(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0133337
(43) 공개일자 2022년10월04일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) *H04N 19/124* (2014.01)
H04N 19/13 (2014.01) *H04N 19/174* (2014.01)
H04N 19/196 (2014.01) *H04N 19/40* (2014.01)
H04N 19/44 (2014.01) *H04N 19/46* (2014.01)
H04N 19/82 (2014.01) *H04N 19/91* (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/124 (2015.01)
(21) 출원번호 10-2022-7033473(분할)
(22) 출원일자(국제) 2015년03월09일
심사청구일자 없음
(62) 원출원 특허 10-2022-7001931
원출원일자(국제) 2015년03월09일
심사청구일자 2022년02월11일
(85) 번역문제출일자 2022년09월26일
(86) 국제출원번호 PCT/US2015/019512
(87) 국제공개번호 WO 2015/142556
국제공개일자 2015년09월24일
(30) 우선권주장
61/953,922 2014년03월16일 미국(US)
62/103,916 2015년01월15일 미국(US)

(71) 출원인
브이아이디 스케일, 인크.
미국 델라웨어 19809, 윌밍턴, 벨뷰 파크웨이 200, 스위트 300
(72) 발명자
예 얀
미국 캘리포니아주 92130 샌 디에고 펄먼 웨이 5001
시우 시아오유
미국 캘리포니아주 92122 샌 디에고 수트 #7322 주디셜 드라이브 9120
헤 유웬
미국 캘리포니아주 92130 샌 디에고 실버 바인 패쓰 13542
(74) 대리인
김태홍, 김진희

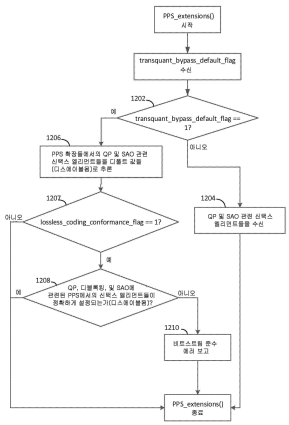
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 **무손실 비디오 코딩의 시그널링을 위한 방법 및 장치**

(57) 요 약

무손실 코딩이 이용된다는 것을 표시하는 하이-레벨 시그널링 무손실 코딩 신택스 엘리먼트를 포함하는 비디오 데이터 비트 스트림을 생성하고 디코딩하기 위한 시스템들 및 방법들이 설명된다. 하이-레벨 시그널링 신택스는 비디오 파라미터 세트(VPS), 시퀀스 파라미터 세트(SPS), 픽처 파라미터 세트(PPS), 또는 슬라이스 세그먼트 헤더 중의 하나이다. 무손실 코딩 신택스 엘리먼트는 양자화, 변환, 변환 스킵, 변환 스킵 회전, 및 인-루프 필터링 프로세스들에 관련된 하나 이상의 SPS, PPS, 및 슬라이스 세그먼트 헤더 신택스 엘리먼트들을 생성하기 위한 조건으로서 이용될 수도 있다.

대 표 도 - 도12



(52) CPC특허분류
H04N 19/13 (2015.01)
H04N 19/174 (2015.01)
H04N 19/196 (2015.01)
H04N 19/40 (2015.01)
H04N 19/44 (2015.01)
H04N 19/46 (2015.01)
H04N 19/82 (2015.01)
H04N 19/91 (2015.01)

명 세 서

청구범위

청구항 1

비디오 인코딩 방법으로서,

비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시를 시그널링하는 단계 - 상기 제1 블록은 인터 코딩으로 예측됨 - ;

상기 제1 블록을 16x16보다 작은 복수의 변환 유닛들로 분리하는 단계; 및

상기 복수의 변환 유닛들을 사용하여 인터 코딩으로 상기 제1 블록을 무손실 인코딩하는 단계

를 포함하는, 비디오 인코딩 방법.

청구항 2

제1항에 있어서, 상기 분리하는 단계는 상기 제1 블록을 8x8보다 작은 변환 유닛들로 분리하는 단계를 포함하는 것인, 비디오 인코딩 방법.

청구항 3

제1항 또는 제2항에 있어서, 상기 제1 블록에 대해 쿼드트리 분리 플래그가 시그널링되지 않는 것인, 비디오 인코딩 방법.

청구항 4

제1항 또는 제2항에 있어서, 비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시는 픽처-레벨 신택스 엘리먼트로 시그널링되는 것인, 비디오 인코딩 방법.

청구항 5

비디오 디코딩 방법에 있어서,

비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시를 수신하는 단계 - 상기 제1 블록은 인터 코딩으로 예측됨 - ;

상기 제1 블록을 16x16보다 작은 복수의 변환 유닛들로 분리하는 단계; 및

상기 복수의 변환 유닛들을 사용하여 인터 코딩으로 상기 제1 블록을 무손실 디코딩하는 단계

를 포함하는, 비디오 디코딩 방법.

청구항 6

제5항에 있어서, 상기 분리하는 단계는 상기 제1 블록을 8x8보다 작은 변환 유닛들로 분리하는 단계를 포함하는 것인, 비디오 디코딩 방법.

청구항 7

제5항 또는 제6항에 있어서, 상기 제1 블록에 대해 쿼드트리 분리 플래그가 수신되지 않는 것인, 비디오 디코딩 방법.

청구항 8

제5항 또는 제6항에 있어서, 비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시는 픽처-레벨 신택스 엘리먼트로 수신되는 것인, 비디오 디코딩 방법.

청구항 9

하나 이상의 프로세서로 하여금 제1항, 제2항, 제5항, 및 제6항 중 어느 한 항의 방법을 수행하도록 하기 위한 명령어들을 포함한 컴퓨터 판독가능 매체.

청구항 10

하나 이상의 프로세서를 포함한 비디오 인코더로서, 상기 하나 이상의 프로세서는,

비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시를 시그널링하고 - 상기 제1 블록은 인터 코딩으로 예측됨 - ;

상기 제1 블록을 16x16보다 작은 복수의 변환 유닛들로 분리하며;

상기 복수의 변환 유닛들을 사용하여 인터 코딩으로 상기 제1 블록을 무손실 인코딩하는 것을

수행하도록 구성되는 것인, 비디오 인코더.

청구항 11

제10항에 있어서, 상기 분리는 상기 제1 블록을 8x8보다 작은 변환 유닛들로 분리하는 것을 포함하는 것인, 비디오 인코더.

청구항 12

제10항 또는 제11항에 있어서, 상기 제1 블록에 대해 쿼드트리 분리 플래그가 시그널링되지 않는 것인, 비디오 인코더.

청구항 13

제10항 또는 제11항에 있어서, 비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시는 픽처-레벨 신택스 엘리먼트로 시그널링되는 것인, 비디오 인코더.

청구항 14

하나 이상의 프로세서를 포함한 비디오 디코더로서, 상기 하나 이상의 프로세서는,

비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시를 수신하고 - 상기 제1 블록은 인터 코딩으로 예측됨 - ;

상기 제1 블록을 16x16보다 작은 복수의 변환 유닛들로 분리하며;

상기 복수의 변환 유닛들을 사용하여 인터 코딩으로 상기 제1 블록을 무손실 디코딩하는 것을

수행하도록 구성되는 것인, 비디오 디코더.

청구항 15

제14항에 있어서, 상기 분리는 상기 제1 블록을 8x8보다 작은 변환 유닛들로 분리하는 것을 포함하는 것인, 비디오 디코더.

청구항 16

제14항 또는 제15항에 있어서, 상기 제1 블록에 대해 쿼드트리 분리 플래그가 수신되지 않는 것인, 비디오 디코더.

청구항 17

제14항 또는 제15항에 있어서, 비디오의 적어도 제1 블록이 무손실 코딩으로 코딩된다는 표시는 픽처-레벨 신택스 엘리먼트로 수신되는 것인, 비디오 디코더.

발명의 설명

기 술 분 야

[0001] **관련 출원들에 대한 상호-참조**

[0002] 본 출원은 2014년 3월 16일자로 출원된 미국 특허 가출원 제61/953,922호, 및 2015년 1월 15일자로 출원된 미국 특허 가출원 제62/103,916호의 정규 출원이며, 이 가출원들로부터 35 U.S.C. §119(e) 하의 이익을 주장한다. 이 출원들의 내용들은 그 전체적으로 참조로 본원에 편입된다.

배 경 기 술

[0003] 과거 20년 동안, 효율적인 디지털 비디오 통신, 분배, 및 소비를 가능하게 하기 위하여, 다양한 디지털 비디오 압축 기술들이 개발되고 표준화되었다. H.261, MPEG-1, MPEG-2 H.263, MPEG-4 (part-2), 및 H.264/AVC(MPEG-4 part 10 Advancd Video Coding)과 같이, 상업적으로 널리 전개된 표준들의 대부분은 ISO/IEC 및 ITU-T에 의해 개발되고 있다. 새롭게 진보된 비디오 압축 기술들, 새로운 비디오 코딩 표준, ITU-T 비디오 코딩 전문가 그룹(Video Coding Experts Group; VCEG) 및 ISO/IEC MPEG에 의한 공동 개발 중인 고효율 비디오 코딩(High Efficiency Video Coding; HEVC)의 출현 및 성숙으로 인해, HEVC(ITU-T H.265/ISO/IEC 23008-2)는 2013년 초창기에 국제 표준으로서 승인되었고, 현재의 최신 H.264/AVC보다 실질적으로 더 높은 코딩 효율을 달성할 수 있다.

[0004] (위성, 케이블, 지상 송신 채널들을 통해 TV 신호들을 전송하는 것과 같은) 전통적인 디지털 비디오 서비스들에 비해, IPTV, 비디오 채팅, 이동 비디오, 및 스트리밍 비디오와 같은 더욱 더 새로운 비디오 애플리케이션들이 이종 환경들에서 전개된다. 이러한 이종성은 클라이언트들 상에서 뿐만 아니라 네트워크에서도 존재한다. 클라이언트 측 상에서는, N-스크린(N-screen) 시나리오, 즉, 스마트폰, 태블릿, PC, 및 TV를 포함하는 다양한 스크린 크기들 및 디스플레이 능력들을 갖는 디바이스들 상에서 비디오 컨텐츠를 소비하는 것이 이미 시장을 지배하고 있고, 그렇게 계속할 것으로 예상된다. 네트워크 측 상에서는, 비디오가 인터넷, WiFi 네트워크들, 이동(3G 및 4G) 네트워크들, 및/또는 이들의 임의의 조합을 걸쳐 송신되고 있다.

발명의 내용

[0005] 무손실 인코딩의 하이-레벨(high-level) 시그널링에 관련된 시스템들 및 방법들이 본원에서 설명된다. 일부의 실시형태들에서, 방법은 무손실 코딩이 이용되고 있다는 것을 표시하는 하이-레벨 신택스 엘리먼트(syntax element)를 포함하는 비디오 데이터 비트 스트림을 생성하는 단계를 포함한다. 하이-레벨 시그널링 신택스는 픽처 파라미터 세트(picture parameter set; PPS), 시퀀스 파라미터 세트(Sequence Parameter Set; SPS), 비디오 파라미터 세트(Video Parameter Set; VPS), 또는 슬라이스 세그먼트 헤더(slice segment header) 중의 하나일 수도 있다. 무손실 코딩 신택스 엘리먼트는 양자화, 변환, 변환 스킵(transform skip), 변환 스킵 회전(transform skip rotation), 및 인-루프 필터링(in-loop filtering) 프로세스들에 관련된 하나 이상의 SPS 신택스 엘리먼트들을 생성하기 위한 조건으로서 이용될 수도 있다.

[0006] 일부의 실시형태들에서, 방법은 transquant_bypass_default_flag를 포함하는 픽처 파라미터 세트(PPS)를 생성하는 단계를 포함한다. transquant_bypass_default_flag는 1로 설정되어, PPS를 참조하는 슬라이스에서의 모든 코딩 유닛들의 cu_transquant_bypass_flag의 디폴트 값(default value)을 표시한다. PPS는 또한, 0과 동일한 transquant_bypass_enabled_flag를 가질 수도 있다.

[0007] 디코더 측에서, 역양자화, 역변환, 디블록킹 필터(deblocking filter), 샘플 적응적 오프셋(sample adaptive offset; SAO) 등등을 포함하는 복수의 프로세싱 블록들은 무손실 코딩이 적용될 때에 우회될 수도 있다. 그러므로, 디코더가 하이-레벨 무손실 코딩 표시를 수신할 경우, 디코더는 코딩 유닛(coding unit; CU)들의 많은 그룹이 이 프로세싱 블록들을 필요로 하지 않을 것으로 결정한다. 디코딩에 앞서서 이 프로세싱 블록들을 정지시키는 것은 전력 감소, 프로세싱 사이클 감소, 더욱 양호한 부하 제공 등의 측면에서 유리할 수도 있다.

[0008] 따라서, 일부의 실시형태들에서, 방법은 비디오 디코더에서, 하이 레벨 무손실 코딩 표시를 수신하는 단계, 및 이에 응답하여, 복수의 프로세싱 블록들을 정지시키는 단계를 포함한다. 복수의 프로세싱 블록들은 다음의 하드웨어 블록들: 역양자화, 역변환, 디블록킹 필터, 및/또는 SAO의 임의의 것 중의 하나 이상을 포함할 수도 있다. 또한, 프로세싱 블록들은 비디오 코딩에 앞서서 정지될 수도 있어서, 프로세싱 블록 하드웨어 컴포넌트(component)들의 적어도 부분의 전력 소비에 있어서 감소를 야기시킬 수도 있다.

도면의 간단한 설명

[0009] 동반된 도면들과 함께 예로서 제시된 다음의 설명으로부터, 더욱 상세한 이해가 행해질 수도 있다.

도 1a는 블록-기반 비디오 인코더의 예를 예시하는 블록도이다.

도 1b는 무손실 블록-기반 비디오 인코더의 예를 예시하는 블록도이다.

도 2a는 블록-기반 비디오 디코더의 예를 예시하는 블록도이다.

도 2b는 무손실 블록-기반 비디오 디코더의 예를 예시하는 블록도이다.

도 3은 8 개의 방향성 예측 모드(directional prediction mode)들의 예의 도면이다.

도 4는 33 개의 방향성 예측 모드들 및 2 개의 비-방향성(non-directional) 예측 모드들의 예를 예시하는 도면이다.

도 5는 수평 예측의 예의 도면이다.

도 6은 평면 모드의 예의 도면이다.

도 7은 모션 예측의 예를 예시하는 도면이다.

도 8은 픽처 내에서의 블록-레벨 이동의 예를 예시하는 도면이다.

도 9는 일 예의 통신 시스템을 예시하는 도면이다.

도 10은 일 예의 무선 송수신 유닛 (wireless transmit/receive unit; WTRU)을 예시하는 도면이다.

도 11은 코딩된 비트스트림 구조의 예를 예시하는 도면이다.

도 12는 플래그 transquant_bypass_default_flag를 이용하여 수정된 PPS 확장들 신택스를 디코딩하는 방법의 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0010] 예시적인 실시형태들의 상세한 설명은 다양한 도면들을 참조하여 지금부터 제공될 것이다. 이 설명은 가능한 구현예들의 상세한 예들을 제공하지만, 제공된 세부사항들은 예시적이도록 의도된 것이며 결코 적용의 범위를 제한하도록 의도된 것이 아니라는 것에 주목해야 한다.

[0011] **비디오 인코딩 및 디코딩**

[0012] 도 1a는 블록-기반 인코더, 예를 들어, 하이브리드 비디오 인코딩 시스템의 예를 예시하는 블록도이다. 비디오 인코더(100)는 입력 비디오 신호(102)를 수신할 수도 있다. 입력 비디오 신호(102)는 블록 대 블록으로 프로세싱될 수도 있다. 비디오 블록은 임의의 크기일 수도 있다. 예를 들어, 비디오 블록 유닛은 16x16 픽셀들을 포함할 수도 있다. 16x16 픽셀들의 비디오 블록 유닛은 매크로블록(macroblock; MB)으로서 지칭될 수도 있다. 고효율 비디오 코딩(HEVC)에서는, (예컨대, 2 개의 용어들이 우리의 목적들에 대하여 동등한, 코딩 트리 유닛(coding tree unit; CTU) 또는 코딩 유닛(coding unit; CU)으로서 지칭될 수도 있는) 확장된 블록 크기들은 높은 해상도(예컨대, 1080p 이상)의 비디오 신호들을 효율적으로 압축하기 위하여 이용될 수도 있다. HEVC에서, CU는 64x64 픽셀들까지 일 수도 있다. CU는, 별도의 예측 방법들이 적용될 수도 있는 예측 유닛(prediction unit; PU)들로 파티셔닝(partitioning)될 수도 있다.

[0013] 입력 비디오 블록(예컨대, MB 또는 CU)에 대하여, 공간적 예측(160) 및/또는 시간적 예측(162)이 수행될 수도 있다. 공간적 예측(예컨대, "인트라 예측(intra prediction)")은 현재의 비디오 블록을 예측하기 위하여, 동일한 비디오 픽처/슬라이스에서의 이미 코딩된 이웃하는 블록들로부터의 픽셀들을 이용할 수도 있다. 공간적 예측은 비디오 신호에서 고유한 공간적 중복성을 감소시킬 수도 있다. 시간적 예측(예컨대, "인터 예측(inter prediction)" 또는 "모션 보상된 예측")은 현재의 비디오 블록을 예측하기 위하여, (예컨대, "참조 픽처들"로서 지칭될 수도 있는) 이미 코딩된 비디오 픽처들로부터의 픽셀들을 이용할 수도 있다. 시간적 예측은 비디오 신호에서 고유한 시간적 중복성을 감소시킬 수도 있다. 비디오 블록에 대한 시간적 예측 신호는 현재의 블록과, 참조 픽처에서의 그 예측 블록과의 사이의 모션(motion)의 양 및/또는 방향을 표시할 수도 있는 하나 이상의 모션 벡터들에 의해 시그널링될 수도 있다. (예컨대, H.264/AVC 및/또는 HEVC에 대하여 그러할 수도 있는 바와 같이) 다수의 참조 픽처들이 지원될 경우, 비디오 블록에 대하여, 그 참조 픽처 인덱스가 전송될 수도 있다. 참조 픽처 인덱스는 시간적 예측 신호가 참조 픽처 저장소(164)에서의 어느 참조 픽처로부터 나오는지를 식별하기 위하여 이용될 수도 있다.

[0014] 인코더에서의 모드 결정 블록(180)은 예를 들어, 공간적 및/또는 시간적 예측 후에 예측 모드를 선택할 수도 있다. 예측 블록은 116에서 현재의 비디오 블록으로부터 감산(subtract)될 수도 있다. 예측 잔차(prediction residual)는 변환(104) 및/또는 양자화(106)될 수도 있다. 양자화 블록(106)은 예측 잔차를 코딩하기 위하여 필요한 비트들의 수를 효과적으로 감소시킬 수 있다. 양자화 파라미터(Quantization Parameter; QP)는 양자화의 극심도(severity)를 제어하기 위하여 이용될 수도 있다. QP 값이 증가함에 따라, 더욱 극심한 양자화가 적용될 수도 있고; 그 결과, 코딩된 비디오 비트 레이트가 감소될 수도 있고, 이와 동시에, 디코딩된 비디오 품질이 열화될 수도 있다. 양자화로 인한 보편적으로 알려진 시각적 아티팩트(visual artifact)들은 블록킹 아티팩트(blocking artifact)들, 블러링(blurring), 스미어링(smearing), 링잉(ringing), 플리커링(flickering) 등등을 포함한다. 도 1a 및 도 2a에서 예시된 비디오 코딩 시스템에서의 다른 프로세싱 블록들은, 특히, 이 프로세싱 블록들이 프로세싱 파이프라인에서 중간 데이터의 비트 심도(bit depth)에 대한 상한을 요구하는 고정-소수점 동작들을 적용할 때에 정보 손실을 또한 야기시킬 수도 있다. 예를 들어, 변환 블록(104)에서는, 수평 방향에서의 변환들이 먼저, 그 다음으로, 수직 방향에서의 변환들이 적용될 수도 있다. 변환은 (승산(multiplication)들로 인해) 데이터 비트 심도를 증가시킬 수도 있으므로, 수평 변환 후에, 수직 변환에 대한 입력 데이터 비트 심도를 감소시키기 위하여, 우측 시프팅(right shifting)이 수평 변환의 출력에 적용될 수도 있다. 이러한 시프팅 동작들은 (데이터 비트 심도를 감소시킴으로써) 구현 코스트를 감소시키는 것을 도울 수도 있지만, 이들은 또한, 프로세싱 파이프라인에서 정보 손실을 야기시킬 수도 있다. 추가적으로, 고정-소수점 동작들을 가능하게 하기 위하여, H.264/AVC 및 HEVC와 같은 최신 비디오 표준들에서의 변환들은 정수 값 변환들이다. 이 정수 변환들의 일부는 직교적(orthogonal)인 것에 근접할 수도 있지만, 완전히 직교적이지 않을 수도 있다. 변환(및 역변환) 행렬들이 완전히 직교적이지 않을 경우, 이들은 완벽한 재구성을 보장할 수 없다. 다시 말해서, 심지어 임의의 양자화 없이도, 비-직교적 변환 및 역변환이 입력 데이터 블록에 적용된 후, 출력 데이터 블록(스케일링 인수(scaling factor)가 출력에 적용될 수도 있음)은 입력 데이터 블록과 수학적으로 동일하게 유지되지 않을 수도 있다.

[0015] 양자화된 잔차 계수들은, 재구성된 비디오 블록을 형성하기 위하여 예측 블록(126)에 다시 가산될 수도 있는 재구성된 잔차를 형성하기 위하여 역양자화(110) 및/또는 역변환(112)될 수도 있다.

[0016] 인-루프 필터링(예컨대, 디블록킹 필터, 샘플 적응적 오프셋, 적응적 루프 필터, 및/또는 등등)은, 재구성된 비디오 블록이 참조 픽처 저장소(164)에 넣어지고 및/또는 미래의 비디오 블록들을 코딩하기 위하여 이용되기 전에, 재구성된 비디오 블록에 적용(166)될 수도 있다. 비디오 인코더(100)는 출력 비디오 스트림(120)을 출력할 수도 있다. 출력 비디오 비트스트림(120)을 형성하기 위하여, 코딩 모드(예컨대, 인터 예측 모드 또는 인트라 예측 모드), 예측 모드 정보, 모션 정보, 및/또는 양자화된 잔차 계수들은 비트스트림을 형성하기 위하여 압축 및/또는 팩킹(packing)되도록, 엔트로피 코딩 유닛(108)으로 전송될 수도 있다. 참조 픽처 저장소(164)는 디코딩된 픽처 버퍼(decoded picture buffer; DPB)로서 지칭될 수도 있다.

[0017] 도 2a는 블록-기반 비디오 디코더의 예를 예시하는 블록도이다. 비디오 디코더(200)는 비디오 비트스트림(202)을 수신할 수도 있다. 비디오 비트스트림(202)은 엔트로피 디코딩 유닛(208)에서 언팩킹(unpacking) 및/또는 엔트로피 디코딩될 수도 있다. 비디오 비트스트림을 인코딩하기 위하여 이용된 코딩 모드 및/또는 예측 정보는 예측 블록을 형성하기 위하여 시간적 예측 유닛(260)(예컨대, 인트라 코딩될 경우) 및/또는 시간적 예측 유닛(262)(예컨대, 인터 코딩될 경우)으로 전송될 수도 있다. 인터 코딩될 경우, 예측 정보는 예측 블록 크기들, (예컨대, 모션의 방향 및 양을 표시할 수도 있는) 하나 이상의 모션 벡터들, 및/또는 (예컨대, 예측 신호가 어느 참조 픽처로부터 획득되어야 하는지를 표시할 수도 있는) 하나 이상의 참조 인덱스들을 포함할 수도 있다.

[0018] 모션 보상된 예측은 시간적 예측 블록을 형성하기 위하여 시간적 예측 유닛(262)에 의해 적용될 수도 있다. 잔차 변환 계수들은 잔차 블록을 재구성하기 위하여, 역양자화 유닛(210) 및 역변환 유닛(212)으로 전송될 수도 있다. 예측 블록 및 잔차 블록은 226에서 함께 가산될 수도 있다. 재구성된 블록은, 그것이 참조 픽처 저장소(264) 내에 저장되기 전에, 루프 필터(266)에 의한 인-루프 필터링을 거칠 수도 있다. 참조 픽처 저장소(264)에서의 재구성된 비디오는 디스플레이 디바이스를 구동하기 위하여 이용될 수도 있고, 및/또는 미래의 비디오 블록들을 예측하기 위하여 이용될 수도 있다. 비디오 디코더(200)는 재구성된 비디오 신호(220)를 출력할 수도 있다. 참조 픽처 저장소(264)는 또한, 디코딩된 픽처 버퍼(DPB)로서 지칭될 수도 있다.

[0019] 비디오 인코더 및/또는 디코더(예컨대, 비디오 인코더(100) 또는 비디오 디코더(200))는 (예컨대, 인트라 예측으로서 지칭될 수도 있는) 공간적 예측을 수행할 수도 있다. 공간적 예측은 (예컨대, 방향성 인트라 예측으로서 지칭될 수도 있는) 복수의 예측 방향들 중의 하나를 따르는 이미 코딩된 이웃하는 픽셀들로부터 예측함으로

써 수행될 수도 있다.

[0020] 도 3은 8 개의 방향성 예측 모드들의 예의 도면이다. 도 3의 8 개의 방향성 예측 모드들은 H.264/AVC에서 지원될 수도 있다. (DC 모드 2를 포함하는) 9 개의 모드들은:

[0021] · 모드 0: 수직 예측.

[0022] · 모드 1: 수평 예측.

[0023] · 모드 2: DC 예측.

[0024] · 모드 3: 대각 하부-좌측 예측.

[0025] · 모드 4: 대각 하부-우측 예측

[0026] · 모드 5: 수직-우측 예측.

[0027] · 모드 6: 수평-하부 예측

[0028] · 모드 7: 수직-좌측 예측.

[0029] · 모드 8: 수평-상부 예측.

[0030] 공간적 예측은 다양한 크기들 및/또는 형상들의 비디오 블록에 대해 수행될 수도 있다. 비디오 신호의 루마 컴포넌트(luma component)의 공간적 예측은 예를 들어, (예컨대, H.264/AVC에서의) 4x4, 8x8, 및 16x16 픽셀들의 블록 크기들에 대하여 수행될 수도 있다. 비디오 신호의 루마 컴포넌트(luma component)의 공간적 예측은 예를 들어, (예컨대, H.264/AVC에서의) 8x8의 블록 크기들에 대하여 수행될 수도 있다. 크기 4x4 또는 8x8의 루마 블록(luma block)에 대하여, 총 9 개의 예측 모드들, 예를 들어, (예컨대, H.264/AVC에서의) 8 개의 방향성 예측 모드들 및 DC 모드가 지원될 수도 있다. 4 개의 예측 모드들이 지원될 수도 있다: 크기 16x16의 루마 블록에 대하여, 예를 들어, 수평, 수직, DC, 및 평면 예측.

[0031] 방향성 인트라 예측 모드들 및 비-방향성 예측 모드들이 지원될 수도 있다. 도 4는 33 개의 방향성 예측 모드들 및 2 개의 비-방향성(non-directional) 예측 모드들의 예를 예시하는 도면이다. 도 4의 33 개의 방향성 예측 모드들 및 2 개의 비-방향성 예측 모드들은 HEVC에 의해 지원될 수도 있다. 더 큰 블록 크기들을 이용한 공간적 예측이 지원될 수도 있다. 예를 들어, 공간적 예측은 예를 들어, 4x4, 8x8, 16x16, 32x32, 또는 64x64의 정사각형 블록 크기들의 임의의 크기의 블록에 대해 수행될 수도 있다. (예컨대, HEVC에서의) 방향성 인트라 예측은 1/32-픽셀 정밀도로 수행될 수도 있다.

[0032] 비-방향성 인트라 예측 모드들은 예를 들어, 방향성 인트라 예측에 추가하여, (예컨대, H.264/AVC, HEVC 등등에서) 지원될 수도 있다. 비-방향성 인트라 예측 모드들은 DC 모드 및/또는 평면 모드를 포함할 수도 있다. DC 모드에 대하여, 예측 값은 이용가능한 이웃하는 픽셀들을 평균화함으로써 획득될 수도 있고, 예측 값은 전체의 블록에 균일하게 적용될 수도 있다. 평면 모드에 대하여, 선형 보간(linear interpolation)은 느린 전환(transition)들을 갖는 평탄한 영역들을 예측하기 위하여 이용될 수도 있다. H.264/AVC는 16x16 루마 블록들 및 크로마 블록(chroma block)들에 대한 평면 모드의 이용을 허용할 수도 있다.

[0033] 인코더(예컨대, 인코더(100))는 비디오 블록에 대한 최상의 코딩 모드를 결정하기 위하여 (예컨대, 도 1a에서의 블록(180)에서) 모드 결정을 수행할 수도 있다. 인코더가 (예컨대, 인터 예측 대신에) 인트라 예측을 적용할 것을 결정할 때, 인코더는 이용가능한 모드들의 세트(set)로부터 최적의 인트라 예측 모드를 결정할 수도 있다. 선택된 방향성 인트라 예측 모드는 입력 비디오 블록에서의 임의의 텍스처(texture), 에지(edge), 및/또는 구조의 방향에 대한 강력한 힌트들을 제공할 수도 있다. 도 5는 (예컨대, 4x4 블록에 대한) 수평 예측의 예의 도면이다. 이미 제구성된 픽셀들 P0, P1, P2, 및 P3(예컨대, 음영표시된 박스들)은 현재의 4x4 비디오 블록에서의 픽셀들을 예측하기 위하여 이용될 수도 있다. 수평 예측에서는, 재구성된 픽셀, 예를 들어, 픽셀들 P0, P1, P2, 및/또는 P3이 4x4 블록을 예측하기 위하여 대응하는 행(row)의 방향을 따라 수평으로 전파될 수도 있다. 예를 들어, 예측은 이하의 수학식 (1)에 따라 수행될 수도 있고, 여기서, L(x, y)는 (x, y), x, y = 0 ... 3에서 예측되어야 할 픽셀일 수도 있다.

$$L(x,0) = P0$$
$$L(x,1) = P1$$
$$L(x,2) = P2 \quad (1)$$
$$L(x,3) = P3$$

[0034]

[0035] 도 6은 평면 모드의 예의 도면이다. 평면 모드는 이에 따라 수행될 수도 있다. (예컨대, T에 의해 표기된) 상부 행에서의 가장 우측 픽셀은 가장 우측 열에서의 픽셀들을 예측하기 위하여 복제될 수도 있다. (예컨대, L에 의해 표기된) 좌측 열에서의 하부 픽셀은 하부 행에서의 픽셀들을 예측하기 위하여 복제될 수도 있다. (예컨대, 좌측 블록에서 도시된 바와 같은) 수평 방향에서의 2중 선형 보간(bilinear interpolation)은 중심 픽셀들의 제 1 예측 H(x, y)를 생성하기 위하여 수행될 수도 있다. (예컨대, 우측 블록에서 도시된 바와 같은) 수직 방향에서의 2중 선형 보간은 중심 픽셀들의 제 2 예측 V(x, y)를 생성하기 위하여 수행될 수도 있다. 수평 예측과 수직 예측 사이의 평균화는 L(x, y) = ((H(x, y)+V(x, y))>>1)를 이용하여 최종 예측 L(x, y)을 획득하기 위하여 수행될 수도 있다.

[0036] 도 7 및 도 8은 (예컨대, 도 1a의 모션 예측 유닛(162)을 이용한) 비디오 블록들의 모션 예측의 예를 예시하는 도면들이다. 도 8은 예를 들어, 참조 픽처들 "Ref pic 0", "Ref pic 1", 및 "Ref pic2"을 포함하는 일 예의 디코딩된 픽처 버퍼를 예시하는 도면이다. 현재의 픽처에서의 블록들 B0, B1, 및 B2는 참조 픽처들 "Ref pic 0", "Ref pic 1", 및 "Ref pic2"에서의 블록들로부터 각각 예측될 수도 있다. 모션 예측은 현재의 비디오 블록을 예측하기 위하여 이웃하는 비디오 프레임들로부터의 비디오 블록들을 이용할 수도 있다. 모션 예측은 시간적 상관(temporal correlation)을 활용할 수도 있고, 및/또는 비디오 신호에서 고유한 시간적 중복성을 제거할 수도 있다. 예를 들어, H.264/AVC 및 HEVC에서, 시간적 예측은 다양한 크기들의 비디오 블록들에 대해 수행될 수도 있다(예컨대, 루마 컴포넌트에 대하여, 시간적 예측 블록 크기들은 H.264/AVC에서 16x16으로부터 4x4로, 그리고 HEVC에서 64x64로부터 4x4로 변동될 수도 있음). (mvx, mvy)의 모션 벡터로, 시간적 예측은 수학식 (1)에 의해 제공된 바와 같이 수행될 수도 있다.

$$P(x,y) = ref(x - mvx, y - mvy) \quad (1)$$

[0037]

[0038] 여기서, ref(x, y)는 참조 픽처에서의 위치 (x, y)에서의 픽셀 값일 수도 있고, P(x, y)는 예측된 블록일 수도 있다. 비디오 코딩 시스템은 분수 픽셀 정밀도(fractional pixel precision)로 인터-예측을 지원할 수도 있다. 모션 벡터 (mvx, mvy)가 분수 픽셀 데이터를 가질 때, 하나 이상의 보간 필터들은 분수 픽셀 위치들에서의 픽셀 값들을 획득하기 위하여 적용될 수도 있다. 블록 기반 비디오 코딩 시스템들은 예를 들어, 예측 신호가 상이한 참조 픽처들로부터의 다수의 예측 신호들을 조합함으로써 형성될 수도 있을 경우에, 시간적 예측을 개선시키기 위하여 멀티-가설(multi-hypothesis) 예측을 이용할 수도 있다. 예를 들어, H.264/AVC 및/또는 HEVC는 2 개의 예측 신호들을 조합할 수도 있는 양-예측(bi-prediction)을 이용할 수도 있다. 양-예측은 다음의 수학식 (2)와 같은 예측을 형성하기 위하여, 각각 참조 픽처로부터의 2 개의 예측 신호들을 조합할 수도 있다.

$$P(x,y) = \frac{P_0(x,y) + P_1(x,y)}{2} = \frac{ref_0(x - mvx_0, y - mvy_0) + ref_1(x - mvx_1, y - mvy_1)}{2} \quad (2)$$

[0039]

[0040] 여기서, $P_0(x, y)$ 및 $P_1(x, y)$는 각각 제 1 및 제 2 예측 블록일 수도 있다. 수학식 (2)에서 예시된 바와 같이, 2 개의 예측 블록들은 2 개의 모션 벡터들 (mvx_0, mvy_0) 및 (mvx_1, mvy_1)를 각각 갖는 2 개의 참조 픽처들 $ref_0(x, y)$ 및 $ref_1(x, y)$로부터의 모션 보상된 예측을 수행함으로써 획득될 수도 있다. 예측 블록 P(x, y)는 예측 잔차 블록을 형성하기 위하여 (예컨대, 가산기(116)에서) 소스 비디오 블록으로부터 감산될 수도 있다. 예측 잔차 블록은 (예컨대, 변환 유닛(104)에서) 변환될 수도 있고, 및/또는 (예컨대, 양자화 유닛(106)에서) 양자화될 수도 있다. 양자화된 잔차 변환 계수 블록들은 비트 레이트를 감소시키기 위하여 엔트로피 코딩되도록, 엔트로피 코딩 유닛(예컨대, 엔트로피 코딩 유닛(108))으로 전송될 수도 있다. 엔트로피 코딩된 잔차 계수들은 출력 비디오 비트스트림(예컨대, 비트스트림(120))의 일부를 형성하기 위하여 팩킹될 수도 있다.

[0041] 도 11은 코딩된 비트스트림 구조의 예를 예시하는 도면이다. 코딩된 비트스트림(1000)은 다수의 NAL(Network Abstraction layer; 네트워크 추상화 계층) 유닛들(1001)로 구성된다. NAL 유닛은 코딩된 슬라이스(1006)와 같은 코딩된 샘플 데이터, 또는 파라미터 세트 데이터, 슬라이스 헤더 데이터(1005), 또는 (SEI 메시지로서 지

칭될 수도 있는) 보충 강화 정보(supplemental enhancement information) 데이터(1007)와 같은 하이 레벨 신택스 메타데이터를 포함할 수도 있다. 파라미터 세트들은, 다수의 비트스트림 계층들에 적용할 수도 있거나(예컨대, 비디오 파라미터 세트(1002)(VPS)), 또는 하나의 계층 내의 코딩된 비디오 시퀀스에 적용할 수도 있거나(예컨대, 시퀀스 파라미터 세트(1003)(SPS)), 또는 하나의 코딩된 비디오 시퀀스 내의 다수의 코딩된 픽처들에 적용할 수도 있는(예컨대, 픽처 파라미터 세트(1004)(PPS)) 필수적인 신택스 엘리먼트들을 포함하는 하이 레벨 신택스 구조들이다. 파라미터 세트들은 비디오 비트 스트림의 코딩된 픽처들과 함께 전송될 수 있거나, 또는 (신뢰성 있는 채널들, 하드 코딩 등을 이용한 대역외(out-of-band) 송신을 포함하는) 다른 수단을 통해 전송될 수 있다. 슬라이스 헤더(1005)는 또한, 상대적으로 작거나 어떤 슬라이스 또는 픽처 타입들에 대해서만 관련되는 일부의 픽처 관련 정보를 포함할 수도 있는 하이 레벨 신택스 구조이다. SEI 메시지들(1007)은, 디코딩 프로세스에 의해 네스팅되지 않을 수도 있지만, 픽처 출력 타이밍 또는 디스플레이 및/또는 손실 검출 및 은닉과 같은 다양한 다른 목적들을 위하여 이용될 수 있는 정보를 반송한다.

[0042] 도 9는 통신 시스템의 예를 예시하는 도면이다. 통신 시스템(1300)은 인코더(1302), 통신 네트워크(1304), 및 디코더(1306)를 포함할 수도 있다. 인코더(1302)는 접속(1308)을 통해 통신 네트워크(1304)와 통신할 수도 있다. 접속(1308)은 유선 접속 또는 무선 접속일 수도 있다. 인코더(1302)는 도 1a의 블록-기반 비디오 인코더와 유사할 수도 있다. 인코더(1302)는 단일 계층 코덱(예컨대, 도 1a) 또는 멀티계층 코덱을 포함할 수도 있다.

[0043] 디코더(1306)는 접속(1310)을 통해 통신 네트워크(1306)와 통신할 수도 있다. 접속(1310)은 유선 접속 또는 무선 접속일 수도 있다. 디코더(1306)는 도 2a의 블록-기반 비디오 디코더와 유사할 수도 있다. 디코더(1306)는 단일 계층 코덱(예컨대, 도 2a) 또는 멀티계층 코덱을 포함할 수도 있다. 예를 들어, 디코더(1306)는 픽처-레벨 ILP 지원을 갖는 멀티-계층(예컨대, 2-계층) 스케일러블 디코딩 시스템일 수도 있다.

[0044] 인코더(1302) 및/또는 디코더(1306)는, 디지털 텔레비전들, 무선 브로드캐스트 시스템들, 네트워크 엘리먼트/단말, (예컨대, 하이퍼텍스트 전송 프로토콜(Hypertext Transfer Protocol; HTTP) 서버와 같은) 컨텐츠 또는 웹 서버들과 같은 서버들, 개인 정보 단말(personal digital assistant; PDA)들, 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 디지털 카메라들, 디지털 레코딩 디바이스들, 비디오 게임용 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화들, 디지털 미디어 플레이어들, 및/또는 등등과 같은, 그러나 이것으로 제한되지는 않는 폭넓게 다양한 유선 통신 디바이스들 및/또는 무선 송순시 유닛(WTRU)들 내로 편입될 수도 있다.

[0045] 통신 네트워크(1304)는 임의의 적당한 타입의 통신 네트워크일 수도 있다. 예를 들어, 통신 네트워크(1304)는 음성, 데이터, 비디오, 메시징, 브로드캐스트 등과 같은 컨텐츠를 다수의 무선 사용자들에게 제공하는 다중 액세스 시스템일 수도 있다. 통신 네트워크(1304)는 다수의 무선 사용자들이 무선 대역폭을 포함하는 시스템 자원들의 공유를 통해 이러한 컨텐츠를 액세스하는 것을 가능하게 할 수도 있다. 예를 들어, 통신 네트워크(1304)는 코드 분할 다중 액세스(code division multiple access; CDMA), 시간 분할 다중 액세스(time division multiple access; TDMA), 주파수 분할 다중 액세스(frequency division multiple access; FDMA), 직교 FDMA(orthogonal FDMA; OFDMA), 단일-캐리어 FDMA(single-carrier FDMA; SC-FDMA) 및/또는 등등과 같은 하나 이상의 채널 액세스 방법들을 채용할 수도 있다. 통신 네트워크(1304)는 다수의 접속된 통신 네트워크들을 포함할 수도 있다. 통신 네트워크(1304)는 인터넷, 및/또는 셀룰러 네트워크들, WiFi 핫스폿(hotspot)들, 인터넷 서비스 제공자 네트워크들(ISP들), 및/또는 등등과 같은 하나 이상의 사설 상업용 네트워크들을 포함할 수도 있다.

[0046] 도 10은 일 예의 WTRU의 시스템 도면이다. WTRU(902)는 프로세서(918), 트랜시버(920), 송수신 엘리먼트(transmit/receive element; 922), 스피커/마이크로폰(924), 키패드 또는 키보드(926), 디스플레이/터치패드(928), 비-분리가능 메모리(930), 분리가능 메모리(932), 전원(934), 글로벌 위치확인 시스템(global positioning system; GPS) 칩셋(936), 및/또는 다른 주변기기들(938)을 포함할 수도 있다. WTRU(902)는 실시형태와 부합하게 유지하면서 상기한 엘리먼트들의 임의의 하위-조합을 포함할 수도 있다는 것이 인식될 것이다. 또한, 인코더(예컨대, 인코더(802)) 및/또는 디코더(예컨대, 디코더(806))가 편입될 수도 있는 단말은 도 10의 WTRU(902)를 참조하여 본원에서 도시되고 설명된 엘리먼트들의 일부 또는 전부를 포함할 수도 있다.

[0047] 프로세서(918)는 범용 프로세서, 특수 목적 프로세서, 기존의 프로세서, 디지털 신호 프로세서(digital signal processor)(DSP), 그래픽 프로세싱 유닛(graphics processing unit; GPU), 복수의 마이크로프로세서들, DSP 코어와 연관된 하나 이상의 마이크로프로세서들, 제어기, 마이크로제어기, 애플리케이션 특정 집적 회로

(Application Specific Integrated Circuit)(ASIC)들, 필드 프로그래밍가능 게이트 어레이(Field Programmable Gate Array)(FPGA) 회로들, 임의의 다른 타입의 집적 회로(integrated circuit)(IC), 상태 머신(state machine) 등등일 수도 있다. 프로세서(918)는 신호 코딩, 데이터 프로세싱, 전력 제어, 입출력 프로세싱, 및/또는 WTRU(902)가 유선 및/또는 무선 환경에서 동작하는 것을 가능하게 하는 임의의 다른 기능성을 수행할 수도 있다. 프로세서(918)는 송수신 엘리먼트(922)에 결합될 수도 있는 트랜시버(920)에 결합될 수도 있다. 도 10은 프로세서(918) 및 트랜시버(920)를 별도의 컴포넌트들로서 도시하고 있지만, 프로세서(918) 및 트랜시버(920)는 전자 패키지 및/또는 칩 내에 함께 집적될 수도 있다는 것이 인식될 것이다.

[0048] 송수신 엘리먼트(922)는 무선 인터페이스(915)를 통해, 신호들을 또 다른 단말로 송신하고, 및/또는 또 다른 단말로부터 신호들을 수신하도록 구성될 수도 있다. 예를 들어, 하나 이상의 실시형태들에서, 송수신 엘리먼트(922)는 RF 신호들을 송신하고 및/또는 수신하도록 구성된 안테나일 수도 있다. 하나 이상의 실시형태들에서, 송수신 엘리먼트(922)는 예를 들어, IR, UV, 또는 가시광 신호들을 송신하고 및/또는 수신하도록 구성된 에미터/검출기(emitter/detector)일 수도 있다. 하나 이상의 실시형태들에서, 송수신 엘리먼트(922)는 RF 및 광 신호들 양자를 송신하고 수신하도록 구성될 수도 있다. 송수신 엘리먼트(922)는 무선 신호들의 임의의 조합을 송신하고 및/또는 수신하도록 구성될 수도 있다는 것이 인식될 것이다.

[0049] 게다가, 송수신 엘리먼트(922)는 도 10에서 단일 엘리먼트로서 도시되어 있지만, WTRU(902)는 임의의 수의 송수신 엘리먼트들(922)을 포함할 수도 있다. 더욱 구체적으로, WTRU(902)는 MIMO 기술을 채용할 수도 있다. 따라서, 일부의 실시형태들에서, WTRU(902)는 무선 인터페이스(915)를 통해 무선 신호들을 송신하고 수신하기 위한 2 개 이상의 송수신 엘리먼트들(922)(예컨대, 다수의 안테나들)을 포함할 수도 있다.

[0050] 트랜시버(920)는 송수신 엘리먼트(922)에 의해 송신되어야 하는 신호들을 변조하도록, 및/또는 송수신 엘리먼트(922)에 의해 수신되는 신호들을 복조하도록 구성될 수도 있다. 위에서 언급된 바와 같이, WTRU(902)는 멀티-모드 능력들을 가질 수도 있다. 따라서, 트랜시버(920)는 WTRU(902)가 예를 들어, UTRA 및 IEEE 802.11과 같은 다수의 RAT들을 통해 통신하는 것을 가능하게 하기 위한 다수의 트랜시버들을 포함할 수도 있다.

[0051] WTRU(902)의 프로세서(918)는 스피커/마이크로폰(924), 키패드(926), 및/또는 디스플레이/터치패드(928)(예컨대, 액정 디스플레이(liquid crystal display; LCD) 디스플레이 유닛 또는 유기 발광 다이오드(organic light-emitting diode; OLED) 디스플레이 유닛)에 결합될 수도 있고 이로부터 사용자 입력 데이터를 수신할 수도 있다. 프로세서(918)는 또한, 사용자 데이터를 스피커/마이크로폰(924), 키패드(926), 및/또는 디스플레이/터치패드(928)로 출력할 수도 있다. 게다가, 프로세서(918)는 비-분리가능 메모리(930) 및/또는 분리가능 메모리(932)와 같은 임의의 타입의 적당한 메모리로부터 정보를 액세스할 수도 있고, 이 메모리에 데이터를 저장할 수도 있다. 비-분리가능 메모리(930)는 랜덤-액세스 메모리(random-access memory)(RAM), 판독전용 메모리(read-only memory)(ROM), 하드 디스크, 또는 임의의 다른 타입의 메모리 저장 디바이스를 포함할 수도 있다. 분리가능 메모리(932)는 가입자 식별 모듈(subscriber identity module; SIM) 카드, 메모리 스틱, 보안 디지털(secure digital; SD) 메모리 카드 등등을 포함할 수도 있다. 하나 이상의 실시형태들에서, 프로세서(918)는 서버 또는 홈 컴퓨터(도시되지 않음) 상에서와 같이, WTRU(902) 상에 물리적으로 위치되지 않은 메모리로부터 정보를 액세스할 수도 있고, 이 메모리에 데이터를 저장할 수도 있다.

[0052] 프로세서(918)는 전원(934)으로부터 전력을 수신할 수도 있고, 전력을 WTRU(902)에서의 다른 컴포넌트들로 분배하고 및/또는 제어하도록 구성될 수도 있다. 전원(934)은 WTRU(902)에 급전하기 위한 임의의 적당한 디바이스일 수도 있다. 예를 들어, 전원(934)은 하나 이상의 건전지 배터리들(예컨대, 니켈-카드뮴(nickel-cadmium; NiCd), 니켈-아연(nickel-zinc; NiZn), 니켈 금속 수소화물(nickel metal hydride; NiMH), 리튬-이온(lithium-ion; Li-ion) 등), 태양 전지(solar cell)들, 연료 전지들 등등을 포함할 수도 있다.

[0053] 프로세서(918)는 WTRU(902)의 현재의 위치에 관한 위치 정보(예컨대, 경도 및 위도)를 제공하도록 구성될 수도 있는 GPS 칩셋(936)에 결합될 수도 있다. GPS 칩셋(936)으로부터의 정보에 추가적으로 또는 이에 대신하여, WTRU(902)는 무선 인터페이스(915)를 통해 단말(예컨대, 기지국)로부터 위치 정보를 수신할 수도 있고, 및/또는 2 개 이상의 근접 기지국들로부터 수신되고 있는 신호들의 타이밍에 기초하여 그 위치를 결정할 수도 있다. WTRU(902)는 실시형태와 부합하게 유지하면서 임의의 적당한 위치-결정 방법을 통해 위치 정보를 취득할 수도 있다는 것이 인식될 것이다.

[0054] 프로세서(918)는, 추가적인 특징들, 기능성, 및/또는 유선 또는 무선 접속성을 제공하는 하나 이상의 소프트웨어 및/또는 하드웨어 모듈들을 포함할 수도 있는 다른 주변기기들(938)에 추가로 결합될 수도 있다. 예를 들어, 주변기기들(938)은 가속도계, 방위 센서들, 모션 센서들, 근접성 센서, 전자-나침판(e-compass), 위성 트

랜시버, (예컨대, 사진들 및/또는 비디오를 위한) 디지털 카메라 및/또는 비디오 레코더, 유니버셜 직렬 버스(universal serial bus; USB) 포트, 진동 디바이스, 텔리비전 트랜시버, 핸즈 프리(hands free) 헤드셋, Bluetooth® 모듈, 주파수 변조된(FM) 라디오 유닛, 및 소프트웨어 모듈들, 예컨대, 디지털 음악 플레이어, 미디어 플레이어, 비디오 게임 플레이어 모듈, 인터넷 브라우저(internet browser) 등등을 포함할 수도 있다.

[0055] 예로서, WTRU(902)는 무선 신호들을 송신하고 및/또는 수신하도록 구성될 수도 있고, 사용자 장비(user equipment; UE), 이동국(mobile station), 고정 또는 이동 가입자 유닛, 페이저(pager), 셀룰러 전화, 개인 정보 단말(PDA), 스마트폰, 랩톱, 넷북, 태블릿 컴퓨터, 개인용 컴퓨터, 무선 센서, 소비자 가전기기, 또는 압축된 비디오 통신신호들을 수신하고 프로세싱할 수 있는 임의의 다른 단말을 포함할 수도 있다.

[0056] WTRU(902) 및/또는 통신 네트워크(예컨대, 통신 네트워크(804))는, 광대역 CDMA(WCDMA)를 이용하여 무선 인터페이스(915)를 확립할 수도 있는 유니버셜 이동 통신 시스템(Universal Mobile Telecommunications System; UMTS) 지상 라디오 액세스(UMTS Terrestrial Radio Access; UTRA)와 같은 라디오 기술을 구현할 수도 있다. WCDMA는 고속 패킷 액세스(High-Speed Packet Access; HSPA) 및/또는 진화형 HSPA(Evolved HSPA; HSPA+)와 같은 통신 프로토콜들을 포함할 수도 있다. HSPA는 고속 다운링크 패킷 액세스(High-Speed Downlink Packet Access; HSDPA) 및/또는 고속 업링크 패킷 액세스(High-Speed Uplink Packet Access; HSUPA)를 포함할 수도 있다. WTRU(902) 및/또는 통신 네트워크(예컨대, 통신 네트워크(804))는, 롱텀 에볼루션(Long Term Evolution; LTE) 및/또는 LTE-어드밴스드(LTE- Advanced; LTE-A)를 이용하여 무선 인터페이스(915)를 확립할 수도 있는 진화형 UMTS 지상 라디오 액세스(Evolved UTRA; E-UTRA)와 같은 라디오 기술을 구현할 수도 있다.

[0057] WTRU(902) 및/또는 통신 네트워크(예컨대, 통신 네트워크(804))는 IEEE 802.16(예컨대, 마이크로파 액세스를 위한 전세계 상호운용성(Worldwide Interoperability for Microwave Access; WiMAX), CDMA2000, CDMA2000 1X, CDMA2000 EV-DO, 잠정 표준 2000(Interim Standard 2000; IS-2000), 잠정 표준 95(IS-95), 잠정 표준 856(IS-856), 이동 통신을 위한 글로벌 시스템(Global System for Mobile communications; GSM), GSM 진화를 위한 증대된 데이터 레이트들(Enhanced Data rates for GSM Evolution; EDGE), GSM EDGE(GERAN) 등등과 같은 라디오 기술들을 구현할 수도 있다. WTRU(902) 및/또는 통신 네트워크(예컨대, 통신 네트워크(804))는 IEEE 802.11, IEEE 802.15 등등과 같은 라디오 기술을 구현할 수도 있다.

[0058] **무손실 코딩**

[0059] 의료용 비디오 애플리케이션들 및 하이-엔드 전문가용 비디오 애플리케이션들과 같은 일부의 비디오 애플리케이션들에 대하여, 임의의 손실 없이 원래의 비디오 신호에서의 모든 정보를 보존하는 것이 바람직할 수도 있다. 이러한 비디오 애플리케이션들에 대해서는, 무손실 코딩이 이용될 수 있다. 무손실 코딩에서는, 변환 및 양자화와 같이, 정보 손실을 야기시킬 수도 있는 비디오 코덱에서의 프로세싱 블록들이 수정될 수도 있고 및/또는 우회될 수도 있다. 각각 도 1b 및 도 2b의 인코더 및 디코더 구성들은 무손실 코딩을 달성하기 위하여 이용될 수도 있다. 무손실 코딩에서는, 변환, 양자화, 역변환, 및 역양자화를 위한 프로세싱 블록들이 적용되지 않는다. 게다가, 도 1b의 합산기(126) 및 도 2b의 합산기(226)의 결과로서의 재구성된 비디오 블록은 원래의 비디오 블록과 수학적으로 동일하므로, 인-루프 필터링은 필요하지 않을 수도 있고, 사실은, 그것은 원하지 않는 왜곡을 실제적으로 도입할 수도 있다. 그러므로, 인-루프 필터링을 위한 프로세싱 블록은 또한, 일부의 실시형태들에서 적용되지 않는다.

[0060] 무선 디스플레이 및 클라우드 컴퓨팅과 같은 급속하게 성장하는 비디오 애플리케이션들로 인해, 스크린 컨텐츠 코딩(screen content coding; SCC)은 최근에 학계 및 산업계로부터 많은 관심을 받았다. HEVC가 선행하는 비디오 코딩 표준들에 비해 코딩 효율에 있어서 상당한 개선을 달성하였지만, 그것은 카메라들에 의해 캡처된 자연적인 비디오를 위하여 주로 설계되었다. 그러나, 텍스트 및 그래픽과 같은 컴퓨터-생성된 컨텐츠로 전형적으로 구성되는 스크린 컨텐츠 비디오는 자연적인 컨텐츠의 속성들과는 상당히 상이한 속성들을 보여준다. 스크린 컨텐츠 코딩을 위하여 HEVC를 확장하는 것이 바람직할 것이라고 가정한다. 인트라 블록 복사(intra block copy; IBC)는 문헌[R. Joshi, J. Xu, HEVC Screen Content Coding Draft Text 2(HEVC 스크린 컨텐츠 코딩 초안 텍스트 2). Document No. JCTVC-S1005, Oct. 2014 (Joshi 2014)]에서 설명된 바와 같이, HEVC 스크린 컨텐츠 코딩 확장으로 채택되었던 하나의 코딩 방법이다. IBC는 동일한 픽처의 이미-재구성된 영역의 픽셀들로부터 현재의 PU의 픽셀들을 예측함으로써, (특히, 픽처가 텍스트 및 그래픽이 풍분한 상당한 양의 스크린 컨텐츠를 포함할 경우) 하나의 픽처에서 고유한 인트라-픽처 중복성을 활용하도록 설계되었다. 인터 모드와 유사하게, IBC 모드로 코딩된 CU들에 대하여, 하나의 예측된 PU와 그 참조 블록 사이의 변위는 블록 벡터(block vector; BV)에 의해 표현된다. BV들은 비트-스트림에서의 대응하는 잔차들과 함께 코딩된다.

[0061] HEVC 및 그 확장들에서, transquant_bypass_enabled_flag로 칭해진 신택스 엘리먼트는 변환 및 양자화가 블록-대-블록에 기초하여 우회될 수도 있는지 여부를 표시하기 위하여 픽처 파라미터 세트(PPS)에서 시그널링된다. 문헌[D. Flynn, M. Naccari, C. Rosewarne, J. Sole, G. Sullivan, T. Suzuki, "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 6(고효율 비디오 코딩(HEVC) 범위 확장들 텍스트 사양: 초안 6)", Document No. JCTVC-P1005, Jan 2014]에서 설명된 바와 같이, HEVC에서의 PPS 신택스 표는 이하의 표 1에서 도시되어 있고, transquant_bypass_enabled_flag는 라인 22에서 도시되어 있다:

표 1

1	pic_parameter_set_rbsp( ) {	**디스크립터**
2	**pps_pic_parameter_set_id**	ue(v)
3	**pps_seq_parameter_set_id**	ue(v)
4	**dependent_slice_segments_enabled_flag**	u(1)
5	**output_flag_present_flag**	u(1)
6	**num_extra_slice_header_bits**	u(3)
7	**sign_data_hiding_enabled_flag**	u(1)
8	**cabac_init_present_flag**	u(1)
9	**num_ref_idx_l0_default_active_minus1**	ue(v)
10	**num_ref_idx_l1_default_active_minus1**	ue(v)
11*	**init_qp_minus26**	se(v)
12	**constrained_intra_pred_flag**	u(1)
13*	**transform_skip_enabled_flag**	u(1)
14*	**cu_qp_delta_enabled_flag**	u(1)
15	if(cu_qp_delta_enabled_flag)	
16*	**diff_cu_qp_delta_depth**	ue(v)
17*	**pps_cb_qp_offset**	se(v)
18*	**pps_cr_qp_offset**	se(v)
19*	**pps_slice_chroma_qp_offsets_present_flag**	u(1)
20	**weighted_pred_flag**	u(1)
21	**weighted_bipred_flag**	u(1)
22	**transquant_bypass_enabled_flag**	u(1)
23	**tiles_enabled_flag**	u(1)
24	**entropy_coding_sync_enabled_flag**	u(1)
25	if(tiles_enabled_flag) {	
26	**num_tile_columns_minus1**	ue(v)
27	**num_tile_rows_minus1**	ue(v)
28	**uniform_spacing_flag**	u(1)
29	if(!uniform_spacing_flag) {	
30	for(i = 0; i < num_tile_columns_minus1; i++)	
31	**column_width_minus1**[i]	ue(v)
32	for(i = 0; i < num_tile_rows_minus1; i++)	
33	**row_height_minus1**[i]	ue(v)
34	}	
35*	**loop_filter_across_tiles_enabled_flag**	u(1)
36	}	
37*	**pps_loop_filter_across_slices_enabled_flag**	u(1)
38*	**deblocking_filter_control_present_flag**	u(1)
39	if(deblocking_filter_control_present_flag) {	

[0062]

40*	**deblocking_filter_override_enabled_flag**	u(1)
41*	**pps_deblocking_filter_disabled_flag**	u(1)
42	if(!pps_deblocking_filter_disabled_flag) {	
43*	**pps_beta_offset_div2**	se(v)
44*	**pps_tc_offset_div2**	se(v)
45	}	
46	}	
47*	**pps_scaling_list_data_present_flag**	u(1)
48	if(pps_scaling_list_data_present_flag)	
49	scaling_list_data( )	
50	**lists_modification_present_flag**	u(1)
51	**log2_parallel_merge_level_minus2**	ue(v)
52	**slice_segment_header_extension_present_flag**	u(1)
53	**pps_extension_present_flag**	u(1)
54	if(pps_extension_present_flag) {	
55	for(i = 0; i < 1; i++)	
56	**pps_extension_flag**[i]	u(1)
57	**pps_extension_7bits**	u(7)
58	}	
59	if(pps_extension_flag[0]) {	
60	if(transform_skip_enabled_flag)	
61*	**log2_max_transform_skip_block_size_minus2**	ue(v)
62	**cross_component_prediction_enabled_flag**	u(1)
63*	**chroma_qp_adjustment_enabled_flag**	u(1)
64	if(chroma_qp_adjustment_enabled_flag) {	
65*	**diff_cu_chroma_qp_adjustment_depth**	ue(v)
66*	**chroma_qp_adjustment_table_size_minus1**	ue(v)
67	for(i = 0; i <= chroma_qp_adjustment_table_size_minus1; i++) {	
68*	**cb_qp_adjustment**[i]	se(v)
69*	**cr_qp_adjustment**[i]	se(v)
70	}	
71	}	
72*	**log2_sao_offset_scale_luma**	ue(v)
73*	**log2_sao_offset_scale_chroma**	ue(v)
74	}	
75	if(pps_extension_7bits)	
76	while(more_rbsp_data( ))	
77	**pps_extension_data_flag**	u(1)
78	rbsp_trailing_bits( )	
79	}	

[0063]

[0064] **표 1: HEVC 범위 확장 초안 6에서의 PPS 신택스 표.**

[0065] 현재의 슬라이스가 transquant_bypass_enabled_flag(표 1의 라인 22)가 1로 설정(이것은 적당한 PPS를 식별하기 위하여 슬라이스 헤더에서의 slice_pic_parameter_set_id(표 2에서의 라인 5)를 적당한 값으로 설정함으로써 행해짐) 되는 PPS를 참조할 경우, 코딩 유닛 또는 CU 레벨에서, cu_transquant_bypass_flag로 칭해진 추가적인 플래그가 현재의 슬라이스에서의 모든 CU들에 대하여 시그널링된다. coding_unit 신택스 표는 표 3에서 도시되어 있다.

표 2

1	slice_segment_header( ) {	**디스크립터**
2	**first_slice_segment_in_pic_flag**	u(1)
3	if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
4	**no_output_of_prior_pics_flag**	u(1)
5	**slice_pic_parameter_set_id**	ue(v)
6	if(!first_slice_segment_in_pic_flag) {	
7	if(dependent_slice_segments_enabled_flag)	
8	**dependent_slice_segment_flag**	u(1)
9	**slice_segment_address**	u(v)
10	}	
11	if(!dependent_slice_segment_flag) {	
12	for(i = 0; i < num_extra_slice_header_bits; i++)	
13	**slice_reserved_flag**[i]	u(1)
14	**slice_type**	ue(v)
15	if(output_flag_present_flag)	
16	**pic_output_flag**	u(1)
17	if(separate_colour_plane_flag = = 1)	
18	**colour_plane_id**	u(2)
19	if(nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) {	
20	**slice_pic_order_cnt_lsb**	u(v)
21	**short_term_ref_pic_set_sps_flag**	u(1)
22	if(!short_term_ref_pic_set_sps_flag)	
23	short_term_ref_pic_set(num_short_term_ref_pic_sets)	
24	else if(num_short_term_ref_pic_sets > 1)	
25	**short_term_ref_pic_set_idx**	u(v)
26	if(long_term_ref_pics_present_flag) {	
27	if(num_long_term_ref_pics_sps > 0)	
28	**num_long_term_sps**	ue(v)
29	**num_long_term_pics**	ue(v)
30	for(i = 0; i < num_long_term_sps + num_long_term_pics; i++) {	
31	if(i < num_long_term_sps) {	
32	if(num_long_term_ref_pics_sps > 1)	
33	**lt_idx_sps**[i]	u(v)
34	} else {	
35	**poc_lsb_lt**[i]	u(v)
36	**used_by_curr_pic_lt_flag**[i]	u(1)
37	}	
38	**delta_poc_msb_present_flag**[i]	u(1)
39	if(delta_poc_msb_present_flag[i])	
40	**delta_poc_msb_cycle_lt**[i]	ue(v)
41	}	
42	}	
43	if(sps_temporal_mvp_enabled_flag)	

[0066]

44	**slice_temporal_mvp_enabled_flag**	u(1)
45	}	
46	if(sample_adaptive_offset_enabled_flag) {	
47*	**slice_sao_luma_flag**	u(1)
48	if(ChromaArrayType != 0)	
49*	**slice_sao_chroma_flag**	u(1)
50	}	
51	if(slice_type = = P | | slice_type = = B) {	
52	**num_ref_idx_active_override_flag**	u(1)
53	if(num_ref_idx_active_override_flag) {	
54	**num_ref_idx_l0_active_minus1**	ue(v)
55	if(slice_type = = B)	
56	**num_ref_idx_l1_active_minus1**	ue(v)
57	}	
58	if(lists_modification_present_flag && NumPocTotalCurr > 1)	
59	ref_pic_lists_modification( )	
60	if(slice_type = = B)	
61	**mvd_l1_zero_flag**	u(1)
62	if(cabac_init_present_flag)	
63	**cabac_init_flag**	u(1)
64	if(slice_temporal_mvp_enabled_flag) {	
65	if(slice_type = = B)	
66	**collocated_from_l0_flag**	u(1)
67	if((collocated_from_l0_flag && num_ref_idx_l0_active_minus1 > 0) | | (!collocated_from_l0_flag && num_ref_idx_l1_active_minus1 > 0))	
68	**collocated_ref_idx**	ue(v)
69	}	
70	if((weighted_pred_flag && slice_type = = P) | | (weighted_bipred_flag && slice_type = = B))	
71	pred_weight_table( )	
72	**five_minus_max_num_merge_cand**	ue(v)
73	}	
74*	**slice_qp_delta**	se(v)
75	if(pps_slice_chroma_qp_offsets_present_flag) {	
76*	**slice_cb_qp_offset**	se(v)
77*	**slice_cr_qp_offset**	se(v)
78	}	
79	if(chroma_qp_adjustment_enabled_flag)	
80*	**slice_chroma_qp_adjustment_enabled_flag**	u(1)
81	if(deblocking_filter_override_enabled_flag)	
82*	**deblocking_filter_override_flag**	u(1)
83	if(deblocking_filter_override_flag) {	
84*	**slice_deblocking_filter_disabled_flag**	u(1)
85	if(!slice_deblocking_filter_disabled_flag) {	
86*	**slice_beta_offset_div2**	se(v)
87*	**slice_tc_offset_div2**	se(v)

[0067]

88	}	
89	}	
90	if(pps_loop_filter_across_slices_enabled_flag && (slice_sao_luma_flag || slice_sao_chroma_flag || !slice_deblocking_filter_disabled_flag))	
91*	**slice_loop_filter_across_slices_enabled_flag**	u(1)
92	}	
93	if(tiles_enabled_flag || entropy_coding_sync_enabled_flag) {	
94	**num_entry_point_offsets**	ue(v)
95	if(num_entry_point_offsets > 0) {	
96	**offset_len_minus1**	ue(v)
97	for(i = 0; i < num_entry_point_offsets; i++)	
98	**entry_point_offset_minus1**[i]	u(v)
99	}	
100	}	
101	if(slice_segment_header_extension_present_flag) {	
102	**slice_segment_header_extension_length**	ue(v)
103	for(i = 0; i < slice_segment_header_extension_length; i++)	
104	**slice_segment_header_extension_data_byte**[i]	u(8)
105	}	
106	byte_alignment( )	
107	}	

[0068]

[0069] **표 2: HEVC 범위 확장 초안 6에서의 슬라이스 세그먼트 헤더 신택스 표.**

표 3

1	coding_unit(x0, y0, log2CbSize) {	**디스크립터**
2	if(transquant_bypass_enabled_flag)	
3	**cu_transquant_bypass_flag**	ae(v)
4	if(slice_type != I)	
5	**cu_skip_flag**[x0][y0]	ae(v)
6	….	
7	}	

[0070]

[0071] **표 3: HEVC 범위 확장들 초안 6에서의 코딩 유닛(CU) 신택스 표.**

[0072] cu_transquant_bypass_flag(표 3의 라인 3)의 값은, (cu_transquant_bypass_flag가 1과 동일할 경우에) 변환 및 양자화가 현재의 CU에 대하여 우회되는지, 또는 (cu_transquant_bypass_flag가 0과 동일할 경우에) 변환 및 양자화가 현재의 CU에 대하여 적용되는지 여부를 표시한다. PPS에서의 transquant_bypass_enabled_flag가 0으로 설정될 때, 추가적인 CU 레벨 플래그 cu_transquant_bypass_flag는 시그널링되지 않고, 0인 것으로 추론된다(즉, 변환 및 양자화가 현재의 CU에 대하여 적용됨).

[0073] 플래그 cu_tranquant_bypass_flag가 현재의 CU에 대하여 1로 설정될 때, 변환 및 양자화는 예측 잔차에 적용되지 않는다. 그 대신에, 예측 잔차는 직접적으로 엔트로피 코딩되고, 그 예측 모드 정보, 모션 정보 등과 함께, 비디오 비트스트림으로 팩킹된다. 추가적으로, 디블록킹 및 샘플 적응적 오프셋들(SAO)은 현재의 CU에 대하여 우회된다. 이러한 방법으로, 블록-레벨 무손실 코딩이 달성되고, 즉, 재구성된 CU는 원래의 CU와 수학적으로 동일하다.

[0074] 무손실 코딩을 요구하는 비디오 애플리케이션들에 대하여, 시퀀스 레벨에서(즉, 전체 시퀀스가 손실 없이 코딩됨) 또는 픽처/슬라이스 레벨에서(즉, 전체 픽처/슬라이스가 손실 없이 코딩됨) 무손실 코딩을 적용하는 것이 바람직할 수도 있다. 제 1 버전 HEVC 표준뿐만 아니라, JCT-VC에서 현재 개발 중인 HEVC 범위 확장들은 시퀀스/픽처/슬라이스 레벨 무손실 코딩을 표시하기 위한 하이 레벨 시그널링을 포함하지 않거나, 또는 그렇지 않을 경우에 제공하지 않는다. 그 대신에, 기존의 시그널링 방식을 이용하여 시퀀스/픽처/슬라이스 레벨 무손실 코딩을 달성하기 위하여, 다음의 동작들이 수행될 수도 있다: (i) transquant_bypass_enabled_flag가 1로 설정되는 PPS의 생성; (ii) 비디오 시퀀스/픽처/슬라이스의 슬라이스 세그먼트 헤더들에서, 1과 동일한 transquant_bypass_enabled_flag를 갖는 PPS에 대한 참조; (iii) 시퀀스/픽처/슬라이스에서의 모든 CU들에 대하여, 플래그 cu_transquant_bypass_flag의 값을 1로 설정.

[0075] 무손실 모드의 블록-레벨 시그널링에만 의존하는 것은 다수의 단점들을 가진다. 특히, 그것은 시퀀스/픽처/슬라이스에서의 모든 CU들에 대한 cu_transquant_bypass_flag를 전송할 것을 요구하고, 이것은 효율적이지 않을 수도 있다. 컨텍스트-적응적 2진 산술 코딩(context-adaptive binary arithmetic coding; CABAC)은, 시그널링 오버헤드(signaling overhead)를 효과적으로 감소시킬 수 있는, 이 CU 레벨 플래그 cu_transquant_bypass_flag를 코딩하기 위하여 이용되지만, 이것은 디코더가 시퀀스/픽처/슬라이스에서 모든 CU들에 대한 추가적인 신택스 엘리먼트를 파싱할 것을 여전히 요구하고, 이것은 중복 동작일 수도 있다.

[0076] 또한, 무손실 모드에 대한 블록 레벨 시그널링만이 이용가능할 경우, 역양자화, 역변환, 디블록킹 필터, SAO 등등과 같은 프로세싱 블록들이 일부의 미래의 CU들에 대하여 여전히 필요하게 될 수도 있다(즉, 일부의 미래의 CU들의 cu_transquant_bypass_flag의 값은 0으로 설정될 수도 있음)는 것이 가능한 것으로 유지되므로, 디코더는 완전히 무손실 코딩을 적당하게 준비할 수 없을 수도 있다. 이것은 불필요한 프로세싱 블록들을 정지시키는 것으로부터 그렇지 않을 경우에 이용가능하게 될 이용가능한 전력 절감들을 제한한다.

[0077] 유사하게, 비디오 비트스트림들은 무손실 인코딩을 위하여 필요하지 않은 다양한 레벨들에서 다수의 신택스 엘리먼트들을 포함한다. 이 엘리먼트들은 하이 레벨 신택스 구조들에서의 역양자화, 역변환, 디블록킹, 및 SAO에 관련된다. 예를 들어, PPS에서, 일부의 신택스 엘리먼트들(예컨대, 표 1에서의 라인들 11, 14, 17, 18 상에서 각각 도시된 바와 같은, init_qp_minus26, cu_qp_delta_enabled_flag, pps_cb/cr_qp_offset)은 역양자화 프로세스에 관련되고; 일부의 신택스 엘리먼트들(예컨대, 표 1에서의 라인들 40, 43, 44 상에서 각각 도시된 바와 같은, deblocking_filter_override_enabled_flag, pps_beta_offset_div2, pps_tc_offset_div2)은 디블록킹 프로세스에 관련되고; 일부의 신택스 엘리먼트들(표 1에서의 라인들 72, 73 상에서 각각 도시된 바와 같은, log2_sao_offset_scale_luma, log2_sao_offset_scale_chroma)은 SAO 프로세스에 관련된다. 유사하게, 슬라이스 세그먼트 헤더에서의 일부의 신택스 엘리먼트들은 역양자화(표 2에서의 라인들 74, 76, 77 상에서 각각 도시된 바와 같은, slice_qp_delta, slice_cb_qp_offset, slice_cr_qp_offset), 디블록킹(라인들 86, 87 상에서 도시된 바와 같은, slice_beta_offset_div2, slice_tc_offset_div2), 및 SAO(표 2에서의 라인들 47 및 49 상에서 도시된 바와 같은, slice_sao_luma_flag, slice_sao_chroma_flag)에 관련된다. 이러한, 그리고 다른 신택스 엘리먼트들은 표들 1 내지 3에서의 별표들로 표기된다. 무손실 코딩이 시퀀스/픽처/슬라이스 레벨 상에서 적용될 때, 시그널링 오버헤드를 감소시키는, 이 신택스 엘리먼트들을 시그널링하는 것이 필요하지 않을 수도 있다. 그러나, 무손실 코딩 모드의 하이 레벨 표시가 없다면, 이 하이-레벨 신택스 엘리먼트들은 비트스트림으로 인코딩되어야 하고, 개개의 디코더로 송신되어야 한다.

[0078] 또한, 비디오 비트스트림은 블록 레벨에서(예컨대, CU 레벨에서) 변환-관련 시그널링을 포함할 수도 있다. 예를 들어, transform_tree() 신택스 구조(transform_tree()의 간략화된 버전이 표 7에서 도시되어 있음)에서는, 변환 유닛들의 쿼드트리 분리(quadtree splitting)이 수행되는지 여부(transform_split_flag), 및/또는 루마 및 크로마에 대한 변환 블록에서 임의의 비제로(nonzero) 계수들이 있는지 여부(cbf_luma, cbf_cb, 및 cbf_cr)를 표시하기 위하여 플래그들이 시그널링된다. 무손실 코딩 모드에서, transform_tree() 신택스는 이 플래그들의 시그널링을 명시적으로 우회시킴으로써, 그리고 그 대신에, 이 플래그들을 적절한 디폴트 값들로 설정함으로써 간략화될 수도 있다.

[0079] **무손실 코딩 모드 시그널링**

[0080] 상기 단점들 중의 하나 이상을 해소할 수도 있는 무손실 코딩 모드에서 이용된 시그널링 방법들의 다양한 실시형태들이 본원에서 설명된다. 하나의 이러한 실시형태에서, 무손실 코딩 모드의 시그널링은 PPS 신택스 구조를 수정함으로써 행해질 수도 있다. 표 4는 일부의 실시형태들에 따라 수정된 PPS 신택스 표를 도시하고, 여기서, 추가적인 플래그, transquant_bypass_default_flag는 이 PPS를 참조하는 슬라이스들에서의 모든 코딩 유닛들의 플래그 cu_transquant_bypass_flag의 디폴트 값을 표시하기 위하여 추가된다. 이 실시형태에서, 비트 스트림은 각각의 개별적인 코딩 유닛에서 플래그 cu_transquant_bypass_flag를 시그널링할 필요가 없다. 이 새로운 플래그를 1로 설정하고, 플래그 transquant_bypass_enabled_flag를 0으로 설정함으로써, 인코더는 시퀀스, 및/또는 픽처, 및/또는 슬라이스 상에서, 무손실 코딩이 적용된다는 것을 디코더에 표시할 수도 있다. 즉, 임의의 CU 레벨 시그널링이 없다면, 이 현재의 PPS를 참조하는 모든 CU들의 변환, 변환 스킵, 양자화, 및 인-루프 필터링 프로세스가 우회된다.

[0081] 새로운 플래그 transquant_bypass_default_flag는, u_transquant_bypass_flag가 존재하지 않을 때에 cu_transquant_bypass_flag의 디폴트 값을 특정하는 것으로 고려될 수도 있다.

[0082] 0인 CU 레벨 플래그 cu_transquant_bypass_flag의 기존의 시맨틱(semantic)들은 다음과 같이 수정될 수도

있다: 1과 동일한 cu_transquant_bypass_flag는 하위조항 8.6에서 특정된 바와 같은 스케일링 및 변환 프로세스와, 하위조항 8.7에서 특정된 바와 같은 인-루프 필터 프로세스가 우회된다는 것을 특정한다. cu_transquant_bypass_flag가 존재하지 않을 때, 그것은 transquant_bypass_default_flag와 동일한 것으로 추론된다. 표 4에서 도시된 바와 같이, 새로운 플래그 transquant_bypass_default_flag(표 4, 라인 11)는, 표 4에서 별표들로 표기되는, 역양자화, 역변환, 및 인-루프 필터링 프로세스에 관련된 PPS에서의 (표 4, 라인들 12, 15, 17, 21, 28, 42, 45, 57, 74, 및 84에서 도시된 바와 같은) 다수의 신택스 엘리먼트들의 존재를 조건화하기 위하여 이용될 수도 있다. 이 조건들은, transquant_bypass_default_flag가 0과 동일할 때(즉, 손실 코딩이 적용될 때)에 이 신택스 엘리먼트들이 오직 전송되도록 설정된다. transquant_bypass_default_flag가 1과 동일할 때(즉, 무손실 코딩이 적용될 때), 이 신택스 엘리먼트들은 전송되지 않고; 그 대신에, 그 값들은 0인 것으로 추론된다. 예를 들어, transquant_bypass_default_flag가 1로 설정될 때에 cu_qp_delta_enabled_flag의 값을 0인 것으로 추론함으로써, 델타 QP 관련 신택스 엘리먼트들이 CU 레벨에서 시그널링되지 않는다는 것이 표시되고, 이에 따라, 비트들을 절감하고 신택스 파싱(syntax parsing)을 간략화한다.

[0083] 또한, 새로운 플래그 transquant_bypass_default_flag는 transform_skip_enabled_flag의 존재를 조건화하기 위하여 이용된다. transform_skip_enabled_flag는 변환 프로세스(그러나 양자화 프로세스는 아님)만을 우회시키기 위하여 이용된다. 그러므로, 그것은 transquant_bypass_enabled_flag의 서브세트(subset)이다. 또한, 새로운 플래그 transquant_bypass_default_flag는 transquant_bypass_enabled_flag의 존재를 조건화하기 위하여 이용된다. 이러한 방법으로, transquant_bypass_default_flag가 1로 설정될 때(즉, 무손실 코딩 모드), transquant_bypass_enabled_flag는 0인 것으로 추론되고, CU 레벨에서의 cu_transquant_bypass_flag의 시그널링은 스킵된다.

표 4

[0084]

1	pic_parameter_set_rbsp( ) {	디스크립터
2	**pps_pic_parameter_set_id**	ue(v)
3	**pps_seq_parameter_set_id**	ue(v)
4	**dependent_slice_segments_enabled_flag**	u(1)
5	**output_flag_present_flag**	u(1)
6	**num_extra_slice_header_bits**	u(3)
7	**sign_data_hiding_enabled_flag**	u(1)
8	**cabac_init_present_flag**	u(1)
9	**num_ref_idx_l0_default_active_minus1**	ue(v)
10	**num_ref_idx_l1_default_active_minus1**	ue(v)
11†	**transquant_bypass_default_flag**	u(1)
12†	if(!transquant_bypass_default_flag)	
13*	**init_qp_minus26**	se(v)
14	**constrained_intra_pred_flag**	u(1)

15†	if(!transquant_bypass_default_flag)	
16*	**transform_skip_enabled_flag**	u(1)
17†	if(!transquant_bypass_default_flag)	
18*	**cu_qp_delta_enabled_flag**	u(1)
19	if(cu_qp_delta_enabled_flag)	
20*	**diff_cu_qp_delta_depth**	ue(v)
21†	if(!transquant_bypass_default_flag) {	
22*	**pps_cb_qp_offset**	se(v)
23*	**pps_cr_qp_offset**	se(v)
24*	**pps_slice_chroma_qp_offsets_present_flag**	u(1)
25†	}	
26	**weighted_pred_flag**	u(1)
27	**weighted_bipred_flag**	u(1)
28†	if(!transquant_bypass_default_flag)	
29	**transquant_bypass_enabled_flag**	u(1)
30	**tiles_enabled_flag**	u(1)
31	**entropy_coding_sync_enabled_flag**	u(1)
32	if(tiles_enabled_flag) {	
33	**num_tile_columns_minus1**	ue(v)
34	**num_tile_rows_minus1**	ue(v)
35	**uniform_spacing_flag**	u(1)
36	if(!uniform_spacing_flag) {	
37	for(i = 0; i < num_tile_columns_minus1; i++)	
38	**column_width_minus1**[i]	ue(v)
39	for(i = 0; i < num_tile_rows_minus1; i++)	
40	**row_height_minus1**[i]	ue(v)
41	}	
42†	if(!transquant_bypass_default_flag)	
43*	**loop_filter_across_tiles_enabled_flag**	u(1)
44	}	
45†	if(!transquant_bypass_default_flag) {	
46*	**pps_loop_filter_across_slices_enabled_flag**	u(1)
47*	**deblocking_filter_control_present_flag**	u(1)
48†	}	
49	if(deblocking_filter_control_present_flag) {	
50*	**deblocking_filter_override_enabled_flag**	u(1)
51*	**pps_deblocking_filter_disabled_flag**	u(1)
52	if(!pps_deblocking_filter_disabled_flag) {	
53*	**pps_beta_offset_div2**	se(v)
54*	**pps_tc_offset_div2**	se(v)
55	}	
56	}	
57†	if(!transquant_bypass_default_flag)	
58*	**pps_scaling_list_data_present_flag**	u(1)
59	if(pps_scaling_list_data_present_flag)	
60	scaling_list_data( )	
61	**lists_modification_present_flag**	u(1)

[0085]

62	**log2_parallel_merge_level_minus2**	ue(v)
63	**slice_segment_header_extension_present_flag**	u(1)
64	**pps_extension_present_flag**	u(1)
65	if(pps_extension_present_flag) {	
66	for(i = 0; i < 1; i++)	
67	**pps_extension_flag**[i]	u(1)
68	**pps_extension_7bits**	u(7)
69	}	
70	if(pps_extension_flag[0]) {	
71	if(transform_skip_enabled_flag)	
72*	**log2_max_transform_skip_block_size_minus2**	ue(v)
73	**cross_component_prediction_enabled_flag**	u(1)
74†	if(!transquant_bypass_default_flag)	
75*	**chroma_qp_adjustment_enabled_flag**	u(1)
76	if(chroma_qp_adjustment_enabled_flag) {	
77*	**diff_cu_chroma_qp_adjustment_depth**	ue(v)
78*	**chroma_qp_adjustment_table_size_minus1**	ue(v)
79	for(i = 0; i <= chroma_qp_adjustment_table_size_minus1; i++) {	
80*	**cb_qp_adjustment**[i]	se(v)
81*	**cr_qp_adjustment**[i]	se(v)
82	}	
83	}	
84†	if(!transquant_bypass_default_flag) {	
85*	**log2_sao_offset_scale_luma**	ue(v)
86*	**log2_sao_offset_scale_chroma**	ue(v)
87†	}	
88	}	
89	if(pps_extension_7bits)	
90	while(more_rbsp_data( ))	
91	**pps_extension_data_flag**	u(1)
92	rbsp_trailing_bits( )	
93	}	

[0086]

[0087] **표 4: 무손실 코딩 모드 시그널링을 갖는 PPS 신택스 표.**

[0088] 또 다른 실시형태에서는, HEVC 범위 확장들을 통해 제안된 추가적인 신택스 엘리먼트를 구현하기 위하여, transquant_bypass_default_flag의 위치는 PPS 확장의 일부(즉, pps_extension_flag[0]의 "if" 조건 내부)로서 더욱 하부로 이동될 수도 있다. 이 배열은 HEVC 범위 확장들의 PPS 신택스가 HEVC 표준의 제 1 버전(B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, Y. K. Wang, T. Wiegand. High Efficiency Video Coding (HEVC) text specification draft 10(고효율 비디오 코딩(HEVC) 텍스트 사양 초안 10). Document No. JCTVC-L1003. Jan. 2013)과 최대로 역호환가능하다는 것을 보장할 수도 있다. 표 5 는 이러한 배열의 예를 도시한다. 이 배열에서, 새로운 플래그 transquant_bypass_default_flag는 transquant_bypass_enabled_flag의 값에 대해 조건화될 수도 있다. 즉, transquant_bypass_default_flag는 transquant_bypass_enabled_flag가 0과 동일할 때에 오직 시그널링된다. transquant_bypass_enabled_flag가 1과 동일할 때, transquant_bypass_default_flag는 시그널링되지 않고, 이 PPS를 참조하는 각각의 코딩 유닛에서의 cu_transquant_bypass_flag의 값은 코딩 유닛 레벨에서 비트스트림으로부터 명시적으로 수신된다. 그러나, 이 실시형태의 배열에서는, 새로운 플래그가 주 PPS 신택스의 일부가 아니므로, 새로운 플래그는 위에서 설명된 바와 같이, 표 4에서 양자화, 변환, 변환 스킵, 및 인-루프 필터링에 관련된 그러한 기존의 PPS 신택스 엘리먼트들(별표로 표기된 신택스 엘리먼트들)의 존재를 조건화하기 위하여 이용되지 않을 수도 있다. 또 다른 플래그 lossless_coding_conformance_flag는 transquant_bypass_default_flag가 1일 경우에 시그널링된다. 플래그 lossless_coding_conformance_flag가 1일 경우, 비트스트림 준수 요건(bitstream conformance requirement)들은 무손실 코딩 모드에서 이용되지 않은 그러한 신택스 엘리먼트들의 시그널링된 값이 적당한 값들을 가지는 것을 보장하기 위하여 적용될 수도 있다. 예를 들어, 준수하는 비트스트림에서, cu_qp_delta_enabled_flag, pps_loop_filter_across_slices_enabled_flag, deblocking_filter_control_present_flag, loop_filter_across_tiles_enabled_flag, pps_scaling_list_data_present_flag 등등을 포함하는 신택스 엘리먼트들의 값들은, 새로운 플래그 transquant_bypass_default_flag가 1로 설정될 경우에 0 으로 설정되도록 요구될 수도 있다. 이러한 비트스트림 준수 요건들은 무손실 코딩에서 이용되지 않은 신택스 엘리먼트들에 관련된 시

그널링 오버헤드를 최소화하는 것을 도울 수도 있다.

[0089] 위에서 설명된 바와 같이, 일부의 실시형태들에서, PPS는 무손실 코딩 표시를 위한 새로운 플래그 transquant_bypass_default_flag를 반송하기 위하여 이용된다. 그러나, 다른 실시형태들에서는, 시퀀스 파라미터 세트(SPS) 또는 비디오 파라미터 세트(VPS)와 같은 다른 하이-레벨 신택스 구조들이 또한, 제안된 플래그를 반송하기 위하여 이용될 수도 있다. 대안적으로, 슬라이스-레벨 무손실 코딩 표시만이 희망될 경우, 슬라이스 세그먼트 헤더는 제안된 새로운 플래그를 반송하기 위하여 이용될 수도 있다.

[0090] 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련된 일부의 신택스 엘리먼트들이 SPS의 일부로서 시그널링될 수도 있다는 것에 주목한다. 이러한 SPS 신택스 엘리먼트들의 예들은 log2_min_transform_block_size_minus2, log2_diffmax_min_transform_block_size, max_transform_hierarchy_depth_inter, max_transform_hierarchy_depth_intra, scaling_list_enabled_flag, sample_adaptive_offset_enabled_flag, pcm_loop_filter_disabled_flag, transform_skip_rotation_enabled_flag, transform_skip_context_enabled_flag 등등을 포함한다. 무손실 코딩 모드가 SPS 레벨 또는 VPS 레벨에서 플래그 transquant_bypass_default_flag를 이용하여 표시될 경우, 제안된 플래그는 양자화, 변환, 변환 스킵, 변환 스킵 회전, 및 인-루프 필터링 프로세스들에 관련되는 이 SPS 신택스 엘리먼트들의 존재를 조건화하기 위하여 이용될 수도 있다. 대안적으로, 유사한 비트스트림 준수 요건들은 또한, 적당한 값들이 이 신택스 엘리먼트들에 대하여 시그널링되는 것을 보장하기 위하여, 예를 들어, 준수 플래그가 설정될 경우에 인-루프 필터들이 디스에이블되는 것을 보장하기 위하여 적용될 수도 있다.

표 5

[0091]

1	pic_parameter_set_rbsp( ) {	**디스크립터**
2	**pps_pic_parameter_set_id**	ue(v)
3	**pps_seq_parameter_set_id**	ue(v)
4	…	
5	if(pps_extension_flag[0]) {	
6†	if(!transquant_bypass_enabled_flag)	
7†	**transquant_bypass_default_flag**	u(1)
8†	if (transquant_bypass_default_flag)	
9†	**lossless_coding_conformance_flag**	u(1)
10†	if(transform_skip_enabled_flag && !transquant_bypass_default_flag)	
11*	**log2_max_transform_skip_block_size_minus2**	ue(v)
12	**cross_component_prediction_enabled_flag**	u(1)
13†	if(!transquant_bypass_default_flag)	
14*	**chroma_qp_adjustment_enabled_flag**	u(1)
15	if(chroma_qp_adjustment_enabled_flag) {	
16*	**diff_cu_chroma_qp_adjustment_depth**	ue(v)
17*	**chroma_qp_adjustment_table_size_minus1**	ue(v)
18	for(i = 0; i <= chroma_qp_adjustment_table_size_minus1; i++) {	
19*	**cb_qp_adjustment**[i]	se(v)
20*	**cr_qp_adjustment**[i]	se(v)
21	}	
22	}	
23†	if(!transquant_bypass_default_flag) {	
24*	**log2_sao_offset_scale_luma**	ue(v)
25*	**log2_sao_offset_scale_chroma**	ue(v)
26†	}	
27	}	
28	…	
29	}	

[0092] **표 5: PPS 확장들에서의 transquant_bypass_default_flag 시그널링.**

[0093] **슬라이스 헤더 시그널링**

[0094] 추가의 실시형태에서는, 슬라이스 헤더 시그널링이 이용될 수도 있다. PPS와 유사하게, 표 2에서의 슬라이스 세그먼트 헤더는 또한, 변환, 양자화, 및 인-루프 필터링 프로세싱 블록들을 위하여 이용되는 다수의 신택스 엘

리먼트들(별표들로 표기된 것들)을 포함한다. 이 신택스 엘리먼트들은 새로운 플래그 transquant_bypass_default_flag의 값에 대해 조건화될 수도 있고, transquant_bypass_default_flag를 1로 설정함으로써 무손실 코딩이 표시될 때에 시그널링될 필요가 없을 수도 있다. 표 6은 이러한 예를 도시한다.

[0095] 표 2에서 도시된 바와 같이, 슬라이스 세그먼트 헤더는 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련되는 몇몇 신택스 엘리먼트들을 포함한다. 이러한 슬라이스 세그먼트 헤더 신택스 엘리먼트들(그 라인 번호들은 별표들로 표기됨)은 slice_sao_luma_flag, slice_sao_chroma_flag, slice_qp_delta, slice_cb_qp_offset, slice_cr_qp_offset, slice_chroma_qp_adjustment_enabled_flag, deblocking_filter_override_flag, slice_deblocking_filter_disabled_flag, slice_beta_offset_div2, slice_tc_offset_div2, 및 slice_loop_filter_across_slices_enabled_flag를 포함한다. 슬라이스-레벨 무손실 코딩이 슬라이스 세그먼트 헤더에서 제안된 플래그 transquant_bypass_default_flag를 시그널링함으로써 인에이블될 경우, 제안된 플래그는 슬라이스 세그먼트 헤더에서 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련된 이 신택스 엘리먼트들의 존재를 조건화하기 위하여 이용될 수도 있다. 따라서, 하나의 실시형태에서, 플래그 transquant_bypass_default_flag는 슬라이스 세그먼트 헤더에서 양자화, 변환, 및 인-루프 필터링에 관련된 그 신택스 엘리먼트들의 전방에 배치된다. 또 다른 실시형태에서, 플래그 transquant_bypass_default_flag는 예를 들어, 양자화, 변환, 및 인-루프 필터링에 관련된 그 신택스 엘리먼트들 이후에, 슬라이스 세그먼트 헤더에서의 대안적인 위치에서 배치될 수도 있다. 이 경우, 비트스트림 준수 요건은, transquant_bypass_default_flag가 1로 설정될 경우에 그 신택스 엘리먼트들의 값들이 적당하게 설정되는 것을 보장하기 위하여 적용되어야 한다. transquant_bypass_default_flag를 1로 설정하는 것은, 비트-스트림에서 각각의 개별적인 코딩 유닛의 cu_transquant_bypass_flag를 시그널링하지 않으면서, 현재의 슬라이스에서의 모든 코딩 유닛들이 무손실 모드에서 코딩된다는 것을 표시한다. 추가적으로, 다음의 비트-스트림 준수 요건이 적용될 수 있다: 현재의 슬라이스가 참조하는 PPS의 플래그 transquant_bypass_enabled_flag의 값은, 제안된 플래그 transquant_bypass_default_flag가 1로 설정될 때에 1과 동일하다.

[0096] 표 6은 제안된 플래그 transquant_bypass_default_flag가 슬라이스 세그먼트 헤더에서 양자화, 변환, 및 인-루프 필터링에 관련된 신택스 엘리먼트들 이전에 시그널링될 때의 수정된 슬라이스 세그먼트 헤더의 하나의 예를 도시한다. 표 6에서의 예는, 제안된 transquant_bypass_default_flag가 슬라이스 세그먼트 헤더에서 시그널링되는 것을 도시하지만, 이 플래그는 그 대신에 PPS(표 5)에서 시그널링될 수도 있고, 별표들로 표시된 슬라이스 세그먼트 헤더 신택스 엘리먼트들의 존재를 조건화하기 위하여 이용될 수도 있다.

표 6

1	slice_segment_header( ) {	**디스크립터**
2	…	
3	if(!dependent_slice_segment_flag) {	
4	for(i = 0; i < num_extra_slice_header_bits; i++)	
5	**slice_reserved_flag**[i]	u(1)
6	**slice_type**	ue(v)
7	if(output_flag_present_flag)	
8	**pic_output_flag**	u(1)
9	if(separate_colour_plane_flag = = 1)	
10	**colour_plane_id**	u(2)
11†	**transquant_bypass_default_flag**	u(1)
12	…	
13†	if(sample_adaptive_offset_enabled_flag && !transquant_bypass_default_flag) {	
14*	**slice_sao_luma_flag**	u(1)
15	if(ChromaArrayType != 0)	
16*	**slice_sao_chroma_flag**	u(1)
17	}	
18	…	
19†	if(!transquant_bypass_default_flag)	
20*	**slice_qp_delta**	se(v)
21†	if(pps_slice_chroma_qp_offsets_present_flag && !transquant_bypass_default_flag) {	
22*	**slice_cb_qp_offset**	se(v)
23*	**slice_cr_qp_offset**	se(v)
24	}	
25†	if(chroma_qp_adjustment_enabled_flag && !transquant_bypass_default_flag)	
26*	**slice_chroma_qp_adjustment_enabled_flag**	u(1)
27†	if(deblocking_filter_override_enabled_flag && !transquant_bypass_default_flag)	
28*	**deblocking_filter_override_flag**	u(1)
29†	if(deblocking_filter_override_flag && !transquant_bypass_default_flag) {	
30*	**slice_deblocking_filter_disabled_flag**	u(1)
31	if(!slice_deblocking_filter_disabled_flag) {	
32*	**slice_beta_offset_div2**	se(v)
33*	**slice_tc_offset_div2**	se(v)
34	}	
35	}	
36†	if(pps_loop_filter_across_slices_enabled_flag && !transquant_bypass_default_flag && (slice_sao_luma_flag || slice_sao_chroma_flag || !slice_deblocking_filter_disabled_flag))	
37*	**slice_loop_filter_across_slices_enabled_flag**	u(1)
38	}	
39	…	
40	}	

[0097]

[0098] **표 6: transquant_bypass_default_flag를 이용한 수정된 슬라이스 세그먼트 헤더 신택스.**

[0099] transquant_bypass_default_flag가 PPS 확장들의 일부로서 시그널링되는 표 5를 예로서 이용하면, 도 12는 디코더 측에서 수정된 PPS 확장된 신택스를 파싱하기 위한 알고리즘의 하나의 실시형태를 예시한다. 수정된 PPS 확장들에서, 제안된 플래그 transquant_bypass_default_flag는 파싱되고, 그 값은 단계(1202)에서 검사된다. transquant_bypass_default_flag가 0과 동일할 경우, 고효율 비디오 코딩(HEVC) 범위 확장들 텍스트 사양(초안 6)에서의 기존의 PPS 확장들 신택스 엘리먼트들이 파싱되고 프로세싱된다(단계(1204)). transquant_bypass_default_flag가 1과 동일할 경우, 신택스 엘리먼트들 chroma_qp_adjustment_enabled_flag, log2_sao_offset_scale_luma, 및 log2_sao_offset_scale_chroma는 파싱되지 않고, 그 대신에, 그 값들은 0인 것으로 추로된다(단계(1206)). 플래그 lossless_coding_conformance_flag가 1일 경우(단계(1207)), 비트스트림 준수 요건들이 적용된다. 준수 요건들은, 이 신택스 엘리먼트들이 적당하게 디스에이블되는 것을 보장하기 위하여, 양자화, 변환, 및 인-루프 필터링에 관련된 기존의 PPS 신택스 엘리먼트들(예를 들어, init_qp_minus26, cu_qp_delta_enabled_flag, pps_cb/cr_qp_offset, cu_qp_delta_enabled_flag, pps_loop_filter_across_slices_enabled_flag, deblocking_filter_control_present_flag, loop_filter_across_tiles_enabled_flag, pps_scaling_list_data_present_flag 등등)의 시그널링된 값들을 체

크함으로서 적용된다(단계(1208)). 이 신택스 엘리먼트들 중의 하나 이상이 디스에이블되지 않을 경우, 디코더는 비트스트림 준수 위반을 보고할 것이고(단계(1210)); 이와 다를 경우, PPS 확장들의 파싱이 정상적으로 완료된다.

[0100] **변환 트리 신택스 시그널링.**

[0101] ***무손실 코딩을 위한 변환 쿼드트리 트리 분리.***

[0102] HEVC 및 HEVC 범위 확장들은 변환 유닛들(TU)의 크기를 시그널링하기 위하여 변환 트리 분리 신택스를 이용한다. 하나의 실시형태에서는, 고효율 비디오 코딩(HEVC) 범위 확장들 텍스트 사양(초안 6)의 섹션 7.3.8.8에서 특정된 변환 트리 신택스가 새로운 제안된 플래그 transquant_bypass_default_flag에 기초하여 변경될 필요가 없다. 또 다른 실시형태에서, 변환 트리 신택스는 transquant_bypass_default_flag가 1과 동일할 때, 쿼드트리 분리 플래그(split_transform_flag) 및/또는 루마 및 크로마 컴포넌트들에 대한 코딩된 블록 플래그들(cbf_luma, cbf_cb, cbf_cr)의 시그널링을 우회시키기 위하여 간략화될 수도 있다. 간략화된 transform_tree() 신택스는 표 7에서 도시되어 있다. transquant_bypass_default_flag는 split_transform_flag의 존재에 대한 추가적인 조건으로서 이용된다. 존재하지 않을 때, split_transform_flag의 값은 대부분의 경우들에 대하여 0(즉, 변환 쿼드트리 분리가 적용되지 않음)인 것으로 추론되고, 변환 쿼드트리 분리가 집행된느 일부의 기존의 특수한 경우들(예를 들어, NxN 파티션이 인트라 코딩에서 이용될 때, 또는 현재의 CU 크기가 32x32의 최대 변환 크기보다 더 클 때 등등)에 대하여 1(즉, 변환 쿼드트리 분리가 적용됨)인 것으로 추론된다. 추가적으로, 표 7에서 도시된 바와 같이, transquant_bypass_default_flag가 1과 동일할 때, transform_tree()에서의 (루마 컴포넌트 및 크로마 컴포넌트들에 대한) 모든 cbf 플래그의 시그널링은 스킵될 수도 있고; 그 대신에, 그 값들은 무손실 코딩에서의 양자화 프로세스의 결여로 인해, cbf 플래그들이 비제로 값들을 아마도 가질 것이므로, 1과 동일한 것으로 추론될 수도 있다.

[0103] cbf_luma, cbf_cb, 및 cbf_cr의 시맨틱들이 또한 수정될 수도 있다. 1과 동일한 cbf_luma[x0] [y0] [trafoDepth]의 값은 루마 변환 블록이 0과 동일하지 않은 하나 이상의 변환 계수 레벨들을 포함한다는 것을 특정하기 위하여 이용될 수도 있다. 어레이 인덱스들 x0, y0는 픽처의 상부-좌측 루마 샘플에 관하여 고려된 변환 블록의 상부-좌측 루마 샘플의 위치 (x0, y0)를 특정한다. 어레이 인덱스 trafoDepth는 변환 코딩의 목적을 위한 블록들로의 코딩 블록의 현재의 재분할 레벨을 특정한다. trafoDepth는 코딩 블록들에 대응하는 블록들에 대하여 0과 동일하다. cbf_luma[x0][y0][trafoDepth]가 존재하지 않을 때, 그것은 1과 동일한 것으로 추론된다.

[0104] 1과 동일한 cbf_cb[x0][y0][trafoDepth]의 값은, Cb 변환 블록이 0과 동일하지 않은 하나 이상의 변환 계수 레벨들을 포함한다는 것을 표시하기 위하여 이용될 수도 있다. 어레이 인덱스들 x0, y0는 고려된 변환 유닛의 상부-좌측 위치 (x0, y0)를 특정한다. 어레이 인덱스 trafoDepth는 변환 코딩의 목적을 위한 블록들로의 코딩 블록의 현재의 재분할 레벨을 특정한다. trafoDepth는 코딩 블록들에 대응하는 블록들에 대하여 0과 동일하다. cbf_cb[x0][y0][trafoDepth]가 존재하지 않을 때, cbf_cb[x0][y0][trafoDepth]의 값은 다음과 같이 추론될 수도 있다.

[0105] • transquant_bypass_default_flag가 1과 동일할 경우, cbf_cb[x0][y0][trafoDepth]는 1과 동일한 것으로 추론된다.

[0106] • 이와 다르게, trafoDepth가 0보다 더 크고 log2TrafoSize가 2와 동일할 경우, cbf_cb[x0][y0][trafoDepth]는 cbf_cb[xBase][yBase][trafoDepth - 1]와 동일한 것으로 추론된다.

[0107] • 이와 다를 경우, cbf_cb[x0][y0][trafoDepth]는 0과 동일한 것으로 추론된다.

[0108] 1과 동일한 cbf_cr[x0][y0][trafoDepth]의 값은, Cr 변환 블록이 0과 동일하지 않은 하나 이상의 변환 계수 레벨들을 포함한다는 것을 특정한다. 어레이 인덱스들 x0, y0는 고려된 변환 유닛의 상부-좌측 위치 (x0, y0)를 특정한다. 어레이 인덱스 trafoDepth는 변환 코딩의 목적을 위한 블록들로의 코딩 블록의 현재의 재분할 레벨을 특정한다. trafoDepth의 값은 코딩 블록들에 대응하는 블록들에 대하여 0과 동일하다.

[0109] cbf_cr[x0][y0][trafoDepth]가 존재하지 않을 때, cbf_cr[x0][y0][trafoDepth]의 값은 다음과 같이 추론될 수도 있다:

[0110] • transquant_bypass_default_flag가 1과 동일할 경우, cbf_cr[x0][y0][trafoDepth]는 1과 동일한 것으

로 추론된다.

[0111] · 이와 다르게, trafoDepth가 0보다 더 크고 log2TrafoSize가 2와 동일할 경우, cbf_cr[x0][y0][trafoDepth]는 cbf_cr[xBase][yBase][trafoDepth - 1]와 동일한 것으로 추론된다.

[0112] · 이와 다를 경우, cbf_cr[x0][y0][trafoDepth]는 0과 동일한 것으로 추론된다.

[0113] 표 7은 제안된 하이 레벨 플래그 transquant_bypass_default_flag를 이용하여 transform_tree() 신택스를 간략화하는 예를 도시하지만, 기존의 블록 레벨 플래그 cu_tranquant_bypass_flag는 동일한 로직을 따르는 split_transform_flag, cbf_luma, cbf_cb, 및 cbf_cr의 존재를 조건화하기 위하여 그 대신에 이용될 수도 있다.

표 7

1	transform_tree(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx) {	디스크립터
2†	if(!transquant_bypass_default_flag && log2TrafoSize <= Log2MaxTrafoSize && log2TrafoSize > Log2MinTrafoSize && trafoDepth < MaxTrafoDepth && !(IntraSplitFlag && (trafoDepth = = 0)))	
3	**split_transform_flag**[x0][y0][trafoDepth]	ae(v)
4†	if(!transquant_bypass_default_flag && log2TrafoSize > 2 && ChromaArrayType != 0) {	
5	if(trafoDepth = = 0 | | cbf_cb[xBase][yBase][trafoDepth - 1]) {	
6	**cbf_cb**[x0][y0][trafoDepth]	ae(v)
7	if(ChromaArrayType = = 2 && ! split_transform_flag[x0][y0][trafoDepth])	
8	**cbf_cb**[x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth]	
9	}	
10	if(trafoDepth = = 0 | | cbf_cr[xBase][yBase][trafoDepth - 1]) {	
11	**cbf_cr**[x0][y0][trafoDepth]	ae(v)
12	if(ChromaArrayType = = 2 && ! split_transform_flag[x0][y0][trafoDepth])	
13	**cbf_cr**[x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth]	
14	}	
15	}	
16	if(split_transform_flag[x0][y0][trafoDepth]) {	
17	x1 = x0 + (1 << (log2TrafoSize - 1))	
18	y1 = y0 + (1 << (log2TrafoSize - 1))	
19	transform_tree(x0, y0, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 0)	
20	transform_tree(x1, y0, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 1)	
21	transform_tree(x0, y1, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 2)	
22	transform_tree(x1, y1, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 3)	
23	} else {	
24†	if(!transquant_bypass_default_flag && ((CuPredMode[x0][y0] = = MODE_INTRA && intra_bc_flag[x0][y0] != 1) | | trafoDepth != 0 | | cbf_cb[x0][y0][trafoDepth] | | cbf_cr[x0][y0][trafoDepth] | | (ChromaArrayType = = 2 && (cbf_cb[x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth] | | cbf_cr[x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth]))))	
25	**cbf_luma**[x0][y0][trafoDepth]	ae(v)
26	transform_unit(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx)	
27	}	
28	}	

[0114]

[0115] **표 7: 간략화된 변환 트리 신택스.**

[0116] 표 7에서 예시된 실시형태에서는, 상이한 크기이며 상이한 코딩 모드들로 코딩된 CU들에 대하여, 쿼드트리 분리 플래그 split_transform_flag 및 코딩된 블록 플래그들 cbf_luma, cbf_cb, 및 cbf_cr은, 플래그 transquant_bypass_default_flag가 1과 동일할 때에 우회되고 대응하는 디폴트 값으로 추론된다. 그러나, 입력 비디오의 특성들에 따라서는, 상이한 블록 크기들 및 코딩 모드들을 갖는 CU들의 잔차들이 구분되는 통계적 특성들을 제시할 수 있다. 이 경우, 하나의 픽처 또는 시퀀스에서의 모든 CU들에 대한 변환 쿼드트리 분리를 디스에이블하는 것은 유익하지 않을 수도 있다. 그 대신에, 코딩 성능을 개선시키기 위하여, 개시물의 하나의 실시형태로서, 제안된 transquant_bypass_default_flag는 블록 크기, 코딩되어야 할 CU의 코딩 모드(즉, 인터, 인트라, 또는 IBC), 또는 블록 크기 및 코딩 모드의 조합에 따라, 변환 쿼드트리 분리 플래그 및/또는 코딩된 블록 플래그들의 시그널링을 조건부로 우회시키기 위하여 이용된다. 예를 들어, 복잡도 및 성능 절충의 측면에

서, 블록이 비-인트라 모드를 이용하여 코딩되고(즉, 인터 또는 IBC 모드를 이용하여 코딩됨) 블록 크기가 8x8 또는 16x16일 경우에 변환 쿼드트리 분리를 오직 허용하는 것이 유익할 수도 있다.

[0117] 하나의 실시형태에서, 다음은 플래그 transquant_bypass_default_flag가 1과 동일할 때에 적용한다. 모든 인트라-코딩된 CU들과, 64x64로부터 32x32까지의 블록 크기를 갖는 모든 인터-코딩되고 IBC-코딩된 CU들에 대하여, split_transform_flag 및/또는 코딩된 블록 플래그들(cbf_luma, cbf_cb, 및 cbf_cr)은 시그널링되지 않고; 이들은 위에서 논의된 바와 같이 대응하는 디폴트 값으로 추론될 것이다. 인터 모드 또는 IBC 모드로 코딩된 8x8 및 16x16 CU들에 대하여, 추가의 분리가 허용될 수도 있다. split_transform_flag 및/또는 코딩된 블록 플래그들(cbf_luma, cbf_cb, 및 cbf_cr)은, 현재의 블록이 4 개의 쿼드런트(quadrant)들로 추가로 파티셔닝되는지 여부, 및/또는 하나의 TU에서의 계수들이 모두 제로들인지 여부를 각각 표시하기 위하여 여전히 시그널링된다.

[0118] 일부의 실시형태들에서, 2 개의 신택스 엘리먼트들 log2_intra_max_no_transform_split_coding_block_size_minus3 및 log2_inter_max_no_transform_split_coding_block_size_minus3은, 인트라 및 인터/IBC 코딩된 CU들에 대하여 변환 쿼드트리 분리가 각각 적용되는 최대 CU 크기를 특정하기 위하여 SPS 또는 PPS에 추가된다. 예를 들어, 상기 조건들 1) 및 2)를 이용하면, log2_intra_max_no_transform_split_coding_block_size_minus3 및 log2_inter_max_no_transform_split_coding_block_size_minus3의 값들은 log2(64)-3 = 3 and log2(16)-3 = 1로 각각 설정된다. 표 8은 2 개의 제안된 신택스 엘리먼트들을 갖는 수정된 SPS 스크린 컨텐츠 코딩 확장 신택스 표를 도시한다.

[0119] 표 8은 2 개의 추가적인 신택스 엘리먼트들을 도시하지만, 이 개시물의 또 다른 실시형태에서, log2_intra_max_no_transform_split_coding_block_size_minus3의 값은 시그널링되지 않을 수도 있다. 그 대신에, 값은 (SPS에서의 신택스 엘리먼트 log2_diff_max_min_luma_coding_block_size에 의해 특정된 바와 같은) 최대 및 최소 CU 크기들 사이의 차이와 함께, (SPS에서의 신택스 엘리먼트 log2_min_luma_coding_block_size_minus3에 의해 특정된 바와 같은) 최소 CU 크기를 추가함으로써 유도될 수 있는 허용된 최대 CU 크기와 항상 동일한 것으로 추론될 수도 있다. 이러한 실시형태에서는, CU가 인트라 모드에서 코딩되고 무손실 코딩이 적용될 때, 변환 쿼드트리 분리가 허용되지 않는다.

표 8

1	sps_scc_extensions( ) {	**디스크립터**
2	**intra_block_copy_enabled_flag**	u(1)
3	**palette_mode_enabled_flag**	u(1)
4	if(palette_mode_enabled_flag) {	
5	**palette_max_size**	ue(v)
6	**palette_max_predictor_size**	ue(v)
7	}	
8	**adaptive_mv_resolution_enabled_flag**	u(1)
9	**intra_boundary_filtering_disabled_flag**	u(1)
10†	**log2_intra_max_no_transform_partition_coding_block_size_minus3**	ue(v)
11†	**log2_inter_max_no_transform_partition_coding_block_size_minus3**	ue(v)
12	}	

[0120]

[0121] **표 1 시퀀스 파라미터 세트 스크린 컨텐츠 코딩 신택스.**

[0122] log2_intra_max_no_transform_partition_coding_block_size_minus3의 값 플러스(plus) 3은, 코딩 유닛이 인트라 코딩될 때, 그리고 cu_transquant_bypass_flag가 1과 동일할 때에 변환 쿼드트리 분리가 적용되는 코딩 유닛들의 최대 블록 크기를 특정한다.

[0123] log2_inter_max_no_transform_partition_coding_block_size_minus3의 값 플러스 3은, 코딩 유닛이 인터 코딩되거나 인트라 블록 복사 코딩될 때, 그리고 cu_transquant_bypass_flag가 1과 동일할 때에 변환 쿼드트리 분리가 적용되는 코딩 유닛들의 최대 블록 크기를 특정한다.

[0124] 표 9는 현재의 CU의 블록 크기 및 코딩 모드들에 대해 조건화된 split_transform_flag, cbf_luma, cbf_cb, 및

cbf_cr의 제안된 시그널링 제약을 갖는 수정된 transform_tree() 신택스 표를 도시한다. 이 예시적인 실시형태는 split_transform_flag 및 cbf 시그널링을 허용할 것인지 여부를 조건화하기 위하여 CU 코딩 모드 및 CU 크기의 양자를 이용하지만, 다른 수정된 조건들이 이용될 수도 있다는 것에 주목한다. 예를 들어, 코딩 모드 및 블록 크기의 어느 하나(그러나 양자는 아님)가 이용될 수도 있다. 추가적으로, 별도의(그리고 상이한) 조건들이 split_transform_flag 시그널링 또는 cbf 시그널링에 적용될 수도 있다.

표 9

1	transform_tree(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx) {	**디스크립터**
2†	if((!cu_transquant_bypass_flag || ((CuPredMode[x0][y0] == MODE_INTRA && log2TrafoSize <= log2_intra_max_no_transform_partition_coding_unit_size_minus3+3) || (CuPredMode[x0][y0] != MODE_INTRA && log2TrafoSize <= log2_inter_max_no_transform_partition_coding_unit_size_minus3+3))) && log2TrafoSize <= Log2MaxTrafoSize && log2TrafoSize > Log2MinTrafoSize && trafoDepth < MaxTrafoDepth && !(IntraSplitFlag && (trafoDepth = = 0)))	
3	**split_transform_flag**[x0][y0][trafoDepth]	ae(v)
4†	if((!cu_transquant_bypass_flag || ((CuPredMode[x0][y0] == MODE_INTRA && log2TrafoSize <= log2_intra_max_no_transform_partition_coding_unit_size_minus3+3) || (CuPredMode[x0][y0] != MODE_INTRA && log2TrafoSize <= log2_inter_max_no_transform_partition_coding_unit_size_minus3+3))) && log2TrafoSize > 2 && ChromaArrayType != 0) {	
5	if(trafoDepth = = 0 || cbf_cb[xBase][yBase][trafoDepth − 1]) {	
6	**cbf_cb**[x0][y0][trafoDepth]	ae(v)
7	if(ChromaArrayType = = 2 && ! split_transform_flag[x0][y0][trafoDepth])	
8	**cbf_cb**[x0][y0 + (1 << (log2TrafoSize − 1))][trafoDepth]	
9	}	
10	if(trafoDepth = = 0 || cbf_cr[xBase][yBase][trafoDepth − 1]) {	
11	**cbf_cr**[x0][y0][trafoDepth]	ae(v)
12	if(ChromaArrayType = = 2 && ! split_transform_flag[x0][y0][trafoDepth])	
13	**cbf_cr**[x0][y0 + (1 << (log2TrafoSize − 1))][trafoDepth]	
14	}	
15	}	
16	if(split_transform_flag[x0][y0][trafoDepth]) {	
17	x1 = x0 + (1 << (log2TrafoSize − 1))	
18	y1 = y0 + (1 << (log2TrafoSize − 1))	
19	transform_tree(x0, y0, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 0)	
20	transform_tree(x1, y0, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 1)	
21	transform_tree(x0, y1, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 2)	
22	transform_tree(x1, y1, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 3)	
23	} else {	
24†	if((!cu_transquant_bypass_flag || ((CuPredMode[x0][y0] == MODE_INTRA && log2TrafoSize <= log2_intra_max_no_transform_partition_coding_unit_size_minus3+3) || (CuPredMode[x0][y0] != MODE_INTRA && log2TrafoSize <= log2_inter_max_no_transform_partition_coding_unit_size_minus3+3))) && ((CuPredMode[x0][y0] = = MODE_INTRA && intra_bc_flag[x0][y0] != 1) || trafoDepth != 0 || cbf_cb[x0][y0][trafoDepth] || cbf_cr[x0][y0][trafoDepth] || (ChromaArrayType = = 2 && (cbf_cb[x0][y0 + (1 << (log2TrafoSize − 1))][trafoDepth] || cbf_cr[x0][y0 + (1 << (log2TrafoSize − 1))][trafoDepth]))))	
25	**cbf_luma**[x0][y0][trafoDepth]	ae(v)
26	transform_unit(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx)	
27	}	
28	}	

[0125]

[0126] **표 2 수정된 변환 트리 신택스.**

[0127] 또 다른 실시형태에서는, 무손실의 경우에 있어서 split_transform_flag의 시그널링을 스킵하기 위하여, SPS에서의 신택스 엘리먼트들 max_transform_hierarchy_depth_inter 및 max_transform_hierarchy_depth_intra가 0으로 설정될 수 있다. 이 방법은 현재의 transform_tree() 신택스에 대한 로우-레벨 변경들을 요구하지 않고, 시퀀스 레벨 무손실 코딩의 경우에 transform_skip_flag의 시그널링을 우회시킨다. 이러한 실시형태에서는, split_transform_flag가 시그널링되지 않고, 그 대신에, TU 쿼드트리 분리가 집행되는 경우들(예를 들어, NxN 파티션이 인트라 코딩을 위하여 이용되거나, CU 크기가 최대 TU 크기보다 더 클 때 등등)을 제외하고는, 대부분의 경우들에 있어서 0인 디폴트 값인 것으로 추론된다. 따라서, 시퀀스 레벨 무손실 코딩이 적용될 때에

max_transform_hierarchy_depth_inter 및 max_transform_hierarchy_depth_intra가 0으로 적당하게 설정될 것을 요구하는 비트스트림 제약을 부과함으로써, transform_split_flag의 시그널링은 임의의 블록 레벨 변경들을 요구하지 않으면서 우회될 수도 있다. 유사하게, 최대 변환 크기 및 최소 변환 크기가 시퀀스 레벨 무손실 코딩의 경우에 동일해야 한다는 비트스트림 제약을 부과함으로써, transform_split_flag의 시그널링은 임의의 블록 레벨 변경들을 요구하지 않으면서 우회될 수도 있다. 이 제약은 시퀀스 레벨 무손실 코딩의 경우에 SPS 신택스 log2_diff_max_min_transform_block_size가 0으로 설정될 것을 요구함으로써 달성될 수 있다. max_transform_hierarchy_depth_inter, max_transform_hierarchy_depth_intra, 및 log2_diff_max_min_transform_block_size는 SPS에서 위치되므로, 이 경우에 SPS 내에 transquant_bypass_default_flag를 또한 넣는 것이 더욱 바람직할 수도 있다.

[0128] 개시물의 또 다른 실시형태에서는, 인코더-단독 방법이 플래그 transquant_bypass_default_flag를 추가하지 않는 무손실 코딩을 위하여 제안된다. 이러한 실시형태에서는, transquant_bypass_default_flag를 이용한 조건적 항들이 칼표(dagger)(†)로 표기되는 표 7 및 표 9에서의 신택스 엘리먼트들로부터 생략될 수 있다. 이 실시형태는 SPS에서의 신택스 엘리먼트들 max_transform_hierarchy_depth_inter 및 max_transform_hierarchy_depth_intra in the SPS, 또는 transform_tree()에서의 신택스 엘리먼트들의 값들의 수정을 요구하지 않는다. 인코딩 복잡도를 감소시키기 위하여, 플래그 split_tranform_flag는 인코더-단독 방법에 대하여 여전히 시그널링되지만, 표 7 또는 표 9에서 설명된 바와 같은 split_transform_flag의 디폴트 값에 의해 표시된 변환 쿼드트리 분리만이 각각의 CU에 대하여 테스트된다.

[0129] 일부의 실시형태들에서, 플래그 cu_transquant_bypass_flag가 현재의 CU에 대하여 1과 동일할 때, 인코더는 대부분의 경우들에 대한 무변환(no transform) 쿼드트리 파티션의 레이트-왜곡(rate-distortion; R-D) 성능과, NxN PU 파티션이 인트라-코딩된 CU에 적용되거나 현재의 CU 크기가 문턱값(threshold)보다 더 클 때와 같은 일부의 특수한 경우들에 대한 1회 변환 쿼드트리 파티션의 R-D 성능을 오직 테스트할 것이다. 이러한 방법으로, 이 인코더-단독 방법은 칼표(†)로 표기된 신택스 엘리먼트들에서 표 7에서의 플래그 split_transform_flag에 적용된 조건들과 부합한다.

[0130] 또 다른 실시형태에서, 플래그 cu_transfquant_bypass_flag가 1과 동일할 때, 인코더는 CU들이 정확하게 하나의 예측 유닛을 포함할 때, 64x64로부터 32x32까지의 블록 크기를 갖는 모든 인트라-코딩된 CU들 및 모든 인터/IBC-코딩된 CU들에 대한 무변환 쿼드트리 파티션의 R-D 성능을 오직 테스트할 것이고; 인코더는 CU들이 적어도 2 개의 예측 유닛들을 포함할 때, 64x64로부터 32x32까지의 블록 크기를 갖는 몯느 인트라-코딩된 CU들 및 모든 인터/IBC-코딩된 CU들에 대한 1회 변환 쿼드트리 파티션의 R-D 성능을 오직 테스트할 것이고; 블록 크기 16x16 및 8x8을 갖는 인터/IBC-코딩된 CU들에 대하여, 무변환 쿼드트리 분리 및 추가의 변환 쿼드트리 분리의 양자의 R-D 성능이 인코더에 의해 테스트된다. 따라서, 이 인코더-단독 방법은 표 9에서의 칼표(†)로 표기된 플래그 split_transform_flag에 적용된 조건들과 부합한다. 추가적으로, 플래그 split_transform_flag는 상기 인코더-단독 방법들에 대하여 비트-스트림에서 여전히 시그널링되지만, 하나의 비트스트림 준수 제약은 신택스 엘리먼트 split_transform_flag의 값들이 블록 크기 및 블록 코딩 모드에 따라 그 디폴트 값들로 설정될 것으을 요구하기 위하여 적용될 수도 있다.

[0131] 일부의 실시형태들에서, 변환 쿼드트리 분리 플래그의 디폴트 값의 결정은 관련된 코딩 유닛에서의 예측 유닛들의 수에 적어도 부분적으로 기초한다. 하나의 이러한 실시형태에서, 디폴트 값의 결정은 코딩 유닛이 인트라 코딩되는지 여부, 코딩 유닛의 크기가 크기 문턱값보다 더 큰지 여부, 및 코딩 유닛이 정확하게 하나의 예측 유닛을 포함하는지 여부를 결정하는 것을 포함한다. 코딩 유닛이 인트라 코딩되지 않고, 코딩 유닛이 크기 문턱값보다 더 크고, 코딩 유닛이 정확하게 하나의 예측 유닛을 포함한다는 결정에 응답하여, 변환 쿼드트리 분리 플래그의 디폴트 값은 무변환 쿼드트리 파티션을 표시하기 위하여 설정된다.

[0132] 또 다른 이러한 실시형태에서, 변환 쿼드트리 분리 플래그의 디폴트 값의 결정은 관련된 코딩 유닛이 인트라 코딩되는지 여부, 코딩 유닛의 크기가 크기 문턱값보다 더 큰지 여부, 및 제 1 코딩 유닛이 적어도 2 개의 예측 유닛들을 포함하는지 여부를 결정하는 것을 포함한다. 코딩 유닛이 인트라 코딩되지 않고, 코딩 유닛이 크기 문턱값보다 더 크고, 코딩 유닛이 적어도 2 개의 예측 유닛들을 포함한다는 결정에 응답하여, 변환 쿼드트리 분리 플래그의 디폴트 값은 1회 변환 쿼드트리 파티션을 표시하기 위하여 설정된다.

[0133] 추가의 실시형태에서, 변환 쿼드트리 분리 플래그의 디폴트 값의 결정은 관련된 코딩 유닛이 인트라 코딩되는지 여부와, 제 1 코딩 유닛이 정확하게 하나의 예측 유닛을 포함하는지 여부를 결정하는 것을 포함한다. 코딩 유닛이 인트라 코딩되고 정확하게 하나의 예측 유닛을 포함함다는 결정에 응답하여, 변환 쿼드트리 분리 플래그의

디폴트 값은 무변환 쿼드트리 파티션을 표시하기 위하여 설정된다.

[0134] 또 다른 실시형태에서, 변환 쿼드트리 분리 플래그의 디폴트 값을 결정하는 것은 관련된 코딩 유닛이 인트라 코딩되는지 여부와, 코딩 유닛이 적어도 2 개의 예측 유닛들을 포함하는지 여부를 결정하는 것을 포함한다. 코딩 유닛이 인트라 코딩되지 않고 코딩 유닛이 적어도 2 개의 예측 유닛들을 포함한다는 결정에 응답하여, 변환 쿼드트리 분리 플래그의 디폴트 값은 1회 변환 쿼드트리 파티션을 표시하기 위하여 설정된다.

[0135] 코딩 유닛이 정확하게 하나의 예측 유닛 또는 적어도 2 개의 예측 유닛들을 포함하는지 여부의 결정은 코딩 유닛의 파티션 모드를 결정함으로써 행해질 수도 있다. 예를 들어, 2Nx2N 파티션 모드를 이용하는 코딩 유닛은 정확하게 하나의 예측 유닛을 포함하는 반면, 예를 들어, 2NxN, Nx2N, 2NxnU, 2NxnD, nLx2N, nRx2N, 또는 NxN의 파티션 모드들을 이용하는 코딩 유닛은 적어도 2 개의 예측 유닛들을 포함한다.

[0136] ***인트라 블록 복사 모드에 대한 변환 트리 분리.***

[0137] 변환 쿼드트리 분리의 최대 심도는 인코딩 및 디코딩 복잡도에 밀접하게 관련된다. 코딩 효율과 연산 복잡도 사이의 유연한 절충을 제공하기 위하여, HEVC 및 그 확장들은 TU 크기들 및 TU 분리 심도를 특정하기 위하여 SPS에서의 신택스 엘리먼트들을 이용한다. 값들 log2_min_luma_transform_block_size_minus2 및 log2_diff_max_min_luma_transform_block_size는 비디오 시퀀스를 코딩하기 위하여 이용된 TU 크기들의 세트를 표시하고, max_transform_hierarchy_depth_inter 및 max_transform_hierarchy_depth_intra는 인트라 및 인터 코딩된 CU들에 대한 최대 분리 심도를 각각 표시한다. 어떤 조건들에서는, 변환 쿼드트리 분리가 적용되지 않을 수도 있다. 예를 들어, max_transform_hierarchy_depth_intra/inter가 0으로 설정될 경우, 변환 쿼드트리 분린느 현재의 인트라/인터 코딩된 CU에 적용되지 않는다.

[0138] HEVC, 그 범위 확장, 및 SCC 초안에서는, 변환 쿼드트리 분리가 디스에이블될 때, 다수의 예측 유닛들(PU)로 파티셔닝되고 인터 모드에서 코딩되는 CU들에 대하여, split_transform_flag의 값이 1(즉, 변환 쿼드트리 분리가 적용됨)인 것으로 항상 추론되도록, 하나의 묵시적 TU 파티션 방법이 손실 및 무손실 코딩의 양자에서 적용된다. 이것은 CU 내부의 PU들의 상이한 모션 벡터들이 인위적인 높은 주파수 정보를 야기시킬 수도 있고, 이웃하는 PU들 사이의 경계들에 걸쳐 부합하지 않는 잔차들을 초래할 수도 있다. 이 경우, CU를 더 작은 TU들로 분리하는 것은 CU의 TU 크기만큼 큰 TU 크기를 이용하는 것보다 더욱 양호한 코딩 효율을 제공할 수 있다.

[0139] HEVC 스크린 컨텐츠 코딩 확장의 작업 초안 [0057]에서는, 상기 묵시적 TU 파티션이 IBC 모드에서 코딩된 CU들에 적용되지 않는다. 더욱 구체적으로, 변환 쿼드트리 분리가 디스에이블될 때, split_transform_flag의 값은 모든 IBC-코딩된 CU들에 대하여 0(즉, 변환 유닛 크기가 CU의 변환 유닛 크기와 동일한 것으로 설정됨)인 것으로 항상 추론된다. IBC 모드와 인터 모드 사이에 본질적인 유사성이 주어지면, IBC-코딩된 CU들의 잔차들은 인터-코딩된 CU들의 특성과 유사한 특성들을 제시할 수도 있다. 그러므로, 변환 코딩의 효율을 추가로 개선시키기 위하여, (즉, split_transform_flag의 값을 1인 것으로 추론하기 위한) 묵시적 TU 파티션을 IBC-코딩된 CU들에 적용하는 것이 또한 유익할 수도 있다. 본원에서 설명된 실시형태에서는, 변환 쿼드트리 파티션이 디스에이블될 때, 인터 모드에 적용되는 동일한 묵시적 TU 파티션 방법이 또한, 손실 코딩 및 무손실 코딩의 양자에서 IBC 코딩된 CU들에 대하여 이용된다. 다시 말해서, split_transform_flag의 값은, 하나를 초과하는 PU 파티션들(예를 들어, 2NxN, Nx2N, 및 NxN)이 IBC 모드로 코딩된 현재의 CU에서 존재할 때에 1(즉, 변환 쿼드트리가 분리됨)인 것으로 추론된다.

[0140] split_transform_flag의 값의 유도 프로세스는 [0057]의 섹션 7.4.9.9에서 특정된다. 본원에서 개시된 예시적인 실시형태들에서의 IBC 모드에 대한 묵시적 TU 파티션의 인에이블에 의하여, split_transform_flag의 시맨틱들은 다음과 같이 동작한다.

[0141] 어레이 split_transform_flag[x0][y0][trafoDepth]는 블록이 변환 코딩의 목적을 위하여 절반의 수평 및 절반의 수직 크기를 갖는 4 개의 블록들로 분리되는지 여부를 특정한다. 어레이 인덱스들 x0, y0는 픽처의 상부-좌측 루마 샘플에 관하여 고려된 블록의 상부-좌측 루마 샘플의 위치 (x0, y0)를 특정한다. 어레이 인덱스 trafoDepth는 변환 코딩의 목적을 위한 블록들로의 코딩 블록의 현재의 재분할 레벨을 특정한다. trafoDepth의 값은 코딩 블록들에 대응하는 블록들에 대하여 0과 동일하다.

[0142] 변수 interSplitFlag는 다음과 같이 유도된다. InterSplitFlag는 다음의 조건들 중의 하나 이상이 적용될 경우에 1과 동일하게 설정된다: max_transform_hierarchy_depth_inter가 0과 동일하고 CuPredMode[x0][y0]가 MODE_INTER와 동일하거나; 또는 intra_bc_flag[x0][y0]가 1과 동일하고, PartMode가 PART_2Nx2N과 동일하지 않고, trafoDepth가 0과 동일하다. 이와 다를 경우, interSplitFlag는 0과 동일하게 설정된다.

[0143] split_transform_flag[x0][y0][trafoDepth]가 존재하지 않을 때, 그 값은 다음과 같이 추론된다. 다음의 조건들 중의 하나 이상이 참(true)일 경우, split_transform_flag[x0][y0][trafoDepth]의 값은 1과 동일한 것으로 추론된다: log2TrafoSize가 MaxTbLog2SizeY보다 더 크고; IntraSplitFlag가 1과 동일하고 trafoDepth가 0과 동일하거나; 또는 interSplitFlag가 1과 동일하다. 이와 다를 경우, split_transform_flag[x0][y0][trafoDepth]의 값은 0과 동일한 것으로 추론된다.

[0144] **실시형태들**

[0145] 예시적인 실시형태에서는, 슬라이스 세그먼트 헤더 및 복수의 코딩 유닛들을 포함하는 비디오 슬라이스를 코딩하는 방법이 제공된다. 방법은 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는지 여부를 표시하는 슬라이스 세그먼트 헤더에서의 우회 플래그를 생성하는 것을 포함한다.

[0146] 일부의 이러한 실시형태들에서, 우회 플래그는 transquant_bypass_default_flag이다. 방법은 transquant_bypass_enabled_flag를 포함하는 픽처 파라미터 세트(PPS)를 생성하는 것을 포함할 수도 있고, 여기서, 슬라이스는 픽처 파라미터 세트를 참조하고, 여기서, transquant_bypass_enabled_flag는 transquant_bypass_default_flag가 1로 설정될 대에 제로로 설정된다.

[0147] 일부의 실시형태들에서, 우회 플래그는 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하고, 방법은 슬라이스 세그먼트 헤더에서, 손실 코딩에 관련된 신택스 엘리먼트들을 생성하는 것을 더 포함한다. 우회 플래그는 손실 코딩에 관련된 신택스 엘리먼트들의 전방에 위치될 수도 있다.

[0148] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더에서, 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련된 신택스 엘리먼트들을 생성하는 것을 더 포함한다. 우회 플래그는 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련된 신택스 엘리먼트들의 전방에 위치될 수도 있다.

[0149] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더에서, slice_sao_luma_flag, slice_sao_chroma_flag, slice_qp_delta, slice_cb_qp_offset, slice_cr_qp_offset, slice_chroma_qp_adjustment_enabled_flag, deblocking_filter_override_flag, slice_deblocking_filter_disabled_flag, slice_beta_offset_div2, slice_tc_offset_div2, 및 slice_loop_filter_across_slices_enabled_flag로 구성되는 그룹으로부터 선택된 하나 이상의 신택스 엘리먼트들을 생성하는 것을 더 포함한다. 우회 플래그는 slice_sao_luma_flag, slice_sao_chroma_flag, slice_qp_delta, slice_cb_qp_offset, slice_cr_qp_offset, slice_chroma_qp_adjustment_enabled_flag, deblocking_filter_override_flag, slice_deblocking_filter_disabled_flag, slice_beta_offset_div2, slice_tc_offset_div2, 및 slice_loop_filter_across_slices_enabled_flag로 구성되는 그룹으로부터 선택된 하나 이상의 신택스 엘리먼트들의 전방에 위치될 수도 있다.

[0150] 일부의 실시형태들에서, 우회 플래그는 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하고, 방법은 슬라이스 세그먼트 헤더로부터, 손실 코딩에 관련된 신택스 엘리먼트들을 제외하는 것을 더 포함한다.

[0151] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더로부터, 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련된 신택스 엘리먼트들을 제외하는 것을 더 포함한다.

[0152] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더로부터, slice_sao_luma_flag, slice_sao_chroma_flag, slice_qp_delta, slice_cb_qp_offset, slice_cr_qp_offset, slice_chroma_qp_adjustment_enabled_flag, deblocking_filter_override_flag, slice_deblocking_filter_disabled_flag, slice_beta_offset_div2, slice_tc_offset_div2, 및 slice_loop_filter_across_slices_enabled_flag로 구성되는 그룹으로부터 선택된 하나 이상의 신택스 엘리먼트들을 제외하는 것을 더 포함한다.

[0153] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스에서의 각각의 코딩 유닛에 대하여 cu_transquant_bypass_flag를 시그널링하는 것을 더 포함한다.

[0154] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스에서의 각각의 코딩 유닛으로부터 cu_transquant_bypass_flag를 제외하는 것을 더 포함한다.

[0155] 예시적인 실시형태에서는, 픽처 파라미터 세트와, 픽처 파라미터 세트(PPS)를 참조하는 적어도 하나의 슬라이스를 포함하는 비디오를 코딩하는 방법이 제공되고, 여기서, 슬라이스는 슬라이스 세그먼트 헤더 및 복수의 코딩 유닛들을 포함한다. 이 실시형태에서, 방법은 픽처 파라미터 세트를 참조하는 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는지 여부를 표시하는 픽처 파라미터 세트에서의 우회 플래그를 생성하는 것을 포함한다.

[0156] 일부의 이러한 실시형태들에서, 우회 플래그는 transquant_bypass_default_flag이다. 픽처 파라미터 세트(PPS)는 transquant_bypass_enabled_flag를 포함하고, transquant_bypass_enabled_flag는 transquant_bypass_default_flag가 1로 설정될 때에 제로로 설정된다.

[0157] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더에서, 손실 코딩에 관련된 신택스 엘리먼트들을 생성하는 것을 더 포함한다.

[0158] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더에서, 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련된 신택스 엘리먼트들을 생성하는 것을 더 포함한다.

[0159] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더에서, slice_sao_luma_flag, slice_sao_chroma_flag, slice_qp_delta, slice_cb_qp_offset, slice_cr_qp_offset, slice_chroma_qp_adjustment_enabled_flag, deblocking_filter_override_flag, slice_deblocking_filter_disabled_flag, slice_beta_offset_div2, slice_tc_offset_div2, 및 slice_loop_filter_across_slices_enabled_flag로 구성되는 그룹으로부터 선택된 하나 이상의 신택스 엘리먼트들을 생성하는 것을 더 포함한다.

[0160] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더로부터, 손실 코딩에 관련된 신택스 엘리먼트들을 제외하는 것을 더 포함한다.

[0161] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더로부터, 양자화, 변환, 및 인-루프 필터링 프로세스들에 관련된 신택스 엘리먼트들을 제외하는 것을 더 포함한다.

[0162] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스 세그먼트 헤더로부터, slice_sao_luma_flag, slice_sao_chroma_flag, slice_qp_delta, slice_cb_qp_offset, slice_cr_qp_offset, slice_chroma_qp_adjustment_enabled_flag, deblocking_filter_override_flag, slice_deblocking_filter_disabled_flag, slice_beta_offset_div2, slice_tc_offset_div2, 및 slice_loop_filter_across_slices_enabled_flag로 구성되는 그룹으로부터 선택된 하나 이상의 신택스 엘리먼트들을 제외하는 것을 포함한다.

[0163] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는 것은 아니라는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스에서의 각각의 코딩 유닛에 대하여 cu_transquant_bypass_flag를 시그널링하는 것을 더 포함한다.

[0164] 우회 플래그가 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩된다는 것을 표시하는 일부의 실시형태들에서, 방법은 슬라이스에서의 각각의 코딩 유닛으로부터 cu_transquant_bypass_flag를 제외하는 것을 더 포함한다.

[0165] 예시적인 실시형태에서는, 비디오를 코딩하는 방법이 제공되고, 여기서, 비디오는 하이 레벨 신택스 구조와, 하이 레벨 신택스 구조를 참조하는 적어도 하나의 슬라이스를 포함하고, 슬라이스는 복수의 코딩 유닛들을 포함한다. 방법은 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는지 여부를 표시하는 하이 레벨 신택스 구조에서의 우회 플래그를 생성하는 것을 포함한다. 각각의 개개의 코딩 유닛에 대하여, 쿼드트리 분리 플래그를 생성할 것인지 여부에 대해 결정이 행해진다. 결정은 개개의 코딩 유닛의 블록 크기 및 코딩 모드로

구성되는 그룹으로부터 선택된 파라미터에 적어도 부분적으로 기초한다. 쿼드트리 분리 플래그는 쿼드트리 분리 플래그를 생성하기 위한 결정을 행한 후에만 개개의 코딩 유닛에 대하여 생성된다.

[0166] 이러한 실시형태에서, 하이 레벨 신택스 구조는 픽처 파라미터 세트, 세그먼트 슬라이스 헤더, 또는 split_transform_flag일 수도 있다. 일부의 이러한 실시형태들에서, 쿼드트리 분리 플래그는, 블록이 비-인트라 모드를 이용하여 코딩되고 블록 크기가 8x8 또는 16x16일 경우에만 생성된다. 쿼드트리 분리 플래그는, 블록 크기가 64x64로부터 32x32까지일 경우에 생성되지 않는다.

[0167] 일부의 실시형태들은 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 하이 레벨 신택스 엘리먼트를 생성하는 것을 더 포함한다. 일부의 실시형태들에서, 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 하이 레벨 신택스 엘리먼트는 시퀀스 파라미터 세트(SPS)에서 생성된다. 일부의 실시형태들에서, 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 하이 레벨 신택스 엘리먼트는 픽처 파라미터 세트(PPS)에서 생성된다.

[0168] 일부의 실시형태들에서, 하이 레벨 신택스 엘리먼트는 비-인트라 모드에서 코딩된 블록들에 대한 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하기 위하여 이용된다.

[0169] 일부의 실시형태들에서, 쿼드트리 분리 플래그는, 블록이 비-인트라 모드를 이용하여 코딩되고 블록 크기가 비-인트라 모드에서 코딩된 블록들에 대한 최대 무변환 쿼드트리 분리 블록 코딩 크기보다 더 크지 않을 경우에 생성된다.

[0170] 일부의 실시형태들에서, 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 하이 레벨 신택스 엘리먼트는 인트라 모드에서 코딩된 블록들에 대한 최대 무변환 분리 블록 코딩 크기를 표시한다.

[0171] 일부의 실시형태들에서, 쿼드트리 분리 플래그는, 블록이 인트라 모드를 이용하여 코딩되고 블록 크기가 인트라 모드에서 코딩된 블록들에 대한 최대 무변환 쿼드트리 분리 블록 코딩 크기보다 더 크지 않을 경우에 생성된다.

[0172] 예시적인 실시형태에서는, 비디오를 코딩하는 방법이 제공되고, 여기서, 비디오는 하이 레벨 신택스 구조와, 하이 레벨 신택스 구조를 참조하는 적어도 하나의 슬라이스를 포함하고, 슬라이스는 복수의 코딩 유닛들을 포함한다. 방법은 슬라이스에서의 코딩 유닛들의 전부가 무손실 코딩으로 코딩되는지 여부를 표시하는 하이 레벨 신택스 구조에서의 우회 플래그를 생성하는 것을 포함한다. 각각의 개개의 코딩 유닛에 대하여, 개개의 코딩 유닛의 블록 크기 및 코딩 모드로 구성되는 그룹으로부터 선택된 파라미터에 적어도 부분적으로 기초하여 코딩된 블록 플래그를 생성할 것인지 여부에 대해 결정이 행해진다. 코딩된 블록 플래그는 코딩된 블록 플래그를 생성하기 위한 결정을 행한 후에만 개개의 코딩 유닛에 대하여 생성된다.

[0173] 이러한 실시형태에서, 하이 레벨 신택스 구조는 픽처 파라미터 세트 또는 세그먼트 슬라이스 헤더일 수도 있다. 코딩된 블록 플래그는 cbf_luma flag, cbf_cb flag, 또는 cbf_cr flag 중의 하나 이상일 수도 있다. 일부의 실시형태들에서, 코딩된 블록 플래그는, 블록이 비-인트라 모드를 이용하여 코딩되고 블록 크기가 8x8 또는 16x16일 경우에만 생성되는 반면; 코딩된 블록 플래그는, 블록이 인트라 모드를 이용하여 코딩될 경우, 또는 블록 크기가 64x64로부터 32x32까지일 경우에는 생성되지 않는다.

[0174] 일부의 실시형태들에서는, 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 하이 레벨 신택스 엘리먼트가 생성된다. 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 하이 레벨 신택스 엘리먼트는 시퀀스 파라미터 세트(SPS) 또는 픽처 파라미터 세트(PPS)에서 생성될 수도 있다.

[0175] 일부의 실시형태들에서, 하이 레벨 신택스 엘리먼트는 비-인트라 모드에서 코딩된 블록들에 대한 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시하는 최대 무변환 쿼드트리 분리 블록 코딩 크기를 표시한다. 코딩된 블록 플래그는, 블록이 비-인트라 모드를 이용하여 코딩되고 블록 크기가 인터 모드에서 코딩된 블록들에 대한 최대 무변환 분리 블록 코딩 크기보다 더 크지 않을 경우에 생성된다.

[0176] 일부의 실시형태들에서, 최대 무변환 분리 블록 코딩 크기를 표시하는 하이 레벨 신택스 엘리먼트는 인트라 모드에서 코딩된 블록들에 대한 최대 무변환 분리 블록 코딩 크기를 표시한다.

[0177] 일부의 실시형태들에서, 코딩된 블록 플래그는, 블록이 인트라 모드를 이용하여 코딩되고 블록 크기가 인트라 모드에서 코딩된 블록들에 대한 최대 무변환 분리 블록 코딩 크기보다 더 크지 않을 경우에 생성된다.

[0178] 예시적인 실시형태에서는, 비디오를 코딩하는 방법이 제공되고, 여기서, 비디오는 하이 레벨 신택스 구조와, 하이 레벨 신택스 구조를 참조하는 적어도 하나의 슬라이스를 포함하고, 슬라이스는 복수의 코딩 유닛들을 포함한

다. 방법은 플래그 cu_transquant_bypass_flag가 코딩 유닛에 대하여 1과 동일한 것으로 결정하는 것을 포함한다. 결정은 1회 변환 쿼드트리 파티션의 레이트-왜곡 성능을 테스트하기 위하여, 코딩 유닛의 코딩 유닛 크기 및 코딩 모드 중의 적어도 하나에 기초하여 행해진다. 결정 후에, 레이트-왜곡 성능은 1회 변환 쿼드트리 파티션에 대해 테스트된다.

[0179] 일부의 이러한 실시형태들에서, 방법은 추가적인 코딩 유닛들에 대하여, 1회 변환 쿼드트리 파티션의 레이트-왜곡 성능을 테스트하지 않기 위한 결정을 행하는 것을 더 포함한다. 레이트-왜곡 성능에 대해 테스트하는 것은 추가적인 코딩 유닛들에 대한 무변환 쿼드트리 파티션에 대해서만 수행된다.

[0180] 예시적인 실시형태에서는, 비디오를 코딩하는 방법이 제공되고, 여기서, 비디오는 split_transform_flag를 포함하는 변환 트리 신택스를 포함하고, 복수의 코딩 유닛들을 더 포함한다. 방법은 개개의 코딩 유닛에 대하여, 코딩 유닛이 인트라-블록 복사 모드에서 코딩되는지 여부와, 하나를 초과하는 예측 유닛 파티션이 코딩 유닛에서 존재하는지 여부를 결정하는 것을 포함한다. 코딩 유닛이 인트라-블록 복사 모드에서 코딩되고 하나를 초과하는 예측 유닛 파티션이 코딩 유닛에서 존재한다는 결정에 응답하여, split_transform_flag의 값은 1인 것으로 추론된다.

[0181] 예시적인 방법에서는, 변환 및 양자화 프로세스가 우회된다는 것을 표시하는 코딩-유닛 플래그에 대한 디폴트 값을 포함하는 하이 레벨 신택스 구조가 생성된다.

[0182] 일부의 이러한 실시형태들에서, 디폴트 값, transquant_bypass_default_flag는 1로 설정되어, 하이 레벨 신택스 구조를 참조하는 슬라이스에서의 모든 코딩 유닛들의 cu_transquant_bypass_flag의 디폴트 값을 표시한다. 하이 레벨 신택스 구조는 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS), 비디오 파라미터 세트(VPS), 또는 슬라이스 세그먼트 헤더 중의 적어도 하나일 수도 있다.

[0183] 일부의 실시형태들에서, 방법은 0과 동일한 transquant_bypass_enabled_flag를 갖는 PPS를 포함하는 비트 스트림을 생성하는 것을 포함한다. 일부의 실시형태들에서, 방법은 비트 스트림을 생성하는 것을 포함하고, 여기서, 코딩 유닛 파라미터들은 cu_transquant_bypass_flag를 포함하지 않는다.

[0184] 예시적인 실시형태에서는, 특정한 하이 레벨 신택스 구조를 참조하는 모든 CU들에 대하여 변환, 변환 스킵, 양자화, 및 인-루프 필터링 프로세스를 우회시키기 위하여, 특정한 하이 레벨 신택스 구조에서 하이 레벨 신택스 엘리먼트를 통해 디코더에 시그널링하는 방법이 제공된다.

[0185] 또 다른 예시적인 실시형태에서는, 역양자화, 역변환, 및 인-루프 필터링 프로세스 중의 임의의 하나 이상에 관련된 복수의 PPS 신택스 엘리먼트들의 존재를 식별하기 위하여 하이 레벨 신택스 엘리먼트를 수신하고 프로세싱하하도록 디코더를 동작시키는 방법이 제공된다.

[0186] 일부의 이러한 방법들에서, 하이 레벨 신택스 엘리먼트는 디폴트 플래그 값(transquant_bypass_default_flag)이다. 일부의 이러한 방법들에서, 하이 레벨 신택스 엘리먼트는 변환 스킵 인에이블된 플래그 엘리먼트의 존재를 식별하기 위하여 이용된다.

[0187] 일부의 실시형태들에서, 방법은 transquant_bypass_enabled_flag가 추론에 의해 0인 것으로 결정하는 것과, 이에 응답하여, CU 레벨에서 cu_transquant_bypass_flag의 시그널링을 제거하거나 스킵하는 것을 포함한다. 일부의 실시형태들에서, 디폴트 플래그는 PPS 확장 파라미터 세트, 또는 SPS 확장 파라미터 세트 중의 적어도 하나 내에 포함된다.

[0188] 일부의 실시형태들에서, 디폴트 플래그의 존재는 transquant_bypass_enabled_flag의 값에 조건화된다. 일부의 실시형태들에서, 디폴트 플래그는 transquant_bypass_default_flag이다.

[0189] 일부의 실시형태들에서, 방법은 무손실 코딩 모드에서 이용되지 않은 신택스 엘리먼트들의 시그널링된 값들이 적당하게 설정된다는 것을 표시하기 위하여 추가적인 준수 플래그의 시그널링을 수신하는 것을 더 포함한다.

[0190] 일부의 실시형태들에서, cu_qp_delta_enabled_flag, pps_loop_filter_across_slices_enabled_flag, deblocking_filter_control_present_flag, loop_filter_across_tiles_enabled_flag, pps_scaling_list_data_present_flag를 포함하는 신택스 엘리먼트들은, 플래그 transquant_bypass_default_flag가 1로 설정될 경우에 0으로 설정된다.

[0191] 예시적인 실시형태에서는, 무손실 코딩이 이용된다는 것을 표시하는 하이-레벨 시그널링 무손실 코딩 신택스 엘리먼트를 포함하는 비디오 데이터 비트 스트림을 수신하는 방법이 제공된다.

[0192] 일부의 이러한 방법들에서, 하이-레벨 시그널링 신택스는 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS), 비디오 파라미터 세트(VPS), 또는 슬라이스 세그먼트 헤더 중의 하나이다. 일부의 이러한 방법들에서, 무손실 코딩 신택스 엘리먼트는 양자화, 변환, 변환 스킵, 변환 스킵 회전, 및 인-루프 필터링 프로세스들에 관련된 하나 이상의 SPS 신택스 엘리먼트들을 제시하기 위한 조건으로서 이용된다.

[0193] 예시적인 실시형태에서는, 슬라이스 세그먼트 헤더가 수신되고, 디폴트 플래그는 변환, 양자화, 및 인-루프 필터링 프로세싱 블록들을 위하여 이용된 슬라이스 세그먼트 신택스 엘리먼트들의 식별을 조건화하기 위한 것이다. 일부의 이러한 방법들에서, 디폴트 플래그는 transquant_bypass_default_flag이다.

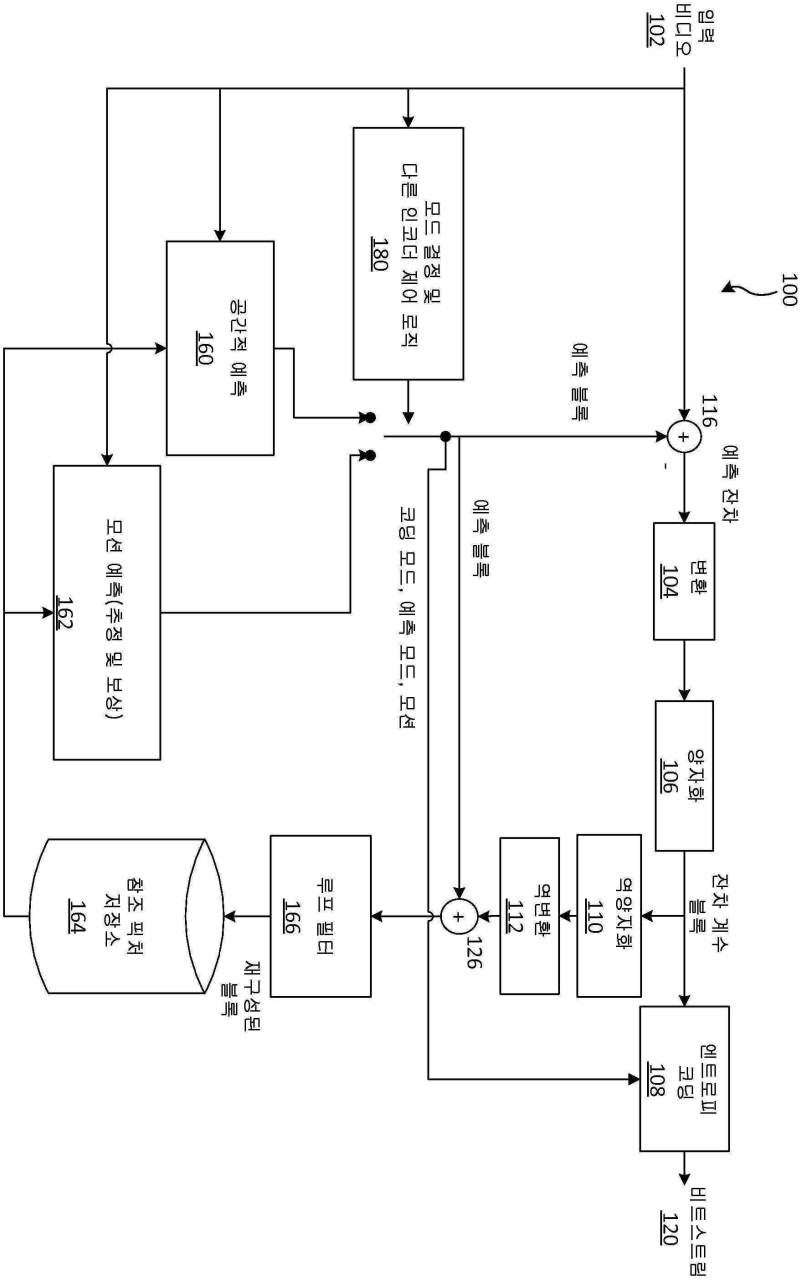
[0194] 예시적인 실시형태에서는, 하이-레벨 무손실 코딩 표시를 수신하고, 이에 응답하여, 복수의 프로세싱 블록들을 정지시키는 방법이 비디오 디코더에서 수행된다. 일부의 이러한 실시형태들에서, 하이-레벨 무손실 코딩 표시는 PPS, SPS, VPS, 또는 슬라이스 헤더 중의 하나에 대한 파라미터 엘리먼트이다. 일부의 실시형태들에서, 복수의 프로세싱 블록들은 다음의 하드웨어 블록들 중의 임의의 것의 하나 이상을 포함한다: 역양자화, 역변환, 디블록킹 필터, SAO.

[0195] 상기한 기법들 중의 임의의 것을 이용하여 인코딩된 비디오는 임의의 적절한 유선 또는 무선 송신 매체를 이용하여 송신될 수도 있고, 및/또는 임의의 적절한 비-일시적 디지털 저장 매체 상에 레코딩될 수도 있다.

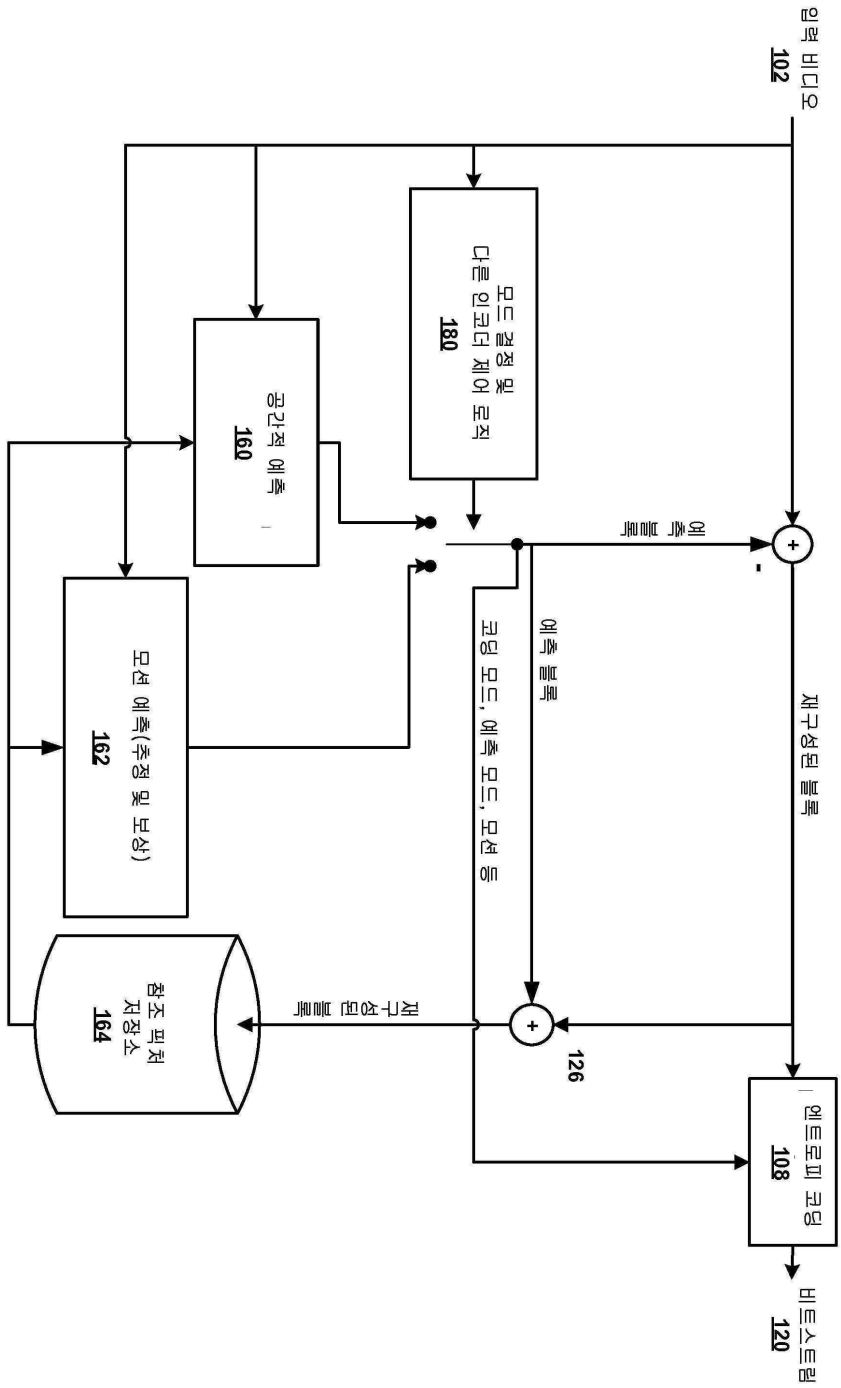
[0196] 특징들 및 엘리먼트들이 특별한 조합들로 위에서 설명되지만, 당해 분야의 당업자는 각각의 특징 및 엘리먼트가 단독으로, 또는 다른 특징들 및 엘리먼트들과의 임의의 조합으로 이용될 수 있다는 것을 인식할 것이다. 게다가, 본원에서 설명된 방법들은 컴퓨터 또는 프로세서에 의한 실행을 위하여 컴퓨터-판독가능 매체 내에 통합된 컴퓨터 프로그램, 소프트웨어, 또는 펌웨어에서 구현될 수도 있다. 컴퓨터-판독가능 매체들의 예들은 (유선 또는 무선 접속들을 통해 송신된) 전자 신호들 및 컴퓨터-판독가능 저장 매체들을 포함한다. 컴퓨터-판독가능 저장 매체들의 예들은 판독전용 메모리(ROM), 랜덤 액세스 메모리(RAM), 레지스터, 캐시 메모리, 반도체 메모리 디바이스들, 내부 하드 디스크들 및 분리가능 디스크들과 같은 자기 매체들, 자기-광 매체들, 및 CD-ROM 디스크들 및 디지털 다기능 디스크(DVD)들과 같은 광학 매체들을 포함하지만, 이것으로 제한되지 않는다. 소프트웨어와 연관된 프로세서는 WTRU, UE, 단말, 기지국, RNC, 또는 임의의 호스트 컴퓨터에서 이용하기 위한 라디오 주파수 트랜시버를 구현하기 위하여 이용될 수도 있다.

도면

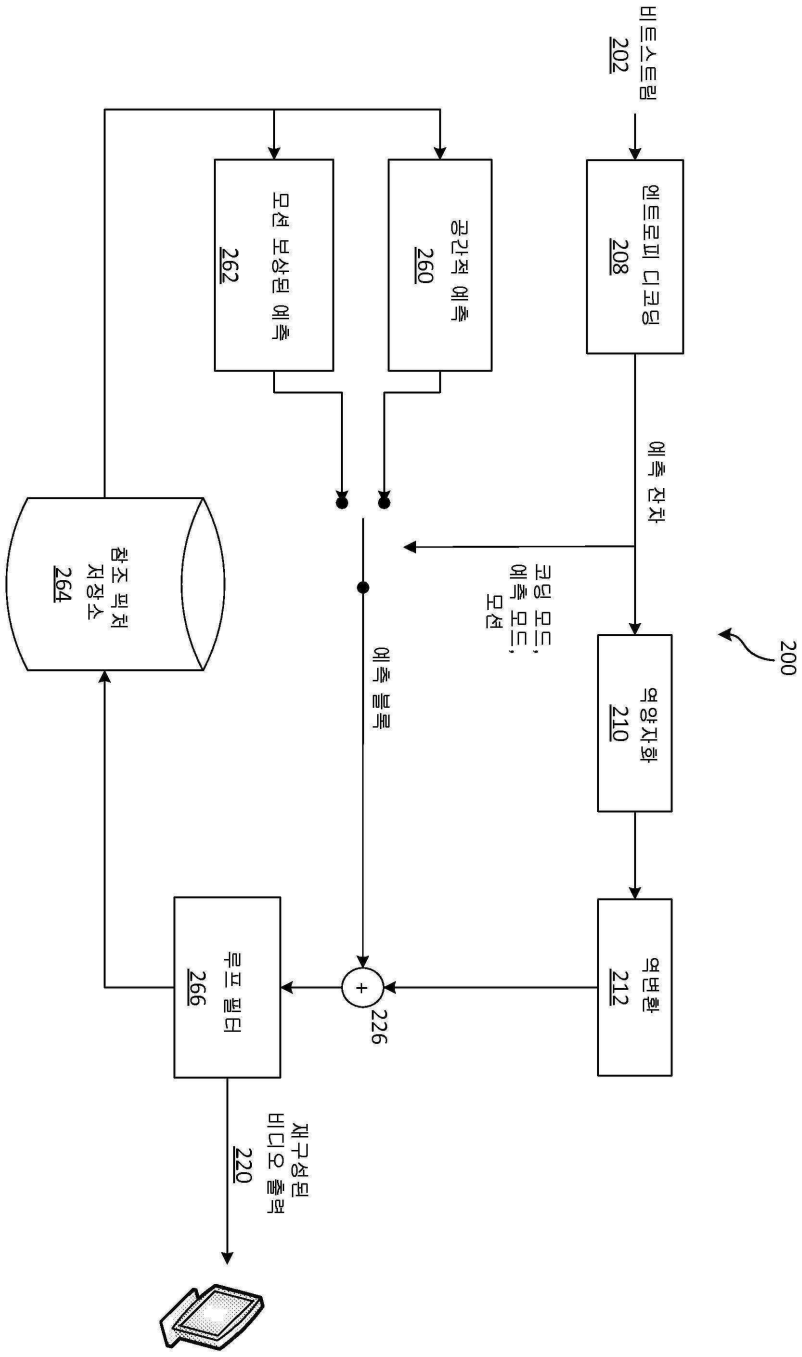
도면1a


입력 비디오
102
100
모드 결정 및 다른 인코더 제어 로직
180
공간적 예측
160
예측 블록
116
예측 잔차
변환
104
양자화
106
모션 예측(추정 및 보상)
162
코딩 모드, 예측 모드, 모션
예측 블록
역양자화
110
역변환
112
126
전자 계수 블록
루프 필터
166
참조 픽처 저장소
164
재구성된 블록
엔트로피 코딩
108
비트스트림
120

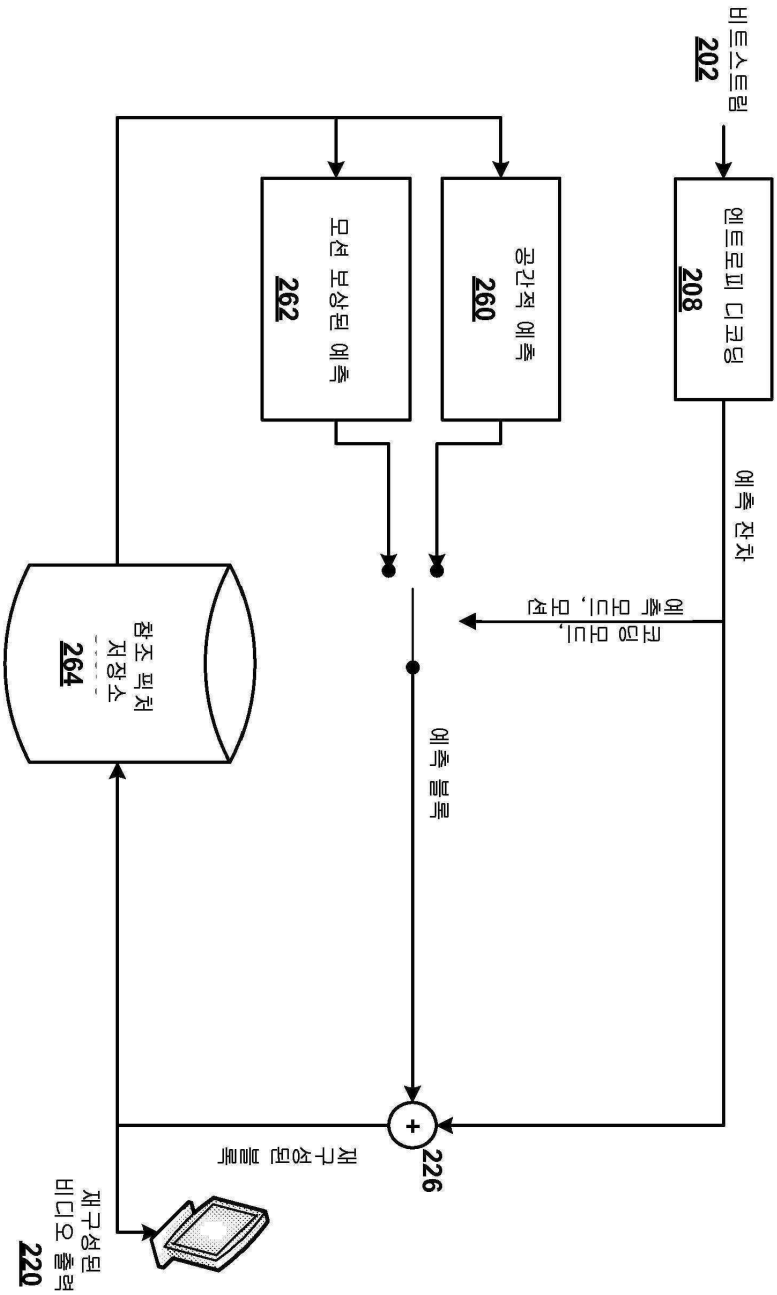
도면1b


입력 비디오
102
모드 결정 및 다른 인코더 제어 로직
180
공간적 예측
160
모션 예측(추정 및 보상)
162
참조 픽처 저장소
164
예측 블록
예측 블록
코딩 모드, 예측 모드, 모션 등
재구성된 블록
재구성된 블록
126
엔트로피 코딩
108
비트스트림
120

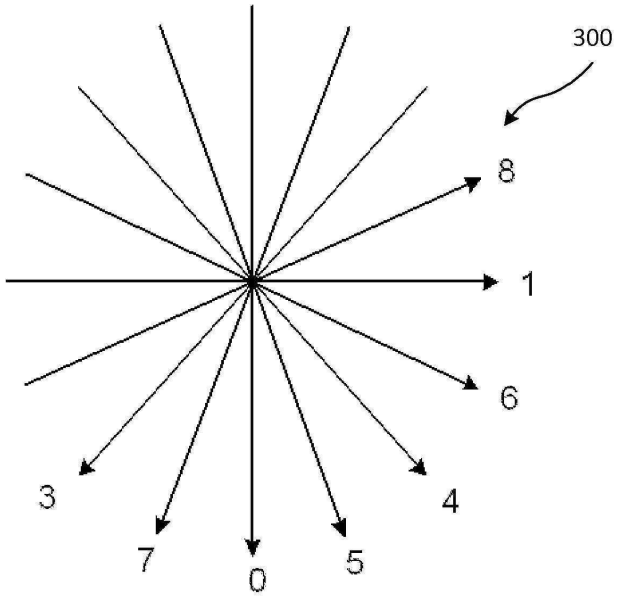
도면2a


비트스트림
202
엔트로피 디코딩
208
공간적 예측
260
모션 보상된 예측
262
예측 잔차
코딩 모드,
예측 모드,
모션
예측 블록
참조 픽처
저장소
264
200
역양자화
210
역변환
212
226
루프 필터
266
재구성된
비디오 출력
220

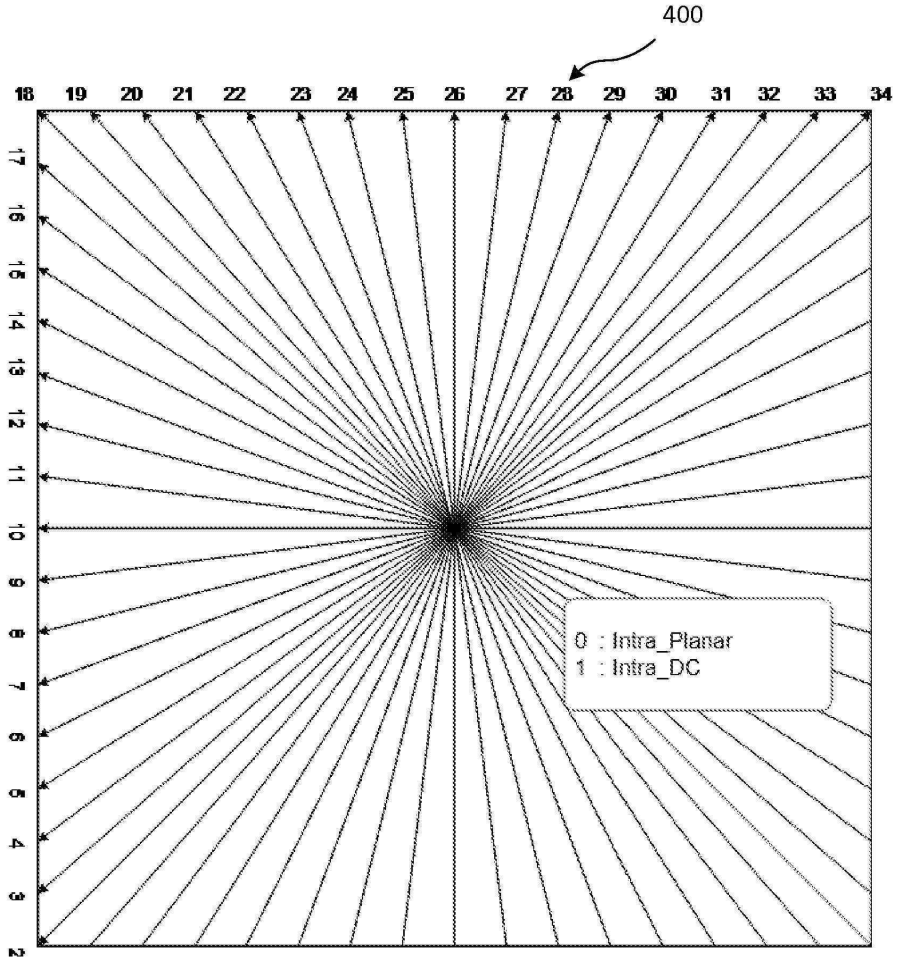
도면2b


비트스트림
202
엔트로피 디코딩
208
공간적 예측
260
모션 보상된 예측
262
예측 잔차
예측 모드, 모션
참조 피처 저장소
264
예측 블록
226
재구성된 블록
재구성된 비디오 출력
220

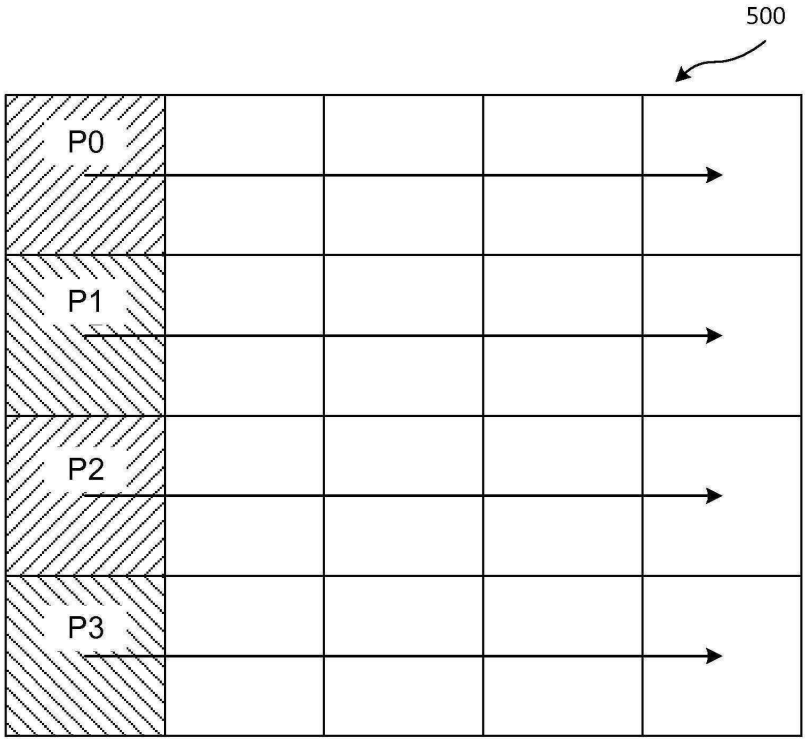
도면3


300
8
1
6
4
5
0
7
3

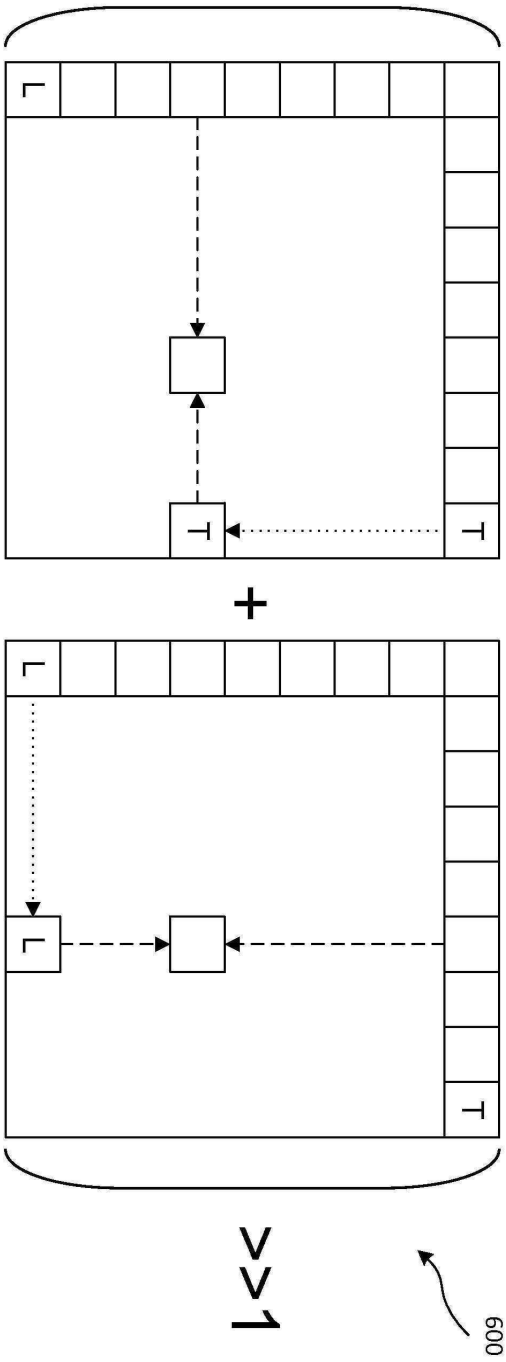
도면4


400
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
0 : Intra_Planar
1 : Intra_DC

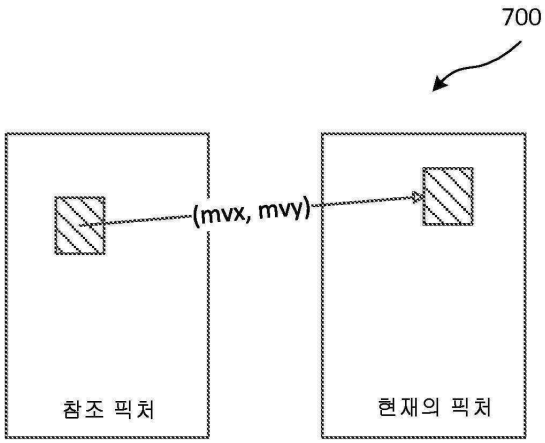
도면5



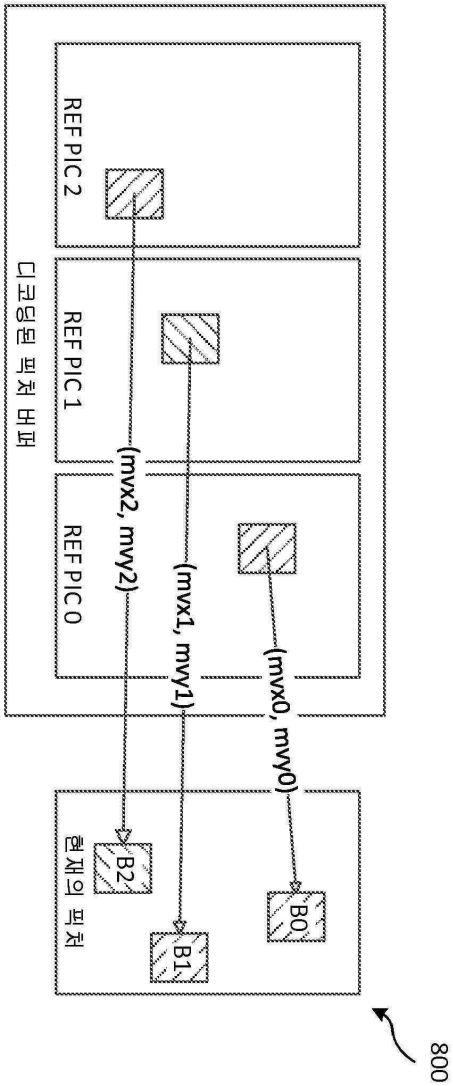
도면6



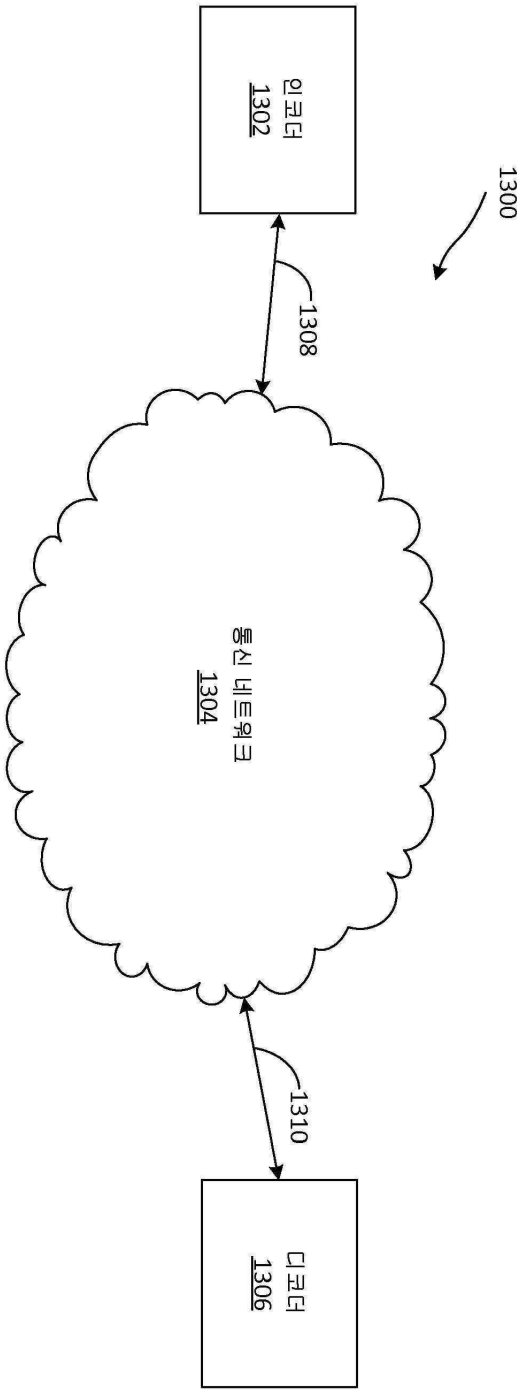
도면7


700
(mvx, mvy)
참조 픽처
현재의 픽처

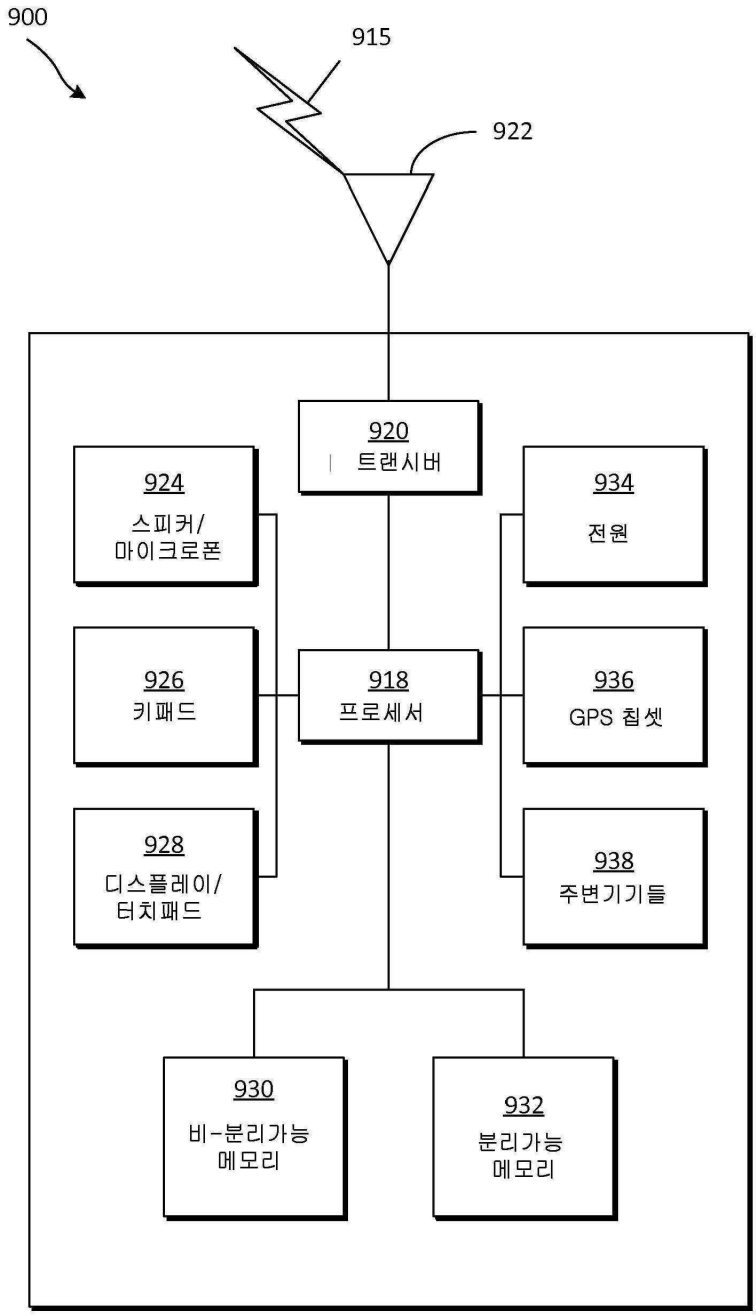
도면8


디코딩된 픽처 버퍼
REF PIC 2
REF PIC 1
REF PIC 0
(mvx2, mvy2)
(mvx1, mvy1)
(mvx0, mvy0)
현재의 픽처
B2
B1
B0
800

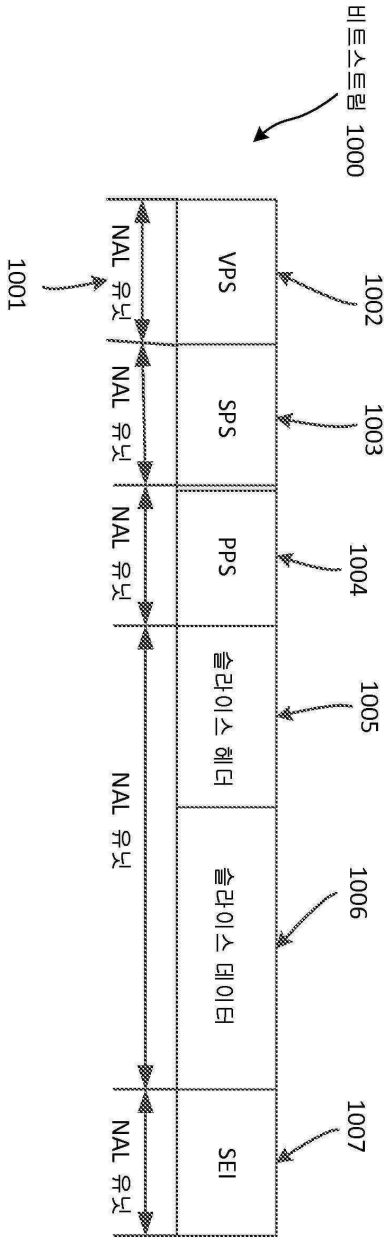
도면9



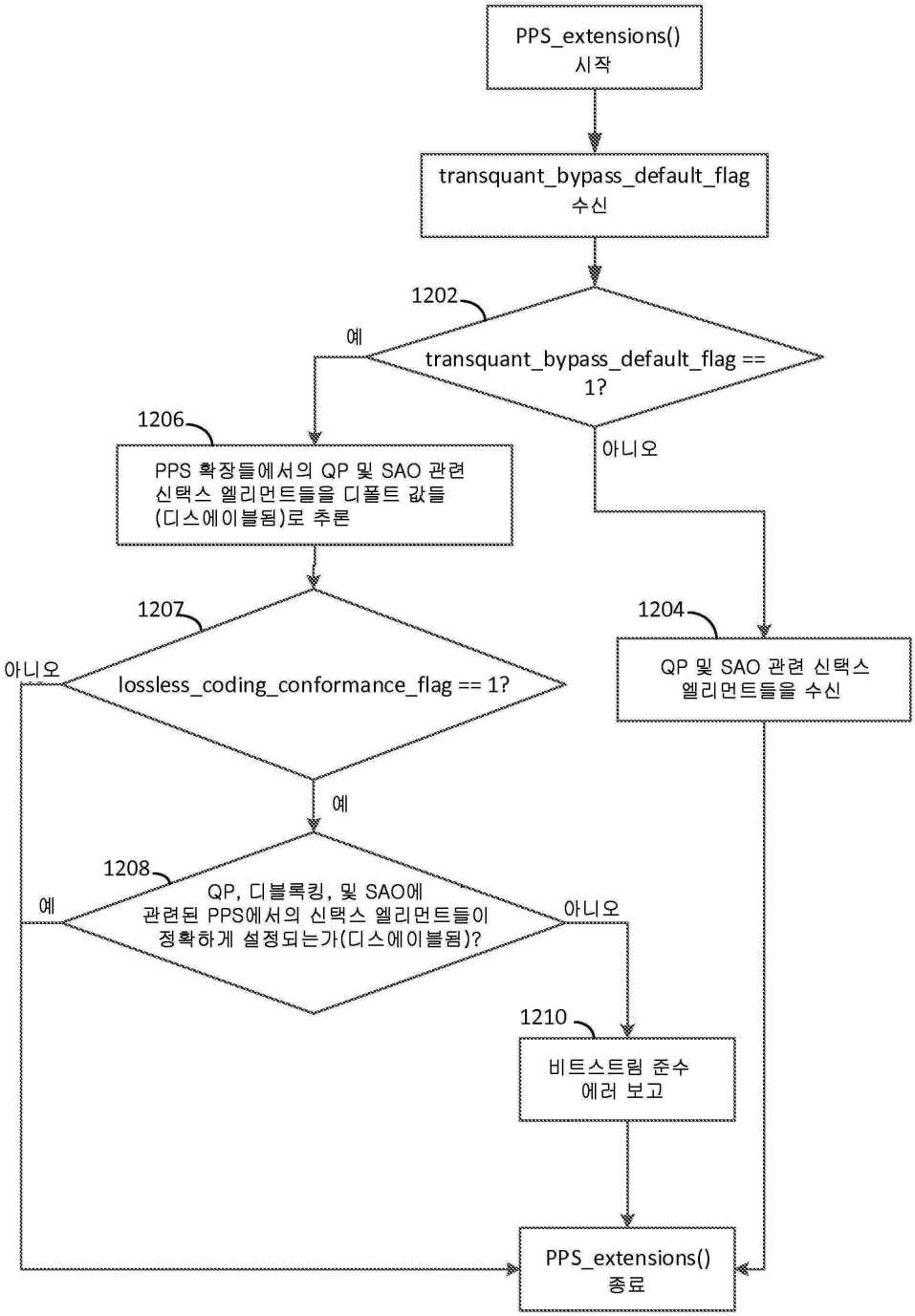
도면10


900
915
922
920
트랜시버
924
스피커/
마이크로폰
926
키패드
928
디스플레이/
터치패드
918
프로세서
934
전원
936
GPS 칩셋
938
주변기기들
930
비-분리가능
메모리
932
분리가능
메모리

도면11


비트스트림 1000
1001
NAL 유닛
VPS
1002
NAL 유닛
SPS
1003
NAL 유닛
PPS
1004
NAL 유닛
슬라이스 헤더
1005
슬라이스 데이터
1006
NAL 유닛
SEI
1007

도면12


PPS_extensions()
시작
transquant_bypass_default_flag
수신
1202
transquant_bypass_default_flag ==
1?
예
아니오
1206
PPS 확장들에서의 QP 및 SAO 관련
신택스 엘리먼트들을 디폴트 값들
(디스에이블됨)로 추론
1207
lossless_coding_conformance_flag == 1?
아니오
예
1204
QP 및 SAO 관련 신택스
엘리먼트들을 수신
1208
QP, 디블록킹, 및 SAO에
관련된 PPS에서의 신택스 엘리먼트들이
정확하게 설정되는가(디스에이블됨)?
예
아니오
1210
비트스트림 준수
에러 보고
PPS_extensions()
종료