



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0165208
(43) 공개일자 2023년12월05일

- | | |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)
H04N 19/13 (2014.01) H04N 19/169 (2014.01)
H04N 19/176 (2014.01) H04N 19/70 (2014.01)</p> <p>(52) CPC특허분류
H04N 19/13 (2015.01)
H04N 19/176 (2015.01)</p> <p>(21) 출원번호 10-2023-7031435</p> <p>(22) 출원일자(국제) 2022년04월08일
심사청구일자 없음</p> <p>(85) 번역문제출일자 2023년09월14일</p> <p>(86) 국제출원번호 PCT/US2022/071628</p> <p>(87) 국제공개번호 WO 2022/217272
국제공개일자 2022년10월13일</p> <p>(30) 우선권주장
63/173,269 2021년04월09일 미국(US)
17/658,396 2022년04월07일 미국(US)</p> | <p>(71) 출원인
켈컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775</p> <p>(72) 발명자
루사노브스키 드미트로
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>(74) 대리인
특허법인코리아나</p> |
|---|--|

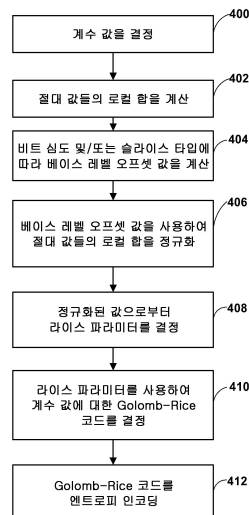
전체 청구항 수 : 총 40 항

(54) 발명의 명칭 높은 비트 심도 비디오 데이터를 코딩하기 위한 라이스 파라미터 도출

(57) 요약

비디오 데이터를 코딩 (인코딩 및/또는 디코딩) 하기 위한 예시적인 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리; 및 회로부에서 구현되는 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하도록 구성된다.

대표도 - 도9



(52) CPC특허분류

H04N 19/1887 (2015.01)

H04N 19/70 (2015.01)

명세서

청구범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

비디오 데이터의 현재 블록을 위한 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계,

상기 비디오 데이터에 대한 비트 심도 또는 상기 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계,

상기 베이스 레벨 오프셋 값을 사용하여 상기 절대 값들의 합에 대한 정규화된 값을 계산하는 단계, 및

상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 코딩하는 단계

를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 베이스 레벨 오프셋 값을 계산하는 단계는, $\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1 : P2) : (\text{isIntra}() ? P3 : P4)$ 를 계산하는 단계를 포함하고, baseLevelOffset 는 상기 베이스 레벨 오프셋 값이고, bitDepth 는 상기 비디오 데이터에 대한 상기 비트 심도를 포함하고, $\text{isIntra}()$ 는 상기 슬라이스에 대한 상기 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 상기 슬라이스에 대한 상기 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키며, T , $P1$, $P2$, $P3$, 및 $P4$ 는 각각의 입력 값들을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 5

제 4 항에 있어서,

$T = 12$ 인, 비디오 데이터를 코딩하는 방법.

청구항 6

제 5 항에 있어서,

$P1 = 5$, $P2 = 10$, $P3 = 6$, 및 $P4 = 20$ 인, 비디오 데이터를 코딩하는 방법.

청구항 7

제 5 항에 있어서,

$P1 = 5$, $P2 = 10$, $P3 = 10$, 및 $P4 = 20$ 인, 비디오 데이터를 코딩하는 방법.

청구항 8

제 4 항에 있어서,

T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 미리 결정되는, 비디오 데이터를 코딩하는 방법.

청구항 9

제 4 항에 있어서,

T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 비트스트림에서 시그널링되는, 비디오 데이터를 코딩하는 방법.

청구항 10

제 1 항에 있어서,

상기 절대 값들의 합에 대한 상기 정규화된 값을 계산하는 단계는 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하는 단계를 포함하고, locSumAbs 는 상기 절대 값들의 합을 포함하고 baseLevelOffset 은 상기 베이스 레벨 오프셋 값을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 11

제 1 항에 있어서,

상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 코딩하는 단계는,

상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 라이스 파라미터를 결정하는 단계, 및

상기 라이스 파라미터를 사용하여 상기 현재 계수에 대한 값을 이진화하여 이진화된 값을 형성하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 12

제 1 항에 있어서,

상기 현재 블록은 제 1 블록을 포함하고, 상기 비디오 데이터의 컬러 컴포넌트는 상기 제 1 블록을 포함하며, 상기 절대 값들의 합은 절대 값들의 제 1 합을 포함하고, 상기 복수의 이웃 계수들은 제 1 복수의 이웃 계수들을 포함하고,

상기 방법은,

상기 비디오 데이터의 상기 컬러 컴포넌트의 계수들을 코딩하기 위해 사용된 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하는 단계,

상기 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 상기 히스토리 카운터에 대한 현재 값으로부터 상기 비디오 데이터의 상기 컬러 컴포넌트의 제 2 블록에 대한 라이스 파라미터 스케일링 값을 도출하는 단계,

상기 제 2 블록의 제 2 계수에 대한 제 2 복수의 이웃 계수들의 절대 값들의 제 2 합을 계산하는 단계,

상기 라이스 파라미터 스케일링 값을 사용하여 상기 절대 값들의 제 2 합에 대한 스케일링된 값을 계산하는 단계, 및

상기 절대 값들의 제 2 합에 대한 상기 스케일링된 값을 사용하여 상기 제 2 블록의 상기 제 2 계수를 코딩하는 단계

를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 13

제 12 항에 있어서,

상기 라이스 파라미터 스케일링 값을 도출하는 단계는, $(\text{StatCoeff}[\text{idx}] \geq T) ? P2 : P1$ 를 계산하는 단계를 포함하고, idx 는 상기 컬러 컴포넌트를 나타내고, $\text{StatCoeff}[\text{idx}]$ 는 상기 히스토리 카운터에 대한 상기 현재 값을 나타내며, T , $P1$, 및 $P2$ 는 입력 값들을 나타내는, 비디오 데이터를 코딩하는 방법.

청구항 14

제 13 항에 있어서,

T , $P1$, 및 $P2$ 에 대한 상기 입력 값들은 미리 결정되는, 비디오 데이터를 코딩하는 방법.

청구항 15

제 13 항에 있어서,

T , $P1$, 및 $P2$ 에 대한 상기 입력 값들은 상기 비디오 데이터에서 시그널링되는, 비디오 데이터를 코딩하는 방법.

청구항 16

제 13 항에 있어서,

$T = 4$, $P2 = 2$, 및 $P1 = 1$ 인, 비디오 데이터를 코딩하는 방법.

청구항 17

제 12 항에 있어서,

상기 라이스 파라미터 스케일링 값을 도출하는 단계는,

상기 비디오 데이터의 상기 컬러 컴포넌트를 포함하는 모든 컬러 컴포넌트들에 대한, 상기 히스토리 카운터를 포함하는 히스토리 카운터들의 집성을 나타내는 집성된 히스토리 카운터 값을 계산하는 단계, 및

상기 집성된 히스토리 카운터 값을 사용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 18

제 17 항에 있어서,

상기 집성된 히스토리 카운터 값을 계산하는 단계는 $((\text{StatCoeff}[0] * w0) + \text{StatCoeff}[1]*w1 + \text{StatCoeff}[2]*w2)/(w1+w2+w3)$ 를 계산하는 단계를 포함하고, $\text{StatCoeff}[0]$ 는 상기 컬러 컴포넌트들의 제 1 컬러 컴포넌트에 대한 상기 히스토리 카운터들의 제 1 히스토리 카운터를 나타내고, $\text{StatCoeff}[1]$ 는 상기 컬러 컴포넌트들의 제 2 컬러 컴포넌트에 대한 상기 히스토리 카운터들의 제 2 히스토리 카운터를 나타내며, $\text{StatCoeff}[2]$ 는 상기 컬러 컴포넌트들의 제 3 컬러 컴포넌트에 대한 상기 히스토리 카운터들의 제 3 히스토리 카운터를 나타내고, $w1$, $w2$, 및 $w3$ 은 각각의 가중치 값들을 나타내는, 비디오 데이터를 코딩하는 방법.

청구항 19

제 18 항에 있어서,

상기 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoefAverage} \geq T) ? P2 : P1$ 를 사용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함하고, StatCoefAverage 는 상기 집성된 히스토리 카운터 값을 나타내고, T , $P1$ 및 $P2$ 는 입력 값들을 나타내는, 비디오 데이터를 코딩하는 방법.

청구항 20

제 19 항에 있어서,

T , $P1$, 및 $P2$ 에 대한 상기 입력 값들은 미리 결정되는, 비디오 데이터를 코딩하는 방법.

청구항 21

제 19 항에 있어서,

T, P1, 및 P2 에 대한 상기 입력 값들은 상기 비디오 데이터에서 시그널링되는, 비디오 데이터를 코딩하는 방법.

청구항 22

제 19 항에 있어서,

T = 4, P2 = 2, 및 P1 = 1 인, 비디오 데이터를 코딩하는 방법.

청구항 23

제 19 항에 있어서,

w0 = 2, w1 = 1, 및 w2 = 1 인, 비디오 데이터를 코딩하는 방법.

청구항 24

제 12 항에 있어서,

상기 제 2 블록은 코딩 트리 유닛 (CTU), 변환 유닛 (TU), 또는 서브-변환 유닛 (서브-TU) 중 하나를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 25

제 1 항에 있어서,

상기 현재 계수를 코딩하는 단계는 상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 디코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 26

제 1 항에 있어서,

상기 현재 계수를 코딩하는 단계는 상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 인코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 27

비디오 데이터를 코딩하기 위한 디바이스로서,

비디오 데이터를 저장하도록 구성된 메모리, 및

회로부에서 구현되는 하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

비디오 데이터의 현재 블록을 위한 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하고,

상기 비디오 데이터에 대한 비트 심도 또는 상기 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하며,

상기 베이스 레벨 오프셋 값을 사용하여 상기 절대 값들의 합에 대한 정규화된 값을 계산하고,

상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 코딩하도록

구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 28

제 27 항에 있어서,

상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 29

제 27 항에 있어서,

상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 30

제 27 항에 있어서,

상기 베이스 레벨 오프셋 값을 계산하기 위해서, 상기 하나 이상의 프로세서들은, $\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1 : P2) : (\text{isIntra}() ? P3 : P4)$ 를 계산하도록 구성되고, baseLevelOffset 는 상기 베이스 레벨 오프셋 값을 포함하고, bitDepth 는 상기 비디오 데이터에 대한 상기 비트 심도를 포함하고, $\text{isIntra}()$ 는 상기 슬라이스에 대한 상기 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 상기 슬라이스에 대한 상기 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키며, T, P1, P2, P3, 및 P4 는 각각의 입력 값들을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 31

제 30 항에 있어서,

T = 12 인, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 32

제 30 항에 있어서,

T, P1, P2, P3, 및 P4 에 대한 입력 값들은 미리 결정되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 33

제 27 항에 있어서,

상기 절대 값들의 합에 대한 상기 정규화된 값을 계산하기 위해서, 상기 하나 이상의 프로세서들은 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하도록 구성되고, locSumAbs 는 상기 절대 값들의 합을 포함하고 baseLevelOffset 은 상기 베이스 레벨 오프셋 값을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 34

제 27 항에 있어서,

상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 코딩하기 위해, 상기 하나 이상의 프로세서들은,

상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 라이스 파라미터를 결정하고,

상기 라이스 파라미터를 사용하여 상기 현재 계수에 대한 값을 이진화하여 이진화된 값을 형성하도록

구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 35

제 27 항에 있어서,

상기 현재 블록은 제 1 블록을 포함하고, 상기 비디오 데이터의 컬러 컴포넌트는 상기 제 1 블록을 포함하며, 상기 절대 값들의 합은 절대 값들의 제 1 합을 포함하고, 상기 복수의 이웃 계수들은 제 1 복수의 이웃 계수들을 포함하고,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 상기 컬러 컴포넌트의 계수들을 코딩하기 위해 사용된 라이스 파라미터들의 히스토리를

나타내는 히스토리 카운터를 유지하고,

상기 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 상기 히스토리 카운터에 대한 현재 값으로부터 상기 비디오 데이터의 상기 컬러 컴포넌트의 제 2 블록에 대한 라이스 파라미터 스케일링 값을 도출하며,

상기 제 2 블록의 제 2 계수에 대한 제 2 복수의 이웃 계수들의 절대 값들의 제 2 합을 계산하고,

상기 라이스 파라미터 스케일링 값을 사용하여 상기 절대 값들의 제 2 합에 대한 스케일링된 값을 계산하며,

상기 절대 값들의 제 2 합에 대한 상기 스케일링된 값을 사용하여 상기 제 2 블록의 상기 제 2 계수를 코딩하도록

추가로 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 36

제 35 항에 있어서,

상기 라이스 파라미터 스케일링 값을 도출하기 위해, 상기 하나 이상의 프로세서들은, $(StatCoeff[idx] \geq T)$? $P2 : P1$ 를 계산하도록 구성되고, idx 는 상기 컬러 컴포넌트를 나타내고, $StatCoeff[idx]$ 는 상기 히스토리 카운터에 대한 현재 값을 나타내며, T , $P1$, 및 $P2$ 는 입력 값들을 나타내는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 37

제 27 항에 있어서,

상기 현재 계수를 코딩하기 위해, 상기 하나 이상의 프로세서들은, 상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 인코딩하는 것 또는 상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 디코딩하는 것 중 적어도 하나를 수행하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 38

제 27 항에 있어서,

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

청구항 39

제 27 항에 있어서,

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

청구항 40

비디오 데이터를 코딩하기 위한 디바이스로서,

비디오 데이터의 현재 블록을 위한 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하기 위한 수단,

상기 비디오 데이터에 대한 비트 심도 또는 상기 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하기 위한 수단,

상기 베이스 레벨 오프셋 값을 사용하여 상기 절대 값들의 합에 대한 정규화된 값을 계산하기 위한 수단, 및

상기 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 코딩하기 위한 수단

을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

발명의 설명

기술 분야

- [0001] 본 출원은 2022년 4월 7일자로 출원된 미국 특허출원 제 17/658,396 호, 및 2021년 4월 9일자로 출원된 미국 가출원 제 63/173,269 호에 대한 우선권을 주장하며, 이 출원들 각각의 전체 내용은 본 명세서에 참조로 포함된다. 2022년 4월 7일자로 출원된 미국 출원 제 17/658,396 호는 2021년 4월 9일자로 출원된 미국 가출원 제 63/173,269 호의 이익을 주장한다.
- [0002] 본 개시는 비디오 인코딩 및 비디오 디코딩을 포함하는 비디오 코딩에 관한 것이다.

배경 기술

- [0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인용 디지털 보조기들(PDA들), 랩톱 또는 데스크톱 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 수도 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩(AVC), ITU-T H.265/고 효율 비디오 코딩(HEVC), TU-T H.266/VVC(Versatile Video Coding)에 의해 정의된 표준들, 및 그러한 표준들의 확장들에서 설명된 것들과 같은 비디오 코딩 기법들 뿐만 아니라 오픈 미디어 연합(Alliance for Open Media)에 의해 개발된 AV1(AOMedia Video 1)과 같은 독점적 비디오 코덱들/포맷들을 구현한다. 비디오 디바이스들은 그러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.
- [0004] 비디오 코딩 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소 또는 제거하기 위해 공간(인트라-화상) 예측 및/또는 시간(인터-화상) 예측을 포함한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스(예를 들어, 비디오 화상 또는 비디오 화상의 일부)는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 코딩 트리 유닛(CTU)들, 코딩 유닛(CU)들 및/또는 코딩 노드들로도 지칭될 수도 있다. 화상의 인트라-코딩된(I) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃하는 블록들에서의 레퍼런스 샘플들에 대한 공간 예측을 사용하여 인코딩된다. 화상의 인터-코딩된(P 또는 B) 슬라이스내의 비디오 블록들은 동일한 화상내의 이웃하는 블록들에 있는 레퍼런스 샘플들에 대한 공간 예측, 또는 다른 레퍼런스 화상들 내의 레퍼런스 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수도 있고, 레퍼런스 화상들은 레퍼런스 프레임들로 지칭될 수도 있다.

발명의 내용

해결하려는 과제

- [0005] 일반적으로, 본 개시는 비디오 데이터, 예를 들어, 높은 비트 심도 비디오 데이터를 코딩하기 위한 라이스 파라미터를 도출하기 위한 기법들을 설명한다. 특히, 라이스 파라미터는 인코딩 및/또는 디코딩을 포함할 수 있는 정규 잔차 코딩(RRC)을 수행할 때 사용될 수 있다. 이러한 기법들은 ITU-T H.266/VVC(Versatile Video Coding) 또는 다른 비디오 코딩 표준들에 적용될 수 있다. 본 개시의 기법들은 일반적으로 비디오 데이터의 변환 블록의 현재 계수를 이진화하기 위해 사용될 라이스 파라미터의 결정에 관한 것이고, 여기서 라이스 파라미터의 결정은 베이스 레벨 오프셋 값을 사용하는 것을 포함한다. 베이스 레벨 오프셋 값은 비디오 데이터에 대한 비트 심도 또는 비디오 데이터의 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 계산될 수도 있다. 그 다음, 베이스 레벨 오프셋 값은 정규화된 값을 계산하기 위해 현재 계수에 대한 이웃 계수들에 대한 절대 값들의 합에 적용될 수 있다. 정규화된 값은, 예를 들어 룩업 테이블로부터 라이스 파라미터를 결정하기 위해 사용될 수 있다. 라이스 파라미터는 그 후 현재 계수에 대한 값을 이진화하기 위해 사용될 수 있다.

과제의 해결 수단

- [0006] 일 예에서, 비디오 데이터를 코딩(인코딩 및/또는 디코딩)하는 방법은, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 비디오 데이터에 대한 비트 심도 또는

현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하는 단계; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계를 포함한다.

[0007] 다른 예에서, 비디오 데이터를 코딩 (인코딩 및/또는 디코딩) 하기 위한 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리; 및 회로부에서 구현되는 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하도록 구성된다.

[0008] 다른 예에서, 컴퓨터 판독가능 저장 매체로서 명령들이 저장되고, 상기 명령들은, 실행될 때, 비디오 데이터를 코딩하기 위한 디바이스의 프로세서로 하여금: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하게 하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하게 하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하게 하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하게 한다.

[0009] 다른 예에서, 비디오 데이터를 코딩 (인코딩 및/또는 디코딩) 하기 위한 디바이스는, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하기 위한 수단; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하기 위한 수단; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하기 위한 수단; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하기 위한 수단을 포함한다.

[0010] 하나 이상의 예들의 상세들이 첨부 도면들 및 하기의 설명에서 제시된다. 다른 특징들, 목적들, 및 이점들은 설명들, 도면들, 및 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0011] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 로컬 절대값들의 합 (locSumAbs) 을 계산하기 위해 사용될 수 있는 현재 계수에 대한 이웃 계수들의 예를 예시하는 개념도이다.

도 3 은 비디오 데이터의 16-비트 테스트 시퀀스에 대한 코딩된 변환 계수들의 경험적 히스토그램을 예시하는 그래프이다.

도 4 는 계수 값들에 대한 엔트로피 코딩 동안 생성된 코드워드 길이들 및 주어진 라이스 파라미터들 사이의 관계를 예시하는 그래프이다.

도 5 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 6 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 7 은 본 개시의 기법들에 따라 현재 블록을 인코딩하기 위한 예시적인 방법을 도시하는 플로우차트이다.

도 8 은 본 개시의 기법들에 따라 현재 블록을 디코딩하기 위한 예시적인 방법을 도시하는 플로우차트이다.

도 9 는 본 개시의 기법들에 따른 비디오 데이터를 인코딩하는 예시의 방법을 나타내는 플로우차트이다.

도 10 은 본 개시의 기법들에 따른 비디오 데이터를 디코딩하는 예시의 방법을 나타내는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0012] 라이스 파라미터 값들은 비디오 데이터의 잔차 블록들의 계수들을 코딩 (인코딩 또는 디코딩) 하는데 사용될 수도 있다. 특히, 라이스 파라미터는 값을 엔트로피 코딩 (인코딩 또는 디코딩) 하는 것과 연관된 값을 이진화할 때 사용될 수 있다. ITU-T H.266/VVC (Versatile Video Coding) 표준의 개발에서는 0 내지 3 (한계 포함) 범위의 라이스 파라미터 값들의 사용이 광범위하게 테스트되었다. 전형적으로, 라이스 파라미터 값들

은 또한 대응하는 비디오 데이터의 비트 심도를 입력으로서 사용하는 클리핑 식을 사용하여 스케일링된다. 원래 VVC 에 대해 개발된 클리핑 식은 8-비트 또는 10-비트의 비트 심도를 전체로 하였다. 그러나, 비트 심도가 10-비트들을 넘어 증가되면, 클리핑 함수는 더 이상 라이스 파라미터 값들을 0 내지 3 비트들의 범위 내에 있도록 효과적으로 클리핑하지 않을 수도 있으며, 이는 코드워드 사이즈들의 급격한 증가, 및 따라서 비트스트림에서의 시그널링된 값들을 초래할 수도 있다. 본 개시는 높은 비트 심도들 (예를 들어, 10 비트를 넘는 비트 심도들) 을 이용하여 비디오 데이터를 코딩하기 위해 라이스 파라미터들을 이용할 때 비트스트림에서 시그널링된 데이터 및 코드워드 사이즈들에서의 이러한 증가들을 회피할 수도 있는 기법들을 설명한다.

[0013] 비디오 코더 (인코더 또는 디코더) 는 변환 블록의 현재 계수에 대한 값을 코딩 (인코딩 또는 디코딩) 하기 위해 본 개시의 기법들을 사용하도록 구성될 수도 있다. 특히, 비디오 코더는 처음에 현재 계수에 대한 이웃 계수들의 절대 값들의 합을 계산할 수도 있다. 그 후, 비디오 코더는 비디오 데이터에 대한 비트 심도 및/또는 변환 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 어느 하나 또는 둘 다에 따라 절대 값들의 합에 적용할 베이스 레벨 오프셋 값을 계산할 수도 있다. 비디오 코더는 그 후 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산할 수도 있다. 비디오 코더는 그 후 정규화된 값을 사용하여 현재 계수를 코딩할 수도 있다. 위에서 언급된 바와 같이, 본 개시의 기법들은 비트 심도가 10 비트들을 넘어 확장될 수 있는 상황들에서 적용될 수도 있다. 따라서, 비트 심도에 따라 베이스 레벨 오프셋 값을 계산하면, 라이스 파라미터가 0 내지 3 비트 이내로 남아있도록 클리핑될 수 있다.

[0014] 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 및 ITU-T H.264 (ISO/IEC MPEG-4 AVC 로도 또한 알려짐) (그의 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장들을 포함함) 를 포함한다. HEVC (High-Efficiency Video Coding) 은 2013년 4월에 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 및 ISO/IEC 모션 화상 전문가 그룹 (MPEG) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 완결되었다.

[0015] JVET (Joint Video Experts Team) 는 MPEG 과 ITU-T Study Group 16 의 VCEG 에 의해 형성된 협력 팀으로서, VVC (Versatile Video Coding) 표준에 대해 그리고 이를 넘어서 비디오 코딩 기법들을 계속 개발하고 있다. VVC 의 주요 목적은 기존의 HEVC 표준에 비해 압축 성능에서 현저한 개선을 제공하여, 360° 전방향 몰입형 멀티미디어 및 HDR (high-dynamic-range) 비디오와 같은 신생의 애플리케이션들 및 더 고품질의 비디오 서비스들의 전개를 보조하는 것이다. VVC 표준의 초안은 Benjamin Bross, Jianle Chen, Shan Liu 및 Ye-Kui Wang 의, “Versatile Video Coding (Draft 10)” Joint Video Experts Team (JVET) (ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JVET- S2001, 19th Meeting) 에서 발견할 수 있다: Teleconference, 22 June ~ 01 July 2020, phenix.it-sudparis.eu/jvet/doc_end_user/documents/19_Teleconference/wg11/JVET-S2001-v17.zip 에서 이용가능하다.

[0016] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (100) 을 예시한 블록도이다. 본 개시의 기법은 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 (예컨대, 재구성된) 비디오, 및 시그널링 데이터와 같은 비디오 메타데이터를 포함할 수도 있다.

[0017] 도 1 에 도시된 바와 같이, 시스템 (100) 은, 이 예에서 목적지 디바이스 (116) 에 의해 디코딩 및 디스플레이 될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (102) 를 포함한다. 특히, 소스 디바이스 (102) 는 비디오 데이터를 컴퓨터 판독가능 매체 (110) 를 통해 목적지 디바이스 (116) 에 제공한다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 모바일 디바이스들, 태블릿 컴퓨터들, 셋탑 박스들, 전화기 핸드셋들, 이를 태면 스마트폰들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스, 브로드캐스트 수신기 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 디바이스들로서 지칭될 수도 있다.

[0018] 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 따라, 소스 디바이스 (102) 의 비디오 인코더 (200) 및 목적지 디바이스 (116) 의 비디오 디코더 (300) 는 라이스 파라미터들을 사용하여 높은 비트

심도 비디오 데이터를 코딩하기 위한 기법들을 적용하도록 구성될 수도 있다. 따라서, 소스 디바이스 (102) 는 비디오 인코딩 디바이스의 예를 나타내는 한편, 목적지 디바이스 (116) 는 비디오 디코딩 디바이스의 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 포함하기 보다는, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0019] 도 1 에 도시된 시스템 (100) 은 단지 일 예일 뿐이다. 일반적으로, 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스는 라이스 파라미터들을 사용하여 높은 비트 심도 비디오 데이터를 코딩하기 위한 기법들을 수행할 수도 있다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는, 소스 디바이스 (102) 가 목적지 디바이스 (116) 로의 송신을 위해 코딩된 비디오 데이터를 생성하는 그러한 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 예들에서, 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 들어, 비디오 스트리밍, 비디오 플래이백, 비디오 브로드캐스팅, 또는 비디오 텔레포니를 위해, 소스 디바이스 (102) 와 목적지 디바이스 (116) 사이의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0020] 일반적으로, 비디오 소스 (104) 는 비디오 데이터 (즉, 원시, 코딩되지 않은 비디오 데이터) 의 소스를 나타내며 화상들에 대한 데이터를 인코딩하는 비디오 인코더 (200) 에 비디오 데이터의 순차적인 일련의 화상들 (또한 "프레임들" 로서도 지칭됨) 을 제공한다. 소스 디바이스 (102) 의 비디오 소스 (104) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 원시 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스를 포함할 수도 있다. 추가적인 대안으로서, 비디오 소스 (104) 는 컴퓨터 그래픽스 기반 데이터를 소스 비디오로서, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합으로서 생성할 수도 있다. 각각의 경우에, 비디오 인코더 (200) 는 캡처되거나, 미리-캡처되거나, 또는 컴퓨터 생성된 비디오 데이터를 인코딩한다. 비디오 인코더 (200) 는 화상들을 수신된 순서 (때때로 "디스플레이 순서" 로서 지칭됨) 로부터 코딩을 위한 코딩 순서로 재배열할 수도 있다. 비디오 인코더 (200) 는 인코딩된 비디오 데이터를 포함하는 비트스트림을 생성할 수도 있다. 그 후, 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.

[0021] 소스 디바이스 (102) 의 메모리 (106) 및 목적지 디바이스 (116) 의 메모리 (120) 는 범용 메모리들을 나타낸다. 일부 예들에서, 메모리들 (106, 120) 은 원시 비디오 데이터, 예컨대, 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, 예컨대, 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해 각각 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 있음이 이해되어야 한다. 더욱이, 메모리들 (106, 120) 은, 예컨대, 비디오 인코더 (200) 로부터 출력되고 비디오 디코더 (300) 에 입력되는 인코딩된 비디오 데이터를 저장할 수도 있다. 일부 예들에서, 메모리들 (106, 120) 의 부분들은 예컨대, 원시의, 디코딩된, 및/또는 인코딩된 비디오 데이터를 저장하기 위해 하나 이상의 비디오 버퍼들로서 할당될 수도 있다.

[0022] 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. 일 예에서, 컴퓨터 판독가능 매체 (110) 는, 소스 디바이스 (102) 로 하여금, 실시간으로, 예를 들어, 라디오 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 임의의 무선 또는 유선 통신 매체, 이를 테면 라디오 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광

역 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 가능하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0023] 일부 예들에서, 소스 디바이스 (102)는 출력 인터페이스 (108)로부터 저장 디바이스 (112)로 인코딩된 데이터를 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 매체들과 같은 다양한 분산된 또는 로컬로 액세스 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다.

[0024] 일부 예들에 있어서, 소스 디바이스 (102)는, 소스 디바이스 (102)에 의해 생성된 인코딩된 비디오 데이터를 저장할 수도 있는 파일 서버 (114) 또는 다른 중간 저장 디바이스로 인코딩된 비디오 데이터를 출력할 수도 있다. 목적지 디바이스 (116)는 스트리밍 또는 다운로드를 통해 파일 서버 (114)로부터의 저장된 비디오 데이터에 액세스할 수도 있다.

[0025] 파일 서버 (114)는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (116)에 송신할 수도 있는 임의의 타입의 서버 디바이스일 수도 있다. 파일 서버 (114)는 (예컨대, 웹사이트에 대한) 웹 서버, (파일 전송 프로토콜 (FTP) 또는 FLUTE (File Delivery over Unidirectional Transport) 프로토콜과 같은) 파일 전송 프로토콜 서비스를 제공하도록 구성된 서버, 콘텐츠 전달 네트워크 (CDN) 디바이스, 하이퍼텍스트 전송 프로토콜 (HTTP) 서버, 멀티미디어 브로드캐스트 멀티캐스트 서비스 (MBMS) 또는 강화된 MBMS (eMBMS) 서버, 및/또는 네트워크 어태치형 스토리지 (NAS) 디바이스를 나타낼 수도 있다. 파일 서버 (114)는 부가적으로 또는 대안으로, HTTP를 통한 동적 적응 스트리밍 (DASH), HTTP 라이브 스트리밍 (HLS), 실시간 스트리밍 프로토콜 (RTSP), HTTP 동적 스트리밍 등과 같은 하나 이상의 HTTP 스트리밍 프로토콜을 구현할 수도 있다.

[0026] 목적지 디바이스 (116)는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 파일 서버 (114)로부터의 인코딩된 비디오 데이터에 액세스할 수도 있다. 이는 파일 서버 (114)상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한, 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, 디지털 가입자 라인 (digital subscriber line; DSL), 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 입력 인터페이스 (122)는 파일 서버 (114)로부터 미디어 데이터를 추출하거나 수신하기 위해 상기 논의된 다양한 프로토콜들 중 임의의 하나 이상의 프로토콜들, 또는 미디어 데이터를 추출하기 위한 다른 그러한 프로토콜들에 따라 동작하도록 구성될 수도 있다.

[0027] 출력 인터페이스 (108) 및 입력 인터페이스 (122)는 무선 송신기들/수신기들, 모뎀들, 유선 네트워킹 컴포넌트들 (예컨대, 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 예들에서, 출력 인터페이스 (108) 및 입력 인터페이스 (122)는 4G, 4G-LTE (롱 텀 에볼루션), 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 예들에서, 출력 인터페이스 (108) 및 입력 인터페이스 (122)는 IEEE 802.11 사양, IEEE 802.15 사양 (예컨대, 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 개별의 시스템-온-칩 (system-on-a-chip; SoC) 디바이스들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102)는 비디오 인코더 (200) 및/또는 출력 인터페이스 (108)에 기인한 기능을 수행하기 위한 SoC 디바이스를 포함할 수도 있고, 목적지 디바이스 (116)는 비디오 디코더 (300) 및/또는 입력 인터페이스 (122)에 기인한 기능을 수행하기 위한 SoC 디바이스를 포함할 수도 있다.

[0028] 본 개시의 기법들은 오버-더-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 예컨대 DASH (dynamic adaptive streaming over HTTP), 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 어플리케이션들과 같은 다양한 멀티미디어 어플리케이션들 중 임의의 것을 지원하여 비디오 코딩에 적용될 수도 있다.

[0029] 목적지 디바이스 (116)의 입력 인터페이스 (122)는 컴퓨터 판독가능 매체 (110) (예컨대, 통신 매체, 저장 디

바이스 (112), 파일 서버 (114) 등)로부터 인코딩된 비디오 비트스트림을 수신한다. 인코딩된 비디오 비트스트림은, 비디오 블록들 또는 다른 코딩된 유닛들 (예컨대, 슬라이스들, 화상들, 화상들의 그룹들, 시퀀스들 등)의 프로세싱 및/또는 특성들을 기술하는 값들을 갖는 선택스 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 화상들을 사용자에게 디스플레이한다. 디스플레이 디바이스 (118)는 액정 디스플레이 (LCD), 플라스마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 나타낼 수도 있다.

[0030] 도 1에 도시되지는 않았지만, 일부 예들에 있어서, 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.

[0031] 비디오 인코더 (200) 및 비디오 디코더 (300)는 각각 다양한 적합한 인코더 및/또는 디코더 회로부, 이를 태면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적 회로들 (ASIC들), 펠드 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 있으며, 이들 중 어느 하나는 개별의 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.

[0032] 비디오 인코더 (200) 및 비디오 디코더 (300)는 고 효율 비디오 코딩 (HEVC)으로서도 또한 지칭되는 ITU-T H.265와 같은 비디오 코딩 표준 또는 그에 대한 확장들, 예컨대 멀티-뷰 및/또는 스케일러블 비디오 코딩 확장들에 따라 동작할 수도 있다. 대안적으로, 비디오 인코더 (200) 및 비디오 디코더 (300)는, 다용도 비디오 코딩 (VVC)로서 또한 지칭되는 ITU-T H.266과 같은 다른 독점 또는 산업 표준들에 따라 동작할 수도 있다. 다른 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300)는 AOMedia Video 1 (AV1), AV1의 확장들, 및/또는 AV1의 후속 버전들 (예를 들어, AV2)과 같은 독점 비디오 코덱/포맷에 따라 동작할 수도 있다. 다른 예들에 있어서, 비디오 인코더 (200) 및 비디오 디코더 (300)는 다른 독점적 포맷 또는 산업 표준에 따라 동작할 수도 있다. 하지만, 본 개시의 기법들은 임의의 특정 코딩 표준 또는 포맷에 한정되지 않는다. 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300)는 라이스 파라미터들을 사용하여 높은 비트 심도 비디오 데이터를 코딩하는 임의의 비디오 코딩 기법들과 함께 본 개시의 기법들을 수행하도록 구성될 수 있다.

[0033] 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300)는 화상들의 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 2차원 매트릭스를 포함할 수도 있다. 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300)는 YUV (예컨대, Y, Cb, Cr) 포맷으로 표현된 비디오 데이터를 코딩할 수도 있다. 즉, 화상의 샘플들에 대한 적색, 녹색, 및 청색 (RGB) 데이터를 코딩하기 보다는, 비디오 인코더 (200) 및 비디오 디코더 (300)는 루미넌스 및 크로미넌스 컴포넌트들을 코딩할 수도 있으며, 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 포맷팅된 데이터를 YUV 표현으로 변환하고, 비디오 디코더 (300)는 YUV 표현을 RGB 포맷으로 변환한다. 대안적으로, 프리- 및 포스트-프로세싱 유닛들 (도시되지 않음)이 이들 변환들을 수행할 수도 있다.

[0034] 본 개시는 일반적으로 화상의 데이터를 인코딩하거나 또는 디코딩하는 프로세스를 포함하도록 화상들의 코딩 (예컨대, 인코딩 및 디코딩)을 참조할 수도 있다. 유사하게, 본 개시는, 블록들에 대한 데이터를 인코딩하거나 또는 디코딩하는 프로세스, 예컨대, 예측 및/또는 잔차 코딩을 포함하도록 화상의 블록들의 코딩을 참조할 수도 있다. 인코딩된 비디오 비트스트림은 일반적으로 코딩 결정들 (예컨대, 코딩 모드들) 및 화상들의 블록들로의 파티셔닝을 나타내는 선택스 엘리먼트들에 대한 일련의 값들을 포함한다. 따라서, 화상 또는 블록을 코딩하는 것에 대한 참조들은 일반적으로 화상 또는 블록을 형성하는 선택스 엘리먼트들에 대한 코딩 값들로

서 이해되어야 한다.

- [0035] HEVC 는 코딩 유닛 (CU) 들, 예측 유닛 (PU) 들, 및 변환 유닛 (TU) 들을 포함하는 다양한 블록들을 규정한다. HEVC 에 따르면, (비디오 인코더 (200) 와 같은) 비디오 코더는 쿼드트리 구조에 따라 코딩 트리 유닛 (CTU) 을 CU 들로 파티셔닝한다. 즉, 비디오 코더는 CTU 들 및 CU 들을 4 개의 동일한 비오버랩하는 정사각형들로 파티셔닝하고, 쿼드트리의 각각의 노드는 0 개 또는 4 개의 자식 노드들 중 어느 하나를 갖는다. 자식 노드들이 없는 노드들은 "리프 노드들" 로서 지칭될 수도 있고, 그러한 리프 노드들의 CU 들은 하나 이상의 PU 들 및/또는 하나 이상의 TU 들을 포함할 수도 있다. 비디오 코더는 PU 들 및 TU 들을 추가로 파티셔닝할 수도 있다. 예를 들어, HEVC 에서, 잔차 쿼드트리 (RQT) 는 TU 들의 파티셔닝을 나타낸다. HEVC 에서, PU 들은 인터-예측 데이터를 나타내는 한편, TU 들은 잔차 데이터를 나타낸다. 인트라-예측되는 CU 들은 인트라 모드 표시와 같은 인트라 예측 정보를 포함한다.
- [0036] 다른 예로서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 VVC 에 따라 동작하도록 구성될 수도 있다. VVC 에 따라, 비디오 코더 (예컨대 비디오 인코더 (200)) 는 화상을 복수의 코딩 트리 유닛 (CTU) 들로 파티셔닝한다. 비디오 인코더 (200) 는 쿼드트리 이진 트리 (QTBT) 구조 또는 멀티-타입 트리 (MTT) 구조와 같은 트리 구조에 따라 CTU 를 파티셔닝할 수도 있다. QTBT 구조는 HEVC 의 CU 들, PU 들, 및 TU 들 사이의 분리와 같은 다중 파티션 타입들의 개념들을 제거한다. QTBT 구조는 2 개의 레벨들: 쿼드트리 파티셔닝에 따라 파티셔닝된 제 1 레벨, 및 이진 트리 파티셔닝에 따라 파티셔닝된 제 2 레벨을 포함한다. QTBT 구조의 루트 노드는 CTU 에 대응한다. 이진 트리들의 리프 노드들은 코딩 유닛들 (CU 들) 에 대응한다.
- [0037] MTT 파티셔닝 구조에서, 블록들은 쿼드트리 (QT) 파티션, 이진 트리 (BT) 파티션, 및 하나 이상의 타입들의 트리플 트리 (TT) (삼진 트리 (TT) 로도 칭함) 파티션들을 사용하여 파티셔닝될 수도 있다. 트리플 또는 삼진 트리 파티션은 블록이 3 개의 서브-블록으로 스플릿되는 파티션이다. 일부 예들에서, 트리플 또는 삼진 트리 파티션은 중심을 통해 원래 블록을 나누지 않으면서 블록을 3 개의 서브-블록들로 나눈다. MTT 에서의 파티셔닝 타입들 (예를 들어, QT, BT 및 TT) 은 대칭적이거나 비대칭적일 수도 있다.
- [0038] AV1 코덱에 따라 동작할 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 비디오 데이터를 블록들로 코딩하도록 구성될 수도 있다. AV1 에서, 프로세싱될 수 있는 최대 코딩 블록은 슈퍼블록으로 지칭된다. AV1 에서, 슈퍼블록은 128x128 루마 샘플들 또는 64x64 루마 샘플들 중 어느 하나일 수 있다. 하지만, 후속 비디오 코딩 포맷들 (예를 들어, AV2) 에서, 슈퍼블록은 상이한 (예를 들어, 더 큰) 루마 샘플 사이즈들에 의해 정의될 수도 있다. 일부 예들에서, 슈퍼블록은 블록 쿼드트리의 최상위 레벨이다. 비디오 인코더 (200) 는 추가로, 슈퍼블록을 더 작은 코딩 블록들로 파티셔닝할 수도 있다. 비디오 인코더 (200) 는 정사각형 또는 비-정사각형 파티셔닝을 사용하여 슈퍼블록 및 다른 코딩 블록들을 더 작은 블록들로 파티셔닝할 수도 있다. 비-정사각형 블록들은 $N/2 \times N$, $N \times N/2$, $N/4 \times N$, 및 $N \times N/4$ 블록들을 포함할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 코딩 블록들의 각각에 대해 별도의 예측 및 변환 프로세스들을 수행할 수도 있다.
- [0039] AV1 은 또한 비디오 데이터의 타일을 정의한다. 타일은, 다른 타일들과 독립적으로 코딩될 수도 있는 슈퍼블록들의 직사각형 어레이이다. 즉, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다른 타일들로부터의 비디오 데이터를 사용하지 않고도 타일 내의 코딩 블록들을 각각 인코딩 및 디코딩할 수도 있다. 하지만, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 타일 경계들에 걸쳐 필터링을 수행할 수도 있다. 타일들은 사이즈가 균일하거나 균일하지 않을 수도 있다. 타일 기반 코딩은 인코더 및 디코더 구현들을 위한 병렬 프로세싱 및/또는 멀티스레딩 (multi-threading) 을 인에이블할 수도 있다.
- [0040] 일부 예들에 있어서, 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 예들에 있어서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 컴포넌트에 대한 하나의 QTBT/MTT 구조 및 양자 모두의 크로미넌스 컴포넌트들에 대한 다른 QTBT/MTT 구조 (또는 개별의 크로미넌스 컴포넌트들에 대한 2 개의 QTBT/MTT 구조들) 와 같은 2 이상의 QTBT 또는 MTT 구조들을 사용할 수도 있다.
- [0041] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 쿼드트리 파티셔닝, QTBT 파티셔닝, MTT 파티셔닝, 슈퍼블록 파티셔닝, 또는 다른 파티셔닝 구조들을 사용하도록 구성될 수도 있다.
- [0042] 일부 예들에 있어서, CTU 는 루마 샘플들의 코딩 트리 블록 (CTB), 3 개의 샘플 어레이들을 갖는 화상의 크로마 샘플들의 2 개의 대응하는 CTB 들, 또는 샘플들을 코딩하는데 사용된 3 개의 별도의 컬러 평면들 및 선택스 구조

들을 사용하여 코딩되는 화상 또는 모노크롬 화상의 샘플들의 CTB 를 포함한다. CTB 는 컴포넌트의 CTB들로의 분할이 파티셔닝이 되도록 N 의 일부 값에 대한 샘플들의 NxN 블록일 수도 있다. 컴포넌트는 모노크롬 포맷의 화상을 위한 어레이 또는 어레이의 단일 샘플 또는 4:2:0, 4:2:2 또는 4:4:4 컬러 포맷의 화상을 위한 2 개의 어레이 (루마 및 2 개의 크로마) 중 하나로부터의 어레이 또는 단일 샘플일 수도 있다. 일부 예들에서, 코딩 블록은, CTB 의 코딩 블록들로의 분할이 파티셔닝이 되도록 M 및 N 의 일부 값들에 대한 샘플들의 MxN 블록이다.

[0043] 블록들 (예를 들어, CTU들 또는 CU들) 은 화상에서 다양한 방식으로 그룹화될 수도 있다. 일 예로서, 브릭은 화상에서의 특정 타일 내의 CTU 행들의 직사각형 영역을 지칭할 수도 있다. 타일은 화상에서의 특정 타일 컬럼 및 특정 타일 로우 내에서 CTB들의 직사각형 영역일 수도 있다. 타일 열은, 화상의 높이와 동일한 높이 및 (예를 들어, 화상 파라미터 세트에서와 같이) 선택스 엘리먼트들에 의해 특정된 폭을 갖는 CTU들의 직사각형 영역을 지칭한다. 타일 행은, (예를 들어, 화상 파라미터 세트에서와 같이) 선택스 엘리먼트들에 의해 특정된 높이 및 화상의 폭과 동일한 폭을 갖는 CTU들의 직사각형 영역을 지칭한다.

[0044] 일부 예들에서, 타일은 다중의 브릭들로 파티셔닝될 수도 있으며, 그 각각은 타일 내의 하나 이상의 CTU 행들을 포함할 수도 있다. 다중의 브릭들로 파티셔닝되지 않은 타일이 또한, 브릭으로서 지칭될 수도 있다. 그러나, 타일의 진정한 서브세트인 브릭은 타일로서 지칭되지 않을 수도 있다. 화상에서의 브릭들은 또한 슬라이스로 배열될 수도 있다. 슬라이스는 단일의 네트워크 추상화 계층 (NAL) 유닛에 배타적으로 포함될 수도 있는 화상의 정수 개의 브릭들일 수도 있다. 일부 예들에서, 슬라이스는 다수의 완전한 타일들 또는 하나의 타일의 완전한 브릭들의 연속적인 시퀀스만을 포함한다.

[0045] 본 개시는 수직 및 수평 치수들의 관점에서 (CU 또는 다른 비디오 블록과 같은) 블록의 샘플 치수들을 지칭하기 위해 상호교환가능하게 "NxN" 및 "N 바이 N", 예컨대, 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 가지며, 여기서, N 은 음이 아닌 정수 값을 나타낸다. CU 에서의 샘플들은 행들 및 열들로 배열될 수도 있다. 더욱이, CU들은 수직 방향에서와 동일한 수의 샘플들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, CU들은 NxM 샘플들을 포함할 수도 있고, 여기서 M 은 N 과 반드시 동일한 것은 아니다.

[0046] 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 차이들을 나타낸다.

[0047] 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 화상의 데이터로부터 CU 를 예측하는 것을 지칭하는 반면, 인트라-예측은 일반적으로 동일한 화상의 이전에 코딩된 데이터로부터 CU 를 예측하는 것을 지칭한다. 인터-예측을 수행하기 위해, 비디오 인코더 (200) 는 하나 이상의 모션 벡터를 사용하여 예측 블록을 생성할 수도 있다. 비디오 인코더 (200) 는 일반적으로, 예를 들어 CU 와 레퍼런스 블록 사이의 차이에 관하여, CU 와 밀접하게 매칭하는 레퍼런스 블록을 식별하기 위해 모션 검색을 수행할 수도 있다. 비디오 인코더 (200) 는 레퍼런스 블록이 현재 CU 와 밀접하게 매칭하는지 여부를 결정하기 위해 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of squared differences; SSD), 평균 절대차 (mean absolute difference; MAD), 평균 제곱차 (mean squared differences; MSD) 또는 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.

[0048] VVC 의 일부 예들은 또한, 인터-예측 모드로 고려될 수도 있는 아핀 모션 보상 모드를 제공한다. 아핀 모션 보상 모드에서, 비디오 인코더 (200) 는 줌 인 또는 아웃, 회전, 원근 모션, 또는 다른 불규칙한 모션 타입들과 같은 비-병진 모션을 나타내는 2 개 이상의 모션 벡터들을 결정할 수도 있다.

[0049] 인트라-예측을 수행하기 위해, 비디오 인코더 (200) 는 예측 블록을 생성하기 위해 인트라-예측 모드를 선택할 수도 있다. VVC 의 일부 예들은 평면 모드 및 DC 모드 뿐만 아니라, 다양한 방향성 모드들을 포함한, 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 (좌측에서 우측으로, 상단에서 하단으로) CTU들 및 CU들을 코딩하는 것을 가정하여, 현재 블록과 동일한 화상에서 현재 블록의 상측, 상측 및 좌측에, 또

는 좌측에 있을 수도 있다.

[0050] 비디오 인코더 (200) 는 현재 블록을 위한 예측 모드를 나타내는 데이터를 인코딩한다. 예를 들어, 인터-예측 모드의 경우, 비디오 인코더 (200) 는 다양한 이용가능한 인터-예측 모드들 중 어느 것이 사용되는지를 나타내는 데이터 뿐만 아니라, 대응하는 모드를 위한 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.

[0051] AV1 은 비디오 데이터의 코딩 블록을 인코딩 및 디코딩하기 위한 2 개의 일반적인 기법들을 포함한다. 2 개의 일반적인 기법들은 인트라 예측 (예를 들어, 인트라 프레임 예측 또는 공간적 예측) 및 인터 예측 (예를 들어, 인터 프레임 예측 또는 시간적 예측) 이다. AV1 의 컨텍스트에서, 인트라 예측 모드를 사용하여 비디오 데이터의 현재 프레임의 블록들을 예측할 경우, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 비디오 데이터의 다른 프레임들로부터의 비디오 데이터를 사용하지 않는다. 대부분의 인트라 예측 모드들에 대해, 비디오 인코더 (200) 는 현재 블록에서의 샘플 값들과 동일한 프레임에서의 레퍼런스 샘플들로부터 생성된 예측된 값들 사이의 차이에 기초하여 현재 프레임의 블록들을 인코딩한다. 비디오 인코더 (200) 는 인트라 예측 모드에 기초하여 레퍼런스 샘플들로부터 생성된 예측된 값들을 결정한다.

[0052] 블록의 인트라-예측 또는 인터-예측과 같은 예측에 후속하여, 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 변환 (DCT), 정수 변환, 웨이브릿 변환, 또는 개념적으로 유사한 변환을 잔차 비디오 데이터에 적용할 수도 있다. 추가적으로, 비디오 인코더 (200) 는 모드 의존적 비-분리가 능 2 차 변환 (MDNSTT), 신호 의존적 변환, Karhunen-Loeve 변환 (KLT) 등과 같은 제 1 변환에 후속하여 2 차 변환을 적용할 수도 있다. 비디오 인코더 (200) 는 하나 이상의 변환들의 적용에 후속하여 변환 계수들을 생성한다.

[0053] 상기 언급된 바와 같이, 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (200) 는 변환 계수들의 양자화를 수행할 수도 있다. 일반적으로 양자화는 변환 계수들이 그 변환 계수들을 나타내는데 사용되는 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.

[0054] 양자화에 이어, 비디오 인코더 (200) 는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 포함한 2 차원 매트릭스로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 더 높은 에너지 (및 따라서 더 낮은 주파수) 변환 계수들을 벡터의 전방에 배치하고 그리고 더 낮은 에너지 (및 따라서 더 높은 주파수) 변환 계수들을 벡터의 후방에 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 양자화된 변환 계수들을 스캔하기 위해 미리정의된 스캔 순서를 활용하여 직렬화된 벡터를 생성하고, 그 다음, 벡터의 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (200) 는 적응 스캔을 수행할 수도 있다. 1 차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캔한 후, 비디오 인코더 (200) 는, 예컨대, 컨텍스트 적응적 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 선택스 엘리먼트들에 대한 값들을 엔트로피 인코딩할 수도 있다.

[0055] 변환 블록의 변환 계수들을 인코딩할 때, 비디오 인코더 (200) 는 예를 들어, (계수가 0 보다 큰 절대 값을 갖는지 여부를 표시하는) 중요한 변환 계수 플래그, (계수가 1 보다 큰 절대 값을 갖는지 여부를 표시하는) 1 보다 큰 플래그, (계수가 2 보다 큰 절대 값을 갖는지 여부를 표시하는) 2 보다 큰 플래그, (계수의 절대 값이 2 를 초과하는 양을 나타내는) 나머지 값, 및 (계수에 대한 값이 포지티브인지 네가티브인지 여부를 나타내는) 부호 플래그를 포함하는, 선택스 엘리먼트들에 대한 값들을 인코딩할 수도 있다.

[0056] 현재 계수의 나머지 값을 인코딩할 때, 비디오 인코더 (200) 는 라이스 파라미터를 사용하여 나머지 값을 이진화할 수도 있다. 본 개시의 기법들에 따르면, 라이스 파라미터를 결정하기 위해, 비디오 인코더 (200) 는

현재 계수에 대한 이웃하는 계수들의 절대 값들의 합을 계산할 수도 있다. 예를 들어, 비디오 인코더 (200) 는, 아래의 도 2 에 도시된 바와 같이, 현재 계수의 우측, 2 개 우측, 아래, 2 개 아래 및 아래와 우측에 대한 계수들에 대한 절대값들을 계산할 수도 있다. 비디오 인코더 (200) 는 그 후 절대 값들의 합을 형성하기 위해 계산된 절대 값들의 각각을 가산할 수도 있다. 비디오 인코더 (200) 는 그 후 현재 블록을 포함하는 슬라이스에 대한 비트 심도 및/또는 슬라이스 타입 중 적어도 하나를 결정할 수도 있다.

[0057] 비디오 인코더 (200) 는 그 후 예를 들어, 아래에서 더 상세히 논의되는 바와 같이 식 (1) 을 이용하여, 비트 심도 및/또는 슬라이스 타입에 따라 베이스 레벨 오프셋 값을 계산할 수도 있다. 비디오 인코더 (200) 는 그 후 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산할 수도 있다. 예를 들어, 비디오 인코더 (200) 는 아래에서 더 상세히 논의되는 바와 같이 식 (2) 를 사용하여 정규화된 값을 계산할 수도 있다. 비디오 인코더 (200) 는 그 후 예를 들어, 아래에서 더 상세히 논의되는 표 1 에 따라, 절대 값들의 로컬 합에 대한 정규화된 값을 사용하여 라이스 파라미터를 도출할 수도 있다. 비디오 인코더 (200) 는 그 후 라이스 파라미터를 사용하여 절대 값들의 합에 대한 정규화된 값에 대한 Golomb-Rice 코드를 형성할 수도 있다.

[0058] CABAC 을 수행하기 위해, 비디오 인코더 (200) 는 컨텍스트 모델 내의 컨텍스트를, 송신될 심볼에 할당할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 제로 값인지 여부와 관련될 수도 있다. 확률 결정은 심볼에 배정된 컨텍스트에 기초할 수도 있다. 상기 예에서 논의된 바와 같은 변환 계수의 경우, 심볼은 Golomb-Rice 코드의 비트들 (빈들) 중 하나에 대응할 수도 있다.

[0059] 비디오 인코더 (200) 는 선택스 데이터, 이를 테면 블록 기반 선택스 데이터, 화상 기반 선택스 데이터, 및 시퀀스 기반 선택스 데이터를, 비디오 디코더 (300) 에, 예컨대, 화상 헤더, 블록 헤더, 슬라이스 헤더, 또는 다른 선택스 데이터, 이를 테면 시퀀스 파라미터 세트 (SPS), 화상 파라미터 세트 (PPS), 또는 비디오 파라미터 세트 (VPS) 에서 추가로 생성할 수도 있다. 마찬가지로, 비디오 디코더 (300) 는 대응하는 비디오 데이터를 디코딩하는 방법을 결정하기 위해 그러한 선택스 데이터를 디코딩할 수도 있다.

[0060] 이러한 방식으로, 비디오 인코더 (200) 는 인코딩된 비디오 데이터, 예컨대, 화상의 블록들 (예컨대, CU들) 로의 파티셔닝을 기술하는 선택스 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.

[0061] 일반적으로, 비디오 디코더 (300) 는 비트스트림의 인코딩된 비디오 데이터를 디코딩하기 위해 비디오 인코더 (200) 에 의해 수행되는 것과 가역적인 프로세스를 수행한다. 예를 들어, 비디오 디코더 (300) 는 비디오 인코더 (200) 의 CABAC 인코딩 프로세스와 실질적으로 유사하지만 상반되는 방식으로 CABAC 을 사용하여 비트스트림의 선택스 엘리먼트들에 대한 값들을 디코딩할 수도 있다. 선택스 엘리먼트들은 화상의 CTU들로의 파티셔닝, 및 QTBT 구조와 같은 대응하는 파티션 구조에 따른 각각의 CTU 의 파티셔닝을 위한 파티셔닝 정보를 정의하여, CTU 의 CU들을 정의할 수도 있다. 선택스 엘리먼트들은 비디오 데이터의 블록들 (예컨대, CU들) 에 대한 예측 및 잔차 정보를 추가로 정의할 수도 있다.

[0062] 잔차 정보는, 예를 들어, 양자화된 변환 계수들에 의해 표현될 수도 있다. 비디오 디코더 (300) 는 블록에 대한 잔차 블록을 재현하기 위해 블록의 양자화된 변환 계수들을 역 양자화 및 역 변환할 수도 있다. 비디오 디코더 (300) 는 시그널링된 예측 모드 (인트라-예측 또는 인터-예측) 및 관련된 예측 정보 (예컨대, 인터-예측을 위한 모션 정보) 를 사용하여 블록에 대한 예측 블록을 형성한다. 그 다음, 비디오 디코더 (300) 는 예측 블록과 잔차 블록을 (샘플 바이 샘플 기준으로) 결합하여 원래 블록을 재현할 수도 있다. 비디오 디코더 (300) 는 블록의 경계들을 따라 시각적 아티팩트들을 감소시키기 위해 디블록킹 프로세스를 수행하는 것과 같은 추가적인 프로세싱을 수행할 수도 있다.

[0063] 본 개시의 기법들에 따르면, 비디오 디코더 (300) 는 변환 블록의 현재 계수에 대한 엔트로피 인코딩된 값을 수신할 수도 있다. 비디오 디코더 (300) 는 엔트로피 인코딩된 값을 엔트로피 디코딩하여 현재 계수의 값을 나타내는 Golomb-Rice 코드를 형성할 수도 있다. 비디오 디코더 (300) 는 현재 계수에 대한 이웃하는 계수들의 로컬 절대 값들의 합을 계산할 수도 있다. 비디오 디코더 (300) 는 또한 예를 들어, 아래에서 논의되는 바와 같이 식 (1) 을 사용하여 베이스 레벨 오프셋 값을 계산할 수도 있다. 비디오 디코더 (300) 는 베이스 레벨 오프셋 값을 사용하여, 예를 들어 아래에서 논의되는 바와 같은 식 (2) 를 사용하여, 절대 값들의 합에 대한 정규화된 값을 계산할 수도 있다. 비디오 디코더 (300) 는 그 후 예를 들어, 아래에서 논의되는 바와 같이 표 1 을 이용하여 정규화된 값으로부터 라이스 파라미터를 결정할 수도 있다. 비디오 디코더 (300)

는 그 후 Golomb-Rice 코드로부터 그리고 라이스 파라미터를 사용하여 현재 계수에 대한 값 (예를 들어, 절대 나머지 값) 을 이진화할 수도 있다. 비디오 디코더 (300) 는 절대 나머지 값 및 현재 계수에 대한 다른 디코딩된 값들, 예를 들어 부호 값, 유효 변환 계수 값, 1 초과 값, 및 2 초과 값으로부터 현재 계수를 재구성할 수도 있다.

[0064] 본 개시는 일반적으로 선택스 엘리먼트들과 같은 특정 정보를 "시그널링" 하는 것을 언급할 수도 있다. 용어 "시그널링" 은 일반적으로, 인코딩된 비디오 데이터를 디코딩하는데 사용되는 선택스 엘리먼트들에 대한 값들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 즉, 비디오 인코더 (200) 는 비트스트림에서 선택스 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 선택스 엘리먼트들을 저장할 때 발생할 수도 있는 바와 같이, 비실시간으로 또는 실질적으로 실시간으로 비트스트림을 목적지 디바이스 (116) 로 전송할 수도 있다.

[0065] 도 2 는 로컬 절대값들의 합 (locSumAbs) 을 계산하기 위해 사용될 수 있는 현재 계수에 대한 이웃 계수들의 예를 예시하는 개념도이다. VVC 의 현재 설계에서, 템플릿에 따른 이웃 변환 계수들의 계수 값들을 고려한 룩업 테이블을 사용하여 RRC (regular residual coding) 를 위한 라이스 파라미터를 도출한다. 도 2 에서 회색 음영처리된 이웃 계수는 도 2 에서 회색 음영처리된 현재 계수에 대한 템플릿을 나타낸다.

[0066] 특히, 도 2 는 다양한 변환 계수들을 포함하는 현재 블록 (130) 을 도시한다. 현재 블록 (130) 은 현재 계수 (132) 및 이웃 계수들 (134A-124E)(이웃 계수들 (134)) 을 포함한다. 현재 계수 (132) 의 포지션에 대해, 이웃 계수 (134A) 는 우측-이웃 계수를 나타내고, 이웃 계수 (134B) 는 2-우측-이웃 계수를 나타내고, 이웃 계수 (134C) 는 하방-우측-이웃 계수를 나타내고, 이웃 계수 (134D) 는 하방-이웃 계수를 나타내고, 이웃 계수 (134E) 는 2-하방-이웃 계수를 나타낸다.

[0067] 비디오 코더는 로컬 절대 값들의 합 (locSumAbs), 예를 들어 현재 계수 (132) 에 대한 이웃 계수들 (134) 의 절대 값들을 계산할 수도 있다. VVC 에 따르면, 비디오 코더는 locSumAbs 에 대한 값을 도 2 에서와 같이 5 개의 이용가능한 이웃 계수들의 절대 값들의 합으로서 계산한다. VVC 의 일 예에 대해, 비디오 코더는 그 후 다음의 식에 따라 (감산 및 클리핑을 사용하여) locSumAbs 의 값을 정규화한다:

[0068]
$$\text{locSumAbs} = \text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevel} * 5)$$

[0069] 그 후, VVC 마다, 비디오 코더는 아래의 표 1 을 이용하여 locSumAbs 의 정규화된 값을 이용하여 라이스 파라미터를 도출한다:

표 1

locSumAbs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
cRiceParam	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2	2
locSumAbs	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
cRiceParam	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3

[0070]

[0071] VVC 에서, 라이스 파라미터 범위는 0 내지 3 (한계 포함) 으로 제한된다.

[0072] 본 개시의 기법들에 따르면, 비디오 코더는 이하에서 더 상세히 논의되는 바와 같이 식 (1) 에 따라 베이스 레벨 오프셋을 계산할 수도 있다. 즉, 비디오 코더는 현재 블록 (130) 을 포함하는 슬라이스에 대한 비트 심도 및/또는 슬라이스 타입에 따라 베이스 레벨 오프셋 값을 계산할 수도 있다.

[0073] 도 3 은 비디오 데이터의 16-비트 테스트 시퀀스에 대한 코딩된 변환 계수들의 경험적 히스토그램을 예시하는 그래프 (136) 이다.

[0074] 도 4 는 계수 값들에 대한 엔트로피 코딩 동안 생성된 코드워드 길이들 및 주어진 라이스 파라미터들 사이의 관계를 예시하는 그래프 (140) 이다.

- [0075] 라이스 파라미터는 8 비트-깊이 또는 10 비트-깊이의 비디오 소스들을 인코딩하기 위해 광범위하게 테스트되었다. 하나의 예시적인 설계에서, 라이스 파라미터 값은 locSumAbs 의 값에 의존하고, 클립핑에 의해 0 내지 3 의 범위로 제한된다. 입력 비디오의 비트깊이가 증가하거나, 확장된 정밀도가 인에이블되거나, 양자화 파라미터가 매우 낮게 설정되면, 계수들의 범위 및 이에 따라서 locSumAbs 값은 상당히 증가할 수 있다. 이러한 경우들에서, 허용된 라이스 파라미터의 현재 VVC 범위는, 큰 값들의 코딩된 계수들에 대해 긴 코드 워드들, 예를 들어, 32 비트들의 코드워드들의 사용을 요구할 것이기 때문에, 충분하지 않다.
- [0076] 도 3 및 도 4 는 그 결과 발생할 수 있는 문제들의 시각화를 제공한다. 특히, 도 3 은 16 비트 신호에 대한 (2 의 거듭제곱 형태의) 코딩된 계수들의 히스토그램을 도시한다. 도 4 는 코딩된 값들에 대한 VVC 엔트로피 코딩 방법들에 의해 생성된 코드워드 길이들을 도시한다. 도 4 로부터 알 수 있는 바와 같이, (VVC 에서 규정된 바와 같이) 3 으로 제한된 라이스 파라미터의 규범적 범위는 10 보다 큰 코딩된 값들에 대한 코드워드 길이의 현저한 증가를 초래한다. 이 문제는 더 큰 크기의 계수들에 대해 라이스 파라미터의 큰 값을 허용함으로써 해결될 수 있다.
- [0077] 비디오의 다양한 입력 비트-심도들에서의 현재 VVC 의 라이스 파라미터 도출의 제한을 해결하고 따라서 압축 효율을 개선하기 위한 일 설계 (JVET-U 미팅의 CE1.1) 에서, 라이스 파라미터들의 지원되는 범위는 VVC 에서의 3 으로부터 큰 수, 예를 들어, 16 까지 확장될 수 있으며, 이는 더 효율적인 이진화 프로세스를 제공하여, 변환 계수들의 큰 값들에 대해 더 적은 수의 비트들을 초래한다.
- [0078] localSumbAbs 의 값은, 예를 들어 VVC 사양의 식 (1517) 에 따라, 라이스 파라미터를 도출하는데 사용되기 전에 변환 계수들의 비트심도 증가 또는 동적 범위를 핸들링하도록 스케일링/정규화될 수도 있다. 스케일링 팩터의 양은 값 localSumAbs - 변환 계수들의 로컬 활동을 나타내기 위해 계산된 템플릿 도출의 출력, 또는 비트스 트림 또는 테이블화된 값들의 세트에서 시그널링된 선택스 엘리먼트에 의존할 수도 있다.
- [0079] VVC 비디오 인코더는 localSumAbs 의 결정된 값을 테이블화된 임계치들의 세트 $T_x = \{T_{id}\}$ 와 비교하여, 값을 포함하는 동적 범위에 대한 동적 범위 id 를 식별할 수 있다. 그 후, 비디오 코더는 범위 id 에 의해 결정된 테이블화된 $R_x = \{R_{id}\}$ 의 주어진 세트로부터의 값 스칼라 normShift 에 의해 localSumAbs 의 값을 스케일링 및/또는 정규화할 수도 있다. 그 후, 비디오 코더는 미리 정해진 룩업 테이블, 예를 들어 VVC 에서의 현재 룩업 테이블, 테이블 128 을 이용하여 라이스 파라미터를 도출하기 위해 정규화된 localSumAbs 를 이용할 수도 있다. localSumbAbs 의 값이 제안된 설계의 제 1 단계에서 정규화되면, 라이스 파라미터는 미리 정해진 테이블, 예를 들어 테이블 128 로부터 도출될 수 있고, 최종적으로 라이스 파라미터 범위의 동적 범위를 확장하기 위해 R_{id} 와 동일한 오프셋을 추가함으로써 수정될 수 있다.
- [0080] 임계치들 T_x 및 스케일러 R_x 의 세트는 다음과 같이 규정될 수 있다:
- [0081] $T_x[8] = \{ 32, 128, 512, 2048, 8192, 32768, 131072, 524288 \};$
- [0082] $R_x[9] = \{ 0, 2, 4, 6, 8, 10, 12, 14, 16 \};$
- [0083] 어레이 T_x 의 값들은 2 의 거듭제곱으로 제한될 수 있고, 다른 실시형태들에서, 값들은 2 의 거듭제곱 (2^x) 에 의해 동기화되고, 제한된 수의 동작들을 통해, 오프셋, 스케일링 또는 우측/좌측 시프트와 같은 것으로 변환될 수 있다.
- [0084] 어레이의 사이즈는 N 과 동일할 수 있으며, 여기서 N 은 하부 경계보다 큰 양의 정수, 예를 들어 1 초과 또는 2 초과이고 상부 경계보다 작은 양의 정수, 예를 들어 9 또는 5 이다. 일부 실시형태들에서, R_x 어레이의 사이즈는 1 엔트리만큼 T_x 의 어레이 보다 더 크다. 일부 실시형태들에서, T_x 및 R_x 어레이들의 사이즈는 4 로 제한될 수 있다.
- [0085] 일부 예들에서, VVC 에서의 관련된 라이스 파라미터 도출 부분은 다음과 같이 수정될 수도 있고, 테이블화된 어레이들의 사이즈는 4 와 동일하다.

9.3.3.2 abs_remainder[] 및 dec_abs_level[] 에 대한 라이스 파라미터 도출 프로세스

이 프로세스에 대한 입력들은 베이스 레벨 baseLevel, 컬러 컴포넌트 인덱스 cldx, 현재 화상의 상단-좌측 샘플에 상대적인 현재 변환 블록의 상단-좌측 샘플을 특징하는 루마 위치 (x0, y0), 현재 계수 스캔 위치 (xC, yC), 변환 블록 폭의 이진 로그 log2TbWidth, 및 변환 블록 높이의 이진 로그 log2TbHeight 이다.

이 프로세스의 출력은 라이스 파라미터 cRiceParam 이다.

컴포넌트 인덱스 cldx 및 상단-좌측 루마 위치 (x0, y0) 를 갖는 변환 블록에 대한 어레이 AbsLevel[x][y] 가 주어지면, 변수 locSumAbs 는 다음의 의사-코드 프로세스에 의해 특정된 바와 같이 도출된다:

```
locSumAbs = 0
if( xC < ( 1 << log2TbWidth ) - 1 ) {
    locSumAbs += AbsLevel[ xC + 1 ][ yC ]
    if( xC < ( 1 << log2TbWidth ) - 2 )
        locSumAbs += AbsLevel[ xC + 2 ][ yC ]
    if( yC < ( 1 << log2TbHeight ) - 1 )
        locSumAbs += AbsLevel[ xC + 1 ][ yC + 1 ]
}
if( yC < ( 1 << log2TbHeight ) - 1 ) {
    locSumAbs += AbsLevel[ xC ][ yC + 1 ]
    if( yC < ( 1 << log2TbHeight ) - 2 )
        locSumAbs += AbsLevel[ xC ][ yC + 2 ]
}

sum = localSumAbs
if( sum < g_riceT[0] )
    rangeIdx = 0
else if( sum < g_riceT[1] )
    rangeIdx = 1
else if( sum < g_riceT[2] )
    rangeIdx = 2
else if( sum < g_riceT[3] )
    rangeIdx = 3
else
    rangeIdx = 4

normShift = g_riceShift[rangeIdx]
locSumAbs = locSumAbs >> normShift
locSumAbs = Clip3( 0, 31, locSumAbs - baseLevel * 5 )
```

[0086]

g_riceT 및 g_riceShift 는 다음과 같이 규정된다:

g_riceT[4] = { 32, 128, 512, 2048 };

g_riceShift[5] = { 0, 2, 4, 6, 8 };

변수 locSumAbs 가 주어지면, 라이스 파라미터 cRiceParam 은 테이블 128 에 특정된 바와 같이 도출된다.

그 후, cRiceParam 는 다음과 같이 세밀하게 구분된다:

cRiceParam = cRiceParam + normShift

[0087]

[0088] 라이스 파라미터들의 범위를 확장하기 위해, U 를 충족하는 CE1.2 에서, locSumAbs 가 특정 임계치를 초과하면, 시프트를 갖는 locSumAbs 값들을 스케일링하여, 스케일링된 값이 클리핑 없이 VVC 사양의 테이블 128 의 허용된 범위에 맞추어지도록 하는 것이 제안된다. 이에 후속하여, 테이블 128 의 출력은, 스케일링 프로세스가 주어진 locSumAbs 값에 대해 이루어졌다면, 시프트 값으로 증분함으로써 조정되고 있다.

[0089] 인코더 (예를 들어, 비디오 인코더 (200)) 는 값들 1 또는 2 를 취할 수도 있는 선택스 엘리먼트 sh_scale_rice 에 대한 값을 선택할 수도 있다.

[0090]
$$\text{scaledCoef} = (\text{localSumAbs} * \text{sh_scale_rice}) \gg 3$$

[0091]
$$\text{shift} = (\text{scaledCoef} == 0) ? 0 : \text{Floor}(\text{Log2}(\text{scaledCoef}))$$

[0092] 이용된 값 sh_scale_rice 는 슬라이스 헤더에서 선택스 엘리먼트로서 시그널링되고 인코더 측에서 QP 값들의 함수로서 도출된다.

[0093] VVC 에 대한 사양 텍스트는 다음과 같을 수 있다:

[0094] 1 과 동일한 sps_high_bit_depth 는, abs_remaining[] 및 dec_abs_level 의 이진화를 위한 라이스 파라미터에 대한 높은 비트심도 모드를 가능하게 하기 위해 sh_rice_parameter 선택스 엘리먼트가 슬라이스 헤더들에 존재할 수 있다는 것을 특정한다. 0 과 동일한 sps_high_bit_depth 는 sh_rice_parameter 선택스 엘리먼트가 슬라이스 헤더들에 존재하지 않고 라이스 파라미터 도출에 대한 높은 비트심도 모드가 인에이블될 수 없음을 특정한다. 존재하지 않을 때, sps_high_bit_depth 의 값은 0 과 동일한 것으로 추론된다.

[0095] 7.3.7 슬라이스 헤더 선택스

...	
if(sps_dep_quant_enabled_flag)	
sh_dep_quant_used_flag	u(1)
if(sps_high_bit_depth)	
sh_scale_rice	ue(v)
if(sps_sign_data_hiding_enabled_flag && !sh_dep_quant_used_flag)	
sh_sign_data_hiding_used_flag	u(1)

[0096]

[0097] sh_scale_rice 는 라이스 파라미터의 도출을 위해 이용되는 스케일 값을 특정하며, sh_scale_rice 의 값은 1 내지 2 (한계 포함) 의 범위에 있을 것이다. sh_scale_rice 가 존재하지 않을 때, 0 과 동일한 것으로 추론된다.

9.3.3.2 abs_remainder[] 및 dec_abs_level[] 에 대한 라이스 파라미터 도출 프로세스

이 프로세스에 대한 입력들은 베이스 레벨 baseLevel, 컬러 컴포넌트 인덱스 cldx, 현재 화상의 상단-좌측 샘플에 상대적인 현재 변환 블록의 상단-좌측 샘플을 특징하는 루마 위치 (x0, y0), 현재 계수 스캔 위치 (xC, yC), 변환 블록 폭의 이진 로그 log2TbWidth, 및 변환 블록 높이의 이진 로그 log2TbHeight 이다.

이 프로세스의 출력은 라이스 파라미터 cRiceParam 이다.

컴포넌트 인덱스 cldx 및 상단-좌측 루마 위치 (x0, y0) 를 갖는 변환 블록에 대한

어레이 AbsLevel[x][y] 가 주어지면, 변수 locSumAbs 는 다음의 의사-코드 프로세스에 의해 특정된 바와 같이 도출된다:

```
locSumAbs = 0
if( xC < ( 1 << log2TbWidth ) - 1 ) {
    locSumAbs += AbsLevel[ xC + 1 ][ yC ]
    if( xC < ( 1 << log2TbWidth ) - 2 )
        locSumAbs += AbsLevel[ xC + 2 ][ yC ]
    if( yC < ( 1 << log2TbHeight ) - 1 )
        locSumAbs += AbsLevel[ xC + 1 ][ yC + 1 ]
}
if( yC < ( 1 << log2TbHeight ) - 1 ) {
    locSumAbs += AbsLevel[ xC ][ yC + 1 ]
    if( yC < ( 1 << log2TbHeight ) - 2 )
        locSumAbs += AbsLevel[ xC ][ yC + 2 ]
}
변수 시프트는 다음과 같이 도출된다:
scaledCoef = (localSumAbs * sh_scale_rice) >> 3
shift = (scaledCoef == 0) ? 0 : Floor( Log2( scaledCoef ) )
locSumAbs = locSumAbs >> shift
변수 localSumAbs 는 다음과 같이 업데이트된다:
locSumAbs = Clip3( 0, 31, locSumAbs - baseLevel * 5 )
변수 locSumAbs 가 주어지면, 라이스 파라미터 cRiceParam 은 테이블 128 에 특정된 바와 같이 도출된다.
그 후, cRiceParam 는 다음과 같이 세밀하게 구분된다:
cRiceParam = cRiceParam + shift
```

[0098]

[0099]

위에서 논의된 RRC 방법은 아래에서 논의되는 바와 같이 사용된 라이스 파라미터들의 히스토리로 확장될 수 있다:

[0100]

컬러 컴포넌트당 단일 카운터가 사용되고, StatCoeff[3] 은 제 1 논-제로 Golomb-Rice 코딩된 변환 계수 (abs_remainder[] 또는 dec_abs_level[]) 로부터 TU 당 한번 업데이트된다.

[0101]

TU 에서의 제 1 논-제로 Golomb-Rice 코딩된 변환 계수가 abs_remainder 로서 코딩될 때, 컬러 컴포넌트 cIdx 에 대한 히스토리 카운터 (StatCoeff) 는 다음과 같이 업데이트된다:

[0102]

```
StatCoeff[cIdx] = ( StatCoeff[cIdx] + Floor(Log2(abs_remainder[ ])) + 2 ) >> 1
```

[0103]

TU 에서의 제 1 논-제로 Golomb-Rice 코딩된 변환 계수가 abs_remainder 로서 코딩될 때, 컬러 컴포넌트 cIdx 에 대한 히스토리 카운터는 다음과 같이 업데이트된다:

[0104] $\text{StatCoeff}[\text{cIdx}] = (\text{StatCoeff}[\text{cIdx}] + \text{Floor}(\text{Log2}(\text{dec_abs_level}[]))) \gg 1$

[0105] 히스토리 카운터 $\text{StatCoeff}[3]$ 는 디폴트 값으로 CTU 레벨에서 리셋될 수 있거나, 히스토리는 상이한 CTU들 사이에서 전파될 수 있다.

[0106] 히스토리 라이스 파라미터 값은 템플릿 도출, 조항 9.3.3.2 에서 이용되며, 이 방법의 구현은 조항 9.3.3.2 에 대한 VVC 사양 텍스트 추출이 도시되고, 녹색으로 하이라이트된 텍스트는 CE-1.2 로부터의 변경들과 관련되고, 황색 하이라이트된 텍스트는 CE-1.4 에서 이용되는 분류와 연관된다. 변수 g_historyValue 는 현재 컬러 컴포넌트 id 와 연관된 히스토리 카운터로부터 현재 TU 의 코딩 전에 도출된다.

[0107] $\text{histValue} = 1 \ll \text{StatCoeff}[\text{cIdx}]$

9.3.3.2 abs_remainder[] 및 dec_abs_level[] 에 대한 라이스 파라미터 도출 프로세스

이 프로세스에 대한 입력들은 베이스 레벨 baseLevel , 컬러 컴포넌트 인덱스 cIdx , 현재 화상의 상단-좌측 샘플에 상대적인 현재 변환 블록의 상단-좌측 샘플을 특정하는 루마 위치 (x_0, y_0), 현재 계수 스캔 위치 (x_C, y_C), 변환 블록 폭의 이진 로그 $\log_2 \text{TbWidth}$, 및 변환 블록 높이의 이진 로그 $\log_2 \text{TbHeight}$ 이다.

이 프로세스의 출력은 라이스 파라미터 cRiceParam 이다.

컴포넌트 인덱스 cIdx 및 상단-좌측 루마 위치 (x_0, y_0) 를 갖는 변환 블록에 대한 어레이 $\text{AbsLevel}[x][y]$ 가 주어지면, 변수 locSumAbs 는 다음의 의사-코드 프로세스에 의해 특정된 바와 같이 도출된다:

```
locSumAbs = 0
if(  $x_C < ( 1 \ll \log_2 \text{TbWidth} ) - 1$  ) {
    locSumAbs += AbsLevel[  $x_C + 1$  ][  $y_C$  ]
    if(  $x_C < ( 1 \ll \log_2 \text{TbWidth} ) - 2$  )
        locSumAbs += AbsLevel[  $x_C + 2$  ][  $y_C$  ]
    else
        locSumAbs += histValue
    if(  $y_C < ( 1 \ll \log_2 \text{TbHeight} ) - 1$  )
        locSumAbs += AbsLevel[  $x_C + 1$  ][  $y_C + 1$  ]
    else
        locSumAbs += histValue
}
else
    locSumAbs += 2 * histValue
if(  $y_C < ( 1 \ll \log_2 \text{TbHeight} ) - 1$  ) {
    locSumAbs += AbsLevel[  $x_C$  ][  $y_C + 1$  ]
    if(  $y_C < ( 1 \ll \log_2 \text{TbHeight} ) - 2$  )
        locSumAbs += AbsLevel[  $x_C$  ][  $y_C + 2$  ]
    else
        locSumAbs += histValue
}
```

[0108]

```

else
    locSumAbs += histValue
변수 시프트는 다음과 같이 도출된다:
scaledCoef = (localSumAbs * ScaleRice) >> 3
shift = (scaledCoef == 0) ? 0 : Floor( Log2( scaledCoef ) )
locSumAbs = locSumAbs >> shift
변수 localSumAbs 는 다음과 같이 업데이트된다:
locSumAbs = Clip3( 0, 31, locSumAbs - baseLevel * 5 )
변수 locSumAbs 가 주어지면, 라이스 파라미터 cRiceParam 은
테이블 128 에 특정된 바와 같이 도출된다.
그 후, cRiceParam 는 다음과 같이 세밀하게 구분된다:
cRiceParam = cRiceParam + shift

```

[0109]

[0110] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 위에서 논의된 바와 같이, 본 개시의 기법들에 따라 구성될 수도 있다. 이들 기법들은 위에서 논의된 기법들에 대한 개정들을 포함한다.

[0111] 일부 예들에서, 위에서 논의된 로컬-기반 라이스 도출 기법들에 대해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 입력 신호 비트 심도 및/또는 슬라이스 타입의 함수로서 베이스 레벨 오프셋을 도출하고, 아래에 도시된 바와 같이, locSumAbs 값을 스케일링하기 위해 baseLevel 오프셋 값을 사용할 수도 있다.

[0112] 일부 예들에 있어서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다음과 같이 baseLevelOffset 을 도출할 수 있다:

[0113]
$$\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1 : P2) : (\text{isIntra}() ? P3 : P4) \quad (1)$$

[0114] 식 (1) 에서, 물음표 마크 연산자 (?) 는 $\langle \text{condition} \rangle ? X : Y$ 의 형태로 수식을 표현하고, 여기서 $\langle \text{condition} \rangle$ 이 진실이면, X 의 값이 복귀되지만, $\langle \text{condition} \rangle$ 이 거짓이면, Y 의 값이 복귀된다. baseLevelOffset 값은 베이스 레벨 오프셋을 나타내고, bitDepth 는 비디오 비트스트림의 비트 심도를 나타내며, T 는 임계치이고, P1, P2, P3, 및 P4 는 비트스트림에서 미리 결정되거나 시그널링될 수도 있는 값들이다.

[0115] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다음과 같이 localSumAbs 를 정규화하기 위해 베이스 레벨 오프셋 값을 사용할 수도 있다:

[0116]
$$\text{locSumAbs} = \text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset}) \quad (2)$$

[0117] 일반적으로, T 는 임계치 값을 나타내고, P1, P2, P3, P4 값은 정수값을 나타낸다. 일부 예에서, T = 12, P1 = 5, P2 = 10, P3 = 6, P4 = 20 이다. 일부 예에서, T = 12, P1 = 5, P2 = 10, P3 = 10, P4 = 20 이다. T, P1, P2, P3, 및 P4 에 대한 다른 값들이 또한 사용될 수 있다. T, P1, P2, P3, 및 P4 는 미리 정해지거나, 비트스트림에서 시그널링되거나, 또는 다양한 기준들에 기반하여 도출될 수도 있다.

[0118] 일부 예들에서, 위에서 논의된 히스토리-기반 라이스 도출 기법들에 대해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 특정 블록들 (예를 들어, 특정 CTU들) 사이의 히스토리 카운터들 StatCoeff[cIdx] 의 상태를 보존

할 수도 있다. 추가적으로 또는 대안적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 StatCoeff[] 카운터의 현재 상태에서부터 블록 레벨, 예를 들어 CTU 레벨에서 (명시적으로 시그널링된 sh_scale_rice 선택스 엘리먼트를 대체하는) ScaleRice 파라미터를 도출할 수도 있다. 따라서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 sh_rice_init 선택스 엘리먼트를 명시적으로 시그널링할 필요가 없다.

[0119] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 StatCoeff[] 의 상태를 임계치에 대해 비교하고 정수 값들의 범위에서 파라미터의 정수 값을 특정함으로써 ScaleRice 파라미터 값을 도출할 수도 있다. 도출의 하나의 이러한 예가 아래에 도시된다:

[0120] $\text{ScaleRice} = (\text{StatCoeff}[\text{idx}] \geq T) ? P2 : P1;$

[0121] 일부 예들에서, $T = 4$, $P2 = 2$ 및 $P1 = 1$ 이다.

[0122] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각의 컬러 컴포넌트 (예를 들어, 루마, 청색 색조 크로마, 및 적색 색조 크로마) 에 대해 독립적으로 파라미터 ScaleRice 를 도출할 수도 있다.

[0123] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는, ScaleRice 파라미터에 대한 값을 도출할 때, 아래에 도시된 바와 같이, 집성 함수, 예를 들어 컬러 컴포넌트들에 걸친 가중된 함수를 통해 2 개 이상의 (예를 들어, 모든) 컬러 컴포넌트들에 대한 히스토리 카운터들에 대한 값들을 고려할 수도 있다:

[0124] $\text{StatCoefAverage} = ((\text{StatCoeff}[0] * w0) + \text{StatCoeff}[1]*w1 + \text{StatCoeff}[2]*w2)/(w1+w2+w3);$

[0125] $\text{ScaleRice} = (\text{StatCoefAverage} \geq T) ? P2 : P1;$

[0126] 일부 예들에서, $w0 = 2$, $w1 = 1$, $w2=1$ 이다.

[0127] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 블록 레벨에서, 예를 들어 CTU들, TU들, 또는 서브-TU 의 레벨에서 파라미터 ScaleRice 를 도출할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 데이터의 후속 코딩된 블록의 데이터를 디코딩할 때 ScaleRice 파라미터를 사용할 수도 있다. 대안적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는, 예를 들어 다음 블록 이후의 블록을 디코딩하기 위해, 도출된 ScaleRice 값을 일부 블록 사이즈 만큼 지연된 방식으로 사용할 수도 있다.

[0128] 따라서, if(sps_high_bit_depth) 및 sh_scale_rice 선택스 엘리먼트에 대한 체크는 선택스 테이블로부터 제거될 수 있다.

[0129] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 디폴트 또는 시그널링된 값들을 갖는 StatCoeff 의 초기화에 후속하여 상술된 프로세스를 통해 ScaleRice 값을 도출할 수도 있다.

[0130] 도 5 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코더 (200) 를 예시하는 블록 다이어그램이다. 도 5 는 설명의 목적들로 제공되며, 본 개시에 대체로 예시화되고 설명된 바와 같은 기법들의 한정으로서 고려되지 않아야 한다. 설명의 목적으로, 본 개시는 VVC (ITU-T H.266, 개발중) 및 HEVC (ITU-T H.265) 의 기법들에 따른 비디오 인코더 (200) 를 기술한다. 그러나, 본 개시의 기법들은 AV1 및 AV1 비디오 코딩 포맷에 대한 후속자들 (successors) 과 같은 다른 비디오 코딩 표준들 및 비디오 코딩 포맷들로 구성되는 비디오 인코딩 디바이스들에 의해 수행될 수도 있다.

[0131] 도 5 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), 복원 유닛 (214), 필터 유닛 (216), 디코딩된 화상 버퍼 (DPB) (218), 및 엔트로피 인코딩 유닛 (220) 을 포함한다. 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), 복원 유닛 (214), 필터 유닛 (216), DPB (218), 및 엔트로피 인코딩 유닛 (220) 중 임의의 것 또는 전부는 하나 이상의 프로세서들에서 또는 프로세싱 회로에서 구현될 수도 있다. 예를 들어, 비디오 인코더 (200) 의 유닛들은 하드웨어 회로부의 일부로서 또는 FPGA 의 프로세서, ASIC 의 일부로서 하나 이상의 회로들 또는 로직 엘리먼트들로서 구현될 수도 있다. 더욱이, 비디오 인코더 (200) 는 이들 및 다른 기능들을 수행하기 위해 추가적인 또는 대안적인 프로세서들 또는 프로세싱 회로부를 포함할 수도 있다.

[0132] 비디오 데이터 메모리 (230) 는 비디오 인코더 (200) 의 컴포넌트들에 의해 인코딩된 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 비

디오 데이터의 예측에서의 사용을 위한 레퍼런스 비디오 데이터를 저장하는 레퍼런스 화상 메모리로서 작용할 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동기식 DRAM (SDRAM) 을 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 같은 비디오 인코더 (200) 의 다른 컴포넌트들과 온-칩이거나 또는 그들 컴포넌트들에 대해 오프-칩일 수도 있다.

[0133] 본 개시에서, 비디오 데이터 메모리 (230) 에 대한 참조는, 이와 같이 구체적으로 설명되지 않으면 비디오 인코더 (200) 내부의 메모리, 또는 이와 같이 구체적으로 설명되지 않으면 비디오 인코더 (200) 외부의 메모리로 한정되는 것으로서 해석되지 않아야 한다. 오히려, 비디오 데이터 메모리 (230) 에 대한 참조는, 비디오 인코더 (200) 가 인코딩을 위해 수신하는 비디오 데이터 (예컨대, 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.

[0134] 도 5 의 다양한 유닛들은 비디오 인코더 (200) 에 의해 수행되는 동작들의 이해를 돕기 위해 예시된다. 그 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 고정 기능 회로들은 특정 기능을 제공하는 회로들을 지칭하고, 수행될 수 있는 동작들에 대해 미리 설정된다. 프로그래밍가능 회로들은 다양한 태스크들을 수행하도록 프로그래밍될 수도 있는 회로들을 지칭하고, 수행될 수도 있는 동작들에서 유연한 기능성을 제공한다. 예를 들어, 프로그래밍가능 회로들은, 프로그래밍가능 회로들로 하여금 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, 유닛들 중 하나 이상은 집적 회로들일 수도 있다.

[0135] 비디오 인코더 (200) 는 프로그래밍가능 회로들로부터 형성된, 산술 로직 유닛들 (ALU들), EFU들 (elementary function units), 디지털 회로들, 아날로그 회로들, 및/또는 프로그래밍가능 코어들을 포함할 수도 있다. 비디오 인코더 (200) 의 동작들이 프로그래밍가능 회로들에 의해 실행된 소프트웨어를 사용하여 수행되는 예들에 있어서, 메모리 (106) (도 1) 는 비디오 인코더 (200) 가 수신하고 실행하는 소프트웨어의 명령들 (예컨대, 물체 코드) 을 저장할 수도 있거나, 또는 비디오 인코더 (200) 내의 다른 메모리 (도시 안됨) 가 그러한 명령들을 저장할 수도 있다.

[0136] 비디오 데이터 메모리 (230) 는 수신된 비디오 데이터를 저장하도록 구성된다. 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 비디오 데이터의 화상을 추출하고, 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.

[0137] 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 추가적인 기능 유닛들을 포함할 수도 있다. 예를로서, 모드 선택 유닛 (202) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 추정 유닛 (222) 및/또는 모션 보상 유닛 (224) 의 부분일 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다.

[0138] 모드 선택 유닛 (202) 은 일반적으로 인코딩 파라미터들의 조합들 및 그러한 조합들에 대한 결과적인 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 더 나은 레이트-왜곡 값들을 갖는 인코딩 파라미터들의 조합을 선택할 수도 있다.

[0139] 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 추출된 화상을 일련의 CTU들로 파티셔닝하고, 슬라이스 내에 하나 이상의 CTU들을 캡슐화할 수도 있다. 모드 선택 유닛 (202) 은 MTT 구조, QTBT 구조, 수퍼블록 구조, 또는 상기 설명된 쿼드트리 구조와 같은 트리 구조에 따라 화상의 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" 으로서 또한 지칭될

수도 있다.

- [0140] 일반적으로, 모드 선택 유닛 (202) 은 또한 그의 컴포넌트들 (예컨대, 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라-예측 유닛 (226)) 을 제어하여 현재 블록 (예컨대, 현재 CU, 또는 HEVC 에서, PU 및 TU 의 오버랩하는 부분) 에 대한 예측 블록을 생성한다. 현재 블록의 인트라-예측을 위해, 모션 추정 유닛 (222) 은 하나 이상의 레퍼런스 화상들 (DPB (218) 에 저장된 하나 이상의 이전에 코딩된 화상들) 에서 하나 이상의 근접하게 매칭하는 레퍼런스 블록들을 식별하기 위해 모션 검색을 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 블록에 가장 근접하게 매칭하는 레퍼런스 블록을 표시하는, 이들 계산들로부터 발생하는 최저 값을 갖는 레퍼런스 블록을 식별할 수도 있다.
- [0141] 모션 추정 유닛 (222) 은 현재 화상에서의 현재 블록의 위치에 대한 레퍼런스 화상들에서의 레퍼런스 블록들의 위치들을 정의하는 하나 이상의 모션 벡터들 (MV들) 을 형성할 수도 있다. 그 다음, 모션 추정 유닛 (222) 은 모션 벡터들을 모션 보상 유닛 (224) 에 제공할 수도 있다. 예를 들어, 단방향 인트라-예측에 대해, 모션 추정 유닛 (222) 은 단일 모션 벡터를 제공할 수도 있는 반면, 양방향 인트라-예측에 대해, 모션 추정 유닛 (222) 은 2 개의 모션 벡터들을 제공할 수도 있다. 그 다음, 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 추출 (retrieve) 할 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. 더욱이, 양방향 인트라-예측에 대해, 모션 보상 유닛 (224) 은 개별의 모션 벡터들에 의해 식별된 2 개의 레퍼런스 블록들에 대한 데이터를 추출하고, 예컨대, 샘플 별 평균화 또는 가장 평균화를 통해 추출된 데이터를 결합할 수도 있다.
- [0142] AV1 비디오 코딩 포맷에 따라 동작할 때, 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 병진 모션 보상, 아핀 모션 보상, 중첩된 블록 모션 보상 (OBMC), 및/또는 복합 인트라-인트라 예측을 사용하여 비디오 데이터의 코딩 블록들 (예를 들어, 루마 및 크로마 코딩 블록들 둘 다) 을 인코딩하도록 구성될 수도 있다.
- [0143] 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 파플레이팅하여 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 결과적인 평균을 포함할 수도 있다.
- [0144] AV1 비디오 코딩 포맷에 따라 동작할 때, 인트라 예측 유닛 (226) 은 방향성 인트라 예측, 비-방향성 인트라 예측, 재귀적 필터 인트라 예측, CFL (chroma-from-luma) 예측, 인트라 블록 카피 (IBC), 및/또는 컬러 팔레트 모드를 사용하여 비디오 데이터의 코딩 블록들 (예를 들어, 루마 및 크로마 코딩 블록들 둘 다) 을 인코딩하도록 구성될 수도 있다. 모드 선택 유닛 (202) 은 다른 예측 모드들에 따라 비디오 예측을 수행하기 위해 추가적인 기능 유닛들을 포함할 수도 있다.
- [0145] 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 원시, 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 샘플별 차이들은 현재 블록에 대한 잔차 블록을 정의한다. 일부 예들에서, 잔차 생성 유닛 (204) 은 또한 RDPCM (residual differential pulse code modulation) 을 사용하여 잔차 블록을 생성하기 위해 잔차 블록에서의 샘플 값들 사이의 차이들을 결정할 수도 있다. 일부 예들에서, 잔차 생성 유닛 (204) 은 이진 감산을 수행하는 하나 이상의 감산기 회로들을 사용하여 형성될 수도 있다.
- [0146] 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 나타난 바와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있고 PU 의 사이즈는 PU 의 루마 예측 블록의 사이즈를 지칭할 수도 있다. 특정 CU 의 사이즈가 2Nx2N 임을 가정하면, 비디오 인코더 (200) 는 인트라 예측을 위해 2Nx2N 또는 NxN 의 PU 사이즈들을 지원하고, 인트라 예측을 위해 2Nx2N, 2NxN, Nx2N, NxN, 기타 등등의 대칭적인 PU 사이즈들을 지원할 수도 있다. 비

디오 인코더 (200) 및 비디오 디코더 (300) 는 또한, 인터 예측을 위해 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

- [0147] 모드 선택 유닛 (202) 이 CU 를 PU들로 추가로 파티셔닝하지 않는 예들에서, 각각의 CU 는 루마 코딩 블록 및 대응하는 크로마 코딩 블록들과 연관될 수도 있다. 상기에서와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 $2N \times 2N$, $2N \times N$, 또는 $N \times 2N$ 의 CU 사이즈들을 지원할 수도 있다.
- [0148] 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, 인코딩되고 있는 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 인코딩될 엔트로피 인코딩 유닛 (220) 에 제공할 수도 있다.
- [0149] 상술한 바와 같이, 잔차 생성 유닛 (204) 은 현재 블록 및 대응하는 예측 블록에 대한 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 샘플별 차이들을 계산한다.
- [0150] 변환 프로세싱 유닛 (206) 은 잔차 블록에 하나 이상의 변환들을 적용하여 변환 계수들의 블록 (본 명세서에서 "변환 계수 블록" 으로서 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, 예컨대 1 차 변환 및 2 차 변환, 이를 테면 회전 변환을 수행할 수도 있다. 일부 예들에 있어서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 변환들을 적용하지 않는다.
- [0151] AVI 에 따라서 동작할 때, 변환 프로세싱 유닛 (206) 은 잔차 블록에 하나 이상의 변환들을 적용하여 변환 계수들의 블록 (본 명세서에서는 "변환 계수 블록" 으로서 지칭됨) 을 생성할 수 있다. 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), 비대칭 이산 사인 변환 (ADST), 반전된 ADST (예를 들어, 역순으로 ADST) 및 항등 변환 (IDTX) 을 포함할 수 있는 수평/수직 변환 조합을 적용할 수 있다. 항등 변환을 사용하는 경우 수직 또는 수평 방향 중 하나로 변환을 스킵한다. 일부 예들에서, 변환 프로세싱은 스킵될 수도 있다.
- [0152] 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 (200) 는 (예컨대, 모드 선택 유닛 (202) 을 통해) CU 와 연관된 QP 값을 조정함으로써 현재 블록과 연관된 변환 계수 블록들에 적용되는 양자화를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서, 양자화된 변환 계수들은 변환 프로세싱 유닛 (206) 에 의해 생성된 원래 변환 계수들보다 더 낮은 정밀도를 가질 수도 있다.
- [0153] 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 재구성 잔차 블록의 샘플들을, 모드 선택 유닛 (202) 에 의해 생성된 예측 블록으로부터의 대응하는 샘플들에 가산하여 재구성된 블록을 생성할 수도 있다.
- [0154] 필터 유닛 (216) 은 재구성된 블록들에 대해 하나 이상의 필터 동작을 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.
- [0155] AVI 에 따라 동작할 때, 필터 유닛 (216) 은 재구성된 블록들에 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. 다른 예들에서, 필터 유닛 (216) 은

디블로킹 후에 적용될 수 있는 제약된 방향성 항상 필터 (CDEF) 를 적용할 수 있고, 추정된 예지 방향에 기반한 비분리형 비선형 저역 통과 방향성 필터의 적용을 포함할 수 있다. 필터 유닛 (216) 은 또한 CDEF 이후에 적용되는 루프 복원 필터를 포함할 수 있으며, 분리 가능한 대칭 정규화 Wiener 필터 또는 이중 자체 유도 필터를 포함할 수 있다.

[0156] 비디오 인코더 (200) 는 재구성된 블록들을 DPB (218) 에 저장한다. 예를 들어, 필터 유닛 (216) 의 동작들이 수행되지 않은 예들에 있어서, 복원 유닛 (214) 이, 복원된 블록들을 DPB (218) 에 저장할 수도 있다. 필터 유닛 (216) 의 동작들이 수행되는 예들에 있어서, 필터 유닛 (216) 은, 필터링된 재구성된 블록들을 DPB (218) 에 저장할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 복원된 (및 잠재적으로 필터링된) 블록들로부터 형성된 DPB (218) 로부터의 레퍼런스 화상을 추출하여, 후속적으로 인코딩된 화상들의 블록들을 인터-예측할 수도 있다. 추가로, 인트라-예측 유닛 (226) 은 현재 화상에서의 다른 블록들을 인트라-예측하기 위해 현재 화상의 DPB (218) 내의 복원된 블록들을 사용할 수도 있다.

[0157] 일반적으로, 엔트로피 인코딩 유닛 (220) 은 비디오 인코더 (200) 의 다른 기능 컴포넌트들로부터 수신된 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 양자화 유닛 (208) 으로부터의 양자화된 변환 계수 블록들을 엔트로피 인코딩할 수도 있다. 다른 예로서, 엔트로피 인코딩 유닛 (220) 은 모드 선택 유닛 (202) 으로부터 예측 선택스 엘리먼트들 (예를 들어, 인트라-예측에 대한 인트라-모드 정보 또는 인터-예측에 대한 모션 정보) 를 엔트로피 인코딩할 수도 있다. 엔트로피 인코딩 유닛 (220) 은 엔트로피 인코딩된 데이터를 생성하기 위해, 비디오 데이터의 다른 예인, 선택스 엘리먼트들에 대해 하나 이상의 엔트로피 인코딩 동작을 수행할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 컨텍스트 적응 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 선택스 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 선택스 엘리먼트들이 엔트로피 인코딩되지 않은 바이패스 모드에서 동작할 수도 있다.

[0158] 일부 예들에서, 엔트로피 인코딩 유닛 (220) 은 비디오 데이터, 예를 들어 변환 계수들의 하나 이상의 선택스 엘리먼트들에 대한 값들을 엔트로피 인코딩할 때 본 개시에서 논의된 라이스 파라미터 도출 기법들 중 임의의 것 또는 전부를 적용하도록 구성될 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 양자화 유닛 (208) 으로부터 변환 블록들을 수신할 수도 있다. 변환 블록에서의 각각의 계수에 대해, 엔트로피 인코딩 유닛 (220) 은 중요한 변환 계수 선택스 엘리먼트, 1 보다 큰 선택스 엘리먼트, 2 보다 큰 선택스 엘리먼트, 부호 선택스 엘리먼트, 및 나머지 값 중 하나 이상에 대한 값들을 인코딩할 수도 있다.

[0159] 다른 값들 중에서, 나머지 값 (또는 dec_abs_remainder 값) 을 엔트로피 인코딩할 때, 다른 값들 중에서, 엔트로피 인코딩 유닛 (220) 은, 도 2 와 관련하여 위에서 논의된 바와 같이, 현재 계수에 대한 이웃하는 계수들의 절대 값들의 합으로서 절대 값들의 로컬 합을 초기에 계산할 수도 있다. 엔트로피 인코딩 유닛 (220) 은, 그 후 예를 들어, 상기 식 (1) 에 따라, 현재 블록을 포함하는 슬라이스에 대한 비트 심도 및/또는 슬라이스 타입에 따라 베이스 레벨 오프셋 값을 계산할 수도 있다. 엔트로피 인코딩 유닛 (220) 은, 예를 들어 상기 식 (2) 에 따라, 베이스 레벨 오프셋 값을 사용하여 절대 값들의 로컬 합 값에 대한 정규화된 값을 계산할 수도 있다. 엔트로피 인코딩 유닛 (220) 은, 그 후 예를 들어 상기 표 1 에 따라 정규화된 값으로부터 라이스 파라미터를 계산할 수도 있다. 엔트로피 인코딩 유닛 (220) 은 그 후 라이스 파라미터를 사용하여 나머지 값에 대한 Golomb-Rice 코드를 결정하고 Golomb-Rice 코드의 각각의 비트 (빈) 를 엔트로피 인코딩할 수도 있다.

[0160] 비디오 인코더 (200) 는 화상 또는 슬라이스의 블록들을 재구성하는데 필요한 엔트로피 인코딩된 선택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 특히, 엔트로피 인코딩 유닛 (220) 은 비트스트림을 출력할 수도 있다.

[0161] AV1 에 따르면, 엔트로피 인코딩 유닛 (220) 은 심볼-대-심볼 적응적 멀티-심볼 산술 코더로서 구성될 수도 있다. AV1 에서의 선택스 엘리먼트는 N개의 엘리먼트들의 알파벳을 포함하고, 컨텍스트 (예를 들어, 확률 모델) 는 N개의 확률들의 세트를 포함한다. 엔트로피 인코딩 유닛 (220) 은 n-비트 (예를 들면, 15-비트) 누적 분포 함수 (CDF) 들로서 확률을 저장할 수 있다. 엔트로피 인코딩 유닛 (220) 은 컨텍스트를 업데이트하기 위해 알파벳 크기에 기초한 업데이트 팩터로 재귀적 스케일링을 수행할 수 있다.

[0162] 상기 설명된 동작들은 블록에 관하여 설명된다. 그러한 설명은 루마 코딩 블록 및/또는 크로마 코딩 블록들에 대한 동작들인 것으로서 이해되어야 한다. 상술한 바와 같이, 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 CU 의 루마 및 크로마 컴포넌트들이다. 일부 예들에 있어서, 루마 코딩 블록 및 크로마 코딩

블록들은 PU 의 루마 및 크로마 컴포넌트들이다.

- [0163] 일부 예들에 있어서, 루마 코딩 블록에 대해 수행된 동작들은 크로마 코딩 블록들에 대해 반복될 필요가 없다. 일 예로서, 루마 코딩 블록에 대한 모션 벡터 (MV) 및 레퍼런스 화상을 식별하기 위한 동작들이, 크로마 블록들에 대한 MV 및 레퍼런스 화상을 식별하기 위해 반복될 필요는 없다. 오히려, 루마 코딩 블록에 대한 MV 는 크로마 블록들에 대한 MV 를 결정하도록 스케일링될 수도 있으며, 레퍼런스 화상은 동일할 수도 있다. 다른 예로서, 인트라-예측 프로세스는 루마 코딩 블록 및 크로마 코딩 블록들에 대해 동일할 수도 있다.
- [0164] 이러한 방식으로, 비디오 인코더 (200) 는 비디오 데이터를 인코딩 (및 디코딩) 하기 위한 디바이스의 일예를 나타내고, 상기 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리; 및 회로부에서 구현되는 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하도록 구성된다.
- [0165] 도 6 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더 (300) 를 예시하는 블록 다이어그램이다. 도 6 은 설명의 목적들로 제공되며, 본 개시에 대체로 예시화되고 설명된 바와 같은 기법들의 한정으로서 고려되지 않아야 한다. 설명의 목적으로, 본 개시는 VVC (ITU-T H.266, 개발중) 및 HEVC (ITU-T H.265) 의 기법들에 따른 비디오 디코더 (300) 를 기술한다. 그러나, 본 개시의 기법들은 다른 비디오 코딩 표준들로 구성되는 비디오 코딩 디바이스들에 의해 수행될 수도 있다.
- [0166] 도 6 의 예에서, 비디오 디코더 (300) 는 코딩된 화상 버퍼 (CPB) 메모리 (320), 엔트로피 디코딩 유닛 (302), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 복원 유닛 (310), 필터 유닛 (312), 및 디코딩된 화상 버퍼 (DPB) (314) 를 포함한다. CPB 메모리 (320), 엔트로피 디코딩 유닛 (302), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 복원 유닛 (310), 필터 유닛 (312) 및 DPB (314) 중 임의의 것 또는 전부가 하나 이상의 프로세서에서 또는 프로세싱 회로부에서 구현될 수도 있다. 예를 들어, 비디오 디코더 (300) 의 유닛들은 하드웨어 회로부의 부분으로서, 또는 프로세서, ASIC, 또는 FPGA 의 부분으로서 하나 이상의 회로들 또는 로직 엘리먼트들로서 구현될 수도 있다. 더욱이, 비디오 디코더 (300) 는 이들 및 다른 기능들을 수행하기 위해 추가적인 또는 대안적인 프로세서들 또는 프로세싱 회로부를 포함할 수도 있다.
- [0167] 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 부가 유닛들을 포함할 수도 있다. 예들로서, 예측 프로세싱 유닛 (304) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 보상 유닛 (316) 의 부분을 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 기능 컴포넌트들을 포함할 수도 있다.
- [0168] AV1 에 따라 동작할 때, 보상 유닛 (316) 은 전술한 바와 같이 병진 모션 보상, 아핀 모션 보상, OBMC, 및/또는 복합 인트라-인트라 예측을 사용하여 비디오 데이터의 코딩 블록(예를 들어, 루마 및 크로마 코딩 블록 양자 모두)을 디코딩하도록 구성될 수 있다. 인트라 예측 유닛 (318) 은 위에서 설명한 바와같이 방향성 인트라 예측, 비방향성 인트라 예측, 채귀 필터 인트라 예측, CFL, 인트라 블록 복사(IBC), 및/또는 색상 팔레트 모드를 사용하여 비디오 데이터의 코딩 블록(예를 들어, 루마 및 크로마 코딩 블록 양자 모두)을 디코딩하도록 구성될 수 있다.
- [0169] CPB 메모리 (320) 는 비디오 디코더 (300) 의 컴포넌트들에 의해 디코딩된 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. CPB 메모리 (320) 에 저장된 비디오 데이터는, 예를 들어 컴퓨터 판독 가능 매체 (110) (도 1) 로부터 획득될 수도 있다. CPB 메모리 (320) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터 (예를 들어, 선택스 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 화상의 선택스 엘리먼트들의 비디오 데이터를 저장할 수도 있다. DPB (314) 는 일반적으로, 인코딩된 비디오 비트스트림의 후속 데이터 또는 화상들을 디코딩할 때 레퍼런스 비디오 데이터로서 비디오 디코더 (300) 가 출력 및/또는 사용할 수도 있는 디코딩된 화상들을 저장한다. CPB 메모리 (320) 및 DPB (314) 는 다양한 메모리 디바이스들, 이를 테면, SDRAM (synchronous DRAM) 을 포함하는 DRAM (Dynamic random access memory), MRAM (magnetoresistive RAM), RRAM (resistive RAM), 또는 다른 타입들의 메모리 디바이스들 중 임

의의 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 그들 컴포넌트들에 대해 오프-칩일 수도 있다.

[0170] 부가적으로 또는 대안적으로, 일부 예들에 있어서, 비디오 디코더 (300) 는 메모리 (120) (도 1) 로부터 코딩된 비디오 데이터를 취출할 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 기능의 일부 또는 전부가 비디오 디코더 (300) 의 프로세싱 회로에 의해 실행될 소프트웨어에서 구현될 때, 비디오 디코더 (300) 에 의해 실행될 명령들을 저장할 수도 있다.

[0171] 도 6 에 도시된 다양한 유닛들은 비디오 디코더 (300) 에 의해 수행되는 동작들의 이해를 돕기 위해 예시된다. 그 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 도 5 와 유사하게, 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하고, 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 회로들로 하여금 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, 유닛들 중 하나 이상은 집적 회로들일 수도 있다.

[0172] 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 예들에 있어서, 온-칩 또는 오프-칩 메모리는, 비디오 디코더 (300) 가 수신하고 실행하는 소프트웨어의 명령들 (예컨대, 오브젝트 코드) 을 저장할 수도 있다.

[0173] 엔트로피 디코딩 유닛 (302) 은 인코딩된 비디오 데이터를 CPB 로부터 수신하고, 비디오 데이터를 엔트로피 디코딩하여 신택스 엘리먼트들을 재현할 수도 있다. 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 복원 유닛 (310), 및 필터 유닛 (312) 은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0174] 일부 예들에서, 엔트로피 디코딩 유닛 (302) 은 비디오 데이터, 예를 들어 변환 계수들의 하나 이상의 신택스 엘리먼트들에 대한 값들을 엔트로피 디코딩할 때 본 개시에서 논의된 라이스 파라미터 도출 기법들 중 임의의 것 또는 전부를 적용하도록 구성될 수도 있다. 예를 들어, 엔트로피 디코딩 유닛 (302) 은 중요한 변환 계수 신택스 엘리먼트, 1 보다 큰 신택스 엘리먼트, 2 보다 큰 신택스 엘리먼트, 부호 신택스 엘리먼트, 및 나머지 값 중 하나 이상에 대한 엔트로피 인코딩된 값들을 수신할 수도 있다.

[0175] 다른 값들 중에서, 현재 계수에 대한 나머지 값 (또는 `dec_abs_remainder` 값) 을 디코딩할 때, 엔트로피 디코딩 유닛 (302) 은 현재 계수의 나머지 값에 대한 비트스트림으로부터 Golomb-Rice 코드를 엔트로피 디코딩할 수도 있다. 엔트로피 디코딩 유닛 (302) 은, 그 후 예를 들어 도 2 와 관련하여 위에서 논의된 바와 같이, 현재 계수에 대한 이웃하는 계수들의 절대 값들의 합으로서 절대 값들의 로컬 합을 계산할 수도 있다. 엔트로피 디코딩 유닛 (302) 은, 그 후 예를 들어, 상기 식 (1) 에 따라, 현재 블록을 포함하는 슬라이스에 대한 비트 심도 및/또는 슬라이스 타입에 따라 베이스 레벨 오프셋 값을 계산할 수도 있다. 엔트로피 디코딩 유닛 (302) 은, 예를 들어 상기 식 (2) 에 따라, 베이스 레벨 오프셋 값을 사용하여 절대 값들의 로컬 합 값에 대한 정규화된 값을 계산할 수도 있다. 엔트로피 디코딩 유닛 (302) 은, 그 후 예를 들어 상기 표 1 에 따라 정규화된 값으로부터 라이스 파라미터를 계산할 수도 있다. 그 후, 엔트로피 디코딩 유닛 (302) 은 라이스 파라미터를 사용하여 Golomb-Rice 코드로부터 나머지 값을 이진화할 수도 있다. 엔트로피 디코딩 유닛 (302) 은 현재 계수를 재구성하기 위해 유효 계수 값, 1 보다 큰 값, 2 보다 큰 값, 및 나머지 값을 연결할 수도 있다.

[0176] 일반적으로, 비디오 디코더 (300) 는 블록별로 화상을 재구성한다. 비디오 디코더 (300) 는 개별적으로 각각의 블록에 대해 재구성 동작을 수행할 수도 있다 (여기서 현재 재구성되고 있는, 즉 디코딩되고 있는 블록은 "현재 블록" 으로 지칭될 수도 있음).

[0177] 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 그리고 마찬가지로, 역 양자화 유닛 (306) 이 적용할 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. 역 양자화 유닛 (306) 은 이에 의해 변환 계수들을 포함하는 변환 계수 블록을 형성할 수도 있다.

[0178] 역 양자화 유닛 (306) 이 변환 계수 블록을 형성한 후, 역 변환 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 변환 계수 블록에 적용할 수도 있다.

[0179] 또한, 예측 프로세싱 유닛 (304) 은 엔트로피 디코딩 유닛 (302) 에 의해 엔트로피 디코딩된 예측 정보 선택스 엘리먼트들에 따라 예측 블록을 생성한다. 예를 들어, 예측 정보 선택스 엘리먼트들이 현재 블록이 인터-예측됨을 표시하면, 모션 보상 유닛 (316) 은 예측 블록을 생성할 수도 있다. 이 경우에, 예측 정보 선택스 엘리먼트들은 레퍼런스 블록을 추출할 DPB (314) 에서의 레퍼런스 화상 뿐만 아니라 현재 화상에서의 현재 블록의 위치에 대한 레퍼런스 화상에서의 레퍼런스 블록의 위치를 식별하는 모션 벡터를 표시할 수도 있다. 모션 보상 유닛 (316) 은 일반적으로, 모션 보상 유닛 (224) (도 5) 에 대하여 설명된 것과 실질적으로 유사한 방식으로 인터 예측 프로세스를 수행할 수도 있다.

[0180] 다른 예로서, 예측 정보 선택스 엘리먼트들이 현재 블록이 인트라 예측됨을 표시하면, 인트라 예측 유닛 (318) 은 예측 정보 선택스 엘리먼트들에 의해 표시된 인트라 예측 모드에 따라 예측 블록을 생성할 수도 있다. 다시, 인트라-예측 유닛 (318) 은 일반적으로 인트라-예측 유닛 (226) (도 5) 에 대하여 설명된 것과 실질적으로 유사한 방식으로 인트라-예측 프로세스를 수행할 수도 있다. 인트라-예측 유닛 (318) 은 DPB (314) 로부터 현재 블록에 대한 이웃 샘플들의 데이터를 추출할 수도 있다.

[0181] 재구성 유닛 (310) 은 예측 블록 및 잔차 블록을 사용하여 현재 블록을 재구성할 수도 있다. 예를 들어, 재구성 유닛 (310) 은 잔차 픽셀 블록의 샘플들을 예측 블록의 대응하는 샘플들에 가산하여 현재 블록을 재구성할 수도 있다.

[0182] 필터 유닛 (312) 은 재구성된 블록들에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (312) 은 재구성된 블록들의 에지들을 따라 블록키스 아티팩트들을 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (312) 의 동작들이 모든 예들에서 반드시 수행되는 것은 아니다.

[0183] 비디오 디코더 (300) 는 DPB (314) 에 재구성된 블록들을 저장할 수도 있다. 예를 들어, 필터 유닛 (312) 의 동작들이 수행되지 않은 예들에 있어서, 복원 유닛 (310) 이, 복원된 블록들을 DPB (314) 에 저장할 수도 있다. 필터 유닛 (312) 의 동작들이 수행되는 예들에 있어서, 필터 유닛 (312) 은, 필터링된 재구성된 블록들을 DPB (314) 에 저장할 수도 있다. 전술한 바와 같이, DPB (314) 는 예측 프로세싱 유닛 (304) 에 인트라-예측을 위한 현재 화상의 샘플들 및 후속 모션 보상을 위해 이전에 디코딩된 화상들과 같은 레퍼런스 정보를 제공할 수도 있다. 더욱이, 비디오 디코더 (300) 는 도 1 의 디스플레이 디바이스 (118) 와 같은 디스플레이 디바이스 상으로의 후속 프리젠테이션을 위해 DPB (314) 로부터 디코딩된 화상들 (예컨대, 디코딩된 비디오) 을 출력할 수도 있다.

[0184] 이러한 방식으로, 비디오 디코더 (300) 는 비디오 데이터를 디코딩하기 위한 디바이스의 일예를 나타내고, 상기 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리; 및 회로부에서 구현되는 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하도록 구성된다.

[0185] 도 7 은 본 개시의 기법들에 따라 현재 블록을 인코딩하기 위한 예시적인 방법을 도시하는 플로우차트이다. 현재 블록은 현재 CU 를 포함할 수도 있다. 비디오 인코더 (200) (도 1 및 도 5) 에 대해 설명되지만, 다른 디바이스들이 도 7 의 방법과 유사한 방법을 수행하도록 구성될 수도 있음이 이해되어야 한다.

[0186] 이 예에 있어서, 비디오 인코더 (200) 는 초기에 현재 블록을 예측한다 (350). 예를 들어, 비디오 인코더 (200) 는 현재 블록에 대한 예측 블록을 형성할 수도 있다. 비디오 인코더 (200) 는 그 후에, 현재 블록에 대한 잔차 블록을 계산할 수도 있다 (352). 잔차 블록을 계산하기 위해, 비디오 인코더 (200) 는 원래, 코

딩되지 않은 블록과 현재 블록에 대한 예측 블록 사이의 차이를 계산할 수도 있다. 그 다음, 비디오 인코더 (200)는 잔차 블록을 변환하고 잔차 블록의 변환 계수들을 양자화할 수도 있다 (354). 다음으로, 비디오 인코더 (200)는 잔차 블록의 양자화된 변환 계수들을 스캔할 수도 있다 (356). 스캔 동안, 또는 스캔에 후속하여, 비디오 인코더 (200)는 변환 계수들을 엔트로피 인코딩할 수도 있다 (358). 예를 들어, 비디오 인코더 (200)는 CAVLC 또는 CABAC를 사용하여 변환 계수들을 인코딩할 수도 있다. 비디오 인코더 (200)는 변환 계수들을 엔트로피 인코딩할 때 본 개시에서 논의된 라이스 파라미터 도출 기법들 중 임의의 것 또는 전부를 적용할 수도 있다. 그 다음, 비디오 인코더 (200)는 블록의 엔트로피 인코딩된 데이터를 출력할 수도 있다 (360).

[0187] 비디오 인코더 (200)는 또한, (예를 들어, 인터- 또는 인트라-예측 모드들에서) 후속적으로 코딩된 데이터에 대한 레퍼런스 데이터로서 현재 블록의 디코딩된 버전을 사용하기 위해, 현재 블록을 인코딩한 후에 현재 블록을 디코딩할 수도 있다. 따라서, 비디오 인코더 (200)는 계수들을 역 양자화 및 역 변환하여 잔차 블록을 재현할 수도 있다 (362). 비디오 인코더 (200)는 잔차 블록을 예측 블록과 조합하여 디코딩된 블록을 형성할 수도 있다 (364). 그 다음, 비디오 인코더 (200)는 디코딩된 블록을 DPB (218)에 저장할 수도 있다 (366).

[0188] 도 8은 본 개시의 기법들에 따른 비디오 데이터의 현재 블록을 디코딩하기 위한 예시적인 방법을 나타내는 플로우차트이다. 현재 블록은 현재 CU를 포함할 수도 있다. 비디오 인코더 (300) (도 1 및 도 6)와 관련하여 설명되었지만, 다른 디바이스들이 도 8의 방법과 유사한 방법을 수행하도록 구성될 수도 있음을 이해해야 한다.

[0189] 비디오 디코더 (300)는 현재 블록에 대응하는 잔차 블록의 변환 계수들에 대한 엔트로피 인코딩된 예측 정보 및 엔트로피 인코딩된 데이터와 같은, 현재 블록에 대한 엔트로피 인코딩된 데이터를 수신할 수도 있다 (370). 비디오 디코더 (300)는 현재 블록에 대한 예측 정보를 결정하고 잔차 블록의 변환 계수들을 재현하기 위해 엔트로피 인코딩된 데이터를 엔트로피 디코딩할 수도 있다 (372). 비디오 디코더 (300)는 변환 계수들을 엔트로피 디코딩할 때 본 개시에서 논의된 라이스 파라미터 도출 기법들 중 임의의 것 또는 전부를 적용할 수도 있다.

[0190] 비디오 디코더 (300)는 현재 블록에 대한 예측 블록을 계산하기 위해, 예를 들어, 현재 블록에 대한 예측 정보에 의해 표시된 바와 같은 인트라- 또는 인터-예측 모드를 사용하여 현재 블록을 예측할 수도 있다 (374). 그 다음, 비디오 디코더 (300)는 양자화된 변환 계수들의 블록을 생성하기 위해, 재현된 변환 계수들을 역 스케닝할 수도 있다 (376). 그 다음, 비디오 디코더 (300)는 변환 계수들을 역 양자화하고, 변환 계수들에 역 변환을 적용하여 잔차 블록을 생성할 수도 있다 (378). 비디오 디코더 (300)는 예측 블록 및 잔차 블록을 조합함으로써 종국적으로 현재 블록을 디코딩할 수도 있다 (380).

[0191] 도 9는 본 개시의 기법들에 따른 비디오 데이터를 인코딩하는 예시의 방법을 나타내는 플로우차트이다. 도 9의 방법은 일반적으로 도 7의 단계 (358)에 대응할 수도 있다.

[0192] 초기에, 비디오 인코더 (200)는 비디오 데이터의 현재 블록의 현재 계수 값 (400)을 결정할 수도 있다. 비디오 인코더 (200)는 유효 계수 선택스 엘리먼트, 1보다 큰 선택스 엘리먼트, 2보다 큰 선택스 엘리먼트, 나머지 선택스 엘리먼트, 및 부호 선택스 엘리먼트와 같은 선택스 엘리먼트들에 대한 각각의 값들로 현재 계수 값을 파티셔닝할 수도 있다. 도 9의 방법은 나머지 선택스 엘리먼트의 값의 인코딩하는 단계를 설명한다. 그러나, `dec_abs_level[]` 선택스 엘리먼트와 같은 다른 선택스 엘리먼트들은 이 또는 유사한 방법을 사용하여 인코딩될 수도 있다.

[0193] 비디오 인코더 (200)는 그 후 절대 값들의 로컬 합을 계산할 수도 있다 (402). 예를 들어, 비디오 인코더 (200)는 현재 계수에 대한 이웃 계수들의 값들에 대한 절대 값들을 결정할 수도 있다. 도 2의 예에서, 예를 들어, 이웃 계수들은, 우측 이웃 계수, 2-우측 이웃 계수, 하단-우측 이웃 계수, 하단 이웃 계수 및 2-하단-이웃 계수를 포함할 수 있다. 비디오 인코더 (200)는 이웃 계수들의 절대 값들의 총 합을 절대 값들의 로컬 합으로서 가산할 수도 있다.

[0194] 비디오 인코더 (200)는 그 후 현재 블록을 포함하는 슬라이스에 대한 비트 심도 및/또는 슬라이스 타입에 따라 베이스 레벨 오프셋 값을 계산할 수도 있다 (404). 예를 들어, 비디오 인코더 (200)는 다음에 따라 베이스 레벨 오프셋 값 (`baseLevelOffset`)을 계산할 수도 있다:

[0195]
$$\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1 : P2) : (\text{isIntra}() ? P3 : P4).$$

- [0196] 따라서, 비디오 인코더 (200) 는 비디오 데이터에 대한 비트 심도 (bitDepth) 가 임계치 값 (T) 보다 큰지 여부를 결정할 수도 있다. 비트 심도가 임계치 값보다 크면, 비디오 인코더 (200) 는 슬라이스 타입이 인트라-예측인지 여부를 결정할 수도 있다. 슬라이스 타입이 인트라 예측이면, 비디오 인코더 (200) 는 베이스 레벨 오프셋 값이 P1 이라고 결정할 수도 있다. 슬라이스 타입이 인트라-예측이 아니면 (예를 들어, P-슬라이스 또는 B-슬라이스와 같은 인터-예측이면), 비디오 인코더 (200) 는 베이스 레벨 오프셋 값이 P2 라고 결정할 수도 있다. 비트 심도가 임계치 값 이하이면, 비디오 인코더 (200) 는 슬라이스 타입이 인트라-예측인지 여부를 결정할 수도 있다. 슬라이스 타입이 인트라 예측이면, 비디오 인코더 (200) 는 베이스 레벨 오프셋 값이 P3 이라고 결정할 수도 있다. 슬라이스 타입이 인트라-예측이 아니면 (예를 들어, P-슬라이스 또는 B-슬라이스와 같은 인터-예측이면), 비디오 인코더 (200) 는 베이스 레벨 오프셋 값이 P4 라고 결정할 수도 있다. P1, P2, P3, 및 P4 는 미리 정해진 값들일 수도 있거나, 또는 비디오 인코더 (200) 는 P1, P2, P3, 및 P4 에 대한 값들을 결정하고 비트스트림에서 결정된 값들을 시그널링할 수도 있다.
- [0197] 비디오 인코더 (200) 는 그 후 계산된 베이스 레벨 오프셋 값을 사용하여 절대 값들의 로컬 합을 정규화할 수도 있다 (406). 예를 들어, 비디오 인코더 (200) 는 다음을 계산할 수도 있다:
- [0198]
$$\text{locSumAbs} = \text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset}).$$
- [0199] 이 예에서, locSumAbs (절대 값들의 로컬 합) 는, 예를 들어 VVC 에 대한 클리핑 연산 Clip3 을 이용하여, 그리고 계산된 baseLevelOffset 값을 이용하여 업데이트된다.
- [0200] 비디오 인코더 (200) 는 그 후 정규화된 값으로부터 라이스 파라미터를 결정할 수도 있다 (408). 예를 들어, 비디오 인코더 (200) 는 룩업 테이블로서 표 1 을 사용하여 라이스 파라미터를 결정할 수도 있다.
- [0201] 비디오 인코더 (200) 는 그 후 라이스 파라미터를 사용하여 계수 값 (예를 들어, 나머지 값) 에 대한 Golomb-Rice 코드를 결정할 수도 있다 (410). Golomb-Rice 코드는 일련의 비트들 (빈들) 일 수 있다. 비디오 인코더 (200) 는 그 후 Golomb-Rice 코드를 엔트로피 인코딩할 수도 있다 (412).
- [0202] 이러한 방식으로, 도 9 의 방법은 비디오 데이터를 인코딩하는 방법의 일예를 나타내고, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하는 단계; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계를 포함한다.
- [0203] 도 10 은 본 개시의 기법들에 따른 비디오 데이터를 디코딩하는 예시의 방법을 나타내는 플로우차트이다. 도 10 의 방법은 일반적으로 도 8 의 단계 (372) 에 대응할 수도 있다.
- [0204] 초기에, 비디오 디코더 (300) 는 비디오 데이터의 현재 블록의 현재 계수에 대한 Golomb-Rice 코드를 엔트로피 디코딩할 수도 있다 (420). Golomb-Rice 코드는 현재 계수의 잔차 값에 대응할 수도 있고, 현재 계수는 유효 계수 선택스 엘리먼트, 1 보다 큰 선택스 엘리먼트, 2 보다 큰 선택스 엘리먼트, 및 부호 선택스 엘리먼트를 더 포함할 수도 있다. 도 10 의 예는 현재 계수에 대한 나머지 값을 디코딩하는 것에 관한 것이지만, 도 10 의 방법은 또한 dec_abs_level[] 선택스 엘리먼트에 대한 값과 같은 다른 값들을 디코딩하는데 사용될 수도 있다.
- [0205] 비디오 디코더 (300) 는 그 후 절대 값들의 로컬 합을 계산할 수도 있다 (422). 예를 들어, 비디오 디코더 (300) 는 현재 계수에 대한 이웃 계수들의 값들에 대한 절대 값들을 결정할 수도 있다. 도 2 의 예에서, 예를 들어, 이웃 계수들은, 우측 이웃 계수, 2-우측 이웃 계수, 하단-우측 이웃 계수, 하단 이웃 계수 및 2-하단-이웃 계수를 포함할 수 있다. 비디오 디코더 (300) 는 이웃 계수들의 절대 값들의 총 합을 절대 값들의 로컬 합으로서 가산할 수도 있다.
- [0206] 비디오 디코더 (300) 는 그 후 현재 블록을 포함하는 슬라이스에 대한 비트 심도 및/또는 슬라이스 타입에 따라 베이스 레벨 오프셋 값을 계산할 수도 있다 (424). 예를 들어, 비디오 디코더 (300) 는 다음에 따라 베이스 레벨 오프셋 값 (baseLevelOffset) 을 계산할 수도 있다:
- [0207]
$$\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1: P2) : (\text{isIntra}() ? P3: P4).$$
- [0208] 따라서, 비디오 디코더 (300) 는 비디오 데이터에 대한 비트 심도 (bitDepth) 가 임계치 값 (T) 보다 큰지 여부를 결정할 수도 있다. 비트 심도가 임계치 값보다 크면, 비디오 디코더 (300) 는 슬라이스 타입이 인트라-예측인지 여부를 결정할 수도 있다. 슬라이스 타입이 인트라 예측이면, 비디오 디코더 (300) 는 베이스 레

벨 오프셋 값이 P1 이라고 결정할 수도 있다. 슬라이스 타입이 인트라-예측이 아니면 (예를 들어, P-슬라이스 또는 B-슬라이스와 같은 인터-예측이면), 비디오 디코더 (300) 는 베이스 레벨 오프셋 값이 P2 라고 결정할 수도 있다. 비트 심도가 임계치 값 이하이면, 비디오 디코더 (300) 는 슬라이스 타입이 인트라-예측인지 여부를 결정할 수도 있다. 슬라이스 타입이 인트라 예측이면, 비디오 디코더 (300) 는 베이스 레벨 오프셋 값이 P3 이라고 결정할 수도 있다. 슬라이스 타입이 인트라-예측이 아니면 (예를 들어, P-슬라이스 또는 B-슬라이스와 같은 인터-예측이면), 비디오 디코더 (300) 는 베이스 레벨 오프셋 값이 P4 라고 결정할 수도 있다.

P1, P2, P3, 및 P4 는 미리 정해진 값들일 수도 있거나, 또는 비디오 디코더 (300) 는 비트스트림으로부터 P1, P2, P3, 및 P4 에 대한 값들을 디코딩할 수도 있다.

[0209] 비디오 디코더 (300) 는 그 후 계산된 베이스 레벨 오프셋 값을 사용하여 절대 값들의 로컬 합을 정규화할 수도 있다 (426). 예를 들어, 비디오 디코더 (300) 는 다음을 계산할 수도 있다:

[0210] $\text{locSumAbs} = \text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$.

[0211] 이 예에서, locSumAbs (절대 값들의 로컬 합) 는, 예를 들어 VVC 에 대한 클리핑 연산 Clip3 을 이용하여, 그리고 계산된 baseLevelOffset 값을 이용하여 업데이트된다.

[0212] 비디오 디코더 (300) 는 그 후 정규화된 값으로부터 라이스 파라미터를 결정할 수도 있다 (428). 예를 들어, 비디오 디코더 (300) 는 룩업 테이블로서 표 1 을 사용하여 라이스 파라미터를 결정할 수도 있다.

[0213] 비디오 디코더 (300) 는 그 후 라이스 파라미터를 사용하여 Golomb-Rice 코드로부터 나머지 값을 나타내는 이진화된 (이진) 값을 형성할 수도 있다 (430). 비디오 디코더 (300) 는 또한 이진화된 값으로부터 계수 값을 재구성할 수도 있다 (432). 예를 들어, 비디오 디코더 (300) 는 유효 계수 선택스 엘리먼트, 1 보다 큰 선택스 엘리먼트, 2 보다 큰 선택스 엘리먼트, 및 나머지 값에 대한 값들을 연결할 수도 있다.

[0214] 이러한 방식으로, 도 9 의 방법은 비디오 데이터를 디코딩하는 방법의 일예를 나타내고, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하는 단계; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계를 포함한다.

[0215] 본 개시의 다양한 기법들의 특정 예들이 다음의 조항들에서 요약된다:

[0216] 조항 1: 비디오 데이터를 코딩하는 방법으로서, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하는 단계; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계를 포함한다.

[0217] 조항 2: 조항 1 의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함한다.

[0218] 조항 3: 조항 1 및 조항 2 중 임의의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수를 포함한다.

[0219] 조항 4: 조항 1 내지 조항 3 중 임의의 방법에서, 상기 베이스 레벨 오프셋 값을 계산하는 단계는 $\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1 : P2) : (\text{isIntra}() ? P3 : P4)$ 를 계산하는 단계를 포함하고, bitDepth 는 비디오 데이터에 대한 비트 심도를 포함하고, isIntra() 는 슬라이스에 대한 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 슬라이스에 대한 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키고, T, P1, P2, P3, 및 P4 는 각각의 입력 값들을 포함한다.

[0220] 조항 5: 조항 4 의 방법에서, T, P1, P2, P3, 및 P4 에 대한 입력 값들은 미리 결정된다.

[0221] 조항 6: 조항 4 의 방법에서, T, P1, P2, P3, 및 P4 에 대한 입력 값들은 비트스트림에서 시그널링된다.

[0222] 조항 7: 조항 4 내지 조항 6 중 임의의 방법에서, T = 12, P1 = 5, P2 = 10, P3 = 6, P4 = 20 이다.

[0223] 조항 8: 조항 4 내지 조항 6 중 임의의 방법에서, T = 12, P1 = 5, P2 = 10, P3 = 10, P4 = 20 이다.

- [0224] 조항 9: 조항 1 내지 조항 8 중 임의의 방법에서, 절대 값들의 합에 대한 정규화된 값을 계산하는 단계는 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하는 단계를 포함하고, locSumAbs 는 절대 값들의 합을 포함하고 baseLevelOffset 은 베이스 레벨 오프셋 값을 포함한다.
- [0225] 조항 10: 비디오 데이터를 코딩하는 방법으로서, 비디오 데이터의 컬러 컴포넌트의 계수들을 코딩하는데 이용되는 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하는 단계; 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 히스토리 카운터에 대한 현재 값으로부터 비디오 데이터의 컬러 컴포넌트의 블록에 대한 라이스 파라미터 스케일링 값을 도출하는 단계; 블록의 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 라이스 파라미터 스케일링 값을 이용하여 절대 값들의 합에 대한 스케일링된 값을 계산하는 단계; 및 절대 값들의 합에 대한 스케일링된 값을 이용하여 현재 계수를 코딩하는 단계를 포함한다.
- [0226] 조항 11: 방법은 조항 1 내지 조항 9 중 임의의 방법 및 조항 10 의 방법의 조합을 포함한다.
- [0227] 조항 12: 조항 10 및 조항 11 중 임의의 방법에서, 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoeff}[\text{idx}] \geq T) ? P2 : P1$ 를 계산하는 단계를 포함하고, idx 는 컬러 컴포넌트를 나타내고, $\text{StatCoeff}[\text{idx}]$ 는 히스토리 카운터에 대한 현재 값을 나타내며, T , $P1$, 및 $P2$ 는 입력 값들을 나타낸다.
- [0228] 조항 13: 조항 12 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 미리 결정된다.
- [0229] 조항 14: 조항 12 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0230] 조항 15: 조항 12 내지 조항 14 중 임의의 방법에서, $T = 4$, $P2 = 2$, 및 $P1 = 1$ 이다.
- [0231] 조항 16: 조항 10 및 조항 11 중 임의의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는, 상기 비디오 데이터의, 상기 컬러 컴포넌트를 포함하는 모든 컬러 컴포넌트들에 대한, 상기 히스토리 카운터를 포함하는 히스토리 카운터들의 집성을 나타내는 집성된 히스토리 카운터 값을 계산하는 단계; 및 상기 집성된 히스토리 카운터 값을 이용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함한다.
- [0232] 조항 17: 조항 16 의 방법에서, 상기 집성된 히스토리 카운터 값을 계산하는 단계는 $((\text{StatCoeff}[0] * w0) + \text{StatCoeff}[1]*w1 + \text{StatCoeff}[2]*w2)/(w1+w2+w3)$ 를 계산하는 단계를 포함하고, $\text{StatCoeff}[0]$ 는 컬러 컴포넌트들의 제 1 컬러 컴포넌트에 대한 히스토리 카운터들의 제 1 히스토리 카운터를 나타내고, $\text{StatCoeff}[1]$ 는 컬러 컴포넌트들의 제 2 컬러 컴포넌트에 대한 히스토리 카운터들의 제 2 히스토리 카운터를 나타내며, $\text{StatCoeff}[2]$ 는 컬러 컴포넌트들의 제 3 컬러 컴포넌트에 대한 히스토리 카운터들의 제 3 히스토리 카운터를 나타내고, $w1$, $w2$, 및 $w3$ 은 각각의 가중치 값들을 나타낸다.
- [0233] 조항 18: 조항 17 의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoefAverage} \geq T) ? P2 : P1$ 를 사용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함하고, StatCoefAverage 는 집성된 히스토리 카운터 값을 나타내고, T , $P1$ 및 $P2$ 는 입력 값들을 나타낸다.
- [0234] 조항 19: 조항 18 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 미리 결정된다.
- [0235] 조항 20: 조항 18 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0236] 조항 21: 조항 18 내지 조항 20 중 임의의 방법에서, $T = 4$, $P2 = 2$, 및 $P1 = 1$ 이다.
- [0237] 조항 22: 조항 18 내지 조항 21 중 임의의 방법에서, $w0 = 2$, $w1 = 1$, 및 $w2 = 1$ 이다.
- [0238] 조항 23: 조항 10 내지 조항 22 중 임의의 방법에서, 상기 블록은 코딩 트리 유닛 (CTU) 을 포함한다.
- [0239] 조항 24: 조항 10 내지 조항 22 중 임의의 방법에서, 상기 블록은 변환 유닛 (TU) 을 포함한다.
- [0240] 조항 25: 조항 10 내지 조항 22 중 임의의 방법에서, 상기 블록은 서브-변환 유닛 (서브-TU) 을 포함한다.
- [0241] 조항 26: 조항 10 내지 조항 22 중 임의의 방법에서, 상기 블록은 제 1 블록을 포함하고, 상기 라이스 파라미터 스케일링 값은 제 1 라이스 파라미터 스케일링 값을 포함하며, 방법은 제 1 라이스 파라미터 스케일링 값을 이용하여 제 2 블록에 대한 제 2 라이스 파라미터 스케일링 값을 도출하는 단계를 더 포함한다.
- [0242] 조항 27: 비디오 데이터를 코딩하는 방법으로서, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계; 베이스 레벨

오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하는 단계; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계를 포함한다.

- [0243] 조항 28: 조항 27 의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함한다.
- [0244] 조항 29: 조항 27 중 임의의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수를 포함한다.
- [0245] 조항 30: 조항 27 의 방법에서, 상기 베이스 레벨 오프셋 값을 계산하는 단계는 $\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1 : P2) : (\text{isIntra}() ? P3 : P4)$ 를 계산하는 단계를 포함하고, bitDepth 는 비디오 데이터에 대한 비트 심도를 포함하고, $\text{isIntra}()$ 는 슬라이스에 대한 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 슬라이스에 대한 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키고, T , $P1$, $P2$, $P3$, 및 $P4$ 는 각각의 입력 값들을 포함한다.
- [0246] 조항 31: 조항 30 의 방법에서, T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 미리 결정된다.
- [0247] 조항 32: 조항 30 의 방법에서, T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 비트스트림에서 시그널링된다.
- [0248] 조항 33: 조항 30 의 방법에서, $T = 12$, $P1 = 5$, $P2 = 10$, $P3 = 6$, $P4 = 20$ 이다.
- [0249] 조항 34: 조항 30 의 방법에서, $T = 12$, $P1 = 5$, $P2 = 10$, $P3 = 10$, $P4 = 20$ 이다.
- [0250] 조항 35: 조항 27 의 방법에서, 절대 값들의 합에 대한 정규화된 값을 계산하는 단계는 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하는 단계를 포함하고, locSumAbs 는 절대 값들의 합을 포함하고 baseLevelOffset 은 베이스 레벨 오프셋 값을 포함한다.
- [0251] 조항 36: 비디오 데이터를 코딩하는 방법으로서, 비디오 데이터의 컬러 컴포넌트의 계수들을 코딩하는데 이용되는 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하는 단계; 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 히스토리 카운터에 대한 현재 값으로부터 비디오 데이터의 컬러 컴포넌트의 블록에 대한 라이스 파라미터 스케일링 값을 도출하는 단계; 블록의 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 라이스 파라미터 스케일링 값을 이용하여 절대 값들의 합에 대한 스케일링된 값을 계산하는 단계; 및 절대 값들의 합에 대한 스케일링된 값을 이용하여 현재 계수를 코딩하는 단계를 포함한다.
- [0252] 조항 37: 조항 36 의 방법에서, 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoeff}[\text{idx}] \geq T) ? P2 : P1$ 를 계산하는 단계를 포함하고, idx 는 컬러 컴포넌트를 나타내고, $\text{StatCoeff}[\text{idx}]$ 는 히스토리 카운터에 대한 현재 값을 나타내며, T , $P1$, 및 $P2$ 는 입력 값들을 나타낸다.
- [0253] 조항 38: 조항 37 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 미리 결정된다.
- [0254] 조항 39: 조항 37 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0255] 조항 40: 조항 37 의 방법에서, $T = 4$, $P2 = 2$, 및 $P1 = 1$ 이다.
- [0256] 조항 41: 조항 36 의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는, 상기 비디오 데이터의, 상기 컬러 컴포넌트를 포함하는 모든 컬러 컴포넌트들에 대한, 상기 히스토리 카운터를 포함하는 히스토리 카운터들의 집성을 나타내는 집성된 히스토리 카운터 값을 계산하는 단계; 및 상기 집성된 히스토리 카운터 값을 이용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함한다.
- [0257] 조항 42: 조항 41 의 방법에서, 상기 집성된 히스토리 카운터 값을 계산하는 단계는 $((\text{StatCoeff}[0] * w0) + \text{StatCoeff}[1] * w1 + \text{StatCoeff}[2] * w2) / (w1 + w2 + w3)$ 를 계산하는 단계를 포함하고, $\text{StatCoeff}[0]$ 는 컬러 컴포넌트들의 제 1 컬러 컴포넌트에 대한 히스토리 카운터들의 제 1 히스토리 카운터를 나타내고, $\text{StatCoeff}[1]$ 는 컬러 컴포넌트들의 제 2 컬러 컴포넌트에 대한 히스토리 카운터들의 제 2 히스토리 카운터를 나타내며, $\text{StatCoeff}[2]$ 는 컬러 컴포넌트들의 제 3 컬러 컴포넌트에 대한 히스토리 카운터들의 제 3 히스토리 카운터를 나타내고, $w1$, $w2$, 및 $w3$ 은 각각의 가중치 값들을 나타낸다.
- [0258] 조항 43: 조항 42 의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoefAverage} \geq T) ? P2 : P1$ 를 사용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함하고, StatCoefAverage 는 집

성된 히스토리 카운터 값을 나타내고, T, P1 및 P2 는 입력 값들을 나타낸다.

- [0259] 조항 44: 조항 43 의 방법에서, T, P1, 및 P2 에 대한 입력 값들은 미리 결정된다.
- [0260] 조항 45: 조항 43 의 방법에서, T, P1, 및 P2 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0261] 조항 46: 조항 43 의 방법에서, $T = 4$, $P2 = 2$, 및 $P1 = 1$ 이다.
- [0262] 조항 47: 조항 43 의 방법에서, $w0 = 2$, $w1 = 1$, 및 $w2 = 1$ 이다.
- [0263] 조항 48: 조항 36 의 방법에서, 상기 블록은 코딩 트리 유닛 (CTU) 을 포함한다.
- [0264] 조항 49: 조항 36 의 방법에서, 상기 블록은 변환 유닛 (TU) 을 포함한다.
- [0265] 조항 50: 조항 36 의 방법에서, 상기 블록은 서브-변환 유닛 (서브-TU) 을 포함한다.
- [0266] 조항 51: 조항 36 의 방법에서, 상기 블록은 제 1 블록을 포함하고, 상기 라이스 파라미터 스케일링 값은 제 1 라이스 파라미터 스케일링 값을 포함하며, 방법은 제 1 라이스 파라미터 스케일링 값을 이용하여 제 2 블록에 대한 제 2 라이스 파라미터 스케일링 값을 도출하는 단계를 더 포함한다.
- [0267] 조항 52: 조항 1 내지 조항 51 중 임의의 방법에서, 코딩하는 단계는 디코딩하는 단계를 포함한다.
- [0268] 조항 53: 조항 1 내지 조항 52 중 임의의 방법에서, 코딩하는 단계는 인코딩하는 단계를 포함한다.
- [0269] 조항 54: 비디오 데이터를 코딩하기 위한 디바이스로서, 디바이스는 조항 1 내지 조항 53 중 임의의 방법을 수행하기 위한 하나 이상의 수단을 포함한다.
- [0270] 조항 55: 조항 54 의 디바이스에서, 하나 이상의 수단은 회로부에서 구현된 하나 이상의 프로세서들을 포함한다.
- [0271] 조항 56: 조항 54 내지 조항 55 중 임의의 디바이스에서, 디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함한다.
- [0272] 조항 57: 조항 54 내지 조항 56 중 임의의 디바이스에서, 디바이스는 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스, 또는 셋탑 박스 중 하나 이상을 포함한다.
- [0273] 조항 58: 조항 54 내지 조항 57 의 디바이스에서, 비디오 데이터를 저장하도록 구성된 메모리를 더 포함한다.
- [0274] 조항 59: 명령들을 저장한 컴퓨터 판독가능 저장 매체로서, 명령들은, 실행될 때, 비디오 데이터를 디코딩하기 위한 디바이스의 프로세서로 하여금 조항 1 내지 조항 53 중 임의의 방법을 수행하게 한다.
- [0275] 조항 60: 비디오 데이터를 코딩하기 위한 디바이스로서, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하기 위한 수단; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하기 위한 수단; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하기 위한 수단; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하기 위한 수단을 포함한다.
- [0276] 조항 61: 비디오 데이터를 코딩하기 위한 디바이스로서, 비디오 데이터의 컬러 컴포넌트의 계수들을 코딩하는데 이용되는 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하기 위한 수단; 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 히스토리 카운터에 대한 현재 값으로부터 비디오 데이터의 컬러 컴포넌트의 블록에 대한 라이스 파라미터 스케일링 값을 도출하기 위한 수단; 블록의 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하기 위한 수단; 라이스 파라미터 스케일링 값을 이용하여 절대 값들의 합에 대한 스케일링된 값을 계산하기 위한 수단; 및 절대 값들의 합에 대한 스케일링된 값을 이용하여 현재 계수를 코딩하기 위한 수단을 포함한다.
- [0277] 조항 62: 비디오 데이터를 코딩하는 방법으로서, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하는 단계; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계를 포함한다.
- [0278] 조항 63: 조항 62 의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함

한다.

- [0279] 조항 64: 조항 62 의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수를 포함한다.
- [0280] 조항 65: 조항 62 의 방법에서, 상기 베이스 레벨 오프셋 값을 계산하는 단계는 $\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1 : P2) : (\text{isIntra}() ? P3 : P4)$ 를 계산하는 단계를 포함하고, bitDepth 는 비디오 데이터에 대한 비트 심도를 포함하고, $\text{isIntra}()$ 는 슬라이스에 대한 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 슬라이스에 대한 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키고, T , $P1$, $P2$, $P3$, 및 $P4$ 는 각각의 입력 값들을 포함한다.
- [0281] 조항 66: 조항 65 의 방법에서, $T = 12$ 이다.
- [0282] 조항 67: 조항 66 의 방법에서, $P1 = 5$, $P2 = 10$, $P3 = 6$, 및 $P4 = 20$ 이다.
- [0283] 조항 68: 조항 66 의 방법에서, $P1 = 5$, $P2 = 10$, $P3 = 10$, 및 $P4 = 20$ 이다.
- [0284] 조항 69: 조항 65 의 방법에서, T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 미리 결정된다.
- [0285] 조항 70: 조항 65 의 방법에서, T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 비트스트림에서 시그널링된다.
- [0286] 조항 71: 조항 62 의 방법에서, 절대 값들의 합에 대한 정규화된 값을 계산하는 단계는 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하는 단계를 포함하고, locSumAbs 는 절대 값들의 합을 포함하고 baseLevelOffset 은 베이스 레벨 오프셋 값을 포함한다.
- [0287] 조항 72: 조항 62 의 방법에서, 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계는, 절대 값들의 합에 대한 정규화된 값에 대응하는 라이스 파라미터를 결정하는 단계, 및 라이스 파라미터를 사용하여 현재 계수에 대한 값을 이진화하여 이진화된 값을 형성하는 단계를 포함한다.
- [0288] 조항 73: 조항 62 의 방법에서, 상기 현재 블록은 제 1 블록을 포함하고, 상기 비디오 데이터의 컬러 컴포넌트는 상기 제 1 블록을 포함하며, 상기 절대 값들의 합은 절대 값들의 제 1 합을 포함하고, 상기 복수의 이웃 계수들은 제 1 복수의 이웃 계수들을 포함하며, 상기 방법은, 상기 비디오 데이터의 상기 컬러 컴포넌트의 계수들을 코딩하는데 사용된 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하는 단계; 상기 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 상기 히스토리 카운터에 대한 현재 값으로부터 상기 비디오 데이터의 상기 컬러 컴포넌트의 제 2 블록에 대한 라이스 파라미터 스케일링 값을 도출하는 단계; 제 2 복수의 이웃 계수들의 절대 값들의 제 2 합을 상기 제 2 블록의 제 2 계수에 대해 계산하는 단계; 상기 라이스 파라미터 스케일링 값을 사용하여 상기 절대 값들의 제 2 합에 대한 스케일링된 값을 계산하는 단계; 및 상기 절대 값들의 제 2 합에 대한 상기 스케일링된 값을 사용하여 상기 제 2 블록의 상기 제 2 계수를 코딩하는 단계를 더 포함한다.
- [0289] 조항 74: 조항 12 의 방법에서, 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoeff}[\text{idx}] \geq T) ? P2 : P1$ 를 계산하는 단계를 포함하고, idx 는 컬러 컴포넌트를 나타내고, $\text{StatCoeff}[\text{idx}]$ 는 히스토리 카운터에 대한 현재 값을 나타내며, T , $P1$, 및 $P2$ 는 입력 값들을 나타낸다.
- [0290] 조항 75: 조항 13 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 미리 결정된다.
- [0291] 조항 76: 조항 13 의 방법에서, T , $P1$, 및 $P2$ 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0292] 조항 77: 조항 13 의 방법에서, $T = 4$, $P2 = 2$, 및 $P1 = 1$ 이다.
- [0293] 조항 78: 조항 12 의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는, 상기 비디오 데이터의, 상기 컬러 컴포넌트를 포함하는 모든 컬러 컴포넌트들에 대한, 상기 히스토리 카운터를 포함하는 히스토리 카운터들의 집성을 나타내는 집성된 히스토리 카운터 값을 계산하는 단계; 및 상기 집성된 히스토리 카운터 값을 이용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함한다.
- [0294] 조항 79: 조항 17 의 방법에서, 상기 집성된 히스토리 카운터 값을 계산하는 단계는 $((\text{StatCoeff}[0] * w0) + \text{StatCoeff}[1] * w1 + \text{StatCoeff}[2] * w2) / (w1 + w2 + w3)$ 를 계산하는 단계를 포함하고, $\text{StatCoeff}[0]$ 는 컬러 컴포넌트들의 제 1 컬러 컴포넌트에 대한 히스토리 카운터들의 제 1 히스토리 카운터를 나타내고, $\text{StatCoeff}[1]$ 는 컬러 컴포넌트들의 제 2 컬러 컴포넌트에 대한 히스토리 카운터들의 제 2 히스토리 카운터를 나타내며,

StatCoeff[2] 는 컬러 컴포넌트들의 제 3 컬러 컴포넌트에 대한 히스토리 카운터들의 제 3 히스토리 카운터를 나타내고, w_1 , w_2 , 및 w_3 은 각각의 가중치 값들을 나타낸다.

- [0295] 조항 80: 조항 18 의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는 ($\text{StatCoefAverage} \geq T$) ? $P_2 : P_1$ 를 사용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함하고, StatCoefAverage 는 집계된 히스토리 카운터 값을 나타내고, T , P_1 및 P_2 는 입력 값들을 나타낸다.
- [0296] 조항 81: 조항 19 의 방법에서, T , P_1 , 및 P_2 에 대한 입력 값들은 미리 결정된다.
- [0297] 조항 82: 조항 19 의 방법에서, T , P_1 , 및 P_2 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0298] 조항 83: 조항 19 의 방법에서, $T = 4$, $P_2 = 2$, 및 $P_1 = 1$ 이다.
- [0299] 조항 84: 조항 19 의 방법에서, $w_0 = 2$, $w_1 = 1$, 및 $w_2 = 1$ 이다.
- [0300] 조항 85: 조항 12 의 방법에서, 상기 제 2 블록은 코딩 트리 유닛 (CTU), 변환 유닛 (TU), 또는 서브-변환 유닛 (서브-TU) 중 하나를 포함한다.
- [0301] 조항 86: 조항 62 의 방법에서, 상기 현재 계수를 코딩하기 위해, 상기 하나 이상의 프로세서들은 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 디코딩하도록 구성된다.
- [0302] 조항 87: 조항 62 의 방법에서, 상기 현재 계수를 코딩하기 위해, 상기 하나 이상의 프로세서들은 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 인코딩하도록 구성된다.
- [0303] 조항 88: 비디오 데이터를 코딩하기 위한 디바이스로서, 비디오 데이터를 저장하도록 구성된 메모리; 및 회로부에서 구현되는 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하도록 구성된다.
- [0304] 조항 89: 조항 27 의 디바이스에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함한다.
- [0305] 조항 90: 조항 27 중 디바이스에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수들을 포함한다.
- [0306] 조항 91: 조항 27 의 디바이스에서, 상기 베이스 레벨 오프셋 값을 계산하기 위해, 상기 하나 이상의 프로세서들은, $\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P_1 : P_2) : (\text{isIntra}() ? P_3 : P_4)$ 를 계산하도록 구성되고, bitDepth 는 비디오 데이터에 대한 비트 심도를 포함하고, $\text{isIntra}()$ 는 슬라이스에 대한 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 슬라이스에 대한 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키고, T , P_1 , P_2 , P_3 , 및 P_4 는 각각의 입력 값들을 포함한다.
- [0307] 조항 92: 조항 30 의 디바이스에서, $T = 12$ 이다.
- [0308] 조항 93: 조항 30 의 디바이스에서, T , P_1 , P_2 , P_3 , 및 P_4 에 대한 입력 값들은 미리 결정된다.
- [0309] 조항 94: 조항 27 의 디바이스에서, 절대 값들의 합에 대한 정규화된 값을 계산하기 위해서, 하나 이상의 프로세서들은 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하도록 구성되고, locSumAbs 는 절대 값들의 합을 포함하고 baseLevelOffset 은 베이스 레벨 오프셋 값을 포함한다.
- [0310] 조항 95: 조항 27 의 디바이스에서, 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하기 위해, 하나 이상의 프로세서들은, 절대 값들의 합에 대한 정규화된 값에 대응하는 라이스 파라미터를 결정하고, 그리고 라이스 파라미터를 사용하여 현재 계수에 대한 값을 이진화하여 이진화된 값을 형성하도록 구성된다.
- [0311] 조항 96: 조항 27 의 디바이스에서, 상기 현재 블록은 제 1 블록을 포함하고, 상기 비디오 데이터의 컬러 컴포넌트는 상기 제 1 블록을 포함하며, 상기 절대 값들의 합은 절대 값들의 제 1 합을 포함하고, 상기 복수의 이웃 계수들은 제 1 복수의 이웃 계수들을 포함하며, 상기 하나 이상의 프로세서들은, 상기 비디오 데이터의 상기 컬러 컴포넌트의 계수들을 코딩하는데 사용된 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하고; 상기 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 상기 히스토리 카운터에 대한

현재 값으로부터 상기 비디오 데이터의 상기 컬러 컴포넌트의 제 2 블록에 대한 라이스 파라미터 스케일링 값을 도출하며; 제 2 복수의 이웃 계수들의 절대 값들의 제 2 합을 상기 제 2 블록의 제 2 계수에 대해 계산하고; 상기 라이스 파라미터 스케일링 값을 사용하여 상기 절대 값들의 제 2 합에 대한 스케일링된 값을 계산하며; 그리고 상기 절대 값들의 제 2 합에 대한 상기 스케일링된 값을 사용하여 상기 제 2 블록의 상기 제 2 계수를 코딩하도록 추가로 구성된다.

- [0312] 조항 97: 조항 35 의 디바이스에서, 라이스 파라미터 스케일링 값을 도출하기 위해, 하나 이상의 프로세서들은, $(StatCoeff[idx] \geq T) ? P2 : P1$ 를 계산하도록 구성되고, idx 는 컬러 컴포넌트를 나타내고, $StatCoeff[idx]$ 는 히스토리 카운터에 대한 현재 값을 나타내며, T , $P1$, 및 $P2$ 는 입력 값들을 나타낸다.
- [0313] 조항 98: 조항 27 의 디바이스에서, 현재 계수를 코딩하기 위해, 하나 이상의 프로세서들은, 절대 값들의 합에 대한 정규화된 값을 사용하여 상기 현재 계수를 인코딩하는 것 또는 절대 값들의 합에 대한 정규화된 값을 사용하여 상기 현재 계수를 디코딩하는 것 중 적어도 하나를 수행하도록 구성된다.
- [0314] 조항 99: 조항 27 의 디바이스에서, 디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함한다.
- [0315] 조항 100: 조항 27 의 디바이스에서, 상기 디바이스는 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스, 또는 셋-톱 박스 중 하나 이상을 포함한다.
- [0316] 조항 101: 컴퓨터 판독가능 저장 매체로서 명령들이 저장되고, 상기 명령들은, 실행될 때, 비디오 데이터를 코딩하기 위한 디바이스의 프로세서로 하여금: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하게 하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하게 하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하게 하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하게 한다.
- [0317] 조항 102: 비디오 데이터를 코딩하기 위한 디바이스로서, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하기 위한 수단; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하기 위한 수단; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하기 위한 수단; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하기 위한 수단을 포함한다.
- [0318] 조항 103: 비디오 데이터를 코딩하는 방법으로서, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하는 단계; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하는 단계; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하는 단계; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계를 포함한다.
- [0319] 조항 104: 조항 103 의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함한다.
- [0320] 조항 105: 조항 103 및 조항 104 중 임의의 방법에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수를 포함한다.
- [0321] 조항 106: 조항 103 내지 조항 105 중 임의의 방법에서, 상기 베이스 레벨 오프셋 값을 계산하는 단계는 $baseLevelOffset = (bitDepth > T) ? (isIntra() ? P1 : P2) : (isIntra() ? P3 : P4)$ 를 계산하는 단계를 포함하고, $bitDepth$ 는 비디오 데이터에 대한 비트 심도를 포함하고, $isIntra()$ 는 슬라이스에 대한 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 슬라이스에 대한 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키고, T , $P1$, $P2$, $P3$, 및 $P4$ 는 각각의 입력 값들을 포함한다.
- [0322] 조항 107: 조항 106 의 방법에서, $T = 12$ 이다.
- [0323] 조항 108: 조항 107 의 방법에서, $P1 = 5$, $P2 = 10$, $P3 = 6$, 및 $P4 = 20$ 이다.
- [0324] 조항 109: 조항 107 의 방법에서, $P1 = 5$, $P2 = 10$, $P3 = 10$, 및 $P4 = 20$ 이다.
- [0325] 조항 110: 조항 106 내지 조항 109 중 임의의 방법에서, T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 미리 결정

된다.

- [0326] 조항 111: 조항 106 내지 조항 109 중 임의의 방법에서, T, P1, P2, P3, 및 P4 에 대한 입력 값들은 비트스트림에서 시그널링된다.
- [0327] 조항 112: 조항 103 내지 조항 111 중 임의의 방법에서, 절대 값들의 합에 대한 정규화된 값을 계산하는 단계는 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하는 단계를 포함하고, locSumAbs 는 절대 값들의 합을 포함하고 baseLevelOffset 은 베이스 레벨 오프셋 값을 포함한다.
- [0328] 조항 113: 조항 103 내지 조항 112 중 임의의 방법에서, 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하는 단계는, 절대 값들의 합에 대한 정규화된 값에 대응하는 라이스 파라미터를 결정하는 단계, 및 라이스 파라미터를 사용하여 현재 계수에 대한 값을 이진화하여 이진화된 값을 형성하는 단계를 포함한다.
- [0329] 조항 114: 조항 103 내지 조항 113 중 임의의 방법에서, 상기 현재 블록은 제 1 블록을 포함하고, 상기 비디오 데이터의 컬러 컴포넌트는 상기 제 1 블록을 포함하며, 상기 절대 값들의 합은 절대 값들의 제 1 합을 포함하고, 상기 복수의 이웃 계수들은 제 1 복수의 이웃 계수들을 포함하며, 상기 방법은, 상기 비디오 데이터의 상기 컬러 컴포넌트의 계수들을 코딩하는데 사용된 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하는 단계; 상기 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 상기 히스토리 카운터에 대한 현재 값으로부터 상기 비디오 데이터의 상기 컬러 컴포넌트의 제 2 블록에 대한 라이스 파라미터 스케일링 값을 도출하는 단계; 제 2 복수의 이웃 계수들의 절대 값들의 제 2 합을 상기 제 2 블록의 제 2 계수에 대해 계산하는 단계; 상기 라이스 파라미터 스케일링 값을 사용하여 상기 절대 값들의 제 2 합에 대한 스케일링된 값을 계산하는 단계; 및 상기 절대 값들의 제 2 합에 대한 상기 스케일링된 값을 사용하여 상기 제 2 블록의 상기 제 2 계수를 코딩하는 단계를 더 포함한다.
- [0330] 조항 115: 조항 114 의 방법에서, 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoeff}[\text{idx}] \geq T) ? P2 : P1$ 를 계산하는 단계를 포함하고, idx 는 컬러 컴포넌트를 나타내고, $\text{StatCoeff}[\text{idx}]$ 는 히스토리 카운터에 대한 현재 값을 나타내며, T, P1, 및 P2 는 입력 값들을 나타낸다.
- [0331] 조항 116: 조항 115 의 방법에서, T, P1, 및 P2 에 대한 입력 값들은 미리 결정된다.
- [0332] 조항 117: 조항 115 의 방법에서, T, P1, 및 P2 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0333] 조항 118: 조항 115 내지 조항 117 중 임의의 방법에서, $T = 4$, $P2 = 2$, 및 $P1 = 1$ 이다.
- [0334] 조항 119: 조항 114 내지 조항 118 중 임의의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는, 상기 비디오 데이터의, 상기 컬러 컴포넌트를 포함하는 모든 컬러 컴포넌트들에 대한, 상기 히스토리 카운터를 포함하는 히스토리 카운터들의 집성을 나타내는 집성된 히스토리 카운터 값을 계산하는 단계; 및 상기 집성된 히스토리 카운터 값을 이용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함한다.
- [0335] 조항 120: 조항 119 의 방법에서, 상기 집성된 히스토리 카운터 값을 계산하는 단계는 $((\text{StatCoeff}[0] * w0) + \text{StatCoeff}[1] * w1 + \text{StatCoeff}[2] * w2) / (w1 + w2 + w3)$ 를 계산하는 단계를 포함하고, $\text{StatCoeff}[0]$ 는 컬러 컴포넌트들의 제 1 컬러 컴포넌트에 대한 히스토리 카운터들의 제 1 히스토리 카운터를 나타내고, $\text{StatCoeff}[1]$ 는 컬러 컴포넌트들의 제 2 컬러 컴포넌트에 대한 히스토리 카운터들의 제 2 히스토리 카운터를 나타내며, $\text{StatCoeff}[2]$ 는 컬러 컴포넌트들의 제 3 컬러 컴포넌트에 대한 히스토리 카운터들의 제 3 히스토리 카운터를 나타내고, $w1$, $w2$, 및 $w3$ 은 각각의 가중치 값들을 나타낸다.
- [0336] 조항 121: 조항 120 의 방법에서, 상기 라이스 파라미터 스케일링 값을 도출하는 단계는 $(\text{StatCoefAverage} \geq T) ? P2 : P1$ 를 사용하여 상기 라이스 파라미터 스케일링 값을 도출하는 단계를 포함하고, StatCoefAverage 는 집성된 히스토리 카운터 값을 나타내고, T, P1 및 P2 는 입력 값들을 나타낸다.
- [0337] 조항 122: 조항 121 의 방법에서, T, P1, 및 P2 에 대한 입력 값들은 미리 결정된다.
- [0338] 조항 123: 조항 121 의 방법에서, T, P1, 및 P2 에 대한 입력 값들은 비디오 데이터에서 시그널링된다.
- [0339] 조항 124: 조항 121 내지 조항 123 중 임의의 방법에서, $T = 4$, $P2 = 2$, 및 $P1 = 1$ 이다.
- [0340] 조항 125: 조항 121 내지 조항 124 중 임의의 방법에서, $w0 = 2$, $w1 = 1$, 및 $w2 = 1$ 이다.
- [0341] 조항 126: 조항 114 내지 조항 125 중 임의의 방법에서, 상기 제 2 블록은 코딩 트리 유닛 (CTU), 변환 유닛 (TU), 또는 서브-변환 유닛 (서브-TU) 중 하나를 포함한다.

- [0342] 조항 127: 조항 103 내지 조항 126 중 임의의 방법에서, 상기 현재 계수를 코딩하기 위해, 상기 하나 이상의 프로세서들은 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 디코딩하도록 구성된다.
- [0343] 조항 128: 조항 103 내지 조항 127 중 임의의 방법에서, 상기 현재 계수를 코딩하기 위해, 상기 하나 이상의 프로세서들은 절대 값들의 합에 대한 상기 정규화된 값을 사용하여 상기 현재 계수를 인코딩하도록 구성된다.
- [0344] 조항 129: 비디오 데이터를 코딩하기 위한 디바이스로서, 비디오 데이터를 저장하도록 구성된 메모리; 및 회로 부에서 구현되는 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하도록 구성된다.
- [0345] 조항 130: 조항 129 의 디바이스에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 5 개의 이웃 계수들을 포함한다.
- [0346] 조항 131: 조항 129 및 조항 130 중 임의의 디바이스에서, 상기 복수의 이웃 계수들은 상기 현재 계수에 대한 2 개의 우측-이웃 계수들, 상기 현재 계수에 대한 2 개의 하방-이웃 계수들, 및 상기 현재 계수에 대한 하나의 하방-좌측 이웃 계수를 포함한다.
- [0347] 조항 132: 조항 129 내지 조항 131 중 임의의 디바이스에서, 상기 베이스 레벨 오프셋 값을 계산하기 위해, 상기 하나 이상의 프로세서들은, $\text{baseLevelOffset} = (\text{bitDepth} > T) ? (\text{isIntra}() ? P1: P2) : (\text{isIntra}() ? P3: P4)$ 를 계산하도록 구성되고, bitDepth 는 비디오 데이터에 대한 비트 심도를 포함하고, $\text{isIntra}()$ 는 슬라이스에 대한 슬라이스 타입이 인트라-예측된 슬라이스이면 참의 값을 복귀시키거나, 슬라이스에 대한 슬라이스 타입이 인터-예측된 슬라이스이면 거짓의 값을 복귀시키고, T , $P1$, $P2$, $P3$, 및 $P4$ 는 각각의 입력 값들을 포함한다.
- [0348] 조항 133: 조항 132 의 디바이스에서, $T = 12$ 이다.
- [0349] 조항 134: 조항 132 및 조항 133 중 임의의 디바이스에서, T , $P1$, $P2$, $P3$, 및 $P4$ 에 대한 입력 값들은 미리 결정된다.
- [0350] 조항 135: 조항 129 내지 조항 134 중 임의의 디바이스에서, 절대 값들의 합에 대한 정규화된 값을 계산하기 위해서, 하나 이상의 프로세서들은 $\text{Clip3}(0, 31, \text{locSumAbs} - \text{baseLevelOffset})$ 을 계산하도록 구성되고, locSumAbs 는 절대 값들의 합을 포함하고 baseLevelOffset 은 베이스 레벨 오프셋 값을 포함한다.
- [0351] 조항 136: 조항 135 의 디바이스에서, 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하기 위해, 하나 이상의 프로세서들은, 절대 값들의 합에 대한 정규화된 값에 대응하는 라이스 파라미터를 결정하고, 그리고 라이스 파라미터를 사용하여 현재 계수에 대한 값을 이진화하여 이진화된 값을 형성하도록 구성된다.
- [0352] 조항 137: 조항 129 내지 조항 136 중 임의의 디바이스에서, 상기 현재 블록은 제 1 블록을 포함하고, 상기 비디오 데이터의 컬러 컴포넌트는 상기 제 1 블록을 포함하며, 상기 절대 값들의 합은 절대 값들의 제 1 합을 포함하고, 상기 복수의 이웃 계수들은 제 1 복수의 이웃 계수들을 포함하며, 상기 하나 이상의 프로세서들은, 상기 비디오 데이터의 상기 컬러 컴포넌트의 계수들을 코딩하는데 사용된 라이스 파라미터들의 히스토리를 나타내는 히스토리 카운터를 유지하고; 상기 라이스 파라미터 스케일링 값에 대한 값을 명시적으로 코딩하지 않고, 상기 히스토리 카운터에 대한 현재 값으로부터 상기 비디오 데이터의 상기 컬러 컴포넌트의 제 2 블록에 대한 라이스 파라미터 스케일링 값을 도출하며; 제 2 복수의 이웃 계수들의 절대 값들의 제 2 합을 상기 제 2 블록의 제 2 계수에 대해 계산하고; 상기 라이스 파라미터 스케일링 값을 사용하여 상기 절대 값들의 제 2 합에 대한 스케일링된 값을 계산하며; 그리고 상기 절대 값들의 제 2 합에 대한 상기 스케일링된 값을 사용하여 상기 제 2 블록의 상기 제 2 계수를 코딩하도록 추가로 구성된다.
- [0353] 조항 138: 조항 137 의 디바이스에서, 라이스 파라미터 스케일링 값을 도출하기 위해, 하나 이상의 프로세서들은, $(\text{StatCoeff}[\text{idx}] \geq T) ? P2 : P1$ 를 계산하도록 구성되고, idx 는 컬러 컴포넌트를 나타내고, $\text{StatCoeff}[\text{idx}]$ 는 히스토리 카운터에 대한 현재 값을 나타내며, T , $P1$, 및 $P2$ 는 입력 값들을 나타낸다.
- [0354] 조항 139: 조항 129 내지 조항 138 중 임의의 디바이스에서, 현재 계수를 코딩하기 위해, 하나 이상의 프로세서들은, 절대 값들의 합에 대한 정규화된 값을 사용하여 상기 현재 계수를 인코딩하는 것 또는 절대 값들의 합에 대한 정규화된 값을 사용하여 상기 현재 계수를 디코딩하는 것 중 적어도 하나를 수행하도록 구성된다.

- [0355] 조항 140: 조항 129 내지 조항 139 중 임의의 디바이스에서, 디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함한다.
- [0356] 조항 141: 조항 129 내지 조항 140 중 임의의 디바이스에서, 디바이스는 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스, 또는 셋탑 박스 중 하나 이상을 포함한다.
- [0357] 조항 142: 컴퓨터 판독가능 저장 매체로서 명령들이 저장되고, 상기 명령들은, 실행될 때, 비디오 데이터를 코딩하기 위한 디바이스의 프로세서로 하여금: 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하게 하고; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하게 하며; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하게 하고; 그리고 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하게 한다.
- [0358] 조항 143: 비디오 데이터를 코딩하기 위한 디바이스로서, 비디오 데이터의 현재 블록에 대해 현재 계수에 대한 복수의 이웃 계수들의 절대 값들의 합을 계산하기 위한 수단; 비디오 데이터에 대한 비트 심도 또는 현재 블록을 포함하는 슬라이스에 대한 슬라이스 타입 중 적어도 하나에 따라 베이스 레벨 오프셋 값을 계산하기 위한 수단; 베이스 레벨 오프셋 값을 사용하여 절대 값들의 합에 대한 정규화된 값을 계산하기 위한 수단; 및 절대 값들의 합에 대한 정규화된 값을 사용하여 현재 계수를 코딩하기 위한 수단을 포함한다.
- [0359] 예에 의존하여, 본 명세서에서 설명된 기법들 중 임의의 것의 소정의 액트들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있고, 전체적으로 부가되거나, 병합되거나, 또는 제거될 수도 있음 (예를 들어, 설명된 모든 액트들 또는 이벤트들이 그 기법들의 실시를 위해 필수적인 것은 아님) 이 인식되어야 한다. 더욱이, 소정의 예들에 있어서, 액트들 또는 이벤트들은 순차적인 것보다는, 예를 들어, 다중-스레딩된 프로세싱, 인터럽트 프로세싱, 또는 다중의 프로세서들을 통해 동시에 수행될 수도 있다.
- [0360] 하나 이상의 예에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합에서 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 컴퓨터 판독가능 매체 상의 하나 이상의 명령 또는 코드로서 저장되거나 이를 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예를 들어 통신 프로토콜에 따라 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 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 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수도 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품이 컴퓨터 판독가능 매체를 포함할 수도 있다.
- [0361] 예로서, 그리고 비제한적으로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 희망하는 프로그램 코드를 저장하기 위해 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 커넥션이 컴퓨터 판독가능 매체로 적절히 명명된다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 트위스티드 페어, 디지털 가입자 라인 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되면, 동축 케이블, 광섬유 케이블, 트위스티드 페어, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 캐리어 파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않고, 대신 비일시적, 유형의 저장 매체들에 관련된다는 것을 이해해야 한다. 본 명세서에서 사용된 바와 같이, 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 및 블루-레이 디스크를 포함하며, 여기서, 디스크 (disk) 들은 보통 데이터를 자기적으로 재현하는 한편, 디스크 (disc) 들은 레이저들로 데이터를 광학적으로 재현한다. 상기의 조합들이 또한, 컴퓨터 판독가능 매체들의 레인지 내에 포함되어야 한다.
- [0362] 명령들은 하나 이상의 프로세서들, 이를 테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 또는 다른 등가의 집적 또는 이산 로직 회로에 의해 실행될 수도 있다. 이에 따라, 본 명세서에서 사용된 바와 같은 용어들 "프로세서" 및 "프로세싱 회로" 는 전술한 구조들 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 부가적으로, 일부 양태들에 있어서, 본 명세서에서 설명된 기능성

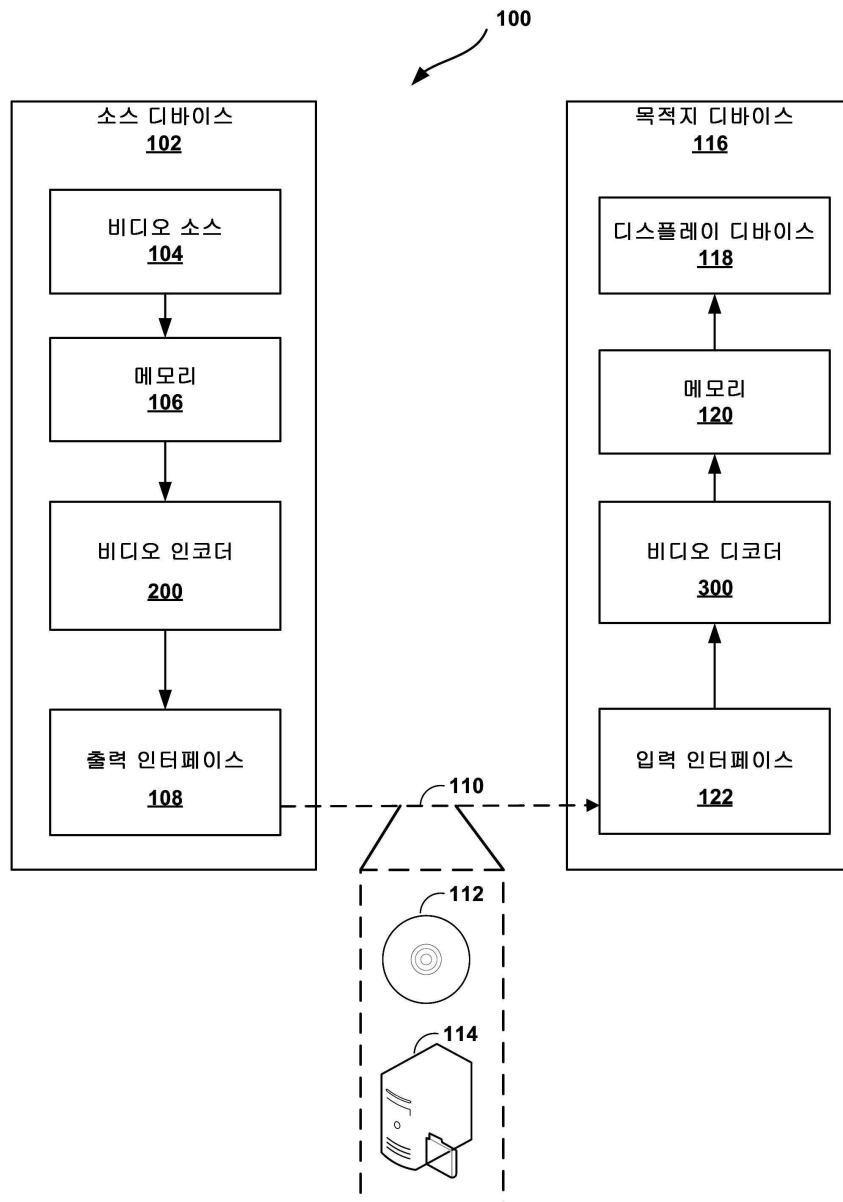
은 인코딩 및 디코딩을 위해 구성되거나 또는 결합된 코덱에서 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 그 기법들은 하나 이상의 회로들 또는 로직 요소들에서 완전히 구현될 수도 있을 것이다.

[0363] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들면, 칩 세트) 를 포함하는, 매우 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, 위에서 설명된 바와 같이, 다양한 유닛들이 코덱 하드웨어 유닛에 결합될 수도 있거나, 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는 상호동작 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

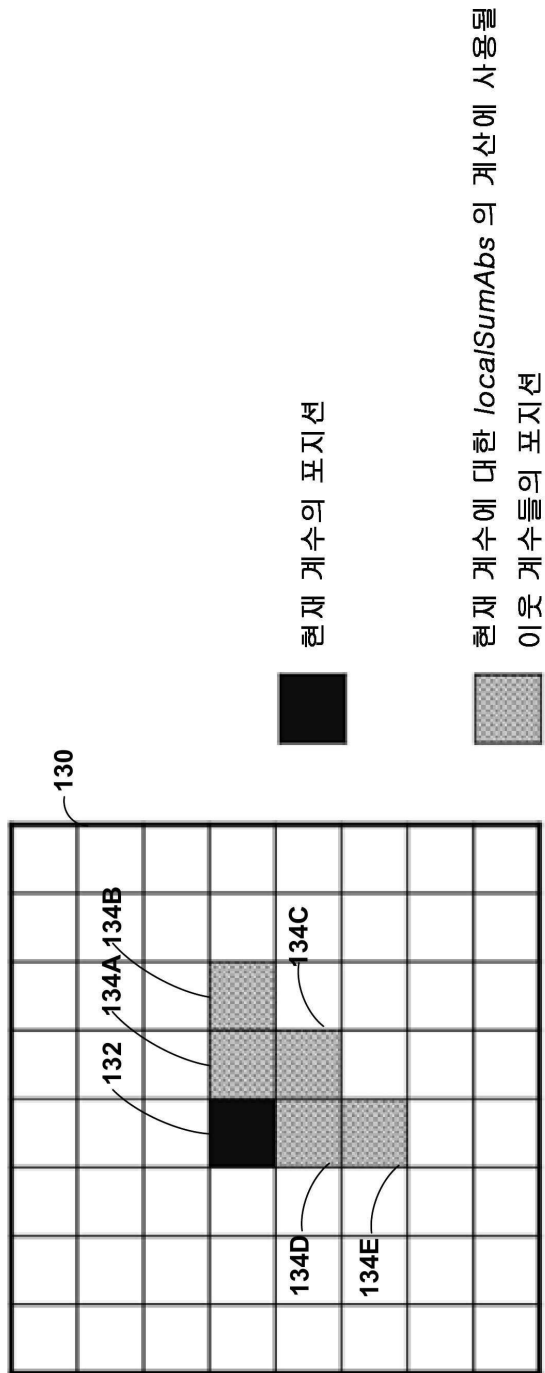
[0364] 다양한 예들이 개시되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

도면1

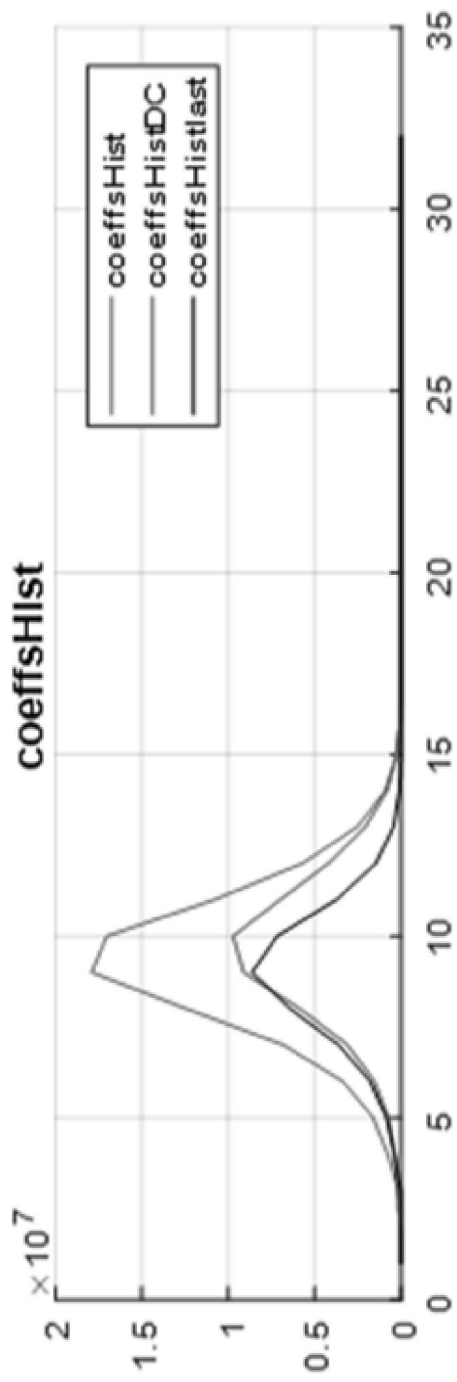


도면2

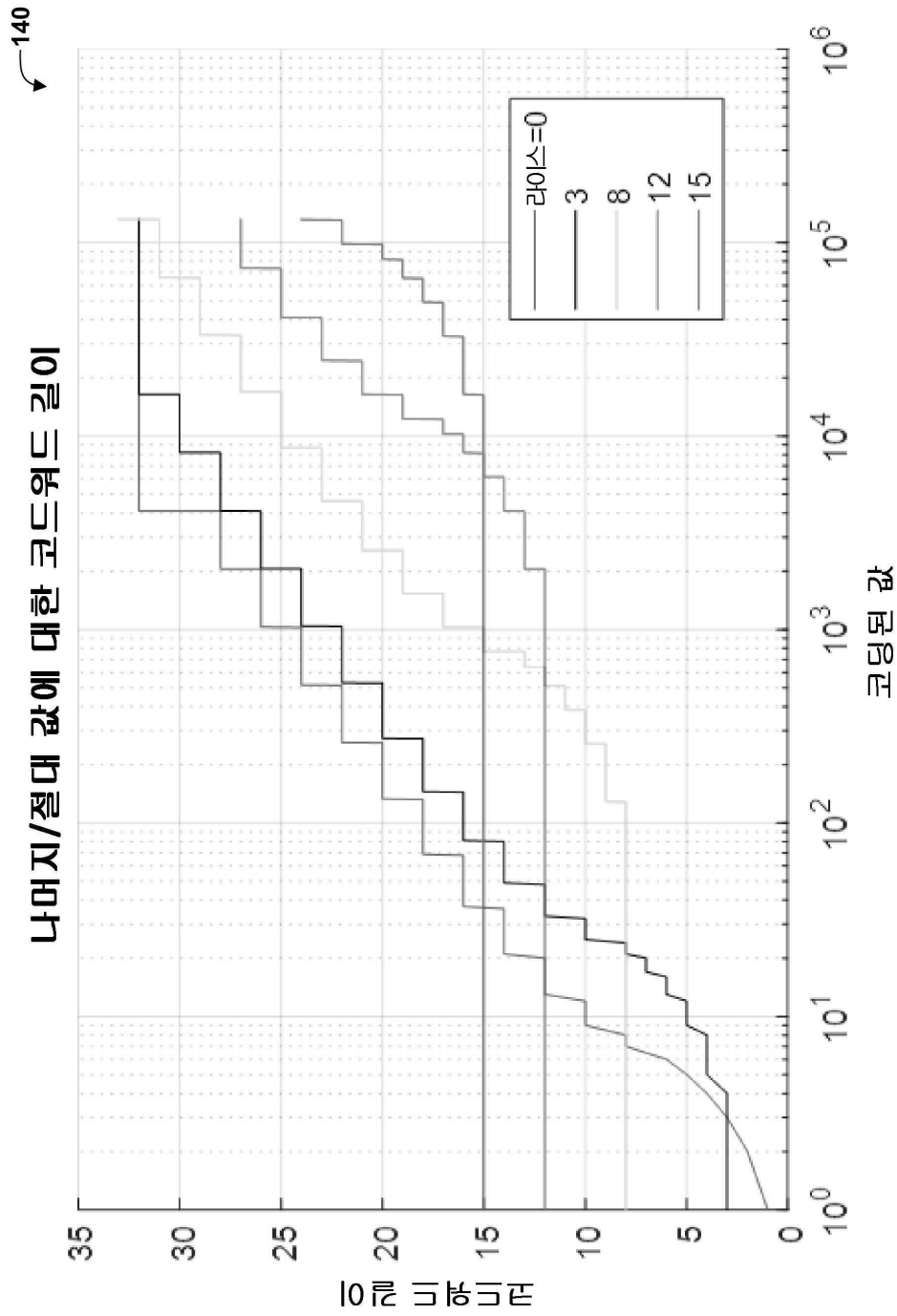


도면3

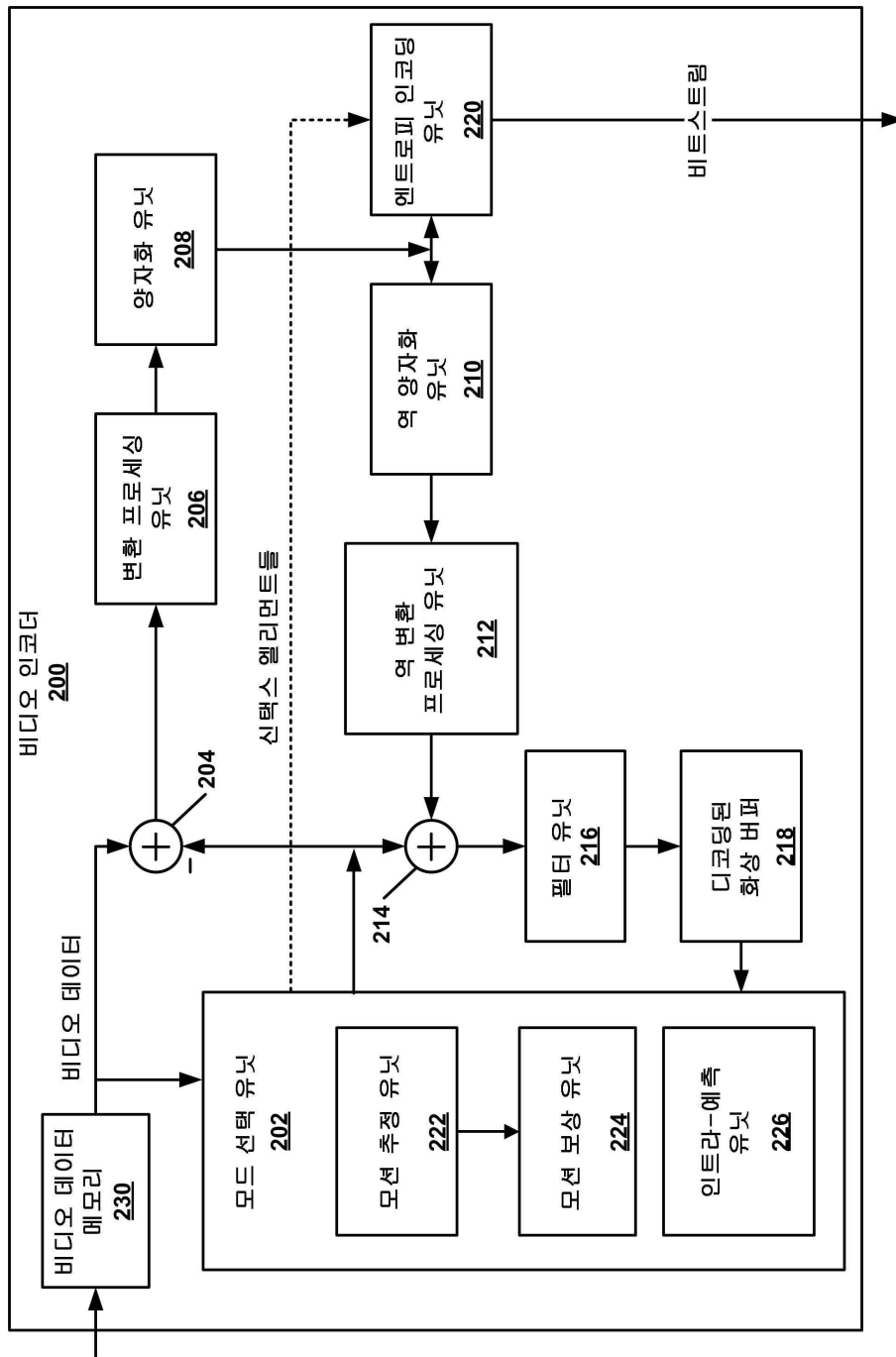
136



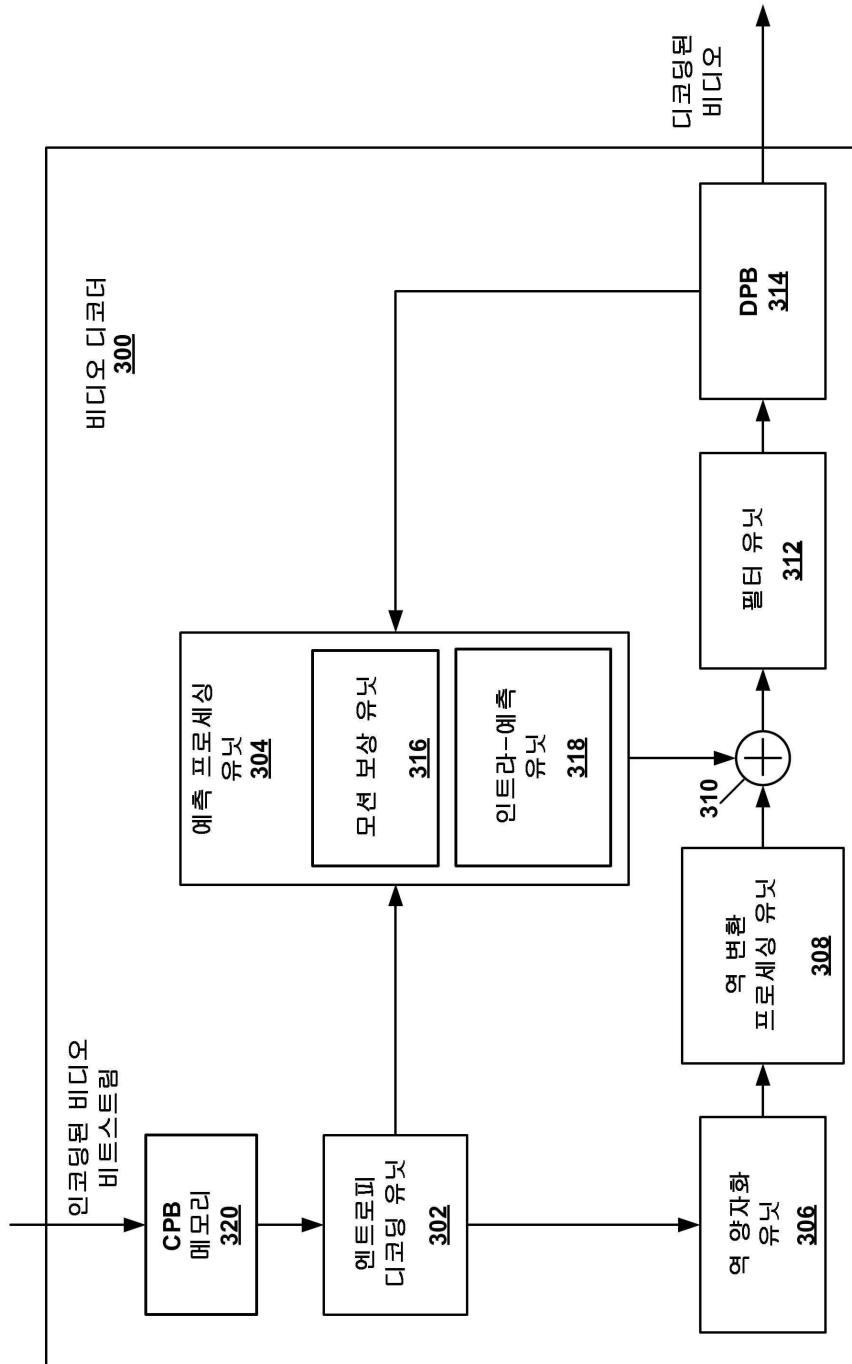
도면4



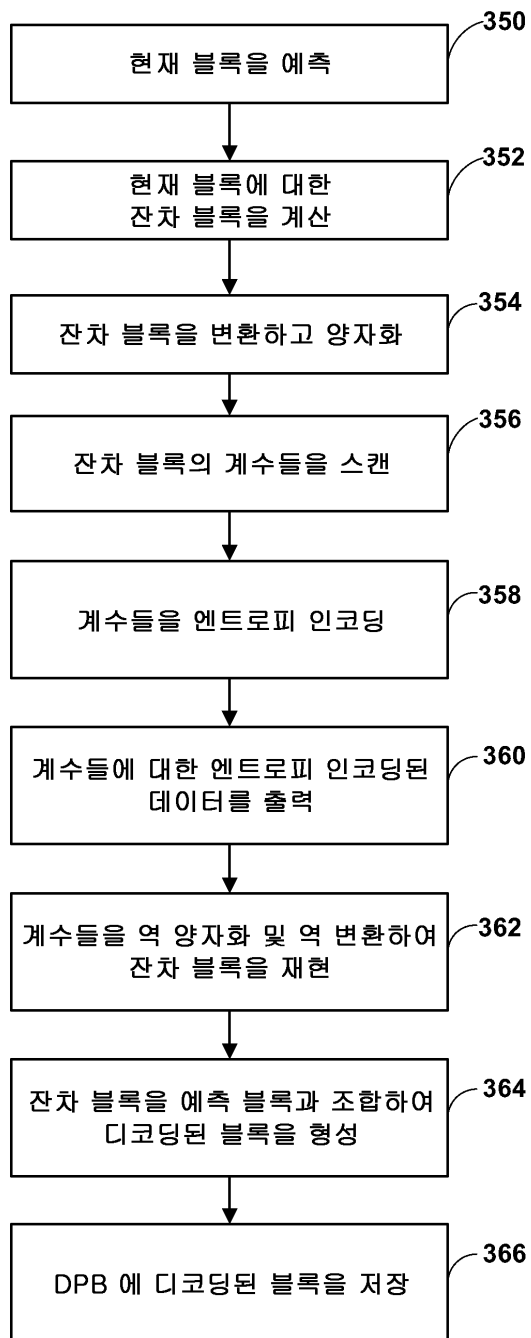
도면5



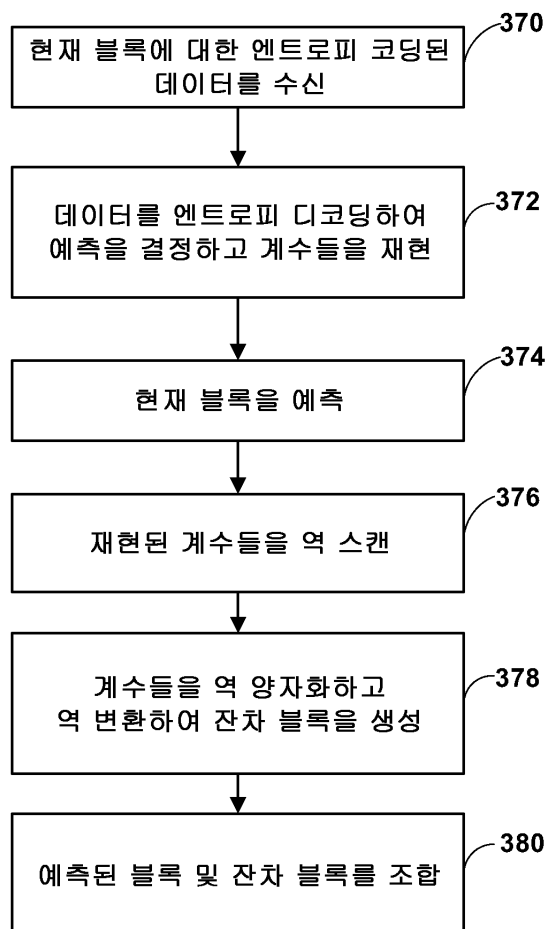
도면6



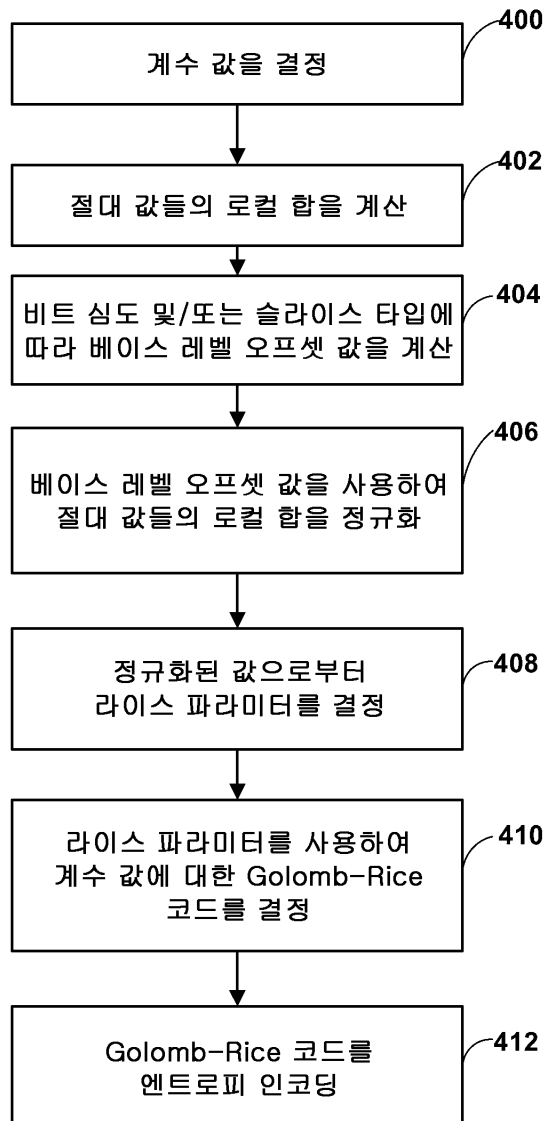
도면7



도면8



도면9



도면10

