(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0096114
(43) 공개일자 2021년08월04일

(51) 국제특허분류(Int. Cl.)
H04N 19/13 (2014.01) *G06F 17/18* (2006.01)
H04N 19/176 (2014.01) *H04N 19/18* (2014.01)
H04N 19/44 (2014.01) *H04N 19/46* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/13 (2015.01)
G06F 17/18 (2013.01)
(21) 출원번호 10-2021-7016652
(22) 출원일자(국제) 2019년12월06일
심사청구일자 없음
(85) 번역문제출일자 2021년05월31일
(86) 국제출원번호 PCT/US2019/065002
(87) 국제공개번호 WO 2020/118212
국제공개일자 2020년06월11일
(30) 우선권주장
62/776,379 2018년12월06일 미국(US)
(뒷면에 계속)

(71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
코반 무하메드 제이드
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인코리아나

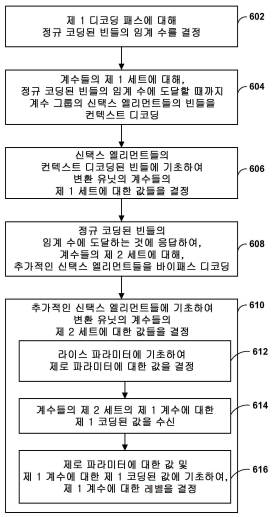
전체 청구항 수 : 총 38 항

(54) 발명의 명칭 **임계치 및 라이스 파라미터를 이용한 계수 디코딩을 위한 정규 코딩된 빈 감축**

(57) 요 약

비디오 코더는, 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하고, 그 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하며; 계수들의 제 2 세트의 제 1 계수에 대한 제 1 코딩된 값을 수신하고; 그리고, 상기 제로 파라미터에 대한 값 및 상기 제 1 계수에 대한 제 1 코딩된 값에 기초하여, 제 1 계수에 대한 레벨을 결정하도록 구성될 수도 있다.

대 표 도 - 도17



(52) CPC특허분류
H04N 19/176 (2015.01)
H04N 19/18 (2015.01)
H04N 19/44 (2015.01)
H04N 19/46 (2015.01)
H04N 19/70 (2015.01)

(30) 우선권주장
62/787,681 2019년01월02일 미국(US)
16/704,995 2019년12월05일 미국(US)

명 세 서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 방법은,

제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정하는 단계;

계수들의 제 1 세트에 대해, 상기 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 단계로서, 컨텍스트 디코딩된 상기 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 상기 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 상기 하나 이상의 패리티 레벨 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 짝수 또는 홀수인지를 표시하고, 상기 하나 이상의 제 1 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 단계;

상기 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 상기 제 1 세트에 대한 값들을 결정하는 단계;

상기 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 단계로서, 상기 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것은, 상기 계수들의 제 2 세트의 계수에 대해, 라이스 (Rice) 파라미터에 대한 값을 도출하는 것을 포함하는, 상기 바이패스 디코딩하는 단계; 및

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하는 단계를 포함하고,

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하는 단계는:

상기 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 단계로서, 상기 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 단계;

상기 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하는 단계; 및

상기 제로 파라미터에 대한 값 및 상기 제 1 계수에 대한 상기 제 1 코딩된 값에 기초하여, 상기 제 1 계수에 대한 레벨을 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 계수에 대한 레벨은 잔여 레벨을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 1 계수에 대한 레벨은 절대 레벨을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 라이스 파라미터에 기초하여 상기 제로 파라미터에 대한 값을 결정하는 단계는, 상기 라이스 파라미터에 기초하여 그리고 상태 머신의 현재 상태에 기초하여 상기 제로 파라미터에 대한 값을 결정하는 단계를

포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 제로 파라미터에 대한 값이 제 1 코딩된 값과 동일한 것에 응답하여, 상기 제 1 계수에 대한 레벨이 제로와 동일한 것을 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 제 1 코딩된 값이 상기 제로 파라미터에 대한 값보다 큰 것에 응답하여, 상기 제 1 계수에 대한 레벨이 상기 제 1 코딩된 값과 동일한 것을 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 제 1 코딩된 값이 상기 제로 파라미터에 대한 값보다 작은 것에 응답하여, 상기 제 1 계수에 대한 레벨이 상기 제 1 코딩된 값 플러스 1 과 동일한 것을 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

룩업 테이블로부터 상기 라이스 파라미터에 대한 값을 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 계수 그룹의 상기 신택스 엘리먼트들을 컨텍스트 디코딩하는 단계는, 상기 계수 그룹의 상기 신택스 엘리먼트들을 디코딩하기 위해 컨텍스트-적응 이진 산술 디코딩을 수행하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 10

제 1 항에 있어서,

상기 정규 코딩된 빈들의 임계 수에 도달할 때까지 상기 계수 그룹의 신택스 엘리먼트들을 컨텍스트 디코딩하는 단계는:

상기 계수들의 제 1 세트의 계수에 대한 신택스 엘리먼트를 코딩하는 동안 상기 정규 코딩된 빈들의 임계 수에 도달한 것을 결정하는 단계;

상기 계수들의 제 1 세트의 계수에 대한 하나 이상의 잔여 신택스 엘리먼트들을 컨텍스트 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 11

제 1 항에 있어서,

상기 계수들의 제 1 세트에 대한 값들 및 상기 계수들의 제 2 세트에 대한 값들에 기초하여 디코딩된 변환 블록을 결정하는 단계;

디코딩된 상기 변환 블록을 예측 블록에 가산하여 재구성된 블록을 결정하는 단계;

비디오 데이터의 디코딩된 블록을 결정하기 위해 상기 재구성된 블록에 대해 하나 이상의 필터링 동작들을 수행하는 단계; 및

비디오 데이터의 상기 디코딩된 블록을 포함하는 비디오 데이터의 디코딩된 픽처를 출력하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 12

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 디바이스는,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

회로에서 구현되는 하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은:

제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정하고;

계수들의 제 1 세트에 대해, 상기 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 것으로서, 컨텍스트 디코딩된 상기 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 상기 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 상기 하나 이상의 패리티 레벨 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 짝수 또는 홀수인지를 표시하고, 상기 하나 이상의 제 1 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 것을 행하고;

상기 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 상기 제 1 세트에 대한 값들을 결정하며;

상기 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것으로서, 상기 추가적인 신택스 엘리먼트들을 바이패스 디코딩하기 위해, 상기 하나 이상의 프로세서들은, 상기 계수들의 제 2 세트의 계수에 대해, 라이스 (Rice) 파라미터에 대한 값을 도출하도록 구성되는, 상기 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것을 행하고; 그리고

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하도록 구성되고,

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하기 위해, 상기 하나 이상의 프로세서들은:

상기 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 것으로서, 상기 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 것을 행하고;

상기 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하며;

상기 제로 파라미터에 대한 값 및 상기 제 1 계수에 대한 상기 제 1 코딩된 값에 기초하여, 상기 제 1 계수에 대한 레벨을 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 13

제 12 항에 있어서,

상기 제 1 계수에 대한 레벨은 잔여 레벨을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 14

제 12 항에 있어서,

상기 제 1 계수에 대한 레벨은 절대 레벨을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 15

제 12 항에 있어서,

상기 라이스 파라미터에 기초하여 상기 제로 파라미터에 대한 값을 결정하기 위해, 상기 하나 이상의 프로세서들은, 상기 라이스 파라미터에 기초하여 그리고 상태 머신의 현재 상태에 기초하여 상기 제로 파라미터에 대한 값을 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 16

제 12 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제로 파라미터에 대한 값이 제 1 코딩된 값과 동일한 것에 응답하여, 상기 제 1 계수에 대한 레벨이 제로와 동일한 것을 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 17

제 12 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제 1 코딩된 값이 상기 제로 파라미터에 대한 값보다 큰 것에 응답하여, 상기 제 1 계수에 대한 레벨이 상기 제 1 코딩된 값과 동일한 것을 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 18

제 12 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제 1 코딩된 값이 상기 제로 파라미터에 대한 값보다 작은 것에 응답하여, 상기 제 1 계수에 대한 레벨이 상기 제 1 코딩된 값 플러스 1 과 동일한 것을 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 19

제 12 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

룩업 테이블로부터 상기 라이스 파라미터에 대한 값을 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 20

제 12 항에 있어서,

상기 계수 그룹의 상기 신택스 엘리먼트들을 컨텍스트 디코딩하기 위해, 상기 하나 이상의 프로세서들은, 상기 계수 그룹의 상기 신택스 엘리먼트들을 디코딩하기 위해 컨텍스트-적응 이진 산술 디코딩을 수행하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 21

제 12 항에 있어서,

상기 정규 코딩된 빈들의 임계 수에 도달할 때까지 상기 계수 그룹의 신택스 엘리먼트들을 컨텍스트 디코딩하기 위해, 상기 하나 이상의 프로세서들은:

상기 계수들의 제 1 세트의 계수에 대한 신택스 엘리먼트를 코딩하는 동안 상기 정규 코딩된 빈들의 임계 수에 도달한 것을 결정하고;

상기 계수들의 제 1 세트의 계수에 대한 하나 이상의 잔여 신택스 엘리먼트들을 컨텍스트 디코딩하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 22

제 12 항에 있어서,

상기 하나 이상의 프로세서들은 또한:

상기 계수들의 제 1 세트에 대한 값들 및 상기 계수들의 제 2 세트에 대한 값들에 기초하여 디코딩된 변환 블록을 결정하고;

디코딩된 상기 변환 블록을 예측 블록에 가산하여 재구성된 블록을 결정하며;

비디오 데이터의 디코딩된 블록을 결정하기 위해 상기 재구성된 블록에 대해 하나 이상의 필터링 동작들을 수행하고; 그리고

비디오 데이터의 상기 디코딩된 블록을 포함하는 비디오 데이터의 디코딩된 픽처를 출력하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 23

제 12 항에 있어서,

상기 디바이스는 무선 통신 디바이스를 포함하고, 인코딩된 비디오 데이터를 수신하도록 구성된 수신기를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 24

제 23 항에 있어서,

상기 무선 통신 디바이스는 전화기 핸드셋을 포함하고, 상기 수신기는, 상기 인코딩된 비디오 데이터를 포함하는 신호를 무선 통신 표준에 따라 복조하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 25

제 12 항에 있어서,

디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 26

제 12 항에 있어서,

상기 디바이스는, 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스, 또는 셋톱 박스 중 하나 이상을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 27

명령들을 저장하는 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 때, 상기 하나 이상의 프로세서들로 하여금:

제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정하게 하고;

계수들의 제 1 세트에 대해, 상기 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 것으로서, 컨텍스트 디코딩된 상기 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 상기 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 상기 하나 이상의 패리티 레벨 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 짝수 또는 홀수인지를 표시하고, 상기 하나 이상의 제 1 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 것을 행하게 하고;

상기 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 상기 제 1 세트에 대한 값들을 결정하게 하며;

상기 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것으로서, 상기 추가적인 신택스 엘리먼트들을 바이패스 디코딩하기 위해, 상

기 명령들은 상기 하나 이상의 프로세서들로 하여금, 상기 계수들의 제 2 세트의 계수에 대해, 라이스 (Rice) 파라미터에 대한 값을 도출하게 하는, 상기 바이패스 디코딩하는 것을 행하게 하고; 그리고

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하게 하며,

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하기 위해, 상기 명령들은 상기 하나 이상의 프로세서들로 하여금:

상기 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 것으로서, 상기 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 것을 행하게 하고;

상기 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하게 하며; 그리고

상기 제로 파라미터에 대한 값 및 상기 제 1 계수에 대한 상기 제 1 코딩된 값에 기초하여, 상기 제 1 계수에 대한 레벨을 결정하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 28

제 27 항에 있어서,

상기 제 1 계수에 대한 레벨은 잔여 레벨을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 29

제 27 항에 있어서,

상기 제 1 계수에 대한 레벨은 절대 레벨을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 30

제 27 항에 있어서,

상기 라이스 파라미터에 기초하여 상기 제로 파라미터에 대한 값을 결정하기 위해, 상기 명령들은 상기 하나 이상의 프로세서들로 하여금, 상기 라이스 파라미터에 기초하여 그리고 상태 머신의 현재 상태에 기초하여 상기 제로 파라미터에 대한 값을 결정하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 31

제 27 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금,

상기 제로 파라미터에 대한 값이 제 1 코딩된 값과 동일한 것에 응답하여, 상기 제 1 계수에 대한 레벨이 제로와 동일한 것을 결정하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 32

제 27 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금,

상기 제 1 코딩된 값이 상기 제로 파라미터에 대한 값보다 큰 것에 응답하여, 상기 제 1 계수에 대한 레벨이 상기 제 1 코딩된 값과 동일한 것을 결정하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 33

제 27 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금,

상기 제 1 코딩된 값이 상기 제로 파라미터에 대한 값보다 작은 것에 응답하여, 상기 제 1 계수에 대한 레벨이 상기 제 1 코딩된 값 플러스 1 과 동일한 것을 결정하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 34

제 27 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금,

룩업 테이블로부터 상기 라이스 파라미터에 대한 값을 결정하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 35

제 27 항에 있어서,

상기 계수 그룹의 상기 신택스 엘리먼트들을 컨텍스트 디코딩하기 위해, 상기 명령들은 상기 하나 이상의 프로세서들로 하여금, 상기 계수 그룹의 상기 신택스 엘리먼트들을 디코딩하기 위해 컨텍스트-적응 이진 산술 디코딩을 수행하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 36

제 27 항에 있어서,

상기 정규 코딩된 빈들의 임계 수에 도달할 때까지 상기 계수 그룹의 신택스 엘리먼트들을 컨텍스트 디코딩하기 위해, 상기 명령들은 상기 하나 이상의 프로세서들로 하여금:

상기 계수들의 제 1 세트의 계수에 대한 신택스 엘리먼트를 코딩하는 동안 상기 정규 코딩된 빈들의 임계 수에 도달한 것을 결정하게 하고;

상기 계수들의 제 1 세트의 계수에 대한 하나 이상의 잔여 신택스 엘리먼트들을 컨텍스트 디코딩하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 37

제 27 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금:

상기 계수들의 제 1 세트에 대한 값들 및 상기 계수들의 제 2 세트에 대한 값들에 기초하여 디코딩된 변환 블록을 결정하게 하고;

디코딩된 상기 변환 블록을 예측 블록에 가산하여 재구성된 블록을 결정하게 하며;

비디오 데이터의 디코딩된 블록을 결정하기 위해 상기 재구성된 블록에 대해 하나 이상의 필터링 동작들을 수행하게 하고; 그리고

비디오 데이터의 상기 디코딩된 블록을 포함하는 비디오 데이터의 디코딩된 픽처를 출력하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 38

비디오 데이터를 디코딩하기 위한 장치로서,

상기 장치는,

제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정하는 수단;

계수들의 제 1 세트에 대해, 상기 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 수단으로서, 컨텍스트 디코딩된 상기 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 상기 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 상기 하나 이상의 패리티 레벨 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 짝수 또는 홀수인지를 표시하고, 상기 하나 이상의 제 1 플래그들의 각각은 상기 대응하는 계수에 대한 상기 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 수단;

상기 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 상기 제 1 세트에 대한 값

들을 결정하는 수단;

상기 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 수단으로서, 상기 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 수단은, 상기 계수들의 제 2 세트의 계수에 대해, 라이스 (Rice) 파라미터에 대한 값을 도출하는 수단을 포함하는, 상기 바이패스 디코딩하는 수단; 및

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하는 수단을 포함하고,

상기 추가적인 신택스 엘리먼트들에 기초하여 상기 변환 유닛의 계수들의 상기 제 2 세트에 대한 값들을 결정하는 수단은:

상기 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 수단으로서, 상기 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 수단;

상기 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하는 수단; 및

상기 제로 파라미터에 대한 값 및 상기 제 1 계수에 대한 상기 제 1 코딩된 값에 기초하여, 상기 제 1 계수에 대한 레벨을 결정하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

발명의 설명

기 술 분 야

[0001] 이 출원은:

[0002] 2018년 12월 6일자로 출원된 미국 가 특허 출원 제 62/776,379 호; 및

[0003] 2019년 1월 2일자로 출원된 미국 가 특허 출원 제 62/787,681 호의 이익을 주장하는,

[0004] 2019년 12월 5일자로 출원된 미국 특허 출원 제 16/704,995 호에 대해 우선권을 주장하고,

[0005] 이들 각각의 전체 내용은 참조에 의해 본원에 통합된다.

[0006] **기술 분야**

[0007] 본 개시는 비디오 인코딩 및 비디오 디코딩에 관한 것이다.

배 경 기 술

[0008] **배경**

[0009] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인용 디지털 보조기들 (PDA들), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰들", 비디오 텔레컨퍼런싱 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC), 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, ITU-T H.265/고효율 비디오 코딩 (HEVC), 및 이러한 표준들의 확장들에 의해 정의된 표준들에 기술된 것들과 같은, 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 그러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0010] 비디오 코딩 기법들은 비디오 시퀀스들에 내재한 리던던시를 감소 또는 제거하기 위해 공간 (인트라 픽처) 예측 및/또는 시간 (인터 픽처) 예측을 포함한다. 블록 기반 비디오 코딩에 대해, 비디오 슬라이스 (즉, 비디오 픽처 또는 비디오 픽처의 일부) 는 비디오 블록들로 파티셔닝될 수도 있으며, 이 비디오 블록들은 또한 코딩 트리 유닛들 (CTU들), 코딩 유닛들 (CU들) 및/또는 코딩 노드들로서 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처에 있어서 이웃하는 블록들에서의 레퍼런스 샘플들에 대한 공간적 예측을 사용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일 픽처의 이웃하는 블록들에서의 레퍼런스 샘플들에 대한 공간 예측, 또는 다른 레퍼런스 픽처들에서의 레퍼런스

샘플들에 대한 시간 예측을 이용할 수도 있다. 픽처들은 프레임들로 지칭될 수도 있고, 레퍼런스 픽처들은 레퍼런스 프레임들로 지칭될 수도 있다.

발명의 내용

[0011] **요약**

[0012] 비디오 코딩 (예컨대, 비디오 인코딩 및/또는 비디오 디코딩) 은 통상적으로 동일한 픽처에서 이미 코딩된 비디오 데이터의 블록 (예를 들어, 인트라 예측) 또는 상이한 픽처에서 이미 코딩된 비디오 데이터의 블록 (예를 들어, 인터 예측) 으로부터의 비디오 데이터의 블록을 예측하는 것을 수반한다. 일부 경우들에서, 비디오 인코더는 또한 예측 블록을 원래 블록과 비교함으로써 잔차 데이터를 계산한다. 따라서, 잔차 데이터 (residual data) 는 비디오 데이터의 예측 블록과 원래 블록 사이의 차이를 나타낸다. 잔차 데이터를 시그널링하는데 필요한 비트들의 수를 감소시키기 위해, 비디오 인코더는 잔차 데이터를 변환 계수들로 변환하고, 변환 계수들을 양자화하고, 그리고 변환된 및 양자화된 계수들을 인코딩된 비트스트림에서 시그널링한다. 변환 및 양자화 프로세스들에 의해 달성되는 압축은 손실성일 수도 있으며, 이는 변환 및 양자화 프로세스들이 디코딩된 비디오 데이터에 왜곡을 도입할 수도 있다는 것을 의미한다. 본 개시는 변환 계수 코딩에 관련된 기법들을 기술한다.

[0013] 비디오 데이터를 디코딩하는 방법은, 제 1 디코딩 패스에 대한 정규 코딩된 빈 (bin) 들의 임계 수를 결정하는 단계; 계수들의 제 1 세트에 대해, 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 단계로서, 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들 (significance flags), 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 하나 이상의 패리티 레벨 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 짝수 또는 홀수인지를 표시하고, 하나 이상의 제 1 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 단계; 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 제 1 세트에 대한 값들을 결정하는 단계; 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 단계로서, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것은, 계수들의 제 2 세트의 계수에 대해, 라이스 (Rice) 파라미터에 대한 값을 도출하는 것을 포함하는, 상기 바이패스 디코딩하는 단계; 및, 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하는 단계를 포함하고, 상기 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하는 단계는, 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 단계로서, 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 단계; 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하는 단계; 및, 제로 파라미터에 대한 값 및 제 1 계수에 대한 제 1 코딩된 값에 기초하여, 제 1 계수에 대한 레벨을 결정하는 단계를 포함한다.

[0014] 비디오 데이터를 디코딩하기 위한 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리; 및, 회로에서 구현되는 하나 이상의 프로세서들을 포함하며, 상기 하나 이상의 프로세서들은, 제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정하고; 계수들의 제 1 세트에 대해, 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 것으로서, 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 하나 이상의 패리티 레벨 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 짝수 또는 홀수인지를 표시하고, 하나 이상의 제 1 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 것을 행하고; 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 제 1 세트에 대한 값들을 결정하며; 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것으로서, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하기 위해, 하나 이상의 프로세서들은, 계수들의 제 2 세트의 계수에 대해, 라이스 파라미터에 대한 값을 도출하도록 구성되는, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것을 행하고; 그리고, 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하도록 구성되고, 상기 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하기 위해, 상기 하나 이상의 프로세서들은, 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 것으로서,

제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 것을 행하고; 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하며; 제로 파라미터에 대한 값 및 제 1 계수에 대한 제 1 코딩된 값에 기초하여, 제 1 계수에 대한 레벨을 결정하도록 구성된다.

[0015] 하나 이상의 예들에 따르면, 컴퓨터 판독가능 저장 매체는 명령들을 저장하고, 그 명령들은, 하나 이상의 프로세서들에 의해 실행될 때, 그 하나 이상의 프로세서들로 하여금, 제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정하게 하고; 계수들의 제 1 세트에 대해, 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 것으로서, 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 하나 이상의 패리티 레벨 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 짝수 또는 홀수인지를 표시하고, 하나 이상의 제 1 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 것을 행하게 하고; 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 제 1 세트에 대한 값들을 결정하게 하며; 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 것으로서, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하기 위해, 명령들은 하나 이상의 프로세서들로 하여금, 계수들의 제 2 세트의 계수에 대해, 라이스 파라미터에 대한 값을 도출하게 하는, 상기 바이패스 디코딩하는 것을 행하게 하고; 그리고, 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하게 하며, 상기 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하기 위해, 명령들은 하나 이상의 프로세서들로 하여금, 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 것으로서, 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 것을 행하게 하고; 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하게 하며; 그리고, 제로 파라미터에 대한 값 및 제 1 계수에 대한 제 1 코딩된 값에 기초하여, 제 1 계수에 대한 레벨을 결정하게 한다.

[0016] 하나의 예에 따르면, 비디오 데이터를 디코딩하기 위한 장치는, 제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정하는 수단; 계수들의 제 1 세트에 대해, 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들의 빈들을 컨텍스트 디코딩하는 수단으로서, 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들은 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함하고, 하나 이상의 유의성 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 제로와 동일한지를 표시하고, 하나 이상의 패리티 레벨 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 짝수 또는 홀수인지를 표시하고, 하나 이상의 제 1 플래그들의 각각은 대응하는 계수에 대한 절대 레벨이 2 보다 큰지를 표시하는, 상기 컨텍스트 디코딩하는 수단; 컨텍스트 디코딩된 신택스 엘리먼트들의 빈들에 기초하여 변환 유닛의 계수들의 제 1 세트에 대한 값들을 결정하는 수단; 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 수단으로서, 추가적인 신택스 엘리먼트들을 바이패스 디코딩하는 수단은, 계수들의 제 2 세트의 계수에 대해, 라이스 파라미터에 대한 값을 도출하는 수단을 포함하는, 상기 바이패스 디코딩하는 수단; 및, 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하는 수단을 포함하고, 상기 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하는 수단은, 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하는 수단으로서, 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별하는, 상기 제로 파라미터에 대한 값을 결정하는 수단; 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하는 수단; 및, 제로 파라미터에 대한 값 및 제 1 계수에 대한 제 1 코딩된 값에 기초하여, 상기 제 1 계수에 대한 레벨을 결정하는 수단을 포함한다.

[0017] 하나 이상의 예들의 상세들이 첨부 도면들 및 이하의 설명에서 전개된다. 다른 피처들, 목적들, 및 이점들은 설명, 도면들, 및 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0018] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 나타내는 블록도이다.

도 2a 및 도 2b 는 예시적인 쿼드트리 바이너리 트리 (QTBT) 구조, 및 대응하는 코딩 트리 유닛 (CTU) 을 나타내는 개념도들이다.

도 3 은 코딩 그룹 (CG) 내의 계수들에 대한 절대 레벨 값들을 나타내는 신택스 엘리먼트들에 대한 예시적인 순서를 도시한다.

도 4 는 확률 모델들을 선택하기 위해 사용되는 템플릿의 예시를 나타낸다.

도 5 는 Par 플래그 후의 제 1 패스에서의 인터리빙된 Gt2 플래그의 예를 도시한다.

도 6 은 Gt1 플래그 후의 제1 패스에서의 인터리빙된 Gt2 플래그의 예를 도시한다.

도 7 은 제 1 코딩 패스에서 SIG-Gt1-Par-Gt2 코딩에 대해 정규 코딩된 빈 한계에 도달하는 마지막 계수의 부분 코딩의 예를 도시한다.

도 8 은 제1 코딩 패스에서 SIG-Gt1-Gt2-Par 코딩에 대해 정규 코딩된 빈 한계에 도달하는 마지막 계수의 부분 코딩의 예를 도시한다.

도 9 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코더를 나타내는 블록도이다.

도 10 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더를 나타내는 블록도이다.

도 11a 및 도 11b 는 이진 산술 코딩에서의 레인지 업데이트 프로세스를 나타내는 개념도들이다.

도 12 는 이진 산술 코딩에서의 출력 프로세스를 나타내는 개념도이다.

도 13 은 비디오 인코더에서의 컨텍스트 적응 이진 산술 코딩 (CABAC) 코더를 나타내는 블록도이다.

도 14 는 비디오 디코더에서의 CABAC 코더를 나타내는 블록도이다.

도 15 는 비디오 인코더의 예시적인 동작을 나타내는 플로우차트이다.

도 16 은 비디오 디코더의 예시적인 동작을 나타내는 플로우차트이다.

도 17 은 비디오 디코더의 예시적인 동작을 나타내는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0019] **상세한 설명**

[0020] 비디오 코딩 (예컨대, 비디오 인코딩 및/또는 비디오 디코딩) 은 통상적으로 동일한 픽처에서 이미 코딩된 비디오 데이터의 블록 (예를 들어, 인트라 예측) 또는 상이한 픽처에서 이미 코딩된 비디오 데이터의 블록 (예를 들어, 인터 예측) 으로부터의 비디오 데이터의 블록을 예측하는 것을 수반한다. 일부 경우들에서, 비디오 인코더는 또한 예측 블록을 원래 블록과 비교함으로써 잔차 데이터를 계산한다. 따라서, 잔차 데이터 (residual data) 는 비디오 데이터의 예측 블록과 원래 블록 사이의 차이를 나타낸다. 잔차 데이터를 시그널링하는데 필요한 비트들의 수를 감소시키기 위해, 비디오 인코더는 잔차 데이터를 변환 및 양자화하고, 변환 및 양자화된 잔차 데이터를 인코딩된 비트스트림에서 시그널링한다. 변환 및 양자화 프로세스들에 의해 달성되는 압축은 손실성일 수도 있으며, 이는 변환 및 양자화 프로세스들이 디코딩된 비디오 데이터에 왜곡을 도입할 수도 있다는 것을 의미한다.

[0021] 비디오 디코더는 잔차 데이터를 디코딩하고 예측 블록에 가산하여 예측 블록 단독보다 더 가깝게 원래 비디오 블록과 매칭하는 재구성된 비디오 블록을 생성한다. 잔차 데이터의 변환 및 양자화에 의해 도입된 손실로 인해, 재구성된 블록은 왜곡 (distortion) 또는 아티팩트들 (artifacts) 을 가질 수도 있다. 아티팩트 또는 왜곡의 하나의 통상적인 유형은 블록키니스 (blockiness) 로서 지칭되며, 여기서 비디오 데이터를 코딩하기 위해 사용되는 블록들의 경계들이 가시적이다.

[0022] 디코딩된 비디오의 품질을 더 향상시키기 위해, 비디오 디코더는 재구성된 비디오 블록들에 대해 하나 이상의 필터링 동작들을 수행할 수 있다. 이들 필터링 동작들의 예들은 디블록킹 필터링, 샘플 적응 오프셋 (SAO) 필터링, 및 적응 루프 필터링 (ALF) 을 포함한다. 이들 필터링 동작들을 위한 파라미터들은 비디오 인코더에 의해 결정될 수도 있고 인코딩된 비디오 비트스트림에서 명시적으로 시그널링될 수도 있거나, 또는 파라미터들이 인코딩된 비디오 비트스트림에서 명시적으로 시그널링될 필요 없이 비디오 디코더에 의해 암시적으로 결정될 수도 있다.

[0023] 위에서 도입된 바와 같이, 비디오 인코더는 잔차 데이터를 변환하여 변환 계수들을 생성한다. 이들 변환 계수들은 추가적으로 양자화될 수도 있다. 본 개시에서, 용어 변환 계수, 또는 계수는 양자화된 변환 계수 또

는 양자화되지 않은 변환 계수를 지칭할 수도 있다. 본 개시물은 변환 계수, 예를 들어, 양자화된 변환 계수들의 값들을 비디오 인코더로부터 비디오 디코더로 시그널링하기 위한 기법들을 설명한다. 보다 구체적으로, 본 개시물은 비트들의 바이너리 표현을 일련의 비-바이너리 값의 양자화된 변환 계수들로 변환하는 엔트로피 디코딩 프로세스와 관련된 기법들을 설명한다. 일반적으로 엔트로피 디코딩의 역 프로세스인 대응하는 엔트로피 인코딩 프로세스가 또한 본 개시에서 설명된다.

[0024] 일 예에서, 본 개시물은, 컨텍스트 적응 이진 산술 코딩 (CABAC) 이 1 보다 큰 계수 레벨들 및 2 보다 큰 계수 레벨들과 같은, 유의성 계수들의 다른 표시들을 코딩하는데 사용되는 계수들의 블록에 대한 계수 레벨들의 잔여 절대 값들을 코딩하기 위해, 코드들, 예를 들어, 골롬-라이스 (Golomb-Rice) 코드들 또는 지수-골롬 (Exponential-Golomb) 코드들을 정의하는데 사용되는 라이스 파라미터를 결정하기 위한 기법들을 설명한다. 계수 레벨들은 손실 코딩의 경우 변환 계수들의 레벨들일 수도 있고, 변환 생략 모드에서 무손실 코딩 또는 손실 코딩의 경우 변환이 적용되지 않는 계수들의 레벨들 (즉, 잔차 픽셀 값들) 일 수도 있다. 이하에서 더 상세히 설명되는 바와 같이, 계수 레벨은 계수 레벨에 대한 절대 값 또는 계수 레벨에 대한 잔여 레벨 (remaining level) 일 수도 있다.

[0025] 라이스 파라미터는 골롬 코드들의 패밀리, 예컨대, 골롬-라이스 코드들 또는 지수-골롬 코드들로부터 코드워드 세트를 선택하는데 사용되는 튜닝가능한 (tunable) 값이다. 라이스 파라미터에 의해 정의된 코드들은 변환 유닛 (TU) 또는 계수 그룹 (CG), 즉 계수들의 블록에서 적어도 하나의 계수에 대한 계수 레벨의 잔여 절대 값을 코딩하는데 사용될 수도 있다. CG들의 각각은 비디오 데이터의 변환 블록의 4×4 변환 블록 또는 4×4 서브블록일 수도 있다. CG 들은 손실 코딩의 경우 변환 계수들을 포함할 수도 있고, 변환 생략 모드의 무손실 코딩 또는 손실 코딩의 경우 변환이 적용되지 않는 계수들을 포함할 수도 있다.

[0026] 본 개시물은 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하기 위한 기법들을 추가로 설명한다. 제로 파라미터는 제로의 계수 레벨에 대응하는 비트스트림 값을 나타낸다. 계수 레벨이 제로일 확률이 상대적으로 낮은 경우, 더 긴 코드워드 또는 비트스트림 값은 더 짧은 코드워드들이 비-제로 값들에 대해 사용될 수도 있도록 제로의 계수 레벨에 할당될 수도 있다. 본 개시물의 기법들은 비트들이 계수 레벨들의 코딩에서 절약될 수도 있도록 제로 파라미터들의 선택을 개선함으로써 비디오 압축을 개선할 수도 있다.

[0027] 본 개시의 기법들은 고 효율 비디오 코딩 (HEVC) 과 같은 기존의 비디오 코덱들의 임의의 것에 적용될 수도 있거나, 또는 현재 개발 중인 다용도 비디오 코딩 (VVC) 과 같은 새로운 비디오 코딩 표준들을 위한 유망한 코딩 툴일 수도 있다.

[0028] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다. 본 개시의 기법들은 일반적으로 비디오 데이터를 코딩 (인코딩 및/또는 디코딩) 하는 것과 관련된다. 일반적으로, 비디오 데이터는 비디오를 프로세싱하기 위한 임의의 데이터를 포함한다. 따라서, 비디오 데이터는 원시, 코딩되지 않은 비디오, 인코딩된 비디오, 디코딩된 (예를 들어, 재구성된) 비디오, 및 비디오 메타데이터, 이를 테면 시그널링 데이터를 포함할 수도 있다.

[0029] 도 1 에 도시된 바와 같이, 시스템 (100) 은 이 예에서 목적지 디바이스 (116) 에 의해 디코딩 및 디스플레이될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (102) 를 포함한다. 특히, 소스 디바이스 (102) 는 컴퓨터 판독 가능 매체 (110) 를 통해 목적지 디바이스 (116) 에 비디오 데이터를 제공한다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 전화기 핸드셋, 예컨대 스마트폰들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 무선 통신을 위해 장비될 수도 있고, 따라서 무선 통신 디바이스들로 지칭될 수도 있다.

[0030] 도 1 의 예에서, 소스 디바이스 (102) 는 비디오 소스 (104), 메모리 (106), 비디오 인코더 (200), 및 출력 인터페이스 (108) 를 포함한다. 목적지 디바이스 (116) 는 입력 인터페이스 (122), 비디오 디코더 (300), 메모리 (120), 및 디스플레이 디바이스 (118) 를 포함한다. 본 개시에 따르면, 소스 디바이스 (102) 의 비디오 인코더 (200) 및 목적지 디바이스 (116) 의 비디오 디코더 (300) 는 본원에 기술된 계수 코딩을 위한 기법들을 적용하도록 구성될 수도 있다. 따라서, 소스 디바이스 (102) 는 비디오 인코딩 디바이스의 일 예를 나타내는 한편, 목적지 디바이스 (116) 는 비디오 디코딩 디바이스의 일 예를 나타낸다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102) 는 외부 카메라와 같은 외부 비디오 소스로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로,

목적지 디바이스 (116) 는 통합된 디스플레이 디바이스를 포함하는 것보다는, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0031] 도 1 에서 도시된 시스템 (100) 은 단지 하나의 예일 뿐이다. 일반적으로, 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스는 본원에 기술된 계수 코딩을 위한 기법들을 수행할 수도 있다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 소스 디바이스 (102) 가 목적지 디바이스 (116) 로의 송신을 위한 코딩된 비디오 데이터를 생성하는 이러한 코딩 디바이스들의 예들일 뿐이다. 본 개시는 데이터의 코딩 (인코딩 및/또는 디코딩) 을 수행하는 디바이스로서 "코딩" 디바이스를 지칭한다. 따라서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 코딩 디바이스들, 특히 각각 비디오 인코더 및 비디오 디코더의 예들을 나타낸다. 일부 예들에 있어서, 소스 디바이스 (102) 및 목적지 디바이스 (116) 는, 소스 디바이스 (102) 및 목적지 디바이스 (116) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (100) 은 예를 들면, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오 텔레포니를 위해, 소스 디바이스 (102) 와 목적지 디바이스 (116) 간의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0032] 일반적으로, 비디오 소스 (104) 는 비디오 데이터 (즉, 원시, 코딩되지 않은 비디오 데이터) 의 소스를 나타내며 픽처들에 대한 데이터를 인코딩하는 비디오 인코더 (200) 에 비디오 데이터의 순차적인 일련의 픽처들 (또한 "프레임들" 로도 지칭됨) 을 제공한다. 소스 디바이스 (102) 의 비디오 소스 (104) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 원시 비디오를 포함하는 비디오 아카이브 (video archive), 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스 (video feed interface) 를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (104) 는 컴퓨터 그래픽 기반 데이터를 소스 비디오로서, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합으로서 생성할 수도 있다. 각각의 경우에 있어서, 비디오 인코더 (200) 는 캡처된, 사전 캡처된, 또는 컴퓨터 생성된 비디오 데이터를 인코딩한다. 비디오 인코더 (200) 는 픽처들을 수신된 순서 (때때로 "디스플레이 순서" 로 지칭됨) 로부터 코딩을 위한 코딩 순서로 재배열할 수도 있다. 비디오 인코더 (200) 는 인코딩된 비디오 데이터를 포함하는 비트스트림을 생성할 수도 있다. 소스 디바이스 (102) 는 예를 들어, 목적지 디바이스 (116) 의 입력 인터페이스 (122) 에 의한 수신 및/또는 취출을 위해 인코딩된 비디오 데이터를 출력 인터페이스 (108) 를 통해 컴퓨터 판독가능 매체 (110) 상으로 출력할 수도 있다.

[0033] 소스 디바이스 (102) 의 메모리 (106) 및 목적지 디바이스 (116) 의 메모리 (120) 는 범용 메모리들을 나타낸다. 일부 예에서, 메모리들 (106, 120) 은 원시 비디오 데이터, 예를 들어, 비디오 소스 (104) 로부터의 원시 비디오 및 비디오 디코더 (300) 로부터의 원시, 디코딩된 비디오 데이터를 저장할 수도 있다. 추가적으로 또는 대안적으로, 메모리들 (106, 120) 은 예를 들어, 각각 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해 실행가능한 소프트웨어 명령들을 저장할 수도 있다. 메모리 (106) 및 메모리 (120) 는 이 예에서 비디오 인코더 (200) 및 비디오 디코더 (300) 와 별도로 도시되지만, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 또한 기능적으로 유사하거나 또는 동등한 목적들을 위한 내부 메모리들을 포함할 수도 있음을 이해해야 한다. 또한, 메모리들 (106, 120) 은 예를 들어, 비디오 인코더 (200) 로부터 출력되고 비디오 디코더 (300) 에 입력되는 인코딩된 비디오 데이터를 저장할 수도 있다. 일부 예들에서, 메모리들 (106, 120) 의 부분들은 예를 들어, 원시, 디코딩된, 및/또는 인코딩된 비디오 데이터를 저장하기 위해 하나 이상의 비디오 버퍼들로서 할당될 수도 있다.

[0034] 컴퓨터 판독가능 매체 (110) 는 인코딩된 비디오 데이터를 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로 전송할 수 있는 임의의 타입의 매체 또는 디바이스를 나타낼 수도 있다. 하나의 예에서, 컴퓨터 판독가능 매체 (110) 는, 소스 디바이스 (102) 로 하여금, 실시간으로, 예를 들어, 무선 주파수 네트워크 또는 컴퓨터 기반 네트워크를 통해 직접 목적지 디바이스 (116) 로 인코딩된 비디오 데이터를 송신할 수 있게 하기 위한 통신 매체를 나타낸다. 무선 통신 프로토콜과 같은 통신 표준에 따라, 출력 인터페이스 (108) 는 인코딩된 비디오 데이터를 포함하는 송신 신호를 변조할 수도 있고, 입력 인터페이스 (122) 는 수신된 송신 신호를 복조할 수도 있다. 통신 매체는 무선 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷 기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0035] 일부 예들에서, 소스 디바이스 (102) 는 출력 인터페이스 (108) 로부터 저장 디바이스 (112) 로 인코딩된 데이

터를 출력할 수도 있다. 유사하게, 목적지 디바이스 (116) 는 입력 인터페이스 (122) 를 통해 저장 디바이스 (112) 로부터의 인코딩된 데이터에 액세스할 수도 있다. 저장 디바이스 (112) 는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체와 같은 다양한 분산된 또는 로컬 액세스된 데이터 저장 매체 중 임의의 것을 포함할 수도 있다.

[0036] 일부 예들에서, 소스 디바이스 (102) 는 소스 디바이스 (102) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 (114) 또는 다른 중간 저장 디바이스에 인코딩된 비디오 데이터를 출력할 수도 있다. 목적지 디바이스 (116) 는 스트리밍 또는 다운로드를 통해 파일 서버 (114) 로부터 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버 (114) 는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (116) 에 송신할 수 있는 임의의 타입의 서버 디바이스일 수도 있다. 파일 서버 (114) 는 (예를 들어, 웹 사이트를 위한) 웹 서버, 파일 전송 프로토콜 (FTP) 서버, 콘텐츠 전달 네트워크 디바이스, 또는 NAS (network attached storage) 디바이스를 나타낼 수도 있다. 목적지 디바이스 (116) 는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 파일 서버 (114) 로부터 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 (114) 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한, 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, 디지털 가입자 라인 (DSL), 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 파일 서버 (114) 및 입력 인터페이스 (122) 는 스트리밍 송신 프로토콜, 다운로드 송신 프로토콜, 또는 이들의 조합에 따라 동작하도록 구성될 수도 있다.

[0037] 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 무선 송신기/수신기, 모뎀들, 유선 네트워킹 컴포넌트들 (예를 들어, 이더넷 카드들), 다양한 IEEE 802.11 표준들 중 임의의 것에 따라 동작하는 무선 통신 컴포넌트들, 또는 다른 물리적 컴포넌트들을 나타낼 수도 있다. 출력 인터페이스 (108) 및 입력 인터페이스 (122) 가 무선 컴포넌트들을 포함하는 예들에 있어서, 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 4G, 4G-LTE (Long-Term Evolution), LTE 어드밴스드, 5G 등과 같은 셀룰러 통신 표준에 따라, 인코딩된 비디오 데이터와 같은 데이터를 전송하도록 구성될 수도 있다. 출력 인터페이스 (108) 가 무선 송신기를 포함하는 일부 예들에 있어서, 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 IEEE 802.11 사양, IEEE 802.15 사양 (예를 들어, ZigBee™), Bluetooth™ 표준 등과 같은 다른 무선 표준들에 따라, 인코딩된 비디오 데이터와 같은 데이터를 전송하도록 구성될 수도 있다. 일부 예들에서, 소스 디바이스 (102) 및/또는 목적지 디바이스 (116) 는 개개의 시스템-온-칩 (SoC) 디바이스들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102) 는 비디오 인코더 (200) 및/또는 출력 인터페이스 (108) 에 기인하는 기능성을 수행하기 위한 SoC 디바이스를 포함할 수도 있고, 목적지 디바이스 (116) 는 비디오 디코더 (300) 및/또는 입력 인터페이스 (122) 에 기인하는 기능성을 수행하기 위한 SoC 디바이스를 포함할 수도 있다.

[0038] 본 개시의 기법들은 오버-디-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 이를 테면 DASH (dynamic adaptive streaming over HTTP), 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들 중 임의의 것을 지원하여 비디오 코딩에 적용될 수도 있다.

[0039] 목적지 디바이스 (116) 의 입력 인터페이스 (122) 는 컴퓨터 판독가능 매체 (110) (예를 들어, 통신 매체, 저장 디바이스 (112), 파일 서버 (114) 등) 로부터 인코딩된 비디오 비트스트림을 수신한다. 인코딩된 비디오 비트스트림은 비디오 블록들 또는 다른 코딩된 유닛들 (예를 들어, 슬라이스들, 픽처들, 픽처들의 그룹들, 시퀀스들 등) 의 프로세싱 및/또는 특성들을 기술하는 값들을 갖는 신택스 엘리먼트들과 같은, 비디오 디코더 (300) 에 의해 또한 사용되는 비디오 인코더 (200) 에 의해 정의된 시그널링 정보를 포함할 수도 있다. 디스플레이 디바이스 (118) 는 디코딩된 비디오 데이터의 디코딩된 픽처들을 사용자에게 디스플레이한다. 디스플레이 디바이스 (118) 는 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디스플레이 디바이스를 포함할 수도 있다.

[0040] 도 1 에 도시되지는 않았지만, 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각 오디오 인코더 및/또는 오디오 디코더와 통합될 수도 있고, 공통 데이터 스트림에서 오디오 및 비디오 양자 모두를 포함하는 멀티플렉싱된 스트림들을 핸들링하기 위해, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및/또는 소프트웨어를 포함할 수도 있다. 적용가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다

른 프로토콜들, 이를 테면 사용자 데이터그램 프로토콜 (UDP) 에 따를 수도 있다.

[0041] 비디오 인코더 (200) 및 비디오 디코더 (300) 각각은 다양한 적합한 인코더 및/또는 디코더 회로부, 이를 테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들 중 임의의 것으로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어에서 구현되는 경우, 디바이스는 적합한 비일시적 컴퓨터 판독가능 매체에 소프트웨어에 대한 명령들을 저장하고, 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있는데, 이들 중 어느 하나는 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다. 비디오 인코더 (200) 및/또는 비디오 디코더 (300) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 예컨대 셀룰러 전화기를 포함할 수도 있다.

[0042] 비디오 인코더 (22) 및 비디오 디코더 (300) 는 고 효율 비디오 코딩 (HEVC) 으로도 지칭되는 ITU-T H.265 와 같은 비디오 코딩 표준 또는 멀티-뷰 및/또는 스케일러블 비디오 코딩 확장들과 같은 그의 확장들에 따라 동작할 수도 있다. 대안적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 JEM (Joint Exploration Test Model) 또는 VVC (Versatile Video Coding) 로도 또한 지칭되는 ITU-T H.266 과 같은 다른 독점 또는 산업 표준들에 따라 동작할 수도 있다. VVC 표준의 최신 초안은 Bross 등의, "Versatile Video Coding (Draft 6)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Gothenburg, SE, 3-12 July 2019, JVET-O2001-vE (이하 "VVC Draft 6" 라 함) 에서 설명되어 있다. 하지만, 본 개시의 기법들은 임의의 특정 코딩 표준에 제한되지 않는다.

[0043] 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 픽처들의 블록 기반 코딩을 수행할 수도 있다. 용어 "블록" 은 일반적으로 프로세싱될 (예를 들어, 인코딩될, 디코딩될, 또는 다르게는 인코딩 및/또는 디코딩 프로세스에서 사용될) 데이터를 포함하는 구조를 지칭한다. 예를 들어, 블록은 루미넌스 및/또는 크로미넌스 데이터의 샘플들의 2 차원 행렬을 포함할 수도 있다. 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 YUV (예를 들어, Y, Cb, Cr) 포맷으로 표현된 비디오 데이터를 코딩할 수도 있다. 즉, 픽처의 샘플들에 대한 적색, 녹색, 및 청색 (RGB) 데이터를 코딩하는 것보다는, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 및 크로미넌스 컴포넌트들을 코딩할 수도 있고, 여기서 크로미넌스 컴포넌트들은 적색 색조 및 청색 색조 크로미넌스 컴포넌트들 양자 모두를 포함할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 인코딩 이전에 수신된 RGB 포매팅된 데이터를 YUV 표현으로 변환하고, 비디오 디코더 (300) 는 YUV 표현을 RGB 포맷으로 변환한다. 대안적으로는, 사전 및 사후 프로세싱 유닛들 (도시되지 않음) 이 이들 변환들을 수행할 수도 있다.

[0044] 본 개시는 일반적으로 픽처의 데이터를 인코딩 또는 디코딩하는 프로세스를 포함하는 픽처들의 코딩 (예를 들어, 인코딩 및 디코딩) 을 참조할 수도 있다. 유사하게, 본 개시는, 예를 들어, 예측 및/또는 잔차 코딩과 같은, 블록들에 대한 데이터를 인코딩 또는 디코딩하는 프로세스를 포함하는 픽처의 블록들의 코딩을 참조할 수도 있다. 인코딩된 비디오 비트스트림은 일반적으로 코딩 결정들 (예를 들어, 코딩 모드들) 및 픽처들의 블록들로의 파티셔닝을 나타내는 신택스 엘리먼트들에 대한 일련의 값들을 포함한다. 따라서, 픽처 또는 블록을 코딩하는 것에 대한 참조들은 일반적으로 픽처 또는 블록을 형성하는 신택스 엘리먼트들에 대한 코딩 값들로서 이해되어야 한다.

[0045] HEVC 는 코딩 유닛들 (CU들), 예측 유닛들 (PU들), 및 변환 유닛들 (TU들) 을 포함한 다양한 블록들을 정의한다. HEVC 에 따르면, 비디오 코더 (이를 테면 비디오 인코더 (200)) 는 쿼드트리 구조에 따라 코딩 트리 유닛 (CTU) 을 CU들로 파티셔닝한다. 즉, 비디오 코더는 CTU들 및 CU들을 4 개의 동일한 비오버랩하는 정사각형들로 파티셔닝하고, 쿼드트리의 각각의 노드는 0 개 또는 4 개 중 어느 하나의 자식 노드들을 갖는다. 자식 노드들 없는 노드들은 "리프 노드들" 로 지칭될 수도 있고, 그러한 리프 노드들의 CU들은 하나 이상의 PU들 및/또는 하나 이상의 TU들을 포함할 수도 있다. 비디오 코더는 PU들 및 TU들을 추가로 파티셔닝할 수도 있다. 예를 들어, HEVC 에서, 잔차 쿼드트리 (RQT) 는 TU들의 파티셔닝을 나타낸다. HEVC 에서, PU들은 인터 예측 데이터를 나타내는 한편, TU들은 잔차 데이터를 나타낸다. 인트라 예측되는 CU들은 인트라 모드 표시와 같은 인트라 예측 정보를 포함한다.

[0046] 다른 예로서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 JEM 또는 VVC 에 따라 동작하도록 구성될 수도 있다. JEM 또는 VVC 에 따르면, 비디오 코더 (이를 테면 비디오 인코더 (200)) 는 픽처를 복수의 코딩 트리

유닛들 (CTU들) 로 파티셔닝한다. 비디오 인코더 (200) 는 쿼드트리 바이너리 트리 (quadtree-binary tree; QTBT) 구조 또는 멀티 타입 트리 (Multi-Type Tree; MTT) 구조와 같은 트리 구조에 따라 CTU 를 파티셔닝할 수도 있다. QTBT 구조는 HEVC 의 CU들, PU들, 및 TU들 간의 분리와 같은 다중 파티션 타입들의 개념들을 제거한다. QTBT 구조는 2 개의 레벨들: 쿼드트리 파티셔닝에 따라 파티셔닝된 제 1 레벨, 및 바이너리 트리 파티셔닝에 따라 파티셔닝된 제 2 레벨을 포함한다. QTBT 구조의 루트 노드는 CTU 에 대응한다. 바이너리 트리들의 리프 노드들은 코딩 유닛들 (CU들) 에 대응한다.

[0047] MTT 파티셔닝 구조에서, 블록들은 쿼드트리 (QT) 파티션, 바이너리 트리 (BT) 파티션, 및 하나 이상의 타입들의 트리플 트리 (TT) (터너리 (ternary) 트리 (TT) 로도 불림) 파티션들을 사용하여 파티셔닝될 수도 있다. 트리플 또는 터너리 트리 파티션은 블록이 3 개의 서브-블록들로 스플릿팅되는 파티션이다. 일부 예들에서, 트리플 또는 터너리 트리 파티션은 센터를 통해 원래 블록을 분할하지 않고 블록을 3 개의 서브-블록들로 분할한다. MTT 에서의 파티셔닝 타입들 (예를 들어, QT, BT, 및 TT) 은 대칭적 또는 비대칭적일 수도 있다.

[0048] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 및 크로미넌스 컴포넌트들의 각각을 나타내기 위해 단일 QTBT 또는 MTT 구조를 사용할 수도 있는 한편, 다른 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 2 개 이상의 QTBT 또는 MTT 구조들, 이를 테면 루미넌스 컴포넌트를 위한 하나의 QTBT/MTT 구조 및 양자의 크로미넌스 컴포넌트들을 위한 다른 QTBT/MTT 구조 (또는 각각의 크로미넌스 컴포넌트들을 위한 2 개의 QTBT/MTT 구조들) 를 사용할 수도 있다.

[0049] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 HEVC 당 쿼드트리 파티셔닝, QTBT 파티셔닝, MTT 파티셔닝, 또는 다른 파티셔닝 구조들을 사용하도록 구성될 수도 있다. 설명의 목적들을 위해, 본 개시의 기법들의 설명은 QTBT 파티셔닝에 대하여 제시된다. 그러나, 본 개시의 기법들은 또한, 쿼드트리 파티셔닝, 또는 다른 타입들의 파티셔닝에도 사용하도록 구성된 비디오 코더들에 적용될 수도 있음이 이해되어야 한다.

[0050] 블록들 (예를 들어, CTU들 또는 CU들) 은 픽처에서 다양한 방식으로 그룹화될 수도 있다. 하나의 예로서, 브릭 (brick) 은 픽처에서의 특정 타일 내의 CTU 행들의 직사각형 영역을 지칭할 수도 있다. 타일은 픽처에서의 특정 타일 행 및 특정 타일 열 (column) 내의 CTU들의 직사각형 영역일 수도 있다. 타일 열은 (예를 들어, 픽처 파라미터 세트에서와 같은) 신택스 엘리먼트들에 의해 특정된 폭 및 픽처의 높이와 동일한 높이를 갖는 CTU들의 직사각형 영역을 지칭한다. 타일 행은, 픽처의 폭과 동일한 폭 및 (예를 들어, 픽처 파라미터 세트에서와 같은) 신택스 엘리먼트들에 의해 특정된 높이를 갖는 CTU들의 직사각형 영역을 지칭한다.

[0051] 일부 예들에서, 타일은 다중 브릭들로 파티셔닝될 수도 있고, 그 브릭들의 각각은 타일 내의 하나 이상의 CTU 행들을 포함할 수도 있다. 다중 브릭들로 파티셔닝되지 않은 타일은 또한 브릭으로 지칭될 수도 있다. 하지만, 타일의 진정한 서브세트인 브릭은 타일로서 지칭되지 않을 수도 있다.

[0052] 픽처에서의 브릭들은 또한 슬라이스로 배열될 수도 있다. 슬라이스는 단일 네트워크 추상화 계층 (NAL) 유닛에 배타적으로 포함될 수도 있는 픽처의 정수 개의 브릭들일 수도 있다. 일부 예들에서, 슬라이스는 다수의 완전한 타일들 또는 오직 하나의 타일의 연속적인 시퀀스의 완전한 브릭들 중 어느 하나를 포함한다.

[0053] 본 개시는 수직 및 수평 차원들의 관점에서 블록 (예컨대 CU 또는 다른 비디오 블록) 의 샘플 차원들을 지칭하기 위해 상호교환가능하게 "NxN" 및 "N 바이 N", 예를 들어, 16x16 샘플들 또는 16 바이 16 샘플들을 사용할 수도 있다. 일반적으로, 16x16 CU 는 수직 방향에서 16 샘플들 (y = 16) 그리고 수평 방향에서 16 샘플들 (x = 16) 을 가질 것이다. 마찬가지로, NxN CU 는 일반적으로 수직 방향에서 N 샘플들 및 수평 방향에서 N 샘플들을 갖고, 여기서 N 은 음이 아닌 정수 값을 나타낸다. CU 에서의 샘플들은 행들 (rows) 및 열들 (columns) 로 배열될 수도 있다. 더욱이, CU들은 수직 방향에서와 동일한 수의 샘플들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들면, CU들은 NxM 샘플들을 포함할 수도 있고, 여기서 M 은 N 과 반드시 동일한 것은 아니다.

[0054] 비디오 인코더 (200) 는 예측 및/또는 잔차 정보를 나타내는 CU들에 대한 비디오 데이터, 및 다른 정보를 인코딩한다. 예측 정보는 CU 에 대한 예측 블록을 형성하기 위하여 CU 가 어떻게 예측될지를 표시한다. 잔차 정보는 일반적으로 인코딩 이전의 CU 의 샘플들과 예측 블록 사이의 샘플 별 (sample-by-sample) 차이들을 나타낸다.

[0055] CU 를 예측하기 위해, 비디오 인코더 (200) 는 일반적으로 인터 예측 또는 인트라 예측을 통해 CU 에 대한 예측 블록을 형성할 수도 있다. 인터 예측은 일반적으로 이전에 코딩된 픽처의 데이터로부터 CU 를 예측하는 것을 지칭하는 반면, 인트라 예측은 일반적으로 동일한 픽처의 이전에 코딩된 데이터로부터 CU 를 예측하는 것을

지칭한다. 인터 예측을 수행하기 위해, 비디오 인코더 (200) 는 하나 이상의 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 비디오 인코더 (200) 는 일반적으로 CU 와 레퍼런스 블록 사이의 차이들의 관점에서, CU 에 밀접하게 매칭하는 레퍼런스 블록을 식별하기 위해 모션 탐색을 수행할 수도 있다. 비디오 인코더 (200) 는 절대 차이의 합 (sum of absolute difference; SAD), 제곱 차이들의 합 (sum of squared differences; SSD), 평균 절대 차이 (mean absolute difference; MAD), 평균 제곱 차이들 (mean squared differences; MSD), 또는 레퍼런스 블록이 현재 CU 에 밀접하게 매칭하는지 여부를 결정하기 위한 다른 그러한 차이 계산들을 사용하여 차이 메트릭을 계산할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 단방향 예측 또는 양방향 예측을 사용하여 현재 CU 를 예측할 수도 있다.

[0056] JEM 및 VVC 의 일부 예들은 또한, 인터 예측 모드로 고려될 수도 있는 아핀 모션 보상 모드 (affine motion compensation mode) 를 제공한다. 아핀 모션 보상 모드에서, 비디오 인코더 (200) 는 줌 인 또는 아웃, 회전, 원근 모션 (perspective motion), 또는 다른 불규칙한 모션 타입들과 같은 비-병진 모션을 나타내는 2 개 이상의 모션 벡터들을 결정할 수도 있다.

[0057] 인트라 예측을 수행하기 위해, 비디오 인코더 (200) 는 예측 블록을 생성하기 위해 인트라 예측 모드를 선택할 수도 있다. JEM 및 VVC 의 일부 예들은 다양한 방향 모드들 뿐만 아니라 평면 모드 및 DC 모드를 포함하여 67 개의 인트라 예측 모드들을 제공한다. 일반적으로, 비디오 인코더 (200) 는 현재 블록의 샘플들을 예측할 현재 블록 (예를 들어, CU 의 블록) 에 대한 이웃하는 샘플들을 기술하는 인트라 예측 모드를 선택한다. 그러한 샘플들은 일반적으로, 비디오 인코더 (200) 가 래스터 스캔 순서로 (왼쪽에서 오른쪽으로, 상단에서 하단으로) CTU들 및 CU들을 코딩하는 것을 가정하여, 현재 블록과 동일한 픽처에서 현재 블록의 상측, 상측 및 좌측에, 또는 좌측에 있을 수도 있다.

[0058] 비디오 인코더 (200) 는 현재 블록에 대한 예측 모드를 나타내는 데이터를 인코딩한다. 예를 들어, 인터 예측 모드들에 대해, 비디오 인코더 (200) 는 다양한 이용가능한 인터 예측 모드들 중 어느 것이 사용되는지를 나타내는 데이터 뿐만 아니라 대응하는 모드에 대한 모션 정보를 인코딩할 수도 있다. 단방향 또는 양방향 인터 예측을 위해, 예를 들어, 비디오 인코더 (200) 는 어드밴스드 모션 벡터 예측 (AMVP) 또는 병합 모드를 사용하여 모션 벡터들을 인코딩할 수도 있다. 비디오 인코더 (200) 는 유사한 모드들을 사용하여 아핀 모션 보상 모드에 대한 모션 벡터들을 인코딩할 수도 있다.

[0059] 블록의 인트라 예측 또는 인터 예측과 같은 예측에 이어, 비디오 인코더 (200) 는 블록에 대한 잔차 데이터를 계산할 수도 있다. 잔차 블록과 같은 잔차 데이터는 대응하는 예측 모드를 사용하여 형성되는, 블록과 블록에 대한 예측 블록 사이의 샘플 별 차이들을 나타낸다. 비디오 인코더 (200) 는 샘플 도메인 대신에 변환 도메인에서 변환된 데이터를 생성하기 위해, 잔차 블록에 하나 이상의 변환들을 적용할 수도 있다. 예를 들어, 비디오 인코더 (200) 는 이산 코사인 (DCT), 정수 변환, 웨이브릿 변환, 또는 개념적으로 유사한 변환을 잔차 비디오 데이터에 적용할 수도 있다. 추가적으로, 비디오 인코더 (200) 는 MDNSST (mode-dependent non-separable secondary transform), 신호 의존적 변환, Karhunen-Loeve 변환 (KLT) 등과 같은 제 1 변환에 후속하는 2 차 변환을 적용할 수도 있다. 비디오 인코더 (200) 는 하나 이상의 변환들의 적용에 이어 변환 계수들을 생성한다.

[0060] 상기 언급된 바와 같이, 변환 계수들을 생성하기 위한 임의의 변환들에 이어, 비디오 인코더 (200) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로, 변환 계수들이 그 변환 계수들을 나타내는데 사용된 데이터의 양을 가능하게는 감소시키도록 양자화되어 추가 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스를 수행함으로써, 비디오 인코더 (200) 는 변환 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, 비디오 인코더 (200) 는 양자화 동안 n-비트 값을 m-비트 값으로 라운딩 다운할 수도 있고, 여기서 n 은 m 보다 크다. 일부 예들에서, 양자화를 수행하기 위해, 비디오 인코더 (200) 는 양자화될 값의 비트단위 우측-시프트를 수행할 수도 있다.

[0061] 양자화에 이어, 비디오 인코더 (200) 는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 포함한 2 차원 행렬로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 벡터의 전방에 더 높은 에너지 (및 따라서 더 낮은 주파수) 변환 계수들을 배치하고 벡터의 후방에 더 낮은 에너지 (및 따라서 더 높은 주파수) 변환 계수들을 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 양자화된 변환 계수들을 스캔하기 위해 미리정의된 스캔 순서를 활용하여 직렬화된 벡터를 생성한 후, 벡터의 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (200) 는 적응적 스캔을 수행할 수도 있다. 1 차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캔한 후, 비디오 인코더 (200) 는, 예를 들어, 컨텍스트 적응

이진 산술 코딩 (CABAC) 에 따라, 1 차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (200) 는 또한, 비디오 데이터를 디코딩하는데 있어서 비디오 디코더 (300) 에 의한 사용을 위해 인코딩된 비디오 데이터와 연관된 메타데이터를 기술하는 신택스 엘리먼트들에 대한 값들을 엔트로피 인코딩할 수도 있다.

[0062] CABAC 을 수행하기 위해, 비디오 인코더 (200) 는 송신될 심볼에 컨텍스트 모델 내의 컨텍스트를 할당할 수도 있다. 컨텍스트 (context) 는 예를 들어, 심볼의 이웃하는 값들이 제로 값인지 여부와 관련될 수도 있다. 확률 결정은 심볼에 할당된 컨텍스트에 기초할 수도 있다.

[0063] 비디오 인코더 (200) 는 신택스 데이터, 예컨대 블록-기반 신택스 데이터, 픽처-기반 신택스 데이터, 및 시퀀스-기반 신택스 데이터를, 비디오 디코더 (300) 에, 예를 들어, 픽처 헤더, 블록 헤더, 슬라이스 헤더, 또는 다른 신택스 데이터, 예컨대 시퀀스 파라미터 세트 (SPS), 픽처 파라미터 세트 (PPS), 또는 비디오 파라미터 세트 (VPS) 에서 추가로 생성할 수도 있다. 비디오 디코더 (300) 는 마찬가지로 대응하는 비디오 데이터를 디코딩하는 방법을 결정하기 위해 그러한 신택스 데이터를 디코딩할 수도 있다.

[0064] 이러한 방식으로, 비디오 인코더 (200) 는 인코딩된 비디오 데이터, 예를 들어, 픽처의 블록들 (예를 들어, CU들) 로의 파티셔닝을 기술하는 신택스 엘리먼트들 및 블록들에 대한 예측 및/또는 잔차 정보를 포함하는 비트스트림을 생성할 수도 있다. 궁극적으로, 비디오 디코더 (300) 는 비트스트림을 수신하고 인코딩된 비디오 데이터를 디코딩할 수도 있다.

[0065] 일반적으로, 비디오 디코더 (300) 는 비트스트림의 인코딩된 비디오 데이터를 디코딩하기 위해 비디오 인코더 (200) 에 의해 수행되는 것과 상반되는 프로세스를 수행한다. 예를 들어, 비디오 디코더 (300) 는 비디오 인코더 (200) 의 CABAC 인코딩 프로세스와 실질적으로 유사하지만, 상반되는 방식으로 CABAC 을 사용하여 비트스트림의 신택스 엘리먼트들에 대한 값들을 디코딩할 수도 있다. 신택스 엘리먼트들은 픽처의 CTU들로의 파티셔닝 정보, 및 QTBT 구조와 같은 대응하는 파티션 구조에 따른 각각의 CTU 의 파티셔닝을 정의하여, CTU 의 CU들을 정의할 수도 있다. 신택스 엘리먼트들은 비디오 데이터의 블록들 (예를 들어, CU들) 에 대한 예측 및 잔차 정보를 추가로 정의할 수도 있다.

[0066] 잔차 정보는 예를 들어 양자화된 변환 계수들에 의해 표현될 수도 있다. 비디오 디코더 (300) 는 블록에 대한 잔차 블록을 재생하기 위해 블록의 양자화된 변환 계수들을 역 양자화 및 역 변환할 수도 있다. 비디오 디코더 (300) 는 시그널링된 예측 모드 (인트라 또는 인터 예측) 및 관련된 예측 정보 (예를 들어, 인터 예측을 위한 모션 정보) 를 사용하여 블록에 대한 예측 블록을 형성한다. 비디오 디코더 (300) 는 그 후 예측 블록과 잔차 블록을 (샘플 별 기준으로) 조합하여 원래의 블록을 재생할 수도 있다. 비디오 디코더 (300) 는 블록의 경계들을 따라 시각적 아티팩트들을 감소시키기 위해 디블로킹 프로세스를 수행하는 것과 같은 추가적인 프로세싱을 수행할 수도 있다.

[0067] 본 개시는 일반적으로 신택스 엘리먼트들과 같은, 소정의 정보를 "시그널링" 하는 것을 참조할 수도 있다. 용어 "시그널링" 은 일반적으로 인코딩된 비디오 데이터를 디코딩하는데 사용되는 신택스 엘리먼트들에 대한 값들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 즉, 비디오 인코더 (200) 는 비트스트림에서 신택스 엘리먼트들에 대한 값들을 시그널링할 수도 있다. 일반적으로, 시그널링은 비트스트림에서 값을 생성하는 것을 지칭한다. 상기 언급된 바와 같이, 소스 디바이스 (102) 는 목적지 디바이스 (116) 에 의한 추후 취출을 위해 저장 디바이스 (112) 에 신택스 엘리먼트들을 저장할 때 발생할 수도 있는 바와 같은, 비실시간으로, 또는 실질적으로 실시간으로 비트스트림을 목적지 디바이스 (116) 로 전송할 수도 있다.

[0068] 도 2a 및 도 2b 는 예시적인 쿼드트리 바이너리 트리 (QTBT) 구조 (130), 및 대응하는 코딩 트리 유닛 (CTU) (132) 을 예시하는 개념도이다. 실선들은 쿼드트리 스플릿팅을 나타내고, 점선들은 바이너리 트리 스플릿팅을 나타낸다. 바이너리 트리의 각각의 스플릿팅된 (즉, 비-리프) 노드에서, 어느 스플릿팅 타입 (즉, 수평 또는 수직) 이 사용되는지를 나타내기 위해 하나의 플래그가 시그널링되며, 여기서 0 은 수평 스플릿팅을 표시하고 1 은 수직 스플릿팅을 표시한다. 쿼드트리 스플릿팅에 대해, 스플릿팅 타입을 표시할 필요는 없는데, 이는 쿼드트리 노드들이 동일한 사이즈를 가진 4 개의 서브-블록들로 수평으로 및 수직으로 블록을 스플릿팅하기 때문이다. 이에 따라, QTBT 구조 (130) 의 영역 트리 레벨 (즉, 실선들) 에 대한 신택스 엘리먼트들 (이를 테면 스플릿팅 정보) 및 QTBT 구조 (130) 의 예측 트리 레벨 (즉, 점선들) 에 대한 신택스 엘리먼트들 (이를 테면 스플릿팅 정보) 을, 비디오 인코더 (200) 가 인코딩할 수도 있고, 비디오 디코더 (300) 가 디코딩할 수도 있다. QTBT 구조 (130) 의 종단 리프 노드들에 의해 표현된 CU들에 대해, 예측 및 변환 데이터와 같은 비디오 데이터를, 비디오 인코더 (200) 가 인코딩할 수도 있고, 비디오 디코더 (300) 가 디코딩할 수도 있다.

[0069] 일반적으로, 도 2b 의 CTU (132) 는 제 1 및 제 2 레벨들에서 QTBT 구조 (130) 의 노드들에 대응하는 블록들의 사이즈들을 정의하는 파라미터들과 연관될 수도 있다. 이들 파라미터들은 CTU 사이즈 (샘플들에서 CTU (132) 의 사이즈를 나타냄), 최소 쿼드트리 사이즈 (MinQTSize, 최소 허용된 쿼드트리 리프 노드 사이즈를 나타냄), 최대 바이너리 트리 사이즈 (MaxBTSize, 최대 허용된 바이너리 트리 루트 노드 사이즈를 나타냄), 최대 바이너리 트리 심도 (MaxBTDepth, 최대 허용된 바이너리 트리 심도를 나타냄), 및 최소 바이너리 트리 사이즈 (MinBTSize, 최소 허용된 바이너리 트리 리프 노드 사이즈를 나타냄) 를 포함할 수도 있다.

[0070] CTU 에 대응하는 QTBT 구조의 루트 노드는 QTBT 구조의 제 1 레벨에서 4 개의 자식 노드들을 가질 수도 있고, 이들의 각각은 쿼드트리 파티셔닝에 따라 파티셔닝될 수도 있다. 즉, 제 1 레벨의 노드들은 리프 노드들 (자식 노드들이 없음) 이거나 또는 4 개의 자식 노드들을 갖는다. QTBT 구조 (130) 의 예는 그러한 노드들을 브랜치들에 대한 실선들을 갖는 자식 노드들 및 부모 노드를 포함하는 것으로서 나타낸다. 제 1 레벨의 노드들이 최대 허용된 바이너리 트리 루트 노드 사이즈 (MaxBTSize) 보다 크지 않으면, 노드들은 개별의 바이너리 트리들에 의해 추가로 파티셔닝될 수 있다. 하나의 노드의 바이너리 트리 스플릿팅은 스플릿으로부터 발생하는 노드들이 최소 허용된 바이너리 트리 리프 노드 사이즈 (MinBTSize) 또는 최대 허용된 바이너리 트리 심도 (MaxBTDepth) 에 도달할 때까지 반복될 수 있다. QTBT 구조 (130) 의 예는 그러한 노드들을 브랜치들에 대한 점선들을 갖는 것으로서 나타낸다. 바이너리 트리 리프 노드는, 임의의 추가 파티셔닝 없이, 예측 (예를 들어, 인트라 픽처 또는 인터 픽처 예측) 및 변환을 위해 사용되는 코딩 유닛 (CU) 으로 지칭된다. 상기 논의된 바와 같이, CU들은 또한, "비디오 블록들" 또는 "블록들" 로 지칭될 수도 있다.

[0071] QTBT 파티셔닝 구조의 하나의 예에서, CTU 사이즈는 128x128 (루마 샘플들 및 2 개의 대응하는 64x64 크로마 샘플들) 로서 설정되고, MinQTSize 는 16x16 으로서 설정되고, MaxBTSize 는 64x64 로서 설정되고, (폭 및 높이 양자 모두에 대한) MinBTSize 는 4 로서 설정되고, 그리고 MaxBTDepth 는 4 로서 설정된다. 쿼드트리 파티셔닝은 쿼드트리 리프 노드들을 생성하기 위해 먼저 CTU 에 적용된다. 쿼드트리 리프 노드들은 16x16 (즉, MinQTSize) 으로부터 128x128 (즉, CTU 사이즈) 까지의 사이즈를 가질 수도 있다. 리프 쿼드트리 노드가 128x128 인 경우, 사이즈가 MaxBTSize (즉, 이 예에서는 64x64) 를 초과하기 때문에 그것은 바이너리 트리에 의해 추가로 스플릿팅되지 않을 것이다. 그렇지 않으면, 리프 쿼드트리 노드는 바이너리 트리에 의해 추가로 파티셔닝될 것이다. 따라서, 쿼드트리 리프 노드는 또한 바이너리 트리에 대한 루트 노드이고 바이너리 트리 심도를 0 으로서 갖는다. 바이너리 트리 심도가 MaxBTDepth (이 예에서는 4) 에 도달할 때, 추가의 스플릿팅이 허용되지 않는다. 바이너리 트리 노드가 MinBTSize (이 예에서는 4) 와 동일한 폭을 가질 때, 그것은 추가의 수평 스플릿팅이 허용되지 않음을 암시한다. 유사하게, 높이가 MinBTSize 와 동일한 바이너리 트리 노드는 그 바이너리 트리 노드에 대해 추가의 수직 스플릿팅이 허용되지 않음을 암시한다. 상기 언급된 바와 같이, 바이너리 트리의 리프 노드들은 CU들로 지칭되고, 추가의 파티셔닝 없이 예측 및 변환에 따라 추가로 프로세싱된다.

[0072] TCQ (Trellis coded quantization) 는 H. Schwarz, T. Nguyen, D. Marpe, T. Wiegand, M. Karczewicz, M. Coban, J. Dong, “CE7: Transform coefficient coding with reduced number of regular-coded bins (tests 7.1.3a, 7.1.3b)”, JVET document JVET-L0274, Macao, CN, Oct 2018 (이하, JVET-L0274) 에서 제안되었다. JVET-L0274 의 기법들에서, 2 개의 스칼라 양자화기들이 양자화/역양자화를 위해 스위칭가능하게 사용된다. 현재 변환/양자화된 계수에 대해 사용되는 스칼라 양자화기는 스캐닝 순서에서 현재 변환/양자화된 계수에 선행하는 양자화된 계수의 패리티 (최하위 비트) 에 의해 결정된다.

[0073] TCQ 와 결합된 계수 코딩 방식이 또한 JVET-L0274 에서 제안되었으며, 이에 의해 양자화된 계수를 디코딩하기 위한 컨텍스트 선택은 사용된 양자화기에 의존한다. 구체적으로, 계수가 제로 또는 비-제로인 것을 표시하는 계수의 유의성 플래그 (significance flag; SIG) 는 컨텍스트 모델들의 3 개의 세트들을 가지며, 특정 SIG 에 대해 선택된 세트는 연관된 계수에 대해 사용된 양자화기에 의존한다. 따라서, 현재 계수의 SIG 를 디코딩하기 시작할 때, 엔트로피 디코더는 이전 스캐닝 포지션에서의 계수의 패리티를 알아야 하며, 이는 현재 계수에 대한 양자화기 및 따라서 그 계수의 SIG 에 대한 컨텍스트 세트를 결정한다.

[0074] TU 는 코딩 그룹 (CG) 이라 불리는 비-중첩 서브블록들로 분할되며, 그 사이즈는 보통 4x4이다. 본 명세서에 설명된 디코딩 프로세스는 때때로 4x4 CG에 대해 설명될 수도 있지만, 임의의 다른 CG 사이즈들로 쉽게 확장될 수 있다. 본 개시의 기법들, 및 따라서 여기에 포함된 설명은 주로 CG 에서의 계수의 절대 레벨에 대한 인코딩 및 디코딩 프로세스들에 관한 것이다. 부호들과 같은 CG 와 연관된 다른 정보는 JVET-L0274 에서 설명된 방식으로 인코딩 또는 디코딩될 수도 있지만, 또한 대안적인 기법들을 사용하여 인코딩 및 디코딩될 수도

있다.

[0075] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 비트스트림들에서 신택스 엘리먼트들을 프로세싱하도록 구성될 수도 있다. 예를 들어, 계수에 대한 절대 레벨 값 (absLevel) 을 표현하기 위해 다음과 같은 신택스 엘리먼트들이 사용될 수도 있다.

[0076] · sig_coeff_flag: absLevel 이 0 이면 이 플래그는 0 과 같고; 그렇지 않으면, 플래그는 1 과 같다.

[0077] · abs_level_gt1_flag: sig_coeff_flag 가 1 과 같으면, 플래그가 비트스트림에 존재한다. absLevel 이 1 보다 크면 그것은 1 과 같고; 그렇지 않으면, 플래그는 0 과 같다.

[0078] · par_level_flag: rem_abs_gt1_flag 가 1 과 같으면, 플래그가 비트스트림에 존재한다. absLevel 이 홀수이면 그것은 0 과 같고, absLevel 이 짝수이면 그것은 1 과 같다.

[0079] · abs_level_gt3_flag: abs_level_gt1_flag 가 1 과 같으면, 플래그가 비트스트림에 존재한다. absLevel 이 3보다 크면 그것은 1 과 같고; 그렇지 않으면, 플래그는 0 과 같다.

[0080] · abs_remainder: abs_level_gt3_flag 가 1 과 동일하면, 이 신택스 엘리먼트는 비트스트림에 존재한다. 그것은 골롬-라이스 코드로 코딩된 변환 계수 레벨의 잔여 절대 값이다.

[0081] · abs_level: 이것은 골롬-라이스 코드로 코딩된 변환 계수 레벨의 절대값이다.

[0082] 이하, 설명의 간략화를 위해 신택스 엘리먼트들 sig_coeff_flag, par_level_flag, abs_level_gt1_flag, abs_level_gt3_flag, abs_remainder, 및 abs_level 은 각각 SIG, Par, Gt1, Gt2, remLevel, absLevel 로서 표시된다.

[0083] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 비트스트림으로부터 파싱되지 않은 상기 신택스 엘리먼트들 중 임의의 것을 0 과 같은 디폴트 값으로 설정하도록 구성될 수도 있다. 5 개의 신택스 엘리먼트들 중 제 1 신택스 엘리먼트의 값들이 주어지면, 계수의 절대 레벨에 대한 값은 다음과 같이 계산될 수 있다:

[0084] absoluteLevel = SIG + Gt1 + Par + (Gt2 << 1) + (remLevel << 1) (1)

[0085] 대안적으로, 계수가 완전히 바이패스 코딩된 모드로 코딩되면, absoluteLevel 은 abs_level 로서 직접 코딩될 수도 있다.

[0086] 도 3 은 JVET-L0274 에서와 같이 CG 에서 absoluteLevels 를 나타내는 신택스 엘리먼트들에 대한 예시적인 순서를 나타낸다. 다른 순서들이 또한 사용될 수도 있다. 알 수 있는 바와 같이, 모든 5 개의 신택스 엘리먼트들은 absLevel 이 4 보다 클 때 비트스트림으로부터 파싱된다.

[0087] 도 3 의 예에서, 비디오 디코더 (300) 는 4 개까지의 패스들에서 CG 에서의 포지션들을 스캔한다. 제 1 패스 (136) 에서, 비디오 디코더 (300) 는 SIG들, Par들, 및 Gt1들에 대한 값들을 파싱한다. 비-제로 SIG 들만이 대응하는 Gt1들 및 Par들이 뒤따른다. 즉, 비디오 디코더 (300) 는 SIG 가 제로의 값을 갖는 것으로 결정하면, 이는 계수 레벨이 제로와 동일함을 의미하며, 비디오 디코더 (300) 는 Gt1 의 인스턴스들을 수신하지 않고 그 계수에 대해 par 를 수신한다. 제 1 패스 (136) 후에, 식 (2) 에 나타낸 바와 같이, 각각의 포지션에 대해 absLevel1 로서 표시된 부분 absoluteLevel 에 대한 값이 재구성될 수도 있다.

[0088] absLevel1 = SIG + Par + Gt1 (2)

[0089] 일부 구현들에서, 비디오 디코더 (300) 는 4x4 서브블록들에 대한 제 1 패스 (136) 에서 최대 28 개의 정규 코딩된 빈들 및 2x2 서브블록들에 대한 최대 6 개의 정규 코딩된 빈들을 파싱하도록 구성될 수도 있다. 정규 코딩된 빈들의 수에 대한 제한들은 SIG, Gt1, Par 빈들의 그룹들에서 강제될 수도 있으며, 이는 SIG, Gt1, 및 Par 빈들의 각각의 그룹이 세트로서 코딩되고 세트의 중간에서 바이패스 코딩으로의 스위칭이 허용되지 않음을 의미한다.

[0090] 제 1 패스에 적어도 하나의 비-제로 Gt1 이 존재하면, 비디오 디코더 (300) 는 제 2 패스 (138) 를 스캔하도록 구성될 수도 있다. 제 2 패스 (138) 에서, 비디오 디코더 (300) 는 비-제로 Gt1들을 갖는 포지션들에 대해 Gt2들을 파싱한다. 제 1 패스 (136) 및 제 2 패스(138) 내의 빈들은 모두 정규 코딩될 수도 있으며, 이는 빈의 확률 분포가 선택된 컨텍스트 모델에 의해 모델링됨을 의미한다. 제 2 패스 (138) 에 적어도 하나의 비-제로 Gt2 가 존재하면, 비디오 디코더 (300) 는 제 3 패스 (140) 를 스캔한다. 제 3 패스 (140) 동안, 비디오 디코더 (300) 는 비-제로 Gt2들을 갖는 포지션들의 remLevel들을 파싱한다. remLevel 은 바이너리가

아니며, 비디오 디코더 (300) 는 rem 의 이진화된 버전의 빈들을 바이패스-코딩할 수도 있는데, 이는 빈들이 균일하게 분포되는 것으로 가정되고 아무런 컨텍스트 선택도 필요하지 않다는 것을 의미한다.

[0091] 제 4 패스 (142) 에서, 비디오 디코더 (300) 는 이전의 3 개의 패스들에서 정규 코딩된 빈들로 부분적으로 표현되지 않은, 모든 잔여 계수들을 스캔한다. 추가 패스 (142) 의 계수들 레벨들은 바이패스 코딩된 빈들을 사용하여 절대 값들로서 코딩된다.

[0092] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 컨텍스트 모델링을 수행할 수도 있다. JVET-L0274에서 사용된 컨텍스트 모델링은 또한 본 개시에 의해 제안된 변형들과 함께 여기에 간략하게 소개된다. 아래에서 더 상세히 논의되는 컨텍스트 모델링은 일반적으로 빈-투-디코드 (bin-to-decode) 를 위한, 컨텍스트들로도 지칭되는, 확률 모델들의 선택을 지칭한다. JVET-L0274 에서, 신택스 엘리먼트들 SIG, Par, Gt1, 및 Gt2 는 컨텍스트 모델링을 사용하여 코딩된다. 컨텍스트의 선택은, N 으로서 표시된, 로컬 이웃 (local neighborhood) 에서의 absLevel1 의 값들에 의존한다. 도 4 는 사용되는 이웃의 템플릿을 나타낸다. 템플릿 내부의, 하지만 현재 TU 외부의 그 포지션들은 N 으로부터 배제될 수도 있다.

[0093] 도 4 는 확률 모델들을 선택하기 위해 사용되는 템플릿의 예시를 나타낸다. "X" 로 마킹된 사각형은 현재 스캔 포지션을 지정하고, "Y" 로 마킹된 사각형은 사용된 로컬 이웃을 나타낸다.

[0094] 현재 포지션 (도 4 에서 X 를 갖는 정사각형 참조) 에 대해, 비디오 디코더 (300) 는 ctxIdxSIG, ctxIdxPar, ctxIdxGt1, 및 ctxIdxGt2 로 표시된, 그것의 SIG, Par, Gt1, 및 Gt2 의 컨텍스트 인덱스들을 결정한다. 컨텍스트 인덱스들을 결정하기 위해, 비디오 디코더 (300) 는 3 개의 변수들 - numSIG, sumAbs1, 및 d 를 먼저 결정할 수도 있다. 변수 numSIG 는 N 에서 비-제로 SIG들의 수를 나타내며, 이는 아래 식 (3) 에 의해 표현된다.

[0095] $$\mathbf{sumSIG} = \sum_{i \in N} \mathbf{SIG}(i) \quad (3)$$

[0096] 변수 sumAbs1 은 N 에서 absLevel1 의 합을 나타내며, 이는 아래 식 4 에 의해 표현된다.

[0097] $$\mathbf{sumAbs1} = \sum_{i \in N} \mathbf{absLevel1}(i) \quad (4)$$

[0098] 변수 d 는 아래의 식 (5) 에 의해 표현되는 바와 같이, TU 내부의 현재 포지션의 대각선 측정치를 나타낸다:

[0099] d = x + y (5)

[0100] 여기서, x 및 y 는 TU 내의 현재 포지션의 좌표들을 나타낸다.

[0101] sumAbs1 및 d 가 주어지면, 비디오 디코더 (300) 는 다음과 같이 SIG 를 디코딩하기 위한 컨텍스트 인덱스를 결정한다:

[0102] • 루마에 대해, ctxIdxSIG 는 식 (6) 에 의해 결정된다:

[0103] ctxIdxSIG = 18*max(0, state-1) + min(sumAbs1, 5) + (d < 2 ? 12 : (d < 5 ? 6 : 0)) (6)

[0104] • 크로마에 대해, ctxIdxSIG는 식 (7) 에 의해 결정된다:

[0105] ctxIdxSIG = 12 * max(0, state-1) + min(sumAbs1, 5) + (d < 2 ? 6 : 0)) (7)

[0106] 식 (6) 및 (7) 에서, 변수 "state" 는 JVET-L0274 에서 정의된 바와 같은 상태 머신의 현재 상태를 나타낸다.

[0107] sumSIG, sumAbs1 및 d 가 주어지면, 비디오 디코더 (300) 는 다음과 같이 Par 를 디코딩하기 위한 컨텍스트 인덱스를 결정한다:

[0108] • 현재 스캔 포지션이 마지막 비-제로 계수의 포지션과 동일하면, ctxIdxPar 은 0 이다.

[0109] • 그렇지 않으면

[0110] o 루마에 대해, ctxIdxPar 은 식 (8) 에 의해 결정된다:

[0111] ctxIdxPar = 1 + min(sumAbs1 - numSIG, 4) + (d == 0 ? 15 : (d < 3 ? 10 : (d < 10 ? 5 : 0))) (8)

[0112] o 크로마에 대해, ctxIdxPar 은 (9) 에 의해 결정된다

[0113] ctxIdxPar = 1 + min(sumAbs1 - numSIG, 4) + (d == 0 ? 5 : 0) (9)

[0114] ctxIdxGt1 및 ctxIdxGt2 는 ctxIdxPar 의 값으로 설정된다.

[0115] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 RemLevel 코딩을 수행하도록 구성될 수도 있다. 비디오 디코더 (300) 는 다음과 같이 비-바이너리 신택스 엘리먼트 remRemainder (remLevel) 및 absLevel 을 코딩하기 위한 라이스 파라미터 (ricePar) 를 도출한다:

[0116] • 각 서브블록의 시작에서, ricePar 은 0 과 동일하게 설정된다;

[0117] • 신택스 엘리먼트 나머지를 코딩한 후, 라이스 파라미터 (ricePar) 는 다음과 같이 수정된다:

[0118] ricePar 가 3 미만이고 나머지의 최종 코딩된 값이 ((3 < ricePar) - 1) 보다 크면, ricePar 은 1 만큼 증분된다.

[0119] 완전히 바이패스-코딩되는 절대 양자화 인덱스들을 나타내는 비-바이너리 신택스 엘리먼트 absLevel 을 코딩하기 위해, 다음이 적용된다:

[0120] • 로컬 템플릿 내의 절대 값들의 합 sumAbs 이 결정된다.

[0121] • 변수들 ricePar 및 posZero 는,

[0122] ricePar = riceParTable[min(31, sumAbs)]

[0123] posZero = posZeroTable[max(0, state - 1)][min(31, sumAbs)]

[0124] 에 따라 테이블 룩업에 의해 결정되고, 여기서, 변수 상태는 종속 양자화를 위한 상태를 나타내고 (그것은 종속 양자화가 디스에이블될 때 0 과 동일함),

[0125] riceParTable[] 및 posZeroTable[]] 은,

[0126] riceParTable[32] = {

[0127] 0,0,0,0,0,0,0,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3

[0128] };

[0129] posZeroTable[3][32] = {

[0130] {0,0,0,0,0,1,2,2,2,2,2,2,4,4,4,4,4,4,4,4,4,4,8,8,8,8,8,8,8,8},

[0131] {1,1,1,1,2,3,4,4,4,6,6,6,8,8,8,8,8,12,12,12,12,12,12,12,12,16,16,16,16,16,16},
{1,1,2,2,2,3,4,4,4,6,6,6,8,8,8,8,8,8,12,12,12,12,12,12,12,16,16,16,16,16,16,16}

[0132] };

[0133] 에 의해 주어진다.

[0134] • 중간 변수 codeValue 는 다음과 같이 도출된다:

[0135] o absLevel 이 0 과 동일한 경우, codeValue 는 posZero 와 동일하게 설정된다;

[0136] o 그렇지 않고, absLevel posZero 보다 작거나 같으면, codeValue 는 absLevel - 1 과 같도록 설정된다;

[0137] o 그렇지 않으면 (absLevel 이 posZero 보다 큼), codeValue 는 absLevel 과 동일하게 설정된다.

[0138] • codeValue 의 값은 라이스 파라미터 ricePar 로 골롬-라이스 코드를 사용하여 코딩된다.

[0139] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 absoluteLevel 재구성을 수행하도록 구성될 수도 있다. absoluteLevel 재구성은 JVET-L0274 에서와 동일할 수도 있으며, 이는 비트스트림에서의 신택스 엘리먼트들에 대해 위에서 논의되었다.

[0140] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 인터리빙된 방식으로 Gt2 플래그들을 코딩하도록 구성될 수도 있다. 일부 예들에서, 제 1 패스에서, SIG, Gt1, Par 플래그들이 코딩되고 제 2 패스에서, Gt2 플래그들이

코딩되는 설명된 방식 대신에, Gt2 플래그들은 아래의 도면들에서 나타낸 바와 같이 Par 플래그 후에 또는 Gt1 플래그 후에 제1 패스에 통합될 수 있고, 이는 코딩 패스들을 4 로부터 3 으로 감소시킨다.

[0141] 도 5 는 Par 플래그 후의 제 1 패스에서의 인터리빙된 Gt2 플래그의 예를 나타낸다. 도 5 와 관련하여, 비디오 디코더 (300) 는 도 3 과 관련하여 위에서 설명된 것과 동일한 방식으로 absLevel1 에 대한 값을 결정할 수도 있지만, 다양한 신택스 엘리먼트들이 수신되는 순서는 변경된다. 예를 들어, 도 5 에서, 비디오 디코더 (300) 는 Gt2 에 대한 값들을 제 2 패스 (예를 들어, 도 3 에서의 제 2 패스 (138)) 의 부분 대신에 제 1 패스 (162) 의 부분으로서 결정한다. 따라서, 도 5 에서, 도 3 의 제 1 패스 (136) 및 제2 패스 (138) 는 단일 패스 (제 1 패스 (162))로 효과적으로 결합되고, 도 3 의 제3 패스 (140) 및 제 4 패스 (142) 는 각각 도 5 의 제 2 패스 (164) 및 제 3 패스 (166) 가 된다. 따라서, 도 5 의 예에서, 모든 신택스 엘리먼트들을 전달하기 위해 단지 3 개의 패스들이 필요하다.

[0142] 도 6 은 Gt1 플래그 후의 제 1 패스에서의 인터리빙된 Gt2 플래그의 예를 나타낸다. 이 경우 absLevel1 은 다음과 같이 계산될 수 있고:

[0143] absLevel1 = SIG + Par + Gt1 + (Gt2<<1)

[0144] 컨텍스트 모델링과 관련하여 위에서 소개한 식들에서 컨텍스트 도출에 사용될 수 있다. 도 6 과 관련하여, 비디오 디코더 (300) 는 도 3 과 관련하여 위에서 설명된 것과 동일한 방식으로 absLevel1 에 대한 값을 결정할 수도 있지만, 다양한 신택스 엘리먼트들이 수신되는 순서는 변경된다. 예를 들어, 도 6 에서, 비디오 디코더 (300) 는 Gt2 에 대한 값들을 제 2 패스 (예를 들어, 도 3 에서의 제 2 패스 (138)) 의 부분 대신에 제 1 패스 (172) 의 부분으로서 결정한다. 따라서, 도 6 에서, 도 3 의 제 1 패스 (136) 및 제 2 패스 (138) 는 단일 패스 (제 1 패스 (172)) 로 효과적으로 결합되고, 도 3 의 제 3 패스 (140) 및 제 4 패스 (142) 는 각각 도 6 의 제 2 패스 (174) 및 제 3 패스 (176) 가 된다. 따라서, 도 6 의 예에서, 모든 신택스 엘리먼트들을 전달하기 위해 단지 3 개의 패스들이 필요하다. 도 6 에서, 제 1 패스 (172) 의 신택스 엘리먼트들은 도 5 의 제1 패스 (162) 의 신택스 요소들과는 상이한 순서로 스캔되지만, 다른 패스들은 일반적으로 동일하다.

[0145] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 부분적인 최종 정규 빈 코딩된 계수 표현을 이용하도록 구성될 수도 있으며, 여기서 일부 계수들에 대한 값들은 바이패스 코딩을 이용하여 전달된 잔여 값과 함께 정규 코딩된 빈들을 이용하여 부분적으로 전달될 수도 있다. JVET-L0274 에서 설명된 코딩 방식에서, 제 1 코딩 패스에 대한 정규 코딩된 빈 버짓이 도달되는 마지막 정규 빈 코딩된 계수 (예를 들어, 도 3 의 Coeff K), SIG, Gt1, Par 빈들은 모두 정규 코딩된 빈들로서 코딩된다. 정규 빈 코딩은 SIG-Gt1-Par 그룹의 중간에서 종결되지 않는다. 유사하게, SIG-Gt1-Par-Gt2 그룹 또는 SIG-Gt1-Gt2-Par 그룹 (예를 들어, 도 5 및 도 6) 에 대해, Coeff K 의 SIG, Gt1, Par, Gt2 플래그들에 대한 코딩은 정규 모드에서 코딩된다. 본 개시는 도 7 및 도 8 에 도시된 바와 같이 SIG 및 Gt1 플래그들의 코딩 후에 정규 코딩된 빈들의 가능한 종결을 허용함으로써 이러한 제약을 깨기 위한 기법들을 제안한다.

[0146] 도 7 은 제 1 코딩 패스 (182) 에서 SIG-Gt1-Par-Gt2 코딩에 대해 정규 코딩된 빈 한계가 도달되는 마지막 계수의 부분 코딩의 예를 나타낸다. 도 7 의 예에서, 비디오 디코더 (300) 는 remLevel 값들 및 absLevel 값들 양자를 포함하는 제 3 패스 (186) 를 스캔한다. remLevel 에 대한 값은 계수에 대한 실제 값과 제1 패스 (182) 및 제 2 패스 (184) 로부터 결정된 부분 값 사이의 잔여 값을 나타낸다. absLevel 에 대한 값은, 반대로, 계수 값의 절대값을 나타낸다.

[0147] 도 8 은 제 1 코딩 패스 (192) 에서 SIG-Gt1-Gt2-Par 코딩에 대해 정규 코딩된 빈 한계가 도달되는 마지막 계수의 부분 코딩의 예를 나타낸다. 도 8 에서, 제 1 패스 (192) 의 신택스 엘리먼트들은 도 7 의 제 1 패스 (182) 의 신택스 엘리먼트들과는 상이한 순서로 스캔된다. 제 2 패스 (194) 및 제3 패스 (196) 는 도 7 의 제 2 패스 (184) 및 제 3 패스 (186) 와 일반적으로 동일하다.

[0148] 도 7 및 도 8 의 예들에서, Coeff K 의 잔여 레벨은 바이패스 코딩되는 absLevel 에 대한 값들과 함께 제 3 패스 (186/196) 에서 바이패스 코딩되는 remLevelFull 로서 코딩된다. 계수에 대한 값은 다음과 같이 표현된다:

[0149] absoluteLevel = SIG + Gt1 + remLevelFull,

[0150] 또는

[0151] absoluteLevel = SIG + remLevelFull.

[0152] 다른 예들에서, 빈들의 정규 코딩은 Par 및 Gt2 플래그들의 코딩 후에 종료될 수 있거나, 또는 그 반대일 수 있다. 이 경우, 마지막 계수의 잔여 레벨은 잔여 레벨의 절반으로서 코딩될 것이다, 즉,

[0153] absoluteLevel = SIG +GT1 + Par + (remLevel << 1),

[0154] 또는

[0155] absoluteLevel = SIG +GT1 + (GT2 << 1) + (remLevel << 1).

[0156] 정규 코딩된 빈들의 총 수는 인터리빙된 SIG, Gt1, Gt2 및 Par 플래그들에 부과되는 총 수로서 규정될 수도 있다.

[0157] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 잔여 레벨 코딩을 수행하도록 구성될 수도 있다. 제 2 코딩 패스에서의 remLevel 코딩은 RemLevel 코딩에 대해 위에서 설명된 것과 동일할 수도 있다. 비디오 디코더 (300) 는 Coeff K-1 의 종료까지 라이스 파라미터 업데이트 및 도출을 수행할 수도 있으며, 여기서, Coeff K-1 은 마지막 정규 코딩된 계수 (Coeff K) 전의 제 2 내지 마지막 정규 코딩된 계수를 나타낸다. 비디오 디코더 (300) 는 완전히 정규 코딩을 사용하여 Coeff K-1 을 디코딩할 수도 있고, 완전히 정규 코딩을 사용하여 또는 정규 코딩과 바이패스 코딩의 조합을 사용하여 Coeff K 를 디코딩할 수도 있다. Coeff K 의 remLevelFull 의 코딩을 위해, 비디오 디코더 (300) 는 다음과 같이 라이스 파라미터를 업데이트할 수도 있다:

[0158] riceParBypass = 2 x ricePar + lastCodedGt2Flag,

[0159] riceParBypass = riceParBypass == 1 ? riceParBypass - 1 : riceParBypass

[0160] 여기서 ricePar 은 제 2 패스에서 remLevel 의 코딩을 위해 사용되는 ricePar 이고, lastCodedGt2Flag 는 제 1 코딩 패스에서 마지막 코딩된 Gt2 플래그의 값이다. 대안적으로, 2 x ricePar 인 ricePar 에 대한 값이 사용될 수 있거나, 또는 Coeff K 에 대한 잔여 레벨의 최적 코딩에 매칭하는 ricePar 이 사용될 수 있다.

[0161] 일부 예들에서, Coeff K 의 remLevelFull 의 코딩을 위해, 비디오 디코더 (300) 는 다음과 같이 라이스 파라미터를 업데이트할 수도 있다:

[0162] 1- riceParBypass = min(ricePar>0 ? ricePar + 1 : lastCodedGt2Flag, 3)

[0163] 2- riceParBypass = min(2*ricePar + lastCodedGt2Flag, 3)

[0164] 3- riceParBypass = min(2*ricePar, 3)

[0165] 바이패스 코딩을 사용하여 완전히 코딩되는 계수들에 대한 absLevel 값들의 나머지에 대해, 비디오 디코더 (300) 는 다음과 같이 riceParBypass 를 업데이트할 수도 있다. 바이패스 코딩된 계수를 코딩하기 전에, 비디오 디코더 (300) 는 다음과 같이 riceParBypass 를 업데이트한다:

[0166] if (riceParBypass < 3 && absoluteLevelPrevCoeff > ((3<<riceParBypass)-1) { riceParBypass++;}

[0167] remLevel 대신에 이전 코딩된 계수 (Coeff K) 의 완전한 절대 값이 임계치 체크를 위해 사용된다는 점을 제외하고 ricePar 가 remLevel 코딩을 위해 업데이트되는 방식과 유사하다.

[0168] 비디오 디코더 (300) 는 임의의 다양한 상이한 기법들의 absLevel 레벨을 결정하기 위해 posZero 파라미터를 도출할 수도 있다. 하나의 예에서, 비디오 디코더 (300) 는 다음과 같은 룩업 테이블을 사용하여 absLevel 레벨을 결정하기 위한 posZero 파라미터를 도출할 수도 있다:

[0169] posZero = posZeroTableBypass[max(0, state - 1)][riceParBypass]

[0170] posZeroTableBypass [3][4]={{ 1, 2, 4, 8 },{ 3, 6, 12, 16 },{ 4, 6, 12, 16 }};

[0171] 비디오 디코더 (300) 는 다음과 같이 코딩될 중간 변수 codeValue 를 도출할 수도 있다:

[0172] o absLevel 또는 remLevelFull 이 0 과 동일한 경우, codeValue 는 posZero 와 동일하게 설정된다;

[0173] o 그렇지 않고, absLevel 또는 remLevelFull 이 posZero 보다 작거나 같으면, codeValue 는 각각 absLevel - 1 또는 remLevelFull - 1 과 동일하게 설정된다.

[0174] o 그렇지 않으면 (absLevel 또는 remLevelFull 이 posZero 보다 큼), codeValue 는 각각 absLevel 또는 remLevelFull 과 동일하게 설정된다.

[0175] 비디오 디코더 (300) 는 라이스 파라미터 riceParBypass 로 골롬-라이스 코드를 사용하여 codeValue 의 값을 코딩할 수도 있다.

[0176] 도 9 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코더 (200) 를 나타내는 블록도이다. 도 9 는 설명의 목적으로 제공되며 본 개시에 폭넓게 예시되고 기재되는 바와 같이 기법들을 제한하는 것으로 고려되지 않아야 한다. 설명의 목적으로, 본 개시는 HEVC 비디오 코딩 표준 및 개발 중인 H.266 비디오 코딩 표준과 같은 비디오 코딩 표준들의 맥락에서 비디오 인코더 (200) 를 설명한다. 그러나, 본 개시의 기법들은 이들 비디오 코딩 표준들에 한정되지 않으며, 일반적으로 비디오 인코딩 및 디코딩에 적용가능하다.

[0177] 도 9 의 예에서, 비디오 인코더 (200) 는 비디오 데이터 메모리 (230), 모드 선택 유닛 (202), 잔차 생성 유닛 (204), 변환 프로세싱 유닛 (206), 양자화 유닛 (208), 역 양자화 유닛 (210), 역 변환 프로세싱 유닛 (212), 재구성 유닛 (214), 필터 유닛 (216), 디코딩된 픽처 버퍼 (DPB) (218), 및 엔트로피 인코딩 유닛 (220) 을 포함한다.

[0178] 비디오 데이터 메모리 (230) 는, 비디오 인코더 (200) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 인코더 (200) 는 예를 들어, 비디오 소스 (104) (도 1) 로부터 비디오 데이터 메모리 (230) 에 저장된 비디오 데이터를 수신할 수도 있다. DPB (218) 는 비디오 인코더 (200) 에 의한 후속 비디오 데이터의 예측에 사용하기 위해 레퍼런스 비디오 데이터를 저장하는 레퍼런스 픽처 메모리로서 작용할 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동기식 동적 랜덤 액세스 메모리 (SDRAM) 를 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (230) 는 예시된 바와 같이 비디오 인코더 (200) 의 다른 컴포넌트들과 온-칩이거나, 또는 그 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0179] 본 개시에서, 비디오 데이터 메모리 (230) 에 대한 참조는 이처럼 구체적으로 기재되지 않으면 비디오 인코더 (200) 내부의 메모리 또는 이처럼 구체적으로 기재되지 않으면 비디오 인코더 (200) 외부의 메모리로 제한되는 것으로 해석되지 않아야 한다. 오히려, 비디오 데이터 메모리 (230) 에 대한 참조는 비디오 인코더 (200) 가 인코딩을 위해 수신하는 비디오 데이터 (예를 들어, 인코딩될 현재 블록에 대한 비디오 데이터) 를 저장하는 레퍼런스 메모리로서 이해되어야 한다. 도 1 의 메모리 (106) 는 또한 비디오 인코더 (200) 의 다양한 유닛들로부터의 출력들의 일시적 저장을 제공할 수도 있다.

[0180] 도 9 의 다양한 유닛들은 비디오 인코더 (200) 에 의해 수행되는 동작들의 이해를 돕기 위해 도시된다. 이 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하며, 수행될 수 있는 동작들에 대해 미리설정된다. 프로그래밍가능 회로들은 다양한 태스크들을 수행하도록 프로그래밍될 수 있는 회로들을 지칭하고, 수행될 수 있는 동작들에서 유연한 기능성을 제공한다. 예를 들어, 프로그래밍가능 회로들은, 프로그래밍가능 회로들이 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 (예를 들어, 파라미터들을 수신하거나 또는 파라미터들을 출력하기 위해) 소프트웨어 명령들을 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작들의 타입들은 일반적으로 불변이다. 일부 예들에서, 유닛들 중 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그래밍가능) 일 수도 있고, 일부 예들에서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.

[0181] 비디오 인코더 (200) 는 프로그래밍가능 회로들로부터 형성된, 산술 논리 유닛들 (arithmetic logic unit; ALU 들), 기본 함수 유닛들 (elementary function unit; EFU들), 디지털 회로들, 아날로그 회로들, 및/또는 프로그래밍가능 코어들을 포함할 수도 있다. 비디오 인코더 (200) 의 동작들이 프로그래밍가능 회로들에 의해 실행되는 소프트웨어에 의해 수행되는 예들에서, 메모리 (106) (도 1) 는 비디오 인코더 (200) 가 수신하고 실행하는 소프트웨어의 오브젝트 코드를 저장할 수도 있거나 또는 비디오 인코더 (200) 내의 다른 메모리 (미도시) 가 이러한 명령들을 저장할 수도 있다 .

[0182] 비디오 데이터 메모리 (230) 는 수신된 비디오 데이터를 저장하도록 구성된다. 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 비디오 데이터의 픽처를 취출하고 비디오 데이터를 잔차 생성 유닛 (204) 및 모드 선택 유닛 (202) 에 제공할 수도 있다. 비디오 데이터 메모리 (230) 에서의 비디오 데이터는 인코딩될 원시 비디오 데이터일 수도 있다.

[0183] 모드 선택 유닛 (202) 은 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라-예측 유닛 (226) 을 포함한다. 모드 선택 유닛 (202) 은 다른 예측 모드들에 따라 비디오 예측을 수행하기 위해 부가적인 기능 유닛들을 포함할 수도 있다. 예를 들어, 모드 선택 유닛 (202) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 추정 유닛 (222) 및/또는 모션 보상 유닛 (224) 의 일부일 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다.

[0184] 모드 선택 유닛 (202) 은 일반적으로 인코딩 파라미터들의 조합들 및 그러한 조합들에 대한 결과의 레이트-왜곡 값들을 테스트하기 위해 다중 인코딩 패스들을 조정한다. 인코딩 파라미터들은 CTU들의 CU들로의 파티셔닝, CU들에 대한 예측 모드들, CU들의 잔차 데이터에 대한 변환 타입들, CU들의 잔차 데이터에 대한 양자화 파라미터들 등을 포함할 수도 있다. 모드 선택 유닛 (202) 은 궁극적으로 다른 테스트된 조합들보다 우수한 레이트-왜곡 값들을 갖는 인코딩 파라미터들의 조합을 선택할 수도 있다.

[0185] 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 취출된 픽처를 일련의 CTU들로 파티셔닝하고, 슬라이스 내에 하나 이상의 CTU들을 캡슐화할 수도 있다. 모드 선택 유닛 (202) 은 상기 설명된 HEVC 의 쿼드트리 구조 또는 QTBT 구조와 같은, 트리 구조에 따라 픽처의 CTU 를 파티셔닝할 수도 있다. 상기 설명된 바와 같이, 비디오 인코더 (200) 는 트리 구조에 따라 CTU 를 파티셔닝하는 것으로부터 하나 이상의 CU들을 형성할 수도 있다. 그러한 CU 는 일반적으로 "비디오 블록" 또는 "블록" 으로도 또한 지칭될 수도 있다.

[0186] 일반적으로, 모드 선택 유닛 (202) 은 또한 그것의 컴포넌트들 (예를 들어, 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라 예측 유닛 (226)) 을 제어하여 현재 블록 (예를 들어, 현재 CU, 또는 HEVC 에서, PU 및 TU 의 오버랩하는 부분) 에 대한 예측 블록을 생성한다. 현재 블록의 인터 예측을 위해, 모션 추정 유닛 (222) 은 하나 이상의 레퍼런스 픽처들 (DPB (218) 에 저장된 하나 이상의 이전에 코딩된 픽처들) 에서 하나 이상의 밀접하게 매칭하는 레퍼런스 블록들을 식별하기 위해 모션 탐색을 수행할 수도 있다. 특히, 모션 추정 유닛 (222) 은, 예를 들어, 절대 차이의 합 (SAD), 제곱 차이들의 합 (SSD), 평균 절대 차이 (MAD), 평균 제곱 차이들 (MSD) 등에 따라, 잠재적 레퍼런스 블록이 현재 블록에 얼마나 유사한지를 나타내는 값을 계산할 수도 있다. 모션 추정 유닛 (222) 은 일반적으로 고려되는 레퍼런스 블록과 현재 블록 사이의 샘플 별 차이들을 사용하여 이들 계산들을 수행할 수도 있다. 모션 추정 유닛 (222) 은 현재 블록과 가장 근접하게 매칭하는 레퍼런스 블록을 표시하는, 이러한 계산들로부터 야기되는 최저 값을 갖는 레퍼런스 블록을 식별할 수도 있다.

[0187] 모션 추정 유닛 (222) 은 현재 픽처에서의 현재 블록의 포지션에 대한 레퍼런스 픽처들에서의 레퍼런스 블록들의 포지션들을 정의하는 하나 이상의 모션 벡터 (MV) 들을 형성할 수도 있다. 모션 추정 유닛 (222) 은 그 후 모션 벡터들을 모션 보상 유닛 (224) 에 제공할 수도 있다. 예를 들어, 단방향 인터-예측에 대해, 모션 추정 유닛 (222) 은 단일 모션 벡터를 제공할 수도 있는 반면, 양방향 인터-예측에 대해, 모션 추정 유닛 (222) 은 2 개의 모션 벡터들을 제공할 수도 있다. 그 후, 모션 보상 유닛 (224) 은 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 예를 들어, 모션 보상 유닛 (224) 은 모션 벡터를 사용하여 레퍼런스 블록의 데이터를 취출할 수도 있다. 다른 예로서, 모션 벡터가 분수 샘플 정밀도를 갖는다면, 모션 보상 유닛 (224) 은 하나 이상의 보간 필터들에 따라 예측 블록에 대한 값들을 보간할 수도 있다. 또한, 양방향 인터 예측에 대해, 모션 보상 유닛 (224) 은 각각의 모션 벡터들에 의해 식별된 2 개의 레퍼런스 블록들에 대한 데이터를 취출하고, 예를 들어 샘플 별 평균화 또는 가중된 평균화를 통해 취출된 데이터를 결합할 수도 있다.

[0188] 다른 예로서, 인트라 예측, 또는 인트라 예측 코딩에 대해, 인트라 예측 유닛 (226) 은 현재 블록에 이웃하는 샘플들로부터 예측 블록을 생성할 수도 있다. 예를 들어, 방향성 모드들에 대해, 인트라 예측 유닛 (226) 은 일반적으로 이웃하는 샘플들의 값들을 수학적으로 결합하고 현재 블록에 걸쳐 정의된 방향에서 이들 계산된 값들을 팝퓰레이트 (populate) 하여 예측 블록을 생성할 수도 있다. 다른 예로서, DC 모드에 대해, 인트라 예측 유닛 (226) 은 현재 블록에 대한 이웃하는 샘플들의 평균을 계산하고 예측 블록을 생성하여 예측 블록의 각각의 샘플에 대해 이러한 결과의 평균을 포함할 수도 있다.

[0189] 모드 선택 유닛 (202) 은 예측 블록을 잔차 생성 유닛 (204) 에 제공한다. 잔차 생성 유닛 (204) 은 비디오 데이터 메모리 (230) 로부터의 현재 블록의 원시의, 코딩되지 않은 버전 및 모드 선택 유닛 (202) 으로부터의 예측 블록을 수신한다. 잔차 생성 유닛 (204) 은 현재 블록과 예측 블록 사이의 샘플 별 차이들을 계산한다. 결과의 샘플 별 차이들은 현재 블록에 대한 잔차 블록을 정의한다. 일부 예들에서, 잔차 생성 유닛 (204) 은 또한 잔차 차분 펄스 코드 변조 (residual differential pulse code modulation; RDPCM) 를 사용하여 잔차 블록을 생성하기 위해 잔차 블록에서의 샘플 값들 사이의 차이들을 결정할 수도 있다. 일부 예들에서, 잔차 생성 유닛 (204) 은 바이너리 감산을 수행하는 하나 이상의 감산 회로들을 사용하여 형성될 수

도 있다.

[0190] 모드 선택 유닛 (202) 이 CU들을 PU들로 파티셔닝하는 예들에서, 각각의 PU 는 루마 예측 유닛 및 대응하는 크로마 예측 유닛들과 연관될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다양한 사이즈를 갖는 PU들을 지원할 수도 있다. 상기 나타낸 바와 같이, CU 의 크기는 CU 의 루마 코딩 블록의 크기를 나타낼 수도 있고 PU 의 크기는 PU 의 루마 예측 블록의 크기를 나타낼 수도 있다. 특정 CU 의 사이즈가 2Nx2N 이라고 가정하면, 비디오 인코더 (200) 는 인트라-예측을 위해 2Nx2N 또는 NxN 의 PU 사이즈들을 지원하고, 인터-예측을 위해 2Nx2N, 2NxN, Nx2N, NxN, 기타 등등의 대칭적인 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 인터-예측을 위해 2NxnU, 2NxnD, nLx2N, 및 nRx2N 의 PU 사이즈에 대한 비대칭적 파티셔닝을 지원할 수도 있다.

[0191] 모드 선택 유닛이 CU 를 PU들로 추가로 파티셔닝하지 않는 예들에서, 각각의 CU 는 루마 코딩 블록 및 대응하는 크로마 코딩 블록들과 연관될 수도 있다. 위에서와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 2Nx2N, 2NxN, 또는 Nx2N 의 CU 사이즈들을 지원할 수도 있다.

[0192] 인트라 블록 카피 모드 코딩, 아핀 모드 코딩 및 선형 모델 (LM) 모드 코딩과 같은 다른 비디오 코딩 기법들에 대해, 몇몇 예들에서와 같이, 모드 선택 유닛 (202) 은 코딩 기술과 연관된 개개의 유닛들을 통해, 인코딩될 현재 블록에 대한 예측 블록을 생성한다. 팔레트 모드 코딩과 같은 일부 예에서, 모드 선택 유닛 (202) 은 예측 블록을 생성하지 않을 수도 있고, 대신에 선택된 팔레트에 기초하여 블록을 재구성하는 방식을 표시하는 신택스 엘리먼트들을 생성할 수도 있다. 이러한 모드들에서, 모드 선택 유닛 (202) 은 이들 신택스 엘리먼트들을 인코딩되도록 엔트로피 인코딩 유닛 (220) 에 제공할 수도 있다.

[0193] 상술한 바와 같이, 잔차 생성 유닛 (204) 은 현재 블록 및 대응하는 예측 블록에 대해 비디오 데이터를 수신한다. 잔차 생성 유닛 (204) 은 그 후 현재 블록에 대한 잔차 블록을 생성한다. 잔차 블록을 생성하기 위해, 잔차 생성 유닛 (204) 은 현재 블록과 예측 블록 사이의 샘플 별 차이들을 계산한다.

[0194] 변환 프로세싱 유닛 (206) 은 잔차 블록에 하나 이상의 변환들을 적용하여 변환 계수들의 블록 (본 명세서에서는 "변환 계수 블록" 으로 지칭됨) 을 생성한다. 변환 프로세싱 유닛 (206) 은 다양한 변환들을 잔차 블록에 적용하여 변환 계수 블록을 형성할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (206) 은 이산 코사인 변환 (DCT), 방향성 변환, Karhunen-Loeve 변환 (KLT), 또는 개념적으로 유사한 변환을 잔차 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 대한 다중 변환들, 예를 들어 1 차 변환 및 2 차 변환, 이를 테면 회전 변환을 수행할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 변환들을 적용하지 않는다.

[0195] 양자화 유닛 (208) 은 양자화된 변환 계수 블록을 생성하기 위해 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 유닛 (208) 은 현재 블록과 연관된 양자화 파라미터 (QP) 값에 따라 변환 계수 블록의 변환 계수들을 양자화할 수도 있다. 비디오 인코더 (202) 는 (예를 들어, 모드 선택 유닛 (202) 을 통해) CU 와 연관된 QP 값을 조정함으로써 현재 블록과 연관된 계수 블록들에 적용된 양자화도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서 양자화된 변환 계수들은 변환 프로세싱 유닛 (206) 에 의해 생성된 원래의 변환 계수들보다 낮은 정밀도를 가질 수도 있다.

[0196] 역 양자화 유닛 (210) 및 역 변환 프로세싱 유닛 (212) 은 각각 양자화된 변환 계수 블록에 역 양자화 및 역 변환들을 적용하여, 변환 계수 블록으로부터 잔차 블록을 재구성할 수도 있다. 재구성 유닛 (214) 은 모드 선택 유닛 (202) 에 의해 생성된 예측 블록 및 재구성된 잔차 블록에 기초하여 (잠재적으로 어느 정도의 왜곡을 가짐에도 불구하고) 현재 블록에 대응하는 재구성된 블록을 생성할 수도 있다 . 예를 들어, 재구성 유닛 (214) 은 재구성된 잔차 블록의 샘플들을, 모드 선택 유닛 (202) 에 의해 생성된 예측 블록으로부터의 대응하는 샘플들에 가산하여 재구성된 블록을 생성할 수도 있다.

[0197] 필터 유닛 (216) 은 재구성된 블록에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (216) 은 CU들의 에지들을 따라 블록키니스 아티팩트들 (blockiness artifacts) 을 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (216) 의 동작들은 일부 예들에서 생략될 수도 있다.

[0198] 비디오 인코더 (200) 는 DPB (218) 에 재구성된 블록들을 저장한다. 예를 들어, 필터 유닛 (216) 의 동작들이 수행되지 않는 예들에서, 재구성 유닛 (214) 은 재구성된 블록들을 DPB (218) 에 저장할 수도 있다. 필터 유닛 (216) 의 동작들이 수행되는 예들에서, 필터 유닛 (216) 은 필터링된 재구성된 블록들을 DPB (218) 에

저장할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 재구성된 (및 잠재적으로 필터링된) 블록들로부터 형성된 DPB (218) 로부터 레퍼런스 픽처를 취출하여, 후속 인코딩된 픽처들의 블록들을 인터 예측할 수도 있다. 또한, 인트라 예측 유닛 (226) 은 현재 픽처에서의 다른 블록들을 인트라 예측하기 위해 현재 픽처의 DPB (218) 에서 재구성된 블록들을 사용할 수도 있다.

[0199] 일반적으로, 엔트로피 인코딩 유닛 (220) 은, 계수 코딩에 대해 상술된 신택스 엘리먼트들을 포함하는, 비디오 인코더 (200) 의 다른 기능 컴포넌트들로부터 수신된 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 양자화 유닛 (208) 으로부터 양자화된 변환 계수 블록들을 엔트로피 인코딩할 수도 있다. 다른 예로서, 엔트로피 인코딩 유닛 (220) 은 모드 선택 유닛 (202) 으로부터 예측 신택스 엘리먼트들 (예를 들어, 인터 예측에 대한 모션 정보 또는 인트라 예측에 대한 인트라 모드 정보) 을 엔트로피 인코딩할 수도 있다. 엔트로피 인코딩 유닛 (220) 은 엔트로피 인코딩된 데이터를 생성하기 위해, 비디오 데이터의 다른 예인, 신택스 엘리먼트들에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행할 수도 있다.

예를 들어, 엔트로피 인코딩 유닛 (220) 은 컨텍스트 적응적 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 신택스 기반 컨텍스트 적응 이진 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-골롬 인코딩 동작, 또는 다른 타입의 엔트로피 인코딩 동작을 데이터에 대해 수행할 수도 있다. 일부 예들에서, 엔트로피 인코딩 유닛 (220) 은 신택스 엘리먼트들이 엔트로피 인코딩되지 않는 바이패스 모드에서 동작할 수도 있다.

[0200] 비디오 인코더 (200) 는 픽처 또는 슬라이스의 블록들을 재구성하는데 필요한 엔트로피 인코딩된 신택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 특히, 엔트로피 인코딩 유닛 (220) 이 비트스트림을 출력할 수도 있다.

[0201] 상기 설명된 동작들은 블록에 대하여 설명된다. 이러한 설명은 루마 코딩 블록 및/또는 크로마 코딩 블록들에 대한 동작들인 것으로 이해되어야 한다. 상술한 바와 같이, 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 CU 의 루마 및 크로마 컴포넌트들이다. 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 PU 의 루마 및 크로마 컴포넌트들이다.

[0202] 일부 예들에서, 루마 코딩 블록에 대해 수행되는 동작들은 크로마 코딩 블록들에 대해 반복될 필요가 없다. 하나의 예로서, 크로마 블록들에 대한 모션 벡터 (MV) 및 레퍼런스 픽처를 식별하기 위해 루마 코딩 블록에 대한 MV 및 레퍼런스 픽처를 식별하는 동작들이 반복될 필요는 없다. 오히려, 루마 코딩 블록에 대한 MV 는 크로마 블록들에 대한 MV 를 결정하도록 스케일링될 수도 있고, 레퍼런스 픽처는 동일할 수도 있다. 다른 예로서, 인트라 예측 프로세스는 루마 코딩 블록들 및 크로마 코딩 블록들에 대해 동일할 수도 있다.

[0203] 비디오 인코더 (200) 는 비디오 데이터를 저장하도록 구성된 메모리, 및 고정 기능 및 / 또는 프로그램 가능 회로에서 구현되고 본 개시에 설명 된 예시적인 기법들을 수행하도록 구성된 하나 이상의 프로세싱 유닛을 포함하는 비디오 데이터를 인코딩하도록 구성된 디바이스의 예를 나타낸다.

[0204] 도 10 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더 (300) 를 나타내는 블록도이다. 도 10 은 설명의 목적들을 위해 제공되며, 본 개시에서 넓게 예시화되고 설명된 바와 같은 기법들에 대해 한정하는 것은 아니다. 설명의 목적으로, 본 개시는 비디오 디코더 (30) 가 JEM 및 HEVC 의 기법들에 따라 기술되는 것을 설명한다. 그러나, 본 개시의 기법들은 다른 비디오 코딩 표준들로 구성되는 비디오 코딩 디바이스들에 의해 수행될 수도 있다.

[0205] 도 10 의 예에 있어서, 비디오 디코더 (300) 는 코딩된 픽처 버퍼 (CPB)(320), 엔트로피 디코딩 유닛 (302), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (310), 재구성 유닛 (310), 필터 유닛 (312), 및 디코딩된 픽처 버퍼 (DPB) (314) 를 포함한다. 예측 프로세싱 유닛 (304) 은 모션 보상 유닛 (316) 및 인트라 예측 유닛 (318) 을 포함한다. 예측 프로세싱 유닛 (304) 은 다른 예측 모드들에 따라 예측을 수행하기 위해 부가적인 유닛들을 포함할 수도 있다. 예를 들어, 예측 프로세싱 유닛 (304) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 보상 유닛 (316) 의 일부를 형성할 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다. 다른 예들에서, 비디오 디코더 (300) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0206] CPB 메모리 (320) 는, 비디오 디코더 (300) 의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. CPB 메모리 (320) 에 저장된 비디오 데이터는, 예를 들어 컴퓨터 판독가능 매체 (110) (도 1) 로부터 획득될 수도 있다. CPB 메모리 (320) 는 인코딩된 비디오 비트스트림으로부터

인코딩된 비디오 데이터 (예를 들어, 신택스 엘리먼트들) 를 저장하는 CPB 를 포함할 수도 있다. 또한, CPB 메모리 (320) 는 비디오 디코더 (300) 의 다양한 유닛들로부터의 출력들을 나타내는 일시적 데이터와 같은, 코딩된 픽처의 신택스 엘리먼트들 이외의 비디오 데이터를 저장할 수도 있다. DPB (314) 는 일반적으로, 인코딩된 비디오 비트스트림의 후속 데이터 또는 픽처들을 디코딩할 때, 레퍼런스 비디오 데이터로서 비디오 디코더 (300) 가 출력 및/또는 사용할 수도 있는 디코딩된 픽처들을 저장한다. CPB 메모리 (320) 및 DPB (314) 는 다양한 메모리 디바이스들, 예컨대 동기식 동적 랜덤 액세스 메모리 (SDRAM) 를 포함한 DRAM, 자기저항성 RAM (MRAM), 저항성 RAM (RRAM) 과 같은 다양한 메모리 디바이스들, 또는 다른 타입들의 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. CPB 메모리 (320) 및 DPB (314) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, CPB 메모리 (320) 는 비디오 디코더 (300) 의 다른 컴포넌트들과 온-칩이거나, 또는 그 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0207] 추가적으로 또는 대안적으로, 일부 예들에서, 비디오 디코더 (300) 는 메모리 (120) (도 1) 로부터 코딩된 비디오 데이터를 취출 (retrieve) 할 수도 있다. 즉, 메모리 (120) 는 CPB 메모리 (320) 로 상기 논의된 바와 같이 데이터를 저장할 수도 있다. 마찬가지로, 메모리 (120) 는 비디오 디코더 (300) 의 기능성의 일부 또는 전부가 비디오 디코더 (300) 의 프로세싱 회로부에 의해 실행되는 소프트웨어에서 구현될 때, 비디오 디코더 (300) 에 의해 실행될 명령들을 저장할 수도 있다.

[0208] 도 10 에 나타낸 다양한 유닛들은 비디오 인코더 (300) 에 의해 수행되는 동작들의 이해를 돕기 위해 도시된다. 이 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 도 9 와 유사하게, 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하며, 수행될 수 있는 동작들에 대해 미리설정된다. 프로그래밍가능 회로들은 다양한 태스크들을 수행하도록 프로그래밍될 수 있는 회로들을 지칭하고, 수행될 수 있는 동작들에서 유연한 기능성을 제공한다. 예를 들어, 프로그래밍가능 회로들은, 프로그래밍가능 회로들이 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 (예를 들어, 파라미터들을 수신하거나 또는 파라미터들을 출력하기 위해) 소프트웨어 명령들을 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작들의 타입들은 일반적으로 불변이다. 일부 예들에서, 유닛들 중 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그래밍가능) 일 수도 있고, 일부 예들에서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.

[0209] 비디오 디코더 (300) 는 프로그래밍가능 회로들로부터 형성된, ALU들, EFU들, 디지털 회로들, 아날로그 회로들, 및/또는 프로그래밍가능 코어들을 포함할 수도 있다. 비디오 디코더 (300) 의 동작들이 프로그래밍가능 회로들 상에서 실행하는 소프트웨어에 의해 수행되는 예들에서, 온-칩 또는 오프-칩 메모리는 비디오 디코더 (300) 가 수신하고 실행하는 소프트웨어의 명령들 (예를 들어, 오브젝트 코드) 을 저장할 수도 있다.

[0210] 엔트로피 디코딩 유닛 (302) 은 CPB 로부터 인코딩된 비디오 데이터를 수신하고, 계수 코딩에 대해 상술된 신택스 엘리먼트들을 포함하는 신택스 엘리먼트들을 재생하기 위해 비디오 데이터를 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 재구성 유닛 (310), 및 필터 유닛 (312) 은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0211] 일반적으로, 비디오 디코더 (300) 는 블록 별로 픽처를 재구성한다. 비디오 디코더 (300) 는 개별적으로 각각의 블록에 대해 재구성 동작을 수행할 수도 있다 (여기서 현재 재구성되는, 즉 디코딩되는 블록은 "현재 블록" 으로 지칭될 수도 있음).

[0212] 엔트로피 디코딩 유닛 (302) 은 양자화 파라미터 (QP) 및/또는 변환 모드 표시(들)와 같은 변환 정보 뿐만 아니라, 양자화된 변환 계수 블록의 양자화된 변환 계수들을 정의하는 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 역 양자화 유닛 (306) 은 양자화된 변환 계수 블록과 연관된 QP 를 사용하여, 양자화도 및 유사하게, 적용할 역 양자화 유닛 (306) 에 대한 역 양자화도를 결정할 수도 있다. 역 양자화 유닛 (306) 은 예를 들어, 양자화된 변환 계수들을 역 양자화하기 위해 비트단위 좌측-시프트 동작을 수행할 수도 있다. 따라서, 역 양자화 유닛 (306) 은 변환 계수들을 포함하는 변환 계수 블록을 형성할 수도 있다.

[0213] 역 양자화 유닛 (306) 이 변환 계수 블록을 형성한 후, 역변환 프로세싱 유닛 (308) 은 현재 블록과 연관된 잔차 블록을 생성하기 위해 변환 계수 블록에 하나 이상의 역 변환들을 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (308) 은 역 DCT, 역 정수 변환, 역 Karhunen-Loeve 변환 (KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 계수 블록에 적용할 수도 있다.

[0214] 또한, 예측 프로세싱 유닛 (304) 은 엔트로피 디코딩 유닛 (302) 에 의해 엔트로피 디코딩된 예측 정보 신택스 엘리먼트들에 따라 예측 블록을 생성한다. 예를 들어, 예측 정보 신택스 엘리먼트들이 현재 블록이 인터 예측됨을 표시하면, 모션 보상 유닛 (316) 은 예측 블록을 생성할 수도 있다. 이 경우, 예측 정보 신택스 엘리먼트들은 레퍼런스 블록을 취출할 DPB (314) 에서의 레퍼런스 픽처뿐만 아니라 현재 픽처에서의 현재 블록의 위치에 대한 레퍼런스 픽처에서의 레퍼런스 블록의 위치를 식별하는 모션 벡터를 표시할 수도 있다. 모션 보상 유닛 (316) 은 일반적으로 모션 보상 유닛 (224) (도 9) 과 관련하여 설명된 것과 실질적으로 유사한 방식으로 인터 예측 프로세스를 수행할 수도 있다.

[0215] 다른 예로서, 예측 정보 신택스 엘리먼트가 현재 블록이 인트라 예측되는 것을 표시하면, 인트라 예측 유닛 (318) 은 예측 정보 신택스 엘리먼트들에 의해 표시된 인트라 예측 모드에 따라 예측 블록을 생성할 수도 있다. 다시, 인트라 예측 유닛 (318) 은 일반적으로 인트라 예측 유닛 (226)(도 9) 과 관련하여 설명된 것과 실질적으로 유사한 방식으로 인트라 예측 프로세스를 수행할 수도 있다. 인트라 예측 유닛 (318) 은 DPB (314) 로부터 현재 블록에 이웃하는 샘플들의 데이터를 취출할 수도 있다.

[0216] 재구성 유닛 (310) 은 예측 블록 및 잔차 블록을 사용하여 현재 블록을 재구성한다. 예를 들어, 재구성 유닛 (310) 은 잔차 블록의 샘플들을 예측 블록의 대응하는 샘플들에 가산하여 현재 블록을 재구성할 수도 있다.

[0217] 필터 유닛 (312) 은 재구성된 블록들에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (312) 은 재구성된 블록들의 에지들을 따라 블록키니스 아티팩트를 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (312) 의 동작들이 모든 예들에서 반드시 수행되는 것은 아니다.

[0218] 비디오 디코더 (300) 는 DPB (314) 에 재구성된 블록들을 저장할 수도 있다. 상기 논의된 바와 같이, DPB (314) 는 예측 프로세싱 유닛 (304) 에, 인트라 예측을 위한 현재 픽처의 샘플들 및 후속 모션 보상을 위한 이전에 디코딩된 픽처들과 같은 레퍼런스 정보를 제공할 수도 있다. 또한, 비디오 디코더 (300) 는 도 1 의 디스플레이 디바이스 (118) 와 같은 디스플레이 디바이스 상으로의 후속 프리젠테이션을 위해 DPB 로부터 디코딩된 픽처들을 출력할 수도 있다.

[0219] 이러한 방식으로, 비디오디코더 (300) 는, 비디오 디코딩 디바이스의 예를 나타내고, 그 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리, 및 회로에서 구현되고 본 개시에서 설명된 바와 같이 계수들을 디코딩하도록 구성된 하나 이상의 프로세싱 유닛들을 포함한다.

[0220] 도 11a 및 도 11b 는 빈 n 에서의 CABAC 프로세스의 예들을 도시한다. 도 11a 의 예 (400) 에서, 빈 n 에서, 빈 2 에서의 레인지는 특정 컨텍스트 상태 (σ) 가 주어지는 최소 가능 심볼 (LPS) 의 확률 (σ) 에 의해 주어지는 RangeMPS 및 RangeLPS 를 포함한다. 예 (400) 는 빈 n 의 값이 최대 가능 심볼 (MPS)과 동일할 때 빈 n+1 에서의 레인지의 업데이트를 나타낸다. 이 예에서, 로우 (low) 는 동일하게 유지되지만, 빈 n+1 에서의 레인지의 값은 빈 n 에서의 RangeMPS 의 값으로 감소된다. 도 11b 의 예 (402) 는 빈 n 의 값이 MPS 와 동일하지 않을 때 (즉, LPS 와 동일할 때) 빈 n+1 에서의 레인지의 업데이트를 나타낸다. 이 예에서, 로우는 빈 n 에서 RangeLPS 의 더 낮은 레인지 값으로 이동된다. 또한, 빈 n+1 에서의 레인지의 값은 빈 n 에서의 RangeLPS 의 값으로 감소된다.

[0221] HEVC 비디오 코딩 프로세스의 일 예에서, 레인지는 9 비트들로 표현되고 10 비트들로 로우로 표현된다. 레인지 및 로우 값을 충분한 정밀도로 유지하기 위한 재정규화 (renormalization) 프로세스가 존재한다. 레인지가 256 보다 적을 때마다 재정규화가 발생한다. 따라서 레인지는 재정규화 후 항상 256 보다 크거나 같다. 레인지 및 로우의 값들에 따라, 이진 산술 코더 (BAC) 는 '0' 또는 '1'인 비트스트림으로 출력하거나, 또는 미래의 출력을 유지하기 위해 (BO: bits-outstanding 라고 불리는) 내부 변수를 업데이트한다. 도 12 는 레인지에 의존하는 BAC 출력의 예들을 나타낸다. 예를 들어, 레인지 및 로우가 특정 임계치 (예를 들어, 512) 초과일 때 비트스트림에 '1' 이 출력된다. 레인지 및 로우가 특정 임계치 (예를 들어, 512) 미만일 때, '0' 이 비트스트림으로 출력된다. 레인지와 하한이 특정 임계치들 사이에 있을 때 비트스트림에 아무것도 출력되지 않는다. 대신에, BO 값이 증분될 수도 있고, 다음 빈이 인코딩된다.

[0222] H.264/AVC 의 CABAC 컨텍스트 모델 및 HEVC의 일부 예들에서, 128 개의 상태들이 존재한다. 0 내지 63 일 수 있는 64 개의 가능한 LPS 확률들 (*상태* σ 에 의해 표시됨) 이 존재한다. 각 MPS는 0 또는 1 일 수 있다. 이와 같이, 128 개의 상태들은 MPS 에 대한 2 개의 가능한 값들 (0 또는 1) 에 대한 64 개의 상태 확률들이다. 따라서, 상태는 7 비트들로 인덱싱될 수 있다.

[0223] LPS 레인지들 ($rangeLPS_{\sigma}$) 을 도출하는 계산을 감소시키기 위해, 모든 경우들에 대한 결과들은 미리 계산되고

룩-업 테이블에 근사치들로서 저장될 수도 있다. 따라서 간단한 테이블 룩업을 이용함으로써 곱셈 없이 LPS 레인지가 획득될 수 있다. 이러한 동작은 많은 하드웨어 아키텍처들에서 상당한 레이턴시를 야기할 수도 있기 때문에, 곱셈을 피하는 것은 일부 디바이스들 또는 애플리케이션들에 대해 중요할 수 있다.

[0224] 곱셈 대신 4-열 미리 계산된 LPS 레인지 테이블이 사용될 수도 있다. 레인지는 네 개의 세그먼트로 나누어진다. 세그먼트 인덱스는 질문 (range>>6)&3 에 의해 도출될 수 있다. 사실상, 세그먼트 인덱스는 실제 레인지로부터 비트들을 시프트 및 드롭함으로써 도출된다. 하기 표 1 은 가능한 레인지들 및 그것들의 대응하는 인덱스들을 나타낸다.

표 1

[0225] **테이블 1 - 레인지 인덱스**

레인지	256-319	320-383	384-447	448-511
(range>>6) & 3	0	1	2	3

[0226] 그 후, LPS 레인지 테이블은 64 개의 엔트리들 (각각의 확률 상태에 대해 하나) 곱하기 4 (각각의 레인지 인덱스에 대해 하나) 를 갖는다. 각 엔트리는 Range LPS, 즉, 레인지에 LPS 확률을 곱한 값이다. 이 테이블의 부분의 일 예가 하기 테이블 2 에서 나타내어진다. 테이블 2 는 확률 상태들 9-12 를 나타낸다. HEVC에 대한 하나의 제안에서, 확률 상태들은 0 내지 63 의 범위일 수도 있다.

표 2

[0227] **테이블 2 - RangeLPS**

프로브 상태(σ)	RangeLPS 인덱스 0	인덱스 1	인덱스 2	인덱스 3
…	…	…	…	…
9	90	110	130	150
10	85	104	123	142
11	81	99	117	135
12	77	94	111	128
…	…	…	…	…

[0228] 각각의 세그먼트 (즉, 레인지 값) 에서, 각각의 확률 상태$_{\sigma}$ 의 LPS 레인지가 미리 정의된다. 다시 말해, 확률 상태$_{\sigma}$ 의 LPS 레인지는 4개의 값들 (즉, 각 레인지 인덱스에 대해 하나의 값) 로 양자화된다. 주어진 포인트에서 사용되는 특정 LPS 레인지는 레인지는 어느 세그먼트에 속하는지에 의존한다. 테이블에서 사용된 가능한 LPS 레인지들의 수는 표 열들의 수 (즉, 가능한 LPS 레인지 값들의 수) 와 LPS 레인지 정밀도 사이의 트레이드-오프 (trade-off) 이다. 일반적으로 말하면, 더 많은 열들은 LPS 레인지 값들의 더 작은 양자화 에러들을 초래하지만, 또한 테이블을 저장하기 위해 더 많은 메모리에 대한 필요성을 증가시킨다. 열의 수가 적을수록 양자화 오류들이 증가하지만 테이블을 저장하는 데 필요한 메모리를 감소시킨다.

[0229] 앞서 설명된 바와 같이, 각각의 LPS 확률 상태는 대응하는 확률을 갖는다. 각 상태에 대한 확률 p는 다음과 같이 도출된다:

[0230] $p_{\sigma} = \alpha\, p_{\sigma-1}$

[0231] 여기서, 상태 σ 는 0 내지 63 이다. 상수 α 는 각 컨텍스트 상태 간의 확률 변화량을 나타낸다. 일 예에서, α=0.9493, 또는 더 정확하게는, α=(0.01875/0.5) 이다. 상태 σ=0 에서의 확률은 0.5 (즉, p_0 = 1/2) 와 동일하다. 즉, 컨텍스트 상태 0 에서, LPS 및 MPS 는 동등하게 가능하다. 각각의 연속적인 상태에서의 확률은 이전 상태에 α 를 곱함으로써 도출된다. 이와 같이, 컨텍스트 상태 α = 1 에서 LPS 가 발생할 확률은 p_0 * 0.9493 (0.5 * 0.9493 = 0.47465) 이다. 이와 같이, 상태 α 의 인덱스가 증가함에 따라 LPS 가 발생할 확률은 낮아진다.

[0232] CABAC 는 신호 통계 (즉, 이전에 코딩된 빈들의 값들) 를 따르기 위해 확률 상태들이 업데이트되기 때문에 적응

적이다. 그 업데이트 프로세스는 다음과 같다. 주어진 확률 상태에 대해, 업데이트는 상태 인덱스 및 LPS 또는 MPS 중 어느 일방으로서 식별된 인코딩된 심볼의 값에 의존한다. 업데이트 프로세스의 결과로서, 잠재적으로 수정된 LPS 확률 추정치 및, 필요한 경우, 수정된 MPS 값으로 이루어진 새로운 확률 상태가 도출된다.

[0233] MPS 와 동일한 빈 값의 경우, 주어진 상태 인덱스는 1만큼 증분될 수도 있다. 이것은 MPS 가 상태 인덱스 (62) 에서 발생할 때를 제외한 모든 상태들에 대한 것이며, 여기서, LPS 확률은 이미 그것의 최소에 있다 (또는 동등하게는, 최대 MPS 확률에 도달된다). 이 경우에, 상태 인덱스 (62) 는 LPS 가 보여질 때까지, 또는 마지막 빈 값이 인코딩될 때까지 고정된 채로 유지된다 (상태 (63) 는 마지막 빈 값의 특별한 경우에 대해 사용된다). LPS 가 발생할 때, 아래의 식에서 나타낸 바와 같이 상태 인덱스를 소정 량 만큼 감소시킴으로써 상태 인덱스가 변경된다. 이 규칙은 일반적으로 다음 예외와 함께 LPS 의 각 발생에 적용된다. LPS가 인덱스 σ=0 를 갖는 상태에서 인코딩되었다고 가정하면, 이는 등확률적 경우에 대응하며, 상태 인덱스는 고정된 채로 유지되지만, MPS 값은 LPS 와 MPS 의 값이 상호교환되도록 토글될 것이다. 다른 모든 경우들에서, 어느 심볼이 인코딩되었든, MPS 값은 변경되지 않을 것이다. LPS 확률에 대한 전이 규칙들의 도출은 주어진 LPS 확률 p_{old} 와 그것의 업데이트된 대응물 p_{new} 사이의 다음 관계에 기초한다:

[0234] $p_{new} = \max(\alpha\ p_{old} , p_{62})$ MPS 가 발생하는 경우

[0235] $p_{new} = (1- \alpha) + \alpha\ p_{old}$ LPS 가 발생하는 경우

[0236] CABAC 에서의 확률 추정 프로세스의 실제 구현과 관련하여, 모든 전이 규칙들은 6-비트 무부호 정수 값들의 63 개의 엔트리들을 각각 갖는 최대 2 개의 테이블들에 의해 실현될 수도 있다는 것에 유의하는 것이 중요하다. 일부 예들에서, 상태 전이들은, 주어진 상태 인덱스 σ 에 대해, LPS 가 관찰된 경우에 새로운 업데이트된 상태 인덱스 TransIdxLPS [σ] 를 결정하는 단일 테이블 TransIdxLPS 로 결정될 수도 있다. MPS-구동 전이들은 1 의 고정된 값에 의한 상태 인덱스의 단순한 (포화된) 증분에 의해 획득될 수 있고, 그 결과 업데이트된 상태 인덱스 min(σ+1, 62) 가 된다. 아래의 테이블 3 은 부분 TransIdxLPS 테이블의 일 예이다.

표 3

[0237] **테이블 3 - TransIdxLPS**

프로브 상태 (σ)	새로운 상태 TransIdxLPS [σ]
…	…
9	6
10	8
11	8
12	8
…	…

[0238] 도 11a, 도 11b, 및 도 12 와 관련하여 상술된 기술들은 단지 CABAC 의 하나의 예시적인 구현을 나타낸다. 본 개시의 기술들은 CABAC 의 이 설명된 구현에만 한정되지 아니함을 이해하여야 한다. 예를 들어, 더 오래된 BAC 접근법들 (예를 들어, H.264/AVC 에서 사용된 BAC 접근법) 에서, 테이블들 RangeLPS 및 TransIdxLPS 는 저 해상도 비디오들 (즉, 공통 중간 포맷 (CIF) 및 쿼터-CIF (QCIF) 비디오들) 에 대해 튜닝되었다. HEVC 및 VVC 와 같은 미래의 코덱들로, 많은 양의 비디오 콘텐츠는 HD (high definition) 이고, 일부 경우들에서, HD 보다 크다. HD 또는 HD 해상도 이상의 비디오 콘텐츠는 H.264/AVC 를 개발하기 위해 사용된 10년 된 QCIF 시퀀스들과는 다른 통계를 갖는 경향이 있다. 이와 같이, H.264/AVC 로부터의 테이블들 RangeLPS 및 TransIdxLPS 는 너무 빠른 방식으로 상태들 간의 적응 (adaptation) 을 야기할 수도 있다. 즉, 특히 LPS 가 발생할 때, 확률 상태들 사이의 전이들은 HD 비디오의 더 부드럽고 더 높은 해상도 콘텐츠에 대해 너무 클 수 있다. 따라서, 종래의 기법들에 따라 사용되는 확률 모델들은 HD 및 엑스트라-HD 콘텐츠에 대해 정확하지 않을 수도 있다. 또한, HD 비디오 콘텐츠가 더 큰 레인지의 픽셀 값들을 포함함에 따라, H.264/AVC 테이블들은 HD 콘텐츠에서 존재할 수도 있는 보다 극단적인 값들을 고려하기에 충분한 엔트리들을 포함하지 않는다.

[0239] 이와 같이, HEVC 에 대해 그리고 VVC 와 같은 미래의 코딩 표준들에 대해, RangeLPS 및 TransIdxLPS 테이블들은 이러한 새로운 콘텐츠의 특성들을 고려하도록 수정될 수도 있다. 특히, HEVC 및 미래의 코딩 표준들에 대한 BAC 프로세스들은 더 느린 적응 프로세스를 허용하는 테이블들을 사용할 수도 있고, 더 극단적인 경우들 (즉,

왜곡된 확률들) 을 설명할 수도 있다. 따라서, 일 예로서, RangeLPS 및 TransIdxLPS 테이블들은 H.264/AVC 또는 HEVC 에 있어서 BAC 에서 사용되는 것보다 더 많은 확률 상태들 및 레인지들을 포함함으로써 이들 목적들을 달성하기 위해 수정될 수도 있다.

[0240] 도 13 은 본 개시의 기법들에 따라 CABAC 를 수행하도록 구성될 수도 있는 예시적인 엔트로피 인코딩 유닛 (220) 의 블록도이다. 신택스 엘리먼트 (418) 는 엔트로피 인코딩 유닛 (220) 내로 입력된다. 신택스 엘리먼트가 이미 바이너리 값 신택스 엘리먼트 (즉, 0 및 1 의 값만을 갖는 신택스 엘리먼트) 인 경우, 이진화 단계는 생략될 수도 있다. 신택스 엘리먼트가 비-바이너리 값의 신택스 엘리먼트 (예를 들어, 변환 계수 레벨들과 같은 다수의 비트들에 의해 표현된 신택스 엘리먼트) 인 경우, 비-바이너리 값의 신택스 엘리먼트는 이진화기 (420) 에 의해 이진화된다. 이진화기 (420) 는 바이너리 결정들 (binary decisions) 의 시퀀스로의 비-바이너리 값의 맵핑을 수행한다. 이들 바이너리 결정들은 종종 "빈들 (bins)"이라고 불립니다. 예를 들어, 변환 계수 레벨들에 대해, 레벨의 값은 연속적인 빈들로 분해될 수도 있으며, 각각의 빈은 계수 레벨의 절대 값이 어떤 값보다 큰지 여부를 나타낸다. 예를 들어, 빈 0 (때때로 유의성 플래그로 지칭됨) 은 변환 계수 레벨의 절대 값이 0 보다 큰지 아닌지 여부를 나타낸다. 빈 1 은 변환 계수 레벨의 절대 값이 1 보다 큰지 아닌지 등을 나타낸다. 각각의 비-바이너리 값의 신택스 엘리먼트에 대해 고유한 맵핑이 개발될 수도 있다.

[0241] 이진화기 (420) 에 의해 생성된 각각의 빈은 엔트로피 인코딩 유닛 (220) 의 이진 산술 코딩 측에 공급된다. 즉, 비-바이너리 값의 신택스 엘리먼트들의 미리 결정된 세트에 대해, 각각의 빈 유형 (예를 들어, 빈 0) 은 다음 빈 유형 (예를 들어, 빈 1) 전에 코딩된다. 코딩은 정규 모드 (regular mode) 또는 바이패스 모드 (bypass mode) 중 어느 일방에서 수행될 수도 있다. 바이패스 모드에서, 바이패스 코딩 엔진 (426) 은 고정 확률 모델을 사용하여, 예를 들어, 골롬-라이스 또는 지수 골롬 코딩을 사용하여, 산술 코딩을 수행한다. 바이패스 모드는 일반적으로 보다 예측가능한 신택스 엘리먼트들에 사용된다.

[0242] 정규 모드에서의 코딩은 CABAC 를 수행하는 것을 수반한다. 정규 모드 CABAC 는, 이전에 코딩된 빈들의 값들이 주어지면 빈 값의 확률이 예측 가능한 빈 값들을 코딩하기 위해 수행된다. 빈이 LPS 일 확률은 컨텍스트 모델러 (422) 에 의해 결정된다. 컨텍스트 모델러 (422) 는 빈 값 및 컨텍스트 모델 (예를 들어, 확률 상태 σ) 을 출력한다. 컨텍스트 모델은 일련의 빈들에 대한 초기 컨텍스트 모델일 수도 있거나, 또는 이전에 인코딩된 빈들의 코딩된 값들에 기초하여 결정될 수도 있다. 전술된 바와 같이, 컨텍스트 모델러는 이전에 코딩된 빈이 MPS 또는 LPS 였는지 여부에 기초하여 상태를 업데이트할 수도 있다.

[0243] 컨텍스트 모델 및 확률 상태 σ 가 컨텍스트 모델러 (422) 에 의해 결정된 후, 정규 코딩 엔진 (424) 은 빈 값에 대해 BAC 를 수행한다. 본 개시의 기법들에 따르면, 정규 코딩 엔진 (424) 은 64 개 초과의 확률 상태들 σ 을 포함하는 TransIdxLPS 테이블 (430) 을 이용하여 BAC 를 수행한다. 하나의 예에서, 확률 상태들의 수는 128 이다. TransIdxLPS 는 이전 빈 (빈 n) 이 LPS일 때 다음 빈 (빈 n+1) 에 대해 어느 확률 상태가 사용되는지를 결정하는데 사용된다. 정규 코딩 엔진 (424) 은 또한 특정 확률 상태 σ 가 주어지면 LPS 에 대한 레인지 값을 결정하기 위해 RangeLPS 테이블 (128) 을 사용할 수도 있다. 그러나, 본 개시의 기법들에 따르면, TransIdxLPS 테이블 (430) 의 모든 가능한 확률 상태들 σ 을 이용하기보다는, 확률 상태 인덱스들 σ 은 RangeLPS 테이블에서 이용하기 위해 그룹화된 인덱스들에 맵핑된다. 즉, RangeLPS 테이블 (428) 내로의 각각의 인덱스는 확률 상태들의 총 수 중 2 개 이상을 나타낼 수도 있다. 그룹화된 인덱스들에 대한 확률 상태 인덱스 σ 의 맵핑은 (예를 들어, 2 로 나눔으로써) 선형일 수도 있거나, 비선형 (예를 들어, 로그 함수 또는 맵핑 테이블) 일 수도 있다.

[0244] 본 개시의 다른 예들에서, 파라미터 α 를 0.9493 보다 크도록 설정함으로써 연속적인 확률 상태들 사이의 차이가 더 작게 될 수도 있다. 하나의 예에서 $\alpha = 0.9689$ 이다. 본 개시의 다른 예에서, LPS 가 발생할 최고 확률 (p_0) 은 0.5 보다 낮게 설정될 수도 있다. 하나의 예에서, p_0 는 0.493 과 동일할 수도 있다.

[0245] 본 개시의 하나 이상의 기법들에 따르면, 이진 산술 코딩 프로세스에서 확률 상태를 업데이트하는데 사용되는 변수의 동일한 값 (예를 들어, 윈도우 사이즈, 스케일링 팩터 (α), 및 확률 업데이트 속도 중 하나 이상) 을 사용하는 것과 대조적으로, 엔트로피 인코딩 유닛 (220) 은 상이한 컨텍스트 모델들 및/또는 상이한 신택스 엘리먼트들에 대해 변수의 상이한 값들을 사용할 수도 있다. 실례로, 엔트로피 인코딩 유닛 (220) 은 복수의 컨텍스트 모델들 중 하나의 컨텍스트 모델에 대해, 이진 산술 코딩 프로세스에서 확률 상태를 업데이트하기 위해 사용되는 변수의 값을 결정하고, 그 결정된 값에 기초하여 확률 상태를 업데이트할 수도 있다.

[0246] 도 14 는 본 개시의 기법들에 따라 CABAC 를 수행하도록 구성될 수도 있는 예시적인 엔트로피 인코딩 유닛

(302) 의 블록도이다. 도 14 의 엔트로피 디코딩 유닛 (302) 은 도 13 에서 설명된 엔트로피 인코딩 유닛 (220) 과 역의 방식으로 CABAC 를 수행한다. 비트스트림 (448) 으로부터의 코딩된 비트들은 엔트로피 디코딩 유닛 (302) 으로 입력된다. 코딩된 비트들이 바이패스 모드 또는 정규 모드를 사용하여 엔트로피 코딩되었는지 여부에 기초하여, 컨텍스트 모델러 (450) 또는 바이패스 디코딩 엔진 (452) 중 어느 하나에 코딩된 비트들이 공급된다. 코딩된 비트들이 바이패스 모드로 코딩된 경우에, 바이패스 디코딩 엔진 (452) 은, 예를 들어, 비-바이너리 신택스 엘리먼트들의 빈들 또는 바이너리 값의 신택스 엘리먼트들을 취출하기 위해 골롬-라이스 또는 지수 골롬 디코딩을 사용할 수도 있다.

[0247] 코딩된 비트들이 정규 모드에서 코딩된 경우에, 컨텍스트 모델러 (450) 는 코딩된 비트들에 대한 확률 모델을 결정할 수도 있고 정규 디코딩 엔진 (454) 은 코딩된 비트들을 디코딩하여 비-바이너리 값의 신택스 엘리먼트들의 빈들 (또는 바이너리 값인 경우 신택스 엘리먼트들 자체) 를 생성할 수도 있다. 컨텍스트 모델 및 확률 상태 σ 가 컨텍스트 모델러 (450) 에 의해 결정된 후, 정규 디코딩 엔진 (454) 은 빈 값에 대해 BAC 를 수행한다. 본 개시의 기법들에 따르면, 정규 디코딩 엔진 (454) 은 64 개 초과의 확률 상태들 σ 을 포함하는 TransIdxLPS 테이블 (458) 을 이용하여 BAC 를 수행한다. 하나의 예에서, 확률 상태들의 수는 128 이지만, 본 개시의 기법들과 일치하는, 다른 수들의 확률 상태들이 정의될 수 있을 것이다. TransIdxLPS 테이블 (458) 은 이전 빈 (빈 n) 이 LPS일 때 다음 빈 (빈 n+1) 에 대해 어느 확률 상태가 사용되는지를 결정하는데 사용된다. 정규 디코딩 엔진 (454) 은 또한 특정 확률 상태 σ 가 주어지면 LPS 에 대한 레인지 값을 결정하기 위해 RangeLPS 테이블 (456) 을 사용할 수도 있다. 그러나, 본 개시의 기법들에 따르면, TransIdxLPS 테이블 (458) 의 모든 가능한 확률 상태들 σ 을 이용하기보다는, 확률 상태 인덱스들 σ 은 RangeLPS 테이블 (456) 에서 이용하기 위해 그룹화된 인덱스들에 맵핑된다. 즉, RangeLPS 테이블 (456) 내로의 각각의 인덱스는 확률 상태들의 총 수 중 2 개 이상을 나타낼 수도 있다. 그룹화된 인덱스들에 대한 확률 상태 인덱스 σ 의 맵핑은 (예를 들어, 2 로 나눔으로써) 선형일 수도 있거나, 비선형 (예를 들어, 로그 함수 또는 맵핑 테이블) 일 수도 있다.

[0248] 본 개시의 다른 예들에서, 파라미터 α 를 0.9493 보다 크도록 설정함으로써 연속적인 확률 상태들 사이의 차이가 더 작게 될 수도 있다. 하나의 예에서 α = 0.9689 이다. 본 개시의 다른 예에서, LPS 가 발생할 최고 확률 (p_0) 은 0.5 보다 낮게 설정될 수도 있다. 하나의 예에서, p_0 는 0.493 과 동일할 수도 있다.

[0249] 정규 디코딩 엔진 (224) 에 의해 빈들이 디코딩된 후에, 역 이진화기 (460) 는 빈들을 다시 비-바이너리 값의 신택스 엘리먼트들의 값들로 변환하기 위해 역 맵핑을 수행할 수도 있다.

[0250] 도 15 는 비디오 데이터의 현재 블록을 인코딩하기 위한 비디오 인코더의 예시적인 동작을 나타내는 플로우차트이다. 현재 블록은 현재 CU 를 포함할 수도 있다. 비디오 인코더 (200) (도 1 및 도 9) 와 관련하여 설명되었지만, 다른 디바이스들이 도 15 의 동작과 유사한 동작을 수행하도록 구성될 수도 있음을 이해하여야 한다.

[0251] 이 예에서, 비디오 인코더 (200) 는 초기에 현재 블록을 예측한다 (550). 예를 들어, 비디오 인코더 (200) 는 현재 블록에 대한 예측 블록을 형성할 수도 있다. 그 후, 비디오 인코더 (200) 는 현재 블록에 대한 잔여 블록을 계산할 수도 있다 (552). 잔차 블록을 계산하기 위해, 비디오 인코더 (200) 는 원래의 코딩되지 않은 블록과 현재 블록에 대한 예측 블록 간의 차이를 계산할 수도 있다. 그 후에, 비디오 인코더 (200) 는 잔차 블록의 계수들을 변환하고 양자화할 수도 있다 (554). 다음으로, 비디오 인코더 (200) 는 잔차 블록의 양자화된 변환 계수들을 스캔할 수도 있다 (556). 스캔 동안 또는 스캔 후에, 비디오 인코더 (200) 는 계수들을 엔트로피 인코딩할 수도 있다 (558). 예를 들어, 비디오 인코더 (200) 는 CAVLC 또는 CABAC 를 사용하여 계수들을 인코딩할 수도 있다. 비디오 인코더 (200) 는 블록의 엔트로피 코딩된 데이터를 출력할 수도 있다 (560).

[0252] 도 16 은 비디오 데이터의 현재 블록을 디코딩하기 위한 비디오 디코더의 예시적인 동작을 나타내는 플로우차트이다. 현재 블록은 현재 CU 를 포함할 수도 있다. 비디오 디코더 (300) (도 1 및 도 3) 와 관련하여 설명되었지만, 다른 디바이스들이 도 16 의 동작과 유사한 동작을 수행하도록 구성될 수도 있음을 이해하여야 한다.

[0253] 비디오 디코더 (300) 는 현재 블록에 대응하는 잔차 블록의 계수들에 대한 엔트로피 코딩된 예측 정보 및 엔트로피 코딩된 데이터와 같은, 현재 블록에 대한 엔트로피 코딩된 데이터를 수신할 수도 있다 (570). 비디오 디코더 (300) 는 엔트로피 코딩된 데이터를 엔트로피 디코딩하여 현재 블록에 대한 예측 정보를 결정하고 잔차 블록의 계수들을 재생할 수도 있다 (572). 비디오 디코더 (300) 는 현재 블록에 대한 예측 블록을 계산하기

위해, 예를 들어 현재 블록에 대한 예측 정보에 의해 표시된 것과 같은 인트라 예측 또는 인터 예측 모드를 사용하여 현재 블록을 예측할 수도 있다 (574). 비디오 디코더 (300) 는 그 다음, 양자화된 변환 계수들의 블록을 생성하기 위해, 재생된 계수들을 역 스캔할 수도 있다 (576). 그 후에, 비디오 디코더 (300) 는 계수들을 역 양자화 및 역 변환하여 잔차 블록을 생성할 수도 있다 (578). 비디오 디코더 (300) 는 예측 블록과 잔차 블록을 결합함으로써 궁극적으로 현재 블록을 디코딩할 수도 있다 (580).

[0254] 도 17 은 계수 값들을 디코딩하기 위한 비디오 디코더의 예시적 동작을 나타내는 플로우차트이다. 비디오 디코더 (300) (도 1 및 도 10) 와 관련하여 설명되었지만, 다른 디바이스들이 도 17 의 동작과 유사한 동작을 수행하도록 구성될 수도 있음을 이해하여야 한다.

[0255] 비디오 디코더 (300) 는 제 1 디코딩 패스에 대한 정규 코딩된 빈들의 임계 수를 결정한다 (602).

[0256] 계수들의 제 1 세트에 대해, 비디오 디코더 (300) 는 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들을 컨텍스트 디코딩한다 (604). 신택스 엘리먼트들의 컨텍스트 디코딩된 빈들은, 예를 들어, 전술한 바와 같이, 하나 이상의 유의성 플래그들, 하나 이상의 패리티 레벨 플래그들, 및 하나 이상의 제 1 플래그들을 포함할 수도 있다. 하나 이상의 유의성 플래그들의 각각은 계수에 대한 절대 레벨이 0과 동일한지를 표시할 수도 있고, 하나 이상의 패리티 레벨 플래그들의 각각은 계수가 짝수 또는 홀수인 절대 레벨을 갖는지를 표시할 수도 있다. 하나 이상의 제 1 플래그들의 각각은 계수가 2 보다 큰 절대 레벨을 갖는지를 나타낼 수도 있다.

[0257] 계수 그룹의 신택스 엘리먼트들을 컨텍스트 디코딩하기 위해, 비디오 디코더 (300) 는 계수 그룹의 신택스 엘리먼트들을 디코딩하기 위해 컨텍스트-적응적 이진 산술 디코딩을 수행할 수도 있다. 다른 예들에서, 정규 코딩된 빈들의 임계 수에 도달할 때까지 계수 그룹의 신택스 엘리먼트들을 컨텍스트 디코딩하기 위해, 비디오 디코더 (300) 는, 계수들의 제 1 세트의 계수에 대한 신택스 엘리먼트를 코딩하는 동안 정규 코딩된 빈들의 임계 수에 도달한 것으로 결정하고, 계수들의 제 1 세트의 계수에 대한 하나 이상의 잔여 신택스 엘리먼트들을 컨텍스트 디코딩할 수도 있다.

[0258] 비디오 디코더 (300) 는 신택스 엘리먼트들의 컨텍스트 디코딩된 빈들에 기초하여 변환 유닛의 계수들의 제 1 세트에 대한 값들을 결정한다 (606). 정규 코딩된 빈들의 임계 수에 도달하는 것에 응답하여, 계수들의 제 2 세트에 대해, 비디오 디코더 (300) 는 추가적인 신택스 엘리먼트들을 바이패스 디코딩한다 (608). 추가적인 신택스 엘리먼트들을 바이패스 디코딩하기 위해, 비디오 디코더 (300) 는, 계수들의 제 2 세트의 계수에 대해, 라이스 파라미터에 대한 값을 도출할 수도 있다.

[0259] 비디오 디코더 (300) 는 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정한다 (610). 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하기 위해, 비디오 디코더 (300) 는 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정한다 (612). 라이스 파라미터에 기초하여 제로 파라미터에 대한 값을 결정하기 위해, 비디오 디코더 (300) 는, 예를 들어, 라이스 파라미터에 기초하여 그리고 또한 상태 머신의 현재 상태에 기초하여 제로 파라미터에 대한 값을 결정할 수도 있다. 전술한 바와 같이, 제로 파라미터에 대한 값은 제로의 계수 레벨에 대응하는 코딩된 값을 식별한다. 비디오 디코더 (300) 는 예를 들어, 룩업 테이블 (look up table) 로부터 또는 일부 다른 방식으로 라이스 파라미터에 대한 값을 결정할 수도 있다.

[0260] 추가적인 신택스 엘리먼트들에 기초하여 변환 유닛의 계수들의 제 2 세트에 대한 값들을 결정하기 위해, 비디오 디코더 (300) 는 또한 계수들의 제 2 세트 중 제 1 계수에 대한 제 1 코딩된 값을 수신하고 (614), 제로 파라미터에 대한 값 및 제 1 계수에 대한 제 1 코딩된 값에 기초하여, 제 1 계수에 대한 레벨을 결정한다 (616). 제1 계수에 대한 레벨은, 예를 들어, 잔여 레벨 또는 절대 레벨 중 어느 일방일 수도 있다.

[0261] 제로 파라미터에 대한 값이 제 1 코딩된 값과 동일한 것에 응답하여, 비디오 디코더 (300) 는 제 1 계수에 대한 레벨이 제로와 동일하다고 결정할 수도 있다. 제 1 코딩된 값이 제로 파라미터에 대한 값보다 큰 것에 응답하여, 비디오 디코더 (300) 는 제 1 계수에 대한 레벨이 제 1 코딩된 값과 동일한 것을 결정할 수도 있다. 다른 경우들에서, 제 1 코딩된 값이 제로 파라미터에 대한 값보다 작은 것에 응답하여, 비디오 디코더 (300) 는 제 1 계수에 대한 레벨이 제 1 코딩된 값 플러스 1 과 동일한 것을 결정할 수도 있다.

[0262] 비디오 디코더 (300) 는 또한, 계수들의 제 1 세트에 대한 값들 및 계수들의 제 2 세트에 대한 값들에 기초하여 디코딩된 변환 블록을 결정하고; 디코딩된 변환 블록을 예측 블록에 가산하여 재구성된 블록을 결정하고; 재구성된 블록에 대해 하나 이상의 필터링 동작들을 수행하여 비디오 데이터의 디코딩된 블록을 결정하고; 비디오

데이터의 디코딩된 블록을 포함하는 비디오 데이터의 디코딩된 픽처를 출력할 수도 있다.

[0263] 예에 의존하여, 본 명세서에서 설명된 기법들의 임의의 것의 특정 행위들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있고, 전체적으로 부가되거나 병합되거나 또는 제거될 수도 있음 (예를 들어, 설명된 모든 행위들 또는 이벤트들이 그 기법들의 실시를 위해 필수적인 것은 아님) 이 인식되어야 한다. 더욱이, 특정 예들에 있어서, 행위들 또는 이벤트들은 순차적인 것보다는, 예를 들어, 다중-스레딩된 프로세싱, 인터럽트 프로세싱, 또는 다중의 프로세서들을 통해 동시에 수행될 수도 있다.

[0264] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 그 기능들은 컴퓨터 판독가능 매체 상의 하나 이상의 명령 또는 코드로서 저장되거나 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예를 들어, 통신 프로토콜에 따라, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 캐리어파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품이 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0265] 제한이 아닌 일 예로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 원하는 프로그램 코드를 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체 중 하나 이상을 포함할 수 있다. 또한, 임의의 커넥션이 컴퓨터 판독가능 매체로 적절히 명명된다. 예를 들어, 동축 케이블, 광섬유 케이블, 꼬임쌍선, 디지털 가입자 라인 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 소프트웨어가 송신되는 경우, 그 동축 케이블, 광섬유 케이블, 꼬임쌍선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 캐리어파들, 신호들 또는 다른 일시적 매체들을 포함하는 것이 아니라, 대신에 비일시적, 유형의 저장 매체에 관련된다는 것이 이해되야 한다. 디스크 (disk) 및 디스크 (disc) 는, 본 명세서에서 사용된 바와 같이, 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 및 블루-레이 디스크를 포함하고, 여기서 디스크 (disk) 들은 보통 데이터를 자기적으로 재생하는 한편, 디스크 (disc) 들은 레이저들로 데이터를 광학적으로 재생한다. 상기의 조합들이 또한, 컴퓨터 판독가능 매체들의 레인지 내에 포함되어야 한다.

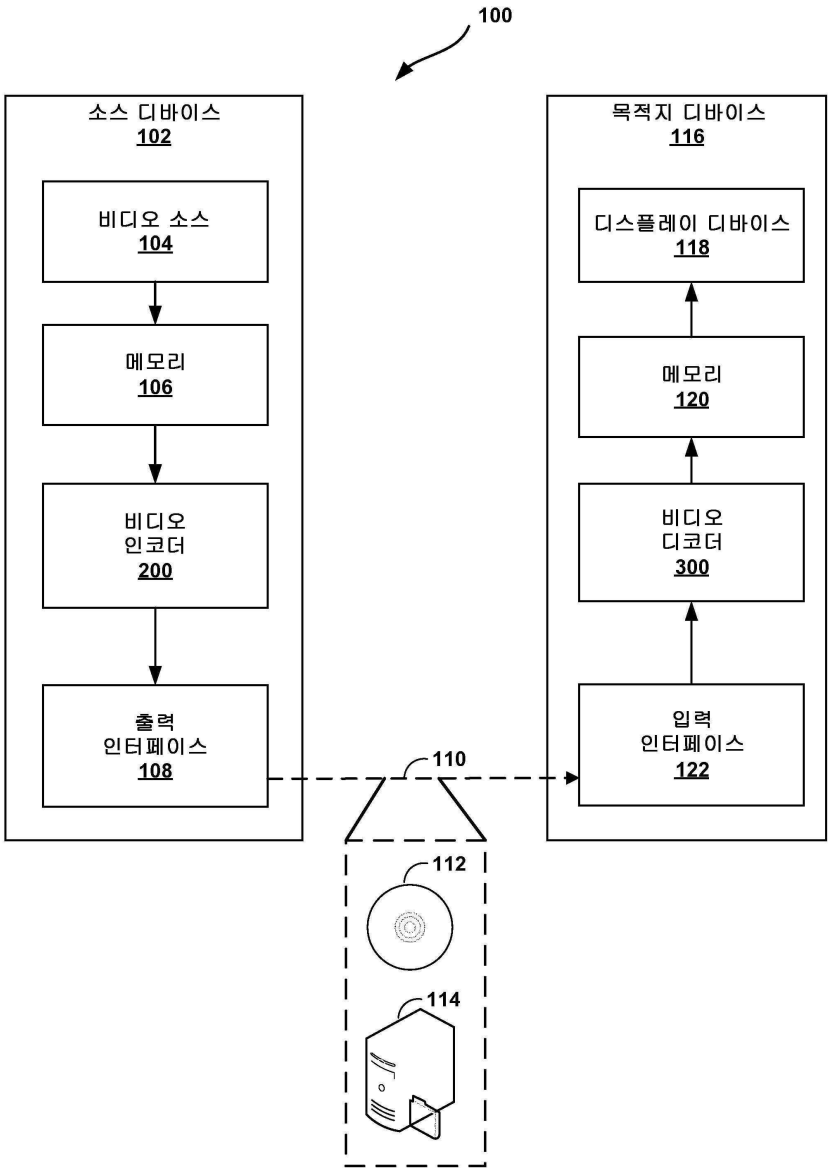
[0266] 명령들은 하나 이상의 프로세서들, 이를 테면 하나 이상의 DSP들, 범용 마이크로프로세서들, ASIC들, FPGA들, 또는 다른 등가의 집적 또는 이산 로직 회로부에 의해 실행될 수도 있다. 이에 따라, 용어 "프로세서" 는, 본 명세서에서 사용된 바와 같이, 전술한 구조 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 추가로, 일부 양태들에서, 본 명세서에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되거나, 또는 결합된 코덱에 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 그 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0267] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들어, 칩 세트) 를 포함하여, 광범위하게 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시에 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하는 것은 아니다. 오히려, 상기 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에서 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상기 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호동작가능한 하드웨어 유닛들의 콜렉션에 의해 제공될 수도 있다.

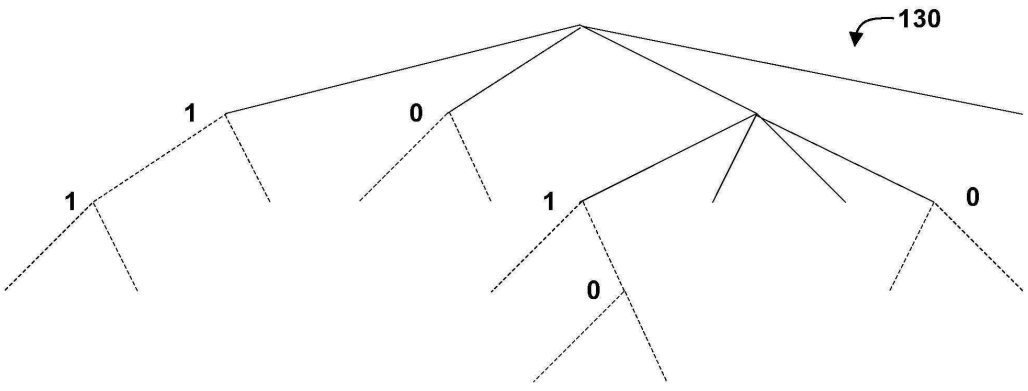
[0268] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

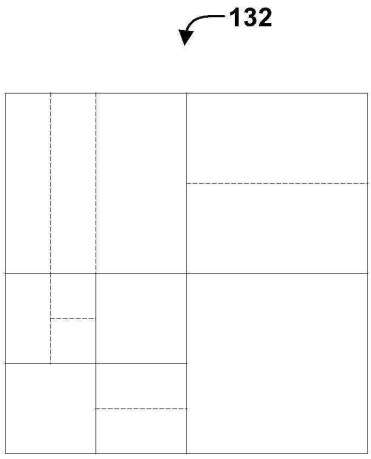
도면1


100
소스 디바이스
102
비디오 소스
104
메모리
106
비디오
인코더
200
출력
인터페이스
108
110
112
114
목적지 디바이스
116
디스플레이 디바이스
118
메모리
120
비디오
디코더
300
입력
인터페이스
122

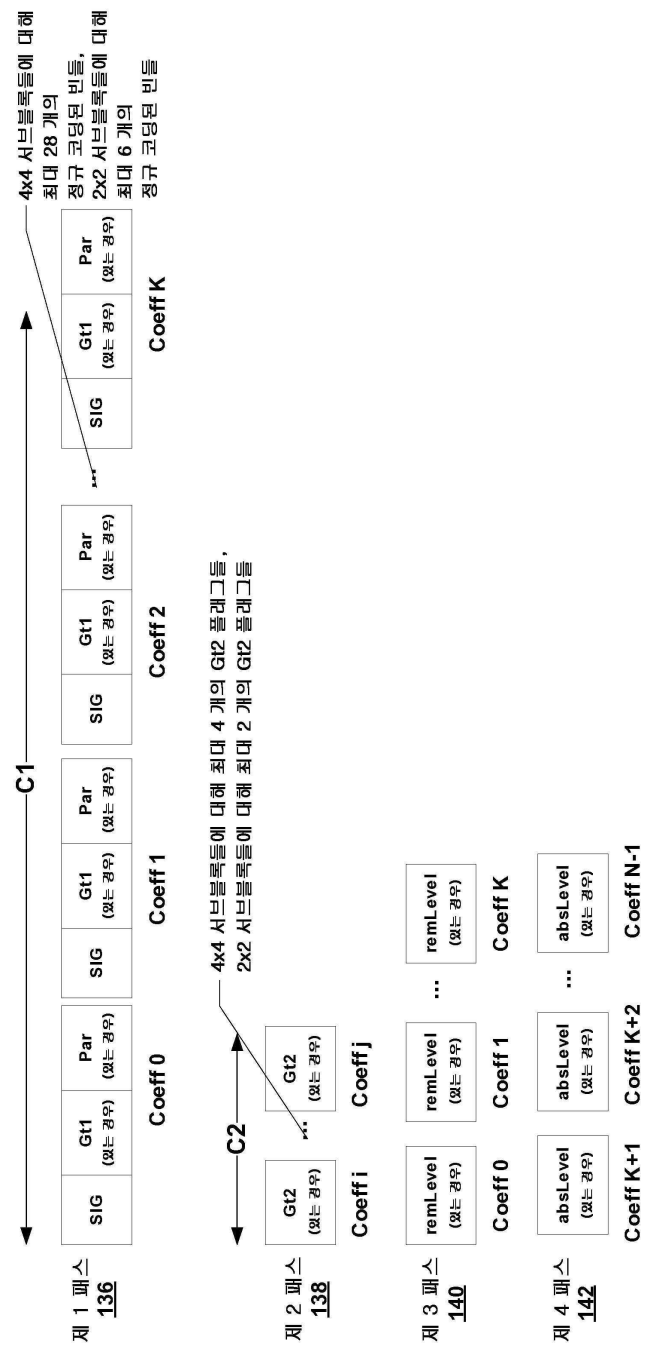
도면2a


130
1
0
1
1
0
0

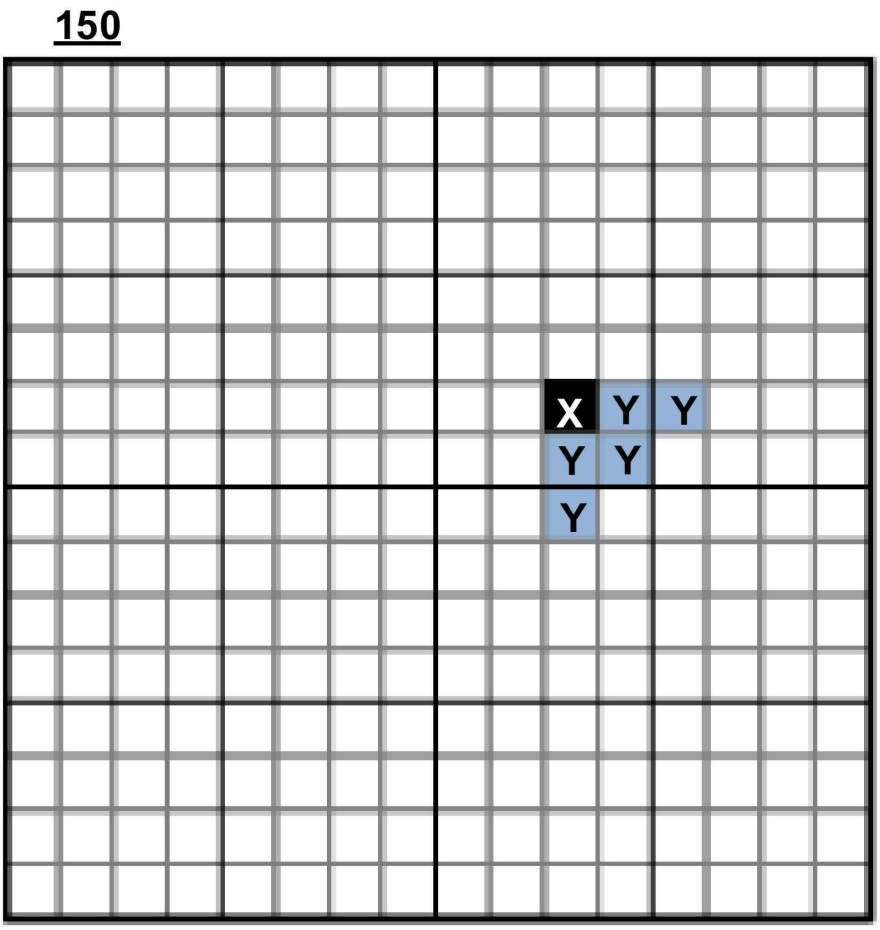
도면2b



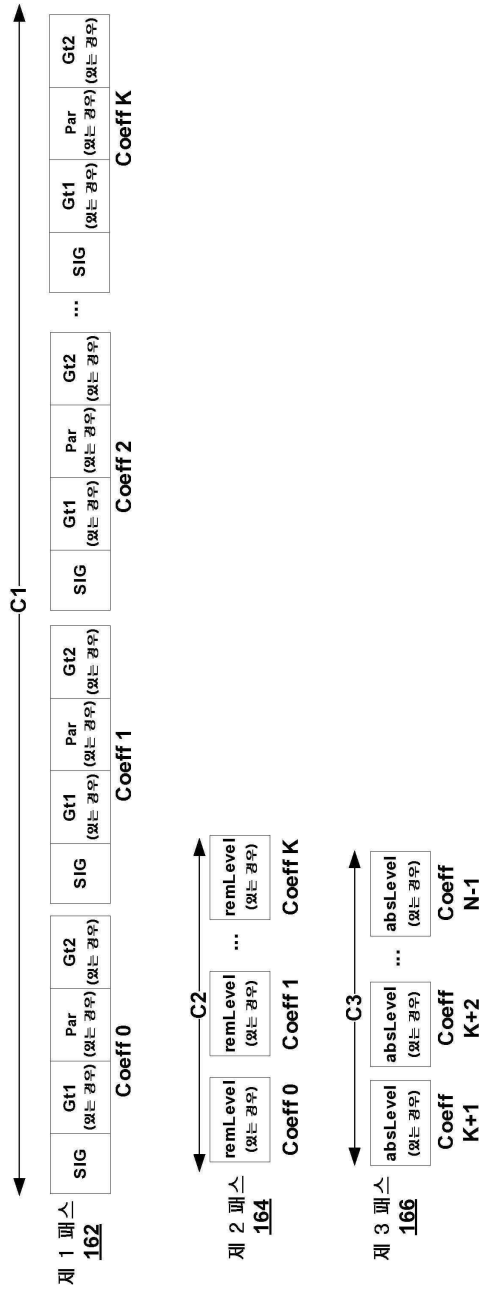
도면3


4x4 서브블록들에 대해
최대 28 개의
정규 코딩된 빈들,
2x2 서브블록들에 대해
최대 6 개의
정규 코딩된 빈들
C1
C2
4x4 서브블록들에 대해 최대 4 개의 Gt2 플래그들,
2x2 서브블록들에 대해 최대 2 개의 Gt2 플래그들
제 1 패스
136
제 2 패스
138
제 3 패스
140
제 4 패스
142
SIG
Gt1
(있는 경우)
Par
(있는 경우)
Gt2
(있는 경우)
remLevel
(있는 경우)
absLevel
(있는 경우)
Coeff 0
Coeff 1
Coeff 2
Coeff K
Coeff i
Coeff j
Coeff K+1
Coeff K+2
Coeff N-1

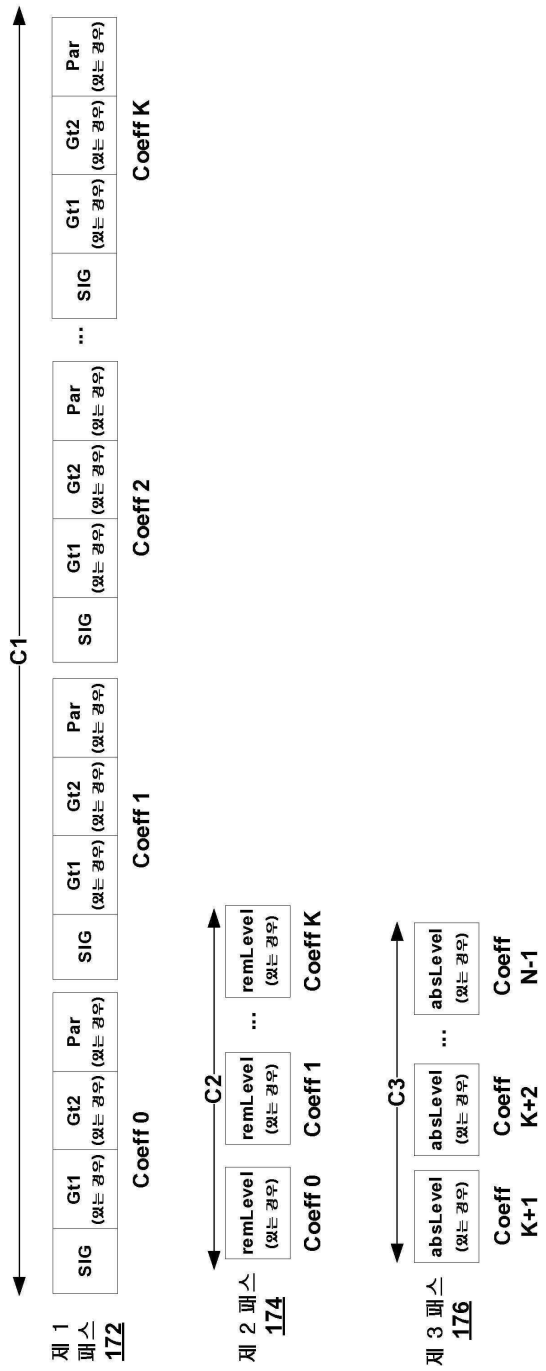
도면4



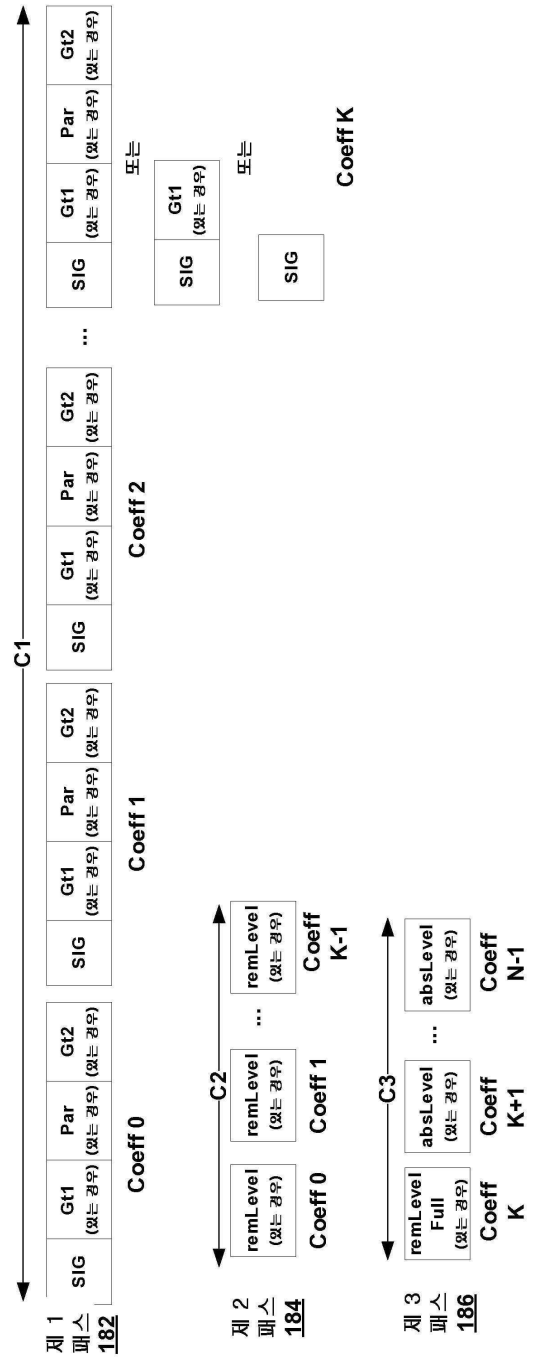
도면5


C1
제 1 패스
162
SIG
Gt1 (있는 경우)
Par (있는 경우)
Gt2 (있는 경우)
Coeff 0
Coeff 1
Coeff 2
...
Coeff K
C2
제 2 패스
164
remLevel (있는 경우)
Coeff 0
Coeff 1
Coeff K
C3
제 3 패스
166
absLevel (있는 경우)
Coeff K+1
Coeff K+2
Coeff N-1

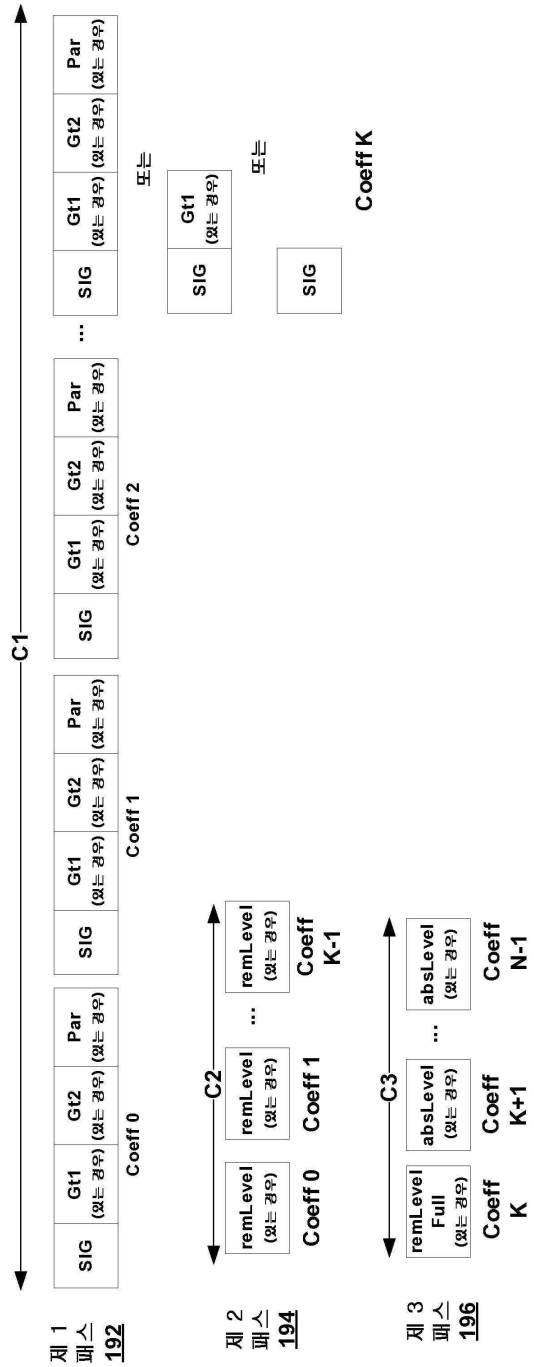
도면6


C1
제 1 패스 172
SIG
Gt1 (있는 경우)
Gt2 (있는 경우)
Par (있는 경우)
Coeff 0
Coeff 1
Coeff 2
...
Coeff K
C2
제 2 패스 174
remLevel (있는 경우)
Coeff 0
Coeff 1
Coeff K
C3
제 3 패스 176
absLevel (있는 경우)
Coeff K+1
Coeff K+2
Coeff N-1

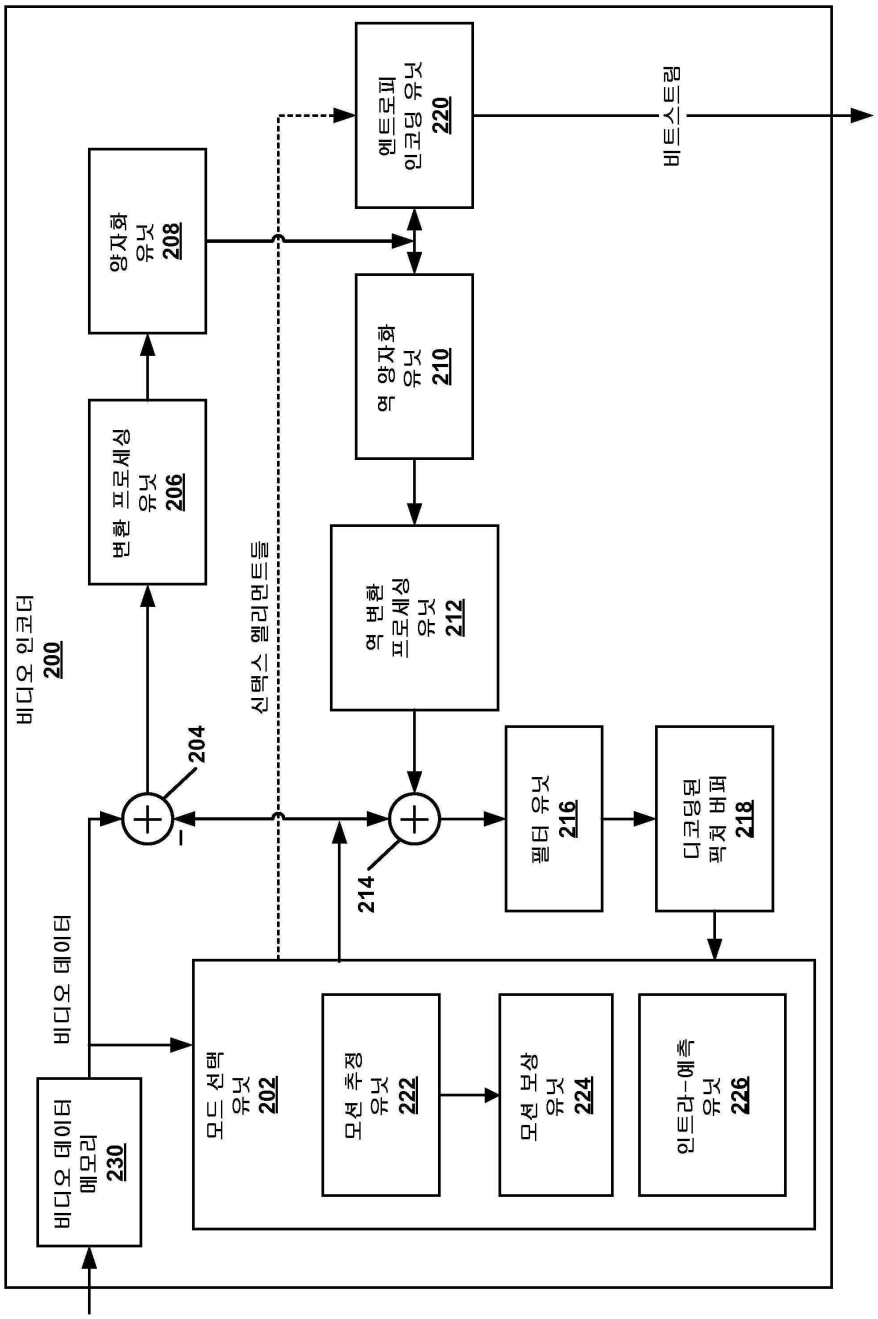
도면7


C1
제 1 패스 182
SIG
Gt1 (있는 경우)
Par (있는 경우)
Gt2 (있는 경우)
Coeff 0
Coeff 1
Coeff 2
...
또는
Coeff K
C2
제 2 패스 184
remLevel (있는 경우)
Coeff 0
Coeff 1
Coeff K-1
C3
제 3 패스 186
remLevel Full (있는 경우)
absLevel (있는 경우)
Coeff K
Coeff K+1
Coeff N-1

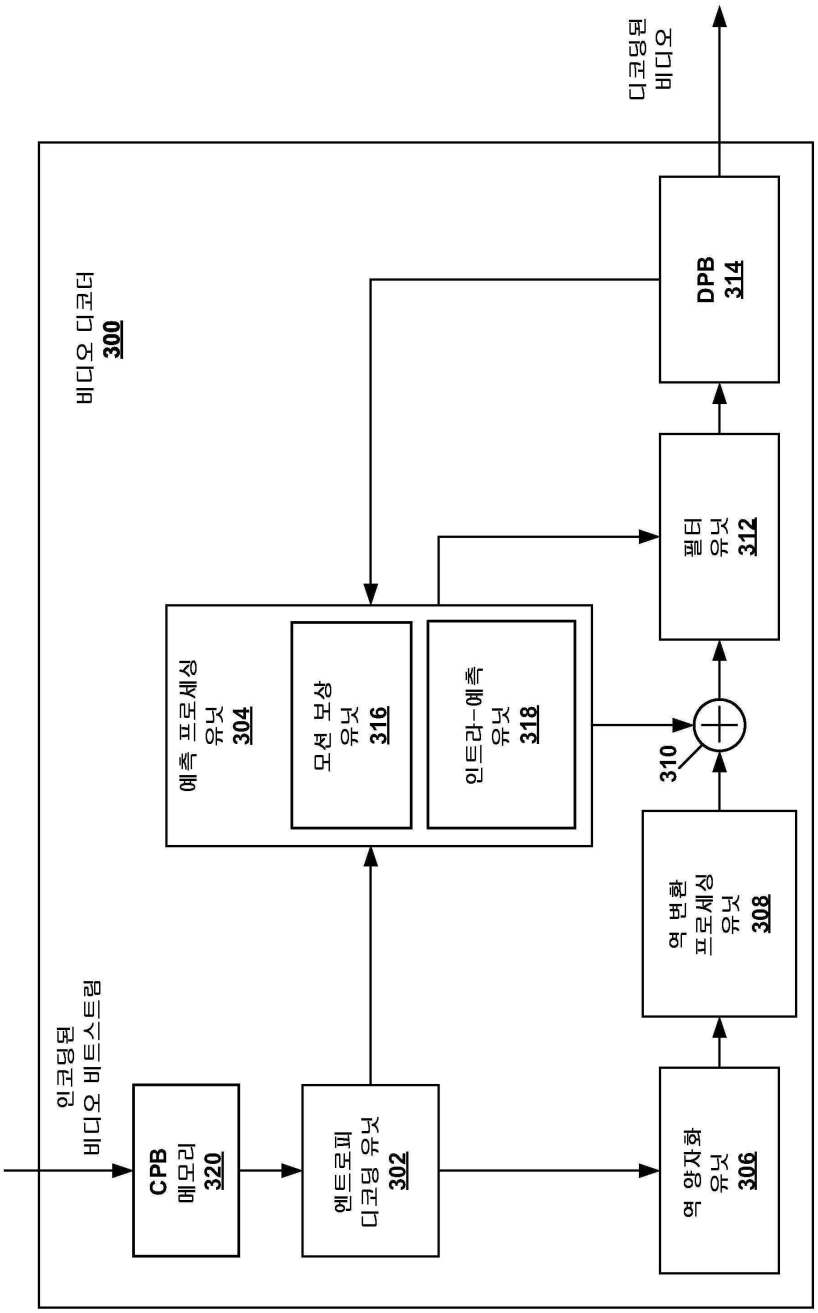
도면8


C1
제 1 패스 192
SIG
Gt1 (있는 경우)
Gt2 (있는 경우)
Par (있는 경우)
Coeff 0
Coeff 1
Coeff 2
...
Coeff K
또는
C2
제 2 패스 194
remLevel (있는 경우)
Coeff 0
Coeff 1
Coeff K-1
C3
제 3 패스 196
remLevel Full (있는 경우)
absLevel (있는 경우)
Coeff K
Coeff K+1
Coeff N-1

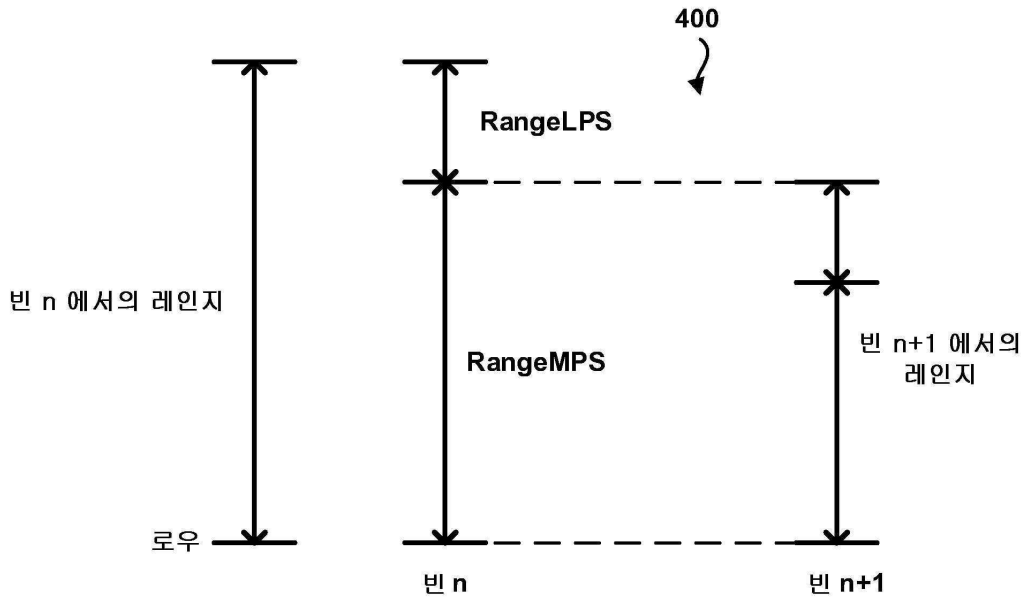
도면9


비디오 인코더
200
비디오 데이터
비디오 데이터
메모리
230
모드 선택
유닛
202
모션 추정
유닛
222
모션 보상
유닛
224
인트라-예측
유닛
226
204
-
214
필터 유닛
216
디코딩된
픽처 버퍼
218
신택스 엘리먼트들
역 변환
프로세싱
유닛
212
역 양자화
유닛
210
변환 프로세싱
유닛
206
양자화
유닛
208
엔트로피
인코딩 유닛
220
비트스트림

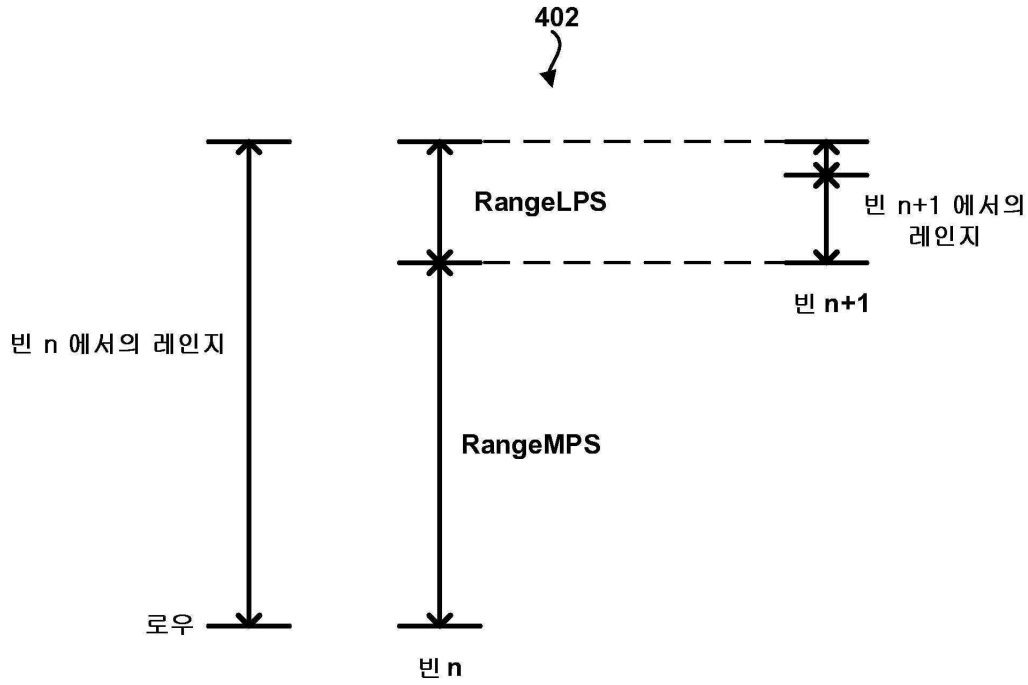
도면10


인코딩된
비디오 비트스트림
비디오 디코더
300
CPB
메모리
320
엔트로피
디코딩 유닛
302
역 양자화
유닛
306
역 변환
프로세싱
유닛
308
310
예측 프로세싱
유닛
304
모션 보상
유닛
316
인트라-예측
유닛
318
필터
유닛
312
DPB
314
디코딩된
비디오

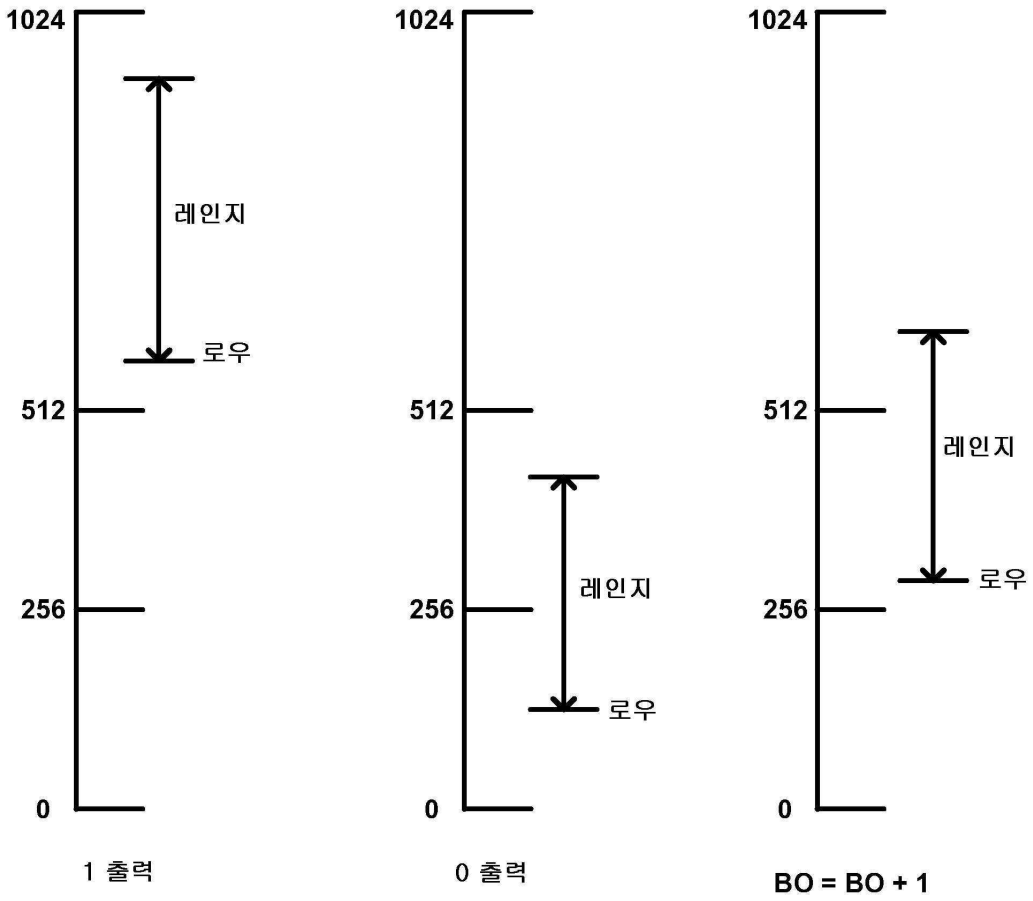
도면11a


400
RangeLPS
빈 n 에서의 레인지
RangeMPS
빈 n+1 에서의
레인지
로우
빈 n
빈 n+1

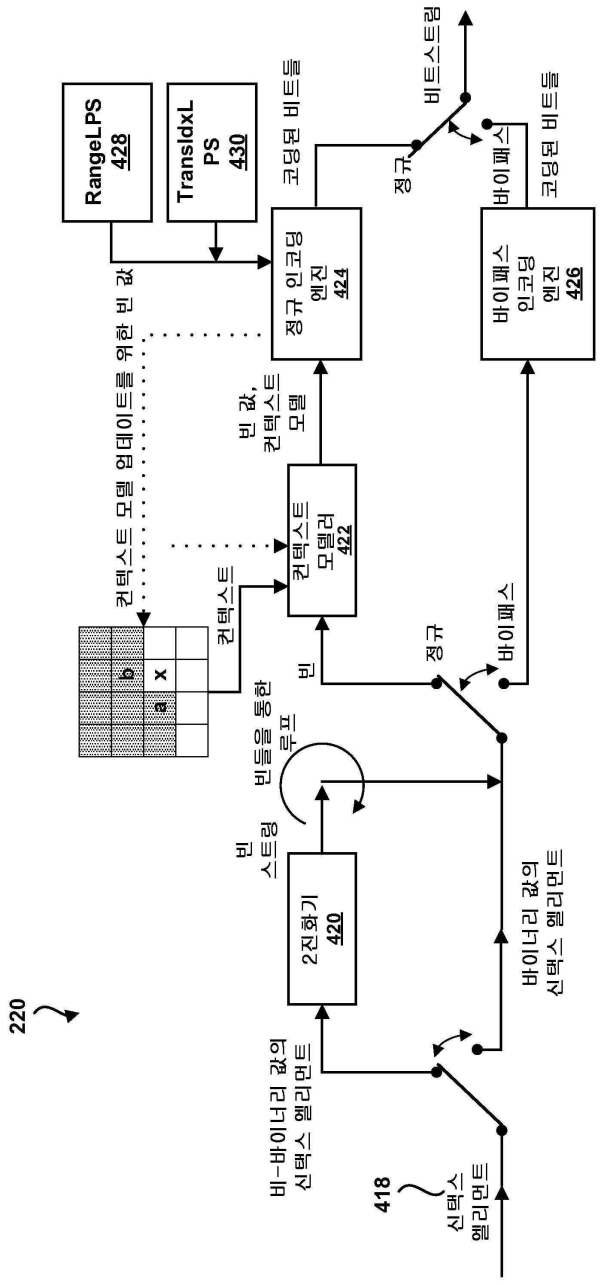
도면11b


402
RangeLPS
빈 n+1 에서의
레인지
빈 n+1
빈 n 에서의 레인지
RangeMPS
로우
빈 n

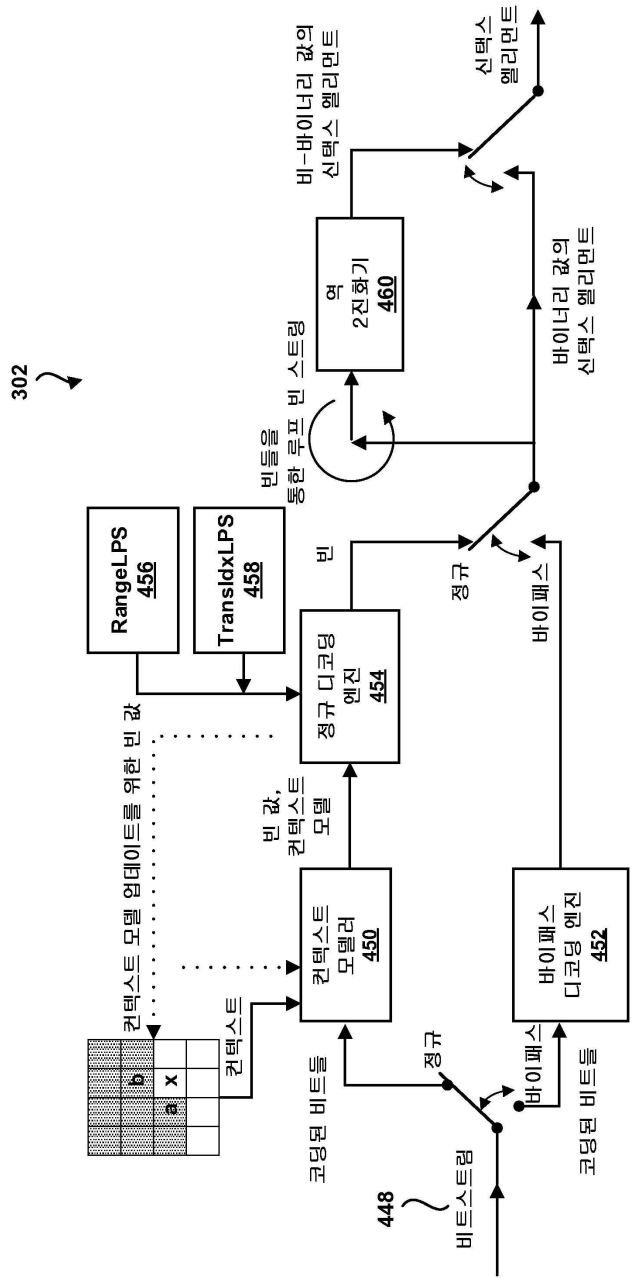
도면12


1024
512
256
0
레인지
로우
1 출력
1024
512
256
0
레인지
로우
0 출력
1024
512
256
0
레인지
로우
BO = BO + 1

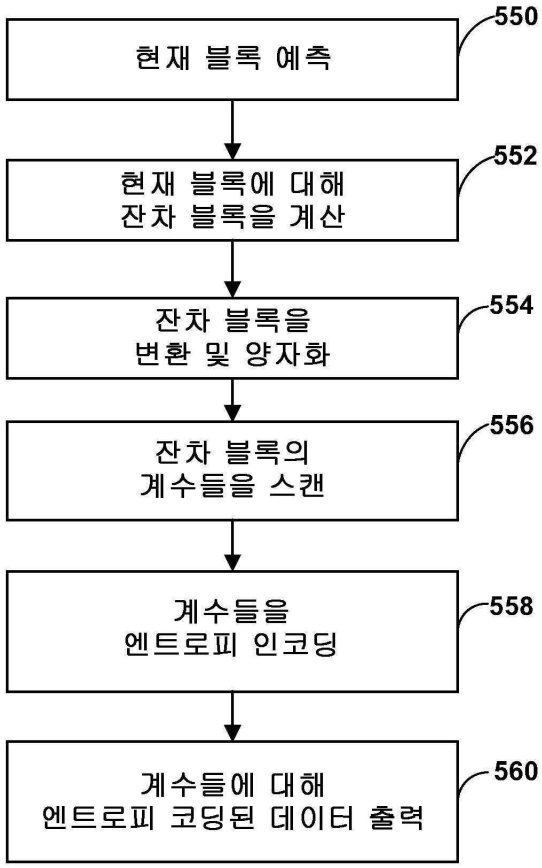
도면13


220
RangeLPS 428
TransIdxL PS 430
정규 인코딩 엔진 424
바이패스 인코딩 엔진 426
코딩된 비트들
비트스트림
정규
바이패스
빈 값, 컨텍스트 모델
컨텍스트 모델 업데이트를 위한 빈 값
컨텍스트 모델러 422
컨텍스트
빈
정규
바이패스
빈들을 통한 루프
빈 스트링
2진화기 420
비-바이너리 값의 신택스 엘리먼트
418
신택스 엘리먼트
바이너리 값의 신택스 엘리먼트

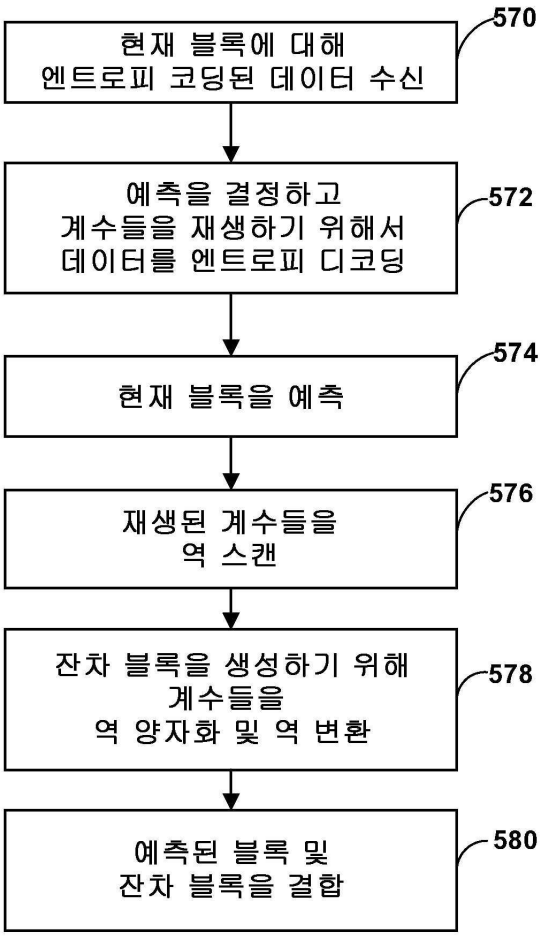
도면14


302
RangeLPS 456
TransldxLPS 458
컨텍스트 모델 업데이트를 위한 빈 값
컨텍스트
b
x
a
컨텍스트 모델러 450
빈 값, 컨텍스트 모델
정규 디코딩 엔진 454
빈
빈들을 통한 루프
역 2진화기 460
비-바이너리 값의 신택스 엘리먼트
신택스 엘리먼트
바이너리 값의 신택스 엘리먼트
정규
바이패스
바이패스 디코딩 엔진 452
코딩된 비트들
448
비트스트림

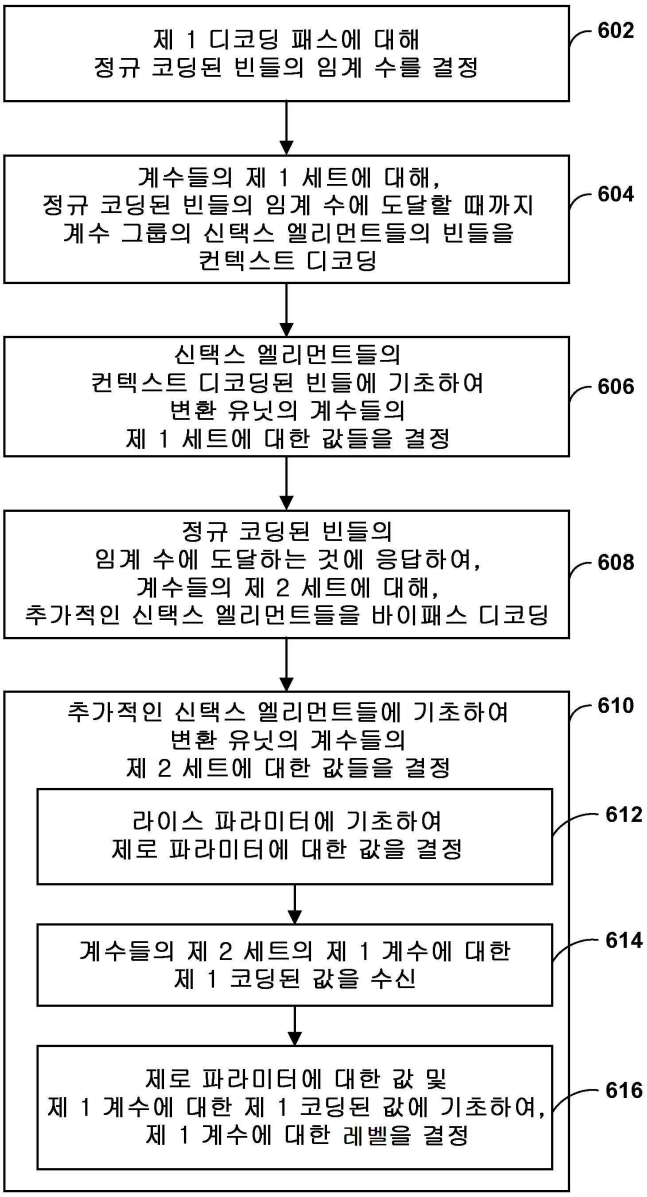
도면15


현재 블록 예측
550
현재 블록에 대해
잔차 블록을 계산
552
잔차 블록을
변환 및 양자화
554
잔차 블록의
계수들을 스캔
556
계수들을
엔트로피 인코딩
558
계수들에 대해
엔트로피 코딩된 데이터 출력
560

도면16


현재 블록에 대해
엔트로피 코딩된 데이터 수신
570
예측을 결정하고
계수들을 재생하기 위해서
데이터를 엔트로피 디코딩
572
현재 블록을 예측
574
재생된 계수들을
역 스캔
576
잔차 블록을 생성하기 위해
계수들을
역 양자화 및 역 변환
578
예측된 블록 및
잔차 블록을 결합
580

도면17


제 1 디코딩 패스에 대해
정규 코딩된 빈들의 임계 수를 결정
602
계수들의 제 1 세트에 대해,
정규 코딩된 빈들의 임계 수에 도달할 때까지
계수 그룹의 신택스 엘리먼트들의 빈들을
컨텍스트 디코딩
604
신택스 엘리먼트들의
컨텍스트 디코딩된 빈들에 기초하여
변환 유닛의 계수들의
제 1 세트에 대한 값들을 결정
606
정규 코딩된 빈들의
임계 수에 도달하는 것에 응답하여,
계수들의 제 2 세트에 대해,
추가적인 신택스 엘리먼트들을 바이패스 디코딩
608
추가적인 신택스 엘리먼트들에 기초하여
변환 유닛의 계수들의
제 2 세트에 대한 값들을 결정
610
라이스 파라미터에 기초하여
제로 파라미터에 대한 값을 결정
612
계수들의 제 2 세트의 제 1 계수에 대한
제 1 코딩된 값을 수신
614
제로 파라미터에 대한 값 및
제 1 계수에 대한 제 1 코딩된 값에 기초하여,
제 1 계수에 대한 레벨을 결정
616