



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0048077
(43) 공개일자 2023년04월10일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/70 (2014.01) HO4N 19/30 (2014.01)
HO4N 19/91 (2014.01) HO4N 19/98 (2014.01)
- (52) CPC특허분류
HO4N 19/70 (2015.01)
HO4N 19/30 (2015.01)
- (21) 출원번호 10-2023-7006687
- (22) 출원일자(국제) 2022년04월29일
심사청구일자 2023년02월24일
- (85) 번역문제출일자 2023년02월24일
- (86) 국제출원번호 PCT/US2022/072015
- (87) 국제공개번호 WO 2023/056105
국제공개일자 2023년04월06일
- (30) 우선권주장
63/250,166 2021년09월29일 미국(US)
17/710,778 2022년03월31일 미국(US)

- (71) 출원인
텐센트 아메리카 엘엘씨
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
- (72) 발명자
최 병두
미국 94306 캘리포니아주 팔로 알토 2747 파크 블러바드 텐센트 아메리카 엘엘씨 내
- 리우 산
미국 94306 캘리포니아주 팔로 알토 2747 파크 블러바드 텐센트 아메리카 엘엘씨 내
- 헝거 스테판
미국 94306 캘리포니아주 팔로 알토 2747 파크 블러바드 텐센트 아메리카 엘엘씨 내
- (74) 대리인
유미특허법인

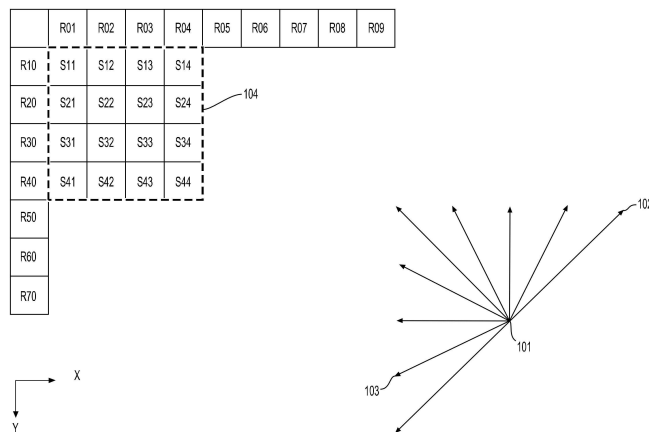
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **잔차 라이스 코딩 확장을 통한 범위 확장을 위한 제약 플래그 시그널링 기술**

(57) 요약

본 개시의 측면들은 비디오 데이터 처리를 위한 방법 및 장치를 제공한다. 일부 예에서, 비디오 데이터 처리를 위한 장치는 처리 회로를 포함한다. 예를 들어, 상기 처리 회로는 비트스트림 내의 코딩된 비디오 데이터의 제1 범위에서의 코딩 제어를 위한 제1 신택스 요소를 결정한다. 상기 제1 신택스 요소는 잔차 코딩에서의 라이스 파라미터 도출을 위한 제1 코딩 도구에 대한 대안인 제2 코딩 도구와 연관된다. 상기 제1 신택스 요소가 상기 제1 범위에서 상기 제2 코딩 도구의 디스에이블을 지시하는 제1 값인 것에 응답하여, 상기 처리 회로는 코딩된 비디오 데이터의 하나 이상의 제2 범위를 포함하는 상기 코딩된 비디오 데이터의 제1 범위를 상기 제2 코딩 도구를 호출하지 않고 디코딩한다.

대표도 - 도1a



(52) CPC특허분류

H04N 19/91 (2015.01)

H04N 19/98 (2015.01)

명세서

청구범위

청구항 1

디코더에서의 비디오 디코딩 방법으로서,

프로세서가 비트스트림 내의 코딩된 비디오 데이터의 제1 범위에서의 코딩 제어를 위한 제1 선택스 요소를 결정하는 단계 - 상기 제1 선택스 요소는 잔차 코딩에서의 라이스 파라미터(Rice parameter) 도출을 위한 제1 코딩 도구에 대한 대안인 제2 코딩 도구와 연관됨 -; 및

상기 제1 선택스 요소가 상기 제1 범위에서 상기 제2 코딩 도구의 디스에이블을 지시하는 제1 값인 것에 응답하여, 상기 프로세서가 코딩된 비디오 데이터의 하나 이상의 제2 범위를 포함하는 상기 코딩된 비디오 데이터의 제1 범위를 상기 제2 코딩 도구를 호출하지 않고 디코딩하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 제1 선택스 요소는 상기 디코더에서 출력 계층 세트 내의 픽처의 코딩 제어를 위한 일반 제약 정보(general constraint information)에 있는, 방법.

청구항 3

제2항에 있어서,

상기 제1 선택스 요소의 상기 제1 값은 상기 출력 계층 세트 내의 각각의 코딩된 계층 비디오 시퀀스(coded layer video sequence, CLVS)에서의 상기 제2 코딩 도구의 디스에이블을 지시하는, 방법.

청구항 4

제2항에 있어서,

상기 비트스트림 내의 코딩된 계층 비디오 시퀀스(CLVS)의 코딩 제어를 위한 제2 선택스 요소를, 상기 CLVS를 디코딩하기 위한 상기 제2 코딩 도구를 호출하지 않음을 지시하는 값을 갖도록 제약하는 단계를 더 포함하는 방법.

청구항 5

제2항에 있어서,

상기 제1 선택스 요소가 제2 값인 것에 응답하여, 상기 비트스트림 내의 코딩된 계층 비디오 시퀀스(CLVS)의 코딩 제어를 위한 제2 선택스 요소의 값을 결정하는 단계 - 상기 제2 선택스 요소는 상기 CLVS에서의 상기 제2 코딩 도구의 인에이블/디스에이블을 지시함 -를 더 포함하는 방법.

청구항 6

제5항에 있어서,

상기 제2 선택스 요소의 값을 결정하는 단계는,

상기 CLVS에 대한 시퀀스 파라미터 세트(sequence parameter set, SPS)에 존재하지 않는 상기 제2 선택스 요소에 응답하여, 상기 CLVS에서 상기 제2 코딩 도구의 디스에이블을 지시하기 위한 상기 제2 선택스 요소의 값을 추론하는 단계를 포함하는, 방법.

청구항 7

제1항에 있어서,

상기 제1 선택스 요소를 결정하는 단계는,

선택스 구조의 일반 제약 정보에 대한 추가 비트를 지시하는 상기 선택스 구조의 선택스 요소에 응답하여 일반 제약 정보에 대한 상기 선택스 구조로부터 상기 제1 선택스 요소를 디코딩하는 단계를 더 포함하는, 방법.

청구항 8

제1항에 있어서,

상기 제2 코딩 도구는 상기 디코더에 의해 지원되지 않는 범위 확장으로 정의되는, 방법.

청구항 9

제8항에 있어서,

상기 제2 코딩 도구는 잔차 코딩 동안 절대값의 이진화를 위한 라이스 파라미터 도출을 위한 것인, 방법.

청구항 10

처리 회로를 포함하는 비디오 디코딩을 위한 장치로서,

상기 처리 회로는,

비트스트림 내의 코딩된 비디오 데이터의 제1 범위에서의 코딩 제어를 위한 제1 선택스 요소를 결정하고 - 상기 제1 선택스 요소는 잔차 코딩에서의 라이스 파라미터 도출을 위한 제1 코딩 도구에 대한 대안인 제2 코딩 도구와 연관됨 -; 및

상기 제1 선택스 요소가 상기 제1 범위에서 상기 제2 코딩 도구의 디스에이블을 지시하는 제1 값인 것에 응답하여, 코딩된 비디오 데이터의 하나 이상의 제2 범위를 포함하는 상기 코딩된 비디오 데이터의 제1 범위를 상기 제2 코딩 도구를 호출하지 않고 디코딩하도록 구성되는,

장치.

청구항 11

제10항에 있어서,

상기 제1 선택스 요소는 출력 계층 세트 내의 픽처의 코딩 제어를 위한 일반 제약 정보에 있는, 장치.

청구항 12

제11항에 있어서,

상기 제1 선택스 요소의 상기 제1 값은 상기 출력 계층 세트 내의 각각의 코딩된 계층 비디오 시퀀스(CLVS)에서의 상기 제2 코딩 도구의 디스에이블을 지시하는, 장치.

청구항 13

제11항에 있어서,

상기 처리 회로는,

상기 비트스트림 내의 코딩된 계층 비디오 시퀀스(CLVS)의 코딩 제어를 위한 제2 선택스 요소를, 상기 CLVS를 디코딩하기 위한 상기 제2 코딩 도구를 호출하지 않음을 지시하는 값을 갖도록 제약하도록 구성되는, 장치.

청구항 14

제11항에 있어서,

상기 처리 회로는,

상기 제1 선택스 요소가 제2 값인 것에 응답하여, 상기 비트스트림 내의 코딩된 계층 비디오 시퀀스(CLVS)에서의 코딩 제어를 위한 제2 선택스 요소의 값을 결정하도록 구성되며, 상기 제2 선택스 요소는 상기 CLVS에서 상

기 제2 코딩 도구의 인에이블/디스에이블을 지시하는, 장치.

청구항 15

제14항에 있어서,

상기 처리 회로는,

상기 CLVS에 대한 시퀀스 파라미터 세트(SPS)에 존재하지 않는 상기 제2 신택스 요소에 응답하여, 상기 CLVS에서 상기 제2 코딩 도구의 디스에이블을 지시하기 위한 상기 제2 신택스 요소의 값을 추론하도록 구성되는, 장치.

청구항 16

제10항에 있어서,

상기 처리 회로는,

신택스 구조의 일반 제약 정보에 대한 추가 비트를 지시하는 상기 신택스 구조의 신택스 요소에 응답하여, 일반 제약 정보에 대한 상기 신택스 구조로부터 상기 제1 신택스 요소를 디코딩하도록 구성되는, 장치.

청구항 17

제10항에 있어서,

상기 제2 코딩 도구는 상기 장치에 의해 지원되지 않는 범위 확장으로 정의되는, 장치.

청구항 18

제17항에 있어서,

상기 제2 코딩 도구는 잔차 코딩 동안 절대값의 이진화를 위한 라이스 파라미터 도출을 위한 것인, 장치.

청구항 19

명령어를 저장하는, 컴퓨터로 판독 가능한 비일시적 저장 매체로서,

상기 명령어는 적어도 하나의 프로세서에 의해 실행될 때 상기 적어도 하나의 프로세서로 하여금,

비트스트림 내의 코딩된 비디오 데이터의 제1 범위에서의 코딩 제어를 위한 제1 신택스 요소를 결정하는 것 - 상기 제1 신택스 요소는 잔차 코딩에서의 라이스 파라미터 도출을 위한 제1 코딩 도구에 대한 대안인 제2 코딩 도구와 연관됨 -; 및

상기 제1 신택스 요소가 상기 제1 범위에서 상기 제2 코딩 도구의 디스에이블을 지시하는 제1 값인 것에 응답하여, 코딩된 비디오 데이터의 하나 이상의 제2 범위를 포함하는 상기 코딩된 비디오 데이터의 제1 범위를 상기 제2 코딩 도구를 호출하지 않고 디코딩하는 것을 수행하도록 하는,

컴퓨터로 판독 가능한 비일시적 저장 매체.

청구항 20

제19항에 있어서,

상기 제1 신택스 요소는 출력 계층 세트 내의 픽처의 코딩 제어를 위한 일반 제약 정보에 있는, 컴퓨터로 판독 가능한 비일시적 저장 매체.

발명의 설명

기술 분야

[0001] 참조에 의한 통합

[0002] 본 출원은 2021년 9월 29일자로 출원된 미국 가출원 제63/250,166호, "TECHNIQUES FOR CONSTRAINT FLAG SIGNALING FOR RANGE EXTENSION WITH RESIDUAL RICE CODING EXTENSION(잔차 라이스 코딩 확장을 통한 범위 확

장을 위한 제약 플래그 시그널링 기술)"에 대한 우선권의 혜택을 주장하여, 2022년 3월 31일자로 출원된 미국 특허출원 제17/710,778호 "TECHNIQUES FOR CONSTRAINT FLAG SIGNALING FOR RANGE EXTENSION WITH RESIDUAL RICE CODING EXTENSION(잔차 라이스 코딩 확장을 통한 범위 확장을 위한 제약 플래그 시그널링 기술)"의 혜택을 주장한다. 상기 선행 출원들의 개시 내용은 그 전체가 참조에 의해 본 출원에 통합된다.

[0003] 본 개시는 일반적으로 비디오 코딩에 관련된 실시예를 설명한다.

배경 기술

[0004] 본 명세서에 제공되는 배경 설명은 일반적으로 본 개시의 맥락을 제시하기 위한 것이다. 이 배경기술 섹션에 기술되는 저작물의 범위에서, 현재 지명된 발명자들의 저작물과 출원 시에 선행 기술로 인정되지 않을 수 있는 설명의 측면들은 명시적으로도 묵시적으로도 본 개시의 선행 기술로 인정되지 않는다.

[0005] 비디오 코딩 및 디코딩은 움직임 보상과 함께 인터 픽처 예측을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처를 포함할 수 있으며, 각각의 픽처는, 예를 들어, 1920×1080 휘도 샘플(luminance sample, 루마 샘플이라고도 함) 및 연관된 색차 샘플(chrominance sample, 크로마 샘플이라고도 함)의 공간 차원을 갖는다. 일련의 픽처는 예를 들어 초당 60장의 픽처 또는 60 Hz의 고정 또는 가변 픽처 레이트(비공식적으로 프레임 레이트라고 알려져 있음)를 가질 수 있다. 압축되지 않은 비디오는 특정 비트레이트 요건을 갖는다. 예를 들어, 샘플당 8비트에서 1080p60 4:2:0 비디오(60 Hz 프레임 레이트에서 1920×1080 휘도 샘플 해상도)는 1.5Gbit/s에 가까운 대역폭을 필요로 한다. 한 시간의 이러한 비디오는 600GB 이상의 저장 공간을 필요로 한다.

[0006] 비디오 코딩 및 디코딩의 한 가지 목적은 압축을 통해, 입력 비디오 신호의 중복성(redundancy)을 줄이는 것일 수 있다. 압축은 앞서 언급한 대역폭 및/또는 저장 공간 요건을, 경우에 따라서는 두자릿수 이상 감소시키는 데 도움이 될 수 있다. 무손실 압축(lossless compression)과 손실 압축(lossy compress) 둘 다는 물론, 이들의 조합도 사용될 수 있다. 무손실 압축이란 압축된 원본 신호(original signal)에서 원본 신호의 정확한 사본(exact copy)을 재구축할 수 있는 기술을 말한다. 손실 압축을 사용하는 경우, 재구축된 신호는 원본 신호와 동일하지 않을 수 있지만 원본 신호와 재구축된 신호 사이의 왜곡은, 재구축된 신호가 의도된 애플리케이션에 유용하도록 충분히 작다. 비디오의 경우, 손실 압축이 널리 채용된다. 용인 가능한 왜곡의 양은 애플리케이션에 따라 다른데, 예를 들어, 특정 소비자 스트리밍 애플리케이션의 사용자는 영화 또는 텔레비전 배급 애플리케이션(television distribution application)의 사용자보다 더 높은 왜곡을 용인할 수 있다. 달성 가능한 압축 비율은 다음을 반영할 수 있다: 허용 가능한/용인 가능한 왜곡이 높을수록 압축 비율이 높을 수 있다.

[0007] 비디오 인코더 및 디코더는, 예를 들어, 움직임 보상(motion compensation), 푸리에 변환(Fourier transform), 양자화(quantization), 및 엔트로피 코딩(entropy coding)을 포함한, 여러 광범위한 범주로부터의 기술을 이용할 수 있다.

[0008] 비디오 코덱 기술은 인트라 코딩(intra coding)으로 알려진 기술을 포함할 수 있다. 인트라 코딩에서, 샘플 값은 이전에 재구축된 참조 픽처의 샘플 또는 다른 데이터에 대한 참조 없이 표현된다. 일부 비디오 코덱에서, 픽처는 샘플 블록들로 공간적으로 세분화된다. 샘플의 모든 블록이 인트라 모드로 코딩되는 경우, 그 픽처는 인트라 픽처(intra picture)일 수 있다. 인트라 픽처 및 독립 디코더 리프레시 픽처(independent decoder refresh picture)와 같은 그 도출물(derivation)은 디코더 상태를 재설정하는 데 사용될 수 있으며, 따라서 코딩된 비디오 비트스트림 및 비디오 세션의 첫 번째 픽처, 또는 스틸 이미지(still image)로 사용될 수 있다. 인트라 블록의 샘플은 변환에 도출될 수 있고, 변환 계수는 엔트로피 코딩 전에 양자화될 수 있다. 인트라 예측(intra prediction)은 변환 전 도메인(pre-transform domain)에서 샘플 값을 최소화하는 기술일 수 있다. 경우에 따라서는, 변환 후 DC 값이 작을수록, 그리고 AC 계수가 작을수록, 엔트로피 코딩 후 블록을 표현하기 위해 주어진 양자화 스텝 크기(quantization step size)에서 필요한 비트는 더 적다.

[0009] 예를 들어 MPEG-2 생성 코딩 기술로부터 알려진 것과 같은 전통적인 인트라 코딩은 인트라 예측을 사용하지 않는다. 하지만, 일부 더 새로운 비디오 압축 기술은, 예를 들어 공간적으로 이웃하고 디코딩 순서에서 선행하는 데이터 블록의 인코딩/디코딩 동안 획득되는 메타데이터 및/또는 주변 샘플 데이터로부터 시도하는 기술을 포함한다. 이러한 기술은 앞으로 "인트라 예측" 기술이라고 한다. 적어도 일부 경우에, 인트라 예측은 참조 픽처로부터가 아닌 재구축중인 현재 픽처로부터의 참조 데이터만 사용한다.

[0010] 다양한 형태의 인트라 예측이 있을 수 있다. 이러한 기술 중 하나 이상이 주어진 비디오 코딩 기술에서 사용 가능한 경우, 사용 중인 기술은 인트라 예측 모드로 코딩될 수 있다. 어떤 경우에는, 모드는 서브모드 및/또는 파라미터들을 가질 수 있고, 이들 개별적으로 코딩될 수 있거나 집합적으로 모드 코드워드에 포함될 수 있다. 어

편 코드워드가 주어진 모드/서브모드/파라미터 조합에 사용되는지는 인트라 예측을 통한 코딩 효율 이득(coding efficiency gain)에 영향을 미칠 수 있어서, 코드워드를 비트스트림으로 변환하는 데 사용되는 엔트로피 코딩 기술에도 영향을 미칠 수 있다.

- [0011] 특정 모드의 인트라 예측은 H.264와 함께 도입되었고, H.265에서 개선되었으며, 공동 탐사 모델(joint exploration model, JEM), 다용도 비디오 코딩(versatile video coding, VVC) 및 벤치 마크 세트(benchmark set, BMS)와 같은 더 새로운 코딩 기술에서 더욱 개선되었다. 예측자 블록(predictor block)은 이미 사용 가능한 샘플에 속하는 이웃 샘플 값을 사용하여 형성될 수 있다. 이웃 샘플의 샘플 값은 방향에 따라 예측자 블록에 복사될 수 있다. 사용 중의 방향에 대한 참조는 비트스트림에 코딩될 수 있거나 그 자체가 예측될 수 있다.
- [0012] 도 1a를 참조하면, H.265의 33개의 가능한 예측자 방향(35개 인트라 모드의 33개 각도 모드에 대응) 중 알려져 있는 9개의 예측자 방향의 서브세트가 우측 하단에 도시되어 있다. 화살표가 수렴하는 지점(101)은 예측되는 샘플을 나타낸다. 화살표는 샘플이 예측되는 방향을 나타낸다. 예를 들어, 화살표(102)는 샘플(101)이 샘플 또는 샘플들로부터 수평 방향에서 45도 각도로 우측 상단으로 예측된다는 것을 지시한다. 마찬가지로, 화살표(103)는 샘플(101)이 샘플 또는 샘플들로부터 수평 방향에서 22.5도 각도로 샘플(101)의 좌측 하단으로 예측된다는 것을 지시한다.
- [0013] 도 1a를 여전히 참조하면, 좌측 상단에는 (굵은 파선으로 나타낸) 4×4 샘플의 정사각형 블록(104)이 표시되어 있다. 정사각형 블록(104)은 16개의 샘플을 포함하고, 각각은 "S", Y 차원의 위치(예: 행 색인), 및 X 차원의 위치(예: 열 색인)로 레이블이 부여된다. 예를 들어, 샘플 S21은 Y 차원의 (상단에서) 두 번째 샘플이고 X 차원의 (좌측에서) 첫 번째 샘플이다. 마찬가지로, 샘플 S44는 Y 및 X 차원 모두에서 블록(104)의 네 번째 샘플이다. 블록의 크기가 4×4 샘플이므로, S44는 우측 하단에 있다. 유사한 번호 매기기 방식을 따르는 참조 샘플이 추가로 도시되어 있다. 참조 샘플은 R, 블록(104)에 상대적인 Y 위치(예: 행 색인) 및 X 위치(열 색인)로 레이블이 부여된다. H.264 및 H.265 모두에서, 예측 샘플은 재구성중인 블록에 이웃하며, 따라서 음의 값을 사용할 필요는 없다.
- [0014] 인트라 픽처 예측(intra picture prediction)은 시그널링된 예측 방향으로 이웃 샘플로부터 참조 샘플 값을 복사함으로써 작동할 수 있다. 예를 들어, 코딩된 비디오 비트스트림이, 이 블록에 대해, 화살표(102)와 일치하는 예측 방향을 지시하는 시그널링을 포함한다고 가정한다 - 즉, 샘플들은 수평 방향에서 45도 각도로 우측 상단에 대한 하나의 예측 샘플 또는 샘플들로부터 예측된다. 이러한 경우, 샘플 S41, S32, S23 및 S14는 동일한 참조 샘플 R05로부터 예측된다. 그런 다음 샘플 S44는 참조 샘플 R08로부터 예측된다.
- [0015] 어떤 경우에는, 참조 샘플을 계산하기 위해, 특히 방향을 45도로 균등하게 나눌 수 없을 때, 예를 들어 보간을 통해 다수의 참조 샘플의 값이 결합될 수 있다.
- [0016] 가능한 방향의 수는 비디오 코딩 기술이 발전됨에 따라 증가했다. H.264(2003년)에서는, 9개의 서로 다른 방향이 표현될 수 있다. H.265(2013년)에서는 33개로 증가했으며, 본 개시 당시 JEM/VVC/BMS는 최대 65개의 방향을 지원할 수 있다. 가장 가능성이 있는 방향을 식별하기 위한 실험이 수행되었으며, 엔트로피 코딩에서의 특정 기술이 그러한 가장 가능성이 있는 방향을 적은 수의 비트로 인코딩하는 데 사용하여, 가능성이 적은 방향에 대해 일정한 패널티를 받아들인다. 또한, 방향 자체는 때로 이미 디코딩된 이웃 블록에 사용된 이웃 방향으로부터 예측될 수 있다.
- [0017] 도 1b는 시간 경과에 따라 증가하는 예측 방향의 수를 설명하기 위해 JEM에 따른 65개의 인트라 예측 방향을 나타낸 개략도(180)를 도시한다.
- [0018] 방향을 나타내는 비트를 코딩된 비디오 비트스트림에서의 인트라 예측 방향비트의 매핑은 비디오 코딩 기술에 따라 다를 수 있으며; 예를 들어, 예측 방향의 단순한 직접 매핑에서부터 인트라 예측 모드, 코드워드, 가장 가능성이 높은 모드를 포함하는 복잡한 적응형 방식, 및 유사한 기술에 이르기까지 다양하다. 하지만, 모든 경우에, 비디오 콘텐츠에서 통계적으로 다른 특정 방향보다 발생할 가능성이 낮은 예측에 대한 특정 방향이 있을 수 있다. 비디오 압축의 목표는 중복성을 줄이는 것이므로, 잘 설계된 비디오 코딩 기술에서, 가능성이 낮은 방향은 가능성이 더 높은 방향보다 더 많은 수의 비트로 표현될 수 있다.
- [0019] 움직임 보상은 손실 압축 기술일 수 있으며 이전에 재구성된 픽처 또는 그 일부(참조 픽처)로부터의 샘플 데이터가 움직임 벡터(motion vector, 이하 MV)에 의해 지시되는 방향으로 공간적으로 시프트된 후, 새로 재구성되는 픽처 또는 픽처 일부의 예측에 사용되는 기술에 관련될 수 있다. 경우에 따라서는, 참조 픽처는 현재 재구성 중인 픽처와 동일할 수 있다. MV는 X와 Y의 2차원, 또는 3차원일 수 있으며, 세 번째 차원은 사용중인 참조 픽

처의 표시이다(후자는 간접적으로 시간 차원일 수 있음).

[0020] 일부 비디오 압축 기술에서, 샘플 데이터의 특정 영역(area)에 적용할 수 있는 MV는 다른 MV, 예를 들어 재구축 중인 영역에 공간적으로 인접하고 디코딩 순서에서 MV보다 선행하는 샘플 데이터의 다른 영역과 관련되어 있는 다른 MV들로부터 예측될 수 있다. 이렇게 하면 MV를 코딩하는 데 필요한 데이터의 양을 상당히 줄일 수 있으므로, 중복성을 없애고 압축을 높일 수 있다. 예를 들어, 카메라로부터 도출된 입력 비디오 신호(자연 비디오로 알려져 있음)를 코딩할 때 단일 MV가 적용될 수 있는 영역보다 큰 영역이 비디오 시퀀스에서 유사한 방향으로 이동하는 통계적 가능성이 있기 때문에, MV 예측은 효과적으로 작동할 수 있고, 따라서 경우에 따라서는 이웃 영역의 MV로부터 도출된 유사한 움직임 벡터를 사용하여 예측될 수 있다. 그 결과 주어진 영역에 대해 발견된 MV는 주변 MV로부터 예측된 MV와 유사하거나 동일하며, 엔트로피 코딩 후, MV를 직접 코딩한다면 사용되었을 비트 수보다 더 적은 수의 비트로 표시될 수 있다. 경우에 따라서는, MV 예측은 원래 신호(즉, 샘플 스트림)로부터 도출된 신호(즉, MV)의 무손실 압축의 일례일 수 있다. 다른 경우에는 MV 예측 자체가 손실될 수 있는데, 예를 들어 주변의 여러 MV로부터 예측자를 계산할 때 라운딩 오차(rounding error)가 발생하기 때문이다.

[0021] 다양한 MV 예측 메커니즘이 H.265/HEVC(ITU-T Rec. H.265, "High Efficiency Video Coding(고효율 비디오 코딩)", 2016년 12월)에 기술되어 있다. H.265가 제공하는 많은 MV 예측 메커니즘 중에서 여기에 설명된 기술은 이하 "공간 병합(spatial merge)"이라 한다.

[0022] 도 2를 참조하면, 현재 블록(201)은 공간적으로 시프트된 동일한 크기의 이전 블록으로부터 예측 가능하도록 움직임 검색 프로세스 동안 인코더에 의해 발견된 샘플을 포함할 수 있다. MV를 직접 코딩하는 대신, MV는 A0, A1 및 B0, B1, B2(각각 202~206)으로 표시된 5개의 주변 샘플 중 하나와 연관된 MV를 사용하여, 예를 들어 가장 최근의(디코딩 순서상) 참조 픽처로부터, 하나 이상의 참조 픽처와 연관된 메타데이터로부터 도출될 수 있다. H.265에서, MV 예측은 이웃 블록이 사용하는 것과 동일한 참조 픽처로부터의 예측자를 사용할 수 있다.

발명의 내용

[0023] 본 개시의 측면은 비디오 데이터 처리를 위한 방법 및 장치를 제공한다. 일부 예에서, 비디오 데이터 처리를 위한 장치는 처리 회로를 포함한다. 예를 들어, 상기 처리 회로는 비트스트림 내의 코딩된 비디오 데이터의 제1 범위에서의 코딩 제어를 위한 제1 신택스 요소를 결정한다. 상기 제1 신택스 요소는 잔차 코딩에서의 라이스 파라미터 도출을 위한 제1 코딩 도구에 대한 대안인 제2 코딩 도구와 연관되어 있다. 상기 제1 신택스 요소가 상기 제1 범위에서 상기 제2 코딩 도구의 디스에이블을 지시하는 제1 값인 것에 응답하여, 상기 처리 회로는 코딩된 비디오 데이터의 하나 이상의 제2 범위를 포함하는 상기 코딩된 비디오 데이터의 제1 범위를 상기 제2 코딩 도구를 호출하지 않고 디코딩한다.

[0024] 일부 실시예에서, 상기 제1 신택스 요소는 일반적으로 출력 계층 세트 내의 픽처의 코딩 제어를 위한 제약 정보이다. 일부 예에서, 상기 제1 신택스 요소의 상기 제1 값은 상기 출력 계층 세트 내의 각각의 코딩된 계층 비디오 시퀀스(coded layer video sequence, CLVS)에서의 상기 제2 코딩 도구의 디스에이블을 지시한다.

[0025] 일부 예에서, 상기 처리 회로는 상기 비트스트림 내의 코딩된 계층 비디오 시퀀스(CLVS)의 코딩 제어를 위한 제2 신택스 요소를, 상기 CLVS를 디코딩하기 위한 상기 제2 코딩 도구를 호출하지 않음을 나타내는 값을 갖도록 제약할 수 있다.

[0026] 일부 실시예에서, 상기 제1 신택스 요소가 제2 값인 것에 응답하여, 상기 처리 회로는 상기 비트스트림 내의 코딩된 계층 비디오 시퀀스(CLVS)의 코딩 제어를 위한 제2 신택스 요소의 값을 결정한다. 상기 제2 신택스 요소는 상기 CLVS에서 상기 제2 코딩 도구의 인에이블/디스에이블을 나타낸다. 상기 제2 코딩 도구는 잔차 코딩 동안 절대값의 이진화를 위한 라이스 파라미터 도출을 위한 것이다.

[0027] 일부 예에서, 상기 처리 회로는 상기 CLVS에 대한 시퀀스 파라미터 세트(SPS)에 존재하지 않는 상기 제2 신택스 요소에 응답하여, 상기 CLVS에서 상기 제2 코딩 도구의 디스에이블을 지시하기 위한 상기 제2 신택스 요소의 값을 추론한다.

[0028] 일부 예에서, 상기 처리 회로는 신택스 구조의 일반 제약 정보에 대한 추가 비트를 나타내는 상기 신택스 구조의 신택스 요소에 응답하여, 일반 제약 정보에 대한 상기 신택스 구조로부터 상기 제1 신택스 요소를 디코딩한다.

[0029] 일부 예에서, 상기 제2 코딩 도구는 표준의 범위 확장으로 정의되며, 상기 범위 확장은 상기 장치에 의해 지원되지 않는다.

[0030] 본 개시의 측면은 또한 명령어를 저장하는, 컴퓨터로 판독 가능한 비일시적 저장 매체를 제공하며, 상기 명령어는 컴퓨터에 의해 실행될 때 상기 컴퓨터로 하여금 상기 비디오 디코딩을 위한 방법을 수행하게 한다.

도면의 간단한 설명

[0031] 개시된 주제의 추가 특징, 성질 및 다양한 이점은 이하의 상세한 설명 및 첨부도면으로부터 더욱 명백해질 것이다.

- 도 1a는 인트라 예측 모드의 예시적인 서브세트의 개략도이다.
- 도 1b는 예시적인 인트라 예측 방향의 예시이다.
- 도 2는 하나의 예에서 현재 블록과 그 주변 공간 병합 후보의 개략도이다.
- 도 3은 일 실시예에 따른 통신 시스템(300)의 간략화된 블록도의 개략도이다.
- 도 4는 일 실시예에 따른 통신 시스템(400)의 간략화된 블록도의 개략도이다.
- 도 5는 일 실시예에 따른 디코더의 간략화된 블록도의 개략도이다.
- 도 6은 일 실시예에 따른 인코더의 간략화된 블록도의 개략도이다.
- 도 7은 다른 실시예에 따른 인코더의 블록도를 도시한다.
- 도 8은 다른 실시예에 따른 디코더의 블록도를 도시한다.
- 도 9는 본 개시의 실시예에 따른 적응적 해상도 변경(adaptive resolution change, ARC) 파라미터를 시그널링하기 위한 예를 도시한다.
- 도 10은 업샘플 또는 다운샘플 인자, 코드워드 및 Ext-Golomb 코드의 매핑을 위한 표(1000)의 예를 도시한다.
- 도 11은 본 개시의 일부 실시예에 따른 ARC 파라미터 시그널링의 일부 예를 도시한다.
- 도 12는 일부 예에서 PTL 신택스 요소 세트의 신택스 구조 예를 도시한다.
- 도 13은 일부 예에서 일반 제약 정보의 신택스 구조 예를 도시한다.
- 도 14a 및 도 14b는 본 개시의 일부 실시예에 따른 PTL 신택스 구조 및 일반 제약 정보 신택스 구조를 포함하는 PTL 정보의 예를 도시한다.
- 도 15a 및 도 15b는 본 개시의 일 실시예에 따른 일반 제약 정보 신택스 구조의 예를 도시한다.
- 도 16은 본 개시의 일부 실시예에 따른 일반 제약 정보의 신택스 구조를 도시한다.
- 도 17은 본 개시의 일부 실시예에 따른 시퀀스 파라미터 세트(SPS) 범위 확장의 신택스 구조 예를 도시한다.
- 도 18은 본 개시의 일 실시예에 따른 프로세스를 개략적으로 설명하는 흐름도를 도시한다.
- 도 19는 본 개시의 일 실시예에 따른 프로세스를 개략적으로 설명하는 흐름도를 도시한다.
- 도 20은 일 실시예에 따른 컴퓨터 시스템의 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0032] 도 3은 본 개시의 일 실시예에 따른 통신 시스템(300)의 간략화된 블록도를 나타낸다. 통신 시스템(300)은 예를 들어 네트워크(350)를 통해 서로 통신할 수 있는 복수의 단말 기기를 포함한다. 예를 들어, 통신 시스템(300)은 네트워크(350)를 통해 상호 연결된 제1의 단말 기기 쌍(310, 320)을 포함한다. 도 3의 예에서, 제1의 단말 기기 쌍(310, 320)은 데이터의 단방향 송신을 수행한다. 예를 들어, 단말 기기(310)는 네트워크(350)를 통해 다른 단말 기기(320)로의 송신을 위해 비디오 데이터(예: 단말 기기(310)에 의해 캡처된 비디오 픽처의 스트림)를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림의 형태로 송신될 수 있다. 단말 기기(320)는 네트워크(350)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처를 복원하고, 복원된 비디오 데이터에 따라 비디오 픽처를 표시할 수 있다. 단방향 데이터 송신은 미디어 서버 애플리케이션 등에 일반적일 수 있다.

[0033] 다른 예에서, 통신 시스템(300)은, 예를 들어 화상 회의 동안, 발생할 수 있는 코딩된 비디오 데이터의 양방향

송신을 수행하는 제2의 단말 기기 쌍(330, 340)을 포함한다. 데이터의 양방향 송신의 경우, 일례에서, 단말 기기 쌍(330, 340)의 각각의 단말 기기는 네트워크(350)를 통해 단말 기기 쌍(330, 340) 중 다른 단말 기기로의 송신을 위해 비디오 데이터(예: 단말 기기에 의해 캡처된 비디오 픽처의 스트림)를 코딩할 수 있다. 단말 기기 쌍(330, 340)의 각각의 단말 기기는 또한 단말 기기 쌍(330, 340)의 다른 단말 기기에 의해 송신되는 코딩된 비디오 데이터를 수신할 수 있으며, 코딩된 비디오 데이터를 디코딩하여 비디오를 복원할 수 있고 복원된 비디오 데이터에 따라 액세스 가능한 디스플레이 기기에 비디오 픽처를 표시할 수 있다.

[0034] 도 3의 예에서, 단말 기기(310, 320, 330, 340)는 서버, 개인용 컴퓨터 및 스마트폰으로 예시될 수 있지만, 본 개시의 원리는 그렇게 한정되지 않을 수 있다. 본 개시의 실시예는 랩톱 컴퓨터, 태블릿 컴퓨터, 미디어 플레이어 및/또는 전용 화상 회의 장비와의 애플리케이션을 찾아낸다. 네트워크(350)는, 예를 들어 유선(wired) 및/또는 무선 통신 네트워크를 포함한, 단말 기기(310, 320, 330, 340) 사이에서 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크를 나타낸다. 통신 네트워크(350)는 회선 교환 및/또는 패킷 교환 채널에서 데이터를 교환할 수 있다. 대표적인 네트워크로는 통신 네트워크, 근거리 통신망, 광역 통신망 및/또는 인터넷을 포함한다. 본 논의를 목적을 위해, 네트워크(350)의 아키텍처 및 토폴로지는 이하에서 설명되지 않는 한 본 개시의 동작에 중요하지 않을 수 있다.

[0035] 도 4는 개시된 주제에 대한 애플리케이션의 일례로서, 스트리밍 환경에서 비디오 인코더 및 비디오 디코더의 배치를 예시한다. 개시된 주제는 예를 들어, 화상 회의, 디지털 TV, CD, DVD, 메모리 스틱 등을 포함하는 디지털 미디어에 압축된 비디오의 저장 등을 포함한, 다른 비디오 지원 애플리케이션(video enabled application)에도 동일하게 적용될 수 있다.

[0036] 스트리밍 시스템은 예를 들어 압축되지 않은 비디오 픽처의 스트림(402)을 생성하는 비디오 소스(401), 예를 들어 디지털 카메라를 포함할 수 있는 캡처 서브시스템(413)을 포함할 수 있다. 일례에서, 비디오 픽처의 스트림(402)은 디지털 카메라로 촬영되는 샘플을 포함한다. 인코딩된 비디오 데이터(404)(또는 코딩된 비디오 비트 스트림)와 비교할 때 높은 데이터 볼륨을 강조하기 위해 굵은 선으로 표시된 비디오 픽처의 스트림(402)은 비디오 소스(401)에 결합된 비디오 인코더(403)를 포함하는 전자 기기(420)에 의해 처리될 수 있다. 비디오 인코더(403)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 측면들을 가능하게 하거나 구현할 수 있는 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다.

[0037] 비디오 픽처의 스트림(402)과 비교할 때 더 낮은 데이터 볼륨을 강조하기 위해 가는 선으로 표시된 인코딩된 비디오 데이터(404)(또는 인코딩된 비디오 비트스트림(404))은 나중에 사용하기 위해 스트리밍 서버(405)에 저장될 수 있다. 도 4에서의 클라이언트 서브시스템(406, 408)과 같은, 하나 이상의 스트리밍 클라이언트 서브시스템은 인코딩된 비디오 데이터(404)의 사본(407, 409)을 검색하기 위해 스트리밍 서버(405)에 액세스할 수 있다. 클라이언트 서브시스템(406)은 예를 들어, 전자 기기(430)에 비디오 디코더(410)를 포함할 수 있다. 비디오 디코더(410)는 인코딩된 비디오 데이터의 인커밍 사본(incoming copy)(407)을 디코딩하고 디스플레이(412)(예: 디스플레이 스크린) 또는 다른 렌더링 기기(도시되지 않음)에서 렌더링될 수 있는 비디오 픽처(411)의 아웃고잉 스트림을 생성한다. 일부 스트리밍 시스템에서, 인코딩된 비디오 데이터(404, 407, 409)(예: 비디오 비트 스트림)은 특정 비디오 코딩/압축 표준에 따라 인코딩될 수 있다. 이러한 표준의 예로는 ITU-T Recommendation H.265를 포함한다. 일례에서, 개발중인 비디오 코딩 표준은 비공식적으로 VVC(Versatile Video Coding)로 알려져 있다. 개시된 주제는 VVC의 컨텍스트에 사용될 수 있다.

[0038] 전자 기기(420, 430)는 다른 구성요소(도시되지 않음)를 포함할 수 있음에 유의한다. 예를 들어, 전자 기기(420)는 비디오 디코더(도시되지 않음)를 포함할 수 있고, 전자 기기(430)도 비디오 인코더(도시되지 않음)를 포함할 수 있다.

[0039] 도 5는 본 개시의 일 실시예에 따른 비디오 디코더(510)의 블록도를 도시한다. 비디오 디코더(510)는 전자 기기(530)에 포함될 수 있다. 전자 기기(530)는 수신기(531)(예: 수신 회로)를 포함할 수 있다. 비디오 디코더(510)는 도 4의 예에서의 비디오 디코더(410) 대신에 사용될 수 있다.

[0040] 수신기(531)는 비디오 디코더(510)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스를 수신할 수 있고; 동일한 실시예 또는 다른 실시예에서, 하나의 코딩된 비디오 시퀀스를 한 번에 수신할 수 있으며, 여기서 각각의 코딩된 비디오 시퀀스의 디코딩은 다른 코딩된 비디오 시퀀스와 독립적이다. 코딩된 비디오 시퀀스는 인코딩된 비디오 데이터를 저장하는 저장 기기에 대한 하드웨어/소프트웨어 링크일 수 있는 채널(501)로부터 수신될 수 있다. 수신기(531)는 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림과 함께 인코딩된 비디오 데이터를 수신할 수 있으며, 이는 엔터티(도시되지 않음)를 사용하여 그들 각각에 포워딩될 수 있다.

수신기(531)는 코딩된 비디오 시퀀스를 다른 데이터와 분리할 수 있다. 네트워크 지터(network jitter)를 방지하기 위해, 버퍼 메모리(515)가 수신기(531)와 엔트로피 디코더/과서(520)(이하 "과서(520)") 사이에 결합될 수 있다. 특정 애플리케이션에서, 버퍼 메모리(515)는 비디오 디코더(510)의 일부이다. 다른 애플리케이션의 경우에는 비디오 디코더(510) 외부에 있을 수 있다(도시되지 않음).

[0041] 또 다른 애플리케이션의 경우에는 예를 들어 네트워크 지터를 방지하기 위해 비디오 디코더(510) 외부에 버퍼 메모리(도시되지 않음)가 있을 수 있으며, 예를 들어 플레이아웃 타이밍(playout timing)을 처리하기 위해 비디오 디코더(510) 내부에 추가로 다른 버퍼 메모리(515)가 있을 수 있다. 수신기(531)가 충분한 대역폭과 제어성(controllability)을 갖는 저장/포워딩 기기로부터 또는 등시성 네트워크(isosynchronous network)로부터 데이터를 수신하고 있을 때, 버퍼 메모리(515)는 필요하지 않거나 작을 수 있다. 인터넷과 같은 최선형 패킷 네트워크(best effort packet network)에서 사용하기 위해, 버퍼 메모리(515)가 필요할 수 있고, 비교적 클 수 있으며 적응적인 크기가 유리할 수 있으며, 운영 체제 또는 비디오 디코더(510)의 외부의 유사한 요소(도시되지 않음)에서 적어도 부분적으로 구현될 수 있다.

[0042] 비디오 디코더(510)는 코딩된 비디오 시퀀스로부터 심볼(521)을 재구성하기 위해 과서(520)를 포함할 수 있다. 이러한 심볼의 카테고리로는 비디오 디코더(510)의 동작을 관리하는 데 사용되는 정보와, 도 5에 도시된 바와 같이, 전자 기기(530)의 일체화된 부분이 아니라 전자 기기(530)에 결합될 수 있는 렌더 기기(512)(예: 디스플레이 스크린)와 같은 렌더링 기기를 제어하기 위한 잠재적인 정보를 포함한다. 렌더링 기기(들)에 대한 제어 정보는 보충 강화 정보(Supplemental Enhancement Information, SEI 메시지) 또는 비디오 유용성 정보(Video Usability Information, VUI) 파라미터 세트 프래그먼트(도시되지 않음)의 형태일 수 있다. 과서(520)는 수신되는 코딩된 비디오 시퀀스를 과싱/엔트로피 디코딩할 수 있다. 코딩된 비디오 시퀀스의 코딩은 비디오 코딩 기술 또는 표준에 따른 것일 수 있으며, 가변 길이 코딩, Huffman 코딩, 컨텍스트 민감도가 있거나 없는 산술 코딩 등을 포함한 다양한 원리를 따를 수 있다. 과서(520)는 비디오 디코더에서의 픽셀의 서브그룹 중 적어도 하나에 대한 서브그룹 파라미터의 세트를, 그 그룹에 대응하는 적어도 하나의 파라미터에 기초하여, 코딩된 비디오 시퀀스로부터 추출할 수 있다. 서브그룹은 픽처의 그룹(Group of Picture, GOP), 픽처, 타일(tile), 슬라이스(slice), 매크로블록(macroblock), 코딩 유닛(Coding Unit, CU), 블록, 변환 유닛(Transform Unit, TU), 예측 유닛(Prediction Unit, PU) 등을 포함할 수 있다. 과서(520)는 또한 변환 계수, 양자화기 파라미터 값, 움직임 벡터 등과 같은 코딩된 비디오 시퀀스 정보로부터 추출할 수 있다.

[0043] 과서(520)는 버퍼 메모리(515)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/과싱 동작을 수행하여 심볼(521)을 생성할 수 있다.

[0044] 심볼(521)의 재구성은 코딩된 비디오 픽처 또는 그 일부(예: 인터 및 인트라 픽처, 인터 및 인트라 블록)의 유형 및 기타 인자에 따라 다수의 다른 유닛과 관련 있을 수 있다. 관련되는 유닛 및 어떻게 관련되는지는 과서(520)에 의해 코딩된 비디오 시퀀스로부터 과싱된 서브그룹 제어 정보에 의해 제어될 수 있다. 과서(520)와 그 아래의 다수의 유닛 사이의 그러한 서브그룹 제어 정보의 흐름은 단순함을 위해 나타내지 않았다.

[0045] 이미 언급된 기능 블록을 넘어, 비디오 디코더(510)는 아래에 설명된 바와 같이 개념적으로 다수의 기능 유닛으로 세분될 수 있다. 상업적 제약 하에서 작동하는 실제 구현에서, 이러한 유닛 중 다수는 서로 밀접하게 상호작용하며, 적어도 부분적으로, 서로 통합될 수 있다. 그러나 개시된 주제를 명확하게 설명하기 위해서는 아래의 기능 유닛들로의 개념적 세분화가 적절하다.

[0046] 제1 유닛은 스케일러/역변환 유닛(551)이다. 스케일러/역변환 유닛(551)은 과서(520)로부터의 심볼(들)(521)인 사용할 역변환, 블록 크기, 양자화 인자, 양자화 스케일링 행렬 등을 포함한 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 스케일러/역변환 유닛(551)은 집성기(agggregator)(555)에 입력될 수 있는, 샘플 값을 포함하는 블록을 출력할 수 있다.

[0047] 경우에 따라서는, 스케일러/역변환 유닛(551)의 출력 샘플은 인트라 코딩된 블록에 관련될 수 있다. 즉, 이전에 재구성된 픽처로부터의 예측 정보를 사용하지 않지만, 현재 픽처의 이전에 재구성된 부분으로부터의 예측 정보를 사용할 수 있는 블록이다. 이러한 예측 정보는 인트라 픽처 예측 유닛(552)에 의해 제공될 수 있다. 경우에 따라서는, 인트라 픽처 예측 유닛(552)은 현재 픽처 버퍼(558)에서 인출되는 이미 재구성된 주변 정보를 사용하여, 재구성중인 블록과 동일한 크기 및 모양의 블록을 생성한다. 현재 픽처 버퍼(558)는, 예를 들어 부분적으로 재구성된 현재 픽처 및/또는 완전히 재구성된 현재 픽처를 버퍼링한다. 집성기(555)는, 경우에 따라서는, 샘플 단위로, 인트라 예측 유닛(552)이 생성한 예측 정보를 스케일러/역변환 유닛(551)에 의해 제공되는 출력 샘플 정보에 추가할 수 있다.

- [0048] 다른 경우에, 스케일러/역변환 유닛(551)의 출력 샘플은 인터 코딩되고 잠재적으로 움직임 보상된 블록에 관련될 수 있다. 이러한 경우, 움직임 보상 예측 유닛(553)은 인터 픽처 예측에 사용되는 샘플을 인출하기 위해 참조 픽처 메모리(557)에 액세스할 수 있다. 인출된 샘플을 블록에 관련되는 심볼(521)에 따라 움직임 보상한 후, 이러한 샘플은 짐성기(555)에 의해 스케일러/역변환 유닛(551)의 출력(이 경우 잔차 샘플 또는 잔차 신호라고 할 수 있음)에 더해져, 출력 샘플 정보를 생성할 수 있다. 움직임 보상 예측 유닛(553)이 예측 샘플을 인출하는 참조 픽처 메모리(557) 내의 주소는 움직임 벡터에 의해 제어될 수 있으며, 예를 들어 X, Y 및 참조 픽처 성분을 가질 수 있는 심볼(521)의 형태로 움직임 보상 예측 유닛(553)에 사용할 수 있다. 움직임 보상은 또한 서브 샘플의 정확한 움직임 벡터가 사용되고 있을 때에 참조 픽처 메모리(557)로부터 인출된 샘플 값의 보간, 움직임 벡터 예측 메커니즘 등도 포함할 수 있다.
- [0049] 짐성기(555)의 출력 샘플은 루프 필터 유닛(556)에서 다양한 루프 필터링 기술의 대상이 될 수 있다. 비디오 압축 기술은 코딩된 비디오 시퀀스(코딩된 비디오 비트 스트림이라고도 함)에 포함된 파라미터에 의해 제어되고 파서(520)로부터의 심볼(521)로서 루프 필터 유닛(556)에 사용 가능하게 되는 인 루프(in-loop) 필터 기술을 포함할 수 있지만, 또한 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서상) 부분을 디코딩하는 동안 획득된 메타 정보에 응답할 수 있을 뿐만 아니라 이전에 재구축되고 루프 필터링된 샘플 값에 응답할 수도 있다.
- [0050] 루프 필터 유닛(556)의 출력은 렌더 기기(512)에 출력될 수 있고 미래의 인터 픽처 예측에 사용하기 위해 참조 픽처 메모리(557)에 저장될 수 있는 샘플 스트림일 수 있다.
- [0051] 일단 완전히 재구축된 특정 코딩된 픽처는 나중의 예측을 위한 참조 픽처로서 사용할 수 있다. 예를 들어, 현재 픽처에 대응하는 코딩된 픽처가 완전히 재구축되고 코딩된 픽처가 참조 픽처로 식별되면(예: 파서(520)에 의해), 현재 픽처 버퍼(558)는 참조 픽처 메모리(557)의 일부가 될 수 있고, 새로운 현재 픽처 버퍼는 다음 코딩된 픽처의 재구축을 시작하기 전에 재할당될 수 있다.
- [0052] 비디오 디코더(510)는 ITU-T Rec. H.265와 같은, 표준에서의 미리 정해진 비디오 압축 기술에 따라 디코딩 동작을 수행할 수 있다. 코딩된 비디오 시퀀스는 비디오 압축 기술 또는 표준의 선택스와 비디오 압축 기술에 문서화된 프로파일 모두에 준거한다는 점에서, 사용되는 비디오 압축 기술 또는 표준에 의해 지정된 선택스를 따를 수 있다. 구체적으로, 프로파일은 비디오 압축 기술 또는 표준에서 사용 가능한 모든 도구 중에서 그 프로파일에서 사용할 수 있는 유일한 도구로서 특정 도구를 선택할 수 있다. 또한 준거에 필요한 것은 코딩된 비디오 시퀀스의 복잡도가 비디오 압축 기술 또는 표준의 레벨에 정의된 범위 내에 있는 것이다. 경우에 따라서는, 레벨은 최대 픽처 크기, 최대 프레임 레이트, 최대 재구축 샘플 레이트(예: 초당 메가 샘플로 측정됨), 최대 참조 픽처 크기 등을 제한한다. 레벨에 의해 설정된 제한은, 경우에 따라서는 코딩된 비디오 시퀀스에서 시그널링된 가상 참조 디코더(Hypothetical Reference Decoder, HRD) 버퍼 관리를 위한 HRD 사양 및 메타 데이터를 통해 추가로 제한될 수 있다.
- [0053] 일 실시예에서, 수신기(531)는 인코딩된 비디오와 함께 추가(중복) 데이터를 수신할 수 있다. 추가 데이터는 코딩된 비디오 시퀀스(들)의 일부로서 포함될 수 있다. 추가 데이터는 데이터를 적절하게 디코딩하고/하거나 원본 비디오 데이터를 더 정확하게 재구축하기 위해 비디오 디코더(510)에 의해 사용될 수 있다. 추가 데이터는, 예를 들어 시간 계층, 공간 계층 또는 SNR 강화 계층(enhancement layer), 중복 슬라이스(redundant slice), 중복 픽처(redundant picture), 순방향 오류 정정 코드(forward error correction code) 등의 형태일 수 있다.
- [0054] 도 6은 본 개시의 일 실시예에 따른 비디오 인코더(603)의 블록도를 도시한다. 비디오 인코더(603)는 전자 기기(620)에 포함된다. 전자 기기(620)는 송신기(640)(예: 송신 회로)를 포함한다. 비디오 인코더(603)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용될 수 있다.
- [0055] 비디오 인코더(603)는 비디오 인코더(603)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(601)(도 6의 예에서 전자 기기(620)의 일부가 아님)로부터 비디오 샘플을 수신할 수 있다. 다른 예에서, 비디오 소스(601)는 전자 기기(620)의 일부이다.
- [0056] 비디오 소스(601)는 임의의 적절한 비트 심도(예: 8비트, 10비트, 12비트, ...), 임의의 색 공간(color space)(예: BT.601 Y CrCb, RGB, ...) 및 임의의 적절한 샘플링 구조(예: Y CrCb 4:2:0, Y CrCb 4:4:4)일 수 있는 디지털 비디오 샘플 스트림의 형태로 비디오 인코더(603)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서버 시스템에서, 비디오 소스(601)는 미리 준비된 비디오를 저장할 수 있는 저장 기기일 수 있다. 영상 회의 시스템에서, 비디오 소스(601)는 로컬 이미지 정보를 비디오 시퀀스로서 캡처하는 카메라일 수 있다.

비디오 데이터는 순차적으로 보여질 때 움직임 부여하는 복수의 개별 픽처로서 제공될 수 있다. 픽처 자체는 픽셀의 공간 배열(spatial array)로 구성될 수 있으며, 여기서 각각의 픽셀은 사용중인 샘플링 구조, 색 공간 등에 따라 하나 이상의 샘플을 포함할 수 있다. 당업자라면 픽셀과 샘플 사이의 관계를 쉽게 이해할 수 있을 것이다. 아래 설명은 샘플에 중점을 둔다.

[0057] 일 실시예에 따르면, 비디오 인코더(603)는 소스 비디오 시퀀스의 픽처를 실시간으로 또는 애플리케이션에 의해 요구되는 임의의 다른 시간 제약하에서 코딩된 비디오 시퀀스(643)로 코딩하고 압축할 수 있다. 적절한 코딩 속도를 강제하는 것은 제어기(650)의 한 가지 기능이다. 일부 실시예에서, 제어기(650)는 후술되는 바와 같이 다른 기능 유닛을 제어하고 다른 기능 유닛에 기능적으로 결합된다. 명확함을 위해 그 결합은 표시되지 않는다. 제어기(650)에 의해 설정되는 파라미터로는 레이트 제어 관련 파라미터(픽처 스킵, 양자화기, 레이트 왜곡 최적화 기술의 램다 값, ...), 픽처 크기, 픽처 그룹(GOP) 레이아웃, 최대 움직임 벡터 검색 범위, 등을 포함할 수 있다. 제어기(650)는 특정 시스템 설계에 최적화된 비디오 인코더(603)에 관련 있는 다른 적절한 기능을 갖도록 구성될 수 있다.

[0058] 일부 실시예에서, 비디오 인코더(603)는 코딩 루프에서 동작하도록 구성된다. 과도하게 단순화된 설명으로서, 일례에서, 코딩 루프는 소스 코더(630)(예: 코딩될 입력 픽처에 기초하여 심볼 스트림과 같은 심볼 및 참조 픽처(들)의 생성을 담당), 및 비디오 인코더(603)에 내장된 (로컬) 디코더(633)를 포함할 수 있다. 디코더(633)는 (원격) 디코더도 생성하였을 것(심볼과 코딩된 비디오 비트 스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술에서 무손실일 수 있는 것)과 유사한 방식으로 샘플 데이터를 생성하기 위해 심볼을 재구성한다. 재구성된 샘플 스트림(샘플 데이터)은 참조 픽처 메모리(634)에 입력된다. 심볼 스트림의 디코딩은 디코더 위치(로컬 또는 원격)와 무관하게 비트가 정확한 결과(bit-exact result)로 이어지므로, 참조 픽처 메모리(634)의 내용도 로컬 인코더와 원격 인코더 사이에서 비트가 정확하다(bit exact). 다시 말해, 인코더의 예측 부분은 디코딩 동안에 예측을 사용하는 경우에 디코더가 "인식하는(see)" 것과 정확히 동일한 샘플 값을 참조 픽처 샘플로서 "인식한다". 참조 픽처 동시성(reference picture synchronicity)(및 예를 들어 채널 오차로 인해 동시성이 유지될 수 없는 경우, 결과 드리프트)의 이러한 기본 원리는 일부 관련 기술에서도 사용된다.

[0059] "로컬" 디코더(633)의 동작은 비디오 디코더(510)와 같은, "원격" 디코더의 동작과 동일할 수 있으며, 이는 이미 도 5와 관련하여 상세하게 설명되었다. 도 5를 간단히 다시 참조하면, 심볼이 이용 가능하고 엔트로피 코더(645) 및 파서(420)에 의해 코딩된 비디오 시퀀스로의 심볼의 인코딩/디코딩이 무손실일 수 있으므로, 비디오 디코더(510)의 엔트로피 디코딩 부분은, 버퍼 메모리(515) 및 파서(520)를 포함하여, 로컬 디코더(633)에 인코더에 전적으로 구현되지 않을 수 있다.

[0060] 이 시점에서 관찰할 수 있는 것은 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 모든 디코더 기술이 반드시 상응하는 인코더에 실질적으로 동일한 기능적 형태로 존재할 필요가 있다는 것이다. 이러한 이유로, 개시된 주제는 디코더 동작에 중점을 둔다. 따라서 인코더 기술에 대한 설명은 포괄적으로 설명된 디코더 기술의 반대이므로 생략될 수 있다. 특정 영역에서만 더 자세한 설명이 필요하며 아래에 제공된다.

[0061] 동작하는 동안, 일부 예에서, 소스 코더(630)는 "참조 픽처"로 지정된 비디오 시퀀스로부터 하나 이상의 이전에 코딩된 픽처를 참조하여 입력 픽처를 예측적으로 코딩하는 움직임 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(632)은 입력 픽처에 대한 예측 참조(들)로서 선택될 수 있는 입력 픽처의 픽셀 블록과 참조 픽처(들)의 픽셀 블록 간의 차이를 코딩한다.

[0062] 로컬 비디오 디코더(633)는 소스 코더(630)에 의해 생성된 심볼에 기초하여, 참조 픽처로서 지정될 수 있는 픽처의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(632)의 동작은 유리하게 손실 프로세스일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 6에 도시되지 않음)에서 디코딩될 수 있을 때, 재구성된 비디오 시퀀스는 일반적으로 약간의 오류가 있는 소스 비디오 시퀀스의 복제본(replica)일 수 있다. 로컬 비디오 디코더(633)는 참조 픽처에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스를 복제하고, 재구성된 참조 픽처가 참조 픽처 캐시(634)에 저장되도록 할 수 있다. 이러한 방식으로, 비디오 인코더(603)는 원단(far-end) 비디오 디코더(송신 오류 없음)에 의해 획득될 재구성된 참조 픽처로서 공통 콘텐츠를 갖는 재구성된 참조 픽처의 사본을 로컬에 저장할 수 있다.

[0063] 예측기(predictor)(635)는 코딩 엔진(632)에 대한 예측 검색을 수행할 수 있다. 즉, 코딩될 새로운 픽처를 위해, 예측기(635)는 참조 픽처 메모리(634)에서 새로운 픽처에 대한 적절한 예측 참조의 역할을 할 수 있는 샘플 데이터(후보 참조 픽셀 블록임), 또는 참조 픽처 움직임 벡터, 블록 모양 등과 같은 특정 메타데이터를 검색할 수 있다. 예측기(635)는 적절한 예측 참조를 찾기 위해 샘플 블록별 픽셀 블록(sample block-by-pixel

block) 단위로 동작할 수 있다. 경우에 따라서는, 예측기(635)에 의해 획득된 검색 결과에 의해 결정되는 바와 같이, 입력 픽처는 참조 픽처 메모리(634)에 저장된 다수의 참조 픽처로부터 끌어낸 예측 참조를 가질 수 있다.

- [0064] 제어기(650)는, 예를 들어 비디오 데이터를 인코딩하기 위해 사용되는 파라미터 및 서브그룹 파라미터의 설정을 포함하는 소스 코더(630)의 코딩 동작을 관리할 수 있다.
- [0065] 진술한 모든 기능 유닛의 출력은 엔트로피 코더(645)에서의 엔트로피 코딩 대상일 수 있다. 엔트로피 코더(645)는 호프만(Huffman) 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술에 따라 심볼을 무손실 압축함으로써, 다양한 기능 유닛에 의해 생성된 심볼을 코딩된 비디오 시퀀스로 변환한다.
- [0066] 송신기(640)는 엔트로피 코더(645)에 의해 생성된 코딩된 비디오 시퀀스(들)를 버퍼링하여, 인코딩된 비디오 데이터를 저장할 저장 기기에 대한 하드웨어/소프트웨어 링크일 수 있는 통신 채널(660)을 통한 송신을 준비할 수 있다. 송신기(640)는 비디오 코더(603)로부터 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어 코딩된 오디오 데이터 및/또는 보조 데이터 스트림(소스는 도시되지 않음)과 병합할 수 있다.
- [0067] 제어기(650)는 비디오 인코더(603)의 동작을 관리할 수 있다. 코딩하는 동안, 제어기(650)는 각각의 코딩된 픽처에 특정 코딩된 픽처 유형을 할당할 수 있으며, 이는 각각의 픽처에 적용될 수 있는 코딩 기술에 영향을 미칠 수 있다. 예를 들어, 픽처는 종종 다음 픽처 유형 중 하나로서 지정될 수 있다:
- [0068] 인트라 픽처(Intra Picture: I 픽처)는 예측 소스로서 시퀀스 내의 어떤 다른 픽처도 사용하지 않고 코딩 및 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱은, 예를 들어 독립 디코더 리프레시 픽처(Independent Decoder Refresh Picture, "IDR")을 포함한, 상이한 유형의 인트라 픽처를 허용한다. 당업자는 I 픽처의 이러한 변형 및 각각의 적용 및 특징을 알고 있다.
- [0069] 예측 픽처(Predictive picture: P 픽처)는 각각의 블록의 샘플 값을 예측하기 위해 최대 하나의 움직임 벡터 및 참조 색인을 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다.
- [0070] 양 방향 예측 픽처(Bi-directionally Predictive Picture: B 픽처)는 각각의 블록의 샘플 값을 예측하기 위해 최대 두 개의 움직임 벡터 및 참조 색인을 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다. 유사하게, 다중 예측 픽처(multiple-predictive picture)는 단일 블록의 재구축을 위해 두 개보다 많은 참조 픽처와 연관 메타 데이터를 사용할 수 있다.
- [0071] 소스 픽처는 일반적으로 공간적으로 복수의 샘플 블록(예: 4×4, 8×8, 4×8 또는 16×16 샘플 블록)으로 세분화되고 블록 단위로 코딩될 수 있다. 블록은 블록의 픽처 각각에 적용된 코딩 할당에 의해 결정된 대로 다른 (이미 코딩된) 블록을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 픽처의 블록은 비 예측적으로 코딩되거나, 동일한 픽처의 이미 코딩된 블록(공간 예측(spatial prediction) 또는 인트라 예측)을 참조하여 예측적으로 코딩될 수 있다. P 픽처의 픽셀 블록은 이전에 코딩된 하나의 참조 픽처를 참조하여 공간 예측을 통해 또는 시간 예측(temporal prediction)을 통해 예측적으로 코딩될 수 있다. B 픽처의 블록은 이전에 코딩된 하나 또는 두 개의 참조 픽처를 참조하여 공간 예측을 통해 또는 시간 예측을 통해 예측적으로 코딩될 수 있다.
- [0072] 비디오 인코더(603)는 ITU-T Rec. H.265와 같은 미리 결정된 비디오 코딩 기술 또는 표준에 따라 코딩 동작을 수행할 수 있다. 그 동작에서, 비디오 인코더(603)는 입력 비디오 시퀀스에서의 시간적 및 공간적 중복성을 이용하는 예측 코딩 동작을 포함한, 다양한 압축 동작을 수행할 수 있다. 따라서, 코딩된 비디오 데이터는 사용되는 비디오 코딩 기술 또는 표준에 의해 지정된 신택스를 따를 수 있다.
- [0073] 일 실시예에서, 송신기(640)는 인코딩된 비디오와 함께 추가 데이터를 송신할 수 있다. 소스 코더(630)는 코딩된 비디오 시퀀스의 일부로서 이러한 데이터를 포함할 수 있다. 추가 데이터는 시간/공간/SNR 강화 계층, 중복 픽처 및 슬라이스와 같은 다른 형태의 중복 데이터, SEI 메시지, VUI 파라미터 세트 프래그먼트 등을 포함할 수 있다.
- [0074] 비디오는 시간적 시퀀스로 복수의 소스 픽처(비디오 픽처)로서 캡처될 수 있다. 인트라 픽처 예측(종종 인트라 예측으로 축약됨)은 주어진 픽처에서 공간적 상관관계를 이용하고, 인터 픽처 예측(inter-picture prediction)은 픽처 사이의 (시간적 또는 기타) 상관관계를 이용한다. 일례에서, 현재 픽처라고 하는 인코딩/디코딩 중인 특정 픽처는 블록으로 파티셔닝될 수 있다. 현재 픽처 내의 블록이 이전에 코딩되고 비디오에서 여전히 버퍼링된 참조 픽처의 참조 블록과 유사한 경우, 현재 픽처 내의 블록은 움직임 벡터라고 하는 벡터에 의해 코딩될 수 있다. 움직임 벡터는 참조 픽처 내의 참조 블록을 가리키며, 다수의 참조 픽처가 사용 중인 경우, 참조 픽처를 식별하는 3차원을 가질 수 있다.

- [0075] 일부 실시예에서, 양-예측 기술(bi-prediction technique)이 인터 픽처 예측에 사용될 수 있다. 양-예측 기술에 따르면, 비디오에서 디코딩 순서상 현재 픽처에 둘다 선행하는 (하지만 각각 표시 순서상 과거 또는 미래일 수 있는) 제1 참조 픽처와 제2 참조 픽처와 같은, 2개의 참조 픽처가 사용된다. 현재 픽처의 블록은 제1 참조 픽처의 제1 참조 블록을 가리키는 제1 움직임 벡터 및 제2 참조 픽처의 제2 참조 블록을 가리키는 제2 움직임 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 예측될 수 있다.
- [0076] 또한, 인터 픽처 예측에 병합 모드 기술을 사용하여 코딩 효율을 향상시킬 수 있다.
- [0077] 본 개시의 일부 실시예에 따르면, 인터 픽처 예측 및 인트라 픽처 예측과 같은 예측은 블록 단위로 수행된다. 예를 들어, HEVC 표준에 따라, 비디오 픽처의 시퀀스에서의 픽처는 압축을 위해 코딩 트리 유닛(coding tree unit, CTU)로 파티셔닝되고, 픽처 내의 CTU는 64×64 픽셀, 32×32 픽셀 또는 16×16 픽셀과 같이, 동일한 크기를 가질 수 있다. 일반적으로, CTU는 루마 CTB 1개와 크로마 CTB 2개인 코딩 트리 블록(coding tree block, CTB)을 포함할 수 있다. 각각의 CTU는 반복적으로 하나 또는 복수의 코딩 유닛(CU)으로 쿼드트리 분할될 수 있다. 예를 들어 64×64 픽셀의 CTU는 64×64 픽셀의 CU 1개 또는 32×32 픽셀의 CU 4개 또는 16×16 픽셀의 CU 16개로 분할될 수 있다. 일례에서, 각각의 CU는 인터 예측 유형 또는 인트라 예측 유형과 같은 CU에 대한 예측 유형을 결정하기 위해 분석될 수 있다. CU는 시간적 및/또는 공간 예측 가능성에 따라 하나 이상의 예측 유닛(prediction unit, PU)으로 분할될 수 있다. 일반적으로, 각각의 PU는 루마 예측 블록(PB)과 두 개의 크로마 PB를 포함한다. 일 실시예에서, 코딩(인코딩/디코딩)에서의 예측 동작은 예측 블록의 단위로 수행된다. 루마 예측 블록을 예측 블록의 일례로 사용하면, 예측 블록은 8×8 픽셀, 16×16 픽셀, 8×16 픽셀, 16×8 픽셀 등과 같은, 픽셀에 대한 값(예: 루마 값)의 행렬을 포함한다.
- [0078] 도 7은 본 개시의 다른 실시예에 따른 비디오 인코더(703)의 도면을 도시한다. 비디오 인코더(703)는 비디오 픽처의 시퀀스에서 현재 비디오 픽처 내의 샘플 값의 처리 블록(예: 예측 블록)을 수신하고, 처리 블록을 코딩된 비디오 시퀀스의 일부인 코딩된 픽처로 인코딩하도록 구성된다. 일례에서, 비디오 인코더(703)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용될 수 있다.
- [0079] HEVC의 예에서, 비디오 인코더(703)는 8×8 샘플의 예측 블록 등과 같은 처리 블록에 대한 샘플 값의 행렬을 수신한다. 비디오 인코더(703)는 처리 블록이, 예를 들어 레이트 왜곡 최적화(rate-distortion optimization)를 사용하는 인트라 모드, 인터 모드 또는 양-예측 모드를 사용하여 가장 잘 코딩되는지를 판정한다. 처리 블록이 인트라 모드로 코딩되는 것으로 결정되는 경우, 비디오 인코더(703)는 처리 블록을 코딩된 픽처로 인코딩하기 위해 인트라 예측 기술을 사용할 수 있고; 처리 블록이 인터 모드 또는 양-예측 모드로 코딩되는 것으로 결정되는 경우, 비디오 인코더(703)는 처리 블록을 코딩된 픽처로 인코딩하기 위해 인터 예측 또는 양-예측 기술을 각각 사용할 수 있다. 특정 비디오 코딩 기술에서, 병합 모드는 움직임 벡터가 예측기 외부의 코딩된 움직임 벡터 성분의 이점 없이 하나 이상의 움직임 벡터 예측기로부터 도출되는 인터 픽처 예측의 서브모드일 수 있다. 특정 다른 비디오 코딩 기술에서, 대상 블록(subject block)에 적용 가능한 움직임 벡터 성분이 존재할 수 있다. 일례에서, 비디오 인코더(703)는 처리 블록의 예측 모드를 결정하기 위한 모드 결정 모듈(도시되지 않음)과 같은, 구성요소를 포함할 수 있다.
- [0080] 도 7의 예에서, 비디오 인코더(703)는 인터 인코더(730), 인트라 인코더(722), 잔차 계산기(723), 스위치(726), 잔차 인코더(724), 일반 제어기(721), 및 도 7에 도시된 바와 같이 함께 결합된 엔트로피 인코더(725)를 포함한다.
- [0081] 인터 인코더(730)는 현재 블록(예: 처리 블록)의 샘플을 수신하고, 블록을 참조 픽처 내의 하나 이상의 참조 블록(예: 이전 픽처 및 이후 픽처 내의 블록)과 비교하고, 인터 예측 정보(예: 인터 인코딩 기술에 따른 중복 정보의 설명, 움직임 벡터, 병합 모드 정보)를 생성하고, 임의의 적절한 기술을 사용하여 인터 예측 정보에 기초하여 인터 예측 결과(예: 예측된 블록)를 계산하도록 구성된다. 일부 예에서, 참조 픽처는 인코딩된 비디오 정보에 기초하여 디코딩되는 디코딩된 참조 픽처이다.
- [0082] 인트라 인코더(722)는 현재 블록(예: 처리 블록)의 샘플을 수신하고, 경우에 따라서는 그 블록을 동일한 픽처에서 이미 코딩된 블록과 비교하고, 변환 후 양자화된 계수를, 경우에 따라서는 또한 인트라 예측 정보(예: 하나 이상의 인트라 인코딩 기술에 따른 인트라 예측 방향 정보)를 생성하도록 구성된다. 일례에서, 인트라 인코더(722)는 또한 동일한 픽처의 인트라 예측 정보 및 참조 블록에 기초하여 인트라 예측 결과(예: 예측된 블록)를 계산한다.
- [0083] 일반 제어기(721)는 일반 제어 데이터를 결정하고 일반 제어 데이터에 기초하여 비디오 인코더(703)의 다른 구

성요소를 제어하도록 구성된다. 일례에서, 일반 제어기(721)는 블록의 모드를 결정하고, 모드에 기초하여 제어 신호를 스위치(726)에 제공한다. 예를 들어, 모드가 인트라 모드인 경우, 일반 제어기(721)는 스위치(726)를 제어하여 잔차 계산기(723)에서 사용하기 위한 인트라 모드 결과를 선택하도록 하고, 엔트로피 인코더(725)를 제어하여 인트라 예측 정보를 선택하여 비트 스트림에 인트라 예측 정보를 포함시키도록 하고; 모드가 인터 모드인 경우, 일반 제어기(721)는 스위치(726)를 제어하여 잔차 계산기(723)에서 사용하기 위한 인터 예측 결과를 선택하도록 하고, 엔트로피 인코더(725)를 제어하여 인터 예측 정보를 선택하여 인터 예측 정보를 비트 스트림에 포함시키도록 한다.

[0084] 잔차 계산기(723)는 수신된 블록과 인트라 인코더(722) 또는 인터 인코더(730)로부터 선택된 예측 결과 사이의 차이(잔차 데이터)를 계산하도록 구성된다. 잔차 인코더(724)는 잔차 데이터에 기초하여 동작하여 잔차 데이터를 인코딩하여 변환 계수를 생성하도록 구성된다. 일례에서, 잔차 인코더(724)는 잔차 데이터를 공간 도메인에서 주파수 도메인으로 변환하고, 변환 계수를 생성하도록 구성된다. 그런 다음 변환 계수를 얻기 위한 양자화 처리를 거친다. 다양한 실시예에서, 비디오 인코더(703)는 또한 잔차 디코더(728)를 포함한다. 잔차 디코더(728)는 역변환을 수행하고, 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(722) 및 인터 인코더(730)에 의해 적절하게 사용될 수 있다. 예를 들어, 인터 인코더(730)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기초하여 디코딩된 블록을 생성할 수 있고, 인트라 인코더(722)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기초하여 디코딩된 블록을 생성할 수 있다. 디코딩된 블록은 디코딩된 픽처를 생성하기 위해 적절하게 처리되고 디코딩된 픽처는 메모리 회로(도시되지 않음)에서 버퍼링될 수 있고 일부 예에서 참조 픽처로서 사용될 수 있다.

[0085] 엔트로피 인코더(725)는 인코딩된 블록을 포함하도록 비트 스트림을 포맷하도록 구성된다. 엔트로피 인코더(725)는 HEVC 표준과 같은 적절한 표준에 따라 다양한 정보를 포함하도록 구성된다. 일례에서, 엔트로피 인코더(725)는 일반 제어 데이터, 선택된 예측 정보(예: 인트라 예측 정보 또는 인터 예측 정보), 잔차 정보 및 기타 적절한 정보를 비트스트림에 포함하도록 구성된다. 개시된 주제에 따르면, 인터 모드 또는 양-예측 모드의 병합 서브모드에서 블록을 코딩하는 경우, 잔차 정보가 없을 수 있음에 유의한다.

[0086] 도 8은 본 개시의 다른 실시예에 따른 비디오 디코더(810)의 도면을 도시한다. 비디오 디코더(810)는 코딩된 비디오 시퀀스의 일부인 코딩된 픽처를 수신하고, 코딩된 픽처를 디코딩하여 재구성된 픽처를 생성하도록 구성된다. 일례에서, 비디오 디코더(810)는 도 4의 예에서 비디오 디코더(410) 대신에 사용될 수 있다.

[0087] 도 8의 예에서, 비디오 디코더(810)는 도 8의 도시된 바와 같이 함께 결합된 엔트로피 디코더(871), 인터 디코더(880), 잔차 디코더(873), 재구성 모듈(874) 및 인트라 디코더(872)를 포함한다.

[0088] 엔트로피 디코더(871)는 코딩된 픽처로부터, 코딩된 픽처를 구성하는 신택스 요소를 나타내는 특정 심볼을 재구성하도록 구성될 수 있다. 이러한 심볼은, 예를 들어 블록이 코딩되는 모드(예: 인트라 모드, 인터 모드, 양-예측 모드, 두 후자는 병합 서브모드(merge submode) 또는 다른 서브모드), 인트라 디코더(872) 또는 인터 디코더(880) 각각에 의해 예측에 사용되는 특정 샘플 또는 메타 데이터를 식별할 수 있는 예측 정보(예: 인트라 예측 정보 또는 인터 예측 정보), 예를 들어 양자화된 변환 계수의 형태로 된 잔차 정보 등을 포함할 수 있다. 일례에서, 예측 모드가 인터 모드 또는 양-예측 모드인 경우, 인터 예측 정보는 인터 디코더(880)에 제공되고; 예측 유형이 인트라 예측 유형인 경우, 인트라 예측 정보는 인트라 디코더(872)에 제공된다. 잔차 정보는 역양자화될 수 있으며 잔차 디코더(873)에 제공된다.

[0089] 인터 디코더(880)는 인터 예측 정보를 수신하고 인터 예측 정보에 기초하여 인터 예측 결과를 생성하도록 구성된다.

[0090] 인트라 디코더(872)는 인트라 예측 정보를 수신하고 인트라 예측 정보에 기초하여 예측 결과를 생성하도록 구성된다.

[0091] 잔차 디코더(873)는 역양자화를 수행하여 역양자화된 변환 계수를 추출하고 역양자화된 변환 계수를 처리하여 잔차를 주파수 도메인에서 공간 도메인으로 변환하도록 구성된다. 잔차 디코더(873)는 또한 (양자화기 파라미터(Quantizer Parameter, QP)를 포함하기 위해) 특정 제어 정보를 필요로 할 수 있고, 그 정보는 엔트로피 디코더(871)에 의해 제공될 수 있다(데이터 경로는 소량의 제어 정보뿐일 수 있으므로 표시되지 않음).

[0092] 재구성 모듈(874)은 공간 도메인에서, 잔차 디코더(873)에 의해 출력되는 잔차와 예측 결과(경우에 따라 인터 예측 모듈 또는 인트라 예측 모듈에 의해 출력됨)를 결합하여 재구성된 블록을 형성하도록 구성되고, 재구성된 블록은 재구성된 픽처의 일부일 수 있으며, 재구성된 픽처는 결과적으로 재구성된 비디오의 일부일 수 있다. 디

블로킹(deblocking) 동작 등과 같은, 다른 적절한 동작이 시각적 품질을 향상시키기 위해 수행될 수도 있다는 점에 유의한다.

- [0093] 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 임의의 적절한 기술을 사용하여 구현될 수 있음에 유의한다. 일 실시예에서, 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 하나 이상의 집적 회로를 사용하여 구현될 수 있다. 다른 실시예에서, 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 소프트웨어 명령어를 실행하는 하나 이상의 프로세서를 사용하여 구현될 수 있다.
- [0094] 본 개시의 측면들은 코딩된 비디오 스트림에서 제약 플래그를 갖는 코딩 도구(들) 및 기능의 제어 기술을 제공한다.
- [0095] 본 개시의 일 측면에 따르면, 비트스트림에서의 픽처 크기는 동일하게 유지되거나 변경될 수 있다. 일부 관련된 예에서, 비디오 인코더 및 디코더는 코딩된 비디오 시퀀스(coded video sequence, CVS), 그룹 픽처(GOP) 또는 유사한 다중 픽처 시간 프레임(multi-picture timeframe)에 대해 정의되고 일정하게 유지되는 주어진 픽처 크기에 대해 동작할 수 있다. MPEG-2와 같은 예에서, 시스템 설계는 장면의 활동도(activity)와 같은 요인에 따라 수평 해상도(따라서 화면 크기)를 변경하는 것으로 알려져 있지만, I 픽처에만이고, 따라서 픽처 크기가 정의되고 일반적으로 GOP에 대해 일정하게 유지된다. CVS 내에서 다른 해상도를 사용하기 위한 참조 픽처의 리샘플링은 예를 들어 ITU-T Rec. H.263 Annex P에서 알려져 있다. 그러나, CVS에서의 픽처 크기는 변경되지 않고 참조 픽처만 리샘플링되므로, 그 결과 잠재적으로 픽처 캔버스의 일부만 사용되거나(예: 다운 샘플링의 경우), 장면의 일부만 캡처된다(예: 업샘플링의 경우). H.263 Annex Q와 같은 일부 예에서, 각 차원(예: 상향 또는 하향)에서 2배로 개별 매크로블록의 리샘플링이 허용된다. 그러나 픽처 크기는 동일하게 유지된다. 예를 들어 H.263에서 매크로블록의 크기는 고정될 수 있고, 따라서 매크로블록의 크기를 시그널링할 필요가 없다.
- [0096] 일부 관련된 예에서, 예측 픽처의 픽처 크기는 변경될 수 있다. VP9와 같은 일례에서, 참조 픽처 리샘플링 및 전체 픽처에 대한 해상도 변경이 허용된다. 일부 예에서(예를 들어, Hendry, et. al, "Onadaptive resolution change (ARC) for VVC", Joint Video Team document JVET-M0135-v1, Jan 9-19, 2019, 그 전체 내용이 본 명세서에 통합됨), 전체 참조 픽처의 다른 해상도(예: 더 높은 해상도 또는 더 낮은 해상도)로의 리샘플링이 허용된다. 서로 다른 후보 해상도는 시퀀스 파라미터 세트(Set)에서 코딩될 수 있으며 픽처 파라미터 세트(picture parameter set, PPS)에서 픽처별 선택 요소로 참조될 수 있다.
- [0097] 본 발명의 일 측면에 따르면, 소스 비디오는 상이한 해상도와 같은 상이한 품질을 갖는 하나 이상의 계층을 포함하는 비트스트림으로 픽처를 인코딩할 수 있는 층이 있는(layered) 코딩에 의해 압축될 수 있다. 비트스트림은 디코더 측에서 출력할 수 있는 계층(또는 계층 세트)을 지정하는 선택 요소를 가질 수 있다. 출력될 계층 세트는 출력 계층 세트로 정의될 수 있다. 예를 들어, 다중 계층 및 확장성을 지원하는 비디오 코덱에서, 하나 이상의 출력 계층 세트가 비디오 파라미터 세트(video parameter set, VPS)에서 시그널링될 수 있다. 전체 비트스트림 또는 하나 이상의 출력 계층 세트에 대한 프로파일 티어층 레벨(profile tier level, PTL)을 지정하는 선택 요소는 VPS, 일부 예에서 디코더 능력 정보(decoder capability information, DCI)라고 할 수 있는 디코더 파라미터 세트(decoder parameter set, DPS), SPS, PPS, SEI 메시지 등에서 시그널링될 수 있다. PTL 정보에는 코딩 도구나 기능에 대한 제약을 지정할 수 있는 일반 제약 정보가 존재할 수 있다. 다양한 코딩 도구 및 기능에 대한 제약 정보를 효율적으로 표현하고 시그널링하는 것이 바람직하다.
- [0098] 일부 예에서, 용어 "서브픽처(sub-picture)"는, 예를 들어 샘플, 블록, 매크로블록, 코딩 유닛, 또는 의미론적으로 그룹화되고 변경된 해상도로 독립적으로 코딩될 수 있는 유사한 엔티티의 직사각형 배열을 지칭하기 위해 사용될 수 있다. 하나 이상의 서브픽처는 픽처를 형성할 수 있다. 하나 이상의 코딩된 서브픽처는 코딩된 픽처를 형성할 수 있다. 하나 이상의 서브픽처를 하나의 픽처로 조립될 수 있으며, 픽처로부터 하나 이상의 서브픽처가 추출될 수 있다. 일부 예에서, 하나 이상의 코딩된 서브픽처는 샘플 레벨에서 코딩된 픽처로 트랜스코딩하지 않고 압축된 도메인에서 조립될 수 있다. 일부 예에서, 하나 이상의 코딩된 서브픽처는 압축된 도메인에서 코딩된 픽처로부터 추출될 수 있다.
- [0099] 일부 예에서, 예를 들어 참조 픽처를 리샘플링함으로써 CVS에서 픽처 또는 서브픽처의 해상도 변경을 허용하는 메커니즘은 적응적 해상도 변경(ARC)으로 지칭될 수 있다. 적응적 해상도 변경을 수행하기 위해 사용되는 제어 정보는 ARC 파라미터로 지칭될 수 있다. ARC 파라미터는 필터 파라미터, 스케일링 인자, 출력 및/또는 참조 픽처의 해상도, 다양한 제어 플래그 등을 포함할 수 있다.
- [0100] 일부 예에서, ARC의 인코딩/디코딩은 픽처 단위로 이루어지므로, 제어 정보(ARC 파라미터) 세트는 단일하고 의

미론적으로 독립적인 코딩된 비디오 픽처를 인코딩/디코딩하는 데 사용된다. 일부 예에서, ARC의 인코딩/디코딩은 서브픽처 단위로 이루어지므로, 픽처 내의 다수의 서브픽처는 독립적인 ARC 파라미터로 인코딩/디코딩될 수 있다. ARC 파라미터는 다양한 기술을 사용하여 시그널링될 수 있음에 유의한다.

- [0101] 도 9는 본 개시의 일부 실시예에 따른 ARC 파라미터를 시그널링하기 위한 기술의 예(예: 옵션)를 도시한다. 코딩 효율, 복잡도 및 아키텍처는 예마다 다를 수 있다. 비디오 코딩 표준 또는 기술은 ARC 파라미터를 시그널링하기 위해 하나 이상의 예 또는 다른 변형을 선택할 수 있다. 예들은 상호 배타적이지 않을 수 있고, 애플리케이션 요구, 표준 기술, 인코더의 선택 등에 기초하여 교환될 수 있다.
- [0102] 본 개시의 일 측면에 따르면, ARC 파라미터는 다양한 방식으로 ARC 파라미터의 클래스로 제공될 수 있다. 일부 예에서, ARC 파라미터의 클래스는 X 차원 및 Y 차원에서 분리되거나 결합된 업샘플 및/또는 다운샘플 인자를 포함한다. 일례에서, 업샘플 및/또는 다운샘플 인자를 포함하는 표를 가리킬 수 있는 하나 이상의 짧은 신택스 요소가 코딩될 수 있다.
- [0103] 일부 예에서, ARC 파라미터의 클래스는 주어진 수의 픽처에 대한 일정한 속도 줌인 및/또는 줌아웃을 지시하는 시간 차원의 추가와 함께, 업샘플 인자 및/또는 다운샘플 인자를 포함한다. 일례에서, 시간 차원이 추가된 업샘플 인자 및/또는 다운샘플 인자를 포함하는 표를 가리킬 수 있는 하나 이상의 짧은 신택스 요소가 코딩될 수 있다.
- [0104] 일부 예에서, ARC 파라미터의 클래스는 X 차원 또는 Y 차원으로, 샘플, 블록, 매크로블록, CU 또는 임의의 다른 적절한 그레놀래리티(granularity)의 단위로, 조합되거나 개별적으로 입력 픽처, 출력 픽처, 참조 픽처, 코딩된 픽처의 해상도를 포함한다. 일부 예에서, 비디오 코딩에서 사용되는 하나 이상의 해상도(예: 입력 픽처에 대한 하나의 해상도, 참조 픽처에 대한 다른 해상도)가 있고, (해상도 중 하나에 대응하는) 값들의 세트가 (해상도 중 다른 것에 대응하는) 값들의 다른 세트로부터 추론될 수 있다. 값의 결정은 예를 들어 플래그의 사용에 기초하여 게이팅될 수 있다. 게이팅을 위한 플래그의 사용은 추가 설명에서 자세히 설명될 것이다.
- [0105] 일부 예에서, ARC 파라미터의 클래스는 H.263 Annex P에서 사용된 것과 유사한 워핑 좌표(warping coordinate)를 전술한 바와 같이 적절한 그레놀래리티로 포함한다. H.263 Annex P는 워핑 좌표를 코딩하는 효율적인 방법을 정의한다. 다른 효율적인 방법이 고안될 수 있다. 예를 들어, Annex P의 워핑 좌표의 가변 길이 가역(variable length reversible), 허프만 스타일 코딩(Huffman-style coding)은 적절한 길이의 이진 코딩으로 대체될 수 있으며, 여기서 이진 코드 워드의 길이는 최대 픽처 크기의 경계를 벗어나는 워핑을 허용하는 값으로 계수 및 오프셋을 곱한 최대 화상 크기로부터 도출될 수 있다.
- [0106] 일부 예에서, ARC 파라미터의 클래스는 업샘플 필터 파라미터 및/또는 다운샘플 필터 파라미터를 포함한다. 일례에서는 업샘플링 및/또는 다운샘플링을 위한 단일 필터만 있다. 다른 예에서는 다수의 필터를 사용할 수 있다. 일부 예에서, 필터 파라미터는 필터 설계에서 더 많은 유연성을 허용하기 위해 시그널링될 수 있다. 가능한 필터 설계 목록의 인덱스를 사용하여 필터 파라미터를 선택할 수 있다. 필터는 완전히 지정될 수 있고(예: 적절한 엔트로피 코딩 기술을 사용하여 필터 계수 목록을 지정함으로써), 필터는 위에서 설명한 임의의 메커니즘에 따라 시그널링되는 업샘플 또는 다운샘플 비율을 통해 묵시적으로 선택될 수 있는 등이다.
- [0107] 이하의 설명에서, 업샘플 또는 다운샘플 인자(X 차원 및 Y 차원 모두에 사용되는 동일한 인자)의 유한 세트는 코드워드를 통한 시그널링 ARC 파라미터를 설명하는 데 사용된다. 일부 예에서, 코드워드는 예를 들어 비디오 코딩 사양(예: H.264 및 H.265)에서의 특정 신택스 요소에 대한 Ext-Golomb 코드를 사용하여 가변 길이 코딩될 수 있다.
- [0108] 도 10은 업샘플 또는 다운샘플 인자, 코드워드 및 Ext-Golomb 코드의 매핑을 위한 표(1000)의 예를 도시한다.
- [0109] 비디오 압축 기술 또는 표준에서 이용 가능한 업스케일 및 다운스케일 메커니즘의 애플리케이션 및 능력에 따라 다른 유사한 매핑이 고안될 수 있음에 유의한다. 일부 예에서 표 1은 추가 값으로 적절하게 확장될 수 있다. Ext-Golomb 코드 이외의 엔트로피 코딩 메커니즘, 예를 들어 이진 코딩을 사용하여 값을 나타낼 수 있음에 유의한다. 일례에서, Ext-Golomb 코드 이외의 엔트로피 코딩 메커니즘은, 예를 들어 미디어 인식 네트워크 요소(media-aware network element, MANE)에 의해 리샘플링 인자가 비디오 처리 엔진(예: 인코더 및 디코더) 외부에서 중요한 경우에 특정 이점을 가질 수 있다. 일부 예에서 해상도 변경이 필요하지 않은 경우(예: 원래/타겟 해상도가 표 1에서 1임), 짧은 Ext-Golomb 코드(예: 표 1에 표시된 단일 비트만)가 선택될 수 있으며, 예를 들어 가장 일반적인 경우에 이진 코드를 사용하는 것보다 유리한 코딩 효율 이점이 있을 수 있다.
- [0110] 본 개시의 일 측면에 따르면, 표 1과 같은 매핑표를 구성할 수 있다. 예를 들어, 표 1의 엔트리 수와 대응하는

시맨틱은 완전히 또는 부분적으로 구성할 수 있다. 일부 예에서, 매핑 표의 기본 개요는 SPS 또는 DPS와 같은 고 레벨의 파라미터 세트로 전달된다. 대안적으로 또는 추가적으로, 일부 예에서, 표 1과 유사한 하나 이상의 표이 비디오 코딩 기술 또는 표준에서 정의될 수 있고, 예를 들어 SPS 또는 DPS를 통해 표 중 하나가 선택될 수 있다.

- [0111] 전술한 바와 같이 코딩된 업샘플 인자 또는 다운샘플 인자와 같은 ARC 정보는 비디오 코딩 기술 또는 표준 선택스에 포함될 수 있다. 업샘플 필터 또는 다운샘플 필터와 같은 다른 클래스의 ARC 정보를 제어하기 위해 하나 이상의 코드워드가 사용될 수 있음에 유의한다. 일부 예에서는 필터 또는 기타 데이터 구조에 비교적 많은 양의 데이터가 필요하다.
- [0112] 도 9를 참조하면, H.263 Annex P와 같은 예(910)에서 ARC 정보(912)는 4개의 워핑 좌표 형태일 수 있으며, H.263 PLUSPTYPE(913) 헤더 확장에서와 같은 픽처 헤더(911)에 포함된다. 예(910)는 i) 픽처 헤더를 사용할 수 있고, ii) ARC 정보의 빈번한 변경이 예상되는 경우에 적용될 수 있다. 그러나, 예(910)에 도시된 바와 같은 H.263-style 시그널링을 사용하는 경우의 오버헤드는 높을 수 있고, 픽처 헤더가 일시적인 성질(transient nature)일 수 있기 때문에 픽처 경계 사이에서 스케일링 인자가 적용되지 않을 수 있다.
- [0113] 도 9를 참조하면, JVCET-M135-v1과 같은 예(920)에서, ARC 참조 정보(925)(예: 색인)는 PPS(924)에 배치될 수 있고 타깃 해상도(예: 해상도 1-3)를 포함하는 표(또는 타깃 해상도 표)(926)를 가리킬 수 있다. 예를 들어, 표(926)는 SPS(927) 내부에 있다. SPS(927) 내의 표(926)에 타깃 해상도를 배치하는 것은 기능 교환 중에 SPS를 상호 운용성 네고시에이션 지점(interoperability negotiation point)으로 사용하여 정당화될 수 있다. 적절한 PPS(924)에서 참조(예: ARC 참조 정보(925))에 의해 하나의 픽처에서 다른 픽처로 표(926)에 있는 값의 제한된 세트(예: 해상도 1-3) 내에서 해상도가 변경될 수 있다.
- [0114] 도 9는 또한 예(930), 예(940) 및 예(950)와 같은, 비디오 비트스트림에서 ARC 정보를 전달하는 데 사용될 수 있는 추가 기술을 도시한다. 이들 기술은 동일한 비디오 코딩 기술 또는 표준에서 개별적으로 사용될 수 있거나 적절한 조합으로 사용될 수 있다.
- [0115] 도 9를 참조하면, 예(930)에서, 리샘플링 인자(또는 줌 인자)와 같은 ARC 정보(939)는 슬라이스 헤더, GOB 헤더, 타일 헤더, 타일 그룹 헤더 등의 헤더에 존재할 수 있다. 예를 들어 도 9에는 타일 그룹 헤더(938)가 표시되어 있다. 예(930)로 나타난 기술은 ARC 정보(939)가 단일 가변 길이 ue(v) 또는 몇 비트의 고정 길이 코드워드와 같은 적은 수의 비트로 코딩될 수 있는 경우에 사용될 수 있다.
- [0116] 본 개시의 일 측면에 따르면, 헤더(예: 도 9의 타일 그룹 헤더(938), 슬라이스 헤더 또는 타일 헤더)에 ARC 정보(939)를 직접 갖는 것은 ARC 정보(939)가 전체 픽처가 아닌, 예를 들어 대응하는 타일 그룹(또는 슬라이스, 타일)이 나타내는 서브픽처에 적용될 수 있다는 점에서 추가적인 이점을 가질 수 있다. 또한, 일례에서, 비디오 압축 기술 또는 표준이 전체 픽처 적응적 해상도 변경만을 상정하더라도(예를 들어, 타일 그룹 기반 적응적 해상도 변경과 대조적으로), 예(930)는 오류 회복력 관점에서 예(910)보다 특정한 이점을 가질 수 있다.
- [0117] 도 9를 참조하면, 예(940)에서, ARC 정보(942)는 PPS, 헤더 파라미터 세트, 타일 파라미터 세트, 적응 파라미터 세트(APS) 또는 적응 파라미터 세트(adaptation parameter set, APS) 등과 같은, 파라미터 세트(941)에 존재할 수 있다. APS(941)는 예를 들어 도 9에 도시되어 있다. 일부 예에서, 파라미터 세트(941)의 범위는 픽처보다 크지 않을 수 있으며, 예를 들어 픽처, 타일 그룹 등일 수 있다. ARC 정보(예: ARC 정보(942))의 사용은 관련 파라미터 세트(예: APS(941))의 활성화를 통해 암묵적일 수 있다. 예를 들어, 비디오 코딩 기술 또는 표준이 픽처 기반 ARC만을 고려하는 경우, PPS 또는 등가물이 적절할 수 있다.
- [0118] 도 9를 참조하면, 예(950)에서, ARC 참조 정보(953)는 타일 그룹 헤더(954) 또는 전술한 바와 같이 유사한 데이터 구조(예: 픽처 헤더, 슬라이스 헤더, 타일 헤더 또는 GOP 헤더)에 존재할 수 있다. 타일 그룹 헤더(954)는 도 9에 예로서 나타나 있다. ARC 참조 정보(953)는 예를 들어 SPS, DPS 등과 같이 단일 픽처 이상의 범위를 갖는 파라미터 세트(956)에서 이용 가능한 ARC 정보(955)의 서브세트를 참조할 수 있다. SPS(956)는 도 9에 예로서 나타나 있다.
- [0119] 도 11은 본 개시의 일부 실시예에 따른 ARC 파라미터 시그널링의 일부 예를 도시한다. 도 11은 비디오 코딩 표준에서 사용되는 선택스 다이어그램 예를 보여준다. 일례에서, 선택스 다이어그램의 표기법은 대략 C 스타일 프로그래밍을 따른다. 굵은 글씨체(boldface)의 행은 비트스트림에 존재하는 선택스 요소를 지시할 수 있고, 굵은 글씨체가 없는 행은 제어 흐름(들) 또는 변수의 설정을 지시할 수 있다.
- [0120] 도 11을 참조하면, 타일 그룹 헤더(1101)는 픽처의 일부(예: 직사각형 부분)에 적용 가능한 헤더의 선택스 구조

를 포함한다. 일례에서, 타일 그룹 헤더(1101)는 가변 길이, Exp-Golomb 코딩된 선택스 요소 dec_pic_size_idx(1102)(굵게 표시됨)를 조건부로 포함할 수 있다. 타일 그룹 헤더(1101) 내의 선택스 요소(예: dec_pic_size_idx(1102))의 존재는 예를 들어 플래그(예: adaptive_pic_resolution_change_flag)(1103)로 표현되는 적응적 해상도에 기초하여 게이팅될 수 있다. 플래그(예: adaptive_pic_resolution_change_flag)(1103)의 값은 굵은 글씨체로 표시되지 않았으므로, 선택스 다이어그램에서 플래그가 발생하는 지점의 비트스트림에 플래그가 존재한다. 적응적 해상도가 픽처 또는 픽처의 일부에 대해 사용되는지 여부는 비트스트림 내부 또는 외부에서 고 레벨의 선택스 구조(예: 도 11의 SPS(1110))에서 시그널링될 수 있다.

[0121] 도 11을 참조하면, SPS(1110)의 일부가 도시되어 있다. SPS(1110)는 플래그(1111)(예: adaptive_pic_resolution_change_flag)인 제1 선택스 요소(1111)를 포함한다. 플래그(1111)가 참일 때, 플래그(1111)는 특정 제어 정보를 요구할 수 있는 적응적 해상도의 사용을 지시할 수 있다. 일례에서, 특정 제어 정보는 SPS(1110) 및 타일 그룹 헤더(1101)에서의 if() 문(statement)(1112)에 의해 표시된 바와 같이 플래그(1111)의 값에 기초하여 조건부로 존재한다.

[0122] 도 11의 예에 도시된 바와 같이, 적응적 해상도가 사용 중인 경우, 샘플 단위의 출력 해상도(또는 출력 픽처의 해상도)(1113)가 코딩될 수 있다. 일례에서, 출력 해상도(1113)는 너비 해상도(예: output_pic_width_in_luma_samples) 및 높이 해상도(예: output_pic_height_in_luma_samples)에 기초하여 코딩된다. 비디오 코딩 기술 또는 표준에서, 출력 해상도(1113)의 값에 대한 특정 제한이 정의될 수 있다. 예를 들어, 레벨 정의는 총 출력 샘플의 수(예: output_pic_width_in_luma_samples와 output_pic_height_in_luma_samples의 곱)를 제한할 수 있다. 일부 예에서, 비디오 코딩 기술 또는 표준, 또는 외부 기술 또는 표준(예: 시스템 표준)은 너비 해상도 및/또는 높이 해상도(예: 너비 해상도 및/또는 높이 해상도는 2의 거듭제곱으로 나눌 수 있음), 높이 해상도에 대한 너비 해상도의 종횡비(예: 높이 해상도에 대한 너비 해상도의 비율은 4:3 또는 16:9임) 등에 대한 변조지정 범위를 제한할 수 있다. 일례에서, 위의 제한사항은 하드웨어 구현을 용이하게 하기 위해 도입될 수 있다.

[0123] 특정 애플리케이션에서, 인코더는 크기가 출력 픽처 크기라고 암묵적으로 가정하기보다는 특정 참조 픽처 크기를 사용하도록 디코더에 명령할 수 있다. 예를 들어, 선택스 요소(예: reference_pic_size_present_flag)(1114)는 참조 픽처 치수(1115)의 조건부 존재를 게이팅한다. 참조 픽처 치수(1115)는 일례에서 너비(예: reference_pic_width_in_luma_samples)와 높이(예: reference_pic_height_in_luma_samples)를 모두 포함할 수 있다.

[0124] 또한 도 11에는 적용 가능한 디코딩 픽처 너비 및 높이의 표가 도시되어 있다. 일례에서, 표의 엔트리의 수는 표 지시(예: 선택스 요소 num_dec_pic_size_in_luma_samples_minus1)(1116)에 의해 표현될 수 있다. "minus1"은 선택스 요소(1116)의 값의 해석을 의미할 수 있다. 예를 들어, 코딩된 값이 0이면, 하나의 표 엔트리가 존재한다. 코딩된 값이 5이면, 6개의 표 엔트리가 존재한다. 표의 엔트리 각각에 대해, 디코딩된 픽처 너비 및 높이가 선택스 요소(1117)로서 포함된다.

[0125] 선택스 요소(1117)에 의해 표현된 표 엔트리는 타일 그룹 헤더(1101)에서의 선택스 요소 dec_pic_size_idx(1102)를 사용하여 색인화될 수 있고, 따라서 타일 그룹마다 상이한 디코딩된 크기 및 줌 인자를 허용한다.

[0126] 본 개시의 일 측면에 따르면, 특정 비디오 코딩 기술 또는 표준(예: VP9)은 시간적 확장성과 함께 특정 형태의 참조 픽처 리샘플링을 구현함으로써 공간적 확장성을 가능하게 할 수 있다. 일 실시예에서, 참조 픽처는 ARC 스타일 기술을 사용하여 공간 강화 계층(spatial enhancement layer)의 기반을 형성하기 위해 더 높은 해상도로 업샘플링된다. 업샘플링된 픽처는 예를 들어 세부 사항을 추가하기 위해 고해상도에서 일반 예측 메커니즘(예: 참조 픽처로부터의 인터 예측을 위한 움직임 보상된 예측)을 사용하여 개선될 수 있다.

[0127] 일부 예에서, 네트워크 추상화 계층(network abstraction layer, NAL) 유닛 헤더, 예를 들어 시간 ID 필드의 값은 시간 계층 정보 및 또한 공간 계층 정보를 디시하는 데 사용된다. NAL 유닛 헤더의 값을 사용하여 시간 계층 정보와 공간 계층 정보를 모두 지시하면 수정 없이 확장 가능한 환경에 대해 기존 선택된 포워딩 유닛들(selected forwarding units, SFUs)을 사용할 수 있다. 예를 들어, 기존 SFUs는 NAL 유닛 헤더 시간 ID 값에 기초하여, 선택된 시간 계층 포워딩을 위해 생성 및 최적화될 수 있다. 그러면 기존 SFUs는 일부 예에서 수정 없이 공간 확장성(예: 공간 계층의 선택)에 사용될 수 있다. 일부 예에서, 코딩된 픽처 크기와 NAL 유닛 헤더 내의 시간 ID 필드에 의해 지시되는 시간 계층 사이의 매핑이 제공될 수 있다.

- [0128] 본 개시의 일 측면에 따르면, 코딩된 비트스트림의 일부 특징은 프로파일, 티어, 레벨 및 일반 제약 정보를 포함하는 프로파일(profile), 티어(tier), 및 레벨 조합(profile, tier and level combination, PTL) 정보를 사용하여 지정될 수 있다. 일부 예에서, 프로파일은 색 재현, 해상도, 추가 비디오 압축 등과 같은 비트스트림 특징의 서브세트를 정의한다. 비디오 코덱은 베이스라인 프로파일(baseline profile)(예: 압축률이 낮은 단순 프로파일), 고 프로파일(압축률이 높은 복합 프로파일), 메인 프로파일(Main profil)(압축률이 베이스라인 프로파일과 고 프로파일 사이인 프로파일, 디폴트 프로파일 설정될 수 있음) 등과 같은 다양한 프로파일을 정의할 수 있다.
- [0129] 또한, 티어 및 레벨은 최대 비트 레이트, 최대 루마 샘플 레이트, 최대 루마 픽처 크기, 최소 압축률, 허용되는 최대 슬라이스 수, 허용되는 최대 타일 수, 등과 관련하여 비트스트림을 정의하는 특정 제약을 지정하는 데 사용될 수 있다. 하위 티어(lower tier)는 상위 티어(higher tier)보다 더 제약을 받으며, 하위 레벨은 상위 레벨보다 더 제약을 받는다. 일례에서, 표준은 메인(Main)과 하이(High)의 두 개의 티어를 정의할 수 있다. 메인 티어는 하이 티어보다 하위 티어이다. 티어들은 최대 비트 레이트의 면에서 다른 애플리케이션을 처리하도록 만들어진다. 메인 티어는 대부분의 애플리케이션용으로 설계되고, High 티어는 일례에서 매우 까다로운 애플리케이션용으로 설계된다. 표준은 다수의 레벨을 정의할 수 있다. 레벨은 비트스트림에 대한 제약의 세트이다. 일례에서, 레벨 4 미만의 레벨의 경우는 메인 티어만 허용된다. 일부 예에서, 주어진 티어/레벨에 준거하는 디코더는 해당 티어/레벨 및 모든 하위 티어/레벨에 대해 인코딩되는 모든 비트스트림을 디코딩할 수 있어야 한다.
- [0130] 일반 제약 정보는 비디오 소스 유형, 코딩 도구 및 기능에 대한 제약 정보를 포함할 수 있다. 예를 들어, 제약 플래그는 인터 코딩 도구, 인트라 코딩 도구, DBF, 엔트로피 코딩, 변환, 파티셔닝(예: 타일, 슬라이스), 버퍼 관리, 랜덤 액세스(예: IDR), 파라미터 세트(예: SPS, PPS) 등이 코딩된 비디오 비트스트림에 존재하거나 사용되는지를 지시할 수 있다. 제약 정보는 파라미터 세트(예: SPS, VPS 또는 DCI)에서 시그널링될 수 있다. 제약 플래그는 고 레벨 신택스 구조(예: SPS, VPS, DCI)에서 시그널링될 수 있다.
- [0131] 본 개시의 일부 측면에 따르면, PTL 정보는 범위(예: 비트스트림에 있는 코딩된 비디오 데이터의 일부)와 연관될 수 있다. 일부 예에서, PTL 정보는 예를 들어 전체 비트스트림, 비트스트림의 CVS, 비트스트림의 각각 출력 계층 세트(output layer set, OLS) 등에 대해 지정될 수 있고, VPS, DPS, DCI, SPS, PPS, APS, GOP, 시퀀스, 헤더, SEI 메시지 등과 같은, 고 레벨 신택스(high-level syntax, HLS) 구조에서 시그널링될 수 있다.
- [0132] 일부 예에서, 고 레벨 신택스(HLS)는 블록 레벨에 대해 정의된다. 블록 레벨 코딩 도구는 픽처를 재구축하기 위해 픽처 내의 픽셀 또는 샘플을 디코딩하는 데 사용될 수 있다. 블록 레벨 코딩 도구로는 인터 예측을 위한 코딩 도구(또는 인터 코딩 도구), 인트라 예측을 위한 코딩 도구(또는 인트라 코딩 도구), 적응적 루프 필터(adaptive loop filter, ALF), 디블로킹 필터(deblocking filter, DBF), 엔트로피 코딩, 변환 등과 같은, 코딩 블록의 재구축에 사용되는 임의 적절한 코딩 도구를 포함할 수 있다.
- [0133] 고 레벨 신택스(high-level syntax, HLS)는 기능, 시스템 인터페이스, 도구의 픽처 레벨 제어 및 버퍼 제어 등에 대한 정보를 지정할 수 있다. 예를 들어, HLS는 파티션(예: 타일, 슬라이스, 서브픽처), 버퍼 관리, 랜덤 액세스(예: IDR, 클린 랜덤 액세스(clean random access, CRA)), 파라미터 세트(들)(예: VPS, SPS, PPS, APS), 참조 픽처 리샘플링(reference picture resampling, RPR), 확장성 등을 지정할 수 있다. 고 레벨 신택스는 블록 레벨 위에 있을 수 있다.
- [0134] 제어 정보는 SPS 레벨 도구 제어 정보, PPS 레벨 도구 제어 정보, 시퀀스 레벨 제어 정보, 비트스트림 레벨 제어 정보 등과 같은 적절한 레벨을 가질 수 있다. 일부 예에서, PTL 정보는 제어 정보의 일부이고 HLS 구조로 제약 플래그로서 시그널링될 수 있고, HLS 구조에 대응하는 범위에서 도구의 제어 또는 제약을 지시할 수 있다. 예를 들어, PTL 정보에 대한 제약 플래그는 시퀀스 레벨 제어 정보 및 비트스트림 레벨 제어 정보 중 하나로 제공될 수 있다. 일례에서, HLS 구조에서 제약 플래그에 의해 특정 도구가 디스에블되면, HLS에 대응하는 범위 내의 코딩 블록에 대해, 그 도구는 사용되지 않는다.
- [0135] 도 12 및 도 13은 본 개시의 일부 실시예에 따른 PTL 정보의 일례를 도시한다. 도 12는 PTL 신택스 요소의 세트의 신택스 구조 예(1200)를 도시하고, 도 13은 일반 제약 정보의 신택스 구조 예(1300)를 도시한다.
- [0136] 도 12에서, PTL 신택스 요소의 세트는 general_profile_idc, general_tier_flag, general_level_idc, num_sub_profiles, general_sub_profile_idc, sublayer_level_present_flag, ptl_alignment_zero_bit 및 sublayer_level_idc를 포함할 수 있다.
- [0137] 도 13에서, 일반 제약 정보는 복수의 제약 플래그를 포함할 수 있다. 일례에서, 1과 동일한 제약 플래그(예:

intra_only_constraint_flag)(1305)는 파라미터 sh_slice_type이 I이어야 함(즉, 인트라 슬라이스인 슬라이스)를 지시할 수 있다. 파라미터 sh_slice_type은 유형 I, P 및 B의 사이에서 슬라이스의 코딩 유형을 지정하는 슬라이스 헤더의 파라미터이다.

[0138] 0과 동일한 제약 플래그(예: intra_only_constraint_flag)(1305)는 제약을 가하지 않는다(예: sh_slice_type은 다른 정보(예: profile_idc)가 비인트라 슬라이스를 허용할 수 있는 PTL 정보 범위 내의 모든 코딩된 그림에 대해 I이어야 한다. 다른 예에서, 1과 동일한 제약 플래그(예: no_alf_constraint_flag)(1306)는 sps_alf_enabled_flag가 PTL 정보의 범위 내의 모든 CVS에 대해 0과 동일함을 지시할 수 있으며, 따라서 적응적 루프 필터링은 예를 들어 profile_idc를 기반으로 허용되더라도 적응적 루프 필터링이 사용중이 아님을 지시할 수 있다. 0과 동일한 제약 플래그(예: no_alf_constraint_flag)(1306)는 상기 제약을 가하지 않는다.

[0139] 다른 예에서, 제약 플래그(예: no_lossless_coding_tool_constraint_flag)(1301)는 도 13에 도시된 바와 같이, 일반 제약 정보에서 시그널링될 수 있다. 1과 동일한 제약 플래그(예: no_lossless_coding_tool_constraint_flag)(1301)는 제약 플래그(1301)를 포함하는 PTL 정보의 범위 내에서 무손실 코딩과 관련된 코딩 도구가 사용될 수 없음을 지시한다. 0과 동일한 제약 플래그(예: no_lossless_coding_tool_constraint_flag)(1301)는 상기 제약을 가하지 않는다.

[0140] 다른 예에서, 제약 플래그(예: no_lossy_coding_tool_constraint_flag)(1302)는 도 13에 도시된 바와 같이 일반 제약 정보에서 시그널링될 수 있다. 1과 동일한 제약 플래그(예: no_lossy_coding_tool_constraint_flag)(1302)는 제약 플래그(1302)를 포함하는 PTL 정보의 범위 내에서 손실 코딩과 관련된 코딩 도구(들)가 사용될 수 없음을 지시한다. 0과 동일한 제약 플래그(예: no_lossy_coding_tool_constraint_flag)(1302)는 상기 제약을 가하지 않는다.

[0141] 일 실시예에서, 제약 플래그(예: no_lossless_coding_tool_constraint_flag)(1301)는 제약 플래그(예: no_lossy_coding_tool_constraint_flag)(1302)가 1과 동일한 경우에 1과 동일하지 않을 수 있다. 대안적으로 제약 플래그(예: no_lossy_coding_tool_constraint_flag)(1302)는 제약 플래그(예: no_lossless_coding_tool_constraint_flag)(1301)가 1과 동일한 경우에 1과 동일하지 않을 수 있다.

[0142] 일반 제약 정보에 있는 복수의 제약 플래그는 특정 순서로 정렬될 수 있다. 순서는 예를 들어 PTL의 범위에서 사용되지 않는 각각의 메커니즘 및/또는 도구의 가능성에 기초하여 설정될 수 있다. 순서는 우선 순위라고 할 수 있다. 순서는 일반 제약 정보 선택스 구조에서 높은 우선 순위에서 낮은 우선 순위로 표시될 수 있으며, 여기서 높은 우선 순위는 도구(또는 메커니즘)를 사용하지 않을 가능성이 높음을 지시하고 낮은 우선 순위는 도구(또는 메커니즘)를 사용하지 않을 가능성이 낮음을 지시한다. 순서에 영향을 미치는 추가 요인으로는 특정 사용 사례에만 사용될 가능성이 높은 도구(예: 서브픽처, 확장성 및/또는 인터페이스 지원용 도구), 인코더/디코더/구현 복잡도에 대한 도구의 영향, 등을 포함할 수 있다.

[0143] 도 14a~도 14b는 본 개시의 일부 실시예에 따른 PTL 선택스 구조(PTL 브래킷(bracket)이라고도 함)의 선택스 구조 예(1410) 및 일반 제약 정보 선택스 구조(일반 제약 정보 브래킷이라고도 함)에 대한 선택스 예(1420)를 포함하는 PTL 정보의 예를 도시한다. 일부 예에서, 제약 플래그의 수를 지시하는 선택스 요소(예: num_available_constraint_flags)가 시그널링될 수 있다. 일례에서, 제약 플래그의 수를 지시하는 선택스 요소는 일반 제약 정보 브래킷에 대한 선택스 예(1420)의 밖에 있을 수 있는 도 14a에 도시된 선택스 예(1410)에서 (1401)로 도시된 바와 같이, PTL 선택스 구조로 시그널링될 수 있다. 대안적으로, 제약 플래그의 수를 지시하는 선택스 요소는 선택스 예(1420)의 시작과 같은, 일반 제약 정보 브래킷의 시작 부분에서 시그널링될 수 있다. 선택스 요소(예: num_available_constraint_flags)가 존재하고 선택스 요소(예: num_available_constraint_flags)의 값이 N과 동일한 경우, 처음 N개의 제약 플래그는 일반 제약 정보 선택스 구조에 존재할 수 있다. 또한, 다른 제약 플래그는 존재하지 않을 수 있으며 특정 값과 동일하다고 추론할 수 있다. N은 음수가 아닌 정수일 수 있다.

[0144] 일 실시예에서, 값 N(예: num_available_constraint_flags)은 0 내지 제약 플래그의 최대 수(예: 파라미터 MaxNumConstraintFlags의 값)의 범위에 있다. 제약 플래그의 최대 수는 양의 정수일 수 있다. 제약 플래그(예: MaxNumConstraintFlags)의 최대 수의 값은 16, 32, 64, 128 등으로 미리 정의될 수 있다. 값 N(예: num_available_constraint_flags)이 0과 동일한 경우, 일반 제약 정보 선택스 구조에 제약 플래그가 존재하지 않는다. 값 N의 코딩(예: num_available_constraint_flags)은 값 N에 대한 대응하는 엔트로피 코딩 표현과 제약 플래그가 바이트 정렬을 보장하기 위해 8로 나눌 수 있는 수까지 더할 수 있도록 선택될 수 있다.

- [0145] 일부 예에서, 제약 플래그는 하나 이상의 제약 정보 그룹으로 분류될 수 있다. 각각의 제약 정보 그룹은 하나 이상의 제약 플래그를 포함할 수 있으며 대응하는 게이트 플래그를 가질 수 있다. 대응하는 제약 정보 그룹의 게이트 플래그는 대응하는 제약 정보 그룹의 제약 플래그(들)가 존재할 수 있는지를 지시할 수 있다. 일례에서, 게이트 플래그는 제약 그룹 존재 플래그로 지칭될 수 있다. 일반적으로 게이트 플래그는 대응하는 제약 정보 그룹과 연관되며, 대응하는 제약 정보 그룹의 제약 플래그(들)와 연관된다. 일 실시예에서, 게이트 플래그는 대응하는 제약 정보 그룹의 제약 플래그(들)가 제약 정보에 존재하는지(또는 시그널링되는지) 여부를 게이트한다. 예를 들어, 대응하는 제약 정보 그룹의 게이트 플래그가 1과 동일하면, 제약 정보 그룹에 대응하는 제약 플래그(들)는 예를 들어 일반 제약 정보에 존재할 수 있다. 대응하는 제약 정보 그룹의 게이트 플래그가 0과 동일하면, 제약 정보 그룹에 대응하는 제약 플래그(들)는 예를 들어 일반 제약 정보에 존재하지 않을 수 있다. 일례에서, 모든 게이트 플래그가 0과 동일하면, 제약 플래그는 존재하지 않는다.
- [0146] 제약 플래그들은 상이한 범위를 가질 수 있다. 예를 들어, DCI의 제약 플래그 범위는 코딩된 비디오 비트스트림일 수 있다. VPS의 제약 플래그 범위는 다수의 여러 계층이 있는 CLVS일 수 있다. SPS의 제약 플래그 범위는 단일 CLVS일 수 있다.
- [0147] 도 15a~도 15b는 본 개시의 일 실시예에 따른 일반 제약 정보 선택스 구조(1500)의 예를 도시한다. 일반 제약 정보 선택스 구조(1500)는 일반 제약 정보를 나타내는 플래그를 포함한다. 구체적으로, 일반 제약 정보 선택스 구조(1500)는 도 15a에서의 게이트 플래그(예: `general_frame_structure_constraint_group_flag`)(1501), 게이트 플래그(예: `high_level_functionality_constraint_group_flag`)(1502), 게이트 플래그(예: `scalability_constraint_group_flag`)(1503), 게이트 플래그(예: `partitioning_constraint_group_flag`)(1504), 게이트 플래그(예: `intra_coding_tool_constraint_group_flag`)(1505), 게이트 플래그(예: `inter_coding_tool_constraint_group_flag`)(1506), 게이트 플래그(예: `transform_constraint_group_flag`)(1507), 게이트 플래그(예: `inloop_filtering_constraint_group_flag`)(1508)과 같은, 하나 이상의 게이트 플래그를 포함한다. 도 15a에 도시된 바와 같이, 하나 이상의 게이트 플래그(예: 게이트 플래그(1501)-(1508))는 일반 제약 정보 선택스 구조(1500)의 시작 부분에 존재할 수 있다.
- [0148] 게이트 플래그(예: `general_frame_structure_constraint_group_flag`)(1501)는 제약 정보 그룹(1510)과 연관되고, 제약 정보 그룹(1510)에 있는 제약 플래그(1511)~(1514)와 연관된다. 1과 동일한 게이트 플래그(예: `general_frame_structure_constraint_group_flag`)(1501)는 제약 정보 그룹(1510)에 있는 제약 플래그(1511)~(1514)가 존재할 수 있음을 지정할 수 있다.
- [0149] 제약 정보 그룹(1510)(또는 제약 플래그(1511)~(1514))은 입력 소스 및 프레임 패키징(예: 패키징되거나 투영된 프레임)과 관련될 수 있다. 도 15a를 참조하면, 제약 플래그(1511)~(1514)는 `general_non_packed_constraint_flag`(1511), `general_frame_only_constraint_flag`(1512), `general_non_projected_constraint_flag`(1513) 및 `general_one_picture_only_constraint_flag`(1514)에 대응한다. 그렇지 않으면, 0과 동일한 게이트 플래그(예: `general_frame_structure_constraint_group_flag`)(1501)는 제약 정보 그룹(1510)에 있는 제약 플래그(1511)~(1514)가 일반 제약 정보 선택스 구조(1500)에 존재하지 않을 수 있음을 지정할 수 있다.
- [0150] 또한, 일부 예에서, 1과 동일한 게이트 플래그(예: `high_level_functionality_constraint_group_flag`)(1502)는 제약 정보 그룹(1520)에 있는 고 레벨 기능(예: 참조 픽처 리샘플링)에 관련된 제약 플래그가 도 15b에 도시된 바와 같이 존재할 수 있음을 지정할 수 있다. 그렇지 않으면, 0과 동일한 게이트 플래그(예: `high_level_functionality_constraint_group_flag`)(1502)는 제약 정보 그룹(1520)에 있는 제약 플래그가 일반 제약 정보 선택스 구조(1500)에 존재하지 않을 수 있음을 지정할 수 있다.
- [0151] 다시 도 15a를 참조하면, 1과 동일한 게이트 플래그(예: `scalability_constraint_group_flag`)(1503)는 확장성(예: 계층 간 예측)과 관련된 제약 플래그(들)가 존재할 수 있음을 지정할 수 있다. 그렇지 않으면 확장성과 관련된 제약 플래그(들)는 일반 제약 정보 선택스 구조(1500)에 존재하지 않을 수 있다.
- [0152] 1과 동일한 게이트 플래그(예: `partitioning_constraint_group_flag`)(1504)는 고 레벨 파티셔닝(예: 서브픽처 또는 타일)과 관련된 제약 플래그(들)가 존재할 수 있음을 지정할 수 있다. 그렇지 않으면, 고 레벨 파티셔닝과 관련된 제약 플래그가 일반 제약 정보 선택스 구조(1500)에 존재하지 않을 수 있다.
- [0153] 1과 동일한 게이트 플래그(예: `intra_coding_tool_constraint_group_flag`)(1505)는 인트라 코딩(예: 인트라 예측)과 관련된 제약 플래그(들)가 존재할 수 있음을 지정할 수 있다. 그렇지 않으면, 인트라 코딩과 관련된 제약

플래그(들)는 일반 제약 정보 선택스 구조(1500)에 존재하지 않을 수 있다.

- [0154] 1과 동일한 게이트 플래그(예: `inter_coding_tool_constraint_group_flag`)(1506)는 인터 코딩(예: 인터 픽처 예측을 위한 움직임 보상)에 관련된 제약 플래그(들)가 존재할 수 있음을 지정할 수 있다. 그렇지 않으면, 인터 코딩과 관련된 제약 플래그는 일반 제약 정보 선택스 구조(1500)에 존재하지 않을 수 있다.
- [0155] 1과 동일한 게이트 플래그(예: `transfom_contraint_group_flag`)(1507)는 변환 코딩(예: 다중 변환 행렬)과 관련된 제약 플래그(들)가 존재할 수 있음을 지정할 수 있다. 그렇지 않으면, 변환 코딩과 관련된 제약 플래그는 일반 제약 정보 선택스 구조(1500)에 존재하지 않을 수 있다.
- [0156] 일 실시예에서, 모든 게이트 플래그(예: 도 15a의 게이트 플래그(1501)~(1508))가 0과 동일한 경우, 일반 제약 정보 선택스 구조(예: 일반 제약 정보 선택스 구조(1500))에 제약 플래그가 존재하지 않을 수 있다.
- [0157] 본 개시의 측면에 따르면, 선택스는 게이트 플래그(예: 게이트 플래그(1501)~(1508)), 연관된 제약 플래그(예: 제약 플래그(1511)~(1512) 및 제약 정보 그룹(1520)의 제약 플래그), 추가 제어 정보 등을 포함하는 제어 정보가 바이트 정렬될 수 있도록, 예를 들어 다수의 플래그는 바이트 정렬을 유지하기 위해 8로 나눌 수 있도록 설계될 수 있다. 예를 들어, 제약 정보(예: 일반 제약 정보 선택스 구조(1500))의 게이트 플래그 및 제약 플래그의 수는 8로 나눌 수 있다. 제어 정보의 바이트 정렬을 달성하기 위해 바이트 정렬 메커니즘이 사용될 수 있다. 도 15b를 참조하면, 바이트 정렬을 위해 선택스(예: `while 루프`)(1530)가 사용될 수 있다.
- [0158] 일부 실시예에서, 제약 정보의 게이트 플래그(들)와 연관된 각각의 제약 정보 그룹(들)에서 제약 플래그(들)의 제시를 돕기 위해 오프셋(예: 선택스 요소 `constraint_info_offset[]`)를 사용함)과 같은 오프셋 정보 및 길이(예: 선택스 요소 `constraint_info_length[]`)를 사용함)와 같은 길이 정보가 제약 정보(예: 일반 제약 정보 선택스 구조의 시작 부분)에 존재한다. 일 실시예에서, 적어도 하나의 제약 정보 그룹 중 하나 이상이 코딩된 비디오 비트스트림에 존재한다. 제약 정보 그룹의 경우, 제약 정보 그룹에 대한 제약 정보에 오프셋 및 길이가 존재할 수 있다. 오프셋은 제약 정보 그룹에서 제1 제약 플래그에 대한 오프셋을 지시할 수 있고, 길이는 제약 정보 그룹에서의 제약 플래그의 수를 지시할 수 있다. 일부 예에서, 제약 정보 그룹의 수는 예를 들어 선택스 요소 `num_constraint_info_set`에 의해 명시적으로 지시될 수 있다. `num_constraint_info_set`의 값은 0보다 크거나 같은 정수일 수 있다. `num_constraint_info_set`의 값이 0인 경우, `constraint_info_offset[]`, `constraint_info_length[]` 및 제약 플래그는 일반 제약 정보 선택스 구조에 존재하지 않는다.
- [0159] 일 실시예에서, 제약 정보 오프셋(예: 선택스 요소 `constraint_info_offset[i]`) 및 제약 정보 길이(예: 선택스 요소 `constraint_info_length[i]`)는 제약 정보(예: 일반 제약 정보 선택스 구조)에서 제약 정보 그룹 i (i 는 양의 정수임)에 대한 제약 플래그 제시를 지원할 수 있다. 일례에서, 정보 오프셋의 값(예: 선택스 요소 `constraint_info_offset[i]`)이 5와 동일하고, 제약 정보 길이의 값(예: 선택스 요소 `constraint_info_length[i]`)이 3과 동일한 경우, 다섯 번째, 여섯 번째 및 일곱 번째 제약 플래그는 제약 정보 그룹 i 와 연관되고 제약 정보(예: 일반 제약 정보 선택스 구조)에 존재한다.
- [0160] 일례에서, 미리 결정된 순서(또는 주어진 순서)로 지정되는 제약 플래그를 코딩하기 위해 실행 길이 코딩(`run-length coding`)이 사용될 수 있다.
- [0161] 일 실시예에서, 제약 플래그가 미리 결정된 순서(또는 주어진 순서)로 지정되는 실행 코딩(`run coding`)이 사용될 수 있다. 제약 플래그를 직접 코딩하는 대신, "스킵(`skip`)" 값의 적절하게 코딩된 목록은 0과 동일한 제약 플래그를 지시할 수 있으며, 다음 제약 플래그가 1과 동일하다는 것을 암시한다. 전술한 실행 코딩은 (i) 제약 플래그의 수가 많고 (ii) 작은 백분율의 제약 플래그가 1인 경우에 특히 효율적일 수 있다.
- [0162] 일 실시예에서, 적어도 하나의 제약 정보 그룹 중 하나 이상이 코딩된 비디오 비트스트림에 존재한다. 적어도 하나의 제약 정보 그룹 중 하나 이상에 있는 복수의 제약 플래그는 미리 결정된 순서에 따라 시그널링된다. 따라서, 복수의 제약 플래그는 실행 코딩(예: 실행 인코딩 또는 실행 디코딩)될 수 있다. 또한, 코딩 블록의 서브 세트에 대한 예측 정보는 복수의 제약 플래그에 기초하여 결정될 수 있다.
- [0163] 일 실시예에서, 게이트 플래그의 제약 정보 그룹 내의 적어도 하나의 제약 플래그는 미리 결정된 순서에 따라 시그널링되는 복수의 제약 플래그를 포함한다. 따라서, 복수의 제약 플래그는 실행 코딩(예: 실행 인코딩 또는 실행 디코딩)될 수 있다.
- [0164] 일 실시예에서, 제약 플래그의 전체 목록은 비디오 코딩 표준(예: VVC 사양), 외부 표 등에 지정될 수 있다. 일례에서, 제약 플래그 중 이용 가능한 제약 플래그(들)만이 예를 들어 다음 중 하나 이상에 의해 지시된다: 다수

의 이용 가능한 제약 플래그(예: num_available_constraint_flags), 게이트 플래그(들)(또는 제약 그룹 존재 플래그(들)), 제약 정보 오프셋 정보 및 제약 정보 길이 정보 등이 코딩된 비디오 스트림에 존재한다.

[0165] 일례에서, 제약 플래그의 전체 목록이 지정되고 인코더 및 디코더에 사용 가능하다. 제약 플래그의 전체 목록은 디코더에 저장될 수 있다. 제약 플래그의 전체 목록은 100개의 제약 플래그를 포함할 수 있다. 100개의 제약 플래그 중 10개는 CLVS에 대한 제약 정보에 존재하므로 CLVS에서의 코딩 블록 서브세트에 사용할 수 있다. 100개의 제약 플래그 중 10개는 이용 가능한 10개의 제약 플래그로 지칭된다. 일례에서, 다수의 이용 가능한 제약 플래그(예: 10개)가 시그널링된다. 일례에서, 10개의 이용 가능한 제약 플래그는 2개의 제약 정보 그룹에 있고 제1 게이트 플래그 및 제2 게이트 플래그에 의해 게이팅된다. 따라서, 제1 게이트 플래그 및 제2 게이트 플래그는 10개의 이용 가능한 제약 플래그를 지시하기 위해 시그널링될 수 있다.

[0166] 일례에서, 제1 제약 정보 오프셋(예: 선택스 요소 constraint_info_offset[0]) 및 제1 제약 정보 길이(예: 선택스 요소 constraint_info_length[0])가 시그널링된다. 제2 제약 정보 오프셋(예: 선택스 요소 constraint_info_offset[1]) 및 제2 제약 정보 길이(예: 선택스 요소 constraint_info_length[1])가 시그널링된다. 예를 들어, 선택스 요소 constraint_info_offset[0]은 15이고 선택스 요소 constraint_info_length[0]은 3이고, 선택스 요소 constraint_info_offset[1]은 82이고 선택스 요소 constraint_info_length[1]은 7이므로, 전체 리스트(예: 100개의 제약 플래그)에서 15번째 내지 17번째 제약 플래그 및 82번째 내지 88번째 제약 플래그가 제약 정보에서 이용 가능하거나 존재함을 지시한다.

[0167] 일 실시예에서, 제약 플래그의 효율적인 코딩을 위한 다양한 기술(또는 방법, 실시예, 예) 중 임의의 것이 적절한 제어 정보를 채용하여 조합될 수 있다. 조합은 이러한 기술 중 둘 이상의 적절한 조합일 수 있다. 대안적으로, 다양한 기술(또는 방법, 실시예, 예) 중 하나는 독립적으로 사용될 수 있다. 제약 플래그는 그룹화될 수 있다. 특정 그룹(들)에서는 실행 코딩을 사용할 수 있는 반면 다른 그룹(들)은 간단한 이진 코딩을 사용할 수 있다.

[0168] 제약 플래그의 최대 수(예: MaxNumConstraintFlags)의 값은 16, 32, 64, 128 등으로 미리 정의될 수 있다.

[0169] 제약 플래그의 최대 수(예: MaxNumConstraintFlags)의 값은 general_profile_idc 또는 general_sub_profile_idc와 같은 프로파일 정보, 또는 코덱 버전 정보에 의해 결정될 수 있으므로, 제약 플래그(예: num_available_constraint_flags(1401))는 프로파일 정보 또는 버전 정보에 의해 제한될 수 있다. 예를 들어, 메인 프로파일(예: 여기서 MaxNumConstraintFlags = 64)의 제약 플래그의 수(예: num_available_constraint_flags(1401))의 값은 0에서 64 범위일 수 있는 반면 어드밴스드 프로파일(예: 여기서 MaxNumConstraintFlags = 128)의 제약 플래그의 수(예: num_available_constraint_flags(1401))의 값은 0에서 128 범위에 있을 수 있다.

[0170] 일 실시예에서, 제약 플래그의 수(예: num_available_constraint_flags)의 값은 general_profile_idc 또는 general_sub_profile_idc와 같은 프로파일 정보 또는 코덱 버전 정보에 의해 미리 정의된 값과 동일한 것으로 추론될 수 있으므로, num_available_constraint_flags의 값은 명시적으로 시그널링 없이 결정될 수 있다.

[0171] 일부 실시예에서, 예비 바이트 정보는 일반 제약 정보 선택스 구조에 존재할 수 있다. 예를 들어, 도 13에 도시된 바와 같이, 플래그 gci_num_reserved_bytes(1303) 및 gci_reserved_bytes[](1304)는 일반 제약 정보 선택스 구조의 확장을 위해 일반 제약 정보 선택스 구조에 존재할 수 있다. 플래그 gci_num_reserved_bytes는 예비된 제약 바이트의 수를 지정할 수 있다. 일례에서, 예비된 제약 바이트는 추가 플래그(예: 추가 제약 플래그)를 시그널링하기 위한 것이다. 플래그 gci_reserved_byte[]는 적절한 값을 가질 수 있다.

[0172] 일 실시예에서, gci_num_reserved_bytes의 값은 general_profile_idc 또는 general_sub_profile_idc와 같은 프로파일 정보 또는 코덱 버전 정보에 의해 제한되거나 결정될 수 있다. 기본 프로파일(또는 메인 프로파일)을 사용하면, 플래그 gci_num_reserved_bytes의 값은 0이 될 수 있다. 확장 프로파일(또는 어드밴스드 프로파일)을 사용하면, gci_num_reserved_bytes 값은 0보다 클 수 있다.

[0173] 일부 실시예에서, 필드 시퀀스 플래그는 코딩된 비디오 비트스트림에서 시그널링될 수 있다. 필드 시퀀스 플래그는 출력 계층에서의 픽처가 필드 코딩으로 코딩되었는지를 지시할 수 있다. 일부 예에서, 필드 시퀀스 플래그는 선택스 요소 sps_field_seq_flag를 사용하여 SPS에서 시그널링될 수 있다. 일 실시예에서, 플래그 sps_field_seq_flag는 SPS에 존재할 수 있다. 1과 동일한 sps_field_seq_flag 플래그는 CLVS가 필드를 나타내는 픽처를 전달하는 CLVS를 지시할 수 있다. 0과 동일한 sps_field_seq_flag 플래그는 CLVS가 프레임에 나타내는 픽처를 전달함을 지시할 수 있다.

- [0174] 도 13의 일반 제약 정보 선택스 구조에서, 플래그 `general_frame_only_constraint_flag`가 존재할 수 있다. 1과 동일한 플래그 `general_frame_only_constraint_flag`는 출력 계층 세트(예: `01sInScope`)에 대한 범위가 프레임 을 나타내는 픽처를 전달함을 지정할 수 있다. 0과 동일한 플래그 `general_frame_only_constraint_flag`는 출력 계층 세트(예: `01sInScope`)에 대한 범위가 프레임을 나타낼 수도 있고 나타내지 않을 수도 있는 픽처를 전달함을 지정한다. 일 실시예에서, 플래그 `general_frame_only_constraint_flag`는 출력 계층 세트 내의 픽처가 필드 코딩으로 코딩되는지를 지시한다. 출력 계층 세트는 코딩 블록의 서브셋을 포함할 수 있다. 플래그 `sps_field_seq_flag`는 픽처의 서브셋이 필드 코딩으로 코딩되지 않음을 지시하는 플래그 `general_frame_only_constraint_flag`(예: 1임)에 기초하여 거짓(false)일 수 있다. 픽처의 서브셋은 출력 계층 세트의 하나 계층에 있을 수 있다.
- [0175] 플래그 `general_frame_only_constraint_flag`가 1과 동일한 경우, 플래그 `sps_field_seq_flag`의 값은 0과 동일할 수 있다.
- [0176] 일 실시예에서, 플래그 `pps_mixed_nalu_types_in_pic_flag`는 PPS에 존재할 수 있다. 1과 동일한 플래그 `pps_mixed_nalu_types_in_pic_flag`는 PPS를 참조하는 각각의 픽처가 하나보다 많은 VCL NAL 유닛을 갖고 VCL NAL 유닛이 동일한 값의 `nal_unit_type`을 갖지 않음을 지정할 수 있다. 0과 동일한 플래그 `pps_mixed_nalu_types_in_pic_flag`는 PPS를 참조하는 각각의 픽처가 하나 이상의 VCL NAL 유닛을 갖고 PPS를 참조하는 각각의 픽처의 VCL NAL 유닛이 `nal_unit_type`의 동일한 값을 갖는다는 것을 지정할 수 있다. 도 13의 일반 제약 정보 선택스 구조에서, 플래그 `no_mixed_nalu_types_in_pic_constraint_flag`가 존재할 수 있다. 1과 동일한 플래그 `no_mixed_nalu_types_in_pic_constraint_flag`는 `pps_mixed_nalu_types_in_pic_flag`의 값이 0과 동일하도록 지정할 수 있다. 0과 동일한 플래그 `no_mixed_nalu_types_in_pic_constraint_flag`는 그러한 제약을 가하지 않는다.
- [0177] 일 실시예에서, 플래그 `general_one_picture_only_constraint_flag`는 도 13에 도시된 바와 같은 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 `general_one_picture_only_constraint_flag`는 비트스트림에 단 하나의 코딩된 픽처가 있음을 지정할 수 있다. 0과 동일한 플래그 `general_one_picture_only_constraint_flag`는 이러한 제약을 가하지 않는다.
- [0178] 일 실시예에서, 플래그 `single_layer_constraint_flag`는 도 13에 도시된 바와 같은 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 `single_layer_constraint_flag`는 `sps_video_parameter_set_id`가 0과 동일할 것임을 지정할 수 있다. 플래그 `single_layer_constraint_flag`는 0과 동일한 경우, 이러한 제약을 가하지 않는다. 플래그 `general_one_picture_only_constraint_flag`가 1과 동일한 경우, 플래그 `single_layer_constraint_flag`의 값은 1과 동일할 수 있다.
- [0179] 일 실시예에서, 플래그 `all_layers_independent_constraint_flag`는 도 13에 도시된 바와 같은 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 `all_layers_independent_constraint_flag`는 플래그 `vps_all_independent_layers_flag`가 1과 동일할 수 있음을 지정할 수 있다. 0과 동일한 플래그 `all_layers_independent_constraint_flag`는 그러한 제약을 가하지 않는다. 플래그 `single_layer_constraint_flag`가 1과 동일한 경우, 플래그 `all_layers_independent_constraint_flag`의 값은 1과 동일할 수 있다.
- [0180] 일 실시예에서, 플래그 `no_res_change_in_clvs_constraint_flag`는 도 13에 도시된 바와 같은 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 `no_res_change_in_clvs_constraint_flag`는 플래그 `sps_res_change_in_clvs_allowed_flag`가 0과 동일할 수 있음을 지정할 수 있다. 0과 동일한 플래그 `no_res_change_in_clvs_constraint_flag`는 그러한 제약을 가하지 않는다. 플래그 `no_ref_pic_resampling_constraint_flag`가 1일 때, 플래그 `no_res_change_in_clvs_constraint_flag`의 값은 1과 동일할 수 있다.
- [0181] 일 실시예에서, 플래그 `no_mixed_nalu_types_in_pic_constraint_flag`는 도 13의 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 `no_mixed_nalu_types_in_pic_constraint_flag`는 플래그 `pps_mixed_nalu_types_in_pic_flag`의 값이 0과 동일할 수 있음을 지정한다. 플래그는 `no_mixed_nalu_constraint_in_flag`이다. `to 0`은 그러한 제약을 가하지 않는다. 플래그 `one_subpic_per_pic_constraint_flag`가 1과 동일한 경우, 플래그 `no_mixed_nalu_types_in_pic_constraint_flag`의 값은 1과 동일할 수 있다.

- [0182] 일 실시예에서, 플래그 no_trail_constraint_flag는 도 13의 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 no_trail_constraint_flag는 OlsInScope에 존재하는 TRAIL_NUT와 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음을 지정할 수 있다(OlsInScope는 DPS를 참조하는 전체 비트스트림의 모든 계층을 포함하는 출력 계층 세트이다). 0과 동일한 플래그 no_trail_constraint_flag는 그러한 제약을 가하지 않는다. 플래그 general_one_picture_only_constraint_flag가 1과 동일한 경우, 플래그 no_trail_constraint_flag는 1과 동일할 수 있다.
- [0183] 일 실시예에서, 플래그 no_stsa_constraint_flag는 도 13의 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 no_stsa_constraint_flag는 OlsInScope에 존재하는 STSA_NUT와 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음을 지정할 수 있다. 0과 동일한 플래그 no_stsa_constraint_flag는 그러한 제약을 가하지 않는다. 플래그 general_one_picture_only_constraint_flag가 1과 동일한 경우, 플래그 no_stsa_constraint_flag는 1과 동일할 수 있다.
- [0184] 일 실시예에서, 플래그 no_trail_constraint_flag는 도 13의 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 no_trail_constraint_flag는 OlsInScope에 존재하는 TRAIL_NUT과 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음을 지정할 수 있다. 0과 동일한 플래그 no_trail_constraint_flag는 그러한 제약이 가하지 않는다. 플래그 general_one_picture_only_constraint_flag가 1과 동일한 경우, 플래그 no_trail_constraint_flag는 1과 동일할 수 있다.
- [0185] 일 실시예에서, 플래그 no_stsa_constraint_flag는 도 13의 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 no_stsa_constraint_flag는 OlsInScope에 존재하는 STSA_NUT와 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음을 지정할 수 있다. 0과 동일한 플래그 no_stsa_constraint_flag는 그러한 제약을 가하지 않는다. 플래그 general_one_picture_only_constraint_flag가 1과 동일한 경우, 플래그 no_stsa_constraint_flag는 1과 동일할 수 있다.
- [0186] 일 실시예에서, 플래그 no_idr_constraint_flag는 도 13에 도시된 바와 같은 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 no_idr_constraint_flag는 IDR_W_RADL과 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음 또는 IDR_N_LP가 OlsInScope에 존재할 수 있음을 지시할 수 있다. 0과 동일한 플래그 no_idr_constraint_flag는 그러한 제약을 가하지 않는다
- [0187] 일 실시예에서, 플래그 no_cra_constraint_flag는 도 13에 도시된 바와 같은 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 no_cra_constraint_flag는 OlsInScope에 존재하는 CRA_NUT과 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음을 지정할 수 있다. 0과 동일한 플래그 no_cra_constraint_flag는 그러한 제약을 가하지 않는다
- [0188] 일 실시예에서, 플래그 no_rasl_constraint_flag는 도 13의 일반 제약 정보 선택스 구조에 존재할 수 있다(플래그 no_rasl_constraint_flag는 도시되지 않음). 1과 동일한 플래그 no_rasl_constraint_flag는 OlsInScope에 존재하는 RASL_NUT과 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음을 지정할 수 있다. 0과 동일한 플래그 no_rasl_constraint_flag는 그러한 제약을 가하지 않는다. 플래그 no_cra_constraint_flag가 1과 동일한 경우, 플래그 no_rasl_constraint_flag의 값은 1과 동일할 수 있다.
- [0189] 일 실시예에서, 플래그 no_radl_constraint_flag는 도 13에 도시된 바와 같은 일반 제약 정보 선택스 구조에 존재할 수 있다. 1과 동일한 플래그 no_radl_constraint_flag는 OlsInScope에 존재하는 RADL_NUT과 동일한 nuh_unit_type을 갖는 NAL 유닛이 없을 수 있음을 지정할 수 있다. 0과 동일한 플래그 no_radl_constraint_flag는 그러한 제약을 가하지 않는다. 플래그 no_idr_constraint_flag가 1과 동일하고 플래그 no_cra_constraint_flag가 1과 동일한 경우, 플래그 no_rasl_constraint_flag의 값은 1과 동일할 수 있다.
- [0190] 본 개시의 일부 측면은 잔차 라이스 코딩 확장을 사용한 범위 확장과 같은 범위 확장을 위한 제약 플래그 시그널링을 위한 기술을 제공한다.
- [0191] 본 개시의 일 측면에 따르면, 라이스 코딩에 기초하여 잔차 코딩이 수행될 수 있다. 라이스 코딩은 조정 가능한 파라미터를 사용하여 입력 값을 두 부분으로 나눈 다음 서로 다른 코딩 기술을 사용하여 두 부분을 코딩한다. 일부 예에서는 조정 가능 파라미터를 라이스 파라미터라고 한다. 라이스 파라미터는 라이스 코딩의 코딩 효율에 영향을 줄 수 있다. 특정 애플리케이션에 대한 코딩 효율을 개선하기 위해 특정 애플리케이션에 대한 라이스 파라미터를 결정하는 기술이 사용될 수 있다. 라이스 파라미터를 결정하기 위한 일부 기술은 비디오 표준의 범위

확장에 포함될 수 있다.

- [0192] 본 개시의 일 측면에 따르면, 일부 표준은 원래 특정 애플리케이션을 위해 개발될 수 있다. 다른 애플리케이션에 적용할 수 있는 표준을 만들기 위해, 다른 애플리케이션을 지원하는 도구를 사용하여 범위 확장이 개발된다. 예를 들어 HEVC는 원래 샘플당 8-10비트의 4:2:0 크로마 포맷의 애플리케이션을 타깃으로 한다. 특정 크로마 포맷 및 특정 비트 심도 외에 다른 포맷 및 비트 심도에 적용할 수 있는 표준을 만들기 위해, 다른 크로마 포맷 및/또는 더 높은 비트 심도를 사용하는 애플리케이션을 지원하도록 범위 확장이 개발될 수 있다.
- [0193] 특정 그룹의 애플리케이션에 필요한 것으로 특정 세트를 제한하기 위해, 비디오 코딩 표준은 이러한 특징을 사용하는 인코더와의 상호 운용성을 위해 지원되는 정의된 디코더 기능 세트를 포함할 수 있는 프로파일을 정의한다. 예를 들어, 프로파일은 적합한 비트스트림(conforming bitstream)을 생성하는 데 사용할 수 있는 코딩 도구 또는 알고리즘 세트를 정의할 수 있다. 프로파일 외에도 일부 표준(예: VVC, HEVC 등)도 레벨과 티어를 정의한다. 레벨은 디코더 처리 부하 및 메모리 능력에 대응할 수 있는 공간 해상도, 픽셀 레이트, 비트 레이트 값 및 변형과 관련된 비트스트림에 제한을 가한다. 레벨 제한은 최대 샘플 레이트, 최대 픽처 크기, 최대 비트 레이트, 최소 압축률, 코딩된 픽처 버퍼의 용량 등의 관점에서 나타낼 수 있다. 레벨의 값이 높을수록 더 높은 복잡도 한계에 대응할 수 있다. 티어는 각각의 레벨에 대한 비트 레이트 값과 변형 제한을 수정한다. 예를 들어, 메인 티어는 대부분의 애플리케이션을 위한 것이고, High 높은 계층은 비디오 배포 애플리케이션보다 훨씬 더 높은 비트 레이트 값을 갖는 것과 같은, 더 까다로운 비디오 제공 애플리케이션(video contribution application)을 처리하도록 설계된다. 프로파일, 티어 및 레벨 각각은 구현 및 디코딩 복잡도에 영향을 미치며, 이 셋의 조합은 비트스트림 및 디코더에 대한 상호 운용성 지점을 지정한다.
- [0194] 일부 예에서, 특정 티어 및 레벨에 준거하는 디코더는 동일한 티어 또는 해당 레벨의 하위 티어 또는 그 아래의 임의의 레벨에 준거하는 모든 비트스트림을 디코딩할 수 있어야 하며, 특정 계층 및 레벨에 준거하는 디코더는 해당 프로파일의 모든 특징을 지원할 수 있다. 일부 예에서, 인코더는 프로파일에서 지원되는 특정 특징 세트를 사용할 필요가 없지만, 준거 비트스트림(conforming bitstream), 즉 준거 디코더에 의해 디코딩될 수 있도록 하는 지정된 제약을 따르는 비트스트림을 생성해야 한다.
- [0195] PTL 정보에 더하여, PTL 선택스 구조는 비트스트림의 특정 제약 속성을 지시하는 비플래그 선택스 요소 및 제약 플래그의 목록을 포함하는 일반 제약 정보(general constraints information, GCI) 선택스 구조를 또한 포함할 수 있다.
- [0196] 일례에서, HEVC는 본래 메인(Main) 프로파일, 메인 10 프로파일 및 메인 스틸 픽처(Main Still Picture) 프로파일로 지칭되는 3개의 프로파일을 포함한다. 2개의 프로파일에는 4:2:0 크로마 샘플링만 지원하는 것과 같은 몇 가지 제한사항이 있다. 메인 및 메인 스틸 픽처 프로파일에서는 샘플당 8비트의 비디오 정밀도만 지원되는 반면, 메인 10 프로파일은 샘플당 최대 10비트까지 지원한다. 메인 스틸 픽처 프로파일에서는 전체 비트스트림은 하나의 코딩된 픽처만 포함한다.
- [0197] 일부 예에서, 범위 확장이 있는 HEVC는 추가 프로파일을 지원할 수 있다. 일례에서, 다음 프로파일은 집합적으로 범위 확장 프로파일로 지칭된다: 모노크롬(Monochrome) 프로파일, 모노크롬 10 프로파일, 모노크롬 12 프로파일, 모노크롬 16 프로파일, 메인 12 프로파일, 메인 4:2:2 프로파일, 메인 인트라 프로파일, 메인 10 인트라 프로파일, 메인 12 인트라 프로파일, 메인 4:2:2 10 인트라 프로파일, 메인 4:2:2 12 인트라 프로파일, 메인 4:4:4 인트라 프로파일, 메인 4:4:4 10 인트라 프로파일, 메인 4:4:4 12 인트라 프로파일, 메인 4:4:4 16 인트라 프로파일, 메인 4:4:4 스틸 픽처 프로파일 및 메인 4:4:4 16 스틸 픽처 프로파일.
- [0198] 범위 확장 프로파일 중 일부는 더 높은 비트 심도를 지원할 수 있고 높은 비트 심도를 가진 동작 범위 확장을 위한 프로파일이라고 할 수 있다. 일부 예에서, 비트 심도가 높은 작동 범위 확장을 위한 프로파일에는 메인 12 프로파일, 메인 12 4:4:4 프로파일, 메인 16 4:4:4 프로파일, 메인 12와 같이 샘플당 10비트 이상을 지원하는 프로파일도 포함된다. 인트라 프로파일, 메인 12 4:4:4 인트라 프로파일, 메인 16 4:4:4 인트라 프로파일, 메인 12 스틸 픽처 프로파일, 메인 12 4:4:4 스틸 픽처 프로파일, 메인 16 4:4:4 스틸 픽처 프로파일 등.
- [0199] 구체적으로, 메인 12 프로파일은 인트라 예측 및 인터 예측 모드 모두에서 4:0:0 및 4:2:0 크로마 샘플링을 지원하여 샘플당 8비트에서 12비트의 비트 심도를 허용한다. 일부 예에서 메인 12 프로파일에 준거하는 디코더는 다음 프로파일로 만들어진 비트스트림을 디코딩할 수 있다: 모노크롬, 모노크롬 12, 메인, 메인 10 및 메인 12
- [0200] 메인 12 4:4:4 프로파일은 4:0:0, 4:2:0, 4:2:2 및 4:4:4 크로마 샘플링을 지원하여 샘플당 8비트에서 12비트의 비트 심도, 그리고 인트라 예측 및 인터 예측 모드 둘다를 허용한다. 일부 예에서, 메인 12 4:4:4 프로파일

에 준거하는 디코더는 다음 프로파일로 만들어진 비트스트림을 디코딩할 수 있다: 모노크롬, 메인, 메인 10, 메인 12, 메인 10 4:2:2, 메인 12 4: 2:2, 메인 4:4:4, 메인 10 4:4:4, 메인 12 4:4:4, 모노크롬 12.

- [0201] 메인 16 4:4:4 프로파일은 4:0:0, 4:2:0, 4:2:2 및 4:4:4 크로마 샘플링, 그리고 인트라 예측 및 인터 예측 모드 둘다를 지원하여 샘플당 8비트에서 16비트의 비트 심도를 허용한다.
- [0202] 메인 12 인트라 프로파일은 4:0:0 및 4:2:0 크로마 샘플링, 그리고 인트라 예측 모드를 지원하여 샘플당 8비트에서 12비트의 비트 심도를 허용한다.
- [0203] 메인 12 4:4:4 인트라 프로파일은 4:0:0, 4:2:0, 4:2:2 및 4:4:4 크로마 샘플링, 그리고 인트라 예측 모드를 지원하여 샘플당 8비트에서 12비트의 비트 심도를 허용한다.
- [0204] 메인 16 4:4:4 인트라 프로파일은 4:0:0, 4:2:0, 4:2:2 및 4:4:4 크로마 샘플링, 그리고 인트라 예측 모드를 지원하여 샘플당 8비트에서 16비트의 비트 심도를 허용한다.
- [0205] 메인 12 스틸 픽처 프로파일은 4:0:0 및 4:2:0 크로마 샘플링을 지원하여 샘플당 8비트에서 12비트의 비트 심도를 허용한다. 메인 12 스틸 픽처 프로파일에서, 전체 비트스트림은 단 하나의 코딩된 픽처를 포함한다.
- [0206] 메인 12 4:4:4 스틸 픽처 프로파일은 4:0:0, 4:2:0, 4:2:2 및 4:4:4 크로마 샘플링을 지원하여 샘플당 8비트에서 12비트의 비트 심도를 허용한다.
- [0207] 메인 12 4:4:4 스틸 픽처 프로파일에서 전체 비트스트림은 단 하나의 코딩된 픽처만 포함한다.
- [0208] 메인 16 4:4:4 스틸 픽처 프로파일은 4:0:0, 4:2:0, 4:2:2, 및 4:4:4 크로마 샘플링을 지원하여 샘플당 8비트에서 12비트의 비트 심도를 허용한다. 메인 16 4:4:4 스틸 픽처 프로파일에서, 전체 비트스트림은 하나의 코딩된 픽처만 포함한다.
- [0209] 본 개시의 일부 측면에 따르면, 코딩 도구 제어는 비트스트림의 범위, 코딩된 계층 비디오 시퀀스(CLVS)의 범위, 픽처, 픽처의 슬라이스 등과 같은, 다양한 범위(예: 코딩 도구 제어를 위한 신택스 요소의 인스턴스의 지속성으로 코딩되는 코딩된 비디오 데이터의 일부)에서 수행될 수 있다. 일부 예에서, 코딩 도구 제어는 일반적으로 디코더에 대한 출력 계층 세트를 실어 전달하는 비트스트림에 대한 제약 정보를 포함하는 일반 제약 정보(GCI) 신택스 구조에서 제공될 수 있다. 일부 예에서, 코딩 도구 제어는 CLVS와 연관된 시퀀스 파라미터 세트(SPS)에서 제공될 수 있으며, SPS는 일반적으로 CLVS에 대한 정보를 포함한다. 일부 예에서, 코딩 도구 제어는 픽처에 대한 픽처 헤더에 제공될 수 있고, 픽처 헤더는 일반적으로 픽처에 대한 정보를 포함한다. 일부 예에서, 코딩 도구 제어는 슬라이스의 슬라이스 헤더에 제공될 수 있으며, 슬라이스 헤더는 일반적으로 슬라이스에 대한 정보를 포함한다.
- [0210] 본 개시의 일 측면에 따르면, 범위 확장에서의 코딩 도구에 대한 제어 정보는 다양한 범위에서 제공될 수 있다. 일부 예에서 더 큰 범위의 신택스 요소를 사용하면 코딩 효율을 향상시킬 수 있다. 예를 들어, 0보다 큰 GCI 신택스 요소 값은 일반적으로 특정 코딩 도구가 비트스트림에서 사용되지 않음을 지시하기 위해 비트스트림이 특정 방식으로 제한됨을 지시한다. 또한, 값 0과 동일한 GCI 신택스 요소 값은 연관된 제약이 적용되지 않을 수 있다고 시그널링하여, 연관된 코딩 도구가 비트스트림(지시된 프로파일에서 사용이 지원되는 경우)에서 사용될 수 있도록 한다(필수는 아님).
- [0211] 본 개시의 다른 측면에 따르면, 비트스트림에서 비디오 데이터의 코딩에 코딩 도구가 사용되지 않는 경우, 예를 들어 PTL 정보 및/또는 일반 제약 정보에서 코딩 도구의 사용 없음을 지시하는 경우, 코딩 도구의 지원이 없는 비디오 디코더는 비디오 디코더가 PTL 정보 및/또는 일반 제약 정보의 시그널링에 기초하여 비트스트림을 디코딩할 수 있음을 결정할 수 있으며, 비디오 디코더의 기능은 확장될 수 있다.
- [0212] 일부 실시예에서, 인코더는 범위 확장이 있는 비디오 표준에 준거하는 비트스트림을 생성할 수 있지만, 범위 확장에서 지원되는 하나 이상의 특징을 사용하지 않는다. 일부 예에서, 범위 확장에서 하나 이상의 특징을 사용하지 않는다는 것을 알고 있으면, 비디오 표준에 준거하지만 범위 확장에서 하나 이상의 특징을 지원하지 않는 디코더는, 디코더가 비트스트림을 디코딩할 수 있는 것으로 결정할 수 있고, 비트스트림을 거부하는 대신 디코딩을 위해 비트스트림을 수락할 수 있다.
- [0213] 도 16은 본 개시의 일부 실시예에 따른 일반 제약 정보의 신택스 구조(1600)를 도시한다. 일부 예에서, 신택스 구조(1600)는 디코더에 대한 출력 계층 세트를 포함하는 비트스트림과 같은, 비트스트림에 적용될 제약을 포함한다. 도 16의 예에서, 신택스 구조(1600)에서 gci_num_additional_bits로 표시된 신택스 요소는 정렬 제로 비

트 선택스 요소(존재하는 경우) 이외의 일반 제약 정보 선택스 구조(1600)에서 추가의 일반 제어 정보(GCI) 비트의 수를 지정하는 데 사용된다. 일부 표준에서, gci_num_additional_bits의 값은 0 또는 1이어야 한다. 일부 표준에서, 디코더는 1보다 큰 gci_num_additional_bits의 값이 선택스 구조에 나타나도록 허용할 수 있다.

[0214] 도 16의 예에서, 선택스 구조(1600)는 general_no_extended_precision_constraint_flag, general_no_ts_residual_coding_rice_present_in_sh_constraint_flag, general_no_rrc_rice_extension_constraint_flag, general_no_persistent_rice_adaptation_constraint_flag, 및 general_no_reverse_last_sig_coeff_constraint_flag로 표시되는 5개의 추가 GCI 비트(선택스 요소)(1601)~(1605)을 포함한다. 5개의 추가 GCI 비트(1601)~(1605)는 각각 일부 예에서 출력 계층 세트의 비트스트림의 범위에서 코딩 도구의 코딩 제어 정보를 제공한다.

[0215] 도 17은 본 개시의 일부 실시예에 따른 시퀀스 파라미터 세트(SPS) 범위 확장의 선택스 구조(1700) 예를 도시한다. 선택스 구조(1700)는 CLVS의 SPS에 추가되어 CLVS에 대한 범위 확장의 코딩 도구의 제어를 제공할 수 있다. 선택스 구조(1700)는 sps_extended_precision_flag, sps_ts_residual_coding_rice_present_in_sh_flag, sps_rrc_rice_extension_flag, sps_persistent_rice_adaptation_enabled_flag 및 sps_reverse_last_sig_coeff_enabled_flag로 표시되는 5개의 선택스 요소(1701)~(1705)를 포함한다. 5개의 선택스 요소(1701)~(1705)는 일부 예에서 CLVS 범위 내의 코딩 도구의 코딩 제어 정보를 제공한다.

[0216] 구체적으로, 일 실시예에서, GCI 비트(1601) 및 선택스 요소(1701)는 스케일링 및 변환 프로세스에서의 변환 계수 및 abs_remainder[] 및 dec_abs_level[] 등과 같은 일부 선택스 요소의 이진화를 위해 확장된 동적 범위의 코딩 도구의 제어와 같은, 확장된 정밀도를 사용하는 제어를 다른 범위에서 제공하는 데 사용된다.

[0217] 1과 동일한 선택스 요소(1701)는 확장된 동적 범위가 스케일링 및 변환 프로세스의 변환 계수에 대해 그리고 abs_remainder[] 및 dec_abs_level[] 등과 같은 일부 선택스 요소의 이진화에 사용됨을 지정한다. 선택스 요소 abs_remainder[스캐닝 위치 n]는 스캐닝 위치 n에서 Golomb-Rice 코드로 코딩되는 변환 계수 레벨의 나머지 절댓값이다. abs_remainder[]는 존재하지 않는 경우, 0과 동일한 것으로 추론된다. 선택스 요소 dec_abs_level[스캐닝 위치 n]은 스캐닝 위치 n에서 Golomb-Rice 코드로 코딩되는 중간 값에 대응할 수 있으며 스캐닝 위치 n에서의 변환 계수의 레벨을 결정하는 데 사용될 수 있다. 0과 동일한 선택스 요소(1701)는 확장된 동적 범위가 스케일링 및 변환 프로세스에서 사용되지 않고, 예를 들어 선택스 요소 abs_remainder[] 및 dec_abs_level[] 등의 이진화에 사용되지 않음을 지정한다. 존재하지 않는 경우, 선택스 요소(1701)의 값은 0과 동일한 것으로 추론된다.

[0218] 일례에서, Log2TransformRange로 표시된 변수는 스케일링 및 변환 프로세스에서 변환 계수 및 특정 선택스 요소의 이진화에 대한 동적 범위를 결정하는 데 사용된다. 예를 들어, 변수 Log2TransformRange는 스케일링 및 변환 프로세스에서 변환 계수를 나타내기 위한 비트 수 및 특정 선택스 요소의 이진화를 위한 비트 수일 수 있다. 동적 범위는 비트 수를 사용하여 표현되는 최대 수와 최소 수의 차이일 수 있다. 일례에서, 변수 Log2TransformRange는 식 (1)과 사용하는 것과 같이, 선택스 요소(1701) sps_extended_precision_flag에 따라 도출된다:

$$\text{Log2TransformRange} = \text{sps_extended_precision_flag} ? \text{Max}(15, \text{Min}(20, \text{BitDepth} + 6)) : 15$$

식 (1)

[0219] .

[0220] 스케일링 및 변환 프로세스에서의 변환 계수 및 특정 선택스 요소의 이진화에 대한 동적 범위는 변수 Log2TransformRange에 기초하여 결정될 수 있다. 일부 예에서, 플래그 sps_extended_precision_flag의 값이 0인 경우, 확장된 동적 범위 특징(예: 확장된 동적 범위의 코딩 도구)이 사용되지 않고, 변환 계수의 동적 범위가 15비트와 같은 고정된 수의 비트에 기초한다. 플래그 sps_extended_precision_flag의 값이 1인 경우, 확장된 동적 범위 특징이 인에이블되며, 스케일링 및 변환 처리에서 변환 계수를 나타내는 비트 수는 식 (1)의 예에서 비트 심도 BitDepth에 기초한 15비트, 16비트, 17비트, 18비트, 19비트 및 20비트 중 하나일 수 있다. 변환 계수의 동적 범위는 비트 수에 기초하여 결정될 수 있다.

[0221] 본 개시의 일 측면에 따르면, 선택스 요소(예: sps_bitdepth_minus8로 표시됨)는 루마 및 크로마 배열의 샘플의 비트 심도(예: BitDepth로 표시됨), 그리고 루마 및 크로마 양자화 파라미터 범위 오프셋(예: QpBdOffset으로 표시됨)를 시그널링하는 데 사용될 수 있다. 일례에서, 비트 심도 BitDepth는 식 (2)에 따라 계산될 수 있으며, QP 범위 오프셋 QpBdOffset은 식 (3)에 따라 계산될 수 있다:

- [0222] $BitDepth = 8 + sps_bitdepth_minus8$ 식 (2)
- [0223] $QpBdOffset = 6 \times sps_bitdepth_minus8$ 식 (3)
- [0224] 일부 예에서, 1과 동일한 GCI 비트(1601)는 출력 계층 세트(01sInScope)에 대한 범위 내의 모든 픽처에 대한 선택 요소(1701)가 0과 동일할 수 있음을 지정한다. 0과 동일한 GCI 비트(1601)는 이러한 제약이 가하지 않는다. 따라서, 1과 동일한 GCI 비트(1601)는 비트스트림의 코딩에서 확장된 동적 범위 코딩 도구를 사용하지 않음을 지정할 수 있다.
- [0225] 일부 실시예에서, GCI 비트(1602) 및 선택 요소(1702)는 상이한 범위에서, 변환 스킵 모드에서 잔차 코딩을 위한 슬라이스 기반 라이스 파라미터 선택과 같은, 변환 스킵 모드에서 잔차 코딩을 위한 슬라이스 기반 라이스 코딩의 코딩 도구의 제어를 제공하는 데 사용된다.
- [0226] 본 개시의 일 측면에 따르면, 변환 스킵 잔차 코딩을 위한 슬라이스 기반 라이스 파라미터 선택은 비디오 표준의 범위 확장에 포함될 수 있다. 일부 예에서, 하나의 제어 플래그(예: `sps_ts_residual_coding_rice_present_in_sh_flag`로 표시됨, 선택 요소(1702))는 변환 스킵 모드가 인에이블되는 경우(예: 선택 요소 `sps_transform_skip_enabled_flag`가 참), 시퀀스 파라미터 세트(SPS)에서 시그널링되어 변환 스킵 슬라이스에 대한 라이스 파라미터가 도 17에 도시된 바와 같이 인에이블 또는 디스에이블됨을 지시한다.
- [0227] 제어 플래그가 인에이블(예: 1과 동일함)된 것으로 시그널링되는 경우, 하나의 선택 요소(예: `sh_ts_residual_coding_rice_idx_minus1`로 표시됨)가 예를 들어 슬라이스 헤더에서 각각의 변환 스킵 슬라이스에 대해 추가로 시그널링되어, 그 변환 스킵 슬라이스의 라이스 파라미터 선택의 선택을 지시한다. 제어 플래그가 디스에이블(예: "0"과 동일함)으로 시그널링되는 경우, 변환 건너뛰기 슬라이스에 대한 라이스 파라미터 선택을 지시하기 위해 슬라이스 레벨(예: 슬라이스 헤더)에서 추가 선택 요소가 시그널링되지 않고, 디폴트 라이스 파라미터가 일례에서 SPS를 참조하는 코딩된 비디오 데이터의 모든 변환 스킵 슬라이스에 사용될 수 있다.
- [0228] 예를 들어, SPS에서 1과 동일한 선택 요소(1702)는 `sh_ts_residual_coding_rice_idx_minus1`로 표시된 슬라이스 헤더 플래그가 SPS를 참조하는 슬라이스의 슬라이스 헤더(예: `slice_header()`) 선택 구조에 존재할 수 있음을 지정한다. SPS에서 0과 동일한 선택 요소(1702)는 슬라이스 헤더 플래그 `sh_ts_residual_coding_rice_idx_minus1`이 SPS를 참조하는 슬라이스의 `slice_header()` 선택 구조에 존재하지 않음을 지정한다. 존재하지 않는 경우, `sps_ts_residual_coding_rice_present_in_sh_flag`의 값은 일부 예에서 0과 동일한 것으로 추론된다.
- [0229] 일부 예에서, 선택 요소는 출력 계층 세트의 범위에서, 변환 스킵 모드에서 잔차 코딩을 위한 슬라이스 기반 라이스 코딩의 코딩 도구의 사용을 제어하기 위해 일반 제약 정보에 포함될 수 있다. 예를 들어, 1과 동일한 선택 요소(1602)는 출력 계층 세트(01sInScope)에 대한 범위 내의 모든 픽처에 대한 선택 요소(1702)가 0과 동일할 수 있음을 지정한다. 0과 동일한 선택 요소(1602)는 그러한 제약을 가하지 않는다. 따라서, 일부 예에서, 비트스트림에서 1과 동일한 GCI 비트(1602)는 비트스트림을 코딩하기 위한 변환 스킵 잔차 코딩을 위한 슬라이스 기반 라이스 파라미터 선택의 사용을 지정하지 않을 수 있다.
- [0230] 일부 실시예에서, GCI 비트(1603) 및 선택 요소(1703)는 상이한 범위에서, 일반 잔차 코딩(regular residual coding, RRC)에서 `abs_remainder[]` 및 `dec_abs_level[]` 등과 같은, 일부 선택 요소의 이진화를 위한 라이스 파라미터 도출을 위한 하나 이상의 코딩 도구의 제어를 제공하는 데 사용된다. 일부 예에서, 일반 잔차 코딩(RRC)는 변환 및 양자화에 의해 획득되는 블록을 코딩하기 위한 일부 기술을 말한다. 일부 예에서, RRC는 양자화에 의해서만 획득된 블록들에 대해 수정될 수 있다. 일부 예에서, 변환 스킵 잔차 코딩(transform skip residual coding, TSRC)은 변환을 우회하여(변환 스킵으로도 지칭됨) 획득되는 코딩 블록에 전용인 일부 기술을 말한다.
- [0231] 일부 예에서, 비디오 코딩 표준은 `abs_remainder[]` 및 `dec_abs_level[]`과 같은 일부 선택 요소의 이진화를 위한 라이스 파라미터 도출을 위한 하나 이상의 코딩 도구를 포함할 수 있으며, 비디오 코딩 표준의 범위 확장은 `abs_remainder[]` 및 `dec_abs_level[]`과 같은 일부 선택 요소의 이진화를 위한 라이스 파라미터 도출을 위한 하나 이상의 대안 코딩 도구를 포함한다.
- [0232] 일부 예에서, 비디오 표준은 라이스 파라미터 도출을 위한 로컬 템플릿 기반 기술을 사용한다. 예를 들어, 하나 이상(예: 예에서 5개)의 이웃 계수 레벨을 포함하는 템플릿이 라이스 파라미터 도출에 사용된다. 예를 들어, 템

플릿 내부의 절대 계수 값의 합이 계산될 수 있으며, 그러면 그 합에 기초하여 라이스 파라미터가 결정된다. 일례에서, 순람표(look up table)를 사용하여 합에 기초하여 라이스 파라미터가 결정될 수 있다.

- [0233] 라이스 파라미터는 다른 적절한 코딩 도구에 의해 결정될 수 있음에 유의한다. 일례에서, 합에 기초하여 식을 사용하여 라이스 파라미터를 결정할 수 있다. 다른 예에서, 이웃 계수 레벨의 통계에 기초하여 라이스 파라미터를 결정하기 위해 컨텍스트 모델링을 사용할 수 있다. 일부 예에서, 비디오 표준의 범위 확장은 라이스 파라미터 도출을 위한 하나 이상의 대안 코딩 도구를 지정할 수 있다.
- [0234] 일부 예에서, 비디오 표준의 범위 확장은 다른 시나리오에서의 사용을 위해 RRC에 대한 수정을 포함할 수 있다. 일례에서, 범위 확장은 변환 스킵 모드에서 잔차 코딩을 위한 다른 컨텍스트 모델링 도구 및 잔차 시그널링 회선 도구를 포함할 수 있다.
- [0235] 일부 예에서, 1과 동일한 SPS의 선택 요소(1703)는 `abs_remainder[]` 및 `dec_abs_level[]`의 이진화를 위한 대안 라이스 파라미터 도출(예: 범위 확장에서 라이스 파라미터 도출을 위한 대안 코딩 도구)이 SPS를 참조하는 CLVS를 코딩하는 데 사용됨을 지정한다. 0과 동일한 선택 요소(1703)는 `abs_remainder[]` 및 `dec_abs_level[]`의 이진화를 위한 대안 라이스 파라미터 도출이 SPS를 참조하는 CLVS를 코딩하는 데 사용되지 않음을 지정한다. 존재하지 않는 경우, 선택 요소(1703)의 값은 0과 동일한 것으로 추론된다.
- [0236] 일부 예에서, 1과 동일한 선택 요소(1603)는 출력 계층 세트(0IsInScope)의 범위 내의 모든 픽처에 대한 선택 요소(1703)가 0과 동일할 수 있음을 지정한다. 0과 동일한 선택 요소(1603)는 이러한 제약을 가하지 않는다. 따라서, 일부 예에서, 1인 GCI 비트(1603)는 비트스트림을 코딩하기 위한 `abs_remainder[]` 및 `dec_abs_level[]`의 이진화를 위해 대안 라이스 파라미터 도출(예: 지정된 범위 확장에 지정된 라이스 파라미터 도출을 위한 대안 코딩 도구)을 사용하지 않음을 지정할 수 있다.
- [0237] 일부 실시예에서, GCI 비트(1604) 및 선택 요소(1704)는 상이한 범위에서 `abs_remainder[]` 및 `dec_abs_level[]`의 이진화를 위한 통계 기반 라이스 파라미터 도출의 제어를 제공하는 데 사용된다.
- [0238] 본 개시의 일 측면에 따르면, `abs_remainder[]` 및 `dec_abs_level[]`의 이진화를 위한 라이스 파라미터 도출은 이전 TU로부터 누적된 통계를 사용하여 각각의 변환 유닛(TU)의 시작에서 초기화될 수 있다. 일부 예에서, 통계 기반 라이스 파라미터 도출은 비디오 표준의 범위 확장에 포함될 수 있다.
- [0239] 일부 예에서, 제어 플래그, 예컨대 SPS에서 `sps_persistent_rice_adaptation_enabled_flag`로 표시되는 선택 요소(1704)는 통계 기반 라이스 파라미터 도출을 제어하는 데 사용된다. 예를 들어, SPS에서 1과 동일한 선택 요소(1704)는 `abs_remainder[]` 및 `dec_abs_level[]`의 이진화를 위한 Rice 파라미터 도출이 이전 TU로부터 누적된 통계를 사용하여 각각의 TU의 시작에서 초기화됨을 지정한다. 0과 동일한 선택 요소(1704)는 현재 TU의 라이스 파라미터 도출에 이전 TU 상태가 사용되지 않음을 지정한다. 존재하지 않는 경우, 선택 요소(1704)의 값은 0과 동일한 것으로 추론된다.
- [0240] 또한, 일 실시예에서, 1과 동일한 선택 요소(1604)는 출력 계층 세트(0IsInScope)에 대한 범위 내의 모든 픽처에 대한 선택 요소(1704)가 0과 동일할 수 있음을 지정한다. 0과 동일한 선택 요소(1604)는 이러한 제약을 가하지 않는다. 따라서, 일부 예에서, 1과 동일한 GCI 비트(1604)는 비트스트림을 코딩하기 위한 통계 기반 라이스 파라미터 도출을 사용하지 않음을 지정할 수 있다.
- [0241] 일부 실시예에서, GCI 비트(1605) 및 선택 요소(1705)는 서로 다른 범위에서, 변환 계수의 엔트로피 코딩 동안 마지막 유효 계수의 위치를 코딩하는 데 사용되는 코딩 도구의 제어를 제공하는 데 사용된다. 일례에서, 마지막 유효 계수의 위치는 다른 코딩 도구에 의해 코딩될 수 있다. 예를 들어, 비디오 표준은 `LastSignificantCoeffX` 및 `LastSignificantCoeffY` 변수에 의해 표시되는 위치의 두 좌표를 코딩함으로써 마지막 유효 계수의 위치를 결정할 수 있는 제1 코딩 도구를 지정할 수 있고; 비디오 표준의 범위 확장은 일례에서 제로 아웃(zero-out) 변환 블록의 우측 하단 모서리를 참조하여 마지막 유효 계수의 상대 좌표를 코딩함으로써 마지막 유효 계수의 위치를 결정할 수 있는 제2 코딩 도구와 같은, 대안 코딩 도구를 지정할 수 있다.
- [0242] 일부 예에서, SPS에서 1과 동일한 선택 요소(1705)는 `sh_reverse_last_sig_coeff_flag`로 표시되는 슬라이스 헤더 플래그(슬라이스 범위)가 SPS를 참조하는 슬라이스 헤더 선택 구조(예: 일부 예에서 `slice_header()`)에 존재함을 지정한다. SPS에서 0과 동일한 선택 요소(1705)는 슬라이스 헤더 플래그 `sh_reverse_last_sig_coeff_flag`가 SPS를 참조하는 슬라이스 헤더 선택 구조에 존재하지 않음을 지정하고, 슬라이스 헤더 플래그 `sh_reverse_last_sig_coeff_flag`는 0으로 추론될 수 있다. 존재하지 않는 경우, 선택 요소

소(1705)의 값은 0과 동일한 것으로 추론된다.

- [0243] 일부 예에서, 슬라이스의 슬라이스 헤더 플래그 `sh_reverse_last_sig_coeff_flag`의 값은 슬라이스의 코딩에 있어 스케일링 및 변환 프로세스에서 변환 계수 중 마지막 유효 계수의 위치 도출을 결정하는 데 사용된다. 일례에서, `sh_reverse_last_sig_coeff_flag`가 1과 동일한 경우, 마지막 유효 계수 위치는 제2 코딩 도구와 같은, 비디오 표준의 범위 확장에서 대안 코딩 도구에 의해 코딩되고; 그렇지 않으면 마지막 유효 계수 위치에 대한 현재 좌표가 제1 코딩 도구에 의해 코딩된다.
- [0244] 일부 예에서, 1과 동일한 GCI 비트(1605)는 출력 계층 세트(0IsInScope)에 대한 범위 내의 모든 픽처에 대한 선택 요소(1705)가 0과 동일할 수 있음을 지정한다. 0과 동일한 GCI 비트(1605)는 이러한 제약을 가하지 않는다. 따라서, 1과 동일한 GCI 비트(1605)는 비트스트림의 범위에 대한 마지막 유효 계수의 위치 도출에서 제2 코딩 도구를 사용하지 않음을 지정할 수 있다.
- [0245] 도 18은 본 개시의 실시예에 따른 프로세스(1800)를 개략적으로 설명하는 흐름도를 도시한다. 프로세스(1800)는 비디오 디코더에서 사용될 수 있다. 다양한 실시예에서, 프로세스(1800)는 단말 기기(310, 320, 330, 340)의 처리 회로, 비디오 디코더(410)의 기능을 수행하는 처리 회로, 비디오 디코더(510)의 기능을 수행하는 처리 회로 등과 같은, 처리 회로에 의해 실행된다. 일부 실시예에서, 프로세스(1800)는 소프트웨어 명령어로 구현되며, 따라서 처리 회로가 소프트웨어 명령어를 실행할 때, 처리 회로는 프로세스(1800)를 수행한다. 프로세스는 (S1801)에서 시작하여 (S1810)으로 진행한다.
- [0246] (S1810)에서, 비트스트림 내의 코딩된 비디오 데이터의 제1 범위(예: 출력 계층 세트)에서의 코딩 제어를 위한 제1 선택 요소(예: `general_no_rrc_rice_extension_constraint_flag`)의 값이 결정된다. 제1 선택 요소는 잔차 코딩에서의 라이스 파라미터 도출을 위한 제1 코딩 도구에 대한 대안인 제2 코딩 도구와 연관된다.
- [0247] 일례에서, 제1 선택 요소는 선택 구조에서의 일반 제약 정보에 대한 추가 비트를 지시하는 선택 구조의 선택 요소(예: `gci_num_additional_bits`)에 응답하여 일반 제약 정보에 대한 선택 구조로부터 디코딩된다.
- [0248] 일부 예에서, 제1 코딩 도구는 표준에서 정의되고, 제2 코딩 도구는 제1 코딩 도구에 대한 대안 코딩 도구로서 표준의 범위 확장에서 정의된다. 일례에서, 비디오 디코더는 표준을 지원할 수 있지만 표준의 범위 확장에서 제2 코딩 도구를 지원하지 않을 수 있다.
- [0249] (S1820)에서, 제1 선택 요소의 값이 제1 값인 경우, 프로세스는 (1830)으로 진행하고; 그렇지 않으면 (S1840)으로 진행한다. 제1 값은 코딩된 비디오 데이터의 하나 이상의 제2 범위(예: 출력 계층 세트 내의 하나 이상의 CLVS)를 포함하는 비트스트림에서 코딩된 비디오 데이터의 제1 범위의 코딩에서 제2 코딩 도구를 사용하지 않음을 나타낸다).
- [0250] 일부 예에서, 제1 선택 요소는 디코더에서 출력되는 출력 계층 세트 내의 픽처의 코딩 제어를 위한 일반 제약 정보이다. 일례에서, 제1 선택 요소의 제1 값은 출력 계층 세트 내의 각각의 코딩된 계층 비디오 시퀀스(CLVS)의 코딩에서 제2 코딩 도구를 사용하지 않음을 지시한다.
- [0251] (S1830)에서, 제1 선택 요소가 제1 값인 것에 응답하여, 비트스트림 내의 코딩된 비디오 데이터의 제1 범위는 제2 코딩 도구를 호출하지 않고 디코딩된다.
- [0252] 일부 예에서, 비트스트림 내의 코딩된 계층 비디오 시퀀스(CLVS)의 코딩 제어를 위한 제2 선택 요소(예: `sps_extended_precision_flag`)는 CLVS를 디코딩하기 위한 제2 코딩 도구를 호출하지 않음을 지시하는 값을 갖도록 제약된다.
- [0253] (S1840)에서, 제1 선택 요소가 제2 값인 것에 응답하여, 비트스트림 내의, 코딩된 계층 비디오 시퀀스(CLVS)와 같은, 코딩된 비디오 데이터의 제2 범위의 코딩 제어를 위한 제2 선택 요소(예: `sps_extended_precision_flag`)의 값이 제2 범위에서의 코딩된 비디오 데이터의 디코딩을 위해 결정된다. 제2 선택 요소는 CLVS에서의 제2 코딩 도구의 사용/미사용을 지시한다. 일례에서, 제2 선택 요소는 CLVS에 대한 CLVS에 대한 시퀀스 파라미터 세트(SPS)에 존재하지 않고, 제2 선택 요소의 값은 CLVS에서의 제2 코딩 도구를 사용하지 않음을 지시하기 위해 추론된다.
- [0254] 코딩된 비디오 데이터의 제2 범위는 그 다음에 제2 선택 요소의 값에 따라 (예: 제2 코딩 도구를 호출하거나 호출하지 않고) 디코딩된다.
- [0255] 프로세스(1800)는 적절하게 적용될 수 있다. 프로세스(1800)의 단계(들)는 수정 및/또는 생략될 수 있다. 추가

단계(들)가 추가될 수 있다. 임의의 적절한 구현 순서가 사용될 수 있다.

- [0256] 도 19는 본 개시의 일 실시예에 따른 프로세스(1900)를 개략적으로 설명하는 흐름도를 도시한다. 프로세스(1900)는 비디오 인코더에 사용될 수 있다. 다양한 실시예에서, 프로세스(1900)는 단말 기기(310, 320, 330, 340)의 처리 회로, 비디오 인코더(403)의 기능을 수행하는 처리 회로, 비디오 인코더(603)의 기능을 수행하는 처리 회로, 비디오 인코더(703)의 기능을 수행하는 처리 회로 등과 같은, 처리 회로에 의해 실행된다. 일부 실시예에서, 프로세스(1900)는 소프트웨어 명령어로 구현되며, 따라서 처리 회로가 소프트웨어 명령어를 실행할 때, 처리 회로는 프로세스(1900)를 수행한다. 프로세스는 (S1901)에서 시작하여 (S1910)으로 진행된다.
- [0257] (S1910)에서, 처리 회로는 비트스트림 내의 코딩된 비디오 데이터의 제1 범위(예: 출력 계층 세트)에서의 인코딩 시에 제1 코딩 도구의 대안인 제2 코딩 도구가 사용되는지를 판정한다. 제2 코딩 도구는 잔차 코딩에서 라이스 파라미터 도출을 위한 제1 코딩 도구에 대한 대안 코딩 도구이다. 코딩된 비디오 데이터의 제1 범위는 코딩된 비디오 데이터의 하나 이상의 제2 범위(예: CLVSs)를 포함한다.
- [0258] 일부 예에서, 처리 회로는 비트스트림에서 코딩된 계층 비디오 시퀀스(CLVS)의 코딩 제어를 위해 제2 선택스 요소(예: sps_extended_precision_flag)에 기초하여 제2 코딩 도구가 사용되는지를 판정할 수 있다.
- [0259] (S1920)에서, 코딩된 비디오 데이터의 제1 범위의 코딩에 제2 코딩 도구가 사용되지 않는 경우, 프로세스는 (S1930)으로 진행하고; 그렇지 않으면 (S1940)으로 진행한다.
- [0260] (S1930)에서, 제1 값을 갖는 제1 선택스 요소(예: general_no_rrc_rice_extension_constraint_flag)가 비트스트림에 인코딩된다. 제1 선택스 요소는 비트스트림 내의 코딩된 비디오 데이터(예: 출력 계층 세트)의 제1 범위에서의 코딩 제어를 위한 것이다. 제1 선택스 요소는 잔차 코딩에서 라이스 파라미터 도출을 위한 제2 코딩 도구와 연관된다. 제1 값은 코딩된 비디오 데이터의 제1 범위의 코딩에서 제2 코딩 도구를 사용하지 않음을 지시한다.
- [0261] 일례에서, 제1 선택스 요소는 일반 제약 정보에 대한 선택스 구조에 인코딩되고, 선택스 구조의 선택스 요소(예: gci_num_additional_bits)는 선택스 구조에서 일반 제약 정보에 대한 추가 비트를 지시하도록 조정된다.
- [0262] (S1940)에서, 제2 값을 갖는 제1 선택스 요소가 비트스트림에 인코딩된다. 일부 예에서, 예를 들어, 제2 값이 제1 선택스 요소에 대한 디폴트 값인 경우, 제1 선택스 요소는 비트스트림에 인코딩되지 않으며, 따라서 제1 선택스 요소는 존재하지 않으면 추론될 수 있고, 그러면 (S1940)을 스킵할 수 있다.
- [0263] 프로세스(1900)는 적절하게 적용될 수 있다. 프로세스(1900)의 단계(들)는 수정 및/또는 생략될 수 있다. 추가 단계가 추가될 수 있다. 구현의 임의의 적절한 순서가 사용될 수 있다.
- [0264] 전술한 기술(예: 제약 플래그, 적응적 해상도 파라미터 등을 시그널링하기 위한 기술)은 컴퓨터로 판독 가능한 명령어를 사용하고 하나 이상의 컴퓨터로 판독 가능한 매체에 물리적으로 저장될 수 있는 컴퓨터 소프트웨어로서 구현될 수 있다. 예를 들어, 도 20은 개시된 주제의 특정 실시예를 구현하기에 적합한 컴퓨터 시스템(2000)을 도시한다.
- [0265] 컴퓨터 소프트웨어는 컴퓨터 중앙 처리 유닛(central processing unit, CPU), 그래픽 처리 유닛(graphics processing unit, GPU) 등에 의해, 어셈블리(assembly), 컴파일(compilation), 링킹(linking), 또는 이와 유사한 메커니즘을 거쳐 직접, 또는 해석(interpretation), 마이크로 코드 실행(micro-code execution) 등을 통해 실행될 수 있는 명령어를 포함하는 코드를 생성할 수 있는 임의의 적절한 기계어 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.
- [0266] 명령어는, 예를 들어 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게이밍 기기, 사물 인터넷 기기 등을 포함한, 다양한 유형의 컴퓨터 또는 그 구성요소에서 실행될 수 있다.
- [0267] 도 20에 도시된 컴퓨터 시스템(2000)의 구성요소는 본질적으로 예시적인 것이며, 본 개시의 실시예를 구현하는 컴퓨터 소프트웨어의 사용 또는 기능의 범위에 대한 어떠한 한정도 시사하려는 것은 아니다. 구성요소의 구성은 컴퓨터 시스템(2000)의 예시적인 실시예에 나타난 구성요소 중 어느 하나 또는 조합과 관련된 임의의 종속성 또는 요건을 가지는 것으로 해석되어서는 안 된다.
- [0268] 컴퓨터 시스템(2000)은 특정 휴먼 인터페이스 입력 기기(human interface input device)를 포함할 수 있다. 이러한 휴먼 인터페이스 입력 기기는 한 명 이상의 인간 사용자에게 의한 입력, 예를 들어 촉각 입력(예: 키 누름

(keystroke), 스와이프(swip), 데이터 장갑(data glove) 움직임), 오디오 입력(예: 음성, 박수), 시각적 입력(예: 제스처), 후각 입력(도시되지 않음)에 응답할 수 있다. 휴먼 인터페이스 기기는 또한 오디오(예: 음성, 음악, 주변 소리), 이미지(예: 스캔된 이미지, 정지 이미지 카메라로부터 획득한 사진 이미지), 비디오(예: 2차원 비디오, 입체 비디오(stereoscopic video)를 포함한 3차원 비디오)와 같은, 사람에게 의한 의식적 입력과 반드시 직접 관련이 있는 것은 아닌 특정 미디어를 캡처하는 데 사용될 수도 있다.

[0269] 입력 휴먼 인터페이스 기기는 키보드(2001), 마우스(2002), 트랙 패드(2003), 터치 스크린(2010), 데이터 장갑(도시되지 않음), 조이스틱(2005), 마이크로폰(2006), 스캐너(2007), 카메라(2008) 중 하나 이상(각각 하나만 표시됨)을 포함할 수 있다.

[0270] 컴퓨터 시스템(2000)은 특정 휴먼 인터페이스 출력 기기를 포함할 수 있다. 이러한 휴먼 인터페이스 출력 기기는 예를 들어 촉각 출력, 소리, 빛 및 냄새/맛을 통해, 한 명 이상의 인간 사용자의 감각을 자극할 수 있다. 이러한 휴먼 인터페이스 출력 기기는 촉각 출력 기기(예: 터치 스크린(2010), 데이터 장갑(도시되지 않음), 또는 조이스틱(2005)에 의한 촉각 피드백이지만, 입력 기기의 역할을 하지 않는 촉각 피드백 기기도 있을 수 있음), 오디오 출력 기기(예: 스피커(2009), 헤드폰(도시되지 않음)), 시각적 출력 기기(예: 각각 터치 스크린 입력 기능이 있거나 없는, 각각 촉각 피드백 기능이 있거나 없는, CRT 스크린, LCD 스크린, 플라즈마 스크린, OLED 스크린을 포함한, 스크린(2010) - 그 일부는 스테레오그래픽 출력(stereographic), 가상 현실 안경(virtual-reality glasses)(도시되지 않음), 홀로그래픽 디스플레이(holographic display) 및 연기 탱크(smoke tank)(도시되지 않음)와 같은 수단을 통해 2차원 시각적 출력 또는 3차원 이상의 출력을 할 수 있음 -), 및 프린터(도시되지 않음)를 포함할 수 있다.

[0271] 컴퓨터 시스템(2000)은 또한 CD/DVD 등의 매체(2021)를 갖는 CD/DVD ROM RW(2020)을 포함한 광학 매체, 썸 드라이브/thumb-drive)(2022), 착탈 가능한 하드 드라이브 또는 솔리드 스테이트 드라이브(2023), 테이프 및 플로피 디스크(도시되지 않음)와 같은 레거시 자기 매체, 보안 동글(security dongle)(도시되지 않음)과 같은 특수한 ROM/ASIC/PLD 기반 기기 등의 인간이 액세스 가능할 수 있는 저장 기기 및 그 연관 매체를 포함할 수도 있다.

[0272] 당업자는 또한 현재 개시된 주제와 관련하여 사용된 바와 같이 컴퓨터로 판독 가능한 매체"라는 용어가 송신 매체, 반송파(carrier wave) 또는 기타 일시적인 신호를 포함하지 않는다는 것을 이해해야 한다.

[0273] 컴퓨터 시스템(2000)은 또한 하나 이상의 통신 네트워크(2055)에 대한 인터페이스(2054)를 포함할 수 있다. 네트워크는 예를 들어 무선, 유선, 광 등의 네트워크일 수 있다. 네트워크는 또한 로컬, 광역, 대도시, 차량 및 산업, 실시간, 지연 허용 등의 네트워크일 수 있다. 네트워크의 예로는 이더넷, 무선 LAN, GSM, 3G, 4G, 5G, LTE, 등을 포함하는 셀룰러 네트워크, 케이블 TV, 위성 TV, 및 지상파 방송 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크, CANBus를 포함하는 차량 및 산업용, 등을 포함한다. 특정 네트워크는 일반적으로 특정 범용 데이터 포트 또는 주변 버스(2049)(예: 컴퓨터 시스템(2000)의 USB 포트)에 부착된 외부 네트워크 인터페이스 어댑터를 필요로 하며; 다른 것은 일반적으로 이하에 설명하는 바와 같이 시스템 버스에 부착함으로써 컴퓨터 시스템(2000)의 코어에 통합된다(예: PC 컴퓨터 시스템에의 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템에의 셀룰러 네트워크 인터페이스). 이러한 네트워크 중 임의의 것을 사용하여, 컴퓨터 시스템(2000)은 다른 네트워크와 통신할 수 있다. 이러한 통신은 단방향, 수신 전용(예: TV 방송), 단방향 전송 전용(예: CANbus에서 특정 CANbus 기기로) 또는 양방향(예: 로컬 또는 광역 디지털 네트워크를 사용하여 다른 컴퓨터 시스템으로)일 수 있다. 특정 프로토콜 및 프로토콜 스택이 전송한 바와 같은 네트워크 및 네트워크 인터페이스 각각에 사용될 수 있다.

[0274] 전송한 휴먼 인터페이스 기기, 인간이 액세스 가능한 저장 기기 및 네트워크 인터페이스는 컴퓨터 시스템(2000)의 코어(2040)에 부착될 수 있다.

[0275] 코어(2040)는 하나 이상의 중앙 처리 유닛(CPU)(2041), 그래픽 처리 유닛(GPU)(2042), 필드 프로그래머블 게이트 어레이(Field Programmable Gate Area, FPGA)(2043) 형태의 특화된 프로그램 가능한 처리 유닛, 특정 태스크를 위한 하드웨어 가속기(2044), 그래픽 어댑터(2050) 등을 포함할 수 있다. 판독 전용 메모리(Read-only memory, ROM)(2045), 랜덤 액세스 메모리(2046), 사용자가 액세스할 수 없는 내부 하드 드라이브, SSD 등의 내부 대용량 저장장치(2047)와 함께, 이러한 기기는 시스템 버스(2048)을 통해 연결될 수 있다. 일부 컴퓨터 시스템에서, 시스템 버스(2048)는 추가적인 CPU, GPU 등에 의한 확장을 가능하게 하는 하나 이상의 물리 플러그의 형태로 액세스 가능할 수 있다. 주변 기기는 코어의 시스템 버스(2048)에 직접 연결되거나 주변 버스(2049)를 통해 연결될 수 있다. 일례에서, 스크린(2010)이 그래픽 어댑터(2050)에 연결될 수 있다. 주변 버스를 위한 아

키택처로는 PCI, USB 등을 포함한다.

- [0276] CPU(2041), GPU(2042), FPGA(2043) 및 가속기(2044)는, 조합하여, 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령어를 실행할 수 있다. 그 컴퓨터 코드는 ROM(2045) 또는 RAM(2046)에 저장될 수 있다. 이행 데이터 (transitional data)는 RAM(2046)에도 저장될 수 있는 반면, 영구 데이터는 예를 들어, 내부 대용량 저장장치 (2047)에 저장될 수 있다. 메모리 소자 중 어느 것에 대한 빠른 저장 및 검색은, 하나 이상의 CPU(2041), GPU(2042), 대용량 저장장치(2047), ROM(2045), RAM(2046) 등과 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 가능해질 수 있다.
- [0277] 컴퓨터로 판독 가능한 매체는 다양한 컴퓨터로 구현되는 동작(computer-implemented operation)을 수행하기 위한 컴퓨터 코드를 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시의 목적을 위해 특별히 설계되고 구축된 것일 수 있거나, 컴퓨터 소프트웨어 분야의 당업자에게 잘 알려져 있고 이용 가능한 종류일 수 있다.
- [0278] 한정이 아닌 예로서, 아키텍처(2000), 구체적으로 코어(2040)를 갖는 컴퓨터 시스템은 하나 이상의 유형의 컴퓨터로 판독 가능한 매체에 구현된 소프트웨어 실행하는 프로세서(들)(CPU, GPU, FPGA, 가속기 등을 포함)의 결과로서 기능을 제공할 수 있다. 이러한 컴퓨터로 판독 가능한 매체는 위에서 소개한 바와 같이 사용자가 액세스할 수 있는 대용량 저장장치와 연관된 매체일 수 있을 뿐만 아니라 코어 내부 대용량 저장장치(2047) 또는 ROM(2045)과 같은, 비일시적인 성질의 코어(2040)의 특정 저장장치일 수 있다. 본 개시의 다양한 실시예를 구현하는 소프트웨어는 이러한 기기에 저장되고 코어(2040)에 의해 실행될 수 있다. 컴퓨터로 판독 가능한 매체는 구체적인 필요에 따라, 하나 이상의 메모리 소자 또는 칩을 포함할 수 있다. 소프트웨어는 코어(2040) 및 특히 내부의 프로세서(CPU, GPU, FPGA 등 포함)로 하여금 RAM(2046)에 저장된 데이터 구조를 정의하고 소프트웨어에 의해 정의된 프로세스에 따라 이러한 데이터 구조를 수정하는 것을 포함하여, 여기에 설명된 특정 프로세스 또는 특정 프로세스의 특정 부분을 실행하도록 할 수 있다. 추가로 또는 대안으로서, 컴퓨터 시스템은 여기에 설명된 특정 프로세스 또는 특정 프로세스의 특정 부분을 실행하기 위해 소프트웨어 대신 또는 소프트웨어와 함께 작동할 수 있는, 논리 배선(logic hardwired)의 결과로서 그렇지 않으면 회로(예: 가속기(2044))에 다른 방식으로 구현되는 기능을 제공할 수 있다. 소프트웨어에 대한 언급은 논리를 포함할 수 있으며, 적절한 경우 그 반대도 마찬가지이다. 컴퓨터로 판독 가능한 매체에 대한 언급은 실행을 위한 소프트웨어를 저장하는 회로(예: 집적 회로(IC)), 실행을 위한 논리를 구현하는 회로, 또는 적절한 경우 둘 다를 포함할 수 있다. 본 개시는 하드웨어와 소프트웨어의 임의의 적절한 조합을 포함한다.
- [0279] [부록 A: 약어]
- [0280] JEM: joint exploration model
- [0281] VVC: versatile video coding
- [0282] BMS: benchmark set
- [0283] MV: Motion Vector
- [0284] HEVC: High Efficiency Video Coding
- [0285] SEI: Supplementary Enhancement Information
- [0286] VUI: Video Usability Information
- [0287] GOPs: Groups of Pictures
- [0288] TUs: Transform Units,
- [0289] PUs: Prediction Units
- [0290] CTUs: Coding Tree Units
- [0291] CTBs: Coding Tree Blocks
- [0292] PBs: Prediction Blocks
- [0293] HRD: Hypothetical Reference Decoder
- [0294] SNR: Signal Noise Ratio

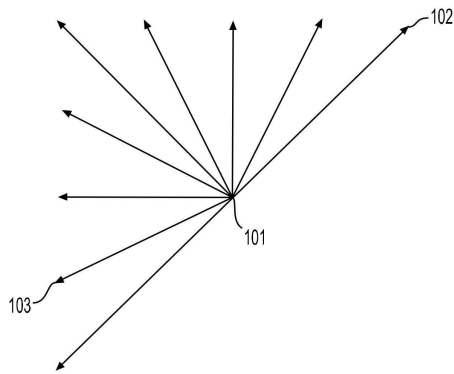
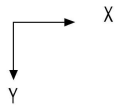
- [0295] CPUs: Central Processing Units
- [0296] GPUs: Graphics Processing Units
- [0297] CRT: Cathode Ray Tube
- [0298] LCD: Liquid-Crystal Display
- [0299] OLED: Organic Light-Emitting Diode
- [0300] CD: Compact Disc
- [0301] DVD: Digital Video Disc
- [0302] ROM: Read-Only Memory
- [0303] RAM: Random Access Memory
- [0304] ASIC: Application-Specific Integrated Circuit
- [0305] PLD: Programmable Logic Device
- [0306] LAN: Local Area Network
- [0307] GSM: Global System for Mobile communications
- [0308] LTE: Long-Term Evolution
- [0309] CANBus: Controller Area Network Bus
- [0310] USB: Universal Serial Bus
- [0311] PCI: Peripheral Component Interconnect
- [0312] FPGA: Field Programmable Gate Areas
- [0313] SSD: solid-state drive
- [0314] IC: Integrated Circuit
- [0315] CU: Coding Unit
- [0316] 본 개시는 몇몇 예시적인 실시예를 설명했지만, 본 개시의 범위 내에 속하는 변경, 순열 및 다양한 대체 등가물이 있다. 따라서, 당업자는 본 명세서에 명시적으로 도시되지 않거나 설명되지 않았지만, 본 개시의 원리를 구현하고 따라서 본 개시의 사상 및 범위 내에 있는 수많은 시스템 및 방법을 고안할 수 있음을 이해할 것이다.

도면

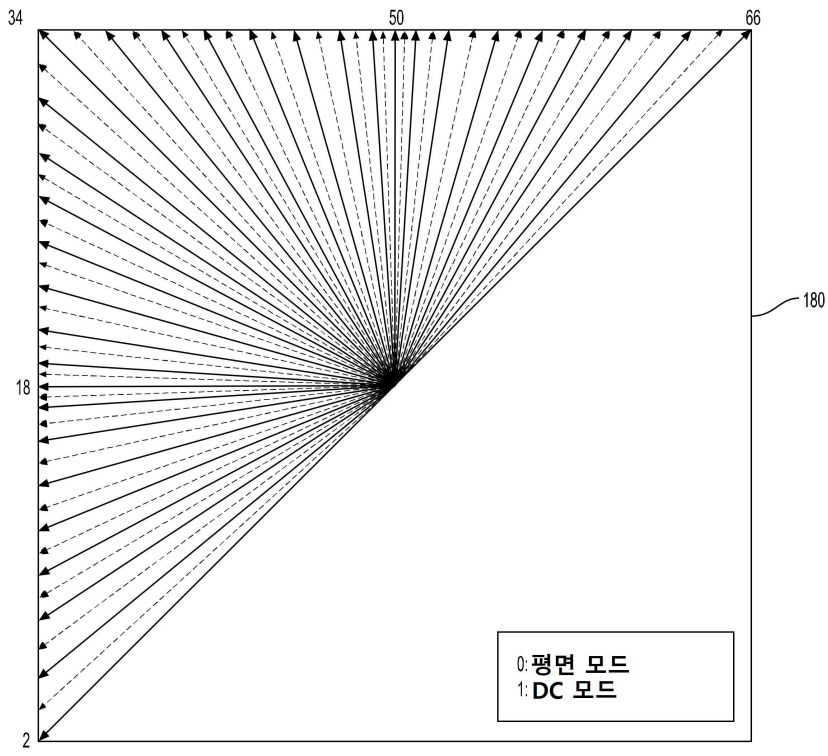
도면1a

	R01	R02	R03	R04	R05	R06	R07	R08	R09
R10	S11	S12	S13	S14					
R20	S21	S22	S23	S24					
R30	S31	S32	S33	S34					
R40	S41	S42	S43	S44					
R50									
R60									
R70									

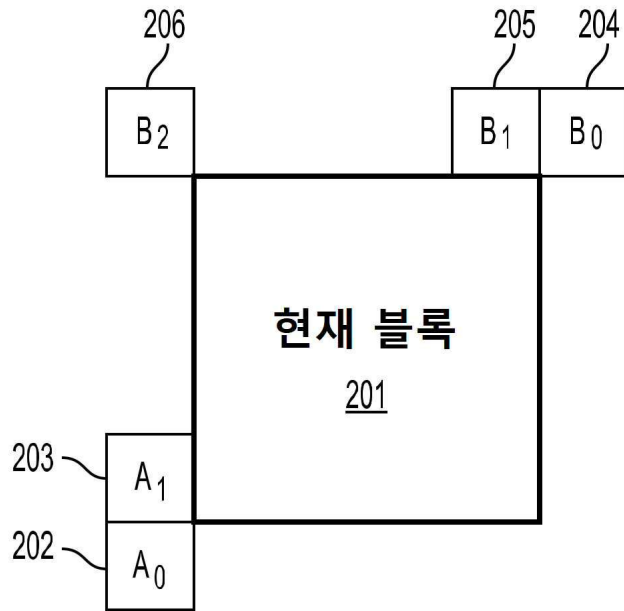
104



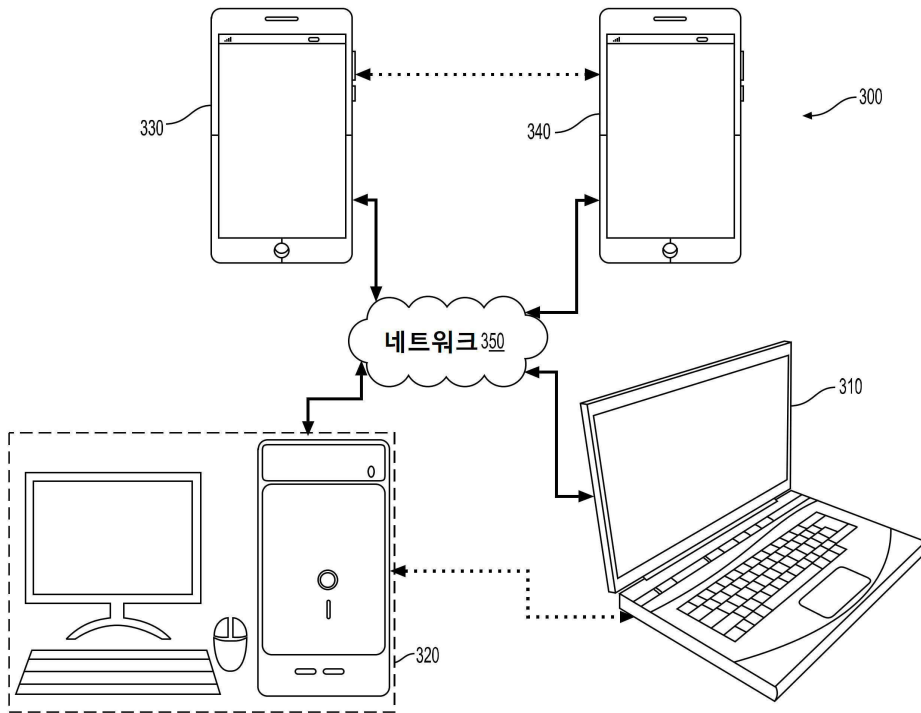
도면1b



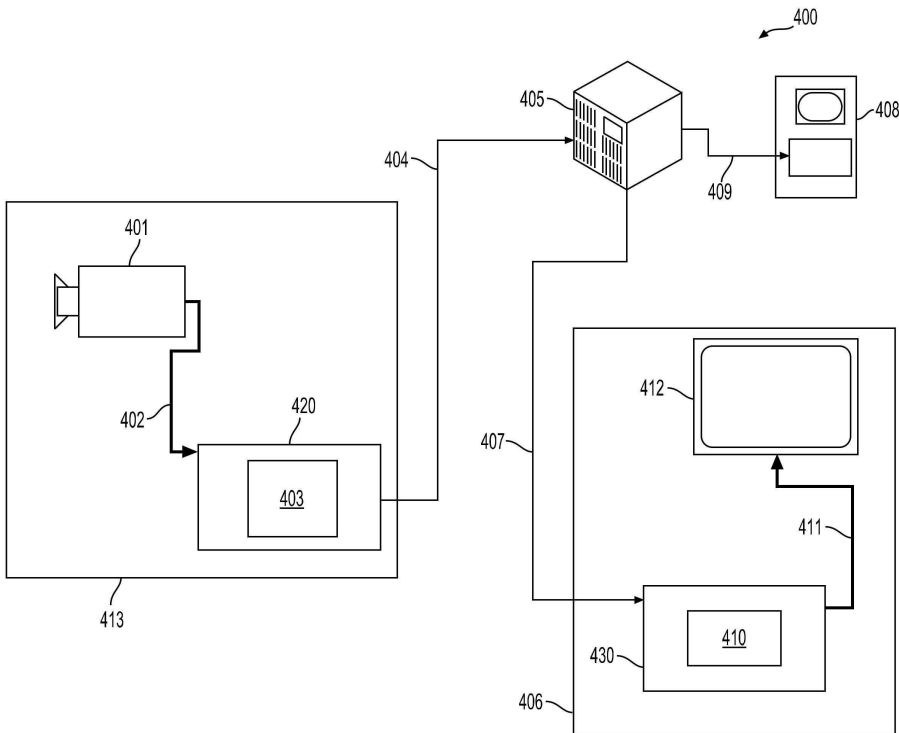
도면2



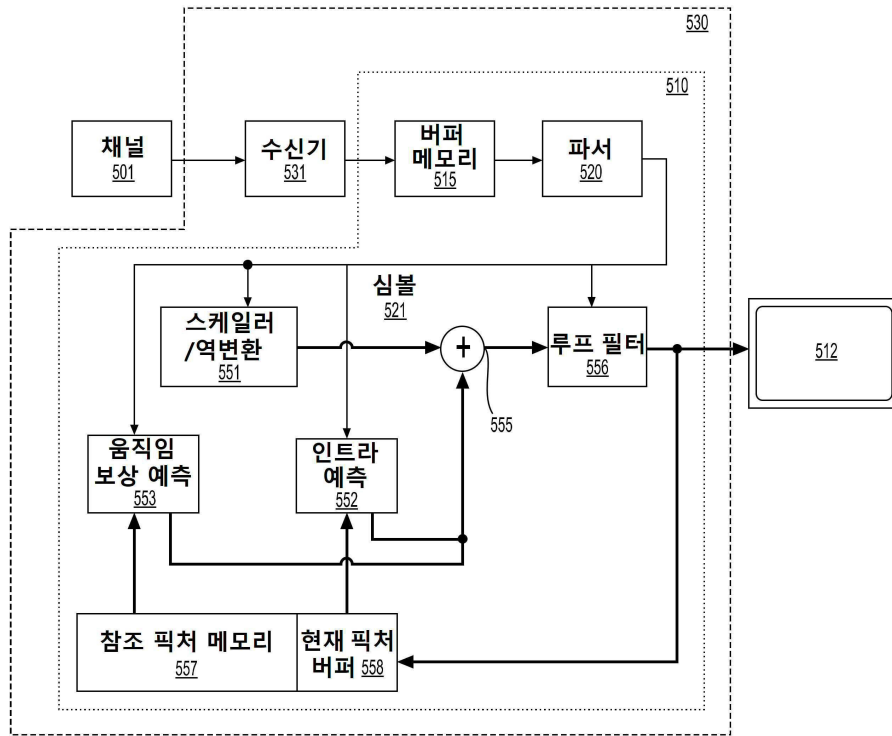
도면3



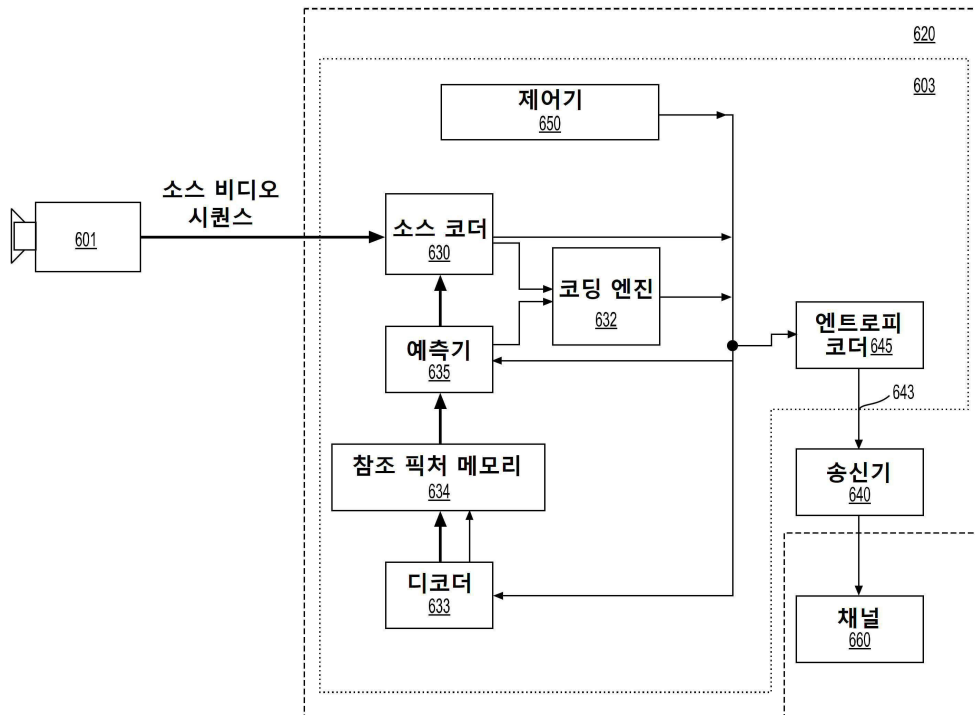
도면4



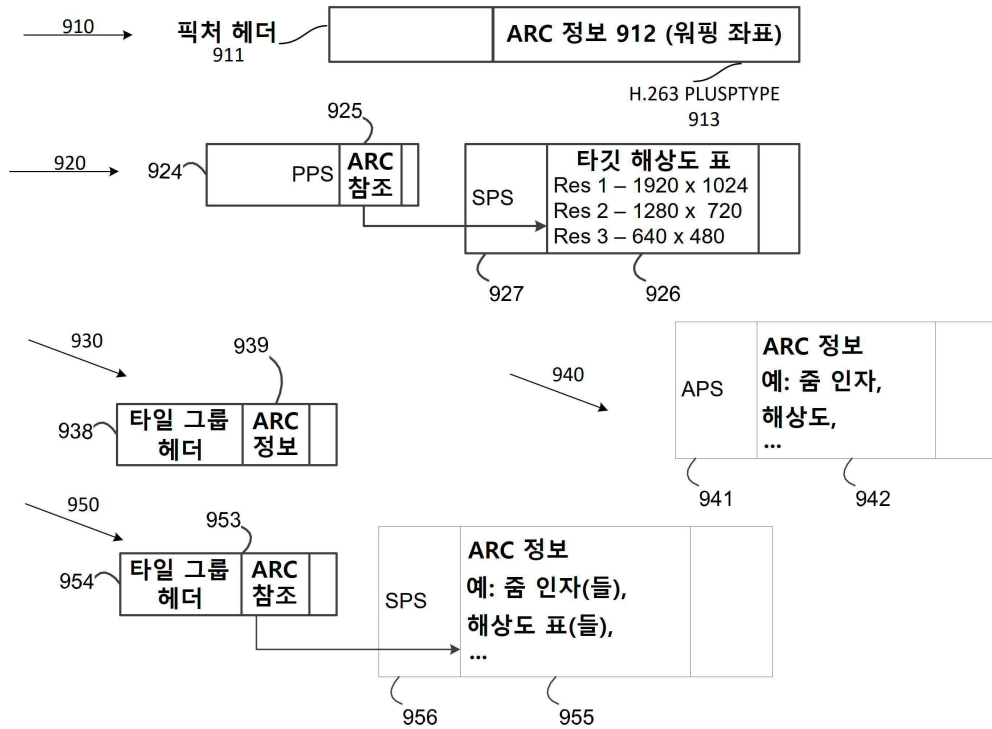
도면5



도면6



도면9



도면10

1000

표 1: 업샘플 또는 다운샘플 인자, 코드워드 및 Ext-Golomb 코드의 매핑

코드워드	Ext-Golomb 코드	원래/타겟 해상도
0	1	1/1
1	010	1/1.5(50% 업스케일)
2	011	1.5/1(50% 다운스케일)
3	00100	1/2(100% 업스케일)
4	00101	2/1(100% 다운스케일)

도면11

1101	tile_group_header() {	
	...	
1103	if(adaptive_pic_resolution_change_flag) {	
1102	dec_pic_size_idx	u(1)
	}	
	...	
	}	

1110	seq_parameter_set_rbsp() {	Descriptor
	...	
1111	adaptive_pic_resolution_change_flag	u(1)
1112	if(adaptive_pic_resolution_change_flag) {	
1113 [output_pic_width_in_luma_samples	ue(v)
	output_pic_height_in_luma_samples	ue(v)
1114	reference_pic_size_present_flag	u(1)
	if(reference_pic_size_present_flag)	
	{	
1115 [reference_pic_width_in_luma_samples	ue(v)
	reference_pic_height_in_luma_samples	ue(v)
	}	
1116	num_dec_pic_size_in_luma_samples_minus1	ue(v)
	for(i = 0; i <=	
	num_dec_pic_size_in_luma_samples_minus1;	
	i++) {	
1117 [dec_pic_width_in_luma_samples[i]	ue(v)
	dec_pic_height_in_luma_samples[i]	ue(v)
	}	
	}	
	}	
	...	
	}	

도면12

1200

	Descriptor
profile_tier_level(profileTierPresentFlag, maxNumSubLayersMinus1) {	
if(profileTierPresentFlag) {	
general_profile_idc	u(7)
general_tier_flag	u(1)
general_constraint_info()	
}	
general_level_idc	u(8)
if(profileTierPresentFlag) {	
num_sub_profiles	u(8)
for(i=0; i<num_sub_profiles; i++)	
general_sub_profile_idc[i]	u(32)
}	
for(i=0; i<maxNumSubLayersMinus1; i++)	
sublayer_level_present_flag[i]	u(1)
while(!byte_aligned())	
ptl_alignment_zero_bit	f(1)
for(i=0; i<maxNumSubLayersMinus1; i++)	
if(sublayer_level_present_flag[i])	
sublayer_level_idc[i]	u(8)
}	

도면13

1300

	Descriptor
general_constraint_info() {	
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
general_one_picture_only_constraint_flag	u(1)
no_lossless_coding_tool_constraint_flag	u(1)
no_lossy_coding_tool_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
single_layer_constraint_flag	u(1)
all_layers_independent_constraint_flag	u(1)
no_ref_pic_resampling_constraint_flag	u(1)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
...	
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)
no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
gci_num_reserved_bytes	u(8)
for(i=0; i<gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

도면14a

1410

	Descriptor
profile tier level(profileTierPresentFlag, maxNumSubLayersMinus1) {	
if(profileTierPresentFlag) {	
general profile idc	u(7)
general tier flag	u(1)
num available constraint flags	u(8)
general constraint info(num available constraint flags)	
}	
general level idc	u(8)
if(profileTierPresentFlag) {	
num sub profiles	u(8)
for(i = 0; i < num sub profiles; i++)	
general sub profile idc[i]	u(32)
}	
for(i = 0; i < maxNumSubLayersMinus1; i++)	
sublayer level present flag[i]	u(1)
while(!byte aligned())	
ptl alignment zero bit	f(1)
for(i = 0; i < maxNumSubLayersMinus1; i++)	
if(sublayer level present flag[i])	
sublayer level idc[i]	u(8)
}	
}	

1401

도면14b

1420

	Descriptor
general constraint info(num available constraint flags) {	
count = 0	
if(count < num available constraint flags)	
general non packed constraint flag	u(1)
count++	
if(count < num available constraint flags)	
general frame only constraint flag	u(1)
count++	
if(count < num available constraint flags)	
general non projected constraint flag	u(1)
count++	
...	
while(!byte aligned())	
gci alignment zero bit	f(1)
gci num reserved bytes	u(8)
for(i = 0; i < gci num reserved bytes; i++)	
gci reserved byte[i]	u(8)
}	

1402

도면15a

1500

	general_constraint_info() {	Descriptor
1501	general_frame_structure_constraint_group_flag	u(1)
1502	high_level_functionality_constraint_group_flag	u(1)
1503	scalability_constraint_group_flag	u(1)
1504	partitioning_constraint_group_flag	u(1)
1505	intra_coding_tool_constraint_group_flag	u(1)
1506	inter_coding_tool_constraint_group_flag	u(1)
1507	transform_constraint_group_flag	u(1)
1508	inloop_filtering_constraint_group_flag	u(1)
	...	
	if(general_frame_structure_constraint_group_flag) {	
1511	general_non_packed_constraint_flag	u(1)
1512	general_frame_only_constraint_flag	u(1)
1513	general_non_projected_constraint_flag	u(1)
1514	general_one_picture_only_constraint_flag	u(1)
	}	
	}	

1510

도면15b

도 15a에 계속

1500

	if(high_level_functionality_constraint_group_flag) {	
	pic_header_in_slice_header_constraint_flag	u(1)
	no_radl_constraint_flag	u(1)
	no_idr_constraint_flag	u(1)
	no_cra_constraint_flag	u(1)
	no_gdr_constraint_flag	u(1)
	no_aps_constraint_flag	u(1)
	...	
	}	
	if(intra_coding_tool_constraint_group_flag) {	
	...	
	}	
	if(inter_coding_tool_constraint_group_flag) {	
	...	
	}	
	while(!byte_aligned())	
	gci_alignment_zero_bit	f(1)
	gci_num_reserved_bytes	u(8)
	for(i = 0; i < gci_num_reserved_bytes; i++)	
	gci_reserved_byte[i]	u(8)
	}	

1430

도면16

1600

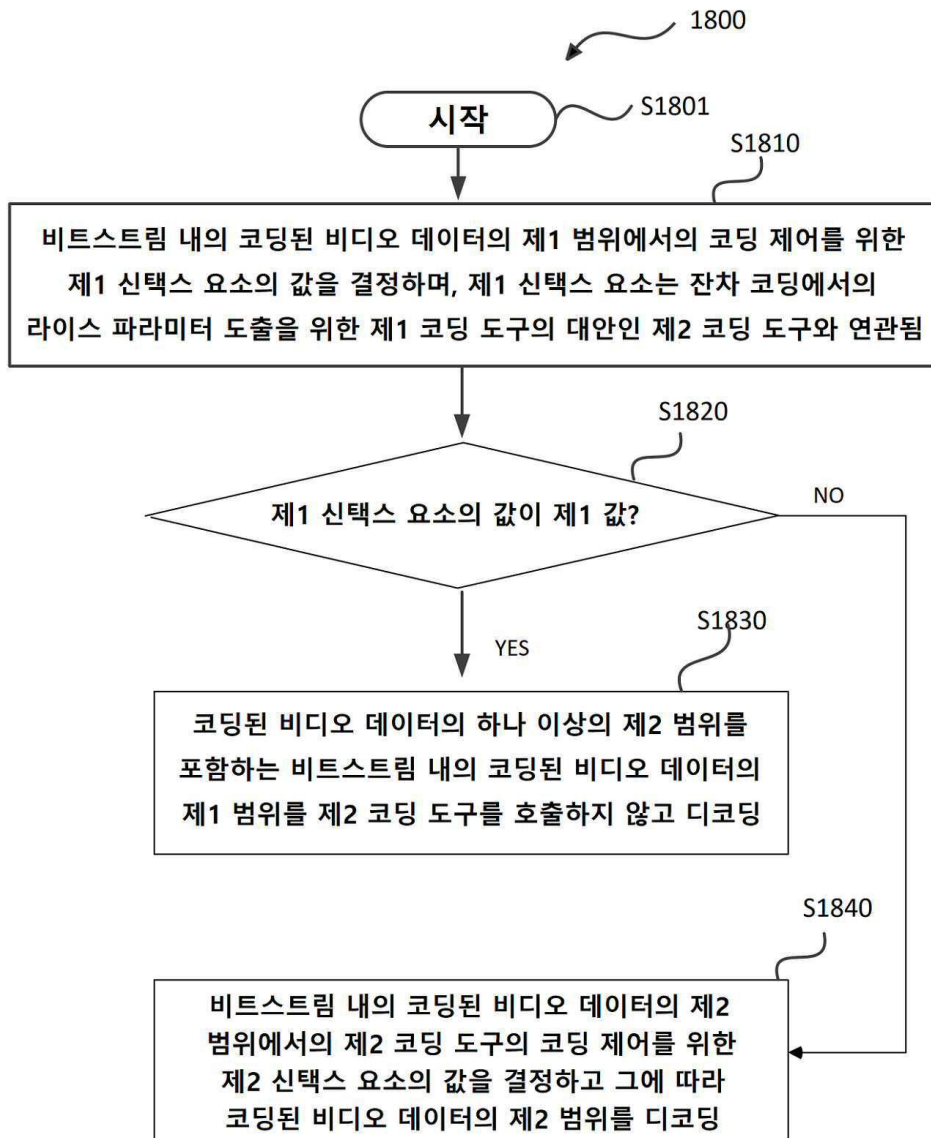
		Descriptor
	general_constraint_info () {	
	...	u(1)
	gci_num_additional_bits	u(8)
	if(gci_num_additional_bits>0) {	
1601	→ general_no_extended_precision_constraint_flag	u(1)
1602	→ general_no_ts_residual_coding_rice_present_in_sh_constraint_flag	u(1)
1603	→ general_no_rrc_rice_extension_constraint_flag	u(1)
1604	→ general_no_persistent_rice_adaptation_constraint_flag	u(1)
1605	→ general_no_reserve_last_sig_coeff_constraint_flag	u(1)
	numAdditionalBitsUsed=5	
	} else	
	numAdditionalBitsUsed=0	
	for(i=0; i<gci_num_additional_bits-numAdditionalBitsUsed;i++)	
	gci_reserved_zero_bit[i]	u(1)
	}	

도면17

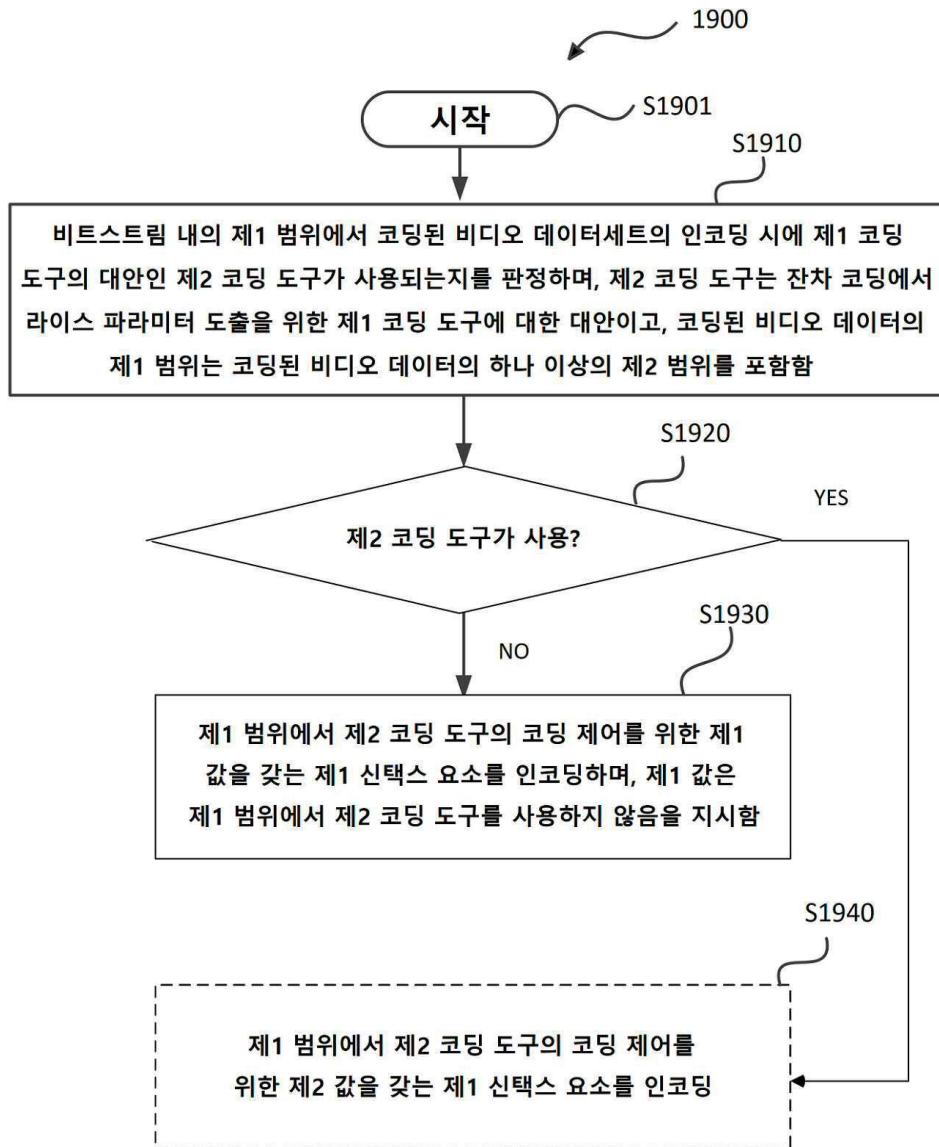
1700

		Descriptor
	sps_range_extension () {	
1701	→ sps_extended_precision_flag	u(1)
	if(sps_transform_skip_enabled_flag)	
1702	→ sps_ts_residual_coding_rice_present_in_sh_flag	u(1)
1703	→ sps_rrc_rice_extension_flag	u(1)
1704	→ sps_persistent_rice_adaptation_enabled_flag	u(1)
1705	→ sps_reverse_last_sig_coeff_enabled_flag	u(1)
	}	

도면18



도면19



도면20

