(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0024142
(43) 공개일자 2022년03월03일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) *H04N 19/105* (2014.01)
H04N 19/184 (2014.01) *H04N 19/44* (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2021-7041898
(22) 출원일자(국제) 2020년06월17일
심사청구일자 없음
(85) 번역문제출일자 2021년12월21일
(86) 국제출원번호 PCT/US2020/038090
(87) 국제공개번호 WO 2020/263646
국제공개일자 2020년12월30일
(30) 우선권주장
19305818.7 2019년06월24일
유럽특허청(EPO)(EP)

(71) 출원인
인터디지털 브이씨 홀딩스 인코포레이티드
미국 19809 델라웨어주 윌밍턴 스위트 300 벨뷰 파크웨이 200
(72) 발명자
보르드, 필립
프랑스 35576 쎄쏭 쎄비네 아브뉘 데 샹 블랑 975
케르드랑바, 미셸
프랑스 35576 쎄쏭 쎄비네 아브뉘 데 샹 블랑 975
프랑쑤와 에두아르
프랑스 35576 쎄쏭 쎄비네 아브뉘 데 샹 블랑 975
(74) 대리인
양영준, 김연송, 백만기

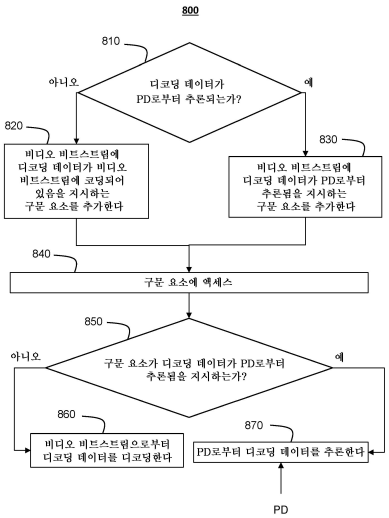
전체 청구항 수 : 총 11 항

(54) 발명의 명칭 **하이 레벨 구문 요소들을 이용하여 디코딩 데이터를 시그널링하기 위한 방법 및 장치**

(57) 요 약

비디오 비트스트림에서 디코딩 데이터를 시그널링하기 위한 방법들(800, 900, 1600, 1700) 및 장치들(2100)로서, 상기 디코딩 데이터가 상기 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 이용한다. 비트스트림, 컴퓨터 판독가능 저장 매체 및 컴퓨터 프로그램 제품도 설명된다.

대 표 도 - 도8



(52) CPC특허분류
H04N 19/184 (2015.01)
H04N 19/44 (2015.01)

명 세 서

청구범위

청구항 1

방법으로서,

픽처 영역에 대응하는 디코딩 데이터가 상기 픽처 영역을 나타내는 비디오 비트스트림의 제1 부분에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 제2 부분의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 이용하는 단계를 포함하고, 상기 이전 데이터는 상기 비트스트림의 제3 부분들에 의해 표현되는 여러 픽처 영역들에 의해 공유되는, 방법.

청구항 2

장치로서,

픽처 영역에 대응하는 디코딩 데이터가 상기 픽처 영역을 나타내는 비디오 비트스트림의 제1 부분에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 제2 부분의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 이용하는 수단을 포함하고, 상기 이전 데이터는 상기 비트스트림의 제3 부분들에 의해 표현되는 여러 픽처 영역들에 의해 공유되는, 장치.

청구항 3

제1항의 방법 또는 제2항의 장치에 있어서, 디코딩 데이터의 세트가 디코딩 데이터의 서브세트들로 분할되고 서브세트마다의 구문 요소가 서브세트의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시하는, 방법 또는 장치.

청구항 4

제1항 또는 제3항의 방법 또는 제2항 또는 제3항의 장치에 있어서, 상기 디코딩 데이터가 코딩 툴들의 활성화를 제어하는 제약-플래그(constraint-flag)들이고, 상기 구문 요소는 상기 제약-플래그들이 상기 비디오 비트스트림의 제1 부분에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 제2 부분의 이전 데이터로부터 추론되는지를 지시하는, 방법 또는 장치.

청구항 5

제1항 또는 제3항의 방법 또는 제2항 또는 제3항의 장치에 있어서, 상기 디코딩 데이터는 참조 픽처들의 목록이고 상기 이전 데이터는 GOP 구조 및 참조 프레임들의 수인, 방법 또는 장치.

청구항 6

제1항 또는 제3항의 방법 또는 제2항 또는 제3항의 장치에 있어서, 상기 디코딩 데이터는 슬라이스 파라미터들이고 상기 이전 데이터는 GOP 구조 및 디코딩된 픽처 순서인, 방법 또는 장치.

청구항 7

제1항, 제3항 내지 제6항 중 어느 한 항에 따른 방법을 포함하는, 비디오를 인코딩 또는 디코딩하기 위한 방법.

청구항 8

제2항 내지 제6항 중 어느 한 항에 따른 장치를 포함하는, 비디오를 인코딩 또는 디코딩하기 위한 장비.

청구항 9

제1항, 제3항 내지 제7항 중 어느 한 항의 방법을 구현하기 위한 명령어들이 저장되어 있는 비일시적 컴퓨터 판독가능 저장 매체.

청구항 10

제1항, 제3항 내지 제7항 중 어느 한 항의 방법을 구현하기 위한 명령어들을 포함하는 컴퓨터 프로그램.

청구항 11

비트스트림으로서,

비디오 데이터를 디코딩하기 위해 사용되는 디코딩 데이터; 및

픽처 영역에 대응하는 상기 디코딩 데이터가 상기 픽처 영역을 나타내는 비트스트림의 제1 부분에 명시적으로 코딩되어 있는지 또는 상기 비트스트림의 제2 부분의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 포함하도록 포맷팅되고, 상기 이전 데이터는 상기 비트스트림의 제3 부분들에 의해 표현되는 여러 픽처 영역들에 의해 공유되는, 비트스트림.

발명의 설명

기 술 분 야

[0001] 본 실시예들은 일반적으로 하이 레벨 구문 요소들을 이용하여 디코딩 데이터를 시그널링하는 것에 관한 것이다.

배 경 기 술

[0002] 본 섹션은 아래에 설명 및/또는 청구되는 본 실시예들 중 적어도 하나의 실시예의 다양한 기술 양태와 관련될 수 있는 본 기술분야의 다양한 양태들을 독자에게 소개하려고 의도되어 있다. 이 논의는 적어도 하나의 실시예의 다양한 양태들의 더 나은 이해를 가능하게 하기 위해 배경 정보를 독자에게 제공하는 데 도움이 될 것으로 여겨진다. 따라서, 이들 진술은 이러한 관점에서 읽혀져야 한다는 점을 이해해야 한다.

[0003] 높은 압축 효율을 달성하기 위해, 비디오 코딩 방식들은 보통 예측 및 변환을 이용하여 비디오 콘텐츠에서의 공간적 및 시간적 중복성(spatial and temporal redundancy)을 활용(leverage)한다. 일반적으로, 프레임 내(intra frame) 또는 프레임 간(inter frame) 상관관계를 활용하기 위해 인트라 또는 인터 예측이 이용된다. 그 후 비디오의 원본 픽처와 예측된 픽처 간의 차이들 - 종종 예측 오류들 또는 예측 잔차들로서 나타내어짐 - 이 변환되고, 양자화되고 엔트로피 코딩된다. 픽처를 재구성하기 위해, 압축된 데이터는 예측, 변환, 양자화 및 엔트로피 코딩에 대응하는 역 프로세스들에 의해 디코딩된다.

발명의 내용

[0004] 본 섹션은 본 개시내용의 일부 양태들의 기본적인 이해를 제공하기 위해 본 실시예들 중 적어도 하나의 실시예의 간략화된 요약을 제공한다. 이 요약은 실시예의 광범위한 개요는 아니다. 그것은 실시예의 핵심적인 또는 중요한 요소들을 식별하려고 의도된 것은 아니다. 다음의 요약은 본 문서의 다른 곳에서 제공되는 더 상세한 설명에 대한 서문으로서 간략화된 형식으로 본 실시예들 중 적어도 하나의 실시예의 일부 양태들을 제시하는 것에 불과하다.

[0005] 본 실시예들 중 적어도 하나의 실시예의 일반적인 양태에 따르면, 비디오 비트스트림에서 디코딩 데이터를 시그널링하기 위한 방법이 제공되는데, 상기 방법은 상기 디코딩 데이터가 상기 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 이용하는 단계를 포함한다.

[0006] 본 실시예들 중 적어도 하나의 실시예의 다른 일반적인 양태에 따르면, 비디오 비트스트림에서 디코딩 데이터를 시그널링하기 위한 장치가 제공되는데, 상기 장치는 상기 디코딩 데이터가 상기 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 이용하는 수단을 포함한다.

[0007] 일 실시예에서는, 디코딩 데이터의 세트가 디코딩 데이터의 서브세트들로 분할되고 상기 단계 또는 상기 수단은 서브세트의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시하는 서브세트마다의 구문 요소를 이용한다.

[0008] 일 실시예에서는, 상기 디코딩 데이터가 코딩 툴들의 활성화를 제어하는 제약-플래그(constraint-flag)들이고

상기 구문 요소는 상기 제약-플래그들이 상기 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시한다.

[0009] 일 실시예에서는, 상기 디코딩 데이터가 픽처 부분들의 그룹의 헤더의 디코딩 데이터이고 상기 구문 요소는 상기 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 상기 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 상기 비디오 비트스트림의 픽처 부분들의 그룹의 다른 헤더의 디코딩 데이터로부터 추론되는지를 지시한다.

[0010] 일 실시예에서는, 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 참조 픽처들의 목록이고 상기 이전 데이터는 GOP 구조 및 참조 프레임들의 수이다.

[0011] 일 실시예에서는, 상기 디코딩 데이터는 슬라이스 파라미터들이고 상기 이전 데이터는 GOP 구조 및 디코딩된 픽처 순서이다.

[0012] 일 실시예에서는, 상기 디코딩 데이터가 타일 그룹의 디코딩 데이터이고 상기 이전 데이터는 다른 타일 그룹의 디코딩 데이터이다.

[0013] 일 실시예에서는, 상기 디코딩 데이터가 GOP 구조이고 상기 이전 데이터는 상기 GOP 및 미리 정의된 GOP 구조들의 세트 내의 일부 먼저 디코딩된 픽처들이다.

[0014] 적어도 하나의 실시예의 다른 일반적인 양태에 따르면, 상기 방법들 중 하나에 따라 비디오 비트스트림에서 디코딩 데이터를 시그널링하는 것을 포함하는 비디오를 인코딩 또는 디코딩하기 위한 방법 및 디바이스가 제공된다.

[0015] 적어도 하나의 실시예의 다른 일반적인 양태들에 따르면, 비일시적 컴퓨터 판독가능 저장 매체, 컴퓨터 프로그램 제품 및 비트스트림이 제공된다.

[0016] 본 실시예들 중 적어도 하나의 실시예의 구체적인 본질 뿐만 아니라 본 실시예들 중 상기 적어도 하나의 실시예의 다른 목적들, 이점들, 특징들 및 용도들은 첨부 도면들과 관련하여 취해지는 예들에 대한 다음의 설명으로부터 명백해질 것이다.

도면의 간단한 설명

[0017] 도면들에는, 여러 실시예들의 예들이 예시된다. 도면들은 다음의 것들을 보여준다:

도 1은 실시예에 따른 예시적인 인코더의 간략화된 블록도를 예시한다.

도 2는 적어도 하나의 실시예에 따른 예시적인 디코더의 간략화된 블록도를 예시한다.

도 3은 종래 기술에 따른 HLS 요소 시그널링 제약 플래그들의 예를 예시한다.

도 4는 종래 기술에 따른 타일 슬라이스들 슬라이스 그룹들, 타일들 및 브릭들의 예들을 예시한다.

도 5는 종래 기술에 따른 타일들에 대한 구문 요소들의 예를 예시한다.

도 6 및 도 7은 종래 기술에 따른 타일-그룹들의 예들을 예시한다.

도 7a는 적어도 하나의 실시예에 따른 타일들, 슬라이스들 및 브릭들의 예를 예시한다.

도 8은 적어도 하나의 실시예에 따른 디코딩 데이터의 세트를 시그널링하기 위한 방법(800)의 플로차트를 예시한다.

도 9는 적어도 하나의 실시예에 따른 도 8 또는 도 9의 방법의 변형의 플로차트를 예시한다.

도 10은 적어도 하나의 실시예에 따른 세트를 서브세트들로 분할하는 예를 예시한다.

도 11은 16개의 디코딩 데이터의 세트의 2개의 계층 레벨 분할의 예를 예시한다.

도 12는 적어도 하나의 실시예에 따른 HLS 요소 시그널링 제약 플래그들의 예를 예시한다.

도 13은 적어도 하나의 실시예에 따른 HLS 요소의 예를 예시한다.

도 14 및 도 15는 종래 기술에 따른 슬라이스 세그먼트들의 예를 예시한다.

도 16은 디코딩 데이터의 세트가 픽처 부분들의 그룹의 헤더의 디코딩 데이터를 포함할 때의 방법(800 또는 900)의 실시예의 플로차트(1600)를 예시한다.

도 17은 적어도 하나의 실시예에 따른 GOP와 연관된 구문 요소들의 시간적 구성(temporal organization)의 예를 예시한다.

도 18은 적어도 하나의 실시예에 따른 슬라이스의 구문의 예를 예시한다.

도 19는 디코딩 데이터의 세트가 픽처 그룹(Group of Pictures)의 구조를 포함할 때의 방법(800 또는 900)의 실시예의 플로차트(1900)를 예시한다.

도 20은 도 19의 방법의 변형을 예시한다.

도 21은 본 개시내용의 양태들이 구현되고 실행될 수 있는 컴퓨팅 환경의 블록도를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 상세한 설명은 본 실시예들의 원리들을 예시한다. 따라서 본 기술분야의 숙련자들은, 비록 본 명세서에 명시적으로 설명되거나 도시되지 않더라도, 본 실시예들의 원리를 구현하고 그 범위 내에 포함되는 다양한 배열을 고안할 수 있을 것이라는 점을 알 것이다.

[0019] 본 명세서에 언급된 모든 예들 및 조건부 언어는 독자에게 본 실시예들의 원리들 및 발명자가 기술을 발전시키는 데 기여한 개념들을 이해하는 데 도움을 줄 교육적 목적으로 의도되어 있고 그러한 구체적으로 언급된 예들 및 조건들에 제한되지 않는 것으로 해석되어야 한다.

[0020] 또한, 본 개시내용의 원리들, 양태들, 및 실시예들, 뿐만 아니라 그의 구체적인 예들을 언급하는 본 명세서에서의 모든 진술들은 그의 구조적 및 기능적 양쪽 모두의 동등물들을 포함하려고 의도되어 있다. 추가적으로, 그러한 동등물들은 현재 알려진 동등물들 뿐만 아니라 미래에 개발되는 동등물들, 즉, 구조에 관계없이, 동일한 기능을 수행하는 개발된 임의의 요소들 양쪽 모두를 포함하는 것이 의도된다.

[0021] 따라서, 예를 들어, 본 기술분야의 숙련자들은, 본 명세서에 제시된 블록도들이 본 실시예들의 원리들을 구현하는 예시적인 회로의 개념도들을 나타낸다는 점을 알 것이다. 유사하게, 임의의 플로 차트들, 플로 다이어그램들, 상태 천이 다이어그램들, 의사코드 등은 컴퓨터 판독가능 매체에 실질적으로 표현될 수 있고 따라서 컴퓨터 또는 프로세서 - 그러한 컴퓨터 또는 프로세서가 명시적으로 도시되어 있는지 여부에 관계없이 - 에 의해 실행될 수 있는 다양한 프로세스들을 나타낸다는 점을 알 것이다.

[0022] 본 실시예들은, 상기 본 실시예들의 예들이 도시되는, 첨부 도면들을 참조하여 이하에서 더 상세히 설명될 것이다. 그러나, 실시예는 많은 대안적인 형식으로 구현될 수 있고 본 명세서에 제시된 예들로 제한되는 것으로 해석되어서는 안 된다. 따라서, 실시예들을 개시된 특정 형태들로 제한하려는 의도는 없다는 점을 이해해야 한다. 오히려, 본 실시예들은 이 출원의 사상 및 범위 내에 포함되는 모든 수정들, 동등물들, 및 대안들을 커버하려고 의도되어 있다.

[0023] 도면이 플로 다이어그램으로서 제시될 때, 그것은 또한 대응하는 장치의 블록도를 제공한다는 점을 이해해야 한다. 유사하게, 도면이 블록도로서 제시될 때, 그것은 또한 대응하는 방법/프로세스의 플로 다이어그램을 제공한다는 점을 이해해야 한다.

[0024] 도면들에 도시된 다양한 요소들의 기능들은 전용 하드웨어 뿐만 아니라 적절한 소프트웨어와 관련하여 소프트웨어를 실행할 수 있는 하드웨어의 사용을 통해 제공될 수 있다. 프로세서에 의해 제공될 때, 기능들은 단일 전용 프로세서에 의해, 단일 공유 프로세서에 의해, 또는 일부가 공유될 수 있는 복수의 개별 프로세서에 의해 제공될 수 있다. 또한, 용어 "프로세서" 또는 "컨트롤러"의 명시적 사용은 소프트웨어를 실행할 수 있는 하드웨어를 오로지 언급하는 것으로 해석되어서는 안 되고, 제한 없이, DSP(digital signal processor) 하드웨어, 소프트웨어를 저장하기 위한 ROM(read only memory), RAM(random access memory), 및 비휘발성 저장소를 암시적으로 포함할 수 있다.

[0025] 종래의 및/또는 맞춤형의 다른 하드웨어가 또한 포함될 수 있다. 유사하게, 도면들에 도시된 임의의 스위치들은 단지 개념적이다. 그것들의 기능은 프로그램 로직의 동작을 통해, 전용 로직을 통해, 프로그램 제어 및 전용 로직의 상호작용을 통해, 또는 심지어 수동으로 수행될 수 있고, 특정 기법은 컨텍스트로부터 더 구체적으로 이해되는 바와 같이 구현자에 의해 선택될 수 있다.

[0026] 도면들의 유사하거나 동일한 요소들은 동일한 참조 번호들로 참조된다.

[0027] 일부 도면들은 비디오 압축 표준들을 따르는 비트스트림의 구조를 정의하기 위해 상기 비디오 압축 표준들의 사

양에서 널리 사용되는 구문 표들을 나타낸다. 해당 구문 표들에서, 용어 '...'는 비디오 압축 표준의 사양에서 주어지고 판독을 용이하게 하기 위해 도면들에서 제거된 널리 알려진 정의에 대한 구문의 변경되지 않은 부분들을 나타낸다. 구문 표들에서의 굵은 용어들은 비트스트림을 파싱함으로써 이 용어에 대한 값이 획득된다는 것을 지시한다. 구문 표들의 우측 열은 구문 요소의 데이터를 인코딩하기 위한 비트들의 수를 지시한다. 예를 들어, u(4)는 데이터를 인코딩하기 위해 4개의 비트가 사용된다는 것을 지시하고, u(8)는 8개의 비트를 지시하고, ae(v)는 컨텍스트 적응 산술 엔트로피 코딩된 구문 요소를 지시한다.

[0028] 본 명세서의 청구항들에서, 특정된 기능을 수행하기 위한 수단으로서 표현된 임의의 요소는, 예를 들어, a) 해당 기능을 수행하는 회로 요소들의 조합, 또는 b) 그 기능을 수행하기 위해 해당 소프트웨어를 실행하기 위한 적절한 회로와 조합된 임의의 형식의 소프트웨어(그러므로, 펌웨어, 마이크로코드 등을 포함함)를 포함하여, 해당 기능을 수행하는 임의의 방식을 포함하려고 의도되어 있다. 그러한 청구항들에 의해 정의된 바와 같은 본 실시예들은 다양한 언급된 수단들에 의해 제공되는 기능성들이 청구항들이 요구하는 방식으로 조합되고 합쳐진다는 사실에 있다. 따라서, 그러한 기능들을 제공할 수 있는 임의의 수단은 본 명세서에 도시된 것들과 동등한 것으로 간주된다.

[0029] 도면들 및 설명들은, 명료성의 목적으로, 전형적인 인코딩 및/또는 디코딩 디바이스들에서 발견되는 많은 다른 요소들을 제거하면서, 본 실시예들의 분명한 이해와 관련되는 요소들을 예시하도록 간략화되었다는 점을 이해해야 한다.

[0030] 비록 제1 및 제2라는 용어들이 다양한 요소들을 설명하기 위해 본 명세서에서 사용될 수 있지만, 이들 요소는 이들 용어들에 의해 제한되어서는 안 된다는 점을 이해할 것이다. 이들 용어는 하나의 요소를 다른 요소와 구별하기 위해서만 사용된다. 다양한 방법들이 위에서 설명되었고, 방법들 각각은 설명된 방법을 달성하기 위한 하나 이상의 단계 또는 액션을 포함한다. 방법의 적절한 동작을 위해 단계들 또는 액션들의 특정 순서가 요구되지 않는 한, 특정 단계들 및/또는 액션들의 순서 및/또는 사용은 수정되거나 조합될 수 있다.

[0031] 다음의 섹션들에서는, "재구성된" 및 "디코딩된"이라는 단어가 호환하여 사용될 수 있다. 반드시 그런 것은 아니지만 보통 "재구성된"은 인코더 측에서 사용되는 반면 "디코딩된"은 디코더 측에서 사용된다. 또한, "코딩된" 및 "인코딩된"이라는 단어들은 호환하여 사용될 수 있다. 또한, "이미지", "픽처" 및 "프레임"이라는 단어들은 호환하여 사용될 수 있다. 더욱이, "코딩", "소스 코딩" 및 "압축"이라는 단어들은 호환하여 사용될 수 있다.

[0032] 본 개시내용의 "일 실시예" 또는 "실시예" 또는 "일 구현" 또는 "구현", 뿐만 아니라 그의 다른 변형들에 대한 언급은 그 실시예와 관련하여 설명된 특정 특징, 구조, 특성 등이 본 개시내용의 적어도 하나의 실시예에 포함된다는 것을 의미한다는 점을 이해해야 한다. 따라서, 본 명세서에 걸쳐 다양한 곳들에서 나타나는 문구 "일 실시예에서" 또는 "실시예에서" 또는 "일 구현에서" 또는 "구현에서", 뿐만 아니라 임의의 다른 변형들의 출현들은 반드시 모두가 동일한 실시예를 언급하는 것은 아니다.

[0033] 추가적으로, 본 실시예들 또는 그의 청구항들은 다양한 정보들을 "결정하는 것"을 언급할 수 있다. 정보를 결정, 도출하는 것은, 예를 들어, 정보를 추정하는 것, 정보를 계산하는 것, 정보를 예측하는 것, 또는 메모리로부터 정보를 검색하는 것 중 하나 이상을 포함할 수 있다.

[0034] 또한, 본 출원 또는 그의 청구항들은 다양한 정보들을 "제공하는 것"을 언급할 수 있다. 정보를 제공하는 것은, 예를 들어, 정보를 출력하는 것, 정보를 저장하는 것, 정보를 송신하는 것, 정보를 전송하는 것, 정보를 디스플레이하는 것, 정보를 보여주는 것, 또는 정보를 이동시키는 것 중 하나 이상을 포함할 수 있다.

[0035] 또한, 본 출원 또는 그의 청구항들 또는 그의 청구항들은 다양한 정보들에 "액세스하는 것"을 언급할 수 있다. 정보에 액세스하는 것은, 예를 들어, 정보를 수신하는 것, 정보를 검색하는 것(예를 들어, 메모리로부터), 정보를 저장하는 것, 정보를 처리하는 것, 정보를 이동시키는 것, 정보를 복사하는 것, 정보를 소거하는 것, 정보를 계산하는 것, 정보를 결정하는 것, 정보를 예측하는 것, 또는 정보를 추정하는 것 중 하나 이상을 포함할 수 있다.

[0036] 또한, 본 출원 또는 그의 청구항들은 다양한 정보들을 "수신하는 것"을 언급할 수 있다. 수신하는 것은, "액세스하는 것"에서와 같이, 광의의 용어인 것으로 의도되어 있다. 정보를 수신하는 것은, 예를 들어, 정보에 액세스하는 것, 또는 정보를 검색하는 것(예를 들어, 메모리로부터) 중 하나 이상을 포함할 수 있다. 또한, "수신하는 것"은 전형적으로, 예를 들어, 정보를 저장하는 것, 정보를 처리하는 것, 정보를 송신하는 것, 정보를 이동시키는 것, 정보를 복사하는 것, 정보를 소거하는 것, 정보를 계산하는 것, 정보를 결정하는 것, 정보를 예측

하는 것, 또는 정보를 추정하는 것과 같은 동작들 동안 하나의 방식 또는 다른 방식으로 수반된다.

[0037] 또한, 본 출원 또는 그의 청구항들은 비디오 비트스트림의 이전 데이터로부터 다양한 데이터를 "추론하는 것"을 언급할 수 있다. "이전 데이터"라는 용어는 추론될 현재 데이터 전에 구축, 재구성, 파싱된 데이터로서 이해되어야 한다. 전에 추론된 데이터로부터 데이터를 추론할 수 있다는 점에 유의한다(재귀적 추론). 추론의 한 가지 특정한 경우는 디폴트 데이터를 추론된 데이터로서 사용하는 디폴트 추론인데, 즉 "이전" 데이터가 디폴트 데이터이다. 비디오 비트스트림의 이전 데이터로부터 추론하는 것, 도출하는 것은, 예를 들어, 이전 데이터를 복사하는 것 또는 다수의 이전 데이터를 조합하여 추론된 데이터를 얻는 것, 또는 추론된 데이터를 도출하기 위한 수단의 지시를 제공하는 이전 데이터로부터의 정보에 액세스하는 것 중 하나 이상을 포함할 수 있다.

[0038] 또한, 본 출원 또는 그의 청구항들은 "비디오 비트스트림에의 디코딩 데이터의 명시적 코딩" 및 "비디오 비트스트림으로부터의 디코딩 데이터의 명시적 디코딩"을 언급할 수 있다. 디코딩 데이터의 명시적 코딩(디코딩)은 비디오 비트스트림에(으로부터) 전용 구문 요소들을 추가(획득)하는 것을 언급한다. 명시적 인코딩 데이터는 이 데이터를 나타내는 적어도 하나의 빈(bin)/정보를 엔트로피 코딩 엔진(예를 들어, CABAC)에 입력/입력(enter/input)하는 것을 언급할 수 있는 반면, 데이터를 추론하는 것은 이 데이터를 나타내는 어떠한 빈/정보도 엔트로피 코딩 엔진에 입력하지 않는 것을 언급한다. 유사하게, 명시적 디코딩 데이터는 이 데이터를 나타내는 적어도 하나의 빈/정보를 엔트로피 디코딩 엔진(즉, CABAC)으로부터 출력하는 것을 언급하는 반면, 데이터를 추론하는 것은 이 데이터를 나타내는 어떠한 빈/정보도 엔트로피 디코딩 엔진에 출력하지 않는 것을 언급한다. 데이터가 명시적으로 코딩되어 있는지 또는 추론되는지의 정보는 또한 명시적으로 코딩되어 있거나 추론될 수 있는 다른 구문 요소일 수도 있다는 점에 유의한다.

[0039] 예를 들어, 디코딩 데이터는 미리 정의된 GOP 구조들의 표의 인덱스 또는 슬라이스 파라미터일 수 있다. 그 후, 슬라이스 파라미터들을 명시적으로 인코딩하는 것은 이 디코딩 데이터가 디코딩 프로세스 동안 사용될 때마다 비디오 비트스트림 내의 미리 정의된 GOP 구조들의 표의 인덱스 또는 이 슬라이스 파라미터에 대한 구문 요소를 추가하는 것을 언급한다.

[0040] 도시되고 설명된 다양한 특징들은 호환된다는 점을 알아야 한다. 달리 지시되지 않는 한, 일 실시예에 도시된 특징이 다른 실시예에 통합될 수 있다. 또한, 다양한 실시예들에서 설명된 특징들이 분리할 수 없거나 조합할 수 없는 것으로 달리 지시되지 않는 한 조합되거나 분리될 수 있다.

[0041] 전에 언급된 바와 같이, 도면들에 도시된 다양한 요소들의 기능들은 전용 하드웨어 뿐만 아니라 적절한 소프트웨어와 관련하여 소프트웨어를 실행할 수 있는 하드웨어의 사용을 통해 제공될 수 있다. 또한, 프로세서에 의해 제공될 때, 기능들은 단일 전용 프로세서에 의해, 단일 공유 프로세서에 의해, 또는 일부가 공유될 수 있는 복수의 개별 프로세서에 의해 제공될 수 있다.

[0042] 첨부 도면들에 묘사된 구성 시스템 컴포넌트들 및 방법들 중 일부는 바람직하게는 소프트웨어로 구현되기 때문에, 시스템 컴포넌트들 또는 프로세스 기능 블록들 간의 실제 연결들은 본 개시내용의 프로세스들이 프로그래밍되는 방식에 따라 상이할 수 있다는 점을 추가로 이해해야 한다. 본 명세서의 교시내용들이 주어지면, 관련 기술분야의 통상의 기술자는 본 개시내용의 이들 및 유사한 구현들 또는 구성들을 고려할 수 있을 것이다.

[0043] 비록 첨부 도면들을 참조하여 예시적인 실시예들이 본 명세서에 기술되었지만, 본 개시내용은 그 정확한 실시예들로 제한되지 않는다는 점과, 본 개시내용의 범위를 벗어나지 않고 관련 기술분야의 통상의 기술자에 의해 그 안에서 다양한 변경들 및 수정들이 달성될 수 있다는 점을 이해해야 한다. 또한, 본 개시내용의 범위를 벗어나지 않고, 개별 실시예들이 조합될 수 있다. 모든 그러한 변경들 및 수정들은 첨부된 청구항들에 제시된 바와 같은 본 개시내용의 범위 내에 포함되는 것으로 의도되어 있다.

[0044] 예를 들어, "A/B", "A 및/또는 B" 및 "A 및 B 중 적어도 하나"의 경우들에서, 다음의 "/", "및/또는 ", 및 "~ 중 적어도 하나" 중 임의의 것의 사용은 첫 번째로 열거된 옵션(A)만의 선택, 또는 두 번째로 열거된 옵션(B)만의 선택, 또는 양쪽 옵션들(A 및 B)의 선택을 포함하려고 의도되어 있다는 점을 알아야 한다. 추가의 예로서, "A, B, 및/또는 C" 및 "A, B, 및 C 중 적어도 하나"의 경우들에서, 그러한 문구는 첫 번째 열거된 옵션(A)만의 선택, 또는 두 번째 열거된 옵션(B)만의 선택, 또는 세 번째 열거된 옵션(C)만의 선택, 또는 첫 번째 및 두 번째 열거된 옵션들(A 및 B)만의 선택, 또는 첫 번째 및 세 번째 열거된 옵션들(A 및 C)만의 선택, 또는 두 번째 및 세 번째 열거된 옵션들(B 및 C)만의 선택, 또는 3개의 옵션(A 및 B 및 C) 모두의 선택을 포함하려고 의도되어 있다. 이는, 이 기술분야 및 관련 기술분야의 통상의 기술자라면 쉽게 알 수 있는 바와 같이, 열거된 만큼 많은 항목들에 대해 확장될 수 있다.

[0045] 본 기술분야의 통상의 기술자에게 명백하겠지만, 구현들은, 예를 들어, 저장 또는 송신될 수 있는 정보를 운반하도록 포맷팅된 다양한 신호들을 생성할 수 있다. 정보는, 예를 들어, 방법을 수행하기 위한 명령어들, 또는 설명된 구현들 중 하나에 의해 생성된 데이터를 포함할 수 있다. 예를 들어, 신호는 설명된 실시예의 비트스트림을 운반하도록 포맷팅될 수 있다. 그러한 신호는, 예를 들어, 전자기파로서(예를 들어, 스펙트럼의 무선 주파수 부분을 사용하여) 또는 기저대역 신호로서 포맷팅될 수 있다. 포맷팅은, 예를 들어, 데이터 스트림을 인코딩하는 것 및 인코딩된 데이터 스트림으로 캐리어를 변조하는 것을 포함할 수 있다. 신호가 운반하는 정보는, 예를 들어, 아날로그 또는 디지털 정보일 수 있다. 신호는, 알려진 바와 같이, 다양한 상이한 유선 또는 무선 링크들을 통해 송신될 수 있다. 신호는 프로세서 판독가능 매체 상에 저장될 수 있다.

[0046] 픽처(이미지 또는 프레임을 또한 나타냄)는 흑백 포맷의 루마 샘플들의 어레이, 또는 4:2:0, 4:2:2, 또는 4:4:4 컬러 포맷의 루마 샘플들의 어레이 및 크로마 샘플들의 2개의 대응하는 어레이 또는 3개의 컬러 성분(예를 들어, RGB)의 3개의 어레이일 수 있다는 점을 이해해야 한다.

[0047] 비디오 압축 표준에서는, 픽처가 아마도 상이한 크기 및/또는 상이한 형상의 블록들로 파티셔닝된다.

[0048] 블록은 2차원 어레이 또는 행렬이라는 점을 이해해야 한다. 수평 또는 x 방향(또는 축)은 폭을 나타내고, 수직 또는 y 방향(또는 축)은 높이를 나타낸다. 인덱스들은 0에서 시작된다. x 방향은 열들을 나타내고 y 방향은 행들을 나타낸다. 최대 x 인덱스는 폭 - 1이다. 최대 y 인덱스는 높이 - 1이다.

[0049] 인코딩

[0050] 도 1은 적어도 하나의 실시예에 따른 예시적인 인코더(100)의 간략화된 블록도를 예시한다.

[0051] 인코더(100)는 통신 시스템 내의 송신기 또는 헤드엔드에 포함될 수 있다.

[0052] 하나 이상의 픽처를 갖는 비디오 시퀀스를 인코딩하기 위해, 픽처는 아마도 상이한 크기 및/또는 상이한 형상의 블록들로 파티셔닝된다(모듈 110)). 예를 들어, HEVC(*"ITU-T H.265 TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU (10/2014), SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services - Coding of moving video, High efficiency video coding, Recommendation ITU-T H.265"*)에서는, 픽처가 구성가능한 크기를 갖는 정사각형 형상의 CTU(Coding Tree Unit)들로 파티셔닝될 수 있다. CTU들의 연속적인 세트가 슬라이스로 그룹화될 수 있다. CTU는 이하에서 블록들로 나타낸 코딩 단위들(CU들)로의 쿼드-트리 파티셔닝의 루트이다.

[0053] 예시적인 인코더(100)에서는, 픽처가 아래에 설명되는 바와 같이 블록 기반 인코딩 모듈들에 의해 인코딩된다.

[0054] 각각의 블록은 인트라-예측 모드 또는 인터-예측 모드를 이용하여 인코딩된다.

[0055] 블록이 인트라-예측 모드에서 인코딩될 때(모듈 160), 인코더(100)는, 동일한 픽처 내의 하나의 블록의 적어도 하나의 샘플에 기초하여(또는 픽처 또는 슬라이스의 제1 블록에 대한 미리 정의된 값에 기초하여), 인트라-예측(공간 예측으로도 나타냄)을 수행한다. 예로서, 재구성된 이웃 샘플들로부터 블록을 인트라-예측함으로써 예측 블록이 획득된다.

[0056] 블록이 인터-예측 모드에서 인코딩될 때, 인코더(100)는, (참조 픽처 버퍼에 저장된) 적어도 하나의 참조 픽처 또는 슬라이스의 적어도 하나의 참조 블록에 기초하여, 인터-예측(시간 예측으로도 나타냄)을 수행한다.

[0057] 참조 픽처 버퍼(180)에 저장된 참조 블록을 모션-추정(모듈 175) 및 모션-보상(모듈 170)을 수행하는 것에 의해 인터-예측 코딩이 수행된다.

[0058] 유니-인터-예측(uni-inter-prediction)(단방향 예측(uni-directional prediction)이라고도 알려짐) 모드에서는, 예측 블록이 일반적으로(반드시 그런 것은 아니지만) 이전 참조 픽처에 기초할 수 있다.

[0059] 바이-인터-예측(bi-inter-prediction)(양방향 예측(bi-prediction)이라고도 알려짐) 모드에서는, 예측 블록이 일반적으로(반드시 그런 것은 아니지만) 이전 및 이후 참조 픽처에 기초할 수 있다.

[0060] 인코더(100)는 블록을 인코딩하기 위해 인트라-예측 모드 또는 인터-예측 모드 중 어느 것을 사용할지를 결정하고(모듈 105) 예측 모드 구문 요소에 의해 인트라/인터 결정을 지시한다.

[0061] 블록으로부터 예측 블록(예측자(predictor)라고도 알려짐)을 빼는 것(모듈 120)에 의해 예측 잔차 블록이 계산된다.

[0062] 예측 잔차 블록은 변환되고(모듈 125) 양자화된다(모듈 130). 변환 모듈(125)은 예측 잔차 블록을 픽셀(공간) 도메인으로부터 변환(주파수) 도메인으로 변환할 수 있다. 변환은, 예를 들어, 코사인 변환, 사인 변환, 웨이블릿 변환 등일 수 있다. 양자화(모듈 130)는, 예를 들어, 레이트 왜곡 기준(rate distortion criterion)에 따라 수행될 수 있다.

[0063] 양자화된 변환 계수들, 뿐만 아니라 모션 벡터들 및 다른 구문 요소들을 엔트로피 코딩하여(모듈 145) 비트스트림을 출력한다. 엔트로피 코딩은, 예를 들어, CABAC(Context Adaptive Binary Arithmetic Coding), CAVLC(Context Adaptive Variable Length Coding), 허프만, 산술, exp-Golomb 등일 수 있다.

[0064] 인코더는 또한 변환을 스킵하고 변환되지 않은 예측 잔차 블록에 직접 양자화를 적용할 수 있다. 인코더는 또한 변환 및 양자화 둘 다를 바이패스할 수 있는데, 즉 변환 또는 양자화 프로세스의 적용 없이 예측 잔차 블록이 직접 코딩된다.

[0065] 직접 PCM 코딩에서는, 예측이 적용되지 않고 블록 샘플들이 비트스트림으로 직접 코딩된다.

[0066] 인코더(100)는 디코딩 루프를 포함하고, 따라서 추가 예측들을 위한 참조를 제공하기 위해 인코딩된 블록을 디코딩한다. 양자화된 변환 계수들을 탈양자화(de-quantize)하고(역양자화(inverse quantization)라고도 알려짐)(모듈 140) 역변환(inverse transform)하여(모듈 150) 예측 잔차 블록들을 디코딩한다. 그 후 디코딩된 예측 잔차 블록과 예측 블록을 조합하는 것(모듈 155)에 의해 블록이 재구성된다. 예를 들어, 코딩 아티팩트들을 감소시키기 위한 디블로킹/샘플 적응 오프셋(SAO, Sample Adaptive Offset) 필터링을 수행하기 위해 재구성된 픽처에 하나 이상의 인-루프 필터(165)가 적용될 수 있다. 필터링된 픽처는 참조 픽처 버퍼(180)에 저장된다.

[0067] 인코더(100)의 모듈들은 소프트웨어로 구현되고 프로세서에 의해 실행될 수 있거나 또는 압축 기술분야의 숙련자에게 잘 알려진 회로 컴포넌트들을 이용하여 구현될 수 있다. 특히, 비디오 인코더(100)는 집적 회로(IC)로서 구현될 수 있다.

[0068] 디코딩

[0069] 도 2는 적어도 하나의 실시예에 따른 예시적인 디코더(200)의 간략화된 블록도를 예시한다.

[0070] 디코더(200)는 통신 시스템 내의 수신기에 포함될 수 있다.

[0071] 비록 디코더 내의 모든 동작이 인코딩 프로세스의 역 동작들인 것은 아니지만(예를 들어, 인트라- 및 인터-예측), 디코더(200)는 일반적으로 도 1에 기술된 바와 같이 인코더(100)에 의해 수행되는 인코딩 패스(encoding pass)에 상반되는 디코딩 패스(decoding pass)를 수행한다.

[0072] 특히, 디코더(200)의 입력은, 인코더(100)에 의해 생성될 수 있는, 비디오 비트스트림을 포함한다.

[0073] 비디오 비트스트림을 먼저 엔트로피 디코딩하여(모듈 230), 예를 들어, 변환 계수들, 모션 벡터들 MV, 픽처 파티셔닝 정보, 아마도 예측 모드 플래그, 구문 요소들 및 다른 디코딩 데이터를 획득한다.

[0074] 예를 들어, HEVC에서는, 픽처 파티셔닝 정보가 CTU들의 크기, 및 CTU가 CU들로 분할되는 방식을 지시한다. 그러므로 디코더는, 픽처 파티셔닝 정보에 따라, 픽처를 CTU들로, 그리고 각각의 CTU를 CU들로 분할(235)할 수 있다.

[0075] 변환 계수들은 탈양자화하고(모듈 240) 역변환하여(모듈 250) 예측 잔차 블록들을 디코딩한다. 그 후 디코딩된 예측 잔차 블록들을 예측 블록들(예측자라고도 알려짐)과 조합하여(모듈 255) 디코딩된/재구성된 블록들을 획득한다.

[0076] 아마도 예측 모드 플래그에 따라 인트라-예측(모듈 260) 또는 모션-보상된 예측(즉, 인터-예측)(모듈 270)으로부터 예측 블록이 획득될 수 있다(모듈 205). 재구성된 픽처에 인-루프 필터(모듈 265)가 적용될 수 있다. 인-루프 필터는 디블로킹 필터 및/또는 SAO 필터를 포함할 수 있다. 필터링된 픽처는 참조 픽처 버퍼(280)에 저장된다.

[0077] 디코더(200)의 모듈들은 소프트웨어로 구현되고 프로세서에 의해 실행될 수 있거나 또는 압축 기술분야의 숙련자에게 잘 알려진 회로 컴포넌트들을 이용하여 구현될 수 있다. 특히, 디코더(200)는 집적 회로(IC)로서, 단독으로 또는 코덱으로서 인코더(100)와 조합되어 구현될 수 있다.

[0078] 본 실시예들은 레거시 블록 기반 비디오 압축 표준들에서 사용되는 현재 구문보다 적은 수의 비트들을 사용함으로써 블록 기반 비디오 코딩을 위한 하이 레벨 구문(High Level Syntax, HLS)을 개선하기 위한 메커니즘을 제공

하는 것에 관한 것이다.

[0079] HLS는, 예를 들어 DVB, ATSC, 3GPP와 같은, MPEG 또는 ITU와는 다른 파티들에 의해 정의된 비트스트림 상호운용성 포인트들을 가능하게 하는 시그널링 메커니즘이다. 예를 들어, VVC 드래프트 표준(B.Bross, J. Chen, S.Liu, "Versatile Video Coding (Draft 4)," in JVET document JVET-N1001, 14th Meeting: Geneva, CH, 19-27 March 2019)에서는, HLS 제약 플래그들이 VCC 디코더에서 코딩 툴들의 활성화를 제어하는 것을 허용한다. HLS 제약 플래그들은 도 3에 묘사된 바와 같이 SPS(시퀀스 파라미터 세트) 및/또는 프로파일 및 레벨, VPS(비디오 파라미터 세트) 및 (디코딩 파라미터 세트) DPS 비트스트림 부분에 포함된 구문 요소 "*general_constraint_info()*"로 그룹화된다. 제약 플래그들은 비디오 비트스트림 전체에서 위반될 수 없는 속성들을 지시한다. SPS(또는 유사한) 구문 구조에 제약 플래그가 설정되어 있을 때, 디코더는 툴이 비디오 비트스트림에서 사용되지 않을 것이라고 안전하게 가정할 수 있다.

[0080] 코딩 제약 플래그들에 대한 현재의 구문은 비디오 비트스트림 코딩 효율을 위태롭게 할 수 있는 상당히 많은 수의 비트들을 사용한다. 사실, 이 정보는, 전형적으로 각각의 랜덤 액세스 포인트(RAP)에서 전송되는 SPS에 존재하기 때문에 비디오 비트스트림 내의 여러 위치에서 반복될 수 있고, 그것은 디코더 파라미터 세트(DPS)와 같은 다른 곳들에서도 반복될 수 있다.

[0081] HLS의 다른 예는 시퀀스, 픽처, 슬라이스 또는 타일 헤더들, SEI(Supplemental Enhancement Information), VUI(Video Usability Information)와 같은 비-코딩된 픽처 부분들을 그룹화하고 코딩된 매크로-블록들, CTU들 또는 CU들을 배제하는 하이-레벨 구문이다. 이하에서는, 매크로-블록들, CTU들, 또는 CU들의 서브세트들에 관련된 정보를 예를 들어 픽처들, 슬라이스들, 타일들, 슬라이스들의 그룹 또는 브릭들 헤더들로 그룹화하는 것을 허용하는 HLS를 "HLS 파티셔닝"이라고 할 것이다.

[0082] AVC 및 HEVC 비디오 코딩 표준들에서는, 하이-레벨 파티셔닝이 슬라이스들(래스터 스캔에서 연속적인 CTU들) 또는 타일들(균일한 또는 불균일한 타일 그리드)을 이용하여 수행된다. VVC 드래프트 표준에서는, 하이-레벨 파티셔닝이 타일들 및 아마도 타일들을 수직으로 직사각형 서브-영역들로 파티셔닝하는 브릭들을 이용하여 수행된다(도 4). 불균일한 타일 간격의 경우에는, 타일들이 불균일한 그리드(상부-우측)에 따라 분포된다. 이 경우, HLS에서는, "타일-열-폭(tile-column-width)" 및 "타일-열-높이(tile-column-height)" 값들을 시그널링할 수 있다(도 5).

[0083] 여러 타일들이 하나의 "타일-그룹"으로 그룹화될 수 있다. 타일-그룹들을 정의하기 위해, 타일들은 도 6에 묘사된 바와 같이 래스터 스캔 순서로 암시적으로 라벨링되고, 각각의 타일 그룹 "i"에 대해 HLS에서 상부-좌측 타일 인덱스(*top_left_tile_idx[i]*) 및 하부-우측 타일 인덱스(*bottom_right_tile_idx[i]*)가 시그널링된다. 예를 들어, 8개의 타일(도 4의 상부-우측)이 4개의 "타일-그룹"(도 6)으로 그룹화되었다. 예를 들어, 제1 타일-그룹 0이 타일 0, 1 및 2로 구성되고 top_left_tile_idx[0]=0, bottom_right_tile_idx[0]=2 및 bottom_right_tile_idx_delta[0]=2을 갖는다(*bottom_right_tile_idx_delta[i] = bottom_right_tile_idx[i] - top_left_tile_idx[i]*)(도 7).

[0084] 변형에서는, 타일이 하나 이상의 브릭으로 분할될 수 있다. 그 각각은 타일 내의 다수의 CTU 행들로 구성된다. 슬라이스는 다수의 타일들 또는 다수의 브릭들을 포함할 수 있다. 2개의 슬라이스 모드, 즉 래스터-스캔 슬라이스 모드 및 직사각형 슬라이스 모드가 지원될 수 있다. 래스터-스캔 슬라이스 모드에서는, 슬라이스가 픽처의 타일 래스터 스캔에서 타일들의 시퀀스를 포함할 수 있다. 직사각형 슬라이스 모드에서는, 슬라이스가 픽처의 직사각형 영역을 집합적으로 형성하는 픽처의 다수의 브릭들을 포함할 수 있다. 직사각형 슬라이스 내의 브릭들은 슬라이스의 브릭 래스터 스캔의 순서일 수 있다. 브릭들은 래스터 스캔 순서로 암시적으로 라벨링될 수 있고, 각각의 슬라이스 "i"에 대해 HLS에서 상부-좌측 브릭 인덱스(*top_left_brick_idx[i]*) 및 하부-우측 브릭 인덱스(*bottom_right_brick_idx[i]*)가 시그널링될 수 있다. 예를 들어, 첫 번째 슬라이스 0(도 7a에서 좌측)은 2개의 타일 및 3개의 브릭을 포함한다. 그것은 "brick_idx" 0, 1 및 2로 구성되고 top_left_brick_idx[0]=0, bottom_right_brick_idx[0]=2 및 bottom_right_brick_idx_delta[0]=2를 갖는다(*bottom_right_brick_idx_delta[i] = bottom_right_brick_idx[i] - top_left_brick_idx[i]*). *두 번째 슬라이스 1(도 7a의 상부-우측)은 2개의 브릭을 포함한다.* 그것은 "brick_idx" 3 및 4로 구성되고 top_left_brick_idx[0]=3, bottom_right_brick_idx[0]=4 및 bottom_right_brick_idx_delta[0]=1을 갖는다. *세 번째 슬라이스 2(도 7a의 중간-우측)는 3개의 브릭을 포함한다.* 그것은 "brick_idx" 5, 6 및 7로 구성되고 top_left_brick_idx[0]=5, bottom_right_brick_idx[0]=7 및 bottom_right_brick_idx_delta[0]=2를 갖는다. *네 번째 슬라이스 3(도 7a의 하부-우측)은 3개의 브릭을 포함한다.* 그것은 "brick_idx" 8, 9 및 10으로 구성되

고 top_left_brick_idx[0]=8, bottom_right_brick_idx[0]=10 및 bottom_right_brick_idx_delta[0]=2를 갖는다.

[0085] 이들 타일, 브릭 및 타일 그룹(group-of-tiles)의 이름은 변경될 수 있지만, 기본 원리는 유지된다는 점에 유의한다.

[0086] 비디오 비트스트림에서, 슬라이스 헤더들, 타일 헤더들 등은, 대부분의 시간 동안 그것들이 동일한 값을 공유하더라도, 각각의 코딩된 픽처에 대해 적어도 한 번 반복된다.

[0087] 본 실시예들은 디코딩 데이터가 비디오 비트스트림에 코딩되어 있는지 또는 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 이용함으로써 비디오 비트스트림에서 디코딩 데이터를 시그널링한다.

[0088] 도 8은 적어도 하나의 실시예에 따른 디코딩 데이터의 세트를 시그널링하기 위한 방법(800)의 플로차트를 예시한다.

[0089] 단계 810에서는, 방법이 세트의 디코딩된 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 비디오 비트스트림의 이전 데이터 PD로부터 추론되는지를 체크할 수 있다. 구문 요소 F가 비디오 비트스트림에 추가될 수 있다. 세트의 디코딩된 데이터가 이전 데이터 PD로부터 추론된다면(단계 830), 구문 요소 F는 세트의 디코딩 데이터가 이전 데이터 PD로부터 추론됨을 지시한다. 세트의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있다면(단계 820), 구문 요소 F는 세트의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시한다.

[0090] 단계 840에서는, 방법이 비디오 비트스트림으로부터 구문 요소 F에 액세스할 수 있다.

[0091] 단계 850에서는, 방법이 구문 요소 F가 디코딩 데이터의 세트의 디코딩 데이터가 비디오 비트스트림의 이전 데이터 PD로부터 추론됨을 지시하는지를 체크할 수 있다. 그 경우(단계 870), 세트의 디코딩 데이터가 이전 데이터 PD로부터 추론된다. 구문 요소 F가 세트의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시한다면, (단계 860), 세트의 디코딩 데이터가 비디오 비트스트림으로부터 명시적으로 디코딩된다.

[0092] 도 9는 구문 요소가 세트의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시할 때의 도 8의 방법의 변형의 플로차트를 예시한다.

[0093] 단계 910에서는, 디코딩 데이터의 세트가 4개의 서브세트 i로 분할되는 도 10에 예시된 바와 같이 디코딩 데이터의 세트가 서브세트들로 분할된다.

[0094] 단계 910 다음에는 방법(800)(도 8)의 단계들 810-830의 반복이 이어진다. 각각의 반복에서는, 디코딩 데이터의 현재 서브세트 i가 단계 810의 입력으로 간주되고(단계 920) 그 과정에서 방법은 현재 서브세트 i의 디코딩된 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 비디오 비트스트림의 이전 데이터 PDi로부터 추론되는지를 체크할 수 있다. 구문 요소 Fi가 비디오 비트스트림에 추가될 수 있다. 현재 서브세트 i의 디코딩된 데이터가 이전 데이터 PDi로부터 추론된다면(단계 830), 구문 요소 Fi는 현재 서브세트 i의 디코딩 데이터가 이전 데이터 PDi로부터 추론됨을 지시한다. 현재 서브세트 i의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있다면(단계 820), 구문 요소 Fi는 서브세트 i의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시한다.

[0095] 모든 서브세트들 i가 고려될 때, 방법(800)(도 8)의 단계들 840-870이 반복된다. 각각의 반복에서는, 디코딩 데이터의 현재 서브세트 i가 단계 840의 입력으로 간주되고(단계 930) 그 과정에서 방법은 비디오 비트스트림으로부터 현재 서브세트 i에 대한 구문 요소 Fi에 액세스할 수 있다.

[0096] 단계 850에서는, 방법이 구문 요소 Fi가 디코딩 데이터의 서브세트 i의 디코딩 데이터가 비디오 비트스트림의 이전 데이터 PDi로부터 추론됨을 지시하는지를 체크할 수 있다. 그 경우(단계 870), 서브세트 i의 디코딩 데이터가 비디오 비트스트림의 이전 데이터 PDi로부터 추론된다. 구문 요소 Fi가 서브세트 i의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시한다면, (단계 860), 서브세트 i의 디코딩 데이터가 비디오 비트스트림으로부터 명시적으로 디코딩된다.

[0097] 인덱스 i(Fi, FDi)는 구문 요소 및 이전 데이터가 서브세트 i의 디코딩 데이터에 전용됨을 지시한다. 일부 Fi 및/또는 FDi는 동일할 수 있다.

[0098] 도 10은 디코딩 데이터의 세트가 서브세트들로 분할될 때의 방법(800)의 이용을 예시한다.

[0099] 변형에서는, 방법(800)이 디코딩 데이터의 세트가 다음과 같이 재귀적으로 분할되어 다수의 계층 레벨들을 생성할 때 이용될 수 있다. 구문 요소가 계층 레벨 L에서의 서브세트의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시할 때(단계 830) 계층 레벨 L에서의 서브세트가 계층 레벨 (L+1)에서의 서브세트들로 분할된다. 재귀적 분할은, 예를 들어, 계층 레벨들의 최대 수 또는 서브세트의 최소 카디널리티(cardinality)와 같은 조건이 충족될 때 정지한다.

[0100] 계층 레벨 L의 디코딩 데이터의 모든 서브세트들이 고려되었을 때, 방법(900)은 새로운 계층 레벨 (L+1)의 서브세트들(존재한다면)에 대해 반복적으로 적용된다.

[0101] 도 11은 16개의 디코딩 데이터의 세트의 2개의 계층 레벨 분할의 예를 예시한다.

[0102] 제1 계층 레벨 0에서는, 세트가 먼저 8개의 디코딩 데이터의 2개의 서브세트(i=0 및 i=1)로 분할된다(810). 단계 820에서는, 디코딩 데이터의 서브세트(i=0)가, 예를 들어, 2개의 서브-서브세트(i=00 및 i=01)로 분할됨을 지시하기 위해 구문 요소가 비트스트림에 추가된다. 두 번째 서브세트(i=1)의 디코딩 데이터가 비디오 비트스트림의 이전 데이터로부터 추론되고(단계 830) 두 번째 서브세트(i=1)가 분할되지 않음을 지시하기 위해 다른 구문 요소가 또한 비디오 비트스트림에 추가된다.

[0103] 제2 계층 레벨 1에서는, 서브-서브세트(i=00)의 디코딩 데이터가 비디오 비트스트림의 이전 데이터로부터 추론되고(단계 830) 서브-서브세트(i=00)가 분할되지 않음을 지시하기 위해 다른 구문 요소가 또한 비디오 비트스트림에 추가된다. 서브-서브세트(i=01)의 디코딩 데이터가 비디오 비트스트림의 이전 데이터로부터 추론되고(단계 830) 서브-서브세트(i=01)가 분할되지 않음을 지시하기 위해 다른 구문 요소가 또한 비디오 비트스트림에 추가된다.

[0104] 일 실시예에서는, 디코딩 데이터의 세트가 코딩 툴의 활성화를 제어하는 제약-플래그들을 포함할 수 있다. 그러면 구문 요소 F(또는 Fi)가 세트(또는 세트의 서브세트)의 제약-플래그들이 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 비디오 비트스트림의 이전 데이터 PD로부터 추론되는지를 지시할 수 있다.

[0105] 도 12는 적어도 하나의 실시예에 따른 HLS 요소 "general_constraint_info"의 예를 예시한다.

[0106] 구문 요소 "general_constraint_info"는 구문 요소 "use_default_constraint_flag"(구문 요소 F 또는 Fi) 하에서 함께 그룹화되는 제약-플래그들을 포함한다. 이 구문 요소 "use_default_constraint_flag"는 이들 제약-플래그가 비디오 비트스트림에 명시적으로 코딩되어 있거나(단계 820), 비디오 비트스트림으로부터 명시적으로 디코딩될 때(단계 860) 거짓(false), 또는 이들 제약-플래그가 모두 비디오 비트스트림의 이전 데이터 PD로부터 추론될 때(단계 830, 870) 참(true)인 플래그일 수 있다.

[0107] 제약-플래그들이 추론될 때 다른 제약-플래그들(여기에 도시되지 않음)이 명시적으로 코딩/디코딩될 수 있다는 점에 유의한다.

[0108] 변형에서는, 이전 데이터 PD가 단일 이진 값(거짓 또는 참)이고 모든 추론된 제약-플래그 값들은 이 단일 이진 값 FD와 동등하다.

[0109] 변형에서는, 이전 데이터 PD가 비디오 비트스트림의 디코딩 프로파일, 레벨 및/또는 적어도 하나의 디코딩 파라미터에 의존한다. 예를 들어, "하이 프로파일"에 대해서는 모든 추론된 제약-플래그 값들이 거짓(아마도 모든 툴들이 인에이블됨)과 동등하고 "로우 프로파일"에 대해서는 제약-플래그 값들의 일부 미리 결정된 서브세트가 참(툴들이 디스에이블됨)으로 추론되고 다른 제약-플래그 값들은 거짓(아마도 툴들이 인에이블됨)으로 추론된다.

[0110] 변형에서는, 구문 요소 F(또는 Fi)의 상태가 디코딩 파라미터들이 비디오 비트스트림에 존재하는지 여부를 결정한다.

[0111] 예를 들어, 도 13의 구문 요소들의 세트 "profile_tier_level( )"은 구문 요소 "use_default_constraint-flag"(구문 요소 F 또는 Fi) 하의 제약-플래그들 "general_constraint_info()"를 포함한다. 도 13에서, 구문 요소 "use_default_constraint_flag"가 참이면, 구문 요소들의 세트 "general_constraint_info()"는 비디오 비트스트림에 명시적으로 코딩되어 있다. "use_default_constraint_flag"가 거짓이면, 이 구문 요소들의 세트가 존재하지 않고 구문 요소들이 추론된다.

[0112] 도 12에 예시된 변형에서는, 구문 요소 "use_default_constraint_flag"가 "profile_tier_level()"이 아니라 "general_constraint_info()" 내에 코딩되어 있다. 구문 요소 "use_default_constraint_flag"가 거짓이면 일

부 툴들이 아마도 비트스트림에서 인에이블되는지 또는 디스에이블되는지를 지시하는 구문 요소들의 세트가 비디오 비트스트림에 존재하고(명시적으로 코딩되어 있고) 그렇지 않으면 이 구문 요소들의 세트가 존재하지 않고 구문 요소들이 추론된다.

[0113] 블록 기반 비디오 코덱 사양들에서는, 비디오 비트스트림이 디코딩 순서로 하나 이상의 액세스 유닛(AU)을 포함하는 코딩된 비디오 시퀀스들(CVS)로 파티셔닝된다. 하나의 AU는 정확히 하나의 코딩된 픽처를 포함한다.

[0114] 예를 들어, HEVC에서는, AU들이 슬라이스들로 파티셔닝된다. 상이한 슬라이스 헤더들(또는 타일 헤더들, 또는 타일 그룹 헤더들) 내의 일부 디코딩 데이터가 여러 슬라이스들에 대해 동일하다. 그 때문에, HEVC에서는, 도 14에 묘사된 바와 같이, 픽처 내의 첫 번째 슬라이스는 반드시 독립적인 반면 후속 슬라이스들은 종속적일 수 있다. 종속적 슬라이스 헤더들은 동일한 픽처 내의 첫 번째 독립적 슬라이스와 공통 디코딩 데이터를 공유한다(도 15).

[0115] 도 16은 디코딩 데이터의 세트가, 예를 들어, 픽처의 슬라이스 헤더 및/또는 타일 헤더 및/또는 타일-그룹 헤더와 같은 픽처 부분들의 그룹의 헤더의 디코딩 데이터를 포함할 때의 방법(800 또는 900)의 실시예의 플로차트(1600)를 예시한다. HEVC에서는, 슬라이스들이 헤더들을 갖고 타일들을 포함할 수 있지만, 타일들은 헤더들을 갖지 않는다. 다른 코덱들에서는 슬라이스들이 타일들로 대체되고 타일들은 헤더들을 갖는다.

[0116] HEVC에서는, 그러한 디코딩의 예들이 다음과 같은 디코딩 데이터(슬라이스 헤더) 중 적어도 하나를 포함할 수 있다:

[0117] - **(slice_sao_luma_flag, slice_sao_chroma_flag, slice_alf_luma_flag, slice_alf_chroma_flag)**와 같은 포스트 필터(post filters) 플래그들.

[0118] - **참조 픽처 목록**,

[0119] - **weighted_pred/bipred_flag**: 슬라이스에서 WP(Weighted Prediction)의 인에이블 여부,

[0120] - 슬라이스들, 타일들 또는 타일-그룹들, 브릭들로의 파티셔닝(예를 들어, num_bricks_in_slice_minus1)

[0121] - **Slice_type**: I(인트라만), P(단방향) 또는 B(양방향 예측),

[0122] - **pic_output_flag**: 현재 픽처가 출력될 것인지의 여부,

[0123] - **colour_plane_flag**: 컬러 채널들이 개별적으로 코딩되어 있는지 여부,

[0124] - **POC**: Picture Order Count,

[0125] - **long-term-ref info**: 장기 참조 픽처 인덱스들(long-term reference pictures indexes) 및 POC들

[0126] - **slice_temporal_mvp_enabled_flag**: 시간 모션 벡터들 예측 툴 인에이블 여부,

[0127] - long-term-ref info (POC)

[0128] - **slice_qp_delta**: PPS에 init_QP가 특정된 QP(Quantization Parameter) start diff,

[0129] - **디블로킹 필터** 파라미터들(beta, tc).

[0130] 다른 코덱들에서는, 이들 데이터가 타일 헤더에 포함될 수 있다. 이하에서는, 픽처가 영역들(또는 픽처 부분들)로 파티셔닝되는 것을 고려하고, 하나의 영역은 여러 블록들(예를 들어, CTU들 또는 매크로-블록들)로 구성된다. 영역은, 코덱의 파티셔닝 토폴로지에 따라, 예를 들어 슬라이스, 타일, 타일 그룹 또는 브릭일 수 있다. 그것들 중 일부는 영역을 구성하는 블록들과 관련된 정보를 포함하는 헤더와 연관될 수 있다.

[0131] 단계 1610(도 8의 단계 810의 실시예)에서는, 방법이 픽처 부분들의 그룹(또는 영역들)의 헤더의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 비디오 비트스트림의 이전 데이터 IPH("inter_picture-header")로부터 추론되는지를 체크할 수 있다. "infer_from_inter_picture_header_flag"로 나타내어진 구문 요소 F(또는 Fi)가 비디오 비트스트림에 추가될 수 있다. 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 이전 데이터 IPH로부터 추론된다면, 단계 1630(도 8의 단계 830의 실시예)에서, 구문 요소 F(또는 Fi)가 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 픽처 부분들의 그룹의 다른 헤더의 디코딩 데이터(이전 데이터 IPH)로부터 추론됨을 지시한다. 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있다면, 단계 1520(도 8의 단계 820의 실시예)에서, 구문 요소 F(또는 Fi)가 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시할 수 있다.

[0132] 이전 데이터 IPH는, 예를 들어, 동일한 픽처의 다수의 슬라이스 헤더들 및/또는 공통인 다수의 타일 헤더들 및/또는 다수의 타일-그룹 헤더들의 디코딩 데이터를 그룹화한다.

[0133] 전형적으로 AU들, CTU들/CU들, 타일들 또는 타일 그룹들의 헤더들에 정의된 공통 디코딩 데이터를 추론하는 것은 비디오를 송신하기 위한 대역폭을 감소시킨다.

[0134] 예를 들어, 각각의 AU, 및 아마도 이 AU 내의 각각의 슬라이스에서, 구문 요소 "infer_from_inter_picture_header_flag"는 이들 디코딩 데이터가 AU에(이 AU 내의 각각의 슬라이스에), 예를 들어, 슬라이스 헤더, 타일 또는 타일-그룹 헤더에 명시적으로 정의되어 있는지, 또는, 그것들이 이전 데이터 IPH로부터 추론되는지 - 따라서 아마도 여러 픽처들의 여러 픽처 부분들의 그룹들에 의해 공유될 수 있는지 - 를 지시할 수 있다.

[0135] 단계 1640(도 8의 단계 840의 실시예)에서는, 방법이 비디오 비트스트림으로부터 구문 요소 F(또는 Fi)에 액세스할 수 있다.

[0136] 단계 1650(도 8의 단계 850의 실시예)에서는, 방법이 구문 요소 F(또는 Fi)가 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 비디오 비트스트림 내의 이전 데이터 IPH로부터 추론됨을 지시하는지를 체크할 수 있다. 그 후, 단계 1670(도 8의 단계 870의 실시예)에서는, 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 이전 데이터 IPH로부터 추론된다. 구문 요소 F(또는 Fi)가 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 비디오 비트스트림에 코딩되어 있음을 지시한다면, 단계 1660(도 8의 단계 860의 실시예)에서, 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 비디오 비트스트림으로부터 명시적으로 디코딩된다.

[0137] 유리하게는, 이전 데이터 IPH가 RAP 구문 요소 뒤에 그리고 RAP에 대해 AU 구문 요소들 앞에 GOP(Group Of Pictures)와 연관된 비트스트림에 위치한다. 그 후 도 17에 예시된 바와 같이 AU 구문 요소들의 디코딩 데이터가 이전 데이터 IPH로부터 추론될 수 있다.

[0138] 방법(1600)의 일 실시예에서는, 참조 픽처들의 목록(픽처 부분들의 그룹의 헤더의 디코딩 데이터)이 GOP 구조 및 참조 프레임들의 수(이전 데이터 IPH)로부터 추론될 수 있다.

[0139] 방법(1600)의 일 실시예에서는, 슬라이스 파라미터들(픽처 부분들의 그룹의 헤더의 디코딩 데이터)(예를 들어 슬라이스 유형)이 GOP 구조 및 디코딩된 픽처 순서(이전 데이터 IPH)로부터 추론될 수 있다.

[0140] 일 실시예에서는, 타일 그룹들의 코딩/디코딩을 제어하는 타일 그룹의 디코딩 데이터가 비디오 비트스트림의 다른 타일 그룹의 디코딩 데이터(이전 데이터 IPH)로부터 추론될 수 있다.

[0141] 비디오 비트스트림 내의 이전 데이터로부터 타일-그룹 정보를 추론하는 것은 픽처를 송신하기 위한 대역폭을 감소시킨다.

[0142] 변형에서는, 타일 그룹이 픽처 또는 슬라이스의 타일들의 수("num-tiles"로 나타내어진 디코딩 데이터)를 포함한다. 그러면 구문 요소(F, Fi, ...)가 열 당 타일들의 수("num_tile_columns_minus1"로 나타내어짐) 및 행 당 타일들의 수("num_tile_rows_minus1"로 나타내어짐)를 나타내는 타일들의 수 i가 비디오 비트스트림에 코딩되어 있는지 또는 이전 데이터 IPH로부터 추론되는지를 지시할 수 있다.

[0143] 예를 들어, "num-tiles"는 다음과 같이 "num_tile_columns_minus1" 및 "num_tile_rows_minus1"로부터 추론된다:

[0144] num_tiles = (num_tile_columns_minus1 + 1) x (num_tile_rows_minus1 + 1).

[0145] 실시예에서는, 타일 그룹은 픽처의 마지막 타일-그룹의 "bottom-right-tile-idx" 값 또는 "bottom-right-tile-idx-delta" 값을 추가로 포함한다.

[0146] 마지막 타일-그룹(디코딩 데이터)의 *bottom-right-tile-idx* 값 또는 *bottom-right-tile-idx-delta* 값(디코딩 데이터)은 다음과 같이 타일들의 수(비디오 비트스트림의 이전 데이터)로부터 추론될 수 있다:

[0147] bottom_right_tile_idx[num_tile_groups_in_pic_minus1] = num_tile - 1 bottom_right_tile_idx_delta[num_tile_groups_in_pic_minus1] = num_tile - 1 -top_left_tile_idx[num_tile_groups_in_pic_minus1].

[0148] 이 실시예의, 도 18에 예시된 이점 중 하나는 "bottom_right_tile_idx_delta[num_tile_groups_in_pic_minus1]"을 코딩하기 위한 비트들을 구문 요소 "pic_parameter_set_rbsp"에 저장하는 것이다.

[0149] 유사하게, 픽처 내의 브릭들의 수(NumBricksInPic)가 이전 데이터 IPH로부터 추론될 수 있다. 직사각형 슬라이스 모드의 경우, 슬라이스들은 픽처의 슬라이스의 "bottom-right-brick-idx" 값 또는 "bottom-right-brick-idx-delta" 값을 추가로 포함할 수 있다.

[0150] 마지막 슬라이스(디코딩 데이터)의 *bottom-right-brick-idx* 값 또는 *bottom-right-brick-idx-delta* 값(디코딩 데이터)은 다음과 같이 브릭들의 수(비디오 비트스트림의 이전 데이터)로부터 추론될 수 있다:

[0151] bottom_right_brick_idx[num_slices_in_pic_minus1] = NumBricksInPic - 1

[0152] bottom_right_brick_idx_delta[num_slices_in_pic_minus1] = NumBricksInPic - 1 - top_left_brick_idx[num_slices_in_pic_minus1]

[0153] 이 실시예의 이점 중 하나는 "bottom_right_brick_idx_delta[num_slices_in_pic_minus1]"을 코딩하기 위한 비트들을 구문 요소 "pic_parameter_set_rbsp"에 저장하는 것이다.

[0154] 도 19는 디코딩 데이터의 세트가 GOP(Group of Pictures)의 구조를 포함할 때의 방법(800 또는 900)의 실시예의 플로차트(1900)를 예시한다.

[0155] 단계 1910(도 8의 단계 810의 실시예)에서는, 방법이 GOP 구조가 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 비디오 비트스트림 내의 이전 데이터 PD로부터 추론되는지를 체크할 수 있다. "GOP_structure_indicator"로 나타내어진 구문 요소 F(또는 Fi)가 비디오 비트스트림에 추가될 수 있다. GOP 구조가 이전 데이터 PD로부터 추론된다면, 단계 1930(도 8의 단계 830의 실시예)에서, 구문 요소 F(또는 Fi)가 GOP 구조가 이전 데이터 PD로부터 추론됨을 지시한다. GOP 구조가 비디오 비트스트림에 명시적으로 코딩되어 있다면, 단계 1920(도 8의 단계 820의 실시예)에서, 구문 요소 F(또는 Fi)가 GOP 구조가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시할 수 있다.

[0156] 변형에서는, GOP 구조가 비디오 비트스트림 내의 이전 데이터 PD로부터 추론될 때, 구문 요소 F가 비디오 비트스트림에 추가되지 않는다.

[0157] GOP 구조들을 추론하는 것은 비디오를 송신하기 위한 대역폭을 감소시킨다.

[0158] 단계 1940(도 8의 단계 830의 실시예)에서는, 방법이 비디오 비트스트림으로부터 구문 요소 F(또는 Fi)에 액세스할 수 있다.

[0159] 변형에서는, 방법이 구문 요소 F(또는 Fi)가 비디오 비트스트림에 존재하는지를 체크한다.

[0160] 단계 1950(도 8의 단계 850의 실시예)에서는, 방법이 구문 요소 F(또는 Fi)가 GOP 구조가 비디오 비트스트림의 이전 데이터 PD로부터 추론됨을 지시하는지, 또는 변형에 따라 구문 요소 F(또는 Fi)가 비디오 비트스트림에 존재하지 않는지를 체크할 수 있다. 그러면, 단계 1970(도 8의 단계 870의 실시예)에서, GOP 구조가 이전 데이터 PD로부터 추론된다. 구문 요소 F(또는 Fi)가 GOP 구조가 비디오 비트스트림에 명시적으로 코딩되어 있음을 지시한다면, 또는 변형에 따라 구문 요소 F(또는 Fi)가 비디오 비트스트림에 존재한다면, 단계 1960(도 8의 단계 860의 실시예)에서, GOP 구조가 비디오 비트스트림으로부터 명시적으로 디코딩된다.

[0161] 일 실시예에서는, 구문 요소 F(또는 Fi)가 미리 정의된 GOP 구조들의 세트의 미리 정의된 GOP 구조 인덱스를 시그널링할 수 있다.

[0162] 미리 정의된 GOP 구조들의 예들은 도 20에 예시된 바와 같이 Random-Access(RA: 계층적 코딩), Low-Delay-B(LB), Low-Delay-P(LP)일 수 있다.

[0163] 도 20에 예시된 방법(1900)의 변형에서는, GOP 길이와 같은 추가 구문 요소가 미리 정의된 GOP 구조 중 어느 것이 시그널링되는지를 정밀화하기 위해 사용될 수 있다.

[0164] 일 실시예에서는, 이전 데이터 PD가 GOP 및 미리 정의된 GOP 구조들의 세트 내의 일부 먼저 디코딩된 픽처들을 포함한다.

[0165] 예를 들어, 도 20의 미리 정의된 GOP 구조에 기초하여, 먼저 디코딩된 픽처 POC들이 0-16-8-4이면, 추론된 GOP 구조는 인덱스 "1"을 갖는 미리 정의된 GOP 구조와 관련된다.

[0166] 도 21은 다양한 양태들 및 실시예들이 구현되는 시스템의 예의 블록도를 예시한다.

[0167] 시스템(2100)은 아래에 설명되는 다양한 컴포넌트들을 포함하는 디바이스로서 구현될 수 있고 이 문서에 설명되

는 양태들 중 하나 이상을 수행하도록 구성된다. 그러한 디바이스들의 예들은 개인용 컴퓨터들, 랩톱 컴퓨터들, 스마트폰들, 태블릿 컴퓨터들, 디지털 멀티미디어 셋톱 박스들, 디지털 텔레비전 수신기들, 개인용 비디오 레코딩 시스템들, 연결된 가전 기기들, 및 서버들과 같은 다양한 전자 디바이스들을 포함하지만, 이에 제한되지 않는다. 시스템(2100)의 요소들은, 단독으로 또는 조합하여, 단일 집적 회로(IC), 다수의 IC들, 및/또는 개별 컴포넌트들로 구현될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 시스템(2100)의 처리 및 인코더/디코더 요소들은 다수의 IC들 및/또는 개별 컴포넌트들에 걸쳐 분산된다. 다양한 실시예들에서는, 시스템(2100)이, 예를 들어, 통신 버스를 통해 또는 전용 입력 및/또는 출력 포트들을 통해 하나 이상의 다른 시스템, 또는 다른 전자 디바이스에 통신가능하게 결합된다. 다양한 실시예들에서는, 시스템(2100)이 이 문서에 설명된 양태들 중 하나 이상을 구현하도록 구성된다.

[0168] 시스템(2100)은, 예를 들어, 이 문서에 설명된 다양한 양태들을 구현하기 위해 그 안에 로딩된 명령어들을 실행하도록 구성된 적어도 하나의 프로세서(2110)를 포함한다. 프로세서(2110)는 임베디드 메모리, 입력 출력 인터페이스, 및 본 기술분야에 알려진 바와 같은 다양한 다른 회로들을 포함할 수 있다. 시스템(2100)은 적어도 하나의 메모리(2120)(예를 들어, 휘발성 메모리 디바이스 및/또는 비휘발성 메모리 디바이스)를 포함한다. 시스템(2100)은, 비휘발성 메모리 및/또는 휘발성 메모리를 포함할 수 있는, 저장 디바이스(2140)를 포함하고, 이는 EEPROM(Electrically Erasable Programmable Read-Only Memory), ROM(Read-Only Memory), PROM(Programmable Read-Only Memory), RAM(Random Access Memory), DRAM(Dynamic Random Access Memory), SRAM(Static Random Access Memory), 플래시, 자기 디스크 드라이브, 및/또는 광 디스크 드라이브를 포함하지만, 이에 제한되지 않는다. 저장 디바이스(2140)는, 비제한적인 예로서, 내부 저장 디바이스, 부착된 저장 디바이스(착탈식 및 비-착탈식 저장 디바이스를 포함함), 및/또는 네트워크 액세스 가능한 저장 디바이스를 포함할 수 있다.

[0169] 시스템(2100)은, 예를 들어, 데이터를 처리하여 인코딩된 비디오 또는 디코딩된 비디오를 제공하도록 구성된 인코더/디코더 모듈(2130)을 포함하고, 인코더/디코더 모듈(2130)은 그 자신의 프로세서 및 메모리를 포함할 수 있다. 인코더/디코더 모듈(2130)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 다를 포함할 수 있다. 추가적으로, 인코더/디코더 모듈(2130)은 시스템(2100)의 개별 요소로서 구현될 수 있거나 본 기술분야의 숙련자들에게 알려진 바와 같은 하드웨어 및 소프트웨어의 조합으로서 프로세서(2110) 내에 통합될 수 있다. 이 문서에 설명된 다양한 양태들을 수행하기 위해 프로세서(2110) 또는 인코더/디코더(2130) 상에 로딩될 프로그램 코드는 저장 디바이스(2140)에 저장될 수 있고 후속하여 프로세서(2110)에 의한 실행을 위해 메모리(2120) 상에 로딩될 수 있다. 다양한 실시예들에 따르면, 프로세서(2110), 메모리(2120), 저장 디바이스(2140), 및 인코더/디코더 모듈(2130) 중 하나 이상은 이 문서에 설명된 프로세스들의 수행 동안 다양한 항목들 중 하나 이상을 저장할 수 있다. 그러한 저장된 항목들은 입력 비디오, 디코딩된 비디오 또는 디코딩된 비디오의 부분들, 비트스트림, 행렬들, 변수들, 및 수학식들, 공식들, 연산들 및 연산 로직의 처리로부터의 중간 또는 최종 결과들을 포함할 수 있지만, 이에 제한되지 않는다.

[0170] 일부 실시예들에서는, 프로세서(2110) 및/또는 인코더/디코더 모듈(2130) 내부의 메모리가 명령어들을 저장하고 인코딩 또는 디코딩 동안에 필요한 처리를 위한 작업 메모리를 제공하기 위해 이용된다. 그러나, 다른 실시예들에서는, 처리 디바이스(예를 들어, 처리 디바이스는 프로세서(2110) 또는 인코더/디코더 모듈(2130) 중 어느 하나일 수 있음) 외부의 메모리가 이들 기능 중 하나 이상을 위해 사용된다. 외부 메모리는 메모리(2120) 및/또는 저장 디바이스(2140), 예를 들어, 동적 휘발성 메모리 및/또는 비휘발성 플래시 메모리일 수 있다. 여러 실시예들에서는, 예를 들어, 텔레비전의 운영 체제를 저장하기 위해 외부 비휘발성 플래시 메모리가 사용된다. 적어도 하나의 실시예에서는, 예컨대 MPEG-2(MPEG는 Moving Picture Experts Group를 언급하고, MPEG-2는 ISO/IEC 13818이라고도 언급되고, 13818-1은 H.222라고도 알려져 있고, 13818-2는 H.262라고도 알려져 있음), HEVC(HEVC는 High Efficiency Video Coding를 언급하고, H.265 및 MPEG-H 파트 2라고도 알려져 있음), 또는 VVC(Versatile Video Coding, JVET(Joint Video Experts Team)에 의해 개발되고 있는 새로운 표준)에 대한, 비디오 코딩 및 디코딩 동작들을 위한 작업 메모리로서 RAM과 같은 고속 외부 동적 휘발성 메모리가 사용된다.

[0171] 시스템(2100)의 요소들에 대한 입력은 블록(2230)에 지시된 바와 같은 다양한 입력 디바이스들을 통해 제공될 수 있다. 그러한 입력 디바이스들은 (i) 예를 들어, 방송사에 의해 무선으로 송신된 RF 신호를 수신하는 무선 주파수(RF) 부분, (ii) 컴포넌트(COMP) 입력 단자(또는 COMP 입력 단자들의 세트), (iii) USB(Universal Serial Bus) 입력 단자, 및/또는 (iv) HDMI(High Definition Multimedia Interface) 입력 단자를 포함하지만, 이에 제한되지 않는다. 도 21에 도시되지 않은 다른 예들은 합성 비디오를 포함한다.

[0172] 다양한 실시예들에서는, 블록(2230)의 입력 디바이스들이 본 기술분야에 알려진 바와 같은 연관된 각자의 입력

처리 요소들을 갖는다. 예를 들어, RF 부분은 (i) 원하는 주파수를 선택하는 것(신호를 선택하는 것, 또는 신호를 주파수들의 대역으로 대역 제한하는 것이라고도 언급됨), (ii) 선택된 신호를 다운-컨버팅하는 것, (iii) 특정 실시예들에서 채널이라고 언급될 수 있는 (예를 들어) 신호 주파수 대역을 선택하기 위해 더 좁은 주파수들의 대역으로 다시 대역 제한하는 것, (iv) 다운-컨버팅되고 대역 제한된 신호를 복조하는 것, (v) 오류 정정을 수행하는 것, 및 (vi) 원하는 데이터 패킷들의 스트림을 선택하기 위해 디멀티플렉싱하는 것에 적합한 요소들과 연관될 수 있다. 다양한 실시예들의 RF 부분은 이들 기능을 수행하기 위한 하나 이상의 요소, 예를 들어, 주파수 선택기들, 신호 선택기들, 대역 제한기들, 채널 선택기들, 필터들, 다운컨버터들, 복조기들, 오류 정정기들 및 디멀티플렉서들을 포함한다. RF 부분은, 예를 들어, 수신된 신호를 더 낮은 주파수(예를 들어, 중간 주파수 또는 근 기저대역 주파수)로 또는 기저대역으로 다운-컨버팅하는 것을 포함하여, 다양한 이들 기능을 수행하는 튜너를 포함할 수 있다. 하나의 셋톱 박스 실시예에서는, RF 부분 및 그것의 연관된 입력 처리 요소가 유선(예를 들어, 케이블) 매체를 통해 송신되는 RF 신호를 수신하고, 필터링, 다운-컨버팅, 및 원하는 주파수 대역으로 다시 필터링에 의해 주파수 선택을 수행한다. 다양한 실시예들은 위에 기술된(및 다른) 요소들의 순서를 재배열하고, 이들 요소 중 일부를 제거하고/하거나, 유사하거나 상이한 기능들을 수행하는 다른 요소들을 추가한다. 요소들을 추가하는 것은, 예를 들어, 증폭기들 및 아날로그-디지털 컨버터를 삽입하는 것과 같은, 기존 요소들 간에 요소들을 삽입하는 것을 포함할 수 있다. 다양한 실시예들에서는, RF 부분이 안테나를 포함한다.

[0173] 추가적으로, USB 및/또는 HDMI 단말은 시스템(2100)을 USB 및/또는 HDMI 연결들을 통해 다른 전자 디바이스에 연결하기 위한 각자의 인터페이스 프로세서를 포함할 수 있다. 입력 처리의 다양한 양태들, 예를 들어, 리드-솔로몬(Reed-Solomon) 오류 정정이, 예를 들어, 별도의 입력 처리 IC 내에 또는 필요에 따라 프로세서(2110) 내에 구현될 수 있다는 점을 이해해야 한다. 유사하게, USB 또는 HDMI 인터페이스 처리의 양태들이 별도의 인터페이스 IC들 내에 또는 필요에 따라 프로세서(2110) 내에 구현될 수 있다. 복조된, 오류 정정된, 및 디멀티플렉싱된 스트림은 출력 디바이스 상에서의 프레젠테이션을 위해 필요에 따라 데이터 스트림을 처리하기 위해, 예를 들어, 프로세서(2110), 및 메모리 및 저장 요소들과 조합하여 동작하는 인코더/디코더(2130)를 포함한 다양한 처리 요소들에 제공된다.

[0174] 시스템(2100)의 다양한 요소들이 통합된 하우징 내에 제공될 수 있고, 통합된 하우징 내에서, 다양한 요소들은 적합한 연결 배열, 예를 들어, I2C(Inter-IC) 버스, 와이어링, 및 인쇄 회로 기판들을 포함하여, 본 기술분야에 알려진 바와 같은 내부 버스를 이용하여 상호연결되고 그들 간에 데이터를 송신할 수 있다.

[0175] 시스템(2100)은 통신 채널(2160)을 통해 다른 디바이스들과의 통신을 가능하게 하는 통신 인터페이스(2150)를 포함한다. 통신 인터페이스(2150)는 통신 채널(2160)을 통해 데이터를 송신하고 수신하도록 구성된 트랜시버를 포함할 수 있지만, 이에 제한되지 않는다. 통신 인터페이스(2150)는 모뎀 또는 네트워크 카드를 포함할 수 있지만, 이에 제한되지 않고, 통신 채널(2160)은, 예를 들어, 유선 및/또는 무선 매체 내에 구현될 수 있다.

[0176] 데이터는, 다양한 실시예들에서, Wi-Fi 네트워크, 예를 들어, IEEE 802.11(IEEE는 Institute of Electrical and Electronics Engineers를 언급함)과 같은 무선 네트워크를 이용하여 시스템(2100)에 스트리밍되거나 달리 제공된다. 이들 실시예의 Wi-Fi 신호는 Wi-Fi 통신을 위해 적응되는 통신 채널(2160) 및 통신 인터페이스(2150)를 통해 수신된다. 이들 실시예의 통신 채널(2160)은 스트리밍 애플리케이션들 및 다른 오버-더-톱(over-the-top) 통신을 허용하기 위해 인터넷을 포함하는 외부 네트워크들로의 액세스를 제공하는 액세스 포인트 또는 라우터에 전형적으로 연결된다. 다른 실시예들은 입력 블록(2130)의 HDMI 연결을 통해 데이터를 전달하는 셋톱 박스를 이용하여 스트리밍된 데이터를 시스템(2100)에 제공한다. 또 다른 실시예들은 입력 블록(2130)의 RF 연결을 이용하여 스트리밍된 데이터를 시스템(2100)에 제공한다. 위에 지시된 바와 같이, 다양한 실시예들은 비-스트리밍 방식으로 데이터를 제공한다. 추가적으로, 다양한 실시예들은 Wi-Fi와는 다른 무선 네트워크들, 예를 들어, 셀룰러 네트워크 또는 블루투스 네트워크를 이용한다.

[0177] 시스템(2100)은 디스플레이(2200), 스피커들(2210), 및 다른 주변 디바이스들(2220)을 포함하는 다양한 출력 디바이스들에 출력 신호를 제공할 수 있다. 다양한 실시예들의 디스플레이(2200)는, 예를 들어, 터치스크린 디스플레이, 유기 발광 다이오드(OLED) 디스플레이, 곡면 디스플레이, 및/또는 폴더블 디스플레이 중 하나 이상을 포함한다. 디스플레이(2200)는 텔레비전, 태블릿, 랩톱, 셀 전화기(휴대폰), 또는 다른 디바이스를 위한 것일 수 있다. 디스플레이(2200)는 또한 다른 컴포넌트들과 통합되거나(예를 들어, 스마트 폰에서와 같이), 또는 분리될 수 있다(예를 들어, 랩톱을 위한 외부 모니터). 다른 주변 디바이스들(2220)은, 실시예들의 다양한 예들에서, 독립형 디지털 비디오 디스크(또는 디지털 다기능 디스크)(양쪽 용어들을 위해, DVR), 디스크 플레이어, 스테레오 시스템, 및/또는 조명 시스템 중 하나 이상을 포함한다. 다양한 실시예들은 시스템(2100)의 출력에

기초하여 기능을 제공하는 하나 이상의 주변 디바이스들(2220)을 사용한다. 예를 들어, 디스크 플레이어가 시스템(2100)의 출력을 재생하는 기능을 수행한다.

[0178] 다양한 실시예들에서는, 제어 신호들은 AV.Link, CEC(Consumer Electronics Control), 또는 사용자 개입과 함께 또는 사용자 개입 없이 디바이스 대 디바이스 제어를 가능하게 하는 다른 통신 프로토콜들과 같은 시그널링을 이용하여 시스템(2100)과 디스플레이(2200), 스피커들(2210), 또는 다른 주변 디바이스들(2220) 간에 통신된다. 출력 디바이스들은 각자의 인터페이스들(2170, 2180, 및 2190)을 통한 전용 연결들을 통해 시스템(2100)에 통신가능하게 결합될 수 있다. 대안적으로, 출력 디바이스들은 통신 인터페이스(2150)를 통해 통신 채널(2160)을 이용하여 시스템(2100)에 연결될 수 있다. 디스플레이(2200) 및 스피커들(2210)은, 예를 들어, 텔레비전과 같은 전자 디바이스 내의 시스템(2100)의 다른 컴포넌트들과 함께 단일 유닛에 통합될 수 있다. 다양한 실시예들에서는, 디스플레이 인터페이스(2170)가, 예를 들어, 타이밍 컨트롤러(T Con) 칩과 같은 디스플레이 드라이버를 포함한다.

[0179] 대안적으로, 예를 들어, 입력(2230)의 RF 부분이 별도의 셋톱 박스의 일부이면, 디스플레이(2200) 및 스피커(2210)는 다른 컴포넌트들 중 하나 이상과 분리될 수 있다. 디스플레이(2200) 및 스피커들(2210)이 외부 컴포넌트들인 다양한 실시예들에서는, 출력 신호가, 예를 들어, HDMI 포트들, USB 포트들, 또는 COMP 출력들을 포함하는 전용 출력 연결들을 통해 제공될 수 있다.

[0180] 본 명세서에 기술된 구현들은, 예를 들어, 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터 스트림, 또는 신호에서 구현될 수 있다. 단일 형식의 구현의 컨텍스트에서만 논의(예를 들어, 방법으로서만 논의)되더라도, 논의된 특징들의 구현은 다른 형식들(예를 들어, 장치 또는 프로그램)로 구현될 수도 있다. 장치는, 예를 들어, 적절한 하드웨어, 소프트웨어, 및 펌웨어로 구현될 수 있다. 방법들은, 예를 들어, 컴퓨터, 마이크로프로세서, 집적 회로, 또는 프로그래머블 로직 디바이스를 포함하는, 일반적으로 처리 디바이스들을 언급하는, 예를 들어, 프로세서와 같은, 예를 들어, 장치에서 구현될 수 있다. 프로세서들은 또한, 예를 들어, 컴퓨터들, 셀 전화기들, PDA(portable/personal digital assistant)들, 및 최종 사용자들 간의 정보의 통신을 용이하게 하는 다른 디바이스들과 같은 통신 디바이스들을 포함한다.

[0181] 본 실시예들의 양태에 따르면, 비디오 인코딩 및/또는 디코딩을 위한 장치(2100)가 제공되는데, 장치는 프로세서(2110), 및 프로세서에 결합된 적어도 하나의 메모리(2120, 2140)를 포함하고, 프로세서(2110)는 위에서 설명된 방법(800, 900, 1600 및/또는 1700)의 실시예들 중 임의의 것을 수행하도록 구성된다.

[0182] 본 개시내용의 양태에 따르면, 디코딩 데이터가 비디오 비트스트림에 명시적으로 코딩되어 있는지 또는 비디오 비트스트림의 이전 데이터로부터 추론되는지를 지시하는 구문 요소를 이용하는 수단을 포함하는 비디오 인코딩 및/또는 디코딩을 위한 장치가 제공된다. 도 1의 비디오 인코더는 장치의 구조 또는 수단을 포함할 수 있다. 비디오 인코딩을 위한 장치는 방법들(800, 900, 1600 및 1700) 중 임의의 것의 실시예들 중 임의의 것을 수행할 수 있다.

[0183] 본 기술분야의 통상의 기술자에게 명백하겠지만, 구현들은, 예를 들어, 저장 또는 송신될 수 있는 정보를 운반하도록 포맷팅된 다양한 신호들을 생성할 수 있다. 정보는, 예를 들어, 방법을 수행하기 위한 명령어들, 또는 설명된 구현들 중 하나에 의해 생성된 데이터를 포함할 수 있다. 예를 들어, 신호는 설명된 실시예의 비트스트림을 운반하도록 포맷팅될 수 있다. 그러한 신호는, 예를 들어, 전자기파로서(예를 들어, 스펙트럼의 무선 주파수 부분을 사용하여) 또는 기저대역 신호로서 포맷팅될 수 있다. 포맷팅은, 예를 들어, 데이터 스트림을 인코딩하는 것 및 인코딩된 데이터 스트림으로 캐리어를 변조하는 것을 포함할 수 있다. 신호가 운반하는 정보는, 예를 들어, 아날로그 또는 디지털 정보일 수 있다. 신호는, 알려진 바와 같이, 다양한 상이한 유선 또는 무선 링크들을 통해 송신될 수 있다. 신호는 프로세서 판독가능 매체 상에 저장될 수 있다.

[0184] 또한, 방법들(800, 900, 1600 및/또는 1700) 중 임의의 것은 프로세서에 의해 실행될 수 있는 컴퓨터 실행가능 명령어들을 포함하는 컴퓨터 프로그램 제품으로서(독립적으로 또는 공동으로) 구현될 수 있다. 컴퓨터 실행가능 명령어들을 갖는 컴퓨터 프로그램 제품은 시스템(2100), 인코더(100) 및/또는 디코더(200)의 각자의 일시적 또는 비일시적 컴퓨터 판독가능 저장 매체에 저장될 수 있다.

[0185] 프로세스들(800, 900, 1600 및/또는 1700) 내의 요소들 중 하나 이상은 본 개시내용의 양태들을 여전히 구현하면서 일부 실시예들에서 조합되거나, 상이한 순서로 수행되거나, 배제될 수 있다는 점에 유의하는 것이 중요하다. 다른 단계들이 병렬로 수행될 수 있고, 여기서 프로세서는 다른 것을 시작하기 전에 단계의 완전한 완료를 기다리지 않는다.

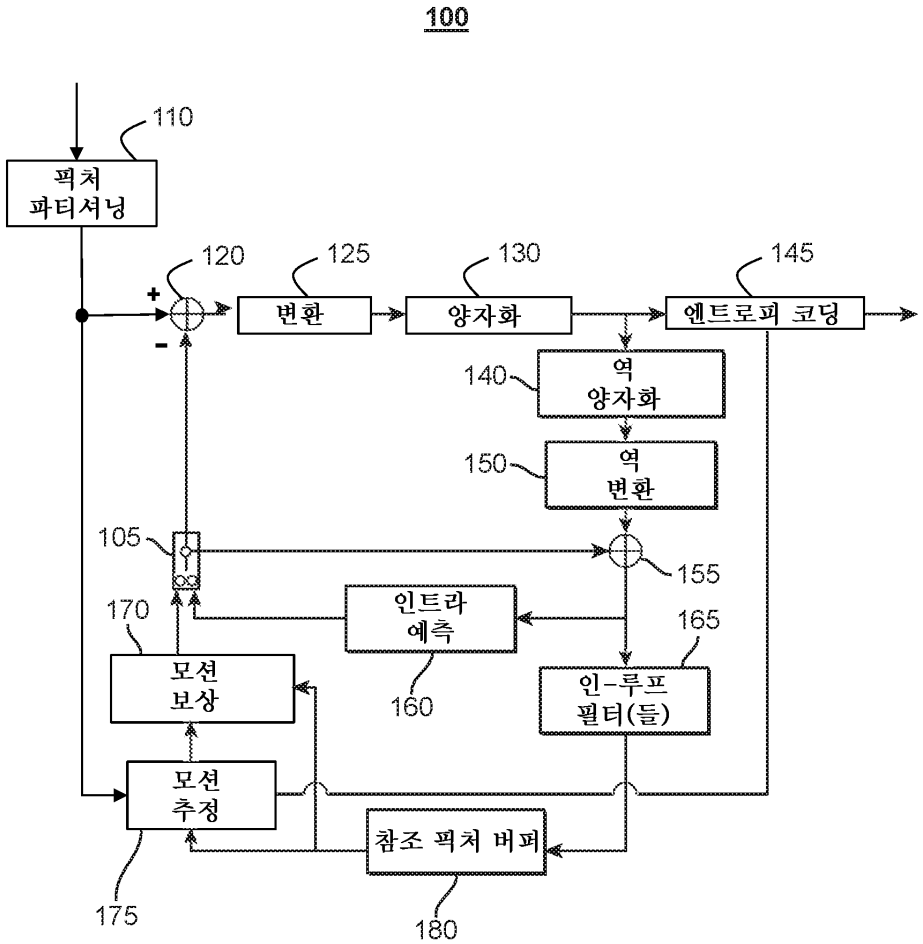
[0186] 더욱이, 본 실시예들의 양태들은 컴퓨터 판독가능 저장 매체의 형식을 취할 수 있다. 하나 이상의 컴퓨터 판독가능 저장 매체(들)의 임의의 조합이 이용될 수 있다. 컴퓨터 판독가능 저장 매체는 하나 이상의 컴퓨터 판독가능 매체(들)에 구현되고 컴퓨터에 의해 실행가능한 컴퓨터 판독가능 프로그램 코드가 구현되어 있는 컴퓨터 판독가능 프로그램 제품의 형식을 취할 수 있다. 본 명세서에서 사용되는 컴퓨터 판독가능 저장 매체는 그 안에 정보를 저장하기 위한 고유 능력 뿐만 아니라 그로부터 정보의 검색을 제공하기 위한 고유 능력이 주어지는 경우에 비일시적 저장 매체로 간주된다. 컴퓨터 판독가능 저장 매체는, 예를 들어, 전자, 자기, 광학, 전자기, 적외선, 또는 반도체 시스템, 장치, 또는 디바이스, 또는 이들의 임의의 적합한 조합일 수 있지만, 이에 제한되지 않는다.

[0187] 이하의 목록은, 본 개시내용이 적용될 수 있는 컴퓨터 판독가능 저장 매체들의 보다 구체적인 예들을 제공하지만, 본 기술분야의 통상의 기술자가 쉽게 알 수 있는 바와 같은 총체적인 것이 아니라 예시적인 목록에 불과하다는 점을 알아야 한다. 예들의 목록은 휴대용 컴퓨터 디스켓, 하드 디스크, ROM, EPROM, 플래시 메모리, 휴대용 CD-ROM(compact disc read-only memory), 광학 저장 디바이스, 자기 저장 디바이스, 또는 이들의 임의의 적합한 조합을 포함한다.

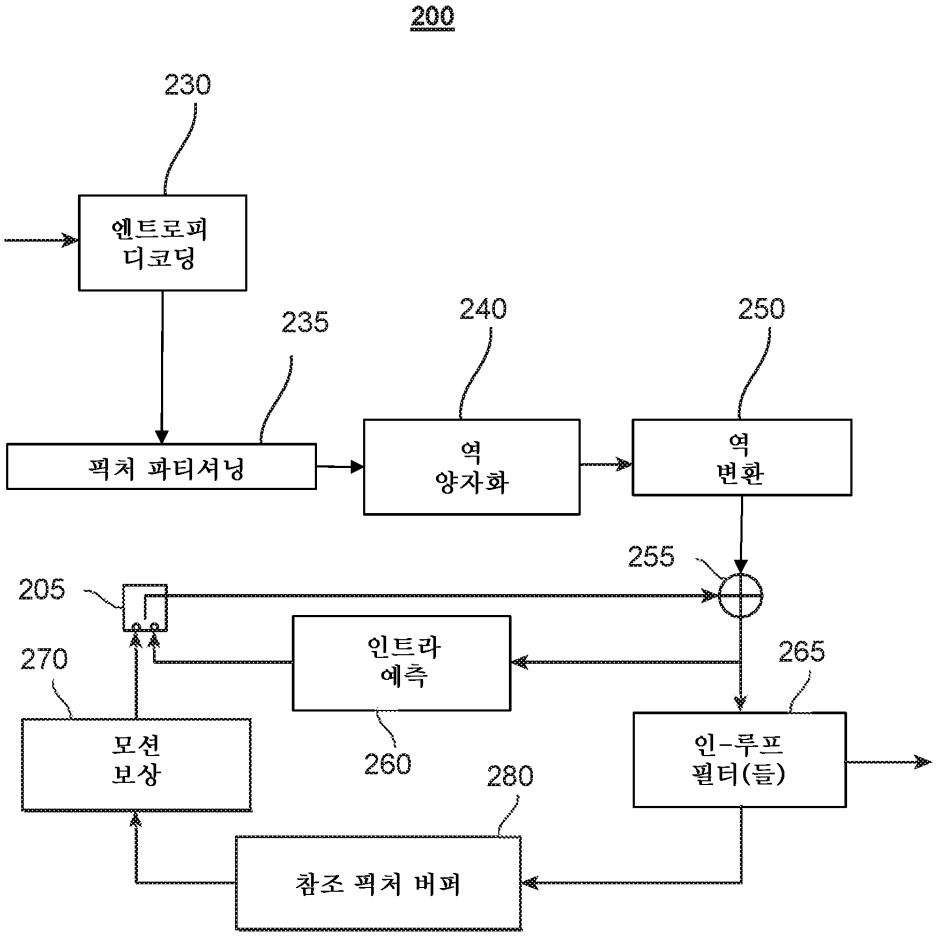
[0188] 본 실시예들의 양태에 따르면, 방법들(800, 900, 1600 및/또는 1700)을 포함하는, 본 실시예들의 방법들 중 임의의 것의 실시예들 중 임의의 것을 수행하기 위한 프로그램 코드 명령어들을 포함하는 소프트웨어 프로그램을 운반하는 컴퓨터 판독가능 저장 매체가 제공된다.

도면

도면1



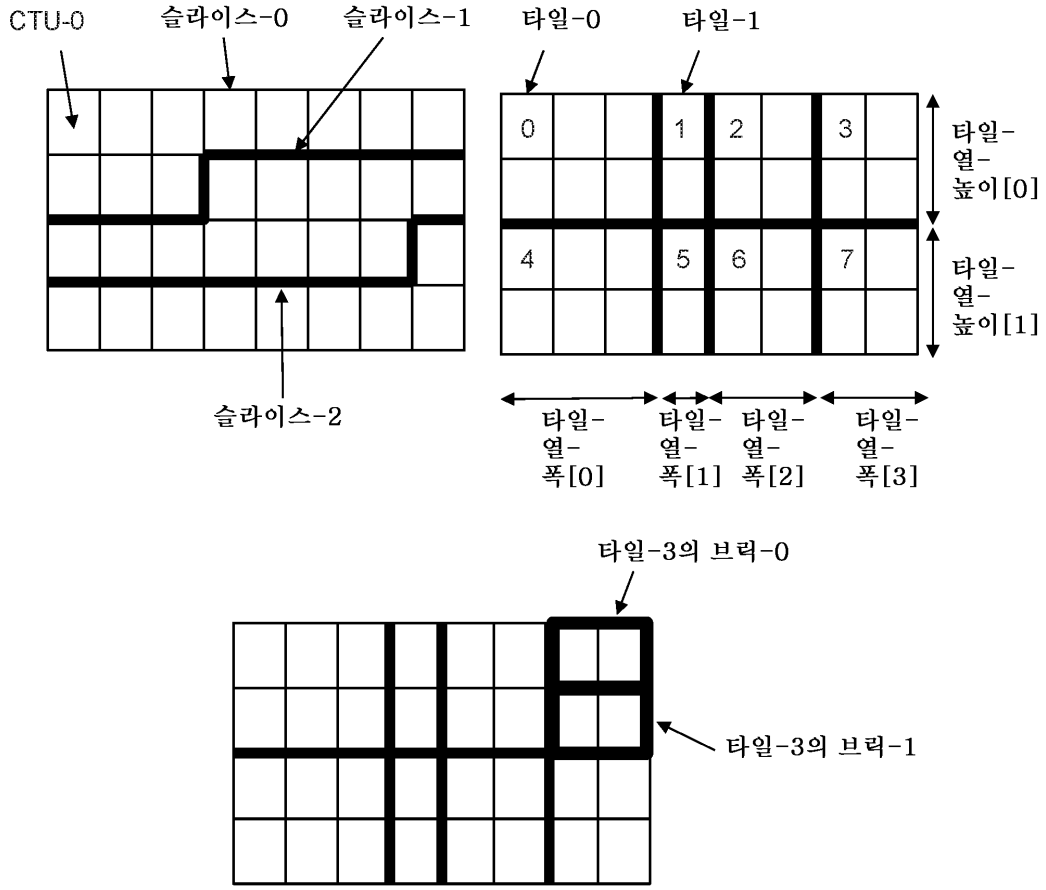
도면2


200
230
엔트로피
디코딩
235
240
250
픽처 파티셔닝
역
양자화
역
변환
255
205
인트라
예측
270
265
모션
보상
260
인-루프
필터(들)
280
참조 픽처 버퍼

도면3

general_constraint_info( ) {	디스크립터
general_progressive_source_flag	u(1)
general_interlaced_source_flag	u(1)
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
frame_only_constraint_flag	u(1)
no_qtbtt_dual_tree_intra constraint_flag	u(1)
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_pcm_constraint_flag	u(1)
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_gbi_constraint_flag	u(1)
no_ibc_constraint_flag	u(1)
no_ciip_constraint_flag	u(1)
no_fpel_mmvd_constraint_flag	u(1)
no_triangle_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_qp_delta_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
// ADD reserved bits for future extensions	
while(!byte_aligned( ))	
gci_alignment_zero_bit	f(1)
}	

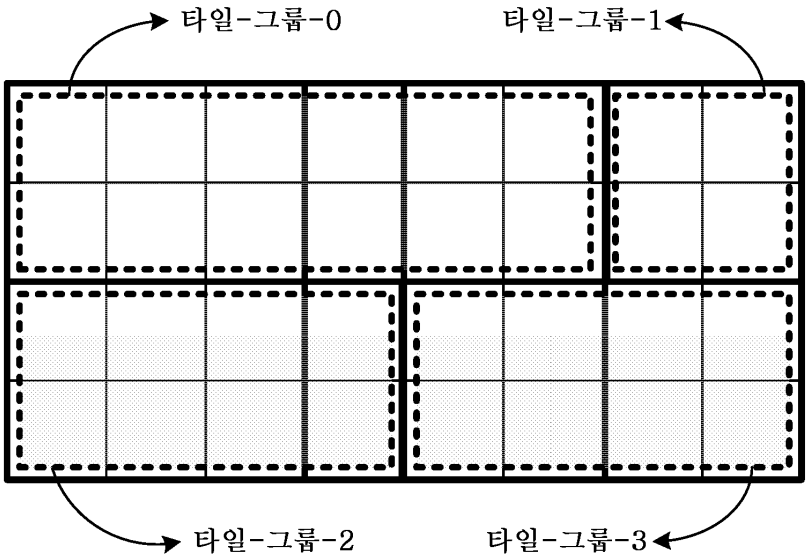
도면4


CTU-0
슬라이스-0
슬라이스-1
슬라이스-2
타일-0
타일-1
0
1
2
3
4
5
6
7
타일-열-높이[0]
타일-열-높이[1]
타일-열-폭[0]
타일-열-폭[1]
타일-열-폭[2]
타일-열-폭[3]
타일-3의 브릭-0
타일-3의 브릭-1

도면5

pic_parameter_set_rbsp( ) {	디스크립터
…	
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_tile_spacing_flag	u(1)
if(!uniform_tile_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
}	
single_tile_per_tile_group_flag	u(1)
if(!single_tile_per_tile_group_flag)	
rect_tile_group_flag	u(1)
if(rect_tile_group_flag && !single_tile_per_tile_group_flag) {	
num_tile_groups_in_pic_minus1	ue(v)
for(i = 0; i <= num_tile_groups_in_pic_minus1; i++) {	
if(i > 0)	
top_left_tile_idx[i]	u(v)
bottom_right_tile_idx_delta[i]	u(v)
}	
}	
…	

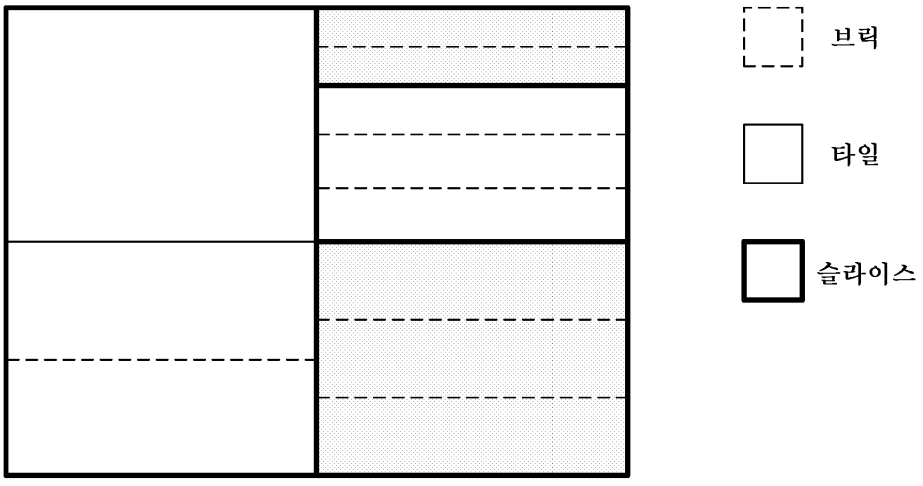
도면6



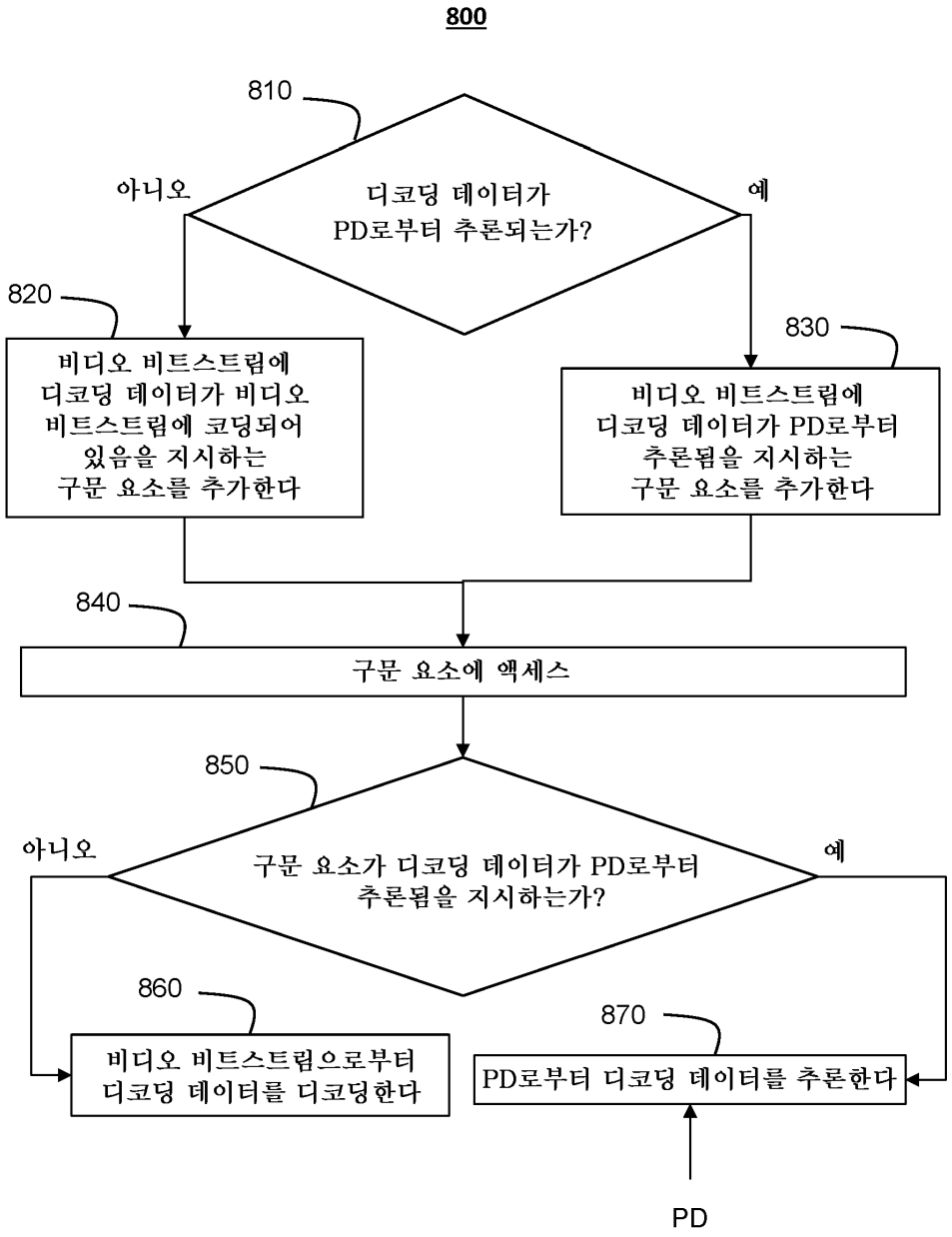
도면7

타일-그룹 인덱스	타일 인덱스 목록	top_left_tile_idx	bottom_right_tile_idx	bottom_right_tile_idx_delta
0	0, 1, 2	0	2	2
1	3	3	3	0
2	4, 5	4	5	1
3	6, 7	6	7	1

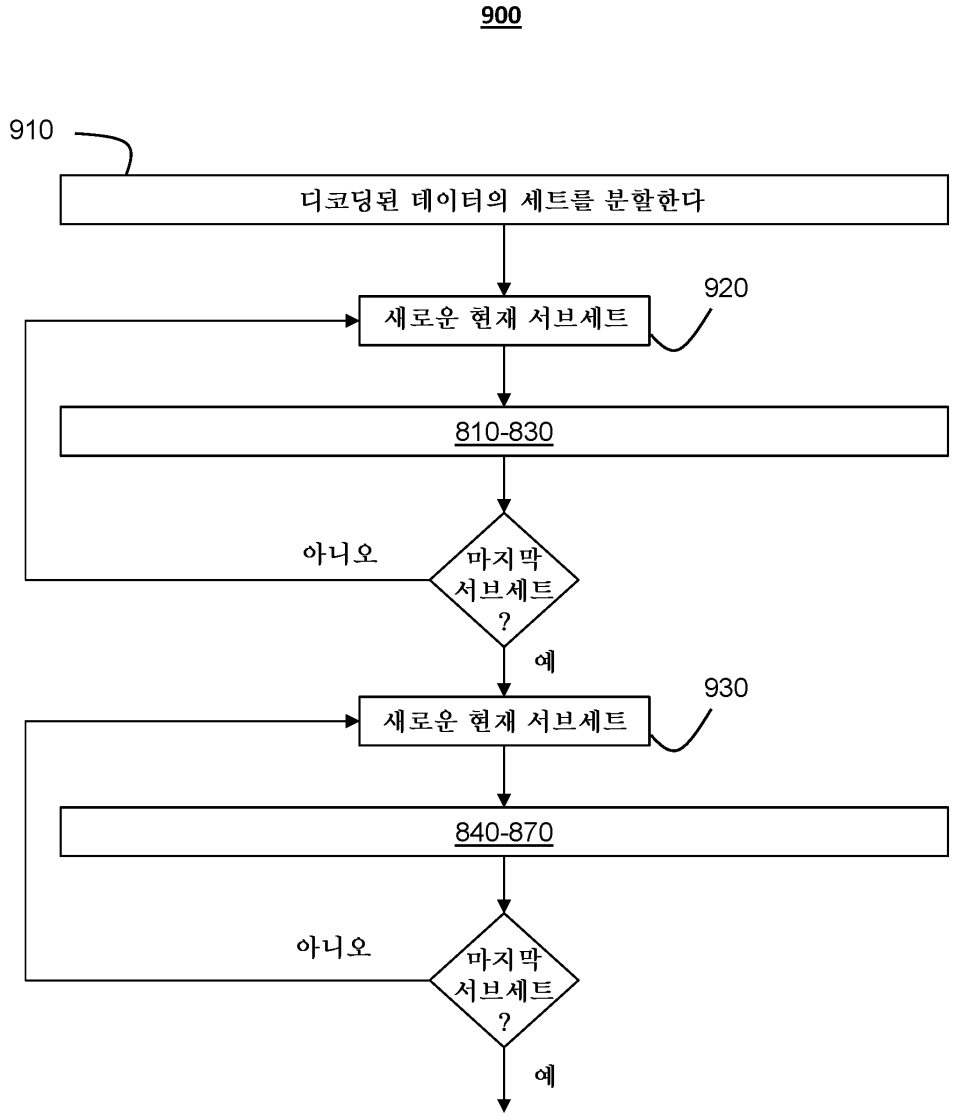
도면7a



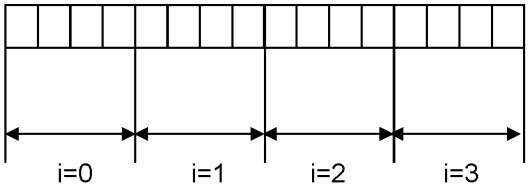
도면8


800
810
디코딩 데이터가
PD로부터 추론되는가?
아니오
예
820
비디오 비트스트림에
디코딩 데이터가 비디오
비트스트림에 코딩되어
있음을 지시하는
구문 요소를 추가한다
830
비디오 비트스트림에
디코딩 데이터가 PD로부터
추론됨을 지시하는
구문 요소를 추가한다
840
구문 요소에 액세스
850
구문 요소가 디코딩 데이터가 PD로부터
추론됨을 지시하는가?
아니오
예
860
비디오 비트스트림으로부터
디코딩 데이터를 디코딩한다
870
PD로부터 디코딩 데이터를 추론한다
PD

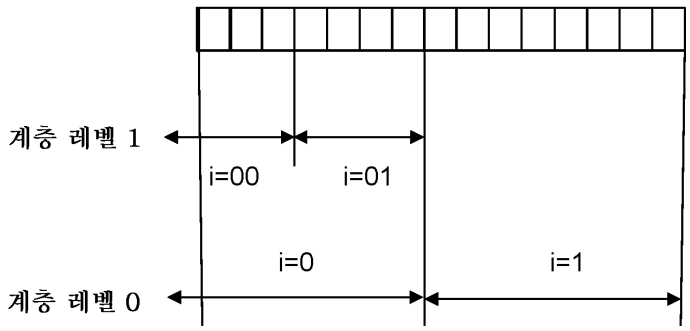
도면9


900
910
디코딩된 데이터의 세트를 분할한다
920
새로운 현재 서브세트
810-830
아니오
마지막 서브세트 ?
예
930
새로운 현재 서브세트
840-870
아니오
마지막 서브세트 ?
예

도면10


i=0
i=1
i=2
i=3

도면11



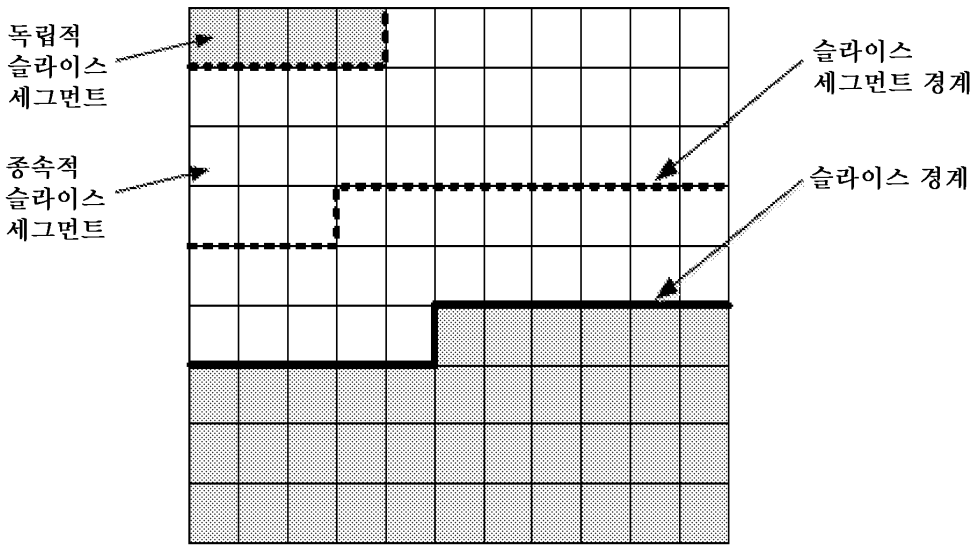
도면12

general_constraint_info( ) {	디스크립터
…	
use_default_constraint_flag	u(1)
if (!use_default_constraint_flag) {	
no_qtbtt_dual_tree_intra constraint_flag	u(1)
no_partition_constraints_override_constrain t_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_pcm_constraint_flag	u(1)
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_gbi_constraint_flag	u(1)
no_ibc_constraint_flag	u(1)
no_ciip_constraint_flag	u(1)
no_fpel_mmvd_constraint_flag	u(1)
no_triangle_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_qp_delta_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
}	
// ADD reserved bits for future extensions	
while(!byte_aligned( ))	
gci_alignment_zero_bit	f(1)
}	

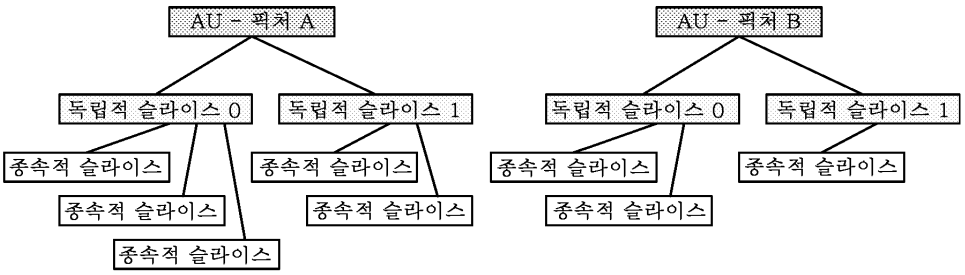
도면13

profile_tier_level(maxNumSubLayersMinus1) {	디스크립터
general_profile_idc	u(7)
general_tier_flag	u(1)
general_sub_profile_idc	u(24)
use_default_constraint_flag	u(1)
if (use-default_constraint_flag)	
general_constraint_info( )	
general_level_idc	u(8)
for(i = 0; i < maxNumSubLayersMinus1; i++)	
sub_layer_level_present_flag[i]	u(1)
while(!byte_aligned( ))	
ptl_alignment_zero_bit	f(1)
for(i = 0; i < maxNumSubLayersMinus1; i++)	
if(sub_layer_level_present_flag[i])	
sub_layer_level_idc[i]	u(8)
}	

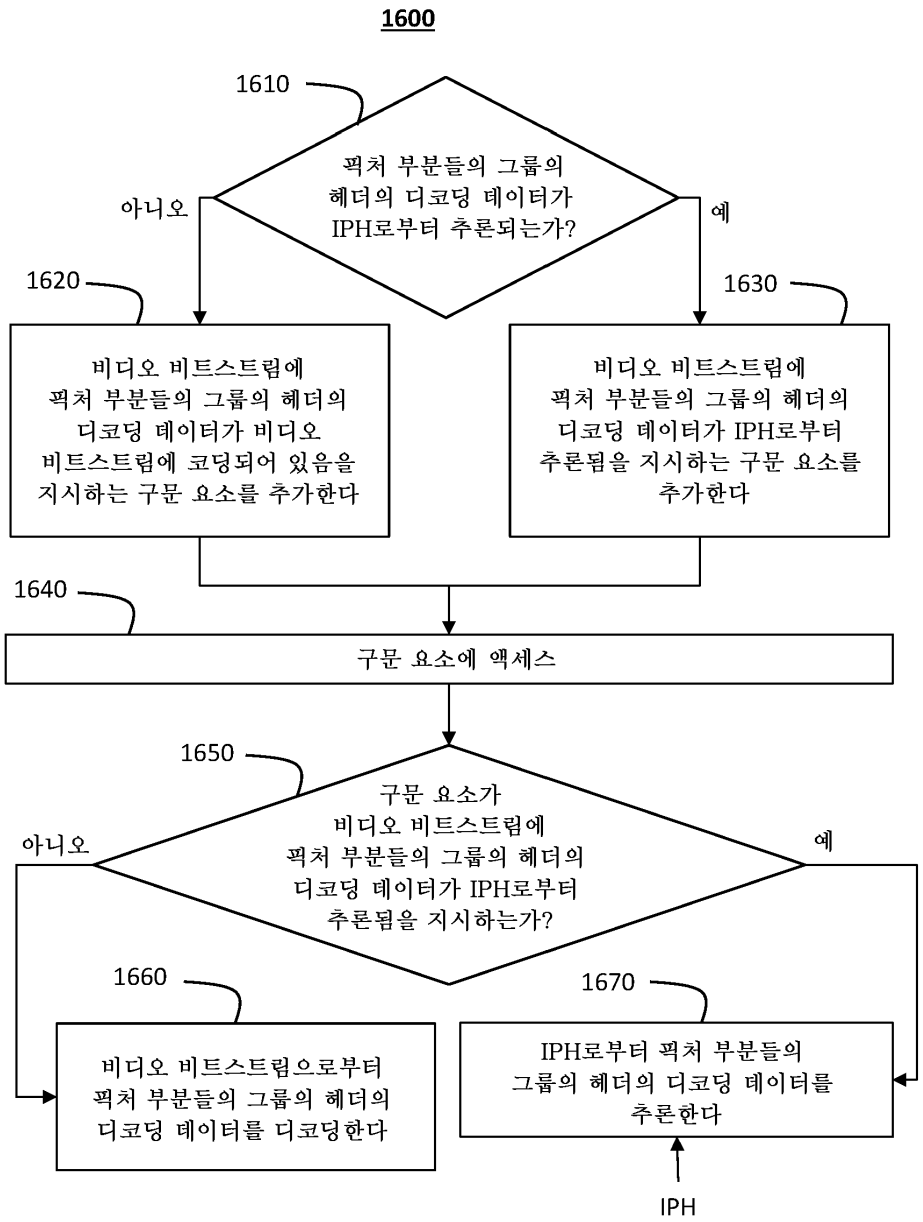
도면14



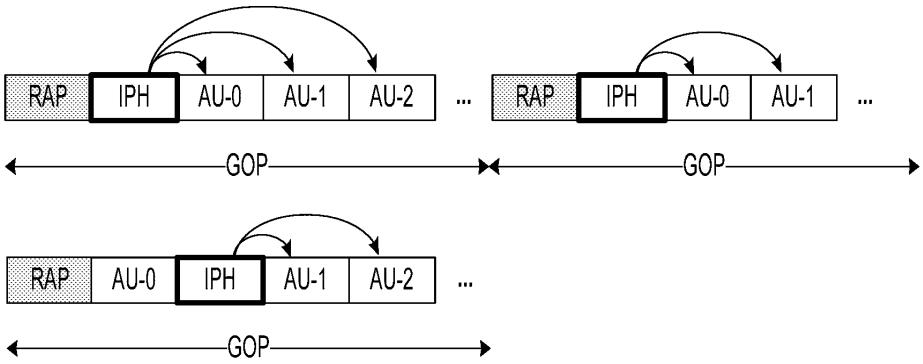
도면15


AU - 픽처 A
독립적 슬라이스 0
독립적 슬라이스 1
종속적 슬라이스
종속적 슬라이스
종속적 슬라이스
종속적 슬라이스
종속적 슬라이스
AU - 픽처 B
독립적 슬라이스 0
독립적 슬라이스 1
종속적 슬라이스
종속적 슬라이스
종속적 슬라이스

도면16


1600
1610
픽처 부분들의 그룹의 헤더의 디코딩 데이터가 IPH로부터 추론되는가?
아니오
예
1620
비디오 비트스트림에 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 비디오 비트스트림에 코딩되어 있음을 지시하는 구문 요소를 추가한다
1630
비디오 비트스트림에 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 IPH로부터 추론됨을 지시하는 구문 요소를 추가한다
1640
구문 요소에 액세스
1650
구문 요소가 비디오 비트스트림에 픽처 부분들의 그룹의 헤더의 디코딩 데이터가 IPH로부터 추론됨을 지시하는가?
아니오
예
1660
비디오 비트스트림으로부터 픽처 부분들의 그룹의 헤더의 디코딩 데이터를 디코딩한다
1670
IPH로부터 픽처 부분들의 그룹의 헤더의 디코딩 데이터를 추론한다
IPH

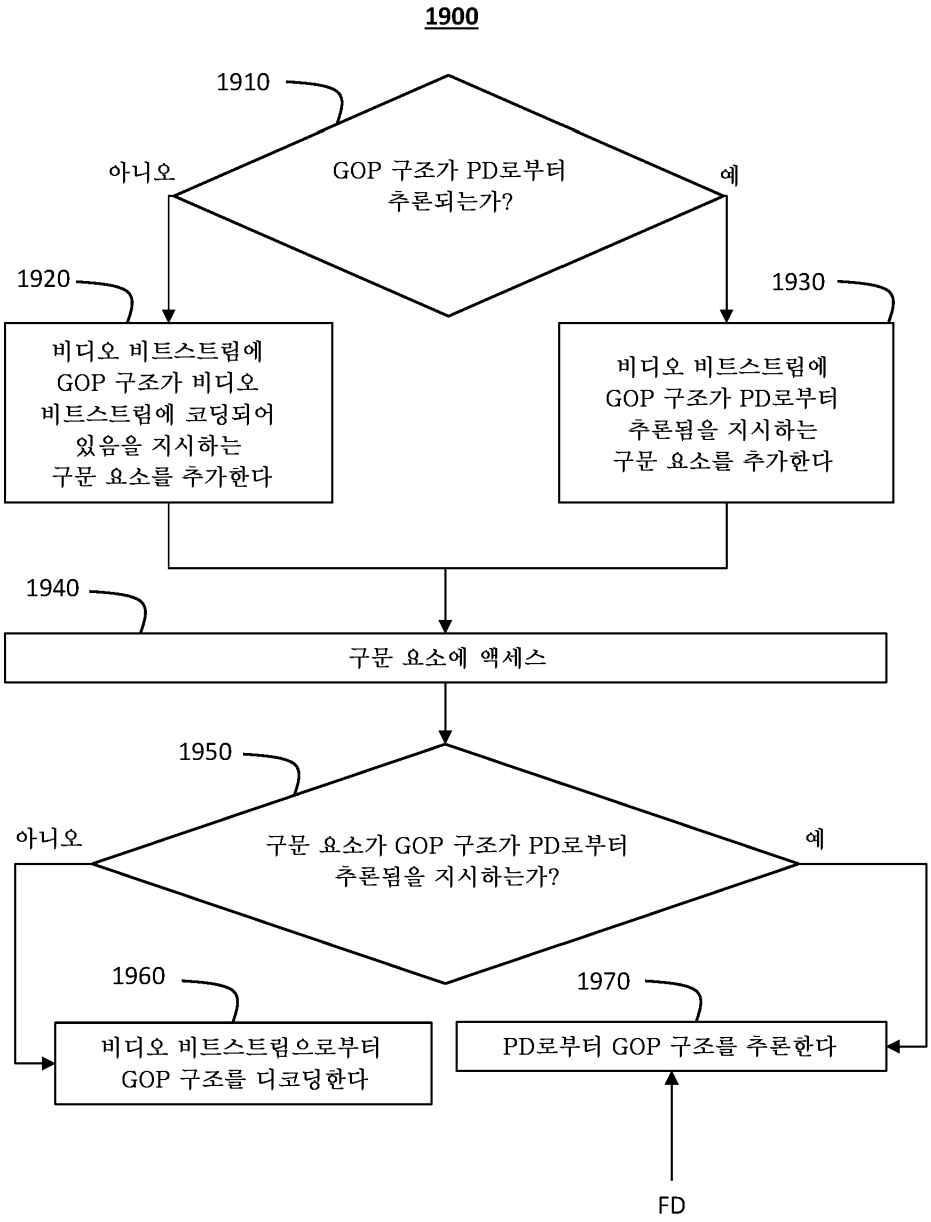
도면17



도면18

pic_parameter_set_rbsp( ) {	디스크립터
…	
single_tile_per_tile_group_flag	u(1)
if(!single_tile_per_tile_group_flag)	
rect_tile_group_flag	u(1)
if(rect_tile_group_flag && !single_tile_per_tile_group_flag) {	
num_tile_groups_in_pic_minus1	ue(v)
for(i = 0; i <= num_tile_groups_in_pic_minus1; i++) {	
if(i > 0)	
top_left_tile_idx[i]	u(v)
if(i < num_tile_groups_in_pic_minus1)	
bottom_right_tile_idx_delta[i]	u(v)
}	
}	
…	

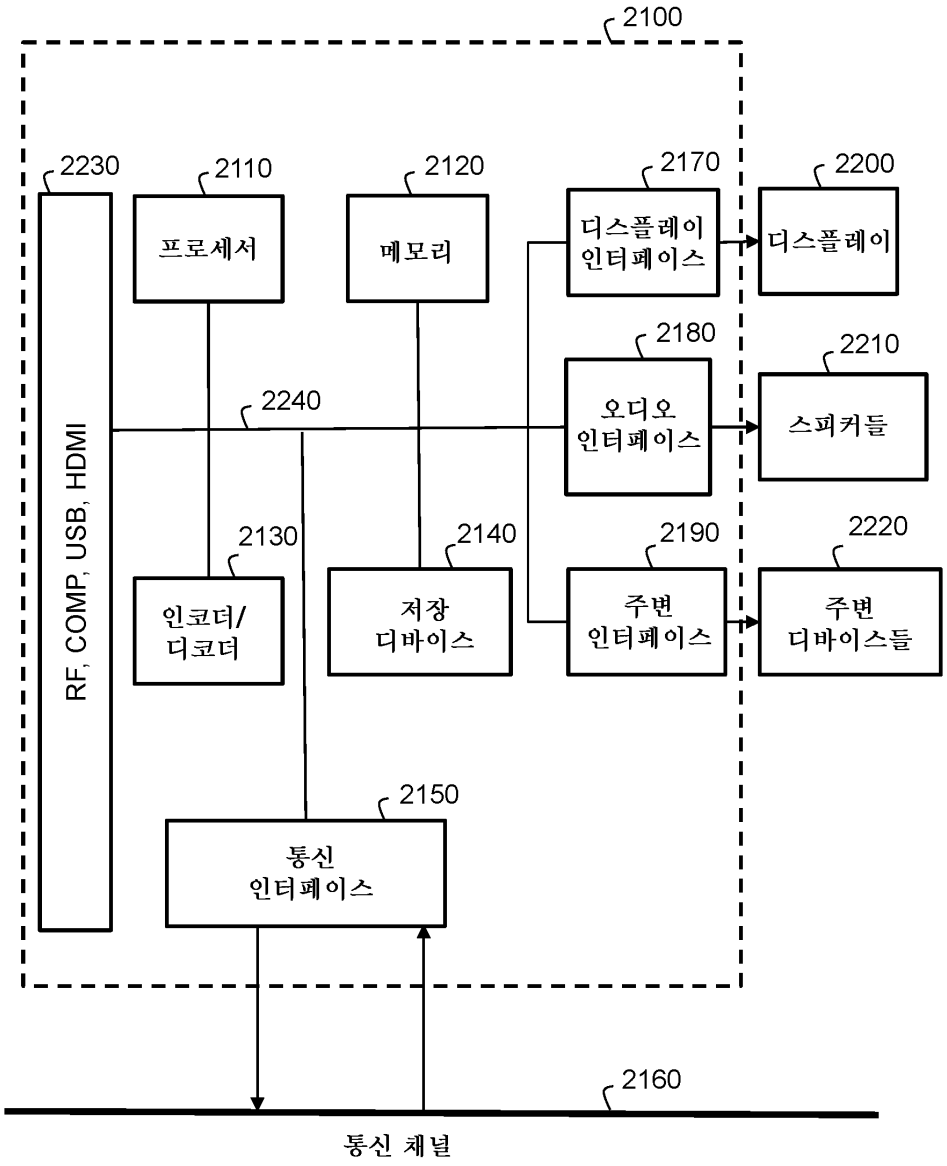
도면19



도면20

이름	GOP 길이	인덱스	GOP 구조
RA	9	0	084213657
RA	16	1	0 16 84213657 12 10 9 11 14 13 15
LB	N	2	0123456... N
LP	N	3	0123456... N

도면21


2100
2230
2110
2120
2170
2200
프로세서
메모리
디스플레이 인터페이스
디스플레이
2180
2210
2240
오디오 인터페이스
스피커들
RF, COMP, USB, HDMI
2130
2140
2190
2220
인코더/ 디코더
저장 디바이스
주변 인터페이스
주변 디바이스들
2150
통신 인터페이스
2160
통신 채널