



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2016년05월10일  
(11) 등록번호 10-1619688  
(24) 등록일자 2016년05월02일

(51) 국제특허분류(Int. Cl.)  
H04N 19/174 (2014.01) H04N 19/13 (2014.01)  
H04N 19/44 (2014.01) H04N 19/70 (2014.01)  
H04N 19/91 (2014.01)  
(21) 출원번호 10-2014-7031387  
(22) 출원일자(국제) 2012년12월21일  
심사청구일자 2015년03월03일  
(85) 번역문제출일자 2014년11월07일  
(65) 공개번호 10-2015-0003289  
(43) 공개일자 2015년01월08일  
(86) 국제출원번호 PCT/SE2012/051477  
(87) 국제공개번호 WO 2013/154482  
국제공개일자 2013년10월17일  
(30) 우선권주장  
61/623,290 2012년04월12일 미국(US)  
(56) 선행기술조사문헌  
Jill Boyce et al., 'Extensible High Layer Syntax for Scalability', JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 5th Meeting: Geneva, CH, 16-23 March, 2011, JCTVC-E279\*  
Rickard Sjoberg et al., 'AHG12: Slice header extension', JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 119th Meeting: Geneva, CH, 27 April - 7 May 2012, JCTVC-I0235  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
텔레호낙티에블라게트 엘엠 에릭슨(피유비엘)  
스웨덴 스톡홀름 83 에스이-164  
(72) 발명자  
사무엘슨, 조나탄  
스웨덴 에스-11250 스톡홀름 스태드샤그스플랜 1  
스조에베르그, 리카르트  
스웨덴 에스-113 36 스톡홀름 로버트 알름스트로  
엠스가탄 9에이  
(74) 대리인  
장수길, 백만기

전체 청구항 수 : 총 26 항

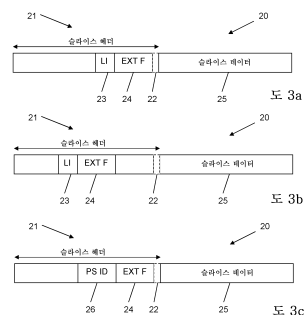
심사관 : 조우연

(54) 발명의 명칭 확장 데이터 처리

(57) 요약

슬라이스(3)의 인코딩된 표현(20)과 연관된 길이 표시자(23)가 파싱된다. 길이 표시자(23)는 인코딩된 표현(20)의 슬라이스 헤더(21)에 존재하는 확장 필드(24)의 길이를 나타낸다. 디코더(40)는, 인코딩된 표현(20)의 디코딩 동안에, 슬라이스 헤더(21) 내의 확장 필드(24)의 임의의 값을 무시하기로 결정할 수 있고, 여기서, 이들  
(뒷면에 계속)

대표도



값들은 길이 표시자(23)에 기초하여 식별된다. 결과적으로, 확장 필드(23)가 슬라이스 헤더(21)에 부가될 수 있고 여전히 레거시 디코더들(40)이 인코딩된 표현(20)을 정확하게 디코딩할 수 있게 한다.

---

## 명세서

### 청구범위

#### 청구항 1

화상(2)의 슬라이스(3)의 인코딩된 표현(20)을 디코딩하는 방법으로서 - 상기 인코딩된 표현(20)은 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함함 -,

상기 슬라이스 헤더(21)에 기초하여 얻어질 수 있는 파라미터 세트 식별자(26)에 기초하여 상기 인코딩된 표현(20)과 연관된 파라미터 세트를 식별하는 단계(S20);

상기 파라미터 세트에 존재하는 슬라이스 헤더 확장 존재 플래그를 파싱하는 단계(S21);

상기 슬라이스 헤더 확장 존재 플래그가 미리정의된 값을 갖는 경우, 상기 인코딩된 표현(20)과 연관되고 상기 슬라이스 헤더(21) 내의 확장 필드(24)의 비트 또는 바이트의 길이를 나타내는 가변 길이 코드의 길이 표시자(23)를 파싱하는 단계(S1); 및

상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 인코딩된 표현(20)의 디코딩 동안에, 상기 길이 표시자(23)에 기초하여 식별된 상기 확장 필드(24)의 임의의 값을 무시하기로 결정하는 단계(S2)

를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

상기 길이 표시자(23)에 기초하여 상기 확장 필드(24)의 길이를 결정하는 단계(S10); 및

상기 길이에 기초하여 상기 슬라이스 헤더(21) 내의 상기 확장 필드(24)의 끝을 식별하는 단계(S11)

를 더 포함하는 방법.

#### 청구항 3

제1항 또는 제2항에 있어서, 상기 길이 표시자(23)를 파싱하는 단계(S1)는, 상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 슬라이스 헤더(21)에 존재하는 상기 길이 표시자(23)를 파싱하는 단계(S1)를 포함하는 방법.

#### 청구항 4

제3항에 있어서, 상기 임의의 값을 무시하기로 결정하는 단계(S2)는, 상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 인코딩된 표현(20)의 디코딩 동안에, 상기 슬라이스 헤더(21) 내의 상기 길이 표시자(23)의 직후부터 상기 길이 표시자(23)에 기초하여 식별된 상기 확장 필드(24)의 끝까지의 상기 확장 필드(24)의 임의의 값을 무시하기로 결정하는 단계(S2)를 포함하는 방법.

#### 청구항 5

제1항 또는 제2항에 있어서, 상기 길이 표시자(23)를 파싱하는 단계(S1)는, 상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 슬라이스 헤더(21)에 기초하여 얻어질 수 있는 파라미터 세트 식별자(26)에 기초하여 식별된 파라미터 세트에 존재하는 상기 길이 표시자(23)를 파싱하는 단계(S1)를 포함하는 방법.

#### 청구항 6

제1항 또는 제2항에 있어서,

길이 표시자(23)에 기초하여 식별된 상기 확장 필드(24)의 끝에 기초하여 상기 슬라이스 데이터(25)의 선두를 식별하는 단계(S30); 및

상기 슬라이스 데이터(25)의 상기 선두로부터 시작하여 상기 슬라이스 데이터(25)를 파싱 및 디코딩하는 단계(S31)

를 더 포함하는 방법.

#### 청구항 7

제6항에 있어서, 상기 슬라이스 데이터(25)를 파싱 및 디코딩하는 단계(S31)는, 상기 슬라이스 헤더(21) 내의 상기 확장 필드(24)의 상기 임의의 값을 무시함으로써 상기 슬라이스 데이터(25)를 파싱 및 디코딩하는 단계(S31)를 포함하는 방법.

#### 청구항 8

화상(2)의 슬라이스(3)를 인코딩하는 방법으로서,

상기 슬라이스(3)의 슬라이스 헤더(21) 내의 확장 필드(24)의 존재를 나타내기 위해 슬라이스 헤더 확장 존재 플래그를 미리정의된 값으로 설정하는 단계(S60);

상기 슬라이스 헤더 확장 존재 플래그를 파라미터 세트 내에 삽입하는 단계(S70);

상기 슬라이스(3)의 상기 슬라이스 헤더(21)에 존재하는 상기 확장 필드(24)의 비트 또는 바이트의 길이를 나타내는 가변 길이 코드(23)의 길이 표시자(23)를 생성하는 단계(S40);

상기 슬라이스(3)를, 상기 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함하는 상기 슬라이스(3)의 인코딩된 표현(20)으로 인코딩하는 단계(S41);

상기 길이 표시자(23)를 상기 인코딩된 표현(20)과 연관시키는 단계(S42); 및

상기 파라미터 세트를 식별하는 파라미터 세트 식별자를 상기 인코딩된 표현(20)과 연관시키는 단계(S71)

를 포함하는 방법.

#### 청구항 9

제8항에 있어서,

상기 슬라이스 헤더(21)에 존재하는 상기 확장 필드(24)의 길이를 결정하는 단계(S51)를 더 포함하고, 상기 길이 표시자(23)를 생성하는 단계(S40)는 상기 확장 필드(24)의 상기 길이에 기초하여 상기 길이 표시자(23)를 생성하는 단계(S40)를 포함하는 방법.

#### 청구항 10

제8항 또는 제9항에 있어서, 상기 길이 표시자(23)를 연관시키는 단계(S42)는 상기 길이 표시자(23)를 상기 슬라이스 헤더(21) 내에 삽입하는 단계를 포함하는 방법.

#### 청구항 11

제10항에 있어서, 상기 길이 표시자(23)를 삽입하는 단계는 상기 슬라이스 헤더(21) 내의 상기 확장 필드(24) 바로 앞에 상기 길이 표시자(23)를 삽입하는 단계를 포함하는 방법.

#### 청구항 12

제8항 또는 제9항에 있어서, 상기 길이 표시자(23)를 연관시키는 단계(S42)는, 상기 길이 표시자(23)를 파라미터 세트 내에 삽입하고 상기 파라미터 세트를 식별하는 파라미터 세트 식별자(26)를 상기 인코딩된 표현(20)과 연관시키는 단계를 포함하는 방법.

#### 청구항 13

화상(2)의 슬라이스(3)의 인코딩된 표현(20)을 디코딩하기 위한 디코더(40, 50)로서 - 상기 인코딩된 표현(20)은 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함함 - ,

상기 디코더(40)는,

상기 슬라이스 헤더(21)에 기초하여 얻어질 수 있는 파라미터 세트 식별자(26)에 기초하여 상기 인코딩된 표현(20)과 연관된 파라미터 세트를 식별하도록 구성된 세트 식별 유닛(45);

상기 파라미터 세트에 존재하는 슬라이스 헤더 확장 존재 플래그를 파싱하고, 상기 슬라이스 헤더 확장 존재 플래그가 미리정의된 값을 갖는 경우, 상기 인코딩된 표현(20)과 연관되고 상기 슬라이스 헤더(21) 내의 확장 필드(24)의 비트 또는 바이트 단위의 길이를 나타내는 가변 길이 코드의 길이 표시자(23)를 파싱하도록 구성된 파싱 유닛(41); 및

상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 디코더(40)로 하여금 상기 인코딩된 표현(20)의 디코딩 동안에 상기 길이 표시자(23)에 기초하여 식별된 상기 확장 필드(24)의 임의의 값을 무시하게 하도록 구성된 처리 유닛(42)

을 포함하는 디코더.

#### 청구항 14

제13항에 있어서,

상기 길이 표시자(23)에 기초하여 상기 확장 필드(24)의 길이를 결정하도록 구성된 결정 유닛(43); 및

상기 길이에 기초하여 상기 슬라이스 헤더(21) 내의 상기 확장 필드(24)의 끝을 식별하도록 구성된 식별 유닛(44)

을 더 포함하는 디코더.

#### 청구항 15

제13항 또는 제14항에 있어서, 상기 파싱 유닛(41)은, 상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 슬라이스 헤더(21)에 존재하는 상기 길이 표시자(23)를 파싱하도록 구성된 디코더.

#### 청구항 16

제15항에 있어서, 상기 처리 유닛(42)은, 상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 디코더(40)로 하여금 상기 인코딩된 표현(20)의 디코딩 동안에 상기 슬라이스 헤더(21) 내의 상기 길이 표시자(23)의 직후부터 상기 길이 표시자(23)에 기초하여 식별된 상기 확장 필드(24)의 끝까지의 상기 확장 필드(24)의 임의의 값을 무시하게 하도록 구성된 디코더.

#### 청구항 17

제13항 또는 제14항에 있어서, 상기 파싱 유닛(41)은, 상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 슬라이스 헤더(21)에 기초하여 얻어질 수 있는 파라미터 세트 식별자(26)에 기초하여 식별된 파라미터 세트에 존재하는 상기 길이 표시자(23)를 파싱하도록 구성된 디코더.

#### 청구항 18

제13항 또는 제14항에 있어서,

길이 표시자(23)에 기초하여 식별된 상기 확장 필드(24)의 끝에 기초하여 상기 슬라이스 데이터(25)의 선두를 식별하도록 구성된 식별 유닛(44); 및

상기 슬라이스 데이터(25)의 상기 선두로부터 시작하여 상기 슬라이스 데이터(25)를 파싱 및 디코딩하도록 구성된 디코딩 유닛(46)

을 더 포함하는 디코더.

#### 청구항 19

제18항에 있어서, 상기 디코딩 유닛(46)은 상기 슬라이스 헤더(21) 내의 상기 확장 필드(24)의 상기 임의의 값을 무시함으로써 상기 슬라이스 데이터를 파싱 및 디코딩하도록 구성된 디코더.

#### 청구항 20

화상(2)의 슬라이스(3)를 인코딩하기 위한 인코더(70)로서,

상기 슬라이스(3)의 슬라이스 헤더(21) 내의 확장 필드(24)의 존재를 나타내기 위해 슬라이스 헤더 확장 존재 플래그를 미리정의된 값으로 설정하도록 구성된 설정 유닛(75);

상기 슬라이스(3)의 상기 슬라이스 헤더(21)에 존재하는 상기 확장 필드(24)의 비트 또는 바이트의 길이를 나타내는 가변 길이 코드의 길이 표시자(23)를 생성하도록 구성된 생성 유닛(71);

상기 슬라이스(3)를, 상기 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함하는 상기 슬라이스(3)의 인코딩된 표현(20)으로 인코딩하도록 구성된 인코딩 유닛(72); 및

상기 길이 표시자(23)를 상기 인코딩된 표현(20)과 연관시키고, 상기 슬라이스 헤더 확장 존재 플래그를 파라미터 세트 내에 삽입하며, 상기 파라미터 세트를 식별하는 파라미터 세트 식별자를 상기 인코딩된 표현(20)과 연관시키도록 구성된 연관 유닛(73)

을 포함하는 인코더.

#### 청구항 21

제20항에 있어서,

상기 슬라이스 헤더(21)에 존재하는 상기 확장 필드(24)의 길이를 결정하도록 구성된 결정 유닛(74)을 더 포함하고, 상기 생성 유닛(71)은 상기 확장 필드(24)의 상기 길이에 기초하여 상기 길이 표시자(23)를 생성하도록 구성되는 인코더.

#### 청구항 22

제20항 또는 제21항에 있어서, 상기 연관 유닛(73)은 상기 길이 표시자(23)를 상기 슬라이스 헤더(21) 내에 삽입하도록 구성되는 인코더.

#### 청구항 23

제22항에 있어서, 상기 연관 유닛(73)은 상기 슬라이스 헤더(21) 내의 상기 확장 필드(24) 바로 앞에 상기 길이 표시자(23)를 삽입하도록 구성되는 인코더.

#### 청구항 24

제20항 또는 제21항에 있어서, 상기 연관 유닛(73)은, 상기 길이 표시자(23)를 파라미터 세트 내에 삽입하고 상기 파라미터 세트를 식별하는 파라미터 세트 식별자(26)를 상기 인코딩된 표현(20)과 연관시키도록 구성되는 인코더.

#### 청구항 25

화상(2)의 슬라이스(3)의 인코딩된 표현(20)을 디코딩하기 위한 컴퓨터 프로그램(130) - 상기 인코딩된 표현(20)은 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함함 - 이 기록된 컴퓨터 판독가능 기록 매체로서,

상기 컴퓨터 프로그램(130)은 프로세서(110)에 의해 실행될 때, 상기 프로세서(110)로 하여금,

상기 슬라이스 헤더(21)에 기초하여 얻어질 수 있는 파라미터 세트 식별자(26)에 기초하여 상기 인코딩된 표현(20)과 연관된 파라미터 세트를 식별하고;

상기 파라미터 세트에 존재하는 슬라이스 헤더 확장 존재 플래그를 파싱하며;

상기 슬라이스 헤더 확장 존재 플래그가 미리정의된 값을 갖는 경우, 상기 인코딩된 표현(20)과 연관되고 상기 슬라이스 헤더(21) 내의 확장 필드(24)의 비트 또는 바이트의 길이를 나타내는 가변 길이 코드의 길이 표시자(23)를 파싱하고;

상기 슬라이스 헤더 확장 존재 플래그가 상기 미리정의된 값을 갖는 경우, 상기 인코딩된 표현(20)의 디코딩 동안에, 상기 길이 표시자(23)에 기초하여 식별된 상기 확장 필드(24)의 임의의 값을 무시하기로 결정하게 하는 코드 수단을 포함하는 컴퓨터 판독가능 기록 매체.

#### 청구항 26

화상(2)의 슬라이스(3)를 인코딩하기 위한 컴퓨터 프로그램(130)이 기록된 컴퓨터 판독가능 기록 매체로서,  
 상기 컴퓨터 프로그램(130)은 프로세서(110)에 의해 실행될 때, 상기 프로세서(110)로 하여금,  
 상기 슬라이스(3)의 슬라이스 헤더(21) 내의 확장 필드(24)의 존재를 나타내기 위해 슬라이스 헤더 확장 존재  
 플래그를 미리정의된 값으로 설정하고;  
 상기 슬라이스 헤더 확장 존재 플래그를 파라미터 세트 내에 삽입하며;  
 상기 슬라이스(3)의 상기 슬라이스 헤더(21)에 존재하는 상기 확장 필드(24)의 비트 또는 바이트의 길이를 나타  
 내는 가변 길이 코드의 길이 표시자(23)를 생성하고;  
 상기 슬라이스(3)를, 상기 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함하는 상기 슬라이스(3)의 인코딩된  
 표현(20)으로 인코딩하며;  
 상기 길이 표시자(23)를 상기 인코딩된 표현(20)과 연관시키고;  
 상기 파라미터 세트를 식별하는 파라미터 세트 식별자를 상기 인코딩된 표현(20)과 연관시키게 하는  
 코드 수단을 포함하는 컴퓨터 판독가능 기록 매체.

#### 청구항 27

삭제

#### 청구항 28

삭제

#### 청구항 29

삭제

#### 청구항 30

삭제

#### 청구항 31

삭제

#### 청구항 32

삭제

#### 청구항 33

삭제

#### 청구항 34

삭제

#### 청구항 35

삭제

#### 청구항 36

삭제

#### 청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

## 발명의 설명

### 기술 분야

[0001] 본 실시예는 대체로 슬라이스의 인코딩 및 디코딩에 관한 것으로, 특히 슬라이스의 인코딩 및 디코딩과 연계하여 확장 데이터를 처리하는 것에 관한 것이다.

### 배경 기술

[0002] 고효율 비디오 코딩(HEVC; High Efficiency Video Coding)은 이제 새로운 비디오 코딩 표준으로서, 현재 JCT-VC(Joint Collaborative Team - Video Coding)에서 개발중에 있다. JCT-VC는 동영상 전문가 그룹(MPEG; Moving Picture Experts Group)과 국제 통신 연합 통신 표준 섹터(ITU-T; International Telecommunication Union Telecommunication standardization sector)간의 협력 프로젝트이다. 현재, 다수의 새로운 톨을 포함하며 H.264 AVC(Advanced Video Coding)보다 상당히 더 효율적인 위원회 드래프트(CD; Committee Draft)가 정의되어 있다.

[0003] 미래에는, 멀티-뷰 확장 및 스케일가능한 확장 등의, HEVC의 확장이 정의될 가능성이 크다. 그러면 이들 확장은, 예를 들어, 베이스 뷰에 의존하는 추가 뷰 또는 베이스 층에 의존하는 추가 층에 관한 데이터를 포함할 수 있다.

[0004] 인코딩된 시퀀스, 예를 들어, 베이스 뷰 또는 베이스 층의 소정 부분들에 대한 하향호환성이 요건이라고 가정될 수 있다. 그러나, 레거시, 즉, 기본 사양 준수형 디코더는 이러한 확장을 이용할 때에도 비트스트림을 정확하게 처리할 수 있다는 것이 중요하다.

[0005] HEVC는, 특히, 확장에 대해 예약된 몇 가지 네트워크 추상층(NAL) 유닛 타입들을 정의한다. 또한, 파라미터 세트들, 예를 들어, 시퀀스 파라미터 세트(SPS; Sequence Parameter Set), 화상 파라미터 세트(PPS; Picture Parameter Set), 및 적응 파라미터 세트(APS; Adaptation Parameter Set)는 그들의 선택스 테이블의 끝에 확장 필드를 포함한다. HEVC의 기본 사양을 따르는 디코더는 파라미터 세트의 끝에 있는 확장 필드 내의 데이터를 무시한다는 것이 명시되어 있다. 확장 데이터를 위한 선택스가, 파라미터 세트로서의 SPS의 형태로 예시로서 아래에 도시되어 있다. 선택스는 SPS 이외의 다른 파라미터 세트를 이용하더라도 유사할 것이다. 파라미터 세트의 총 길이는, 통상 시스템층에 의해 제공되는 선택스 요소 NumBytesInRBSP를 통해 디코더가 알기 때문에, 디코더는 확장 필드가 끝나는 곳을 검출하는데 있어서 아무런 문제를 갖지 않을 것이다.

<code>seq_parameter_set_rbsp()</code>	<b>Descriptor</b>
...	
<code>sps_extension_flag</code>	<code>u(1)</code>

[0006]



```

if( sps_extension_flag )
    while( more_rbsp_data( ) )
        sps_extension_data_flag          u(1)
        rbsp_trailing_bits( )
}

```

[0007]

[0008] 화상의 인코딩된 슬라이스를 포함하는 NAL 유닛은 2개 부분: 첫 번째 슬라이스 헤더와 그 다음 슬라이스 데이터로 구성된다. 인코딩된 슬라이스들에 대한 확장 필드는 현재 없다.

[0009] 슬라이스 헤더는 현재의 슬라이스에 대한 정보, 예를 들어, 슬라이스 타입, 어느 기준 화상이 이용될 것인지 등을 포함한다. HEVC에서 확장과 확장 필드들을 추가할 필요성이 있다. 그러나, 이러한 경우, 기본 사양을 따르는, 즉, 확장을 이용하지 않는 레거시 디코더와, 확장-준수형 디코더 양쪽 모두가 이러한 확장을 포함하는 비트스트림을 정확하게 처리할 수 있어야 한다는 것이 중요하다.

[0010] 슬라이스의 인코딩된 표현에 대한 슬라이스 헤더의 현재의 레이아웃은, 레거시 및 확장-준수형 디코더들 모두를 가질 때 이러한 필요성을 구현하기에 어렵게 하거나 심지어 불가능하게 한다.

### 발명의 내용

#### 해결하려는 과제

[0011] 슬라이스 헤더 확장의 이용을 가능케하는 것이 전반적 목적이다.

[0012] 슬라이스의 인코딩 및 디코딩과 연계하여 슬라이스 헤더 확장을 처리하는 것이 특정한 목적이다.

[0013] 이들 및 기타의 목적은 여기서 개시된 실시예에 의해 충족된다.

#### 과제의 해결 수단

[0014] 실시예의 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하는 방법에 관한 것이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 이 방법은, 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱(parse)하는 단계를 포함한다. 이 방법은 또한, 인코딩된 표현의 디코딩 동안에, 길이 표시자에 기초하여 식별된 확장 필드의 임의의 값을 무시하기로 결정하는 단계를 포함한다.

[0015] 실시예의 관련된 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 디코더를 정의한다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 디코더는 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하도록 구성된 파싱 유닛을 포함한다. 디코더는 또한, 인코딩된 표현의 디코딩 동안에, 길이 표시자에 기초하여 식별된 확장 필드의 임의의 값을 디코더가 무시하게 하도록 구성된 처리 유닛을 포함한다.

[0016] 실시예의 또 다른 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하는 방법에 관한 것이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 이 방법은 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하는 단계를 포함한다. 이 방법은 또한, 길이 식별자에 기초하여 슬라이스 헤더 내의 확장 필드를 식별하는 단계를 포함한다. 슬라이스의 인코딩된 표현은 길이 식별자에 기초하여 식별된 확장 필드의 적어도 하나의 값에 기초하여 디코딩된다.

[0017] 실시예의 또 다른 관련된 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 디코더를 정의한다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 디코더는 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하도록 구성된 파싱 유닛을 포함한다. 디코더는 또한, 길이 식별자에 기초하여 슬라이스 헤더 내의 확장 필드를 식별하도록 구성된 확장 필드 식별 유닛을 포함한다. 디코더의 디코딩 유닛은 확장 필드 식별 유닛에 의해 식별된 확장 필드의 적어도 하나의 값에 기초하여 슬라이스의 인코딩된 표현을 디코딩하도록 구성된다.

[0018] 실시예의 추가 양태는 화상의 슬라이스를 인코딩하는 방법에 관한 것이다. 이 방법은 슬라이스의 슬라이스 헤

데에 존재하는 확장 필드의 길이를 나타내는 길이 표시자를 생성하는 단계를 포함한다. 슬라이스는 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스의 인코딩된 표현으로 인코딩된다. 길이 표시자는 인코딩된 표현과 연관된다.

- [0019] 실시예의 관련된 추가 양태는 화상의 슬라이스를 인코딩하기 위한 인코더를 정의한다. 인코더는 슬라이스의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타내는 길이 표시자를 생성하도록 구성된 생성 유닛을 포함한다. 인코딩 유닛은 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스의 인코딩된 표현으로 슬라이스를 인코딩하도록 구성된다. 인코더는 또한, 길이 표시자를 인코딩된 표현과 연관시키도록 구성된 연관 유닛을 포함한다.
- [0020] 실시예의 추가 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 디코더를 포함하는 수신기에 관한 것이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 디코더는 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하도록 구성된 파싱 유닛을 포함한다. 디코더는 또한, 인코딩된 표현의 디코딩 동안에, 길이 표시자에 기초하여 식별된 확장 필드의 임의의 값을 디코더가 무시하게 하도록 구성된 처리 유닛을 포함한다.
- [0021] 실시예의 또 다른 추가 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 디코더를 포함하는 수신기에 관한 것이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 디코더는 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하도록 구성된 파싱 유닛을 포함한다. 디코더는 또한, 길이 식별자에 기초하여 슬라이스 헤더 내의 확장 필드를 식별하도록 구성된 확장 필드 식별 유닛을 포함한다. 디코더의 디코딩 유닛은 확장 필드 식별 유닛에 의해 식별된 확장 필드의 적어도 한 값에 기초하여 슬라이스의 인코딩된 표현을 디코딩하도록 구성된다.
- [0022] 실시예의 역시 또 다른 양태는 화상의 슬라이스를 인코딩하기 위한 인코더를 포함하는 전송기에 관한 것이다. 인코더는 슬라이스의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타내는 길이 표시자를 생성하도록 구성된 생성 유닛을 포함한다. 인코더는 또한 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스의 인코딩된 표현으로 슬라이스를 인코딩하도록 구성된 인코딩 유닛을 포함한다. 인코더의 연관 유닛은 길이 표시자를 인코딩된 표현과 연관시키도록 구성된다.
- [0023] 실시예의 추가의 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 컴퓨터 프로그램에 관한 것이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 컴퓨터 프로그램은, 프로세서에 의해 실행될 때, 프로세서로 하여금, 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하게 하는 코드 수단을 포함한다. 프로세서는 또한, 인코딩된 표현의 디코딩 동안에, 길이 표시자에 기초하여 식별된 상기 확장 필드의 임의의 값을 무시하기로 결정하게 된다.
- [0024] 실시예의 추가의 관련된 양태는 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 컴퓨터 프로그램을 정의한다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 컴퓨터 프로그램은, 프로세서에 의해 실행될 때, 프로세서로 하여금, 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하게 하는 코드 수단을 포함한다. 프로세서는 또한, 길이 표시자에 기초하여, 상기 슬라이스 헤더 내의 확장 필드를 식별하고, 길이 표시자에 기초하여 식별된 확장 필드의 적어도 하나의 값에 기초하여 슬라이스의 인코딩된 표현을 디코딩하게 된다.
- [0025] 실시예의 역시 추가의 관련된 양태는 화상의 슬라이스를 인코딩하기 위한 컴퓨터 프로그램을 정의한다. 컴퓨터 프로그램은, 프로세서에 의해 실행될 때, 프로세서로 하여금, 슬라이스의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타내는 길이 표시자를 생성하게 하는 코드 수단을 포함한다. 프로세서는 또한, 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스의 인코딩된 표현으로 슬라이스를 인코딩하게 된다. 프로세서는 또한, 길이 표시자를 인코딩된 표현과 연관시키게 된다.
- [0026] 실시예의 또 다른 추가적 양태는, 컴퓨터 판독가능한 매체와 컴퓨터 판독가능한 매체에 저장된 컴퓨터 프로그램을 포함하는 컴퓨터 프로그램 제품에 관한 것이다.
- [0027] 본 실시예는, 슬라이스의 인코딩된 표현의 슬라이스 헤더 내로의 확장 필드의 도입을 가능케하고, 확장 필드와 호환되지 않는 레거시 디코더들에 의해서도 정확히 처리될 수 있는 인코딩된 표현들을 여전히 생성한다.

### 도면의 간단한 설명

- [0028] 본 발명은, 그 추가 목적 및 이점들과 함께, 첨부된 도면과 함께 취해지는 이하의 상세한 설명을 참조하여 최상으로 이해될 수 있다.

- 도 1은 하나 이상의 슬라이스를 포함하는 화상들의 비디오 스트림의 개략적 예시이다;
- 도 2는 NAL 유닛을 포함하는 데이터 패킷의 예시이다;
- 도 3a 내지 도 3c는 슬라이스의 인코딩된 표현의 예시이다;
- 도 4는 실시예에 따른 슬라이스의 인코딩된 표현을 디코딩하는 방법의 흐름도이다;
- 도 5는 실시예에 따른 도 4의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다;
- 도 6은 또 다른 실시예에 따른 도 4의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다;
- 도 7은 추가의 실시예에 따른 도 4의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다;
- 도 8는 또 다른 실시예에 따른 슬라이스의 인코딩된 표현을 디코딩하는 방법의 흐름도이다;
- 도 9는 실시예에 따른 슬라이스를 인코딩하는 방법의 흐름도이다;
- 도 10은 실시예에 따른 도 9의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다;
- 도 11은 또 다른 실시예에 따른 도 9의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다;
- 도 12는 도 11의 연관 단계의 실시예를 나타내는 흐름도이다;
- 도 13은 실시예에 따른 디코더의 개략적 블록도이다;
- 도 14는 또 다른 실시예에 따른 디코더의 개략적 블록도이다;
- 도 15는 추가의 실시예에 따른 디코더의 개략적 블록도이다;
- 도 16은 실시예에 따른 수신기의 개략적 블록도이다;
- 도 17은 실시예에 따른 인코더의 개략적 블록도이다;
- 도 18은 또 다른 실시예에 따른 인코더의 개략적 블록도이다;
- 도 19는 실시예에 따른 전송기의 개략적 블록도이다;
- 도 20은 실시예에 따른 컴퓨터 판독가능한 매체 및 컴퓨터 프로그램을 포함하는 컴퓨터의 개략적 블록도이다.

### 발명을 실시하기 위한 구체적인 내용

- [0029] 도면들을 통틀어, 동일한 참조 번호는 유사하거나 대응하는 요소들에 대해 이용된다.
- [0030] 본 실시예들은 화상들의 슬라이스들의 인코딩 및 디코딩과 연계한 확장 데이터 처리에 관한 것이다. 따라서, 실시예들은, 슬라이스의 슬라이스 헤더에서, 확장 선택스 요소라고 표기된, 소위 확장 필드의 도입과 이용을 가능케한다. 실시예들에 따르면, 이러한 확장 필드는 슬라이스 헤더에 추가될 수 있고, 여전히, 확장-준수형 디코더와 확장 데이터를 처리하지 못하는, 즉, 확장 필드를 이용하지 않는 프로파일에 따르는 디코더들 모두에 의해 정확하게 처리될 수 있는 인코딩된 표현을 제시한다. 이들 디코더들은 일반적으로 레거시 디코더 또는 기본 사양 준수형 디코더로 표기된다.
- [0031] 확장-준수형 및 레거시 디코더들 모두의 존재는 슬라이스 헤더 내의 확장 필드의 이용과 연계하여 문제점을 제기한다. 따라서, 레거시 디코더는, 인코딩된 표현이 그 슬라이스 헤더에서 확장 데이터를 운반하더라도 슬라이스의 인코딩된 표현의 데이터를 정확하게 디코딩하고 해석할 수 있을 필요가 있다.
- [0032] 배경부에서 논의된 파라미터 세트들에서와 동일한 종류의 확장 필드를 추가하는 것은 가능하지 않으며, 여기서, 선택스 구조의 끝에 있는 모든 추가 데이터는 디코더에 의해 무시된다. 그 이유는, 슬라이스 헤더가 어디에서 끝나고 슬라이스 데이터가 어디에서 시작하는지를 식별하기 위한 방법을 디코더가 갖고 있지 않다는데서 시작한다. 따라서, 개별 데이터 구조들, 즉, 슬라이스 헤더와 슬라이스 데이터의 길이는 알려지지 않고, 선택스 요소 NumBytesInRBSP에 의해 총 길이만 알려진다. 따라서, 슬라이스 층 원시 바이트 시퀀스 페이로드(RBSP; raw byte sequence payload) 선택스는 일반적으로 다음과 같이 보인다:

```

slice_layer_rbsp( ) {
    slice_header( )
    slice_data( )
    rbsp_slice_trailing_bits( )
}

```

[0033]

[0034]

따라서, 슬라이스 헤더가 어디에서 끝나고 슬라이스 데이터가 어디에서 시작하는지를 레거시 디코더가 식별할 방법이 없기 때문에 슬라이스 헤더의 끝에서의 모든 추가 데이터가 레거시 디코더에 의해 무시되는 단순 확장 필드를 추가하는 것이 가능하지 않다.

[0035]

길이 표시자를 슬라이스의 인코딩된 표현과 연관시키는 것이 제안되며, 여기서, 이 길이 표시자는 슬라이스의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타낸다. 따라서, 길이 표시자는, 확장 필드를 구성하는 슬라이스 헤더와 슬라이스의 인코딩된 표현의 일부를 디코더가 식별할 수 있게 한다. 따라서, 레거시 디코더는, 디코딩 동안에 인코딩된 표현의 어떤 부분이, 레거시 디코더가 처리할 수 없는 확장 데이터를 포함한다는 이유로 무시될 것인지를 결정할 수 있다.

[0036]

그러나, 레거시 디코더가 디코딩 동안에 확장 데이터를 무시해야 하는 반면 확장-준수형 디코더는 디코딩 동안에 이 확장 데이터를 디코딩하고 이용할 수 있다.

[0037]

도 1은 화상 2의 비디오 시퀀스 또는 스트림 1의 개략적 예시이다. 본 분야에서 공지된 바와 같이, 각각의 화상 2는 하나 이상의 소위 슬라이스(3)를 포함할 수 있다. 따라서, 비디오 스트림 1의 각각의 화상(2)은 하나 이상의 슬라이스(3)로 분할되고, 각각의 슬라이스(3)는 화상 2의 독립적으로 디코딩가능한 조각이다. 즉, 하나의 슬라이스(3)가 소실된다면, 그 화상(2)의 다른 슬라이스(3)는 여전히 디코딩가능하다.

[0038]

슬라이스(3)의 인코딩은 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스(3)의 인코딩된 표현을 생성한다. 인코딩된 표현은 도 2에 도시된 바와 같이 소위 네트워크 적응층(NAL; Network Adaptation Layer) 유닛(11)로서 인코딩 프로세스로부터 출력된다. NAL 유닛(11)의 제1 부분은 NAL 유닛 내의 데이터의 타입의 표시를 포함하는 헤더이다. NAL 유닛(11)의 나머지 부분은 슬라이스 헤더와 슬라이스 데이터의 형태로 페이로드 데이터를 포함한다.

[0039]

그 다음 NAL 유닛(11)에는 헤더(12)가 부가되어 인코더로부터 디코더로의 비트스트림의 일부로서 전송될 수 있는 데이터 패킷(10)을 형성할 수 있다. 예를 들어, 실시간 전송 프로토콜(RTP; Real-time Transport Protocol), 사용자 데이터그램 프로토콜(UDP; User Datagram Protocol), 및 인터넷 프로토콜(IP; Internet Protocol) 헤더(12)가 NAL 유닛(11)에 부가될 수 있다. NAL 유닛(11)의 이러한 형태의 패킷화는 비디오 트랜지스포트와 연계한 예를 구성할 뿐이다. 파일 포맷, MPEG2-트랜스포트 스트림, MPEG-2 프로그램 스트림 등의 NAL 유닛(11)을 처리하는 다른 접근법들도 가능하다.

[0040]

여기서 개시된 실시예들은 비디오 스트림 또는 시퀀스(1)의 화상(2)의 슬라이스(3)의 디코딩 및 인코딩에 특히 적용가능하며 이에 관한 것으로서, 비디오 스트림 또는 시퀀스(1)는 도 1에 도시된 바와 같이 복수의 화상(2)을 포함한다. 따라서 실시예들은 주로 그러한 비디오 디코딩 및 코딩과 연계하여 논의될 것이다.

[0041]

그러나, 실시예들은 또한, 단일 화상 내의 슬라이스들의 디코딩 및 인코딩에도, 예를 들어, 정지 화상 또는 영상의 디코딩 및 인코딩 등과 연계하여 적용될 수 있다. 예를 들어, 확장 데이터는 공간적으로 스케일가능한 정지 화상과 연계하여 이용될 수 있다.

[0042]

도 4는, 선택사항이지만 바람직하게는 비디오 스트림 내의, 화상의 슬라이스의 인코딩된 표현을 디코딩하는 방법을 나타내는 흐름도이다. 도 4의 방법은 바람직하게는 화상의 슬라이스의 인코딩된 표현의 디코딩 동안에 디코더에 의해 수행되는 방법 또는 화상의 슬라이스의 인코딩된 표현에 대한 디코딩 방법이다.

[0043]

도 3a 내지 도 3c는 슬라이스의 이러한 인코딩된 표현(20)의 비제한적 예이다. 인코딩된 표현(20)은 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함한다.

[0044]

도 4의 방법은 인코딩된 표현(20)과 또는 인코딩된 표현(20)에 연관된 길이 표시자(LI; length indicator)(23)가 파싱되는 단계 S1을 포함한다. 이 길이 표시자(23)는 슬라이스 헤더(21) 내의 확장 필드(EXT F;

extension field)(24)의 길이를 나타낸다. 후속하는 단계 S2에서, 인코딩된 표현(20)의 디코딩 동안에, 길이 표시자(23)에 기초하여 식별된 확장 필드(24)의 임의의 값을 무시하기로 결정된다.

[0045] 이것은, 확장 필드(24)에 의해 점유된 인코딩된 표현(20)의 부분을 식별하기 위해 인코딩된 표현(20)과 연관된 길이 표시자(23)가 이용될 수 있다는 것을 의미한다. 따라서, 디코더는 길이 표시자(23)를 이용하여 확장 필드(24)의 값(들) 또는 데이터를 무시하고 이용하지 않기로 판정 또는 결정할 수 있고, 여기서 이 확장 필드(24)는 길이 표시자(23)에 기초하여 슬라이스 헤더(21)에서 식별된다. 따라서, 디코더는 이렇게 길이 표시자(23)를 이용해 슬라이스 헤더(21) 내의 데이터의 소정량 또는 일부를 건너뛰고, 여기서, 건너뛴 데이터의 양 또는 부분은 길이 표시자(23)에 의해 표시된다.

[0046] 특정 실시예에서, 비디오 스트림 내 등의 화상의 슬라이스의 인코딩된 표현(20)을 디코딩하는 방법은, 인코딩된 표현(20)과 연관되고 슬라이스 헤더(21) 내의 확장 필드(24)의 길이를 나타내는 길이 표시자(23)를 파싱하는 단계를 포함한다. 이 방법은 또한, 이 특정 실시예에서, 인코딩된 표현(20)의 디코딩 동안에, 길이 표시자(23)에 기초하여 식별된 확장 필드(24)의 임의의 값을 무시하는 단계를 더 포함한다.

[0047] 인코딩된 표현(20)과 연관된 길이 표시자(23)는 또한 레거시 디코더가 인코딩된 표현(20)을 처리 및 디코딩할 수 있게 한다. 이들 레거시 디코더들은 단순히, 확장 필드(24)에 의해 점유된 슬라이스 헤더(21)의 부분으로부터 얻어진 값(들) 또는 데이터를 무시하기로 결정하고, 여기서, 슬라이스 헤더(21)의 이 부분은 파싱된 길이 표시자(23)에 기초하여 식별된다.

[0048] 본 명세서에서 연관시킨다는 것이란, 인코딩된 표현(20)에 기초하여 관련 길이 표시자(23)를 식별하고 회수하는 것이 가능하다는 것을 암시한다. 여기서 더 개시되는 바와 같이, 이것은, 예를 들어, 인코딩된 표현(20)에, 전형적으로는 슬라이스 헤더(21)에 길이 표시자(23)를 포함하거나(도 3a 및 3b 참조), 인코딩된 표현(20)에 식별자(26) 등의 데이터를 포함시킴으로써(도 3c 참조) -여기서, 이 데이터 또는 식별자(26)는 관련 길이 표시자(23)를 식별하는데 이용될 수 있음- 달성될 수 있다. 후자의 경우, 길이 표시자(23)는 인코딩된 표현(20)이 아닌 다른 데이터 구조로 제공될 수 있다. 따라서 데이터 또는 식별자(26)는 이 다른 데이터 구조를 가리키고 그 식별을 가능케 한다.

[0049] 도 5는 특정 실시예에 따른 방법의 추가 단계들을 나타내는 흐름도이다. 이들 추가적이지만 선택사항적인 단계들은 도 4의 단계 S1으로부터 계속되는 단계 S10에서 시작한다. 단계 S10에서, 확장 필드의 길이는 단계 S1에서 파싱된 길이 표시자(23)에 기초하여 결정된다. 다음 단계 S11에서, 슬라이스 헤더(21) 내의 확장 필드(24)의 끝이 단계 S10에서 결정된 길이에 기초하여 식별된다. 후속 단계 S2에서, 디코더는 그에 따라 단계 S11에서 식별된 확장 필드(24)의 끝까지의 확장 필드(24)의 값 또는 값들을 무시하기로 결정 또는 판정할 수 있다.

[0050] 선택사항으로서, 슬라이스 헤더(21) 내의 확장 필드(24)의 선두도 역시 식별된다. 실시예에서, 선두는, 슬라이스 헤더(21) 내의 정의된 비트 또는 바이트 위치 등의, 슬라이스 헤더(21) 내의 고정되거나 미리정의된 위치에 있을 수 있다. 대안으로서, 확장 필드(24)의 선두는, 슬라이스 헤더(21) 내의 길이 표시자(23) 직후에 후속되는 등의, 길이 표시자(23)에 기초하여 결정될 수 있다(도 3a 및 도 3b 참조). 일반적으로, 슬라이스 헤더(21) 내의 선택스 요소의 위치는 선행하는 선택스 요소의 길이에 의해 주어진다. 이것은 특히, 가변 길이 코딩이 슬라이스 헤더(21) 내의 하나 이상의 선행 선택스 요소에 적용되는 경우에 참이다. 이러한 경우, 디코더는 통상적으로 첫 비트 위치로부터 슬라이스 헤더(21)의 파싱과 디코딩을 시작하고 확장 필드(24)의 선두에 도달할 때까지 다양한 선택스 요소들을 통과한다.

[0051] 실시예에서, 추가 슬라이스 데이터, 즉, 확장 필드(24)의 길이를 나타내는 길이 표시자(23)가 도입된다. 길이 표시자(23)는, 예를 들어, 슬라이스 헤더(21) 또는 또 다른 데이터 구조 내의 선택스 요소의 예시로서, 슬라이스 헤더(21)와 연관되거나, 그 길이가 비트 패턴에 의해 유도될 수 있다. 따라서, 인코더는 추가 데이터의 길이를 시그널링하기 위해 정의된 선택스 요소(23)를 이용하도록 구성된다. 디코더는 추가 데이터의 길이를 결정하도록 구성된다. 추가 데이터는 소정 프로파일을 따르는 디코더에 의해 무시되는 반면 다른 프로파일을 따르는 디코더는 추가 데이터 내의 선택스 요소들의 값들을 파싱하고 이용할 것이다. 추가 데이터를 무시하는 프로파일에 따르는 디코더는, NAL 유닛이 디코딩될 수 있도록 슬라이스 헤더(21)의 끝을 결정하기 위해 길이 정보를 이용할 것이다.

[0052] 따라서 실시예들은 간단하고 비트레이트 효율적인 방식으로 슬라이스 헤더(21)의 확장을 가능케 한다.

[0053] 도 4의 방법 및 도 5에 도시된 실시예에 적용가능한 특정 실시예에서, 길이 표시자(23)는, 도 3a 및 도 3b에 도시된 바와 같이, 슬라이스 헤더(21)에 존재하는 코드워드 또는 선택스 요소이다. 이것은, 단계 S1이 슬라이스



헤더(21)에 존재하는 길이 표시자(23)를 파싱하는 단계를 포함한다는 것을 의미한다.

[0054] 특정 실시예에서, 길이 표시자(23)는 슬라이스 헤더(21)에서 확장 필드(24) 바로 앞에 존재한다. 따라서, 확장 필드(24)는 이 특정 실시예에서는 슬라이스 헤더(21) 내의 길이 표시자(23) 직후에 온다. 이러한 경우, 도 4의 단계 S2는, 실시예에서, 인코딩된 표현(20)의 디코딩 동안에, 슬라이스 헤더(21) 내의 길이 표시자(23) 직후부터 확장 필드(24)의 끝까지 확장 필드(24)의 임의의 값 또는 데이터를 무시하기로 결정하는 단계를 포함할 수 있다. 이 확장 필드(24)의 끝은, 도 5의 단계 S11과 연계하여 상기에서 개시된 바와 같이, 길이 표시자(23)에 기초하여 식별된다.

[0055] 또 다른 특정 실시예에서, 길이 표시자(23)는 슬라이스 헤더(21)에 반드시 존재할 필요는 없지만, 인코딩된 표현(20)과 기타의 방식으로 연관된다. 예를 들어, 슬라이스 헤더(21)는, 통상적으로, 인코딩된 표현(20)과 연관되고 현재의 슬라이스에 적용가능한 데이터를 포함하는 파라미터 세트의 식별자(26)를 포함한다. 도 3c는 슬라이스 헤더(21)에 파라미터 세트(PS) 식별자(ID)(26)를 갖는 이러한 경우를 나타낸다.

[0056] 예를 들어, 적응 파라미터 세트(APS; adaptation parameter set)는 하나보다 많은 슬라이스에 대해 유효한 제어 정보를 포함한다. 제어 정보는 슬라이스들간에 상이할 수 있다. 화상 파라미터 세트(PPS; picture parameter set)는 수개의 화상들에 대해 유효한 제어 정보를 포함하며, 동일한 비디오 시퀀스 또는 스트림의 복수의 화상들에 대해 동일할 수 있다. 시퀀스 파라미터 세트(SPS; sequence parameter set)는 전체의 비디오 시퀀스 또는 스트림에 대해 유효한 제어 정보를 포함한다. HEVC는 또한, 소위 비디오 파라미터 세트(VPS; Video Parameter Set)를 이용한다. 이러한 경우에, 길이 표시자(23)는, 현재의 슬라이스에 적용가능한 APS, VPS, PPS 또는 SPS 내 등의, 이들 파라미터 세트들 중 임의의 것에 존재할 수 있다. 이러한 경우에, 슬라이스 헤더(21)는 바람직하게는, 길이 표시자(23)를 운반하는 관련 파라미터 세트의 식별을 가능케하는 파라미터 세트 식별자(26)를 포함한다. 그러면 디코더는, 정확한 파라미터 세트를 식별한 다음 식별된 파라미터 세트로부터 길이 표시자(23)를 파싱하기 위하여 이 파라미터 세트 식별자(26)를 이용할 수 있다. 파라미터 세트 식별자(26)는 APS 식별자 또는 PPS 식별자 등의 파라미터 세트를 바로 식별할 수 있다. 대안적 접근법에서 파라미터 세트 식별자(26)는, 길이 표시자(23)를 운반하는 또 다른 파라미터 세트에 대한 또 다른 파라미터 세트 식별자를 포함하는 파라미터 세트를 식별한다. 예를 들어, 슬라이스 헤더(21)는 현재의 슬라이스에 적용가능한 PPS에 대한 PPS 식별자를 포함할 수 있다. 이 PPS는, 현재의 슬라이스와 PPS에 적용가능한 SPS에 대한 SPS 식별자를 포함할 수 있고, 이 SPS는 길이 표시자(23)를 포함한다.

[0057] 바이트 또는 비트수의 관점에서 확장 필드(24)의 크기 또는 길이가 비디오 스트림 내의 수 개의 슬라이스들에 대해 유사할 가능성이 높다면, 길이 표시자(23)를 각각의 인코딩된 표현(20)에서 직접 시그널링하는 것 대신에 그것을 파라미터 세트에서 추가하는 것이 바람직할 수 있다. 이러한 경우, 슬라이스 헤더(21)에 길이 표시자(23)의 비트들을 포함해야 할 필요가 없어서 인코딩된 표현(20)의 총 크기 또는 길이는 다소 감소될 수 있다. 이 접근법에서, 복수의 슬라이스들은 그에 따라, 길이 표시자(23)를 포함하는 동일한 파라미터 세트를 참조함으로써 동일한 길이 표시자(23)를 공유할 것이다.

[0058] 그러나, 일반적으로는 각각의 인코딩된 표현(20) 내의 슬라이스 헤더(21)에서 길이 표시자(23)를 직접 포함하는 데 있어서 훨씬 더 큰 융통성이 있다. 결과적으로, 그에 따라 확장 필드(24)의 크기 또는 길이는 상이한 슬라이스들간에 상이한 것이 허용될 수 있다. 또한, 길이 표시자(23)를 파싱하기 위하여 파라미터 세트에 대한 어떠한 참조도 요구되지 않는다.

[0059] 도 4의 단계 S1에 도시된 바와 같은 길이 표시자를 파싱하는 것은 각각의 인코딩된 표현(20)에 대해 반드시 수행되어야 할 필요는 없다. 특히, 길이 표시자(23)가 슬라이스 헤더에 존재하는 파라미터 세트 식별자(26)에 기초하여 식별된 파라미터 세트에 존재한다면, 관련 파라미터 세트는 파라미터 세트 식별자(26)를 포함하는 비디오 시퀀스의 첫 번째 인코딩된 표현(20)에 대해 식별될 수 있다. 그러면 식별된 파라미터 세트 내의 데이터가 파싱되고, 길이 표시자(23)를 포함한, 그 내부에 있는 디코딩된 값들 및 데이터가 통상적으로 디코더의 메모리에 저장된다. 그러면 디코더는 활성화된 파라미터 세트, 즉, 이미 파싱되고 디코딩된 파라미터 세트를 추적한다. 이것은, 비디오 시퀀스의 후속된 인코딩된 표현(20)이 이 파라미터 세트를 식별하는 파라미터 세트 식별자(26)를 포함할 때, 길이 표시자(23)를 포함한 데이터가 메모리 내의 저장된 값들로부터 회수될 수 있다는 것을 의미한다.

[0060] 따라서, 단계 S1에서 길이 표시자(23)를 파싱하는 단계는, 특히 길이 표시자(23)가 파라미터 세트에 존재한다면, 각각의 인코딩된 표현(20)에 대해 반드시 수행될 필요는 없다. 그러나, 이 단계 S1은, 바람직하게는, 그 슬라이스 헤더(21) 내의 관련 파라미터 세트를 가리키는 파라미터 세트 식별자(26)를 갖는 인코딩된 표

현(20)에 대해 적어도 한 번 수행된다. 이 관련 파라미터 세트에 대한 파라미터 세트 식별자(26)를 포함하는 비디오 시퀀스의 임의의 후속하는 인코딩된 표현(20)에 대해, 단계 S1은, 메모리로부터, 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 판독 또는 회수하는 단계를 포함할 수 있다.

[0061] 길이 표시자(23)(도 3a 및 도 3b 참조) 또는 파라미터 세트 식별자(26)(도 3c 참조)는, 도 3a 내지 도 3c에 도시된 바와 같이 반드시 슬라이스 헤더(21) 내의 확장 필드(24) 바로 앞에 올 필요는 없다. 확장 필드(24)가 길이 표시자(23) 또는 파라미터 세트 식별자(26) 직후에 오게 하는 것이 일반적으로 바람직하다. 그러나, 이것이 반드시 필요한 것은 아니다. 대안적 실시예에서, 다른 슬라이스 헤더 데이터, 즉, 적어도 하나의 선택스 요소 또는 코드워드는, 슬라이스 헤더(21)에서 길이 표시자(23) 또는 파라미터 세트 식별자(26)의 위치와 확장 필드(24) 사이에 존재할 수 있다.

[0062] 도 6은 도 4의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다. 단계 S21에서, 인코딩된 표현과 연관된 슬라이스 헤더 확장 존재 플래그가 파싱된다. 이 슬라이스 헤더 확장 존재 플래그는 슬라이스 헤더(21) 내에 존재하거나 기타의 방식으로 인코딩된 표현(20)과 연관될 수 있다. 예를 들어, 슬라이스 헤더 확장 존재 플래그는, PPS 내 등, 슬라이스와 연관된 파라미터 세트에 존재할 수 있다. 그 다음, 이 방법은 바람직하게는 선택사항적 단계 S20을 포함하고, 이 단계에서, 인코딩된 표현(20)과 연관된 PPS 등의 파라미터 세트가, 슬라이스 헤더(21)에 기초하여 얻어질 수 있는 PPS 식별자 등의 파라미터 세트 식별자에 기초하여 식별된다. 파라미터 세트 식별자는 슬라이스 헤더(21)로부터 직접 회수되거나, 슬라이스 헤더(21)로부터 직접 회수된 또 다른 파라미터 세트 식별자에 의해 식별되는 또 다른 파라미터 세트로부터 회수될 수 있다. 그러면 단계 S21은, 바람직하게는, PPS 내 등의, 파라미터 세트에 존재하는 슬라이스 헤더 확장 플래그를 파싱하는 단계를 포함한다.

[0063] 단계 S21의 슬라이스 헤더 확장 존재 플래그를 파싱하는 단계는, 특히, 슬라이스 헤더 확장 존재 플래그가 단계 S20에서 식별된 파라미터 세트에 존재한다면, 비디오 시퀀스의 모든 인코딩된 표현(20)에 대해 반드시 수행될 필요는 없다. 이러한 경우에, 관련 파라미터 세트는, 이 파라미터 세트를 가리키는 파라미터 세트 식별자(26)를 갖는 첫 번째 인코딩된 표현(20)에 대해 식별되고, 파싱되고, 디코딩될 수 있다. 그 다음, 슬라이스 헤더 확장 존재 플래그를 포함한, 파라미터 세트의 디코딩된 데이터는, 디코더의 메모리에 저장될 수 있다. 이러한 경우에, 슬라이스 헤더 확장 존재 플래그의 값은, 이전에 파싱되고 디코딩된 파라미터 세트를 가리키는 파라미터 세트 식별자(26)를 갖는 임의의 후속하는 인코딩된 표현(20)에 대해 메모리로부터 단순히 판독되거나 회수될 수 있다.

[0064] 슬라이스 헤더 확장 존재 플래그는, 슬라이스 헤더(21) 내에 임의의 확장 필드(24)가 존재하는지의 여부를 나타낸다. 따라서, 슬라이스 헤더 확장 존재 플래그가,  $1_{bin}$  등의 제1 미리정의된 값을 갖는 경우, 슬라이스 헤더(21)에는 확장 필드(24)가 존재하고 인코딩된 표현(20)은 길이 표시자(23)와 연관된다. 그러나, 슬라이스 헤더 확장 존재 플래그가,  $0_{bin}$  등의 제2 미리정의된 값을 갖는 경우, 슬라이스 헤더(21)에는 아무런 확장 필드(24)도 존재하지 않고, 따라서, 아무런 길이 표시자(20)도 인코딩된 표현(23)과 연관되지 않는다.

[0065] 다음 단계 S22에서, 파싱된 슬라이스 헤더 확장 존재 플래그가 분석되고, 이것이 제1 미리정의된 값을 갖는지 또는 제2 미리정의된 값을 갖는지가 판정된다. 예를 들어, 슬라이스 헤더 확장 존재 플래그가 세트되어 있다면, 즉,  $1_{bin}$ 과 같으면, 이 방법은 도 4의 단계 S1을 계속하고, 그 곳에서 길이 표시자(23)가 파싱된다. 따라서, 이 경우 확장 필드(24)가 존재하고 레거시 디코더는 확장 필드(24)에 대응하는 슬라이스 헤더(21)의 부분을 식별할 필요가 있다.

[0066] 슬라이스 헤더 확장 존재 플래그가 세트되어 있지 않으면, 즉,  $0_{bin}$ 과 같으면, 이 방법은 대신에 단계 S23을 계속하고, 그 곳에서, 디코더는 간단히 슬라이스 헤더(21)에 후속하는 슬라이스 데이터(25)를 파싱 및 디코딩하기 시작할 수 있다. 따라서, 이 경우 레거시 디코더에 의해 식별될 필요가 있는 슬라이스 헤더(21)에 존재하는 확장 필드는 없다.

[0067] 특정 실시예에서, 화상 파라미터 세트 RBSP 선택스는 다음과 같을 수 있다:

[0068] `pic_parameter_set_rbsp( ) {` **Descriptor**

```

...
    slice_header_extension_present_flag                u(1)
...
}

```

[0069]

[0070] 0과 동등한 **slice\_header\_extension\_present\_flag**는, 화상 파라미터 세트를 참조하는 코딩된 화상들에 대해 슬라이스 헤더 내에 아무런 슬라이스 헤더 확장 선택스 요소들도 존재하지 않는다는 것을 명시한다.

[0071] 그러면 일반적인 슬라이스 헤더 선택스는 이 특정 실시예에서는 다음과 같다:

```

slice_header(){                                     Descriptor
...
    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[i]      u(8)
        }
        byte_alignment()
    }

```

[0072]

[0073] **slice\_header\_extension\_length**는, **slice\_header\_extension\_length** 자체를 시그널링하는데 이용되는 비트들을 포함하지 않는, 바이트 단위의 슬라이스 헤더 확장 데이터의 길이를 명시한다. **slice\_header\_extension\_length**의 값은 0 내지 256(양쪽 끝 범위 포함)의 범위에 있을 수 있다. **slice\_header\_extension\_length**의 크기는 반드시 UVLC(Universal Variable Length Code)일 필요는 없고, 소정의 양의 정수 q에 대한 고정된 길이, 즉 u(q)일 수 있다.

[0074] **slice\_header\_extension\_data\_byte**는 임의의 값을 가질 수 있다. 레거시 디코더들은 **slice\_header\_extension\_data\_byte**의 값을 무시해야 한다. **slice\_header\_extension\_data\_byte**는 대안적으로, 바이트들, 즉, u(8) 대신에, 비트들, 즉, u(1)을 이용할 수 있다.

[0075] 슬라이스 헤더 확장 존재 플래그는, 디코더가, 간단히, 바람직하게는 1비트 플래그 형태의 이 단일 플래그를 판독함으로써, 인코딩된 표현(20)이, 확장-준수형 디코더에 의해 처리, 즉, 파싱 및 이용되거나 레거시 디코더에 의해 무시될 필요가 있는 임의의 확장 필드(24)를 포함하는지를 판정할 수 있다는 것을 암시한다. 따라서, 슬라이스 헤더 확장 존재 플래그의 이용은, 일부 인코딩된 표현(20)이 확장 필드(24)를 포함할 수 있는 반면 다른 인코딩된 표현들은 포함하지 않는 상황에서 인코딩된 표현(20)의 디코딩을 간소화한다. 이러한 경우, 디코더는 임의의 길이 표시자(23)를 파싱 및 이용할 필요가 없고 간단히 슬라이스 헤더 확장 존재 플래그의 값을 조사할 수 있다.

[0076] 슬라이스 헤더 확장 존재 플래그는 유익하게 파라미터 세트에 삽입되는데, 그 이유는 비디오 스트림 내의 복수의 슬라이스들은 확장 필드(24)를 갖지 않거나 확장 필드(24)를 가질 가능성이 높기 때문이다. 각각의 슬라이스 헤더(21)에서 이러한 슬라이스 헤더 확장 존재 플래그를 포함하는 것에 비해 파라미터 세트에서 슬라이스 헤더 확장 존재 플래그를 시그널링하는 것이 더욱 비트-효율적이다.

[0077] 특정한 실시예에서, 슬라이스 데이터(25)는 인코딩된 표현(20) 내의 확장 필드(24)의 끝의 직후에 시작한다. 따라서, 이 접근법에서, 길이 표시자(23)에 기초하여 식별된 확장 필드(24)의 끝에 후속하는 비트 위치는 슬라이스 데이터(25)의 첫 번째 비트를 구성한다.

[0078] 또 다른 특정 실시예에서, 도 3a에 표시되고 상기에서 일반적인 슬라이스 헤더 선택스에서 추가로 언급된 바와 같이 확장 필드(24)와 슬라이스 데이터(25) 사이에 바이트 정렬 데이터(22)가 삽입될 수 있다. 이러한 바이트 정렬 데이터는 슬라이스 헤더(21)의 끝에서 채워져, 바이트 정렬된 슬라이스 데이터(25)의 선두를 얻을 수 있다. 이러한 경우에, 확장 필드(24)에 의해 점유되고 예를 들어 확장 필드(24)의 길이를 나타냄으로써 확장 필



드(24)의 끝을 시그널링할 수 있는 슬라이스 헤더(21)의 부분을 식별하기 위해 길이 표시자(23)가 이용될 수 있다. 그 이후에 바이트 정렬 데이터(22)가 후속되고 그 다음 슬라이스 데이터(25)가 이어진다. 이것은, 바이트 정렬 데이터(22)가 이용되더라도 디코더는 길이 표시자(23)에 기초하여 슬라이스 데이터(25)의 선두와 확장 필드(24)를 구성하는 슬라이스 헤더(21)의 부분을 식별할 수 있을 것임을 의미한다.

[0079] 바이트 정렬 데이터(22)는 바람직하게는 명확한 신택스를 가진다, 예를 들어:

<i>byte_alignment()</i> {	Descriptor
<i>bit_equal_to_one</i> /* equal to 1 */	f(1)
while( ! <i>byte_aligned()</i> )	
<i>bit_equal_to_zero</i> /* equal to 0 */	f(1)
}	

[0080]

[0081] 이 예에서 바이트 정렬 데이터는 항상 적어도 하나의 비트를 포함한다. 확장 필드(24)의 끝이 바이트 정렬된다면, 8-비트 코드워드, 1000 0000<sub>bin</sub>이 이용된다.

[0082]

도 3b는, 길이 표시자(23) 및 확장 필드(24)가 반드시 슬라이스 헤더(21)의 (선택사항적 바이트 정렬 데이터(22)를 제외한) 마지막 신택스 요소를 형성할 필요는 없다는 것을 보여주는 슬라이스의 인코딩된 표현(20)의 실시예를 나타낸다. 명확한 대조로, 길이 표시자(23) 및 확장 필드(24)(도 3b), 또는 사실상 파라미터 세트 식별자(26) 및 확장 필드(도 3c)는 슬라이스 헤더(21)의 임의의 곳에 존재할 수 있다.

[0083]

도 7은 도 4의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다. 이 방법은 도 4의 단계 S2로부터 계속된다. 다음 단계 S30에서, 인코딩된 표현(20)의 슬라이스 데이터(25)의 선두는, 도 5의 단계 S11에 개시된 바와 같이, 길이 표시자(23)에 기초하여 식별된 확장 필드(24)의 끝에 기초하여 식별된다. 앞서 언급된 바와 같이, 슬라이스 데이터(25)의 선두는 확장 필드(24)의 끝 직후에 올 수 있다. 대안적 접근법에서, 바이트 정렬 데이터(22)는 확장 필드(24)와 슬라이스 데이터(25)의 선두 사이에 개재된다. 이러한 경우, 길이 표시자(23)는 확장 필드(24)의 끝을 식별하는데 이용될 수 있다. 인코딩된 표현(20) 내의 후속 비트 또는 비트들은 바이트 정렬 데이터(22)를 구성한다. 인코딩된 표현(20)의 이 부분은, 상기에서 예시된 바와 같이 명확하게 정의된 신택스를 갖기 때문에, 디코더에 의해 바이트 정렬 데이터(22)로서 해석된다. 일단 바이트 정렬 데이터(22)의 끝에 도달하면, 슬라이스 데이터(25)의 선두가 발견된다. 이후에서 더 설명되는 바와 같이, 다른 슬라이스 헤더 데이터가 확장 필드(24)에 후속될 수 있다는 것도 가능하다.

[0084]

디코더가 슬라이스 데이터(25)의 선두를 식별했을 때, 디코더는 슬라이스 데이터(25)의 식별된 선두로부터 단계 S31에서 슬라이스 데이터(25)를 파싱 및 디코딩할 수 있다.

[0085]

레거시 디코더는 바람직하게는, 단계 S31의 파싱 및 디코딩 프로세스 동안에 슬라이스 헤더(21) 내의 확장 필드(24)의 임의의 값 또는 데이터를 무시, 즉, 이용하지 않음으로써 단계 S31 내의 슬라이스 데이터를 파싱 및 디코딩한다. 따라서, 레거시 디코더는 슬라이스 헤더(21) 내의 확장 필드(24)를 이용할 수 없는데, 그 이유는 이 확장 필드(24)는, 레거시 디코더가 준수하는 기본 프로파일 또는 기본 사양에 따르면, 명시되지 않기 때문이다.

[0086]

그러나, 확장-준수형 디코더는, 선택사항적으로는 슬라이스 데이터(25)의 디코딩 이후 프로세싱을 포함한 파싱 및 디코딩 프로세스 동안에 슬라이스 헤더 내의 확장 필드(24) 내의 데이터를 이용해야 한다.

[0087]

본 실시예는 또한, 슬라이스 헤더(21)가 복수의, 즉, 적어도 2개의 확장 필드(24)를 포함하는 경우에도 적용될 수 있다. 이러한 경우에, 각각의 확장 필드(24)는 바람직하게는 각각의 길이 표시자(23)를 포함한다. 확장 필드(24)는 모두 슬라이스 헤더(21)의 끝과 연계하여 정렬될 수 있다, 아마도, 선택사항적인 바이트 정렬 데이터(22)가 이어질 수 있다. 그러나, 슬라이스 헤더(21) 내의 확장 필드(24)의 분산된 구성을 갖는 것도 역시 가능하다. 이 후자의 경우, 확장 필드(24)는 슬라이스 헤더(21) 내의 임의의 곳에 제공될 수 있다. 예를 들어, 제 1 확장 필드(24)는 슬라이스 헤더(21)의 첫 부분 또는 중간 부분에 존재할 수 있는 반면, 두 번째 확장 필드(24)는 슬라이스 헤더(21)의 끝 부분에 존재할 수 있다.

[0088]

슬라이스 헤더(21)에 복수의 확장 필드(24)가 존재할 수 있다면, 각각의 길이 표시자(23)가 슬라이스 헤더(21)에서 하나 이상의 파라미터 세트에 존재하거나 슬라이스 헤더(21)와 하나 이상의 파라미터 세트 사이에 분산될 수 있다. 예를 들어, 제1 확장 필드(24)의 제1 길이 표시자(23)는 PPS 등의 파라미터 세트에 존재할 수 있다.

그러면 제2 확장 필드(24)의 제2 길이 표시자(23)는 슬라이스 헤더(21)에 존재할 수 있다. 확장 필드(24)와 길이 표시자(23) 중 임의의 것의 존재는, 예를 들어, 파라미터 세트 내의 플래그에 의해 시그널링될 수 있다. 예를 들어, PPS는, 제1 길이 표시자(23)와, 제2 확장 필드(24)의 존재 또는 부재를 나타내는 플래그를 포함할 수 있다. 그러면 제2 길이 표시자(23)와 제2 확장 필드(24)는, 플래그가 정의된 값으로 세트되었다면, 슬라이스 헤더(21) 내에 존재한다.

[0089] 복수의 확장 필드(24)의 경우, 도 4의 단계 S1은 바람직하게는 각각의 확장 필드(24)와 길이 표시자(23)에 대해 한번만 수행된다. 따라서, 슬라이스 헤더(21) 내의 각각의 확장 필드(24)의 식별을 가능케 하기 위하여 인코딩된 표현(20)과 연관된 각각의 길이 표시자(23)가 식별 및 파싱되어, 레거시 디코더는 단계 S2에서 확장 필드의 각각의 값 또는 값들을 무시하기로 결정할 수 있다.

[0090] 도 8는 화상의 슬라이스의 인코딩된 표현을 디코딩하는 방법의 또 다른 실시예를 나타내는 흐름도이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 이 실시예는 특히, 확장-준수형 디코더, 즉, 슬라이스 헤더에 존재하는 확장 필드 내의 값 또는 값들을 이해하고 이용할 수 있는 디코더에 의해 수행된다.

[0091] 이 방법은 단계 S80에서 시작하고, 그 곳에서, 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자가 파싱된다. 이 단계 S1은 기본적으로 도 4의 단계 S1에 대응하므로 여기서는 더 이상 논의되지 않는다.

[0092] 다음 단계 S81은 단계 S80에서 파싱된 길이 표시자에 기초하여 슬라이스 헤더 내의 확장 필드를 식별한다. 따라서, 확장 필드에 의해 점유되는 슬라이스 헤더의 부분을 식별하기 위하여 단계 S81에서는 길이 표시자가 이용된다. 이 단계 S81은, 예를 들어, 도 4 및 도 5와 연계하여, 본 명세서에서 앞서 개시된 바와 같이 수행될 수 있다.

[0093] 슬라이스 헤더가 복수의 확장 필드를 포함한다면, 단계들 S80 및 S81은 바람직하게는 각각의 확장 필드 및 연관된 길이 표시자에 대해 한번 수행된다.

[0094] 후속 단계 S82에서, 슬라이스의 인코딩된 표현은, 단계 S80에서 파싱된 길이 표시자에 기초하여 단계 S81에서 식별된 슬라이스 헤더 내의 확장 필드의 적어도 하나의 값, 바람직하게는 모든 값에 기초하여 디코딩된다.

[0095] 따라서, 이 실시예에서 디코더는 확장-준수형이고, 그에 따라 슬라이스의 인코딩된 표현을 디코딩할 때, 특히 인코딩된 표현 내의 슬라이스 데이터의 디코딩 동안에 확장 필드의 값 또는 값들 또는 값들의 적어도 일부를 이용할 것이다.

[0096] 확장 필드는, 예를 들어, 디코더에게 슬라이스 데이터의 디코딩을 어떻게 수행할지를 지시하고 명령을 제공하는 하나 이상의 플래그 및/또는 다른 선택스 요소들을 포함할 수 있다. 따라서, 확장-준수형 디코더는, 선택사항적으로는 슬라이스 데이터의 디코딩 이후 프로세싱을 포함한 파싱 및 디코딩 프로세스 동안에 슬라이스 헤더 내의 확장 필드 내의 데이터를 이용한다.

[0097] 단계 S82에서 확장 필드의 적어도 한 값에 기초한 슬라이스의 인코딩된 표현의 디코딩은, 디코더에 의해 수행되는 슬라이스의 인코딩된 표현의 임의의 프로세싱에 관련된다. 따라서, 단계 S82에서 정의된 디코딩은, 단순히 슬라이스의 인코딩된 표현에 존재하는 슬라이스 데이터로부터의 슬라이스의 화소 데이터의 실제 생성으로 제한되지 않는다. 이것은, 확장 필드의 적어도 하나의 값이, 예를 들어, 출력되기 위해, 예를 들어, 디스플레이되기 위해 임시로 저장되는, 및/또는 디코딩 순서에 따라 후속하는 비디오 시퀀스의 화상들에 대한 디코딩 기준으로서 이용되기 위해 임시로 저장되는, 비디오 시퀀스의 이전에 디코딩된 화상들을 포함하는 기준 화상 버퍼를 처리하기 위해 디코더에 의해 이용될 수 있다는 것을 의미한다. 확장 필드의 적어도 한 값은, 대안으로서, 예를 들어, 비트스트림의 특정 NAL 유닛을 제거 또는 누락하기 위해 디코더에 의해 이용될 수 있다. 예를 들어, 확장 필드는, 특정 NAL 유닛은 누락될 수 있으므로 상위층 인코딩된 표현만이 디코딩되고 있다면 기준 화상 버퍼에 입력되거나 기준 화상 버퍼로부터 제거될 수 없다는 것을 나타낼 수 있다.

[0098] 따라서, 단계 S82에 정의된 디코딩은, 즉, 현재의 슬라이스의 화소 값들의 실제의 생성과 관련되어 발생하는, 즉, 선행하는, 동시에 또는 후속하는, 디코더의 다양한 동작들을 포괄한다.

[0099] 특정 실시예는, 복수의 화상을 포함하는 비디오 시퀀스 내의 화상의 슬라이스의 인코딩된 표현의 디코딩 동안에 수행되는 방법과 관련된다. 비디오 시퀀스의 각각의 화상은 복수의 층들 중의 층에 속한다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 이 방법은 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하는 단계를 포함한다. 슬라이스 헤더 내의 확장 필드는 길이 표시

자에 기초하여 식별된다. 이 방법은 또한, 슬라이스 헤더 내의 확장 필드의 적어도 하나의 값에 기초하여 상기 복수의 화상들의 층을 이룬 구조(layered structure)를 처리하는 단계를 포함한다.

[0100] 이 특정 실시예에서 비디오 시퀀스의 화상들은 복수의 층들을 수반한 층을 이룬 구조로 조직된다. 그러면 이들 층들은, 멀티뷰 뷰의 경우에는 상이한 뷰들에, 스케일가능한 비디오의 경우에는 상이한 시간적 층들에 대응할 수 있다. 디코더에서, 예를 들어 하나 이상의 화상 또는 화상을 운반하는 NAL 유닛을 제거하는 단계를 포함한, 층을 이룬 구조의 처리는, 바람직하게는 적어도 부분적으로 확장 필드의 적어도 하나의 값에 기초하여 수행된다.

[0101] 여기서 본 실시예의 구현의 변형 및 대안을 개시하는 다수의 예시적 실시예들이 후속된다.

[0102] 제1 실시예에서, 슬라이스 헤더(21) 내의 확장 필드(24)의 길이를 나타내는 슬라이스 헤더(21) 내의 코드워드(신택스 요소), 즉, 길이 표시자(23)가 있다. 인코더는 이 제1 실시예에 따라 신택스 요소로서 길이 표시자(23)를 삽입하도록 구성된다. 디코더는, 슬라이스 데이터(25)의 선두를 찾기 위하여 코드워드의 값, 즉, 길이 표시자(23)의 값을 파싱함으로써, 확장 필드(24)를 건너뛰는 것이다.

[0103] 확장 필드(24)의 길이를 나타내는 코드워드의 이름은 예를 들어 slice\_header\_extension\_length일 수 있다.

[0104] 이 실시예에 따르면, 디코더는 다음과 같은 단계들을 수행하도록 구성된다.

[0105] 1. slice\_header\_extension\_length 신택스 요소(23)가 파싱된다.

[0106] 2. 디코더는 slice\_header\_extension\_length 신택스 요소(23) 직후의 slice\_header\_extension\_length 비트수의 값들을 파싱하고 무시한다.

[0107] 3. 디코더는 slice\_header\_extension\_length 비트수의 파싱 후에 뒤따르는 첫 번째 비트에서 (단계 2의 파싱이 끝나는 비트 위치에서) 시작하여 슬라이스 데이터(25)를 파싱 및 디코딩한다.

[0108] 이 실시예에서, 디코더는 그에 따라 헤더 디코딩 동안에 슬라이스 헤더(21) 내의 확장 필드(24)의 값 또는 데이터를 파싱 및 판독한다. 그러나, 값들이 파싱되고 아마도 디코딩되더라도 디코더는 값들을 무시, 즉, 슬라이스 데이터(25)의 디코딩 및 처리 동안에 이들을 더 이상 이용하지 않는다.

[0109] 대안으로서 디코더는 다음과 같은 단계들 수행하도록 구성될 수 있다:

[0110] 1. slice\_header\_extension\_length 신택스 요소(23)가 파싱된다.

[0111] 2. 디코더는 비트스트림(20)에서 slice\_header\_extension\_length 비트수로 전방 점프한다.

[0112] 3. 디코더는 slice\_header\_extension\_length 비트수의 점프 후에 뒤따르는 첫 번째 비트에서 (단계 2의 점프가 끝나는 비트 위치에서) 시작하여 슬라이스 데이터(25)를 파싱 및 디코딩한다.

[0113] 이 실시예에서, 디코더는 확장 필드(24)의 데이터 또는 값을 파싱 또는 판독하지 않고 길이 표시자(23)에 기초하여 정의된 확장 필드(24)에 의해 점유된 슬라이스 헤더(21)의 부분을 점프한다.

[0114] 제2 실시예에서, 제1 실시예에서 설명된 신택스 요소(23)의 존재는, 예를 들어, slice\_header\_extension\_present\_flag라 불리는 또 다른 신택스 요소를 조건으로 한다. slice\_header\_extension\_present\_flag가 0이면, slice\_header\_extension\_length 신택스 요소(23) 및 확장 데이터(24)는 존재하지 않는다. slice\_header\_extension\_present\_flag가 1이면, slice\_header\_extension\_length 신택스 요소(23) 및 확장 데이터(24)는 존재한다. slice\_header\_extension\_present\_flag는 바람직하게는 SPS에서 시그널링되지만 대안으로서 PPS, VPS, APS 또는 슬라이스 헤더(21)에서 시그널링될 수도 있다.

[0115] 따라서, 인코더는 플래그를 이용하여 제1 실시예의 길이 표시자(23)의 존재를 조절하도록 구성된다.

[0116] 이 실시예에 따르면, 디코더는 다음과 같은 순서화된 단계들을 수행하도록 구성된다:

[0117] 1. 활성 SPS로부터의 slice\_header\_extension\_present\_flag가 이용된다.

[0118] 2. slice\_header\_extension\_present\_flag가 1이면:

[0119] a. slice\_header\_extension\_length 신택스 요소(23)가 파싱된다.

[0120] b. 디코더는 slice\_header\_extension\_length 신택스 요소(23) 직후의 slice\_header\_extension\_length 비트수의 값들을 파싱하고 무시한다.

[0121] c. 디코더는 slice\_header\_extension\_length 비트수의 파싱 후에 뒤따르는 첫 번째 비트에서 (단계 b의 파싱이 끝나는 비트 위치에서) 시작하여 슬라이스 데이터(25)를 파싱 및 디코딩한다.

[0122] 3. 그렇지 않고, (slice\_header\_extension\_present\_flag가 0과 같다면), 디코더는 slice\_header\_extension\_length 선택스 요소(23)를 파싱하지 않고 및 임의의 확장 데이터(24)를 파싱하지 않고 슬라이스 데이터(25)를 파싱 및 디코딩한다. slice\_header\_extension\_length 선택스 요소(23)도 확장 데이터(24)도 슬라이스에 존재하지 않을 것이다.

[0123] 대안으로서, 디코더는 다음과 같은 단계들 수행하도록 구성될 수 있다:

[0124] 1. 활성 SPS로부터의 slice\_header\_extension\_present\_flag가 이용된다.

[0125] 2. slice\_header\_extension\_present\_flag가 1이면:

[0126] a. slice\_header\_extension\_length 선택스 요소(23)가 파싱된다.

[0127] b. 디코더는 비트스트림(20)에서 slice\_header\_extension\_length 비트수로 전방 점프한다.

[0128] c. 디코더는 slice\_header\_extension\_length 비트수의 점프 후에 뒤따르는 첫 번째 비트에서 (단계 b의 점프가 끝나는 비트 위치에서) 시작하여 슬라이스 데이터를 파싱 및 디코딩한다.

[0129] 3. 그렇지 않고, (slice\_header\_extension\_present\_flag가 0과 같다면), 디코더는 slice\_header\_extension\_length 선택스 요소(23)를 파싱하지 않고 및 비트스트림(20)에서 점프를 수행하지 않고 슬라이스 데이터(25)를 파싱 및 디코딩한다. slice\_header\_extension\_length 선택스 요소(23)도 확장 데이터(24)도 슬라이스에 존재하지 않을 것이다.

[0130] 선택스 테이블은 예를 들어 다음과 같을 수 있다:

<i>slice_header()</i>	Descriptor
...	
if( <i>slice_header_extension_present_flag</i> ) {	
<i>slice_header_extension_length</i>	ue(v)
for( i = 0; i < <i>slice_header_extension_length</i> ; i++)	
<i>slice_header_extension_data_flag</i>	u(1)
}	
}	

[0131] 제3 실시예에서, 전체 슬라이스 헤더(21)의 길이는 선택스 요소(23)에 의해 주어지고 확장 필드(24)는 파라미터 세트의 확장 필드와 유사하게 슬라이스 헤더(21)의 끝에 추가된다.

[0132] 따라서, 인코더는, 선택스 요소, 즉, 길이 표시자(23)에 의해 전체 슬라이스 헤더(21)의 길이를 표시하도록 구성되고, 확장 필드(24)는 파라미터 세트의 확장 필드와 유사하게 슬라이스 헤더(21)의 끝에 추가된다.

[0133] 이 실시예에 따르면, 디코더는 다음과 같은 단계들을 수행하도록 구성된다:

[0134] 1. 활성 SPS로부터의 slice\_header\_extension\_present\_flag가, 존재한다면, 이용된다. 그렇지 않으면, 단계들 1, 2, 및 3은 생략되고 단계들 a, b, 및 c만이 수행된다.

[0135] 2. slice\_header\_extension\_present\_flag가 1이면:

[0136] a. slice\_header\_extension\_length가 슬라이스 헤더(21)의 길이로부터 현재의 비트 위치를 뺀 값으로서 계산된다.

[0137] b. 디코더는, slice\_header\_extension\_length 선택스 요소, 즉, 전체 슬라이스 헤더(21)의 길이를 나타내는 선택스 요소 직후에 뒤따르는 slice\_header\_extension\_length 비트수의 값들을 파싱하고 무시한다.

[0138] c. 디코더는 slice\_header\_extension\_length 비트수의 파싱 후에 뒤따르는 첫 번째 비트에서



(단계 b의 파싱이 끝나는 비트 위치에서) 시작하여 슬라이스 데이터(25)를 파싱 및 디코딩한다.

- [0141] 3. 그렇지 않고, (slice\_header\_extension\_present\_flag가 0과 같다면), 디코더는 slice\_header\_extension\_length 선택 요소(23)를 파싱하지 않고 및 임의의 확장 데이터(24)를 파싱하지 않고 슬라이스 데이터(25)를 파싱 및 디코딩한다. slice\_header\_extension\_length 선택 요소(23)도 확장 데이터(24)도 슬라이스에 존재하지 않을 것이다.
- [0142] 따라서, 확장 필드(24)의 길이의 결정에 관련된 도 5의 단계 S10은 바람직하게는, 길이 표시자(23)가 나타낸 값으로부터 슬라이스 헤더(21) 내의 길이 표시자(23)의 끝의 비트 또는 바이트 위치를 감산하여 확장 필드(24)의 길이를 계산함으로써 수행된다. 길이 표시자(23)가 나타내는 이 값은 바람직하게는 슬라이스 헤더(21)의 길이이다.
- [0143] 제1 및 제2 실시예의 단계 3과 제3 실시예의 단계 c의 변형에서, 확장 필드(24)와 슬라이스 데이터(25) 사이에는 바이트 정렬 데이터(22)가 개재될 수 있다. 이러한 경우, 디코더는, slice\_header\_extension\_length 비트수의 파싱에 후속하는 첫 번째 비트에서 시작하거나(제1 및 제3 실시예) slice\_header\_extension\_length 비트수의 점프에 후속하는 첫 번째 비트에서 시작하여(제2 실시예) 바이트 정렬 데이터(22)를 파싱한다. 그 다음, 디코더는, 일단 바이트 정렬 데이터(22)의 끝에 도달하고 나면, 슬라이스 데이터(25)를 파싱 및 디코딩한다.
- [0144] 제4 실시예에서, 특정한 비트 패턴("시작코드")이 이용되어 확장 데이터 필드(24)의 끝을 표시한다. 따라서, 인코더는 이러한 특정한 비트-패턴을 삽입하도록 구성된다. 디코더는 이 비트-패턴이 발견될 때까지 모든 데이터를 파싱한다. 비트-패턴은 바이트 정렬될 것이 요구될 수도 아닐 수도 있다. 그 다음, 디코더는 특정한 비트-패턴 이후의 첫 번째 비트에서 시작하여 슬라이스 데이터(25)를 디코딩한다.
- [0145] 디코더는 다음과 같은 단계들을 수행할 수 있다:
- [0146] 1. 확장 필드(24)가 시작하는 지점까지 슬라이스 헤더(21)가 파싱된다.
- [0147] 2. 그 지점으로부터 디코더는 비트스트림을 스캔하여, 확장 데이터(24)가 어디에서 끝나고 슬라이스 데이터(25)가 어디에서 시작하는지를 나타내는 특정한 비트-패턴("시작 코드")을 탐색한다.
- [0148] 3. 특정 비트-패턴이 발견되는 지점 이후에, 디코더는 슬라이스를 파싱 및 디코딩한다.
- [0149] 따라서, 이 실시예에서 도 4에 개시된 방법은, 슬라이스 헤더(21)의 선두로부터 슬라이스 헤더(21) 내의 확장 필드(24)의 선두까지 슬라이스 헤더(21)를 파싱하는 단계를 더 포함한다. 이 방법은, 확장 필드(24)의 선두로부터 특정 시작 코드의 형태로 되어 있는 길이 표시자(23)에 도달할 때까지 슬라이스 헤더(21)를 스캔하는 단계를 더 포함한다. 이 방법은 또한, 확장 필드(24)의 끝을 특정 시작 코드로서 식별하는 단계를 포함한다.
- [0150] 제5 실시예에서, 제1 내지 제3 실시예 중 임의의 것이 이용되지만, slice\_header\_extension\_length(23)가 슬라이스 헤더(21)에서 시그널링되지 않는다는 차이를 가진다. 대신에, 이것은, SPS, VPS, PPS 또는 APS 등의 슬라이스 헤더(21)에 의해 참조될 수 있는 데이터 구조에서 시그널링된다.
- [0151] 제6 실시예에서, 상기 제시된 실시예들 중 임의의 것이 이용되지만, 길이가 비트 대신에 바이트로 표현된다는 차이를 가진다.
- [0152] 제7 실시예에서, slice\_header\_extension\_present\_flag가 반드시 0으로 세트되어야 한다는 비트스트림 요건과 함께 제2 내지 제6 실시예 중 임의의 것이 이용된다. 인코더는 slice\_header\_extension\_present\_flag를 0으로 설정하도록 구성될 수 있다.
- [0153] 특정 실시예에서, 소정 프로파일, 특히 기본 프로파일을 따르는 인코더는 바람직하게는 slice\_header\_extension\_present\_flag를 0으로 설정하도록 강제된다. 그러나, 동일한 프로파일을 따르는 디코더는 길이 표시자를 이용해야 하고 확장 데이터를 무시해야 한다.
- [0154] 제8 실시예에서, 확장 데이터 필드(24)를 위해 도입된 특정 의미와 함께 제2 내지 제6 실시예 중 임의의 것이 이용된다. 이러한 의미는 멀티-뷰 확장 또는 스케일가능한 확장에서 추가 정보에 대응할 수 있다.
- [0155] 인코더는, slice\_header\_extension\_present\_flag를 1로 설정함으로써 실시예를 이용할 수 있고 각각의 슬라이스 헤더(21)에 대해 도입된 의미에 따라 확장 데이터 필드(23)의 값들을 인코딩한다. 인코더는 또한, 확장 데이터 필드(24)의 길이를 계산하고 그것을 slice\_header\_extension\_length 선택 요소(23)에서 시그널링하도록 구성된다. 대안으로서, 확장(24)의 길이가 미리 알려지면(예를 들어, 고정된 길이 확장이 이용되면), 길이는

이것을 먼저 계산할 필요없이 slice\_header\_extension\_length 선택스 요소(23)에서 시그널링될 수 있다.

[0156] 확장 데이터(24)에 대한 도입된 의미를 인식(및 이용)하는 디코더는 다음과 같은 단계들을 수행하도록 계산될 수 있다:

[0157] 1. 활성 SPS로부터의 slice\_header\_extension\_present\_flag가 이용된다.

[0158] 2. slice\_header\_extension\_present\_flag가 1이면:

[0159] a. slice\_header\_extension\_length 선택스 요소(23)가 파싱된다(또는 확장 길이가 이미 알려져 있다면 무시된다).

[0160] b. 디코더는, 비트들의 도입된 의미에 따라 slice\_header\_extension\_length 선택스 요소(23) 직후에 이어지는 slice\_header\_extension\_length 비트수의 값들을 파싱 및 이용한다.

[0161] c. 디코더는 slice\_header\_extension\_length 비트수의 점프 후에 뒤따르는 첫 번째 비트에서 (단계 b의 점프가 끝나는 비트 위치에서) 시작하여 슬라이스 데이터(25)를 파싱 및 디코딩한다.

[0162] 3. 그렇지 않고, (slice\_header\_extension\_present\_flag가 0과 같다면), 디코더는 slice\_header\_extension\_length 선택스 요소(23)를 파싱하지 않고 및 비트스트림(20)에서 점프를 수행하지 않고 슬라이스 데이터(25)를 파싱 및 디코딩한다. slice\_header\_extension\_length 선택스 요소(23)도 확장 데이터(24)도 슬라이스에 존재하지 않을 것이다.

[0163] 확장 데이터(24)에 대한 도입된 의미를 인식하지(또는 이용하지) 않는 디코더는 제2 실시예에서 제시된 단계들을 수행하도록 구성될 수 있다.

[0164] 제9 실시예에서, HEVC의 하나 이상의 프로파일에 예측된 확장 데이터 필드(24)의 특정 의미와 함께 제8 실시예가 이용된다. 즉, (명시된 의미를 갖는) 특정 목적을 위해 확장 데이터 필드(24)를 이용하는 프로파일을 따르는 비트스트림(20)을 디코딩하는 디코더는, 제8 실시예에서 설명된 바와 같이 확장 데이터(24)를 디코딩하고, 정의된 의미에 따라 확장 데이터(24)를 이용할 것이다.

[0165] 특정한 목적을 위해 확장 데이터(24)를 이용하지 않는 프로파일을 따르는 비트스트림(20)을 디코딩하는 디코더는 제2 실시예에서 제시된 단계들을 수행할 것이다.

[0166] 제10 실시예에서, 제2, 제3, 제8 또는 제9 실시예 중 임의의 것이 이용되지만 slice\_header\_extension\_present\_flag가 없으며, 그 대신에 이 플래그가 1인 것으로 유추된다. 제로 확장 길이를 시그널링함으로써 어떠한 확장도 시그널링되지 않는다.

[0167] 제11 실시예에서, 상기 실시예들 중 임의의 것이 이용될 수 있지만 확장(24)의 크기는 시그널링되지 않는다. 대신에 일련의 고정된 및/또는 가변 크기 가변 길이 코드(VLC)가 정의된다.

[0168] 제12 실시예에서, 소정의 슬라이스에 대해 2개의 슬라이스 헤더 메커니즘이 이용될 수 있다. 이 경우, 제1 길이 표시자는 PPS 등의 파라미터 세트에 시그널링되는 반면 제2 길이 표시자는 슬라이스 헤더에서 시그널링된다. 실시예에서, PPS 내의 플래그는 슬라이스 헤더가 제2 길이 표시자를 포함하는지의 여부를 나타낸다.

[0169] 따라서 PPS RBSP 선택스는 다음과 같을 수 있다:

<i>pic_parameter_set_rbsp()</i> {	Descriptor
...	
<b><i>num_slice_header_extension_bits</i></b>	u(5)
<b><i>slice_header_extension_present_flag</i></b>	u(1)

[0170] ...

[0171] 이 예에서, num\_slice\_header\_extension\_bits는 제1 길이 표시자를 나타낸다. 0과 동일한 이 제1 길이 표시자의 값은, PPS를 참조하는 코딩된 화상에 대한 슬라이스 헤더 RBSP에 어떠한 여분의 슬라이스 헤더 비트도 없다는 것을 명시한다. 선택스 요소 num\_slice\_header\_extension\_bits는 기본 프로파일 명세를 따르는 비트스트림에 대해 0과 동일해야 한다. 플래그 slice\_header\_extension\_present\_flag는 슬라이스 헤더 내의 제2 길이 표시자의 존재를 시그널링하는데 이용된다. 이 플래그의 값 0은, PPS를 참조하는 코딩된 화상들에 대해 슬라이스

헤더 내에 아무런 슬라이스 헤더 확장 선택스 요소들도 존재하지 않는다는 것을 명시한다. 플래그 `slice_header_extension_present_flag`는 기본 프로파일 명세를 따르는 비트스트림에 대해 0과 동일해야 한다.

[0173] 그러면 슬라이스 층 RBSP는 다음과 같을 수 있다:

```
slice_layer_rbsp() {
    slice_header()
    slice_data()
    rbsp_slice_trailing_bits()
}
```

[0174] 대응하여, 슬라이스 헤더 선택스는 하기에 정의된 바와 같을 것이다:  
[0175]

<code>slice_header()</code> {	Descriptor
...	
for( i = 0; i < num_slice_header_extension_bits; i++ )	
<b>slice_header_extension_data_flag</b>	u(1)
...	
if( slice_header_extension_present_flag ) {	
<b>slice_header_extension_length</b>	ue(v)
for( i = 0; i < slice_header_extension_length; i++ )	
<b>slice_header_extension_data_byte</b>	u(8)
}	
byte_alignment()	
}	

[0176]  
[0177] 슬라이스 헤더 내의 제1 확장 필드는, 이 예에서, `slice_header_extension_data_flag`로 구성된다. 제2 길이 표시자, `slice_header_extension_length`는, `slice_header_extension_length` 길이 자체를 시그널링하는데 이용되는 비트들을 포함하지 않는, 제2 확장 필드의 길이, 즉, `slice_header_extension_data_byte`를 명시한다. 예에서, `slice_header_extension_length`의 값은 0 내지 256(양쪽 끝 범위 포함)의 범위에 있을 수 있다. 슬라이스 헤더 내의 제2 확장 필드는 임의의 값을 가질 수 있다.

[0178] 제13 실시예는 전술된 제12 실시예와 유사하지만 다음과 같은 슬라이스 층 RBSP 선택스를 가진다.

<code>slice_layer_rbsp()</code> {	Descriptor
<b>slice_header()</b>	
<b>slice_data()</b>	
if( slice_header_extension_present_flag ) {	
<b>slice_extension_flag</b>	u(1)
if( slice_extension_flag )	
while( more_rbsp_data() )	
<b>slice_extension_data_flag</b>	u(1)
}	
rbsp_slice_trailing_bits()	
}	

[0179]

[0180] 제12 및 제13의 구현예에서, 기본 사양은 다음과 같을 수 있다:

```
for(i = 0; i < num_slice_header_extension_bits; i++)
    slice_header_extension_data_flag
```

[0181]

[0182] 확장 명세는 이것을 다음과 같이 할 수 있다:

```
inter_layer_prediction_flag
```

[0183]

[0184] 0과 동일한 inter\_layer\_prediction\_flag는, NAL 유닛과 연관된 화상이 상위 공간 또는 멀티-뷰 층에 속하는 임의의 화상에서의 참조에 이용되지 않는다는 것을 나타낸다. 1과 동일한 inter\_layer\_prediction\_flag는, NAL 유닛과 연관된 화상이 상위층 또는 멀티-뷰 층에 속하는 화상에서의 참조에 이용될 수 있다는 것을 나타낸다.

[0185] 따라서, 이 예시의 구현에서 슬라이스 헤더 내의 제1 확장 필드는, 확장 명세를 따르는 디코더에 의해, inter\_layer\_prediction\_flag를 나타내는 것으로서 해석될 수 있다.

[0186] 구현 실시예에서, 슬라이스 헤더 내의 확장 필드 또는 확장 필드들은, 확장 명세에서, 확장 플래그, 예를 들어 제12 및 제13 실시예를 참조한 slice\_header\_extension\_data\_flag 및 slice\_header\_extension\_data\_byte 대신에, 실제의 코드워드로서 해석되거나 실제의 코드워드로 대체된다. 코드워드는, 플래그, 고정 길이 코드(FLC; fixed length code) 워드, 가변 길이 코드(VLC; variable length code) 워드, 또는 심지어 이들의 혼합의 형태일 수도 있다.

[0187] 또한, 확장-준수형 디코더가 확장 필드 내의 비트들 중 일부를 무시하는 것도 가능하다. 예를 들어, 이것은, 확장 필드의 코드워드가 12 비트에 달하지만 길이 표시자에 의해 시그널링된 확장 필드의 총 길이가 14 비트로 설정되는 경우에 해당될 수 있다. 그러면 나머지 2개 비트들은 확장 준수형 디코더에 의해 단순히 무시된다.

[0188] 앞서 언급된 바와 같이, 확장 필드(24)는, 예에서, 인코딩된 멀티-뷰 확장 데이터 및/또는 인코딩된 스케일가능한 확장 데이터를 포함할 수 있다.

[0189] 전자의 경우, 비디오 스트림은 복수의 (카메라) 뷰, 예를 들어, 입체 비디오의 경우 우측 카메라 뷰 및 좌측 카메라 뷰로부터의 화상들의 스트림이다. 이러한 경우, 이들 복수의, 즉, 적어도 2개의 뷰들 중 하나는 베이스 뷰일 수 있고 하나 이상의 추가 뷰들은 베이스 뷰에 관해 및 아마도 서로에 관해 인코딩될 수 있다. 그러나, 베이스 뷰는 바람직하게는 독립적으로 디코딩가능하다, 즉, 베이스 뷰는 참조를 위한 다른 뷰 내의 화상들을 이용하지 않는다. 이러한 경우, 레거시 디코더는, 추가의 뷰에 관련된 또는 슬라이스 헤더에 존재하는 멀티-뷰 구조에 관련된 확장 데이터가 있더라도 베이스 뷰를 정확하게 디코딩할 수 있다. 이것은, 여기서 개시된 다양한 실시예들에 따라 이용되는 길이 표시자 때문에 가능하다.

[0190] 후자의 경우, 비디오 스트림은 복수의 층으로부터의 화상들의 스트림이다. 이러한 경우, 이들 복수의 층들 중 하나는 하나 이상의 추가 층을 갖는 베이스 층이다. 베이스 층은 바람직하게는 독립적으로, 즉, 추가 층들 중 임의의 층 내의 화상을 참조하지 않고 디코딩가능하다. 대응적으로, 추가 층 N에 속하는 화상은 바람직하게는 베이스 층 및 N까지의 층들 내의 화상을 참조할 수 있지만 N보다 큰 층에 대해서는 참조하지 않는다. 이러한 경우, 레거시 디코더는, 추가의 층들에 관련된 또는 슬라이스 헤더에 존재하는 다중층 구조에 관련된 확장 데이터가 있더라도 베이스 층을 정확하게 디코딩할 수 있다. 이것은, 여기서 개시된 다양한 실시예들에 따라 이용되는 길이 표시자 때문에 가능하다.

[0191] 실시예들은 HEVC로 제한되지 않고, 스케일가능한 확장 또는 멀티-뷰 확장 등의 HEVC의 임의의 확장에, 또는 상이한 비디오 코덱에, 또는 사실상 기타 임의의 비디오 코딩 표준에 적용될 수 있다.

[0192] 도 9는, 비디오 스트림 1 내의 화상 2 등의, 화상 2의 슬라이스 3을 인코딩하는 방법의 개략적 흐름도이다. 이 방법은 단계 S40에서 길이 표시자(23)를 생성하는 단계를 포함한다. 이 길이 표시자(23)는 슬라이스(3)의 슬라이스 헤더(21)에 존재하는 확장 필드(24)의 길이를 나타낸다. 다음 단계 S41에서, 슬라이스(3)는, 슬라이스 헤더(21)와 슬라이스 데이터(25)를 포함하는 슬라이스(3)의 인코딩된 표현(20)으로 인코딩된다. 길이 표시자(23)는 단계 S42에서 인코딩된 표현(20)과 연관된다. 단계 S42는 단계 S41보다 앞에, 실질적으로 병행하여, 또는 그 이후에 수행될 수 있다.

[0193] 인코딩된 표현(20)은 더 처리되어 NAL 유닛(11)을 형성할 수 있고, 이 NAL 유닛은 선택사항으로서 도 2에 도시



된 바와 같은 하나 이상의 헤더(12)를 갖는 데이터 패킷(10)으로 패키징될 수 있다.

- [0194] 슬라이스 헤더가 복수의 확장 필드를 포함할 것이라면, 도 9의 단계들 S40 및 S42는 바람직하게는 각각의 이러한 확장 필드에 대해 한번이다.
- [0195] 도 10은 도 9의 방법의 추가의 선택사항적 단계들을 나타내는 흐름도이다. 단계 S51에서, 슬라이스 헤더(21)에 존재하는 확장 필드(24)의 길이가 결정된다. 그 다음 도 9의 단계 S40에서 확장 필드(24)의 길이에 기초하여 길이 표시자(23)가 생성된다. 따라서, 특정 실시예에서, 길이 표시자(23)는, 바이트 또는 비트 단위의 확장 필드(24)의 길이를 나타내는 값을 부여받는다.
- [0196] 선택사항적 단계 S50에서, 확장 필드(24)의 값들은 확장 필드(24)가 얼마나 길게 되는지를 알기 위하여 결정된다. 대안으로서, 확장 필드(24)는, 확장 데이터가, 멀티-뷰 확장 데이터 또는 스케일가능한 확장 데이터 등의, 어떤 타입인지에 따라, 미리정의된 길이를 가질 수 있다. 앞서 언급된 바와 같이, 확장 필드(24)의 값들은 바람직하게는 슬라이스(3)에 대한 멀티-뷰 확장 데이터 및/또는 스케일가능한 확장 데이터를 나타낸다.
- [0197] 단계 S40에서 생성된 길이 표시자(23)는 바람직하게는 단계 S42에서 슬라이스 헤더(21) 내에 삽입된다. 대안으로서, 길이 표시자(23)는 현재 슬라이스(3)에 적용되는 파라미터 세트 내에 삽입될 수 있다. 이러한 경우에, 슬라이스 헤더(21)는 바람직하게는, 길이 표시자(23)를 포함하는 관련 파라미터 세트의 식별을 가능케하는 파라미터 세트 식별자를 포함한다.
- [0198] 길이 표시자(23)가 단계 S42에서 슬라이스 헤더(21) 내에 삽입되면, 길이 표시자(23)는 바람직하게 슬라이스 헤더(21) 내의 확장 필드(24) 바로 앞에 삽입된다.
- [0199] 실시예에서, 슬라이스 헤더 확장 존재 플래그는 슬라이스 헤더(21) 내의 확장 필드(24)의 존재 또는 부재를 시그널링하는데 이용된다. 이 실시예가 도 11에 개략적으로 도시되어 있다. 이 방법은 단계 S60에서 시작하고, 그 곳에서 슬라이스 헤더 확장 존재 플래그는  $1_{bin}$  등의 미리정의된 값으로 설정되어 슬라이스 헤더(21) 내의 확장 필드(24)의 존재를 표시한다. 다음 단계 S61에서, 슬라이스 헤더 확장 존재 플래그는 인코딩된 표현(20)과 연관된다. 단계 S61의 이 연관은 현재의 슬라이스(3)에 적용가능한 파라미터 세트 내의 슬라이스 헤더 확장 존재 플래그의 삽입될 수 있다. 이러한 경우에, 슬라이스 헤더(21)는 바람직하게는, 길이 표시자(23)를 포함하는 관련 파라미터 세트의 식별을 가능케하는 파라미터 세트 식별자를 포함한다.
- [0200] 도 12는 도 11의 연관 단계 S61의 특정 실시예를 나타낸다. 이 실시예에서, 슬라이스 헤더 확장 존재 플래그가, 단계 S70에서, 파라미터 세트, 예를 들어, PPS 내에 삽입된다. 슬라이스 헤더 확장 존재 플래그를 포함하는 파라미터 세트를 식별하는 파라미터 세트 식별자(26)는 바람직하게는, 파라미터 세트 식별자(26)를 인코딩된 표현(20)의 슬라이스 헤더(21) 내에 삽입하는 등에 의해, 단계 S71에서 인코딩된 표현(20)과 연관된다.
- [0201] 대안적 접근법에서, 슬라이스 헤더 확장 존재 플래그는 슬라이스 헤더(21) 내에 삽입된다.
- [0202] 도 13은 실시예에 따른 디코더(40)의 개략적 블록도이다. 디코더(40)는, 화상의 슬라이스의 인코딩된 표현과 연관된 길이 표시자를 파싱하도록 구성된, 파서 또는 파싱 수단 또는 모듈이라고도 표기되는, 파싱 유닛(41)을 포함한다. 길이 표시자는 인코딩된 표현의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타낸다. 디코더(40)는 또한, 인코딩된 표현의 디코딩 동안에, 길이 표시자에 기초하여 식별된 확장 필드의 임의의 값을 디코더(40)가 무시하게 하도록 구성된, 프로세서 또는 처리 수단 또는 모듈이라고도 표기되는, 처리 유닛(42)을 포함한다.
- [0203] 선택사항적 실시예에서, 디코더(40)는 또한, 길이 표시자에 기초하여 확장 필드의 길이를 결정하도록 구성된, 결정기 또는 결정 수단 또는 모듈이라고도 표기되는, 결정 유닛(43)을 포함한다. 디코더(40)의 식별기 또는 식별 수단 또는 모듈이라고도 표기되는, 선택사항적 식별 유닛(44)은 바람직하게는 결정 유닛(43)에 의해 결정된 길이에 기초하여 슬라이스 헤더 내의 확장 필드의 끝을 식별하도록 구성된다.
- [0204] 파싱 유닛(41)은 바람직하게는 인코딩된 표현의 슬라이스 헤더에 존재하는 길이 표시자를 파싱하도록 구성된다. 대안으로서, 파싱 유닛(41)은, 인코딩된 표현의 슬라이스 헤더에서 파싱된 파라미터 세트 식별자에 기초하여 식별된 파라미터 세트 내의 길이 표시자를 파싱할 수 있다. 후자의 경우, 파싱 유닛(41)은, 이전에 인코딩된 표현에 대해 길이 표시자를 갖는 파라미터 세트가 이미 파싱 및 디코딩되었다면, 여기서 앞서 개시된 바와 같이 연관된 메모리로부터 길이 표시자를 회수할 수 있다.
- [0205] 선택사항적 실시예에서, 처리 유닛(42)은, 디코더(40)가, 인코딩된 표현의 디코딩 동안에, 슬라이스 헤더 내의

길이 표시자 직후부터 예를 들어 식별 유닛(44)에 의해 길이 표시자에 기초하여 식별된 확장 필드의 끝까지의 확장 필드의 임의의 값을 무시하게 하도록 구성된다.

[0206] 파싱 유닛(41)은 또한 바람직하게는 인코딩된 표현과 연관된 슬라이스 헤더 확장 플래그를 파싱하도록 구성된다. 이러한 경우, 슬라이스 헤더 확장 플래그가 미리정의된 값, 바람직하게는  $1_{bin}$ 을 갖는다면, 파싱 유닛(41)은 길이 표시자를 파싱하도록 구성되고, 처리 유닛(42)은 디코더(40)가 확장 필드의 임의의 값을 무시하게 하도록 구성된다.

[0207] 선택사항적 실시예에서, 디코더(40)는, 슬라이스 헤더에 기초하여 얻을 수 있는 파라미터 세트 식별자, 바람직하게는 PPS 식별자에 기초하여 인코딩된 표현과 연관된 파라미터 세트, 바람직하게는 PPS를 식별하도록 구성된, 세트 식별자 또는 세트 식별 수단 또는 모듈이라고도 표기되는, 세트 식별 유닛(45)을 더 포함한다. 특정 실시예에서, 세트 식별 유닛(45)은 바람직하게는, 인코딩된 표현의 슬라이스 헤더로부터 파라미터 세트 식별자를 회수하고 이것을 이용해 슬라이스 헤더 확장 존재 플래그를 운반하는 관련 파라미터 세트를 식별한다. 이러한 경우, 파싱 유닛(41)은, 세트 식별 유닛(45)에 의해 식별되는 파라미터 세트, 바람직하게는 PPS에 존재하는 슬라이스 헤더 확장 존재 플래그를 파싱하도록 구성된다.

[0208] 앞서 언급된 식별 유닛(44)은, 예를 들어 식별 유닛(44)에 의해 길이 표시자에 기초하여 식별된 확장 필드의 끝에 기초하여 인코딩된 표현 내의 슬라이스 데이터의 선두를 식별하도록 구성된 선택사항적 실시예에 있다. 그 다음, 디코더(40)는, 식별 유닛(44)에 의해 식별된 슬라이스 데이터의 선두로부터 시작하여 슬라이스 데이터를 파싱 및 디코딩하도록 구성된, 데이터 디코더 또는 디코딩 수단 또는 모듈이라고도 표시되는, 디코딩 유닛(46)을 포함한다.

[0209] 디코딩 유닛(46)은 바람직하게는, 슬라이스 헤더 내의 확장 필드의 임의의 값을 무시하고 이용하지 않음으로써 슬라이스 데이터를 파싱 및 디코딩하도록 구성된다. 따라서, 디코딩 유닛(46)은, 확장 필드 내의 또는 확장 필드에 의해 표시된 임의의 값을 이용하지 않고 인코딩된 표현(20)의 디코딩을 수행한다.

[0210] 특정 실시예에서, 디코더(40)는, 슬라이스 헤더의 길이를 나타내기 위한 길이 표시자가 이용되는지를 나타내는 플래그를 파싱하기 위한 유닛(41)을 포함한다. 또한, 디코더(40)는 길이 표시자를 파싱하기 위한 유닛(41)을 포함하고 이에 기초해 슬라이스 헤더 내의 확장 필드의 길이를 결정한다. 그러면, 디코더(40)는 인코딩된 NAL 유닛을 디코딩할 때 그 정보를 이용할 수 있다.

[0211] 도 14는 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 또 다른 실시예에 따른 디코더(90)의 개략적 블록도이다. 디코더(90)는, 인코딩된 표현과 연관된 길이 표시자를 파싱하도록 구성된, 파서 또는 파싱 수단 또는 모듈이라고도 표기되는, 파싱 유닛(91)을 포함한다. 길이 표시자는 인코딩된 표현의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타낸다. 확장 필드 식별기 또는 식별 수단 또는 모듈이라고도 표기되는, 확장 필드 식별 유닛(92)은, 파싱 유닛(91)에 의해 파싱된 길이 표시자에 기초하여 슬라이스 헤더 내의 확장 필드를 식별하도록 구성된다. 디코더(90)는 또한, 데이터 디코더 또는 디코딩 수단 또는 모듈이라고도 표기되는, 디코딩 유닛(93)을 포함한다. 이 디코딩 유닛(93)은, 확장 필드 식별 유닛(92)에 의해 식별된 슬라이스 헤더 내의 확장 필드의 적어도 하나의 값에 기초하여 슬라이스의 인코딩된 표현을 디코딩하도록 구성된다.

[0212] 도 13 및 도 14의 디코더들(40, 90)과 그들의 포함하는 유닛들(41-46, 91-93)은 하드웨어로 구현될 수 있다. 디코더(40, 90)의 유닛들(41-46, 91-93)의 기능들을 달성하기 위해 이용되고 결합될 수 있는 회로 요소들의 수많은 변형이 있다. 이러한 변형들은 실시예에 의해 포괄된다. 디코더(40, 90)의 하드웨어 구현의 특정 예는, 범용 전자 회로 및 응용-특유의 회로 양쪽 모두를 포함한, 디지털 신호 처리기(DSP) 하드웨어 및 집적 회로 기술 형태의 구현이다.

[0213] 여기서 설명된 디코더(50)는 대안으로서, 도 15에 도시된 바와 같이, 예를 들어, 적절한 스토리지 또는 메모리(54)와 함께 프로세서(52)와 적절한 소프트웨어 중 하나 이상에 의해, 따라서, 프로그래머블 로직 장치(PLD) 또는 기타의 전자 컴포넌트(들)에 의해 구현될 수 있다. 또한, 디코더(50)는 바람직하게는, NAL 유닛의 형태 등의, 슬라이스의 인코딩된 표현을 수신하도록 구성된 입력 또는 입력 유닛(51)을 포함한다. 대응하는 출력 또는 출력 유닛(53)은 디코딩된 슬라이스를 출력하도록 구성된다.

[0214] 디코더(32)는, 예를 들어, 비디오 카메라, 셋탑 박스 또는 디스플레이 내 등의, 예를 들어 도 16에 도시된 모바일 장치 내의, 수신기(30)에 위치할 수 있다. 수신기(30)는, 도 2에 도시된 NAL 유닛의 데이터 패킷 등의, 코딩된 비트스트림을 수신하도록 구성된 입력 또는 입력 유닛(31)을 포함한다. NAL 유닛의 인코딩된 표현은 여기서 개시된 디코더(32)에 의해 디코딩된다. 디코더(32)는 바람직하게는, 비디오 스트림 내의 다른 화상들에 대

한 기준 화상으로서 이용되는 이미 디코딩된 화상을 임시로 저장하는 기준 화상 버퍼(34)를 포함하거나 이에 접속된다. 디코딩된 화상들은, 출력 또는 출력 유닛(33)에 의해, 기준 화상 버퍼(34)로부터 등의, 수신기(30)로부터 출력된다. 이들 출력된 화상들은, 수신기(30)의 또는 수신기(30)에 무선 접속되는 것을 포함한 수신기(30)에 접속된, 스크린이나 디스플레이 상에서 사용자에게 디스플레이된다.

[0215] 따라서, 실시예는, 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 디코더를 포함하는 수신기(30)에 관한 것이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 수신기의 디코더는 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하도록 구성된 파싱 유닛을 포함한다. 디코더는 또한, 인코딩된 표현의 디코딩 동안에, 길이 표시자에 기초하여 식별된 확장 필드의 임의의 값을 디코더가 무시하게 하도록 구성된 처리 유닛을 포함한다.

[0216] 또 다른 실시예는, 화상의 슬라이스의 인코딩된 표현을 디코딩하기 위한 디코더를 포함하는 수신기(30)에 관한 것이다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 수신기의 디코더는 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하도록 구성된 파싱 유닛을 포함한다. 디코더는 또한, 길이 식별자에 기초하여 슬라이스 헤더 내의 확장 필드를 식별하도록 구성된 확장 필드 식별 유닛을 포함한다. 디코더의 디코딩 유닛은 확장 필드 식별 유닛에 의해 식별된 슬라이스 헤더 내의 확장 필드의 적어도 한 값에 기초하여 슬라이스의 인코딩된 표현을 디코딩하도록 구성된다.

[0217] 도 17은 실시예에 따른 화상의 슬라이스를 인코딩하도록 구성된 인코더(70)의 개략적 블록도이다. 인코더(70)는, 슬라이스의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타내는 길이 표시자를 생성하도록 구성된, 생성기 또는 생성 수단 또는 모듈이라고도 표기되는, 생성 유닛(71)을 포함한다. 인코더(70)의, 슬라이스 인코더 또는 인코딩 수단 또는 모듈이라고도 표기되는, 인코딩 유닛(72)은, 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스의 인코딩된 표현으로 슬라이스를 인코딩하도록 구성된다. 인코더(70)는 또한, 길이 표시자를 인코딩된 표현과 연관시키도록 구성된, 연관기 또는 연관 수단 또는 모듈이라고도 표기되는, 연관 유닛(73)을 포함한다.

[0218] 선택사항적 실시예에서, 인코더(70)는 또한, 슬라이스 헤더에 존재하는 확장 필드의 길이를 결정하도록 구성된, 결정기 또는 결정 수단 또는 모듈이라고도 표기되는, 결정 유닛(74)을 포함한다. 그 다음, 생성 유닛(71)은 결정 유닛(74)에 의해 결정된 확장 필드의 길이에 기초하여 길이 표시자를 생성하도록 구성된다.

[0219] 선택사항적 실시예는, 바람직하게는 인코딩 유닛(72)이 슬라이스를 인코딩하기 이전에, 슬라이스 헤더에 길이 표시자를 삽입하도록 구성된 연관 유닛(73)을 가진다. 이러한 경우에, 연관 유닛(73)은 바람직하게는 슬라이스 헤더 내의 확장 필드 바로 앞에 길이 표시자를 삽입한다.

[0220] 인코더(70)는, 슬라이스 헤더 내의 확장 필드의 존재를 나타내기 위해, 슬라이스 헤더 확장 존재 플래그를 미리 정의된 값, 바람직하게는,  $1_{bin}$ 에 설정하도록 구성된, 설정기 또는 설정 수단 또는 모듈이라고도 표기되는, 설정 유닛(75)을 포함한다. 그 다음, 연관 유닛(73)은 슬라이스 헤더 확장 존재 플래그를 인코딩된 표현과 연관시키도록 구성된다. 특정 실시예에서, 연관 유닛(73)은, 파라미터 세트, 바람직하게는 PPS에 슬라이스 헤더 확장 존재 플래그를 삽입하도록 구성된다. 연관 유닛(73)은 또한, 파라미터 세트, 바람직하게는 PPS를 식별하는 파라미터 세트 식별자, 바람직하게는 PPS 식별자를 인코딩된 표현과 연관시키도록 구성된다. 특히, 연관 유닛(73)은, 파라미터 세트 식별자, 바람직하게는 PPS 식별자를 슬라이스 헤더 내에 삽입하도록 구성된다.

[0221] 특정 실시예에서, 인코더(70)는, 슬라이스 헤더 내의 확장 필드의 길이를 결정하기 위한 유닛(74)과 길이 표시자를 삽입하기 위한 유닛(73)을 포함한다.

[0222] 도 17의 인코더(70)와 그 포함하는 유닛들(71-75)은 하드웨어로 구현될 수 있다. 인코더(70)의 유닛들(71-75)의 기능들을 달성하기 위해 이용되고 결합될 수 있는 회로 요소들의 수많은 변형이 있다. 이러한 변형들은 실시예에 의해 포괄된다. 인코더(70)의 하드웨어 구현의 특정 예는, 범용 전자 회로 및 응용-특유의 회로 양쪽 모두를 포함한, 디지털 신호 처리기(DSP) 하드웨어 및 집적 회로 기술 형태의 구현이다.

[0223] 여기서 설명된 인코더(80)는 대안으로서, 도 18에 도시된 바와 같이, 예를 들어, 적절한 스토리지 또는 메모리(84)와 함께 프로세서(82)와 적절한 소프트웨어 중 하나 이상에 의해, 따라서, 프로그래머블 로직 장치(PLD) 또는 기타의 전자 컴포넌트(들)에 의해 구현될 수 있다. 또한, 인코더(80)는 바람직하게는, 비디오 스트림의 화상들을 수신하도록 구성된 입력 또는 입력 유닛(81)을 포함한다. 대응하는 출력 또는 출력 유닛(83)은, 바람직하게는 NAL 유닛 형태의 슬라이스들의 인코딩된 표현을 출력하도록 구성된다.

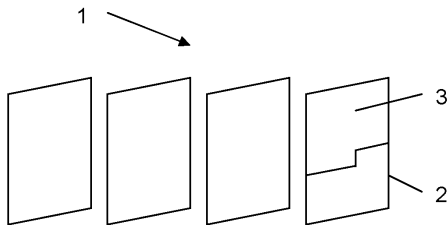
- [0224] 인코더(62)는, 예를 들어, 비디오 카메라 내의, 예를 들어, 도 19에 도시된 모바일 장치 내의, 전송기(60)에 위치할 수 있다. 전송기(60)는 인코딩된 비디오 스트림의 화상을 수신하도록 구성된 입력 또는 입력 유닛(61)을 포함한다. 화상들은 여기서 개시된 바와 같이 인코더(62)에 의해 인코딩된다. 인코딩된 화상들은, 도 2에 도시된 바와 같이 NAL 유닛 또는 NAL 유닛 등을 운반하는 데이터 패킷 등의, 코딩된 비트스트림 형태로 출력 또는 출력 유닛(63)에 의해 전송기(60)로부터 출력된다.
- [0225] 실시예는 화상의 슬라이스를 인코딩하기 위한 인코더를 포함하는 전송기(60)에 관한 것이다. 인코더는 슬라이스의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타내는 길이 표시자를 생성하도록 구성된 생성 유닛을 포함한다. 인코더는 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스의 인코딩된 표현으로 슬라이스를 인코딩하도록 구성된 인코딩 유닛을 포함한다. 인코더의 연관 유닛은 길이 표시자를 인코딩된 표현과 연관시키도록 구성된다.
- [0226] 실시예들은, 디코더, 인코더, 및 네트워크-노드나 미디어 인식 네트워크 요소(Media Aware Network Element) 등의, 비트스트림 상에서 동작하는 및 임의의 요소에 적용된다.
- [0227] 상호작용하는 유닛 또는 모듈 뿐만 아니라 유닛의 명명은 예시의 목적일 뿐이고, 개시된 프로세스 동작들을 실행할 수 있기 위하여 복수의 대안적 방식들로 구성될 수 있다는 것을 이해할 것이다.
- [0228] 본 개시에서 설명된 유닛들 또는 모듈들은 논리적 엔티티로서 간주되어야 하며 반드시 별개의 물리적 엔티티로서 간주되어서는 안 된다는 점에 유의해야 한다. 여기서 개시된 기술의 범위는 당업자에게 자명한 다른 실시예를 완전히 포괄하며, 본 개시의 범위는 그에 따라 제한되지 않는다는 점을 이해할 것이다.
- [0229] 단수 형태의 요소에 대한 언급은 명시적으로 진술되지 않는 한 "단 하나만의(one and only one)"를 의미하는 것이 아니라, "하나 또는 그 이상"을 의미한다. 당업자에게 공지된 전술된 실시예의 요소들의 모든 구조적 및 기능적 등가물들은 참조에 의해 명시적으로 본 명세서에 병합되며 여기에 포함되는 것으로 의도된다. 게다가, 한 장치 또는 방법이, 본 명세서에 포함된다는 이유로 여기서 설명된 기술에 의해 해결하고자 하는 각각의 및 모든 문제를 해결할 필요는 없다.
- [0230] 상기 설명에서, 제한이 아닌 설명의 목적으로, 개시된 기술의 철저한 이해를 제공하기 위하여 특정한 아키텍처, 인터페이스, 기술 등의 특정한 상세사항들이 개시되어 있다. 그러나, 개시된 기술은 이들 특정한 상세사항으로부터 벗어난 다른 실시예들 및/또는 실시예들의 조합으로 실시될 수 있다는 것은 당업자에게 명백할 것이다. 즉, 당업자라면, 여기서 명시적으로 설명되거나 도시되지는 않았지만, 개시된 기술의 원리를 구현하는 다양한 배치를 고안할 수 있을 것이다. 일부 예에서, 공지된 장치, 회로, 및 방법의 상세한 설명은 불필요한 상세사항으로 개시된 기술의 설명을 흐리게 하지 않도록 생략된다. 게다가, 개시된 기술의 원리, 양태, 및 양태들 뿐만 아니라 그 특정한 예를 기재한 본 명세서의 모든 기재사항은 구조적 및 기능적 그 균등물 모두를 포괄하는 것으로 의도되었다. 추가적으로, 이러한 균등물은 현재 알려진 균등물 뿐만 아니라 향후에 개발될 균등물, 예를 들어, 구조에 관계없이 동일한 기능을 수행하도록 개발된 임의의 요소들 모두를 포함하는 것으로 의도한다.
- [0231] 따라서, 예를 들어, 당업자라면, 여기서의 블록도들은 본 기술의 원리를 구현하는 예시적 회로 또는 기타의 기능 회로의 개념적 뷰를 나타낼 수 있다는 것을 이해할 것이다. 유사하게, 임의의 플로차트, 상태 천이도, 의사 코드 등은, 컴퓨터 판독가능한 매체로 실질적으로 표현되고 명시적으로 도시되었든 아니든 컴퓨터 또는 프로세서에 의해 실행될 수 있는 다양한 프로세스들을 나타낸다는 것을 이해할 것이다.
- [0232] 기능 블록들을 포함하는 다양한 요소들의 기능들은, 컴퓨터 판독가능한 매체에 저장된 코딩된 명령어들의 형태로 된 소프트웨어를 실행할 수 있는 회로 하드웨어 및/또는 하드웨어 등의 하드웨어의 이용을 통해 제공될 수 있다. 따라서, 이러한 기능들 및 예시된 기능 블록들은 하드웨어-구현 및/또는 컴퓨터-구현되므로 머신-구현되는 것으로 이해되어야 한다.
- [0233] 도 20은, 프로세서(110)와 컴퓨터 프로그램(130)을 저장하고 있는 컴퓨터 판독가능한 매체(120)를 포함하는 컴퓨터(100)의 개략도이다.
- [0234] 실시예에서, 컴퓨터 프로그램(130)은 화상의 슬라이스의 인코딩된 표현을 디코딩하도록 구성된다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 컴퓨터 프로그램(130)은, 프로세서(110)에 의해 실행될 때, 프로세서(110)로 하여금, 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하게 하는 코드 수단을 포함한다. 프로세서(110)는 또한, 인코딩된 표현의 디코딩 동안에, 길이 표시자에 기초하여 식별된 상기 확장 필드의 임의의 값을 무시하기로 결정하게 된다.



- [0235] 또 다른 실시예에서, 컴퓨터 프로그램(130)은 화상의 슬라이스의 인코딩된 표현을 디코딩하도록 구성된다. 인코딩된 표현은 슬라이스 헤더와 슬라이스 데이터를 포함한다. 컴퓨터 프로그램(130)은, 프로세서(110)에 의해 실행될 때, 프로세서(110)로 하여금, 인코딩된 표현과 연관되고 슬라이스 헤더 내의 확장 필드의 길이를 나타내는 길이 표시자를 파싱하게 하는 코드 수단을 포함한다. 프로세서(110)는 또한, 길이 표시자에 기초하여, 상기 슬라이스 헤더 내의 확장 필드를 식별하고, 길이 표시자에 기초하여 식별된 확장 필드의 적어도 한 값에 기초하여 슬라이스의 인코딩된 표현을 디코딩하게 된다.
- [0236] 추가 실시예에서, 컴퓨터 프로그램(130)은 화상의 슬라이스를 인코딩하도록 구성된다. 컴퓨터 프로그램(130)은, 프로세서(110)에 의해 실행될 때, 프로세서(110)로 하여금, 슬라이스의 슬라이스 헤더에 존재하는 확장 필드의 길이를 나타내는 길이 표시자를 생성하게 하는 코드 수단을 포함한다. 프로세서(110)는 또한, 슬라이스 헤더와 슬라이스 데이터를 포함하는 슬라이스의 인코딩된 표현으로 슬라이스를 인코딩하게 된다. 프로세서(110)는 길이 표시자를 인코딩된 표현과 연관시키게 된다.
- [0237] 실시예는, 컴퓨터 판독가능한 매체(120), 및 컴퓨터 판독가능한 매체(120)에 저장된 상기 개시된 임의의 실시예에 따라 정의되는 컴퓨터 프로그램(130)을 포함하는 컴퓨터 프로그램 제품에 관한 것이다.
- [0238] 하드웨어 구현의 관점에서, 기능 블록들은, 제한없이, 디지털 신호 프로세서(DSP) 하드웨어, 축소된 명령어 세트 프로세서(reduced instruction set processor), 주문형 집적 회로(들)(ASIC) 및 (적절하다면) 이러한 기능들을 수행할 수 있는 상태 머신들을 포함하지만 이들로 제한되지 않는 하드웨어(예를 들어, 디지털 또는 아날로그) 회로를 포함하거나 포괄할 수 있다.
- [0239] 전술된 실시예들은 본 발명의 몇 가지 예시로서 이해되어야 한다. 당업자라면, 본 발명의 범위로부터 벗어나지 않고 실시예들에 대해 다양한 수정, 조합, 및 변경이 이루어질 수 있다는 것을 이해할 것이다. 특히, 상이한 실시예들 내의 상이한 부분적 해결책들은, 기술적으로 가능하다면, 다른 구성들로 조합될 수 있다. 그러나, 본 발명의 범위는 첨부된 청구항들에 정의된다.

## 도면

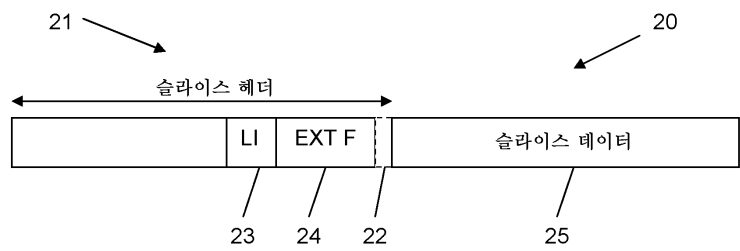
### 도면1



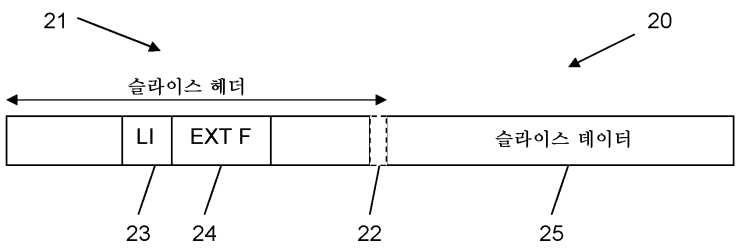
### 도면2



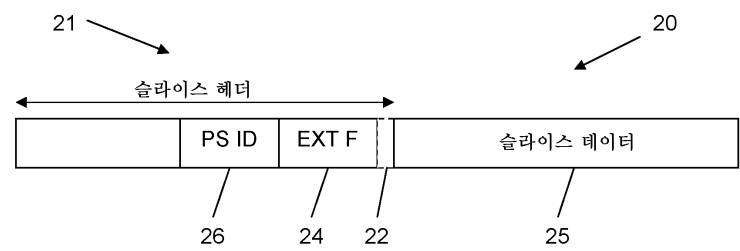
도면3a



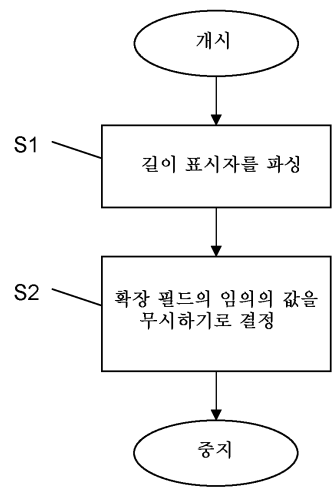
도면3b



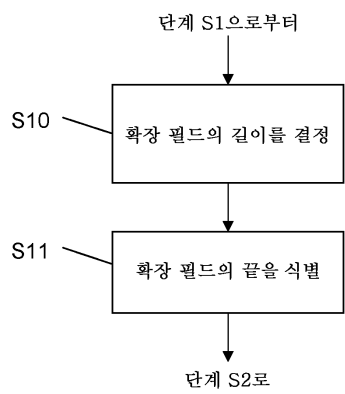
도면3c



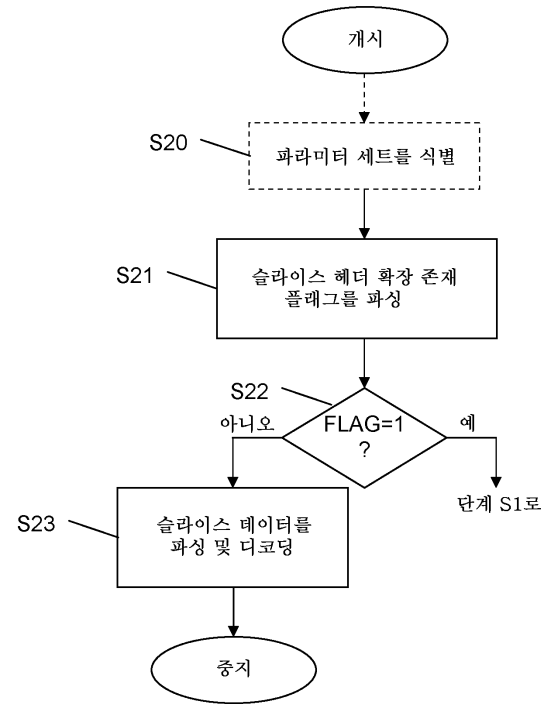
도면4



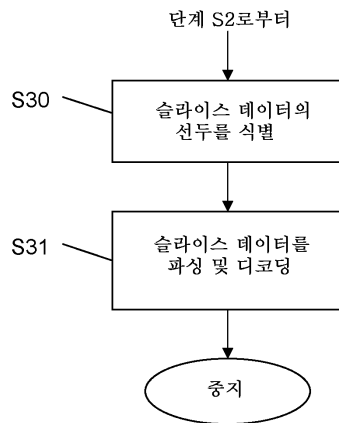
도면5



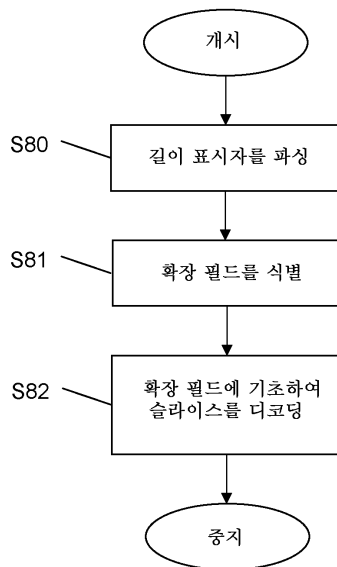
도면6



도면7

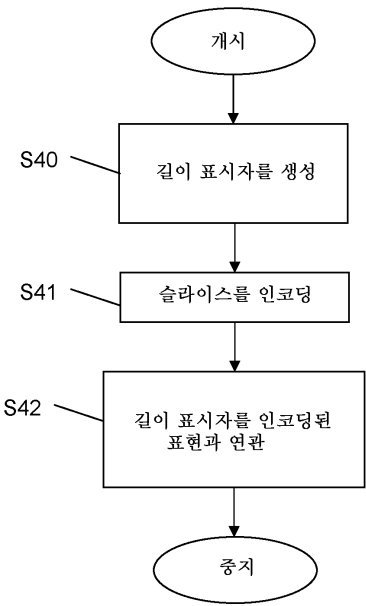


도면8

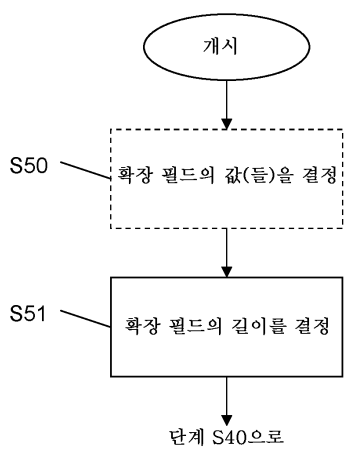




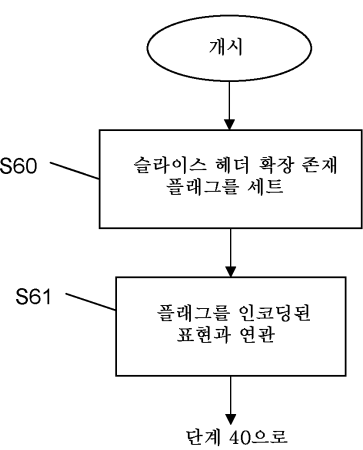
도면9



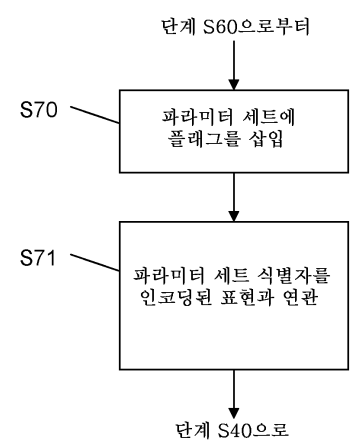
도면10



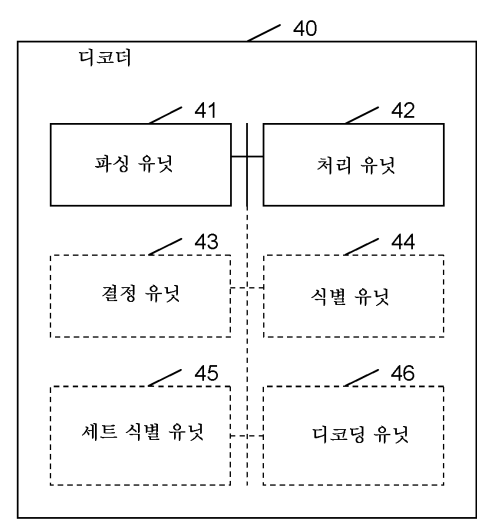
도면11



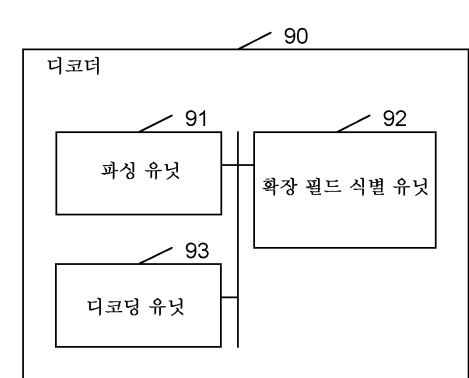
도면12



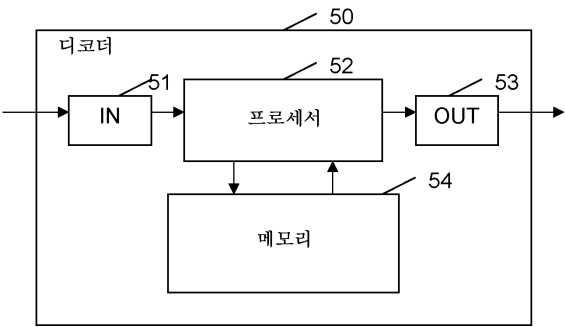
도면13



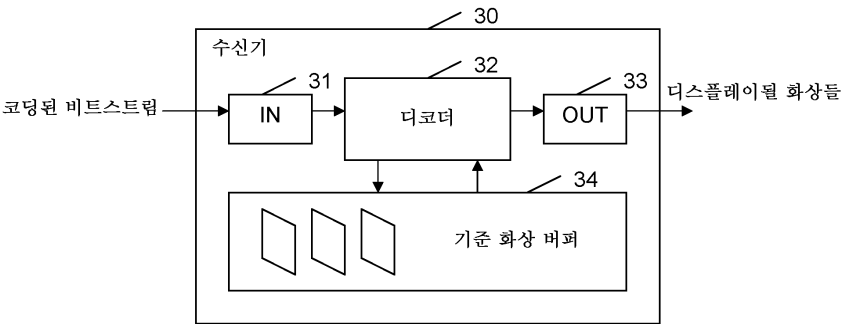
도면14



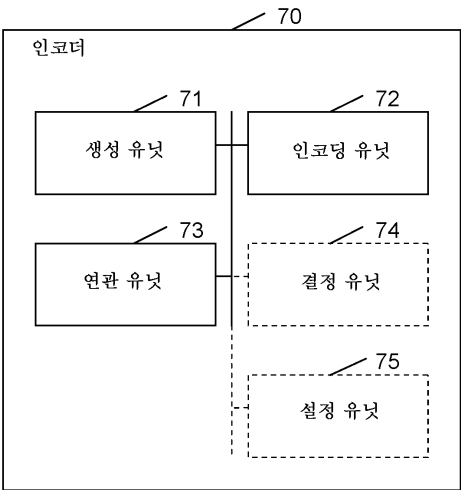
도면15



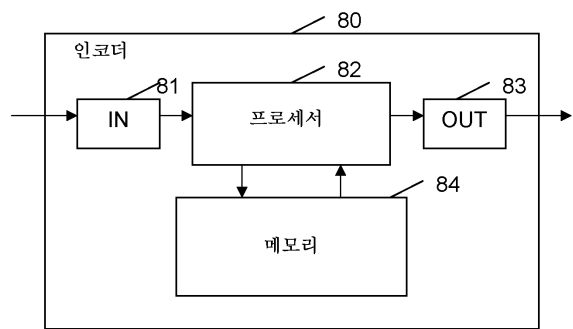
도면16



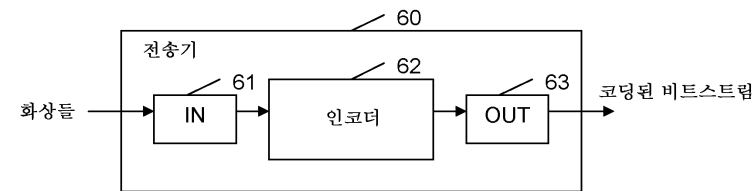
도면17



도면18



도면19



도면20

