter_flag slice_sample_adaptive_offset_flag !disable_deblocking_filter_flag))	
slice_loop_filter_across_slices_enabled_flag	u(1)
}	
if(tiles_or_entropy_coding_sync_idc == 1 tiles_or_entropy_coding_sync_idc == 2) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset[i]	u(v)
}	
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte	u(8)
}	
byte_alignment()	
}	

[0100]

[0101]

비디오 파라미터 세트 RBSP 시맨틱스, 이를테면 위의 표 2에서 도시된 것들이 이제 설명될 것이다. 표 2에서의 선택스 엘리먼트 video_parameter_set_id는 비디오 파라미터 세트에 대한 ID (identification) 를 제공한다. video_parameter_set_id의 값을 사용하여, 다른 선택스 구조, 이를테면 SPS가 특정 VPS를 활성화할 수 있다. 예를 들어, 일 예의 SPS 선택스 구조를 도시하는 표 3은, 선택스 엘리먼트 video_parameter_set_id를 또한 포함한다. SPS에서의 선택스 엘리먼트 video_parameter_set_id의 값에 기초하여, 그 동일한 값을 갖는 특정 VPS는 SPS에 연관된 비디오 블록들을 코딩하기 위해 활성화될 수 있다. 보통, 다수의 SPS들이 동일한 VPS와 연관될 것이다. 일 예로서, 비디오 디코더 (30) 는 비디오 데이터에서 선택스 엘리먼트 video_parameter_set_id에 대한 제 1 값을 포함하는 제 1 SPS를 수신할 수도 있고, 비디오 디코더 (30) 는 선택스 엘리먼트 video_parameter_set_id에 대한 동일한 값을 포함하는 제 2 SPS를 또한 수신할 수도 있다. 제 1 SPS는 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조일 수도 있고, 제 2 SPS는 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조일 수도 있다. 비디오 디코더 (30) 는 제 1 SPS 및 제 2 SPS 양쪽 모두에 연관된 비디오 블록들을 동일한 VPS로부터의 파라미터들에 기초하여 디코딩한다.

[0102]

선택스 엘리먼트들 profile_space, profile_idc, profile_compatibility_flag[i], constraint_flags, level_idc, 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, pic_crop_bottom_offset, temporal_id_nesting_flag 및 separate_colour_plane_flag는 WD7에서 특정된 바와 같이 시퀀스 파라미터 세트에서의 동일한 선택스 엘리먼트 이름들과는 그들 선택스 엘리먼트들의 동일한 시맨틱스를 가지지만, 본 개시물의 제안된 기법들에 따르면 SPS로

부터 VPS로 이동되었다.

- [0103] 신택스 엘리먼트 `profile_space`는 신택스 엘리먼트 `profile_idc`를 해석하기 위한 컨텍스트를 식별하고, 신택스 엘리먼트 `profile_idc`는 프로파일들의 그룹을 식별한다. 신택스 엘리먼트들 `profile_compatibility_flag[i]`는 비디오 데이터가 프로파일 `[i]`와 호환가능한지를 식별할 수도 있다. 비디오 디코더 (30)는, 예를 들어, 비디오 데이터에서 `profile_space` 및 `profile_idc`에 대한 값들을 수신하고, `profile_space`의 값에 기초하여, 신택스 엘리먼트 `profile_idc`를 해석하기 위한 컨텍스트를 식별할 수도 있다. `profile_idc`의 해석된 값에 기초하여, 비디오 디코더 (30)는 프로파일들의 그룹을 식별할 수 있고, 각각의 프로파일에 대해, 비디오 디코더 (30)는 비디오 데이터가 프로파일 `[i]`와 호환가능한지를 식별하기 위해 신택스 엘리먼트 `profile_compatibility_flag[i]`에 대한 값을 수신할 수 있다. 신택스 엘리먼트 `profile_idc`는, 예를 들어, 각각의 플래그가 프로파일의 특정 양태를 나타내는 32 개의 연관된 플래그들을 가질 수도 있다. 예를 들어, 동일한 프로파일이 주어진다고 하면, 하나의 특정 코딩 또는 프로세스 도구가 턴 온 되는지 또는 오프되는지를 플래그가 나타낼 수도 있다.
- [0104] 신택스 엘리먼트 `level_idc`는 비디오 데이터에 연관된 최대 레벨을 식별하고, 신���스 엘리먼트 `level_lower_temporal_layers_present_flag`는 비디오 데이터의 시간 계층이 최대 레벨보다 낮은 레벨을 가지는지를 식별한다. 1과 동일하게 설정된 신택스 엘리먼트 `level_lower_temporal_layers_present_flag`는 `level_idc_temporal_subset[i]`가 존재할 수도 있음을 특징한다. 0과 동일하게 설정된 신택스 엘리먼트 `level_lower_temporal_layers_present_flag`는 `level_idc_temporal_subset[i]`가 존재하지 않음을 특징한다. 신택스 엘리먼트 `level_idc_temporal_subset[i]`는 `i` 이하의 `temporal_id`를 갖는 모든 NAL 단위들로 이루어진 비트스트림 서브셋이 순응하는 레벨을 특징한다.
- [0105] 비디오 디코더 (30)는, 예를 들어, 1과 동일하게 설정된 신택스 엘리먼트 `level_lower_temporal_layers_present_flag`의 수신에 응답하여, 신택스 엘리먼트들 `level_idc_temporal_subset[i]`를 수신할 수도 있다. 신택스 엘리먼트들 `level_idc_temporal_subset[i]`는 시간 계층 `[i]`가 준수하는 레벨을 식별하기 위해 존재할 수도 있다.
- [0106] 신택스 엘리먼트들 `vps_temporal_id_nesting_flag`, `vps_temporal_id_nesting_flag`, `vps_max_dec_pic_buffering[i]`, `vps_num_reorder_pics[i]`, 및 `vps_max_latency_increase[i]`는 HEVC WD 7의 시퀀스 파라미터 세트에서 각각 다음의 신택스 엘리먼트들의 동일한 시맨틱스를 가진다: `sps_temporal_id_nesting_flag`, `sps_temporal_id_nesting_flag`, `sps_max_dec_pic_buffering[i]`, `sps_num_reorder_pics[i]`, `sps_max_latency_increase[i]`.
- [0107] 신택스 엘리먼트 `next_essential_info_byte_offset`은 본 개시물에서 도입된 오프셋 신택스 엘리먼트의 일 예이다. 신택스 엘리먼트 `next_essential_info_byte_offset`은 VPS NAL 단위에서 그 NAL 단위의 시작부분에서부터 시작하여 프로파일과 레벨 정보 및 다른 고정 길이 코딩된 정보의 다음 세트의 바이트 오프셋을 특징한다. MANE (29)는, 예를 들어, 신택스 엘리먼트 `next_essential_info_byte_offset`를 수신하고 신택스 엘리먼트 `next_essential_info_byte_offset`에 의해 나타내어진 바이트들의 수를 결정할 수도 있고, 결정된 바이트들의 수에 기초하여, MANE (29)는 표 2에 도시된 하나 이상의 가변 길이 코딩된 신택스 엘리먼트들, 이를테면 가변 길이 신택스 엘리먼트들 `pic_crop_left_offset`, `pic_crop_right_offset`, `pic_crop_top_offset`, `pic_crop_bottom_offset`, 및 표 2에 도시된 다른 가변 길이 신택스 엘리먼트들을 스킵할 수도 있다. 비디오 디코더 (30)는, 그러나, 신택스 엘리먼트 `next_essential_info_byte_offset`의 수신 시, 그 신택스 엘리먼트의 값을 무시할 수도 있다. 따라서, 신택스 엘리먼트 `next_essential_info_byte_offset`을 파싱한 후, 비디오 디코더 (30)는 가변 길이 신택스 엘리먼트들 `pic_crop_left_offset`, `pic_crop_right_offset`, `pic_crop_top_offset`, `pic_crop_bottom_offset`, 및 표 2에 도시된 다른 가변 길이 신택스 엘리먼트들을 계속 파싱할 수도 있다.
- [0108] HEVC 규격의 미래의 확장, 예를 들어 스케일러블 코딩 확장 또는 3DV 확장에서, 비-기본 계층 또는 뷰에 대한 VPS 정보는 VPS NAL 단위에서 기본 계층 또는 뷰에 대한 VPS 정보 뒤에 포함될 수도 있다. 비-기본 계층 또는 뷰에 대한 VPS 정보는 또한 고정 길이 신택스 엘리먼트들, 이를테면 코딩된 프로파일, 레벨, 및 세션 협상에 필수적인 다른 정보로 시작할 수도 있다. `next_essential_info_byte_offset`에 의해 특정되는 비트 오프셋을 사용하여, MANE (29)는 엔트로피 디코딩을 수행할 필요 없이 VPS NAL 단위에서의 당해 필수적인 정보의 위치를 찾고 액세스할 수도 있다. 비디오 데이터를 수송하고 프로세싱하도록 구성된 일부 네트워크 엔티티들 (예컨대 MANE (29))은 엔트로피 디코딩을 위해 장비되지 않을 수도 있다. 그러나, 본 개시물에서 설명된

바와 같은 오프셋 신택스 엘리먼트를 사용하여, 이러한 네트워크 엔티티들은 파라미터 세트의 일부 양태들을 여전히 프로세싱하고, 비디오 데이터를 위한 라우팅 결정을 하는 경우 프로세싱된 신택스 엘리먼트에 포함된 정보를 사용할 수 있다. 라우팅 결정을 하는 경우에 네트워크 엔티티가 프로세싱할 수도 있는 정보의 일 예는, 세션 협상에 관련된 정보를 포함한다.

[0109] 신택스 엘리먼트들 `nal_hrd_parameters_present_flag[i]` 및 `vcl_hrd_parameters_present_flag[i]`는 WD7의 VUI 파라미터들에 존재하는 `nal_hrd_parameters_present_flag` 및 `vcl_hrd_parameters_present_flag`와 유사한 시맨틱스를 가지지만, i 번째 시간 계층 표현에 적용가능하다. 신택스 엘리먼트 `nal_hrd_parameters_present_flag`는, 예를 들어, 비트레이트, 코딩된 화상 버퍼 (CPB) 사이즈, 및 초기 CPB 제거 지연 (`initial_cpb_removal_delay_length_minus1`), CPB 제거 지연 (`cpb_removal_delay_length_minus1`), DPB 출력 지연 (`dpb_output_delay_length_minus1`), 및 타임 오프셋 길이 (`time_offset_length`) 와 같은 HRD 파라미터들이든 간에 시그널링할 수도 있다. 그 신택스 엘리먼트들은, 예를 들어, 비디오 데이터에 대한 비트레이트가 일정한지 또는 가변적인지를 나타내는 신택스 엘리먼트 (`cbr_flag`) 를 포함할 수도 있다.

[0110] 신택스 엘리먼트 `low_delay_hrd_flag`는 DPB로부터의 디코딩 단위의 제거 시간을 나타내는데 사용될 수도 있다. 1과 동일한 신택스 엘리먼트 `sub_pic_cpb_params_present_flag`는 서브 화상 레벨 CPB 제거 지연 파라미터들이 존재한다는 것을 특정할 수도 있고 CPB는 액세스 단위 레벨 또는 서브 화상 레벨에서 동작할 수도 있다. 0과 동일한 신택스 엘리먼트 `sub_pic_cpb_params_present_flag`는 서브 화상 레벨 CPB 제거 지연 파라미터들이 존재하지 않는다는 것을 특정할 수도 있고 CPB는 액세스 단위 레벨에서 동작한다. 신택스 엘리먼트 `num_units_in_sub_tick`는 서브 화상 클록 틱 카운터의 하나의 증분 (서브 화상 클록 틱이라 지칭됨) 에 대응하는 주파수 `time_scale` Hz에서 동작하는 클록의 시간 단위들의 수를 나타낸다. 위에서 논의된 HRD 파라미터들은 모든 시간 계층 표현들에 적용가능할 수도 있다.

[0111] 1과 동일하게 설정된 신택스 엘리먼트 `vui_video_parameters_present_flag`는 `vui_vps()` 신택스 구조가 VPS 내에 존재함을 특정한다. 0과 동일하게 설정된 이 플래그는 `vui_vps()` 신택스 엘리먼트가 존재하지 않음을 특정한다. 신택스 엘리먼트 `num_vps_short_term_ref_pic_sets`는 비디오 파라미터 세트에서 특정되는 단기 참조 화상 세트들의 수를 특정한다. 1과 동일하게 설정된 신���스 엘리먼트 `bitrate_info_present_flag[i]`는 i 번째 시간 계층에 대한 비트 레이트 정보가 비디오 파라미터 세트에 존재함을 특정한다. 0으로 설정된 신택스 엘리먼트 `bitrate_info_present_flag[i]`는 i 번째 시간 계층에 대한 비트 레이트 정보가 VPS에 존재하지 않음을 특정한다.

[0112] 1로 설정된 신택스 엘리먼트 `frm_rate_info_present_flag[i]`는 i 번째 시간 계층에 대한 프레임 레이트 정보가 비디오 파라미터 세트에 존재함을 특정한다. 0과 동일하게 설정된 신택스 엘리먼트 `frm_rate_info_present_flag[i]`는 i 번째 시간 계층에 대한 프레임 레이트 정보가 비디오 파라미터 세트에 존재하지 않음을 특정한다.

[0113] 신택스 엘리먼트 `avg_bitrate[i]`는 i 번째 시간 계층 표현의 평균 비트 레이트를 나타낸다. 초 당 비트 단위의 i 번째 시간 계층 표현에 대한 평균 비트 레이트는 다음에 의해 특정되는 함수 `BitRateBPS()`를 이용하여 `BitRateBPS(avg_bitrate[i])`에 의해 주어진다:

$$\text{BitRateBPS}(x) = (x \ \& \ (2^{14} - 1)) * 10^{(2 + (x \gg 14))}$$

[0114] 평균 비트 레이트는 HEVC 표준의 부록 C에서 특정된 액세스 단위 제거 시간에 따라 도출될 수도 있다. 다음에서, b_{Total} 은 i 번째 시간 계층 표현의 모든 NAL 단위들에서의 비트들의 수이며, t_1 은 VPS가 적용하는 제 1 액세스 단위의 제거 시간 (초 단위) 이고, t_2 는 VPS가 적용하는 (디코딩 순서에서) 마지막 액세스 단위의 제거 시간 (초 단위) 이다.

[0116] x 가 `avg_bitrate[i]`의 값을 특정하는 경우, 다음이 적용된다:

[0117] - t_1 이 t_2 와 동일하지 않으면, 다음의 조건이 참이 될 수도 있다:

$$(x \ \& \ (2^{14} - 1)) == \text{Round}(b_{\text{Total}} \div ((t_2 - t_1) * 10^{(2 + (x \gg 14))}))$$

[0119] - 그렇지 않으면 (t_1 이 t_2 와 동일하면), 다음의 조건이 참이 될 수도 있다:

- [0120] $(x \& (2^{14} - 1)) == 0$
- [0121] 신택스 엘리먼트 `max_bitrate_layer[i]`는 부록 C에서 특정된 바와 같이 액세스 단위 제거 시간의 임의의 1초 시간 윈도우에서 i 번째 시간 계층 표현의 비트 레이트에 대한 상한을 나타낸다. 초 당 비트 단위의 현재 스케일러블 계층의 비트 레이트에 대한 상한은 수식 G-369에서 특정된 함수 `BitRateBPS()`를 이용하여 `BitRateBPS(max_bitrate_layer[i])`에 의해 주어진다. 비트 레이트 값들은 HEVC 표준의 부록 C에서 특정된 액세스 단위 제거 시간에 따라 도출된다. 다음에서, t_1 은 임의의 시간 포인트이고 (초 단위), t_2 는 $t_1 + \text{max_bitrate_calc_window}[i] \div 100$ 과 동일하게 설정되고, $bTotal$ 은 t_1 이상이고 t_2 미만인 제거 시간을 갖는 액세스 단위들에 속하는 현재 스케일러블 계층의 모든 NAL 단위들에서의 비트들의 수이다. x 가 `max_bitrate_layer[i]`의 값을 특정하는 경우, 다음의 조건이 t_1 의 모든 값들에 대해 지켜질 수도 있다:
- [0122] $(x \& (2^{14} - 1)) \geq bTotal \div ((t_2 - t_1) * 10^{(2 + (x \gg 14))})$.
- [0123] 신택스 엘리먼트 `constant_frm_rate_idc[i]`는 i 번째 시간 계층 표현의 프레임 레이트가 일정한지의 여부를 나타낸다. 다음에서, 시간적 세그먼트 $tSeg$ 는 현재 시간 계층 표현의 디코딩 순서에서의 2 개 이상의 연속적인 액세스 단위들의 임의의 세트이며, $fTotal(tSeg)$ 는 시간적 세그먼트 $tSeg$ 에서의 화상들의 수이며, $t_1(tSeg)$ 는 시간적 세그먼트 $tSeg$ 의 (디코딩 순서에서의) 제 1 액세스 단위의 제거 시간 (초 단위)이며, $t_2(tSeg)$ 는 시간적 세그먼트 $tSeg$ 의 (디코딩 순서에서의) 마지막 액세스 단위의 제거 시간 (초 단위) 이고, $avgFR(tSeg)$ 는 다음에 의해 주어지는 시간적 세그먼트 $tSeg$ 에서의 평균 프레임 레이트이다:
- [0124] $avgFR(tSeg) == \text{Round}(fTotal(tSeg) * 256 \div (t_2(tSeg) - t_1(tSeg)))$
- [0125] i 번째 시간 계층 표현이 하나의 액세스 단위만을 포함하거나 또는 $avgFR(tSeg)$ 의 값이 i 번째 시간 계층 표현의 모든 시간적 세그먼트들에 걸쳐 일정하면, 프레임 레이트는 일정하고; 그렇지 않으면, 프레임 레이트는 일정하지 않다. 0과 동일하게 설정된 신택스 엘리먼트 `constant_frm_rate_idc[i]`는 i 번째 시간 계층 표현의 프레임 레이트가 일정하지 않음을 나타낸다. 1과 동일하게 설정된 신택스 엘리먼트 `constant_frm_rate_idc[i]`는 i 번째 시간 계층 표현의 프레임 레이트가 일정함을 나타낸다.
- [0126] 2와 동일하게 설정된 신택스 엘리먼트 `constant_frm_rate_idc[i]`는 i 번째 시간 계층 표현의 프레임 레이트가 일정할 수도 있거나 또는 일정하지 않을 수도 있음을 나타낼 수도 있다. `constant_frm_rate_idc[i]`의 값은 0부터 2까지의 범위에 있을 수도 있다.
- [0127] 신택스 엘리먼트 `avg_frm_rate[i]`는 256 초 당 프레임들 단위의, i 번째 시간 계층 표현의 평균 프레임 레이트를 나타낸다. $fTotal$ 가 i 번째 시간 계층 표현에서의 화상들의 수이고, t_1 이 VPS가 적용하는 제 1 액세스 단위의 제거 시간 (초 단위)이며, t_2 가 VPS가 적용하는 (디코딩 순서에서의) 마지막 액세스 단위의 제거 시간 (초 단위)인 경우, 다음이 적용된다:
- [0128] t_1 이 t_2 와 동일하지 않으면, 다음의 조건이 참이 될 수도 있다:
- [0129] $avg_frm_rate[i] == \text{Round}(fTotal * 256 \div (t_2 - t_1))$
- [0130] 그렇지 않으면 (t_1 이 t_2 와 동일하면), 다음의 조건이 참이 될 수도 있다:
- [0131] $avg_frm_rate[i] == 0$
- [0132] VUI 파라미터들의 시맨틱스가 이제 설명될 것이다. VUI 파라미터들에서의 각각의 신택스 엘리먼트는 WD7에서 특정된 바와 같은 VUI 파라미터들의 신택스에서 동일한 이름을 갖는 신택스 엘리먼트와 동일한 시맨틱스를 가진다.
- [0133] 시퀀스 파라미터 세트 RBSP 시맨틱스가 이제 설명될 것이다. 1과 동일하게 설정된 신택스 엘리먼트 `use_rps_from_vps_flag`는, 시퀀스 파라미터 세트에 포함된 단기 참조 화상 세트들이 참조되는 비디오 파라미터 세트에 포함된 단기 참조 화상 세트들에 부가적임을 특정한다. 0과 동일하게 설정된 신택스 엘리먼트 `use_rps_from_vps_flag`는, 시퀀스 파라미터 세트에 포함된 단기 참조 화상 세트들이 참조되는 비디오 파라미터 세트에 포함되는 단기 참조 화상 세트들을 오버라이드 (override) 함을 특정한다.

- [0134] 대안으로, 선택스 엘리먼트 `num_short_term_ref_pic_sets`는 SPS에 존재하지 않을 수도 있고 0과 동일하게 설정된 것으로 항상 유추될 수도 있다. 대안으로, 선택스 엘리먼트 `use_rps_from_vps_flag`는 존재하지 않을 수도 있고 1과 동일하게 설정된 것으로 항상 유추될 수도 있다. 대안으로, 선택스 엘리먼트 `use_rps_from_vps_flag`는 존재하지 않을 수도 있고 0과 동일하게 설정된 것으로 항상 유추될 수도 있다.
- [0135] 변수 `NumShortTermRefPicSets`은 다음과 같이 도출될 수 있다.
- [0136] `NumShortTermRefPicSets = num_short_term_ref_pic_sets`
- [0137] `if(use_rps_from_vps_flag)`
- [0138] `NumShortTermRefPicSets += num_vps_short_term_ref_pic_sets`
- [0139] 슬라이스 헤더 시맨틱스가 이제 설명될 것이다. 선택스 엘리먼트 `short_term_ref_pic_set_idx`는 현재 화상의 참조 화상 세트의 생성을 위해 사용될 수도 있는 액티브 시퀀스 파라미터 세트에서 특정된 단기 참조 화상 세트들의 리스트에 대한 인덱스를 특정한다. 선택스 엘리먼트 `short_term_ref_pic_set_idx`는 $\text{Ceil}(\log_2(\text{NumShortTermRefPicSets}))$ 비트들에 의해 표현될 수도 있다. `short_term_ref_pic_set_idx`의 값은 0 부터 `num_short_term_ref_pic_sets - 1`까지의 범위에 있을 수도 있는데, `num_short_term_ref_pic_sets`는 액티브 시퀀스 파라미터 세트로부터의 선택스 엘리먼트이다.
- [0140] 변수 `StRpsIdx`는 다음과 같이 도출될 수도 있다.
- [0141] `if(short_term_ref_pic_set_sps_flag)`
- [0142] `StRpsIdx = short_term_ref_pic_set_idx`
- [0143] `else`
- [0144] `StRpsIdx = NumShortTermRefPicSets`
- [0145] 0과 동일하게 설정된 선택스 엘리먼트 `tiles_fixed_structure_idc`는 코딩된 비디오 시퀀스에서 임의의 화상에 의해 참조되는 각각의 화상 파라미터 세트가 0과 동일하게 설정된 `tiles_or_entropy_coding_sync_idc`를 가짐을 나타낸다. 1과 동일하게 설정된 선택스 엘리먼트 `tiles_fixed_structure_idc`는, 코딩된 비디오 시퀀스에서 임의의 화상에 의해 참조되는 각각의 화상 파라미터 세트가, 존재하는 경우, 선택스 엘리먼트들 `num_tile_columns_minus1`, `num_tile_rows_minus1`, `uniform_spacing_flag`, `column_width[i]`, `row_height[i]` 및 `loop_filter_across_tiles_enabled_flag`의 동일한 값을 가짐을 나타낸다. 2와 동일하게 설정된 선택스 엘리먼트 `tiles_fixed_structure_idc`는, 코딩된 비디오 시퀀스에서 화상들에 의해 참조되는 상이한 화상 파라미터 세트들에서의 타일들의 선택스 엘리먼트들이 동일한 값을 가질 수도 있거나 또는 가지지 않을 수도 있다는 것을 나타낸다. `tiles_fixed_structure_idc`의 값은 0 부터 2까지의 범위에 있을 수도 있다. 선택스 엘리먼트 `tiles_fixed_structure_flag`가 존재하지 않는 경우, 그것은 2와 동일한 것으로 유추된다.
- [0146] 1과 동일하게 설정된 선택스 엘리먼트 `tiles_fixed_structure_flag`의 시그널링은, 코딩된 비디오 시퀀스에서의 각각의 화상이 멀티 스레드식 디코딩의 경우에 작업부하 할당에 유용할 수도 있다는 것과 동일한 방식으로 분산된 동일한 수의 타일들을 가지는 것을 디코더에 보장하는 것일 수도 있다.
- [0147] 위에서 설명된 제 1 예와 유사한 제 2 예가 이제 설명될 것이다. 이 제 2 예에서, SPS 내에 남아있는 선택스 엘리먼트들이 VPS에 존재할 수도 있고 SPS에 조건부로 존재할 수도 있다. 이 예에 따른 VPS 및 SPS의 선택스 및 시맨틱스는 아래의 표 7 내지 표 9에서 변경되고 설명된다.

표 7

비디오 파라미터 세트 RBSP 선택스

video_parameter_set_rbsp() {	기술자
<code>vps_max_temporal_layers_minus1</code>	u(3)
<code>vps_max_layers_minus1</code>	u(5)

[0148]

profile_space	u(3)
profile_idc	u(5)
for(j = 0; j < 32; j++)	
profile_compatibility_flag[i]	u(1)
constraint_flags	u(16)
level_idc	u(8)
level_lower_temporal_layers_present_flag	u(1)
if(level_lower_temporal_layers_present_flag)	
for (i = 0; i < vps_max_temporal_layers_minus1; i++)	
level_idc_temporal_subset[i]	u(8)
video_parameter_set_id	u(5)
vps_temporal_id_nesting_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
bit_depth_luma_minus8	u(2)
bit_depth_chroma_minus8	u(2)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
for (i = 0; i <= vps_max_temporal_layers_minus1; i++) {	
bitrate_info_present_flag[i]	u(1)
frm_rate_info_present_flag[i]	u(1)
if(bitrate_info_present_flag[i]) {	
avg_bitrate[i]	u(16)
max_bitrate[i]	u(16)
}	
if(frm_rate_info_present_flag[i]) {	
constant_frm_rate_idc[i]	u(2)
avg_frm_rate[i]	u(16)
}	
}	
next_essential_info_byte_offset	u(12)
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)
}	
for (i = 0, nalHrdPresent = 0, vclHrdPresent = 0;	
i <= vps_max_temporal_layers_minus1; i++) {	
nal_hrd_parameters_present_flag[i]	u(1)
if(nal_hrd_parameters_present_flag[i]) {	
hrd_parameters(nalHrdPresent)	
nalHrdPresent++	
}	
vcl_hrd_parameters_present_flag[i]	u(1)

[0149]

if(vcl_hrd_parameters_present_flag[i]) {	
hrd_parameters(vclHrdPresent)	
vclHrdPresent++	
}	
if(nalHrdPresent + vclHrdPresent == 1) {	
low_delay_hrd_flag	u(1)
sub_pic_cpb_params_present_flag	u(1)
num_units_in_sub_tick	u(32)
}	
vps_max_dec_pic_buffering[i]	ue(v)
vps_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
vui_parameters_present_flag	u(1)
if (vui_parameters_present_flag)	
vui_parameters()	
num_vps_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_vps_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
optional_sps_parameters()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

[0150]

표 8

시퀀스 파라미터 세트 RBSP 선택스

seq_parameter_set_rbsp() {	기술자
seq_parameter_set_id	ue(v)
video_parameter_set_id	ue(v)
num_short_term_ref_pic_sets	ue(v)
use_rps_from_vps_flag	u(1)
for(i = 0; i < num_short_term_ref_pic_sets; i++){	
idx = use_rps_from_vps_flag ? num_vps_short_term_ref_pic_sets + i : i	
short_term_ref_pic_set(idx)	
}	
sps_parameters_override_flag	u(1)
if(sps_parameters_override_flag) {	
optional_sps_parameters()	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0151]

표 9

옵션의 SPS 파라미터들

optional_sps_parameters() {	
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)
}	
log2_max_pic_order_cnt_lsb_minus4	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	u(1)
if(scaling_list_enable_flag) {	
sps_scaling_list_data_present_flag	u(1)
if(sps_scaling_list_data_present_flag)	

[0152]

scaling_list_param()	
{	
chroma_pred_from_luma_enabled_flag	u(1)
transform_skip_enabled_flag	u(1)
seq_loop_filter_across_slices_enabled_flag	u(1)
asymmetric_motion_partitions_enabled_flag	u(1)
nsrqt_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)
if(log2_min_coding_block_size_minus3 == 0)	
inter_4x4_enabled_flag	u(1)
long_term_ref_pics_present_flag	u(1)
sps_temporal_mvp_enable_flag	u(1)
tiles_fixed_structure_idc	u(2)
}	

[0153]

[0154]

옵션의 SPS 파라미터들의 시맨틱스가 이제 설명될 것이다. 이 선택스 구조에서의 선택스 엘리먼트들 및 선택스 구조들의 시맨틱스는 제 1 예에서 특정된 것과 동일한 선택스 엘리먼트 이름들을 갖는 SPS에서의 그들 선택스 엘리먼트들과 동일한 시맨틱스를 가진다.

[0155]

시퀀스 파라미터 세트 RBSP 시맨틱스가 이제 설명될 것이다. 1과 동일하게 설정된 선택스 엘리먼트 sps_parameters_override_flag는, 시퀀스 파라미터 세트에서 특정된 바와 같은 pcm_enabled_flag로부터 tiles_fixed_structure_idc까지의 선택스 엘리먼트들 및 선택스 구조들의 값들이 참조되는 비디오 파라미터 세트에서 특정된 바와 동일한 선택스 엘리먼트들 및 선택스 구조들의 값들을 오버라이드함을 특정한다. 0과 동일하게 설정된 선택스 엘리먼트 sps_parameters_override_flag는, 참조되는 비디오 파라미터 세트에서 특정된 바와 같은 pcm_enabled_flag로부터 tiles_fixed_structure_idc까지의 선택스 엘리먼트들 및 선택스 구조들의 값들이 사용되고 있음을 특정한다.

[0156]

표 7에 도시된 선택스 엘리먼트 next_essential_byte_offset은 표 2를 참조하여 위에서 설명된 방식으로 MANE (29) 및/또는 비디오 디코더 (30) 에 의해 프로세싱되고 파싱될 수도 있다. 마찬가지로, 선택스 엘리먼트들

video_parameter_set_id, profile_idc 및 profile_space는 또한, 비디오 인코더 (20) 에 의해 생성되고 위에서 설명된 방식으로 비디오 디코더 (30) 에 의해 프로세싱되고 파싱될 수도 있다.

[0157] 제 3 예는 제 1 예의 상위집합이다. 이 제 3 예에서, 신택스는 확장들을 구현에 더 용이하게 하는 방식으로 설계될 수도 있다. 덧붙여서, VPS의 확장은 이 예에서 지원될 수도 있다. 제 1 예에서의 대응부분과 정확히 동일한 신택스 테이블의 신택스 설계 또는 시맨틱스 설계는 존재하지 않는다. 제 3 예는 표 10 내지 표 19를 참조하여 아래에서 설명된다.

표 10

비디오 파라미터 세트 RBSP 신택스 (기본 사양만)

video_parameter_set_rbsp() {	기술자
vps_max_temporal_layers_minus1	u(3)
vps_max_layers_minus1	u(5)
profile_level_info(0, vps_max_temporal_layers_minus1)	
video_parameter_set_id	u(5)
vps_temporal_id_nesting_flag	u(1)
rep_format_info(0, 0)	
bitrate_framerate_info(0, vps_max_temporal_layers_minus1)	
next_essential_info_byte_offset	u(12)
rep_format_info(0, 1)	
for(i = 0; i <= vps_max_temporal_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
hrd_info(0, vps_max_temporal_layers_minus1)	
vui_vps_set(0)	
num_vps_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_vps_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0158]

표 11

비디오 파라미터 세트 RBSP 선택스 (확장을 포함)

video_parameter_set_rbsp() {	기술자
vps_max_temporal_layers_minus1	u(3)
vps_max_layers_minus1	u(5)
profile_level_info(0, vps_max_temporal_layers_minus1)	
video_parameter_set_id	u(5)
vps_temporal_id_nesting_flag	u(1)
rep_format_info(0, 0)	
bitrate_framerate_info(0, vps_max_temporal_layers_minus1)	
next_essential_info_byte_offset	u(12)
rep_format_info(0, 1)	
for(i = 0; i <= vps_max_temporal_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
hrd_info(0, vps_max_temporal_layers_minus1)	
vui_vps_set(0)	
num_vps_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_vps_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
~~vps_extension_flag~~	~~u(1)~~
bit_equal_to_one	u(1)
vps_extension()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

[0159]

표 12

프로파일 및 레벨 정보 테이블 선택스

profile_level_info(index, NumTempLevelMinus1) {	
profile_space	u(3)
profile_idc	u(5)
for(j = 0; j < 32; j++)	
profile_compatibility_flag[j]	u(1)
constraint_flags	u(16)
level_idc	u(8)
level_lower_temporal_layers_present_flag	u(1)
if(level_lower_temporal_layers_present_flag)	
for(i = 0; i < NumTempLevelMinus1; i++)	
level_idc[i]	u(8)
profileLevelInfoIdx = index	
}	

[0160]

표 13

표현 포맷 정보 테이블 선택스

rep_format_info(index, partIdx) {	
if(!partIdx) {	
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
bit_depth_luma_minus8	u(2)
bit_depth_chroma_minus8	u(2)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
}	
else {	
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)
}	
}	
repFormatInfoIdx = index	
}	

[0161]

표 14

비트레이트 및 프레임 레이트 정보 테이블 선택스

bitrate_framerate_info(TempLevelLow, TempLevelHigh) {	
for(i = TempLevelLow; i <= TempLevelHigh; i++) {	
bitrate_info_present_flag[i]	u(1)
frm_rate_info_present_flag[i]	u(1)
if(bitrate_info_present_flag[i]) {	
avg_bitrate[i]	u(16)
max_bitrate[i]	u(16)
}	
if(frm_rate_info_present_flag[i]) {	
constant_frm_rate_idc[i]	u(2)
avg_frm_rate[i]	u(16)
}	
}	
}	

[0162]

표 15

HRD 시간 동작 포인트들의 정보 테이블 선택스

hrd_info(TempLevelLow, TempLevelHigh) {	
for (i = TempLevelLow, nalHrdPresent = 0, vclHrdPresent = 0;	
i <= NumTempLevelMinus1; i++) {	
nal_hrd_parameters_present_flag[i]	u(1)
if(nal_hrd_parameters_present_flag[i]) {	
hrd_parameters(nalHrdPresent)	
nalHrdPresent++	
}	
vcl_hrd_parameters_present_flag[i]	u(1)
if(vcl_hrd_parameters_present_flag[i]) {	
hrd_parameters(vclHrdPresent)	
vclHrdPresent++	
}	
if(nalHrdPresent + vclHrdPresent == 1) {	
low_delay_hrd_flag	u(1)
sub_pic_cpb_params_present_flag	u(1)
num_units_in_sub_tick	u(32)
}	
}	

[0163]

표 16

VUI VPS 세트 테이블 선택스

vui_vps_set(index) {	
vui_video_parameters_present_flag	u(1)
if(vui_video_parameters_present_flag)	
vui_parameters()	
vuiVpsSetIndex = index	
}	

[0164]

표 17

VPS 확장 선택스

vps_extension() {	
byte_aligned_bits	u(v)
num_additional_profile_level_info	u(4)
num_additional_rep_fromat_info	u(3)
num_additional_dependency_operation_points	u(8)
extension_type	u(3)
for(i =0; i< num_additional_profile_level_info; i++)	
profile_level_info(i + 1, vps_max_temporal_layers_minus1)	
for(i = 0; i < num_additional_rep_fromat_info; i++)	
rep_format_info(i+1, 0)	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
if(num_additional_profile_level_info)	
profile_level_index[k]	u(4)
if(num_additional_rep_fromat_info)	
ref_format_index[k]	u(3)
applicable_lowest_temporal_id[k]	u(3)
applicable_highest_temporal_id[k]	u(3)
}	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
bitrate_framerate_info(applicable_lowest_temporal_id[k],	
applicable_highest_temporal_id[k])	
}	
// layer dependency	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
if(extension_type == 0) { /* Condition always true for 3DV */	
depth_included_flag[k]	u(1)
num_target_output_views_minus1[k]	u(5)
num_depended_layers[k]	u(5)
for(j = 0; j < num_target_output_views_minus1[k]; j++)	
layer_id[k][j]	u(5)
for(j = 0; j < num_depended_layers[k]; j++)	
dependent_layer_id[k][j]	u(5)
}	
else if(extension_type == 1)	
layer_id[k]	u(5)
}	
for(i = 0; i < num_additional_rep_fromat_info; i++) {	
rep_format_info(i + 1, 1)	

[0165]

// boundary of the fixed-length and ue(v)	
//vui	
num_additional_vui_vps_set_info	ue(v)
for(i = 0; i < num_additional_vui_vps_set_info; i++)	
vui_vps_set(i + 1)	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
if(num_additional_vui_vps_set_info)	
vui_vps_set_idx	ue(v)
hrd_info(applicable_lowest_temporal_id[k],	
applicable_highest_temporal_id[k])	
}	
}	

[0166]

[0167]

비디오 파라미터 세트 RBSP 시맨틱스가 이제 설명될 것이다. 선택스 엘리먼트 byte_aligned_bits는 VPS NAL 단위에서 num_additional_profile_level_info 바이트 앞의 비트들이 정렬되게 하는 가능한 비트들을 특정한다. 선택스 엘리먼트 byte_aligned_bits는 0 부터 7까지의 범위에 있다. 선택스 엘리먼트 num_additional_profile_level_info는 VPS에서 존재하는 부가적인 프로파일 및 레벨 정보 테이블의 수를 특정한다. 선택스 엘리먼트 num_additional_rep_fromat_info는 VPS에서 존재하는 부가적인 표현 포맷 정보 테이블들의 수를 특정한다. 선택스 엘리먼트 num_additional_dependency_operation_points는, 시간적 확장성에 무관하게, 비트스트림에 추가로 존재하는 의존 동작 포인트들의 수를 특정한다. 각각의 의존 동작 포인트는 동일한 계층 구조를 각각 가지는 시간적 서브 동작 포인트들을 포함할 수도 있다. 선택스 엘리먼트 extension_type은 현재 비트스트림의 확장의 유형을 특정하는데, 0이 3DV에 대응하고 1이 SVC에 대응한다.

신택스 엘리먼트 `profile_level_index[k]`는 현재 k 번째 의존 동작 포인트에 대해 VPS에서 시그널링된 레벨 정보 테이블에 대한 인덱스를 나타낸다. 신택스 엘리먼트 `ref_format_index`는 현재 k 번째 의존 동작 포인트에 대해 VPS에서 시그널링된 표현 포맷 정보 테이블에 대한 인덱스를 나타낸다.

[0168] 신택스 엘리먼트 `applicable_lowest_temporal_id[k]` 및 `applicable_highest_temporal_id[k]`는 k 번째 의존 동작 포인트의 시그널링된 시간적 서브 동작 포인트들에 대응하는 최저 `temporal_id` 값 및 최고 `temporal_id` 값을 각각 특정한다. 대안으로, 신택스 엘리먼트들 `applicable_lowest_temporal_id[k]` 및 `applicable_highest_temporal_id[k]`는 양쪽 모두 시그널링되지 않고 각각 0 및 `vps_max_temporal_layers_minus1`과 동일한 것으로 유추된다. 대안으로, 신택스 엘리먼트 `applicable_lowest_temporal_id[k]`는 시그널링되지 않고 0과 동일한 것으로 유추된다. 대안으로, 신택스 엘리먼트 `applicable_highest_temporal_id[k]`는 시그널링되지 않고 `vps_max_temporal_layers_minus1`과 동일한 것으로 유추된다.

[0169] 1과 동일한 신택스 엘리먼트 `depth_included_flag[k]`는 현재 3DV 의존 동작 포인트가 깊이를 포함함을 나타낸다. 0과 동일한 이 플래그는 현재 3DV 동작 포인트가 깊이를 포함하지 않음을 나타낸다. 대안으로, 신택스 엘리먼트 `depth_included_flag[k]`는 시그널링되지 않으며, 따라서 깊이 VCL NAL 단위가 `layer_id_plus1`에 의존함을 나타낸다.

[0170] 신택스 엘리먼트 `num_target_output_views_minus1[k]` 더하기 1은 k 번째 의존 동작 포인트에서의 타겟 출력 뷰들의 수를 특정한다. 신택스 엘리먼트 `num_dependent_layers[k]`는 현재 k 번째 의존 동작 포인트를 디코딩하기 위한 의존 계층들의 수를 나타낸다. 신택스 엘리먼트 `layer_id[k][j]`는 k 번째 의존 동작 포인트의 j 번째 타겟 출력 뷰의 `layer_id`를 나타낸다. 신택스 엘리먼트 `dependent_layer_id[k][j]`는 k 번째 의존 동작 포인트의 j 번째 의존성 뷰의 `layer_id`를 나타낸다. 하나의 대안에서, 플래그가, `dependent_layer_id[k][j]` 직후에, `direct_dependent_flag[k][j]`로서 시그널링된다.

[0171] 신택스 엘리먼트 `direct_dependent_flag[k][j]`는 뷰간 RPS를 도출하는데 사용될 j 번째 의존성 뷰가 직접 의존성 뷰인지의 여부를 나타낸다. 신택스 엘리먼트 `layer_id[k]`는 현재 k 번째 (SVC) 의존 동작 포인트의 최고 `layer_id`를 나타낸다. 대안으로, `num_target_output_views_minus1[k]`, `num_dependent_layers[k]`, `layer_id[k][j]` 및 `dependent_layer_id[k][j]`는 `ue(v)`로서 시그널링될 수 있다.

[0172] 신택스 엘리먼트 `num_additional_vui_vps_set_info`는 VPS에서 존재하는 추가적인 VUI VPS 세트 테이블의 수를 특정할 수도 있다.

[0173] 프로파일 및 레벨 정보 테이블 시맨틱스에 대해, 신택스 엘리먼트 `profileLevelInfoIdx`는 프로파일 및 레벨 정보 테이블의 인덱스를 나타낸다. 표현 포맷 정보 테이블 시맨틱스에 대해, 신택스 엘리먼트 `repFormatInfoIdx`는 표현 포맷 정보 테이블의 인덱스를 나타낸다.

[0174] 표 7에 도시된 신택스 엘리먼트 `next_essential_byte_offset`은 표 2를 참조하여 위에서 설명된 방식으로 MANE (29) 및/또는 비디오 디코더 (30)에 의해 프로세싱되고 파싱될 수도 있다.

[0175] VUI VPS 세트 테이블 시맨틱스에 대해, 신택스 엘리먼트 `vuiVpsSetIndex`는 VUI VPS 세트 테이블의 인덱스를 나타낸다.

[0176] 대안으로, 각각의 뷰의 뷰 의존성이 다음과 같이 SPS에서 시그널링될 수 있다:

표 18

seq_parameter_set_rbsp() {	기술자
seq_parameter_set_id	ue(v)
video_parameter_set_id	ue(v)
num_short_term_ref_pic_sets	ue(v)
use_rps_from_vps_flag	u(1)
for(i = 0; i < num_short_term_ref_pic_sets; i++){	
idx = use_rps_from_vps_flag ? num_vps_short_term_ref_pic_sets + i : i	
short_term_ref_pic_set(idx)	
}	
sps_parameters_override_flag	u(1)
if(sps_parameters_override_flag) {	
optional_sps_parameters()	
~~sps_extension_flag~~	~~u(1)~~
bit_equal_to_one	u(1)
num_reference_views	ue(v)
for(i = 0; i < num_reference_views ; i++){	
ref_view_layer_id[i]	ue(v)
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0177]

[0178]

신택스 엘리먼트 num_reference_views는 뷰간 RPS 서브세트를 구축하는데 사용되는 텍스처 또는 깊이 뷰들의 최대 수를 나타낸다. 신택스 엘리먼트 ref_view_layer_id[i]는 뷰간 RPS 서브세트에서 i번째 뷰간 (전용) 참조 화상을 나타내는데 사용된 i번째 텍스처/깊이 뷰의 layer_id를 식별한다.

[0179]

대안으로, VPS 확장은 다음과 같이 시그널링될 수 있다. 신택스 엘리먼트 extension_type이 SVC를 나타내는 경우, 신택스 엘리먼트 num_additional_dependency_operation_points는 시그널링되지 않고 vps_max_layers_minus1과 동일한 것으로 도출된다. 액세스 단위 내의 VCL NAL 단위들이 layer_id의 비-내림 차순으로 있다는 제약이 주어진다. MVC에서, 신택스 엘리먼트 layer_id는 view_idx와 동등하다. 3DV에서, 신택스 엘리먼트 view_idx는 layer_id에 의해 다음과 같이 계산될 수도 있다: view_idx = (layer_idx>>1).

표 19

vps_extension() {	
byte_aligned_bits	u(v)
num_additional_profile_level_info	u(4)
num_additional_rep_fromat_info	u(3)
extension_type	u(3)
if(extension_type != 1) {	
num_additional_dependency_operation_points	u(8)
depth_present_flag	u(1)
for(i =0; i< num_additional_profile_level_info; i++)	
profile_level_info(i + 1, vps_max_temporal_layers_minus1)	
for(i = 0; i < num_additional_rep_fromat_info; i++)	
rep_format_info(i+1, 0)	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
if(num_additional_profile_level_info)	
profile_level_index[k]	u(4)
if(num_additional_rep_fromat_info)	
ref_format_index[k]	u(3)
applicable_lowest_temporal_id[k]	u(3)
applicable_highest_temporal_id[k]	u(3)
}	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
bitrate_framerate_info(applicable_lowest_temporal_id[k],	
applicable_highest_temporal_id[k])	
}	
// layer dependency	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
if(extension_type == 0) { /* Condition always true for 3DV */	
if(depth_present_flag)	
depth_included_flag[k]	u(1)
num_target_output_views_minus1[k]	u(5)
num_dependent_layers[k]	u(5)
for(j = 0; j < num_target_output_views_minus1[k]; j++)	
layer_id[k][j]	u(5)
for(j = 0; j < num_dependent_layers[k]; j++)	
dependent_layer_id[k][j]	u(5)
}	
else if(extension_type == 1)	
layer_id[k]	u(5)
}	
for(i = 0; i < num_additional_rep_fromat_info; i++) {	
rep_format_info(i + 1, 1)	
// boundary of the fixed-length and ue(v)	
//vui	
num_additional_vui_vps_set_info	ue(v)
for(i = 0; i < num_additional_vui_vps_set_info; i++)	

[0180]

vui_vps_set(i + 1)	
for(k=0; k< num_additional_dependency_operation_points; k++) {	
if(num_additional_vui_vps_set_info)	
vui_vps_set_idx	ue(v)
hrd_info(applicable_lowest_temporal_id[k],	
applicable_highest_temporal_id[k])	
}	
}	

[0181]

[0182]

1과 동일하게 설정된 선택스 엘리먼트 depth_present_flag는 깊이를 포함하는 동작 포인트들이 있을 수도 있다는 것을 나타낸다. 0과 동일하게 설정된 선택스 엘리먼트 depth_present_flag는 동작 포인트가 깊이를 포함하지 않음을 나타낸다.

[0183]

선택스 엘리먼트 num_target_output_views_minus1[k] 더하기 1은 k번째 의존 동작 포인트에서 타겟 출력 뷰들의 수를 특정하는데 사용될 수도 있다. 선택스 엘리먼트 num_dependent_layers[k]는 현재 k번째 의존 동작 포인트를 디코딩하는 의존 계층들의 수를 나타내는데 사용될 수도 있다. depth_present_flag가 1과 동일하게 설정되는 경우, 의존 계층은 깊이 뷰 또는 텍스처 뷰 중 어느 하나 또는 양쪽 모두일 수도 있다. 선택스

엘리먼트 layer_id[k][j]는 k번째 의존 동작 포인트의 j번째 타겟 출력 텍스처 뷰의 layer_id를 나타낸다. 존재한다면, 텍스처 뷰에 연관된 깊이 뷰의 layer_id는 layer_id[k][j] + 1이다.

[0184] 대안으로, 선택 엘리먼트 view_idx[k][j]는 layer_id[k][j] 대신 각각의 타겟 출력 뷰에 대해 시그널링될 수도 있다. 각각의 view_idx[k][j]에 대해, 대응하는 텍스처 뷰의 layer_id는 (view_idx[k][j] << depth_present_flag) 이다. depth_included_flag[k]가 1과 동일하면, 대응하는 깊이 뷰의 layer_id는 (view_idx[k][j] << depth_present_flag) + 1인데, 이 경우 depth_present_flag가 1이어야만 하므로 이것은 (view_idx[k][j] << 1) + 1이다. 대안으로, 선택 엘리먼트 layer_id[k][j]는 view_idx[k][j]로 변경될 수도 있고, u(v) 코딩되며, 여기서 길이는 5 - depth_present_flag이다. 대안으로, 선택 엘리먼트 layer_id[k][j]는 view_idx[k][j]로 변경될 수도 있고 u(v) 코딩되며, 여기서 길이는 5 - depth_included[k]이다.

[0185] 제 4 예는 제 2 예의 상위집합이다. 선택스는 확장 친화적 방식으로 설계된다. 덧붙여서, VPS의 확장이 이 예에서 제공된다. 제 2 예의 대응부분과 정확히 동일한 선택스 테이블의 선택스 설계 또는 시맨틱스 설계는 존재하지 않는다.

표 20

비디오 파라미터 세트 RBSP 선택스 (기본 사양만)

video_parameter_set_rbsp() {	기술자
vps_max_temporal_layers_minus1	u(3)
vps_max_layers_minus1	u(5)
profile_level_info(0, vps_max_temporal_layers_minus1)	
video_parameter_set_id	u(5)
vps_temporal_id_nesting_flag	u(1)
rep_format_info(0, 0)	
bitrate_framerate_info(0, vps_max_temporal_layers_minus1)	
next_essential_info_byte_offset	u(12)
rep_format_info(0, 1)	
for(i = 0; i <= vps_max_temporal_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
hrd_info(0, vps_max_temporal_layers_minus1)	
vui_vps_set(0)	
num_vps_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_vps_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
optional_sps_parameters()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
}	
rbsp_trailing_bits()	
}	

[0186]

표 21

비디오 파라미터 세트 RBSP 선택스 (확장을 포함)

video_parameter_set_rbsp() {	기술자
vps_max_temporal_layers_minus1	u(3)
vps_max_layers_minus1	u(5)
profile_level_info(0, vps_max_temporal_layers_minus1)	
video_parameter_set_id	u(5)
vps_temporal_id_nesting_flag	u(1)
rep_format_info(0, 0)	
bitrate_framerate_info(0, vps_max_temporal_layers_minus1)	
next_essential_byte_offset	u(12)
rep_format_info(0, 1)	
for(i = 0; i <= vps_max_temporal_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)

[0187]

}	
hrd_info(0, vps_max_temporal_layers_minus1)	
vui_vps_set(0)	
num_vps_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_vps_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
optional_sps_parameters()	
~~vps_extension_flag~~	u(1)
bit_equal_to_one	u(1)
vps_extension()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbbsp_trailing_bits()	
}	

[0188]

[0189]

표 21에 도시된 선택스 엘리먼트 next_essential_byte_offset은 표 2를 참조하여 위에서 설명된 방식으로 MANE (29) 및/또는 비디오 디코더 (30) 에 의해 프로세싱되고 파싱될 수도 있다.

[0190]

도 4는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20) 를 도시하는 블록도이다. 비디오 인코더 (20) 는, 예를 들어, 표 1 내지 표 21을 참조하여 위에서 설명된 선택스 구조들을 생성할 수도 있다. 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라 코딩 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 공간적 예측에 의존하여, 주어진 비디오 프레임 또는 화상 내의 비디오에서 공간적 리던던시를 감소시키거나 또는 제거한다. 인터 코딩은 시간적 예측에 의존하여, 비디오 시퀀스의 인접한 프레임들 또는 화상들 내의 비디오에서 시간적 리던던시를 감소시키거나 또는 제거한다. 인트라 모드 (I 모드) 는 여러 공간 기반 압축 모드들 중 임의의 것을 말할 수도 있다. 단방향 예측 (P 모드) 및 양방향 예측 (B 모드) 과 같은 인터 모드들은 여러 시간 기반 압축 모드들 중 임의의 것을 말할 수도 있다.

[0191]

도 4의 예에서, 비디오 인코더 (20) 는 구획화 유닛 (35), 예측 프로세싱 유닛 (41), 필터 유닛 (63), 화상 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 구비한다. 예측 프로세싱 유닛 (41) 은 모션 추정 유닛 (42), 모션 보상 유닛 (44), 및 인트라 예측 프로세싱 유닛 (46) 을 구비한다. 비디오 블록 복원을 위해, 비디오 인코더 (20) 는 또한 역 양자화 유닛 (58), 역 변환 프로세싱 유닛 (60), 및 합산기 (62) 를 구비한다. 필터 유닛 (63) 은 디블로킹 필터, 적응 루프 필터 (adaptive loop filter; ALF), 및 샘플 적응 오프셋 (sample adaptive offset; SAO) 필터와 같은 하나 이상의 루프 필터들을 나타내고자 하는 것이다. 비록 필터 유닛 (63) 이 도 4에서 인 루프 필터 (in loop filter) 인 것으로 도시되고 있지만, 다른 구성들에서, 필터 유닛 (63) 은 포스트 (post) 루프 필터로서 구현될 수도 있다. 도 4는 비디오 인코더 (20) 에 의해 생성된 인코딩된 비디오 데이터에 대해 추가적인 프로세싱을 수행할 수도 있는 포스트 프로세싱 디바이스 (57) 를 또한 도시한다. 오프셋 선택스 엘리먼트를 갖는 파라미터 세트를 생성하는 것을 포함하는 본 개시물의 기법들은, 일부 경우들에서 비디오 인코더 (20) 에 의해 구현될 수

도 있다. 그러나, 다른 경우들에서, 본 개시물의 기법들은 포스트 프로세싱 디바이스 (57) 에 의해 구현될 수도 있다.

[0192] 도 4에 도시된 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고 구획화 유닛 (35) 은 그 데이터를 비디오 블록들로 구획한다. 이 구획화는 또한 슬라이스들, 타일들, 또는 다른 큰 단위들로의 구획화, 뿐만 아니라, 예컨대, LCU들 및 CU들의 쿼드트리 구조에 따른 비디오 블록 구획화를 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 슬라이스는 다수의 비디오 블록들로 (및 가능하다면 타일들이라고 지칭된 비디오 블록들의 세트들로) 나누어질 수도 있다. 예측 프로세싱 유닛 (41) 은 복수의 가능한 코딩 모드들 중 하나, 이를테면 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나를 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨) 에 기초하여 현재 비디오 블록에 대해 선택할 수도 있다. 예측 프로세싱 유닛 (41) 은 결과적인 인트라 또는 인터 코딩된 블록을 잔차 블록 데이터를 생성하는 합산기 (50) 및 참조 화상로서 사용하기 위해 인코딩된 블록을 복원하는 합산기 (62) 에 제공할 수도 있다.

[0193] 예측 프로세싱 유닛 (41) 내의 인트라 예측 프로세싱 유닛 (46) 은 코딩된 현재 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 대하여 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 하나 이상의 참조 화상들에서의 하나 이상의 예측 블록들에 대하여 현재 비디오 블록의 인터 예측 코딩을 수행하여 시간적 압축을 제공한다.

[0194] 모션 추정 유닛 (42) 은 비디오 시퀀스에 대한 미리 결정된 패턴에 따라 비디오 슬라이스에 대한 인터 예측 모드를 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들, B 슬라이스들 또는 GPB 슬라이스들로서 지정할 수도 있다. 모션 추정 유닛 (42) 과 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만 개념적 목적들을 위해 별개로 예시된다. 모션 추정 유닛 (42) 에 의해 수행되는 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어, 참조 화상 내의 예측 블록에 대한 현재 비디오 프레임 또는 화상 내의 비디오 블록의 PU의 변위 (displacement) 를 나타낼 수도 있다.

[0195] 예측 블록은 차의 절대값 합 (SAD), 차의 제곱 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 화소 차이의 관점에서 코딩될 비디오 블록의 PU에 밀접하게 매칭되는 것으로 확인되는 블록이다. 일부 예들에서, 비디오 인코더 (20) 는 화상 메모리 (64) 에 저장된 참조 화상들의 서브-정수 (sub-integer) 화소 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 화상의 1/4 화소 위치들, 1/8 화소 위치들, 또는 다른 분수 (fractional) 화소 위치들의 값들을 보간할 수도 있다. 그러므로, 모션 추정 유닛 (42) 은 풀 (full) 화소 위치들 및 분수 화소 위치들에 대한 모션 검색을 수행하고 분수 화소 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0196] 모션 추정 유닛 (42) 은 PU의 위치와 참조 화상의 예측 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 화상은 화상 메모리 (64) 에 저장된 하나 이상의 참조 화상들을 각각 식별하는 제 1 참조 화상 리스트 (리스트 0) 또는 제 2 참조 화상 리스트 (리스트 1) 로부터 선택될 수도 있다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0197] 모션 보상 유닛 (44) 에 의해 수행되는 모션 보상은 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐지하는 것 또는 생성하는 것, 가능하다면 서브-화소 (sub-pixel) 정밀도로 보간들을 수행하는 것을 수반할 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터를 수신 시, 모션 보상 유닛 (44) 은 참조 화상 리스트들 중 하나에서 모션 벡터가 가리키는 예측 블록을 위치결정할 수도 있다. 비디오 인코더 (20) 는, 코딩되고 있는 현재 비디오 블록의 화소 값들로부터 예측 블록의 화소 값들을 감산하여 화소 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 화소 차이 값들은 블록에 대한 잔차 데이터를 형성하고, 루마 및 크로마 차이 성분들 양쪽 모두를 포함할 수도 있다. 합산기 (50) 는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 모션 보상 유닛 (44) 은 또한 비디오 슬라이스의 비디오 블록들의 디코딩 시에 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스에 연관된 선택스 엘리먼트들을 생성할 수도 있다.

[0198] 인트라 예측 프로세싱 유닛 (46) 은, 위에서 설명된 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 수행된 인터 예측에 대한 대안으로서 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 프

로세싱 유닛 (46) 은 현재 블록을 인코딩하는데 사용하기 위한 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 프로세싱 유닛 (46) 은 예컨대, 개별 인코딩 패스들 동안에 다양한 인트라 예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 프로세싱 유닛 (46) (또는 일부 예들에서, 모드 선택 유닛 (40)) 은 테스트된 모드들로부터 사용할 적절한 인트라 예측 모드를 선택할 수도 있다. 예를 들어, 인트라 예측 프로세싱 유닛 (46) 은 다양한 테스트된 인트라 예측 모드들에 대한 레이트 왜곡 분석을 이용하여 레이트 왜곡 값들을 계산하고, 테스트된 모드들 중에서 최상의 레이트 왜곡 특성들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록과 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러) 의 양, 뿐만 아니라 인코딩된 블록을 생성하는데 사용된 비트 레이트 (다시 말하면, 비트들의 수) 를 결정한다. 인트라 예측 프로세싱 유닛 (46) 은 어떤 인트라 예측 모드가 그 블록에 대한 최상의 레이트 왜곡 값을 나타내는지 결정하기 위해 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 계산할 수도 있다.

[0199] 어느 경우이나, 블록에 대한 인트라 예측 모드를 선택한 후, 인트라 예측 프로세싱 유닛 (46) 은 그 블록에 대한 선택된 인트라 예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (56) 에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 본 개시물의 기법들에 따라 그 선택된 인트라 예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는, 다양한 블록들에 대한 인코딩 콘텍스트들의 정의들, 및 가장 가능성 있는 인트라 예측 모드, 인트라 예측 모드 인덱스 테이블, 및 그 콘텍스트들의 각각에 대한 사용을 위한 수정된 인트라 예측 모드 인덱스 테이블의 표시들을, 복수의 인트라 예측 모드 인덱스 테이블들 및 복수의 수정된 인트라 예측 모드 인덱스 테이블들 (또한 코드워드 매핑 테이블들이라고 지칭됨) 을 포함할 수도 있는 송신된 비트스트림 구성 데이터에 포함시킬 수도 있다.

[0200] 예측 프로세싱 유닛 (41) 이 인터 예측 또는 인트라 예측 중 어느 하나를 통해 현재 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 인코더 (20) 는 현재 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 잔차 블록에서의 잔차 비디오 데이터는, 하나 이상의 TU들에 포함되고 변환 프로세싱 유닛 (52) 에 적용될 수도 있다. 변환 프로세싱 유닛 (52) 은 변환, 이를테면 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환을 사용하여 잔차 비디오 데이터를 잔차 변환 계수들로 변환시킨다. 변환 프로세싱 유닛 (52) 은 잔차 비디오 데이터를 화소 도메인으로부터 변환 도메인, 이를테면 주파수 도메인으로 컨버팅할 수도 있다.

[0201] 변환 프로세싱 유닛 (52) 은 결과적인 변환 계수들을 양자화 유닛 (54) 에 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더욱 감소시킨다. 양자화 프로세스는 그 계수들의 일부 또는 전부에 연관된 비트 깊이를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 수정될 수도 있다. 일부 예들에서, 양자화 유닛 (54) 은 그 후, 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안으로, 엔트로피 인코딩 유닛 (56) 이 그 스캔을 수행할 수도 있다.

[0202] 양자화에 이어, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 콘텍스트 적응 가변 길이 코딩 (CAVLC), 콘텍스트 적응 이진 산술 코딩 (CABAC), 선택스 기반 콘텍스트 적응 이진 산술 코딩 (SBAC), 확률 간격 구획화 엔트로피 (PIPE) 코딩 또는 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩에 이어, 인코딩된 비트스트림은 비디오 디코더 (30) 로 송신되거나 또는 비디오 디코더 (30) 에 의한 나중의 송신 또는 취출을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (56) 은 코딩되고 있는 현재 비디오 슬라이스에 대한 모션 벡터들 및 다른 선택스 엘리먼트들을 또한 엔트로피 인코딩할 수도 있다.

[0203] 역 양자화 유닛 (58) 과 역 변환 프로세싱 유닛 (60) 은 역 양자화 및 역 변환을 각각 적용하여, 참조 화상의 참조 블록으로서 나중에 사용하기 위해 화소 도메인에서 잔차 블록을 복원한다. 모션 보상 유닛 (44) 은 참조 화상 리스트들 중 하나의 참조 화상 리스트 내의 참조 화상들 중 하나의 참조 화상의 예측 블록에 잔차 블록을 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한 하나 이상의 보간 필터들을 복원된 잔차 블록에 적용하여 모션 추정에서 사용하기 위한 서브-정수 화소 값들을 계산할 수도 있다. 합산기 (62) 는 복원된 잔차 블록을 모션 보상 유닛 (44) 에 의해 생성된 모션 보상된 예측 블록에 가산하여, 화상 메모리 (64) 에서의 저장을 위한 참조 블록을 생성한다. 참조 블록은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 후속 비디오 프레임 또는 화상에서 블록을 인터 예측하기 위한 참조 블록으로서 사용될 수도 있다.

[0204] 이런 방식으로, 도 4의 비디오 인코더 (20) 는 위의 표 1 내지 표 21에서 설명된 선택스를 생성하도록 구성된

비디오 인코더의 일 예를 나타낸다. 비디오 인코더 (20) 는, 예를 들어, 위에서 설명된 바와 같은 VPS, SPS, PPS, 및 APS 파라미터 세트들을 생성할 수도 있다. 하나의 예에서, 비디오 인코더 (20) 는 하나의 초기 고정 길이 신택스 엘리먼트들 다음에 오프셋 신택스 엘리먼트를 포함하는 코딩된 비디오 데이터에 대한 파라미터 세트를 생성할 수도 있다. 하나 이상의 초기 고정 길이 신택스 엘리먼트들은, 예를 들어, 세션 협상에 관련된 정보를 포함할 수도 있다. 오프셋 신택스 엘리먼트는 파라미터 세트가 MANE에 의해 프로세싱되는 경우 스킵될 다수의 바이트들을 나타낼 수도 있다. 스킵될 다수의 바이트들은, 예를 들어, 하나 이상의 가변 길이 신택스 엘리먼트들을 포함할 수도 있다. 비디오 인코더 (20) 는 파라미터 세트 내에서, 스킵된 바이트들 뒤에, 추가적인 고정 길이 신택스 엘리먼트들을 포함할 수도 있다. 추가적인 고정 길이 신택스 엘리먼트들은, 예를 들어, 비디오 데이터의 다른 계층에 관련된 정보를 포함할 수도 있다. 하나의 예에서, 초기 고정 길이 신택스 엘리먼트들은 기본 계층에 대한 세션 협상에 관련된 정보를 포함할 수도 있는 반면, 추가적인 고정 길이 신택스 엘리먼트들은 비-기본 계층에 대한 세션 협상에 관련된 정보를 포함할 수도 있다.

[0205] 비디오 인코더 (20) 는 하나 이상의 가변 길이 신택스 엘리먼트들을 코딩하는데 사용된 비트들의 수에 기초하여 오프셋 신택스 엘리먼트에 대한 값을 결정할 수도 있다. 예를 들어, 제 1 VPS에 대해 스킵될 신택스 엘리먼트들이 2 비트, 3 비트, 및 5 비트의 3 개의 고정 길이 신택스 엘리먼트들 뿐만 아니라 2 비트, 4, 비트, 5 비트, 및 3 비트의 4 개의 가변 길이 신택스 엘리먼트들을 포함한다고 가정한다. 이 예에서, 고정 길이 신택스 엘리먼트들은 총 10 비트를 포함하는 반면 가변 길이 신택스 엘리먼트들은 총 14 비트를 포함한다. 따라서, 제 1 VPS에 대해, 비디오 인코더 (20) 는 오프셋 신택스 엘리먼트의 값을 스킵될 24 비트들 (예컨대 3 바이트) 을 포함하는 24로 설정할 수도 있다. 제 2 VPS에 대해, 고정된 신택스 엘리먼트들에 대한 비트들의 수는 다시 10이 될 것이지만, 가변 길이 신택스 엘리먼트들을 위해 사용된 비트들의 수는 상이할 수도 있다. 따라서, 제 2 VPS에 대해, 비디오 인코더 (20) 는 오프셋 신택스 엘리먼트에 대한 값을 상이한 값으로 설정할 수도 있다.

[0206] 본 개시물의 기법들은 대체로 비디오 인코더 (20) 에 관해 설명되었지만, 위에서 언급했듯이, 본 개시물의 기법들 중 일부는 포스트 프로세싱 디바이스 (57) 에 의해 또한 구현될 수도 있다. 예를 들어, 포스트 프로세싱 디바이스 (57) 는 비디오 인코더 (20) 에 의해 생성된 비디오 데이터의 다수의 계층들에 대한 VPS를 생성할 수도 있다.

[0207] 도 5는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더 (30) 를 도시하는 블록도이다. 비디오 디코더 (30) 는, 예를 들어, 표 1 내지 표 21에 관해 위에서 설명된 신택스 구조들을 프로세싱하고 파싱하도록 구성될 수도 있다. 도 5의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역 양자화 유닛 (86), 역 변환 프로세싱 유닛 (88), 합산기 (90), 필터 유닛 (91), 및 화상 메모리 (92) 를 구비한다. 예측 프로세싱 유닛 (81) 은 모션 보상 유닛 (82) 과 인트라 예측 프로세싱 유닛 (84) 을 구비한다. 비디오 디코더 (30) 는, 일부 예들에서, 도 4로부터 비디오 인코더 (20) 에 관해 설명된 인코딩 패스 (pass) 에 일반적으로 역인 디코딩 패스를 수행할 수도 있다.

[0208] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 는 인코딩된 비디오 비트스트림을 네트워크 엔터티 (79) 로부터 수신할 수도 있다. 네트워크 엔터티 (79) 는, 예를 들어, 서버, MANE, 비디오 편집기/스플라이서 (splicer), 또는 위에서 설명된 기법들 중 하나 이상을 구현하도록 구성된 다른 이러한 디바이스일 수도 있다. 네트워크 엔터티 (79) 는 비디오 인코더 (20) 를 포함할 수도 있거나 또는 포함하지 않을 수도 있다. 위에서 설명된 바와 같이, 본 개시물에서 설명된 기법들 중 일부는 네트워크 엔터티 (79) 가 인코딩된 비디오 비트스트림을 비디오 디코더 (30) 로 송신하기 전에 네트워크 엔터티 (79) 에 의해 구현될 수도 있다. 일부 비디오 디코딩 시스템들에서, 네트워크 엔터티 (79) 와 비디오 디코더 (30) 는 별개의 디바이스들의 일부들일 수도 있지만, 다른 경우들에서, 네트워크 엔터티 (79) 에 관해 설명된 기능성은 비디오 디코더 (30) 를 포함하는 동일한 디바이스에 의해 구현될 수도 있다.

[0209] 네트워크 엔터티 (79) 는 비디오 비트스트림에 연관된 파라미터 세트에 대한 하나 이상의 초기 신택스 엘리먼트들을 프로세싱하며; 파라미터 세트 내에서 스킵될 신택스 엘리먼트들을 식별하는 파라미터 세트에 대한 오프셋 신택스 엘리먼트를 파라미터 세트에서 수신하고, 그 오프셋 신택스 엘리먼트에 기초하여, 파라미터 세트 내에서 신택스 엘리먼트들을 스킵하도록 구성된 비디오 프로세싱 디바이스의 일 예를 나타낸다. 네트워크 엔터티 (79) 는 파라미터 세트에서의 하나 이상의 추가적인 신택스 엘리먼트들을 또한 프로세싱할 수도 있다. 하나 이상의 추가적인 신택스 엘리먼트들은 파라미터 세트에서 스킵된 신택스 엘리먼트들 뒤에 있다.

- [0210] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 선택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 그 비디오 블록들은, 예를 들어, 비디오 인코더 (20) 로부터 하나 이상의 MANE들, 이를테면 도 1의 MANE (29) 또는 도 5의 네트워크 엔티티 (79) 를 통해 비디오 디코더 (30) 로 라우팅될 수도 있다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 그 비트스트림을 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들, 및 다른 선택스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들 및 다른 선택스 엘리먼트들을 예측 프로세싱 유닛 (81) 에 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 선택스 엘리먼트들을 수신할 수도 있다.
- [0211] 위에서 도입된 바와 같이, 엔트로피 디코딩 유닛 (80) 은 하나 이상의 파라미터 세트들, 이를테면 VPS, SPS, PPS, 및 APS에서의 고정 길이 선택스 엘리먼트들 및 가변 길이 선택스 엘리먼트들 양쪽 모두를 프로세싱하고 파싱할 수도 있다. 파라미터 세트들 중 하나 이상, 예를 들어 VPS에서, 비디오 디코더 (30) 는 본 개시물에서 설명된 바와 같은 오프셋 선택스 엘리먼트를 수신할 수도 있다. 오프셋 선택스 엘리먼트의 수신에 응답하여, 비디오 디코더 (30) 는 오프셋 선택스 엘리먼트의 값을 기본적으로 무시할 수 있다. 예를 들어, 비디오 디코더 (30) 는 오프셋 선택스 엘리먼트를 수신할 수도 있지만 임의의 선택스 엘리먼트들을 스킵하는 일 없이 오프셋 선택스 엘리먼트를 뒤따르는 가변 길이 선택스 엘리먼트들을 포함하는 선택스 엘리먼트들을 계속 디코딩할 수도 있다.
- [0212] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩되는 경우, 예측 프로세싱 유닛 (81) 의 인트라 예측 프로세싱 유닛 (84) 은 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를, 현재 프레임 또는 화상의 이전에 디코딩된 블록들로부터의 시그널링된 인트라 예측 모드 및 데이터에 기초하여 생성할 수도 있다. 비디오 프레임이 인터 코딩된 (즉, B, P, 또는 GPB) 슬라이스로서 코딩되는 경우, 예측 프로세싱 유닛 (81) 의 모션 보상 유닛 (82) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 모션 벡터들 및 다른 선택스 엘리먼트들에 기초하여, 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 그 예측 블록들은 참조 화상 리스트들 중 하나 내의 참조 화상들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는 화상 메모리 (92) 에 저장된 참조 화상들에 기초하여 디폴트 구축 기법들을 이용하여, 참조 프레임 리스트들, 리스트 0 및 리스트 1을 구축할 수도 있다.
- [0213] 모션 보상 유닛 (82) 은 모션 벡터들 및 다른 선택스 엘리먼트들을 파싱함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82) 은 수신된 선택스 엘리먼트들의 일부를 사용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예컨대, 인트라 또는 인터 예측), 인터 예측 슬라이스 유형 (예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 화상 리스트들 중 하나 이상에 대한 구축 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터 코딩된 비디오 블록에 대한 인터 예측 스테이터스, 및 현재 비디오 슬라이스에서의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.
- [0214] 모션 보상 유닛 (82) 은 보간 필터들에 기초하여 보간을 또한 수행할 수도 있다. 모션 보상 유닛 (82) 은 비디오 블록들의 인코딩 동안에 비디오 인코더 (20) 에 의해 사용된 것과 같은 보간 필터들을 사용하여 참조 블록들의 서브-정수 화소들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (82) 은 비디오 인코더 (20) 에 의해 사용된 보간 필터들을 수신된 선택스 엘리먼트들로부터 결정하고 그 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.
- [0215] 역 양자화 유닛 (86) 은 비트스트림으로 제공되고 엔트로피 디코딩 유닛 (80) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉, 탈양자화 (dequantize) 한다. 역 양자화 프로세스는 양자화의 정도와, 마찬가지로 적용되어야 할 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다. 역 변환 프로세싱 유닛 (88) 은 화소 도메인에서 잔차 블록들을 생성하기 위하여 역 변환, 예컨대, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 변환 계수들에 적용한다.
- [0216] 모션 보상 유닛 (82) 이 현재 비디오 블록에 대한 예측 블록을 모션 벡터들 및 다른 선택스 엘리먼트들에 기초하여 생성한 후, 비디오 디코더 (30) 는 역 변환 프로세싱 유닛 (88) 으로부터의 잔차 블록들을 모션 보상 유닛 (82) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (90) 는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 원한다면, 루프 필터들이 (코딩 루프

내 또는 코딩 루프 후 중 어느 하나에서) 화소 전이 (transition) 들을 부드럽게 하기 위해, 또는 그렇지 않으면 비디오 품질을 개선하기 위해 또한 사용될 수도 있다. 필터 유닛 (91) 은 디블로킹 필터, 적응 루프 필터 (adaptive loop filter; ALF), 및 샘플 적응 오프셋 (sample adaptive offset; SAO) 필터와 같은 하나 이상의 루프 필터들을 나타내고자 하는 것이다. 비록 필터 유닛 (91) 이 도 5에서 인 루프 필터인 것으로 도시되고 있지만, 다른 구성들에서, 필터 유닛 (91) 은 포스트 루프 필터로서 구현될 수도 있다. 주어진 프레임 또는 화상에서의 디코딩된 비디오 블록들은 그러면 화상 메모리 (92) 에 저장되며, 그 화상 메모리는 후속 모션 보상을 위해 사용되는 참조 화상들을 저장한다. 화상 메모리 (92) 는 또한 디스플레이 디바이스, 이를테면 도 1의 디스플레이 디바이스 (32) 상의 나중의 프레젠테이션을 위해 디코딩된 비디오를 저장한다.

[0217] 이런 방식으로, 도 5의 비디오 디코더 (30) 는 위의 표 1 내지 표 21에서 설명된 신택스를 파싱하도록 구성된 비디오 디코더의 일 예를 나타낸다. 비디오 디코더 (30) 는, 예를 들어, 위에서 설명된 바와 같은 VPS, SPS, PPS, 및 APS 파라미터 세트들을 파싱할 수도 있다.

[0218] 도 6은 네트워크 (150) 의 일부를 형성하는 디바이스들의 일 예의 세트를 도시하는 블록도이다. 이 예에서, 네트워크 (150) 는 라우팅 디바이스들 (154A, 154B) (라우팅 디바이스들 (154)) 과 트랜스코딩 (transcoding) 디바이스 (156) 를 구비한다. 라우팅 디바이스들 (154) 과 트랜스코딩 디바이스 (156) 는 네트워크 (150) 의 일부를 형성할 수도 있는 적은 수의 디바이스들을 나타내려고 하는 것이다. 다른 네트워크 디바이스들, 이를테면 스위치들, 허브들, 게이트웨이들, 방화벽들, 브리지들, 및 다른 이러한 디바이스들이 또한 네트워크 (150) 내에 포함될 수도 있다. 더구나, 부가적인 네트워크 디바이스들이 서버 디바이스 (152) 및 클라이언트 디바이스 (158) 사이에서 네트워크 경로를 따라 제공될 수도 있다. 일부 예들에서, 서버 디바이스 (152) 는 소스 디바이스 (12) (도 1) 에 대응할 수도 있는 반면, 클라이언트 디바이스 (158) 는 목적지 디바이스 (14) (도 1) 에 대응할 수도 있다. 라우팅 디바이스들 (154) 은, 예를 들어, 미디어 데이터를 라우팅하도록 구성된 MANE들일 수도 있다.

[0219] 대체로, 라우팅 디바이스들 (154) 은 네트워크 (150) 를 통해 네트워크 데이터를 교환하기 위해 하나 이상의 라우팅 프로토콜들을 구현한다. 대체로, 라우팅 디바이스들 (154) 은 네트워크 (150) 를 통해 루트들을 발견하기 위해 라우팅 프로토콜들을 실행한다. 이러한 라우팅 프로토콜들을 실행함으로써, 라우팅 디바이스 (154B) 는 자신으로부터 라우팅 디바이스 (154A) 를 통해 서버 디바이스 (152) 까지의 네트워크 루트를 발견할 수도 있다. 도 6의 다양한 디바이스들은 본 개시물의 기법들을 구현할 수도 있는 디바이스들의 예들을 나타낸다. 라우팅 디바이스들 (154) 은, 예를 들어, 본 개시물에 따라 파라미터 세트, 이를테면 VPS의 신택스 엘리먼트들을 파싱하도록 구성되는 미디어 인식 네트워크 엘리먼트들일 수도 있다. 예를 들어, 라우팅 디바이스들 (154) 은 VPS에서 하나 이상의 초기 고정 길이 신택스 엘리먼트들을 수신하고 그 고정 길이 신택스 엘리먼트들을 파싱하고 프로세싱할 수도 있다. 초기 고정 길이 신택스 엘리먼트들은, 예를 들어, 세션 협상에 관련된 신택스 엘리먼트들일 수도 있다. 라우팅 디바이스들 (154) 은, VPS에서, 오프셋 신택스 엘리먼트를 또한 수신할 수도 있다. 오프셋 신택스 엘리먼트는 스킵될 다수의 바이트들을 식별할 수도 있다. 라우팅 디바이스 (154) 는 특정 수의 바이트들을 스킵할 수 있고, 특정 수의 바이트들을 스킵한 후, VPS 내에서 고정 길이 신택스 엘리먼트들의 파싱 및 프로세싱을 재개할 수 있다. 스킵된 바이트들은 라우팅 디바이스들 (154) 이 파싱할 수 없는 하나 이상의 가변 길이 신택스 엘리먼트들을 포함할 수도 있는데, 라우팅 디바이스들 (154) 이 엔트로피 디코딩 동작들을 수행할 수 없기 때문이다.

[0220] 도 7은 본 개시물의 기법들에 따라 오프셋 신택스 엘리먼트를 프로세싱하는 방법의 일 예를 도시하는 흐름도이다. 도 7의 기법들은 도 1의 MANE (29) 또는 도 6에서의 라우팅 디바이스들 (154) 중 하나의 라우팅 디바이스와 같은 네트워크 디바이스에 관해 설명될 것이다. 네트워크 엔티티는 비디오 비트스트림에 연관된 파라미터 세트에 대한 하나 이상의 초기 신택스 엘리먼트들을 프로세싱한다 (171). 하나 이상의 초기 신택스 엘리먼트들은 부가적으로 고정 길이 신택스 엘리먼트들을 포함하고 오프셋 신택스 엘리먼트에 선행할 수도 있다. 하나 이상의 초기 신택스 엘리먼트들은 세션 협상에 관련된 정보를 포함하는 신택스 엘리먼트들을 포함할 수도 있다. 더욱이, 하나 이상의 초기 신택스 엘리먼트들은 비디오 데이터의 기본 계층에 대한 신택스 엘리먼트들을 포함하고 하나 이상의 부가적인 신택스 엘리먼트들은 비디오 데이터의 비-기본 계층에 대한 신택스 엘리먼트들을 포함한다.

[0221] 네트워크 엔티티는 비디오 비트스트림에서 파라미터 세트에 대한 오프셋 신택스 엘리먼트를 수신한다 (172). 오프셋 신택스 엘리먼트는 파라미터 세트 내에서 스킵될 다수의 비트들을 식별한다. 오프셋 신택스 엘리먼트는, 예를 들어, 비디오 파라미터 세트의 일부일 수도 있다. 스킵될 다수의 비트들은, 예를 들어, 가변 길이 코딩을 사용하여 코딩된 하나 이상의 신택스 엘리먼트들에 대응할 수도 있다. 오프셋 신택스 엘리먼트

에 기초하여, 네트워크 엔티티는 파라미터 세트 내에서 다수의 비트들을 스킵한다 (173). 네트워크 엔티티는 파라미터 세트에서의 하나 이상의 부가적인 선택스 엘리먼트들을 프로세싱한다 (174). 하나 이상의 부가적인 선택스 엘리먼트들은 파라미터 세트에서 스킵된 다수의 비트들 뒤에 있다. 하나 이상의 부가적인 선택스 엘리먼트들은 부가적인 고정 길이 선택스 엘리먼트들일 수도 있고, 하나 이상의 부가적인 선택스 엘리먼트들은 오프셋 선택스 엘리먼트를 뒤따르고 스킵될 비트들을 뒤따를 수도 있다.

[0222] 도 8은 본 개시물의 기법들에 따라 오프셋 선택스 엘리먼트를 프로세싱하는 방법의 일 예를 도시하는 흐름도이다. 도 8의 기법들은 비디오 데이터를 인코딩하거나 또는 인코딩된 비디오 데이터를 프로세싱하도록 구성된 비디오 프로세싱 디바이스를 참조하여 설명될 것이다. 인코딩된 비디오 데이터를 프로세싱하도록 구성된 비디오 프로세싱 디바이스들의 예들은 도 1 및 도 4의 비디오 인코더 (20) 와 도 4의 포스트 프로세싱 디바이스 (57) 를 포함한다. 비디오 프로세싱 디바이스들은 비디오 비트스트림에 연관된 파라미터 세트에 대한 하나 이상의 초기 선택스 엘리먼트들을 생성한다 (181). 하나 이상의 초기 선택스 엘리먼트들은 고정 길이 선택스 엘리먼트들을 포함할 수도 있고, 하나 이상의 초기 선택스 엘리먼트들은 오프셋 선택스 엘리먼트에 선행할 수도 있다. 하나 이상의 초기 선택스 엘리먼트들은 세션 협상에 관련된 정보를 포함하는 선택스 엘리먼트들을 포함할 수도 있다. 하나 이상의 초기 선택스 엘리먼트들은 비디오 데이터의 기본 계층에 대한 선택스 엘리먼트들을 포함할 수도 있고, 하나 이상의 부가적인 선택스 엘리먼트들은 비디오 데이터의 비-기본 계층에 대한 선택스 엘리먼트들을 포함할 수도 있다.

[0223] 비디오 프로세싱 디바이스들은 파라미터 세트에 대한 오프셋 선택스 엘리먼트를 생성한다 (182). 오프셋 선택스 엘리먼트는 파라미터 세트 내에서 스킵될 다수의 비트들을 식별할 수도 있다. 오프셋 선택스 엘리먼트는 비디오 파라미터 세트의 일부일 수도 있다. 비디오 프로세싱 디바이스는 스킵될 하나 이상의 선택스 엘리먼트들을 생성한다 (183). 스킵될 비트들은 스킵될 하나 이상의 선택스 엘리먼트들을 포함한다. 스킵될 하나 이상의 선택스 엘리먼트들은 가변 길이 코딩을 사용하여 코딩된 하나 이상의 선택스 엘리먼트들을 포함할 수도 있다. 비디오 프로세싱 디바이스는 파라미터 세트에서 하나 이상의 부가적인 선택스 엘리먼트들을 생성한다 (184). 하나 이상의 부가적인 선택스 엘리먼트들은 파라미터 세트에서 스킵될 다수의 비트들 뒤에 있다. 하나 이상의 부가적인 선택스 엘리먼트들은 부가적인 고정 길이 선택스 엘리먼트들을 포함할 수도 있으며, 하나 이상의 부가적인 선택스 엘리먼트들은 오프셋 선택스 엘리먼트를 뒤따르고 스킵될 비트들을 뒤따를 수도 있다.

[0224] 도 9는 본 개시물의 기법들에 따라 오프셋 선택스 엘리먼트를 디코딩하는 방법의 일 예를 도시하는 흐름도이다. 도 9의 기법들은 비디오 디코더, 이를테면 도 1 및 도 5의 비디오 디코더 (30) 를 참조하여 설명될 것이다. 그 비디오 디코더는 비디오 비트스트림에 연관된 파라미터 세트에 대한 하나 이상의 초기 선택스 엘리먼트들을 디코딩한다 (191). 비디오 디코더는 비디오 비트스트림에서 파라미터 세트에 대한 오프셋 선택스 엘리먼트를 수신한다 (192). 오프셋 선택스 엘리먼트는 파라미터 세트 내에서 스킵될 다수의 비트들을 식별한다. 비디오 디코더는 스킵될 비트들을 디코딩한다 (193). 일부 예들에서, 비디오 디코더는 스킵될 비트들에 포함된 가변 길이 선택스 엘리먼트들을 디코딩하기 위해 엔트로피 디코딩을 수행함으로써 스킵될 비트들을 디코딩한다. 비디오 디코더는, 예를 들어, 스킵될 비트들을 디코딩할 수도 있는데, 그 비트들이 비디오 데이터가 MANE와 같은 비디오 프로세싱 디바이스에 의해 프로세싱되고 있을 때에 스킵될 것이지만 그 비트들은 비디오 데이터를 디코딩하기 위해 필요할 수도 있기 때문이다. MANE는, 비디오 디코더와는 대조적으로, 비디오 데이터를 완전히 디코딩하지 않고서도 비디오 데이터에 대한 특정한 프로세싱을 수행하기 위하여 비트들을 스킵할 수도 있다. 어떤 경우들에서, MANE는 비디오 데이터를 디코딩하는데 필요한 모든 능력들을 심지어 가지지 않을 수도 있다.

[0225] 도 10은 본 개시물의 기법들에 따라 VPS를 프로세싱하는 방법의 일 예를 도시하는 흐름도이다. 도 10의 기법들은 일반 비디오 프로세싱 디바이스를 참조하여 설명될 것이다. 비디오 프로세싱 디바이스는 도 1의 MANE (29) 또는 도 6의 라우팅 디바이스들 (154) 중 하나의 라우팅 디바이스와 같은 네트워크 디바이스에 대응할 수도 있다. 비디오 프로세싱 디바이스는 도 1 및 도 4의 비디오 디코더 (30) 와 같은 비디오 디코더에 부가적으로 대응할 수도 있다. 비디오 프로세싱 디바이스는 비디오 파라미터 세트에서, 세션 협상에 관련된 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 수신한다 (201). 비디오 프로세싱 디바이스는 비디오 데이터에서 비디오 파라미터 세트를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 수신한다 (202). 제 1 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 비디오 데이터에서 비디오 파라미터 세트를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 수

신한다 (203). 제 2 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 하나 이상의 선택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 프로세싱한다 (204).

[0226] 하나 이상의 선택스 엘리먼트들은, 예를 들어, 고정 길이 선택스 엘리먼트들일 수도 있고, 비디오 파라미터 세트에서, 임의의 가변 길이 코딩된 선택스 엘리먼트들에 선행할 수도 있다. 하나 이상의 선택스 엘리먼트들은 비디오 코딩 표준의 프로파일을 식별하는 선택스 엘리먼트를 포함할 수도 있다. 하나 이상의 선택스 엘리먼트들은 비디오 코딩 표준의 레벨을 식별하는 선택스 엘리먼트를 추가로 또는 대안으로 포함할 수도 있다. 레벨은, 예를 들어, 다수의 레벨들 중 비디오 코딩 표준의 프로파일에 연관된 하나의 레벨에 대응할 수도 있다.

[0227] 하나 이상의 선택스 엘리먼트들은 제 1 선택스 엘리먼트와 제 2 선택스 엘리먼트를 포함할 수도 있다. 제 1 선택스 엘리먼트는 제 2 선택스 엘리먼트를 해석하기 위한 콘텍스트를 식별할 수도 있고, 제 2 선택스 엘리먼트는 프로파일들의 그룹을 식별할 수도 있다. 비디오 프로세싱 디바이스는, 비디오 파라미터 세트에서, 프로파일들의 그룹으로부터의 프로파일에 각각 연관되는 하나 이상의 호환성 플래그들을 수신할 수도 있다. 하나 이상의 호환성 플래그들의 각각에 대한 값은 비디오 데이터가 프로파일들의 그룹으로부터의 연관된 프로파일과 호환되는지를 식별할 수도 있다.

[0228] 하나 이상의 선택스 엘리먼트들은 비디오 데이터에 연관된 최대 시간 (temporal) 레벨을 식별하는 제 1 선택스 엘리먼트와 비디오 데이터의 시간 계층이 최대 시간 레벨보다 낮은 레벨을 갖는지를 식별하는 제 2 선택스 엘리먼트를 또한 포함할 수도 있다. 비디오 데이터의 시간 계층이 최대 시간 레벨보다 낮은 레벨을 가짐을 나타내는 제 2 선택스 엘리먼트에 응답하여, 비디오 프로세싱 디바이스는 비디오 데이터의 하나 이상의 시간 계층들에 대한 레벨들을 식별하는 추가적인 선택스 엘리먼트들을 수신할 수도 있다.

[0229] 비디오 프로세싱 디바이스가 비디오 디코더인 경우들에서, 비디오 디코더는 비디오 블록들의 제 1 세트 및 비디오 블록들의 제 2 세트를 디코딩할 수도 있다. 비디오 프로세싱 디바이스가 MANE인 경우들에서, MANE는 비디오 블록들의 제 1 세트 및 비디오 블록들의 제 2 세트를 클라이언트 디바이스로 포워딩할 수도 있다.

[0230] 도 11은 본 개시물의 기법들에 따른 VPS 내의 포함을 위해 선택스 엘리먼트들을 생성하는 방법의 일 예를 도시하는 흐름도이다. 도 8의 기법들은 비디오 데이터를 인코딩하거나 또는 인코딩된 비디오 데이터를 프로세싱하도록 구성된 비디오 프로세싱 디바이스를 참조하여 설명될 것이다. 인코딩된 비디오 데이터를 프로세싱하도록 구성된 비디오 프로세싱 디바이스들의 예들은 도 1 및 도 4의 비디오 인코더 (20) 와 도 4의 포스트 프로세싱 디바이스 (57) 를 포함한다. 비디오 프로세싱 디바이스는 비디오 파라미터 세트 내의 포함을 위해, 세션 협상에 관련된 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 생성한다 (211). 비디오 프로세싱 디바이스는 비디오 데이터 내의 포함을 위해 비디오 파라미터 세트를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 생성한다 (212). 제 1 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 비디오 데이터 내의 포함을 위해 비디오 파라미터 세트를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 생성한다 (213). 제 2 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 하나 이상의 선택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 인코딩한다 (214).

[0231] 하나 이상의 선택스 엘리먼트들은, 예를 들어, 고정 길이 선택스 엘리먼트들일 수도 있고, 비디오 파라미터 세트에서, 임의의 가변 길이 코딩된 선택스 엘리먼트들에 선행할 수도 있다. 하나 이상의 선택스 엘리먼트들은 비디오 코딩 표준의 프로파일을 식별하는 선택스 엘리먼트를 포함할 수도 있다. 하나 이상의 선택스 엘리먼트들은 비디오 코딩 표준의 레벨을 식별하는 선택스 엘리먼트를 추가로 또는 대안으로 포함할 수도 있다. 레벨은, 예를 들어, 다수의 레벨들 중 비디오 코딩 표준의 프로파일에 연관된 하나의 레벨에 대응할 수도 있다.

[0232] 하나 이상의 선택스 엘리먼트들은 제 1 선택스 엘리먼트와 제 2 선택스 엘리먼트를 포함할 수도 있다. 제 1 선택스 엘리먼트는 제 2 선택스 엘리먼트를 해석하기 위한 콘텍스트를 식별할 수도 있고, 제 2 선택스 엘리먼트는 프로파일들의 그룹을 식별할 수도 있다. 비디오 프로세싱 디바이스는, 비디오 파라미터 세트에서, 프로

파일들의 그룹으로부터의 프로파일에 각각 연관되는 하나 이상의 호환성 플래그들을 수신할 수도 있다. 하나 이상의 호환성 플래그들의 각각에 대한 값은 비디오 데이터가 프로파일들의 그룹으로부터의 연관된 프로파일과 호환되는지를 식별할 수도 있다.

[0233] 하나 이상의 선택스 엘리먼트들은 비디오 데이터에 연관된 최대 시간 (temporal) 레벨을 식별하는 제 1 선택스 엘리먼트와 비디오 데이터의 시간 계층이 최대 시간 레벨보다 낮은 레벨을 갖는지를 식별하는 제 2 선택스 엘리먼트를 또한 포함할 수도 있다. 비디오 데이터의 시간 계층이 최대 시간 레벨보다 낮은 레벨을 가짐을 나타내는 제 2 선택스 엘리먼트에 응답하여, 비디오 프로세싱 디바이스는 비디오 데이터의 하나 이상의 시간 계층들에 대한 레벨들을 식별하는 부가적인 선택스 엘리먼트들을 수신할 수도 있다.

[0234] 도 12는 본 개시물의 기법들에 따라 VPS를 프로세싱하는 방법의 일 예를 도시하는 흐름도이다. 도 12의 기법들은 일반 비디오 프로세싱 디바이스를 참조하여 설명될 것이다. 비디오 프로세싱 디바이스는 도 1의 MANE (29) 또는 도 6의 라우팅 디바이스들 (154) 중 하나의 라우팅 디바이스와 같은 네트워크 디바이스에 대응할 수도 있다. 비디오 프로세싱 디바이스는 도 1 및 도 4의 비디오 디코더 (30) 와 같은 비디오 디코더에 부가적으로 대응할 수도 있다. 비디오 프로세싱 디바이스는 비디오 파라미터 세트에서, HRD 파라미터들에 관련된 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 수신한다 (221). 비디오 프로세싱 디바이스는 비디오 데이터에서 비디오 파라미터 세트를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 수신한다 (222). 제 1 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 비디오 데이터에서 비디오 파라미터 세트를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 수신한다 (223). 제 2 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 하나 이상의 선택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 프로세싱한다 (224).

[0235] 도 13은 본 개시물의 기법들에 따른 VPS 내의 포함을 위해 선택스 엘리먼트들을 생성하는 방법의 일 예를 도시하는 흐름도이다. 도 13의 기법들은 비디오 데이터를 인코딩하거나 또는 인코딩된 비디오 데이터를 프로세싱하도록 구성된 비디오 프로세싱 디바이스를 참조하여 설명될 것이다. 인코딩된 비디오 데이터를 프로세싱하도록 구성된 비디오 프로세싱 디바이스들의 예들은 도 1 및 도 4의 비디오 인코더 (20) 와 도 4의 포스트 프로세싱 디바이스 (57) 를 포함한다. 비디오 프로세싱 디바이스는 비디오 파라미터 세트 내의 포함을 위해, HRD 파라미터들에 관련된 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 생성한다 (231). 비디오 프로세싱 디바이스는 비디오 데이터 내의 포함을 위해, 비디오 파라미터 세트를 식별하는 제 1 선택스 엘리먼트를 포함하는 제 1 시퀀스 파라미터 세트를 생성한다 (232). 제 1 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 전체 화상들에 적용하는 선택스 엘리먼트들의 제 1 그룹을 포함하는 제 1 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 비디오 데이터 내의 포함을 위해 비디오 파라미터 세트를 식별하는 제 2 선택스 엘리먼트를 포함하는 제 2 시퀀스 파라미터 세트를 생성한다 (233). 제 2 시퀀스 파라미터 세트는 비디오 데이터의 하나 이상의 상이한 전체 화상들에 적용하는 선택스 엘리먼트들의 제 2 그룹을 포함하는 제 2 선택스 구조를 포함한다. 비디오 프로세싱 디바이스는 하나 이상의 선택스 엘리먼트들에 기초하여, 제 1 파라미터 세트에 연관된 비디오 블록들의 제 1 세트와 제 2 파라미터 세트에 연관된 비디오 블록들의 제 2 세트를 인코딩한다 (234).

[0236] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적 (non-transitory) 인 유형의 (tangible) 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0237] 비제한적인 예로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 저장,

자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 이룰테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들 이룰테면 적외선, 라디오, 및/또는 마이크로파가 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않고, 대신 비일시적 (non-transient), 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc) 는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, 디스크 (disk) 들은 보통 데이터를 자기적으로 재생하지만, 디스크 (disc) 들은 레이저들로 광학적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

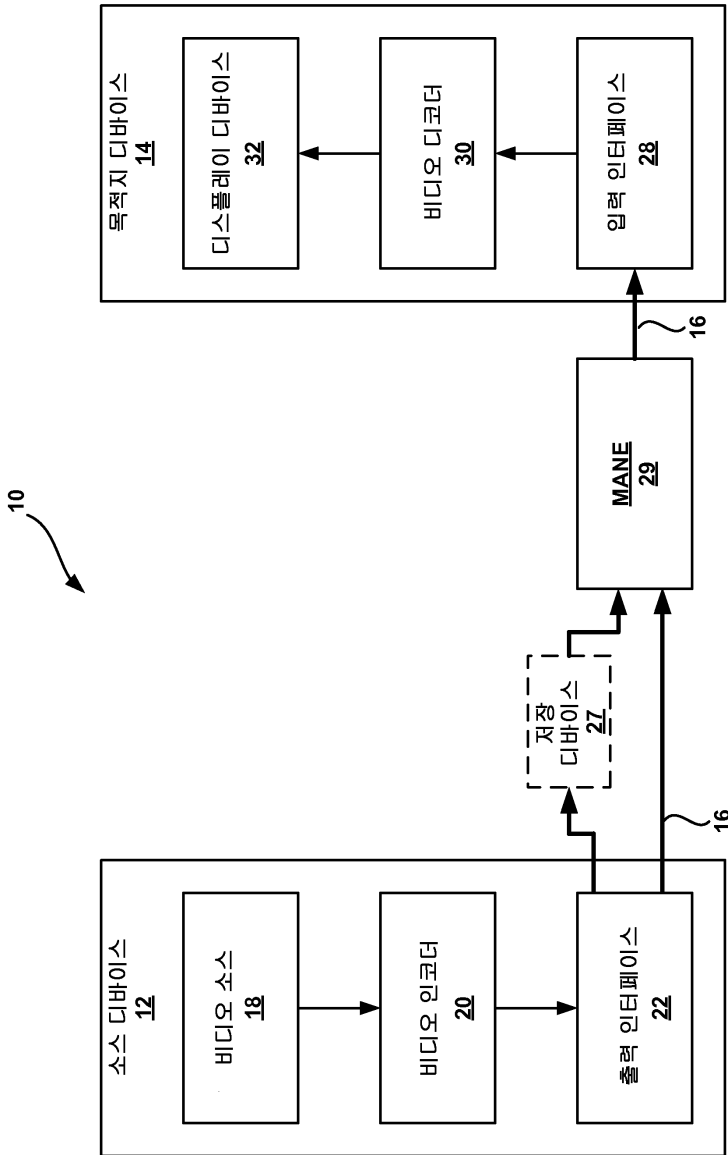
[0238] 명령들은 하나 이상의 프로세서들, 이룰테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서"는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 말할 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0239] 본 개시물의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋) 를 포함하는 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

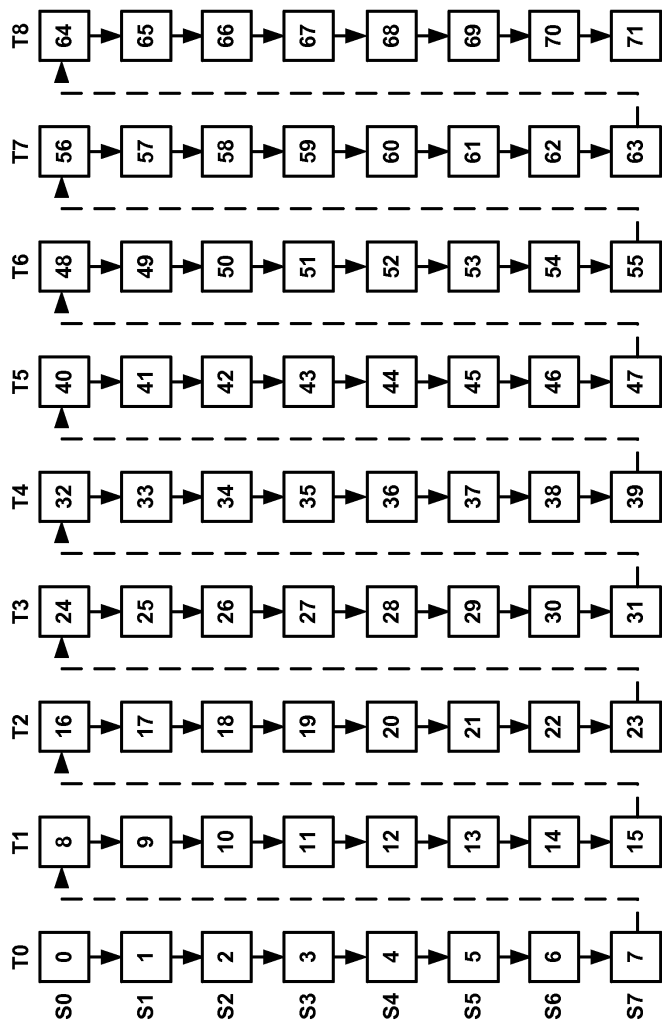
[0240] 다양한 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

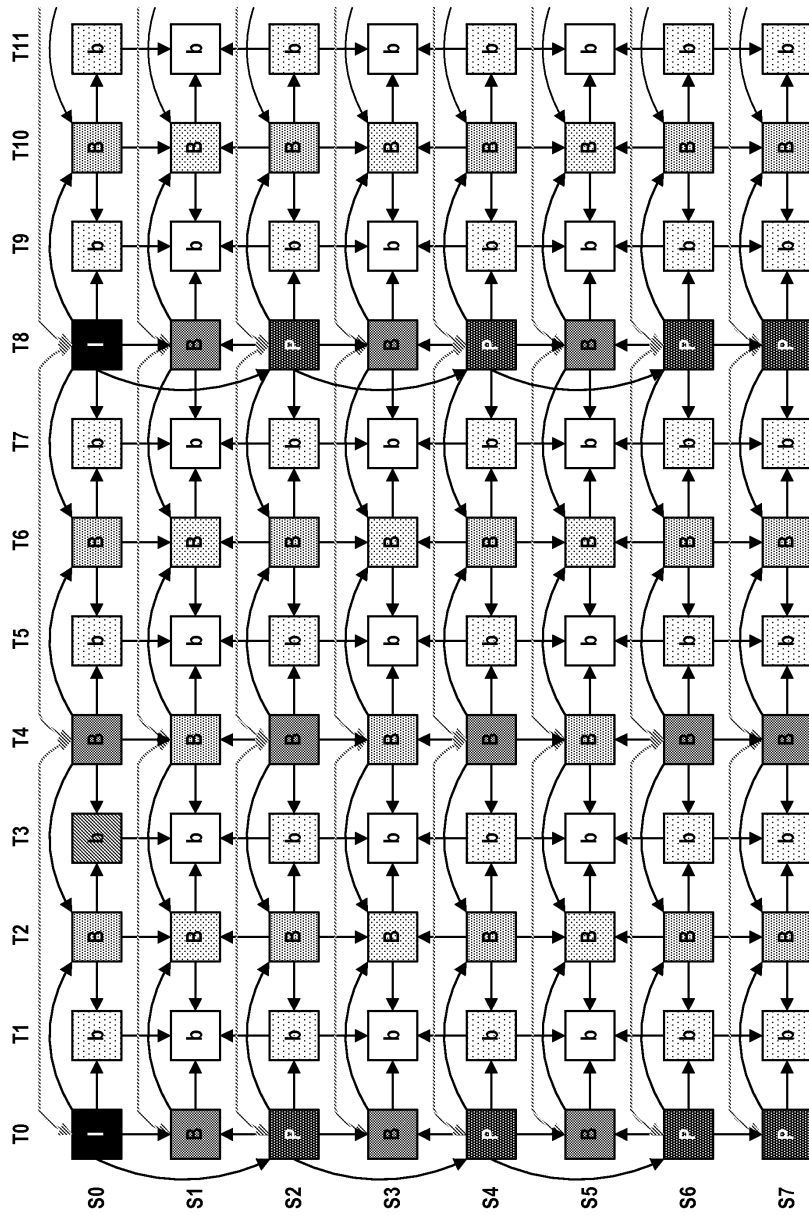
도면1



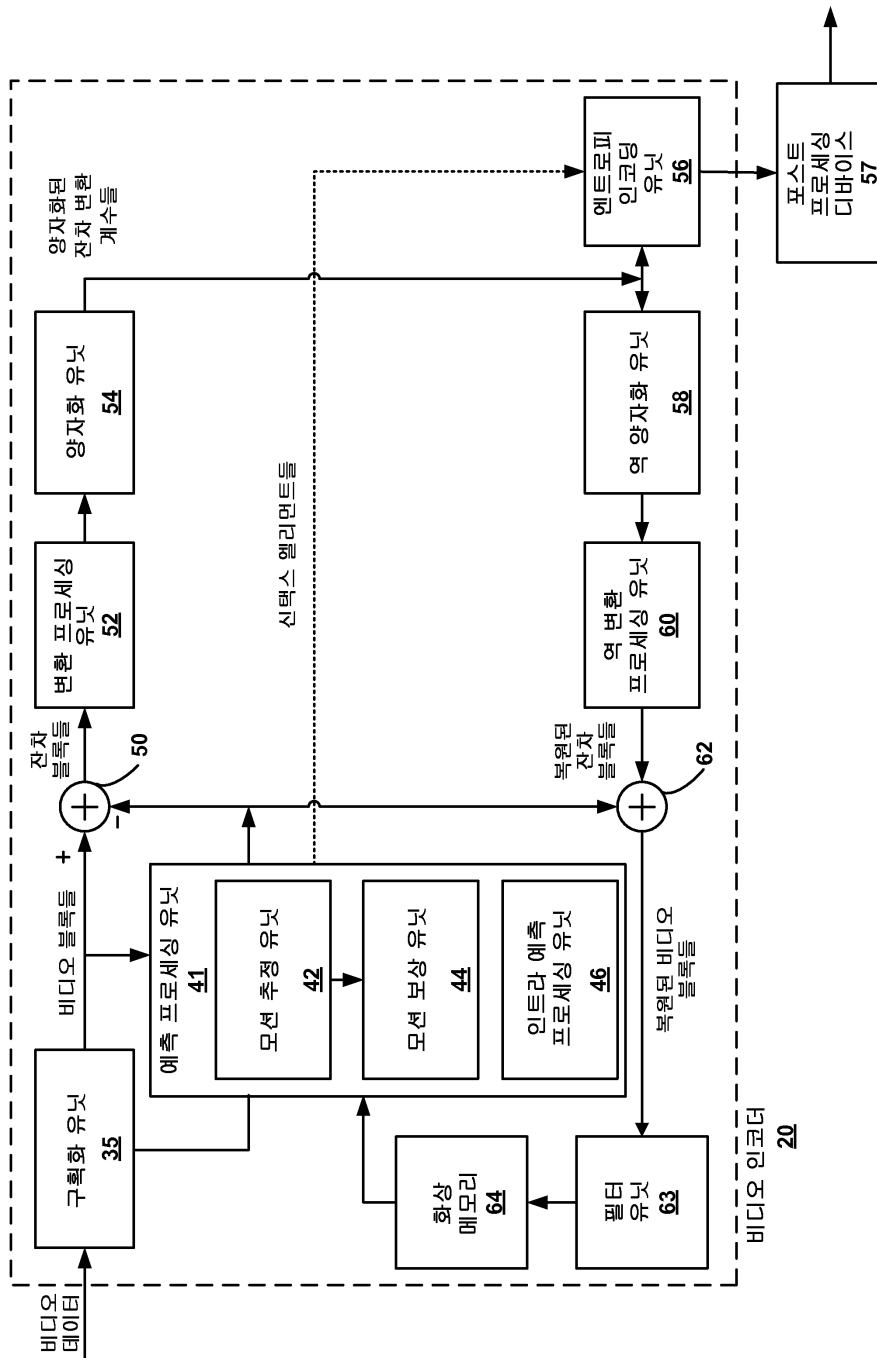
도면2



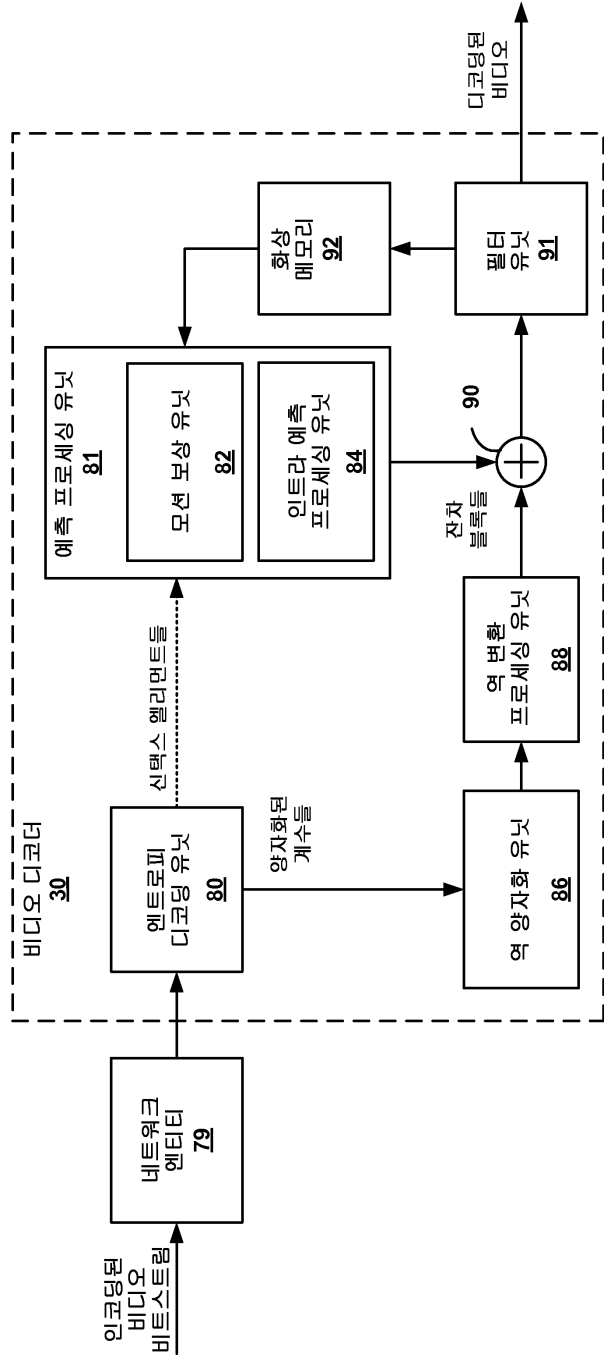
도면3



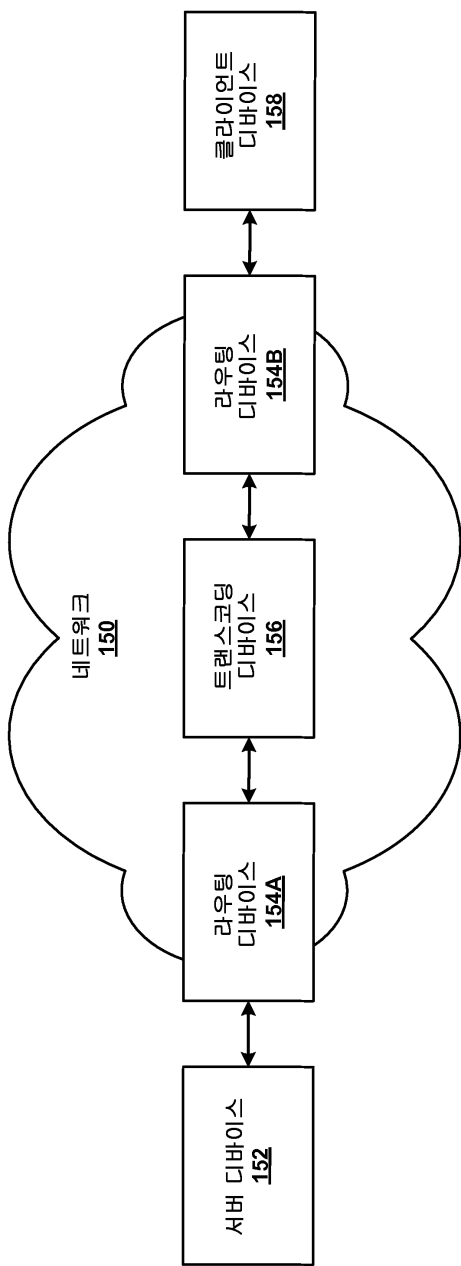
도면4



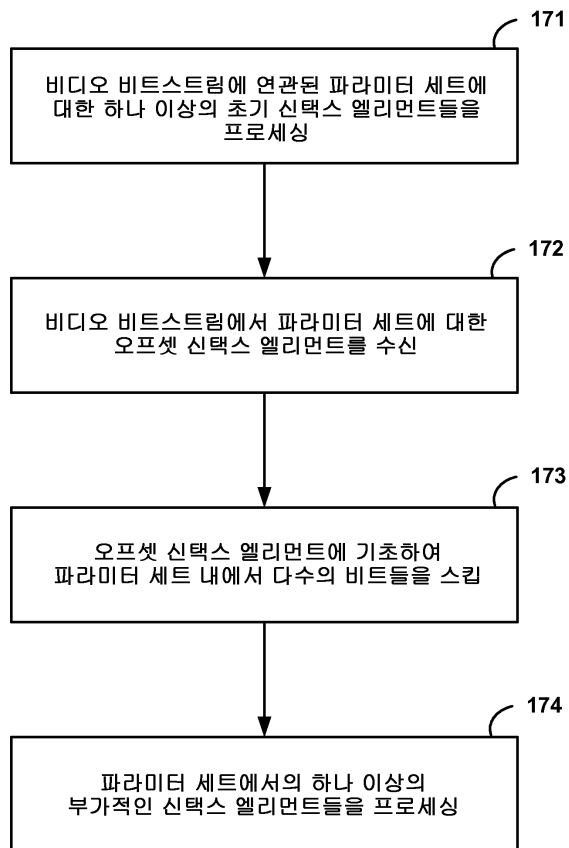
도면5



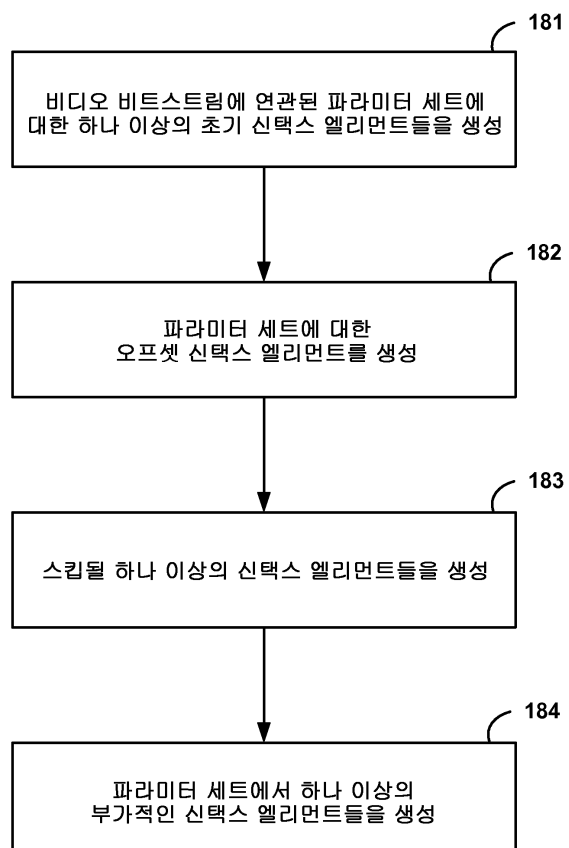
도면6



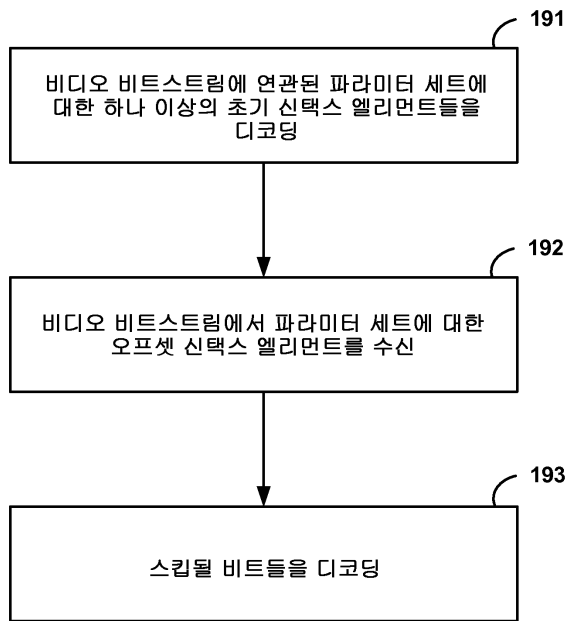
도면7



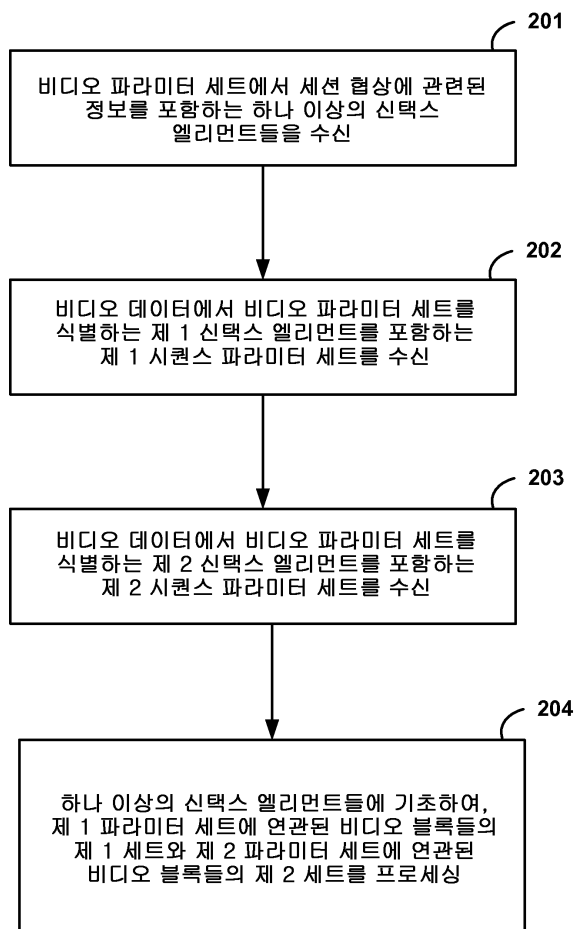
도면8



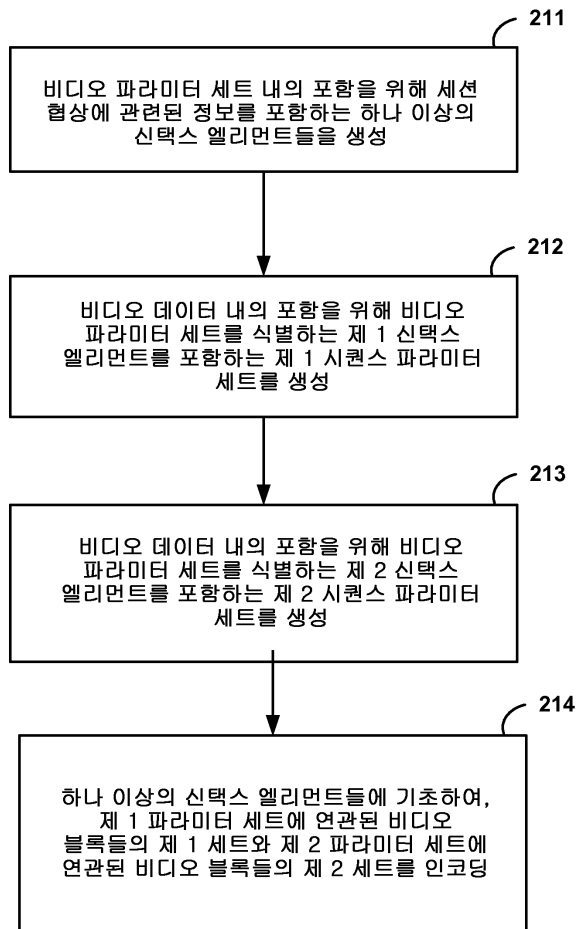
도면9



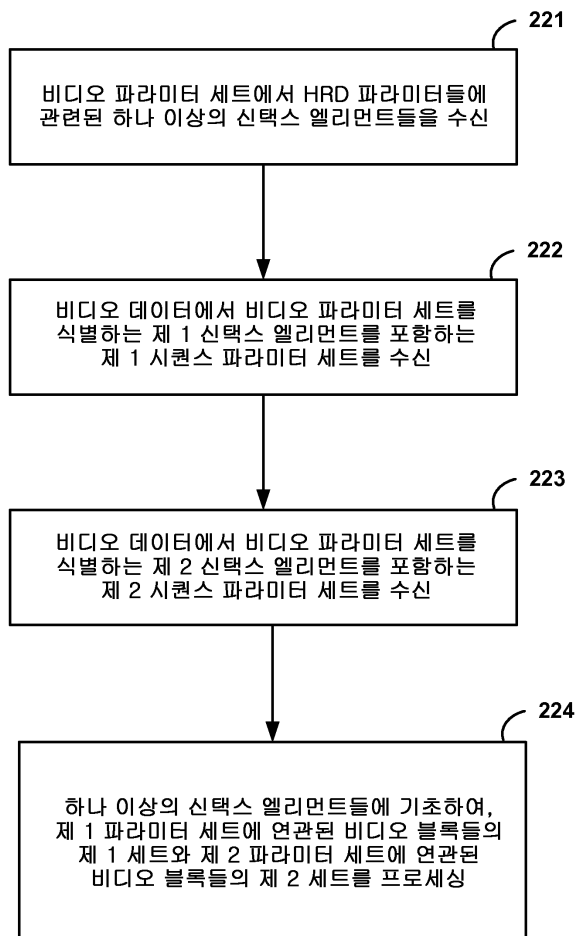
도면10



도면11



도면12



도면13

