(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0100770
(43) 공개일자 2025년07월03일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) *H04N 19/105* (2014.01)
H04N 19/119 (2014.01) *H04N 19/189* (2014.01)
H04N 19/577 (2014.01) *H04N 19/593* (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2025-7020320(분할)
(22) 출원일자(국제) 2019년05월28일
심사청구일자 없음
(62) 원출원 특허 10-2020-7035041
원출원일자(국제) 2019년05월28일
심사청구일자 2022년05월19일
(85) 번역문제출일자 2025년06월18일
(86) 국제출원번호 PCT/US2019/034129
(87) 국제공개번호 WO 2019/236335
국제공개일자 2019년12월12일
(30) 우선권주장
18305693.6 2018년06월07일
유럽특허청(EPO)(EP)
18305849.4 2018년07월02일
유럽특허청(EPO)(EP)

(71) 출원인
인터디지털 브이씨 홀딩스 인코포레이티드
미국 19809 델라웨어주 윌밍턴 스위트 300 벨뷰 파크웨이 200
(72) 발명자
레레아넥, 파브리스
프랑스 35576 쎄쏭 쎄비네 자크 데 샹 블랑 아브뉘 데 샹 블랑 975 떼끄니꼴로르 에르 에 데 프랑스 내
갈팽, 프랑크
프랑스 35576 쎄쏭 쎄비네 자크 데 샹 블랑 아브뉘 데 샹 블랑 975 떼끄니꼴로르 에르 에 데 프랑스 내
(뒷면에 계속)
(74) 대리인
양영준, 김연송, 백만기

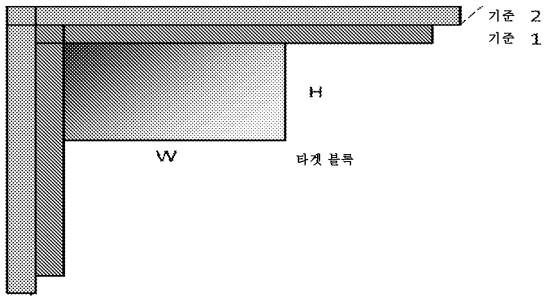
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 **비디오 인코딩 또는 디코딩을 위한 신택스 요소들**

(57) 요 약

적어도 하나의 실시예에서, 새로운 코딩 도구들을 사용함으로써 코딩 효율이 향상된다. 예시적인 실시예에서, 이러한 코딩 도구들의 효율적인 시그널링은 인코딩에 이용되는 코딩 도구들을 나타내는 정보를, 예를 들어, 인코더 디바이스로부터 수신기 디바이스(예를 들어, 디코더 또는 디스플레이)로 전달하여, 적절한 도구들이 디코딩 스테이지에 사용되도록 한다. 이러한 도구들은 새로운 파티셔닝 모드들, 새로운 인트라 예측 모드들, 샘플 적응적 오프셋에 대한 개선된 유연성 및 양방향 예측 블록들에 대한 새로운 조명 보상 모드를 포함한다.

대 표 도 - 도7



(52) CPC특허분류
H04N 19/119 (2015.01)
H04N 19/189 (2015.01)
H04N 19/577 (2015.01)
H04N 19/593 (2015.01)
(72) 발명자
프와레, 탕이
프랑스 35576 쎄쏭 쎄비네 자크 데 샹 블랑 아브뉘 데 샹 블랑 975 떼끄니꼴로르 에르 에 데 프랑스 내
프랑수아, 에두아르
프랑스 35576 쎄쏭 쎄비네 자크 데 샹 블랑 아브뉘 데 샹 블랑 975 떼끄니꼴로르 에르 에 데 프랑스 내

명 세 서

청구범위

청구항 1

비디오의 픽처 데이터를 디코딩하기 위한 방법으로서,

상기 방법은, 상기 비디오의 블록에 대해,

인코딩된 픽처 데이터를 획득하는 단계;

복수의 기준 라인(reference line) - 기준 라인은 상기 블록의 좌측에 위치한 좌측 기준 라인 및 상기 블록 위에 위치한 상부 기준 라인을 포함함 - 에 기초하여 인트라(intra) 예측의 사용을 나타내는 플래그를 획득하는 단계,

상기 플래그의 값에 기초하여 상기 복수의 기준 라인 중에서 선택된 하나 이상의 기준 라인을 사용하여 상기 블록을 예측하는 단계; 및

상기 예측된 블록에 기초하여 픽처 데이터를 디코딩하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

제1 기준 라인은 상기 블록을 둘러싸는 제1 라인 및 제1 열을 포함하고, 상기 제1 라인은 상기 블록 위에 위치하고, 제2 열은 상기 블록의 좌측에 위치하고, 제2 기준 라인은 상기 블록을 둘러싸는 제2 라인 및 제2 열을 포함하고, 상기 제2 라인은 상기 제1 라인 위에 위치하고, 상기 제2 열은 상기 제1 열의 좌측에 위치하는, 방법.

청구항 3

제2항에 있어서,

추가 기준 라인들은 이전 라인들 위에 위치한 추가 라인들 및 이전 열들의 좌측에 위치한 추가 열들을 포함하는, 방법.

청구항 4

제2항에 있어서,

상기 예측은 상기 제1 및 제2 기준 라인들로부터 가중 평균으로서 수행되는, 방법.

청구항 5

제4항에 있어서,

가장 가까운 기준 라인으로부터의 상기 예측은 가장 먼 기준 라인으로부터의 예측보다 높은 가중치를 부여받는, 방법.

청구항 6

제4항에 있어서,

사용된 가중치들은 3과 1인, 방법.

청구항 7

비디오의 픽처 데이터를 디코딩하기 위한 디바이스로서,

상기 디바이스는 프로세서를 포함하고,

상기 프로세서는, 상기 비디오의 블록에 대해,

인코딩된 픽처 데이터를 획득하고;

복수의 기준 라인(reference line) - 기준 라인은 상기 블록의 좌측에 위치한 좌측 기준 라인 및 상기 블록 위에 위치한 상부 기준 라인을 포함함 - 에 기초하여 인트라(intra) 예측의 사용을 나타내는 플래그를 획득하고,

상기 플래그의 값에 기초하여 상기 복수의 기준 라인 중에서 선택된 하나 이상의 기준 라인을 사용하여 상기 블록을 예측하고; 그리고

상기 예측된 블록에 기초하여 픽처 데이터를 디코딩하도록 구성되는, 디바이스.

청구항 8

제7항에 있어서,

제1 기준 라인은 상기 블록을 둘러싸는 제1 라인 및 제1 열을 포함하고, 상기 제1 라인은 상기 블록 위에 위치하고, 제2 열은 상기 블록의 좌측에 위치하고, 제2 기준 라인은 상기 블록을 둘러싸는 제2 라인 및 제2 열을 포함하고, 상기 제2 라인은 상기 제1 라인 위에 위치하고, 상기 제2 열은 상기 제1 열의 좌측에 위치하는, 디바이스.

청구항 9

제8항에 있어서,

추가 기준 라인들은 이전 라인들 위에 위치한 추가 라인들 및 이전 열들의 좌측에 위치한 추가 열들을 포함하는, 디바이스.

청구항 10

제8항에 있어서,

상기 예측은 상기 제1 및 제2 기준 라인들로부터 가중 평균으로서 수행되는, 디바이스.

청구항 11

제10항에 있어서,

가장 가까운 기준 라인으로부터의 상기 예측은 가장 먼 기준 라인으로부터의 예측보다 높은 가중치를 부여받는, 디바이스.

청구항 12

제10항에 있어서,

사용된 가중치들은 3과 1인, 디바이스.

청구항 13

비일시적 컴퓨터 판독가능 매체로서,

제1항에 따른 방법의 단계들을 구현하기 위해 프로세서에 의해 실행가능한 프로그램 코드 명령어들을 저장하는 비일시적 컴퓨터 판독가능 매체.

발명의 설명

기 술 분 야

[0001] 본 실시예들 중 적어도 하나는 일반적으로 비디오 인코딩 또는 디코딩을 위한 신택스 요소들에 관한 것이다.

배 경 기 술

[0002] 높은 압축 효율을 달성하기 위해, 이미지 및 비디오 코딩 스킴들은 비디오 콘텐츠에서의 공간적 및 시간적 리던

던시(spatial and temporal redundancy)를 레버리지(leverage)하기 위해 보통 예측 및 변환을 이용한다. 일반적으로, 인트라(intra) 또는 인터(inter) 예측이 인트라 또는 인터 프레임 상관을 이용하기 위해 사용되고, 이어서, 종종 예측 에러들 또는 예측 잔차들로서 표시되는, 원래의 블록과 예측된 블록 간의 차이들이 변환되고, 양자화되며, 엔트로피 코딩된다. 비디오를 재구성하기 위해, 압축된 데이터는 엔트로피 코딩, 양자화, 변환, 및 예측에 대응하는 역 프로세스들에 의해 디코딩된다.

발명의 내용

과제의 해결 수단

[0003] 적어도 하나의 실시예의 제1 양태에 따르면, 비디오 신호가 제시되며, 이 비디오 신호는 비디오 코딩 표준에 따른 정보를 포함하도록 포맷되고, 비디오 콘텐츠 및 하이 레벨 신택스 정보를 갖는 비트스트림을 포함하고, 하이 레벨 신택스 정보는 파티셔닝의 타입을 나타내는 파라미터, 인트라 예측 모드의 타입을 나타내는 파라미터, 샘플 적응적 오프셋 루프 필터에 대한 블록 크기 적응을 나타내는 파라미터, 및 양방향 예측 블록들에 대한 조명 보상 모드를 나타내는 파라미터 중 적어도 하나의 파라미터를 포함한다.

[0004] 적어도 하나의 실시예의 제2 양태에 따르면, 저장 매체가 제시되며, 이 저장 매체는 인코딩된 비디오 신호 데이터를 가지고, 비디오 신호는 비디오 코딩 표준에 따른 정보를 포함하도록 포맷되고, 비디오 콘텐츠 및 하이-레벨 신택스 정보를 갖는 비트스트림을 포함하고, 하이 레벨 신택스 정보는 파티셔닝의 타입을 나타내는 파라미터, 인트라 예측 모드의 타입을 나타내는 파라미터, 샘플 적응적 오프셋 루프 필터에 대한 블록 크기 적응을 나타내는 파라미터, 및 양방향 예측 블록들에 대한 조명 보상 모드를 나타내는 파라미터 중 적어도 하나의 파라미터를 포함한다.

[0005] 적어도 하나의 실시예의 제3 양태에 따르면, 장치가 제시되며, 이 장치는 픽처 내의 적어도 하나의 블록에 대한 픽처 데이터를 인코딩하기 위한 비디오 인코더를 포함하고, 인코딩은 파티셔닝의 타입을 나타내는 파라미터, 인트라 예측 모드의 타입을 나타내는 파라미터, 샘플 적응적 오프셋 루프 필터에 대한 블록 크기 적응을 나타내는 파라미터, 및 양방향 예측 블록들에 대한 조명 보상 모드를 나타내는 파라미터 중 적어도 하나의 파라미터를 사용하여 수행되고, 파라미터들은 인코딩된 픽처 데이터의 하이 레벨 신택스 요소들에 삽입된다.

[0006] 적어도 하나의 실시예의 제4 양태에 따르면, 방법이 제시되며, 이 방법은 픽처 내의 적어도 하나의 블록에 대한 픽처 데이터를 인코딩하는 단계- 인코딩은 파티셔닝의 타입을 나타내는 파라미터, 인트라 예측 모드의 타입을 나타내는 파라미터, 샘플 적응적 오프셋 루프 필터에 대한 블록 크기 적응을 나타내는 파라미터, 및 양방향 예측 블록들에 대한 조명 보상 모드를 나타내는 파라미터 중 적어도 하나의 파라미터를 사용하여 수행됨 -, 및 파라미터들을 인코딩된 픽처 데이터의 하이 레벨 신택스 요소들에 삽입하는 단계를 포함한다.

[0007] 적어도 하나의 실시예의 제5 양태에 따르면, 장치가 제시되며, 이 장치는 픽처 내의 적어도 하나의 블록에 대한 픽처 데이터를 디코딩하기 위한 비디오 디코더를 포함하고, 디코딩은 파티셔닝의 타입을 나타내는 파라미터, 인트라 예측 모드의 타입을 나타내는 파라미터, 샘플 적응적 오프셋 루프 필터에 대한 블록 크기 적응을 나타내는 파라미터, 및 양방향 예측 블록들에 대한 조명 보상 모드를 나타내는 파라미터 중 적어도 하나의 파라미터를 사용하여 수행되고, 파라미터들은 인코딩된 픽처 데이터의 하이 레벨 신택스 요소들로부터 획득된다.

[0008] 적어도 하나의 실시예의 제6 양태에 따르면, 방법이 제시되며, 이 방법은 인코딩된 픽처 데이터의 하이 레벨 신택스 요소들로부터 파라미터들을 획득하는 단계, 및 픽처 내의 적어도 하나의 블록에 대한 픽처 데이터를 디코딩하는 단계를 포함하고, 디코딩은: 파티셔닝의 타입을 나타내는 파라미터, 인트라 예측 모드의 타입을 나타내는 파라미터, 샘플 적응적 오프셋 루프 필터에 대한 블록 크기 적응을 나타내는 파라미터, 및 양방향 예측 블록들에 대한 조명 보상 모드를 나타내는 파라미터 중 적어도 하나의 파라미터를 사용하여 수행된다.

[0009] 적어도 하나의 실시예의 제7 양태에 따르면, 비일시적 컴퓨터 판독가능 매체가 제시되며, 이 비일시적 컴퓨터 판독가능 매체는 제3 또는 제4 양태에 따라 생성된 데이터 콘텐츠를 포함한다.

[0010] 적어도 하나의 실시예의 제7 양태에 따르면, 프로세서에 의해 실행가능한 프로그램 코드 명령어들을 포함하는 컴퓨터 프로그램이 제시되며, 이 컴퓨터 프로그램은 적어도 제4 또는 제6 양태에 따른 방법의 단계들을 구현한다.

[0011] 적어도 하나의 실시예의 제8 양태에 따르면, 비일시적 컴퓨터 판독가능 매체 상에 저장되고 프로세서에 의해 실행가능한 프로그램 코드 명령어들을 포함하는 컴퓨터 프로그램 제품이 제시되며, 이 컴퓨터 프로그램 제품은 적

어도 제4 또는 제6 양태에 따른 방법의 단계들을 구현한다.

도면의 간단한 설명

[0012] 도 1은 고효율 비디오 코딩(HEVC) 인코더와 같은 비디오 인코더(100)의 예를 도시한다.

도 2는 HEVC 디코더와 같은 비디오 디코더(200)의 예의 블록도를 도시한다.

도 3은 압축된 도메인에서 코딩 트리 유닛 및 코딩 트리의 예를 도시한다.

도 4는 CTU를 코딩 유닛들, 예측 유닛들 및 변환 유닛들로 분할하는 예를 도시한다.

도 5는 QTBT(Quad-Tree plus Binary-Tree) CTU 표현의 예를 도시한다.

도 6은 예시적인 확장된 코딩 유닛 파티셔닝 세트를 도시한다.

도 7은 2개의 기준 계층들을 갖는 인트라 예측 모드의 예를 도시한다.

도 8은 양방향 조명 보상을 위한 보상 모드의 예시적인 실시예를 도시한다.

도 9는 예시적인 실시예에 따른 bt-split-flag의 해석을 도시한다.

도 10은 다양한 양태들 및 실시예들이 구현되는 시스템의 예의 블록도를 도시한다.

도 11은 새로운 인코딩 도구들을 사용하는 실시예에 따른 인코딩 방법의 예의 흐름도를 도시한다.

도 12는 새로운 인코딩 도구들을 사용하는 실시예에 따른 디코딩 방법의 일부의 예의 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0013] 적어도 하나의 실시예에서, 새로운 코딩 도구들을 사용하면 코딩 효율이 향상된다. 예시적인 실시예에서, 이러한 코딩 도구들의 효율적인 시그널링은 인코딩에 이용되는 코딩 도구들을 나타내는 정보를, 예를 들어, 인코더 장치로부터 수신기 디바이스(예를 들어, 디코더 또는 디스플레이)로 전달하여, 적절한 도구들이 디코딩 스테이지에 사용되도록 한다. 이러한 도구들은 새로운 파티셔닝 모드들, 새로운 인트라 예측 모드들, 샘플 적응적 오프셋에 대한 개선된 유연성 및 양방향 예측 블록들에 대한 새로운 조명 보상 모드를 포함한다. 따라서, 복수의 코딩 도구를 수집하는 제안된 새로운 신택스는 보다 효율적인 비디오 코딩을 제공한다.

[0014] 도 1은 고효율 비디오 코딩(HEVC) 인코더와 같은 비디오 인코더(100)의 예를 도시한다. 도 1은 또한 HEVC 표준에 대해 개선들이 이루어지는 인코더 또는 JVET(Joint Video Exploration Team)에 의해 개발 중인 JEM(Joint Exploration Model) 인코더와 같은, HEVC와 유사한 기술들을 이용하는 인코더를 도시할 수 있다.

[0015] 본 출원에서, 용어들 "재구성된" 및 "디코딩된"은 혼용하여 사용될 수 있고, 용어들 "인코딩된" 또는 "코딩된"은 혼용하여 사용될 수 있으며, 용어들 "이미지", "픽처" 및 "프레임"은 혼용하여 사용될 수 있다. 필수적은 아니지만 통상적으로, 용어 "재구성된"은 인코더 측에서 사용되는 한편, "디코딩된"은 디코더 측에서 사용된다.

[0016] 인코딩되기 전에, 비디오 시퀀스는 프리-인코딩 처리(101)를 거칠 수 있다. 이것은 예를 들어, 입력 컬러 픽처에 컬러 변환(예를 들어, RGB 4:4:4로부터 YCbCr 4:2:0으로의 변환)을 적용하거나 또는 (예를 들어, 컬러 성분들 중 하나의 히스토그램 균등화를 사용하여) 압축에 보다 탄력적인 신호 분포를 얻기 위해 입력 픽처 성분들의 리매핑을 수행함으로써 수행된다. 메타데이터는 전처리와 연관되어 비트스트림에 첨부될 수 있다.

[0017] HEVC에서, 하나 이상의 픽처를 갖는 비디오 시퀀스를 인코딩하기 위해, 픽처는 하나 이상의 슬라이스로 파티셔닝되고(102), 각각의 슬라이스는 하나 이상의 슬라이스 세그먼트를 포함할 수 있다. 슬라이스 세그먼트는 코딩 유닛들(coding units), 예측 유닛들(prediction units) 및 변환 유닛들(transform units)로 조직화된다. HEVC 규격은 "블록들" 과 "유닛들" 사이를 구별하며, 여기서 "블록"은 샘플 어레이의 특정 영역(예를 들어, 루마, Y)을 다루고, "유닛"은 모든 인코딩된 컬러 성분들(Y, Cb, Cr, 또는 단색)의 병치된 블록들, 및 블록들과 연관된 신택스 요소들 및 예측 데이터(예를 들어, 모션 벡터들)를 포함한다.

[0018] HEVC에서 코딩의 경우, 픽처는 구성가능한 크기를 갖는 사각형 형상의 코딩 트리 블록들(CTB)로 파티셔닝되고, 코딩 트리 블록들의 연속적 세트는 슬라이스로 그룹화된다. 코딩 트리 유닛(CTU)은 인코딩된 컬러 성분들의 CTB들을 포함한다. CTB는 코딩 블록들(CB)로의 파티셔닝하는 쿼드트리(quadtree)의 루트이고, 코딩 블록은 하나 이상의 예측 블록들(PB)로 파티셔닝될 수 있고, 변환 블록들(TB들)로 파티셔닝하는 쿼드트리의 루트를 형성한다. 코딩 블록, 예측 블록 및 변환 블록에 대응하여, 코딩 유닛(CU)은 예측 유닛들(PU들) 및 변환 유닛들(TU

들)의 트리-구조 세트를 포함하고, PU는 모든 컬러 성분들에 대한 예측 정보를 포함하며, TU는 각각의 컬러 성분에 대한 잔차 코딩 신택스 구조(residual coding syntax structure)를 포함한다. 루마 성분의 CB, PB 및 TB의 크기는 대응하는 CU, PU 및 TU에 적용된다. 본 출원에서, 용어 "블록"은 예를 들어, CTU, CU, PU, TU, CB, PB, 및 TB 중 임의의 것을 지칭하기 위해 사용될 수 있다. 또한, "블록"은 H.264/AVC 또는 다른 비디오 코딩 표준들에서 특정된 바와 같이 매크로블록 및 파티션을 지칭하기 위해, 그리고 더 일반적으로는 다양한 크기들의 데이터의 어레이를 지칭하기 위해 또한 사용될 수 있다.

[0019] 인코더(100)의 예에서, 픽처는 후술하는 바와 같이 인코더 요소들에 의해 인코딩된다. 인코딩될 픽처는 CU들의 단위들로 처리된다. 각각의 CU는 인트라 또는 인터 모드를 사용하여 인코딩된다. CU가 인트라 모드에서 인코딩될 때, 인트라 예측(160)을 수행한다. 인터 모드에서, 모션 추정(175) 및 보상(170)이 수행된다. 인코더는 CU를 인코딩하기 위해 인트라 모드 또는 인터 모드 중 어느 것을 사용할지를 결정하고(105), 예측 모드 플래그에 의해 인트라/인터 결정을 표시한다. 예측 잔차들은 원래의 이미지 블록으로부터 예측된 블록을 감산(110)함으로써 계산된다.

[0020] 인트라 모드의 CU들은 동일한 슬라이스 내의 재구성된 이웃 샘플들로부터 예측된다. 35개의 인트라 예측 모드들의 세트는 DC, 평면 및 33개의 각도 예측 모드들을 포함하는 HEVC에서 이용가능하다. 인트라 예측 기준은 현재 블록에 인접한 행 및 열로부터 재구성된다. 기준은 이전에 재구성된 블록들로부터 이용가능한 샘플들을 사용하여 수평 및 수직 방향에서 블록 크기 2배에 걸쳐 연장된다. 인트라 예측에 대해 각도 예측 모드가 사용될 때, 기준 샘플들은 각도 예측 모드에 의해 표시된 방향을 따라 카피될 수 있다.

[0021] 현재 블록에 대해 적용가능한 루마 인트라 예측 모드는 2개의 상이한 옵션들을 사용하여 코딩될 수 있다. 적용가능한 모드가 3개의 가장 가능성 있는 모드(most probable modes)(MPM)의 구성된 리스트(constructed list)에 포함되는 경우, 모드는 MPM 리스트에서의 인덱스에 의해 시그널링된다. 그렇지 않으면, 모드는 모드 인덱스의 고정 길이 이진화에 의해 시그널링된다. 3개의 가장 가능성있는 모드는 상단 및 좌측 이웃 블록들의 인트라 예측 모드들로부터 도출된다.

[0022] 인터 CU의 경우, 대응하는 코딩 블록은 하나 이상의 예측 블록들로 추가로 파티셔닝된다. 인트라 예측은 PB 레벨에서 수행되고, 대응하는 PU는 인터 예측이 어떻게 수행되는지에 대한 정보를 포함한다. 모션 정보(예를 들어, 모션 벡터 및 기준 픽처 인덱스)는 2개의 방법, 즉, "병합 모드" 및 "진보된 모션 벡터 예측(AMVP)"에서 시그널링될 수 있다.

[0023] 병합 모드에서, 비디오 인코더 또는 디코더는 이미 코딩된 블록들에 기초하여 후보 리스트를 어셈블하고, 비디오 인코더는 후보 리스트에서 후보들 중 하나에 대한 인덱스를 시그널링한다. 디코더 측에서는, 시그널링된 후보에 기초하여 모션 벡터(MV) 및 기준 픽처 인덱스가 재구성된다.

[0024] AMVP에서, 비디오 인코더 또는 디코더는 이미 코딩된 블록들로부터 결정된 모션 벡터들에 기초하여 후보 리스트들을 어셈블한다. 비디오 인코더는 이어서 모션 벡터 예측자(MVP)를 식별해주기 위해 후보 리스트에서의 인덱스를 시그널링하고, 모션 벡터 차이(MVD)를 시그널링한다. 디코더 측에서, 모션 벡터(MV)는 MVP+MVD로서 재구성된다. 적용가능한 기준 픽처 인덱스는 또한 AMVP에 대한 PU 신택스에서 명시적으로 코딩된다.

[0025] 이어서, 후술하는 색차 양자화 파라미터를 적응시키기 위한 적어도 하나의 실시예를 포함하는 예측 잔차들이 변환(125) 되고 양자화(130)된다. 변환들은 일반적으로 분리가능한 변환들에 기초한다. 예를 들어, DCT 변환이 먼저 수평 방향으로, 이어서 수직 방향으로 적용된다. JEM과 같은 최근의 코덱들에서, 양쪽 방향들에서 사용되는 변환들은 상이할 수 있고(예를 들어, 하나의 방향에서는 DCT, 다른 방향에서는 DST), 이는 매우 다양한 2D 변환들로 이어지는 반면, 이전의 코덱들에서는, 주어진 블록 크기에 대한 다양한 2D 변환들이 보통 제한된다.

[0026] 양자화된 변환 계수들 뿐만 아니라 모션 벡터들 및 다른 신택스 요소들은 비트스트림을 출력하도록 엔트로피 코딩된다(145). 인코더는 또한 변환을 스킵하고, 4x4 TU 기반으로 비-변환된 잔차 신호에 직접 양자화를 적용할 수 있다. 인코더는 또한 변환 및 양자화 양쪽 모두를 우회할 수 있는데, 즉, 잔차는 변환 또는 양자화 프로세스의 적용없이 직접 코딩된다. 다이렉트 PCM 코딩에서는, 예측이 적용되지 않으며, 코딩 유닛 샘플들이 비트스트림으로 직접 코딩된다.

[0027] 인코더는 추가적 예측들을 위한 기준을 제공하기 위해 인코딩된 블록을 디코딩한다. 예측 잔차들을 디코딩하기 위해, 양자화된 변환 계수들은 역양자화되고(140) 역변환된다(150). 디코딩된 예측 잔차들 및 예측된 블록을 결합하면(155), 이미지 블록이 재구성된다. 루프 내 필터들(165)은, 예를 들어, 인코딩 아티팩트들을 감소시키기 위한 디블로킹/SAO(Sample Adaptive Offset) 필터링을 수행하기 위해, 재구성된 픽처에 적용된다. 필터링된

이미지는 기준 픽처 버퍼(180)에 저장된다.

[0028] 도 2는 HEVC 디코더와 같은 비디오 디코더(200)의 예의 블록도를 도시한다. 디코더(200)의 예에서, 비트스트림은 아래에서 설명되는 바와 같이 디코더 요소들에 의해 디코딩된다. 비디오 디코더(200)는 일반적으로 도 1에 설명된 바와 같은 인코딩 패스(pass)에 대해 역인 디코딩 패스를 수행하며, 이는 인코딩 비디오 데이터의 일부로서 비디오 디코딩을 수행한다. 도 2는 또한 HEVC 표준 또는 HEVC와 유사한 기술들을 이용하는 디코더, 예컨대 JEM 디코더에 대해 개선들이 이루어지는 디코더를 도시할 수 있다.

[0029] 특히, 디코더의 입력은 비디오 인코더(100)에 의해 생성될 수 있는 비디오 비트스트림을 포함한다. 비트스트림은 먼저 엔트로피 디코딩(230)되어 변환 계수들, 모션 벡터들, 픽처 파티셔닝 정보, 및 다른 코딩된 정보를 획득한다. 픽처 파티셔닝 정보는 CTU들의 크기, 및 CTU가 CU들로, 그리고 가능하게는 적용 가능한 경우 PU들로 분할되는 방식을 표시한다. 따라서, 디코더는 디코딩된 픽처 파티셔닝 정보에 따라 픽처를 CTU들로, 그리고 각각의 CTU를 CU들로 분할할 수 있다(235). 후술하는 색차 양자화 파라미터를 적응시키기 위한 적어도 하나의 실시예를 포함하는 변환 계수들은 역양자화되고(240), 예측 잔차들을 디코딩하기 위해 역변환된다(250).

[0030] 디코딩된 예측 잔차들 및 예측된 블록을 결합하면(255), 이미지 블록이 재구성된다. 예측된 블록은 인트라 예측(260) 또는 모션-보상된 예측(즉, 인터 예측)(275)으로부터 획득될 수 있다(270). 전술한 바와 같이, AMVP 및 병합 모드 기술들은 모션 보상을 위한 모션 벡터들을 도출하기 위해 사용될 수 있고, 이는 기준 블록의 서브-정수 샘플들에 대한 보간된 값들을 계산하기 위해 보간 필터들을 사용할 수 있다. 루프 내 필터들(265)은 재구성된 이미지에 대해 적용된다. 필터링된 이미지는 기준 픽처 버퍼(280)에 저장된다.

[0031] 디코딩된 픽처는 포스트-디코딩 처리(285), 예를 들어, 역 컬러 변환(예를 들어, YCbCr 4:2:0으로부터 RGB 4:4:4로의 변환) 또는 프리-인코딩 처리(101)에서 수행된 리매핑 프로세스의 역을 수행하는 역 리매핑을 더 거칠 수 있다. 포스트-디코딩 처리는 프리-인코딩 처리에서 도출되고 비트스트림에서 시그널링되는 메타데이터를 사용할 수 있다.

[0032] 도 3은 압축된 도메인에서 코딩 트리 유닛 및 코딩 트리의 예를 도시한다. HEVC 비디오 압축 표준에서, 픽처는 소위 CTU(Coding Tree Unit)들로 분할되며, 그 크기는 전형적으로 64x64, 128x128, 또는 256x256 픽셀들이다. 각각의 CTU는 압축된 도메인에서 코딩 트리에 의해 표현된다. 이는 CTU의 쿼드-트리(quad-tree) 부분이고, 여기서 각각의 리프(leaf)는 코딩 유닛(CU)으로 지칭된다.

[0033] 도 4는 CTU를 코딩 유닛들, 예측 유닛들 및 변환 유닛들로 분할하는 예를 도시한다. 그 다음, 각각의 CU에는 일부 인트라 또는 인터 예측 파라미터들(예측 정보)이 부여된다. 그렇게 하기 위해, 각각의 PU에 일부 예측 정보가 할당되는 하나 이상의 예측 유닛(PU)으로 공간적으로 파티셔닝된다. 인트라 또는 인터 코딩 모드는 CU 레벨에서 할당된다.

[0034] 압축된 도메인에서 코딩 트리 유닛 표현을 포함하는 새롭게 떠오르는 비디오 압축 도구들이 픽처 데이터를 보다 유연한 방식으로 표현하기 위해 제안된다. 코딩 트리의 이러한 보다 유연한 표현의 장점은 HEVC 표준의 CU/PU/TU 배열에 비해 증가된 압축 효율을 제공한다는 점이다.

[0035] 도 5는 QTBT(Quad-Tree plus Binary-Tree) CTU 표현의 예를 도시한다. QTBT(Quad-Tree plus Binary-Tree) 코딩 도구는 이러한 증가된 유연성을 제공한다. QTBT는 코딩 유닛들이 쿼드-트리 및 이진-트리 방식으로 분할될 수 있는 코딩 트리로 구성된다. 코딩 유닛의 분할은 최소 레이트 왜곡 비용으로 CTU의 QTBT 표현을 결정하는 레이트 왜곡 최적화 절차를 통해 인코더 측에서 결정된다. QTBT 기술에서, CU는 정사각형 또는 직사각형 형상을 갖는다. 코딩 유닛의 크기는 항상 2의 거듭 제곱이고, 전형적으로 4 내지 128이다. 코딩 유닛에 대한 이러한 다양한 직사각형 형상들에 추가하여, 이러한 CTU 표현은 HEVC와 비교하여 다음의 상이한 특성들을 갖는다. CTU의 QTBT 분해는 2개의 스테이지로 이루어진다: 먼저 CTU가 쿼드-트리 방식으로 분할되고, 그 후 각각의 쿼드-트리 리프는 이진 방식으로 추가로 분할될 수 있다. 이는 도면의 우측에 도시되어 있으며, 여기서 실선들은 쿼드-트리 분해 단계를 나타내고 파선들은 쿼드-트리 리프들에 공간적으로 내장된 이진 분해를 나타낸다. 인트라 슬라이스들에서, 루마 및 크로마 블록 파티셔닝 구조는 분리되고, 독립적으로 결정된다. 예측 유닛들 또는 변환 유닛으로의 CU 파티셔닝은 더 이상 이용되지 않는다. 즉, 각각의 코딩 유닛은 단일 예측 유닛(2Nx2N 예측 유닛 파티션 타입)과 단일 변환 유닛(변환 트리로의 분할 없음)으로 체계적으로 만들어진다.

[0036] 도 6은 예시적인 확장된 코딩 유닛 파티셔닝 세트를 도시한다. 비대칭 이진 및 트리 분할 모드들(asymmetric binary and tree split modes, ABT)에서, 비대칭 이진 분할 모드들 중 하나, 예를 들어, HOR_UP(horizontal-up)를 통해 분할될 수 있는 크기(w, h)(폭 및 높이)를 갖는 직사각형 코딩 유닛은 각각의 직사각형 크기들

$\left(w, \frac{h}{4}\right)$ 및 $\left(w, \frac{3h}{4}\right)$을 갖는 2개의 서브 코딩 유닛들로 이어질 것이다. 또한, 소위 CU의 트리플 트리 파티셔닝(triple tree partitioning)이 사용될 수 있고, 이는 도 5에 주어진 가능한 파티션들의 세트로 이어진다. 트리플 트리는 CU를, 고려되는 배향(예를 들어, 수평 분할 모드에 대한 HOR_TRIPLE)에서, 부모 CU에 관한 크기(1/4, 1/2, 1/4)를 갖는 트리 서브-CU로 분할하는 것으로 구성된다.

[0037] 전술한 새로운 토폴로지들의 하나 이상의 실시예를 사용하여 상당한 코딩 효율 개선이 이루어진다.

[0038] 도 7은 2개의 기준 계층들을 갖는 인트라 예측 모드의 예를 도시한다. 실제로, 새로운 인트라 예측 모드들은 개선된 신택스에서 고려된다. 첫번째 새로운 인트라 퍼디션 모드는 다중 기준 인트라 예측(MRIP)으로 명명된다. 이 도구는 블록의 인트라 예측을 위해 다수의 기준 계층들의 사용을 용이하게 한다. 통상적으로, 인트라 예측을 위해 2개의 기준 계층이 사용되며, 각각의 기준 계층은 좌측 기준 어레이와 상부 기준 어레이로 구성된다. 각각의 기준 계층은 기준 샘플 치환에 의해 구성된 다음 사전 필터링된다.

[0039] 각각의 기준 계층을 사용하여, 블록에 대한 예측이, 전형적으로 HEVC에서 또는 JEM에서 행해진 바와 같이, 구성된다. 최종 예측은 2개의 기준 계층들로부터 행해진 예측들의 가중 평균으로서 형성된다. 가장 가까운 기준 계층으로부터의 예측은 가장 먼 계층으로부터의 예측보다 높은 가중치를 부여받는다. 전형적인 사용된 가중치는 3과 1이다.

[0040] 예측 프로세스의 마지막 부분에서, 모드 의존적 프로세스를 사용하여, 소정의 예측 모드들에 대한 경계 샘플을 평활화하기 위해 복수의 기준 계층이 사용될 수 있다.

[0041] 예시적인 실시예에서, 비디오 압축 도구들은 또한 샘플 적응적 오프셋(SAO)에 대한 적응적 블록 크기를 포함한다. 이 도구는 HEVC에서 특정된 루프 필터이다. HEVC에서, SAO 프로세스는 하나의 블록의 재구성된 샘플들을 몇몇 클래스들로 분류하고, 일부 클래스들에 속하는 샘플들은 오프셋들로 정정된다. SAO 파라미터들은 블록마다 인코딩되거나, 오직 좌측 또는 상부 이웃 블록들로부터만 상속될 수 있다.

[0042] SAO 팔레트 모드를 정의함으로써 추가의 개선들이 제안된다. SAO 팔레트는 블록마다 동일한 SAO 파라미터들을 유지하지만, 블록들은 다른 모든 블록들로부터 이 파라미터들을 상속할 수도 있다. 이것은 하나의 블록에 대해 가능한 SAO 파라미터들의 범위를 확장함으로써 SAO에 더 많은 유연성을 제공한다. SAO 팔레트는 상이한 SAO 파라미터들의 세트로 구성된다. 각각의 블록에 대해, 어떤 SAO 파라미터를 사용할 것인지를 표시하기 위해 인덱스가 코딩된다.

[0043] 도 8은 양방향 조명 보상을 위한 보상 모드의 예시적인 실시예를 도시한다. 조명 보상(IC)은 가능하게는 공간 또는 시간 로컬 조명 변화를 고려함으로써 모션 보상을 통해 획득된 블록 예측 샘플들(SMC)을 정정하는 것을 허용한다.

[0044] $S_{IC} = a_i \cdot S_{MC} + b_i$

[0045] 양방향 예측의 경우, IC 파라미터들은 도 8에 도시된 바와 같이 2개의 기준 CU 샘플의 샘플들을 사용하여 추정된다. 먼저, IC 파라미터들(a, b)은 2개의 기준 블록 사이에서 추정되고, 다음으로 기준 블록과 현재 블록 사이의 IC 파라미터들(ai,bi)i=0,1은 다음과 같이 도출된다:

[0046] $$\begin{cases} a_0 = a.\alpha + (1-\alpha) \\ b_0 = \alpha.b \\ a_1 = \alpha + (1-\alpha)/a \\ b_1 = -b(1-\alpha)/a \end{cases}$$ 여기서: $\alpha = \frac{poc_{cur} - poc_0}{poc_1 - poc_0}$

[0047] CU 크기가 폭 또는 높이면에서 8 이하이면, IC 파라미터들은 단방향 프로세스를 사용하여 추정된다. CU 크기가 8보다 클 때, 양방향 또는 단방향 프로세스로부터 도출된 IC 파라미터들을 사용하는 것 사이의 선택은 IC 보상된 기준 블록들의 평균의 차이를 최소화하는 IC 파라미터들을 선택함으로써 이루어진다. 양방향 광학 흐름(BIO)은 양방향 조명 보상 도구로 인에이블된다.

[0048] 위에 소개된 복수의 도구는 비디오 인코더(100)에 의해 선택되고 사용되며 비디오 디코더(200)에 의해 획득되고 사용될 필요가 있다. 이를 위해, 이들 도구의 사용을 나타내는 정보는 하이 레벨 신택스 정보의 형태로 비디오 인코더에 의해 생성되고 비디오 디코더(200)에 의해 획득되는 코딩된 비트스트림으로 전달된다. 이를 위해, 이하에 대응하는 신택스가 정의되고 기술된다. 복수의 도구를 수집하는 이 신택스는 개선된 코딩 효율을 허용한다.

[0049] 이 신택스 구조는 HEVC 신택스 구조를 기본으로서 사용하고 일부 추가적인 신택스 요소들을 포함한다. 다음의 표들에서 이탤릭체 굵은 폰트로 시그널링되고 회색으로 강조 표시된 신택스 요소들은 예시적인 실시예에 따른 추가적인 신택스 요소들에 대응한다. 신택스 요소들은 동일한 기능들을 여전히 처리하면서 신택스 표들에 도시된 것과 다른 형태들 또는 명칭들을 취할 수 있다는 점에 유의해야 한다. 신택스 요소들은 상이한 레벨들에 존재할 수 있는데, 예를 들어, 일부 신택스 요소들은 SPS(sequence parameter set)에 배치될 수 있고 일부 신택스 요소들은 PPS(picture parameter set)에 배치될 수 있다.

[0050] 아래의 표 1은 시퀀스 파라미터 세트(SPS)를 예시하고 적어도 하나의 예시적인 실시예에 따라 이 파라미터 세트에 삽입된 새로운 신택스 요소들을 소개하는데, 보다 정확하게는, multi_type_tree_enabled_primary, log2_min_cu_size_minus2, log2_max_cu_size_minus4, log2_max_tu_size_minus2, sep_tree_mode_intra, multi_type_tree_enabled_secondary, sps_bdip_enabled_flag, sps_mrip_enabled_flag, use_erp_aqp_flag, use_high_perf_chroma_qp_table, abt_one_third_flag, sps_num_intra_mode_ratio이다.

표 1

seq_parameter_set_rbsp( ) {	기술자
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(1, sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
multi_type_tree_enabled_primary	ue(v)
log2_min_cu_size_minus2	ue(v)
log2_max_cu_size_minus4	ue(v)
log2_max_tu_size_minus2	ue(v)
sep_tree_mode_intra	u(1)
if (sep_tree_mode_intra)	
multi_type_tree_enabled_secondary	ue(v)
sample_adaptive_offset_enabled_flag	u(1)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)

[0051]

log2_min_pcm_luma_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_luma_coding_block_size	ue(v)
pcm_loop_filter_disabled_flag	u(1)
}	
num_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_short_term_ref_pic_sets; i++)	
st_ref_pic_set(i)	
long_term_ref_pics_present_flag	u(1)
if(long_term_ref_pics_present_flag) {	
num_long_term_ref_pics_sps	ue(v)
for(i = 0; i < num_long_term_ref_pics_sps; i++) {	
lt_ref_pic_poc_lsb_sps[i]	u(v)
used_by_curr_pic_lt_sps_flag[i]	u(1)
}	
}	
sps_temporal_mvp_enabled_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters( )	
use_imv	u(1)
fruc_merge_mode	u(1)
if(fruc_merge_mode) {	
idx_num_template	ue(v)
idx_num_template_ic	ue(v)
fruc_template_affine	u(1)
}	
mode_bilateral_TM	ue(v)
optical_flow_filtering	ue(v)
atmvp_flag	u(1)
sps_lic_enabled_flag	u(1)
sps_bidir_ic_present_flag	u(1)
sps_use_intra_emt	u(1)
sps_use_inter_emt	u(1)
sps_nsst_enabled_flag	u(1)
sps_cross_component_prediction_enabled_flag	u(1)
sps_intra_4tap_filter_enabled_flag	u(1)
sps_intra_boundary_filter_enabled_flag	u(1)
sps_obmc_flag	u(1)
if(sps_obmc_flag)	
obmc_blk_size	ue(v)
obmc_for_sub_block	u(1)
sps_affine_enabled_flag	u(1)
use_NL_Bil_flag	u(1)
sps_bdip_enabled_flag	u(1)
sps_mrip_enabled_flag	u(1)
num_predicted_coef_signs	u(4)

[0052]

mc_frame_pad	u(1)
use_erp_aqp_flag	u(1)
use_chroma_qp_table	
if(use_chroma_qp_table)	
use_high_perf_chroma_qp_table	u(1)
log2_intra131modes_min_area_minus6	ue(v)
log2_intra65modes_min_area_minus4	ue(v)
abt_one_third_flag	u(1)
sps_num_intra_mode_ratio	u(1)
if(sps_range_extension_flag)	
sps_range_extension( )	
if(sps_multilayer_extension_flag)	
sps_multilayer_extension( ) /* specified in Annex F */	
if(sps_3d_extension_flag)	
sps_3d_extension( ) /* specified in Annex I */	
if(sps_extension_5bits)	
while(more_rbsp_data( ))	
sps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

[0053]

[0054] 표 1: 수정된 시퀀스 파라미터 세트

[0055] 새로운 신택스 요소들은 다음과 같이 정의된다:

[0056] - use_high_perf_chroma_qp_table: 이 신택스 요소는 슬라이스와 연관된 크로마 성분들의 디코딩에 사용된 QP를 도출하는데 사용된 크로마 QP 표를, 고려된 슬라이스의 크로마 성분들과 연관된 기본 크로마 QP의 함수로서 지정한다.

[0057] - multi_type_tree_enabled_primary: 이 신택스 요소는 코딩된 슬라이스들에서 허용되는 파티셔닝의 타입을 나타낸다. 적어도 하나의 구현에서, 이 신택스 요소는 (SPS에서) 시퀀스 레벨에서 시그널링되므로, 이 SPS를 사용하고 있는 모든 코딩된 슬라이스에 적용된다. 예를 들어, 이 신택스 요소의 제1 값은 QTBT(quad-tree and binary tree) 파티셔닝(도 5의 NO-SPLIT, QT-SPLIT, HOR 및 VER)을 허용하고, 제2 값은 QTBT 플러스 TT(triple-tree) 파티셔닝(도 5의 NO-SPLIT, QT-SPLIT, HOR, VER, HOR_TRIPLE 및 VER_TRIPLE)을 허용하고, 제3 값은 QTBT 플러스 ABT(asymmetric binary tree) 파티셔닝(도 5의 NO-SPLIT, QT-SPLIT, HOR, VER, HOR-UP, HOR_DOWN, VER_LEFT 및 VER_RIGHT)을 허용하고, 제4 값은 QTBT 플러스 TT 플러스 ABT 파티셔닝(도 5의 모든 분할 경우)을 허용한다.

[0058] - sep_tree_mode_intra: 이 신택스 요소는 루마 및 크로마 블록들에 대해 개별 코딩 트리가 사용되는지를 나타낸다. sep_tree_mode_intra가 1과 동일할 때, 루마 및 크로마 블록들은 독립적인 코딩 트리들을 사용하고, 따라서 루마 및 크로마 블록들의 분할은 독립적이다. 적어도 하나의 구현에서, 이 신택스 요소는 (SPS에서) 시퀀스 레벨에서 시그널링되므로, 이 SPS를 사용하고 있는 모든 코딩된 슬라이스에 적용된다.

[0059] - multi_type_tree_enabled_secondary: 이 신택스 요소는 코딩된 슬라이스들에서 크로마 블록들에 대해 파티셔닝의 타입이 허용된다는 것을 나타낸다. 이 신택스 요소가 취할 수 있는 값들은 전형적으로 신택스 요소 multi_type_tree_enabled_primary의 값들과 동일하다. 적어도 하나의 구현에서, 이 신택스 요소는 (SPS에서) 시퀀스 레벨에서 시그널링되므로, 이 SPS를 사용하고 있는 모든 코딩된 슬라이스에 적용된다.

[0060] - sps_bdip_enabled_flag: 이 신택스 요소는 JVET-J0022에 기술된 양방향 인트라 예측이 고려된 코딩된 비디오 시퀀스에 포함된 코딩된 슬라이스들에서 허용된다는 것을 나타낸다.

[0061] - sps_mrip_enabled_flag: 이 신택스 요소는 다중-기준 인트라 예측 도구가 고려된 코딩된 비디오 비트스트림에 포함된 코드 슬라이스들의 디코딩에 사용된다는 것을 나타낸다.

[0062] - use_erp_aqp_flag: 이 신택스 요소는 JVET-J0022에 기술된 바와 같이 VR360 ERP 콘텐츠에 대한 공간적으로 적응적인 양자화가 코딩된 슬라이스들에서 활성화되는지를 나타낸다.

[0063] - abt_one_third_flag: 이 신택스 요소는 초기 CU의 수평 또는 수직 차원의 1/3 및 2/3, 또는 2/3 및 1/3인 수평 또는 수직 차원을 갖는 2개의 파티션으로의 비대칭 파티셔닝이 코딩된 슬라이스들에서 활성화되는지를 나타

낸다.

[0064] - sps_num_intra_mode_ratio: 이 신택스 요소는 폭 또는 높이에서 3의 배수와 동일한 블록 크기에 사용되는 인트라 예측의 수를 도출하는 방법을 특정한다.

[0065] 또한, 코딩 유닛들(CU) 및 변환 유닛들(TU)의 크기를 제어하기 위해 다음의 3개의 신택스 요소들이 사용된다. 적어도 하나의 구현에서, 이들 신택스 요소는 (SPS에서) 시퀀스 레벨에서 시그널링되므로, 이 SPS를 사용하고 있는 모든 코딩된 슬라이스에 적용된다.

[0066] - log2_min_cu_size_minus2는 최소 CU 크기를 특정한다.

[0067] - log2_max_cu_size_minus4는 최대 CU 크기를 특정한다.

[0068] - log2_max_tu_size_minus2는 최대 TU 크기를 특정한다.

[0069] 변형 실시예에서, 위에 소개된 새로운 신택스 요소들은 시퀀스 파라미터 세트에 도입되지 않고, 픽처 파라미터 세트(PPS)에 도입된다.

[0070] 아래의 표 2는 SPS(sequence parameter set)를 예시하고 적어도 하나의 예시적인 실시예에 따라 이 파라미터 세트에 삽입된 새로운 신택스 요소들을 소개하는데, 보다 정확하게는:

[0071] - slice_sao_size_id: 이 신택스 요소는 SAO가 코딩된 슬라이스에 적용되는 블록들의 크기를 특정한다.

[0072] - slice_bidir_ic_enable_flag: 코딩된 슬라이스에 대해 양방향 조명 보상이 활성화되는지를 나타낸다.

표 2

slice_segment_header( ) {	기술자
first_slice_segment_in_pic_flag	u(1)
if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
no_output_of_prior_pics_flag	u(1)
slice_pic_parameter_set_id	ue(v)
if(!first_slice_segment_in_pic_flag) {	
if(dependent_slice_segments_enabled_flag)	
dependent_slice_segment_flag	u(1)
slice_segment_address	u(v)
}	
if(!dependent_slice_segment_flag) {	
for(i = 0; i < num_extra_slice_header_bits; i++)	
slice_reserved_flag[i]	u(1)
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag = = 1)	
colour_plane_id	u(2)
if(nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) {	
slice_pic_order_cnt_lsb	u(v)
short_term_ref_pic_set_sps_flag	u(1)
if(!short_term_ref_pic_set_sps_flag)	
st_ref_pic_set(num_short_term_ref_pic_sets)	
else if(num_short_term_ref_pic_sets > 1)	
short_term_ref_pic_set_idx	u(v)
if(long_term_ref_pics_present_flag) {	
if(num_long_term_ref_pics_sps > 0)	
num_long_term_sps	ue(v)
num_long_term_pics	ue(v)
for(i = 0; i < num_long_term_sps + num_long_term_pics; i++) {	
if(i < num_long_term_sps) {	
if(num_long_term_ref_pics_sps > 1)	
lt_idx_sps[i]	u(v)
} else {	
poc_lsb_lt[i]	u(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
delta_poc_msb_present_flag[i]	u(1)
if(delta_poc_msb_present_flag[i])	
delta_poc_msb_cycle_lt[i]	ue(v)
}	
}	
if(sps_temporal_mvp_enabled_flag)	
slice_temporal_mvp_enabled_flag	u(1)

[0073]

}	
if(sample_adaptive_offset_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(slice_sao_luma_flag || slice_sao_chroma_flag) {	
slice_sao_size_id	u(3)
if(slice_type == P || slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
if(lists_modification_present_flag && NumPicTotalCurr > 1)	
ref_pic_lists_modification( )	
if(slice_type == B)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && num_ref_idx_l0_active_minus1 > 0) || (!collocated_from_l0_flag && num_ref_idx_l1_active_minus1 > 0))	
collocated_ref_idx	ue(v)
}	
if((weighted_pred_flag && slice_type == P) || (weighted_bipred_flag && slice_type == B))	
pred_weight_table( )	
if(sps_ic_present_flag && slice_type != I) {	
slice_ic_enable_flag	u(1)
if(sps_bidir_ic_present_flag && slice_ic_enable_flag)	
slice_bidir_ic_enable_flag	u(1)
max_search_depth_region_tree	ue(v)
for (i = 0; i < max_search_depth_region_tree; i++)	
max_search_depth_prediction_tree[i]	ue(v)
if (slice_type == I)	
for (i = 0; i < max_search_depth_region_tree; i++)	
max_search_depth_prediction_tree_chroma[i]	
if (slice_type != I)	
max_num_merge_cand	ue(v)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)

[0074]

slice_cr_qp_offset	se(v)
}	
if(chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(pps_loop_filter_across_slices_enabled_flag && (slice_sao_luma_flag || slice_sao_chroma_flag || !slice_deblocking_filter_disabled_flag))	
slice_loop_filter_across_slices_enabled_flag	u(1)
}	
if(tiles_enabled_flag || entropy_coding_sync_enabled_flag) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset_minus1[i **]**	u(v)
}	
}	
if(pps_clip_adaptive_enable_flag)	
clip_adaptive_flag	u(1)
if(clip_adaptive_flag) {	
clip_adaptive_y_min	ue(v)
clip_adaptive_y_max	ue(v)
clip_adaptive_flag_chroma	u(1)
if(clip_adaptive_flag_chroma) {	
clip_adaptive_c0_min	ue(v)
clip_adaptive_c0_max	ue(v)
clip_adaptive_c1_min	ue(v)
clip_adaptive_c1_max	ue(v)
}	
}	
}	
if(sps_affine_enabled_flag && slice_type != I) {	
affine_control_flag[0]	u(1)
affine_control_flag[1]	u(1)
}	
if(slice_segment_header_extension_present_flag) {	
slice_segment_header_extension_length	ue(v)

[0075]

for(i = 0; i < slice_segment_header_extension_length; i++)	
slice_segment_header_extension_data_byte[i **]**	u(8)
}	
byte_alignment( )	
}	

[0076]

[0077] 표 2: 슬라이스 헤더

[0078] 아래의 표 3, 4, 5, 6은 적어도 하나의 예시적인 실시예에 따른 추가적인 파티셔닝 모드들을 지원하도록 수정된 코딩 트리 신택스를 예시한다. 특히, 비대칭 파티셔닝을 가능하게 하기 위한 신택스는 coding_binary_tree()에 특정된다. 새로운 신택스 요소들은 코딩 트리 신택스에 삽입되는데, 보다 정확하게는:

[0079] - 비대칭 이진 분할이 현재 CU에 대해 허용되는지를 나타내는 asymmetricSplitFlag 및

[0080] - 비대칭 이진 분할의 타입을 나타내는 asymmetric_type에 삽입된다. 수평 분할이 위쪽 또는 아래쪽이 되거나 수직 분할이 왼쪽 또는 오른쪽이 될 수 있게 하기 위해, 2개의 값을 취할 수 있다.

[0081] 또한, 파라미터 btSplitMode는 상이하게 해석된다.

표 3

[0082]

coding_tree_unit( ) {	기술자
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
if(slice_sao_luma_flag || slice_sao_chroma_flag)	
if(slice_type == I) {	
coding_tree(xCtb, yCtb, CtbSizeX, CtbSizeY, LUMA_TREE, 0, 0)	
coding_tree(xCtb, yCtb, CtbSizeX, CtbSizeY, CHROMA_TREE, 0, 0)	
} else {	
coding_tree(xCtb, yCtb, CtbSizeX, CtbSizeY, LUMA_CHROMA_TREE, 0, 0)	
}	
}	

[0083] 표 3: coding_tree_unit

표 4

[0084]

coding_tree(x0, y0, log2CbSize, cqtDepth) {	기술자
if(x0 + (1 << log2CbSize) <= pic_width_in_luma_samples && y0 + (1 << log2CbSize) <= pic_height_in_luma_samples && log2CbSize > MinCbLog2SizeY)	
split_cu_flag[x0][y0]	ae(v)
if(split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize − 1))	
y1 = y0 + (1 << (log2CbSize − 1))	
coding_tree(x0, y0, log2CbSize − 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples)	
coding_tree(x1, y0, log2CbSize − 1, cqtDepth + 1)	
if(y1 < pic_height_in_luma_samples)	
coding_tree(x0, y1, log2CbSize − 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_tree(x1, y1, log2CbSize − 1, cqtDepth + 1)	
} else	
coding_binary_tree(x0, y0, 1<<log2CbSize,1<< log2CbSize,cqtDepth)	
}	

[0085] 표 4: coding_tree

표 5

coding_binary_tree(x0, y0, width, height, cqtDepth) {	기술자
if(btSplitAllowed(x0,y0,width,height){	
bt_split_mode(x0,y0,width,height,cqtDepth)	
}	
if(btSplitFlag) {	
if(btSplitMode==HOR) {	
x1 = x0	
y1 = y0 + (height >> 1)	
sub_width_1 = sub_width_0 = width;	
sub_height_1 = sub_height_0 = (height >> 1)	
}	
else if(btSplitMode==VER) {	
x1 = x0 + (width >> 1)	
y1 = y0	
sub_width_1 = sub_width_0 = (width >> 1)	
sub_height_1 = sub_height_0 = height	
}	
else if(btSplitMode==HOR_UP) {	
x1 = x0	
y1 = y0 + (height >> 2)	
sub_width_1 = sub_width_0 = width	
sub_height_0 = (height >> 2)	
sub_height_1 = ((height *3) >> 2)	
}	
else if(btSplitMode==HOR_DOWN) {	
x1 = x0	
y1 = y0 + ((height*3) >> 2)	
sub_width_1 = sub_width_0 = width	
sub_height_0 = ((height *3) >> 2)	
sub_height_1 = (height >> 2)	
}	
else if(btSplitMode==VER_LEFT) {	
x1 = x0 + (width >> 2)	
y1 = y0	
sub_width_0 = width >> 2	
sub_width_1 = (width *3) >> 2	
sub_height_1 = sub_height_0 = height	
}	
else if(btSplitMode==VER_RIGHT) {	
x1 = x0 + (width*3) >> 2	
y1 = y0	
sub_width_0 = (width*3) >> 2	
sub_width_1 = width >> 2	
sub_height_1 = sub_height_0 = height	
}	
coding_binary_tree(x0, y0, sub_width, sub_height, cqtDepth)	

[0086]

if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_binary_tree(x1, y1, sub_width, sub_height, cqtDepth)	
}	
} else	
coding_unit(x0, y0, width, height)	
}	

[0087]

[0088] 표 5: coding_binary_tree

표 6

[0089]

bt_split_mode(x0,y0,width,height,cqtDepth){	기술자
if(btSplitAllowed(x0,y0,width,height){	
btSplitFlag[x0][y0][cbSizeX][cbSizeY]	ae(v)
}	
if(btSplitFlag[x0][y0][cbSizeX][cbSizeY]) {	
if(horizontalSplitAllowed && verticalSplitAllowed){	
btSplitOrientation[x0][y0][cbSizeX][cbSizeY]	ae(v)
}	
if(btSplitOrientation==HOR && horizontal_asymmetric_allowed ||	
btSplitOrientation==VER && vertical_asymmetric_allowed){	
asymmetricSplitFlag*[x0][y0][cbSizeX][cbSizeY]*	ae(v)
if(asymmetricSplitFlag==true){	
asymmetric_type*[x0][y0][cbSizeX][cbSizeY]*	ae(v)
}	
}	
}	
}	

[0090] 표 6: bt_split_mode

[0091] 도 9는 적어도 하나의 실시예에 따른 bt-split-flag의 해석을 도시한다. 2개의 신택스 요소 asymmetricSplitFlag와 asymmetric_type는 파라미터 btSplitMode의 값을 특정하는데 사용된다. btSplitMode는 현재 CU에 적용되는 이진 분할 모드를 나타낸다. btSplitMode는 종래 기술과는 상이하게 도출된다. JEM에서, 값들 HOR, VER을 취할 수 있다. 개선된 신택스에서, 도 6에 도시된 파티셔닝에 대응하는 값들 HOR_UP, HOR_DOWN, VER_LEFT, VER_RIGHT를 추가로 취할 수 있다.

[0092] 아래의 표 7, 8, 9 및 10은 코딩 유닛 신택스 요소들을 예시하고 양방향 인트라 예측 모드에 대한 새로운 신택스 요소를 소개한다.

표 7

coding_unit(x0, y0, cuWidth, cuHeight, treeMode) {	기술자
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0])	
cu_data_merge(x0, y0, cuWidth, cuHeight, treeMode)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(cuPredMode[x0][y0] == MODE_INTRA)	
cu_data_intra(x0, y0, cuWidth, cuHeight, treeMode)	
else {	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0])	
cu_data_merge(x0, y0, cuWidth, cuHeight, treeMode)	
else	
cu_data_inter(x0, y0, cuWidth, cuHeight, treeMode)	
if(cuPredMode[x0][y0] != MODE_INTRA && !(merge_flag[x0][y0])	
rqt_root_cbf	ae(v)
if(rqt_root_cbf)	
cu_residual_data(x0, y0, cuWidth, cuHeight, treeMode)	
}	
}	
}	

[0093]

[0094] 표 7: coding_unit

표 8

cu_data_merge(x0, y0, cuWidth, cuHeight, treeMode) {	기술자
if(fruc_merge_enabled_flag)	
fruc_merge_mode[x0][y0]	ae(v)
if(fruc_merge_mode[x0][y0])	
if(lic_enabled_flag && fruc_merge_mode[x0][y0] == 1)	
lic_flag[x0][y0]	ae(v)
if(fruc_merge_mode[x0][y0] == 1 && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(lic_enabled_flag)	
lic_flag[x0][y0]	ae(v)
else	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	
}	

[0095]

[0096] 표 8: cu_data_merge

표 9

cu_data_intra(x0, y0, cuWidth, cuHeight, treeMode) {	기술자
if(treeMode & LUMA_TREE) {	
prev_intra_luma_pred_flag[x0][y0]	ae(v)
if(prev_intra_luma_pred_flag[x0][y0]) {	
mpm_idx[x0][y0]	ae(v)
} else {	
second_mpm_flag[x0][y0]	ae(v)
if(second_mpm_flag[x0][y0]	
second_mpm[x0][y0]	f(5)
else	
rem_intra_luma_pred_mode[x0][y0]	ae(v)
}	
if(bdip_enable_flag && ((luma_mode >= 2 && luma_mode < 18) || (luma_mode >= 114 && luma_mode < 130)	
bdip_enable_flag[x0][y0]	ae(v)
}	
if(treeMode & CHROMA_TREE) {	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
}	

[0097]

[0098] 표 9: cu_data_intra

표 10

cu_data_inter(x0, y0, cuWidth, cuHeight, treeMode) {	기술자
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(affine_enabled_flag && cuWidth > 8 && cuHeight > 8)	
affine_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
if(affine_flag[x0][y0]) {	
mvd_gr0(x0, y0, 0, AFFINE_LEFT)	ae(v)
mvd_gr0(x0, y0, 0, AFFINE_RIGHT)	ae(v)
} else	
mvd_ gr0 (x0, y0, 0, AFFINE_OFF)	
mvp_l0_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
} else if(affine_flag[x0][y0]) {	
mvd_gr0(x0, y0, 1, AFFINE_LEFT)	ae(v)
mvd_gr0(x0, y0, 1, AFFINE_RIGHT)	ae(v)
} else	
mvd_gr0(x0, y0, 1, AFFINE_OFF)	
mvp_l1_flag[x0][y0]	ae(v)
}	

[0099]

if(imv_enabled_flag && CuHasNonZeroMvd)	
imv_mode[x0][y0]	ae(v)
if(affine[x0][y0] || CuHasNonZeroMvd)	
{	
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(affine_flag[x0][y0]) {	
mvd_remain(x0, y0, 0, AFFINE_LEFT)	ae(v)
mvd_remain(x0, y0, 0, AFFINE_RIGHT)	ae(v)
} else	
mvd_ remain(x0, y0, 0, AFFINE_OFF)	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
} else if(affine_flag[x0][y0]) {	
mvd_gr0(x0, y0, 1, AFFINE_LEFT)	ae(v)
mvd_gr0(x0, y0, 1, AFFINE_RIGHT)	ae(v)
} else	
mvd_gr0(x0, y0, 1, AFFINE_OFF)	ae(v)
}	
}	
if(obmc_enabled_flag && cuWidth*cuHeight <= 16*16)	
obmc_flag[x0][y0]	ae(v)
if(lic_enabled_flag && !affine_flag[x0][y0])	
lic_flag[x0][y0]	ae(v)
}	

[0100]

[0101] 표 10: cu_data_inter

표 11

cu_residual_data(x0, y0, cuWidth, cuHeight, treeMode) {	기술자
if(treeMode & CHROMA_TREE) {	
cbf_cb[x0][y0]	ae(v)
cbf_cr[x0][y0]	ae(v)
}	
if(treeMode & LUMA_TREE) {	
if(cuPredMode[x0][y0] == MODE_INTRA || cbf_cb[x0][y0] || cbf_cr[x0][y0])	
cbf_luma[x0][y0]	ae(v)
}	
if(treeMode & CHROMA_TREE) {	
if(cbf_cb[x0][y0])	
residual_coding(x0, y0, cuWidth / 2, cuHeight / 2, treeMode, 1)	
if(cbf_cr[x0][y0])	
residual_coding(x0, y0, cuWidth / 2, cuHeight / 2, treeMode, 2)	
}	
if(treeMode & LUMA_TREE) {	
if(emt_enable_flag && cuWidth <= 64 && cuHeight <= 64)	
emt_cu_flag[x0][y0]	ae(v)
residual_coding(x0, y0, cuWidth, cuHeight, treeMode, 0)	
}	

[0102]

[0103] 표 11: cu_residual_data

[0104] 본 문헌에서 논의되는 신택스 요소들 중 몇몇은 어레이들로서 정의된다. 예를 들어, btSplitFlag는 픽처 내의 수평 및 수직 위치들에 의해 그리고 코딩된 블록 수평 및 수직 크기들에 의해 인덱싱된, 차원 4의 어레이로서 정의된다. 표기를 단순화하기 위해, 신택스 요소들의 시맨틱 설명에서, 인덱스들은 유지되지 않는다 (예를 들어, btSplitFlag[x0][y0][cbSizeX][cbSizeY]는 단순히 btSplitFlag로 표기된다).

[0105] 도 10은 다양한 양태들 및 실시예들이 구현되는 시스템의 예의 블록도를 도시한다. 시스템(1000)은 이하에 설명되는 다양한 컴포넌트들을 포함하는 디바이스로서 구현될 수 있고, 본 출원에 설명된 양태들 중 하나 이상을 수행하도록 구성된다. 그러한 디바이스들의 예들은 개인용 컴퓨터들, 랩톱 컴퓨터들, 스마트폰들, 태블릿 컴퓨터들, 디지털 멀티미디어 셋톱 박스들, 디지털 텔레비전 수신기들, 개인 비디오 기록 시스템들, 접속된 가전 기기들, 인코더들, 트랜스코더들, 및 서버들과 같은 다양한 전자 디바이스들을 포함하지만 이에 한정되지는 않는다. 시스템(1000)의 요소들은, 단독으로 또는 조합하여, 단일 집적 회로, 다중의 IC, 및/또는 개별 컴포넌트들로 구현될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 시스템(1000)의 처리 및 인코더/디코더 요소들은 다중의 IC 및/또는 개별 컴포넌트들에 걸쳐 분산된다. 다양한 실시예들에서, 시스템(1000)은, 예를 들어, 통신 버스를 통해 또는 전용 입력 및/또는 출력 포트들을 통해 다른 유사한 시스템들, 또는 다른 전자 디바이스들에 통신가능하게 결합된다. 다양한 실시예들에서, 시스템(1000)은 본 문서에서 설명된 양태들 중 하나 이상을 구현하도록 구성된다.

[0106] 시스템(1000)은, 예를 들어, 본 문서에 설명된 다양한 양태들을 구현하기 위해, 그 가운데 로딩된 명령어들을 실행하도록 구성된 적어도 하나의 프로세서(1010)를 포함한다. 프로세서(1010)는 내장된 메모리, 입력 출력 인터페이스, 및 본 기술분야에 공지된 다양한 다른 회로들을 포함할 수 있다. 시스템(1000)은 적어도 하나의 메모리(1020)(예를 들어, 휘발성 메모리 디바이스 및/또는 비휘발성 메모리 디바이스)를 포함한다. 시스템(1000)은 EEPROM, ROM, PROM, RAM, DRAM, SRAM, 플래시, 자기 디스크 드라이브, 및/또는 광학 디스크 드라이브를 포함하지만 이에 한정되지 않는 비휘발성 메모리 및/또는 휘발성 메모리를 포함할 수 있는 저장 디바이스(1040)를 포함한다. 저장 디바이스(1040)는 비제한적인 예들로서, 내부 저장 디바이스, 부착된 저장 디바이스, 및/또는 네트워크 액세스가능 저장 디바이스를 포함할 수 있다.

[0107] 시스템(1000)은, 예를 들어, 인코딩된 비디오 또는 디코딩된 비디오를 제공하기 위해 데이터를 처리하도록 구성된 인코더/디코더 모듈(1030)을 포함하고, 인코더/디코더 모듈(1030)은 그 자신의 프로세서 및 메모리를 포함할 수 있다. 인코더/디코더 모듈(1030)은 인코딩 및/또는 디코딩 기능들을 수행하기 위해 디바이스에 포함될 수 있는 모듈(들)을 나타낸다. 알려진 바와 같이, 디바이스는 인코딩 및 디코딩 모듈들 중 하나 또는 둘 모두를 포함할 수 있다. 또한, 인코더/디코더 모듈(1030)은 시스템(1000)의 별도 요소로서 구현될 수 있거나, 또는 본 기술분야의 통상의 기술자에게 공지된 바와 같이 하드웨어 및 소프트웨어의 조합으로서 프로세서(1010) 내에 통합될 수 있다.

[0108] 본 문서에 설명된 다양한 양태들을 수행하기 위해 프로세서(1010) 또는 인코더/디코더(1030) 상에 로딩될 프로그램 코드는 저장 디바이스(1040)에 저장될 수 있고, 후속하여 프로세서(1010)에 의한 실행을 위해 메모리(1020) 상에 로딩될 수 있다. 다양한 실시예들에 따르면, 프로세서(1010), 메모리(1020), 저장 디바이스(1040), 및 인코더/디코더 모듈(1030) 중 하나 이상은 본 문서에 설명된 프로세스들의 수행 동안 다양한 아이템들 중 하나 이상을 저장할 수 있다. 이러한 저장된 아이템들은 입력 비디오, 디코딩된 비디오 또는 디코딩된 비디오의 부분들, 비트스트림, 행렬들, 변수들, 및 수학식들, 공식들, 동작들 및 동작 로직의 처리로부터의 중간 또는 최종 결과들을 포함할 수 있지만, 이에 한정되지는 않는다.

[0109] 몇몇 실시예들에서, 프로세서(1010) 및/또는 인코더/디코더 모듈(1030)의 내부에 있는 메모리는 명령어들을 저장하고 및 인코딩 또는 디코딩 동안 필요한 처리를 위한 작업 메모리를 제공하기 위해 사용된다. 그러나, 다른 실시예들에서, 처리 디바이스 외부의 메모리(예를 들어, 처리 디바이스는 프로세서(1010) 또는 인코더/디코더 모듈(1030) 중 하나일 수 있음)가 이러한 기능들 중 하나 이상을 위해 사용된다. 외부 메모리는 메모리(1020) 및/또는 저장 디바이스(1040), 예를 들어, 동적 휘발성 메모리 및/또는 비휘발성 플래시 메모리일 수 있다. 몇몇 실시예들에서, 외부 비휘발성 플래시 메모리는 텔레비전의 운영 체제를 저장하기 위해 사용된다. 적어도 하나의 실시예에서, RAM과 같은 고속 외부 동적 휘발성 메모리는 MPEG-2, HEVC, 또는 VVC(Versatile Video Coding)와 같은 비디오 코딩 및 디코딩 동작들을 위한 작업 메모리로서 사용된다.

[0110] 시스템(1000)의 요소들에의 입력은 블록(1130)에 나타낸 바와 같이 다양한 입력 디바이스들을 통해 제공될 수 있다. 이러한 입력 디바이스들은 (i) 예를 들어, 브로드캐스터에 의해 OTA(over the air)로 송신되는 RF 신호를 수신하는 RF 부분, (ii) 복합 입력 단자, (iii) USB 입력 단자, 및/또는 (iv) HDMI 입력 단자를 포함하지만,

이에 한정되지는 않는다.

[0111] 다양한 실시예들에서, 블록(1130)의 입력 디바이스들은 본 기술분야에 공지된 바와 같은 연관된 제각기 입력 처리 요소들을 갖는다. 예를 들어, RF 부분은 (i) 원하는 주파수를 선택하는 것(또한 신호를 선택하는 것, 또는 주파수들의 대역으로 신호를 대역 제한하는 것으로 지칭됨), (ii) 선택된 신호를 하향 변환하는 것, (iii) 주파수들의 더 좁은 대역으로 다시 대역 제한하여 특정 실시예들에서 채널로 지칭될 수 있는 (예를 들어) 신호 주파수 대역을 선택하는 것, (iv) 하향 변환된 및 대역 제한된 신호를 복조하는 것, (v) 에러 정정을 수행하는 것, 및 (vi) 원하는 데이터 패킷들의 스트림을 선택하기 위해 역다중화하는 것에 필요한 요소들과 연관될 수 있다. 다양한 실시예들의 RF 부분은 이러한 기능들을 수행하기 위한 하나 이상의 요소, 예를 들어, 주파수 선택기들, 신호 선택기들, 대역 제한기들, 채널 선택기들, 필터들, 다운컨버터들, 복조기들, 에러 정정기들, 및 디멀티플렉서들을 포함한다. RF 부분은, 예를 들어, 수신된 신호를 더 낮은 주파수(예를 들어, 중간 주파수 또는 근 기저대역 주파수)로 또는 기저대역으로 하향 변환하는 것을 포함하여, 다양한 이러한 기능들을 수행하는 튜너를 포함할 수 있다. 하나의 셋톱 박스 실시예에서, RF 부분 및 그것의 연관된 입력 처리 요소는 유선(예를 들어, 케이블) 매체를 통해 송신되는 RF 신호를 수신하고, 필터링, 하향 변환, 및 원하는 주파수 대역으로의 다시 필터링에 의해 주파수 선택을 수행한다. 다양한 실시예들은 전술한(및 다른) 요소들의 순서를 재배열하고, 이 요소들의 일부를 제거하고, 및/또는 유사하거나 상이한 기능들을 수행하는 다른 요소들을 추가한다. 요소들을 추가하는 것은, 예를 들어, 증폭기들과 아날로그-투-디지털 변환기를 삽입하는 것과 같이 기존 요소들 사이 내에 요소들을 삽입하는 것을 포함할 수 있다. 다양한 실시예들에서, RF 부분은 안테나를 포함한다.

[0112] 또한, USB 및/또는 HDMI 단말들은 USB 및/또는 HDMI 접속들을 통해 다른 전자 디바이스들에 시스템(1000)을 접속하기 위한 제각기 인터페이스 프로세서들을 포함할 수 있다. 입력 처리의 다양한 양태들, 예를 들어, 리드-솔로몬(Reed-Solomon) 에러 정정이, 예를 들어, 별도의 입력 처리 IC 내에서 또는 필요에 따라 프로세서(1010) 내에서 구현될 수 있다는 것을 이해해야 한다. 유사하게, USB 또는 HDMI 인터페이스 처리의 양태들은 필요에 따라 별도의 인터페이스 IC들 내에서 또는 프로세서(1010) 내에서 구현될 수 있다. 복조된, 에러 정정된, 및 역다중화된 스트림은, 예를 들어, 프로세서(1010), 및 출력 디바이스 상의 프리젠테이션을 위해 필요한 대로 데이터 스트림을 처리하기 위해 메모리 및 저장 요소들과 조합하여 동작하는 인코더/디코더(1030)를 포함하는 다양한 처리 요소들에 제공된다.

[0113] 시스템(1000)의 다양한 요소들은 통합된 하우징 내에 제공될 수 있고, 통합된 하우징 내에서, 다양한 요소들은 적절한 접속 배열, 예를 들어, I2C 버스, 배선, 및 인쇄 회로 기판들을 포함하는 본 기술분야에 알려진 바와 같은 내부 버스를 사용하여 상호접속되고 그들 사이에 데이터를 송신할 수 있다.

[0114] 시스템(1000)은 통신 채널(1060)을 통해 다른 디바이스들과의 통신을 가능하게 하는 통신 인터페이스(1050)를 포함한다. 통신 인터페이스(1050)는 통신 채널(1060)을 통해 데이터를 송신하고 수신하도록 구성된 송수신기를 포함할 수 있지만, 이에 한정되는 것은 아니다. 통신 인터페이스(1050)는 모뎀 또는 네트워크 카드를 포함할 수 있지만, 이에 한정되지는 않고, 통신 채널(1060)은 예를 들어, 유선 및/또는 무선 매체 내에서 구현될 수 있다.

[0115] 데이터는, 다양한 실시예들에서, IEEE802.11과 같은 Wi-Fi 네트워크를 사용하여 시스템(1000)에 스트리밍된다. 이러한 실시예들의 Wi-Fi 신호는 Wi-Fi 통신들을 위해 적응되는 통신 인터페이스(1050) 및 통신 채널(1060)을 통해 수신된다. 이러한 실시예들의 통신 채널(1060)은 스트리밍 애플리케이션들 및 다른 OTT(over-the-top) 통신들을 허용하기 위한 인터넷을 포함하는 외부 네트워크들에 대한 액세스를 제공하는 액세스 포인트 또는 라우터에 전형적으로 접속된다. 다른 실시예들은 입력 블록(1130)의 HDMI 접속을 통해 데이터를 전달하는 셋톱 박스를 사용하여 스트리밍된 데이터를 시스템(1000)에 제공한다. 또 다른 실시예들은 입력 블록(1130)의 RF 접속을 사용하여 스트리밍된 데이터를 시스템(1000)에 제공한다.

[0116] 시스템(1000)은 디스플레이(1100), 스피커들(1110), 및 다른 주변 기기들(1120)을 포함하는 다양한 출력 디바이스들에 출력 신호를 제공할 수 있다. 다른 주변 기기들(1120)은, 실시예들의 다양한 예들에서, 독립형 DVR, 디스크 플레이어, 스테레오 시스템, 조명 시스템, 및 시스템(1000)의 출력에 기초하는 기능을 제공하는 다른 디바이스들 중 하나 이상을 포함한다. 다양한 실시예들에서, 제어 신호들은 AV.Link, CEC, 또는 사용자 개입을 이용하거나 사용하지 않고 디바이스-투-디바이스 제어를 가능하게 하는 다른 통신 프로토콜들과 같은 시그널링을 이용하여 시스템(1000)과 디스플레이(1100), 스피커들(1110), 또는 다른 주변 기기들(1120) 사이에서 통신된다. 출력 디바이스들은 각각의 인터페이스들(1070, 1080, 및 1090)을 통한 전용 접속들을 통해 시스템(1000)에 통신 가능하게 결합될 수 있다. 대안적으로, 출력 디바이스들은 통신 인터페이스(1050)를 통해 통신 채널(1060)을

사용하여 시스템(1000)에 접속될 수 있다. 디스플레이(1100) 및 스피커들(1110)은, 예를 들어, 텔레비전과 같은 전자 디바이스에서 시스템(1000)의 다른 컴포넌트들과 함께 단일 유닛에 통합될 수 있다. 다양한 실시예들에서, 디스플레이 인터페이스(1070)는, 예를 들어, 타이밍 제어기(T Con) 칩과 같은 디스플레이 드라이버를 포함한다.

[0117] 디스플레이(1100) 및 스피커(1110)는 대안적으로, 예를 들어, 입력(1130)의 RF 부분이 별도의 셋톱 박스의 일부인 경우, 다른 컴포넌트들 중 하나 이상으로부터 분리될 수 있다. 디스플레이(1100) 및 스피커들(1110)이 외부 컴포넌트들인 다양한 실시예들에서, 출력 신호는 예를 들어, HDMI 포트들, USB 포트들, 또는 COMP 출력들을 포함하는 전용 출력 접속들을 통해 제공될 수 있다. 여기에 기술된 구현들은 예컨대 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터 스트림 또는 신호로 구현될 수 있다. 비록 단지 단일 형태의 구현과 관련하여 기술될지라도(예컨대, 단지 방법으로서 논의될지라도), 논의된 특징들의 구현은 다른 형태들(예컨대, 장치 또는 프로그램)로 구현될 수 있다. 장치는 예를 들어, 적절한 하드웨어, 소프트웨어 및 펌웨어로 구현될 수 있다. 예를 들어, 방법들은, 예를 들어, 컴퓨터, 마이크로프로세서, 집적 회로(integrated circuit), 또는 프로그래밍 가능 로직 디바이스를 포함하는 처리 디바이스들을 일반적으로 지칭하는, 예를 들어, 프로세서와 같은 장치로 구현될 수 있다. 프로세서들은 또한, 예를 들어, 컴퓨터들, 셀폰들, PDA(portable/personal digital assistant)들, 및 최종 사용자들 간의 정보의 전달을 용이하게 하는 다른 디바이스들과 같은, 통신 디바이스들을 포함한다.

[0118] 도 11은 새로운 인코딩 도구들을 사용하는 실시예에 따른 인코딩 방법의 예의 흐름도를 도시한다. 이러한 인코딩 방법은 도 10에 기술된 시스템(1000)에 의해 수행될 수 있고, 보다 정확하게는 프로세서(1010)에 의해 구현될 수 있다. 적어도 하나의 실시예에서, 단계(1190)에서, 프로세서(1010)는 사용될 인코딩 도구들을 선택한다. 선택은 상이한 기술들을 사용하여 행해질 수 있다. 선택은 인코딩 및 디코딩 프로세스 동안 사용될 도구들(및 필요한 경우 대응하는 파라미터들)을 나타내는 인코딩 구성 파라미터(전형적으로 플래그)를 통해 사용자에 의해 행해질 수 있다. 예시적인 실시예에서, 선택은 파일 내의 값을 설정함으로써 행해지고, 플래그는 어떤 도구를 사용할지를 선택하기 위해 플래그의 값을 해석하는 인코딩 디바이스에 의해 판독된다. 예시적인 실시예에서, 선택은 인코더 디바이스를 처리하는 그래픽 사용자 인터페이스를 사용하여 인코딩 구성 파라미터를 선택하는 사용자의 수동 동작에 의해 행해진다. 일단 이러한 선택이 수행되면, 다른 인코딩 도구들 중에서 선택된 도구를 이용하여 단계(1193)에서 인코딩이 수행되고, 도구들의 선택은 하이-레벨 신택스 요소들에서(예를 들어, 다음 SPS, PPS 또는 심지어 슬라이스 헤더에서) 시그널링된다.

[0119] 도 12는 새로운 인코딩 도구들을 사용하는 실시예에 따른 디코딩 방법의 일부의 예의 흐름도를 도시한다. 이러한 디코딩 방법은 도 10에 기술된 시스템(1000)에 의해 수행될 수 있고, 보다 정확하게는 프로세서(1010)에 의해 구현될 수 있다. 적어도 하나의 실시예에서, 단계 1200에서, 프로세서(1010)는 (예를 들어, 입력 인터페이스 상에서 수신되거나 또는 미디어 지원으로부터 판독된) 신호에 액세스하고, 하이 레벨 신택스 요소들을 추출 및 분석하여 인코딩 디바이스에서 선택된 도구들을 결정한다. 단계(1210)에서, 이러한 도구들은 디코딩을 수행하고 예를 들어, 디바이스에 제공되거나 또는 디바이스 상에 표시될 수 있는 디코딩된 이미지를 생성하기 위해 사용된다.

[0120] "하나의 실시예" 또는 "실시예" 또는 "하나의 구현" 또는 "구현"은 물론, 그의 다른 변형들에 대한 언급은 실시예와 관련하여 기술된 특정 특징, 구조, 특성 등이 적어도 하나의 실시예에 포함된다는 것을 의미한다. 따라서, 본 명세서 전반에 걸친 다양한 위치들에서 등장하는 "하나의 실시예에서" 또는 "실시예에서" 또는 "하나의 구현에서" 또는 "구현에서" 뿐만 아니라 다른 변형들은, 반드시 동일한 실시예 전부를 참조할 필요는 없다.

[0121] 게다가, 본원 또는 그의 청구항들은 다양한 정보들을 "결정하는 것"을 언급할 수 있다. 정보를 결정하는 것은 예를 들어, 정보를 추정하는 것, 정보를 계산하는 것, 정보를 예측하는 것 또는 메모리로부터의 정보를 검색하는 것 중 하나 이상을 포함할 수 있다.

[0122] 또한, 본 명세서 또는 그것의 청구항들은 다양한 정보에 "접근하는 것"을 언급할 수 있다. 정보에 액세스하는 것은, 예를 들어, 정보를 수신하는 것, (예를 들어, 메모리로부터) 정보를 검색하는 것, 정보를 저장하는 것, 정보를 이동시키는 것, 정보를 카피하는 것, 정보를 계산하는 것, 정보를 예측하는 것, 또는 정보를 추정하는 것 중 하나 이상을 포함할 수 있다.

[0123] 추가로, 본 출원 또는 그것의 청구항들은 다양한 정보를 "수신하는 것"을 언급할 수 있다. 수신하는 것은, "액세스하는 것"에서와 같이 광의의 용어로 의도된다. 정보를 수신하는 것은, 예를 들어, 정보에 액세스하는 것,

또는 (예를 들어, 메모리 또는 광학 매체 스토리지로부터) 정보를 검색하는 것 중 하나 이상을 포함할 수 있다. 또한, "수신하는 것"은 통상적으로 예를 들어, 정보를 저장하는 것, 정보를 처리하는 것, 정보를 송신하는 것, 정보를 이동시키는 것, 정보를 카피하는 것, 정보를 소거하는 것, 정보를 계산하는 것, 정보를 결정하는 것, 정보를 예측하는 것, 또는 정보를 추정하는 것과 같은 동작들 동안 하나의 방식 또는 다른 방식으로 수반된다.

[0124] 예를 들어, "A/B", "A 및/또는 B(A and/or B)" 및 "A 및 B 중 적어도 하나(at least one of A and B)"의 경우들에서, 이하의 "/", "및/또는(and/or)" 및 "~ 중 적어도 하나(at least one of)"중 임의의 것의 사용은, 처음 열거된 옵션(A) 만을 선택함, 또는 2번째로 열거된 옵션(B) 만을 선택함, 또는 옵션들 둘 다(A 및 B)를 선택함을 망라하려는 의도임을 알 것이다. 추가 예로서, "A, B 및/또는 C" 및 "A, B 및 C 중 적어도 하나"의 경우들에서, 이러한 어구는 첫번째로 열거된 옵션(A)만의 선택, 또는 두번째로 열거된 옵션(B)만의 선택, 또는 세번째로 열거된 옵션(C)만의 선택, 또는 첫번째와 두번째로 열거된 옵션들(A 및 B)만의 선택, 또는 첫번째와 세번째로 열거된 옵션들(A 및 C)만의 선택, 또는 두번째와 세번째로 열거된 옵션들(B 및 C)만의 선택, 또는 3개의 옵션(A 및 B 및 C) 전부의 선택을 포함하는 것으로 의도된다. 이것은 본 기술분야 및 관련 기술분야의 통상의 기술자에 의해 손쉽게 명백해지는 바와 같이 열거된 많은 항목들에 대해 확장될 수 있다.

[0125] 이 분야의 기술자에게 명백한 바와 같이, 구현들은 예컨대 저장되거나 또는 송신될 수 있는 정보를 전달(carry)하도록 포맷팅된 다양한 신호들을 생성할 수 있다. 정보는 예를 들어, 방법을 수행하기 위한 명령어들, 또는 설명되는 구현들 중 하나에 의해 생성되는 데이터를 포함할 수 있다. 예를 들어, 신호는 설명되는 실시예의 비트스트림을 전달하도록 포맷팅될 수 있다. 이러한 신호는 예컨대(예컨대, 스펙트럼의 무선 주파수 부분을 사용하여) 전자기파로서 또는 기저대역 신호로서 포맷팅될 수 있다. 포맷은, 예를 들어, 데이터 스트림을 인코딩하고 인코딩된 데이터 스트림으로 캐리어를 변조하는 것을 포함할 수 있다. 신호가 전달하는 정보는, 예를 들어, 아날로그 또는 디지털 정보일 수 있다. 신호는 공지된 바와 같은 다양한 상이한 유선 또는 무선 링크를 통해 송신될 수 있다. 신호는 프로세서 판독가능 매체상에 저장될 수 있다.

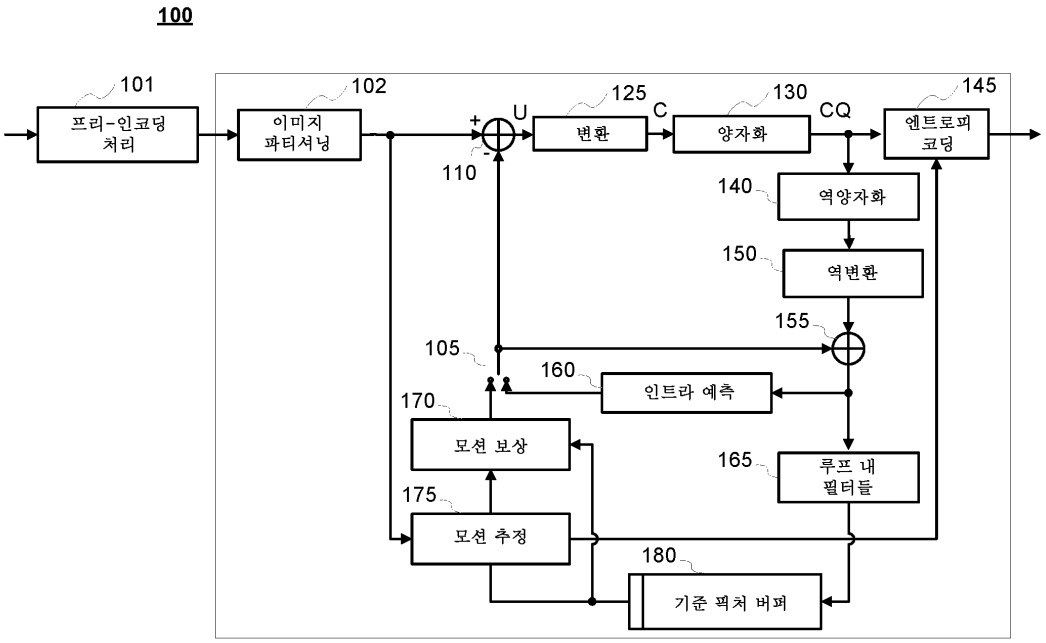
[0126] 적어도 한 실시예의 제1, 제2, 제3, 제4, 제5, 제6, 제7 및 제8 양태의 변형에서, 파티셔닝의 타입을 나타내는 파라미터는 쿼드-트리 및 이진 트리 파티셔닝을 위한 제1 값, 쿼드-트리 및 이진 트리 파티셔닝 플러스 트리플-트리 파티셔닝을 위한 제2 값, 쿼드-트리 및 이진 트리 파티셔닝 플러스 비대칭 이진 트리 파티셔닝을 위한 제3 값, 및 쿼드-트리 및 이진 트리 파티셔닝 플러스 트리플-트리 파티셔닝 플러스 비대칭 이진 트리 파티셔닝을 위한 제4 값을 포함한다.

[0127] 적어도 한 실시예의 제1, 제2, 제3, 제4, 제5, 제6, 제7 및 제8 양태의 변형에서, 파라미터들은 샘플 적응적 오프셋이 코딩된 슬라이스에 적용되는 블록들의 크기를 추가로 포함한다.

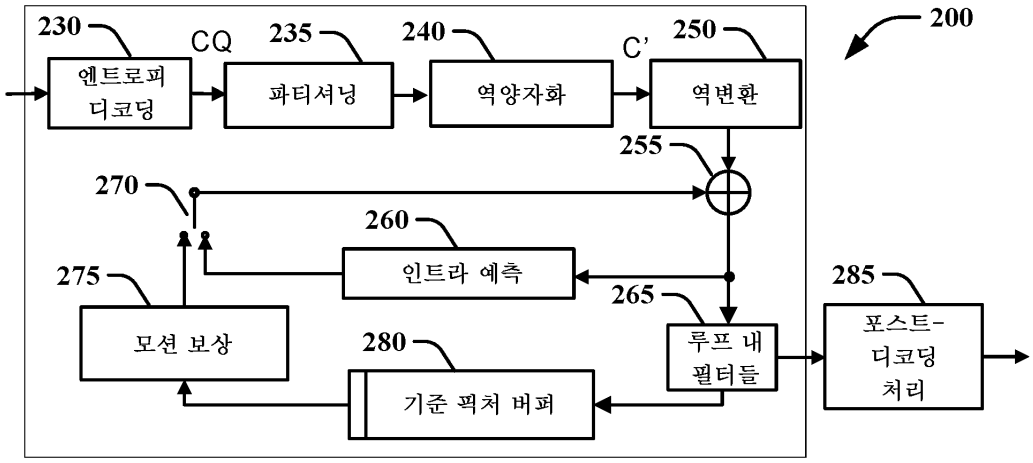
[0128] 적어도 한 실시예의 제1, 제2, 제3, 제4, 제5, 제6, 제7 및 제8 양태의 변형에서, 인트라 예측 모드의 타입은 인트라 양방향 예측 모드 및 다중-기준 인트라 예측 모드 중 적어도 하나를 포함한다.

도면

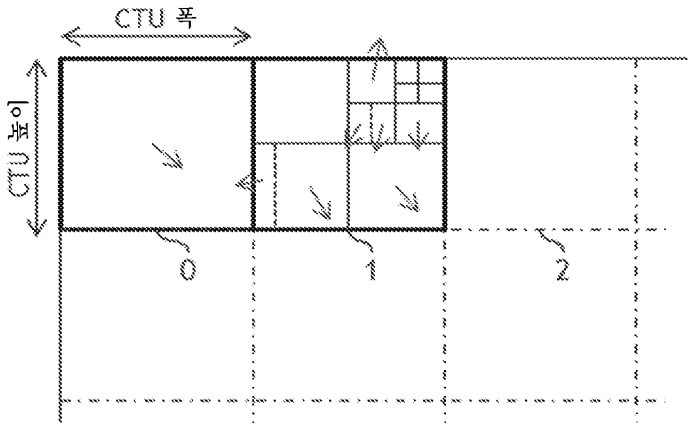
도면1


100
101
프리-인코딩 처리
102
이미지 파티셔닝
+
-
U
110
125
변환
C
130
양자화
CQ
145
엔트로피 코딩
140
역양자화
150
역변환
155
105
160
인트라 예측
170
모션 보상
165
루프 내 필터들
175
모션 추정
180
기준 픽처 버퍼

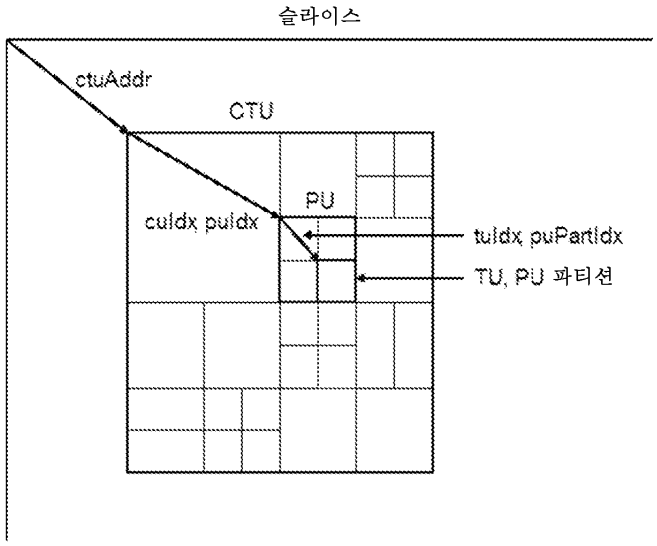
도면2


200
230
엔트로피 디코딩
CQ
235
파티셔닝
240
역양자화
C'
250
역변환
255
270
260
인트라 예측
275
모션 보상
265
280
기준 픽처 버퍼
루프 내 필터들
285
포스트-디코딩 처리

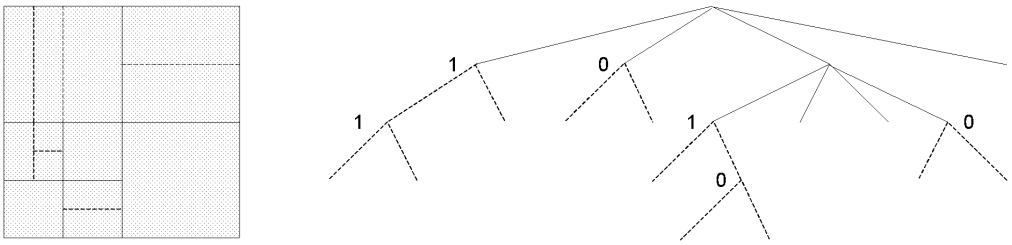
도면3


CTU 폭
CTU 높이
0
1
2

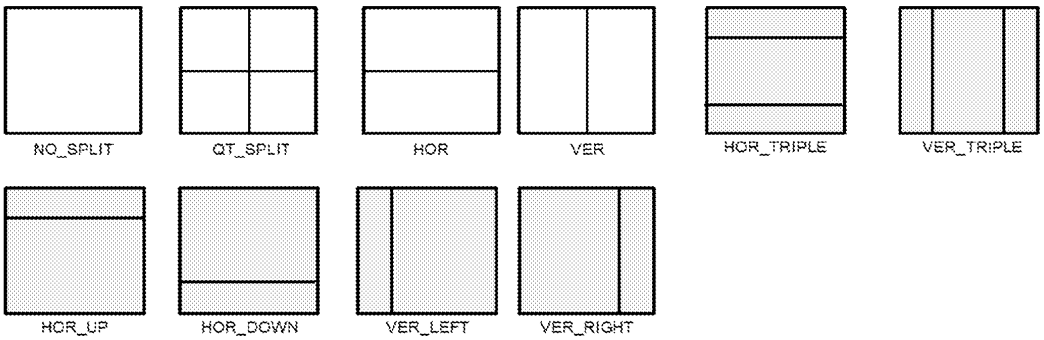
도면4


슬라이스
ctuAddr
CTU
PU
cuIdx puIdx
tuIdx puPartIdx
TU, PU 파티션

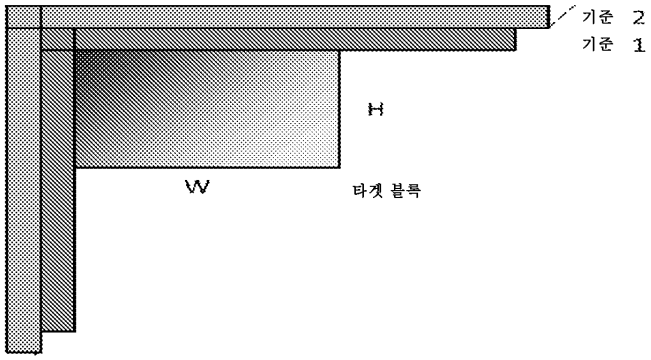
도면5


1
0
1
1
0
0

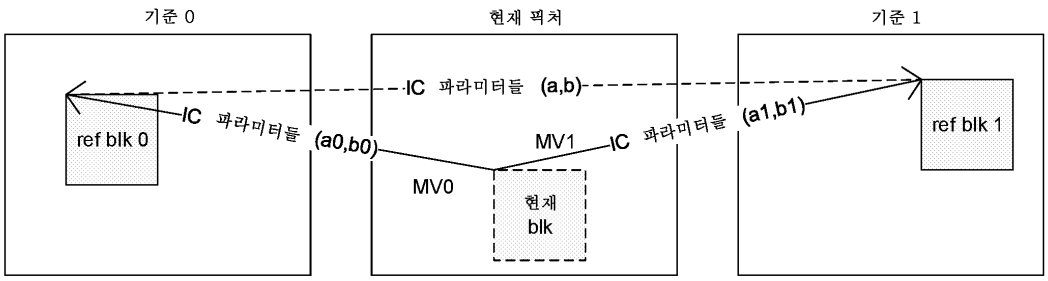
도면6


NO_SPLIT
QT_SPLIT
HOR
VER
HOR_TRIPLE
VER_TRIPLE
HOR_UP
HOR_DOWN
VER_LEFT
VER_RIGHT

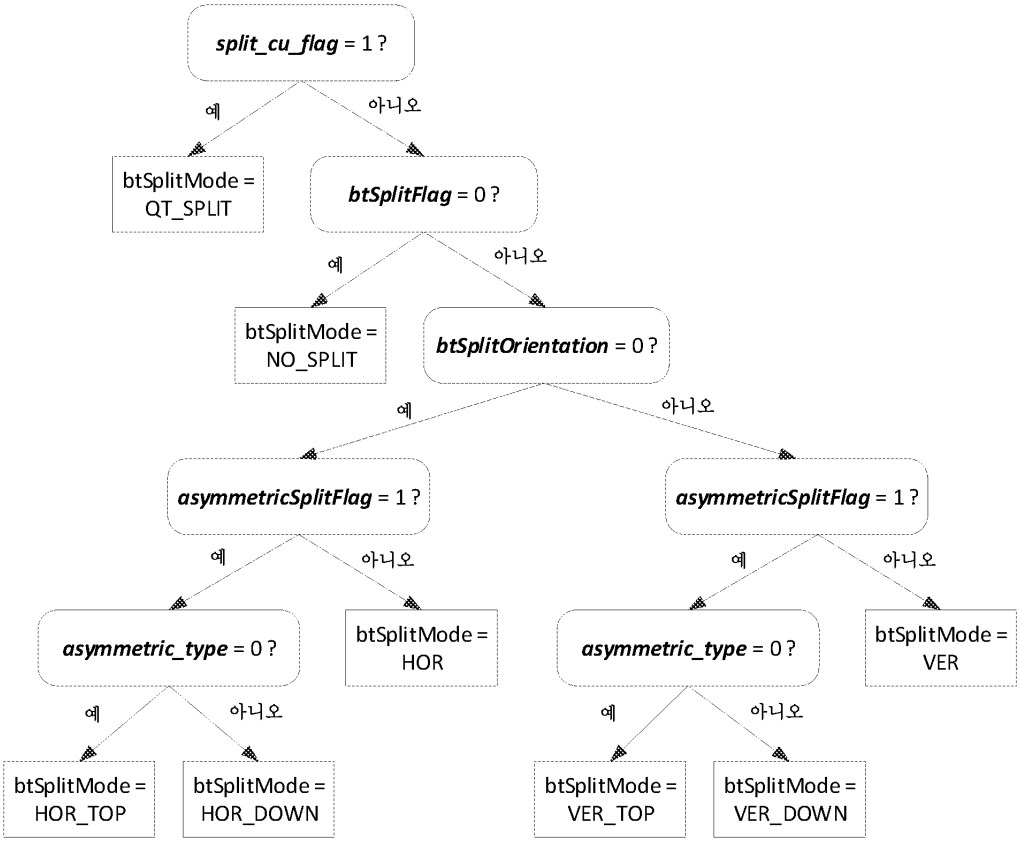
도면7


기준 2
기준 1
H
W
타겟 블록

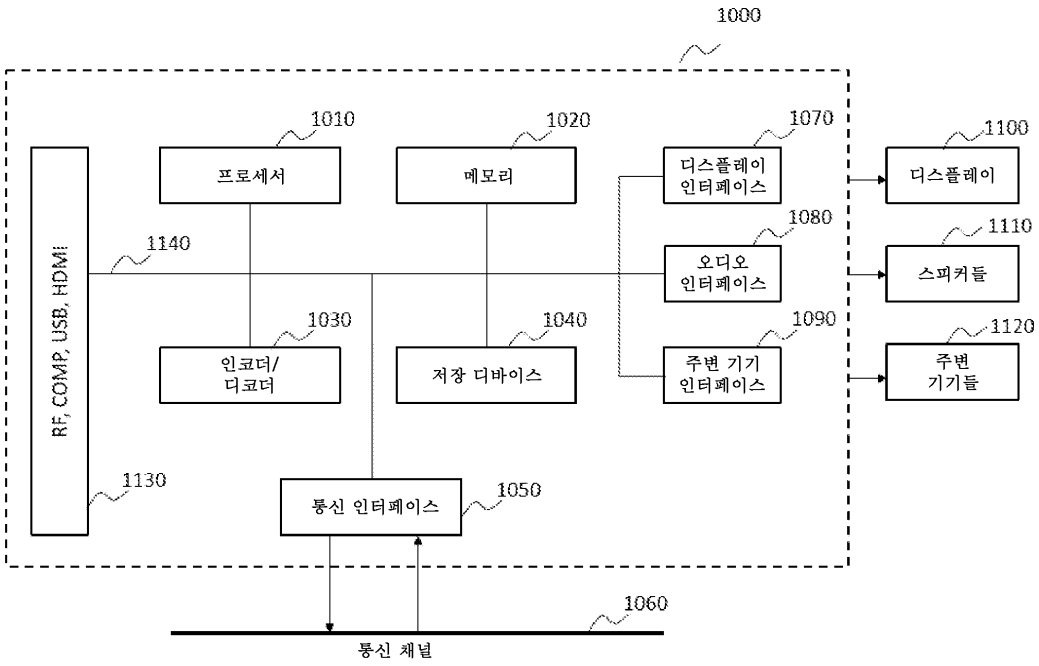
도면8


기준 0
현재 픽처
기준 1
IC 파라미터들 (a,b)
IC 파라미터들 (a0,b0)
IC 파라미터들 (a1,b1)
ref blk 0
MV1
MV0
현재 blk
ref blk 1

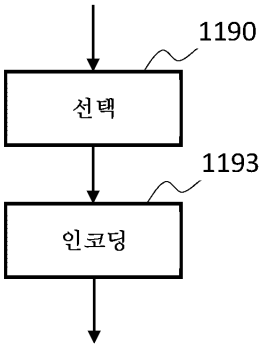
도면9


split_cu_flag = 1 ?
예
아니오
btSplitMode = QT_SPLIT
btSplitFlag = 0 ?
예
아니오
btSplitMode = NO_SPLIT
btSplitOrientation = 0 ?
예
아니오
asymmetricSplitFlag = 1 ?
asymmetricSplitFlag = 1 ?
예
아니오
예
아니오
asymmetric_type = 0 ?
btSplitMode = HOR
asymmetric_type = 0 ?
btSplitMode = VER
예
아니오
예
아니오
btSplitMode = HOR_TOP
btSplitMode = HOR_DOWN
btSplitMode = VER_TOP
btSplitMode = VER_DOWN

도면10


1000
1010
프로세서
1020
메모리
1070
디스플레이 인터페이스
1100
디스플레이
1080
오디오 인터페이스
1110
스피커들
1140
1030
인코더/ 디코더
1040
저장 디바이스
1090
주변 기기 인터페이스
1120
주변 기기들
RF, COMP, USB, HDMI
1130
통신 인터페이스
1050
1060
통신 채널

도면11



도면12