



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년04월09일
(11) 등록번호 10-2794367
(24) 등록일자 2025년04월07일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/11 (2014.01) HO4N 19/105 (2014.01)
HO4N 19/13 (2014.01) HO4N 19/159 (2014.01)
HO4N 19/167 (2014.01) HO4N 19/176 (2014.01)
HO4N 19/186 (2014.01) HO4N 19/46 (2014.01)
HO4N 19/593 (2014.01) HO4N 19/70 (2014.01)
- (52) CPC특허분류
HO4N 19/11 (2015.01)
HO4N 19/105 (2015.01)
- (21) 출원번호 10-2021-7033329
- (22) 출원일자(국제) 2020년03월23일
심사청구일자 2021년10월15일
- (85) 번역문제출일자 2021년10월15일
- (65) 공개번호 10-2021-0134402
- (43) 공개일자 2021년11월09일
- (86) 국제출원번호 PCT/CN2020/080674
- (87) 국제공개번호 WO 2020/192629
국제공개일자 2020년10월01일
- (30) 우선권주장
62/822,865 2019년03월23일 미국(US)
(뒷면에 계속)
- (56) 선행기술조사문헌
W02018232676 A1
Wang (Huawei) B et al: "CE3-3.1.1: Unified
MPM list generation", no.JVET-N0184,
2019.3.17.

- (73) 특허권자
후아웨이 테크놀러지 컴퍼니 리미티드
중국 518129 광둥성 셴젠 룡강 디스트릭트 반티안
후아웨이 어드미니스트레이션 빌딩
- (72) 발명자
왕 바오
독일 80992 뮌헨 리스트트라세 25
에센리크 세미흐
독일 80992 뮌헨 리스트트라세 25
(뒷면에 계속)
- (74) 대리인
유미특허법인

전체 청구항 수 : 총 25 항

심사관 : 김영태

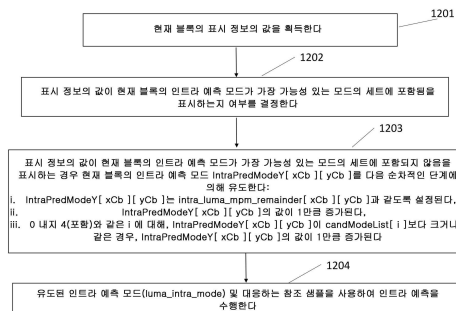
(54) 발명의 명칭 인코더, 디코더 및 대응하는 인트라 예측을 위한 방법

(57) 요약

인트라 예측을 위한 디바이스 및 방법이 제공된다. 이 방법은, 현재 블록의 표시 정보의 값을 획득하는 단계를 포함한다. 표시 정보의 값이 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되지 않음을 표시하는 경우, 다음 순차적인 단계에 의해 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]를 유

(뒷면에 계속)

대표도 - 도12



도하는 단계: i. IntraPredModeY[xCb][yCb]는 intra_luma_mpm_remainder[xCb][yCb]과 같도록 설정된다,
 ii. IntraPredModeY[xCb][yCb]의 값이 1만큼 증가된다;를 포함한다. 이 방법은 현재 블록의 인트라 예측 모
 드를 결정하는 효율을 향상할 수 있다.

(52) CPC특허분류

- H04N 19/13* (2015.01)
- H04N 19/159* (2015.01)
- H04N 19/167* (2015.01)
- H04N 19/176* (2015.01)
- H04N 19/186* (2015.01)
- H04N 19/46* (2015.01)
- H04N 19/593* (2015.01)
- H04N 19/70* (2015.01)

(30) 우선권주장

- 62/824,282 2019년03월26일 미국(US)
- 62/824,360 2019년03월27일 미국(US)

(72) 발명자

코트라 아난드 메헤르

독일 80992 뮌헨 리스트트라세 25

가오 한

독일 80992 뮌헨 리스트트라세 25

천 장러

미국 95050 캘리포니아주 산타클라라 센트럴 익스
 프레스웨이 2330

명세서

청구범위

청구항 1

디코딩 디바이스에 의해 구현되는 인트라 예측을 위한 방법으로서,

비트스트림으로부터 현재 블록의 표시 정보의 값을 획득하는 단계 - 상기 표시 정보의 값은 상기 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 표시하고, 상기 가장 가능성 있는 모드의 세트는 평면 모드와 5개의 인트라 예측 모드를 포함함 - ;

상기 표시 정보의 값이 상기 현재 블록의 인트라 예측 모드가 상기 가장 가능성 있는 모드의 세트에 포함되지 않음을 표시하는 경우, 다음 순차적인 단계에 의해 상기 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 를 유도하는 단계:

- i. $IntraPredModeY[xCb][yCb]$ 는 $intra_luma_mpm_remainder[xCb][yCb]$ 과 같도록 설정된다,
- ii. $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다;를 포함하며,

여기서, $intra_luma_mpm_remainder[xCb][yCb]$ 는 상기 비트스트림으로부터 도출되어 나머지 인트라 예측 모드를 표현하고, 루마 위치(xCb, yCb)는 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 상기 현재 블록의 좌측 상단 샘플을 지정하는, 인트라 예측을 위한 방법.

청구항 2

제1항에 있어서,

상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 $candModeList[i]$ 는,

$i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 $candModeList[i]$ 이 $candModeList[j]$ 보다 큰 경우, 두 값은 다음과 같이 교환됨:

$(candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j])$.

에 의해 유도되는, 인트라 예측을 위한 방법.

청구항 3

제2항에 있어서,

단계 ii 이후에, 상기 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 를 유도하는 단계는,

iii. 0 내지 4(포함)와 같은 i 에 대해, $IntraPredModeY[xCb][yCb]$ 이 $candModeList[i]$ 보다 크거나 같은 경우, $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다;를 더 포함하는, 인트라 예측을 위한 방법.

청구항 4

제1항에 있어서,

상기 표시 정보의 값은 $intra_luma_mpm_flag$ 에 의해 표시되는, 인트라 예측을 위한 방법.

청구항 5

제4항에 있어서,

$intra_luma_mpm_flag$ 의 값이 1과 같지 않은 경우, 상기 현재 블록의 인트라 예측 모드는 상기 가장 가능성 있는 모드의 세트에 포함되지 않는, 인트라 예측을 위한 방법.

청구항 6

인코딩 디바이스에 의해 구현되는 인트라 예측을 위한 방법으로서,

현재 블록의 인트라 예측 모드 $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ 이 가장 가능성 있는 모드의 세트에 포함되지 않는 경우 현재 블록의 인트라 예측 모드 $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ 를 사용하여 다음 순차적인 단계에 의해 변수를 유도하는 단계 - 상기 가장 가능성 있는 모드의 세트는 평면 모드와 5개의 인트라 예측 모드를 포함하고:

- i. $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ 는 $\text{intra_luma_mpm_remainder}[x_{Cb}][y_{Cb}]$ 과 같도록 설정되고,
 - ii. $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ 의 값이 1만큼 감소되며,
 - iii. 0 내지 4(포함)와 같은 i 에 대해, $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ 이 $\text{candModeList}[i]$ 보다 크거나 같은 경우, $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ 의 값은 1만큼 감소되고 - 여기서, $\text{candModeList}[i]$ 는 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드를 표현함 - , 여기서, 루마 위치(x_{Cb} , y_{Cb})는 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 상기 현재 블록의 좌측 상단 샘플을 지정하고, $\text{intra_luma_mpm_remainder}[x_{Cb}][y_{Cb}]$ 는 나머지 인트라 예측 모드를 표현함 - ; 및
- 비트스트림을 디코더에 송신하는 단계 - 여기서, 상기 비트스트림은 상기 변수의 정보를 포함함 - 를 포함하는 인트라 예측을 위한 방법.

청구항 7

제6항에 있어서,
 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 $\text{candModeList}[i]$ 는,
 $i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 $\text{candModeList}[i]$ 이 $\text{candModeList}[j]$ 보다 큰 경우, 두 값은 다음과 같이 교환됨:
 $(\text{candModeList}[i], \text{candModeList}[j]) = \text{Swap}(\text{candModeList}[i], \text{candModeList}[j])$.
 에 의해 유도되는, 인트라 예측을 위한 방법.

청구항 8

제6항에 있어서,
 상기 비트스트림은 현재 블록의 표시 정보의 값을 포함하며, 여기서 상기 표시 정보의 값은 상기 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 가리키는, 인트라 예측을 위한 방법.

청구항 9

제8항에 있어서,
 상기 표시 정보의 값은 $\text{intra_luma_mpm_flag}$ 에 의해 표시되는, 인트라 예측을 위한 방법.

청구항 10

제9항에 있어서,
 $\text{intra_luma_mpm_flag}$ 의 값이 1과 같지 않은 경우, 상기 현재 블록의 인트라 예측 모드는 상기 가장 가능성 있는 모드의 세트에 포함되지 않는, 인트라 예측을 위한 방법.

청구항 11

제1항 내지 제5항 중 어느 한 항에 따른 방법을 수행하기 위한 처리 회로를 포함하는 디코더(30).

청구항 12

제6항 내지 제10항 중 어느 한 항에 따른 방법을 수행하기 위한 처리 회로를 포함하는 인코더(20).

청구항 13

컴퓨터가 판독 가능한 저장 매체에 저장되어 있는 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은 제1항 내지 제10항 중 어느 한 항에 따른 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램.

청구항 14

인트라 예측을 위한 디코더로서,

하나 이상의 프로세서; 및

상기 프로세서에 결합되고 상기 프로세서에 의해 실행되기 위한 프로그래밍을 저장하는 비일시적 컴퓨터-판독 가능한 저장 매체-여기서, 상기 프로그래밍은, 상기 프로세서에 의해 실행되는 경우, 상기 디코더로 하여금 제1항 내지 제5항 중 어느 한 항에 따른 방법을 수행하도록 구성함-;를 포함하는, 디코더.

청구항 15

인트라 예측을 위한 인코더로서,

하나 이상의 프로세서; 및

상기 프로세서에 결합되고 상기 프로세서에 의해 실행되기 위한 프로그래밍을 저장하는 비일시적 컴퓨터-판독 가능한 저장 매체-여기서, 상기 프로그래밍은, 상기 프로세서에 의해 실행되는 경우, 상기 디코더로 하여금 제6항 내지 제10항 중 어느 한 항에 따른 방법을 수행하도록 구성함-;를 포함하는, 인코더.

청구항 16

인트라 예측을 위한 디코딩 디바이스로서,

비트스트림으로부터 현재 블록의 표시 정보의 값을 획득하도록 구성되는 획득 유닛(1401) - 여기서, 상기 표시 정보의 값은 상기 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 표시하고, 상기 가장 가능성 있는 모드의 세트는 평면 모드와 5개의 인트라 예측 모드를 포함함 - ;

상기 표시 정보의 값이 상기 현재 블록의 인트라 예측 모드가 상기 가장 가능성 있는 모드의 세트에 포함되지 않음을 표시하는 경우 상기 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 을 다음 순차적인 단계에 의해 유도하도록 구성되는 유도 유닛(1402):

- i. $IntraPredModeY[xCb][yCb]$ 는 $intra_luma_mpm_remainder[xCb][yCb]$ 과 같도록 설정된다,
- ii. $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다;을 포함하되,

여기서, $intra_luma_mpm_remainder[xCb][yCb]$ 는 상기 비트스트림으로부터 도출되어 나머지 인트라 예측 모드를 표현하고, 루마 위치(xCb , yCb)는 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 상기 현재 블록의 좌측 상단 샘플을 지정하는, 인코딩 디바이스.

청구항 17

제16항에 있어서,

상기 유도 유닛은, 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 $candModeList[i]$ 를,

$i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 $candModeList[i]$ 이 $candModeList[j]$ 보다 큰 경우, 두 값은 다음과 같이 교환됨:

$$(candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j])$$

에 의해 유도하도록 추가적으로 구성되는, 인코딩 디바이스.

청구항 18

제16항에 있어서,

단계 ii 이후에, 상기 유도 유닛은, 상기 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 를,

iii. 0 내지 4(포함)와 같은 i 에 대해, $IntraPredModeY[xCb][yCb]$ 이 $candModeList[i]$ 보다 크거나 같은 경우, $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다;

에 의해 유도하도록 추가적으로 구성되는, 인코딩 디바이스.

청구항 19

제16항에 있어서,

상기 표시 정보의 값은 `intra_luma_mpm_flag`에 의해 표시되는, 인코딩 디바이스.

청구항 20

제19항에 있어서,

`intra_luma_mpm_flag`의 값이 1과 같지 않은 경우, 상기 현재 블록의 인트라 예측 모드는 상기 가장 가능성 있는 모드의 세트에 포함되지 않는, 인코딩 디바이스.

청구항 21

구현되는 인트라 예측을 위한 인코딩 디바이스로서,

현재 블록의 인트라 예측 모드 `IntraPredModeY[xCb][yCb]`이 가장 가능성 있는 모드의 세트에 포함되지 않는 경우 현재 블록의 인트라 예측 모드 `IntraPredModeY[xCb][yCb]`를 사용하여 다음 순차적인 단계에 의해 변수를 유도하도록 구성되는 유도 유닛(1501) - 상기 가장 가능성 있는 모드의 세트는 평면 모드와 5개의 인트라 예측 모드를 포함하고:

- i. `IntraPredModeY[xCb][yCb]`는 `intra_luma_mpm_remainder[xCb][yCb]`과 같도록 설정되고,
- ii. `IntraPredModeY[xCb][yCb]`의 값이 1만큼 감소되며,
- iii. 0 내지 4(포함)와 같은 i 에 대해, `IntraPredModeY[xCb][yCb]`이 `candModeList[i]`보다 크거나 같은 경우, `IntraPredModeY[xCb][yCb]`의 값은 1만큼 감소되고 - 여기서, `candModeList[i]`는 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드를 표현함 - , 여기서, 루마 위치(xCb , yCb)는 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 상기 현재 블록의 좌측 상단 샘플을 지정하고, `intra_luma_mpm_remainder[xCb][yCb]`는 나머지 인트라 예측 모드를 표현함 - ; 및

비트스트림을 디코더로 송신하도록 구성되는 전송 유닛(1502) - 여기서 상기 비트스트림은 상기 변수의 정보를 포함함 -

을 포함하는 인코딩 디바이스.

청구항 22

제21항에 있어서,

상기 유도 유닛은, 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 `candModeList[i]`를,

$i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 `candModeList[i]`이 `candModeList[j]`보다 큰 경우, 두 값은 다음과 같이 교환됨:

$$(\text{candModeList}[i], \text{candModeList}[j]) = \text{Swap}(\text{candModeList}[i], \text{candModeList}[j])$$

에 의해 유도하도록 추가적으로 구성되는, 인코딩 디바이스.

청구항 23

제21항에 있어서,

비트스트림은 현재 블록의 표시 정보의 값을 포함하며, 여기서상기 표시 정보의 값은 상기 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 가리키는, 인코딩 디바이스.

청구항 24

제23항에 있어서,

상기 표시 정보의 값은 `intra_luma_mpm_flag`에 의해 표시되는, 인코딩 디바이스.

청구항 25

제24항에 있어서,

intra_luma_mpm_flag의 값이 1과 같지 않은 경우, 상기 현재 블록의 인트라 예측 모드는 상기 가장 가능성 있는 모드의 세트에 포함되지 않는, 인코딩 디바이스.

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2019년 3월 23일자로 제출된 미국 가출원 번호 제62/822,865호, 2019년 3월 26일자로 제출된 미국 가출원 번호 제62/824,282호, 및 2019년 3월 27일자로 제출된 미국 가출원 번호 제62/824,360호에 대해 우선권을 주장하며, 이들 출원 모두는 원용에 의해 본 명세서에 포함된다.

[0002] 본 출원(개시)의 실시예는 일반적으로 화상 처리 분야, 보다 구체적으로 인트라 예측에 관한 것이다.

배경 기술

[0003] 비디오 코딩(비디오 인코딩 및 디코딩)은, 일례로 브로드캐스트 디지털 TV, 인터넷 및 모바일 네트워크를 통한 비디오 전송, 화상 채팅과 같은 실시간 대화 애플리케이션, 화상 회의, DVD 및 블루-레이 디스크, 비디오 콘텐츠 수집 및 편집 시스템, 및 보안 애플리케이션의 캡코더와 같은, 넓은 범위의 디지털 비디오 애플리케이션에서 사용된다.

[0004] 비교적 짧은 비디오를 묘사하는데 필요한 비디오 데이터의 양은 상당할 수 있으며, 이는 대역폭 용량이 제한된 통신 네트워크를 통해 데이터가 스트리밍되거나 통신될 때 어려움을 초래할 수 있다. 따라서, 비디오 데이터는 일반적으로 현대 통신 네트워크를 통해 통신되기 전에 압축된다. 메모리 자원이 제한될 수 있으므로 비디오가 저장 디바이스에 저장될 때 비디오의 크기가 문제될 수도 있다. 비디오 압축 디바이스는 종종 소스에서 소프트웨어 및/또는 하드웨어를 사용하여 전송 또는 저장 전에 비디오 데이터를 코딩하여 디지털 비디오 이미지를 표

현하는데 필요한 데이터 양을 줄인다. 압축된 데이터는 비디오 데이터를 디코딩하는 비디오 압축 해제 디바이스에 의해 테스트네이션(destination)에서 수신된다. 네트워크 자원이 제한되고 더 높은 비디오 품질에 대한 요구가 계속 증가함에 따라, 화상 품질을 거의 또는 전혀 희생하지 않고 압축률을 향상하는 개선된 압축 및 압축 해제 기법이 바람직하다.

[0005] 특히, 인트라 예측 처리의 맥락에서 프로세서 부하를 줄이기 위한 지속적인 요구가 있다. 업계에서는, 인트라 예측 모드 코딩을 위해 가장 가능성 있는 모드(Most Probable Mode, MPM) 목록을 사용하는 것으로 알려져 있다. MPM 목록은 현재 블록의 인트라 예측 모드를 코딩하는데 요구되는 비트를 줄인다. 현재 블록의 인트라 예측 모드가 MPM 목록의 엔트리에 대응하는 경우, 실제의 모드가 아닌 인덱스를 코딩함으로써 더 적은 비트가 필요하게 된다. 그러나, 인트라 예측 모드의 유도 과정은 인트라 예측 모드가 MPM 목록에 포함되지 않은 경우 인트라 예측 모드 코딩의 효율에 영향을 미친다.

발명의 내용

[0006] 이를 고려하여, 본 발명의 기초가 되는 목적은 종래에 비해 더 효율적인 코딩을 초래하는 인트라 예측 모드를 획득하기 위한 기법을 제공하는 것이다.

[0007] 전술한 및 다른 목적은 독립항의 주제에 의해 달성된다. 추가적인 구현 형태는 종속항, 설명 및 도면으로부터 명백하다.

[0008] 제1 측면에 따르면, 본 발명은 디코딩 디바이스에 의해 수행되는 방법에 관련된다. 상기 방법은, 현재 블록의 표시 정보의 값을 획득하는 단계-상기 표시 정보의 값은 상기 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 표시함-를 포함한다. 상기 표시 정보의 값이 상기 현재 블록의 인트라 예측 모드가 상기 가장 가능성 있는 모드의 세트에 포함되지 않음을 표시하는 경우, 다음 순차적인 단계에 의해 상기 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 를 유도하는 단계:

[0009] i. $IntraPredModeY[xCb][yCb]$ 는 $intra_luma_mpm_remainder[xCb][yCb]$ 과 같도록 설정된다,

[0010] ii. $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다;를 포함하며,

[0011] 여기서, $intra_luma_mpm_remainder[xCb][yCb]$ 는 나머지 인트라 예측 모드를 표현하고, 루마 위치(xCb, yCb)는 상기 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 상기 현재 블록의 좌측 상단 샘플을 지정한다.

[0012] 상기 제1 측면에 따른 방법의 가능한 구현 형태에서, 단계 ii 이후에, 상기 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 를 유도하는 단계는,

[0013] iii. 0 내지 4(포함)와 같은 i 에 대해, $IntraPredModeY[xCb][yCb]$ 이 $candModeList[i]$ 보다 크거나 같은 경우, $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다;를 더 포함한다.

[0014] 상기 제1 측면에 따른 방법의 가능한 구현 형태에서, 상기 가장 가능성 있는 모드의 세트는 5개의 후보 인트라 예측 모드 및 평면 모드를 포함한다. 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 $candModeList[i]$ 는,

[0015] $i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 $candModeList[i]$ 이 $candModeList[j]$ 보다 큰 경우, 두 값은 다음과 같이 교환됨:

[0016] $(candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j])$ 에 의해 유도된다.

[0017] 제2 측면에 따르면, 본 발명은 인코딩 장치에 의해 수행되는 방법에 관련된다. 상기 방법은, 상기 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 이 상기 가장 가능성 있는 모드의 세트에 포함되지 않는 경우 현재 블록의 인트라 예측 모드 $IntraPredModeY[xCb][yCb]$ 를 사용하여 다음 순차적인 단계에 의해 변수(예를 들어, $intra_luma_mpm_remainder$)를 유도하는 단계:

[0018] i. 0 내지 4(포함)와 같은 i 에 대해, $IntraPredModeY[xCb][yCb]$ 이 $candModeList[i]$ 보다 큰 경우, $IntraPredModeY[xCb][yCb]$ 의 값은 1만큼 감소된다-여기서, $candModeList[i]$ 는 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드를 표현함-;

[0019] ii. $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 감소된다;를 포함하며,

- [0020] 여기서, 루마 위치(xCb , yCb)는 상기 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 상기 현재 블록의 좌측 상단 샘플을 지정한다. 상기 방법은 비트스트림을 디코더로 송신하는 단계를 더 포함하며, 여기서 상기 비트스트림은 상기 변수의 정보를 포함한다.
- [0021] 상기 제2 측면에 따른 방법의 가능한 구현 형태에서, 상기 가장 가능성 있는 모드의 세트는 5개의 후보 인트라 예측 모드 및 평면 모드를 포함한다. 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 candModeList[i]는,
- [0022] $i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 candModeList[i]이 candModeList[j]보다 큰 경우, 두 값은 다음과 같이 교환됨:
- [0023] (candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j])에 의해 유도된다.
- [0024] 본 발명의 제1 측면에 따른 방법은 본 발명의 제3 측면에 따른 디코딩 디바이스에 의해 수행될 수 있다. 상기 디코딩 디바이스는 획득 유닛 및 유도 유닛을 포함한다. 상기 획득 유닛은 현재 블록의 표시 정보의 값을 획득하도록 구성되며, 상기 표시 정보의 값은 상기 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 가리킨다. 상기 유도 유닛은, 상기 표시 정보의 값이 상기 현재 블록의 인트라 예측 모드가 상기 가장 가능성 있는 모드의 세트에 포함되지 않음을 표시하는 경우 상기 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]을 다음 순차적인 단계에 의해 유도하도록 구성된다:
- [0025] i. IntraPredModeY[xCb][yCb]는 intra_luma_mpm_remainder[xCb][yCb]과 같도록 설정된다,
- [0026] ii. IntraPredModeY[xCb][yCb]의 값이 1만큼 증가된다.
- [0027] 여기서, intra_luma_mpm_remainder[xCb][yCb]는 나머지 인트라 예측 모드를 표현하고, 루마 위치(xCb , yCb)는 상기 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 상기 현재 블록의 좌측 상단 샘플을 지정한다.
- [0028] 본 발명의 제3 측면에 따른 방법의 추가적인 특징 및 구현 형태는 본 발명의 제1 측면에 따른 장치의 특징 및 구현 형태에 대응한다.
- [0029] 본 발명의 제2 측면에 따른 방법은 본 발명의 제4 측면에 따른 인코딩 디바이스에 의해 수행될 수 있다. 상기 인코딩 디바이스는 유도 유닛 및 전송 유닛을 포함한다. 상기 유도 유닛은, 상기 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]이 상기 가장 가능성 있는 모드의 세트에 포함되지 않는 경우 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]를 사용하여 다음 순차적인 단계에 의해 변수(예를 들어, intra_luma_mpm_remainder)를 유도하도록 구성된다:
- [0030] i. 0 내지 4(포함)와 같은 i 에 대해, IntraPredModeY[xCb][yCb]이 candModeList[i]보다 큰 경우, IntraPredModeY[xCb][yCb]의 값은 1만큼 감소된다-여기서, candModeList[i]는 상기 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드를 표현함-;
- [0031] ii. IntraPredModeY[xCb][yCb]의 값이 1만큼 감소된다.
- [0032] 상기 전송 유닛은 비트스트림을 디코더로 송신하도록 구성되며, 여기서 상기 비트스트림은 상기 변수의 정보를 포함한다.
- [0033] 본 발명의 제4 측면에 따른 방법의 추가적인 특징 및 구현 형태는 본 발명의 제2 측면에 따른 장치의 특징 및 구현 형태에 대응한다.
- [0034] 제5 측면에 따르면, 본 발명은, 프로세서 및 메모리를 포함하는 비디오 스트림을 디코딩하기 위한 장치에 관련된다. 상기 메모리는 상기 프로세서로 하여금 상기 제1 측면에 따른 방법을 수행하게 하는 명령을 저장하고 있다.
- [0035] 제6 측면에 따르면, 본 발명은, 프로세서 및 메모리를 포함하는 비디오 스트림을 인코딩하기 위한 장치에 관련된다. 상기 메모리는 상기 프로세서로 하여금 상기 제2 측면에 따른 방법을 수행하게 하는 명령을 저장하고 있다.
- [0036] 제7 측면에 따르면, 실행되는 경우 하나 이상의 프로세서로 하여금 비디오 데이터를 코딩하게 하는 명령을 저장하는 컴퓨터-판독 가능한 저장 매체가 제안된다. 상기 명령은 상기 하나 이상의 프로세서로 하여금 상기 제1 또는 제2 측면 또는 상기 제1 또는 제2 측면의 임의의 가능한 실시예에 따른 방법을 수행하게 한다.

[0037] 제8 측면에 따르면, 본 발명은 컴퓨터 상에서 실행되는 경우 상기 제1 또는 제2 측면 또는 상기 제1 또는 제2 측면의 가능한 실시예에 따른 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램에 관련된다.

[0038] 평면 모드가 상기 가장 가능성 있는 모드의 세트에 있는 것으로 (목적적 또는 명시적) 항상 간주되는 경우, 이 발명의 실시예는 비-MPM 모드 유도 과정을 단순화한다. 평면(planar)는 인트라 예측 모드의 가장 작은 값에 대응하므로, 비-MPM 모드에 대한 유도 과정이 단순화될 수 있다. 비교 및 증가 단계의 경우, 평면(planar)는 항상 MPM 목록에 있으므로 평면 모드에 대한 비교가 저장될 수 있고, 따라서 나머지 인트라 예측 모드의 초기값은, 전술한 단계 ii에 대응하여, 1만큼 직접 증가될 수 있다. 이에 상응하여, 이 발명의 실시예는 종래에 비해 더 효율적인 코딩을 초래하는 인트라 예측 모드를 획득하기 위한 방법을 제공한다.

[0039] 하나 이상의 실시예의 세부사항은 첨부 도면 및 아래의 설명에 설명되어 있다. 다른 특징, 목적, 및 이점은 설명, 도면, 및 청구범위에서 명백할 것이다.

도면의 간단한 설명

[0040] 본 발명의 다음의 실시예는 첨부된 도면 및 도면을 참조하여 보다 상세하게 설명된다.

- 도 1a는 본 발명의 실시예들을 구현하도록 구성되는 비디오 코딩 시스템의 예시를 도시하는 블록도이다.
 - 도 1b는 본 발명의 실시예들을 구현하도록 구성되는 비디오 코딩 시스템의 다른 예를 도시하는 블록도이다.
 - 도 2는 본 발명의 실시예들을 구현하도록 구성되는 비디오 인코더의 예시를 도시하는 블록도이다.
 - 도 3은 본 발명의 실시예들을 구현하도록 구성되는 비디오 디코더의 예시적인 구조를 도시하는 블록도이다.
 - 도 4는 인코딩 장치 또는 디코딩 장치의 예시를 도시하는 블록도이다.
 - 도 5는 인코딩 장치 또는 디코딩 장치의 다른 예를 도시하는 블록도이다.
 - 도 6은 다중 참조 라인(Multiple Reference Line, MRL) 코딩 도구의 예시를 도시하는 블록도이다.
 - 도 7은 4×8 및 8×4 블록의 분할의 예시를 도시하는 블록도이다.
 - 도 8은 4×8, 8×4 및 4×4를 제외한 모든 블록의 분할의 예시를 도시하는 블록도이다.
 - 도 9은 CABAC 컨텍스트 모델링/코딩의 예시를 도시하는 블록도이다.
 - 도 10 은 현재 블록의 좌측(L) 및 상측(A) 이웃 블록의 예시를 도시하는 블록도이다.
 - 도 11은 인트라 예측 모드를 도시하는 블록도이다.
 - 도 12는 본 발명에 따른 디코딩 디바이스에 의해 구현되는 현재 블록의 인트라 예측의 방법(1200)의 실시예를 도시한다.
 - 도 13는 본 발명에 따른 인코딩 디바이스에 의해 구현되는 현재 블록의 인트라 예측의 방법(1300)의 실시예를 도시한다.
 - 도 14는 본 발명에 따른 이미지 디코더에서 사용되기 위한 디코딩 디바이스(1400)의 실시예를 도시한다.
 - 도 15는 본 발명에 따른 이미지 인코더에서 사용되기 위한 인코딩 디바이스(1500)의 실시예를 도시한다.
 - 도 16은 콘텐츠 전송 서비스를 실현하는 콘텐츠 공급 시스템(3100)의 예시적인 구조를 도시하는 블록도이다.
 - 도 17은 단말 디바이스의 예시의 구조를 도시하는 블록도이다.
- 아래에서, 동일한 참조 부호는, 달리 명시적으로 지정되지 않는 한, 동일하거나 적어도 기능적으로 동등한 특징을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0041] 다음의 설명에서, 첨부된 도면을 참조하며, 이는 본 발명의 일부를 형성하고, 도시를 통해, 본 발명의 실시예의 특정 측면 또는 본 개시의 실시예가 사용될 수 있는 특정 측면을 보여준다. 본 발명의 실시예는 다른 측면에서 사용될 수 있고 도면에 도시되지 않은 구조적 또는 논리적 변경을 포함할 수 있음이 이해된다. 따라서, 다음의 상세한 설명은 제한적인 의미로 받아들여서는 안되며, 본 발명의 범위는 첨부된 청구 범위에 의해 정의된다.

- [0042] 예를 들어, 설명된 방법과 관련된 개시는 방법을 수행하도록 구성되는 대응하는 장치 또는 시스템에 대해 또한 참일 수 있고, 그 반대의 경우도 마찬가지인 것으로 이해된다. 예를 들어, 하나 또는 복수의 특정 방법 단계가 설명되면, 대응하는 장치는 설명된 하나 또는 복수의 방법 단계를 수행하는 하나 또는 복수의 유닛, 예컨대 기능적 유닛(예컨대, 하나 또는 복수의 단계를 수행하는 하나의 유닛, 또는 상기 복수의 단계 중 하나 이상을 각각 수행하는 복수의 유닛)을, 그러한 하나 이상의 유닛이 명시적으로 기술되지 않거나 도면에 도시되지 않더라도, 포함할 수 있다. 한편, 예를 들어, 특정 장치가 하나 또는 복수의 유닛, 예컨대 기능적 유닛에 기초하여 설명된 경우, 대응하는 방법은, 하나 또는 복수의 유닛의 기능을 수행하는 하나의 단계(예컨대, 하나 또는 복수의 유닛의 기능을 수행하는 하나의 단계, 또는 상기 복수의 유닛 중 하나 이상의 기능을 각각 수행하는 복수의 단계)를, 그러한 하나 또는 복수의 단계가 명시적으로 기술되지 않거나 도면에 도시되지 않더라도, 포함할 수 있다. 또한, 특별히 달리 언급되지 않는 한, 여기에서 설명된 다양한 실시예 및/또는 측면의 특징은 서로 조합될 수 있다는 것이 이해된다.
- [0043] 비디오 코딩은 일반적으로 비디오 또는 비디오 시퀀스를 형성하는, 화상의 시퀀스를 프로세싱하는 것을 지칭한다. 용어 "화상" 대신에 용어 "프레임" 또는 "이미지"가 비디오 코딩 분야에서 동의어로 사용될 수 있다. 비디오 코딩(또는 일반적으로 코딩)은 비디오 인코딩 및 비디오 디코딩의 두 부분을 포함한다. 비디오 인코딩은, (더 효율적인 저장 및/또는 전송을 위해) 비디오 화상을 표현하는데 요구되는 데이터의 양을 줄이기 위해 (예를 들어, 압축에 의해) 원본 비디오 화상을 처리하는 것을 통상적으로 포함하여, 소스 측에서 수행된다. 비디오 디코딩은 테스트베이션 측에서 수행되고 일반적으로 비디오 화상을 재구성하기 위해 인코더에 대비한 역처리를 포함한다. 비디오 화상(또는, 일반적으로 화상)의 코딩을 참조하는 실시예는 각각의 비디오 시퀀스에 대한 비디오 화상의 "인코딩" 또는 "디코딩"에 관한 것으로 이해되어야 한다. 인코딩 부분과 디코딩 부분의 조합은 CODEC(Coding and Decoding)으로 또한 지칭된다.
- [0044] 무손실 비디오 코딩의 경우, 원본 비디오 화상을 재구성할 수 있으며, 즉, 재구성된 비디오 화상은 (저장 또는 전송 중에 전송 손실이나 기타 데이터 손실이 없다고 가정하면) 원본 비디오 화상과 동일한 품질을 가진다. 손실 비디오 코딩의 경우, 예컨대 양자화에 의해, 추가 압축이 수행되어, 비디오 화상을 표현하는 데이터의 양을 줄여, 디코더에서 완전히 재구성될 수 없게 되며, 즉, 재구성된 비디오 화상의 품질이 원본 비디오 화상의 품질에 비해 낮거나 나쁘다.
- [0045] 여러 비디오 코딩 표준은, "손실 하이브리드 비디오 코덱(lossy hybrid video codecs)"의 그룹에 속한다(즉, 샘플 도메인에서의 공간적 및 시간적 예측과 변환 도메인에서 양자화를 적용하기 위한 2D 변환 코딩을 조합한다). 비디오 시퀀스의 각각의 화상은 일반적으로 중첩되지 않는 블록의 세트로 분할되고, 코딩은 일반적으로 블록 레벨에서 수행된다. 달리 말하면, 인코더에서는, 비디오는 통상적으로 블록(비디오 블록) 레벨에서 예컨대 공간적(인트라 화상) 예측 및/또는 시간적(인터 화상) 예측을 사용하여 처리 즉 인코딩되어, 예측 블록을 생성하고, 현재 블록(현재 처리되는/처리될 블록)으로부터 예측 블록을 차감하여 잔차 블록을 획득하고, 잔차 블록을 변환하고 변환 도메인에서 잔차 블록을 양자화하여 전송될 데이터의 양을 줄이며(압축), 한편 디코더에서는, 인코더에 대비해 역처리가 인코딩된 또는 압축된 블록에 적용되어, 현재 블록을 표현하기 위해 재구성한다. 또한, 인코더는 디코더 처리 루프를 복제하여, 프로세싱, 즉, 코딩, 후속 블록을 처리, 즉 코딩하기 위해 동일한 예측(예컨대 인트라- 및 인터 예측) 및/또는 재구성을 생성할 것이다.
- [0046] 비디오 코딩 시스템(10)의 다음 실시예에서, 비디오 인코더(20) 및 비디오 디코더(30)가 도 1 내지 도 3에 기초하여 설명된다.
- [0047] 도 1a는 본 출원의 기법을 활용할 수 있는 예시적인 코딩 시스템(10), 예컨대 비디오 코딩 시스템(10)(또는 간단히 코딩 시스템(10))을 도시하는 개략적인 블록도이다. 비디오 코딩 시스템(10)의 비디오 인코더(20)(또는 간단히 인코더 20) 및 비디오 디코더(30)(또는 간단히 디코더(30))는 본 출원에서 기술되는 다양한 예시에 따른 기법을 수행하도록 구성될 수 있는 디바이스의 예시를 나타낸다.
- [0048] 도 1a에 도시된 바와 같이, 코딩 시스템(10)은, 인코딩된 화상 데이터(13)의 디코딩을 위해 인코딩된 화상 데이터(21)을 예컨대 테스트베이션 디바이스(14)로 제공하도록 구성되는 소스 디바이스(12)를 포함한다.
- [0049] 소스 디바이스(12)는 인코더(20)를 포함하고, 추가적으로, 즉, 선택사항으로, 화상 소스(16), 전처리기(또는 전처리 유닛)(18), 예컨대 화상 전처리기(18), 및 통신 인터페이스 또는 통신 유닛(22)을 포함할 수 있다.
- [0050] 화상 소스(16)는, 임의의 종류의 화상 캡처 디바이스, 일례로 실제 화상을 캡처하는 카메라, 및/또는 임의의 종류의 화상 생성 디바이스, 일례로 컴퓨터 애니메이션 화상을 생성하는 컴퓨터-그래픽 프로세서, 또는 실제

화상, 컴퓨터 생성 화상(예컨대 스크린 콘텐츠, 가상 현실(VR) 화상) 및/또는 이들의 임의의 조합(예컨대 증강 현실(AR) 화상)을 획득 및/또는 제공하기 위한 임의의 종류의 다른 디바이스이거나 이를 포함할 수 있다. 화상 소스는 전술한 화상 중 어느 하나를 저장하는 임의의 종류의 메모리 또는 스토리지일 수 있다.

- [0051] 전처리기(18) 및 전처리 유닛(18)에 의해 수행되는 처리와 구별하여, 화상 또는 화상 데이터(17)는 또한 원시(raw) 화상 또는 원시 화상 데이터(17)로 지칭될 수 있다.
- [0052] 전처리기(18)는 (원시) 화상 데이터(17)를 수신하고, 화상 데이터(17)에 대해 전처리를 수행하여 전처리된 화상(19) 또는 전처리된 화상 데이터(19)를 획득하도록 구성된다. 전처리기(18)에 의해 수행되는 전처리는 예를 들어 트리밍(trimming), 컬러 포맷 변환(예를 들어, RGB로부터 YCbCr로), 컬러 보정(color correction) 또는 노이즈 제거(de-noising)를 포함할 수 있다. 전처리 유닛(18)은 선택적 컴포넌트일 수 있음을 이해할 수 있다.
- [0053] 비디오 인코더(20)는 전처리된 화상 데이터(19)를 수신하고, 인코딩된 화상 데이터(21)를 제공하도록 구성된다 (자세한 내용은 예를 들어, 도 2에 기초하여 아래에서 설명될 것이다).
- [0054] 소스 디바이스(12)의 통신 인터페이스(22)는 인코딩된 화상 데이터(21)를 수신하고, 통신 채널(13)을 통해 인코딩된 화상 데이터(21)(또는 그의 추가 처리된 버전)를 다른 디바이스, 예를 들어, 저장 또는 직접 재구성성을 위한, 테스트네이션 디바이스(14) 또는 임의의 다른 디바이스로 전송하도록 구성될 수 있다.
- [0055] 테스트네이션 디바이스(14)는 디코더(30)(예를 들어, 비디오 디코더(30))를 포함하고, 추가적으로, 즉 선택적으로 통신 인터페이스 또는 통신 유닛(28), 후처리기(post-processor)(32)(또는 후처리 유닛(32)) 및 디스플레이 디바이스(34)를 포함할 수 있다.
- [0056] 테스트네이션 디바이스(14)의 통신 인터페이스(28)는 예를 들어 소스 디바이스(12)로부터 직접 또는 임의의 다른 소스, 예를 들어, 저장 디바이스, 예를 들어, 인코딩된 화상 데이터 저장 디바이스로부터, 인코딩된 화상 데이터(21)(또는 그것의 추가 처리된 버전)를 수신하고, 인코딩된 화상 데이터(21)를 디코더(30)에 제공하도록 구성된다.
- [0057] 통신 인터페이스(22) 및 통신 인터페이스(28)는 소스 디바이스(12)와 테스트네이션 디바이스(14) 사이의 직접 통신 링크, 예를 들어 직접 유선 또는 무선 연결을 통해, 또는 임의의 종류의 네트워크, 예를 들어 유선 또는 무선 네트워크 또는 이들의 임의의 조합, 또는 임의의 종류의 사설 및 공용 네트워크, 또는 이들의 임의의 종류의 조합을 통해, 인코딩된 화상 데이터(21) 또는 인코딩된 데이터(13)를 전송 또는 수신하도록 구성될 수 있다.
- [0058] 통신 인터페이스(22)는 예를 들어, 인코딩된 화상 데이터(21)를 적절한 포맷, 예를 들어 패킷으로 패키징하거나, 및/또는 임의의 종류의 전송 인코딩 또는 통신 링크 또는 통신 네트워크를 통한 전송을 위한 처리를 사용하여, 인코딩된 화상 데이터를 처리하도록 구성될 수 있다.
- [0059] 통신 인터페이스(22)의 대응물을 형성하는 통신 인터페이스(28)는 예를 들어, 전송된 데이터를 수신하고 임의의 종류의 대응하는 전송 디코딩 또는 처리 및/또는 디패키징(de-packaging)을 사용하여 전송 데이터를 처리하여 인코딩된 화상 데이터(21)를 획득하도록 구성될 수 있다.
- [0060] 통신 인터페이스(22) 및 통신 인터페이스(28) 모두는 도 1a에서 소스 디바이스(12)에서 테스트네이션 디바이스(14)를 가리키는 통신 채널(13)에 대한 화살표에 의해 지시된 바와 같은 단방향 통신 인터페이스 또는 양방향 통신 인터페이스로서 구성될 수 있으며, 예를 들어, 메시지를 송신 및 수신하여, 예를 들어, 연결을 설정하여, 통신 링크 및/또는 데이터 전송, 예를 들어 인코딩된 화상 데이터 전송에 관련된 기타 정보를 확인 및 교환하도록 구성될 수 있다.
- [0061] 디코더(30)는 인코딩된 화상 데이터(21)를 수신하도록 그리고 디코딩된 화상 데이터(31) 또는 디코딩된 화상(31)(예컨대, 도 3 또는 도 5에 기초하여 아래에서 더욱 상세히 설명된다)을 제공하도록 구성된다.
- [0062] 테스트네이션 디바이스(14)의 후처리기(32)는 디코딩된 화상 데이터(31)(또한 재구성된 화상 데이터라고도 함), 예를 들어, 디코딩된 화상 데이터(33)를 후처리하여, 후처리된 화상 데이터(33), 예를 들어, 후처리된 화상(33)을 획득하도록 구성된다. 후처리 유닛(32)에 의해 수행되는 후처리는 예를 들어, 컬러 포맷 변환(예를 들어, YCbCr로부터 RGB로), 컬러 보정, 트리밍 또는 재샘플링(re-sampling), 또는 임의의 다른 처리, 예를 들어 디스플레이 디바이스(34)에 의한 디스플레이를 위해 디코딩된 화상 데이터(31)를 준비하는 것을 포함할 수 있다.
- [0063] 테스트네이션 디바이스(14)의 디스플레이 디바이스(34)는 예를 들어 사용자 또는 뷰어에게 화상을 디스플레이하기 위해 후처리된 화상 데이터(33)를 수신하도록 구성된다. 디스플레이 디바이스(34)는 재구성된 화상을 표현하

기 위한 임의의 종류의 디스플레이, 예를 들어 통합 또는 외부 디스플레이 또는 모니터이거나 이를 포함할 수 있다. 디스플레이는 예를 들어 액정 디스플레이(liquid crystal display, LCD), 유기 발광 다이오드(organic light emitting diode, OLED) 디스플레이, 플라즈마 디스플레이, 프로젝터, 마이크로 LED 디스플레이, 액정 온 실리콘(liquid crystal on silicon, LCoS), 디지털 광 프로세서(digital light processor, DLP) 또는 모든 종류의 다른 디스플레이를 포함할 수 있다.

[0064] 비록 도 1a는 소스 디바이스(12) 및 테스트네이션 디바이스(14)를 별개의 디바이스로서 도시하지만, 디바이스의 실시예는 또한 둘 또는 둘 모두의 기능, 소스 디바이스(12) 또는 대응하는 기능 및 테스트네이션 디바이스(14) 또는 대응하는 기능을 포함할 수 있다. 이러한 실시예에서, 소스 디바이스(12) 또는 대응하는 기능 및 테스트네이션 디바이스(14) 또는 대응하는 기능은, 동일한 하드웨어 및/또는 소프트웨어를 사용하거나 별개의 하드웨어 및/또는 소프트웨어 또는 이들의 임의의 조합에 의해 구현될 수 있다.

[0065] 설명에 기초하여 당업자에게 명백한 바와 같이, 도 1a에 도시된 바와 같이 소스 디바이스(12) 및/또는 테스트네이션 디바이스(14) 내의 상이한 유닛의 기능 또는 기능의 존재 및 (정확한) 분할(split)은 실제 디바이스 및 애플리케이션에 따라 다를 수 있다.

[0066] 인코더(20)(예: 비디오 인코더(20)) 또는 디코더(30)(예: 비디오 디코더(30)) 혹은 이들 모두는 하나 이상의 마이크로 프로세서, 디지털 신호 프로세서(digital signal processor, DSP), 주문형 집적 회로(application-specific integrated circuit, ASIC), 펠드 프로그래밍 가능 게이트 어레이(field-programmable gate array, FPGA), 이산 로직, 하드웨어 또는 이들의 조합과 같은, 도 1b에 도시된 바와 같이 다양한 적절한 회로 중 임의의 것으로 구현될 수 있다. 인코더(20)는 처리 회로(46)를 통해 구현될 수 있어, 도 2의 인코더(20)와 관련하여 논의되는 다양한 모듈 및/또는 여기에서 기술되는 임의의 다른 인코더 시스템 또는 서브시스템을 구현할 수 있다. 디코더(30)는 처리 회로(46)를 통해 구현될 수 있어, 도 3의 디코더(30)와 관련하여 논의되는 다양한 모듈 및/또는 여기에서 기술되는 임의의 다른 디코더 시스템 또는 서브시스템을 구현할 수 있다. 처리 회로는 후에 논의되는 다양한 동작을 수행하도록 구성될 수 있다. 도 5에 도시된 바와 같이, 기술이 부분적으로 소프트웨어로 구현되는 경우, 디바이스는 적절한 비일시적 컴퓨터-판독 가능한 저장 매체에 소프트웨어에 대한 명령을 저장할 수 있고, 하나 이상의 프로세서를 사용하여 하드웨어에서 명령을 실행하여 본 개시의 방법을 수행할 수 있다. 비디오 인코더(20) 및 비디오 디코더(30) 중 어느 하나는 예를 들어 도 1b에 도시된 바와 같이 단일 디바이스에서 결합된 인코더/디코더(CODEC)의 일부로서 통합될 수 있다.

[0067] 소스 디바이스(12) 및 테스트네이션 디바이스(14)는 임의의 종류의 핸드헬드 또는 고정 디바이스, 예를 들어 노트북 또는 랩톱 컴퓨터, 휴대폰, 스마트 폰, 태블릿 또는 태블릿 컴퓨터, 카메라, 데스크톱 컴퓨터, 셋탑 박스, 텔레비전, 디스플레이 디바이스, 디지털 미디어 플레이어, 비디오 게임 콘솔, 비디오 스트리밍 디바이스(예: 콘텐츠 서비스 서버 또는 콘텐츠 전달 서버), 방송 수신기 디바이스, 방송 송신기 디바이스 등을 포함하는, 광범위한 디바이스를 포함할 수 있으며, 임의의 운영 체제를 사용하지 않거나 또는 사용할 수 있다. 일부 경우에, 소스 디바이스(12) 및 테스트네이션 디바이스(14)는 무선 통신을 장착할 수 있다. 따라서, 소스 디바이스(12) 및 테스트네이션 디바이스(14)는 무선 통신 디바이스일 수 있다.

[0068] 일부 경우에, 도 1a에 도시된 비디오 코딩 시스템(10)은 단지 예일 뿐이고, 본 출원의 기술은 인코딩 디바이스와 디코딩 디바이스 사이의 임의의 데이터 통신을 반드시 포함하지 않는 비디오 코딩 설정(예: 비디오 인코딩 또는 비디오 디코딩)에 적용될 수 있다. 다른 예에서, 데이터가 로컬 메모리로부터 검색되고, 네트워크를 통해 스트리밍된다. 비디오 인코딩 디바이스는 데이터를 인코딩하고 메모리에 저장할 수 있거나, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 검색하고 디코딩할 수 있다. 일부 예에서, 인코딩 및 디코딩은 서로 통신하지 않고 단순히 데이터를 메모리로 인코딩하고 및/또는 메모리로부터 데이터를 검색 및 디코딩하는 디바이스에 의해 수행된다.

[0069] 설명의 편의를 위해, 본 개시의 실시예는 예를 들어, HEVC(High-Efficiency Video Coding) 또는 ITU-T VCEG(Video Coding Experts Group) 및 ISO/IEC MPEG(Motion Picture Experts Group)의 JCT-VC(Joint Collaboration Team on Video Coding)에 의해 개발된 차세대 비디오 코딩 표준인, VVC(Versatile Video Coding)의 참조 소프트웨어를 참조하여 여기에 설명된다. 당업자는 본 개시의 실시예가 HEVC 또는 VVC에 제한되지 않음을 이해할 것이다.

[0070] 인코더 및 인코딩 방법

[0071] 도 2는 본 출원의 기법을 구현하도록 구성되는 예시적인 비디오 인코더(20)의 개략적 블록도를 도시한다. 도 2

의 예에서, 비디오 인코더(20)는, 입력(201)(또는 입력 인터페이스(201)), 잔차 계산 유닛(204), 변환 처리 유닛(206), 양자화 유닛(208), 역 양자화 유닛(210), 및 역변환 처리 유닛(212), 재구성 유닛(214), 루프 필터 유닛(220), 디코딩된 화상 버퍼(DPB)(230), 모드 선택 유닛(260), 엔트로피 인코딩 유닛(270) 및 출력(272)(또는 출력 인터페이스(272))을 포함한다. 모드 선택 유닛(260)은 인터 예측 유닛(244), 인트라 예측 유닛(254) 및 파티셔닝 유닛(262)을 포함할 수 있다. 인터 예측 유닛(244)은 모션 추정(motion estimation) 유닛 및 모션 보상(motion compensation) 유닛(도시하지 않음)을 포함할 수 있다. 도 2에 도시된 바와 같은 비디오 인코더(20)는 하이브리드 비디오 인코더 또는 하이브리드 비디오 코덱에 따른 비디오 인코더로도 또한 지칭될 수 있다.

[0072] 잔차 계산 유닛(204), 변환 처리 유닛(206), 양자화 유닛(208), 모드 선택 유닛(260)은 인코더(20)의 순방향 신호 경로를 형성하는 것으로 지칭될 수 있는 반면, 역 양자화 유닛(210), 역 변환 처리 유닛(212), 재구성 유닛(214), 버퍼(216), 루프 필터(220), 디코딩된 화상 버퍼(DPB)(230), 인터 예측 유닛(244) 및 인트라 예측 유닛(254)은 비디오 인코더의 역방향 신호 경로를 형성하는 것으로 지칭될 수 있으며, 여기서 비디오 인코더(20)의 역방향 신호 경로는 디코더의 신호 경로에 대응한다(도 3의 비디오 디코더(30) 참조). 역 양자화 유닛(210), 역 변환 처리 유닛(212), 재구성 유닛(214), 루프 필터(220), 디코딩 화상 버퍼(DPB)(230), 인터 예측 유닛(244) 및 인트라 예측 유닛(254)도 비디오 인코더(20)의 "내장 디코더(built-in decoder)"를 형성하는 것으로 지칭될 수 있다.

[0073] 화상 & 화상 파티셔닝(화상 & 블록)

[0074] 인코더(20)는 예를 들어 입력(201)을 통해, 화상(17)(또는 화상 데이터(17)), 예를 들어 비디오 또는 비디오 시퀀스를 형성하는 화상 시퀀스의 화상을 수신하도록 구성될 수 있다. 수신된 화상 또는 화상 데이터는 또한 전처리된 화상(19)(또는 전처리된 화상 데이터(19))일 수 있다. 단순화를 위해 다음 설명은 화상(17)을 참조한다. 화상(17)은 또한 (특히, 현재 화상을 다른 화상, 예를 들어, 동일한 비디오 시퀀스, 즉 현재 화상도 포함하는 비디오 시퀀스의 이전에 인코딩 및/또는 디코딩된 화상과 구별하기 위한 비디오 코딩에서) 코딩될 현재 화상 또는 화상으로 지칭될 수 있다.

[0075] (디지털) 화상은 세기(intensity) 값이 있는 샘플들의 2차원 어레이 또는 행렬이거나 이들로 간주될 수 있다. 어레이의 샘플은 픽셀(pixel)(화상 엘리먼트의 짧은 형식) 또는 펠(pel)이라고도 한다. 어레이 또는 화상의 수평 및 수직 방향(또는 축)의 샘플 수는 화상의 크기 및/또는 해상도를 정의한다. 컬러 표현을 위해, 일반적으로 3가지 컬러 컴포넌트가 사용된다. 즉, 화상이 3개의 샘플 어레이로 표현되거나 이를 포함할 수 있다. RGB 형식 또는 색 공간에서 화상은 대응하는 빨강, 녹색 및 파랑 샘플 어레이로 구성된다. 그러나, 비디오 코딩에서 각각의 픽셀은 일반적으로 휘도(luminance) 및 채도(chrominance) 포맷 또는 색 공간, 예를 들어 Y로 지시되는 휘도 컴포넌트(때로는 L이 대신 사용됨)와 Cb 및 Cr로 지시되는 두개의 채도 컴포넌트를 포함하는 YCbCr로 표현된다. 휘도(또는 간단히, 루마(luma)) 컴포넌트 Y는 밝기 또는 그레이 레벨 강도(예컨대, 그레이 스케일 화상에서와 같이)를 나타내는 반면, 2개의 채도(또는 간단히 크로마(chroma)) 컴포넌트 Cb 및 Cr은 채도 또는 색상 정보 컴포넌트를 나타낸다. 따라서, YCbCr 포맷의 화상은 휘도 샘플 값(Y)의 휘도 샘플 어레이와 채도 값(Cb 및 Cr)의 2개의 채도 샘플 어레이로 구성된다. RGB 포맷의 화상은 YCbCr 포맷으로 컨버전(conversion) 또는 변환될 수 있고 그 반대의 경우도 마찬가지이며, 이 프로세스는 색상 변환 또는 색상 컨버전이라고도 한다. 화상이 단색이면, 화상은 휘도 샘플 어레이만 포함할 수 있다. 따라서, 화상은 예를 들어 단색 포맷의 루마 샘플 어레이 또는 4:2:0, 4:2:2 및 4:4:4 컬러 포맷에서의 루마 샘플 어레이와 2개의 대응하는 크로마 샘플 어레이일 수 있다.

[0076] 비디오 인코더(20)의 실시예는 화상(17)을 복수의 (일반적으로 비-중첩) 화상 블록(203)으로 파티셔닝하도록 구성된 화상 파티셔닝 유닛(도 2에 도시되지 않음)을 포함할 수 있다. 이러한 블록은 루트(root) 블록, 매크로 블록(H.264/AVC) 또는 코딩 트리 블록(coding tree block, CTB) 또는 코딩 트리 유닛(coding tree unit, CTU)(H.265/HEVC 및 VVC)이라고도 한다. 화상 파티셔닝 유닛은 비디오 시퀀스의 모든 화상 및 블록 크기를 규정하는 대응하는 그리드에 대해 동일한 블록 크기를 사용하도록, 또는 화상 또는 화상의 서브세트 또는 그룹 사이에서 블록 크기를 변경하고, 각각의 화상을 대응하는 블록으로 파티셔닝하도록, 구성될 수 있다.

[0077] 추가적인 실시예에서, 비디오 인코더는 화상(17)의 블록(203), 예를 들어 화상(17)을 형성하는 하나, 여러 또는 모든 블록을 직접 수신하도록 구성될 수 있다. 화상 블록(203)은 또한 코딩될 화상 블록 또는 현재 화상 블록으로 지칭될 수 있다.

[0078] 화상(17)과 마찬가지로, 화상 블록(203)도 또한, 화상(17)보다는 작은 크기이지만, 세기 값(샘플 값)을 가지는 샘플의 행렬 또는 2차원 어레이이거나 이로 간주될 수 있다. 달리 말하면, 블록(203)은, 예를 들어, 하나의 샘플

플 어레이(예를 들어, 단색 화상(17)의 경우 루마 어레이, 또는 컬러 화상의 경우 루마 또는 크로마 어레이), 3개의 샘플 어레이(예를 들어, 컬러 화상(17)의 경우 하나의 루마 어레이 및 2개의 크로마 어레이), 또는 적용된 컬러 포맷에 따라 임의의 다른 수 및/또는 종류의 어레이를 포함할 수 있다. 블록(203)의 수평 및 수직 방향(또는 축)의 샘플 수량은 블록(203)의 크기를 규정한다. 따라서, 블록은, 예를 들어, 샘플의 $M \times N$ (M -열 및 N -행) 어레이, 또는 변환 계수의 $M \times N$ 어레이일 수 있다.

[0079] 도 2에 도시된 비디오 인코더(20)의 실시예는 화상(17)을 블록별로 인코딩하도록 구성될 수 있으며, 예를 들어, 인코딩 및 예측은 블록(203)마다 수행된다.

[0080] 도 2에 도시된 비디오 인코더(20)의 실시예는 슬라이스(비디오 슬라이스라고도 함)를 사용하여 픽처를 파티션 및/또는 인코딩하도록 추가로 구성될 수 있으며, 여기서 픽처는 하나 이상의 슬라이스(전형적으로 비-중첩), 각 슬라이스는 하나 이상의 블록(예컨대 CTU)을 포함할 수 있다.

[0081] 도 2에 도시된 비디오 인코더(20)의 실시예는 타일 그룹(비디오 타일 그룹이라고도 함) 및/또는 타일(비디오 타일이라고도 함)을 사용하여 픽처를 파티셔닝 및/또는 인코딩하도록 추가로 구성될 수 있다. 픽처는 하나 이상의 타일 그룹(일반적으로 비-중첩)을 사용하여 파티셔닝되거나 인코딩될 수 있으며, 각 타일 그룹은 예를 들어 하나 이상의 블록(예컨대 CTU) 또는 하나 이상의 타일을 포함할 수 있고, 각 타일은, 예컨대 직사각형 모양일 수 있으며 하나 이상의 블록(예컨대 CTU), 예컨대 완전하거나 분수로 나뉘어진 블록을 포함할 수 있다.

[0082] 잔차 계산

[0083] 잔차 계산 유닛(204)은, 화상 블록(203) 및 예측 블록(265)(예측 블록(265)에 대한 더욱 상세한 내용은 추후 제공된다)에 기초하여, 예를 들어, 화상 블록(203)의 샘플 값으로부터 예측 블록(265)의 샘플 값을 차감함으로써 잔차 블록(205)을 계산하여, 샘플 단위로(픽셀 단위로) 샘플 도메인에서 잔차 블록(205)(잔차(205)로도 지칭됨)을 획득하도록 구성될 수 있다.

[0084] 변환

[0085] 변환 처리 유닛(206)은, 잔차 블록(205)의 샘플 값에 변환을, 예를 들어, 이산 코사인 변환(DCT) 또는 이산 사인 변환(DST)을 적용하여, 변환 도메인에서 변환 계수(207)를 획득하도록 구성될 수 있다. 변환 계수(207)는 변환 잔차 계수로도 또한 지칭될 수 있고, 변환 도메인에서 잔차 블록(205)을 표현할 수 있다.

[0086] 변환 처리 유닛(206)은, H.265/HEVC에 대해 지정된 변환과 같은, DCT/DST의 정수 근사를 적용하도록 구성될 수 있다. 직교 DCT 변환과 비교하여, 이러한 정수 근사는 일반적으로 특정 팩터에 의해 스케일링된다. 순방향 및 역 변환에 의해 처리되는 잔차 블록의 놴(norm)을 보존하기 위해, 추가적인 스케일 팩터가 변환 프로세스의 일부로서 적용된다. 스케일링 팩터는 통상적으로, 시프트 연산에 대해 2의 거듭제곱, 변환 계수의 비트 심도, 정확도 및 구현 비용 사이의 절충, 등, 특정 제약 조건에 기초하여 선택된다. 특정 스케일링 팩터는, 예를 들어, 역 변환 처리 유닛(212)에 의한 역 변환(및 예를 들어 비디오 디코더(30)의 역 변환 처리 유닛(312)에 의한 대응하는 역 변환)에 대해 지정되고, 예를 들어, 인코더(20)에서의 변환 처리 유닛(206)에 의한, 순방향 변환에 대응하는 스케일링 팩터가 지정될 수 있다.

[0087] 비디오 인코더(20)(각각 변환 처리 유닛(206))의 실시예는, 예를 들어, 직접 또는 엔트로피 인코딩 유닛(270)을 통해 인코딩 또는 압축된, 예를 들어, 변환 또는 변환들의 유형과 같은 변환 파라미터를 출력하도록 구성될 수 있으므로, 비디오 디코더(30)는 디코딩을 위해 변환 파라미터를 수신하고 사용할 수 있다.

[0088] 양자화

[0089] 양자화 유닛(208)은, 예를 들어, 스칼라 양자화 또는 벡터 양자화를 적용하여, 변환 계수(207)를 양자화하여 양자화된 계수(209)를 획득하도록 구성될 수 있다. 양자화된 계수(209)는 양자화된 변환 계수(209) 또는 양자화된 잔차 계수(209)로도 또한 지칭될 수 있다.

[0090] 양자화 프로세스는 변환 계수(207)의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수 있다. 예를 들어, n -비트 변환 계수는 양자화 동안 m -비트 변환 계수로 반올림될 수 있으며, 여기서 n 은 m 보다 크다. 양자화의 정도는 양자화 파라미터(QP)를 조정함으로써 수정될 수 있다. 예를 들어, 스칼라 양자화의 경우, 더 미세하거나 거친 양자화를 달성하기 위해 다른 스케일링을 적용할 수 있다. 더 작은 양자화 단계 크기는 더 세밀한 양자화에 해당하고, 더 큰 양자화 단계 크기는 더 거친 양자화에 해당한다. 적용 가능한 양자화 단계 크기는 양자화 파라미터(QP)로 표시될 수 있다. 양자화 파라미터는, 예를 들어, 적용 가능한 양자화 단계 크기의 미리 정의된 세트에 대한 인덱스일 수 있다. 예를 들어, 작은 양자화 파라미터는 세밀한 양자화(더 작은 양자화 단계 크기)에 대응

할 수 있고 큰 양자화 파라미터는 거친 양자화(더 큰 양자화 단계 크기)에 대응할 수 있으며, 그 반대일 수 있다. 양자화는 양자화 단계 크기에 의한 나눗셈과, 예를 들어 역 양자화(210)에 의해 수행되는 역 양자화 및/또는 대응하는 양자화를 포함할 수 있거나, 또는 양자화 단계 크기에 의한 곱셈을 포함할 수 있다. 예컨대 HEVC의, 일부 표준에 따른 실시에는, 양자화 단계 크기를 결정하는데 양자화 파라미터를 사용하도록 구성될 수 있다. 일반적으로, 양자화 단계 크기는 나누기를 포함하는 방정식의 고정 소수점 근사를 사용하여 양자화 파라미터에 기초하여 계산될 수 있다. 잔차 블록의 놈(norm)을 복원하기 위해 추가적인 스케일링 팩터가 양자화 및 역양자화에 도입될 수 있고, 여기서 잔차 블록의 놈(norm)은 양자화 파라미터 및 양자화 단계 크기에 대한 수학적 고정 소수점 근사에서 사용되는 스케일링으로 인해 수정될 수 있다. 하나의 예시적인 구현에서, 역변환 및 역양자화의 스케일링이 조합될 수 있다. 대안적으로, 맞춤형 양자화 테이블이 사용되어 인코더로부터 디코더로, 예를 들어 비트스트림으로 시그널링될 수 있다. 양자화는 손실이 있는 연산으로서, 양자화 단계 크기가 증가함에 따라 손실이 증가한다.

[0091] 비디오 인코더(20)(각각 양자화 유닛(208))의 실시 예는, 예를 들어 직접 또는 엔트로피 인코딩 유닛(270)을 통해 인코딩된 양자화 파라미터(QP)를 출력하도록 구성될 수 있으므로, 예를 들어, 비디오 디코더(30)는 디코딩을 위한 양자화 파라미터를 수신하고 적용할 수 있다.

[0092] 역 양자화

[0093] 역 양자화 유닛(210)은, 예를 들어, 양자화 유닛(208)과 동일한 양자화 단계 크기에 기초하거나 사용하여 양자화 유닛(208)에 의해 적용되는 양자화 방식의 역을 적용함으로써, 양자화 유닛(208)의 역 양자화를 양자화된 계수에 적용하여 역양자화된 계수(211)를 획득하도록 구성된다. 역양자화된 계수(211)는 역양자화된 잔차 계수(211)로도 또한 지칭될 수 있고, 통상적으로 양자화에 의한 손실 때문에 변환 계수에 동일하지는 않으나, 변환 계수(207)에 대응한다.

[0094] 역변환

[0095] 역 변환 처리 유닛(212)은 변환 처리 유닛(206)에 의해 적용된 변환의 역 변환, 예를 들어 역 이산 코사인 변환(inverse discrete cosine transform, DCT) 또는 역 이산 사인 변환(inverse discrete sine transform, DST) 또는 다른 역 변환을 적용하여, 샘플 도메인에서 재구성된 잔차 블록(213)(또는 대응하는 역양자화된 계수(213))를 획득하도록 구성된다. 재구성된 잔차 블록(213)은 또한 변환 블록(213)으로 지칭될 수 있다. 재구성된 잔차 블록(213)은 변환 블록(213)으로도 또한 지칭될 수 있다.

[0096] 재구성

[0097] 재구성 유닛(214)(예를 들어, 합산기(adder) 또는 합산기(summer)(214))은, 변환 블록(213)(즉, 재구성된 잔차 블록(213))을 예측 블록(265)에 더함으로써, 예를 들어, 재구성된 잔차 블록(213)의 샘플 값 및 예측 블록의 샘플 값(265)을 샘플 단위로 더함으로써, 샘플 도메인에서 재구성된 블록(215)을 획득하도록 구성된다.

[0098] 필터링

[0099] 루프 필터 유닛(220)(또는 간단히 "루프 필터"(220))은, 재구성된 블록(215)을 필터링하여 필터링된 블록(221)을 획득하도록, 또는 재구성된 샘플을 필터링하여 필터링된 샘플을 획득하도록 구성된다. 루프 필터 유닛은 예를 들어 픽셀 전환(pixel transition)을 평활화하거나(smooth) 비디오 품질을 향상하도록 구성된다. 루프 필터 유닛(220)은 디-블로킹 필터, 샘플-적응형 오프셋(SAO) 필터 또는 하나 이상의 다른 필터, 예컨대 양방향 필터, 적응형 루프 필터(ALF), 샤프닝(sharpening), 스무딩(smoothing) 필터 또는 협업(collaborative) 필터, 또는 이들의 임의의 조합과 같은 하나 이상의 루프 필터를 포함할 수 있다. 비록 루프 필터 유닛(220)은 도 2에서 인 루프 필터(in loop filter)로서 도시되어 있으나, 다른 구성에서는, 루프 필터 유닛(220)은 포스트 루프 필터(post loop filter)로서 구현될 수 있다. 필터링된 블록(221)은 필터링된 재구성된 블록(221)으로도 또한 지칭될 수 있다.

[0100] 비디오 인코더(20)(각각 루프 필터 유닛(220))의 실시예는, 예컨대, 디코더(30)가 디코딩을 위해 동일한 루프 필터 파라미터 또는 각각의 루프 필터를 수신하고 적용하도록, 예컨대, 직접 또는 엔트로피 인코딩 유닛(270)을 통해 인코딩하여, (샘플 적응형 오프셋 정보와 같은) 루프 필터 파라미터를 출력하도록 구성될 수 있다.

[0101] 디코딩된 화상 버퍼

[0102] 디코딩된 화상 버퍼(DPB)(230)는 비디오 인코더(20)가 비디오 데이터를 인코딩하는데 사용하기 위한 참조 화상(또는 일반적으로 화상) 데이터를 저장하는 메모리일 수 있다. DPB(230)는 동기식 DRAM(SDRAM), 자기 저장성

RAM(magneto-resistive RAM, MRAM), 저항성 RAM(resistive RAM, RRAM) 또는 기타 유형의 메모리 장치를 포함하는 동적 랜덤 액세스 메모리(DRAM)와 같은 다양한 메모리 장치로 형성될 수 있다. 디코딩된 화상 버퍼(DPB)(230)은 하나 이상의 필터링된 블록(221)을 저장하도록 구성될 수 있다. 디코딩된 화상 버퍼(230)는, 동일한 현재 화상 또는 상이한 화상, 예를 들어, 이전에 재구성된 화상의, 다른 이전에 필터링된 블록, 예를 들어, 이전에 재구성되고 필터링된 블록(221)을 저장하도록 추가로 구성될 수 있고, 완전한 이전에 재구성된, 즉, 디코딩된 화상(및 대응하는 참조 블록 및 샘플) 및/또는 부분적으로 재구성된 현재 화상(및 대응하는 참조 블록 및 샘플)을, 예를 들어, 인터 예측에 제공할 수 있다. 디코딩된 화상 버퍼(DPB)(230)는 또한 예를 들어, 재구성된 블록(215)이 루프 필터 유닛(220)에 의해 필터링되지 않으면, 하나 이상의 필터링되지 않은 재구성된 블록(215), 또는 일반적으로 필터링되지 않은 재구성된 샘플 또는 임의의 다른 추가 처리 재구성된 블록 또는 샘플의 버전을 저장하도록 구성될 수 있다.

[0103] 모드 선택 (파티셔닝 & 예측)

[0104] 모드 선택 유닛(260)은 파티셔닝 유닛(262), 인터-예측 유닛(244) 및 인트라-예측 유닛(254)을 포함하고, 원본 화상 데이터를, 예컨대 원본 블록(203)(현재 화상(17)의 현재 블록(203))을, 그리고 재구성된 화상 데이터를, 예컨대 동일한(현재) 화상의 및/또는 하나 또는 복수의 이전에 디코딩된 화상으로부터의, 예컨대 디코딩된 화상 버퍼(230) 또는 다른 버퍼(예컨대 라인 버퍼, 도시하지 않음)로부터의 필터링된 및/또는 필터링되지 않은 재구성된 샘플 또는 블록을 수신하거나 획득하도록 구성된다. 재구성된 화상 데이터는 예측 블록(265) 또는 예측자(predictor)(265)를 획득하기 위해 예측, 예를 들어 인터 예측 또는 인트라 예측을 위한 참조 화상 데이터로서 사용된다.

[0105] 모드 선택 유닛(260)은 현재 블록 예측 모드(파티션 없음을 포함) 및 예측 모드(예: 인트라 또는 인터 예측 모드)에 대한 파티셔닝을 결정 또는 선택하며, 대응하는 예측 블록(265)을 생성하도록 구성될 수 있으며, 예측 블록(265)은 잔차 블록(265)의 계산 및 재구성된 블록(215)의 재구성을 위해 사용된다.

[0106] 모드 선택 유닛(260)의 실시예는, (예컨대 모드 선택 유닛(260)에 의해 지원되거나 모드 선택 유닛(260)에 대해 사용 가능한 것들로부터), 최상 매칭 또는 다른 말로 최소 잔차(최소 잔차는 전송 또는 저장을 위한 더 나은 압축을 의미한다), 또는 최소 시그널링 오버헤드(최소 시그널링 오버헤드는 전송 또는 저장을 위한 더 나은 압축을 의미한다)를 제공하거나, 양자를 고려하거나 균형을 이루는, 파티셔닝 및 예측 모드를 선택하도록 구성될 수 있다. 모드 선택 유닛(260)은 레이트 왜곡 최적화(rate distortion optimization, RDO)에 기반하여 파티셔닝 및 예측 모드를 결정하도록, 즉, 최소 레이트 왜곡을 제공하는 예측 모드를 선택하도록 구성된다. 이 문맥에서 "최상", "최소", "최적(optimum)" 등과 같은 용어는 반드시 전체적인 "최상", "최소", "최적" 등을 의미하는 것은 아니지만, 임계 값을 초과하거나 아래로 떨어지는 값 또는 잠재적으로 "서브 최적 선택(sub-optimum selection)"으로 이어지지만 복잡성과 처리 시간을 감소시키는 기타 제약 조건과 같은 종료 또는 선택 기준의 충족을 의미한다.

[0107] 달리 말하면, 파티셔닝 유닛(262)은, 예를 들어, 쿼드-트리-파티셔닝(quad-tree-partitioning, QT), 2진 파티셔닝(binary partitioning, BT), 트리플-트리-파티셔닝(triple-tree-partitioning, TT), 또는 이들의 임의의 조합을 반복적으로 사용하여, 블록(203)을 더 작은 블록 파티션 또는 서브-블록(이는 다시 블록을 형성함)으로 파티셔닝하도록, 그리고 예를 들어, 블록 파티션 또는 서브-블록의 각각에 예측을 수행하도록, 구성될 수 있으며, 여기서 모드 선택은 파티셔닝된 블록(203)의 트리 구조의 선택을 포함하고 예측 모드는 블록 파티션 또는 서브-블록의 각각에 적용된다.

[0108] 다음에, 예시적인 비디오 인코더(20)에 의해 수행되는 파티셔닝(예: 파티셔닝 유닛(260)에 의한) 및 예측 처리(인터 예측 유닛(244) 및 인트라 예측 유닛(254)에 의한)가 더 상세히 설명될 것이다.

[0109] 파티셔닝

[0110] 파티셔닝 유닛(262)은 현재 블록(203)을 더 작은 파티션으로, 예를 들어 정사각형 또는 직사각형 크기의 작은 블록으로 파티셔닝(또는 분할)할 수 있다. 이러한 더 작은 블록(서브-블록이라고도 지칭됨)은 더 작은 파티션으로 더욱 분할될 수 있다. 이것은 또한 트리-파티셔닝 또는 계층적 트리-파티셔닝이라고도 하며, 여기서 예를 들어 루트 트리-레벨 0(계층화-레벨 0, 깊이 0)에서의 루트 블록은 재귀적으로 파티셔닝될 수 있으며, 예를 들어 다음 하위 트리(lower tree) 레벨, 예를 들어, 트리-레벨 1(계층화-레벨 1, 깊이 1)의 노드의 2개 이상의 블록으로 파티셔닝될 수 있으며, 여기서 이들 블록은 파티셔닝이 종료될 때까지, 예를 들어, 종료 기준이 충족되었기 때문에, 예를 들어 최대 트리 깊이 또는 최소 블록 크기에 도달할 때까지, 다시 다음 하위 레벨, 예를 들어,

트리-레벨 2(계층화-레벨 2, 깊이 2)의 2개 이상의 블록으로 파티셔닝될 수 있다. 더 이상 분할되지 않은 블록은 트리의 리프-블록 또는 리프-노드라고도 지칭된다. 2개의 파티션으로 분할하는 트리를 2진-트리(binary-tree, BT)라고 지칭하고, 3개의 파티션으로 분할하는 트리를 삼중 트리(ternary-tree, TT)라고 지칭하며, 4개의 파티션으로 분할하는 트리를 쿼드-트리(quad-tree, QT)라고 지칭한다.

[0111] 앞서 언급한 바와 같이, 여기서 사용되는 용어 "블록"은 화상의 일부, 특히 정사각형 또는 직사각형 부분일 수 있다. 예를 들어, HEVC 및 VVC를 참조하면, 블록은 코딩 트리 유닛(coding tree unit, CTU), 코딩 유닛(coding unit, CU), 예측 유닛(prediction unit, PU) 및 변환 유닛(transform unit, TU) 및/또는 대응하는 블록, 예를 들어 코딩 트리 블록(coding tree block, CTB), 코딩 블록(coding block, CB), 변환 블록(transform block, TB) 또는 예측 블록(prediction block, PB)이거나 이에 대응할 수 있다.

[0112] 예를 들어, 코딩 트리 유닛(CTU)은 루마 샘플의 CTB, 3개의 샘플 어레이를 갖는 화상의 크로마 샘플의 2개의 대응하는 CTB, 또는 단색 화상 또는 샘플을 코딩하는 데 사용되는 3개의 개별 컬러 평면과 신텍스 구조를 사용하여 코딩된 화상의 샘플의 CTB이거나 이들을 포함할 수 있다. 상응하여, 코딩 트리 블록(CTB)은 콤포넌트를 CTB로 나누는 것(division)이 파티셔닝이 되도록 N의 일부 값에 대한 샘플의 N×N 블록일 수 있다. 코딩 유닛(CU)은 루마 샘플의 코딩 블록, 3개의 샘플 어레이를 갖는 화상의 크로마 샘플의 2개의 대응하는 코딩 블록, 또는 단색 화상 또는 샘플을 코딩하는 데 사용되는 3개의 개별 컬러 평면과 신텍스 구조를 사용하여 코딩된 화상의 샘플의 코딩 블록이거나 이들을 포함할 수 있다. 상응하여, 코딩 블록(CB)은 CTB를 코딩블록으로 나누는 것(division)이 파티셔닝이 되도록 M과 N의 일부 값에 대한 샘플의 N×N 블록일 수 있다.

[0113] 실시예에서, 예를 들어 HEVC에 따르면, 코딩 트리 단위(CTU)는 코딩 트리노드 표시된 쿼드-트리 구조를 사용하여 CU로 분할될 수 있다. 인터-화상(시간적) 또는 인트라-화상(공간적) 예측을 사용하여 화상 영역을 코딩할지 여부의 결정은 CU 레벨에서 이루어진다. 각각의 CU는 PU 분할 유형에 따라 1개, 2개 또는 4개의 PU로 더 분할될 수 있다. 하나의 PU 내부에서, 동일한 예측 프로세스가 적용되고 관련 정보가 PU 기반으로 디코더로 전송된다. PU 분할 유형에 기초하여 예측 과정을 적용하여 잔차 블록을 획득한 후, CU는 CU에 대한 코딩 트리와 유사한 다른 쿼드-트리 구조에 따라 변환 유닛(TU)으로 파티셔닝될 수 있다.

[0114] 실시예에서, 예를 들어 VVC(Versatile Video Coding)라고 지칭되는 현재 개발중인 최신 비디오 코딩 표준에 따르면, 쿼드-트리 및 2진 트리(Quad-tree and binary tree, QTBT) 파티셔닝이 예를 들어 코딩 블록을 파티셔닝하는 데 사용된다. QTBT 블록 구조에서, CU는 정사각형 또는 직사각형 모양을 가질 수 있다. 예를 들어, 코딩 트리 유닛(CTU)은 먼저 쿼드트리 구조에 의해 파티셔닝된다. 쿼드트리 리프 노드는 2진 트리 또는 3중(ternary)(또는 3중(triple)) 트리 구조에 의해 추가로 파티셔닝된다. 파티셔닝 트리 리프 노드는 코딩 유닛(CU)이라고 하고, 이 세분화(segmentation)는 추가적인 파티셔닝 없이 예측 및 변환 처리에 사용된다. 이는 CU, PU 및 TU가 QTBT 코딩 블록 구조에서 동일한 블록 크기를 가짐을 의미한다. 병렬적으로, 다중 파티션, 예를 들어 트리플 트리 파티션이 QTBT 블록 구조와 함께 사용될 수 있다.

[0115] 한 예로서, 비디오 인코더(20)의 모드 선택 유닛(260)은 여기서 설명된 파티셔닝 기술의 임의의 조합을 수행하도록 구성될 수 있다.

[0116] 위에서 설명한 바와 같이, 비디오 인코더(20)는, (예컨대 기설정된) 예측 모드 세트로부터 최상의 예측 모드 또는 최적의 예측 모드를 결정 또는 선택하도록 구성된다. 예측 모드의 세트는, 예컨대, 인트라-예측 모드 및/또는 인터-예측 모드를 포함할 수 있다.

[0117] 인트라-예측

[0118] 인트라-예측 모드의 세트는, 35개의 상이한 인트라-예측 모드, 예컨대 DC(또는 평균) 모드 및 평면 모드와 같은 비-방향성 모드, 또는 예컨대 HEVC.265에 규정된 바와 같은, 방향성 모드를 포함할 수 있거나, 67개의 상이한 인트라-예측 모드, 예컨대 DC(또는 평균) 모드 및 평면 모드와 같은 비-방향성 모드, 또는 예컨대 VVC에 대해 규정된 바와 같은, 방향성 모드를 포함할 수 있다.

[0119] 인트라 예측 유닛(254)은 동일한 현재 화상의 이웃 블록의 재구성된 샘플을 사용하여, 인트라 예측 모드 세트의 인트라 예측 모드에 따라 인트라 예측 블록(265)을 생성하도록 구성된다.

[0120] 인트라 예측 유닛(254)(또는 일반적으로 모드 선택 유닛(260))은 추가로, 인코딩된 화상 데이터(21)에 포함시키기 위해, 인트라 예측 파라미터(또는 블록에 대해 선택된 인트라 예측 모드를 나타내는 일반적인 정보)를 신텍스 엘리먼트(266)의 형태로 엔트로피 인코딩 유닛(270)에 출력하도록 구성되므로, 예를 들어, 비디오 디코더

(30)가 디코딩을 위해 예측 파라미터를 수신하고 사용할 수 있다.

- [0121] 인터-예측
- [0122] 인터-예측 모드의 세트(또는 가능한 인터-예측 모드)는, 사용 가능한 참조 화상(즉, 예컨대 DBP(230)에 저장된, 이전의 적어도 부분적으로 디코딩된 화상) 및 다른 인터-예측 파라미터, 예컨대 참조 화상의, 전체 참조 화상 또는 일부만, 예컨대 현재 블록의 영역 주위의 검색 윈도우 영역이 최상 매칭 참조 블록을 검색하는데 사용되는 지 여부, 및/또는 예컨대 1/2 세미-펠(pe1) 및/또는 1/4-펠 보간(interpolation)과 같은 픽셀 보간이 적용되는 지 여부에 의존한다.
- [0123] 위의 예측 모드에 추가로, 스킵 모드 및/또는 다이렉트 모드가 적용될 수 있다.
- [0124] 인터 예측 유닛(244)은 모션 추정(motion estimation, ME) 유닛 및 모션 보상(motion compensation, MC) 유닛(둘 다 도 2에는 도시하지 않음)을 포함할 수 있다. 모션 추정 유닛은, 화상 블록(203)(현재 화상(17)의 현재 화상 블록(203)) 및 디코딩된 화상(231), 또는 적어도 하나의 또는 복수의 이전에 재구성된 블록, 예를 들어, 하나 또는 복수의 다른/상이한 이전에 디코딩된 화상(231)의 재구성된 블록을, 모션 추정을 위해 수신하거나 획득하도록 구성될 수 있다. 예를 들어, 비디오 시퀀스는 현재 화상 및 이전에 디코딩된 화상(231)을 포함할 수 있거나, 달리 말하면, 현재 화상 및 이전에 디코딩된 화상(231)은 비디오 시퀀스를 형성하는 화상의 시퀀스의 일부이거나 이를 형성할 수 있다.
- [0125] 예를 들어, 인코더(20)는, 복수의 다른 화상의 동일한 또는 상이한 화상의 복수의 참조 블록으로부터 참조 블록을 선택하도록, 그리고 참조 블록의 위치(x, y 좌표)와 현재 블록의 위치 사이의 참조 화상(또는 참조 화상 인덱스) 및/또는 오프셋(공간적 오프셋)을 인터 예측 파라미터로서 모션 추정 유닛에 제공하도록, 구성될 수 있다. 이 오프셋은 모션 벡터(MV)로도 지칭된다.
- [0126] 모션 보상 유닛은 인터 예측 파라미터를 획득, 예를 들어 수신하고, 인터 예측 파라미터에 기반하여 또는 인터 예측 파라미터를 사용하여 인터 예측을 수행하여 인터 예측 블록(265)을 획득하도록 구성된다. 모션 보상 유닛에 의해 수행되는 모션 보상은 모션 추정에 의해 결정된 모션/블록 벡터에 기반하여 예측 블록을 페치(fetch)하거나 생성하는 것을 포함할 수 있으며, 가능하게는 서브 픽셀 정밀도(sub-pixel precision)에 대한 보간을 수행할 수 있다. 보간 필터링은 알려진 픽셀 샘플로부터 추가적인 픽셀 샘플을 생성할 수 있으며, 따라서 화상 블록을 코딩하는데 사용될 수 있는 후보 예측 블록의 수를 잠재적으로 증가시킬 수 있다. 현재 화상 블록의 PU에 대한 모션 벡터를 수신하면, 모션 보상 유닛은 참조 화상 목록 중 하나에서 모션 벡터가 가리키는 예측 블록을 찾을 수 있다.
- [0127] 모션 보상 유닛은 또한 비디오 슬라이스(slice)의 화상 블록의 디코딩시에 비디오 디코더(30)에 의해 사용하기 위해 블록 및 비디오 슬라이스와 연관된 신택스 엘리먼트를 생성할 수 있다. 슬라이스 및 각각의 신택스 엘리먼트에 추가로 또는 대안으로서, 타일 그룹 및/또는 타일 및 각각의 신택스 엘리먼트가 생성되거나 사용될 수 있다.
- [0128] 엔트로피 코딩
- [0129] 엔트로피 인코딩 유닛(270)은 예를 들어, 엔트로피 인코딩 알고리즘 또는 방식(예: 가변 길이 코딩(variable length coding, VLC) 방식, 콘텍스트 적응형 VLC 방식(context adaptive VLC scheme, CAVLC), 산술 코딩 방식, 2진화(binanzation), 콘텍스트 적응형 이진 산술 코딩(context adaptive binary arithmetic coding, CABAC), 신택스 기반 콘텍스트 적응형 2진 산술 코딩(syntax-based context-adaptive binary arithmetic coding, SBAC), 확률 간격 파티셔닝 엔트로피(probability interval partitioning entropy, PIPE) 코딩 또는 다른 엔트로피 인코딩 방법론 또는 기술) 또는 우회(비 압축)를, 양자화된 계수, 인터 예측 파라미터, 인트라 예측 파라미터, 루프 필터 파라미터 및/또는 기타 신택스 엘리먼트에 적용하여, 예를 들어, 인코딩된 비트스트림(21)의 형태로 출력(272)을 통해 출력될 수 있는 인코딩된 화상 데이터(21)를 획득하도록 구성되므로, 예를 들어, 비디오 디코더(30)는 디코딩을 위해 파라미터를 수신하고 사용할 수 있다. 인코딩된 비트스트림(21)은 비디오 디코더(30)로 전송되거나, 비디오 디코더(30)에 의한 추후 전송 또는 검색(retrieval)을 위해 메모리에 저장될 수 있다.
- [0130] 비디오 인코더(20)의 다른 구조적 변형은 비디오 스트림을 인코딩하도록 구성될 수 있다. 예를 들어, 비-변환 기반 인코더(20)는 특정 블록 또는 프레임에 대해 변환 처리 유닛(106) 없이 직접 잔차 신호를 양자화할 수 있다. 다른 구현예에서, 인코더(20)는 단일 유닛으로 결합될 수 있는 양자화 유닛(208) 및 역 양자화 유닛(210)을

포함할 수 있다.

[0131] 디코더 및 디코딩 방법

[0132] 도 3은 본 출원의 기법을 구현하도록 구성되는 비디오 디코더(30)의 예시를 도시한다. 비디오 디코더(30)는, 디코딩된 화상(331)을 획득하기 위해, 예컨대, 인코더(20)에 의해 인코딩된, 인코딩된 화상 데이터(21)(예컨대, 인코딩된 비트스트림(21))를 수신하도록 구성된다. 인코딩된 화상 데이터 또는 비트스트림은, 인코딩된 화상 데이터를 디코딩하기 위한 정보, 예컨대 인코딩된 비디오 슬라이스(및/또는 타일 그룹 또는 타일)의 화상 블록 및 연관된 신텍스 엘리먼트를 표현하는 데이터를 포함한다.

[0133] 도 3의 예에서, 디코더(30)는 엔트로피 디코딩 유닛(304), 역 양자화 유닛(310), 역변환 처리 유닛(312), 재구성 유닛(314)(예컨대 합산기(314)), 루프 필터(320), 디코딩된 화상 버퍼(DBP)(330), 모드 적용 유닛(360), 인터 예측 유닛(344) 및 인트라 예측 유닛(354)을 포함한다. 인터 예측 유닛(344)은 모션 보상 유닛이거나 포함할 수 있다. 일부 예에서, 비디오 디코더(30)는 도 2의 비디오 인코더(100)에 관해 기술된 인코딩 패스에 일반적으로 역(inverse)인 디코딩 패스를 수행할 수 있다.

[0134] 인코더(20)에 대해 설명한 바와 같이, 역 양자화 유닛(210), 역 변환 처리 유닛(212), 재구성 유닛(214), 루프 필터(220), 디코딩된 화상 버퍼(DPB)(230), 인터 예측 유닛(344) 및 인트라 예측 유닛(354)은 또한 비디오 인코더(20)의 "내장 디코더"를 형성하는 것으로도 지칭된다. 따라서, 역 양자화 유닛(310)은 역 양자화 유닛(110)에 기능적으로 동일할 수 있고, 역변환 처리 유닛(312)은 역변환 처리 유닛(212)에 기능적으로 동일할 수 있고, 재구성 유닛(314)은 재구성 유닛(214)에 기능적으로 동일할 수 있고, 루프 필터(320)는 루프 필터(220)에 기능적으로 동일할 수 있고, 디코딩된 화상 버퍼(330)는 디코딩된 화상 버퍼(230)에 기능적으로 동일할 수 있다. 따라서, 비디오(20) 인코더의 각각의 유닛 및 기능에 대해 제공된 설명은 비디오 디코더(30)의 각각의 유닛 및 기능에 대응하여 적용된다.

[0135] 엔트로피 디코딩

[0136] 엔트로피 디코딩 유닛(304)은 비트스트림(21)(또는 일반적으로 인코딩된 화상 데이터(21))을 파싱하고(parse), 인코딩된 화상 데이터(21)에 대해 예를 들어, 엔트로피 디코딩을 수행하여, 예를 들어, 양자화된 계수(309) 및/또는 디코딩된 코딩 파라미터(도 3에 도시되지 않음), 예를 들어, 인터 예측 파라미터(예: 참조 화상 인덱스 및 모션 벡터), 인트라 예측 파라미터(예: 인트라 예측 모드 또는 인덱스), 변환 파라미터, 양자화 파라미터, 루프 필터 파라미터 및/또는 기타 신텍스 엘리먼트 중 어느 하나 또는 전부를 획득하도록 구성된다. 엔트로피 디코딩 유닛(304)은 인코더(20)의 엔트로피 인코딩 유닛(270)과 관련하여 설명된 인코딩 방식에 대응하는 디코딩 알고리즘 또는 방식을 적용하도록 구성될 수 있다. 엔트로피 디코딩 유닛(304)은 인터 예측 파라미터, 인트라 예측 파라미터 및/또는 다른 신텍스 엘리먼트를 모드 적용 유닛(360)에 제공하도록 그리고 다른 파라미터를 디코더(30)의 다른 유닛에 제공하도록 추가로 구성될 수 있다. 비디오 디코더(30)는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신텍스 엘리먼트를 수신할 수 있다. 슬라이스 및 각각의 신텍스 엘리먼트에 추가로 또는 대안으로서, 타일 그룹 및/또는 타일 및 각각의 신텍스 엘리먼트가 생성되거나 및/또는 사용될 수 있다.

[0137] 역 양자화

[0138] 역 양자화 유닛(310)은 (예를 들어, 엔트로피 디코딩에 의한 파싱 및/또는 디코딩에 의해) 인코딩된 화상 데이터(21)로부터 양자화 파라미터(QP)(또는 일반적으로 역 양자화와 관련된 정보) 및 양자화된 계수를 수신하고, 양자화 파라미터에 기반하여 디코딩된 양자화된 계수(309)에 역 양자화를 적용하여, 변환 계수(311)로도 지칭될 수 있는 역양자화된 계수(311)를 획득하도록 구성될 수 있다. 역 양자화 프로세스는, 양자화의 정도(및 마찬가지로, 적용되어야 하는 역 양자화의 정도)를 결정하기 위해 비디오 슬라이스(또는 타일 또는 타일 그룹) 내의 각각의 비디오 블록에 대해 비디오 인코더(200)에 의해 결정되는 양자화 파라미터의 사용을 포함할 수 있다.

[0139] 역변환

[0140] 역변환 처리 유닛(312)은, 변환 계수(311)로도 지칭되는, 역양자화된 계수(311)를 수신하도록, 그리고 샘플 도메인에서 재구성된 잔차 블록(213)을 획득하기 위해 역양자화된 계수(311)에 변환을 적용하도록 구성될 수 있다. 재구성된 잔차 블록(213)은 변환 블록(313)으로도 또한 지칭될 수 있다. 변환은, 역변환, 예컨대, 역 DCT, 역 DST, 역 정수 변환, 또는 개념적으로 유사한 역변환 프로세스일 수 있다. 역 변환 처리 유닛(312)은 또한 인코딩된 화상 데이터(21)로부터(예: 예를 들어 엔트로피 디코딩 유닛(304)에 의한 파싱 및/또는 디코딩하는 것에 의해) 변환 파라미터 또는 대응하는 정보를 수신하여, 역양자화된 계수(311)에 적용될 변환을 결정하도록

구성될 수 있다.

- [0141] 재구성
- [0142] 재구성 유닛(314)(예를 들어, 가산기 또는 합산기(314))은 재구성된 잔차 블록(313)을 예측 블록(365)에 추가하여, 예를 들어 재구성된 잔차 블록(313)의 샘플 값 및 예측 블록(365)의 샘플 값을 더하는 것에 의해, 샘플 도메인에서 재구성된 블록(315)을 획득하도록 구성될 수 있다.
- [0143] 필터링
- [0144] 루프 필터 유닛(320)은 (코딩 루프 중에 또는 코딩 루프 이후에), 필터 재구성된 블록(315)을 필터링하여 필터링된 블록(321)을 획득함으로써, 픽셀 전환을 부드럽게 하거나 비디오 품질을 향상하도록 구성된다. 루프 필터 유닛(320)은 디-블로킹 필터, 샘플-적응형 오프셋(SAO) 필터 또는 하나 이상의 다른 필터, 예컨대 양방향 필터, 적응형 루프 필터(ALF), 샤프닝(sharpening), 스무딩(smoothing) 필터 또는 협업(collaborative) 필터, 또는 이들의 임의의 조합과 같은 하나 이상의 루프 필터를 포함할 수 있다. 비록 루프 필터 유닛(320)은 도 3에서 인 루프 필터(in loop filter)로서 도시되어 있으나, 다른 구성에서는, 루프 필터 유닛(320)은 포스트 루프 필터(post loop filter)로서 구현될 수 있다.
- [0145] 디코딩된 화상 버퍼
- [0146] 화상의 디코딩된 비디오 블록(321)은 디코딩된 화상 버퍼(330)에 저장되고, 디코딩된 화상(331)은 다른 화상에 대한 후속 모션 보상 및/또는 각각의 디스플레이를 위한 참조 화상으로서 저장된다.
- [0147] 디코더(30)는 사용자에게 표현 또는 보기를 위해, 예컨대, 출력(312)을 통해 디코딩된 화상(311)을 출력하도록 구성된다.
- [0148] 예측
- [0149] 인터 예측 유닛(344)은 인터 예측 유닛(244)(특히 모션 보상 유닛에 대해)과 동일할 수 있고, 인트라 예측 유닛(354)은 기능면에서 인터 예측 유닛(254)과 동일할 수 있으며, (예를 들어 엔트로피 디코딩 유닛(304)에 의한 예를 들어 파싱 및/또는 디코딩하는 것에 의해) 인코딩된 화상 데이터(21)로부터 수신된 파티셔닝 및/또는 예측 파라미터 또는 각각의 정보에 기반하여, 분할 또는 파티셔닝 결정 및 예측을 수행할 수 있다. 모드 적용 유닛(360)은 재구성된 화상, 블록 또는 각각의 샘플(필터링되거나 필터링되지 않은)에 기반하여 블록 당 예측(인트라 예측 또는 인터 예측)을 수행하여, 예측 블록(365)을 획득하도록 구성될 수 있다.
- [0150] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩되는 경우, 모드 적용 유닛(360)의 인트라 예측 유닛(354)은, 현재 화상의 이전에 디코딩된 블록으로부터 시그널링된 인트라 예측 모드 및 데이터에 기초하여 현재의 비디오 슬라이스의 화상 블록에 대한 예측 블록(365)을 생성하도록 구성된다. 비디오 화상이 인터 인코딩된 (즉, B 또는 P) 슬라이스로서 코딩되는 경우, 모드 적용 유닛(360) 내의 인터 예측 유닛(344)(예를 들어, 모션 보상 유닛)은, 엔트로피 디코딩 유닛(304)으로부터 수신되는 모션 벡터 및 다른 신택스 엘리먼트에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록(265)을 생성하도록 구성된다. 인터 예측을 위해, 예측 블록은 참조 화상 목록 중 하나에 있는 참조 화상 중 하나로부터 생성될 수 있다. 비디오 디코더(30)는 DPB(330)에 저장된 참조 화상에 기초하여 디폴트 구성 기법을 사용하여 참조 프레임 목록, 목록 0 및 목록 1을 구성할 수 있다. 동일 또는 유사한 사항이 추가적으로 또는 대안적으로 타일 그룹(예컨대 비디오 타일 그룹) 및/또는 타일(예컨대 비디오 타일)을 슬라이스(예컨대 비디오 슬라이스)에 사용하는 실시예에 의하여 또는 임베 대하여 적용될 수 있으며, 예컨대 비디오는 I, P 또는 B 타일 그룹 및/또는 타일을 사용하여 코딩될 수 있다.
- [0151] 모드 적용 유닛(360)은 모션 벡터 또는 관련 정보 및 다른 신택스 엘리먼트를 파싱(parsing)함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 예측 정보를 사용하여 디코딩 중인 현재 비디오 블록에 대한 예측 블록을 생성하도록 구성된다. 예를 들어, 모드 적용 유닛(360)은 수신된 신택스 엘리먼트의 일부를 사용하여, 비디오 슬라이스의 비디오 블록을 코딩하는데 사용되는 예측 모드(예컨대, 인트라 또는 인터 예측), 인터-예측 슬라이스 유형(예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 화상 목록 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터, 슬라이스의 각각의 인터 코딩된 비디오 블록에 대한 인터 예측 상태, 및 현재 비디오 슬라이스에서 비디오 블록을 디코딩하기 위한 다른 정보를 결정한다. 동일 또는 유사한 사항이 추가적으로 또는 대안적으로 타일 그룹(예컨대 비디오 타일 그룹) 및/또는 타일(예컨대 비디오 타일)을 슬라이스(예컨대 비디오 슬라이스)에 사용하는 실시예에 의하여 또는 임베 대하여 적용될 수 있으며, 예컨대 비디오는 I, P 또는 B 타일 그룹 및/또는 타일을 사

용하여 코딩될 수 있다.

[0152] 도 3에 도시된 비디오 디코더(30)의 실시예는 슬라이스(비디오 슬라이스로도 또한 지칭됨)를 사용하여 화상을 파티션 및/또는 디코딩하도록 구성될 수 있으며, 여기서, 화상은 하나 이상의 슬라이스(통상적으로 비-중첩)를 사용하여 파티셔닝 또는 디코딩 될 수 있고, 각각의 슬라이스는 하나 이상의 블록(예컨대 CTU)를 포함할 수 있다.

[0153] 도 3에 도시된 비디오 디코더(30)의 실시예는 타일 그룹(비디오 타일 그룹으로도 또한 지칭됨) 및/또는 타일(비디오 타일으로도 또한 지칭됨)을 사용하여 화상을 파티션 및/또는 디코딩하도록 하도록 구성될 수 있으며, 여기서, 화상은 하나 이상의 타일 그룹(통상적으로 비-중첩)를 사용하여 파티셔닝 또는 디코딩 될 수 있고, 각각의 타일 그룹은, 예컨대 하나 이상의 블록(예컨대 CTU) 또는 하나 이상의 타일을 포함할 수 있으며, 각각의 타일은, 예컨대 직사각형 형상일 수 있고 하나 이상의 블록(예컨대 CTU), 예컨대 완전한 또는 일부 블록을 포함할 수 있다.

[0154] 비디오 디코더(30)의 다른 변형은 인코딩된 화상 데이터(21)를 디코딩하는데 사용될 수 있다. 예를 들어, 디코더(30)는 루프 필터링 유닛(320) 없이 출력 비디오 스트림을 생성할 수 있다. 예를 들어, 비-변환 기반 디코더(30)는 특정 블록 또는 프레임에 대해 역 변환 처리 유닛(312) 없이 직접 잔차 신호를 역 양자화할 수 있다. 다른 구현에서, 비디오 디코더(30)는 단일 유닛으로 결합될 수 있는 역-양자화 유닛(310) 및 역-변환 처리 유닛(312)을 가질 수 있다.

[0155] 인코더(20) 및 디코더(30)에서는 현재 단계의 처리 결과가 더 처리된 후 다음 단계로 출력될 수 있음을 이해해야 한다. 예를 들어, 보간 필터링, 모션 벡터 유도 또는 루프 필터링 후, 보간 필터링, 모션 벡터 유도 또는 루프 필터링의 처리 결과에 대해 클립(Clip) 또는 시프트(shift)와 같은 추가 연산이 수행될 수 있다.

[0156] 추가 연산이 현재 블록의 도출된 모션 벡터(아핀 모드의 제어 포인트 모션 벡터, 아핀, 평면(planar), ATMVP 모드에서의 서브 블록 모션 벡터, 시간적 모션 벡터 등을 포함하지만 이에 한정되는 것은 아님)에 적용될 수 있다. 예를 들어, 모션 벡터의 값은 표현 비트에 따라 미리 정의된 범위로 제한된다. 모션 벡터의 표현 비트가 bitDepth이면, 범위는 $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)}-1$ 이며, 여기서 "^"는 지수를 의미한다. 예를 들어, bitDepth가 16으로 설정되면, 범위는 $-32768 \sim 32767$ 이고; bitDepth가 18으로 설정되면, 범위는 $-131072 \sim 131071$ 이다. 예를 들어, 도출된 모션 벡터(예컨대 하나의 8x8 블록 내의 4개의 4x4 서브-블록의 MV)의 값은, 4개의 4x4 서브-블록 MV의 정수 부분 사이의 최대 차이가 N 픽셀보다 크지 않도록, 예를 들어 1 픽셀보다 크지 않도록, 제한된다. 다음은 bitDepth에 따라 모션 벡터를 제한하는 두 가지 방법을 제공한다.

[0157] 방법 1: 다음 연산에 의해 오버플로우 MSB(최상위 비트)를 제거한다.

[0158]
$$ux = (mvx + 2^{bitDepth}) \% 2^{bitDepth} \quad (1)$$

[0159]
$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (2)$$

[0160]
$$uy = (mvy + 2^{bitDepth}) \% 2^{bitDepth} \quad (3)$$

[0161]
$$mvy = (uy \geq 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (4)$$

[0162] 여기서, mvx는 이미지 블록 또는 서브-블록의 모션 벡터의 수평 성분이고, mvy는 이미지 블록 또는 서브-블록의 모션 벡터의 수직 성분이고, 그리고 ux 및 uy는 중간 값을 나타낸다;

[0163] 예를 들어, mvx의 값이 -32769이면, 식 (1) 및 (2)를 적용한 후, 결과 값은 32767이다. 컴퓨터 시스템에서, 10진수는 2의 보수(two's complement)로 저장된다. -32769의 2의 보수는 1,0111,1111,1111,1111(17 비트)이고 MSB는 폐기되므로 결과적인 2의 보수는 0111,1111,1111,1111(10진수는 32767)이며 이는 식 (1) 및 (2)를 적용한 출력과 동일하다.

[0164]
$$ux = (mvp_x + mvdx + 2^{bitDepth}) \% 2^{bitDepth} \quad (5)$$

[0165]
$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (6)$$

[0166]
$$uy = (mvp_y + mvdy + 2^{bitDepth}) \% 2^{bitDepth} \quad (7)$$

[0167]
$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (8)$$

[0168] 이 연산들은 식 (5) 내지 (8)과 같이.mvp와 mvd의 합산에서 적용될 수 있다.

[0169] 방법 2: 값을 클리핑하여 오버플로우 MSB를 제거한다.

[0170]
$$vx = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vx)$$

[0171]
$$vy = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vy)$$

[0172] 여기서, vx는 이미지 블록 또는 서브-블록의 모션 벡터의 수평 성분이고, vy는 이미지 블록 또는 서브-블록의 모션 벡터의 수직 성분이고; x, y 및 z는 MV 클리핑 프로세스의 3개의 입력 값에 각각 대응하고, 함수 Clip3의 정의는 다음과 같다:

[0173]
$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases}$$

[0174] 도 4는 본 개시의 실시예에 따른 비디오 코딩 디바이스(400)의 개략도이다. 비디오 코딩 디바이스(400)는 여기에 개시된 실시예를 구현하는데 적합하다. 일 실시예로서, 비디오 코딩 디바이스(400)는 도 1a의 비디오 디코더(30)와 같은 디코더 또는 도 1a의 비디오 인코더(20)와 같은 인코더일 수 있다.

[0175] 비디오 코딩 디바이스(400)는, 데이터를 수신하기 위한 입구 포트(410)(또는 입력 포트(410)) 및 수신기 유닛(Rx)(420); 데이터를 처리하기 위한 프로세서, 논리 유닛, 또는 중앙 처리 장치(CPU)(430); 데이터를 전송하기 위한 송신기 유닛(Tx)(440) 및 출구 포트(450)(또는 출력 포트(450)); 및 데이터를 저장하기 위한 메모리(460)를 포함한다. 비디오 코딩 디바이스(400)는 광학적 또는 전기적 신호의 유출 또는 유입을 위해 입구 포트(410), 수신기 유닛(420), 송신기 유닛(440), 및 출구 포트(450)에 연결되는 광-전기(OE; optical-to-electrical) 콤포넌트 및 전기-광(EO; electrical-to-optical) 콤포넌트를 또한 포함할 수 있다.

[0176] 프로세서(430)는 하드웨어 및 소프트웨어로 구현된다. 프로세서(430)는 하나 이상의 CPU 칩, 코어(예를 들어, 멀티-코어 프로세서), FPGA, ASIC, 및 DSP로 구현될 수 있다. 프로세서(430)는 입구 포트(410), 수신기 유닛(420), 송신기 유닛(440), 출구 포트(450), 및 메모리(460)와 통신한다. 프로세서(430)는 코딩 모듈(470)을 포함한다. 코딩 모듈(470)은 위에서 설명된 개시된 실시예들을 구현한다. 예를 들어, 코딩 모듈(470)은 다양한 코딩 동작을 구현, 처리, 준비 또는 제공한다. 코딩 모듈(470)을 포함함으로써 비디오 코딩 디바이스(400)의 기능에 상당한 향상이 제공되고 비디오 코딩 디바이스(400)의 상이한 상태로의 변환에 영향을 끼친다. 대안적으로, 코딩 모듈(470)은 메모리(460)에 저장되고 프로세서(430)에 의해 실행되는 명령어로서 구현된다.

[0177] 메모리(460)는 하나 이상의 디스크, 테이프 드라이브, 및 고체 드라이브를 포함할 수 있고, 프로그램이 실행을 위해 선택될 때 프로그램을 저장하고, 프로그램 실행 중에 판독되는 명령 및 데이터를 저장하기 위해 오버-플로우 데이터 저장 디바이스로서 사용될 수 있다. 메모리(460)는 예를 들어 휘발성 및/또는 비휘발성일 수 있으며, ROM(read-only memory), RAM(random-access memory), TCAM(ternary content-addressable memory) 및/또는 SRAM(static random-access memory)일 수 있다.

[0178] 도 5는 예시적인 일 실시예에 따른 도 1로부터 소스 디바이스(12) 및 테스트네이션 디바이스(14) 중 하나 또는 양자 모두로서 사용될 수 있는 장치(500)의 단순화된 블록도이다.

[0179] 장치(500)의 프로세서(502)는 중앙 처리 장치일 수 있다. 대안적으로, 프로세서(502)는, 정보를 조작 또는 처리할 수 있는, 하나 이상의 기존의 또는 향후에 개발될 하나의 디바이스 또는 다중 디바이스의 임의의 다른 유형일 수 있다. 비록 개시된 구현예가 예컨대 프로세서(502)와 같은 단일 프로세서로 실현될 수 있으나, 속도 및 효율의 이점은 둘 이상의 프로세서를 사용하여 달성될 수 있다.

[0180] 장치(500)의 메모리(504)는 구현예에서 판독 전용 메모리(ROM) 디바이스 또는 랜덤 액세스 메모리(RAM) 디바이스일 수 있다. 임의의 다른 적절한 유형의 저장 디바이스가 메모리(504)로서 사용될 수 있다. 메모리(504)는 버스(512)를 사용하여 프로세서(502)에 의해 액세스되는 코드 및 데이터(506)를 포함할 수 있다. 메모리(504)는 운영 체제(508) 및 응용 프로그램(510)을 더 포함할 수 있으며, 응용 프로그램(510)은 프로세서(502)가 여기에 기술된 방법을 수행하도록 허용하는 적어도 하나의 프로그램을 포함한다. 예를 들어, 응용 프로그램(510)은, 여기에 기술된 방법을 수행하는 비디오 코딩 애플리케이션을 더 포함하는, 애플리케이션 1 내지 N을 포함할 수 있

다.

- [0181] 장치(500)는 디스플레이(518)와 같은 하나 이상의 출력 디바이스를 또한 포함할 수 있다. 예로서, 디스플레이(518)는, 한 예로서, 디스플레이 및 터치 입력을 감지하여 작동하는 터치-감응 엘리먼트를 결합하는 터치 감응 디스플레이일 수 있다. 디스플레이(518)는 버스(512)를 통해 프로세서(502)에 연결될 수 있다.
- [0182] 여기서서는 단일 버스로 도시되어 있지만, 장치(500)의 버스(512)는 다중 버스로 구성될 수 있다. 또한, 2차 스토리지(514)는 장치(500)의 다른 구성 요소에 직접 결합될 수 있거나 네트워크를 통해 액세스될 수 있고, 메모리 카드와 같은 단일 통합 유닛 또는 다중 메모리 카드와 같은 다중 유닛을 포함할 수 있다. 따라서 장치(500)는 매우 다양한 구성으로 구현될 수 있다.
- [0183] 백그라운드 인트라 예측 모드 코딩
- [0184] 현재 VTM 4.0에서, 6-MPM (가장 가능성 있는 모드) 목록이 인트라 예측 모드 코딩을 위해 구성된다. MPM 목록은 현재 블록의 인트라 예측 모드를 코딩하는데 요구되는 비트를 줄이기 위해 도입된다. 6-MPM 목록은 현재 블록의 이웃 블록의 인트라 예측 모드에 기초하여 구성된다. 현재 블록의 인트라 예측 모드가 MPM 목록에 해당하는 경우, 실제의 모드가 아닌 인덱스를 코딩하여 더 적은 비트를 사용할 수 있다. 현재 블록의 인트라 예측 모드가 MPM 목록에 해당하지 않는 경우, 절단 2진화(truncated binarization)가 현재 블록의 인트라 예측 모드의 코딩에 사용된다. MPM 목록은 현재 블록의 인트라 예측 모드에 대한 우수한 예측을 제공한다.
- [0185] 다중 참조 라인(MRL)에 대한 MPM 목록.
- [0186] VTM 4.0에서, 다중 참조 라인(Multiple Reference Line, MRL) 코딩 도구는 다중 이웃 참조 라인 중 하나를 사용하여 현재 블록의 샘플을 예측할 수 있다. MRL 라인 인덱스의 값이 0과 같은 경우(즉, 도 6에 도시된 바와 같이, 가장 가까운 이웃 참조 라인이 사용된다), 평면(planar) 및 DC를 포함하는 일반적인 6개의 MPM 목록이 사용된다. MRL 라인 인덱스의 값이 0과 같지 않은 경우, 평면(즉, 0의 값) 및 DC(즉, 1의 값)를 제외한 6-MPM 목록이 사용된다.
- [0187] 인트라 서브-파티션 코딩 모드(ISP)에 대한 MPM 목록
- [0188] 인트라 서브-파티션(ISP) 코딩 모드는 VTM4.0(JVET-M0102)에서 새로이 채용된 도구이다. 루마 인트라-예측 블록은, 표 1에 도시된 바와 같이, 블록 크기 제원에 따라 2 또는 4개의 서브-파티션으로 수직으로 또는 수평으로 분할된다. 도 7 및 도 8은 2가지 가능성의 예를 도시한다. 모든 서브-파티션은 적어도 16 샘플을 가지는 조건을 충족한다.

표 1

블록 크기에 따른 서브-파티션의 수

블록 크기	서브-파티션의 수
4×4	분할되지 않음
4×8 및 8×4	2
다른 모드 경우	4

- [0189]
- [0190] 인트라 서브-파티션 코딩 모드가 활성화되는 경우, DC 모드를 제외한 다른 MPM 목록이 VTM 4.0에서 사용된다. 인트라 서브-파티션 모드는 다중 참조 라인 인덱스의 값이 0과 같은 경우(즉, MRL은 현재 인트라 예측 블록에 적용되지 않음)에 적용될 수 있다. 모든 서브-파티션은 단일의 인트라 예측 모드를 공유하며, 따라서, MPM 목록은 인트라 블록에 대해 한번 구성되고 모든 서브-파티션에 의해 공유된다.
- [0191] MPM의 구성은 ISP의 분할 모드에 따라 달라질 수 있다. 수평 또는 수직의, 2가지 분할 모드가 결정된다. 이들은 도 7 및 도 8에서 도시된 바와 같으며, 여기서, 수평 분할/파티션은 상단에 도시되고, 수직 분할/파티션은 하단에 도시된다.

[0192] 다음 표 2는 VTM 4.0에서 사용되는 3 MPM 목록의 특성을 요약한다.

표 2

[0193] 상이한 상황에서 VTM 4.0에서 사용되는 상이한 6-MPM

MRL config	ISP config	MPM 목록 특성
$MRL \neq 0$	-	6 MPM이지만 평면 모드 및 DC 모드를 제외함
$MRL = 0$	1	6 MPM이지만 DC 모드를 제외함
$MRL = 0$	0	평면 모드 및 DC 모드를 포함하는 6 MPM

[0194] CABAC 컨텍스트 모델링/코딩에 관한 배경.

[0195] 도 9에 도시된 바와 같이, CABAC 코딩은 2진화(binization), 컨텍스트 모델링, 및 2진 산술 코딩을 포함한다. 2진화(binization)는 선택스 엘리먼트를 2진 심볼(bins)로 맵핑한다. 컨텍스트 모델링은 각각의 바이패스되지 않은(즉, 정규로 코딩된) 빈(bin)의 확률을 특정 컨텍스트에 기초하여 추정한다. 결국, 2진 산술 코딩은 추정된 확률에 따라 빈(bins)을 비트로 압축한다.

[0196] 각각의 비-2진 선택스 엘리먼트 값은 빈(bins)의 시퀀스로 분해함으로써, CABAC에서 각각의 빈(bin) 값의 추가적인 처리는 연관된 코딩 모드 결정에 따라 달라지며, 이는 정규의 또는 바이패스 모드로서 선택될 수 있다. 후자는, 균일하게 분포된 것으로 가정하고, 결과적으로 전체 정규의 2진 산술 인코딩(및 디코딩) 프로세스가 단순히 바이패스되는 빈(bins)에 대해 선택된다. 정규 코딩 모드에서, 각각의 빈(bin) 값은 정규 2진 산술 코딩 엔진을 사용하여 인코딩되며, 여기서 연관된 확률 모델은 선택스 엘리먼트의 유형 및 선택스 엘리먼트의 2진화된 표현에서 빈(bin) 위치 또는 빈(bin) 인덱스(binIdx)에 기초하여 고정된 선택에 의해 결정되거나, 관련 부가 정보에 따라 둘 이상의 확률 모델로부터 적응적으로 선택된다.

[0197] 확률 모델의 선택은 컨텍스트 모델링으로 지칭된다. CABAC의 각각의 확률 모델은, 고정된 할당에 의해 결정되거나 주어진 컨텍스트 모델이 지정되는 컨텍스트 유도 로직에 의해 계산되는, 고유한 컨텍스트 인덱스(ctxIdx)를 사용하여 처리된다.

[0198] 좌측 및 상측 이웃을 어떻게 결정할 것인가에 관한 백그라운드.

[0199] VVC의 인트라 예측 모드 코딩은 이의 이웃 블록의 인트라 예측 모드에 따라 달라진다. 즉, 현재 블록의 좌측 및 상측 블록. 이들의 위치는 도 10에 도시되어 있다.

[0200] MPM 인덱스의 시그널링

[0201] 플래그(즉, MPM 플래그)는 현재 블록의 인트라 예측 모드가 MPM 목록 내에 있는지 아닌지 여부를 표시하는데 사용된다. MPM 플래그가 참(true)(즉, 1의 값)인 경우, 인트라 예측 모드는 MPM 인덱스를 사용하여 결정될 수 있다. MPM 인덱스는, MPM 목록의 길이가 6인 경우 다음 표 3과 같이, 절단 단항 코드를 사용하여 코딩된다. MPM 플래그가 참이 아닌 경우, 현재 블록의 인트라 예측 모드는 MPM 목록에 있지 않고, 모드는 절단 2진 코드(truncated binary code)를 사용하여 코딩된다. MPM 목록이 5개의 모드로 구성되는 경우 MPM 인덱스의 다른 예시가 표 4에 도시되어 있다. 인덱스를 코딩하기 위한 빈(bin) 문자열은 절단 단항 코드(비-MPM 모드를 코딩하기 위한 절단 2진 코드(truncated binary code)가 아님)로 지칭된다. 절단 단항 코드의 경우 빈(bin) 0과 1은 일 반성을 잃지 않고 상호 교환 가능하다. 일례로, MPM 목록이 5개의 모드로 구성되는 경우의 MPM 인덱스는 표 5와 같이도 코딩될 수 있다.

표 3

[0202] MPM 목록에 6개의 엔트리가 있는 경우 MPM 플래그 및 MPM 인덱스 코딩

인트라 예측 모드	MPM 플래그	MPM 인덱스	MPM 인덱스의 Bin 문자열
MPM 모드 (6)	1	0	0
		1	10
		2	110
		3	1110
		4	11110
		5	11111

	0		절단 2진(TB) 코드
--	---	--	--------------

표 4

[0203]

MPM 목록에 5개의 엔트리가 있는 경우, MPM 플래그 및 MPM 인덱스 코딩

인트라 예측 모드	MPM 플래그	MPM 인덱스	MPM 인덱스의 Bin 문자열
MPM 모드 (5)	1	0	0
		1	10
		2	110
		3	1110
		4	1111
	0		절단 2진(TB) 코드

표 5

[0204]

MPM 목록에 5개의 엔트리가 있는 경우, MPM 플래그 및 MPM 인덱스 코딩, 및 MPM 인덱스에 대한 대체 코드워드

인트라 예측 모드	MPM 플래그	MPM 인덱스	MPM 인덱스의 Bin 문자열
MPM 모드 (5)	1	0	1
		1	01
		2	001
		3	0001
		4	0000
	0		절단 2진(TB) 코드

[0205]

평면 모드, DC, 수평, 수직 모드 및 각도 모드(angular mode)에 대한 백그라운드: 인트라 예측 모드 평면(0의 값) 및 DC(1의 값). 다른 인트라 예측 모드는 각도 예측 모드로 지칭된다.

[0206]

VTM4.0에서, 표 6에 도시된 바와 같은 다음의 선택스 설계가 인트라 예측 모드 정보를 시그널링하는데 사용된다.

표 6

VTM4.0에서 선택스 파싱

<code>if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)</code>	
<code> intra_luma_mpm_flag[x0][y0]</code>	<code>ae(v)</code>
<code> if(intra_luma_mpm_flag[x0][y0]) {</code>	
<code> intra_luma_mpm_idx[x0][y0]</code>	<code>ae(v)</code>
<code> }</code>	
<code> else</code>	
<code> intra_luma_mpm_remainder[x0][y0]</code>	<code>ae(v)</code>

[0207]

[0208]

`intra_luma_ref_idx[x0][y0]`는 0, 1, 또는 2의 3개의 값을 가질 수 있으며, 어느 참조 라인이 사용되는지를 표시한다. `intra_luma_ref_idx[x0][y0]`이 존재하지 않는 경우, 0과 같은 것으로 추론된다.

[0209]

1과 같은 `intra_subpartitions_mode_flag[x0][y0]`는, 현재 인트라 코딩 유닛이 추가로 파티션됨(즉, ISP 모드가 활성화됨)을 지정한다. 0과 같은 `intra_subpartitions_mode_flag[x0][y0]`는, 현재 인트라 코딩 유닛이 직사각형의 변환 블록 서브파티션으로 파티션됨을 지정한다. `intra_subpartitions_mode_flag[x0][y0]`이 존재하지 않는 경우, 0과 같은 것으로 추론된다.

[0210]

선택스 엘리먼트 `intra_luma_mpm_flag[x0][y0]`, `intra_luma_mpm_idx[x0][y0]` 및 `intra_luma_mpm_remainder[x0][y0]`는 루마 샘플에 대한 인트라 예측 모드를 지정한다.

- [0211] `intra_luma_mpm_flag[x0][y0]`는 현재 블록의 인트라 예측 모드가 MPM 목록으로부터 유도되는지 아닌지 여부를 표시한다. `intra_luma_mpm_flag[x0][y0]`의 값이 1이면, 이는 현재 블록의 인트라 예측 모드가 MPM 목록으로부터 유도됨을 나타내고; `intra_luma_mpm_flag[x0][y0]`의 값이 0이면, 이는 현재 블록의 인트라 예측 모드가 MPM 목록으로부터 유도되지 않음을 나타낸다. `intra_luma_mpm_flag[x0][y0]`이 존재하지 않는 경우, 1과 같은 것으로 추론된다. `intra_luma_mpm_flag[x0][y0]`는 표 3에서 MPM 플래그에 대응한다.
- [0212] `intra_luma_mpm_idx`는 표 3에서 MPM 인덱스에 대응하고 표 3에 도시된 바와 같은 절단 단항 코드를 사용한다.
- [0213] `intra_luma_mpm_remainder`는 MPM 목록에 없는 모드의 절단 2진 코드(truncated binary code)를 표현한다.
- [0214] 실시예 1에서, 신택스 엘리먼트 `intra_luma_planar_flag` 및 `ctx`가 개시된다.
- [0215] 본 발명의 한 실시예에 따르면, 플래그 `intra_luma_planar_flag`이 도입된다.
- [0216] `intra_luma_mpm_flag`의 값이 참(true)이면(예를 들면, 값이 1임), 디코더는 `intra_luma_planar_flag`를 파싱하여 현재 블록의 인트라 예측 모드가 평면 모드인지 아닌지 여부를 결정한다. `intra_luma_planar_flag`의 값이 1인 경우, 현재 블록의 인트라 예측 모드는 평면 모드이다. `intra_luma_planar_flag`의 값이 0인 경우, 현재 블록의 인트라 예측 모드는 평면 모드가 아니다. `intra_luma_planar_flag`이 존재하지 않으면(예를 들어, 이 플래그는 `intra_luma_mpm_flag`이 0과 같은 경우 비트스트림으로부터 파싱될 수 없다), `intra_luma_planar_flag`의 값은 0으로 추론된다.
- [0217] VVC 작업 초안 사양의 스타일로 작성된 경우, 현재 블록의 인트라 예측 모드
- [0218] `IntraPredModeY[xCb][yCb]`는 다음의 절차를 적용하여 유도된다:
- [0219] - `intra_luma_mpm_flag[xCb][yCb]`이 1과 같으면,
- [0220] - `intra_luma_planar_flag[xCb][yCb]`이 1과 같으면, `IntraPredModeY[xCb][yCb]`는 INTRA_PLANAR과 같도록 설정된다.
- [0221] - 그렇지 않으면, `IntraPredModeY[xCb][yCb]`는 `candModeList[intra_luma_mpm_idx[xCb][yCb]`]과 같도록 설정된다.
- [0222] 인트라 예측 모드가 평면 모드가 아니면(즉, `intra_luma_planar_flag`의 값이 0임), 디코더는 신택스 엘리먼트 `intra_luma_mpm_idx`를 추가로 파싱한다.
- [0223] 한 예로서, `intra_luma_planar_flag`의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 일반 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.
- [0224] 한 예로서, `intra_luma_planar_flag`의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 다중 참조 라인(MRL)이 가능한 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.
- [0225] 한 예로서, `intra_luma_planar_flag`의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 인트라 서브-파티션(ISP) 코딩 모드가 활성화되는 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.
- [0226] 예로서, `intra_luma_planar_flag`의 위치가 표 7에서와 같이 배치된다.

표 7

intra_luma_planar_flag를 가지는 선택스

if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]){	
intra_luma_planar_flag [x0][y0]	ae(v)
if (!intra_luma_planar_flag [x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
}	
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)

[0227]

[0228]

실시예 2: intra_luma_not_planar_flag 및 ctx

[0229]

본 발명의 한 실시예에 따르면, 선택스 엘리먼트 intra_luma_not_planar_flag이 도입된다.

[0230]

intra_luma_mpm_flag의 값이 참(true)이면(예를 들어, 값이 1임), 디코더는 intra_luma_not_planar_flag를 파싱하여 현재 블록의 인트라 예측 모드가 평면 모드인지 아닌지 여부를 결정한다. intra_luma_not_planar_flag의 값이 1인 경우, 현재 블록의 인트라 예측 모드는 평면 모드가 아니고; intra_luma_not_planar_flag의 값이 0인 경우, 현재 블록의 인트라 예측 모드는 평면 모드이다. intra_luma_not_planar_flag이 비트스트림 내에 존재하지 않으면, intra_luma_not_planar_flag의 값은 1로 추론된다. 예로서, intra_luma_not_planar_flag의 위치가 표 8에서와 같이 배치된다.

[0231]

현재 블록의 인트라 예측 모드가 평면 모드가 아니면(즉, intra_luma_not_planar_flag의 값이 1임), 디코더는 선택스 엘리먼트 intra_luma_mpm_idx를 추가로 파싱한다.

[0232]

VVC 작업 초안 사양의 스타일로 작성된 경우, 현재 블록의 인트라 예측 모드

[0233]

IntraPredModeY[xCb][yCb]는 다음의 절차를 적용하여 유도된다:

[0234]

- intra_luma_mpm_flag[xCb][yCb]이 1과 같으면,

[0235]

- intra_luma_not_planar_flag[xCb][yCb]이 0과 같으면, IntraPredModeY[xCb][yCb]는 INTRA_PLANAR과 같도록 설정된다.

[0236]

- 그렇지 않으면, IntraPredModeY[xCb][yCb]는 candModeList[intra_luma_mpm_idx[xCb][yCb]]과 같도록 설정된다.

[0237]

한 예로서, intra_luma_not_planar_flag의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 일반 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.

[0238]

한 예로서, intra_luma_not_planar_flag의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 다중 참조 라인(MRL)이 가능한 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.

[0239]

한 예로서, intra_luma_not_planar_flag의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 인트라 서브-파티션(ISP) 코딩 모드가 활성화되는 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.

표 8

intra_luma_not_planar_flag를 가지는 신택스

if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]){	
intra_luma_not_planar_flag [x0][y0]	ae(v)
if(intra_luma_not_planar_flag [x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
}	
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)

[0240]

[0241] 실시예 3: intra_luma_planar_flag 및 MRL, 그리고 바이패스 디코딩

[0242] 본 발명의 한 실시예에 따르면, 플래그 intra_luma_planar_flag이 도입된다.

[0243] intra_luma_mpm_flag의 값이 참(true)이면(예를 들어, 값이 1임), 디코더는 참조 인덱스 라인(intra_luma_ref_idx)의 값이 0인지 아닌지 여부를 파싱한다. 참조 인덱스 라인의 값이 0이면, intra_luma_planar_flag의 값을 파싱하여 현재 블록의 인트라 예측 모드가 평면 모드인지 아닌지 여부를 결정한다. intra_luma_planar_flag의 값이 1인 경우, 현재 블록의 인트라 예측 모드는 평면 모드이다. intra_luma_planar_flag의 값이 0인 경우, 현재 블록의 인트라 예측 모드는 평면 모드가 아니다. intra_luma_planar_flag이 없으면, intra_luma_planar_flag의 값은 0으로 추론된다.

[0244] 현재 블록의 인트라 예측 모드가 평면 모드가 아니면(즉, intra_luma_planar_flag의 값이 0임), 디코더는 intra_luma_mpm_idx를 추가로 파싱한다.

[0245] 한 예로서, intra_luma_planar_flag의 값은 CABAC 바이패스 코딩되고(즉, CABAC 컨텍스트 코딩되지 않음), 디코더는 CABAC 컨텍스트를 사용하지 않고 플래그를 파싱한다.

[0246] intra_luma_planar_flag의 위치가 표 9에서와 같이 배치된다.

표 9

intra_luma_planar_flag를 가지고 바이패스 코딩되는 신택스

if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]){	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_planar_flag[x0][y0]	ae(v)
if(!intra_luma_planar_flag [x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
}	
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)

[0247]

- [0248] 실시예 4: `intra_luma_not_planar_flag` 및 MRL, 그리고 바이패스 디코딩
- [0249] 본 발명의 한 실시예에 따르면, 플래그 `intra_luma_not_planar_flag`이 도입된다.
- [0250] `intra_luma_mpm_flag`의 값이 참(true)이면(예를 들어, 값이 1임), 디코더는 참조 인덱스 라인의 값이 0인지 아닌지 여부를 파악한다. 참조 인덱스 라인의 값이 0이면, `intra_luma_not_planar_flag`를 파악하여 현재 블록의 인트라 예측 모드가 평면 모드인지 아닌지 여부를 결정한다. `intra_luma_not_planar_flag`의 값이 1인 경우, 현재 블록의 인트라 예측 모드는 평면 모드가 아니다. `intra_luma_planar_flag`의 값이 0인 경우, 현재 블록의 인트라 예측 모드는 평면 모드이다. `intra_luma_not_planar_flag`가 없으면, `intra_luma_not_planar_flag`의 값은 1로 추론된다.
- [0251] 현재 블록의 인트라 예측 모드가 평면 모드가 아니면(즉, `intra_luma_not_planar_flag`의 값이 1임), 디코더는 `intra_luma_mpm_idx`를 추가로 파악한다.
- [0252] 한 예로서, `intra_luma_not_planar_flag`의 값은 CABAC 바이패스 코딩되고(즉, CABAC 컨텍스트 코딩되지 않음), 디코더는 CABAC 컨텍스트를 사용하지 않고 플래그를 파악한다.
- [0253] `intra_not_luma_planar_flag`의 위치가 표 10에서와 같이 배치된다.

표 10

`intra_luma_not_planar_flag`를 가지고 바이패스 코딩되는 신택스

<code>if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)</code>	
<code>intra_luma_mpm_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(intra_luma_mpm_flag[x0][y0]){</code>	
<code>if(intra_luma_ref_idx[x0][y0] == 0)</code>	
<code>intra_luma_not_planar_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(intra_luma_not_planar_flag [x0][y0])</code>	
<code>intra_luma_mpm_idx[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>else</code>	
<code>intra_luma_mpm_remainder[x0][y0]</code>	<code>ae(v)</code>

- [0254]
- [0255] 실시예 5: `intra_luma_not_planar_flag` 및 MRL, 그리고 `cabac` 컨텍스트 코딩(실시예 4와의 차이는 `intra_luma_not_planar_flag`은 바이패스 코딩되는 대신에 `cabac` 컨텍스트 코딩되는 점이다).
- [0256] 본 발명의 한 실시예에 따르면, 플래그 `intra_luma_not_planar_flag`이 도입된다.
- [0257] `intra_luma_mpm_flag`의 값이 참(true)이면(예를 들어, 값이 1임), 디코더는 참조 인덱스 라인의 값이 0인지 아닌지 여부를 파악한다. 참조 인덱스 라인의 값이 0이면, `intra_luma_not_planar_flag`를 파악하여 현재 블록의 인트라 예측 모드가 평면 모드인지 아닌지 여부를 결정한다. `intra_luma_not_planar_flag`의 값이 1인 경우, 현재 블록의 인트라 예측 모드는 평면 모드가 아니다. `intra_luma_planar_flag`의 값이 0인 경우, 현재 블록의 인트라 예측 모드는 평면 모드이다. `intra_luma_not_planar_flag`가 없으면, `intra_luma_not_planar_flag`의 값은 1로 추론된다.
- [0258] 현재 블록의 인트라 예측 모드가 평면 모드가 아니면(즉, `intra_luma_not_planar_flag`의 값이 1임), 디코더는 `intra_luma_mpm_idx`를 추가로 파악한다.
- [0259] 한 예로서, `intra_luma_not_planar_flag`의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 일반 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파악한다.
- [0260] 한 예로서, `intra_luma_not_planar_flag`의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음),

디코더는 현재 블록이 다중 참조 라인(MRL)이 가능한 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.

[0261] 한 예로서, `intra_luma_not_planar_flag`의 값은 CABAC 컨텍스트 코딩되고(즉, CABAC 바이패스 코딩되지 않음), 디코더는 현재 블록이 인트라 서브-파티션(ISP) 코딩 모드가 활성화되는 인트라 블록인지 아닌지 여부에 기초하여 컨텍스트를 사용하여 플래그를 파싱한다.

[0262] `intra_not_luma_planar_flag`의 위치가 표 11에서와 같이 배치된다.

표 11

`intra_luma_not_planar_flag`를 가지고 cabac 컨텍스트 코딩되는 신택스

<code>if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)</code>	
<code>intra_luma_mpm_flag[x0][y0]</code>	ae(v)
<code>if(intra_luma_mpm_flag[x0][y0]) {</code>	
<code>if(intra_luma_ref_idx[x0][y0] == 0)</code>	
<code>intra_luma_not_planar_flag[x0][y0]</code>	ae(v)
<code>if(intra_luma_not_planar_flag [x0][y0])</code>	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
<code>}</code>	
<code>else</code>	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)

[0263]

[0264] 예로서,

[0265] `intra_luma_not_planar_flag`를 더한다.

<code>if(intra_luma_mpm_flag[x0][y0]) {</code>	
<code>if(intra_luma_ref_idx[x0][y0] == 0)</code>	
<code>intra_luma_not_planar_flag[x0][y0]</code>	ae(v)
<code>if(intra_luma_not_planar_flag[x0][y0])</code>	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
<code>} else</code>	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)
<code>}</code>	

[0266]

<code>intra_luma_mpm_flag[][]</code>	FL	cMax = 1
<code>intra_luma_not_planar_flag[][]</code>	FL	cMax = 1
<code>intra_luma_mpm_idx[][]</code>	TR	cMax = 4, cRiceParam = 0

[0267]

[0268] `intra_luma_not_planar_flag[x0][y0]`이 존재하지 않는 경우, 1과 같은 것으로 추론된다.

[0269] 아래의 표 12는 `ctxInc`를 컨텍스트 코딩된 빈(bins)을 가지는 신택스 엘리먼트에 할당하는 예시를 도시한다.

표 12

컨텍스트 코딩된 빈(bins)을 가지는 선택스 엘리먼트로 ctxInc의 할당

Syntax element	binIdx					
	0	1	2	3	4	>= 5
intra_subpartition_split_flag	0	na	na	na	na	na
intra_luma_mpm_flag[][]	0	na	na	na	na	na
intra_luma_not_planar_flag[][]	intra_subpartitio ns_mode_flag	na	na	na	na	na
intra_luma_mpm_idx[][]	bypass	bypass	bypass	bypass	na	na
intra_luma_mpm_remainder[][]	bypass	bypass	bypass	bypass	bypass	bypass

[0270]

[0271] 일 실시예로서, MPM 목록은 다음의 프로세스에 따라 구성된다.

[0272] 8.4.2 루마 인트라 예측 모드에 대한 유도 과정

[0273] 이 프로세스에의 입력은 다음과 같다:

- [0274] - 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 현재 루마 코딩 블록의 좌측 상단 샘플을 지정하는 루마 위치(x_{Cb} , y_{Cb}),
- [0275] - 루마 샘플에서 현재 코딩 블록의 폭을 지정하는 변수 $cbWidth$,
- [0276] - 루마 샘플에서 현재 코딩 블록의 높이를 지정하는 변수 $cbHeight$.

[0277] 이 프로세스에서, 루마 인트라 예측 모드 $IntraPredModeY[x_{Cb}][y_{Cb}]$ 이 유도된다.

[0278] 표 13은 인트라 예측 모드 $IntraPredModeY[x_{Cb}][y_{Cb}]$ 에 대한 값 및 연관된 명칭을 지정한다.

표 13

인트라 예측 모드 및 연관된 명칭의 사양

Intra prediction mode	Associated name
0	INTRA_PLANAR
1	INTRA_DC
2..66	INTRA_ANGULAR2..INTRA_ANGULAR66
81..83	INTRA_LT_CCLM, INTRA_L_CCLM, INTRA_T_CCLM

[0279]

[0280] 참고 - : 인트라 예측 모드 INTRA_LT_CCLM, INTRA_L_CCLM 및 INTRA_T_CCLM는 크로마 성분에만 적용 가능하다.

[0281] $IntraPredModeY[x_{Cb}][y_{Cb}]$ 는 다음과 같이 유도된다:

- [0282] - $intra_luma_not_planar_flag[x_{Cb}][y_{Cb}]$ 이 1과 같으면, 다음 순차적인 단계가 적용된다:
- [0283] 1. 이웃 위치(x_{NbA} , y_{NbA}) 및 (x_{NbB} , y_{NbB})는 각각 ($x_{Cb} - 1$, $y_{Cb} + cbHeight - 1$) 및 ($x_{Cb} + cbWidth - 1$, $y_{Cb} - 1$)와 같도록 설정된다.
- [0284] 2. A 또는 B로 대체되는 X에 대해, 변수 $candIntraPredModeX$ 는 다음과 같이 유도된다:
 - [0285] - 블록에 대한 가용성 유도 프로세스가 호출된다: 이웃 블록 가용성 확인 프로세스는 (x_{Cb} , y_{Cb})와 같도록 설정되는 위치(x_{Curr} , y_{Curr}) 및 (x_{NbX} , y_{NbX})와 같도록 설정되는 이웃 위치(x_{NbY} , y_{NbY})를 입력으로 하여 호출되고, 출력은 $availableX$ 로 할당된다.
 - [0286] - 후보 인트라 예측 모드 $candIntraPredModeX$ 는 다음과 같이 유도된다:

- [0287] - 다음 조건 중 하나 이상이 참(true)이면, candIntraPredModeX는 INTRA_PLANAR과 같도록 설정된다.
- [0288] - 변수 availableX는 거짓(FALSE)과 같다.
- [0289] - CuPredMode[xNbX][yNbX]는 MODE_INTRA와 같지 않고 ciip_flag[xNbX][yNbX]는 1과 같지 않다.
- [0290] - pcm_flag[xNbX][yNbX]는 1과 같다.
- [0291] - X는 B와 같고 yCb - 1은 ((yCb >> CtbLog2SizeY) << CtbLog2SizeY)보다 작다.
- [0292] - 그렇지 않으면, candIntraPredModeX는 IntraPredModeY[xNbX][yNbX]과 같도록 설정된다.
- [0293] 여기서, X는 이웃 블록 A 또는 B로 대체되고, 루마 위치(xCb , yCb)는 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 현재 루마 코딩 블록의 좌측 상단 샘플을 지정하고, 이웃 A의 위치(xNbA, yNbA)는 (xCb - 1, yCb + cbHeight - 1)와 같도록 설정되고, 이웃 B의 위치(xNbB, yNbB)는 (xCb + cbWidth - 1, yCb - 1)와 같도록 설정되고, 변수 cbWidth는 루마 샘플에서 현재 코딩 블록의 폭을 지정하고, 변수 cbHeight는 루마 샘플에서 현재 코딩 블록의 높이를 지정하고, CuPredMode는 현재 예측 모드를 표현하고, SizeY는 코딩 트리 블록의 Y 콤포넌트의 크기를 표현한다.
- [0294] 3. x = 0..4인 candModeList[x]는 다음과 같이 유도된다:
- [0295] - candIntraPredModeB는 candIntraPredModeA과 같고 candIntraPredModeA는 INTRA_DC보다 크면, x = 0..4인 candModeList[x]는 다음과 같이 유도된다:
- [0296] candModeList[0] = candIntraPredModeA (8-10)
- [0297] candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64) (8-12)
- [0298] candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64) (8-13)
- [0299] candModeList[3] = INTRA_DC (8-11)
- [0300] candModeList[4] = 2 + ((candIntraPredModeA + 60) % 64) (8-14)
- [0301] - 그렇지 않으면, candIntraPredModeB는 candIntraPredModeA과 같지 않고 candIntraPredModeA 또는 candIntraPredModeB가 INTRA_DC보다 크면, 다음이 적용된다:
- [0302] - 변수 minAB 및 maxAB는 다음과 같이 유도된다:
- [0303] minAB = Min(candIntraPredModeA, candIntraPredModeB) (8-24)
- [0304] maxAB = Max(candIntraPredModeA, candIntraPredModeB) (8-25)
- [0305] - candIntraPredModeA 및 candIntraPredModeB이 둘다 INTRA_DC보다 크면, x = 0..4인 candModeList[x]는 다음과 같이 유도된다:
- [0306] candModeList[0] = candIntraPredModeA (8-27)
- [0307] candModeList[1] = candIntraPredModeB (8-29)
- [0308] candModeList[2] = INTRA_DC (8-29)
- [0309] - maxAB - minAB가 2 내지 62(포함)의 범위에 있으면, 다음이 적용된다:
- [0310] candModeList[3] = 2 + ((maxAB + 61) % 64) (8-30)
- [0311] candModeList[4] = 2 + ((maxAB - 1) % 64) (8-31)
- [0312] - 그렇지 않으면, 다음이 적용된다:
- [0313] candModeList[3] = 2 + ((maxAB + 60) % 64) (8-32)
- [0314] candModeList[4] = 2 + ((maxAB) % 64) (8-33)
- [0315] - 그렇지 않으면(candIntraPredModeA 또는 candIntraPredModeB이 INTRA_DC보다 큼), x = 0..4인 candModeList[x]는 다음과 같이 유도된다:

- [0316] $candModeList[0] = maxAB$ (8-65)
- [0317] $candModeList[1] = INTRA_DC$ (8-66)
- [0318] $candModeList[2] = 2 + ((maxAB + 61) \% 64)$ (8-66)
- [0319] $candModeList[3] = 2 + ((maxAB - 1) \% 64)$ (8-67)
- [0320] $candModeList[4] = 2 + ((maxAB + 60) \% 64)$ (8-68)
- [0321] - 그렇지 않으면, 다음이 적용된다:
- [0322] $candModeList[0] = INTRA_DC$ (8-71)
- [0323] $candModeList[1] = INTRA_ANGULAR50$ (8-72)
- [0324] $candModeList[2] = INTRA_ANGULAR18$ (8-73)
- [0325] $candModeList[3] = INTRA_ANGULAR46$ (8-74)
- [0326] $candModeList[4] = INTRA_ANGULAR54$ (8-75)
- [0327] 4. $IntraPredModeY[xCb][yCb]$ 는 다음의 절차를 적용하여 유도된다:
- [0328] - $intra_luma_mpm_flag[xCb][yCb]$ 이 1과 같으면, $IntraPredModeY[xCb][yCb]$ 는 $candModeList[intra_luma_mpm_idx[xCb][yCb]]$ 과 같도록 설정된다.
- [0329] - 그렇지 않으면, $IntraPredModeY[xCb][yCb]$ 는 다음 순차적인 단계를 적용하여 유도된다:
- [0330] 1. $i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 $candModeList[i]$ 이 $candModeList[j]$ 보다 큰 경우, 두 값은 다음과 같이 교환된다:
- [0331] $(candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j])$ (8-94)
- [0332] 2. $IntraPredModeY[xCb][yCb]$ 는 다음 순차적인 단계에 의해 유도된다:
- [0333] i. $IntraPredModeY[xCb][yCb]$ 는 $intra_luma_mpm_remainder[xCb][yCb]$ 과 같도록 설정된다.
- [0334] ii. $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다.
- [0335] iii. 0 내지 4(포함)와 같은 i 에 대해, $IntraPredModeY[xCb][yCb]$ 이 $candModeList[i]$ 보다 크거나 같은 경우, $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다.
- [0336] - 그렇지 않으면($intra_luma_not_planar_flag[xCb][yCb]$ 는 0과 같음), $IntraPredModeY[xCb][yCb]$ 는 $INTRA_PLANAR$ 과 같도록 설정된다.
- [0337] $x = xCb..xCb + cbWidth - 1$ 및 $y = yCb..yCb + cbHeight - 1$ 을 가지는 변수 $IntraPredModeY[x][y]$ 는 $IntraPredModeY[xCb][yCb]$ 와 같도록 설정된다.
- [0338] 한 예로서, 현재 블록의 인트라 예측 모드(예컨대 변수 명칭이 $luma_intra_mode$ 임)는 5-MPM 목록에서 평면 모드도 인트라 예측 모드도 아니며, 다음과 같이 그 값이 유도되고 이 인트라 예측 모드의 예측이 수행된다:
- [0339] 1. 비트스트림으로부터 $intra_luma_mpm_remainder$ 의 값을 유도하고, $luma_intra_mode$ 는 $intra_luma_mpm_remainder$ 의 값으로 설정된다.
- [0340] 2. $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 증가된다
- [0341] 3. 0 내지 4와 같은 i 에 대해(5-MPM 목록에서 모든 엔트리와 비교), $luma_intra_mode$ 이 $MPM[i]$ 보다 크거나 같은 경우(여기서, $MPM[0]$ 는 MPM 목록에서 첫 번째 엔트리에 대응하고, $MPM[1]$ 은 MPM 목록에서 두 번째 엔트리에 대응하고, 등), $luma_intra_mode$ 의 값이 1만큼 증가된다.
- [0342] 4. 유도된 인트라 예측 모드($luma_intra_mode$) 및 대응하는 참조 샘플을 사용하여 인트라 예측을 수행.
- [0343] (인코더 측)
- [0344] 한 예로서, 현재 블록의 인트라 예측 모드(예컨대 변수 명칭이 $IntraPredModeY[xCb][yCb]$ 임)는 5-MPM 목록에서 평면 모드도 인트라 예측 모드도 아니며, 그 값은 다음과 같이 유도된다:

- [0345] 1. $i = 0..3$ 그리고 각각의 i 에 대해, $j = (i + 1)..4$ 에 대해 $candModeList[i]$ 이 $candModeList[j]$ 보다 큰 경우, 두 값은 다음과 같이 교환된다:
- [0346] $(candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j])$ (8-94)
- [0347] 2. 4 내지 0(포함)과 같은 i 에 대해, $IntraPredModeY[xCb][yCb]$ 이 $candModeList[i]$ 보다 큰 경우, $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 감소된다.
- [0348] 3. $IntraPredModeY[xCb][yCb]$ 의 값이 1만큼 감소된다
- [0349] 4. $IntraPredModeY[xCb][yCb]$ 의 값은 절단 2진 2진화(binanzation) 프로세스를 사용하여 코드워드와 함께 비트스트림에 기록되고, 코드워드 내의 모든 빈(bins)은 바이패스 코딩된다.
- [0350] 인코더에서, 현재 모드는 이미 알고 있고, 유도되는 것은 코드워드(즉, $intra_luma_mpm_remainder$ 이다).
- [0351] 절단 2진 2진화(binanzation) 프로세스의 사양에 대해, 다음과 같이 규정된다(JVET-Q2001-VE.docx의 9.3.3.4): 절단 2진 (TB) 2진화(binanzation) 프로세스
- [0352] 이 프로세스에 대한 입력은 값 $synVal$ 및 $cMax$ 를 가지는 선택스 엘리먼트에 대한 TB 2진화(binanzation)에 대한 요청이다. 이 프로세스의 출력은 선택스 엘리먼트의 TB 2진화(binanzation)이다. 선택스 엘리먼트 $synVal$ 의 TB 2진화(binanzation) 프로세스의 빈(bin) 문자열은 다음과 같이 지정된다:
- [0353] $n = cMax + 1$
- [0354] $k = Floor(Log2(n))$ (1550)
- [0355] $u = (1 \ll (k + 1)) - n$
- [0356] - $synVal$ 이 u 보다 작으면, TB 빈(bin) 문자열은 $(1 \ll k) - 1$ 과 같은 $cMax$ 값을 가지는 $synVal$ 에 대해 9.3.3.7절에서 지정된 FL 2진화(binanzation) 프로세스를 호출함으로써 유도된다.
- [0357] - 그렇지 않으면($synVal$ 이 u 보다 크거나 같음), TB 빈(bin) 문자열은 $(1 \ll (k + 1)) - 1$ 과 같은 $cMax$ 값을 가지는 $(synVal + u)$ 에 대해 9.3.3.7절에서 지정된 FL 2진화(binanzation) 프로세스를 호출함으로써 유도된다.
- [0358] 전술한 실시예의 하나의 구현예로서, 평면 모드는 목시적 6-MPM 목록에서 항상 첫 번째 엔트리인 것으로 목시적으로 간주될 수 있다. 이 목시적 6-MPM 목록은 평면(항상 첫 번째 엔트리) 및 5개의 다른 모드(5-MPM 목록으로서 명시적으로 기술될 수 있음)로 구성된다.
- [0359] 이는, 평면 모드가 항상 목시적 6-MPM 목록의 첫 번째 위치에 있으면, 목시적 MPM 목록으로부터 분리될 수 있고, 그리고 특수 플래그(일레로 플래그 $intra_luma_planar_flag$ 또는 플래그 $intra_luma_not_planar_flag$)가 할당되어 현재 블록의 인트라 예측 모드가 평면인지 아닌지 여부를 표시하기 때문이다.
- [0360] 한 예로서, 평면 모드가 현재 블록의 인트라 예측 모드와 같은지 여부를 표시하기 위한 특수 플래그로서 $intra_luma_not_planar_flag[x0][y0]$ 를 도입하는 것을 고려한다. 평면 모드인 경우, $intra_luma_not_planar_flag$ 는 0으로 설정되고, 그렇지 않으면 1로 설정된다. 이 경우, mpm_idx 의 첫 번째 빈(bin)은 특수 플래그 $intra_luma_not_planar_flag$ 와 동등하다.
- [0361] 현재 블록의 인트라 예측 모드가 평면(planar)인 경우, MPM 인덱스는 0과 같고, 코딩된 빈(bin)은 0이다. 현재 블록의 인트라 예측 모드가 평면이 아닌(Not planar) 경우, MPM 인덱스는 0과 같지 않고, MPM 인덱스의 첫 번째 빈(bin)은 항상 1과 같다.
- [0362] 예로서, $intra_luma_mpm_idx$ 는 다음의 절단 단항(truncated unary, TR) 코드를 사용하여 코딩된다,
- [0363] Value TR
- [0364] Mpm_idx 0 0
- [0365] Mpm_idx 1 1 0
- [0366] Mpm_idx 2 1 1 0
- [0367] Mpm_idx 3 1 1 1 0

- [0368] Mpm_idx 4 1 1 1 1.
- [0369] 본 발명의 한 실시예(비-각형 브랜치(non-angular branch))에 따르면:
- [0370] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0371] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0372] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0373] · Left_mode는 각도 모드가 아니고 above_mode는 각도 모드가 아니다
- [0374] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0375] MPM[1]: DC
- [0376] MPM[2]: 수직 모드 (즉, 50)
- [0377] MPM[3]: 수평 모드 (즉, 18)
- [0378] MPM[4]: 수직-4 (즉, 46)
- [0379] MPM[5]: 수직+4 (즉, 54)
- [0380] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0381] 본 발명의 한 실시예(비-각형 브랜치(non-angular branch))에 따르면:
- [0382] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0383] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0384] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0385] · Left_mode는 각도 모드가 아니고 above_mode는 각도 모드가 아니다
- [0386] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0387] MPM[1]: DC
- [0388] MPM[2]: 수직 모드 (즉, 50)
- [0389] MPM[3]: 수평 모드 (즉, 18)
- [0390] MPM[4]: 66
- [0391] MPM[5]: 2
- [0392] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0393] 본 발명의 한 실시예(비-각형 브랜치(non-angular branch))에 따르면:
- [0394] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라

라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.

- [0395] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0396] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0397] · Left_mode는 각도 모드가 아니고 above_mode는 각도 모드가 아니다
- [0398] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0399] MPM[1]: DC
- [0400] MPM[2]: 수직 모드 (즉, 50)
- [0401] MPM[3]: 수평 모드 (즉, 18)
- [0402] MPM[4]: 2
- [0403] MPM[5]: 34
- [0404] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0405] 본 발명의 한 실시예(비-각형 브랜치(non-angular branch))에 따르면:
- [0406] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0407] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0408] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0409] · Left_mode는 각도 모드가 아니고 above_mode는 각도 모드가 아니다
- [0410] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0411] MPM[1]: DC
- [0412] MPM[2]: 수직 모드 (즉, 50)
- [0413] MPM[3]: 수평 모드 (즉, 18)
- [0414] MPM[4]: 66
- [0415] MPM[5]: 34
- [0416] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0417] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 하나는 각형 나머지는 비-각형)에 따르면:
- [0418] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0419] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0420] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0421] · left_mode 및 above_mode 중 하나는 각도 모드이고, 다른 하나는 비-각도 모드(non-angular mode)이다.

- [0422] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0423] 각도 모드를 ang로 정의하고 다음과 같이 획득한다:
- [0424] left_mode이 각도 모드이면, ang=left mode
- [0425] 그렇지 않으면(above mode가 각도 모드임) ang=above mode
- [0426] MPM[1]: ang
- [0427] MPM[2]: DC
- [0428] MPM[3]: $2 + ((ang + 61) \% 64)$
- [0429] MPM[4]: $2 + ((ang - 1) \% 64)$
- [0430] MPM[5]: $2 + ((ang + 60) \% 64)$
- [0431] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0432] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 하나는 각형 나머지는 비-각형)에 따르면:
- [0433] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0434] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0435] · left_mode 및 above_mode 중 하나는 각도 모드이고, 다른 하나는 비-각도 모드(non-angular mode)이다.
- [0436] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0437] 각도 모드를 ang로 정의하고 다음과 같이 획득한다:
- [0438] left_mode이 각도 모드이면, ang=left mode
- [0439] 그렇지 않으면(above mode가 각도 모드임) ang=above mode
- [0440] MPM[1]: DC
- [0441] MPM[2]: ang
- [0442] MPM[3]: $2 + ((ang + 61) \% 64)$
- [0443] MPM[4]: $2 + ((ang - 1) \% 64)$
- [0444] MPM[5]: $2 + ((ang + 60) \% 64)$
- [0445] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0446] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 하나는 각형 나머지는 비-각형)에 따르면:
- [0447] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0448] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0449] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,

- [0450] · left_mode 및 above_mode 중 하나는 각도 모드이고, 다른 하나는 비-각도 모드(non-angular mode)이다.
- [0451] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0452] 각도 모드를 ang로 정의하고 다음과 같이 획득한다:
- [0453] left_mode이 각도 모드이면, ang=left mode
- [0454] 그렇지 않으면(above mode가 각도 모드임) ang=above mode
- [0455] MPM[1]: ang
- [0456] MPM[2]: $2 + ((ang + 61) \% 64)$
- [0457] MPM[3]: DC
- [0458] MPM[4]: $2 + ((ang - 1) \% 64)$
- [0459] MPM[5]: $2 + ((ang + 60) \% 64)$
- [0460] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0461] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 하나는 각형 나머지는 비-각형)에 따르면:
- [0462] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0463] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0464] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0465] · left_mode 및 above_mode 중 하나는 각도 모드이고, 다른 하나는 비-각도 모드(non-angular mode)이다.
- [0466] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0467] 각도 모드를 ang로 정의하고 다음과 같이 획득한다:
- [0468] left_mode이 각도 모드이면, ang=left mode
- [0469] 그렇지 않으면(above mode가 각도 모드임) ang=above mode
- [0470] MPM[0]: 평면(Planar)
- [0471] MPM[1]: ang
- [0472] MPM[2]: $2 + ((ang + 61) \% 64)$
- [0473] MPM[3]: $2 + ((ang - 1) \% 64)$
- [0474] MPM[4]: DC
- [0475] MPM[5]: $2 + ((ang + 60) \% 64)$
- [0476] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0477] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 하나는 각형 나머지는 비-각형)에 따르면:
- [0478] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나

나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.

- [0479] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0480] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
 - [0481] · left_mode 및 above_mode 중 하나는 각도 모드이고, 다른 하나는 비-각도 모드(non-angular mode)이다.
- [0482] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0483] 각도 모드를 ang로 정의하고 다음과 같이 획득한다:
 - [0484] left_mode이 각도 모드이면, $ang = \text{left mode}$
 - [0485] 그렇지 않으면(above mode가 각도 모드임) $ang = \text{above mode}$
 - [0486] MPM[1]: ang
 - [0487] $MPM[2]: 2 + ((ang + 61) \% 64)$
 - [0488] $MPM[3]: 2 + ((ang - 1) \% 64)$
 - [0489] $MPM[4]: 2 + ((ang + 60) \% 64)$
 - [0490] MPM[5]: DC
- [0491] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0492] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 둘은 동일한 각형)에 따르면:
 - [0493] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
 - [0494] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
 - [0495] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
 - [0496] · left_mode 및 above_mode는 모두 각도 모드이고 이들은 동일하다.
 - [0497] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
 - [0498] MPM[1]: left mode
 - [0499] $MPM[2]: 2 + ((left\ mode + 61) \% 64)$
 - [0500] $MPM[3]: 2 + ((left\ mode - 1) \% 64)$
 - [0501] MPM[4]: DC
 - [0502] $MPM[5]: 2 + ((left\ mode + 60) \% 64)$
 - [0503] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
 - [0504] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 둘은 동일한 각형)에 따르면:
 - [0505] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라

라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.

[0506] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.

[0507] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,

[0508] · left_mode 및 above_mode는 모두 각도 모드이고 이들은 동일하다.

[0509] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :

[0510] MPM[1]: DC

[0511] MPM[2]: left mode

[0512] MPM[3]: $2 + ((\text{left mode} + 61) \% 64)$

[0513] MPM[4]: $2 + ((\text{left mode} - 1) \% 64)$

[0514] MPM[5]: $2 + ((\text{left mode} + 60) \% 64)$

[0515] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.

[0516] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 둘은 동일한 각형)에 따르면:

[0517] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.

[0518] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.

[0519] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,

[0520] · left_mode 및 above_mode는 모두 각도 모드이고 이들은 동일하다.

[0521] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :

[0522] MPM[1]: left mode

[0523] MPM[2]: DC

[0524] MPM[3]: $2 + ((\text{left mode} + 61) \% 64)$

[0525] MPM[4]: $2 + ((\text{left mode} - 1) \% 64)$

[0526] MPM[5]: $2 + ((\text{left mode} + 60) \% 64)$

[0527] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.

[0528] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 둘은 동일한 각형)에 따르면:

[0529] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.

[0530] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.

[0531] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,

[0532] · left_mode 및 above_mode는 모두 각도 모드이고 이들은 동일하다.

[0533] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를

들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :

- [0534] MPM[1]: left mode
- [0535] MPM[2]: $2 + ((\text{left mode} + 61) \% 64)$
- [0536] MPM[3]: DC
- [0537] MPM[4]: $2 + ((\text{left mode} - 1) \% 64)$
- [0538] MPM[5]: $2 + ((\text{left mode} + 60) \% 64)$
- [0539] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0540] 본 발명의 한 실시예(일-각형 브랜치(one-angular branch), 둘은 동일한 각형)에 따르면:
- [0541] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0542] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0543] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0544] · left_mode 및 above_mode는 모두 각도 모드이고 이들은 동일하다.
- [0545] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0546] MPM[1]: left mode
- [0547] MPM[2]: $2 + ((\text{left mode} + 61) \% 64)$
- [0548] MPM[3]: $2 + ((\text{left mode} - 1) \% 64)$
- [0549] MPM[4]: $2 + ((\text{left mode} + 60) \% 64)$
- [0550] MPM[5]: DC
- [0551] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0552] 본 발명의 한 실시예(2-각형 브랜치(two-angular branch))에 따르면:
- [0553] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0554] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0555] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0556] · left_mode 및 above_mode는 2개의 상이한 각도 모드이다.
- [0557] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0558] MPM[1]: left_mode
- [0559] MPM[2]: above_mode
- [0560] MPM[3]: DC

- [0561] - 변수 minAB 및 maxAB는 다음과 같이 유도된다:
- [0562] $\text{minAB} = \text{Min}(\text{above_mode}, \text{left_mode})$
- [0563] $\text{maxAB} = \text{Max}(\text{above_mode}, \text{left_mode})$
- [0564] - $\text{maxAB} - \text{minAB}$ 가 2 내지 62(포함)의 범위에 있으면, 다음이 적용된다:
- [0565] $\text{MPM}[4] = 2 + ((\text{maxAB} + 61) \% 64)$ (8-30)
- [0566] $\text{MPM}[5] = 2 + ((\text{maxAB} - 1) \% 64)$ (8-31)
- [0567] - 그렇지 않으면, 다음이 적용된다:
- [0568] $\text{MPM}[4] = 2 + ((\text{maxAB} + 60) \% 64)$ (8-32)
- [0569] $\text{MPM}[5] = 2 + ((\text{maxAB}) \% 64)$ (8-33)
- [0570] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0571] 본 발명의 한 실시예(2-각형 브랜치(two-angular branch))에 따르면:
- [0572] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.
- [0573] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.
- [0574] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,
- [0575] · left_mode 및 above_mode는 2개의 상이한 각도 모드이다.
- [0576] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :
- [0577] MPM[1]: above_mode
- [0578] MPM[2]: left_mode
- [0579] MPM[3]: DC
- [0580] - 변수 minAB 및 maxAB는 다음과 같이 유도된다:
- [0581] $\text{minAB} = \text{Min}(\text{above_mode}, \text{left_mode})$
- [0582] $\text{maxAB} = \text{Max}(\text{above_mode}, \text{left_mode})$
- [0583] - $\text{maxAB} - \text{minAB}$ 가 2 내지 62(포함)의 범위에 있으면, 다음이 적용된다:
- [0584] $\text{MPM}[4] = 2 + ((\text{maxAB} + 61) \% 64)$ (8-30)
- [0585] $\text{MPM}[5] = 2 + ((\text{maxAB} - 1) \% 64)$ (8-31)
- [0586] - 그렇지 않으면, 다음이 적용된다:
- [0587] $\text{MPM}[4] = 2 + ((\text{maxAB} + 60) \% 64)$ (8-32)
- [0588] $\text{MPM}[5] = 2 + ((\text{maxAB}) \% 64)$ (8-33)
- [0589] ○ 단계 6: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.
- [0590] 본 발명의 한 실시예(2-각형 브랜치(two-angular branch))에 따르면:
- [0591] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라

라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.

[0592] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.

[0593] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,

[0594] · left_mode 및 above_mode는 2개의 상이한 각도 모드이다.

[0595] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :

[0596] MPM[1]: left_mode

[0597] MPM[2]: DC

[0598] MPM[3]: above_mode

[0599] - 변수 minAB 및 maxAB는 다음과 같이 유도된다:

[0600] $minAB = \text{Min}(above_mode, left_mode)$

[0601] $maxAB = \text{Max}(above_mode, left_mode)$

[0602] - $maxAB - minAB$ 가 2 내지 62(포함)의 범위에 있으면, 다음이 적용된다:

[0603] $MPM[4] = 2 + ((maxAB + 61) \% 64) (8-30)$

[0604] $MPM[5] = 2 + ((maxAB - 1) \% 64) (8-31)$

[0605] - 그렇지 않으면, 다음이 적용된다:

[0606] $MPM[4] = 2 + ((maxAB + 60) \% 64) (8-32)$

[0607] $MPM[5] = 2 + ((maxAB) \% 64) (8-33)$

[0608] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다.

[0609] 본 발명의 한 실시예(2-각형 브랜치(two-angular branch))에 따르면:

[0610] ○ 단계 1: 도 10에 따른 좌측 이웃 블록의 인트라 예측 모드 및 상측 이웃 블록의 인트라 예측 모드를 획득한다. 이웃 블록의 인트라 예측 모드가 사용 가능하지 않으면(예를 들어, 이웃 블록이 화상 경계를 벗어나거나 이웃 블록이 인터 코딩됨), 이웃 블록의 인트라 예측 모드는 평면(planar)으로 설정된다. 좌측 이웃의 인트라 예측 모드는 left_mode로 명명하고 상측 이웃 블록의 인트라 예측 모드는 above_mode로 명명한다.

[0611] ○ 단계 2: 이웃 블록의 인트라 예측 모드가 각도 모드인지 아닌지 여부를 결정한다.

[0612] ○ 단계 3: 다음 조건(점 표시 항목)의 모두가 참으로 평가되면,

[0613] · left_mode 및 above_mode는 2개의 상이한 각도 모드이다.

[0614] 그러면, 인트라 예측 모드의 N개의 엔트리를 다음과 같이 포함하는 인트라 예측 모드의 목록이 구성된다(예를 들어, N은 5와 같으며, 여기서, 첫 번째 엔트리는 MPM[1]에 대응하고, 두 번째 엔트리는 MPM[2]에 대응하고, 등) :

[0615] MPM[1]: above_mode

[0616] MPM[2]: DC

[0617] MPM[3]: left_mode

[0618] - 변수 minAB 및 maxAB는 다음과 같이 유도된다:

[0619] $minAB = \text{Min}(above_mode, left_mode)$

[0620] $maxAB = \text{Max}(above_mode, left_mode)$

[0621] - $maxAB - minAB$ 가 2 내지 62(포함)의 범위에 있으면, 다음이 적용된다:

- [0622] $MPM [4] = 2 + ((\max AB + 61) \% 64)$ (8-30)
- [0623] $MPM [5] = 2 + ((\max AB - 1) \% 64)$ (8-31)
- [0624] - 그렇지 않으면, 다음이 적용된다:
- [0625] $MPM [4] = 2 + ((\max AB + 60) \% 64)$ (8-32)
- [0626] $MPM [5] = 2 + ((\max AB) \% 64)$ (8-33)
- [0627] ○ 단계 4: 구성된 목록의 엔트리와 인덱스(MPM 인덱스로 명명됨)에 따라 현재 블록을 예측한다. ○
4단계: 인덱스(MPM 인덱스로 표시)와 구성된 목록의 항목에 따라 현재 블록을 예측한다.
- [0628] 실시예의 하나의 구현예로서, N은 6과 같도록 설정된다.
- [0629] 실시예의 하나의 구현예로서, N은 6과 같도록 설정된다. 이 경우, 인트라 예측 모드의 목록은 처음 5개의 엔트리를 위에서 나열된 대로 포함하고 6번째 엔트리는 포함하지 않는다.
- [0630] 실시예의 하나의 구현예로서, N은 4와 같도록 설정된다. 이 경우, 인트라 예측 모드의 목록은 처음 4개의 엔트리를 위에서 나열된 대로 포함하고 5번째 엔트리 및 6번째 엔트리는 포함하지 않는다.
- [0631] 실시예의 하나의 구현예로서, 상기 인트라 예측 모드의 목록은 MPM 목록으로 지칭된다.
- [0632] 구현예에서, 상기 목록 인덱스(MPM 인덱스)는 비트스트림에서 인디케이터로서 시그널링된다. MPM 인덱스는 컨텍스트 적응형 엔트로피 코더(Context Adaptive Entropy Coder, CABAC)로 코딩될 수 있어 유리하다. MPM 인덱스는 다양한 확률 모델(달리 말하면, 컨텍스트)에 따라 CABAC에 의해 코딩될 수 있다.
- [0633] 하나의 예로서, `mpm_index`에 대한 코드워드의 첫 번째 bin(bin)은 CABAC 컨텍스트 코딩된다.
- [0634] 한 예로서, 이의 컨텍스트는 현재 인트라 블록이 다중 참조 라인, ISP 또는 일반 인트라 블록으로 적용되는지 여부에 따라 결정된다. 총 3개의 컨텍스트가 `mpm_index`에 대한 코드워드의 첫 번째 bin(bin)의 CABAC 컨텍스트에 대해 생성된다.
- [0635] MPM 목록이 평면 및 DC 모드를 모두 포함하면, 구성된 MPM 목록으로부터 평면 및 DC 모드가 제외된 MPM 목록이 다중 참조 라인 코딩 도구에 의해 사용된다. 이 경우, 4-MPM 목록이 다중 참조 라인의 인트라 예측 모드 코딩에 의해 사용된다. 한 예로서, MPM 목록 구성 후 MPM 목록은 {평면(0의 값), DC(1의 값), VER(50의 값), HOR(18의 값), VER-4(46의 값), VER+4(54의 값)}이며, {VER, HOR, VER-4, VER+4}의 4-MPM 목록은 다중 참조 라인이 활성화되는 경우 인트라 예측 모드 코딩에 의해 사용된다.
- [0636] MPM 목록이 평면 모드를 포함하나 DC 모드를 포함하지 않으면, 구성된 MPM 목록으로부터 평면(planar)을 제외한 MPM 목록이 다중 참조 라인 코딩 도구에 의해 사용된다. 이 경우, 5-MPM 목록이 다중 참조 라인의 인트라 예측 모드 코딩에 의해 사용된다. 한 예로서, MPM 목록 구성 후 MPM 목록은 {평면, VER, HOR, 43, 60, 3}이며, {VER, HOR, 43, 60, 3}의 5-MPM 목록은 다중 참조 라인이 활성화되는 경우 인트라 예측 모드 코딩에 의해 사용된다.
- [0637] 다른 예에서 DC 모드가 없는 MPM 목록에 대한 선택 수정이 필요할 수 있다. MPM 목록이 DC를 포함하지 않으면, 아마도 DC 모드는 비-MPM 브랜치(즉, `mpm_flag`는 거짓(false)임)에서 매우 자주 사용될 것이다. 이 경우, 새로운 선택, `DC_mode`이 도입된다. `DC_mode`이 1과 같은 것은 현재 블록의 인트라 예측 모드가 DC 모드임을 나타낸다. `DC_mode`이 0과 같은 것은 현재 블록의 인트라 예측 모드가 DC 모드가 아님을 나타낸다.
- [0638] 즉, 비-MPM의 이전 선택은,
 If (`mpm_flag`)
 ...
 else
 intra_mode (using TB)
- [0639]

[0640]

로부터

If (mpm_flag)

...

else

DC_mode

If (DC_mode==0)

[0641]

intra_mode (using TB)

[0642]

로 변경된다.

[0643]

신택스 DC_mode는 컨텍스트 적응형 엔트로피 코더(Context Adaptive Entropy Coder, CABAC)로 코딩될 수 있다. DC_mode는 다양한 확률 모델(달리 말하면, 컨텍스트)에 따라 CABAC에 의해 코딩될 수 있다. DC_Mode의 도입으로, 절단 2진에 대한 최대 값은 60, 즉, 67(총 모드 수) - 6(MPM) -1(DC)이다. DC_mode이 도입되지 않는 경우, 절단 2진에 대한 최대 값은 61, 즉, 67(총 모드 수) - 6(MPM)이다.

[0644]

특히, 디코딩 디바이스에 의해 구현되는 현재 블록의 예측 코딩의 다음의 방법. 디코딩 디바이스는 도 1a의 비디오 디코더(30), 또는 도 3의 디코더(30)일 수 있다.

[0645]

실시예(1200)(도 12 참조)에 따르면, 디코딩 디바이스에 의해 구현되는 현재 블록의 인트라 예측을 위한 방법은 다음 단계를 포함한다. 단계 1201, 디바이스는 현재 블록의 표시 정보의 값을 획득한다. 표시 정보는 플래그, 예를 들어, intra_luma_mpm_flag에 의해 표시된다.

[0646]

단계 1202, 디바이스는 표시 정보의 값이 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함됨을 표시하는지 여부를 결정한다. 위에서 설명한 바와 같이, intra_luma_mpm_flag의 값이 1과 같은 경우, 현재 블록의 인트라 예측 모드는 가장 가능성 있는 모드의 세트에 포함된다. intra_luma_mpm_flag의 값이 0과 같은 경우, 현재 블록의 인트라 예측 모드는 가장 가능성 있는 모드의 세트에 포함되지 않는다.

[0647]

단계 1203, 표시 정보의 값이 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되지 않음을 표시하는 경우 디바이스는 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]를 다음 순차적인 단계에 의해 유도한다:

[0648]

i. IntraPredModeY[xCb][yCb]는 intra_luma_mpm_remainder[xCb][yCb]과 같도록 설정된다.

[0649]

ii. IntraPredModeY[xCb][yCb]의 값이 1만큼 증가된다.

[0650]

iii. 0 내지 4(포함)와 같은 i 에 대해, IntraPredModeY[xCb][yCb]이 candModeList[i]보다 크거나 같은 경우, IntraPredModeY[xCb][yCb]의 값이 1만큼 증가된다.

[0651]

가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 candModeList[i]는,

[0652]

$i = 0..3$ 그리고 각각의 i에 대해, $j = (i + 1)..4$ 에 대해 candModeList[i]이 candModeList[j]보다 큰 경우, 두 값은 다음과 같이 교환됨:

[0653]

(candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j])에 의해 유도된다.

[0654]

단계 1204, 디바이스는 유도된 인트라 예측 모드(luma_intra_mode) 및 대응하는 참조 샘플을 사용하여 인트라 예측을 수행한다.

[0655]

본 실시예(1200)의 세부 정보는 위에서 언급한 실시예에 나타나 있다..

[0656]

실시예(1300)(도 13 참조)에 따르면, 인코딩 디바이스에 의해 구현되는 현재 블록의 인트라 예측을 위한 방법은 다음 단계를 포함한다. 인코딩 디바이스는 도 1a의 비디오 인코더(20), 또는 도 2의 인코더(20)일 수 있다.

[0657]

단계 1301, 디바이스는 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 결정한다. 단계 1302, 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]이 가장 가능성 있는 모드의 세트에 포함되지 않는 경우 디바이스는 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]를 사용하여 다음 순차적인 단계에 의해 변수를 유도한다:

- [0658] i. 0 내지 4(포함)와 같은 i 에 대해, IntraPredModeY[xCb][yCb]이 candModeList[i]보다 큰 경우, IntraPredModeY[xCb][yCb]의 값이 1만큼 감소된다;
- [0659] ii. IntraPredModeY[xCb][yCb]의 값이 1만큼 감소된다.
- [0660] 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드 candModeList[i]는,
- [0661] $i = 0..3$ 그리고 각각의 i에 대해, $j = (i + 1)..4$ 에 대해 candModeList[i]이 candModeList[j]보다 큰 경우, 두 값은 다음과 같이 교환됨:
- [0662] (candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j]) 에 의해 유도된다.
- [0663] 단계 1303, 디바이스는 비트스트림을 인코딩하고(여기서 비트스트림은 변수, 예를 들어, intra_luma_mpm_remainder[xCb][yCb]의 정보를 포함함), 그리고 인코딩된 비트스트림을 디코딩 디바이스로 송신한다.
- [0664] 본 실시예(1300)의 세부 정보는 위에서 언급한 실시예에 나타나 있다..
- [0665] 도 14는 디코딩 디바이스(1400)의 실시예를 도시한다. 디코딩 디바이스(1400)는 도 1a의 비디오 디코더(30), 또는 도 3의 디코더(30)일 수 있다. 디코딩 디바이스(1400)는 실시예(1200) 및 전술한 다른 실시예를 구현하는데 사용될 수 있다.
- [0666] 디바이스는 획득 유닛(1401) 및 유도 유닛(1402)을 포함한다. 획득 유닛(1401)은 현재 블록의 표시 정보의 값을 획득하도록 구성되며, 표시 정보의 값은 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되는지 여부를 표시한다.
- [0667] 유도 유닛(1402)은, 표시 정보의 값이 현재 블록의 인트라 예측 모드가 가장 가능성 있는 모드의 세트에 포함되지 않음을 표시하는 경우 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]을 다음 순차적인 단계에 의해 유도하도록 구성되며:
- [0668] i. IntraPredModeY[xCb][yCb]는 intra_luma_mpm_remainder[xCb][yCb]과 같도록 설정된다,
- [0669] ii. IntraPredModeY[xCb][yCb]의 값이 1만큼 증가된다;
- [0670] 여기서, intra_luma_mpm_remainder[xCb][yCb]는 나머지 인트라 예측 모드를 표현하고, 루마 위치(xCb , yCb)는 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 현재 블록의 좌측 상단 샘플을 지정한다.
- [0671] 디바이스(1400)는 예측 유닛(1403)(도 14에는 도시하지 않음)을 추가로 포함한다. 예측 유닛(1403)은 유도된 인트라 예측 모드(luma_intra_mode) 및 대응하는 참조 샘플을 사용하여 인트라 예측을 수행하도록 구성된다.
- [0672] 도 15는 인트라 예측을 위한 인코딩 디바이스(1500)의 실시예를 도시한다. 인코딩 디바이스(1500)는 도 1a의 비디오 인코더(20), 또는 도 2의 인코더(20)일 수 있다. 인코딩 디바이스(1500)는 실시예(1300) 및 전술한 다른 실시예를 구현하는데 사용될 수 있다.
- [0673] 디바이스(1500)는 유도 유닛(1501) 및 전송 유닛(1502)을 포함한다. 유도 유닛(1501)은, 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]이 가장 가능성 있는 모드의 세트에 포함되지 않는 경우 현재 블록의 인트라 예측 모드 IntraPredModeY[xCb][yCb]를 사용하여 다음 순차적인 단계에 의해 변수를 유도하도록 구성되며:
- [0674] i. 0 내지 4(포함)와 같은 i 에 대해, IntraPredModeY[xCb][yCb]이 candModeList[i]보다 큰 경우, IntraPredModeY[xCb][yCb]의 값은 1만큼 감소된다-여기서, candModeList[i]는 가장 가능성 있는 모드의 세트에서 후보 인트라 예측 모드를 표현함-;
- [0675] ii. IntraPredModeY[xCb][yCb]의 값이 1만큼 감소된다;
- [0676] 여기서, 루마 위치(xCb , yCb)는 현재 화상의 좌측 상단 루마 샘플에 대해 상대적인 현재 블록의 좌측 상단 샘플을 지정한다.
- [0677] 전송 유닛(1502)은 비트스트림을 디코더로 송신하도록 구성되며, 여기서 비트스트림은 변수의 정보를 포함한다.
- [0678] 평면 모드가 MPM 목록에 있는 것으로 (묵시적 또는 명시적) 항시 간주되고 플래그

(intra_luma_not_planar_flag)로서 표시되는 경우, 이 발명의 실시예는 비-MPM 모드 유도 과정을 단순화한다. 평면(planar)는 인트라 예측 모드의 가장 작은 값에 대응하므로, 비-MPM 모드에 대한 유도 과정이 단순화될 수 있다. 비교 및 증가 단계의 경우, 평면(planar)는 항상 MPM 목록에 있으므로 평면 모드에 대한 비교가 저장될 수 있고, 따라서 나머지 인트라 예측 모드의 초기값은, 전술한 단계 ii에 대응하여, 1만큼 직접 증가될 수 있다.

[0679] 다음은, 전술한 실시예에서 나타낸 인코딩 방법 및 디코딩 방법의 적용 및 이를 사용하는 시스템에 대한 설명이다.

[0680] 도 16은 콘텐츠 배포 서비스를 실현하기 위한 콘텐츠 공급 시스템(3100)을 나타내는 블록도이다. 이 콘텐츠 공급 시스템(3100)은 캡처 디바이스(3102), 단말 디바이스(3106)를 포함하고, 선택사항으로 디스플레이(3126)를 포함한다. 캡처 디바이스(3102)는 통신 링크(3104)를 통해 단말 디바이스(3106)와 통신한다. 통신 링크는 전술한 통신 채널(13)을 포함할 수 있다. 통신 링크(3104)는 WIFI, 이더넷, 케이블, 무선(3G/4G/5G), USB 또는 이들의 임의의 종류의 조합 등을 포함하지만 이에 제한되지 않는다.

[0681] 캡처 디바이스(3102)는 데이터를 생성하고, 상기 실시예에서 도시된 바와 같은 인코딩 방법에 의해 데이터를 인코딩할 수 있다. 대안적으로, 캡처 디바이스(3102)는 데이터를 스트리밍 서버(도면에는 도시하지 않음)에 배포할 수 있고, 서버는 데이터를 인코딩하고 인코딩된 데이터를 단말 디바이스(3106)로 전송한다. 캡처 디바이스(3102)은 카메라, 스마트폰 또는 패드, 컴퓨터 또는 랩탑, 화상 회의 시스템, PDA, 차량 탑재 디바이스, 또는 이들의 임의의 조합, 등을 포함하나 이에 제한되지는 않는다. 예를 들어, 캡처 디바이스(3102)는 전술한 바와 같은 소스 장치(12)를 포함할 수 있다. 데이터가 비디오를 포함하는 경우, 캡처 디바이스(3102)에 포함된 비디오 인코더(20)는 실제로 비디오 인코딩 처리를 수행할 수 있다. 데이터가 오디오(즉, 음성)를 포함하는 경우, 캡처 디바이스(3102)에 포함된 오디오 인코더는 실제로 오디오 인코딩 처리를 수행할 수 있다. 일부 실제 시나리오의 경우, 캡처 디바이스(3102)는 인코딩된 비디오 및 오디오 데이터를 함께 멀티플렉싱하여 분배한다. 다른 실제 시나리오의 경우, 일례로 화상 회의 시스템에서, 인코딩된 오디오 데이터 및 인코딩된 비디오 데이터는 멀티플렉싱되지 않는다. 캡처 디바이스(3102)는 인코딩된 오디오 데이터 및 인코딩된 비디오 데이터를 단말 디바이스(3106)에 개별적으로 배포한다.

[0682] 콘텐츠 공급 시스템(3100)에서, 단말 디바이스(310)는 인코딩된 데이터를 수신하여 재생한다. 단말 디바이스(3106)는, 스마트폰 또는 패드(3108), 컴퓨터 또는 랩탑(3110), 네트워크 비디오 레코더(NVR)/디지털 비디오 레코더(DVR)(3112), TV(3114), 셋톱 박스(STB)(3116), 화상 회의 시스템(3118), 비디오 감시 시스템(3120), 개인용 디지털 단말기(PDA)(3122), 차량 탑재 디바이스(3124), 또는 이들의 임의의 조합, 등과 같이 디코딩 전술한 인코딩된 데이터를 디코딩할 수 있는, 데이터 수신 및 복구 능력을 가지는 디바이스일 수 있다. 예를 들어, 단말 디바이스(3106)는 전술한 바와 같은 테스트네이션 디바이스(14)를 포함할 수 있다. 인코딩된 데이터가 비디오를 포함하는 경우, 단말 디바이스에 포함된 비디오 디코더(30)는 비디오 디코딩을 수행하기 위해 우선 순위를 가진다. 인코딩된 데이터가 오디오를 포함하는 경우, 단말 디바이스에 포함된 오디오 디코더는 오디오 디코딩 처리를 수행하기 위해 우선 순위를 가진다.

[0683] 예를 들어, 스마트폰 또는 패드(3108), 컴퓨터 또는 랩탑(3110), 네트워크 비디오 레코더(NVR)/디지털 비디오 레코더(DVR)(3112), TV(3114), 개인용 디지털 단말기(PDA)(3122), 또는 차량 탑재 디바이스(3124)와 같은, 디스플레이를 가지는 단말 디바이스의 경우, 단말 디바이스는 디코딩된 데이터를 그의 디스플레이에 공급할 수 있다. 예를 들어, STB(3116), 화상 회의 시스템(3118), 또는 비디오 감시 시스템(3120)와 같은, 디스플레이가 없는 단말 디바이스의 경우 외부 디스플레이(3126)가 내부에 접촉하여 디코딩된 데이터를 수신하여 보여줄 수 있다.

[0684] 이 시스템 내의 각각의 디바이스가 인코딩 또는 디코딩을 수행하는 경우, 전술한 실시예에서 도시된 바와 같은 화상 인코딩 디바이스 또는 화상 디코딩 디바이스가 사용될 수 있다.

[0685] 도 17은 단말 디바이스(3106)의 일례의 구조를 나타내는 도면이다. 단말 디바이스(3106)가 캡처 디바이스(3102)로부터 스트림을 수신한 후, 프로토콜 처리 유닛(3202)은 스트림의 전송 프로토콜을 분석한다. 프로토콜은, RTSP(Real Time Streaming Protocol), HTTP(Hyper Text Transfer Protocol), HLS(HTTP Live Streaming Protocol), MPEG-DASH, RTP(Real-time Transport Protocol), RTMP(Real Time Messaging Protocol), 또는 임의의 종류의 이들의 조합, 등을 포함하나 이에 제한되지는 않는다.

[0686] 프로토콜 진행 유닛(3202)가 스트림을 처리 한 후, 스트림 파일이 생성된다. 파일은 디멀티플렉싱 유닛(3204)으

로 출력된다. 디멀티플렉싱 유닛(3204)은 멀티플렉싱된 데이터를 인코딩된 오디오 데이터 및 인코딩된 비디오 데이터로 분리할 수 있다. 위에서 설명한 바와 같이, 일부 실제 시나리오의 경우, 일례로 화상 회의 시스템에서, 인코딩된 오디오 데이터 및 인코딩된 비디오 데이터는 멀티플렉싱되지 않는다. 이 상황에서, 인코딩된 데이터는 디멀티플렉싱 유닛(3204)을 통하지 않고 비디오 디코더(3206) 및 오디오 디코더(3208)로 전송된다.

[0687] 디멀티플렉싱 처리를 통해 비디오 ES(Elementary Stream), 오디오 ES 및 선택사항으로 자막이 생성된다. 전술한 실시예에서 설명한 비디오 디코더(30)를 포함하는 비디오 디코더(3206)는 전술한 실시예에서 나타난 디코딩 방법에 의해 비디오 ES를 디코딩하여 비디오 프레임을 생성하고, 이 데이터를 동기 유닛(3212)에 공급한다. 오디오 디코더(3208)는 오디오 ES를 디코딩하여 오디오 프레임을 생성하고, 이 데이터를 동기 유닛(3212)에 공급한다. 대안적으로, 비디오 프레임은 동기 유닛(3212)에 공급하기 전에 버퍼(도 Y에 도시되지 않음)에 저장할 수 있다. 유사하게, 오디오 프레임은 동기 유닛(3212)에 공급하기 전에 버퍼(도 Y에 도시되지 않음)에 저장할 수 있다.

[0688] 동기 유닛(3212)은 비디오 프레임 및 오디오 프레임을 동기화하고 비디오/오디오를 비디오/오디오 디스플레이(3214)에 공급한다. 예를 들어, 동기 유닛(3212)은 비디오 및 오디오 정보의 표현을 동기화한다. 정보는 코딩된 오디오 및 시각 데이터의 표현에 관한 타임 스탬프 및 데이터 스트림 자체의 전달에 관한 타임 스탬프를 사용하여 신택스로 코딩할 수 있다.

[0689] 스트림에 자막이 포함된 경우, 자막 디코더(3210)는 자막을 디코딩하고, 비디오 프레임 및 오디오 프레임과 동기화하며, 비디오/오디오/자막을 비디오/오디오/자막 디스플레이(3216)에 제공한다.

[0690] 본 발명은 전술한 시스템에 제한되지 않고, 전술한 실시예에서의 화상 인코딩 디바이스 또는 화상 디코딩 디바이스는 다른 시스템, 예를 들어, 자동차 시스템에 통합될 수 있다.

[0691] 수학 연산자

[0692] 본 출원에서 사용되는 수학 연산자는 C 프로그래밍 언어에서 사용되는 것과 유사하다. 그러나, 정수 나눗셈과 산술 시프트 연산의 결과가 보다 정확하게 정의되고, 지수, 실수값 나눗셈과 같은 추가적인 연산이 규정된다. 번호 매기기 및 계산 규칙은 일반적으로 0부터 시작하며, 예컨대, "첫 번째"는 0-번째와 동등하고 "두 번째"는 1-번째와 동등한 등이다.

[0693] 산술 연산자

[0694] 다음의 논리 연산자는 다음과 같이 규정된다:

표 15

[0695] +	덧셈
-	뺄셈(2-인수 연산자) 또는 부정(단항 프리픽스 연산자)
*	곱셈(행렬 곱셈을 포함)
x^y	지수화. x의 y 거듭제곱을 지정한다. 다른 컨텍스트에서, 이러한 표기법은 위 첨자로 사용되며 지수로 해석되는 것을 의도하지 않는다.
/	결과를 0쪽으로 버림(truncation)하는 정수 나눗셈. 예를 들어, 7 / 4 및 -7 / -4는 1로 버림되고 -7 / 4 및 7 / -4는 -1로 버림된다.
$\lfloor x \rfloor$	버림 또는 반올림을 의도하지 않는 수학적의 나눗셈을 나타내는데 사용된다.
$\lceil x \rceil$	버림 또는 반올림을 의도하지 않는 수학적의 나눗셈을 나타내는데 사용된다.
$\sum_{i=x}^y f(i)$	x로부터 y(포함)까지의 모든 정수 값을 취하는 i에 대해 f(i)의 합.
$x \% y$	모듈러스(Modulus). x를 y로 나눈 나머지로써, $x \geq 0$ 이고 $y > 0$ 인 정수 x 및 y에 대해서만 규정된다.

[0696] 논리 연산자

[0697] 다음의 논리 연산자는 다음과 같이 규정된다:

[0698] $x \ \&\& \ y$ x 및 y의 불리언(Boolean) 논리 연산 "and"

- [0699] $x \mid \mid y$ x 및 y 의 불리언(Boolean) 논리 연산 "or"
- [0700] ! 불리언(Boolean) 논리 연산 "not"
- [0701] $x ? y : z$ x 가 참(TRUE)이거나 0이 아니면, y 의 값을 구하고; 그렇지 않으면, z 의 값을 구한다.
- [0702] 관계 연산자
- [0703] 다음의 논리 연산자는 다음과 같이 규정된다:
- [0704] > 보다 크다
- [0705] >= 보다 크거나 같다
- [0706] < 보다 작다
- [0707] <= 보다 작거나 같다
- [0708] == 같다
- [0709] != 같지 않다
- [0710] 관계 연산자가 값 "na"(해당 사항 없음)가 할당된 선택스 엘리먼트 또는 변수에 적용되는 경우, 값 "na"는 선택스 엘리먼트 또는 변수에 대한 고유한 값으로 처리된다. 값 "na"는 여하한 다른 값과 같지 않은 것으로 간주된다.
- [0711] 비트 단위(bit-wise) 연산자
- [0712] 다음의 비트 단위 연산자는 다음과 같이 규정된다:
- [0713] & 비트 단위(bit-wise) "and". 정수 인수(arguments)에 대해 연산할 때, 정수 값의 2의 보수 표현(two's complement representation)에 대해 연산한다. 다른 인수보다 더 적은 수의 비트를 포함하는 2진 인수(argument)에 대해 연산할 때, 더 짧은 인수는 상위 비트에 0을 추가함으로써 확장된다.
- [0714] | 비트 단위(bit-wise) "or". 정수 인수(arguments)에 대해 연산할 때, 정수 값의 2의 보수 표현(two's complement representation)에 대해 연산한다. 다른 인수보다 더 적은 수의 비트를 포함하는 2진 인수(argument)에 대해 연산할 때, 더 짧은 인수는 상위 비트에 0을 추가함으로써 확장된다.
- [0715] ^ 비트 단위(bit-wise) "exclusive or". 정수 인수(arguments)에 대해 연산할 때, 정수 값의 2의 보수 표현(two's complement representation)에 대해 연산한다. 다른 인수보다 더 적은 수의 비트를 포함하는 2진 인수(argument)에 대해 연산할 때, 더 짧은 인수는 상위 비트에 0을 추가함으로써 확장된다.
- [0716] $x \gg y$ x 의 2의 보수(two's complement) 정수 표현을 y 2진 자릿수만큼 산술 오른쪽 시프트. 이 함수는 y 의 음이 아닌 정수 값에 대하여만 규정된다. 오른쪽 시프트의 결과로서 최상위 비트(most significant bits, MSB)로 시프트된 비트는 시프트 연산 이전의 x 의 MSB와 같은 값을 가진다 .
- [0717] $x \ll y$ x 의 2의 보수(two's complement) 정수 표현을 y 2진 자릿수만큼 산술 왼쪽 시프트. 이 함수는 y 의 음이 아닌 정수 값에 대하여만 규정된다. 왼쪽 시프트의 결과로서 최하위 비트(least significant bits, LSB)로 시프트된 비트는 0과 같은 값을 가진다.
- [0718] 할당 연산자
- [0719] 다음의 논리 연산자는 다음과 같이 규정된다:
- [0720] = 할당 연산자
- [0721] ++ 증가, 즉, $x++$ 는 $x = x + 1$ 와 동등하고; 어레이 인덱스에서 사용되는 경우, 증가 연산 이전의 변수 값을 구한다.
- [0722] -- 감소, 즉, $x--$ 는 $x = x - 1$ 와 동등하고; 어레이 인덱스에서 사용되는 경우, 감소 연산 이전의 변수 값을 구한다.
- [0723] += 지정된 양만큼 증가, 즉, $x += 3$ 은 $x = x + 3$ 와 동등하고,
- [0724] $x += (-3)$ 는 $x = x + (-3)$ 와 동등하다.

[0725] -= 지정된 양만큼 감소, 즉, x -= 3은 x = x - 3와 동등하고,

[0726] x -= (-3)는 x = x - (-3)와 동등하다.

[0727] 범위 표기법

[0728] 다음 표기법은 값의 범위를 지정하는데 사용된다.

[0729] x = y..z x는 y로부터 z까지(포함)의 정수 값을 취하며, x, y, 및 z는 정수이고 z는 y보다 크다.

[0730] 수학 함수

[0731] 다음 수학 함수가 정의된다.

[0732]
$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

[0733] Asin(x) 삼각 역사인 함수로서, -1.0 내지 1.0의 범위(포함)에 있는 인수 x에 대해 연산하고, 라디안 단위로 $-\pi \div 2$ 내지 $\pi \div 2$ 의 범위(포함)에서 출력 값을 가짐

[0734] Atan(x) 삼각 역 탄젠트 함수로서, 인수 x에 대해 연산하고, 라디안 단위로 $-\pi \div 2$ 내지 $\pi \div 2$ 의 범위(포함)에서 출력 값을 가짐

[0735]
$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x = 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{otherwise} \end{cases}$$

[0736] Ceil(x) x보다 크거나 같은 가장 작은 정수.

[0737] Clip1_y(x) = Clip3(0, (1 << bitDepth_y) - 1, x)

[0738] Clip1_c(x) = Clip3(0, (1 << bitDepth_c) - 1, x)

[0739]
$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases}$$

[0740] Cos(x) 라디안 단위로 인수 x에 대해 연산하는 삼각 코사인 함수.

[0741] Floor(x) x보다 작거나 같은 가장 큰 정수.

[0742]
$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d / 2 \\ c - d & ; \quad a - b > d / 2 \\ c & ; \quad \text{otherwise} \end{cases}$$

[0743] Ln(x) x의 자연 로그(e를 베이스로 하는 로그, 여기서 e는 자연 로그 기본 상수 2.718 281 828...이다).

[0744] Log2(x) 2를 베이스로 하는 x의 로그.

[0745] Log10(x) 10를 베이스로 하는 x의 로그.

[0746]
$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

[0747]
$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

[0748] $\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$

[0749]
$$\text{Sign}(x) = \begin{cases} 1 & ; x > 0 \\ 0 & ; x = 0 \\ -1 & ; x < 0 \end{cases}$$

[0750] $\text{Sin}(x)$ 라디안 단위로 인수 x 에 대해 연산하는 삼각 사인 함수

[0751] $\text{Sqrt}(x) = \sqrt{x}$

[0752] $\text{Swap}(x, y) = (y, x)$

[0753] $\text{Tan}(x)$ 라디안 단위로 인수 x 에 대해 연산하는 삼각 탄젠트 함수

[0754] 연산 우선 순위의 순서

[0755] 표현에서 우선 순위의 순서가 괄호를 사용하여 명시적으로 표시되지 않은 경우, 다음 규칙이 적용된다:

[0756] - 더 높은 우선 순위의 연산은 더 낮은 우선 순위의 연산에 앞서 계산된다.

[0757] - 동일한 우선 순위의 연산은 왼쪽으로부터 오른쪽으로 순차적으로 계산된다.

[0758] 아래의 표 14는 연산의 우선 순위를 최상위로부터 최하위까지 지정하며; 표 14에서 더 높은 위치는 더 높은 우선 순위를 나타낸다.

[0759] C 프로그래밍 언어에서도 사용되는 연산자에 대하여는, 이 명세서에서 사용되는 우선 순위의 순서는 C 프로그래밍 언어에서 사용되는 것과 동일하다.

표 14

최상위(테이블의 상단)로부터 최하위(테이블의 하단)까지 연산 우선 순위

operations (with operands $x, y,$ and z)
" $x++$ ", " $x--$ "
" $!x$ ", " $-x$ " (as a unary prefix operator)
x^y
" $x * y$ ", " x / y ", " $x \div y$ ", " $\frac{x}{y}$ ", " $x \% y$ "
" $x + y$ ", " $x - y$ " (as a two-argument operator), " $\sum_{i=x}^y f(i)$ "
" $x \ll y$ ", " $x \gg y$ "
" $x < y$ ", " $x \leq y$ ", " $x > y$ ", " $x \geq y$ "
" $x == y$ ", " $x != y$ "
" $x \& y$ "
" $x y$ "
" $x \&\& y$ "
" $x y$ "
" $x ? y : z$ "
" $x..y$ "
" $x = y$ ", " $x += y$ ", " $x -= y$ "

[0760]

[0761] 논리 연산에 대한 텍스트 설명

[0762] 텍스트에서, 수학적으로 다음의 형태로 기술되는 논리 연산의 선언은:

[0763] $\text{if}(\text{조건 } 0)$

[0764] 선언 0

[0765] $\text{else if}(\text{조건 } 1)$

[0766] 선언 1

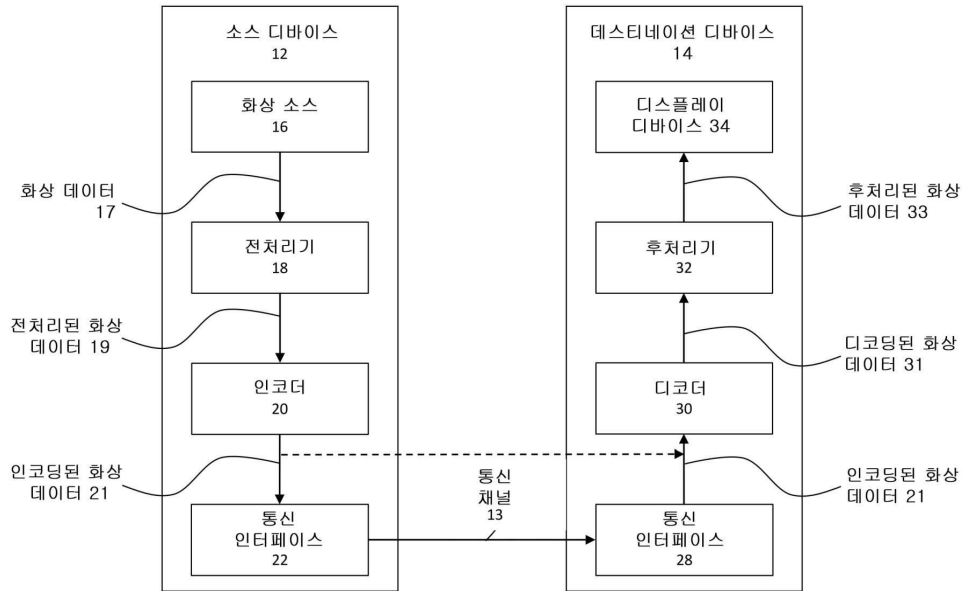
- [0767] ...
- [0768] else /* 나머지 조건에 대한 정보 주석 */
- [0769] 선언 n
- [0770] 다음의 방식으로 기술될 수 있다:
- [0771] ... 다음과 같이 / ... 다음이 적용된다:
- [0772] - If 조건 0, 선언 0
- [0773] - 그렇지 않으면, 조건 1이면, 선언 1
- [0774] - ...
- [0775] - 그렇지 않으면(나머지 조건에 대한 정보 주석), 선언 n
- [0776] 텍스트에서 각각의 "If ... 그렇지 않으면, if ... 그렇지 않으면, ..." 선언은 "... 다음과 같이" 또는 "... 다음이 적용된다"로 시작하고 "If ... "가 바로 이어진다. "If ... 그렇지 않으면, if ... 그렇지 않으면, ..."의 마지막 조건은 항상 "그렇지 않으면, ..."이다. 사이에 끼워진 "If ... 그렇지 않으면, if ... 그렇지 않으면, ..." 선언은 "그렇지 않으면, ..."으로 끝나는 "... 다음과 같이" 또는 "... 다음이 적용된다"를 매칭 함으로써 식별될 수 있다.
- [0777] 텍스트에서, 수학적으로 다음의 형태로 기술되는 논리 연산의 선언은:
- [0778] if(조건 0a && 조건 0b)
- [0779] 선언 0
- [0780] else if(조건 1a || 조건 1b)
- [0781] 선언 1
- [0782] ...
- [0783] else
- [0784] 선언 n
- [0785] 다음의 방식으로 기술될 수 있다:
- [0786] ... 다음과 같이 / ... 다음이 적용된다:
- [0787] - 다음 조건의 모두가 참(true)이면, 선언 0:
- [0788] - 조건 0a
- [0789] - 조건 0b
- [0790] - 그렇지 않으면, 다음 조건 중 하나 이상이 참(true)이면, 선언 1:
- [0791] - 조건 1a
- [0792] - 조건 1b
- [0793] - ...
- [0794] - 그렇지 않으면, 선언 n
- [0795] 텍스트에서, 수학적으로 다음의 형태로 기술되는 논리 연산의 선언은:
- [0796] if(조건 0)
- [0797] 선언 0
- [0798] if(조건 1)
- [0799] 선언 1

- [0800] 다음의 방식으로 기술될 수 있다:
- [0801] 조건 0인 경우, 선언 0
- [0802] 조건 1인 경우, 선언 1
- [0803] 본 개시의 실시예가 주로 비디오 코딩에 기반하여 설명되었지만, 코딩 시스템(10), 인코더(20) 및 디코더(30) (및 이에 상응하는 시스템(10))의 실시예 및 여기에 설명된 다른 실시예에는 또한 여전히 화상 처리 또는 코딩, 즉 비디오 코딩에서와 같이 임의의 선행하는 또는 연속하는 화상에 독립적인 개별 화상의 처리 또는 코딩을 위해 구성될 수 있다. 일반적으로 화상 처리 코딩이 단일 화상(17)으로 제한되는 경우에 인터 예측 유닛(244)(인코더) 및 (344)(디코더)만이 사용 가능하지 않을 수 있다. 비디오 인코더(20) 및 비디오 디코더(30)의 다른 모든 기능(도구 또는 기술이라고도 함)은 정지 화상 처리, 예를 들어 잔차 계산(204/304), 변환(206), 양자화(208), 역 양자화(210/310), (역) 변환(212/312), 파티셔닝(262/362), 인트라 예측(254/354), 및/또는 루프 필터링(220, 320), 엔트로피 코딩(270) 및 엔트로피 디코딩(304)에 동일하게 사용될 수 있다.
- [0804] 예를 들어, 인코더(20) 및 디코더(30)의 실시예 및 여기서 설명된 기능, 예를 들어 인코더(20) 및 디코더(30)를 참조하여, 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수 있다. 소프트웨어로 구현되면, 기능은 컴퓨터가 판독 가능한 매체에 저장되거나 통신 매체를 통해 하나 이상의 명령 또는 코드로 전송되고, 하드웨어 기반 처리 디바이스에 의해 실행될 수 있다. 컴퓨터-판독 가능한 매체는, 데이터 저장 매체와 같은 유체물의 매체에 해당하는 컴퓨터-판독 가능한 저장 매체, 또는 (예컨대 통신 프로토콜에 따라) 컴퓨터 프로그램을 하나의 장소에서 다른 장소로 전송하는 것을 용이하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수 있다. 이러한 방식으로, 컴퓨터-판독 가능한 매체는 일반적으로 (1) 유체물의 컴퓨터-판독 가능한 저장 매체 또는 (2) 신호 또는 캐리어와 같은 통신 매체에 대응할 수 있다. 데이터 저장 매체는 본 출원에 기술된 기술의 구현을 위한 명령, 코드 및/또는 데이터 구조를 검색(retrieve)하기 위해 하나 이상의 컴퓨터 또는 하나 이상의 프로세서에 의해 액세스될 수 있는 임의의 이용 가능한 매체일 수 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독 가능한 매체를 포함할 수 있다.
- [0805] 제한이 아닌 예로서, 이러한 컴퓨터-판독 가능한 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스, 플래시 메모리, 또는 컴퓨터에 의해 액세스할 수 있는 명령 또는 데이터 구조의 형태로 원하는 프로그램 코드를 저장하는데 사용할 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 모든 연결을 컴퓨터-판독 가능한 매체라고 한다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선(twisted pair), 디지털 가입자 회선(DSL), 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술을 사용하여 웹 사이트, 서버, 또는 다른 원격 소스로부터 명령이 전송되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술이 매체의 정의에 포함된다. 그러나, 컴퓨터-판독 가능한 저장 매체 및 데이터 저장 매체는 연결, 캐리어 파(wave), 신호, 또는 다른 일시적인 매체를 포함하지 않고, 대신에 비일시적 유체물의 저장 매체를 카리킴을 이해해야 한다. 여기에 사용된 디스크(disk) 및 디스크(disc)는, 콤팩트 디스크(CD), 레이저 디스크, 광학 디스크, DVD(digital versatile disc), 플로피 디스크 및 블루-레이 디스크를 포함하며, 여기서, 디스크(disk)는 일반적으로 데이터를 자기적으로 재생하는 반면, 디스크(disc)는 레이저를 사용하여 광학적으로 데이터를 재생한다. 위의 조합도 컴퓨터-판독 가능한 매체의 범위 내에 포함되어야 한다.
- [0806] 명령은 하나 이상의 디지털 신호 처리기(digital signal processor; DSP), 범용 마이크로 프로세서, 주문형 집적 회로(application specific integrated circuits; ASIC), 필드 프로그래밍 가능한 로직 어레이(field programmable logic array; FPGA) 또는 다른 동등한 집적 또는 이산 논리 회로와 같은 하나 이상의 프로세서에 의해 실행될 수 있다. 따라서, 여기에서 사용되는 용어 "프로세서"는, 전술한 구조 또는 여기에서 설명된 기술의 구현에 적합한 임의의 다른 구조일 수 있다. 또한, 일부 측면에서, 여기에서 기술되는 기능은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈 내에 제공되거나 결합된 코덱에 통합될 수 있다. 또한, 기법은 하나 이상의 회로 또는 논리 요소에서 완전히 구현될 수 있다.
- [0807] 본 개시의 기법은 무선 핸드셋, 집적 회로(IC) 또는 IC 세트(예를 들어, 칩셋)를 포함하는 다양한 장치 또는 디바이스에서 구현될 수 있다. 개시된 기술을 수행하도록 구성되는 장치의 기능적 측면을 강조하기 위해 다양한 구성 요소, 모듈, 또는 유닛이 본 출원에서 설명되지만, 반드시 상이한 하드웨어 유닛에 의한 구현을 필요로 하는 것은 아니다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛이 코덱 하드웨어 유닛으로 결합되거나, 적절한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 하나 이상의 프로세서를 포함하는 상호 동작하는 하드웨어

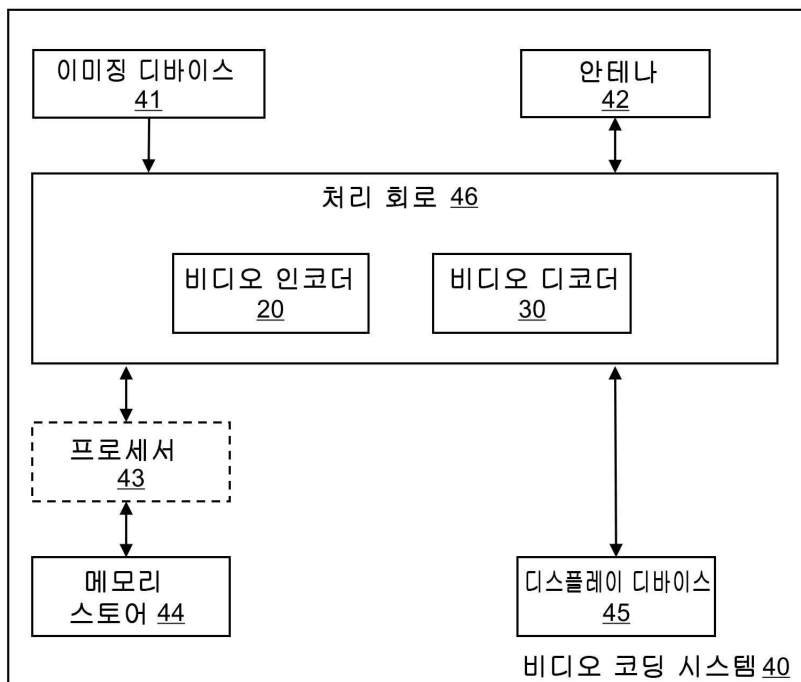
유닛의 집합에 의해 제공될 수 있다.

도면

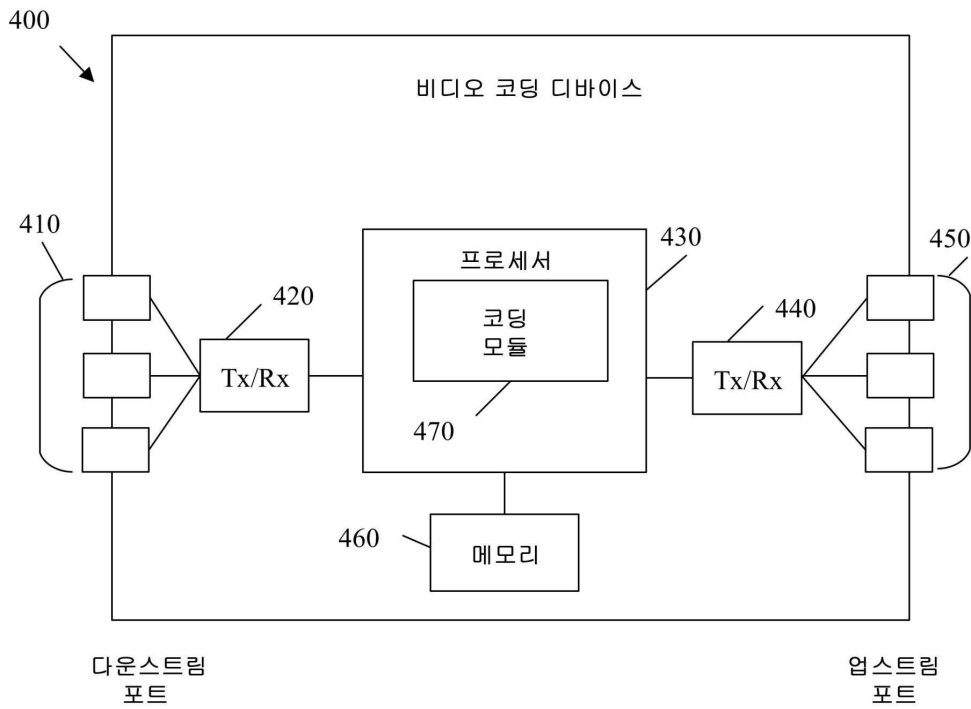
도면1a



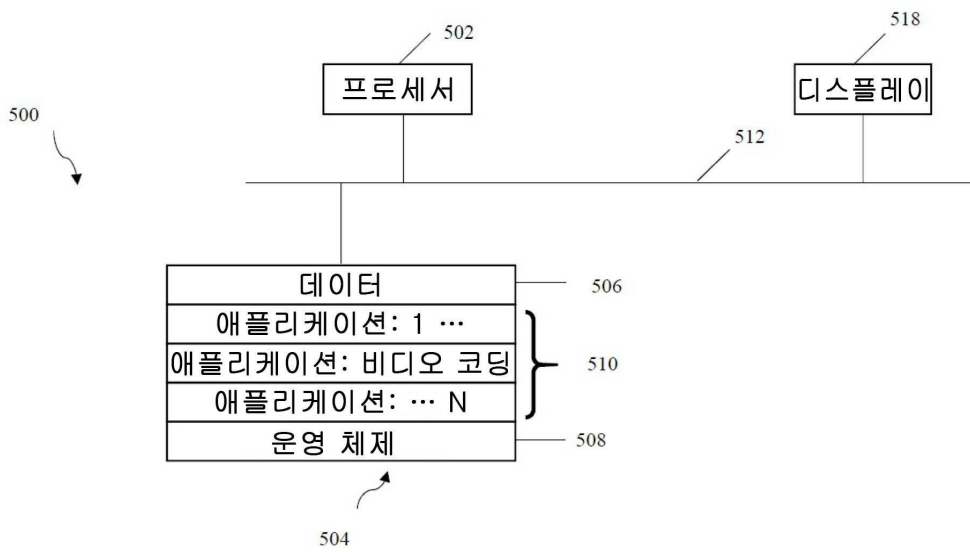
도면1b



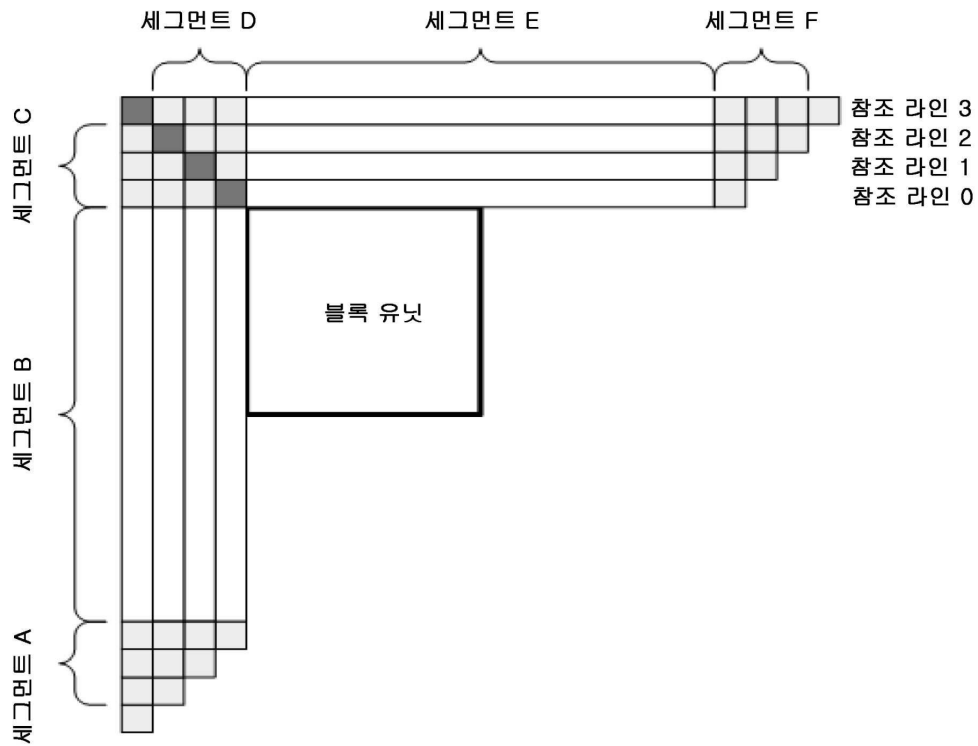
도면4



도면5

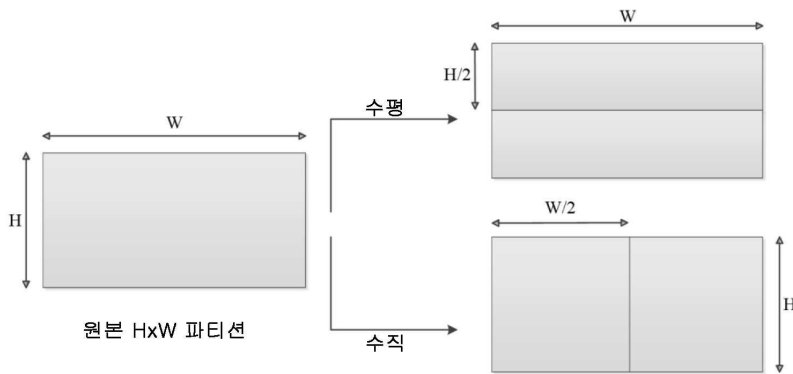


도면6



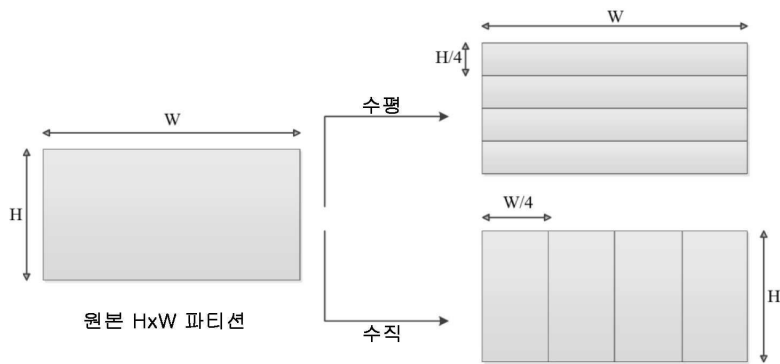
다중 참조 라인 코딩 도구, 이웃 참조 라인의 표시.
 참조 라인 인덱스가 0과 같은 경우, 현재 블록에
 가장 가까운 이웃 참조 라인이 사용된다.

도면7



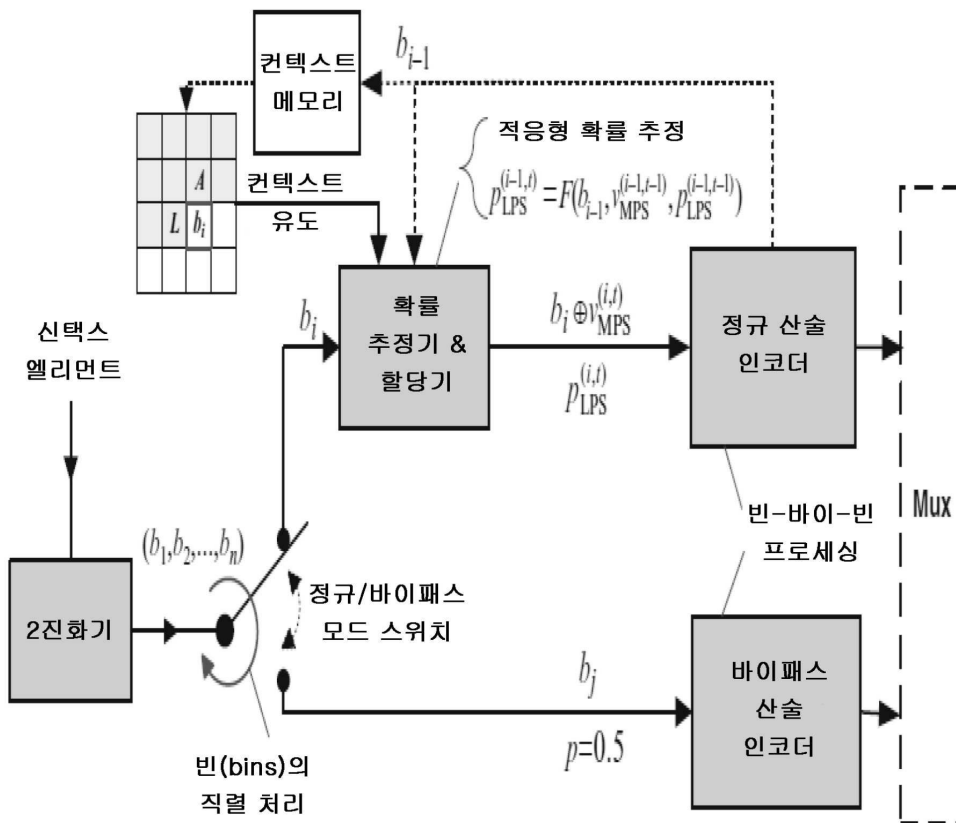
4 × 8 및 8 × 4 블록의 분할의 예시

도면8

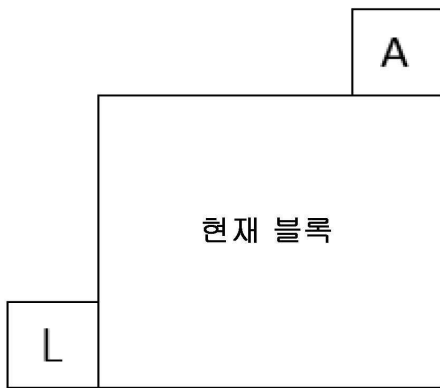


4x8, 8x4 및 4x4를 제외한 모든 블록의 분할의 예시

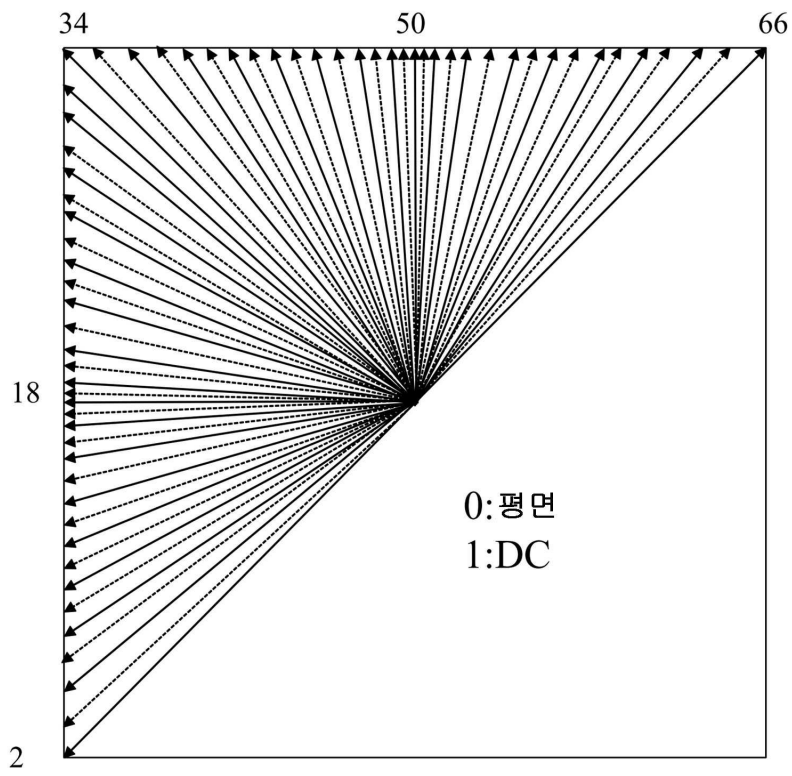
도면9



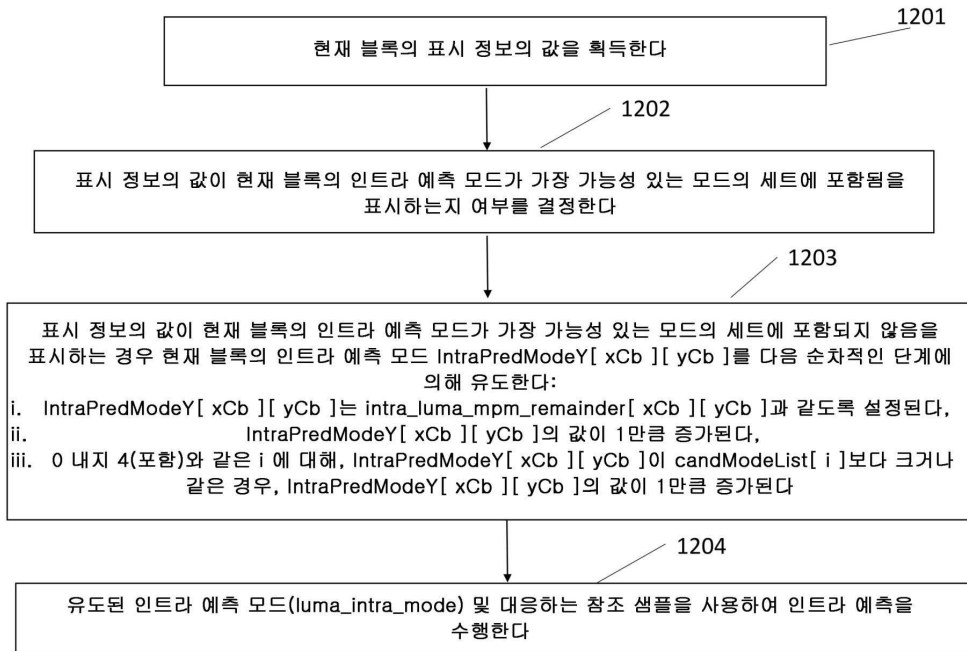
도면10



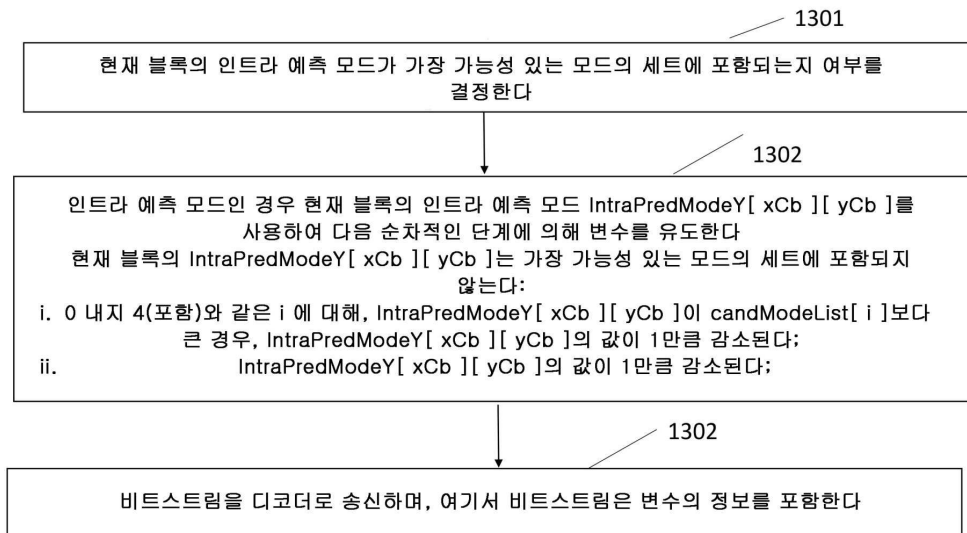
도면11



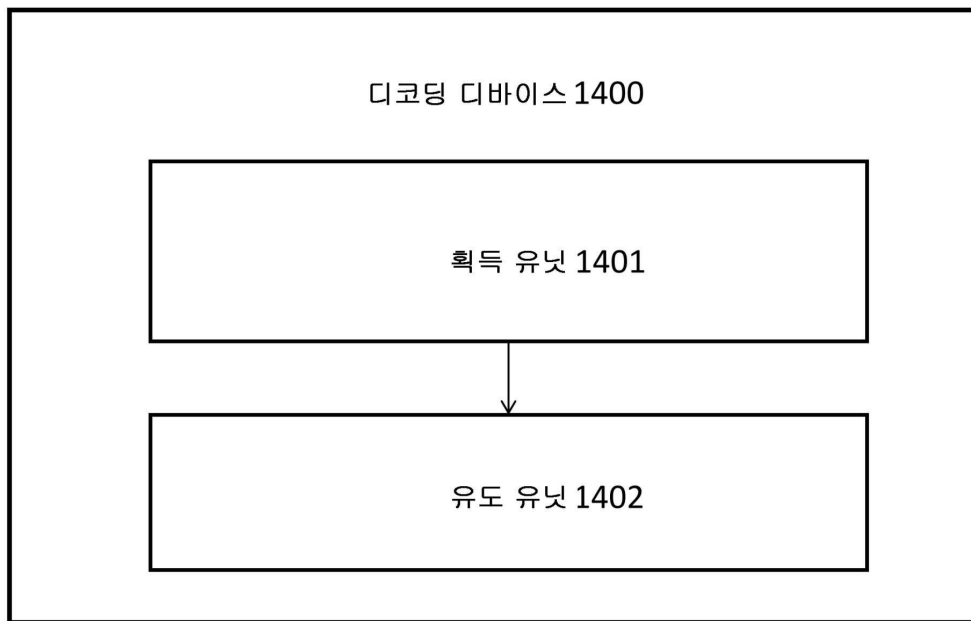
도면12



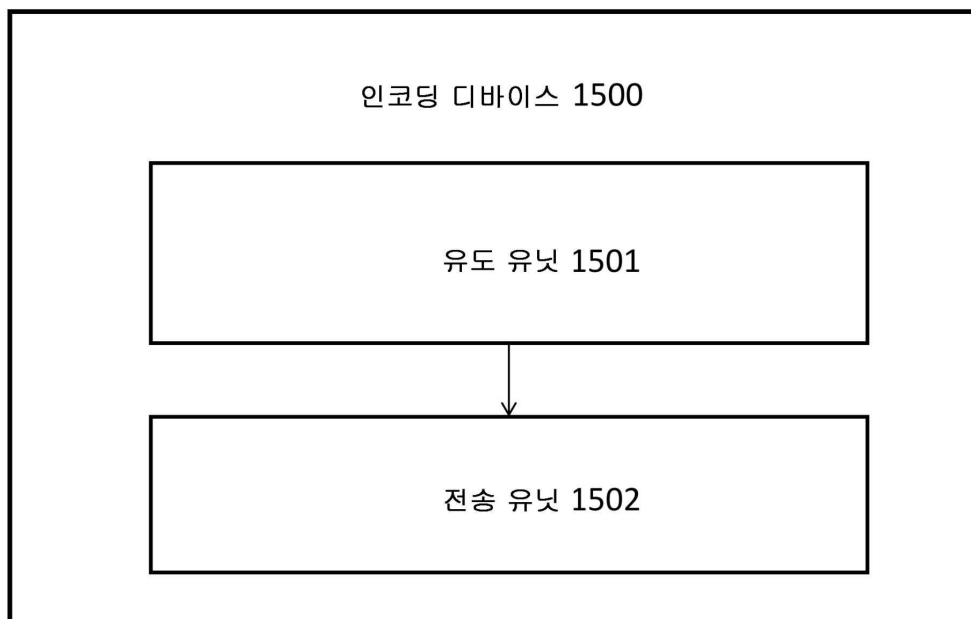
도면13



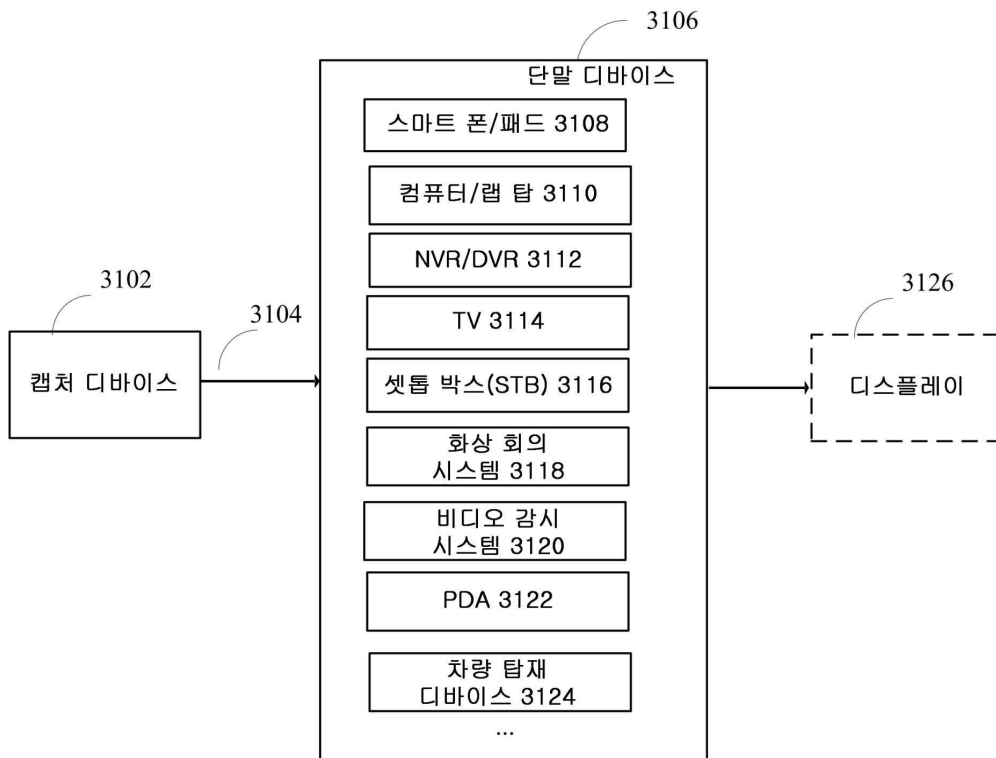
도면14



도면15



도면16



도면17

