



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0120628
(43) 공개일자 2022년08월30일

(51) 국제특허분류(Int. Cl.)

H04N 19/119 (2014.01) H04N 19/117 (2014.01)
H04N 19/136 (2014.01) H04N 19/167 (2014.01)
H04N 19/174 (2014.01) H04N 19/184 (2014.01)
H04N 19/70 (2014.01) H04N 19/82 (2014.01)

(52) CPC특허분류

H04N 19/119 (2015.01)
H04N 19/117 (2015.01)

(21) 출원번호 10-2022-7025413

(22) 출원일자(국제) 2022년12월18일

심사청구일자 없음

(85) 번역문제출일자 2022년07월21일

(86) 국제출원번호 PCT/US2020/066009

(87) 국제공개번호 WO 2021/133672

국제공개일자 2021년07월01일

(30) 우선권주장

62/954,014 2019년12월27일 미국(US)

(71) 출원인

알리바바 그룹 홀딩 리미티드

케이만군도, 그랜드 케이만, 조지 타운, 피.오.
박스 847, 원 캐피탈 플레이스, 플로어 4

(72) 발명자

첸, 지에

중국 베이징 차오양 디스트릭트 왕징 둥 위안 더
4티에이지 디스트릭트 빌딩 엔오. 6

예, 안

미국 캘리포니아 92130 샌디에고 펄맨 웨이 5001
(뒷면에 계속)

(74) 대리인

특허법인 광장리앤코

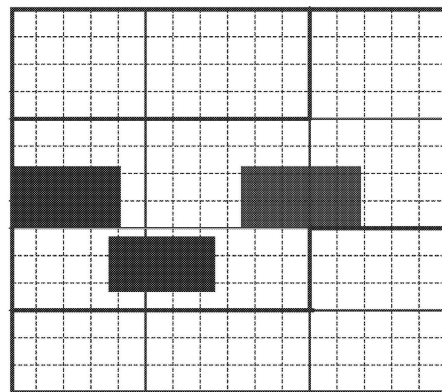
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 서버화상 파티셔닝 정보를 시그널링하기 위한 방법 및 장치

(57) 요약

본 개시내용은 서버화상 파티셔닝 정보를 시그널링하기 위한 방법들 및 장치들을 제공한다. 예시적인 방법은: 비트스트림에서 시그널링되는 서버화상 정보 존재 플래그에 따라, 비트스트림이 서버화상 정보를 포함하는지를 결정하는 단계; 및 비트스트림이 서버화상 정보를 포함하는 것에 대한 응답으로, 화상 내 서버화상들의 수, 타겟 서버화상의 폭, 높이, 포지션 및 식별자(ID) 매핑, subpic_treated_as_pic_flag, 및 loop_filter_across_subpic_enabled_flag 중 적어도 하나를 비트스트림에서 시그널링하는 단계를 포함한다.

대표도



CTU 타일 슬라이스

타일 및 래스터-스캔 슬라이스들로 파티셔닝된 화상의 예

(52) CPC특허분류

HO4N 19/136 (2015.01)

HO4N 19/167 (2015.01)

HO4N 19/174 (2015.01)

HO4N 19/184 (2015.01)

HO4N 19/70 (2015.01)

HO4N 19/82 (2015.01)

(72) 발명자

루오, 지안쿵

미국 뉴저지 08558 스킵맨 리치몬드 드라이브 28

라오, 루-링

중국 베이징 차오양 디스트릭트 왕징 둥 위안 더 4
디스트릭트 라디언스 진 후이 타워

명세서

청구범위

청구항 1

비디오 프로세싱 방법으로서,

비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 상기 비트스트림이 서브화상 정보를 포함하는지를 결정하는 단계; 및

상기 비트스트림이 상기 서브화상 정보를 포함하는 것에 대한 응답으로,

화상 내 서브화상들의 수,

타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑,

subpic_treated_as_pic_flag, 및

loop_filter_across_subpic_enabled_flag 중 적어도 하나를 상기 비트스트림에서 시그널링하는 단계를 포함하는,

비디오 프로세싱 방법.

청구항 2

제1 항에 있어서,

상기 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 상기 화상 내 상기 서브화상들의 수에 기초하는,

비디오 프로세싱 방법.

청구항 3

제2 항에 있어서,

상기 화상에 적어도 2개의 서브화상들이 있을 때, 상기 subpic_treated_as_pic_flag, 상기 loop_filter_across_subpic_enabled_flag, 및 상기 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나를 시그널링하는 단계를 더 포함하고,

상기 화상에 단 하나의 서브화상만이 있을 때, 상기 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 스킵되고,

상기 subpic_treated_as_pic_flag는 코딩 층 비디오 시퀀스(CLVs) 내 각각의 코딩된 화상의 서브화상들이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되는지를 표시하고, 그리고

상기 loop_filter_across_subpic_enabled_flag는 상기 CLVS 내 각각의 코딩된 화상의 서브화상 경계들에 걸쳐 서브화상 경계들에 걸친 인-루프 필터링 동작들이 인에이블되는지를 표시하는,

비디오 프로세싱 방법.

청구항 4

제3 항에 있어서,

상기 타겟 서브화상의 폭이 시그널링되지 않을 때, 상기 화상의 폭으로서 상기 타겟 서브화상의 폭의 값을 결정하는 단계; 및

상기 타겟 서브화상의 높이가 시그널링되지 않을 때, 상기 화상의 높이로서 상기 타겟 서브화상의 높이의 값을 결정하는 단계를 더 포함하는,

비디오 프로세싱 방법.

청구항 5

제4 항에 있어서,

상기 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 상기 화상의 폭으로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 폭의 값을 결정하는 단계; 및

상기 타겟 서브화상의 높이가 시그널링되지 않을 때, 상기 CTB 크기 단위의 상기 화상의 높이로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 높이의 값을 결정하는 단계를 더 포함하는,

비디오 프로세싱 방법.

청구항 6

제3 항에 있어서,

상기 subpic_treated_as_pic_flag가 상기 비트스트림에서 시그널링되지 않을 때, 상기 subpic_treated_as_pic_flag가 1의 값을 갖는 것으로 결정하는 단계; 및

상기 loop_filter_across_subpic_enabled_flag가 상기 비트스트림에서 시그널링되지 않을 때, 상기 loop_filter_across_subpic_enabled_flag가 0의 값을 갖는 것으로 결정하는 단계를 더 포함하는,

비디오 프로세싱 방법.

청구항 7

제1 항에 있어서,

상기 타겟 서브화상이 상기 화상에서 마지막 서브화상일 때, 상기 타겟 서브화상의 폭 및 높이 중 적어도 하나를 시그널링하는 것을 스킵하는 단계를 더 포함하는,

비디오 프로세싱 방법.

청구항 8

제7 항에 있어서,

상기 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 상기 화상의 폭에서 상기 CTB 크기 단위의 상기 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 위치션을 뺀 것으로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 폭의 값을 결정하거나, 상기 화상의 폭에서 상기 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 위치션을 뺀 것으로서 상기 타겟 서브화상의 폭의 값을 결정하는 단계; 및

상기 타겟 서브화상의 높이가 시그널링되지 않을 때, 상기 CTB 크기 단위의 상기 화상의 높이에서 상기 CTB 크기 단위의 상기 타겟 서브화상의 상단 좌측 CTU의 수직 위치션을 뺀 것으로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 높이의 값을 결정하거나, 상기 화상 높이에서 상기 타겟 서브화상의 상단 좌측 CTU의 수직 위치션을 뺀 것으로서 상기 타겟 서브화상의 높이의 값을 결정하는 단계를 더 포함하는,

비디오 프로세싱 방법.

청구항 9

제1 항에 있어서,

상기 타겟 서브화상의 ID 매핑을 시그널링하는 단계는,

상기 비트스트림에서 제1 플래그를 시그널링하는 단계; 및

상기 제1 플래그가 1과 동일한 것에 대한 응답으로, 제1 데이터 유닛 또는 제2 데이터 유닛에서 상기 타겟 서브화상의 ID 매핑을 시그널링하는 단계를 더 포함하고,

0과 동일한 상기 제1 플래그는 상기 타겟 서브화상의 ID 매핑이 상기 비트스트림에서 시그널링되지 않음을 표시

하는,

비디오 프로세싱 방법.

청구항 10

제9 항에 있어서,

상기 제1 플래그가 1과 동일하고 상기 타겟 서브화상의 ID 매핑이 상기 제1 데이터 유닛에서 시그널링되지 않는 것에 대한 응답으로, 상기 제2 데이터 유닛에서 상기 타겟 서브화상의 ID 매핑을 시그널링하는 단계; 또는

상기 제1 플래그가 0과 동일하거나 상기 타겟 서브화상의 ID 매핑이 상기 제1 데이터 유닛에서 시그널링되는 것에 대한 응답으로, 상기 제2 데이터 유닛에서 상기 타겟 서브화상의 ID 매핑을 시그널링하는 것을 스킵하는 단계를 더 포함하는,

비디오 프로세싱 방법.

청구항 11

제10 항에 있어서,

상기 제1 데이터 유닛 및 상기 제2 데이터 유닛 각각은 시퀀스 파라미터 세트(SPS), 화상 파라미터 세트(PPS), 또는 화상 헤더(PH) 중 하나인,

비디오 프로세싱 방법.

청구항 12

비디오 프로세싱 장치로서,

명령들을 저장하기 위한 적어도 하나의 메모리; 및

상기 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고,

상기 명령들은, 상기 장치로 하여금,

비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 상기 비트스트림이 서브화상 정보를 포함하는지를 결정하는 것; 및

상기 비트스트림이 상기 서브화상 정보를 포함하는 것에 대한 응답으로,

화상 내 서브화상들의 수,

타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑,

subpic_treated_as_pic_flag, 및

loop_filter_across_subpic_enabled_flag 중 적어도 하나를 상기 비트스트림에서 시그널링하는 것을 수행하게 하기 위한 것인,

비디오 프로세싱 장치.

청구항 13

제12 항에 있어서,

상기 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 상기 화상 내 상기 서브화상들의 수에 기초하는,

비디오 프로세싱 장치.

청구항 14

제13 항에 있어서,

상기 적어도 하나의 프로세서는 상기 장치로 하여금,

상기 화상에 적어도 2개의 서브화상들이 있을 때, 상기 `subpic_treated_as_pic_flag`, 상기 `loop_filter_across_subpic_enabled_flag`, 및 상기 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나를 시그널링하는 것을 수행하게 하기 위한 상기 명령들을 실행하도록 구성되고,

상기 화상에 단 하나의 서브화상만이 있을 때, 상기 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 스킵되고,

상기 `subpic_treated_as_pic_flag`는 코딩 층 비디오 시퀀스(CLVS) 내 각각의 코딩된 화상의 서브화상들이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되는지를 표시하고, 그리고

상기 `loop_filter_across_subpic_enabled_flag`는 상기 CLVS 내 각각의 코딩된 화상의 서브화상 경계들에 걸쳐 서브화상 경계들에 걸친 인-루프 필터링 동작들이 인에이블되는지를 표시하는,

비디오 프로세싱 장치.

청구항 15

제14 항에 있어서,

상기 적어도 하나의 프로세서는 상기 장치로 하여금,

상기 타겟 서브화상의 폭이 시그널링되지 않을 때, 상기 화상의 폭으로서 상기 타겟 서브화상의 폭의 값을 결정하는 것; 및

상기 타겟 서브화상의 높이가 시그널링되지 않을 때, 상기 화상의 높이로서 상기 타겟 서브화상의 높이의 값을 결정하는 것을 수행하게 하기 위한 상기 명령들을 실행하도록 구성되는,

비디오 프로세싱 장치.

청구항 16

제15 항에 있어서,

상기 적어도 하나의 프로세서는 상기 장치로 하여금,

상기 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 상기 화상의 폭으로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 폭의 값을 결정하는 것; 및

상기 타겟 서브화상의 높이가 시그널링되지 않을 때, 상기 CTB 크기 단위의 상기 화상의 높이로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 높이의 값을 결정하는 것을 수행하게 하기 위한 상기 명령들을 실행하도록 구성되는,

비디오 프로세싱 장치.

청구항 17

제14 항에 있어서,

상기 적어도 하나의 프로세서는 상기 장치로 하여금,

상기 `subpic_treated_as_pic_flag`가 상기 비트스트림에서 시그널링되지 않을 때, 상기 `subpic_treated_as_pic_flag`가 1의 값을 갖는 것으로 결정하는 것; 및

상기 `loop_filter_across_subpic_enabled_flag`가 상기 비트스트림에서 시그널링되지 않을 때, 상기 `loop_filter_across_subpic_enabled_flag`가 0의 값을 갖는 것으로 결정하는 것을 수행하게 하기 위한 상기 명령들을 실행하도록 구성되는,

비디오 프로세싱 장치.

청구항 18

제12 항에 있어서,

상기 적어도 하나의 프로세서는 상기 장치로 하여금,

상기 타겟 서브화상이 상기 화상에서 마지막 서브화상일 때, 상기 타겟 서브화상의 폭 및 높이 중 적어도 하나를 시그널링하는 것을 스킵하는 것을 수행하게 하기 위한 상기 명령들을 실행하도록 구성되는,

비디오 프로세싱 장치.

청구항 19

제18 항에 있어서,

상기 적어도 하나의 프로세서는 상기 장치로 하여금,

상기 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 상기 화상의 폭에서 상기 CTB 크기 단위의 상기 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 포지션을 뺀 것으로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 폭의 값을 결정하거나, 상기 화상의 폭에서 상기 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 포지션을 뺀 것으로서 상기 타겟 서브화상의 폭의 값을 결정하는 것; 및

상기 타겟 서브화상의 높이가 시그널링되지 않을 때, 상기 CTB 크기 단위의 상기 화상의 높이에서 상기 CTB 크기 단위의 상기 타겟 서브화상의 상단 좌측 CTU의 수직 포지션을 뺀 것으로서 상기 CTB 크기 단위의 상기 타겟 서브화상의 높이의 값을 결정하거나, 상기 화상 높이에서 상기 타겟 서브화상의 상단 좌측 CTU의 수직 포지션을 뺀 것으로서 상기 타겟 서브화상의 높이의 값을 결정하는 것을 수행하게 하기 위한 상기 명령들을 실행하도록 구성되는,

비디오 프로세싱 장치.

청구항 20

명령들의 세트를 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체로서,

상기 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 방법을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하고, 상기 방법은,

비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 상기 비트스트림이 서브화상 정보를 포함하는지를 결정하는 단계; 및

상기 비트스트림이 상기 서브화상 정보를 포함하는 것에 대한 응답으로,

화상 내 서브화상들의 수,

타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑,

subpic_treated_as_pic_flag, 및

loop_filter_across_subpic_enabled_flag 중 적어도 하나를 상기 비트스트림에서 시그널링하는 단계를 포함하는,

비-일시적 컴퓨터 판독 가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 개시내용은, 2019년 12월 27일에 출원된 미국 가출원 번호 제62/954,014호를 우선권으로 주장하며, 이 문서는 그 전체가 인용에 의해 본원에 포함된다.

[0002] 본 개시내용은 일반적으로 비디오 프로세싱에 관한 것으로, 보다 구체적으로, 서브화상 파티셔닝 정보를 시그널링하기 위한 방법들 및 장치들에 관한 것이다.

배경 기술

[0003] 비디오는 시각적 정보를 캡처하는 정적 화상들(또는 "프레임들") 세트이다. 저장 메모리 및 송신 대역폭을 감소시키기 위해, 비디오는 저장 또는 송신 전에 압축되고 디스플레이 전에 압축해제될 수 있다. 압축 프로세스는 일반적으로 인코딩으로서 지칭되고 압축해제 프로세스는 일반적으로 디코딩으로서 지칭된다. 가장 일반적인

로 예측, 변환, 양자화, 엔트로피 코딩 및 인-루프 필터링(in-loop filtering)에 기초하는 표준화된 비디오 코딩(coding) 기술들을 사용하는 다양한 비디오 코딩 포맷들이 있다. 특정 비디오 코딩 포맷들을 지정하는 비디오 코딩 표준들 이르면, 고효율 비디오 코딩(High Efficiency Video Coding)(HEVC/H.265) 표준, 다용도 비디오 코딩(Versatile Video Coding)(VVC/H.266) 표준, AVS 표준들이 표준화 기관들에 의해 개발되었다. 점점 더 많은 진보된 비디오 코딩 기술들이 비디오 표준들에서 채택됨에 따라, 새로운 비디오 코딩 표준들의 코딩 효율은 점점 더 높아진다.

발명의 내용

[0004] 일부 실시예들에서, 서브화상 파티셔닝 정보를 시그널링하기 위한 예시적인 방법은: 비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 비트스트림이 서브화상 정보를 포함하는지를 결정하는 단계; 및 비트스트림이 서브화상 정보를 포함하는 것에 대한 응답으로, 화상 내 서브화상들의 수, 타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑, subpic_treated_as_pic_flag, 및 loop_filter_across_subpic_enabled_flag 중 적어도 하나를 비트스트림에서 시그널링하는 단계를 포함한다.

[0005] 일부 실시예들에서, 예시적인 비디오 프로세싱 장치는 명령들을 저장하기 위한 적어도 하나의 메모리 및 적어도 하나의 프로세서를 포함한다. 적어도 하나의 프로세서는 장치로 하여금: 비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 비트스트림이 서브화상 정보를 포함하는지를 결정하는 것; 그리고 비트스트림이 서브화상 정보를 포함하는 것에 대한 응답으로, 화상 내 서브화상들의 수, 타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑, subpic_treated_as_pic_flag, 및 loop_filter_across_subpic_enabled_flag 중 적어도 하나를 비트스트림에서 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성된다.

[0006] 일부 실시예들에서, 예시적인 비-일시적 컴퓨터 판독가능 저장 매체는 명령들의 세트를 저장한다. 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 비트스트림이 서브화상 정보를 포함하는지를 결정하는 것; 그리고 비트스트림이 서브화상 정보를 포함하는 것에 대한 응답으로, 화상 내 서브화상들의 수, 타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑, subpic_treated_as_pic_flag, 및 loop_filter_across_subpic_enabled_flag 중 적어도 하나를 비트스트림에서 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하다.

도면의 간단한 설명

[0007] 본 개시내용의 실시예들 및 다양한 양상들이 하기 상세한 설명 및 첨부 도면들에서 예시된다. 도면들에 보여지는 다양한 특징들은 실체대로 그려진 것은 아니다.

도 1은 본 개시내용의 일부 실시예들에 따른 예시적인 비디오 시퀀스의 구조들을 예시하는 개략도이다.

도 2a는 본 개시내용의 실시예들과 일치하는, 하이브리드 비디오 코딩 시스템의 예시적인 인코딩 프로세스의 개략도를 예시한다.

도 2b는 본 개시내용의 실시예들과 일치하는, 하이브리드 비디오 코딩 시스템의 다른 예시적인 인코딩 프로세스의 개략도를 예시한다.

도 3a는 본 개시내용의 실시예들과 일치하는, 하이브리드 비디오 코딩 시스템의 예시적인 디코딩 프로세스의 개략도를 예시한다.

도 3b는 본 개시내용의 실시예들과 일치하는, 하이브리드 비디오 코딩 시스템의 다른 예시적인 디코딩 프로세스의 개략도를 예시한다.

도 4는 본 개시내용의 일부 실시예들에 따라, 비디오를 인코딩 또는 디코딩하기 위한 예시적인 장치의 블록도를 예시한다.

도 5는 본 개시내용의 일부 실시예들에 따라, 코딩 트리 유닛(CTU)들로 분할된 화상의 예를 예시하는 개략도이다.

도 6은 본 개시내용의 일부 실시예들에 따라, 타일들 및 래스터-스캔 슬라이스들로 파티셔닝된 화상의 예를 예시하는 개략도이다.

도 7은 본 개시내용의 일부 실시예들에 따라, 타일들 및 직사각형 슬라이스들로 파티셔닝된 화상의 예를 예시하

는 개략도이다.

도 8은 본 개시내용의 일부 실시예들에 따라, 타일들 및 직사각형 슬라이스들로 파티셔닝된 화상의 다른 예를 예시하는 개략도이다.

도 9는 본 개시내용의 일부 실시예들에 따라, 서브화상들로 파티셔닝된 화상의 예를 예시하는 개략도이다.

도 10은 본 개시내용의 일부 실시예들에 따라, 서브화상 파티셔닝의 예시적인 시퀀스 파라미터 세트(PPS) 선택스를 도시하는 예시적인 표 1을 예시한다.

도 11은 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 예시적인 PPS 선택스를 도시하는 예시적인 표 2를 예시한다.

도 12는 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 예시적인 화상 파라미터 세트(PPS) 선택스를 도시하는 예시적인 표 3을 예시한다.

도 13은 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 예시적인 화상 헤더(PH) 선택스를 도시하는 예시적인 표 4를 예시한다.

도 14는 본 개시내용의 일부 실시예들에 따라, 예시적인 비트스트림 부합 제약을 예시하는 개략도이다.

도 15는 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 다른 예시적인 PH 선택스를 도시하는 예시적인 표 5를 예시한다.

도 16은 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 다른 예시적인 PH 선택스를 도시하는 예시적인 표 6를 예시한다.

도 17a는 본 개시내용의 일부 실시예들에 따라, 예시적인 PPS 선택스를 도시하는 예시적인 표 7A를 예시한다.

도 17b는 본 개시내용의 일부 실시예들에 따라, 다른 예시적인 PPS 선택스를 도시하는 예시적인 표 7B를 예시한다.

도 18은 본 개시내용의 일부 실시예들에 따라, 다른 예시적인 PPS 선택스를 도시하는 예시적인 표 8를 예시한다.

도 19는 본 개시내용의 일부 실시예들에 따라, 다른 예시적인 PPS 선택스를 도시하는 예시적인 표 9를 예시한다.

도 20은 본 개시내용의 일부 실시예들에 따라, 다른 예시적인 PPS 선택스를 도시하는 예시적인 표 10을 예시한다.

도 21은 본 개시내용의 일부 실시예들에 따른 예시적인 비디오 프로세싱 방법의 흐름도를 예시한다.

도 22는 본 개시내용의 일부 실시예들에 따른 다른 예시적인 비디오 프로세싱 방법의 흐름도를 예시한다.

도 23은 본 개시내용의 일부 실시예들에 따른 다른 예시적인 비디오 프로세싱 방법의 흐름도를 예시한다.

도 24는 본 개시내용의 일부 실시예들에 따른 다른 예시적인 비디오 프로세싱 방법의 흐름도를 예시한다.

도 25는 본 개시내용의 일부 실시예들에 따른 다른 예시적인 비디오 프로세싱 방법의 흐름도를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0008] 이제 예시적인 실시예들이 상세히 참조될 것이며, 그의 예들은 첨부 도면들에 예시된다. 하기 설명은 달리 표현되지 않는 한 상이한 도면들에서 동일한 번호들은 동일하거나 유사한 요소들을 나타내는 첨부 도면들을 참조한다. 예시적인 실시예들의 하기 설명에서 기술된 구현들은 본 발명과 일치하는 모든 구현들을 표현하는 것은 아니다. 대신, 이들은 단지 첨부된 청구항들에서 인용되는 바와 같은 본 발명과 관련된 양상들에 일치하는 장치들 및 방법들의 예들일 뿐이다. 본 개시내용의 특정 양상들은 아래에서 더 상세히 설명된다. 참조로 포함된 용어들 및/또는 정의들과 상충하는 경우, 본원에 제공된 용어들 및 정의들에 따른다.

[0009] ITU-T 비디오 코딩 전문가 그룹(ITU-T VCEG)과 ISO/IEC 동화상 전문가 그룹(ISO/IEC MPEG)의 공동 비디오 전문가 그룹(JVET)은 현재 다용도 비디오 코딩(Versatile Video Coding; VVC/H.266) 표준을 개발하고 있다. VVC 표준은 그의 선행 표준인 고효율 비디오 코딩(HEVC/H.265) 표준의 압축 효율을 배가(doubling)시키는 것을 목표

로 한다. 즉, VVC의 목표는 절반의 대역폭을 사용하여 HEVC/H.265와 동일한 주관적 품질을 달성하는 것이다.

- [0010] 절반의 대역폭을 사용하여 HEVC/H.265와 동일한 주관적 품질을 달성하기 위해, JVET는 공동 탐구 모델(Joint Explosion Model; "JEM") 참조 소프트웨어를 사용하여 HEVC를 뛰어넘은 기술들을 개발하고 있다. 코딩 기술들이 JEM에 통합됨에 따라, JEM은 HEVC보다 실질적으로 더 높은 코딩 성능을 달성했다
- [0011] VVC 표준은 최근에 개발되었으며 더 나은 압축 성능을 제공하는 더 많은 코딩 기술들을 계속 포함하고 있다. VVC는 HEVC, H.264/AVC, MPEG2, H.263 등과 같은 최신 비디오 압축 표준들에 사용된 것과 동일한 하이브리드 비디오 코딩 시스템에 기초한다.
- [0012] 비디오는 시각적 정보를 저장하기 위해 시간적 시퀀스로 배열된 정적 화상들(또는 "프레임들")의 세트이다. 비디오 캡처 디바이스(예컨대, 카메라)가 시간적 시퀀스로 이러한 화상들을 캡처하고 저장하기 위해 사용될 수 있으며, 비디오 재생 디바이스(예컨대, 텔레비전, 컴퓨터, 스마트폰, 태블릿 컴퓨터, 비디오 플레이어 또는 디스플레이 기능이 갖는 임의의 최종-사용자 단말)가 이러한 화상들을 시간적 시퀀스로 디스플레이하는데 사용될 수 있다. 또한, 일부 애플리케이션들에서, 비디오 캡처 디바이스는 감시, 회의 또는 라이브 방송과 같이, 캡처된 비디오를 실시간으로 비디오 재생 디바이스(예컨대, 모니터가 있는 컴퓨터)로 송신할 수 있다.
- [0013] 이러한 애플리케이션들에 의해 필요로 되는 저장 공간 및 송신 대역폭을 감소시키기 위해, 비디오는 저장 및 송신 이전에 압축되고 디스플레이 전에 압축해제될 수 있다. 압축 및 압축해제는 프로세서(예컨대, 일반 컴퓨터의 프로세서) 또는 특수 하드웨어에 의해 실행되는 소프트웨어에 의해 구현될 수 있다. 압축을 위한 모듈은 일반적으로 "인코더"로서 지칭되고, 압축해제를 위한 모듈은 일반적으로 "디코더"로서 지칭된다. 인코더 및 디코더는 집합적으로 "코덱"으로서 지칭될 수 있다. 인코더 및 디코더는 다양한 적합한 하드웨어, 소프트웨어 또는 이들의 조합 중 임의의 것으로서 구현될 수 있다. 예컨대, 인코더 및 디코더의 하드웨어 구현은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들(DSP들), 주문형 집적 회로들(ASIC들), 필드-프로그래밍 가능 게이트 어레이들(FPGA들), 이산 로직, 또는 이들의 임의의 조합과 같은 회로를 포함할 수 있다. 인코더 및 디코더의 소프트웨어 구현은 프로그램 코드들, 컴퓨터 실행-가능 명령들, 펌웨어, 또는 컴퓨터-판독 가능 매체에 고정된 임의의 적합한 컴퓨터-구현 알고리즘 또는 프로세스를 포함할 수 있다. 비디오 압축 및 압축해제는 MPEG-1, MPEG-2, MPEG-4, H.26x 시리즈들 등과 같은 다양한 알고리즘들 또는 표준들에 의해 구현될 수 있다. 일부 애플리케이션들에서, 코덱은 제1 코딩 표준으로부터의 비디오를 압축해제하고 제2 코딩 표준을 사용하여 압축해제된 비디오를 재-압축할 수 있으며, 이 경우 코덱은 "트랜스코더"로서 지칭될 수 있다.
- [0014] 비디오 인코딩 프로세스는 화상을 재구성하는 데 사용될 수 있는 유용한 정보를 식별 및 유지하고 재구성을 위해 중요하지 않은 정보를 무시할 수 있다. 무시되는 중요하지 않은 정보가 완전히 재구성될 수 없는 경우, 이러한 인코딩 프로세스를 "손실성(lossy)"으로서 지칭될 수 있다. 그렇지 않으면, 이는 "무손실성(lossless)"으로서 지칭될 수 있다. 대부분의 인코딩 프로세스들은 손실성이며, 이는 필요한 저장 공간 및 송신 대역폭을 감소시키기 위한 절충안(tradeoff)이다.
- [0015] 다수의 경우들에서, 인코딩되는 화상("현재 화상"으로서 지칭됨)의 유용한 정보는 참조 화상(예컨대, 이전에 인코딩 및 재구성된 화상)에 대한 변화들을 포함할 수 있다. 이러한 변경들은 픽셀들의 포지션 변경들, 광도 변경들 또는 컬러 변경들을 포함될 수 있으며, 그 중 포지션 변경들이 주로 관련된다. 오브젝트를 표현하는 픽셀들의 그룹의 포지션 변경들은 참조 화상과 현재 화상 사이의 오브젝트의 움직임을 반영할 수 있다.
- [0016] 다른 화상을 참조 없이 코딩된 화상(즉, 이는 자체 참조 화상임)은 "I-화상"으로서 지칭된다. 참조 화상으로서 이전 화상을 사용하여 코딩된 화상은 "P-화상"으로서 지칭된다. 참조 화상들로서 이전 화상 및 미래 화상 둘 모두를 사용하여 코딩된 화상(즉, 참조가 "양방향성"임)은 "B-화상"으로서 지칭된다.
- [0017] 도 1은 본 개시내용의 일부 실시예들에 따른 예시적인 비디오 시퀀스(100)의 구조들을 예시한다. 비디오 시퀀스(100)는 라이브 비디오 또는 캡처 및 보관된 비디오일 수 있다. 비디오(100)는 실-생활 비디오, 컴퓨터-생성 비디오(예컨대, 컴퓨터 게임 비디오), 또는 이들의 조합(예컨대, 증강-현실 효과들을 갖는 실-생활 비디오)일 수 있다. 비디오 시퀀스(100)는 비디오 캡처 디바이스(예컨대, 카메라), 이전에 캡처된 비디오를 포함하는 비디오 아카이브(예컨대, 저장 디바이스에 저장된 비디오 파일), 또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스(예컨대, 비디오 브로드캐스트 트랜시버)로부터 입력될 수 있다.
- [0018] 도 1에 도시된 바와 같이, 비디오 시퀀스(100)는, 화상들(102, 104, 106, 108)을 포함하여, 타임라인을 따라 시간적으로 배열된 일련의 화상들을 포함할 수 있다. 화상들(102-106)은 연속적이며, 화상들(106 및 108) 사이에 더 많은 화상들이 있다. 도 1에서, 화상(102)은, 그 참조 화상이 화상(102) 자체인 I-화상이다. 화상(104)은

그 참조 화상이 화살표에 의해 표시된 바와 같이 화상(102)인 P-화상이다. 화상(106)은 그 참조 화상들이 화살표들에 의해 표시된 바와 같이 화상들(104 및 108)인 B-화상이다. 일부 실시예들에서, 화상(예컨대, 화상(104))의 참조 화상은 화상 바로 앞이나 뒤에 있지 않을 수 있다. 예컨대, 화상(104)의 참조 화상은 화상(102) 앞의 화상일 수 있다. 화상들(102-106)의 참조 화상들은 단지 예들일 뿐이고, 본 개시내용은 도 1에 도시된 예시들로 참조 화상들의 실시예들을 제한하지 않는다는 것이 주의되어야 한다.

[0019] 통상적으로, 비디오 코덱들은 이러한 작업들의 계산 복잡성으로 인해 한 번에 전체 화상을 인코딩 또는 디코딩하지 않는다. 오히려, 이들은 화상을 기본 세그먼트들로 분할하고, 세그먼트별로 화상을 인코딩 또는 디코딩할 수 있다. 이러한 기본 세그먼트들은 본 개시내용에서 기본 프로세싱 유닛들(basic processing unit; "BPU들")로서 지칭된다. 예컨대, 도 1의 구조(110)는 비디오 시퀀스(100)의 화상(예컨대, 화상들(102-108) 중 임의의 것)의 예시적인 구조를 도시한다. 구조(110)에서, 화상은 4×4 기본 프로세싱 유닛들로 분할되고, 그 경계들은 파선들로서 도시된다. 일부 실시예들에서, 기본 프로세싱 유닛들은 일부 비디오 코딩 표준들(예컨대, MPEG 패밀리, H.261, H.263, 또는 H.264/AVC)에서 "매크로블록들"로서, 또는 일부 다른 비디오 코딩 표준들(예컨대, H.265/HEVC 또는 H.266/VVC)에서 "코딩 트리 유닛들"(coding tree units; "CTU들")로서 지칭될 수 있다. 기본 프로세싱 유닛들은 화상에서 가변 크기들 이룰때면, 128×128 , 64×64 , 32×32 , 16×16 , 4×8 , 16×32 와 또는 임의의 자의적 형상 및 픽셀 크기들을 가질 수 있다. 기본 프로세싱 유닛들의 크기들 및 형상들은 세부사항의 레벨들 및 코딩 효율의 균형에 기초한 화상이 기본 프로세싱 유닛에서 유지되도록 선택될 수 있다.

[0020] 기본 프로세싱 유닛들은 컴퓨터 메모리에(예컨대, 비디오 프레임 버퍼에) 저장된 상이한 유형들의 비디오 데이터의 그룹을 포함할 수 있는 로직 유닛들일 수 있다. 예컨대, 컬러 화상의 기본 프로세싱 유닛은 무채색 밝기 정보를 표현하는 루마 성분(Y), 컬러 정보를 표현하는 하나 이상의 크로마 성분들(예컨대, Cb 및 Cr) 및 연관된 신텍스 요소들을 포함할 수 있으며, 여기에서 루마 및 크로마 성분들은 기본 프로세싱 유닛과 동일한 크기를 가질 수 있다. 루마 및 크로마 성분들은 일부 비디오 코딩 표준들(예컨대, H.265/HEVC 또는 H.266/VVC)에서 "코딩 트리 블록들(CTB들)"로서 지칭될 수 있다. 기본 프로세싱 유닛에 대해 수행된 모든 동작은 그의 루마 및 크로마 성분들 각각에 대해 반복적으로 수행될 수 있다.

[0021] 비디오 코딩은 여러 동작 스테이지들을 가지며, 그의 예들이 도 2a 및 도 2b 및 도 3a 및 도 3b에 도시된다. 각각의 스테이지에 대해, 기본 프로세싱 유닛들의 크기는 여전히 프로세싱하기에 너무 클 수 있고, 이에 따라 본 개시내용에서 "기본 프로세싱 서브-유닛들"로서 지칭되는 세그먼트들로 추가로 분할될 수 있다. 일부 실시예들에서, 기본 프로세싱 서브-유닛들은 일부 비디오 코딩 표준들(예컨대, MPEG 패밀리, H.261, H.263, 또는 H.264/AVC)에서 "블록들"로서, 또는 일부 다른 비디오 코딩 표준들(예컨대, H.265/HEVC 또는 H.266/VVC)에서 "코딩 유닛들"(coding tree units; "CU들")로서 지칭될 수 있다. 기본 프로세싱 서브-유닛은 기본 프로세싱 유닛과 동일하거나 더 작은 크기를 가질 수 있다. 기본 프로세싱 유닛들과 유사하게, 기본 프로세싱 서브-유닛들은 또한 로직 유닛들이며, 이들은 컴퓨터 메모리(예컨대, 비디오 프레임 버퍼)에 저장된 상이한 유형들의 비디오 데이터(예컨대, Y, Cb, Cr 및 연관된 신텍스 요소들)의 그룹을 포함할 수 있다. 기본 프로세싱 서브-유닛에 대해 수행된 모든 동작은 그의 루마 및 크로마 성분들 각각에 대해 반복적으로 수행될 수 있다. 이러한 분할은 프로세싱 요구들에 의존하여 추가 레벨들로 수행될 수 있다는 것이 주의되어야 한다. 또한, 상이한 스테이지들은 상이한 방식들을 사용하여 기본 프로세싱 유닛들을 분할할 수 있다는 것이 주의되어야 한다.

[0022] 예컨대, 모드 결정 스테이지(그 예가 도 2b에 도시됨)에서, 인코더는 기본 프로세싱 유닛에 대해 어느 예측 모드(예컨대, 화상-내(intra-picture) 예측 또는 화상-간(inter-picture) 예측)를 사용할지를 결정할 수 있으며, 이 기본 프로세싱 유닛은 그러한 결정을 내리기에 너무 클 수 있다. 인코더는 기본 프로세싱 유닛을 여러 기본 프로세싱 서브-유닛들(예컨대, H.265/HEVC 또는 H.266/VVC에서와 같은 CU들)로 분할하고, 개별 기본 프로세싱 서브-유닛 각각에 대한 예측 유형을 결정할 수 있다.

[0023] 다른 예를 들면, 예측 스테이지(그 예가 도 2a 및 도 2b에 도시됨)에서, 인코더가 기본 프로세싱 서브-유닛들(예컨대, CU들)의 레벨에서 예측 동작을 수행할 수 있다. 그러나, 일부 경우들에서, 기본 프로세싱 서브-유닛은 여전히 프로세싱하기에 너무 클 수 있다. 인코더는 기본 프로세싱 서브-유닛을 더 작은 세그먼트들(예컨대, H.265/HEVC 또는 H.266/VVC에서 "예측 블록들" 또는 "PB들"로서 지칭됨)로 추가로 분할할 수 있으며, 그 레벨에서 예측 동작이 수행될 수 있다.

[0024] 다른 예를 들면, 변환 스테이지(그 예는 도 2a 및 도 2b에 도시됨)에서, 인코더는 잔차 기본 프로세싱 서브-유닛들(예컨대, CU들)에 대해 변환 동작을 수행할 수 있다. 그러나, 일부 경우들에서, 기본 프로세싱 서브-유닛은 여전히 프로세싱하기에 너무 클 수 있다. 인코더는 기본 프로세싱 서브-유닛을 더 작은 세그먼트들(예컨대,

H.265/HEVC 또는 H.266/VVC에서 "변환 블록들" 또는 "TB들"로서 지칭됨)로 추가로 분할할 수 있으며, 그 레벨에서 변환 동작이 수행될 수 있다. 동일한 기본 프로세싱 서브-유닛의 분할 방식들은 예측 스테이지 및 변환 스테이지에서 상이할 수 있다는 것이 주의되어야 한다. 예컨대, H.265/HEVC 또는 H.266/VVC에서, 동일한 CU의 예측 블록들 및 변환 블록들은 상이한 크기들 및 개수들을 가질 수 있다.

[0025] 도 1의 구조(110)에서, 기본 프로세싱 유닛(112)은 3×3 기본 프로세싱 서브-유닛들로 추가로 분할되며, 그 경계들은 점선들로서 도시된다. 동일한 화상의 상이한 기본 프로세싱 유닛들은 상이한 방식들의 기본 프로세싱 서브-유닛들로 분할될 수 있다.

[0026] 일부 구현들에서, 비디오 인코딩 및 디코딩에 대한 병렬 프로세싱 및 오류 복원 능력을 제공하기 위해, 화상은 프로세싱을 위한 구역들로 분할될 수 있어서, 화상의 구역에 대해, 인코딩 또는 디코딩 프로세스가 화상의 임의의 다른 구역으로부터의 정보에 의존하지 않을 수 있다. 즉, 화상의 각각의 구역은 독립적으로 프로세싱될 수 있다. 이렇게 함으로써, 코덱은 화상의 상이한 구역들을 병렬로 프로세싱하고, 이에 따라 코딩 효율성을 증가시킬 수 있다. 또한, 프로세싱 시에 구역의 데이터가 손상되거나 네트워크 송신 시에 손실될 때, 코덱은 손상되거나 손실된 데이터를 복원하지 않고 동일한 화상의 다른 구역을 올바르게 인코딩 또는 디코딩할 수 있고, 이에 따라 오류 복원 능력을 제공한다. 일부 비디오 코딩 표준들에서, 화상은 상이한 유형들의 구역들로 분할될 수 있다. 예컨대, H.265/HEVC 및 H.266/VVC는 두 가지 유형들의 구역들: "슬라이스들" 및 "타일들"을 제공한다. 비디오 시퀀스(100)의 상이한 화상들은 화상을 구역들로 분할하기 위한 상이한 파티셔닝 방식들을 가질 수 있다는 것이 또한 주의되어야 한다.

[0027] 예컨대, 도 1에서, 구조(110)는 3개의 구역들(114, 116 및 118)로 분할되고, 그 경계들은 구조(110) 내부에 실선들로서 도시된다. 구역(114)은 4개의 기본 프로세싱 유닛들을 포함한다. 구역들(116 및 118) 각각은 6개의 기본 프로세싱 유닛들을 포함한다. 도 1의 구조(110)의 기본 프로세싱 유닛들, 기본 프로세싱 서브-유닛들 및 구역들은 단지 예들일 뿐이며, 본 개시내용이 그의 실시예들을 제한하지 않는다는 것이 주의되어야 한다.

[0028] 도 2a는 본 개시내용의 실시예들과 일치하는, 예시적인 인코딩 프로세스(200A)의 개략도를 예시한다. 예컨대, 인코딩 프로세스(200A)는 인코더에 의해 수행될 수 있다. 도 2a에 도시된 바와 같이, 인코더는 프로세스(200A)에 따라 비디오 시퀀스(202)를 비디오 비트스트림(228)으로 인코딩할 수 있다. 도 1의 비디오 시퀀스(100)와 유사하게, 비디오 시퀀스(202)는 시간적 순서로 배열된 화상들의 세트("오리지널 화상들"로서 지칭됨)를 포함할 수 있다. 도 1의 구조(110)와 유사하게, 비디오 시퀀스(202)의 각각의 오리지널 화상은 인코더에 의해, 기본 프로세싱 유닛들, 기본 프로세싱 서브-유닛들, 또는 프로세싱을 위한 구역들로 분할될 수 있다. 일부 실시예들에서, 인코더는 비디오 시퀀스(202)의 각각의 오리지널 화상에 대한 기본 프로세싱 유닛들의 레벨에서 프로세스(200A)를 수행할 수 있다. 예컨대, 인코더는 프로세스(200A)를 반복적인 방식으로 수행할 수 있고, 여기서 인코더는 프로세스(200A)의 한 번의 반복에서 기본 프로세싱 유닛을 인코딩할 수 있다. 일부 실시예들에서, 인코더는 비디오 시퀀스(202)의 각각의 오리지널 화상의 구역들(예컨대, 구역들(114 내지 118))에 대해 병렬로 프로세스(200A)를 수행할 수 있다.

[0029] 도 2a에서, 인코더는 비디오 시퀀스(202)의 오리지널 화상의 기본 프로세싱 유닛("오리지널 BPU"로서 지칭됨)을 예측 스테이지(204)로 공급하여 예측 데이터(206) 및 예측된 BPU(208)를 생성할 수 있다. 인코더는 오리지널 BPU로부터 예측된 BPU(208)를 감산하여 잔차 BPU(210)를 생성할 수 있다. 인코더는 잔차 BPU(210)를 변환 스테이지(212) 및 양자화 스테이지(214)로 공급하여 양자화된 변환 계수들(216)을 생성할 수 있다. 인코더는 예측 데이터(206) 및 양자화된 변환 계수들(216)을 바이너리 코딩 스테이지(226)로 공급하여 비디오 비트스트림(228)을 생성할 수 있다. 성분들(202, 204, 206, 208, 210, 212, 214, 216, 226 및 228)은 "순방향 경로"로서 지칭될 수 있다. 프로세스(200A) 동안, 양자화 스테이지(214) 이후에, 인코더는 양자화된 변환 계수들(216)을 역양자화 스테이지(218) 및 역변환 스테이지(220)로 공급하여 재구성된 잔차 BPU(222)를 생성할 수 있다. 인코더는 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 추가하여 예측 참조(224)를 생성할 수 있으며, 이는 프로세스(200A)의 다음 반복을 위해 예측 스테이지(204)에서 사용된다. 프로세스(200A)의 성분들(218, 220, 222 및 224)은 "재구성 경로"로서 지칭될 수 있다. 재구성 경로는 인코더 및 디코더 둘 모두가 예측을 위해 동일한 참조 데이터를 사용하도록 보장하는 데 사용될 수 있다.

[0030] 인코더는 (순방향 경로에서) 오리지널 화상의 각각의 오리지널 BPU를 인코딩하고 (재구성 경로에서) 오리지널 화상의 다음 오리지널 BPU를 인코딩하기 위한 예측된 참조(224)를 생성하기 위해 프로세스(200A)를 반복적으로 수행할 수 있다. 오리지널 화상의 모든 오리지널 BPU들을 인코딩한 후, 인코더는 비디오 시퀀스(202) 내 다음 화상을 인코딩하는 것으로 진행할 수 있다.

- [0031] 프로세스(200A)를 참조하면, 인코더는 비디오 캡처 디바이스(예컨대, 카메라)에 의해 생성된 비디오 시퀀스(202)를 수신할 수 있다. 본원에서 사용되는 "수신"이란 용어는 수신, 입력, 취득, 리트리브, 획득, 판독, 액세스 또는 데이터를 입력하기 위한 임의의 방식의 임의의 액션을 지칭할 수 있다.
- [0032] 예측 스테이지(204)에서, 현재 반복에서 인코더는 오리지널 BPU 및 예측 참조(224)를 수신하고 예측 동작을 수행하여 예측 데이터(206) 및 예측된 BPU(208)를 생성할 수 있다. 예측 참조(224)는 프로세스(200A)의 이전 반복의 재구성 경로로부터 생성될 수 있다. 예측 스테이지(204)의 목적은, 예측 참조(224) 및 예측 데이터(206)로부터의 예측된 BPU(208)로서 오리지널 BPU를 재구성하는데 사용될 수 있는 예측 데이터(206)를 추출함으로써 정보 중복성을 감소시키는 것이다.
- [0033] 이상적으로, 예측된 BPU(208)는 오리지널 BPU와 동일할 수 있다. 그러나, 비-이상적 예측 및 재구성 동작들로 인해, 예측된 BPU(208)는 일반적으로 오리지널 BPU와 약간 상이하다. 그러한 차이들을 레코딩하기 위해, 예측된 BPU(208)를 생성한 후, 인코더는 오리지널 BPU로부터 이 차이를 감산하여 잔차 BPU(210)를 생성할 수 있다. 예컨대, 인코더는 예측된 BPU(208)의 픽셀들의 값들(예컨대, 회색조 값들 또는 RGB 값들)을 오리지널 BPU의 대응하는 픽셀들의 값들로부터 감산할 수 있다. 잔차 BPU(210)의 각각의 픽셀은 예측된 BPU(208)와 오리지널 BPU의 대응하는 픽셀들 사이의 이러한 감산의 결과로서 잔차 값을 가질 수 있다. 오리지널 BPU와 비교하여, 예측 데이터(206) 및 잔차 BPU(210)는 더 적은 비트들을 가질 수 있지만, 현저한 품질 악화 없이 오리지널 BPU를 재구성하는 데 사용될 수 있다. 따라서, 오리지널 BPU가 압축된다.
- [0034] 잔차 BPU(210)를 추가로 압축하기 위해, 변환 스테이지(212)에서, 인코더는 잔차 BPU(210)를 2-차원 "기본 패턴들"의 세트에 분해함으로써 그의 공간적 중복성을 감소시킬 수 있으며, 각각의 기본 패턴은 "변환 계수"와 연관된다. 기본 패턴들은 동일한 크기(예컨대, 잔차 BPU(210)의 크기)를 가질 수 있다. 각각의 기본 패턴은 잔차 BPU(210)의 변동 주파수(예컨대, 밝기 변동 주파수) 성분을 표현할 수 있다. 기본 패턴들 중 어느 것도 임의의 다른 기본 패턴들의 임의의 조합들(예컨대, 선형 조합들)로부터 재생될 수 없다. 즉, 분해는 잔차 BPU(210)의 변동들을 주파수 도메인으로 분해할 수 있다. 이러한 분해는 함수의 이산 푸리에 변환과 유사하며, 여기서 기본 패턴이 이산 푸리에 변환의 기본 함수들(예컨대, 삼각 함수들)과 유사하고, 변환 계수들은 기본 함수들과 연관된 계수들과 유사하다.
- [0035] 상이한 변환 알고리즘들은 상이한 기본 패턴들을 사용할 수 있다. 예컨대, 이산 코사인 변환, 이산 사인 변환 등과 같은 다양한 변환 알고리즘들이 변환 스테이지(212)에서 사용될 수 있다. 변환 스테이지(212)에서의 변환은 가역적이다. 즉, 인코더는 변환의 역동작("역변환"으로서 지칭됨)에 의해 잔차 BPU(210)를 복구할 수 있다. 예컨대, 잔차 BPU(210)의 픽셀을 복구하기 위해, 역변환은 기본 패턴들의 대응하는 픽셀들의 값들을 개개의 연관된 계수들로 곱하고 그 결과들을 더하여 가중합을 생성하는 것일 수 있다. 비디오 코딩 표준의 경우, 인코더 및 디코더 둘 모두는 동일한 변환 알고리즘(따라서 동일한 기본 패턴들)을 사용할 수 있다. 따라서, 인코더는, 인코더로부터 기본 패턴들을 수신하지 않고 디코더가 잔차 BPU(210)를 재구성할 수 있는, 변환 계수들만을 레코딩할 수 있다. 잔차 BPU(210)와 비교하여, 변환 계수들은 더 적은 비트들을 가질 수 있지만, 현저한 품질 악화 없이 잔차 BPU(210)를 재구성하는 데 사용될 수 있다. 따라서, 잔차 BPU(210)는 추가로 압축된다.
- [0036] 인코더는 양자화 스테이지(214)에서 변환 계수들을 추가로 압축할 수 있다. 변환 프로세스에서, 상이한 기본 패턴들은 상이한 변동 주파수들(예컨대, 밝기 변동 주파수들)을 표현할 수 있다. 인간의 눈들은 보통 저-주파수 변동을 더 잘 인식하기 때문에, 인코더는 디코딩 시에 현저한 품질 악화를 야기함 없이 고-주파수 변동에 관한 정보를 목살할 수 있다. 예컨대, 양자화 스테이지(214)에서, 인코더는 각각의 변환 계수를 정수 값("양자화 파라미터"로서 지칭됨)으로 나누고 몫을 가장 가까운 정수로 반올림함으로써 양자화된 변환 계수들(216)을 생성할 수 있다. 이러한 동작 후에, 고-주파수 기본 패턴들의 일부 변환 계수들은 0으로 컨버팅될 수 있고, 저-주파수 기본 패턴들의 변환 계수들은 더 작은 정수들로 컨버팅될 수 있다. 인코더는 0-값 양자화된 변환 계수들(216) 이에 의해 변환 계수들이 추가로 압축됨을 목살할 수 있다. 양자화 프로세스는 또한 가역적이며, 여기서 양자화된 변환 계수들(216)은 양자화의 역동작("역양자화"로서 지칭됨)에서 변환 계수들로 재구성될 수 있다.
- [0037] 인코더가 반올림 동작에서 이러한 나눗셈들의 나머지를 목살하기 때문에, 양자화 스테이지(214)는 손실성일 수 있다. 통상적으로, 양자화 스테이지(214)는 프로세스(200A)에서 가장 많은 정보 손실에 기여할 수 있다. 정보 손실이 클수록, 양자화된 변환 계수들(216)이 필요로 할 수 있는 비트들이 더 적다. 정보 손실의 상이한 레벨들을 획득하기 위해, 인코더는 양자화 파라미터의 상이한 값들 또는 양자화 프로세스의 임의의 다른 파라미터를 사용할 수 있다.
- [0038] 바이너리 코딩 스테이지(226)에서, 인코더는 예컨대, 엔트로피 코딩, 가변 길이 코딩, 산술 코딩, 호프만

(Huffman) 코딩, 맥락-적응 바이너리 산술 코딩, 또는 임의의 다른 무손실성 또는 손실성 압축 알고리즘과 같은 바이너리 코딩 기술을 사용하여 예측 데이터(206) 및 양자화된 변환 계수들(216)을 인코딩할 수 있다. 일부 실시예들에서, 예측 데이터(206) 및 양자화된 변환 계수들(216) 외에도, 인코더는 예컨대, 예측 스테이지(204)에서 사용되는 예측 모드, 예측 동작의 파라미터, 변환 스테이지(212)에서의 변환 유형, 양자화 프로세스의 파라미터들(예컨대, 양자화 파라미터들), 인코더 제어 파라미터(예컨대, 비트레이트 제어 파라미터) 등과 같은 다른 정보를 바이너리 코딩 스테이지(226)에서 인코딩할 수 있다. 인코더는 바이너리 코딩 스테이지(226)의 출력 데이터를 사용하여 비디오 비트스트림(228)을 생성할 수 있다. 일부 실시예들에서, 비디오 비트스트림(228)은 추가로 네트워크 송신을 위해 패킷화될 수 있다.

[0039] 프로세스(200A)의 재구성 경로를 참조하면, 역양자화 스테이지(218)에서, 인코더는 양자화된 변환 계수들(216)에 대해 역양자화를 수행하여 재구성된 변환 계수들을 생성할 수 있다. 역변환 스테이지(220)에서, 인코더는 재구성된 변환 계수들에 기초하여 재구성된 잔차 BPU(222)를 생성할 수 있다. 인코더는 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 추가하여 프로세스(200A)의 다음 반복에서 사용될 예측 참조(224)를 생성할 수 있다.

[0040] 프로세스(200A)의 다른 변형들이 비디오 시퀀스(202)를 인코딩하는 데 사용될 수 있다는 것이 주의되어야 한다. 일부 실시예들에서, 프로세스(200A)의 스테이지들은 인코더에 의해 상이한 순서들로 수행될 수 있다. 일부 실시예들에서, 프로세스(200A)의 하나 이상의 스테이지들은 단일 스테이지로 결합될 수 있다. 일부 실시예들에서, 프로세스(200A)의 단일 스테이지는 다수의 스테이지들로 분할될 수 있다. 예컨대, 변환 스테이지(212) 및 양자화 스테이지(214)는 단일 스테이지로 결합될 수 있다. 일부 실시예들에서, 프로세스(200A)는 부가적인 스테이지들을 포함할 수 있다. 일부 실시예들에서, 프로세스(200A)는 도 2a의 하나 이상의 스테이지들을 생략할 수 있다.

[0041] 도 2b는 본 개시내용의 실시예들과 일치하는, 다른 예시적인 인코딩 프로세스(200B)의 개략도를 예시한다. 프로세스(200B)는 프로세스(200A)로부터 수정될 수 있다. 예컨대, 프로세스(200B)는 하이브리드 비디오 코딩 표준(예컨대, H.26x 시리즈들)에 부합하는 인코더에 의해 사용될 수 있다. 프로세스(200A)와 비교하여, 프로세스(200B)의 순방향 경로는 부가적으로 모드 결정 스테이지(230)를 포함하고 예측 스테이지(204)를 공간적 예측 스테이지(2042) 및 시간적 예측 스테이지(2044)로 분할한다. 프로세스(200B)의 재구성 경로는 부가적으로 루프 필터 스테이지(232) 및 버퍼(234)를 포함한다.

[0042] 일반적으로, 예측 기술들은 두 가지 유형들: 공간적 예측과 시간적 예측으로 카테고리화될 수 있다. 공간적 예측(예컨대, 화상-내 예측 또는 "인트라 예측(intra prediction)")은 현재 BPU를 예측하기 위해 동일한 화상 내 하나 이상의 이미 코딩된 이웃 BPU들로부터의 픽셀들을 사용할 수 있다. 즉, 공간적 예측에서의 예측 참조(224)는 이웃 BPU들을 포함할 수 있다. 공간적 예측은 화상의 고유한 공간 중복성을 감소시킬 수 있다. 시간적 예측(예컨대, 화상-간 예측 또는 "인터 예측(inter prediction)")은 현재 BPU를 예측하기 위해 하나 이상의 이미 코딩된 화상들로부터의 구역들을 사용할 수 있다. 즉, 시간적 예측에서의 예측 참조(224)는 코딩된 화상들을 포함할 수 있다. 시간적 예측은 화상들의 고유한 시간 중복성을 감소시킬 수 있다.

[0043] 프로세스(200B)를 참조하면, 순방향 경로에서, 인코더는 공간적 예측 스테이지(2042) 및 시간적 예측 스테이지(2044)에서 예측 동작을 수행한다. 예컨대, 공간적 예측 스테이지(2042)에서, 인코더는 인트라 예측을 수행할 수 있다. 인코딩되는 화상의 오리지널 BPU에 대해, 예측 참조(224)는 동일한 화상에서 (순방향 경로에서) 인코딩되고 (재구성된 경로에서) 재구성된 하나 이상의 이웃 BPU들을 포함할 수 있다. 인코더는 이웃 BPU들을 외삽함으로써 예측된 BPU(208)를 생성할 수 있다. 외삽 기술들은, 예컨대, 선형 외삽 또는 내삽, 다항식 외삽 또는 내삽 등을 포함할 수 있다. 일부 실시예들에서, 인코더는 이를테면, 예측된 BPU(208)의 각각의 픽셀에 대한 대응하는 픽셀들의 값들을 외삽함으로써, 픽셀 레벨에서 외삽을 수행할 수 있다. 외삽에 대해 사용되는 이웃 BPU들은 오리지널 BPU에 대해, 다양한 방향들로부터 이를테면, 수직 방향(예컨대, 오리지널 BPU의 상단), 수평 방향(예컨대, 오리지널 BPU의 좌측), 대각선 방향(예컨대, 오리지널 BPU의 좌측-아래, 우측-아래, 좌측-위 또는 우측-위) 또는 사용된 비디오 코딩 표준에 정의된 임의의 방향으로 로케이팅될 수 있다. 인트라 예측을 위해, 예측 데이터(206)는, 예컨대, 사용된 이웃 BPU들의 로케이션들(예컨대, 좌표들), 사용된 이웃 BPU들의 크기들, 외삽의 파라미터들, 오리지널 BPU에 대한 사용된 이웃 BPU들의 방향 등을 포함할 수 있다.

[0044] 다른 예를 들면, 시간적 예측 스테이지(2044)에서, 인코더는 인터 예측을 수행할 수 있다. 현재 화상의 오리지널 BPU에 대해, 예측 참조(224)는 (순방향 경로에서) 인코딩되고 (재구성된 경로에서) 재구성된 하나 이상의 화상들("참조 화상들"로서 지칭됨)을 포함할 수 있다. 일부 실시예들에서, 참조 화상은 BPU별로 인코딩 및 재구

성될 수 있다. 예컨대, 인코더는 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 추가하여 재구성된 BPU를 생성할 수 있다. 동일한 화상의 모든 재구성된 BPU들이 생성될 때, 인코더는 참조 화상으로서 재구성된 화상을 생성할 수 있다. 인코더는 참조 화상의 범위("검색 창"으로 지칭됨)에서 매칭 구역을 검색하기 위해 "모션 추정(motion estimation)"의 동작을 수행할 수 있다. 참조 화상에서 검색 창의 로케이션은 현재 화상에서 오리지널 BPU의 로케이션에 기초하여 결정될 수 있다. 예컨대, 검색 창은 참조 화상에서 현재 화상의 오리지널 BPU와 동일한 좌표들을 갖는 로케이션에 중심을 둘 수 있고 미리 결정된 거리만큼 확장될 수 있다. 인코더가 검색 창에서 오리지널 BPU와 유사한 구역을 (예컨대, PRA(pel-recursive algorithm), 블록-매칭 알고리즘 등을 사용함으로써) 식별할 때, 인코더는 그러한 구역을 매칭 구역으로서 결정할 수 있다. 매칭 구역은 오리지널 BPU와 상이한 디멘션들(예컨대, 더 작거나, 동일하거나, 더 크거나, 상이한 형상)을 가질 수 있다. 참조 화상 및 현재 화상은 (예컨대, 도 1에 도시된 바와 같이) 타임라인에서 시간적으로 분리되기 때문에, 시간이 지남에 따라 매칭 구역이 오리지널 BPU의 로케이션으로 "이동한다"고 간주될 수 있다. 인코더는 "모션 벡터"로서 이러한 모션의 방향 및 거리를 레코딩할 수 있다. (예컨대, 도 1의 화상(106)과 같이) 다수의 참조 화상들이 사용될 때, 인코더는 매칭 구역을 검색하고 각각의 참조 화상에 대한 연관된 모션 벡터를 결정할 수 있다. 일부 실시예들에서, 인코더는 개개의 매칭 참조 화상들의 매칭 구역들의 픽셀 값들에 가중치들을 할당할 수 있다.

[0045] 모션 추정은, 예컨대, 병진운동들, 회전들, 주밍(zooming) 등과 같은 다양한 유형들의 모션들을 식별하는 데 사용될 수 있다. 인터 예측에 대해, 예측 데이터(206)는, 예컨대, 매칭 구역의 로케이션들(예컨대, 좌표들), 매칭 구역과 연관된 모션 벡터들, 참조 화상들의 수, 참조 화상들과 연관된 가중치들 등을 포함할 수 있다.

[0046] 예측된 BPU(208)를 생성하기 위해, 인코더는 "모션 보상"의 동작을 수행할 수 있다. 모션 보상은 예측 데이터(206)(예컨대, 모션 벡터) 및 예측 참조(224)에 기초하여 예측된 BPU(208)를 재구성하는데 사용될 수 있다. 예컨대, 인코더는, 인코더가 현재 화상의 오리지널 BPU를 예측할 수 있는 모션 벡터에 따라 참조 화상의 매칭 구역을 이동할 수 있다. (예컨대, 도 1의 화상(106)과 같이) 다수의 참조 화상들이 사용될 때, 인코더는 매칭 구역들의 평균 픽셀 값들 및 개개의 모션 벡터들에 따라 참조 화상들의 매칭 구역들을 이동할 수 있다. 일부 실시예들에서, 인코더가 개개의 매칭 참조 화상들의 매칭 구역들의 픽셀 값들에 가중치들을 할당한 경우, 인코더는 이동된 매칭 구역들의 픽셀 값들의 가중합을 추가할 수 있다.

[0047] 일부 실시예들에서, 인터 예측은 단방향 또는 양방향일 수 있다. 단방향 인터 예측들은 현재 화상에 대해 동일한 시간 방향으로 하나 이상의 참조 화상들을 사용할 수 있다. 예컨대, 도 1의 화상(104)은 참조 화상(즉, 화상(102))이 화상(104)에 선행하는 단방향 인터 예측 화상이다. 양방향 인터 예측들은 현재 화상에 대해 양쪽 시간 방향들에서 하나 이상의 참조 화상들을 사용할 수 있다. 예컨대, 도 1의 화상(106)은 참조 화상들(즉, 화상들(104 및 108))이 화상(104)에 대해 양쪽 시간 방향들에 있는 양방향 인터 예측 화상이다.

[0048] 여전히 프로세스(200B)의 순방향 경로를 참조하면, 공간적 예측(2042) 및 시간적 예측 스테이지(2044) 이후에, 모드 판단 스테이지(230)에서, 인코더는 프로세스(200B)의 현재 반복에 대해 예측 모드(예컨대, 인트라 예측 또는 인터 예측 중 하나)를 선택할 수 있다. 예컨대, 인코더는 레이트-왜곡 최적화 기술을 수행할 수 있는데, 여기에서 인코더는 후보 예측 모드의 비트레이트 및 후보 예측 모드 하의 재구성된 참조 화상의 왜곡에 의존하여 비용 함수의 값을 최소화하도록 예측 모드를 선택할 수 있다. 선택된 예측 모드에 의존하여, 인코더는 대응하는 예측된 BPU(208) 및 예측된 데이터(206)를 생성할 수 있다.

[0049] 프로세스(200B)의 재구성 경로에서, 순방향 경로에서 인트라 예측 모드가 선택된 경우, 예측 참조(224)(예컨대, 현재 화상에서 인코딩 및 재구성된 현재 BPU)를 생성한 후, 인코더는 추후 사용을 위해(예컨대, 현재 화상의 다음 BPU의 외삽을 위해) 예측 참조(224)를 공간적 예측 스테이지(2042)로 직접 공급할 수 있다. 순방향 경로에서 인터 예측 모드가 선택된 경우, 예측 참조(224)(예컨대, 모든 BPU들이 인코딩 및 재구성된 현재 화상)를 생성한 후, 인코더는 예측 참조(224)를 루프 필터 스테이지(232)로 공급할 수 있으며, 여기에서 인코더는 예측 참조(224)에 루프 필터를 적용하여 인터 예측에 의해 도입된 왜곡(예컨대, 블로킹 아티팩트(blocking artifact)들을 감소시키거나 제거할 수 있다. 인코더는 루프 필터 스테이지(232)에서, 예컨대, 디블로킹(deblocking), 샘플 적응적 오프셋들, 적응적 루프 필터들 등과 같은 다양한 루프 필터 기술들을 적용할 수 있다. 루프-필터링된 참조 화상은 후 사용을 위해(예컨대, 비디오 시퀀스(202)의 미래 화상에 대한 인터-예측 참조 화상으로서 사용되도록) 버퍼(234)(또는 "디코딩된 화상 버퍼")에 저장될 수 있다. 인코더는 시간적 예측 스테이지(2044)에서 사용되도록 버퍼(234)에 하나 이상의 참조 화상들을 저장할 수 있다. 일부 실시예들에서, 인코더는 양자화된 변환 계수들(216), 예측 데이터(206) 및 다른 정보와 함께, 바이너리 코딩 스테이지(226)에서 루프 필터의 파라미터들(예컨대, 루프 필터 강도)을 인코딩할 수 있다.

- [0050] 도 3a는 본 개시내용의 실시예들과 일치하는, 예시적인 디코딩 프로세스(300A)의 개략도를 예시한다. 프로세스(300A)는 도 2a의 압축 프로세스(200A)에 대응하는 압축해제 프로세스일 수 있다. 일부 실시예들에서, 프로세스(300A)는 프로세스(200A)의 재구성 경로와 유사할 수 있다. 디코더는 프로세스(300A)에 따라 비디오 비트스트림(228)을 비디오 스트림(304)으로 디코딩할 수 있다. 비디오 스트림(304)은 비디오 시퀀스(202)와 매우 유사할 수 있다. 그러나, 압축 및 압축해제 프로세스(예컨대, 도 2a 및 도 2b의 양자화 스테이지(214))에서의 정보 손실로 인해, 보통, 비디오 스트림(304)은 비디오 시퀀스와 동일하지 않다. 도 2a 및 도 2b의 프로세스들(200A 및 200B)과 유사하게, 디코더는 비디오 비트스트림(228)에 인코딩된 각각의 화상에 대해 기본 프로세싱 유닛들(BPU들)의 레벨에서 프로세스(300A)를 수행할 수 있다. 예컨대, 디코더는 프로세스(300A)를 반복적인 방식으로 수행할 수 있고, 여기서 디코더는 프로세스(300A)의 한번의 반복에서 기본 프로세싱 유닛을 디코딩할 수 있다. 일부 실시예들에서, 디코더는 비디오 비트스트림(228)에 인코딩된 각각의 화상의 구역들(예컨대, 구역들(114-118))에 대해 병렬로 프로세스(300A)를 수행할 수 있다.
- [0051] 도 3a에서, 디코더는 인코딩된 화상의 기본 프로세싱 유닛("인코딩된 BPU"로서 지칭됨)과 연관된 비디오 비트스트림(228)의 부분을 바이너리 디코딩 스테이지(302)로 공급할 수 있다. 바이너리 디코딩 스테이지(302)에서, 디코더는 이 부분을 예측 데이터(206) 및 양자화된 변환 계수들(216)로 디코딩할 수 있다. 디코더는 양자화된 변환 계수들(216)을 역양자화 스테이지(218) 및 역변환 스테이지(220)로 공급하여 재구성된 잔차 BPU(222)를 생성할 수 있다. 디코더는 예측 데이터(206)를 예측 스테이지(204)로 공급하여 예측된 BPU(208)를 생성할 수 있다. 디코더는 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 추가하여 예측된 참조(224)를 생성할 수 있다. 일부 실시예들에서, 예측된 참조(224)는 버퍼(예컨대, 컴퓨터 메모리의 디코딩된 화상 버퍼)에 저장될 수 있다. 디코더는 프로세스(300A)의 다음 반복에서 예측 동작을 수행하기 위해 예측된 참조(224)를 예측 스테이지(204)로 공급할 수 있다.
- [0052] 디코더는 인코딩된 화상의 각각의 인코딩된 BPU를 디코딩하고 인코딩된 화상의 다음 인코딩된 BPU를 인코딩하기 위한 예측된 참조(224)를 생성하기 위해 프로세스(300A)를 반복적으로 수행할 수 있다. 인코딩된 화상의 모든 인코딩된 BPU들을 디코딩한 후, 디코더는 디스플레이를 위해 화상을 비디오 스트림(304)으로 출력하고 비디오 비트스트림(228)에서 다음 인코딩된 화상을 디코딩하도록 진행할 수 있다.
- [0053] 바이너리 디코딩 스테이지(302)에서, 디코더는 인코더에 의해 사용되는 바이너리 코딩 기술(예컨대, 엔트로피 코딩, 가변 길이 코딩, 산술 코딩, 호프만 코딩, 맥락-적용 바이너리 산술 코딩, 또는 임의의 다른 무손실 압축 알고리즘)의 역 동작을 수행할 수 있다. 일부 실시예들에서, 예측 데이터(206) 및 양자화된 변환 계수들(216) 외에도, 디코더는, 예컨대, 예측 모드, 예측 동작의 파라미터들, 변환 유형, 양자화 프로세스의 파라미터들(예컨대, 양자화 파라미터들), 인코더 제어 파라미터(예컨대, 비트레이트 제어 파라미터) 등과 같은 다른 정보를 바이너리 디코딩 스테이지(302)에서 디코딩할 수 있다. 일부 실시예들에서, 비디오 비트스트림(228)이 네트워크를 통해 패킷들에서 송신되는 경우, 디코더는 비디오 비트스트림(228)을 바이너리 디코딩 스테이지(302)로 공급하기 전에 패킷해제(depacketize)할 수 있다.
- [0054] 도 3b는 본 개시내용의 실시예들과 일치하는, 다른 예시적인 디코딩 프로세스(300B)의 개략도를 예시한다. 프로세스(300B)는 프로세스(300A)로부터 수정될 수 있다. 예컨대, 프로세스(300B)는 하이브리드 비디오 코딩 표준(예컨대, H.26x 시리즈들)에 부합하는 디코더에 의해 사용될 수 있다. 프로세스(300A)와 비교하여, 프로세스(300B)는 부가적으로 예측 스테이지(204)를 공간적 예측 스테이지(2042) 및 시간적 예측 스테이지(2044)로 분할하고, 루프 필터 스테이지(232) 및 버퍼(234)를 부가적으로 포함한다.
- [0055] 프로세스(300B)에서, 디코딩 되고있는 인코딩된 화상("현재 화상"으로서 지칭됨)의 인코딩된 기본 프로세싱 유닛("현재 BPU"로서 지칭됨)에 대해, 디코더에 의해 바이너리 디코딩 스테이지(302)로부터 디코딩된 예측 데이터(206)는, 인코더에 의해 현재 BPU를 인코딩하기 위해 어떤 예측 모드가 사용되었는지에 의존하여, 다양한 유형들의 데이터를 포함할 수 있다. 예컨대, 현재 BPU를 인코딩하기 위해 인코더에 의해 인트라 예측이 사용된 경우, 예측 데이터(206)는 인트라 예측을 표시하는 예측 모드 표시자(예컨대, 플래그 값), 인트라 예측 동작의 파라미터들 등을 포함할 수 있다. 인트라 예측 동작의 파라미터들은, 예컨대, 참조로서 사용되는 하나 이상의 이웃 BPU들의 로케이션들(예컨대, 좌표들), 이웃 BPU들의 크기들, 외삽의 파라미터들, 오리지널 BPU에 대한 이웃 BPU들의 방향 등을 포함할 수 있다. 다른 예를 들면, 현재 BPU를 인코딩하기 위해 인코더에 의해 인터 예측이 사용된 경우, 예측 데이터(206)는 인터 예측을 표시하는 예측 모드 표시자(예컨대, 플래그 값), 인터 예측 동작의 파라미터들 등을 포함할 수 있다. 인터 예측 동작의 파라미터들은, 예컨대, 현재 BPU와 연관된 참조 화상들의 수, 참조 화상들과 각자 연관된 가중치들, 개개의 참조 화상들 내 하나 이상의 매칭 구역들의 로케이션들(예

컨대, 좌표들), 매칭 구역들과 각자 연관된 하나 이상의 모션 벡터들 등을 포함할 수 있다.

[0056] 예측 모드 표시자에 기초하여, 디코더는 공간적 예측 스테이지(2042)에서 공간적 예측(예컨대, 인트라 예측)을 수행할지 또는 시간적 예측 스테이지(2044)에서 시간적 예측(예컨대, 인터 예측)을 수행할지를 결정할 수 있다. 그러한 공간적 예측 또는 시간적 예측을 수행하는 것에 대한 세부사항들은 도 2b에서 설명되며 이후에 반복되지 않을 것이다. 그러한 공간적 예측 또는 시간적 예측을 수행한 후, 디코더는 예측된 BPU(208)를 생성할 수 있다. 도 3a에 설명된 바와 같이, 디코더는 예측된 BPU(208) 및 재구성된 잔차 BPU(222)를 더하여 예측 참조(224)를 생성할 수 있다.

[0057] 프로세스(300B)에서, 디코더는 프로세스(300B)의 다음 반복에서 예측 동작을 수행하기 위해 예측된 참조(224)를 공간적 예측 스테이지(2042) 또는 시간적 예측 스테이지(2044)에 공급할 수 있다. 예컨대, 현재 BPU가 공간적 예측 스테이지(2042)에서 인트라 예측을 사용하여 디코딩되는 경우, 예측 참조(224)(예컨대, 디코딩된 현재 BPU)를 생성한 후에, 디코더는 추후 사용을 위해(예컨대, 현재 화상의 다음 BPU의 외삽을 위해) 예측 참조(224)를 공간적 예측 스테이지(2042)로 직접 공급할 수 있다. 현재 BPU가 시간적 예측 스테이지(2044)에서 인터 예측을 사용하여 디코딩되는 경우, 예측 참조(224)(예컨대, 모든 BPU들이 디코딩된 참조 화상)를 생성한 후, 인코더는 예측 참조(224)를 루프 필터 스테이지(232)로 공급하여 왜곡(예컨대, 블로킹 아티팩트들)을 감소시키거나 제거할 수 있다. 디코더는 도 2b에 설명된 바와 같은 방식으로 루프 필터를 예측 참조(224)에 적용할 수 있다. 루프-필터링된 참조 화상은 추후 사용을 위해(예컨대, 비디오 비트스트림(228)의 미래 인코딩된 화상에 대한 인터-예측 참조 화상으로서 사용되도록) 버퍼(234)(예컨대, 컴퓨터 메모리의 디코딩된 화상 버퍼)에 저장될 수 있다. 디코더는 시간적 예측 스테이지(2044)에서 사용되도록 버퍼(234)에 하나 이상의 참조 화상들을 저장할 수 있다. 일부 실시예들에서, 현재 BPU를 인코딩하기 위해 인터 예측이 사용되었음을 예측 데이터(206)의 예측 모드 표시자가 표시할 때, 예측 데이터는 루프 필터의 파라미터들(예컨대, 루프 필터 강도)을 더 포함할 수 있다.

[0058] 도 4는 본 개시내용의 실시예들과 일치하는, 비디오를 인코딩 또는 디코딩하기 위한 예시적인 장치(400)의 블록도를 예시한다. 도 4에 도시된 바와 같이, 장치(400)는 프로세서(402)를 포함할 수 있다. 프로세서(402)가 본원에 설명된 명령들을 실행할 때, 장치(400)는 비디오 인코딩 또는 디코딩을 위한 특수 기계가 될 수 있다. 프로세서(402)는 정보를 조작하거나 프로세싱할 수 있는 임의의 유형의 회로일 수 있다. 예컨대, 프로세서(402)는 중앙 처리 장치(또는 "CPU"), 그래픽 프로세싱 장치(또는 "GPU"), 뉴럴 프로세싱 유닛("NPU"), 마이크로제어기 유닛("MCU"), 광학 프로세서, 프로그래밍 가능 로직 제어기, 마이크로제어기, 마이크로프로세서, 디지털 신호 프로세서, 지적 재산(IP) 코어, 프로그래밍 가능 로직 어레이(PLA), 프로그래밍 가능 어레이 로직(PAL), 일반 어레이 로직(GAL), 복합 프로그래밍 가능 로직 디바이스(CPLD), 필드-프로그래밍 가능 게이트 어레이(FPGA), 시스템 온 시스템(SoC), 주문형 집적 회로(ASIC) 등의 임의의 수의 임의의 조합을 포함할 수 있다. 일부 실시예들에서, 프로세서(402)는 또한 단일 로직 구성요소로서 그룹화된 프로세서들의 세트일 수 있다. 예컨대, 도 4에 도시된 바와 같이, 프로세서(402)는 프로세서(402a), 프로세서(402b) 및 프로세서(402n)를 포함하는 다수의 프로세서들을 포함할 수 있다.

[0059] 장치(400)는 또한 데이터(예컨대, 명령들의 세트, 컴퓨터 코드들, 중간 데이터 등)를 저장하도록 구성된 메모리(404)를 포함할 수 있다. 예컨대, 도 4에 도시된 바와 같이, 저장된 데이터는 프로그램 명령들(예컨대, 프로세스들(200A, 200B, 300A 또는 300B)의 스테이지들을 구현하기 위한 프로그램 명령들) 및 프로세싱을 위한 데이터(예컨대, 비디오 시퀀스(202), 비디오 비트스트림(228) 또는 비디오 스트림(304))을 포함할 수 있다. 프로세서(402)는 (예컨대, 버스(410)를 통해) 프로그램 명령들 및 프로세싱을 위한 데이터에 액세스할 수 있고, 프로그램 명령들을 실행하여 프로세싱을 위한 데이터에 대한 동작 또는 조작을 수행할 수 있다. 메모리(404)는 고속 랜덤-액세스 저장 디바이스 또는 비-휘발성 저장 디바이스를 포함할 수 있다. 일부 실시예들에서, 메모리(404)는 랜덤-액세스 메모리(RAM), 판독-전용 메모리(ROM), 광학 디스크, 자기 디스크, 하드 드라이브, 솔리드-스테이트 드라이브, 플래시 드라이브, 보안 디지털(SD) 카드, 메모리 스틱, 콤팩트 플래시(CF) 카드 등의 임의의 수의 임의의 조합을 포함할 수 있다. 메모리(404)는 또한 단일의 로직 구성요소로서 그룹화된 메모리들의 그룹(도 4에 도시되지 않음)일 수 있다.

[0060] 버스(410)는 내부 버스(예컨대, CPU-메모리 버스), 외부 버스(예컨대, 범용 직렬 버스 포트, 주변 구성요소 인터connect 익스프레스 포트) 등과 같이, 장치(400) 내부의 구성요소들 사이에서 데이터를 전달하는 통신 디바이스일 수 있다.

[0061] 모호함을 야기함 없이 설명의 용이함을 위해, 프로세서(402) 및 다른 데이터 프로세싱 회로들은 본 개시내용에

서 "데이터 프로세싱 회로"로서 집합적으로 지칭된다. 데이터 프로세싱 회로는 전체적으로 하드웨어로서 구현되거나 소프트웨어, 하드웨어 또는 펌웨어의 조합으로서 구현될 수 있다. 또한, 데이터 프로세싱 회로는 단일 독립 모듈일 수 있거나 장치(400)의 임의의 다른 구성요소에 전체적으로 또는 부분적으로 결합될 수 있다.

[0062] 장치(400)는 네트워크(예컨대, 인터넷, 인트라넷, 로컬 영역 네트워크, 모바일 통신 네트워크 등)와의 유선 또는 무선 통신을 제공하기 위한 네트워크 인터페이스(406)를 더 포함할 수 있다. 일부 실시예들에서, 네트워크 인터페이스(406)는 네트워크 인터페이스 제어기(NIC), 라디오 주파수(RF) 모듈, 트랜스폰더, 트랜시버, 모뎀, 라우터, 게이트웨이, 유선 네트워크 어댑터, 무선 네트워크 어댑터, 블루투스 어댑터, 적외선 어댑터, 근거리 통신("NFC") 어댑터, 셀룰러 네트워크 칩 등의 임의의 수의 임의의 조합을 포함할 수 있다.

[0063] 일부 실시예들에서, 선택적으로, 장치(400)는 하나 이상의 주변 디바이스들에 대한 연결을 제공하기 위해 주변 인터페이스(408)를 더 포함할 수 있다. 도 4에 도시된 바와 같이, 주변 디바이스는 커서 제어 디바이스(예컨대, 마우스, 터치패드 또는 터치스크린), 키보드, 디스플레이(예컨대, 음극선관 디스플레이, 액정 디스플레이, 또는 발광 다이오드 디스플레이), 비디오 입력 디바이스(예컨대, 비디오 아카이브에 커플링된 입력 인터페이스 또는 카메라) 등을 포함할 수 있지만, 이에 제한되지 않는다.

[0064] 비디오 코덱들(예컨대, 코덱 수행 프로세스(200A, 200B, 300A 또는 300B))은 장치(400)에서 임의의 소프트웨어 또는 하드웨어 모듈의 임의의 조합으로서 구현될 수 있다는 것이 주의되어야 한다. 예컨대, 프로세스(200A, 200B, 300A 또는 300B)의 일부 또는 모든 스테이지들은 장치(400)의 하나 이상의 소프트웨어 모듈들 이블테면, 메모리(404) 내에 로딩될 수 있는 프로그램 명령들로서 구현될 수 있다. 다른 예를 들면, 프로세스(200A, 200B, 300A 또는 300B)의 일부 또는 모든 스테이지들은, 장치(400)의 하나 이상의 하드웨어 모듈들 이블테면, 특수 데이터 프로세싱 회로(예컨대, FPGA, ASIC, NPU 등)로서 구현될 수 있다.

[0065] 양자화 및 역양자화 기능 블록들(예컨대, 도 2a 또는 도 2b의 양자화(214) 및 역양자화(218), 도 3a 또는 도 3b의 역양자화(218))에서, 예측 잔차들에 적용되는 양자화(및 역양자화)의 양을 결정하기 위하여 양자화 파라미터(QP)가 사용된다. 화상 또는 슬라이스의 코딩에 대해 사용되는 초기 QP 값들은, 예컨대, 화상 파라미터 세트(Picture Parameter Set; PPS)의 init_qp_minus26 선택스 요소를 사용하여 그리고 슬라이스 헤더의 slice_qp_delta 선택스 요소를 사용하여 코레벨에서 시그널링될 수 있다. 또한, QP 값들은 양자화 그룹들의 입도(granularity)에서 전송된 델타 QP 값들을 사용하여 각각의 CU에 대한 로컬 레벨에서 적응될 수 있다.

[0066] 개시된 실시예들에서, 프레임은 인코딩하기 위해, 화상은 코딩 트리 유닛(CTU)들의 시퀀스로 파티셔닝된다. 다수의 CTU들은 타일, 슬라이스 또는 서브화상을 형성할 수 있다. 화상들은 CTU들의 시퀀스로 분할된다. 3개의 샘플 어레이들을 갖는 화상에 대해, CTU는 크로마 샘플들의 2개의 대응하는 블록들과 함께 루마 샘플들의 N×N 블록들로 구성된다. 도 5는 본 개시내용의 일부 실시예들에 따라, 복수의 CTU들로 분할되는 화상의 예를 예시한다.

[0067] 일부 실시예들에 따르면, CTU에서 루마 블록의 최대 허용된 크기는 128×128로 지정되고(그러나, 루마 변환 블록들의 최대 크기는 64x64일 수 있음) CTU에서 루마 블록의 최소 허용된 크기는 32×32로 지정된다.

[0068] 화상은 하나 이상의 타일 행들 및 하나 이상의 타일 열들로 분할된다. 타일은 화상의 직사각형 구역을 커버하는 CTU들의 시퀀스이다. 슬라이스는 정수 개의 완전한 타일들 또는 화상의 타일 내에서 정수 개의 연속적 완전한 CTU 행들을 포함한다. 슬라이스들의 2개의 모드들: 래스터-스캔 슬라이스 모드 및 직사각형 슬라이스 모드가 지원될 수 있다. 래스터-스캔 슬라이스 모드에서, 슬라이스는 화상의 타일 래스터-스캔에서 완전한 타일들의 시퀀스를 포함한다. 직사각형 슬라이스 모드에서, 슬라이스는 화상의 직사각형 구역을 집합적으로 형성하는 다수의 완전한 타일들 또는 화상의 직사각형 구역을 집합적으로 형성하는 하나의 타일의 다수의 연속적인 완전한 CTU 행들을 포함한다. 직사각형 슬라이스 내의 타일들은 그 슬라이스에 대응하는 직사각형 구역 내에서 타일 래스터-스캔 순서로 스캔된다.

[0069] 서브화상은 화상의 직사각형 구역을 집합적으로 커버하는 하나 이상의 슬라이스들을 포함한다.

[0070] 도 6은 본 개시내용의 일부 실시예들에 따라, 타일들 및 래스터-스캔 슬라이스들로 파티셔닝된 화상의 예를 예시한다. 도 6에 도시된 바와 같이, 화상은 12개의 타일들(4개의 타일 행들 및 3개의 타일 열들) 및 3개의 래스터-스캔 슬라이스들로 분할된다.

[0071] 도 7은 본 개시내용의 일부 실시예들에 따라, 타일들 및 직사각형 슬라이스들로 파티셔닝된 화상의 예를 예시한다. 도 7에 도시된 바와 같이, 화상은 20개의 타일들(5개의 타일 열들 및 4개의 타일 행들) 및 9개의 직사각형

슬라이스들로 분할된다.

- [0072] 도 8은 본 개시내용의 일부 실시예들에 따라, 타일들 및 직사각형 슬라이스들로 파티셔닝된 화상의 다른 예를 예시한다. 도 8에 도시된 바와 같이, 화상은 4개의 타일들(2개의 타일 열들 및 2개의 타일 행들) 및 4개의 직사각형 슬라이스들로 분할된다.
- [0073] 도 9는 본 개시내용의 일부 실시예들에 따라, 서브화상들로 파티셔닝된 화상의 예를 예시한다. 도 9에 도시된 바와 같이, 화상은 4×4 CTU들의 하나의 슬라이스를 각각 커버하는 왼손 측 상의 12개 및 2×2 CTU들의 2개의 수직-스택 슬라이스들을 각각 커버하는 오른손 측 상의 8개의 타일들인 20개의 타일들(5개의 타일 열들 및 4개의 타일 행들)로 파티셔닝되며, 함께 다양한 디멘션들의 28개의 슬라이스들 및 28개의 서브화상들(각각의 슬라이스는 서브화상임)을 발생시킨다.
- [0074] 일부 개시된 실시예들에 따르면, 서브화상 파티셔닝 정보는 시퀀스 파라미터 세트(PS)에서 시그널링된다. 도 10은 본 개시내용의 일부 실시예들에 따라, 서브화상 파티셔닝의 예시적인 SPS 선택스를 도시하는 예시적인 표 1을 예시한다.
- [0075] 표 1에서, 선택스 요소 `sps_num_subpics_minus1 + 1`은 하나의 화상 내 서브화상들의 수를 지정하고, 선택스 요소들 `subpic_ctu_top_left_x[ i ]` 및 `subpic_ctu_top_left_y[ i ]`는 CtbSizeY 단위로 i-번째 서브화상의 상단 좌측 CTU의 포지션을 지정하고, 선택스 요소 `subpic_width_minus1[ i ] + 1` 및 선택스 요소 `subpic_height_minus1[ i ] + 1`은 CtbSizeY 단위로 i-번째 서브화상의 폭 및 높이를 각각 지정한다. 이러한 선택스 요소들의 시맨틱은 다음과 같다.
- [0076] 1과 동일한 `subpics_present_flag`는 서브화상 파라미터들이 SPS RBSP 선택스에 존재함을 지정하고; 0과 동일한 `subpics_present_flag`는 서브화상 파라미터들이 SPS RBSP 선택스에 존재하지 않음을 지정한다.
- [0077] `sps_num_subpics_minus1 + 1`은 서브화상들의 수를 지정한다. 선택스 요소들 `sps_num_subpics_minus1`은 0 내지 254의 범위에 있다. 존재하지 않을 때, 선택스 요소 `sps_num_subpics_minus1`의 값은 0과 동일한 것으로 추론된다.
- [0078] `subpic_ctu_top_left_x[ i ]`는 CtbSizeY 단위로 i-번째 서브화상의 상단 좌측 CTU의 수평 포지션을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 `subpic_ctu_top_left_x[ i ]`의 값은 0과 동일한 것으로 추론된다.
- [0079] `subpic_ctu_top_left_y[ i ]`는 CtbSizeY 단위로 i-번째 서브화상의 상단 좌측 CTU의 수직 포지션을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 `subpic_ctu_top_left_y[ i ]`의 값은 0과 동일한 것으로 추론된다.
- [0080] `subpic_width_minus1[ i ] + 1`은 CtbSizeY 단위로 i-번째 서브화상의 폭을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 `subpic_width_minus1[ i ]`의 값은 $\text{Ceil}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 과 동일한 것으로 추론된다.
- [0081] `subpic_height_minus1[ i ] + 1`은 CtbSizeY 단위로 i-번째 서브화상의 높이를 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 `subpic_height_minus1[ i ]`의 값은 $\text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 과 동일한 것으로 추론된다.
- [0082] 1과 동일한 `subpic_treated_as_pic_flag[ i ]`는 코딩된 층 비디오 시퀀스(CLV) 내 각각의 코딩된 화상의 i-번째 서브화상이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되는 것을 지정한다. 0과 동일한 선택스 요소 `subpic_treated_as_pic_flag[ i ]`는 CLVS 내 각각의 코딩된 화상의 i-번째 서브화상이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되지 않는 것을 지정한다. 존재하지 않을 때, 선택스 요소 `subpic_treated_as_pic_flag[ i ]`의 값은 0과 동일한 것으로 추론된다.
- [0083] 1과 동일한 `loop_filter_across_subpic_enabled_flag[ i ]`는 CLVS 내 각각의 코딩된 화상의 i-번째 서브화상의 경계들에 걸쳐 인-루프 필터링 동작들이 수행될 수 있음을 지정한다. 0과 동일한 선택스 요소 `loop_filter_across_subpic_enabled_flag[ i ]`는 CLVS 내 각각의 코딩된 화상의 i-번째 서브화상의 경계들에 걸쳐 인-루프 필터링 동작들이 수행되지 않음을 지정한다. 존재하지 않을 때, 선택스 요소

loop_filter_across_subpic_enabled_pic_flag[i]의 값은 1과 동일한 것으로 추론된다.

- [0084] 본 개시내용의 개시된 실시예들과 일치하여, 각각의 서브화상에는 식별자가 할당될 수 있다. 서브화상들의 식별자 정보는 시그널링 파라미터 세트(SPS), 화상 파라미터 세트(PPS) 또는 화상 헤더(PH)에서 시그널링될 수 있다. 도 11은 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 예시적인 SPS 선택스를 도시하는 예시적인 표 2를 예시한다. 도 12는 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 예시적인 PPS 선택스를 도시하는 예시적인 표 3을 예시한다. 도 13은 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 예시적인 PH 선택스를 도시하는 예시적인 표 4를 예시한다.
- [0085] 표들 2-4에 도시된 바와 같이, 선택스 요소 sps_subpic_id_present_flag는 서브화상 ID 매핑이 SPS에 존재하는지를 표시하고; 선택스 요소들 sps_subpic_id_signaling_present_flag, pps_subpic_id_signaling_present_flag 및 ph_subpic_id_signaling_present_flag는 서브화상 ID 매핑이 SPS, PPS 또는 PH에서 각각 시그널링되는지를 표시하고; 선택스 요소 sps_subpic_id_len_minus1 + 1, 선택스 요소 pps_subpic_id_len_minus1 + 1 및 선택스 요소 ph_subpic_id_len_minus1 + 1은 각각 SPS, PPS 및 PH에서 시그널링되는 서브화상 ID인 선택스 요소 sps_subpic_id[i], pps_subpic_id[i] 및 ph_subpic_id[i]를 각각 제시하는 데 사용되는 비트들의 수를 지정한다.
- [0086] 위의 선택스 요소들의 시맨틱 및 관련된 비트스트림 부합 요건은 다음과 같이 설명된다.
- [0087] 1과 동일한 sps_subpic_id_present_flag는 서브화상 ID 매핑이 SPS에 존재함을 지정하고; 0과 동일한 선택스 요소 sps_subpic_id_present_flag는 서브화상 ID 매핑이 SPS에 존재하지 않음을 지정한다.
- [0088] 1과 동일한 sps_subpic_id_signaling_present_flag는 서브화상 ID 매핑이 SPS에서 시그널링됨을 지정하고; 0과 동일한 선택스 요소 sps_subpic_id_signaling_present_flag는 서브화상 ID 매핑이 SPS에서 시그널링되지 않음을 지정한다. 존재하지 않을 때, 선택스 요소 sps_subpic_id_signaling_present_flag의 값은 0과 동일한 것으로 추론된다.
- [0089] sps_subpic_id_len_minus1 + 1은 선택스 요소 sps_subpic_id[i]를 표현하는 데 사용되는 비트들의 수를 지정한다. 선택스 요소 sps_subpic_id_len_minus1의 값은 0 내지 15 범위에 있다(해당 값들을 포함함).
- [0090] sps_subpic_id[i]는 i-번째 서브화상의 서브화상 ID를 지정한다. 선택스 요소 sps_subpic_id[i]의 길이는 sps_subpic_id_len_minus1 + 1 비트이다. 존재하지 않을 때, 그리고 선택스 요소 sps_subpic_id_present_flag가 0과 동일할 때, 선택스 요소 sps_subpic_id[i]의 값은 0 내지 sps_num_subpics_minus1의 범위(해당 값들을 포함함) 내 각각의 i에 대해 i와 동일한 것으로 추론된다.
- [0091] 1과 동일한 pps_subpic_id_signaling_present_flag는 서브화상 ID 매핑이 PPS에서 시그널링됨을 지정한다. 0과 동일한 선택스 요소 pps_subpic_id_signaling_present_flag는 서브화상 ID 매핑이 PPS에서 시그널링되지 않음을 지정한다. 선택스 요소 sps_subpic_id_present_flag가 0이거나 선택스 요소 sps_subpic_id_signaling_present_flag가 1과 동일할 때, 선택스 요소 pps_subpic_id_signaling_present_flag는 0과 동일할 수 있다.
- [0092] pps_num_subpics_minus1 + 1은 PPS를 참조하는 코딩된 화상들 내 서브화상들의 수를 지정한다. 선택스 요소 pps_num_subpic_minus1의 값이 선택스 요소 sps_num_subpics_minus1과 동일한 것이 비트스트림 부합의 요건일 수 있다.
- [0093] pps_subpic_id_len_minus1 + 1은 선택스 요소 pps_subpic_id[i]를 표현하는 데 사용되는 비트들의 수를 지정한다. 선택스 요소 pps_subpic_id_len_minus1의 값은 0 내지 15 범위에 있다(해당 값들을 포함함). CLVS 내 코딩된 화상들에 의해 참조되는 모든 PPS들에 대해 선택스 요소 pps_subpic_id_len_minus1의 값이 동일하다는 것이 비트스트림 부합의 요건일 수 있다.
- [0094] pps_subpic_id[i]는 i-번째 서브화상의 서브화상 ID를 지정한다. 선택스 요소 pps_subpic_id[i]의 길이는 pps_subpic_id_len_minus1 + 1 비트이다.
- [0095] 1과 동일한 ph_subpic_id_signaling_present_flag는 서브화상 ID 매핑이 PH에서 시그널링됨을 지정하고; 0과 동일한 선택스 요소 ph_subpic_id_signaling_present_flag는 서브화상 ID 매핑이 PH에서 시그널링되지 않음을 지정한다.
- [0096] ph_subpic_id_len_minus1 + 1은 선택스 요소 ph_subpic_id[i]를 표현하는 데 사용되는 비트들의 수를 지정한다.

다. 선택스 요소 `pic_subpic_id_len_minus1`의 값은 0 내지 15 범위에 있다(해당 값들을 포함함). CLVS 내 코딩된 화상들에 의해 참조되는 모든 PH들에 대해 선택스 요소 `ph_subpic_id_len_minus1`의 값이 동일하다는 것이 비트스트림 부합의 요건일 수 있다.

[0097] `ph_subpic_id[ i ]`는 i -번째 서브화상의 서브화상 ID를 지정한다. 선택스 요소 `ph_subpic_id[ i ]`의 길이는 `ph_subpic_id_len_minus1 + 1` 비트이다.

[0098] 서브화상 ID와 관련된 이러한 선택스 요소들을 파싱한 후, 다음 선택스(1)를 사용하여 서브화상 ID 목록 `SubpicIdList`가 도출된다:

```
for( i = 0; i <= sps_num_subpics_minus1; i++)
```

```
    SubpicIdList[ i ] = sps_subpic_id_present_flag ?
```

Syntax (1)

```
( sps_subpic_id_signaling_present_flag ? sps_subpic_id[ i ] :
```

```
( ph_subpic_id_signaling_present_flag ? ph_subpic_id[ i ] : pps_subpic_id[ i ] ) )
```

```
    ; i
```

[0099]

[0100] 그러나, 서브화상 파티셔닝의 위에서 설명된 시그널링에는 여러 문제들이 있다. 첫째로, 시맨틱에 따르면, 선택스 요소들 `sps_subpic_id_present_flag` 및 `sps_subpic_id_signaling_present_flag` 둘 모두는 서브화상 ID가 SPS에 존재하는지를 지정한다. 그리고 표 2에 따르면, 이들 선택스 요소들 둘 모두가 참인 경우에만, 서브화상 id 정보가 SPS에서 시그널링된다. 따라서 시그널링에 중복성이 있다. 둘째로, `sps_subpics_id_present_flag`가 참일 때조차도, `pps_subpic_id_signalling_present_flag` 또는 `ph_subpic_id_signalling_present_flag`가 참인 경우 서브화상 ID가 PPS 또는 PH에서 또한 시그널링될 수 있다. 셋째로, Syntax (1)에 따르면, `sps_subpic_id_present_flag`가 참이 아닐 때, 서브화상 인덱스와 동일한 디폴트 ID가 각각의 서브화상에 할당된다. `sps_subpic_id_present_flag`가 참일 때, `SubpicIdList[ i ]`는 `sps_subpic_id[ i ]`, `ph_subpic_id[ i ]` 또는 `pps_subpic_id[ i ]`로서 도출된다. 그러나 `sps_subpic_id_present_flag`가 참인 경우 SPS, PPS 또는 PH에 서브화상 ID가 존재하지 않아서, 이 경우에 선택스 요소 `pps_subpic_id`의 정의되지 않은 값이 `SubpicIdList`에 할당되는 것이 가능하다. Syntax (1)에서, 선택스 요소 `sps_subpic_id_present_flag`가 참이고, 선택스 요소들 `sps_subpic_id_signaling_present_flag` 및 `ph_subpic_id_signaling_present_flag` 둘 모두는 거짓일 때, 선택스 요소 `pps_subpic_id[ i ]`는 선택스 요소 `pps_subpic_id_signaling_present_flag`의 값에 관계없이 `SubpicIdList[ i ]`에 할당된다. 선택스 요소 `ps_subpic_id_signaling_present_flag`가 거짓인 경우, 선택스 요소 `pps_subpic_id[ i ]`는 정의되지 않는다.

[0101] 또한, 도 10의 표 1에 도시된 바와 같이, 선택스 요소 `subpics_present_flag`가 참인 경우, 서브화상들의 수가 먼저 시그널링되고 나서, 각각의 서브화상의 상단-좌측 포지션, 폭 및 높이 및 2개의 제어 플래그들 `subpic_treated_as_pic_flag` 및 `loop_filter_across_subpic_enabled_flag`가 시그널링된다. 단 하나의 서브화상의 경우에도(선택스 요소 `sps_num_subpics_minus1`이 0과 동일할 때), 상단-좌측 포지션, 폭 및 높이, 그리고 이들 2개의 제어 플래그들이 시그널링된다. 그러나, 화상에 1개의 서브화상만이 있는 경우에는 서브화상이 화상과 동일하기 때문에 이러한 것들을 표시할 필요가 없어서, 시그널링되는 정보는 화상 자체로부터 도출될 수 있다.

[0102] 또한, 화상을 분할함으로써 서브화상이 획득되고, 모든 서브화상들을 병합함으로써 화상이 형성된다. 마지막 서브화상의 포지션 및 크기는 전체 화상의 크기 및 모든 이전 서브화상 포지션들 및 크기들로부터 도출될 수 있다. 따라서 마지막 서브화상의 포지션, 폭, 높이 정보를 시그널링할 필요가 없다.

[0103] 또한, 도 11의 표 2에 도시된 바와 같이, 선택스 요소 `sps_subpic_id_present_flag`는 선택스 요소 `subpics_present_flag`의 값에 관계없이 항상 시그널링된다. 따라서, 위에서 설명된 시그널링 방법들에서, 서브화상이 존재하지 않는 경우에도, 서브화상 식별자는 여전히 시그널링될 수 있으며, 이는 의미가 없다.

[0104] 본 개시내용은 위의 문제들을 해결하기 위한 시그널링 방법들을 제공한다. 일부 예시적인 실시예들이 아래에서 상세히 설명된다.

- [0105] 본 개시내용의 일부 실시예들에서, 선택스 요소 `sps_subpic_id_present_flag`가 참이지만 선택스 요소들 `sps_subpic_id_signaling_present_flag` 및 `pps_subpic_id_signaling_present_flag` 둘 모두가 거짓인 경우, 서브화상 ID는 화상 헤더에서 시그널링되도록 강제될 수 있다. 이는 서브화상 ID가 정의되지 않은 경우를 회피할 수 있다.
- [0106] 예컨대, 비트스트림 부합 제약은 다음 두 가지 방식으로 부과될 수 있다. 제1 방식에서, 비트스트림 부합 제약에 대한 시맨틱은 다음과 같다(이탤릭체로 강조됨): 1과 동일한 `ph_subpic_id_signaling_present_flag`는 서브화상 ID 매핑이 PH에서 시그널링됨을 지정한다. 0과 동일한 선택스 요소 `ph_subpic_id_signaling_present_flag`는 서브화상 ID 매핑이 PH에서 시그널링되지 않음을 지정한다. 선택스 요소 `sps_subpic_id_present_flag`가 1과 동일하고, 선택스 요소 `sps_subpic_id_signaling_present_flag`가 0과 동일하고 선택스 요소 `pps_subpic_id_signaling_present_flag`가 0과 동일할 때, 선택스 요소 `ph_subpic_id_signal`의 값은 1이라는 것이 비트스트림 부합 요건일 수 있다.
- [0107] 제2 방식에서, 비트스트림 부합 제약에 대한 시맨틱은 다음과 같다(이탤릭체로 강조됨): 1과 동일한 `ph_subpic_id_signaling_present_flag`는 서브화상 ID 매핑이 PH에서 시그널링됨을 지정한다. 0과 동일한 선택스 요소 `ph_subpic_id_signaling_present_flag`는 서브화상 ID 매핑이 PH에서 시그널링되지 않음을 지정한다. 선택스 요소 `sps_subpic_id_present_flag`가 1과 동일하고 선택스 요소 `sps_subpic_id_signaling_present_flag`가 0과 동일하고 CLVS 내 코딩된 화상들에 의해 참조되는 모든 PPS의 선택스 요소 `pps_subpic_id_signaling_present_flag`가 0과 동일할 때,
- [0108] 참조하는 CLVS 내 코딩된 화상들에 의해 참조되는 모든 PH 중, 선택스 요소 `ph_subpic_id_signaling_present_flag`의 값이 1과 동일한 적어도 하나의 PH가 있는 것이 비트스트림 부합 요건일 수 있다.
- [0109] 도 14는 본 개시내용의 일부 실시예들에 따라, 제2 방식의 예시적인 비트스트림 부합 제약을 예시하는 개략도이다.
- [0110] 선택스 요소 `sps_subpic_id_present_flag`의 시맨틱은 현재 VVC 드래프트에서 명확하게 정의되지 않고 다음과 같이 변경될 수 있다(이탤릭체로 강조됨).
- [0111] 1과 동일한 `sps_subpic_id_present_flag`는 서브화상 ID 매핑이 SPS, PPS 또는 PH에 존재함을 지정한다. 0과 동일한 선택스 요소 `sps_subpic_id_present_flag`는 서브화상 ID 매핑이 SPS, PPS 및 PH에 존재하지 않음을 지정한다.
- [0112] 이는 선택스 요소 `sps_subpic_id_present_flag`가 참이지만 선택스 요소들 `sps_subpic_id_signaling_present_flag` 및 `pps_subpic_id_signaling_present_flag` 둘 모두가 거짓일 때, 선택스 요소 `ph_subpic_id`가 시그널링되는 것을 보장할 수 있다. 선택스는 도 11 내지 도 13의 표들 2-4에서 보여진다.
- [0113] 위의 실시예들에서, 서브화상 ID 존재 플래그가 참(선택스 요소 `sps_subpic_id_present_flag` = 1)인 경우, 사용된 서브화상 ID는 비트스트림에서(SPS, PPS 또는 PH 중 어느 하나에서) 시그널링되고 어떠한 추론 규칙도 필요하지 않다. 서브화상 ID 존재 플래그가 참(선택스 요소 `sps_subpic_id_present_flag` = 1)일 때, 서브화상 ID가 SPS, PPS 또는 PH 중 하나에서 시그널링되도록 강제하는 것은, 서브화상 ID가 존재함을 선택스 요소 `sps_subpic_id_present_flag`가 표시하기 때문에 비트스트림에서 서브화상 ID를 시그널링하지 않고 서브화상 ID를 도출하기 위해 추론 규칙을 사용하는 것보다 더 나을 수 있다.
- [0114] 다른 예로서, 도 15는 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 다른 예시적인 PH 선택스를 도시하는 예시적인 표 5를 예시한다. 표 5는 표 4에 보여진 PH 선택스의 수정(박스(1501)에서 보여지고 이탤릭체로 강조됨)을 보여준다. 표 5를 참조하면, 선택스 요소 `sps_subpic_id_present_flag`가 참이지만 선택스 요소들 `sps_subpic_id_signaling_present_flag` 및 `pps_subpic_id_signaling_present_flag` 둘 모두가 거짓일 때, 선택스 요소 `ph_subpic_id`는 선택스 요소 `ph_subpic_id_signaling_present_flag`를 참으로 추론함으로써 시그널링되도록 강제된다.
- [0115] 선택스 요소 `ph_subpic_id_signaling_present_flag`는 다음의 2개의 대안적인 시맨틱을 가질 수 있다(이탤릭체로 강조됨).
- [0116] 제1 시맨틱은 다음을 포함한다(이탤릭체로 강조): 1과 동일한 `ph_subpic_id_signaling_present_flag`는 서브화

상 ID 매핑이 PH에서 시그널링됨을 지정한다. 0과 동일한 선택스 요소 `ph_subpic_id_signaling_present_flag`는 서브화상 ID 매핑이 PH에서 시그널링되지 않음을 지정한다. 존재하지 않을 때, 선택스 요소 `ph_subpic_id_signaling_present_flag`의 값은 1인 것으로 추론된다.

[0117] 제2 시맨틱은 다음을 포함한다(이탤릭체로 강조됨): 1과 동일한 `ph_subpic_id_signaling_present_flag`는 서브화상 ID 매핑이 PH에서 시그널링됨을 지정하고; 0과 동일한 선택스 요소 `ph_subpic_id_signaling_present_flag`는 서브화상 ID 매핑이 PH에서 시그널링되지 않음을 지정한다. 존재하지 않을 때, 선택스 요소 `sps_subpic_id_present_flag`가 1과 동일하고 선택스 요소 `sps_subpic_id_signaling_present_flag`가 0과 동일한 경우, 선택스 요소 `ph_subpic_id_signaling_present_flag`의 값은 1인 것으로 추론된다.

[0118] 선택스 요소 `sps_subpic_id_present_flag`의 시맨틱은 다음과 같이 변경될 수 있다(이탤릭체로 강조됨).

[0119] 1과 동일한 `sps_subpic_id_present_flag`는 서브화상 ID 매핑이 SPS, PPS 또는 PH에 존재함을 지정한다. 0과 동일한 선택스 요소 `sps_subpic_id_present_flag`는 서브화상 ID 매핑이 SPS, PPS 및 PH에 존재하지 않음을 지정한다.

[0120] 일부 실시예들에서, 선택스 요소 `sps_subpic_id_present_flag`가 참일 때, 서브화상 ID의 추론 규칙은 필요하지 않다. 따라서, 선택스 요소 `ph_subpic_id_signaling_present_flag`의 시그널링은, 서브화상 ID 존재 플래그가 참(선택스 요소 `sps_subpic_id_present_flag=1`)이지만 서브화상 ID가 SPS 또는 PPS에서 시그널링되지 않을 때(선택스 요소 `sps_subpic_id_signaling_present_flag=0` 및 선택스 요소 `pps_subpic_id_signaling_present_flag=0`), 스킵된다. 그러면, 1 비트가 절약될 수 있다.

[0121] 다른 예로서, 선택스 요소 `ph_subpic_id`는, 선택스 요소 `sps_subpic_id_present_flag`가 참이지만 선택스 요소들 `sps_subpic_id_signaling_present_flag` 및 `pps_subpic_id_signaling_present_flag` 둘 모두가 거짓(즉, 서브화상 ID가 SPS 또는 PPS에서 시그널링되지 않음)인 경우 시그널링되도록 강제되며, 이는 이 경우에 서브화상 ID가 PH에서 시그널링됨을 의미한다. 선택스 요소 `pps_subpic_id`가 시그널링되는 경우(선택스 요소 `sps_subpic_id_present_flag`는 참이고, 선택스 요소 `sps_subpic_id_signaling_present_flag`는 거짓이고, 선택스 요소 `pps_subpic_id_signaling_present_flag`는 참임), 선택스 요소 `ph_subpic_id`는 시그널링될 수 없다. 도 16은 본 개시내용의 일부 실시예들에 따라, 서브화상 식별자의 다른 예시적인 PH 선택스(박스(1601)로 강조하여 보여지고 이탤릭체로 강조됨)를 도시하는 예시적인 표 6를 예시한다.

[0122] 목록 `SubpicIdList[ i ]`는 다음과 같이 선택스(2)에 따라 도출된다:

```
for( i = 0; i <= sps_num_subpics_minus1; i++ )
```

```
    SubpicIdList[ i ] = sps_subpic_id_present_flag ?
```

Syntax (2)

```
( sps_subpic_id_signaling_present_flag ? sps_subpic_id[ i ] :
```

```
( pps_subpic_id_signaling_present_flag ? pps_subpic_id[ i ] : ph_subpic_id[ i ] )
```

```
); i
```

[0123]

[0124] 일부 실시예들에서, 선택스 요소 `sps_subpic_id_present_flag`가 참일 때, 서브화상 ID의 추론 규칙은 필요하지 않다. 선택스 요소 `ph_subpic_id_signaling_present_flag`는 제거될 수 있다. 따라서, 선택스 요소 `sps_subpic_id_present_flag`가 1과 동일하고, 선택스 요소 `sps_subpic_id_signaling_present_flag`가 0과 동일하고, 선택스 요소 `pps_subpic_id_signaling_present_flag`가 1과 동일한 경우 1 비트가 절약된다.

[0125] 서브화상 ID가 PPS에서 이미 시그널링된 경우에, 일부 실시예들은 PH에서 서브화상 ID를 재차 시그널링함으로써 인코더가 PPS 내 서브화상 ID를 무시하는 옵션을 제공할 수 있다. 이는 인코더에 대해 보다 유연한 방식이다.

[0126] 본 개시내용의 일부 실시예들에서, 서브화상 ID 목록 `SubpicIdList`가 도출될 수 있음을 보장하기 위해 추론 규칙이 주어진다.

[0127] 일 예로서, 추론 규칙이 선택스 요소 `pps_subpic_id`에 주어져서, 서브화상 ID가 SPS, PPS 또는 PH에서 시그널링되지 않을 때, 추론 규칙에 의해 추론된 선택스 요소 `pps_subpic_id`의 디폴트 값은 서브화상 ID 목록

SubpicIdList를 도출하는 데 사용된다.

[0128] 선택스 요소 pps_subpic_id의 시맨틱은 다음과 같다(이탤릭체로 강조): pps_subpic_id[i]는 i-번째 서브화상의 서브화상 ID를 지정한다. 선택스 요소 pps_subpic_id[i]의 길이는 pps_subpic_id_len_minus1 + 1 비트이다. 존재하지 않을 때, 선택스 요소 pps_subpic_id[i]의 값은 0 내지 pps_num_subpics_minus1 범위 내 각각의 i에 대해 i인 것으로 추론된다.

[0129] 다른 예로서, SubpicIdList의 도출 프로세스에서 추론 규칙이 주어진다. 선택스 요소 sps_subpic_id_present_flag가 참이고, 선택스 요소들 sps_subpic_id_signaling_present_flag, pps_subpic_id_signaling_present_flag 및 ph_subpic_id_signaling_present_flag 모두가 거짓일 때, 디폴트 값이 SubpicIdList[i]에 할당된다.

[0130] SubpicIdList의 도출은 다음과 같이 선택스(3)에 따른다(이탤릭체로 강조됨):

```
for( i = 0; i <= sps_num_subpics_minus1; i++ )
    SubpicIdList[ i ] = sps_subpic_id_present_flag ?
        ( sps_subpic_id_signaling_present_flag ? sps_subpic_id[ i ] :
          ( ph_subpic_id_signaling_present_flag ? ph_subpic_id[ i ] :
            ( pps_subpic_id_signaling_present_flag ? pps_subpic_id[ i ] : i ) ) ) : i
```

Syntax (3)

[0131]

[0132] 일부 실시예들에서, pps_subpic_id에 추론 규칙을 부과함으로써, 심지어 비트스트림에서 서브화상 ID의 시그널링이 전혀 없어도 SubpicIdList가 도출될 수 있다는 것이 보장될 수 있다. 따라서, 서브화상 ID를 시그널링하는데 전용되는 비트들이 절약될 수 있다.

[0133] 본 개시내용의 일부 실시예들에서, 서브화상 ID 목록 SubpicIdList의 도출 규칙은 PH보다는 PPS에서 시그널링되는 서브화상 ID에 더 높은 우선순위를 부여하도록 변경될 수 있다. 따라서, 선택스 요소 pps_subpic_id_signaling_present_flag는 선택스 요소 ph_subpic_id_signaling_present_flag에 앞서 체크된다.

[0134] 일 예로서, SubpicIDList의 도출 규칙은 아래에 보여진 바와 같이 선택스(4)에 따르며(이탤릭체로 강조됨), 선택스 요소 ph_subpic_id에 대한 추론 규칙이 제공된다.

```
for( i = 0; i <= sps_num_subpics_minus1; i++ )
    SubpicIdList[ i ] = sps_subpic_id_present_flag ?
        ( sps_subpic_id_signaling_present_flag ? sps_subpic_id[ i ] :
          ( pps_subpic_id_signaling_present_flag ? pps_subpic_id[ i ] : ph_subpic_id[ i ] ) ) : i
```

Syntax (4)

[0135]

[0136] 선택스(4)와 일치하는 시맨틱은 다음과 같다(이탤릭체로 강조): ph_subpic_id[i]는 i 번째 서브화상의 서브화상 ID를 지정한다. 선택스 요소 ph_subpic_id[i]의 길이는 ph_subpic_id_len_minus1 + 1 비트이다. 존재하지 않을 때, 선택스 요소 ph_subpic_id[i]의 값은 0 내지 ph_num_subpics_minus1 범위 내 각각의 i에 대해 i인 것으로 추론된다.

[0137] 다른 예로서, SubpicIDList의 도출 규칙은 아래에 보여진 바와 같이 선택스(5)에 따르며(이탤릭체로 강조됨), 이 예에서 선택스 요소 ph_subpic_id에 대한 어떠한 부가적인 추론 규칙도 없다.

```
for( i = 0; i <= sps_num_subpics_minus1; i++)
```

```
    SubpicIdList[ i ] = sps_subpic_id_present_flag ?
```

Syntax (5)

```
    ( sps_subpic_id_signaling_present_flag ? sps_subpic_id[ i ] :
```

```
    ( pps_subpic_id_signaling_present_flag ? pps_subpic_id[ i ] :
```

```
    ( ph_subpic_id_signaling_present_flag ? ph_subpic_id[ i ] : i ) ) : i
```

[0138]

[0139]

비트스트림에서 서브화상 ID를 전혀 시그널링하지 않고도 SubpicIdList가 올바르게 도출될 수 있음을 보장하도록 추론 규칙이 선택 요소 ph_subpic_id 또는 SubpicIdList 도출 프로세스에 제공된다. 따라서, 추론 규칙에 의해 추론된 디폴트 서브화상 ID가 사용되는 경우, 서브화상 시그널링에 전용되는 비트들이 절약될 수 있다.

[0140]

본 개시내용의 일부 실시예들에서, 서브화상의 수가 1과 동일할 때 서브화상에 대해 시그널링되는 중복 정보가 제거될 수 있다.

[0141]

일 예로서, SPS 선택 요소는 도 17a의 표 7A(박스들(1701-1702)로 강조하여 보여지고 이탤릭체로 강조표시됨) 또는 도 17b의 표 7B(박스들(1711-1712)로 강조하여 보여지고 이탤릭체로 강조표시됨)에서 보여진다. 표 7A 및 표 7B는 동등하다는 것이 인지된다. 표 7A 및 표 7B 내 선택 요소와 일치하는 시맨틱(이탤릭체로 강조됨)이 아래에서 보여진다.

[0142]

subpic_ctu_top_left_x[i]는 CtbSizeY 단위로 i-번째 서브화상의 상단 좌측 CTU의 수평 위치선을 지정한다. 선택 요소의 길이는 $\text{Ceil}(\log_2(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택 요소 subpic_ctu_top_left_x[i]의 값은 0과 동일한 것으로 추론된다.

[0143]

subpic_ctu_top_left_y[i]는 CtbSizeY 단위로 i-번째 서브화상의 상단 좌측 CTU의 수직 위치선을 지정한다. 선택 요소의 길이는 $\text{Ceil}(\log_2(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택 요소 subpic_ctu_top_left_y[i]의 값은 0과 동일한 것으로 추론된다.

[0144]

subpic_width_minus1[i] + 1은 CtbSizeY 단위로 i-번째 서브화상의 폭을 지정한다. 선택 요소의 길이는 $\text{Ceil}(\log_2(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택 요소 subpic_width_minus1[i]의 값은 $\text{Ceil}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 과 동일한 것으로 추론된다. 여기서 "Ceil()" 은 가장 가까운 정수로 반올림하는 함수이다. 따라서 $\text{Ceil}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 은 $(\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 과 동일하며, 여기서 "/" 는 정수 나눗셈이다.

[0145]

subpic_height_minus1[i] + 1은 CtbSizeY 단위로 i-번째 서브화상의 높이를 지정한다. 선택 요소의 길이는 $\text{Ceil}(\log_2(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택 요소 subpic_height_minus1[i]의 값은 $\text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 과 동일한 것으로 추론된다. 여기서 "Ceil()" 은 가장 가까운 정수로 반올림하는 함수이다. 따라서 $\text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 은 $(\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 과 동일하며, 여기서 "/" 는 정수 나눗셈이다.

[0146]

1과 동일한 subpic_treated_as_pic_flag[i]는 CLVS 내 각각의 코딩된 화상의 i-번째 서브화상이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되는 것을 지정한다. 0과 동일한 선택 요소 subpic_treated_as_pic_flag[i]는 CLVS 내 각각의 코딩된 화상의 i-번째 서브화상이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되지 않는 것을 지정한다. 존재하지 않을 때, 선택 요소 subpics_present_flag가 1과 동일하고 선택 요소 sps_num_subpics_minus1이 0과 동일한 경우, 선택 요소 subpic_treated_as_pic_flag[i]의 값은 1과 동일한 것으로 추론되고; 그렇지 않으면, 선택 요소 subpic_treated_as_pic_flag[i]의 값은 0과 동일한 것으로 추론된다.

[0147]

1과 동일한 loop_filter_across_subpic_enabled_flag[i]는 CLVS 내 각각의 코딩된 화상의 i-번째 서브화상의 경계들에 걸쳐 인-루프 필터링 동작들이 수행될 수 있음을 지정한다. 0과 동일한 선택 요소 loop_filter_across_subpic_enabled_flag[i]는 CLVS 내 각각의 코딩된 화상의 i-번째 서브화상의 경계들에 걸쳐 인-루프 필터링 동작들이 수행되지 않음을 지정한다. 존재하지 않을 때, 선택 요소 subpics_present_flag가 1과 동일하고 선택 요소 sps_num_subpics_minus1이 0과 동일한 경우, 선택 요소

loop_filter_across_subpic_enabled_flag [*i*]의 값은 0과 동일한 것으로 추론되고; 그렇지 않으면, 선택스 요소 *loop_filter_across_subpic_enabled_pic_flag*[*i*]의 값은 1과 동일한 것으로 추론된다.

- [0148] 다른 예로서, SPS 선택스는 도 18의 표 8(박스들(1801-1802)로 강조하여 보여지고 이탤릭체로 강조표시됨)에서 보여진다. 표 8의 선택스와 일치하는 시맨틱(이탤릭체로 강조됨)은 아래에서 설명된다.
- [0149] *subpic_ctu_top_left_x*[*i*]는 CtbSizeY 단위로 *i*-번째 서브화상의 상단 좌측 CTU의 수평 위치션을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 *subpic_ctu_top_left_x*[*i*]의 값은 0과 동일한 것으로 추론된다.
- [0150] *subpic_ctu_top_left_y*[*i*]는 CtbSizeY 단위로 *i*-번째 서브화상의 상단 좌측 CTU의 수직 위치션을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 *subpic_ctu_top_left_y*[*i*]의 값은 0과 동일한 것으로 추론된다.
- [0151] *subpic_width_minus1*[*i*] + 1은 CtbSizeY 단위로 *i*-번째 서브화상의 폭을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 *subpic_width_minus1*[*i*]의 값은 $\text{Ceil}((\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) - 1)$ 과 동일한 것으로 추론된다. 여기서 "Ceil()"은 가장 가까운 정수로 반올림하는 함수이다. 따라서 $\text{Ceil}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 은 $(\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 와 동일하며, 여기서 "/"는 정수 나눗셈이다.
- [0152] *subpic_height_minus1*[*i*] + 1은 CtbSizeY 단위로 *i*-번째 서브화상의 높이를 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 *subpic_height_minus1*[*i*]의 값은 $\text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 과 동일한 것으로 추론된다. 여기서 "Ceil()"은 가장 가까운 정수로 반올림하는 함수이다. 따라서 $\text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 은 $(\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 과 동일하며, 여기서 "/"는 정수 나눗셈이다.
- [0153] 본 개시내용의 일부 실시예들에서, 마지막 서브화상의 위치션 및/또는 크기 정보는 스킵되고 전체 화상의 크기 및 모든 이전 서브화상의 크기 및 위치션으로부터 도출될 수 있다. 도 19는 본 개시내용의 일부 실시예들에 따라, 다른 예시적인 SPS 선택스를 도식하는 예시적인 표 9를 예시한다. 표 9(박스들(1901-1902)로 강조하여 보여지고 이탤릭체로 강조표시됨)에서, 선택스 요소 *sps_num_subpics_minus1*과 동일한 인덱스를 갖는 서브화상인 마지막 서브화상의 폭 및 높이가 스킵된다.
- [0154] 마지막 서브화상의 폭 및 높이는 전체 화상의 폭 및 높이, 및 마지막 서브화상의 상단-좌측 위치션으로부터 도출된다.
- [0155] 다음은 표 9의 선택스와 일치하는 시맨틱(이탤릭체로 강조됨)이다.
- [0156] *subpic_ctu_top_left_x*[*i*]는 CtbSizeY 단위로 *i*-번째 서브화상의 상단 좌측 CTU의 수평 위치션을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 *subpic_ctu_top_left_x*[*i*]의 값은 0과 동일한 것으로 추론된다.
- [0157] *subpic_ctu_top_left_y*[*i*]는 CtbSizeY 단위로 *i*-번째 서브화상의 상단 좌측 CTU의 수직 위치션을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 *subpic_ctu_top_left_y*[*i*]의 값은 0과 동일한 것으로 추론된다.
- [0158] *subpic_width_minus1*[*i*] + 1은 CtbSizeY 단위로 *i*-번째 서브화상의 폭을 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 *subpic_width_minus1*[*i*]의 값은 $\text{Ceil}((\text{pic_width_max_in_luma_samples}) \div \text{CtbSizeY}) - 1$ ($i == \text{sps_num_subpics_minus1} ? \text{subpic_ctu_top_left_x}[\text{sps_num_subpics_minus1}]:0$)와 동일한 것으로 추론된다. 여기서 "Ceil()"은 가장 가까운 정수로 반올림하는 함수이다. 즉, $\text{Ceil}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 은 $(\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 와 동일하며, 여기서 "/"는 정수 나눗셈이다. "*sps_num_subpics_minus1*"은 화상 내의 서브화상들의 수이다. 화상 내 마지막 서브화상의 경우, *i*는 *sps_num_subpics_minus1*과 동일하며, 이 경우에, *subpic_width_minus1*[*i*]은 $((\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1 - \text{subpic_ctu_top_left_x}[\text{sps_num_subpics_minus1}])$ 와 동일한 것으로 추론된다. 화상 내 단 하나의 서브화상만

있을 때, i 는 단지 0일 수 있고 $\text{subpic_ctu_top_left_x}[0]$ 는 0이다. 따라서, $\text{subpic_width_minus1}[i]$ 는 $(\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1 - \text{subpic_ctu_top_left_x}[0]$ 또는 $(\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 과 동일한 것으로 추론된다.

[0159] $\text{subpic_height_minus1}[i] + 1$ 은 CtbSizeY 단위로 i -번째 서브화상의 높이를 지정한다. 선택스 요소의 길이는 $\text{Ceil}(\log_2(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}))$ 비트이다. 존재하지 않을 때, 선택스 요소 $\text{subpic_height_minus1}[\text{sps_num_subpics_minus1}]$ 의 값은 $(\text{Ceil}(\text{pic_height_max_in_luma_samples}) \div \text{CtbSizeY}) - 1 - (i == \text{sps_num_subpics_minus1} ? \text{subpic_ctu_top_left_y}[i]:0)$ 와 동일한 것으로 추론된다. 여기서 "Ceil()"은 가장 가까운 정수로 반올림하는 함수이다. 즉, $\text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ 은 $(\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 와 동일하며, 여기서 "/"는 정수 나눗셈이다. "sps_num_subpics_minus1"은 화상 내의 서브화상들의 수이다. 화상 내 마지막 서브화상의 경우, i 는 $\text{sps_num_subpics_minus1}$ 과 동일하며, 이 경우에, $\text{subpic_height_minus1}[i]$ 은 $(\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1 - \text{subpic_ctu_top_left_y}[\text{sps_num_subpics_minus1}]$ 와 동일한 것으로 추론된다. 화상 내 단 하나의 서브화상만 있을 때, i 는 단지 0일 수 있고 $\text{subpic_ctu_top_left_y}[0]$ 는 0이다. 따라서, $\text{subpic_width_minus1}[i]$ 는 $(\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1 - \text{subpic_ctu_top_left_y}[0]$ 또는 $(\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) / \text{CtbSizeY} - 1$ 과 동일한 것으로 추론된다.

[0160] 본 개시내용의 일부 실시예들에서, 서브화상 ID는 서브화상이 존재할 때만 시그널링된다. 도 20은 본 개시내용의 일부 실시예들에 따라, 다른 예시적인 SPS 선택스(박스들(2001-2002)로 강조하여 보여지고 이탤릭체로 강조 표시됨)를 도시하는 예시적인 표 10을 예시한다.

[0161] 도 21은 본 개시내용의 일부 실시예들에 따라 예시적인 비디오 프로세싱 방법(2100)의 흐름도를 예시한다. 방법(2100)은 인코더(예컨대, 도 2a의 프로세서(200A) 또는 도 2b의 200B), 디코더(예컨대, 도 3a의 프로세서(300A) 또는 도 3b의 300B)에 의해 수행되거나 장치(예컨대, 도 4의 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 구성요소들에 의해 수행될 수 있다. 예컨대, 프로세서(예컨대, 도 4의 프로세서(402))는 방법(2100)을 수행할 수 있다. 일부 실시예들에서, 방법(2100)은 컴퓨터(예컨대, 도 4의 장치(400))에 의해 실행되는 컴퓨터 실행 가능 명령들 이블테면, 프로그램 코드를 포함하는 컴퓨터 판독 가능 매체에서 구체화되는 컴퓨터 프로그램 제품에 의해 구현될 수 있다.

[0162] 단계(2101)에서, 서브화상 ID 매핑이 비트스트림에 존재하는지에 관한 결정이 내려질 수 있다. 일부 실시예들에서, 방법(2100)은 서브화상 ID 매핑이 비트스트림에 존재하는지를 표시하는 플래그를 시그널링하는 것을 포함할 수 있다. 예컨대, 플래그는 도 11의 표 2, 도 13의 표 4, 도 15의 표 5, 또는 도 16의 표 6에 도시된 바와 같이 $\text{sps_subpic_id_present_flag}$ 일 수 있다.

[0163] 단계(2103)에서, 하나 이상의 서브화상 ID들이 제1 선택스 또는 제2 선택스에서 시그널링되는지에 관한 결정이 내려질 수 있다. 단계(2105)에서, 서브화상 ID 매핑이 존재하는 것으로 결정되고 하나 이상의 서브화상 ID들이 제1 선택스 및 제2 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링된다. 제1 선택스, 제2 선택스 또는 제3 선택스는 SPS, PPS 및 PH 중 하나이다. 예컨대, 제1 선택스, 제2 선택스, 및 제3 선택스는 각각 SPS, PPS 및 PH이다. 따라서, 서브화상 ID 매핑이 존재할 때(예컨대, $\text{sps_subpic_id_present_flag} = 1$), 서브화상 ID들은 SPS, PPS 또는 PH 중 어느 하나에서 시그널링되도록 강제될 수 있다.

[0164] 일부 실시예들에서, 방법(2100)은 하나 이상의 서브화상 ID들이 제1 선택스(예컨대, 도 11의 표 2에 도시된 SPS)에서 시그널링됨을 표시하는 제1 플래그(예컨대, 도 11의 표 2, 도 15의 표 5 또는 도 16의 표 6에 도시된 바와 같은 $\text{sps_subpic_id_signaling_present_flag}$)를 시그널링하는 것을 포함할 수 있다. 일부 실시예들에서, 방법(2100)은 하나 이상의 서브화상 ID들이 제2 선택스(예컨대, 도 12의 표 3에 도시된 PPS)에서 시그널링됨을 표시하는 제2 플래그(예컨대, 도 12의 표 3, 도 15의 표 5 또는 도 16의 표 6에 도시된 바와 같은 $\text{pps_subpic_id_signaling_present_flag}$)를 시그널링하는 것을 포함할 수 있다. 일부 실시예들에서, 방법(2100)은 하나 이상의 서브화상 ID들이 제3 선택스(예컨대, 도 15의 표 5에 도시된 PH)에서 시그널링됨을 표시하는 제3 플래그(예컨대, 도 15의 표 5에 도시된 바와 같은 $\text{ph_subpic_id_signaling_present_flag}$)를 시그널링하는 것을 포함할 수 있다.

[0165] 일부 실시예들에서, 방법(2100)은 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링됨을 표시하는 제3 플래그를 비트스트림이 포함하는지를 결정하는 것, 그리고 비트스트림이 제3 플래그를 포함하지 않는다는 것에 대

한 응답으로, 하나 이상의 서브화상 ID들을 제3 선택스에서 시그널링하는 것을 포함할 수 있다. 예컨대, 제3 선택스는 PH일 수 있고 제3 플래그는 `ph_subpic_id_signaling_present_flag`일 수 있다. `ph_subpic_id_signaling_present_flag`가 PH에서 시그널링되지 않을 때, 그것은 1인 것으로 추론될 수 있고 하나 이상의 서브화상 ID들은 PH에서 시그널링된다.

- [0166] 일부 실시예들에서, 방법(2100)은 하나 이상의 서브화상 ID들이 제1 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 제2 선택스 및 제3 선택스(예컨대, 도 15의 표 5)에서 하나 이상의 서브화상 ID들을 시그널링하는 것을 포함할 수 있다.
- [0167] 도 22는 본 개시내용의 일부 실시예들에 따라 예시적인 비디오 프로세싱 방법(2200)의 흐름도를 예시한다. 방법(2200)은 인코더(예컨대, 도 2a의 프로세스(200A) 또는 도 2b의 200B), 디코더(예컨대, 도 3a의 프로세스(300A) 또는 도 3b의 300B)에 의해 수행되거나 장치(예컨대, 도 4의 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 구성요소들에 의해 수행될 수 있다. 예컨대, 프로세서(예컨대, 도 4의 프로세서(402))는 방법(2200)을 수행할 수 있다. 일부 실시예들에서, 방법(2200)은 컴퓨터(예컨대, 도 4의 장치(400))에 의해 실행되는 컴퓨터 실행 가능 명령들 이블테면, 프로그램 코드를 포함하는 컴퓨터 판독 가능 매체에서 구체화되는 컴퓨터 프로그램 제품에 의해 구현될 수 있다.
- [0168] 단계(2201)에서, 하나 이상의 서브화상 ID들이 SPS, PH, 또는 PPS 중 적어도 하나에서 시그널링되는지에 관한 결정이 내려질 수 있다. 일부 실시예들에서, 방법(2200)은 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하기 전에 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하는 것을 포함할 수 있다. 일부 실시예들에서, 방법(2200)은 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하기 전에 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하는 것을 포함할 수 있다.
- [0169] 단계(2203)에서, 하나 이상의 서브화상 ID들이 SPS, PH 및 PPS에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 하나 이상의 서브화상 ID들은 디폴트 값을 갖는 것으로 결정될 수 있다.
- [0170] 도 23은 본 개시내용의 일부 실시예들에 따라 예시적인 비디오 프로세싱 방법(2300)의 흐름도를 예시한다. 방법(2300)은 인코더(예컨대, 도 2a의 프로세스(200A) 또는 도 2b의 200B), 디코더(예컨대, 도 3a의 프로세스(300A) 또는 도 3b의 300B)에 의해 수행되거나 장치(예컨대, 도 4의 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 구성요소들에 의해 수행될 수 있다. 예컨대, 프로세서(예컨대, 도 4의 프로세서(402))는 방법(2300)을 수행할 수 있다. 일부 실시예들에서, 방법(2300)은 컴퓨터(예컨대, 도 4의 장치(400))에 의해 실행되는 컴퓨터 실행 가능 명령들 이블테면, 프로그램 코드를 포함하는 컴퓨터 판독 가능 매체에서 구체화되는 컴퓨터 프로그램 제품에 의해 구현될 수 있다.
- [0171] 단계(2301)에서, 코딩된 화상의 서브화상 번호가 1과 동일한지에 관한 결정이 내려질 수 있다. 예컨대, 도 18의 표 8에 도시된 바와 같이, `sps_num_subpic_minus1`이 0보다 큰지에 관한 결정이 내려질 수 있다.
- [0172] 단계(2303)에서, 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 코딩된 화상의 서브화상은 디코딩 프로세스에서 화상으로서 처리될 수 있다. 예컨대, 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 플래그 `subpic_treated_as_pic_flag[i]`는 1과 동일한 것으로 추론될 수 있다. 일부 실시예들에서, 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 인-루프 필터링 동작들이 제외될 수 있다.
- [0173] 도 24는 본 개시내용의 일부 실시예들에 따라 예시적인 비디오 프로세싱 방법(2400)의 흐름도를 예시한다. 방법(2400)은 인코더(예컨대, 도 2a의 프로세스(200A) 또는 도 2b의 200B), 디코더(예컨대, 도 3a의 프로세스(300A) 또는 도 3b의 300B)에 의해 수행되거나 장치(예컨대, 도 4의 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 구성요소들에 의해 수행될 수 있다. 예컨대, 프로세서(예컨대, 도 4의 프로세서(402))는 방법(2400)을 수행할 수 있다. 일부 실시예들에서, 방법(2400)은 컴퓨터(예컨대, 도 4의 장치(400))에 의해 실행되는 컴퓨터 실행 가능 명령들 이블테면, 프로그램 코드를 포함하는 컴퓨터 판독 가능 매체에서 구체화되는 컴퓨터 프로그램 제품에 의해 구현될 수 있다.
- [0174] 단계(2401)에서, 서브화상이 화상의 마지막 서브화상인지에 관한 결정이 내려질 수 있다. 단계(2403)에서, 서브화상이 마지막 서브화상인 것으로 결정되는 것에 대한 응답으로, 서브화상의 포지션 또는 크기 정보는 화상의 크기 및 화상의 이전 서브화상들의 크기들 및 포지션들로부터 도출될 수 있다.
- [0175] 도 25는 본 개시내용의 일부 실시예들에 따라 예시적인 비디오 프로세싱 방법(2500)의 흐름도를 예시한다. 방법(2500)은 인코더(예컨대, 도 2a의 프로세스(200A) 또는 도 2b의 200B), 디코더(예컨대, 도 3a의 프로세스

(300A) 또는 도 3b의 300B)에 의해 수행되거나 장치(예컨대, 도 4의 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 구성요소들에 의해 수행될 수 있다. 예컨대, 프로세서(예컨대, 도 4의 프로세서(402))는 방법(2500)을 수행할 수 있다. 일부 실시예들에서, 방법(2500)은 컴퓨터(예컨대, 도 4의 장치(400))에 의해 실행되는 컴퓨터 실행 가능 명령들 이블테면, 프로그램 코드를 포함하는 컴퓨터 판독 가능 매체에서 구체화되는 컴퓨터 프로그램 제품에 의해 구현될 수 있다.

- [0176] 단계(2501)에서, 서브화상이 화상에 존재하는지에 관한 결정이 내려질 수 있다. 예컨대, 이러한 결정은 플래그(예컨대, 도 20의 표 10에 도시된 바와 같은 `subpics_present_flag`)에 기초하여 내려질 수 있다.
- [0177] 단계(2503)에서, 하나 이상의 서브화상들이 화상에 존재하는 것으로 결정되는 것에 대한 응답으로, 서브화상 ID 매핑이 SPS에 존재하는지를 표시하는 제1 플래그가 시그널링될 수 있다. 예컨대, 제1 플래그는 도 20의 표 10에 도시된 바와 같은 `sps_subpic_id_present_flag`일 수 있다.
- [0178] 일부 실시예들에서, 방법(2500)은 서브화상 ID 매핑이 SPS에 존재한다는 것을 제1 플래그가 표시하는 것에 대한 응답으로, 서브화상 ID 매핑이 SPS에서 시그널링되는지를 표시하는 제2 플래그를 시그널링하는 것을 포함할 수 있다. 서브화상 ID 매핑이 SPS에서 시그널링됨을 제2 플래그가 표시하는 것에 대한 응답으로, 하나 이상의 서브화상들의 서브화상 ID들이 SPS에서 시그널링될 수 있다. 예컨대, 제2 플래그는 도 20의 표 10에 도시된 바와 같이 `sps_subpic_id_signaling_present_flag`일 수 있다. `sps_subpic_id_signaling_present_flag`가 참일 때, `sps_subpic_id[ i ]`가 시그널링될 수 있다.
- [0179] 실시예들은 다음 조항(clauses)을 사용하여 추가로 설명될 수 있다:
- [0180] 1. 비디오 프로세싱 방법으로서,
- [0181] 서브화상 ID 매핑이 비트스트림에 존재하는지를 결정하는 단계;
- [0182] 하나 이상의 서브화상 ID들이 제1 선택스 또는 제2 선택스에서 시그널링되는지를 결정하는 단계; 및
- [0183] 서브화상 ID 매핑이 존재하는 것으로 결정되고 하나 이상의 서브화상 ID들이 제1 선택스 및 제2 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 단계를 포함하는, 비디오 프로세싱 방법.
- [0184] 2. 조항 1에 있어서,
- [0185] 제1 선택스, 제2 선택스, 또는 제3 선택스는 시퀀스 파라미터 세트(SPS), 화상 파라미터 세트(PPS), 및 화상 헤더(PH) 중 하나인, 비디오 프로세싱 방법.
- [0186] 3. 조항 1 및 조항 2 중 어느 한 조항에 있어서,
- [0187] 하나 이상의 서브화상 ID들이 제1 선택스에서 시그널링됨을 표시하는 제1 플래그를 시그널링하는 단계; 또는
- [0188] 하나 이상의 서브화상 ID들이 제2 선택스에서 시그널링됨을 표시하는 제2 플래그를 시그널링하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0189] 4. 조항 1 내지 조항 3 중 어느 한 조항에 있어서,
- [0190] 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링됨을 표시하는 제3 플래그를 시그널링하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0191] 5. 조항 1 내지 조항 4 중 어느 한 조항에 있어서,
- [0192] 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링됨을 표시하는 제3 플래그를 비트스트림이 포함하는지를 결정하는 단계; 및
- [0193] 비트스트림이 제3 플래그를 포함하지 않는 것에 대한 응답으로, 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0194] 6. 조항 1 내지 조항 5 중 어느 한 조항에 있어서,
- [0195] 하나 이상의 서브화상 ID들이 제1 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 제2 선택스 및 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 단계를 더 포함하는, 비디오 프로세싱 방법.

- [0196] 7. 조항 1 내지 조항 6 중 어느 한 조항에 있어서,
- [0197] 서브화상 ID 매핑이 비트스트림에 존재하는지를 표시하는 제4 플래그를 시그널링하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0198] 8. 비디오 프로세싱 장치로서,
- [0199] 명령들을 저장하기 위한 적어도 하나의 메모리; 및
- [0200] 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고,
- [0201] 명령들은, 장치로 하여금,
- [0202] 서브화상 ID 매핑이 비트스트림에 존재하는지를 결정하는 것;
- [0203] 하나 이상의 서브화상 ID들이 제1 선택스 또는 제2 선택스에서 시그널링되는지를 결정하는 것; 및
- [0204] 서브화상 ID 매핑이 존재하는 것으로 결정되고 하나 이상의 서브화상 ID들이 제1 선택스 및 제2 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 것을 수행하게 하기 위한 것인, 비디오 프로세싱 장치.
- [0205] 9. 조항 8에 있어서,
- [0206] 제1 선택스, 제2 선택스, 또는 제3 선택스는 시퀀스 파라미터 세트(СПS), 화상 파라미터 세트(PPS), 및 화상 헤더(PH) 중 하나인, 비디오 프로세싱 장치.
- [0207] 10. 조항 8 및 조항 9 중 어느 한 조항에 있어서,
- [0208] 적어도 하나의 프로세서는 장치로 하여금,
- [0209] 하나 이상의 서브화상 ID들이 제1 선택스에서 시그널링됨을 표시하는 제1 플래그를 시그널링하는 것; 또는
- [0210] 하나 이상의 서브화상 ID들이 제2 선택스에서 시그널링됨을 표시하는 제2 플래그를 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0211] 11. 조항 8 내지 조항 10 중 어느 한 조항에 있어서,
- [0212] 적어도 하나의 프로세서는 장치로 하여금,
- [0213] 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링됨을 표시하는 제3 플래그를 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0214] 12. 조항 8 내지 조항 11 중 어느 한 조항에 있어서,
- [0215] 적어도 하나의 프로세서는 장치로 하여금,
- [0216] 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링됨을 표시하는 제3 플래그를 비트스트림이 포함하는지를 결정하는 것; 및
- [0217] 비트스트림이 제3 플래그를 포함하지 않는 것에 대한 응답으로, 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0218] 13. 조항 8 내지 조항 12 중 어느 한 조항에 있어서,
- [0219] 적어도 하나의 프로세서는 장치로 하여금,
- [0220] 하나 이상의 서브화상 ID들이 제1 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 제2 선택스 및 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0221] 14. 조항 8 내지 조항 13 중 어느 한 조항에 있어서,
- [0222] 적어도 하나의 프로세서는 장치로 하여금,
- [0223] 서브화상 ID 매핑이 비트스트림에 존재하는지를 표시하는 제4 플래그를 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.

- [0224] 15. 명령들의 세트를 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체로서,
- [0225] 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 방법을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하고, 방법은,
- [0226] 서브화상 ID 매핑이 비트스트림에 존재하는지를 결정하는 단계;
- [0227] 하나 이상의 서브화상 ID들이 제1 선택스 또는 제2 선택스에서 시그널링되는지를 결정하는 단계; 및
- [0228] 서브화상 ID 매핑이 존재하는 것으로 결정되고 하나 이상의 서브화상 ID들이 제1 선택스 및 제2 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 단계를 포함하는, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0229] 16. 조항 15에 있어서,
- [0230] 제1 선택스, 제2 선택스, 또는 제3 선택스는 시퀀스 파라미터 세트(SPS), 화상 파라미터 세트(PPS), 및 화상 헤더(PH) 중 하나인, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0231] 17. 조항 15 및 조항 16 중 어느 한 조항에 있어서,
- [0232] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0233] 하나 이상의 서브화상 ID들이 제1 선택스에서 시그널링됨을 표시하는 제1 플래그를 시그널링하는 것; 또는
- [0234] 하나 이상의 서브화상 ID들이 제2 선택스에서 시그널링됨을 표시하는 제2 플래그를 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0235] 18. 조항 15 내지 조항 17 중 어느 한 조항에 있어서,
- [0236] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0237] 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링됨을 표시하는 제3 플래그를 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0238] 19. 조항 15 내지 조항 18 중 어느 한 조항에 있어서,
- [0239] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0240] 하나 이상의 서브화상 ID들이 제3 선택스에서 시그널링됨을 표시하는 제3 플래그를 비트스트림이 포함하는지를 결정하는 것; 및
- [0241] 비트스트림이 제3 플래그를 포함하지 않는 것에 대한 응답으로, 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0242] 20. 조항 15 내지 조항 19 중 어느 한 조항에 있어서,
- [0243] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0244] 하나 이상의 서브화상 ID들이 제1 선택스에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 제2 선택스 및 제3 선택스에서 하나 이상의 서브화상 ID들을 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0245] 21. 조항 15 내지 조항 20 중 어느 한 조항에 있어서,
- [0246] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0247] 서브화상 ID 매핑이 비트스트림에 존재하는지를 표시하는 제4 플래그를 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0248] 22. 비디오 프로세싱 방법으로서,
- [0249] 하나 이상의 서브화상 ID들이 시퀀스 파라미터 세트(SPS), 화상 헤더(PH), 또는 화상 파라미터 세트(PPS) 중 적어도 하나에서 시그널링되는지를 결정하는 단계; 및
- [0250] 하나 이상의 서브화상 ID들이 SPS, PH 및 PPS에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 하

나 이상의 서브화상 ID들은 디폴트 값을 갖는 것으로 결정하는 단계를 포함하는, 비디오 프로세싱 방법.

- [0251] 23. 조항 22에 있어서,
- [0252] 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하는 단계는 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하기 전에 수행되는, 비디오 프로세싱 방법.
- [0253] 24. 조항 22에 있어서,
- [0254] 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하는 단계는 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하기 전에 수행되는, 비디오 프로세싱 방법.
- [0255] 25. 비디오 프로세싱 방법으로서,
- [0256] 코딩된 화상의 서브화상 번호가 1과 동일한지를 결정하는 단계; 및
- [0257] 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 디코딩 프로세스에서 화상으로서 코딩된 화상의 서브화상을 처리하는 단계를 포함하는, 비디오 프로세싱 방법.
- [0258] 26. 조항 25에 있어서,
- [0259] 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 인-루프 필터링 동작들을 제외시키는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0260] 27. 비디오 프로세싱 방법으로서,
- [0261] 서브화상이 화상의 마지막 서브화상인지를 결정하는 단계; 및
- [0262] 서브화상이 마지막 서브화상인 것으로 결정되는 것에 대한 응답으로, 화상의 크기 및 화상의 이전 서브화상들의 크기들 및 포지션들로부터 서브화상의 포지션 또는 크기 정보를 도출하는 단계를 포함하는, 비디오 프로세싱 방법.
- [0263] 28. 비디오 프로세싱 방법으로서,
- [0264] 화상에 서브화상이 존재하는지를 결정하는 단계; 및
- [0265] 하나 이상의 서브화상들이 화상에 존재하는 것으로 결정되는 것에 대한 응답으로, 서브화상 ID 매핑이 시퀀스 파라미터 세트(SPS)에 존재하는지를 표시하는 제1 플래그를 시그널링하는 단계를 포함하는, 비디오 프로세싱 방법.
- [0266] 29. 조항 28에 있어서,
- [0267] 서브화상 ID 매핑이 SPS에 존재한다는 것을 제1 플래그가 표시하는 것에 대한 응답으로, 서브화상 ID 매핑이 SPS에서 시그널링되는지를 표시하는 제2 플래그를 시그널링하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0268] 30. 조항 29에 있어서,
- [0269] 서브화상 ID 매핑이 SPS에서 시그널링됨을 제2 플래그가 표시하는 것에 대한 응답으로, SPS에서 하나 이상의 서브화상들의 서브화상 ID들을 시그널링하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0270] 31. 비디오 프로세싱 장치로서,
- [0271] 명령들을 저장하기 위한 적어도 하나의 메모리; 및
- [0272] 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 명령들은, 장치로 하여금,
- [0273] 하나 이상의 서브화상 ID들이 시퀀스 파라미터 세트(SPS), 화상 헤더(PH), 또는 화상 파라미터 세트(PPS) 중 적어도 하나에서 시그널링되는지를 결정하는 것; 및
- [0274] 하나 이상의 서브화상 ID들이 SPS, PH 및 PPS에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 하나 이상의 서브화상 ID들은 디폴트 값을 갖는 것으로 결정하는 것을 수행하게 하기 위한 것인, 비디오 프로세싱 장치.
- [0275] 32. 조항 31에 있어서,
- [0276] 적어도 하나의 프로세서는 장치로 하여금,

- [0277] 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하기 전에 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0278] 33. 조항 31에 있어서,
- [0279] 적어도 하나의 프로세서는 장치로 하여금,
- [0280] 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하기 전에 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0281] 34. 비디오 프로세싱 장치로서,
- [0282] 명령들을 저장하기 위한 적어도 하나의 메모리; 및
- [0283] 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 명령들은, 장치로 하여금,
- [0284] 코딩된 화상의 서브화상 번호가 1과 동일한지를 결정하는 것; 및
- [0285] 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 디코딩 프로세스에서 화상으로서 코딩된 화상의 서브화상을 처리하는 것을 수행하게 하기 위한 것인, 비디오 프로세싱 장치.
- [0286] 35. 조항 34에 있어서,
- [0287] 적어도 하나의 프로세서는 장치로 하여금,
- [0288] 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 인-루프 필터링 동작들을 제외시키는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0289] 36. 비디오 프로세싱 장치로서,
- [0290] 명령들을 저장하기 위한 적어도 하나의 메모리; 및
- [0291] 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 명령들은, 장치로 하여금,
- [0292] 서브화상이 화상의 마지막 서브화상인지를 결정하는 것; 및
- [0293] 서브화상이 마지막 서브화상인 것으로 결정되는 것에 대한 응답으로, 화상의 크기 및 화상의 이전 서브화상들의 크기들 및 포지션들로부터 서브화상의 포지션 또는 크기 정보를 도출하는 것을 수행하게 하기 위한 것인, 비디오 프로세싱 장치.
- [0294] 37. 비디오 프로세싱 장치로서,
- [0295] 명령들을 저장하기 위한 적어도 하나의 메모리; 및
- [0296] 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 명령들은, 장치로 하여금,
- [0297] 화상에 서브화상이 존재하는지를 결정하는 것; 및
- [0298] 하나 이상의 서브화상들이 화상에 존재하는 것으로 결정되는 것에 대한 응답으로, 서브화상 ID 매핑이 시퀀스 파라미터 세트(SPS)에 존재하는지를 표시하는 제1 플래그를 시그널링하는 것을 수행하게 하기 위한 것인, 비디오 프로세싱 장치.
- [0299] 38. 조항 37에 있어서,
- [0300] 적어도 하나의 프로세서는 장치로 하여금,
- [0301] 서브화상 ID 매핑이 SPS에 존재한다는 것을 제1 플래그가 표시하는 것에 대한 응답으로, 서브화상 ID 매핑이 SPS에서 시그널링되는지를 표시하는 제2 플래그를 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0302] 39. 조항 38에 있어서,
- [0303] 적어도 하나의 프로세서는 장치로 하여금,
- [0304] 서브화상 ID 매핑이 SPS에서 시그널링됨을 제2 플래그가 표시하는 것에 대한 응답으로, SPS에서 하나 이상의 서브화상들의 서브화상 ID들을 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프

로세싱 장치.

- [0305] 40. 명령들의 세트를 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체로서, 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 방법을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하고,
- [0306] 방법은,
- [0307] 하나 이상의 서브화상 ID들이 시퀀스 파라미터 세트(PS), 화상 헤더(PH), 또는 화상 파라미터 세트(PPS) 중 적어도 하나에서 시그널링되는지를 결정하는 단계; 및
- [0308] 하나 이상의 서브화상 ID들이 PS, PH 및 PPS에서 시그널링되지 않는 것으로 결정되는 것에 대한 응답으로, 하나 이상의 서브화상 ID들은 디폴트 값을 갖는 것으로 결정하는 단계를 포함하는, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0309] 41. 조항 40에 있어서,
- [0310] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0311] 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하기 전에 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0312] 42. 조항 40에 있어서,
- [0313] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0314] 하나 이상의 서브화상 ID들이 PH에서 시그널링되는지를 결정하기 전에 하나 이상의 서브화상 ID들이 PPS에서 시그널링되는지를 결정하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0315] 43. 명령들의 세트를 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체로서,
- [0316] 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 방법을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하고,
- [0317] 방법은,
- [0318] 코딩된 화상의 서브화상 번호가 1과 동일한지를 결정하는 단계; 및
- [0319] 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 디코딩 프로세스에서 화상으로서 코딩된 화상의 서브화상을 처리하는 단계를 포함하는, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0320] 44. 조항 43에 있어서,
- [0321] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0322] 서브화상 번호가 1과 동일한 것으로 결정되는 것에 대한 응답으로, 인-루프 필터링 동작들을 제외시키는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0323] 45. 명령들의 세트를 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체로서,
- [0324] 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 방법을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하고,
- [0325] 방법은,
- [0326] 서브화상이 화상의 마지막 서브화상인지를 결정하는 단계; 및
- [0327] 서브화상이 마지막 서브화상인 것으로 결정되는 것에 대한 응답으로, 화상의 크기 및 화상의 이전 서브화상들의 크기들 및 포지션들로부터 서브화상의 포지션 또는 크기 정보를 도출하는 단계를 포함하는, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0328] 46. 명령들의 세트를 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체로서,

- [0329] 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 방법을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하고,
- [0330] 방법은,
- [0331] 화상에 서브화상이 존재하는지를 결정하는 단계; 및
- [0332] 하나 이상의 서브화상들이 화상에 존재하는 것으로 결정되는 것에 대한 응답으로, 서브화상 ID 매핑이 시퀀스 파라미터 세트(SPS)에 존재하는지를 표시하는 제1 플래그를 시그널링하는 단계를 포함하는, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0333] 47. 조항 46에 있어서,
- [0334] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0335] 서브화상 ID 매핑이 SPS에 존재한다는 것을 제1 플래그가 표시하는 것에 대한 응답으로, 서브화상 ID 매핑이 SPS에서 시그널링되는지를 표시하는 제2 플래그를 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0336] 48. 조항 47에 있어서,
- [0337] 명령들의 세트는, 비디오 프로세싱 장치로 하여금,
- [0338] 서브화상 ID 매핑이 SPS에서 시그널링됨을 제2 플래그가 표시하는 것에 대한 응답으로, SPS에서 하나 이상의 서브화상들의 서브화상 ID들을 시그널링하는 것을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능한, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0339] 49. 비디오 프로세싱 방법으로서,
- [0340] 비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 비트스트림이 서브화상 정보를 포함하는지를 결정하는 단계; 및
- [0341] 비트스트림이 서브화상 정보를 포함하는 것에 대한 응답으로,
- [0342] 화상 내 서브화상들의 수,
- [0343] 타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑,
- [0344] subpic_treated_as_pic_flag, 및
- [0345] loop_filter_across_subpic_enabled_flag 중 적어도 하나를 비트스트림에서 시그널링하는 단계를 포함하는, 비디오 프로세싱 방법.
- [0346] 50. 조항 49에 있어서,
- [0347] 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 화상 내 서브화상들의 수에 기초하는, 비디오 프로세싱 방법.
- [0348] 51. 조항 50에 있어서,
- [0349] 화상에 적어도 2개의 서브화상들이 있을 때, subpic_treated_as_pic_flag, loop_filter_across_subpic_enabled_flag, 및 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나를 시그널링하는 단계를 더 포함하고,
- [0350] 화상에 단 하나의 서브화상만이 있을 때, 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 스킵되고,
- [0351] subpic_treated_as_pic_flag는 코딩 층 비디오 시퀀스(CLVs) 내 각각의 코딩된 화상의 서브화상들이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되는지를 표시하고, 그리고
- [0352] loop_filter_across_subpic_enabled_flag는 CLVS 내 각각의 코딩된 화상의 서브화상 경계들에 걸쳐 서브화상 경계들에 걸친 인-루프 필터링 동작들이 인에이블되는지를 표시하는, 비디오 프로세싱 방법.
- [0353] 52. 조항 51에 있어서,

- [0354] 타겟 서브화상의 폭이 시그널링되지 않을 때, 화상의 폭으로서 타겟 서브화상의 폭의 값을 결정하는 단계; 및
- [0355] 타겟 서브화상의 높이가 시그널링되지 않을 때, 화상의 높이로서 타겟 서브화상의 높이의 값을 결정하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0356] 53. 조항 52에 있어서,
- [0357] 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 화상의 폭으로서 CTB 크기 단위의 타겟 서브화상의 폭의 값을 결정하는 단계; 및
- [0358] 타겟 서브화상의 높이가 시그널링되지 않을 때, CTB 크기 단위의 화상의 높이로서 CTB 크기 단위의 타겟 서브화상의 높이의 값을 결정하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0359] 54. 조항 51에 있어서,
- [0360] subpic_treated_as_pic_flag가 비트스트림에서 시그널링되지 않을 때, subpic_treated_as_pic_flag가 1의 값을 갖는 것으로 결정하는 단계; 및
- [0361] loop_filter_across_subpic_enabled_flag가 비트스트림에서 시그널링되지 않을 때, loop_filter_across_subpic_enabled_flag가 0의 값을 갖는 것으로 결정하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0362] 55. 조항 49에 있어서,
- [0363] 타겟 서브화상이 화상에서 마지막 서브화상일 때, 타겟 서브화상의 폭 및 높이 중 적어도 하나를 시그널링하는 것을 스킵하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0364] 56. 조항 55에 있어서,
- [0365] 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 화상의 폭에서 CTB 크기 단위의 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 위치션을 뺀 것으로서 CTB 크기 단위의 타겟 서브화상의 폭의 값을 결정하거나, 화상의 폭에서 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 위치션을 뺀 것으로서 타겟 서브화상의 폭의 값을 결정하는 단계; 및
- [0366] 타겟 서브화상의 높이가 시그널링되지 않을 때, CTB 크기 단위의 화상의 높이에서 CTB 크기 단위의 타겟 서브화상의 상단 좌측 CTU의 수직 위치션을 뺀 것으로서 CTB 크기 단위의 타겟 서브화상의 높이의 값을 결정하거나, 화상 높이에서 타겟 서브화상의 상단 좌측 CTU의 수직 위치션을 뺀 것으로서 타겟 서브화상의 높이의 값을 결정하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0367] 57. 조항 49에 있어서,
- [0368] 타겟 서브화상의 ID 매핑을 시그널링하는 단계는,
- [0369] 비트스트림에서 제1 플래그를 시그널링하는 단계; 및
- [0370] 제1 플래그가 1과 동일한 것에 대한 응답으로, 제1 데이터 유닛 또는 제2 데이터 유닛에서 타겟 서브화상의 ID 매핑을 시그널링하는 단계를 더 포함하고,
- [0371] 0과 동일한 제1 플래그는 타겟 서브화상의 ID 매핑이 비트스트림에서 시그널링되지 않음을 표시하는, 비디오 프로세싱 방법.
- [0372] 58. 조항 57에 있어서,
- [0373] 제1 플래그가 1과 동일하고 타겟 서브화상의 ID 매핑이 제1 데이터 유닛에서 시그널링되지 않는 것에 대한 응답으로, 제2 데이터 유닛에서 타겟 서브화상의 ID 매핑을 시그널링하는 단계; 또는
- [0374] 제1 플래그가 0과 동일하거나 타겟 서브화상의 ID 매핑이 제1 데이터 유닛에서 시그널링되는 것에 대한 응답으로, 제2 데이터 유닛에서 타겟 서브화상의 ID 매핑을 시그널링하는 것을 스킵하는 단계를 더 포함하는, 비디오 프로세싱 방법.
- [0375] 59. 조항 58에 있어서,
- [0376] 제1 데이터 유닛 및 제2 데이터 유닛 각각은 시퀀스 파라미터 세트(SPS), 화상 파라미터 세트(PPS), 또는 화상

헤더(PH) 중 하나인, 비디오 프로세싱 방법.

- [0377] 60. 비디오 프로세싱 장치로서,
- [0378] 명령들을 저장하기 위한 적어도 하나의 메모리; 및
- [0379] 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고,
- [0380] 명령들은, 장치로 하여금,
- [0381] 비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 비트스트림이 서브화상 정보를 포함하는지를 결정하는 것; 및
- [0382] 비트스트림이 서브화상 정보를 포함하는 것에 대한 응답으로,
- [0383] 화상 내 서브화상들의 수,
- [0384] 타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑,
- [0385] subpic_treated_as_pic_flag, 및
- [0386] loop_filter_across_subpic_enabled_flag 중 적어도 하나를 비트스트림에서 시그널링하는 것을 수행하게 하기 위한 것인, 비디오 프로세싱 장치.
- [0387] 61. 조항 60에 있어서,
- [0388] 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 화상 내 서브화상들의 수에 기초하는, 비디오 프로세싱 장치.
- [0389] 62. 조항 61에 있어서,
- [0390] 적어도 하나의 프로세서는 장치로 하여금,
- [0391] 화상에 적어도 2개의 서브화상들이 있을 때, subpic_treated_as_pic_flag, loop_filter_across_subpic_enabled_flag, 및 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나를 시그널링하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되고,
- [0392] 화상에 단 하나의 서브화상만이 있을 때, 타겟 서브화상의 폭, 높이 및 포지션 중 적어도 하나의 시그널링은 스킵되고,
- [0393] subpic_treated_as_pic_flag는 코딩 층 비디오 시퀀스(CLVs) 내 각각의 코딩된 화상의 서브화상들이 인-루프 필터링 동작들을 제외한 디코딩 프로세스에서 화상으로서 처리되는지를 표시하고, 그리고
- [0394] loop_filter_across_subpic_enabled_flag는 CLVS 내 각각의 코딩된 화상의 서브화상 경계들에 걸쳐 서브화상 경계들에 걸친 인-루프 필터링 동작들이 인에이블되는지를 표시하는, 비디오 프로세싱 장치.
- [0395] 63. 조항 62에 있어서,
- [0396] 적어도 하나의 프로세서는 장치로 하여금,
- [0397] 타겟 서브화상의 폭이 시그널링되지 않을 때, 화상의 폭으로서 타겟 서브화상의 폭의 값을 결정하는 것; 및
- [0398] 타겟 서브화상의 높이가 시그널링되지 않을 때, 화상의 높이로서 타겟 서브화상의 높이의 값을 결정하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0399] 64. 조항 63에 있어서,
- [0400] 적어도 하나의 프로세서는 장치로 하여금,
- [0401] 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 화상의 폭으로서 CTB 크기 단위의 타겟 서브화상의 폭의 값을 결정하는 것; 및
- [0402] 타겟 서브화상의 높이가 시그널링되지 않을 때, CTB 크기 단위의 화상의 높이로서 CTB 크기 단위의 타겟 서브화상의 높이의 값을 결정하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0403] 65. 조항 62에 있어서,

- [0404] 적어도 하나의 프로세서는 장치로 하여금,
- [0405] subpic_treated_as_pic_flag가 비트스트림에서 시그널링되지 않을 때, subpic_treated_as_pic_flag가 1의 값을 갖는 것으로 결정하는 것; 및
- [0406] loop_filter_across_subpic_enabled_flag가 비트스트림에서 시그널링되지 않을 때, loop_filter_across_subpic_enabled_flag가 0의 값을 갖는 것으로 결정하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0407] 66. 조항 60에 있어서,
- [0408] 적어도 하나의 프로세서는 장치로 하여금,
- [0409] 타겟 서브화상이 화상에서 마지막 서브화상일 때, 타겟 서브화상의 폭 및 높이 중 적어도 하나를 시그널링하는 것을 스킵하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0410] 67. 조항 66에 있어서,
- [0411] 적어도 하나의 프로세서는 장치로 하여금,
- [0412] 타겟 서브화상의 폭이 시그널링되지 않을 때, 코딩 트리 블록(CTB) 크기 단위의 화상의 폭에서 CTB 크기 단위의 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 포지션을 뺀 것으로서 CTB 크기 단위의 타겟 서브화상의 폭의 값을 결정하거나, 화상의 폭에서 타겟 서브화상의 상단 좌측 코딩 트리 유닛(CTU)의 수평 포지션을 뺀 것으로서 타겟 서브화상의 폭의 값을 결정하는 것; 및
- [0413] 타겟 서브화상의 높이가 시그널링되지 않을 때, CTB 크기 단위의 화상의 높이에서 CTB 크기 단위의 타겟 서브화상의 상단 좌측 CTU의 수직 포지션을 뺀 것으로서 CTB 크기 단위의 타겟 서브화상의 높이의 값을 결정하거나, 화상 높이에서 타겟 서브화상의 상단 좌측 CTU의 수직 포지션을 뺀 것으로서 타겟 서브화상의 높이의 값을 결정하는 것을 수행하게 하기 위한 명령들을 실행하도록 구성되는, 비디오 프로세싱 장치.
- [0414] 68. 명령들의 세트를 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체로서,
- [0415] 명령들의 세트는, 비디오 프로세싱 장치로 하여금, 방법을 수행하게 하도록 하나 이상의 프로세싱 디바이스들에 의해 실행 가능하고,
- [0416] 방법은,
- [0417] 비트스트림에서 시그널링되는 서브화상 정보 존재 플래그에 따라, 비트스트림이 서브화상 정보를 포함하는지를 결정하는 단계; 및
- [0418] 비트스트림이 서브화상 정보를 포함하는 것에 대한 응답으로,
- [0419] 화상 내 서브화상들의 수,
- [0420] 타겟 서브화상의 폭, 높이, 포지션 및 식별자(ID) 매핑,
- [0421] subpic_treated_as_pic_flag, 및
- [0422] loop_filter_across_subpic_enabled_flag 중 적어도 하나를 비트스트림에서 시그널링하는 단계를 포함하는, 비-일시적 컴퓨터 판독 가능 저장 매체.
- [0423] 일부 실시예들에서, 명령들을 포함하는 비-일시적 컴퓨터-판독 가능 저장 매체가 또한 제공되고, 명령들은 위에서 설명된 방법들을 수행하기 위한 디바이스(이들테면, 개시된 인코더 및 디코더)에 의해 실행될 수 있다. 비-일시적 매체들의 일반적인 형태들은, 예컨대, 플로피 디스크, 플렉서블 디스크, 하드 디스크, 솔리드 스테이트 드라이브, 자기 테이프 또는 임의의 다른 자기 데이터 저장 매체, CD-ROM, 임의의 다른 광학 데이터 저장 매체, 구멍들의 패턴들을 갖는 임의의 물리적 매체, RAM, PROM 및 EPROM, FLASH-EPROM 또는 임의의 다른 플래시 메모리, NVRAM, 캐시, 레지스터, 임의의 다른 메모리 칩 또는 카트리지 및 이의 네트워크된 버전들을 포함한다. 디바이스는 하나 이상의 프로세서들(CPU들), 입력/출력 인터페이스, 네트워크 인터페이스 및/또는 메모리를 포함할 수 있다.
- [0424] 본원에서 "제1" 및 "제2"와 같은 관계 용어들은 엔티티 또는 동작을 다른 엔티티 또는 동작으로부터 구별하기 위해서만 사용되며, 이러한 엔티티들 또는 동작들 사이의 임의의 실제 관계 또는 시퀀스를 요구하거나 암시하지

않는다는 것이 주의되어야 한다. 또한, "포함하는(comprising)", "갖는(having)", "함유하는(containing)" 및 "포함하는(including)"이라는 단어 및 다른 유사한 형태들은 의미가 동등하고 이러한 단어들 중 임의의 것을 뒤따르는 아이템 또는 아이템들이 그러한 아이템 또는 아이템들의 총망라하는 목록을 의미하거나 나열된 아이템 또는 아이템들만으로 제한되는 것을 의미하지 않는다는 점에서 개방형(open ended)인 것으로 의도된다.

[0425] 본원에서 사용되는 바와 같이, 달리 구체적으로 언급되지 않는 한, 용어 "또는"은 실행 불가능한 경우를 제외하고 모든 가능한 조합들을 포괄한다. 예컨대, 데이터베이스가 A 또는 B를 포함할 수 있다고 언급되는 경우, 구체적으로 달리 언급되거나 실행 불가능하지 않다면, 데이터베이스는 A 또는 B 또는 A 및 B를 포함할 수 있다. 제2의 예로서, 데이터베이스가 A, B 또는 C를 포함할 수 있다고 언급되는 경우, 구체적으로 달리 언급되거나 실행 불가능하지 않다면, 데이터베이스는 A 또는 B 또는 C 또는 A와 B 또는 A와 C 또는 B와 C 또는 A와 B와 C를 포함할 수 있다.

[0426] 위에서 설명된 실시예들은 하드웨어, 또는 소프트웨어(프로그램 코드들) 또는 하드웨어 및 소프트웨어의 조합에 의해 구현될 수 있다는 것이 인지된다. 소프트웨어에 의해 구현되는 경우, 이는 위에서 설명된 컴퓨터-판독 가능 매체들에 저장될 수 있다. 프로세서에 의해 실행될 때, 소프트웨어는 개시된 방법들을 수행할 수 있다. 본 개시내용에서 설명된 컴퓨팅 유닛들 및 다른 기능 유닛들은 하드웨어, 소프트웨어 또는 하드웨어 및 소프트웨어의 조합에 의해 구현될 수 있다. 당업자는 또한 위에서 설명된 모듈들/유닛들 중 다수가 하나의 모듈/유닛으로서 결합될 수 있고, 위에서 설명된 모듈들/유닛들 각각은 복수의 서브-모듈들/서브-유닛들로 추가로 분할될 수 있다는 것을 이해할 것이다.

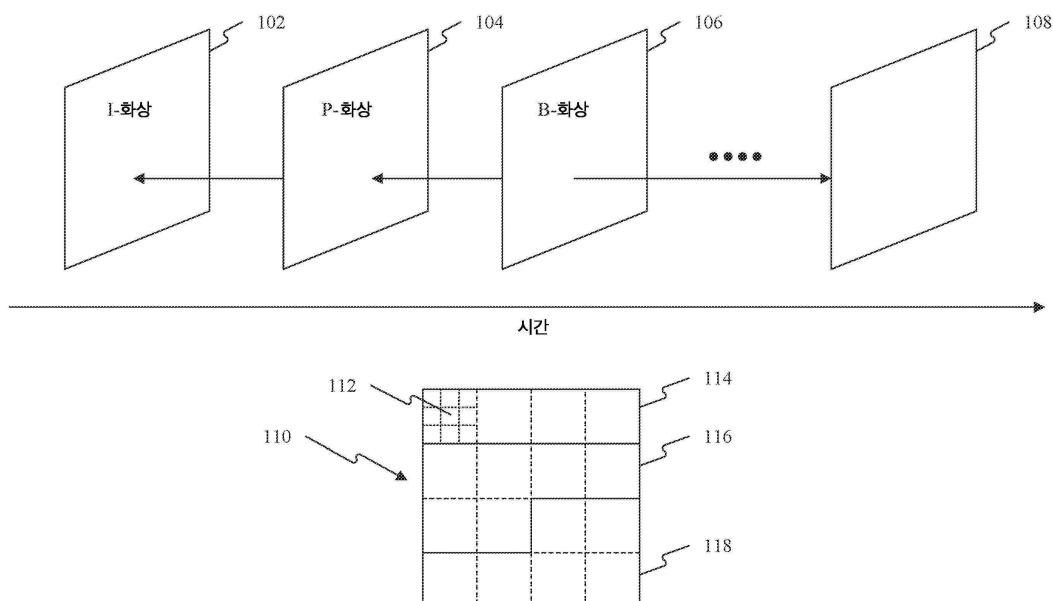
[0427] 전술한 명세서에서, 실시예들은 구현마다 변동될 수 있는 수많은 특정 세부사항들을 참조하여 설명되었다. 설명된 실시예들의 특정 적응들 및 수정들이 이루어질 수 있다. 다른 실시예들은 본원에서 개시된 발명의 실시 및 명세서를 고려하여 당업자들에게 명백해질 수 있다. 본 명세서 및 예들은 단지 예시적인 것으로서만 고려되며, 본 발명의 실제 범위 및 사상은 다음의 청구항들에 의해 표시되는 것으로 의도된다. 또한 도면들에 도시된 단계들의 시퀀스는 단지 예시를 위한 것으로 의도되며 단계들의 임의의 특정 시퀀스로 제한되는 것으로 의도되지 않는다. 따라서, 당업자들은 이러한 단계들이 동일한 방법을 구현하면서 상이한 순서로 수행될 수 있다는 것을 인지할 수 있다.

[0428] 도면들 및 명세서에서, 예시적인 실시예들이 개시되어 있다. 그러나, 이들 실시예에 대해 다수의 변동들 및 수정들이 이루어질 수 있다. 이에 따라 특정 용어들이 이용되지만, 그들은 제한의 목적들이 아니라 일반적이고 설명적인 의미로 사용된다.

도면

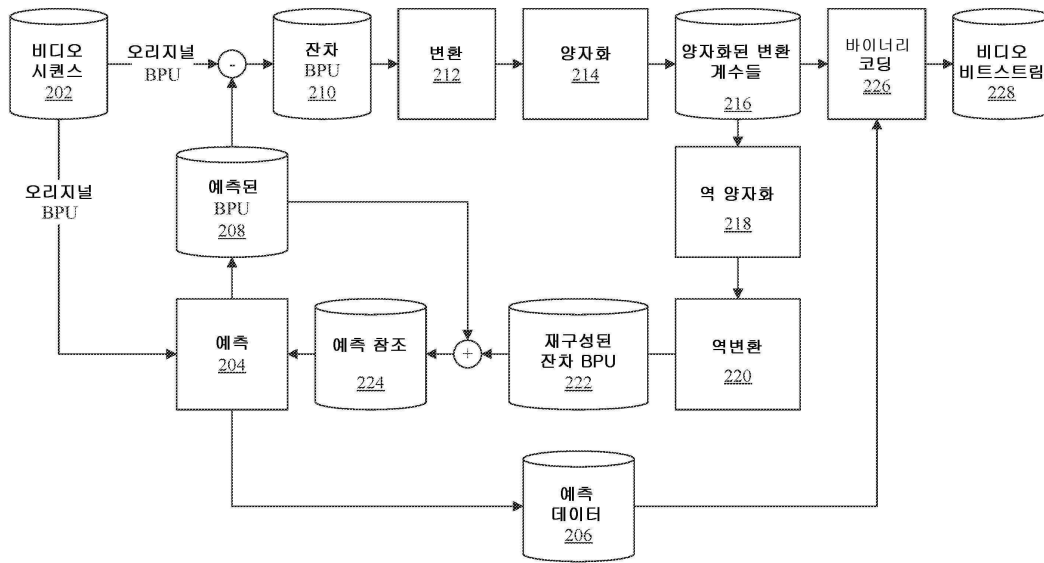
도면1

100



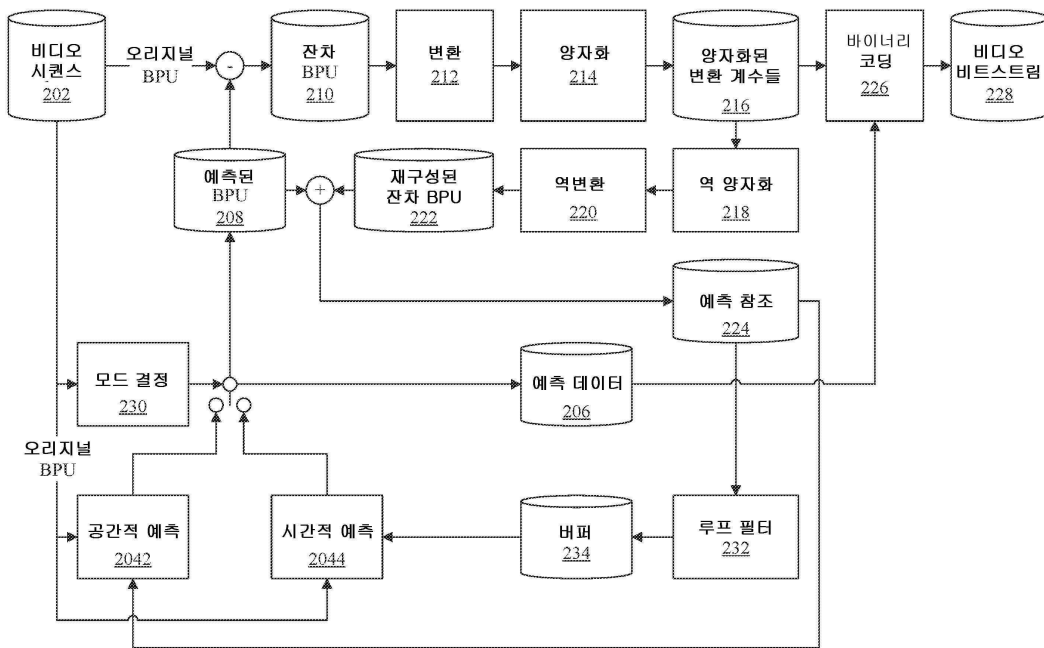
도면2a

200A



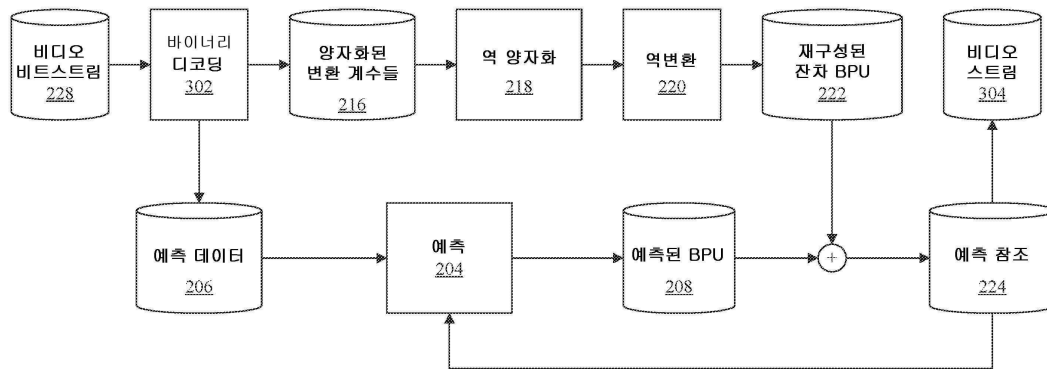
도면2b

200B



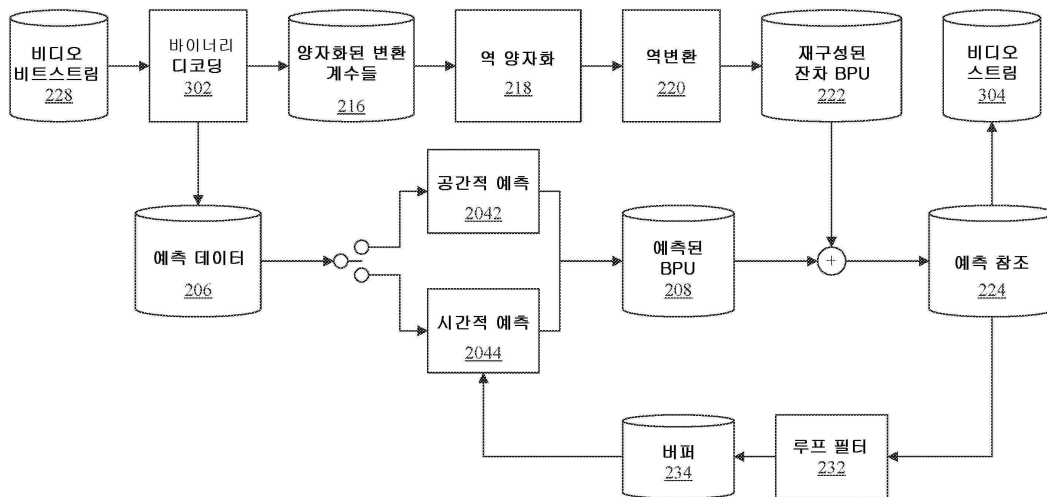
도면3a

300A

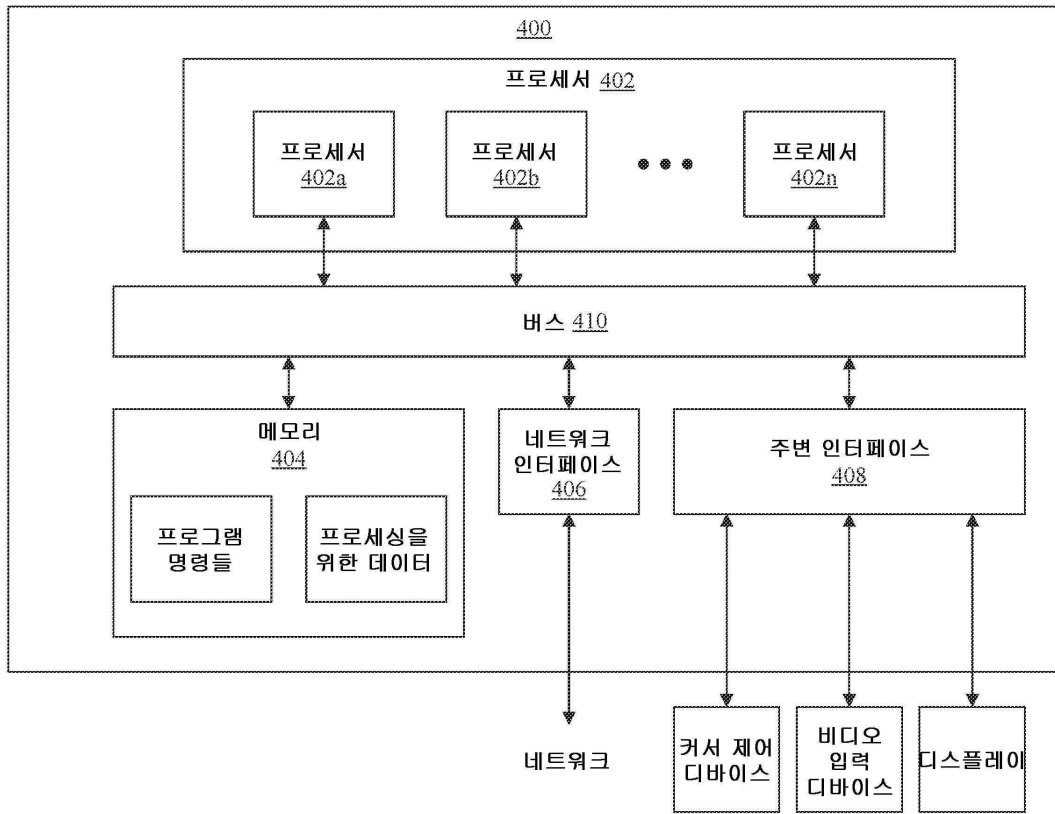


도면3b

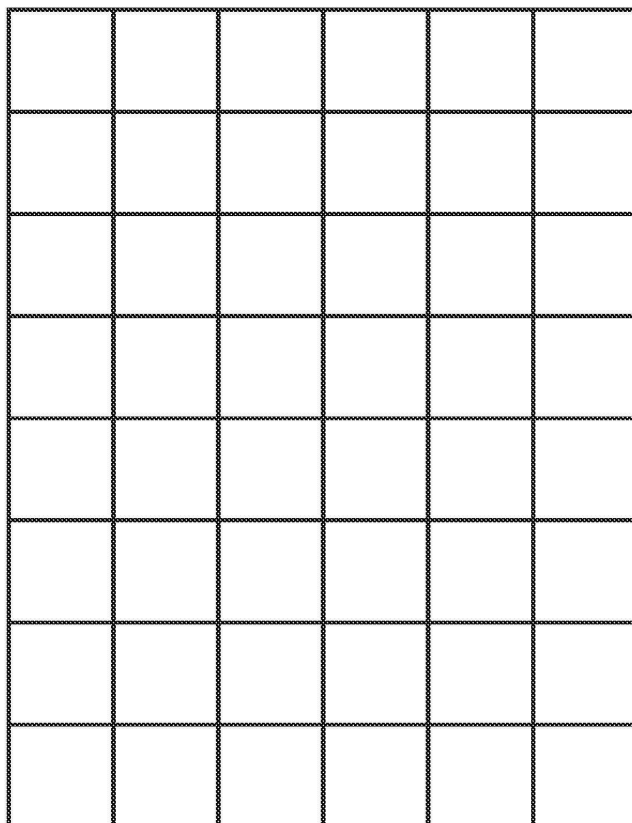
300B



도면4

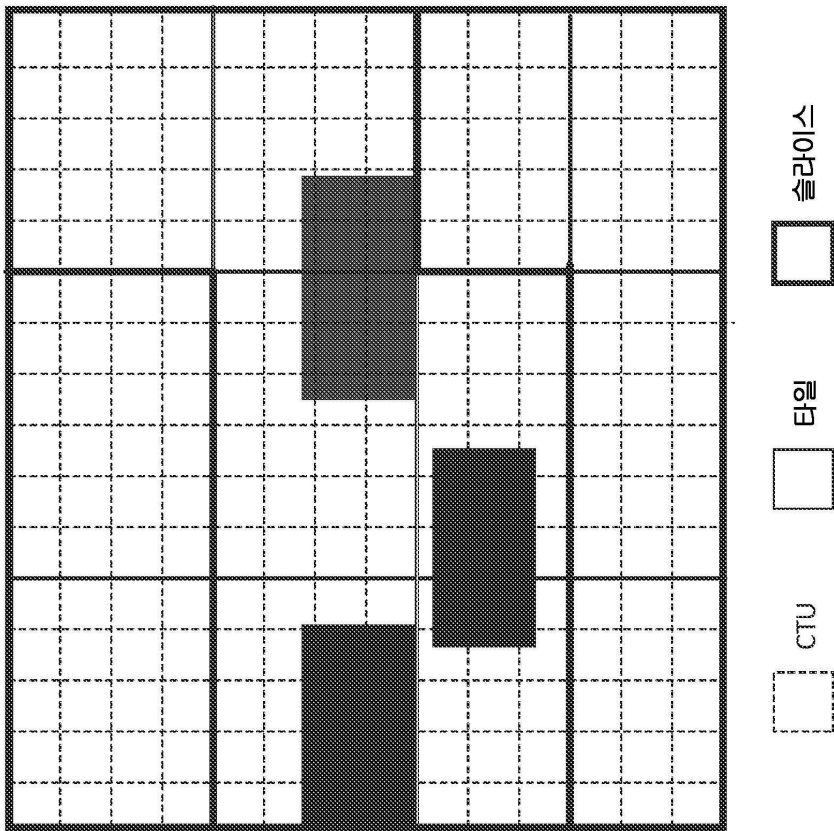


도면5



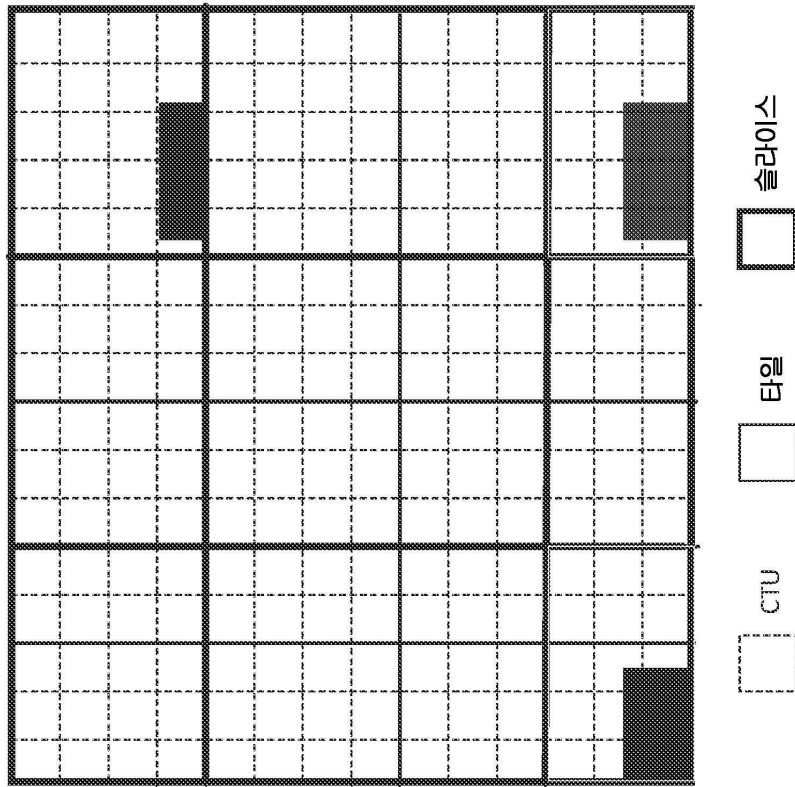
예시의 화상 변환된 CTU

도면6



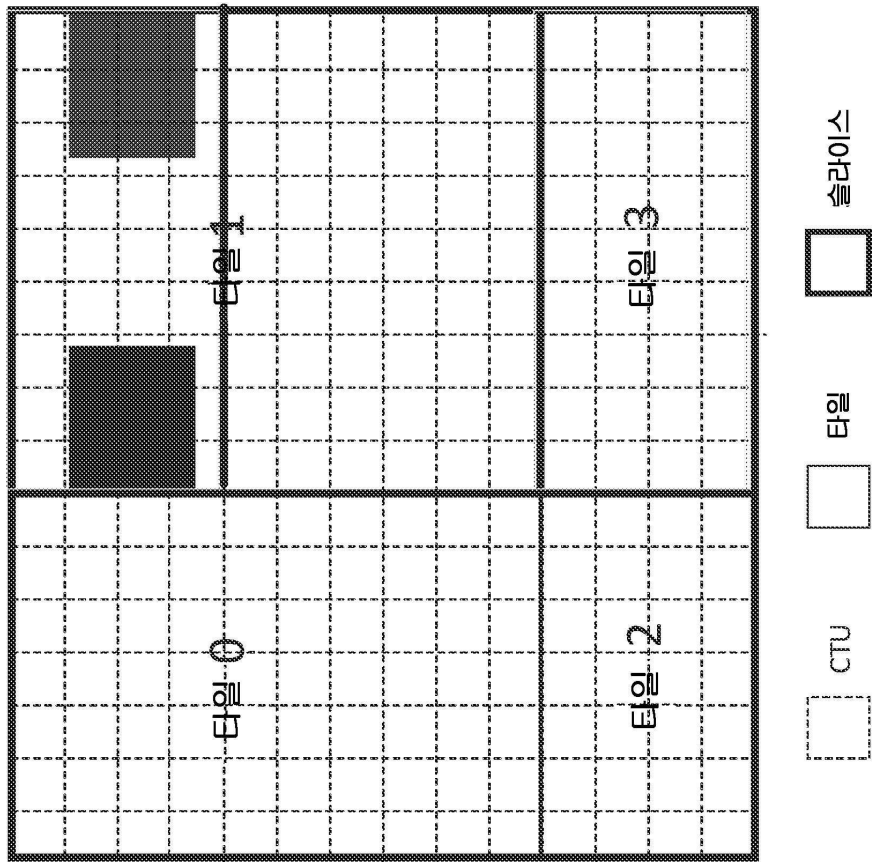
타일을 및 래스터-스캔 슬라이스들로 파티셔닝된 화상의 예

도면7



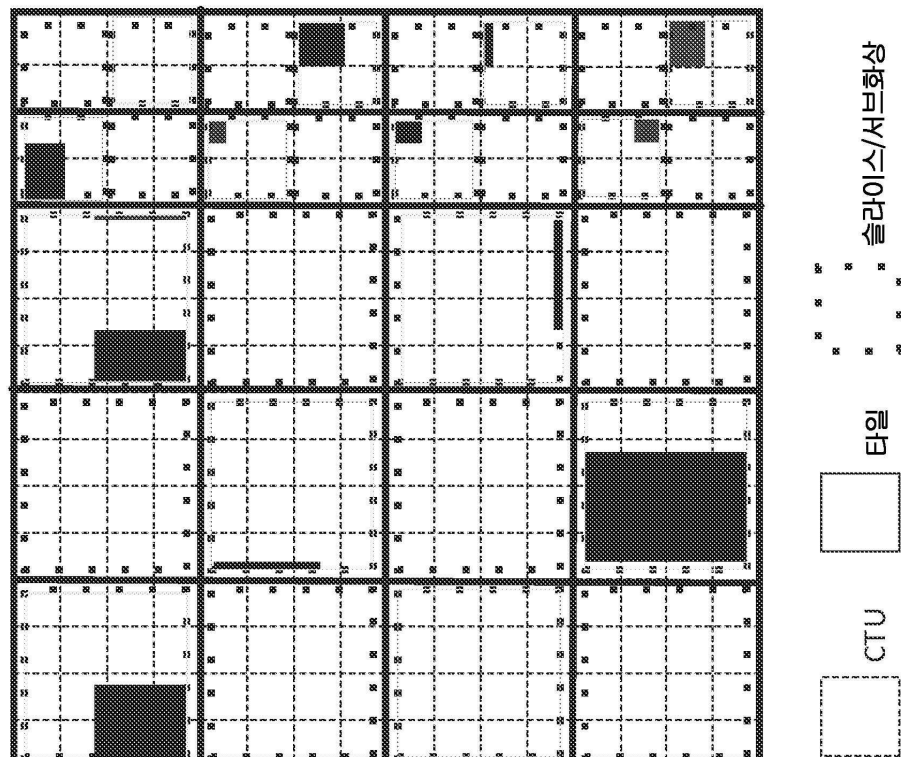
타일을 및 직사각형 슬라이스들로 파티셔닝된 화상의 예

도면8



4개의 타일들 및 4개의 직사각형 슬라이스들로 파티셔닝된 화상의 예

도면9



28개의 서브화상들로 파티셔닝된 화상의 예

도면10

표 1: 서브화상 파티셔닝의 예시적인 SPS 신택스

seq_parameter_set_rbsp() {	Descriptor
...	
subpics_present_flag	u(1)
if(subpics_present_flag) {	
sps_num_subpics_minus1	u(8)
for(i = 0; i <= sps_num_subpics_minus1; i++) {	
subpic_ctu_top_left_x[i]	u(v)
subpic_ctu_top_left_y[i]	u(v)
subpic_width_minus1[i]	u(v)
subpic_height_minus1[i]	u(v)
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	

도면11

표 2: 서브화상 식별자의 예시적인 SPS 신택스

seq_parameter_set_rbsp() {	Descriptor
...	
sps_subpic_id_present_flag	u(1)
if(sps_subpic_id_present_flag) {	
sps_subpic_id_signalling_present_flag	u(1)
if(sps_subpic_id_signalling_present_flag) {	
sps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
sps_subpic_id[i]	u(v)
}	
}	

도면12

표 3: 서브화상 식별자의 예시적인 PPS 신택스

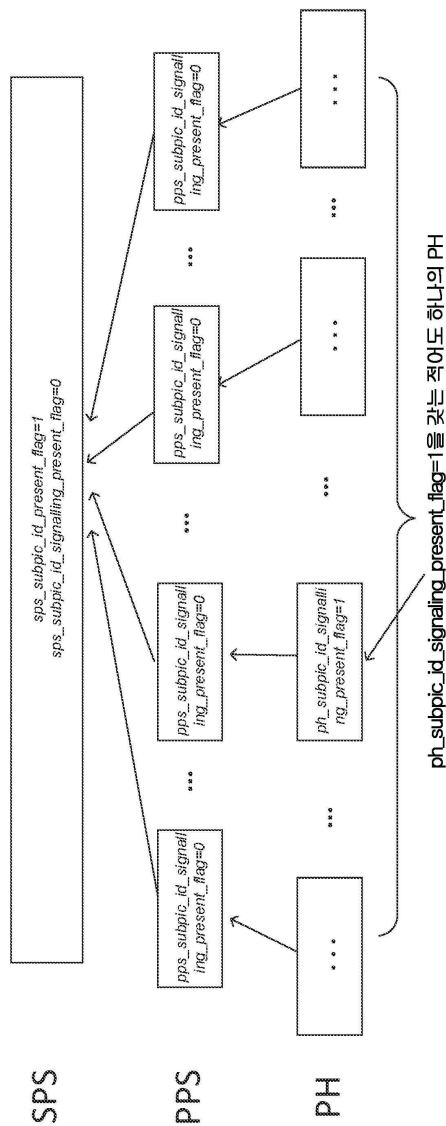
pic_parameter_set_rbsp() {	Descriptor
...	
pps_subpic_id_signalling_present_flag	u(1)
if(pps_subpic_id_signalling_present_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	

도면13

표 4: 서브화상 식별자의 예시적인 PH 선택스

picture_header_rbsp() {	Descriptor
...	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_present_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpic_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	

도면14



도면15

표 5: 서브화상 식별자의 예시적인 PH 신택스

	picture_header_rbsp() {	Descriptor
	...	
1501	if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_present_flag) {	
	if(pps_subpic_id_signalling_present_flag)	
	ph_subpic_id_signalling_present_flag	u(1)
	if(ph_subpic_id_signalling_present_flag) {	
	ph_subpic_id_len_minus1	ue(v)
	for(i = 0; i <= sps_num_subpics_minus1; i++)	
	ph_subpic_id[i]	u(v)
	}	
	}	

도면16

표 6: 서브화상 식별자의 예시적인 PH 신택스

	picture_header_rbsp() {	Descriptor
	...	
1601	if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_present_flag)	
	&& !pps_subpic_id_signalling_present_flag) {	
	ph_subpic_id_len_minus1	ue(v)
	for(i = 0; i <= sps_num_subpics_minus1; i++)	
	ph_subpic_id[i]	u(v)
	}	

도면17a

표 7A: 예시적인 SPS 신택스

	seq_parameter_set_rbsp() {	Descriptor
	...	
	subpics_present_flag	u(1)
	if(subpics_present_flag) {	
	sps_num_subpics_minus1	u(8)
1701	if(sps_num_subpics_minus1 > 0) {	
	for(i = 0; i <= sps_num_subpics_minus1; i++) {	
	subpic_ctu_top_left_x[i]	u(v)
	subpic_ctu_top_left_y[i]	u(v)
	subpic_width_minus1[i]	u(v)
	subpic_height_minus1[i]	u(v)
	subpic_treated_as_pic_flag[i]	u(1)
	loop_filter_across_subpic_enabled_flag[i]	u(1)
	}	
1702	}	
	}	

도면17b

표 7B: 예시적인 SPS 신택스

	seq_parameter_set_rbsp() {	Descriptor
	...	
	subpics_present_flag	u(1)
	if(subpics_present_flag) {	
	sps_num_subpics_minus1	u(8)
	for(i = 0; i <= sps_num_subpics_minus1; i++) {	
1711	if(sps_num_subpics_minus1>0){	
	subpic_ctu_top_left_x[i]	u(v)
	subpic_ctu_top_left_y[i]	u(v)
	subpic_width_minus1[i]	u(v)
	subpic_height_minus1[i]	u(v)
	subpic_treated_as_pic_flag[i]	u(1)
	loop_filter_across_subpic_enabled_flag[i]	u(1)
1712	}	
	}	
	}	
	...	

도면18

표 8: 예시적인 SPS 신택스

	seq_parameter_set_rbsp() {	Descriptor
	...	
	subpics_present_flag	u(1)
	if(subpics_present_flag) {	
	sps_num_subpics_minus1	u(8)
	for(i = 0; i <= sps_num_subpics_minus1; i++) {	
1801	if(sps_num_subpic_minus1>0){	
	subpic_ctu_top_left_x[i]	u(v)
	subpic_ctu_top_left_y[i]	u(v)
	subpic_width_minus1[i]	u(v)
	subpic_height_minus1[i]	u(v)
1802	}	
	subpic_treated_as_pic_flag[i]	u(1)
	loop_filter_across_subpic_enabled_flag[i]	u(1)
	}	
	}	

도면19

표 9: 예시적인 SPS 신택스

		Descriptor
	seq_parameter_set_rbsp() {	
	...	
	subpics_present_flag	u(1)
	if(subpics_present_flag) {	
	sps_num_subpics_minus1	u(8)
	for(i = 0; i <= sps_num_subpics_minus1; i++) {	
	subpic_ctu_top_left_x[i]	u(v)
	subpic_ctu_top_left_y[i]	u(v)
1901	if(i != sps_num_subpics_minus1) {	
	subpic_width_minus1[i]	u(v)
	subpic_height_minus1[i]	u(v)
1902	}	
	subpic_treated_as_pic_flag[i]	u(1)
	loop_filter_across_subpic_enabled_flag[i]	u(1)
	}	
	}	

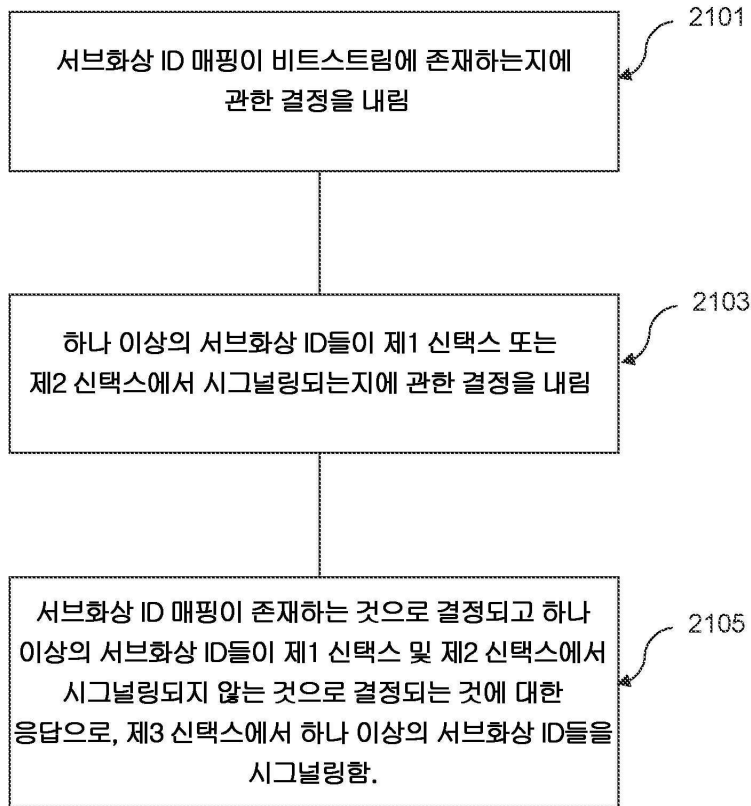
도면20

표 10: 예시적인 SPS 신택스

		Descriptor
	seq_parameter_set_rbsp() {	
	...	
	subpics_present_flag	u(1)
	if(subpics_present_flag) {	
	sps_num_subpics_minus1	u(8)
	for(i = 0; i <= sps_num_subpics_minus1; i++) {	
	subpic_ctu_top_left_x[i]	u(v)
	subpic_ctu_top_left_y[i]	u(v)
	subpic_width_minus1[i]	u(v)
	subpic_height_minus1[i]	u(v)
	subpic_treated_as_pic_flag[i]	u(1)
	loop_filter_across_subpic_enabled_flag[i]	u(1)
	}	
2001	/	
	sps_subpic_id_present_flag	u(1)
	if(sps_subpic_id_present_flag) {	
	sps_subpic_id_signalling_present_flag	u(1)
	if(sps_subpic_id_signalling_present_flag) {	
	sps_subpic_id_len_minus1	ue(v)
	for(i = 0; i <= sps_num_subpics_minus1; i++)	
	sps_subpic_id[i]	u(v)
	}	
	}	
2002	/	
	...	

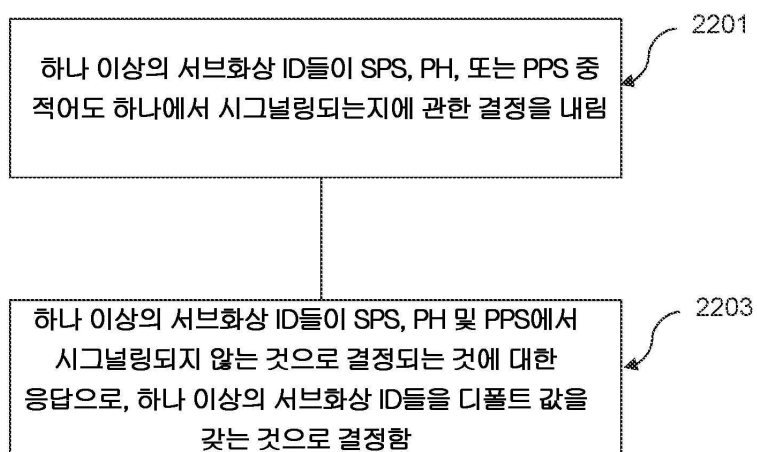
도면21

2100

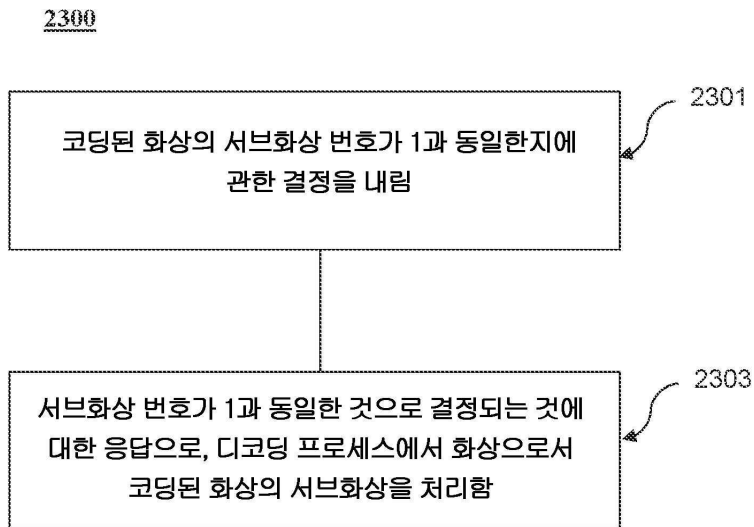


도면22

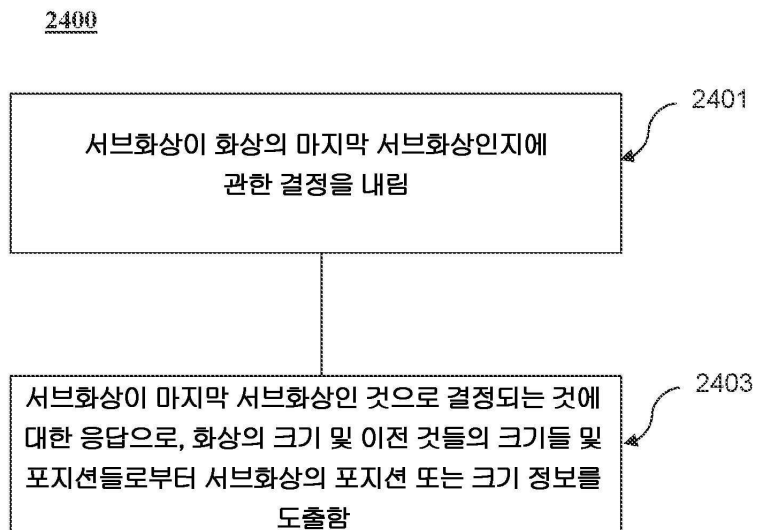
2200



도면23



도면24



도면25

