(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0006057
(43) 공개일자 2022년01월14일

(51) 국제특허분류(Int. Cl.)
*H04N 19/593* (2014.01) *H04N 19/105* (2014.01)
*H04N 19/11* (2014.01) *H04N 19/157* (2014.01)
*H04N 19/159* (2014.01) *H04N 19/176* (2014.01)
*H04N 19/436* (2014.01) *H04N 19/91* (2014.01)
(52) CPC특허분류
*H04N 19/593* (2015.01)
*H04N 19/105* (2015.01)
(21) 출원번호 10-2021-7035523
(22) 출원일자(국제) 2020년05월06일
심사청구일자 없음
(85) 번역문제출일자 2021년10월29일
(86) 국제출원번호 PCT/US2020/031651
(87) 국제공개번호 WO 2020/227393
국제공개일자 2020년11월12일
(30) 우선권주장
62/845,790 2019년05월09일 미국(US)
*(뒷면에 계속)*

(71) 출원인
**퀄컴 인코포레이티드**
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
**라마수브라모니안 아다르쉬 크리쉬난**
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
**판 더 아우베라 게르트**
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
*(뒷면에 계속)*
(74) 대리인
**특허법인코리아나**

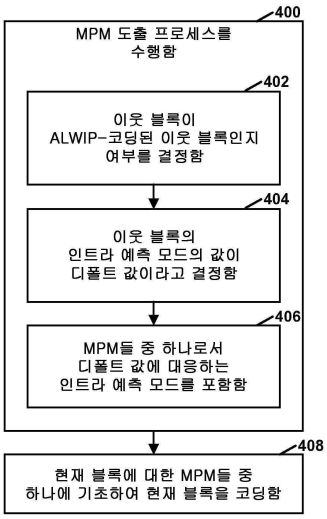
전체 청구항 수 : 총 28 항

(54) 발명의 명칭 **비디오 코딩에서 아핀 선형 가중 인트라 예측**

## (57) 요 약

비디오 코더는 아핀 선형 가중 인트라 예측 (ALWIP) 을 사용하여 코딩되지 않은 현재 블록에 대한 하나 이상의 최확 모드 (Most-Probable Mode; MPM) 들을 도출하는 MPM 도출 프로세스를 수행한다. MPM 도출 프로세스를 수행하는 것의 일부로서, 비디오 코더는 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정한다. 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 비디오 코더는 이웃 블록의 인트라 예측 모드의 값이 평면 모드를 표시하는 값이라고 결정한다. 비디오 코더는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩한다.

*대 표 도* - 도19



(52) CPC특허분류
*H04N 19/11* (2015.01)
*H04N 19/157* (2015.01)
*H04N 19/159* (2015.01)
*H04N 19/176* (2015.01)
*H04N 19/436* (2015.01)
*H04N 19/91* (2015.01)

(72) 발명자
**팜 반 루옹**
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
**카르체비츠 마르타**
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장
62/864,320 2019년06월20일 미국(US)
16/867,208 2020년05월05일 미국(US)

# 명 세 서

## 청구범위

### 청구항 1

비디오 데이터를 코딩하는 방법으로서,

상기 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (Most-Probable Mode; MPM) 들을 도출하는 MPM 도출 프로세스를 수행하는 단계로서, 상기 현재 블록은 아핀 선형 가중 인트라 예측 (affine linear weighted intra prediction; ALWIP) 을 사용하여 코딩되지 않으며, 상기 MPM 도출 프로세스를 수행하는 단계는:

상기 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 단계; 및

상기 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여:

상기 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고; 그리고

상기 하나 이상의 MPM들 중 하나로서 상기 디폴트 값에 대응하는 인트라 예측 모드를 포함하는 단계를 포함하는,

상기 MPM 도출 프로세스를 수행하는 단계; 및

상기 현재 블록에 대한 상기 MPM들 중 하나에 기초하여 상기 현재 블록을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

### 청구항 2

제 1 항에 있어서,

상기 디폴트 값에 대응하는 상기 인트라 예측 모드는 평면 모드인, 비디오 데이터를 코딩하는 방법.

### 청구항 3

제 1 항에 있어서,

상기 현재 블록은 제 1 블록이고, 상기 비디오 데이터의 제 2 블록은 다이렉트 모드 (Direct Mode; DM) 모드로 코딩된 크로마 블록이며, 상기 방법은,

병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정하는 단계로서, 상기 병치된 루마 블록은 상기 제 2 블록과 병치되는, 상기 ALWIP 로 코딩되는지 여부를 결정하는 단계; 및

상기 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여:

상기 병치된 루마 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고; 그리고

상기 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 상기 제 2 블록을 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

### 청구항 4

제 3 항에 있어서,

상기 디폴트 값에 대응하는 상기 인트라 예측 모드는 평면 모드인, 비디오 데이터를 코딩하는 방법.

### 청구항 5

제 1 항에 있어서,

ALWIP 신택스 엘리먼트에 대한 컨텍스트를 결정하는 단계로서, 상기 ALWIP 신택스 엘리먼트는 상기 현재 블록이 ALWIP 로 코딩되는지 여부를 표시하고, 상측 이웃 블록은 상기 상측 이웃 블록이 상기 현재 블록과 상이한 코딩 트리 유닛 (CTU) 에 있는 것에 기초하여 상기 컨텍스트를 결정하는데 이용불가능한 것으로 결정되며, 상기 상측

이웃 블록은 상기 현재 블록의 상측 이웃인, 상기 컨텍스트를 결정하는 단계; 및

상기 컨텍스트에 기초하여 상기 ALWIP 신택스 엘리먼트를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

**청구항 6**

제 1 항에 있어서,

비디오 코딩 표준은 임계치 폭보다 작은 폭 또는 임계치 높이보다 작은 높이를 갖는 블록들에 대해 ALWIP 를 디스에이블하는 제약을 부과하는, 비디오 데이터를 코딩하는 방법.

**청구항 7**

제 6 항에 있어서,

상기 임계치 폭은 8 이고 상기 임계치 높이는 8 인, 비디오 데이터를 코딩하는 방법.

**청구항 8**

제 1 항에 있어서,

상기 현재 블록을 코딩하는 단계는, 상기 현재 블록을 디코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

**청구항 9**

제 8 항에 있어서,

상기 현재 블록을 포함하는 픽처를 디스플레이하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

**청구항 10**

제 1 항에 있어서,

상기 현재 블록을 코딩하는 단계는 상기 현재 블록을 인코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

**청구항 11**

제 10 항에 있어서,

상기 현재 블록을 포함하는 픽처를 캡처하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

**청구항 12**

비디오 데이터를 코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하는 메모리; 및

회로부에서 구현된 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (Most-Probable Mode; MPM) 들을 도출하는 MPM 도출 프로세스를 수행하는 것으로서, 상기 현재 블록은 아핀 선형 가중 인트라 예측 (affine linear weighted intra prediction; ALWIP) 을 사용하여 코딩되지 않으며, 상기 MPM 도출 프로세스를 수행하는 것은:

상기 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 것; 및

상기 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여:

상기 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고;

상기 하나 이상의 MPM들 중 하나로서 상기 디폴트 값들에 대응하는 인트라 예측 모드를 포함하는 것을 포함하는,

상기 MPM 도출 프로세스를 수행하며; 그리고

상기 현재 블록에 대한 상기 MPM들 중 하나에 기초하여 상기 현재 블록을 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 13**

제 12 항에 있어서,

상기 디폴트 값에 대응하는 상기 인트라 예측 모드는 평면 모드인, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 14**

제 12 항에 있어서,

상기 현재 블록은 제 1 블록이고, 상기 비디오 데이터의 제 2 블록은 다이렉트 모드 (Direct Mode; DM) 모드로 코딩된 크로마 블록이며, 상기 하나 이상의 프로세서들을 또한,

병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정하는 것으로서, 상기 병치된 루마 블록은 상기 제 2 블록과 병치되는, 상기 ALWIP 로 코딩되는지 여부를 결정하고; 그리고

상기 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여:

상기 병치된 루마 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고;

상기 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 상기 제 2 블록을 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 15**

제 14 항에 있어서,

상기 디폴트 값에 대응하는 상기 인트라 예측 모드는 평면 모드인, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 16**

제 12 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 현재 블록이 ALWIP 로 코딩되는지 여부를 표시하는 ALWIP 신택스 엘리먼트에 대한 컨텍스트를 결정하는 것으로서, 상측 이웃 블록은 상기 상측 이웃 블록이 상기 현재 블록과 상이한 코딩 트리 유닛 (CTU) 에 있는 것에 기초하여 상기 컨텍스트를 결정하는데 이용불가능한 것으로 결정되며, 상기 상측 이웃 블록은 상기 현재 블록의 상측 이웃인, 상기 컨텍스트를 결정하고; 그리고

상기 컨텍스트에 기초하여 상기 ALWIP 신택스 엘리먼트를 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 17**

제 12 항에 있어서,

비디오 코딩 표준은 임계치 폭보다 작은 폭 또는 임계치 높이보다 작은 높이를 갖는 블록들에 대해 ALWIP 를 디스에이블하는 제약을 부과하는, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 18**

제 17 항에 있어서,

상기 임계치 폭은 8 이고 상기 임계치 높이는 8 인, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 19**

제 12 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 현재 블록을 코딩하는 것의 일부로서, 상기 하나 이상의 프로세서들이 상기 현재 블록을 디코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 20**

제 19 항에 있어서,

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

**청구항 21**

제 12 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 현재 블록을 코딩하는 것의 일부로서, 상기 하나 이상의 프로세서들이 상기 현재 블록을 인코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 22**

제 12 항에 있어서,

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

**청구항 23**

비디오 데이터를 코딩하기 위한 디바이스로서,

상기 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (Most-Probable Mode; MPM) 들을 도출하는 MPM 도출 프로세스를 수행하는 수단으로서, 상기 현재 블록은 아핀 선형 가중 인트라 예측 (affine linear weighted intra prediction; ALWIP) 을 사용하여 코딩되지 않으며, 상기 MPM 도출 프로세스를 수행하는 수단은:

상기 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 수단; 및

상기 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 상기 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고, 상기 하나 이상의 MPM들 중 하나로서 상기 디폴트 값에 대응하는 인트라 예측 모드를 포함하는 수단을 포함하는,

상기 MPM 도출 프로세스를 수행하는 수단; 및

상기 현재 블록에 대한 상기 MPM들 중 하나에 기초하여 상기 현재 블록을 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 24**

제 23 항에 있어서,

상기 디폴트 값에 대응하는 상기 인트라 예측 모드는 평면 모드인, 비디오 데이터를 코딩하기 위한 디바이스.

**청구항 25**

제 23 항에 있어서,

상기 현재 블록은 제 1 블록이고, 상기 비디오 데이터의 제 2 블록은 다이렉트 모드 (Direct Mode; DM) 모드로 코딩된 크로마 블록이며, 상기 디바이스는,

병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정하는 수단으로서, 상기 병치된 루마 블록은 상기 제 2 블록과 병치되는, 상기 ALWIP 로 코딩되는지 여부를 결정하는 수단;

상기 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여, 상기 병치된 루마 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하는 수단; 및

상기 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 상기 제 2 블록을 코딩하는 수단을 더 포함하는, 비디

오 데이터를 코딩하기 위한 디바이스.

**청구항 26**

명령들이 저장된 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행될 때, 하나 이상의 프로세서들로 하여금,

비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (Most-Probable Mode; MPM) 들을 도출하는 MPM 도출 프로세스를 수행하게 하는 것으로서, 상기 현재 블록은 아핀 선형 가중 인트라 예측 (affine linear weighted intra prediction; ALWIP) 을 사용하여 코딩되지 않으며, 상기 MPM 도출 프로세스를 수행하는 것은:

상기 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 것; 및

상기 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여:

상기 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고;

상기 하나 이상의 MPM들 중 하나로서 상기 디폴트 값에 대응하는 인트라 예측 모드를 포함하는 것을 포함하는,

상기 MPM 도출 프로세스를 수행하게 하며; 그리고

상기 현재 블록에 대한 상기 MPM들 중 하나에 기초하여 상기 현재 블록을 코딩하게 하는, 컴퓨터 판독가능 저장 매체.

**청구항 27**

제 26 항에 있어서,

상기 디폴트 값에 대응하는 상기 인트라 예측 모드는 평면 모드인, 컴퓨터 판독가능 저장 매체.

**청구항 28**

제 26 항에 있어서,

상기 현재 블록은 제 1 블록이고, 상기 비디오 데이터의 제 2 블록은 다이렉트 모드 (Direct Mode; DM) 모드로 코딩된 크로마 블록이며, 상기 하나 이상의 프로세서들은 또한,

병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정하는 것으로서, 상기 병치된 루마 블록은 상기 제 2 블록과 병치되는, 상기 ALWIP 로 코딩되는지 여부를 결정하고; 그리고

상기 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여:

상기 병치된 루마 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고;

상기 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 상기 제 2 블록을 코딩하도록 구성되는, 컴퓨터 판독가능 저장 매체.

## *발명의 설명*

### *기 술 분 야*

[0001] 본 출원은 2019 년 5 월 9 일 출원된 미국 가특허출원 제 62/845,790 호 2019 년 6 월 20 일 출원된 미국 가특허출원 제 62/864,320 호의 이익을 주장하는, 2020 년 5 월 5 일 출원된 미국 특허 출원 제 16/867,208 호에 대해 우선권을 주장하며, 이들 각각의 전체 내용은 참조로 통합된다.

[0002] 본 개시는 비디오 인코딩 및 비디오 디코딩에 관한 것이다.

### *배 경 기 술*

[0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인용 디지털 보조기 (PDA) 들, 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 텔레컨퍼런싱 디바이스들, 비디오 스트리

밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC), ITU-T H.265, 고효율 비디오 코딩 (HEVC) 표준에 의해 정의된 표준들, 및 그러한 표준들의 확장들에서 설명된 기법들과 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 그러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 코딩 기법들은 비디오 시퀀스들에 내재된 리던던시를 감소 또는 제거하기 위해 공간적 (인트라-픽처) 예측 및/또는 시간적 (인터-픽처) 예측을 포함한다. 블록 기반 비디오 코딩에 대해, 비디오 슬라이스 (즉, 비디오 픽처 또는 비디오 픽처의 일부) 는 비디오 블록들로 파티셔닝될 수도 있으며, 이 비디오 블록들은 또한 코딩 트리 유닛들 (CTU들), 코딩 유닛들 (CU들) 및/또는 코딩 노드들로서 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처에 있어서 이웃하는 블록들에서의 참조 샘플들에 대한 공간적 예측을 사용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처에 있어서 이웃하는 블록들에서의 참조 샘플들에 대한 공간적 예측 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들로 지칭될 수도 있고, 참조 픽처들은 참조 프레임들로 지칭될 수도 있다.

## *발명의 내용*

### *과제의 해결 수단*

[0005] 일반적으로, 본 개시는 선형 가중 인트라 예측을 위한 모드들의 도출 및 시그널링을 포함하는, 인트라 예측을 위한 기법들을 설명한다. 예를 들어, 일 예에서, 본 개시는 아핀 선형 가중 인트라 예측 (affine linear weighted intra prediction; ALWIP) 을 사용하여 코딩되지 않은 현재 블록에 대한 하나 이상의 최확 모드 (Most-Probable Mode; MPM) 들을 도출하는 MPM 도출 프로세스를 수행하는, 비디오 인코더 또는 비디오 디코더와 같은 비디오 코더를 설명한다. 현재 블록에 대한 MPM들은 현재 블록을 코딩하기 위해 가장 많이 사용될 가능성이 있는 것으로 결정된 인트라 예측 모드들일 수도 있다. MPM 도출 프로세스를 수행하는 것의 일부로서, 비디오 코더는 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정한다. 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 비디오 코더는 이웃 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정한다. 비디오 코더는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩한다.

[0006] 일 예에서, 본 개시는 비디오 데이터를 코딩하는 방법을 기재하며, 방법은 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (MPM) 들을 도출하는 MPM 도출 프로세스를 수행하는 단계로서, 현재 블록은 아핀 선형 가중 인트라 예측 (affine linear weighted intra prediction; ALWIP) 을 사용하여 코딩되지 않으며, MPM 도출 프로세스를 수행하는 단계는: 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 단계; 및 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고; 그리고 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함하는 단계를 포함하는, 상기 MPM 도출 프로세스를 수행하는 단계; 및 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩하는 단계를 포함한다.

[0007] 다른 예에서, 본 개시는 비디오 데이터를 코딩하기 위한 디바이스를 기재하며, 디바이스는 비디오 데이터를 저장하도록 구성된 메모리; 및 회로부에서 구현된 하나 이상의 프로세서들을 포함하고, 하나 이상의 프로세서들은, 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (MPM) 들을 도출하는 MPM 도출 프로세스를 수행하는 것으로서, 현재 블록은 아핀 선형 가중 인트라 예측 (ALWIP) 을 사용하여 코딩되지 않으며, MPM 도출 프로세스를 수행하는 것은, 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 것; 및 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고; 그리고 하나 이상의 MPM들 중 하나로서 디폴트 값들에 대응하는 인트라 예측 모드를 포함하는 것을 포함하는, 상기 MPM 도출 프로세스를 수행하며; 그리고 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩하도록 구성된다.

[0008] 다른 예에서, 본 개시는 비디오 데이터를 코딩하기 위한 디바이스를 기재하며, 디바이스는 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (MPM) 들을 도출하는 MPM 도출 프로세스를 수행하는 수단으로서, 현재 블록은 아핀 선형 가중 인트라 예측 (ALWIP) 을 사용하여 코딩되지 않으며, MPM 도출 프로세스를 수행하는 수단은, 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 수단; 및 이웃 블록이 ALWIP-코딩된

이웃 블록인 것에 기초하여, 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고, 그리고 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함하는 수단을 포함하는, 상기 MPM 도출 프로세스를 수행하는 수단; 및 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩하는 수단을 포함한다.

[0009] 다른 예에서, 본 개시는 명령들이 저장된 컴퓨터 판독가능 저장 매체를 기재하며, 명령들은 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (MPM) 들을 도출하는 MPM 도출 프로세스를 수행하게 하는 것으로서, 현재 블록은 아핀 선형 가중 인트라 예측 (ALWIP) 을 사용하여 코딩되지 않으며, MPM 도출 프로세스를 수행하는 것은, 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 것; 및 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여,: 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정하고; 그리고 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함하는 것을 포함하는, 상기 MPM 도출 프로세스를 수행하게 하며; 그리고 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩하게 한다.

[0010] 하나 이상의 예들의 상세들은 첨부 도면들 및 하기의 설명에서 기술된다. 다른 피처들, 목적들 및 이점들은 설명 및 도면들, 그리고 청구항들로부터 명백해질 것이다.

## *도면의 간단한 설명*

[0011] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템을 도시하는 블록 다이어그램이다.

도 2 는 참조 샘플들을 가리키는 화살표들을 갖는, 인트라 예측의 예를 도시하는 블록 다이어그램이다.

도 3 은 '가까운' 참조 샘플들은 사용되지 않지만, -135 도 내지 45 도 범위에 있도록 인트라 예측 방향이 제한되는 것으로 인해 더 먼 참조 샘플들이 사용될 수도 있는, 8x4 직사각형 블록의 개념적 다이어그램이다.

도 4 는 65 개의 각도 모드들에 부가하여 도시된 넓은 각도들 (-1 내지 -10, 그리고 67 내지 76) 을 도시하는 개념적 다이어그램이다.

도 5a, 도 5b 및 도 5c 는 대각선 방향 범위 외부의 인트라 예측 모드들에 대한 모드 매핑 프로세스들의 예시들이다.

도 6 은 총 93 개의 각도 모드들에 대해 모드들 2 및 66 을 넘어 VVC 테스트 모델 3 (VTM3) 에서 넓은 각도들 (-1 내지 -14, 및 67 내지 80) 을 도시하는 개념적 다이어그램이다.

도 7 은 인트라 예측 각도들의 사양을 제공하는 표이다.

도 8 은 코딩 블록의 인트라 예측을 위해 사용될 수도 있는 다중 참조 라인들로부터의 참조 샘플들의 예시이다.

도 9 는 4x8 및 8x4 블록들의 분할 예를 도시하는 개념적 다이어그램이다.

도 10 은 4x8 및 8x4, 및 4x4 를 제외한 모든 블록들의 분할을 도시하는 개념적 다이어그램이다.

도 11 은 8x8 블록에 대한 아핀 선형 가중 인트라 예측 (ALWIP) 프로세스의 예를 도시하는 개념적 다이어그램이다.

도 12 는 VVC WD4 (Versatile Video Coding Working Draft 4) 에서 각각의 ALWIP 타입에 대한 오프셋 파라미터들 및 매트릭스 파라미터들의 수를 도시하는 표이다.

도 13 은 본 개시의 하나 이상의 기법들에 따른, 현재 블록에 대한 ALWIP 최확 모드 (MPM) 들을 도출하는데 사용된 예시의 이웃 블록들을 도시하는 개념적 다이어그램이다.

도 14a 및 도 14b 는 예시의 쿼드트리 이진 트리 (QTBT) 구조 및 대응하는 코딩 트리 유닛 (CTU) 을 도시하는 개념적 다이어그램들이다.

도 15 는 본 개시의 기법들을 수행할 수도 있는 예시의 비디오 인코더를 도시하는 블록 다이어그램이다.

도 16 은 본 개시의 기법들을 수행할 수도 있는 예시의 비디오 디코더를 도시하는 블록 다이어그램이다.

도 17 은 현재 블록을 인코딩하기 위한 예시의 방법을 도시하는 플로우챠트이다.

도 18 은 비디오 데이터의 현재 블록을 디코딩하기 위한 예시의 방법을 도시하는 플로우챠트이다.

도 19 는 본 개시의 하나 이상의 기법들에 따른, 비디오 데이터를 코딩하는 예시의 방법을 도시하는 플로우챠트이다.

도 20 은 본 개시의 하나 이상의 기법들에 따른, 크로마 블록을 코딩하는 예시의 방법을 도시하는 플로우챠트이다.

## *발명을 실시하기 위한 구체적인 내용*

[0012] 아핀 선형 가중 인트라 예측 (ALWIP) 은 블록의 참조 샘플들을 다운샘플링하고, 다운샘플링된 참조 샘플들의 벡터에 매트릭스를 곱하고 바이어스 벡터를 부가하고, 결과의 샘플 값들을 예측 블록 내의 미리결정된 포지션들에 매핑하며, 예측 블록에서의 샘플 값들에 대해 선형 보간을 수행하여 예측 블록 내의 나머지 포지션들에 대한 샘플 값들을 결정함으로써 예측 블록이 생성되는 인트라 예측의 형태이다. ALWIP 의 사용은 일부 타입의 블록들에 대해 우수한 코딩 효율성을 유도할 수도 있지만, 정규 인트라 예측 기법들이 다른 타입의 블록들에 대해 ALWIP 보다 우수한 코딩 효율성을 유도할 수도 있다. 따라서, 픽처 내의 일부 블록들은 ALWIP 를 사용하여 코딩될 수도 있고, 픽처 내의 일부 블록들은 평면 인트라 예측 모드, DC 인트라 예측 모드, 또는 방향성 인트라 예측 모드와 같은 정규 인트라 예측 기법들을 사용하여 코딩될 수도 있다. 위의 예는 ALWIP 에 대한 여러 단계들을 설명하지만, 일부 구현들은 위의 단계들 중 하나 이상을 제외하도록 선정할 수도 있고, 다른 구현들은 도출 프로세스에 대한 다른 단계들을 포함할 수도 있다. 그러나, ALWIP 의 핵심 아이디어는 참조 샘플들로부터 도출된 샘플들의 벡터에 매트릭스를 곱하는 것이다 (또는 동등한 연산).

[0013] 인트라 예측 모드 (예를 들어, 평면, DC 또는 방향성 인트라 예측 모드) 를 시그널링하는 효율성을 개선하기 위해, 비디오 코더 (예를 들어, 비디오 인코더 또는 비디오 디코더) 는 비디오 데이터의 현재 블록에 대해 하나 이상의 최확 모드 (MPM) 들을 도출하는 MPM 도출 프로세스를 수행할 수도 있다. MPM들의 각각은 인트라 예측 모드일 수도 있다. 현재 블록에 대한 최적의 인트라 예측 모드가 MPM들 중 하나인 경우, MPM들 중 하나를 표시하는 인덱스는 비트스트림에서 시그널링될 수도 있다. 현재 블록에 대한 최적의 인트라 예측 모드가 MPM들 중 하나가 아닌 경우, 비-MPM 인트라 예측 모드들 중 하나를 표시하는 인덱스가 시그널링될 수도 있다. 비-MPM 인트라 예측 모드들보다 더 적은 MPM들이 있기 때문에, MPM들을 표시하는 인덱스는 비-MPM 인트라 예측 모드들 중 하나를 표시하는 인덱스보다 상당히 적은 비트를 사용하여 코딩될 수도 있다. 따라서, MPM 도출 프로세스의 사용은 일부 블록들에 대한 코딩 효율을 증가시킬 수도 있다.

[0014] 비디오 코더는 현재 블록에 이웃하는 블록들의 인트라 예측 모드들에 기초하여 MPM들을 도출할 수도 있다. 예를 들어, 좌측 이웃 블록이 DC 인트라 예측 모드를 사용하여 코딩되는 경우, 비디오 코더는 MPM들 중 하나가 DC 인트라 예측 모드라고 결정할 수도 있다. 그러나, MPM 도출 프로세스에서, 이웃 블록들 중 하나가 ALWIP 를 사용하여 코딩되는 경우, 비디오 코더는 이웃 블록의 ALWIP 파라미터들을 방향성 인트라 예측 모드들 중 하나에 매핑하는 매핑 데이터를 사용하는 프로세스를 수행한다. ALWIP 파라미터들은 매트릭스 계수들 및 바이어스 벡터를 포함한다. 그 후 비디오 코더는 결정된 방향성 인트라 예측 모드를 MPM들 중 하나로서 사용할 수도 있다.

[0015] 매핑 데이터를 저장하면 비디오 인코더 및 비디오 디코더의 데이터 저장 요건들을 증가시킨다. 증가된 데이터 저장 요건들은 비디오 인코더 및 비디오 디코더의 비용과 복잡성을 증가시킬 수도 있다. 게다가, 매핑 데이터를 사용하기 위해 수행되는 룩업 동작들은 시간이 걸리며, 이는 인코딩 및 디코딩 프로세스 속도를 늦출 수도 있다. 부가적으로, 매핑 데이터를 사용하여 결정된 인트라 예측 모드를 사용하여 생성된 예측 블록은 이웃 블록의 ALWIP 파라미터들이 사용되었던 경우 생성될 예측 블록의 근사치일 뿐이다. 이러한 근사치는 코딩 효율을 감소시킬 수도 있다.

[0016] 본 개시의 기법들은 이러한 쟁점들 중 하나 이상을 해결할 수도 있다. 예를 들어, 본 개시의 예에서 설명된 바와 같이, 비디오 코더 (예를 들어, 비디오 인코더 또는 비디오 디코더) 는 비디오 데이터의 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 도출 프로세스를 수행할 수도 있다. 이 예에서, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. MPM 도출 프로세스를 수행하는 것의 일부로서, 비디오 코더는 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정할 수도 있다. 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 비디오 코더는 이웃 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정할 수도 있다. 따라서, 비디오 코더는 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함할 수도 있다. 비디오 코더는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩할 수도

있다.

[0017] 이웃 블록이 ALWIP-코딩될 때 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 사용함으로써, 비디오 코더는 매핑 데이터를 저장할 필요성을 회피할 수도 있다. 매핑 데이터를 저장할 필요성을 회피하면 비디오 코더의 데이터 저장 요건들을 감소시킬 수도 있으며, 이는 비디오 코더의 비용 및 복잡성을 감소시킬 수도 있다. 또한, 매핑 데이터의 사용을 회피하면 매핑 데이터의 사용으로 수반된 룩업 동작들을 회피함으로써 인코딩 및 디코딩 프로세스들을 가속화할 수도 있다.

[0018] 도 1 은 본 개시의 기법들을 수행할 수도 있는 비디오 인코딩 및 디코딩 시스템 (100) 의 예를 도시하는 블록 다이어그램이다. 본 개시의 기법들은 일반적으로 비디오 데이터를 코딩 (인코딩 및/또는 디코딩) 하는 것에 관련된다. 일반적으로, 비디오 데이터는 비디오를 프로세싱하기 위한 임의의 데이터를 포함한다. 따라서, 비디오 데이터는 원시, 인코딩되지 않은 비디오, 인코딩된 비디오, 디코딩된 (예를 들어, 복원된) 비디오, 및 비디오 메타데이터, 예컨대 시그널링 데이터를 포함할 수도 있다.

[0019] 도 1 에 나타낸 바와 같이, 시스템 (100) 은 이 예에서 목적지 디바이스 (14) 에 의해 디코딩 및 디스플레이될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (102) 를 포함한다. 특히, 소스 디바이스 (102) 는 컴퓨터 판독가능 매체 (110) 를 통해 목적지 디바이스 (116) 에 비디오 데이터를 제공한다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 데스크탑 컴퓨터, 노트북 (즉, 랩탑) 컴퓨터, 셋탑 박스, 모바일 디바이스, 예컨대 전화기 핸드셋 (예를 들어, 스마트폰) 및 태블릿 컴퓨터, 텔리비전, 카메라, 디스플레이 디바이스, 디지털 미디어 플레이어, 비디오 게이밍 콘솔, 비디오 스트리밍 디바이스, 브로드캐스트 수신기 디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 무선 통신을 위해 장비될 수도 있고, 따라서 무선 통신 디바이스로 지칭될 수도 있다.

[0020] 도 1 의 예에서, 소스 디바이스 (102) 는 비디오 소스 (104), 메모리 (106), 비디오 인코더 (200) 및 출력 인터페이스 (108) 를 포함한다. 목적지 디바이스 (116) 는 입력 인터페이스 (122), 비디오 디코더 (300), 메모리 (120), 및 디스플레이 디바이스 (118) 를 포함한다. 본 개시에 따라, 소스 디바이스 (102) 의 비디오 인코더 (200) 및 목적지 디바이스 (116) 의 비디오 디코더 (300) 는 선형 가중 인트라 예측을 위한 모드들의 도출 및 시그널링을 포함하는, 인트라 예측을 수행하기 위한 기법들을 적용하도록 구성될 수도 있다. 따라서, 소스 디바이스 (102) 는 비디오 인코딩 디바이스의 예를 나타내는 한편, 목적지 디바이스 (116) 는 비디오 디코딩 디바이스의 예를 나타낸다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102) 는 외부 카메라와 같은 외부 비디오 소스로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (116) 는 통합된 디스플레이 디바이스를 포함하기 보다는, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0021] 도 1 에 나타낸 시스템 (10) 은 단지 하나의 예일 뿐이다. 일반적으로, 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스는 선형 가중 인트라 예측을 위한 모드들의 도출 및 시그널링을 포함하는, 인트라 예측을 수행하기 위한 기법들을 수행할 수도 있다. 소스 디바이스 (102) 및 목적지 디바이스 (102) 는 단지 소스 디바이스 (116) 가 목적지 디바이스 (116) 로의 송신을 위한 인코딩된 비디오 데이터를 생성하는 그러한 코딩 디바이스들의 예들일 뿐이다. 본 개시는 데이터의 코딩 (인코딩 및/또는 디코딩) 을 수행하는 디바이스로서 "코딩" 디바이스를 언급한다. 따라서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 코딩 디바이스, 특히 각각 비디오 인코더 및 비디오 디코더의 예들을 나타낸다. 일부 예들에서, 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 소스 디바이스 (102) 및 목적지 디바이스 (116) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭 방식으로 동작할 수도 있다. 이로써, 시스템 (100) 은 예를 들어, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅 또는 비디오 텔레포니를 위해, 소스 디바이스 (102) 와 목적지 디바이스 (116) 사이의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0022] 일반적으로, 비디오 소스 (104) 는 비디오 데이터 (즉, 원시, 인코딩되지 않은 비디오 데이터) 의 소스를 나타내며 픽처들에 대한 데이터를 인코딩하는 비디오 인코더 (200) 에 비디오 데이터의 순차적인 일련의 픽처들 (또한 "프레임들" 로서 지칭됨) 을 제공한다. 소스 디바이스 (102) 의 비디오 소스 (104) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 원시 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스를 포함할 수도 있다. 추가적인 대안으로서, 비디오 소스 (104) 는 컴퓨터 그래픽 기반 데이터를 소스 비디오로서, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합으로서 생성할 수도 있다. 각각의 경우, 비디오 인코더 (200) 는 캡처되거나, 사전-캡처되거나, 또는 컴퓨터 생성된 비디오 데이터를 인코딩한다. 비디오 인코더 (200) 는 수신된 순

서 (때때로 "디스플레이 순서"로 지칭됨) 로부터 픽처들을 코딩을 위한 코딩 순서로 재배열할 수도 있다. 비디오 인코더 (200) 는 인코딩된 비디오 데이터를 포함하는 비트스트림을 생성할 수도 있다. 소스 디바이스 (102) 는 그 후 예를 들어, 목적지 디바이스 (116) 의 입력 인터페이스 (122) 에 의한 수신 및/또는 취출을 위해 컴퓨터 판독가능 매체 (110) 상으로 출력 인터페이스 (108) 를 통해 인코딩된 비디오 데이터를 출력할 수도 있다.

[0023] 소스 디바이스 (102) 의 메모리 (106) 및 목적지 디바이스 (116) 의 메모리 (120) 는 범용 메모리들을 나타낸다. 일부 예에서, 메모리들 (106, 120) 은 원시 비디오 데이터, 예를 들어 비디오 소스 (104) 로부터의 원시 비디오, 및 비디오 디코더 (300) 로부터의 원시, 디코딩된 비디오 데이터를 저장할 수도 있다. 부가적으로 또는 대안으로, 메모리들 (106, 120) 은 예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해 각각 실행가능한 소프트웨어 명령들을 저장할 수도 있다. 이 예에서는 메모리 (106) 및 메모리 (120) 가 비디오 인코더 (200) 및 비디오 디코더 (300) 와 별도로 나타나 있지만, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 또한 기능적으로 유사하거나 동등한 목적을 위한 내부 메모리들을 포함할 수 있음을 이해해야 한다. 또한, 메모리들 (106, 120) 은 예를 들어, 비디오 인코더 (200) 로부터 출력되고 비디오 디코더 (300) 에 입력된 인코딩된 비디오 데이터를 저장할 수도 있다. 일부 예들에서, 메모리들 (106, 120) 의 부분들은 예를 들어, 원시, 디코딩된 및/또는 인코딩된 비디오 데이터를 저장하기 위해 하나 이상의 비디오 버퍼들로서 할당될 수도 있다.

[0024] 컴퓨터 판독가능 매체 (110) 는 인코딩된 비디오 데이터를 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로 전송할 수 있는 임의의 타입의 매체 또는 디바이스를 나타낼 수도 있다. 일 예에서, 컴퓨터 판독가능 매체 (110) 는 소스 디바이스 (102) 가 실시간으로, 예를 들어 무선 주파수 네트워크 또는 컴퓨터 기반 네트워크를 통해 직접 목적지 디바이스 (116) 로 인코딩된 비디오 데이터를 송신하는 것을 가능하게 하기 위한 통신 매체를 나타낸다. 출력 인터페이스 (108) 는 인코딩된 비디오 데이터를 포함하는 송신 신호를 변조할 수도 있고, 입력 인터페이스 (122) 는 무선 통신 프로토콜과 같은 통신 표준에 따라 수신된 송신 신호를 변조할 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 예컨대 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예컨대 인터넷의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0025] 일부 예들에서, 컴퓨터 판독가능 매체 (110) 는 저장 디바이스 (112) 를 포함할 수도 있다. 소스 디바이스 (102) 는 출력 인터페이스 (108) 로부터 저장 디바이스 (112) 로 인코딩된 데이터를 출력할 수도 있다. 유사하게, 목적지 디바이스 (116) 는 입력 인터페이스 (122) 를 통해 저장 디바이스 (112) 로부터의 인코딩된 데이터에 액세스할 수도 있다. 저장 디바이스 (112) 는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체와 같은 다양한 분산된 또는 로컬 액세스된 데이터 저장 매체 중 임의의 것을 포함할 수도 있다.

[0026] 일부 예들에서, 컴퓨터 판독가능 매체 (110) 는, 소스 디바이스 (102) 에 의해 생성된 인코딩된 비디오 데이터를 저장할 수도 있는 파일 서버 (114) 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 소스 디바이스 (102) 는 소스 디바이스 (102) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 (114) 또는 다른 중간 저장 디바이스로 인코딩된 비디오 데이터를 출력할 수도 있다. 목적지 디바이스 (116) 는 스트리밍 또는 다운로드를 통해 파일 서버 (114) 로부터 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버 (114) 는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (116) 로 송신할 수 있는 임의의 타입의 서버 디바이스일 수도 있다. 파일 서버 (114) 는 (예를 들어, 웹 사이트를 위한) 웹 서버, 파일 전송 프로토콜 (FTP) 서버, 콘텐츠 전달 네트워크 디바이스, 또는 NAS (network attached storage) 디바이스를 나타낼 수도 있다. 목적지 디바이스 (116) 는 인터넷 접속을 포함하는, 임의의 표준 데이터 접속을 통해 파일 서버 (114) 로부터 인코딩된 비디오 데이터에 액세스할 수도 있다. 이는 파일 서버 (114) 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한, 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, 디지털 가입자 라인 (DSL), 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 파일 서버 (114) 및 입력 인터페이스 (122) 는 스트리밍 송신 프로토콜, 다운로드 송신 프로토콜 또는 이들의 조합에 따라 동작하도록 구성될 수도 있다.

[0027] 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 무선 송신기/수신기, 모뎀, 유선 네트워킹 컴포넌트 (예를

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

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

[0029] 목적지 디바이스 (116) 의 입력 인터페이스 (122) 는 컴퓨터 판독가능 매체 (110)(예를 들어, 통신 매체, 저장 디바이스 (112), 파일 서버 (114) 등) 로부터 인코딩된 비디오 비트스트림을 수신한다. 인코딩된 비디오 비트스트림은 비디오 블록들 또는 다른 코딩된 유닛들 (예를 들어, 슬라이스들, 픽처들, 픽처들의 그룹들, 시퀀스들 등) 의 프로세싱 및/또는 특징들을 기술하는 값들을 갖는 신택스 엘리먼트들과 같은, 비디오 디코더 (300) 에 의해 또한 사용되는, 비디오 인코더 (200) 에 의해 정의된 시그널링 정보를 포함할 수도 있다. 디스플레이 디바이스 (118) 는 디코딩된 비디오 데이터의 디코딩된 픽처들을 사용자에게 디스플레이한다. 디스플레이 디바이스 (118) 는 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 나타낼 수도 있다.

[0030] 도 1 에 나타내지는 않았지만, 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각 오디오 인코더 및/또는 오디오 디코더와 통합될 수도 있고, 공통 데이터 스트림에서 오디오 및 비디오 양자 모두를 포함하는 멀티플렉싱된 스트림을 핸들링하기 위해, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및/또는 소프트웨어를 포함할 수도 있다. 적용가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜 또는 다른 프로토콜들, 예컨대 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0031] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각 다양한 적합한 인코더 및/또는 디코더 회로부, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서(DSP)들, 주문형 집적회로(ASIC)들, 필드 프로그램가능 게이트 어레이(FPGA)들, 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들으로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어로 구현될 때, 디바이스는 적합한 비일시적 컴퓨터 판독가능 매체에 소프트웨어를 위한 명령들을 저장하고 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 명령들을 실행할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있는데, 이들 중 어느 하나는 개개의 디바이스에 있어서 조합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다. 비디오 인코더 (200) 및/또는 비디오 디코더 (300) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 셀룰러 전화기와 같은 무선 통신 디바이스를 포함할 수도 있다.

[0032] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 고효율 비디오 코딩 (HEVC) 으로서 또한 지칭되는 ITU-T H.265 와 같은 비디오 코딩 표준 또는 그에 대한 확장들, 예컨대 멀티-뷰 및/또는 스케일러블 비디오 코딩 확장들에 따라 동작할 수도 있다. 대안으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다기능 비디오 코딩 (VVC) 으로서 또한 지칭되는 ITU-T H.266 또는 공동 탐구 테스트 모델 (JEM) 과 같은, 다른 독점 또는 산업 표준들에 따라 동작할 수도 있다. VVC 표준의 초안은 Bross 등의 "Versatile Video Coding (Draft 4)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 13th Meeting: Marrakech, MA, 9-18 Jan. 2019, JVET-M1001 (이하 "VVC WD4") 에 기재되어 있다. VVC 표준의 다른 초

안은 Bross 등의 "Versatile Video Coding (Draft 5)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting: Geneva, CH, 19-27 March 2019, JVET-N1001-v5 (이하 "VVC Draft 5") 에 기재되어 있다. VVC 의 목적은 보다 높은 품질의 비디오 서비스 및 부상하는 애플리케이션들 예컨대 360° 전방위 몰입형 멀티미디어 및 HDR (high-dynamic-range) 비디오의 전개를 보조하면서, 기존의 HEVC 표준에 비해 압축 성능의 상당한 개선을 제공하는 것이다. 하지만, 본 개시의 기법들은 임의의 특정 코딩 표준에 제한되지 않는다.

[0033] 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 픽처들의 블록 기반 코딩을 수행할 수도 있다. 용어 "블록" 은 일반적으로 프로세싱될 (예를 들어, 인코딩될, 디코딩될, 또는 인코딩 및/또는 디코딩 프로세스에서 사용될) 데이터를 포함하는 구조를 지칭한다. 예를 들어, 블록은 루미넌스 및/또는 크로미넌스 데이터의 샘플들의 2 차원 매트릭스를 포함할 수도 있다. 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 YUV (예를 들어, Y, Cb, Cr) 포맷으로 표현된 비디오 데이터를 코딩할 수도 있다. 즉, 픽처의 샘플들에 대한 적색, 녹색, 및 청색 (RGB) 을 코딩하기 보다, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 (루마) 및 크로미넌스 성분들을 코딩할 수도 있으며, 여기서 크로미넌스 성분들은 적색 색조 및 청색 색조 크로미넌스 성분들 양자 모두를 포함할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 인코딩 이전에 수신된 RGB 포맷된 데이터를 YUV 표현으로 변환하고, 비디오 디코더 (300) 는 YUV 표현을 RGB 포맷으로 변환한다. 대안으로, 프리-프로세싱 및 포스트-프로세싱 유닛들 (도시되지 않음) 이 이들 변환들을 수행할 수도 있다.

[0034] 본 개시는 일반적으로 픽처의 데이터를 인코딩 또는 디코딩하는 프로세스를 포함하기 위해 픽처들의 코딩 (예를 들어, 인코딩 및 디코딩) 을 언급한다. 유사하게, 본 개시는 블록들에 대한 데이터를 인코딩 또는 디코딩하는 프로세스, 예를 들어 예측 및/또는 잔차 코딩을 포함하기 위해 픽처의 블록들의 코딩을 언급할 수도 있다. 인코딩된 비디오 비트스트림은 일반적으로 픽처들의 블록들로의 파티셔닝 및 코딩 판정들 (예를 들어, 코딩 모드들) 을 나타내는 신택스 엘리먼트들에 대한 일련의 값들을 포함한다. 따라서, 픽처 또는 블록을 코딩하는 것에 대한 참조들은 일반적으로 픽처 또는 블록을 형성하는 신택스 엘리먼트에 대한 코딩 값들로서 이해되어야 한다.

[0035] HEVC 는 코딩 유닛 (CU), 예측 유닛 (PU) 및 변환 유닛 (TU) 을 포함하는 다양한 블록들을 정의한다. HEVC 에 따라, (비디오 인코더 (200) 와 같은) 비디오 코더는 쿼드트리 구조에 따라 코딩 트리 유닛 (CTU) 을 CU들로 파티셔닝한다. 즉, 비디오 코더는 CTU들 및 CU들을 4 개의 동등한, 오버랩하지 않는 정사각형들로 파티셔닝하고, 쿼드트리의 각 노드는 0 또는 4 개의 자식 노드들을 갖는다. 자식 노드가 없는 노드들은 "리프 노드들" 로서 지칭될 수도 있으며, 이러한 리프 노드들의 CU들은 하나 이상의 PU들 및/또는 하나 이상의 TU들을 포함할 수도 있다. 비디오 코더는 PU들 및 TU들을 추가로 파티셔닝할 수도 있다. 예를 들어, HEVC 에서, 잔차 쿼드트리 (RQT) 는 TU들의 파티셔닝을 나타낸다. HEVC 에서, PU들은 인터 예측 데이터를 나타내는 한편, TU들은 잔차 데이터를 나타낸다. 인트라 예측되는 CU들은 인트라 예측 모드 표시와 같은 인트라 예측 정보를 포함한다.

[0036] 다른 예로서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 JEM 또는 VVC 에 따라 동작하도록 구성될 수도 있다. JEM 또는 VVC 에 따르면, (비디오 인코더 (200) 와 같은) 비디오 코더는 픽처를 복수의 CTU들로 파티셔닝한다. 비디오 인코더 (200) 는 쿼드트리-이진 트리 (QTBT) 구조 또는 멀티-타입 트리 (MTT) 구조와 같은 트리 구조에 따라 CTU 를 파티셔닝할 수도 있다. QTBT 구조는 HEVC 의 CU들, PU들, 및 TU들 사이의 분리와 같은, 다중 파티션 타입들의 개념들을 제거한다. QTBT 구조는 2 개의 레벨들: 쿼드트리 파티셔닝에 따라 파티셔닝된 제 1 레벨, 및 이진 트리 파티셔닝에 따라 파티셔닝된 제 2 레벨을 포함한다. QTBT 구조의 루트 노드는 CTU 에 대응한다. 이진 트리들의 리프 노드들은 코딩 유닛들 (CU들) 에 대응한다.

[0037] MTT 파티셔닝 구조에서, 블록들은 쿼드트리 (QT) 파티션, 이진 트리 (BT) 파티션, 및 트리플 트리 (TT) 파티션들 중 하나 이상의 타입들을 사용하여 파티셔닝될 수도 있다. 트리플 트리 파티션은 블록이 3 개의 서브-블록들로 스플릿되는 파티션이다. 일부 예들에서, 트리플 트리 파티션은 중앙을 통해 원래 블록을 분할하지 않으면서 블록을 3 개의 서브-블록들로 분할한다. MTT 에서의 파티셔닝 타입들 (예를 들어, QT, BT 및 TT) 은 대칭적이거나 비대칭적일 수도 있다.

[0038] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 및 크로미넌스 성분들의 각각을 나타내기 위해 단일 QTBT 또는 MTT 구조를 사용할 수도 있는 한편, 다른 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 성분에 대한 하나의 QTBT/MTT 구조 및 양자의 크로미넌스 성분들에 대한 다른

QTBT/MTT 구조 (또는 개개의 크로미넌스 성분들에 대한 2 개의 QTBT/MTT 구조들) 와 같은, 2 이상의 QTBT 또는 MTT 구조들을 사용할 수도 있다.

[0039] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 쿼드트리 파티셔닝, QTBT 파티셔닝, MTT 파티셔닝, 또는 다른 파티셔닝 구조들을 사용하도록 구성될 수도 있다. 설명의 목적을 위해, 본 개시의 기법들의 기재는 QTBT 파티셔닝에 대하여 제시된다. 그러나, 본 개시의 기법들은 또한, 쿼드트리 파티셔닝, 또는 다른 타입들의 파티셔닝에도 사용하도록 구성된 비디오 코더들에 적용될 수도 있음이 이해되어야 한다.

[0040] 본 개시는 수직 및 수평 치수들에 관하여 (CU 또는 다른 비디오 블록과 같은) 블록의 샘플 치수들을 지칭하기 위해 "NxN"및 "N 바이 N", 예를 들어 16x16 샘플들 또는 16 바이 16 샘플들을 상호교환가능하게 사용할 수도 있다. 일반적으로, 16×16 CU 는 수직 방향에서 16 개의 샘플들 (y = 16) 및 수평 방향에서 16 개의 샘플들 (x = 16) 을 갖는다. 마찬가지로, NxN CU 는 수직 방향에서 N 샘플들 및 수평 방향에서 N 샘플들을 갖고, 여기서 N 은 음이 아닌 정수 값을 나타낸다. CU 에서의 샘플들은 로우들 및 컬럼들로 배열될 수도 있다. 더욱이, CU들은 수직 방향에서의 동일한 수의 샘플들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, CU들은 N×M 샘플들을 포함할 수도 있고, 여기서 M 은 반드시 N 과 동일한 것은 아니다.

[0041] 비디오 인코더 (200) 는 예측 및/또는 잔차 정보를 나타내는 CU들에 대한 비디오 데이터, 및 다른 정보를 인코딩한다. 예측 정보는 CU 에 대한 예측 블록을 형성하기 위해 CU 가 어떻게 예측될지를 표시한다. 잔차 정보는 일반적으로 인코딩 전의 CU 의 샘플들과 예측 블록 사이의 샘플 별 차이들을 나타낸다.

[0042] CU 를 예측하기 위해, 비디오 인코더 (200) 는 일반적으로 인터-예측 또는 인트라-예측을 통해 CU 에 대한 예측 블록을 형성할 수도 있다. 인터 예측은 일반적으로 하나 이상의 이전에 코딩된 픽처들의 데이터로부터 CU 를 예측하는 것을 지칭하는 반면, 인트라 예측은 일반적으로 동일한 픽처의 이전에 코딩된 데이터로부터 CU 를 예측하는 것을 지칭한다. 인터 예측을 수행하기 위해, 비디오 인코더 (200) 는 하나 이상의 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 비디오 인코더 (200) 는 일반적으로, 예를 들어 CU 와 참조 블록 사이의 차이에 관하여, CU 와 밀접하게 매칭하는 참조 블록을 식별하기 위해 모션 탐색을 수행할 수도 있다. 비디오 인코더 (200) 는 참조 블록이 현재 CU 와 밀접하게 매칭하는지 여부를 결정하기 위해 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of squared differences; SSD), 평균 절대차 (mean absolute difference; MAD), 평균 제곱차 (mean squared differences; MSD) 또는 다른 그러한 차이 계산들을 사용하여 차이 메트릭을 계산할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 단방향 예측 또는 양방향 예측을 사용하여 현재 CU 를 예측할 수도 있다.

[0043] JEM 및 VVC 의 일부 예들은 또한, 인터 예측 모드로 고려될 수도 있는 아핀 모션 보상 모드를 제공한다. 아핀 모션 보상 모드에서, 비디오 인코더 (200) 는 줌인 또는 줌아웃, 회전, 원근 모션 (perspective motion), 또는 다른 불규칙한 모션 타입들과 같은 비-병진 모션을 나타내는 2 이상의 모션 벡터들을 결정할 수도 있다.

[0044] 인트라 예측을 수행하기 위해, 비디오 인코더 (200) 는 예측 블록을 생성하기 위해 인트라 예측 모드를 선택할 수도 있다. JEM 및 VVC 의 일부 예들은 평면 모드 및 DC 모드 뿐만 아니라, 다양한 방향성 모드들을 포함한, 67 개의 인트라-예측 모드들을 제공한다. 일반적으로, 비디오 인코더 (200) 는 현재 블록의 샘플들을 예측할 현재 블록 (예를 들어, CU 의 블록) 에 대한 이웃 샘플들을 기술하는 인트라-예측 모드를 선택한다. 이러한 샘플들은 일반적으로, 비디오 인코더 (200) 가 래스터 스캔 순서로 (좌측에서 우측으로, 상단에서 하단으로) CTU들 및 CU들을 코딩한다고 가정하여, 현재 블록과 동일한 픽처에서 현재 블록의 상측, 상측 및 좌측으로, 또는 좌측으로 있을 수도 있다.

[0045] 일반적으로, 비디오 인코더 (200) 또는 비디오 디코더 (300) 와 같은 비디오 코더는, 이웃 샘플들로부터 추정된 그래디언트로서 평면 모드를 사용하여 예측 블록을 생성할 수도 있다. 예를 들어, VVC WD4 에서, 비디오 코더는 다음과 같이 평면 인트라 예측 모드를 사용하여 예측 블록을 생성할 수도 있다.

[0046] **8.3.4.2.5 INTRA_PLANAR 인트라 예측 모드의 사양**

[0047] 이 프로세스에 대한 입력들은:

[0048] - 변환 블록 폭을 특정하는 변수 nTbW,

[0049] - 변환 블록 높이를 특정하는 변수 nTbH,

[0050] - 이웃 샘플들 p[ x ][ y ], 여기서 x = -1, y = -1..nTbH 이고 x = 0..nTbW, y = -1.

[0051] 이 프로세스의 출력들은 예측된 샘플들 predSamples[ x ][ y ] 이며, 여기서 x = 0..nTbW - 1, y = 0..nTbH - 1 이다.

[0052] 변수들 nW 및 nH 는 다음과 같이 도출된다:

$$nW = Max( nTbW, 2 ) \quad (8\text{-}115)$$

$$nH = Max( nTbH, 2 ) \quad (8\text{-}116)$$

[0053]

[0054] 예측 샘플들 predSamples[ x ][ y ] 의 값들은 다음과 같이 도출되며, 여기서 x = 0..nTbW - 1 이고 y = 0..nTbH - 1 이다.

$$predV[ x ][ y ] = ( ( nH - 1 - y ) * p[ x ][ -1 ] + ( y + 1 ) * p[ -1 ][ nTbH ] ) << Log2 ( nW ) \quad (8\text{-}117)$$

$$predH[ x ][ y ] = ( ( nW - 1 - x ) * p[ -1 ][ y ] + ( x + 1 ) * p[ nTbW ][ -1 ] ) << Log2 ( nH ) \quad (8\text{-}118)$$

$$predSamples[ x ][ y ] = ( predV[ x ][ y ] + predH[ x ][ y ] + nW * nH ) >> (Log2 ( nW ) + Log2 ( nH ) + 1 ) \quad (8\text{-}119)$$

[0055]

[0056] 비디오 인코더 (200) 는 현재 블록에 대한 예측 모드를 나타내는 데이터를 인코딩한다. 예를 들어, 인터 예측 모드들에 대해, 비디오 인코더 (200) 는 이용가능한 다양한 인터-예측 모드들 중 어느 것이 사용되는지를 나타내는 데이터 뿐만 아니라, 대응하는 모드에 대한 모션 정보를 인코딩할 수도 있다. 단방향 또는 양방향 인터-예측을 위해, 예를 들어, 비디오 인코더 (200) 는 어드밴스드 모션 벡터 예측 (AMVP) 또는 병합 모드를 사용하여 모션 벡터들을 인코딩할 수도 있다. 비디오 인코더 (200) 는 유사한 모드들을 사용하여 아핀 모션 보상 모드에 대한 모션 벡터들을 인코딩할 수도 있다.

[0057] 블록의 인트라 예측 또는 인터 예측과 같은 예측에 후속하여, 비디오 인코더 (200) 는 블록에 대한 잔차 데이터를 계산할 수도 있다. 잔차 블록과 같은 잔차 데이터는 대응하는 예측 모드를 사용하여 형성된, 블록과 블록에 대한 예측 블록 사이의 샘플 별 차이들을 나타낸다. 비디오 인코더 (200) 는 샘플 도메인 대신 변환 도메인에서 변환된 데이터를 생성하기 위해, 하나 이상의 변환들을 잔차 블록에 적용할 수도 있다. 예를 들어, 비디오 인코더 (200) 는 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환을 잔차 비디오 데이터에 적용할 수도 있다. 부가적으로, 비디오 인코더 (200) 는 MDNSST (mode-dependent non-separable secondary transform), 신호 의존적 변환, Karhunen-Loeve 변환 (KLT) 등과 같은, 제 1 변환에 후속하는 2차 변환을 적용할 수도 있다. 비디오 인코더 (200) 는 하나 이상의 변환들의 적용을 통해 변환 계수들을 생성한다.

[0058] 위에 언급된 바와 같이, 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (200) 는 변환 계수들의 양자화를 수행할 수도 있다. 일반적으로 양자화는 변환 계수들이 양자화되어 그 변환 계수들을 나타내는데 사용된 데이터의 양을 감소시킬 수 있어서, 추가 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스를 수행함으로써, 비디오 인코더 (200) 는 변환 계수들의 일부 또는 모두와 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, 비디오 인코더 (200) 는 양자화 동안 n-비트 값을 m-비트 값으로 라운딩 다운할 수도 있으며, 여기서 n 은 m 보다 크다. 일부 예들에서, 양자화를 수행하기 위해, 비디오 인코더 (200) 는 양자화될 값의 비트단위 우측-시프트를 수행할 수도 있다.

[0059] 양자화에 후속하여, 비디오 인코더 (200) 는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 포함하는 2차원 매트릭스로부터 1차원 벡터를 생성할 수도 있다. 스캔은 더 높은 에너지 (및 따라서 더 낮은 주파수) 변환 계수들을 벡터의 전방에 배치하고 그리고 더 낮은 에너지 (및 따라서 더 높은 주파수) 변환 계수들을 벡터의 후방에 배치하도록 설계될 수도 있다. 일부 예들에 있어서, 비디오 인코더 (200) 는 양자화된 변환 계수들을 스캐닝하기 위해 미리정의된 스캔 순서를 활용하여 직렬화된 벡터를 생성하고, 그 후 벡터의 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (200) 는 적응 스캔을 수행할 수도 있다. 1차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캐닝한 후, 비디오 인코더 (200) 는 예를 들어, 컨텍스트 적응 이진 산술 코딩 (CABAC) 에 따라 1차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더

(200) 는 또한, 비디오 데이터를 디코딩하는데 있어서 비디오 디코더 (300) 에 의한 사용을 위해 인코딩된 비디오 데이터와 연관된 메타데이터를 기술하는 신택스 엘리먼트들에 대한 값들을 엔트로피 인코딩할 수도 있다. CABAC 을 수행하기 위해, 비디오 인코더 (200) 는 송신될 심볼에 컨텍스트 모델 내의 컨텍스트를 할당할 수도 있다.

[0060] 위에 언급된 바와 같이, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 CABAC 인코딩 및 디코딩을 신택스 엘리먼트들의 값들에 적용할 수도 있다. CABAC 인코딩을 신택스 엘리먼트에 적용하기 위해, 비디오 인코더 (200) 는 신택스 엘리먼트의 값을 이진화하여 "빈들" 로서 지칭되는 일련의 하나 이상의 비트들을 형성할 수도 있다. 또한, 비디오 인코더 (200) 는 코딩 컨텍스트를 식별할 수도 있으며, 이는 간단히 "컨텍스트"로 지칭될 수도 있다. 코딩 컨텍스트는 특정 값을 갖는 빈들의 확률을 식별할 수도 있다. 예를 들어, 코딩 컨텍스트는 0-값의 빈을 코딩하는 0.7 확률 및 1-값의 빈을 코딩하는 0.3 확률을 표시할 수도 있다. 코딩 컨텍스트를 식별한 후, 비디오 인코더 (200) 는 인터벌을 하위 서브-인터벌 및 상위 서브-인터벌로 분할할 수도 있다. 하위-인터벌들 중 하나는 값 0 과 연관될 수도 있고 다른 하위-인터벌은 값 1 과 연관될 수도 있다. 서브 간격들의 폭은 식별된 코딩 컨텍스트에 의해 연관된 값들에 대해 표시된 확률에 비례할 수도 있다. 신택스 엘리먼트의 빈이 하위 서브-인터벌과 연관된 값을 갖는 경우, 인코딩된 값은 하위 서브-인터벌의 하위 경계와 동일할 수도 있다. 신택스 엘리먼트의 동일한 빈이 상위 서브-인터벌과 연관된 값을 갖는 경우, 인코딩된 값은 상위 서브-인터벌의 하위 경계와 동일할 수도 있다. 신택스 엘리먼트의 다음 빈을 인코딩하기 위해, 비디오 인코더 (200) 는 인코딩된 비트의 값과 연관된 서브-인터벌인 인터벌로 이러한 단계들을 반복할 수도 있다. 비디오 인코더 (200) 가 다음 빈에 대해 이러한 단계들을 반복할 때, 비디오 인코더 (200) 는 인코딩된 빈들의 실제 값 및 식별된 코딩 컨텍스트에 의해 표시된 확률에 기초하여 수정된 확률을 사용할 수도 있다.

[0061] 비디오 디코더 (300) 가 신택스 엘리먼트의 값에 대해 CABAC 디코딩을 수행할 때, 비디오 디코더 (300) 는 코딩 컨텍스트를 식별할 수도 있다. 비디오 디코더 (300) 는 그 후 인터벌을 하위 서브-인터벌 및 상위 서브-인터벌로 분할할 수도 있다. 하위-인터벌들 중 하나는 값 0 과 연관될 수도 있고 다른 하위-인터벌은 값 1 과 연관될 수도 있다. 서브 간격들의 폭은 식별된 코딩 컨텍스트에 의해 연관된 값들에 대해 표시된 확률에 비례할 수도 있다. 인코딩된 값이 하위 서브-인터벌 내에 있는 경우, 비디오 디코더 (300) 는 하위 서브-인터벌과 연관된 값을 갖는 빈을 디코딩할 수도 있다. 인코딩된 값이 상위 서브-인터벌 내에 있는 경우, 비디오 디코더 (300) 는 상위 서브-인터벌과 연관된 값을 갖는 빈을 디코딩할 수도 있다. 신택스 엘리먼트의 다음 빈을 디코딩하기 위해, 비디오 디코더 (300) 는 인코딩된 값을 포함하는 서브-인터벌인 인터벌로 이러한 단계들을 반복할 수도 있다. 비디오 디코더 (300) 가 다음 빈에 대해 이러한 단계들을 반복할 때, 비디오 디코더 (300) 는 디코딩된 빈들 및 식별된 코딩 컨텍스트에 의해 표시된 확률에 기초하여 수정된 확률을 사용할 수도 있다. 비디오 디코더 (300) 는 그 후 신택스 엘리먼트의 값을 복구하기 위해 빈들을 이진화해제할 수도 있다.

[0062] 일부 경우들에서, 비디오 인코더 (200) 는 바이패스 CABAC 코딩을 이용하여 빈들을 인코딩할 수도 있다. 빈에 대해 정규 CABAC 코딩을 수행하기 보다는 빈에 대해 바이패스 CABAC 코딩을 수행하는 것이 계산적으로 비용이 덜 들 수도 있다. 또한, 바이패스 CABAC 코딩을 수행하는 것은 고도의 병렬화 (parallelization) 및 스루풋 (throughput) 을 허용할 수도 있다. 바이패스 CABAC 코딩을 이용하여 인코딩된 빈들은 "바이패스 빈들" 로서 지칭될 수도 있다. 바이패스 빈들을 함께 그룹핑하는 것은 비디오 인코더 (200) 및 비디오 디코더 (300) 의 스루풋을 증가시킬 수도 있다. 바이패스 CABAC 코딩 엔진은 단일 사이클에서 수 개의 빈들을 코딩하는 것이 가능할 수도 있는 반면, 정규 CABAC 코딩 엔진은 사이클에서 단지 단일 빈만을 코딩하는 것이 가능할 수도 있다. 바이패스 CABAC 코딩 엔진은 그 바이패스 CABAC 코딩 엔진이 컨텍스트들을 선택하지 않고 양 심볼들 (0 및 1) 에 대해 1/2 의 확률을 가정할 수도 있기 때문에 더 단순할 수도 있다. 결과적으로, 바이패스 CABAC 코딩에서, 간격들은 직접 절반으로 분할된다.

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

[0064] 이러한 방식으로, 비디오 인코더 (200) 는 인코딩된 비디오 데이터, 예를 들어 픽처의 블록들 (예를 들어, CU들) 로의 파티셔닝을 기술하는 신택스 엘리먼트들 및 블록들에 대한 예측 및/또는 잔차 정보를 포함하는 비트

스트림을 생성할 수도 있다. 궁극적으로, 비디오 디코더 (300) 는 비트스트림을 수신하고 인코딩된 비디오 데이터를 디코딩할 수도 있다.

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

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

[0067] 인트라 예측은 DC 예측 모드, 평면 예측 모드 및 방향성 (또는 각도) 예측 모드들을 포함한다. 정사각형 블록들에 대한 방향성 예측은 도 2 에 도시된 바와 같이, VVC 테스트 모델 2 (VTM2)(J. Chen, Y. Ye, S. Kim, "Algorithm description for Versatile Video Coding and Test Model 2 (VTM2)," 11th JVET Meeting, Ljubljana, SI, July 2018, JVET-K1002) 에서 현재 블록의 - 135 도와 45 도 사이의 방향들을 사용한다. 도 2 는 참조 샘플들을 가리키는 화살표들을 갖는, 인트라 예측의 예를 도시하는 블록 다이어그램이다.

[0068] VTM2 에서, 인트라 예측을 위하여 예측 블록을 규정하는데 사용된 블록 구조는 정사각형 (폭 w = 높이 h) 으로 제한되지 않는다. 직사각형 또는 비정사각형 예측 블록들 (w > h 또는 w < h) 은 콘텐츠의 특징들에 기초하여 코딩 효율을 증가시킬 수 있다.

[0069] 이러한 직사각형 블록들에서, -135 도 내지 45 도 내에 있도록 인트라 예측의 방향을 제한하면 더 가까운 참조 샘플들보다는 더 먼 참조 샘플들이 인트라 예측을 위하여 사용되는 상황들을 초래할 수 있다. 이러한 설계는 코딩 효율성에 영향을 미칠 가능성이 있다: (-135 내지 45 도 각도를 넘어서는) 더 가까운 참조 샘플들이 예측에 사용될 수도 있도록 완화된 제한 범위를 갖는 것이 더 유익하다. 이러한 경우의 예가 도 3 에 주어진다. 도 3 은 '더 가까운' 참조 샘플들 (상부 우측 파선 원 (502)) 은 사용되지 않지만, -135 도 내지 45 도 범위에 있도록 인트라 예측 방향이 제한되는 것으로 인해 더 먼 참조 샘플들 (하부 좌측 파선 원 (504)) 이 사용될 수도 있는, 예시의 8x4 직사각형 블록의 개념적 다이어그램이다. 도 4 는 VTM2 에서 채택되는 넓은 각도들의 예시이다.

[0070] 제 12 회 JVET 미팅 동안, 넓은 각도 인트라 예측의 수정이 VTM3 에 채택되었다 (L. Zhao, X. Zhao, S. Liu, X. Li, "CE3-related: Unification of angular intra prediction for square and non-square blocks," 12th JVET Meeting, Macau SAR, CN, Oct. 2018, JVET-L0279; J. Chen, Y. Ye, S. Kim, "Algorithm description for Versatile Video Coding and Test Model 3 (VTM3)," 12th JVET Meeting, Macau SAR, CN, Oct. 2018, JVET-L1002; B. Bross, J. Chen, S. Liu, "Versatile Video Coding (Draft 3)," 12th JVET Meeting, Macau SAR, CN, Oct. 2018, JVET-L1001 참조). 이 채택은 정사각형 블록 및 비정사각형 블록에 대해 각도 인트라 예측을 통합하기 위한 2개의 수정들을 포함한다. 첫번째로, 모든 블록 형상들의 대각선 방향들을 커버하도록 각도 예측 방향들이 수정된다. 두번째로, 도 5a 내지 도 5c 에 도시된 바와 같이 모든 각도 방향들이 모든 블록 종횡비들 (정사각형 및 비정사각형) 에 대해 하단 좌측 대각선 방향과 상단 우측 대각선 방향 사이의 범위 내에서 유지된다. 도 5a, 도 5b 및 도 5c 는 대각선 방향 범위 외의 인트라 예측 모드들에 대한 모드 매핑 프로세스의 예시들이다. 도 5a 는 각도 모드 리매핑을 필요로 하지 않는 정사각형 블록을 도시한다. 도 5b 는 수평 비정사각형 블록에 대한 각도 모드 리매핑을 도시한다. 도 5c 는 수직 비정사각형 블록에 대한 각도 모드 리매핑을 도시한다.

[0071] 추가로, 상부 참조 로우 및 좌측 참조 컬럼에서의 참조 샘플들의 수는 모든 블록 형상들에 대해 2 * 폭 + 1 및

2 * 높이 + 1 로 제한된다. VTM3 에 채택되는 더 넓은 각도의 예시는 도 6 에 제공된다. 도 6 은 총 93 개의 각도 모드들에 대해 모드들 2 및 66 을 넘어 VVC 테스트 모델 3 (VTM3) 에서 넓은 각도들 (-1 내지 -14, 및 67 내지 80) 을 도시하는 개념적 다이어그램이다. VTM3 은 임의의 블록 사이즈에 대해 95 개 모드들을 정의하지만, 단지 67 개의 모드들만이 허용된다. 허용되는 정확한 모드는 블록 폭 대 높이의 비율에 의존한다. 이는 소정의 블록 사이즈들에 대해 모드 범위를 제한하는 것으로 행해진다.

[0072] 도 7 에 나타낸 표 1 은 VTM3 (JVET-L1001) 에서 각도 파라미터 (intraPredAngle) 와 인트라 예측 모드 (predModeIntra) 사이의 매핑 표를 특정한다. 비정사각형 블록 대각선과 대응하는 각도 모드들은 밑줄로 나타낸다. 수직 및 수평 모드들은 이탤릭체로 나타낸다. 정사각형 블록 대각선 모드들은 밑줄과 이탤릭체로 나타낸다. 블록에 대한 대각선 모드들은 각각 블록의 하단 우측 코너들과 상단 좌측 또는 블록의 상단 우측 코너들과 하부 좌측 사이의 라인에 대한 각도와 동일한 각도를 갖는다. 이러한 개시에서, 양의 intraPredAngle 값을 갖는 각도 모드들은 양의 각도 모드들 (모드 인덱스 <18 또는 >50) 로서 지칭되는 한편, 음의 intraPredAngle 값을 갖는 각도 모드들은 음의 각도 모드들 (모드 인덱스 >18 및 <50) 로서 지칭된다.

[0073] 역 각도 파라미터 (invAngle) 는 다음과 같이 intraPredAngle 에 기초하여 도출된다:

[0074] $$\mathrm{invAngle} = \mathrm{Round}\left(\frac{256*32}{\mathrm{intraPredAngle}}\right) \qquad (2\text{-}1)$$

[0075] 32 의 배수들인 intraPredAngle 값들 (0, 32, 64, 128, 256, 512) 은 VTM3 사양에서의 경우와 같이, 비-분수의 참조 어레이 샘플들로부터의 예측과 대응함을 유의한다.

[0076] 하기의 표 2 는 다양한 종횡비들을 갖는 블록들에 대한 대각선 모드들 (즉, 대각선 인트라 예측 모드들) 을 나타낸다.

[0077] *표 2: 다양한 블록 종횡비들과 대응하는 대각선 모드들.*

| **블록 종횡비 (폭/높이)** | **대각선 모드들** |
|---|---|
| 1 **(정사각형)** | 2, 34, 66 |
| 2 | 8, 28, 72 |
| 4 | 12, 24, 76 |
| 8 | 14, 22, 78 |
| 16 | 16, 20, 80 |
| 1/2 | -6, 40, 60 |
| 1/4 | -10, 44, 56 |
| 1/8 | -12, 46, 54 |
| 1/16 | -14, 48, 52 |

[0078]

[0079] 코딩 블록의 이웃에 있는 샘플들은 블록의 인트라 예측을 위해 사용된다. 통상적으로, 코딩 블록의 좌측 및 상단 경계들에 가장 가까운 복원된 참조 샘플 라인들은 인트라 예측에 대한 참조 샘플들로서 사용된다. 그러나, VVC WD4 는 또한 코딩 블록의 이웃에 있는 다른 샘플들이 참조 샘플들로서 사용되는 것을 가능하게 한다. 도 8 은 인트라 예측을 위해 사용될 수도 있는 참조 샘플 라인들을 도시한다. 도 8 의 예에서, 참조 샘플 라인들은 다중 참조 라인 인덱스 (MRLIdx) 를 사용하여 나타낸다. 각각의 코딩 블록에 대해, 사용되는 참조 라인을 표시하는 인덱스가 시그널링된다.

[0080] 시그널링된 인덱스의 값은 시그널링된 인덱스에 의해 표시된 MRLIdx 와 상이할 수도 있다. 예를 들어, VVC WD4 에서는, MRLIdx 가 0, 1, 3 인 참조 라인들만이 사용될 수 있다. VVC WD4 에서, 블록을 코딩하기 위해 사용된 참조 라인에 시그널링된 인덱스 (MRLIdx 0, 1 및 3 각각을 갖는 라인들을 표시하는 값들 0, 1) 는 절단된 단항 코드워드로 코딩된다. MRLIdx > 0일 때 평면 및 DC 모드들은 사용되지 않는다. 최신 버전의 VVC 에서는, 0, 1, 2 인 MRLIdx 가 사용될 수 있으며, MRLIdx > 0 일 때 DC 모드들이 사용될 수도 있다.

[0081] 인트라 예측 모드들을 시그널링하는 프로세스를 보다 효율적으로 행하기 위해, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 하나 이상의 인트라 예측 모드를 블록에 대한 "최확 모드" (MPM) 들인 것으로 식별할 수도 있

다. 블록의 인트라 예측 모드가 블록에 대한 MPM들 중 하나인 경우, MPM 에 대한 인덱스가 시그널링된다.

블록의 인트라 예측 모드가 MPM들 중 하나가 아닌 경우, 블록에 대한 비-MPM 인트라 예측 모드들 중 하나에 대한 인덱스가 시그널링될 수도 있다. 통상적으로 비-MPM 인트라 예측 모드들보다 블록에 대한 MPM들이 훨씬 더 적기 때문에, MPM 에 대한 인덱스는 비-MPM 인트라 예측 모드에 대한 인덱스보다 더 적은 비트를 사용하여 시그널링될 수도 있다. 블록에 대한 MPM들을 결정하는 인트라 예측 모드를 결정하기 위한 프로세스의 예에서 VVC WD4 로부터의 다음의 텍스트.

[0082] 이 프로세스에 대한 입력은:

[0083] 현재 픽처의 상단 좌측 샘플에 대한 현재 루마 코딩 블록의 상단 좌측 샘플을 특정하는 루마 위치 (xCb, yCb),

[0084] 루마 샘플들에서 현재 코딩 블록의 폭을 특정하는 변수 cbWidth,

[0085] 루마 샘플들에서 현재 코딩 블록의 높이를 특정하는 변수 cbHeight,

[0086] 이 프로세스에서, 루마 인트라 예측 모드 IntraPredModeY[ xCb ][ yCb ]가 도출된다.

[0087] 표 8-1 은 인트라 예측 모드 IntraPredModeY[ xCb ][ yCb ] 에 대한 값 및 연관된 명칭들을 특정한다.

[0088] 표 8-1 - 인트라 예측 모드 및 연관된 명칭들의 지정

| 인트라 예측 모드 | 연관된 명칭 |
|---|---|
| 0 | INTRA_PLANAR |
| 1 | INTRA_DC |
| 2..66 | INTRA_ANGULAR2..INTRA_ANGULAR66 |
| 81..83 | INTRA_LT_CCLM, INTRA_L_CCLM, INTRA_T_CCLM |

[0089]

[0090] 비고: 인트라 예측 모드들 INTRA_LT_CCLM, INTRA_L_CCLM 및 INTRA_T_CCLM 은 크로마 성분들에만 적용가능하다.

[0091] IntraPredModeY[ xCb ][ yCb ] 는 다음의 순서화된 단계들로 도출된다:

[0092] 이웃 위치들 ( xNbA, yNbA ) 및 ( xNbB, yNbB ) 은 각각 ( xCb - 1, yCb + cbHeight - 1 ) 및 ( xCb + cbWidth - 1, yCb - 1 ) 과 동일하게 설정된다.

[0093] X 가 A 또는 B 로 대체되는 것에 대해, 변수들 candIntraPredModeX 는 다음과 같이 도출된다:

[0094] 조항 6.4.X 에 특정된 바와 같은 블록에 대한 가용성 도출 프로세스 [Ed. (BB): 이웃 블록 가용성 체크 프로세스 tbd] 는 입력들로서 ( xCb, yCb ) 와 동일하게 설정된 위치( xCurr, yCurr ) 및 ( xNbX, yNbX ) 동일하게 설정된 이웃 위치( xNbY, yNbY ) 로 호출되고, 출력들이 이용가능한X 에 할당된다.

[0095] 후보 인트라 예측 모드 candIntraPredModeX 는 다음과 같이 도출된다:

[0096] 다음 조건들 중 하나 이상이 참이면, candIntraPredModeX 는 INTRA_PLANAR 와 동일하게 설정된다.

[0097] 변수 availableX 는 FALSE 이다.

[0098] CuPredMode[ xNbX ][ yNbX ]는 MODE_INTRA 와 동일하지 않고 ciip_flag[ xNbX ][ yNbX ] 는 1 이 아니다.

[0099] pcm_flag[ xNbX ][ yNbX ]는 1 이다.

[0100] X 는 B 와 같고 yCb 1 은 ( ( yCb >> CtbLog2SizeY ) << CtbLog2SizeY ) 보다 작다.

[0101] 그렇지 않으면, candIntraPredModeX 는 IntraPredModeY[ xNbX ][ yNbX ] 와 동일하게 설정된다.

[0102] 변수들 ispDefaultMode1 및 ispDefaultMode2 는 다음과 같이 정의된다:

[0103] IntraSubPartitionsSplitType 가 ISP_HOR_SPLIT 와 동일하면, ispDefaultMode1 는 INTRA_ANGULAR18 와 동일하게 설정되고 ispDefaultMode2 는 INTRA_ANGULAR5 와 동일하게 설정된다.

[0104] 그렇지 않으면, ispDefaultMode1 이 INTRA_ANGULAR50 와 동일하게 설정되고 ispDefaultMode2 는 INTRA_ANGULAR63 와 동일하게 설정된다.

[0105] x = 0..5 인 candModeList[ x ]는 다음과 같이 도출된다:

[0106] candIntraPredModeB 가 candIntraPredModeA 와 동일하고 candIntraPredModeA 가 INTRA_DC 보다 크면, x = 0..5 인 candModeList[ x ] 가 다음과 같이 도출된다:

[0107] IntraLumaRefLineIdx[ xCb ][ yCb ] 가 0 이고 IntraSubPartitionsSplitType 가 ISP_NO_SPLIT 와 동일하면, 다음이 적용된다:

[0108]
$$candModeList[\,0\,] = candIntraPredModeA \quad (8\text{-}8)$$

$$candModeList[\,1\,] = INTRA_PLANAR \quad (8\text{-}9)$$

$$candModeList[\,2\,] = INTRA_DC \quad (8\text{-}10)$$

$$candModeList[\,3\,] = 2 + (\,(\,candIntraPredModeA + 61\,)\,\%\,64\,) \quad (8\text{-}11)$$

$$candModeList[\,4\,] = 2 + (\,(\,candIntraPredModeA - 1\,)\,\%\,64\,) \quad (8\text{-}12)$$

$$candModeList[\,5\,] = 2 + (\,(\,candIntraPredModeA + 60\,)\,\%\,64\,) \quad (8\text{-}13)$$

[0109]

[0110] 그렇지 않으면 (IntraLumaRefLineIdx[ xCb ][ yCb ] 이 0 이 아니거나 IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 와 동일하지 않으면), 다음이 적용된다:

$$candModeList[\,0\,] = candIntraPredModeA \quad (8\text{-}14)$$

$$candModeList[\,1\,] = 2 + (\,(\,candIntraPredModeA + 61\,)\,\%\,64\,) \quad (8\text{-}15)$$

$$candModeList[\,2\,] = 2 + (\,(\,candIntraPredModeA - 1\,)\,\%\,64\,) \quad (8\text{-}16)$$

[0111]

[0112] 다음의 조건들 중 하나가 참이면,

[0113] IntraSubPartitionsSplitType 는 ISP_HOR_SPLIT 와 동일하고 candIntraPredModeA 는 INTRA_ANGULAR34 보다 작고,

[0114] IntraSubPartitionsSplitType 이 ISP_VER_SPLIT 와 동일하고 candIntraPredModeA 가 INTRA_ANGULAR34 보다 크거나 같고,

[0115] IntraLumaRefLineIdx[ xCb ][ yCb ] 가 0 이 아니며,

[0116] 다음이 적용된다:

$$candModeList[\,3\,] = 2 + (\,(\,candIntraPredModeA + 60\,)\,\%\,64\,) \quad (8\text{-}17)$$

$$candModeList[\,4\,] = 2 + (\,candIntraPredModeA\,\%\,64\,) \quad (8\text{-}18)$$

$$candModeList[\,5\,] = 2 + (\,(\,candIntraPredModeA + 59\,)\,\%\,64\,) \quad (8\text{-}19)$$

Otherwise, the following applies:

$$candModeList[\,3\,] = ispDefaultMode1 \quad (8\text{-}20)$$

$$candModeList[\,4\,] = ispDefaultMode2 \quad (8\text{-}21)$$

$$candModeList[\,5\,] = INTRA_PLANAR \quad (8\text{-}22)$$

[0117]

[0118] 그렇지 않고 candIntraPredModeB 가 candIntraPredModeA 와 동일하지 않고 candIntraPredModeA 또는 candIntraPredModeB 가 INTRA_DC 보다 크면, 다음이 적용된다:

[0119] 변수들 minAB 및 maxAB 는 다음과 같이 도출된다:

[0120]
$$minAB = Min(\,candIntraPredModeA, candIntraPredModeB\,) \quad (8\text{-}23)$$

$$maxAB = Max(\,candIntraPredModeA, candIntraPredModeB\,) \quad (8\text{-}24)$$

[0121] candIntraPredModeA 및 candIntraPredModeB 양자 모두가 INTRA_DC 보다 크면, candModeList[ x ] (x = 0..5) 가 다음과 같이 도출된다:

candModeList[ 0 ] = candIntraPredModeA (8-25)

candModeList[ 1 ] = candIntraPredModeB (8-26)

[0122]

[0123] IntraLumaRefLineIdx[ xCb ][ yCb ] 가 0 이고 IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 와 동일하면, 다음이 적용된다:

candModeList[ 2 ] = INTRA_PLANAR (8-27)

candModeList[ 3 ] = INTRA_DC (8-28)

[0124]

[0125] maxAB minAB 이 2 내지 62 포함 범위에 있으면, 다음이 적용된다:

candModeList[ 4 ] = 2 + ( ( maxAB + 61 ) % 64 ) (8-29)

candModeList[ 5 ] = 2 + ( ( maxAB − 1 ) % 64 ) (8-30)

[0126]

[0127] 그렇지 않으면, 다음이 적용된다:

candModeList[ 4 ] = 2 + ( ( maxAB + 60 ) % 64 ) (8-31)

[0128] candModeList[ 5 ] = 2 + ( ( maxAB ) % 64 ) (8-32)

[0129] 그렇지 않으면 (IntraLumaRefLineIdx[ xCb ][ yCb ] 이 0 이 아니거나 IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 와 동일하지 않으면), 다음이 적용된다:

[0130] IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 와 동일하지 않고, abs( candIntraPredModeB ispDefaultMode1 ) 가 abs( candIntraPredModeA ispDefaultMode1 ) 보다 작을 때, 다음이 적용된다:

candModeList[ 0 ] = candIntraPredModeB (8-33)

[0131] candModeList[ 1 ] = candIntraPredModeA (8-34)

[0132] maxAB minAB 이 1 과 동일하면, 다음이 적용된다:

candModeList[ 2 ] = 2 + ( ( minAB + 61 ) % 64 ) (8-35)

candModeList[ 3 ] = 2 + ( ( maxAB − 1 ) % 64 ) (8-36)

candModeList[ 4 ] = 2 + ( ( minAB + 60 ) % 64 ) (8-37)

[0133] candModeList[ 5 ] = 2 + ( maxAB % 64 ) (8-38)

[0134] 그렇지 않고 maxAB minAB 이 2 와 동일하면, 다음이 적용된다:

candModeList[ 2 ] = 2 + ( ( minAB − 1 ) % 64 ) (8-39)

candModeList[ 3 ] = 2 + ( ( minAB + 61 ) % 64 ) (8-40)

candModeList[ 4 ] = 2 + ( ( maxAB − 1 ) % 64 ) (8-41)

[0135] candModeList[ 5 ] = 2 + ( ( minAB + 60 ) % 64 ) (8-42)

[0136] 그렇지 않고 maxAB minAB 이 61 보다 크면, 다음이 적용된다:

candModeList[ 2 ] = 2 + ( ( minAB − 1 ) % 64 ) (8-43)

candModeList[ 3 ] = 2 + ( ( maxAB + 61 ) % 64 ) (8-44)

candModeList[ 4 ] = 2 + ( minAB % 64 ) (8-45)

[0137] candModeList[ 5 ] = 2 + ( ( maxAB + 60 ) % 64 ) (8-46)

[0138] 그렇지 않으면, 다음이 적용된다:

[0139]
$$candModeList[\ 2\ ] = 2 + (\ (\ minAB + 61\ )\ \%\ 64\ ) \quad (8\text{-}47)$$

$$candModeList[\ 3\ ] = 2 + (\ (\ minAB - 1\ )\ \%\ 64\ ) \quad (8\text{-}48)$$

$$candModeList[\ 4\ ] = 2 + (\ (\ maxAB + 61\ )\ \%\ 64\ ) \quad (8\text{-}49)$$

$$candModeList[\ 5\ ] = 2 + (\ (\ maxAB - 1\ )\ \%\ 64\ ) \quad (8\text{-}50)$$

[0140]

[0141] 그렇지 않으면 (candIntraPredModeA 또는 candIntraPredModeB 이 INTRA_DC 보다 크면), candModeList[ x ] (x = 0..5) 가 다음과 같이 도출된다:

[0142] IntraLumaRefLineIdx[ xCb ][ yCb ] 이 0 이고 IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 와 동일하면, 다음이 적용된다:

$$candModeList[\ 0\ ] = candIntraPredModeA \quad (8\text{-}51)$$

$$candModeList[\ 1\ ] = candIntraPredModeB \quad (8\text{-}52)$$

$$candModeList[\ 2\ ] = 1 - minAB \quad (8\text{-}53)$$

$$candModeList[\ 3\ ] = 2 + (\ (\ maxAB + 61\ )\ \%\ 64\ ) \quad (8\text{-}54)$$

$$candModeList[\ 4\ ] = 2 + (\ (\ maxAB - 1\ )\ \%\ 64\ ) \quad (8\text{-}55)$$

[0143]
$$candModeList[\ 5\ ] = 2 + (\ (\ maxAB + 60\ )\ \%\ 64\ ) \quad (8\text{-}56)$$

[0144] 그렇지 않고, IntraLumaRefLineIdx[ xCb ][ yCb ] 이 0 이 아니면, 다음이 적용된다:

$$candModeList[\ 0\ ] = maxAB \quad (8\text{-}57)$$

$$candModeList[\ 1\ ] = 2 + (\ (\ maxAB + 61\ )\ \%\ 64\ ) \quad (8\text{-}58)$$

$$candModeList[\ 2\ ] = 2 + (\ (\ maxAB - 1\ )\ \%\ 64\ ) \quad (8\text{-}59)$$

$$candModeList[\ 3\ ] = 2 + (\ (\ maxAB + 60\ )\ \%\ 64\ ) \quad (8\text{-}60)$$

$$candModeList[\ 4\ ] = 2 + (\ maxAB\ \%\ 64\ ) \quad (8\text{-}61)$$

[0145]
$$candModeList[\ 5\ ] = 2 + (\ (\ maxAB + 59\ )\ \%\ 64\ ) \quad (8\text{-}62)$$

[0146] 그렇지 않으면 (IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 와 동일하지 않으면), 다음이 적용된다:

$$candModeList[\ 0\ ] = INTRA_PLANAR \quad (8\text{-}63)$$

$$candModeList[\ 1\ ] = maxAB \quad (8\text{-}64)$$

$$candModeList[\ 2\ ] = 2 + (\ (\ maxAB + 61\ )\ \%\ 64\ ) \quad (8\text{-}65)$$

$$candModeList[\ 3\ ] = 2 + (\ (\ maxAB - 1\ )\ \%\ 64\ ) \quad (8\text{-}66)$$

$$candModeList[\ 4\ ] = 2 + (\ (\ maxAB + 60\ )\ \%\ 64\ ) \quad (8\text{-}67)$$

[0147]
$$candModeList[\ 5\ ] = 2 + (\ maxAB\ \%\ 64\ ) \quad (8\text{-}68)$$

[0148] 그렇지 않으면, 다음이 적용된다:

[0149] IntraLumaRefLineIdx[ xCb ][ yCb ] 이 0 이고 IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 와 동일하면, 다음이 적용된다:

[0150]
$$candModeList[\ 0\ ] = candIntraPredModeA \quad (8\text{-}69)$$

candModeList[ 1 ] =
( candModeList[0] == INTRA_PLANAR ) ? INTRA_DC : (8-70)
INTRA_PLANAR
candModeList[ 2 ] = INTRA_ANGULAR50(8-71)
candModeList[ 3 ] = INTRA_ANGULAR18(8-72)
candModeList[ 4 ] = INTRA_ANGULAR46(8-73)

[0151] candModeList[ 5 ] = INTRA_ANGULAR54(8-74)

[0152] 그렇지 않고, IntraLumaRefLineIdx[ xCb ][ yCb ] 이 0 이 아니면, 다음이 적용된다:

candModeList[ 0 ] = INTRA_ANGULAR50(8-75)
candModeList[ 1 ] = INTRA_ANGULAR18(8-76)
candModeList[ 2 ] = INTRA_ANGULAR2 (8-77)
candModeList[ 3 ] = INTRA_ANGULAR34(8-78)
candModeList[ 4 ] = INTRA_ANGULAR66(8-79)

[0153] candModeList[ 5 ] = INTRA_ANGULAR26(8-80)

[0154] 그렇지 않고, IntraSubPartitionsSplitType 이 ISP_HOR_SPLIT 와 동일하면, 다음이 적용된다:

candModeList[ 0 ] = INTRA_PLANAR (8-81)
candModeList[ 1 ] = INTRA_ANGULAR18(8-82)
candModeList[ 2 ] = INTRA_ANGULAR25(8-83)
candModeList[ 3 ] = INTRA_ANGULAR10(8-84)
candModeList[ 4 ] = INTRA_ANGULAR65(8-85)

[0155] candModeList[ 5 ] = INTRA_ANGULAR50(8-86)

[0156] 그렇지 않고, IntraSubPartitionsSplitType 이 ISP_VER_SPLIT 와 동일하면, 다음이 적용된다:

candModeList[ 0 ] = INTRA_PLANAR (8-87)
candModeList[ 1 ] = INTRA_ANGULAR50(8-88)
candModeList[ 2 ] = INTRA_ANGULAR43(8-89)
candModeList[ 3 ] = INTRA_ANGULAR60(8-90)
candModeList[ 4 ] = INTRA_ANGULAR3 (8-91)

[0157] candModeList[ 5 ] = INTRA_ANGULAR18(8-92)

[0158] IntraPredModeY[ xCb ][ yCb ] 는 다음의 절차를 적용함으로써 도출된다:

[0159] intra_luma_mpm_flag[ xCb ][ yCb ] 이 1 이면, IntraPredModeY[ xCb ][ yCb ] 는 candModeList[ intra_luma_mpm_idx[ xCb ][ yCb ] ] 와 동일하게 설정된다.

[0160] 그렇지 않으면, IntraPredModeY[ xCb ][ yCb ] 가 다음의 순서화된 단계들을 적용함으로써 도출된다:

[0161] i = 0..4 에 대해 candModeList[ i ] 가 candModeList[ j ] 보다 크고 각각의 i 에 대해, j = ( i + 1 )..5 일 때, 양자의 값들은 다음과 같이 스와핑된다:

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

[0162] candModeList[ j ] ) (8-93)

[0163] IntraPredModeY[ xCb ][ yCb ] 는 다음의 순서화된 단계들에 의해 도출된다:

[0164] IntraPredModeY[ xCb ][ yCb ] 는 intra_luma_mpm_remainder[ xCb ][ yCb ] 와 동일하게 설정된다.

[0165] i 가 0 내지 5 인 경우, IntraPredModeY[ xCb ][ yCb ] 가 candModeList[ i ] 보다 크거나 같을 때, IntraPredModeY[ xCb ][ yCb ] 의 값은 1 씩 증분된다.

[0166] 변수 IntraPredModeY[ x ][ y ](x = xCb..xCb + cbWidth - 1 및 y = yCb..yCb + cbHeight - 1) 은 IntraPredModeY[ xCb ][ yCb ] 와 동일하게 설정된다.

[0167] 인트라 서브-파티션 코딩 (ISP)(S. De Luxαn Hernαndez, V. George, J. Ma, T. Nguyen, H. Schwarz, D. Marpe, T. Wiegand (HHI), "CE3: Intra Sub-Partitions Coding Mode," JVET-M0102 참조) 은 코딩 블록이 2 또는 4 개의 서브블록들로 분할되는 방법이다. 블록 내의 각각의 서브블록은 디코딩 순서로 후속 서브블록의 복원 전에 디코딩 순서로 복원된다. VVC WD4 에서, ISP 는 루마 코딩 블록들에만 적용된다. ISP-코딩된 블록들에 대한 참조 샘플들은 코딩 블록에 대해 가장 가까운 참조 라인으로부터인 것으로 제한된다 (예를 들어, 도 8 에 나타낸 바와 같이 MRLIdx = 0). 스플릿의 예들은 도 9 및 도 10 에 나타낸다.

[0168] 일 비트는 코딩 블록이 ISP들로 스플릿되는지 여부를 시그널링하는데 사용되고, 제 2 비트는 스플릿 타입: 수평 또는 수직을 표시하는데 사용된다. 사용된 스플릿 타입 및 인트라 모드에 기초하여, 프로세싱 순서들의 2 개의 상이한 클래스들이 사용될 수도 있으며, 이는 정상 순서 및 반전된 순서로서 지칭된다. 정상 순서에서, 프로세싱될 제 1 서브-파티션은 CU 의 상단 좌측 샘플을 포함하고 그 후 하향방향 (수평 스플릿) 또는 우측방향 (수직 스플릿) 으로 계속하는 것이다. 다른 한편으로, 역 프로세스 순서는 CU 의 하단 좌측 샘플을 포함하는 서브-파티션으로 시작하고 (수평 스플릿) 상향으로 계속하거나 CU 의 상단 우측 샘플을 포함하는 서브-파티션으로 시작하여 좌측방향으로 계속한다 (수직 스플릿).

[0169] 정상 프로세싱 순서만을 사용하는 ISP 의 변형은 JVET WD4 에서 사용된다. 이 문서에서 용어 서브블록 및 서브-파티션은 상호교환가능하게 사용되고, 양자는 모두 ISP 를 사용하여 코딩 블록을 파티셔닝하는 것에 의해 획득된 블록들을 지칭한다.

[0170] JVET WD4 에서 ISP 와 연관된 일부 신택스 및 시맨틱스는 하기에 나타나 있으며, 관련 신택스는 태그들 <!> … </!> 로 둘러싸여 있다.

[0171] **코딩 유닛의 신택스 표**

| coding_unit( x0, y0, cbWidth, cbHeight, treeType ) { | 디스크립터 |
|---|---|
| … | |
| } else { | |
| if( treeType = = SINGLE_TREE \|\| treeType = = DUAL_TREE_LUMA ) { | |
| if( ( y0 % CtbSizeY ) > 0 ) | |
| **intra_luma_ref_idx**[ x0 ][ y0 ] | ae(v) |
| <!> if (intra_luma_ref_idx[ x0 ][ y0 ] = = 0 && ( cbWidth <= MaxTbSizeY \|\| cbHeight <= MaxTbSizeY ) && ( cbWidth * cbHeight > MinTbSizeY * MinTbSizeY )) | |
| **intra_subpartitions_mode_flag**[ x0 ][ y0 ] | ae(v) |
| if( intra_subpartitions_mode_flag[ x0 ][ y0 ] = = 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY ) | |
| **intra_subpartitions_split_flag**[ x0 ][ y0 ] | ae(v)</!> |
| 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 ] ) | |

[0172]

| **intra_luma_mpm_idx**[ x0 ][ y0 ] | ae(v) |
|---|---|
| … | |

[0173]

[0174] **변환 트리의 신택스 테이블**

| transform_tree( x0, y0, tbWidth, tbHeight , treeType) { | 디스크립터 |
|---|---|
| <!> InferTuCbfLuma = 1 | |
| if( IntraSubPartSplitType = = NO_ISP_SPLIT ) {</!> | |
| if( tbWidth > MaxTbSizeY \|\| tbHeight > MaxTbSizeY ) { | |
| trafoWidth = ( tbWidth > MaxTbSizeY ) ? (tbWidth / 2) : tbWidth | |
| trafoHeight = ( tbHeight > MaxTbSizeY ) ? (tbHeight / 2) : tbHeight | |
| transform_tree( x0, y0, trafoWidth, trafoHeight ) | |
| if( tbWidth > MaxTbSizeY ) | |
| transform_tree( x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType ) | |
| if( tbHeight > MaxTbSizeY ) | |
| transform_tree( x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType ) | |
| if( tbWidth > MaxTbSizeY && tbHeight > MaxTbSizeY ) | |
| transform_tree( x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType ) | |
| } else { | |
| transform_unit( x0, y0, tbWidth, tbHeight, treeType, 0 ) | |
| } | |
| <!> } else if( IntraSubPartitionsSplitType = = ISP_HOR_SPLIT ) { | |

[0175]

| | |
|---|---|
| trafoHeight = tbHeight / NumIntraSubPartitions | |
| for( partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++ ) | |
| transform_unit( x0, y0 + trafoHeight * partIdx, tbWidth, trafoHeight, treeType, partIdx ) | |
| } else if( IntraSubPartitionsSplitType = = ISP_VER_SPLIT ) { | |
| trafoWidth = tbWidth / NumIntraSubPartitions | |
| for( partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++ ) | |
| transform_unit( x0 + trafoWidth * partIdx, y0, trafoWidth, tbHeight, treeType, partIdx ) | |
| }</!> | |
| } | |

[0176]

[0177] **코딩 유닛의 시맨틱스**

[0178] <!> **1 과 동일한 intra_subpartitions_mode_flag**[ x0 ][ y0 ] 는 현재 인트라 코딩 유닛이 NumIntraSubPartitions[ x0 ][ y0 ] 직사각형 변환 블록 서브파티션들로 파티셔닝되는 것을 특정한다. 0 과 동일한 intra_subpartitions_mode_flag[ x0 ][ y0 ] 는 현재 인트라 코딩 유닛이 직사각형 변환 블록 서브파티션들로 파티셔닝되지 않는 것을 특정한다. intra_subpartitions_mode_flag[ x0 ][ y0 ] 이 존재하지 않을 때, 이는 0 과 동일한 것으로 추론된다.

[0179] **intra_subpartitions_split_flag**[ x0 ][ y0 ] 는 인트라 서브파티션 스플릿 타입이 수평인지 또는 수직인지를 특정한다. intra_subpartitions_mode_flag[ x0 ][ y0 ] 이 존재하지 않을 때, 이는 0 과 동일한 것으로 추론된다. 변수 IntraSubPartitionsSplitType 는 표 2-3 에 예시된 바와 같이 현재 루마 코딩 블록을 위해 사용된 스플릿의 타입을 특정한다. IntraSubPartitionsSplitType 은 다음과 같이 도출된다:

[0180] - intra_subpartitions_mode_flag[ x0 ][ y0 ] 가 0 과 동일하면, IntraSubPartitionsSplitType 이 0 과 동일하게 설정된다.

[0181] - 그렇지 않으면, IntraSubPartitionsSplitType 은 1 + intra_subpartitions_split_flag[ x0 ][ y0 ] 과 동일하게 설정된다.

[0182] **표 2-3 - IntraSubPartitionsSplitType 에 대한 명칭 연관성**

| IntraSubPartitionsSplitType | IntraSubPartitionsSplitType 의 명칭 |
|---|---|
| 0 | ISP_NO_SPLIT |
| 1 | ISP_HOR_SPLIT |
| 2 | ISP_VER_SPLIT |

[0183]

[0184] 변수 NumIntraSubPartitions 은 인트라 루마 코딩 블록이 분할되는 변환 블록 서브파티션들의 수를 특정한다. NumIntraSubPartitions 은 다음과 같이 도출된다:

[0185] - IntraSubPartitionsSplitType 이 ISP_NO_SPLIT 과 동일하면, NumIntraSubPartitions 이 1 과 동일하게 설정된다.

[0186] - 그렇지 않고, 다음 조건들 중 하나가 참이면, NumIntraSubPartitions 은 2 와 동일하게 설정된다:

[0187] - cbWidth 는 4 와 동일하고 cbHeight 는 8 과 동일하다.

[0188] - cbWidth 는 8 와 동일하고 cbHeight 는 4 와 동일하다.

[0189] - 그렇지 않으면, NumIntraSubPartitions 은 4 와 동일하게 설정된다.</!>

[0190] 아핀 선형 가중 인트라 예측 (ALWIP)( 예를 들어, J. Pfaff 등의 "Affine linear weighted intra prediction (CE3-4.1, CE3-4.2)", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 JVET (Joint Video Experts Team), 14th Meeting: Geneva, CH, 19-27 March 2019, document JVET-N0217 (이하, "JVET-N0217") 에 기재된 바와 같음) 은 아핀 선형 가중 예측 모델을 사용하여 이웃 참조 샘플들로부터 블록의 예측 (즉, 예측 블록) 을 생성한다. 이웃 샘플들이 먼저 프로세싱되고 (예를 들어, 일부 경우들에서, 이웃 샘플들은 다운-샘플링됨) 프로세싱된 (예를 들어, 다운-샘플링된) 샘플들은 그 후 예측된 샘플들의 중간 다운-샘플링된 버전과 유사한 감소된 샘플들의 세트를 (아핀 모델을 사용하여) 도출하는데 사용된다. 최종 예측은 (필요한 경우) 중간 값들을 업-샘플링함으로써 획득된다. ALWIP 는 또한 매트릭스 인트라 예측 (Matrix Intra Prediction; MIP) 으로 지칭될 수도 있다.

[0191] ALWP 프로세스의 예시는 도 11 에 주어진다. 도 11 의 ALWIP 프로세스는 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해 수행될 수도 있다. 블록의 참조 샘플들 (경계 샘플들로 또한 지칭됨) 은 감소된 경계 샘플들을 획득하기 위해 다운-샘플링된다. 경계 샘플들의 벡터 표현, $bdry_{red}$ 은 매트릭스 $A_k$ 로 승산되고, 오프셋/바이어스 항 $b_k$ 는 예측된 블록 $pred_{red}$ 의 다운-샘플링된 버전을 획득하기 위해 부가된다. 최종 예측은 경계 샘플들과 함께 이러한 예측된 샘플들 $pred_{red}$ 을 업-샘플링함으로써 획득된다. 매트릭스 $A_k$ 및 오프셋/바이어스 벡터 $b_k$ 는 블록에 대해 표시된 모드 값에 기초하여 선정된다. 매트릭스 $A_k$ 및 오프셋/바이어스 벡터 $b_k$ 의 조합은 본 명세서에서 "ALWIP 모드" 로서 지칭될 수도 있다.

[0192] 중간 예측된 샘플들의 도출는 아핀 선형 가중 예측 모델을 사용한다. 3 가지 타입들 (즉, ALWIP 타입들) 이 정의되고, 도출된 중간 샘플들의 수는 다음과 같이 각각의 타입에 대해 상이하다:

[0193] 1) 폭 및 높이가 모두 4 인 블록 사이즈들에 대해 4x4. 즉, 4x4 사이즈를 갖는 블록들에 대해 16 개의 중간 샘플들이 있다.

[0194] 2) 폭 및 높이가 모두 4 인 경우 (즉, 4x8, 8x4 및 8x8 블록들) 를 제외하고 폭 및 높이가 모두 8 이하인 블록 사이즈들에 대해 8x8. 즉, $H$x$W$ 의 사이즈들을 갖는 블록들에 대해 64 개의 중간 샘플들이 있으며, 여기서 $4 < H \leq 8$ 및 $4 < W \leq 8$, 또는 $H = 4$ 및 $4 < W \leq 8$, 또는 $W = 4$ 및 $4 < H \leq 8$ 이다.

[0195] 3) 폭 및 높이 중 적어도 하나가 8 보다 큰 경우의 블록들에 대해 16x16. 즉, *$H$x$W$ 의 사이즈들을 갖는 블록들에 대해 256 개의 중간 샘플들이 있으며, $H > 8$* 이거나 $W > 8$ 이다.

[0196] 이러한 3 가지 경우들 각각에서, 상이한 수의 ALWIP 모드들이 사용된다: 각각 35, 19 및 11. 즉, ALWIP 타입 1 에 대해, 35 개의 상이한 이용가능한 ALWIP 모드들이 있고; ALWIP 타입 2 에 대해, 19 개의 상이한 이용가능한 ALWIP 모드들이 있으며; ALWIP 타입 3 에 대해, 11 개의 상이한 이용가능한 ALWIP 모드들이 있다. 상이한 ALWIP 모드들의 각각은 매트릭스 $A_k$ 및 오프셋/바이어스 값 $b_k$ 의 상이한 조합에 대응할 수도 있다.

[0197] ALWIP 의 시그널링은 다음을 포함한다:

[0198] a) 현재 블록이 ALWIP 로 코딩됨을 표시하는 플래그 (alwip_flag).

[0199] b) 현재 블록이 ALWIP 로 코딩될 때, 현재 블록이 ALWIP-최확 모드 (MPM) 모드인지 여부를 표시하기 위해 또 다른 플래그가 시그널링된다.

[0200] a. 블록이 ALWIP-MPM 모드로 코딩되는 경우, MPM 인덱스가 시그널링된다. ALWIP-MPM 모드의 도출은 하기 본 개시에서 설명된다.

[0201] b. 그 외, 나머지 모드 값에 대한 인덱스가 시그널링된다. 나머지 모드 값은 ALWIP-MPM 모드 이외의 ALWIP 모드를 표시한다.

[0202] alwip_flag 는 허용된 4 개의 컨텍스트들로 컨텍스트 코딩된다.

[0203] - 블록 폭 > 2*높이 또는 높이 > 2*폭인 경우, 컨텍스트 3 이 사용된다.

[0204] - 그 외, 컨텍스트 ctxId 가 사용되며, 여기서 ctxId 는 다음과 같이 도출된다:

[0205] o ctxId 를 0 으로 초기화한다.

[0206] o 좌측 이웃 블록이 ALWIP 로 코딩되는 경우, ctxId++ (즉, ctxId 증분).

[0207] o 상측 이웃 블록이 ALWIP 로 코딩되는 경우, ctxId++.

[0208] ALWIP MPM 모드들의 도출은 다음의 단계들을 수반한다:

[0209] 1) LeftIntraMode 및 AboveIntraMode 가 -1 로 초기화된다.

[0210] 2) 좌측 이웃 블록이 인트라 코딩되는 경우:

[0211] a. 좌측 이웃 블록이 ALWIP 모드 *L* 로 코딩되는 경우 (즉, 좌측 이웃 블록이 ALWIP 로 코딩되는 경우, 좌측 이웃 블록의 ALWIP 모드는 *L* 로 표기된다고 한다).

[0212] *i*. *L* 이 현재 블록과 동일한 ALWIP 타입의 것이면, LeftIntraMode 는 *L* 과 동일하게 설정된다.

[0213] b. 그렇지 않고, 좌측 이웃 블록이 ALWIP 모드로 코딩되지 않은 경우, 좌측 이웃 블록의 인트라 모드는 현재 블록과 동일한 타입의 ALWIP 모드에 매핑되고 LeftIntraMode 에 할당된다.

[0214] 3) 상측 이웃 블록이 인트라 코딩되는 경우:

[0215] a. 상측 이웃 블록이 ALWIP 모드*A* 로 코딩되는 경우 (즉, 상측 이웃 블록이 ALWIP 로 코딩되는 경우, 상측 이웃 블록의 ALWIP 모드는 *A* 로 표기된다고 한다).

[0216] *i*. *A* 가 현재 블록과 동일한 ALWIP 타입의 것이면, AboveIntraMode 는 *A* 와 동일하게 설정된다.

[0217] b. 그렇지 않고, 상측 이웃 블록이 ALWIP 모드로 코딩되지 않은 경우, 상측 이웃 블록의 인트라 모드는 현재 블록과 동일한 타입의 ALWIP 모드에 매핑되고 AboveIntraMode 에 할당된다.

[0218] 4) 그 후 MPM들은 LeftIntraMode 및 AboveIntraMode 에 기초하여 도출된다. 즉, 제 1 ALWIP-MPM 은 LeftIntraMode 와 동일하게 설정되고 제 2 ALWIP-MPM 은 AboveIntraMode 와 동일하게 설정된다. LeftIntraMode 가 -1 과 동일하게 유지되거나 AboveIntraMode 가 -1 과 동일하게 유지되는 경우, 대응 MPM 들은 이용불가능하다.

[0219] 문서의 나머지에서, ALWIP 로 코딩된 블록들은 ALWIP-코딩된 블록들 또는 ALWIP 블록들로 지칭될 수도 있으며; 다른 블록들 (예를 들어, 정규 인트라 예측, 인트라 서브-파티션들, 또는 다중 참조 라인들로 코딩된 블록들) 은 비-ALWIP 블록들로 지칭될 수도 있다.

[0220] ALWIP (즉, MIP) 의 설계에는 몇 가지 문제가 있다. 예를 들어, 현재 블록이 ALWIP 모드를 사용하여 코딩되는지 여부를 표시하는 플래그 (예를 들어, alwip_flag) 의 컨텍스트 도출은 좌측 및 상측 이웃 블록들로부터의 alwip_flag 값을 사용한다. 상측 이웃 블록의 alwip_flag 값은 상측 이웃 블록이 상이한 CTU 로우에 속할 때에도 사용된다. 이는 부가적인 저장 요건들을 생성한다. 예를 들어, *r* 이 현재 CTU 로우와 이전 CTU 로우 사이의 CTU 경계를 지칭하는 경우, 경계 *r* 을 공유하는 이전 CTU 로부터의 모든 블록들의 alwip_flag 는 저장되어야 한다. 4K 픽처에서 4x4 블록들의 최악의 시나리오에서는, 필요한 바이트 수가 3840/4 (블록들의 수) x 1 (플래그 값 당 비트) = 120 바이트이다. 이전 CTU 로우로부터의 모든 블록들의 alwip_flag 값들에 대한 저장을 제공하는 것은 비디오 인코더 (200) 및 비디오 디코더 (300) 의 메모리 저장 요건들을 증가시킴으로써, 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 잠재적으로 증가시킨다.

[0221] ALWIP 설계에 의한 문제의 또 다른 예에서는, 4x4 블록에 대해 특정된 35 개의 매트릭스들/모드들이 있다. 이러한 35 개의 매트릭스들/모드들은 ALWIP 에 필요한 파라미터들의 거의 25% 를 나타낸다. ALWIP 4x4 블록들을 인에이블할 때의 압축 효율성은 거의 0.05 % 이고 저장은 (35 개의 매트릭스들 및 오프셋/바이어스 벡터들을 저장하기 위해) 약 2KB 를 필요로 한다. 따라서, 인코더/디코더 구현들에 대한 저장 부담은 획득된 효율성에 비해 과도하다. 도 12 는 VVC WD4 에서 각각의 ALWIP 타입에 대한 오프셋 파라미터들 및 매트릭스 파라미터들의 수를 도시하는 표이다.

[0222] 비-ALWIP 블록들의 최확 모드 (MPM) 도출 동안, 좌측 및 상측 이웃 블록들에 사용된 인트라 모드가 체크되고 이용가능할 때 사용된다. 예를 들어, ALWIP MPM 모드들을 도출할 때, 비디오 코더 (예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300)) 는 좌측 이웃 블록이 이용가능한지 여부를 결정하고, 이용가능한 경우, 비디오 코더는 좌측 이웃 블록이 인트라 코딩되는지 여부를 결정하며, 인트라 코딩되는 경우, 비디오 코더는 좌측 이웃 블록이 ALWIP 모드를 사용하여 코딩되는지 여부를 결정한다.. 유사하게, 비디오 코더는 또한 상측 이웃 블록이 이용가능한지 여부를 결정하고, 이용가능한 경우, 비디오 코더는 상측 이웃 블록이 인트라 코딩되는지 여

부를 결정하며, 이용가능한 경우, 비디오 코더는 상측 이웃 블록이 ALWIP 모드를 사용하여 코딩되는지 여부를 결정한다. 비-ALWIP 블록들 (즉, ALWIP 모드를 사용하여 코딩되지 않은 블록들) 에 대해, 모드 값들은 각도 인트라 예측 모드들 또는 평면/DC 인트라 예측 모드들에 대응한다. 사소한 경우들을 제외하고, ALWIP 에 사용된 매트릭스 계수들의 일 세트 (예를 들어, ALWIP 모드에 대한 매트릭스 계수들의 세트) 는 비-ALWIP 블록들에서 사용된 인트라 모드들 중 어느 것에도 대응하지 않을 수도 있다. ALWIP 는 현재 제 1 매핑 표을 사용하여 인트라 모드들 중 하나에 ALWIP 매트릭스의 매핑을 근사화한다. 이 근사 값은 그 후 비-ALWIP 블록의 MPM 도출에 사용될 이웃 블록의 인트라 모드로 간주된다. 즉, 좌측 이웃 블록이 비-ALWIP 인트라 예측 모드를 사용하여 코딩될 때, 비디오 코더 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더 (300)) 는 제 1 ALWIP-MPM 을 좌측 이웃 블록에 대한 근사 값으로 설정한다. 유사하게, 상측 이웃 블록이 비-ALWIP 인트라 예측 모드를 사용하여 코딩될 때, 비디오 코더는 제 2 ALWIP-MPM 을 상측 이웃 블록에 대한 근사 값으로 설정한다. 이는 예를 들어 근사치의 에러 때문에, 비효율적인 압축을 초래할 수도 있다. 또한, 매핑 표는 부가 스토리지를 필요로 하고 MPM 도출 프로세스에서 룩업 동작들을 부가한다. 부가 스토리지는 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 부가한다. 부가 룩업 동작들은 인코딩 및 디코딩 프로세스의 속도를 늦출 수도 있다.

[0223] 이 문제는 크로마 다이렉트 모드 (DM) 모드에 또한 적용될 수도 있으며, 여기서 크로마 블록에 사용된 인트라 모드는 병치된 인트라 블록으로부터 카피된다. 병치된 인트라 블록이 ALWIP 코딩될 때, 제 1 매핑 표는은 ALWIP 블록 매트릭스/모드로부터 크로마 DM 모드를 도출하는데 사용된다. 즉, 크로마 블록과 함께 병치된 루마 블록은 ALWIP 모드를 사용하여 코딩될 수도 있지만 크로마 블록은 임의의 ALWIP 모드를 사용하여 코딩될 수 없다. 따라서, 크로마 블록이 루마 블록의 인트라 예측 모드가 크로마 블록에 의해 상속되는 크로마 DM 모드를 사용하여 코딩되어야 하는 경우, 비디오 코더 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더 (300)) 가 루마 블록의 ALWIP 모드에서 크로마 블록에서 사용하기 위한 비- ALWIP 인트라 예측 모드로 변환하는데 필요할 수도 있다. 이 변환은 코딩 효율성을 감소시킬 수도 있는 근사치이다. 더욱이, 이 변환은 매핑 표를 위한 부가 스토리지를 필요로 할 수도 있고 룩업 동작들을 부가할 수도 있으며, 이는 비디오 코더의 비용 및 복잡성을 증가시킬 수도 있고 인코딩 및 디코딩 프로세스들을 느리게 할 수도 있다.

[0224] ALWIP 블록의 MPM 도출 동안, 이전 단락들에서의 예들과 유사하게, 좌측 및 상측 이웃 블록들로부터의 인트라 모드들이 체크되고 이용가능할 때 사용된다. 이웃 블록이 비-ALWIP 블록일 때, 비-ALWIP 블록의 인트라 모드는 ALWIP 블록에 의해 사용된 매트릭스들/모드들에 대응하지 않을 수도 있다. 즉, 현재 블록의 사이즈 및 현재 블록의 상측 또는 좌측 이웃 블록의 인트라 모드에 의존하여 현재 블록으로 사용하는데 이용가능한 상이한 ALWIP 모드들은 현재 블록과 사용하는데 이용가능한 ALWIP 모드들 중 어느 것에도 대응하지 않을 수도 있다. 따라서, 제 2 매핑 표는 비-ALWIP 블록의 인트라 모드를 ALWIP 블록의 매트릭스/모드에 매핑하는데 사용된다. 이는 압축에서 비효율성을 야기할 수도 있다. 또한, 제 2 매핑 표는 부가 스토리지를 필요로 하고 MPM 도출 프로세스에서 룩업 동작들을 부가한다. 이웃 블록이 조합된 인트라/인터 모드 (CIIP) 로 코딩되는 경우, 이웃 블록은 (예를 들어, 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해) 인트라 코딩되지 않은 모드로 간주되고 이용 불가능한 것으로 마킹된다. 이웃 블록이 인트라 블록 카피 모드 (IBC) 로 코딩되는 경우, 블록은 인트라 코딩되지 않은 모드로 간주되고 이용불가능한 것으로 마킹된다.

[0225] 본 개시는 ALWIP 모드의 설계를 개선할 수도 있는 기법들을 설명한다. 본 개시의 기법들은 비디오 인코더 (200) 및 비디오 디코더 (300) 와 같은 비디오 코더들의 비용 및 복잡성을 감소시킬 수도 있고, 인코딩 및 디코딩 프로세스들의 속도를 증가시킬 수도 있다. 본 개시의 기법들 및 예들은 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해 개별적으로 또는 임의의 조합으로 사용될 수도 있다.

[0226] 본 개시의 제 1 기법에 따라, 상측 이웃 블록이 현재 블록 (즉, 비디오 코더 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더(300)) 가 현재 코딩하고 있는 블록) 과 상이한 CTU 에 속할 때, 상측 이웃 블록은 alwip_flag 에 대한 컨텍스트의 도출를 위해 이용가능한 것으로 간주된다. 예를 들어, alwip_flag 에 대한 컨텍스트의 도출 프로세스는 다음과 같이 수정될 수도 있으며, <!> …</!> 태그들은 부가들을 표시한다:

[0227] alwip_flag 는 허용된 4 개의 컨텍스트들로 컨텍스트 코딩된다:

[0228] - 블록 폭 > 2*높이 또는 높이 > 2*폭인 경우, 컨텍스트 3 이 사용된다.

[0229] - 그 외, 컨텍스트 ctxId 가 사용되며, 여기서 ctxId 는 다음과 같이 도출된다:

[0230] o ctxId 를 0 으로 초기화한다

[0231] o 좌측 이웃 블록이 ALWIP 로 코딩되는 경우, ctxId++.

[0232] o 상측 이웃 블록이 ALWIP 로 코딩되고 <!> 상측 이웃 블록이 현재 블록과 동일한 CTU 에 있는 경우 </!> , ctxId++

[0233] 따라서, 이 예에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 현재 블록을 포함하는 CTU 로우 위의 CTU 로우에서의 블록들이 ALWIP 모드를 사용하여 코딩되는지 여부를 표시하는 데이터를 저장할 필요성을 회피할 수도 있다. 이러한 데이터를 저장할 필성을를 회피하면 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 감소시킬 수도 있다.

[0234] 일부 예들에서, 상측 CTU 로우로부터의 alwip_flag 의 값은 현재 블록에 대한 alwip_flag 의 컨텍스트를 도출할 목적으로 특정 값 (예를 들어, 0) 과 동일한 것으로 추론될 수도 있다. 즉, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 상측 CTU 로우에 있는 상측 이웃 블록들에 대한 alwip_flag 값이 모두 동일한 특정 값을 갖는다고 가정할 수도 있다. 이러한 방식으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 현재 블록을 포함하는 CTU 로우 위의 CTU 로우에서의 블록들이 ALWIP 모드를 사용하여 코딩되는지 여부를 표시하는 데이터를 저장할 필요성을 회피할 수도 있으며, 결과적으로 비디오 인코더 (200) 및 비디오 디코더 (300) 에 대한 비용 및 복잡성이 잠재적으로 감소된다.

[0235] 따라서, 본 개시의 제 1 기법의 예에 따라, 비디오 코더 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더 (300)) 는 ALWIP 신택스 엘리먼트 (예를 들어, alwip_flag) 에 대한 컨텍스트를 결정할 수도 있다. 이 예에서, ALWIP 신택스 엘리먼트는 현재 블록이 ALWIP 로 코딩되는지 여부를 표시한다. 또한, 이 예에서, 상측 이웃 블록은 상측 이웃 블록이 현재 블록과 상이한 CTU 에 있는 것에 기초하여 컨텍스트를 결정하는데 이용불가능한 것으로 결정되며, 상측 이웃 블록은 현재 블록의 상측 이웃이다. 이 예에서, 비디오 코더는 컨텍스트에 기초하여 ALWIP 신택스 엘리먼트를 코딩할 수도 있다. 예를 들어, 비디오 코더는 ALWIP 신택스 엘리먼트의 CABAC 코딩에서 컨텍스트를 사용할 수도 있다.

[0236] 본 개시의 제 2 기법에 따라, 폭 $w_T$ 및 높이 $h_T$ 의 특정 임계값보다 작은 블록들에 대해 ALWIP 를 디스에이블하기 위한 제약이 도입된다. 즉, 비디오 코딩 표준 (예를 들어, VVC 등) 은 임계치 폭보다 작은 폭 또는 임계치 높이보다 작은 높이를 갖는 블록들에 대해 ALWIP 를 디스에이블하는 제약을 부과할 수도 있다. 따라서, 비디오 코딩 표준은 $w_T$ 보다 작은 폭 및 $h_T$ 보다 작은 높이를 갖는 블록들이 ALWIP 를 사용하여 코딩될 수 없다.

예를 들어, 제약은 8 보다 작은 폭 및 8 보다 작은 높이를 갖는 블록들에 대해 디스에이블된다. 일부 예들에서, ALWIP-관련 신택스 엘리먼트들은 이러한 블록 사이즈 제한들에 기초하여 컨디셔닝된다. 예를 들어, 도 12 와 관련하여, 위에 언급된 바와 같이, ALWIP 에 의한 작은 블록들의 코딩은 높은 저장 요건들을 야기할 수도 있지만 코딩 효율에서 비교적 완만한 이득을 야기할 수도 있다. 이러한 제약을 도입하는 것은 비디오 인코더 (200) 및 비디오 디코더 (300) 가 ALWIP 를 구현하기 위해 저장해야 할 수도 있는 파라미터들의 수를 감소시킴으로써, 비디오 인코더 (200) 및 비디오 디코더 (300) 의 복잡성 및 비용을 잠재적으로 감소시킬 수도 있다. 따라서, 본 개시의 제 2 기법에 따라, 임계치 폭 (예를 들어, 8) 보다 작은 폭 또는 임계치 높이 (예를 들어, 8) 보다 작은 높이를 갖는 블록들에 대해 ALWIP 를 디스에이블하는 제약이 부과될 수도 있다.

[0237] 본 개시의 제 3 기법에 따라, ALWIP 코딩된 이웃 블록이 비-ALWIP 코딩된 블록의 MPM 도출에서 고려/테스트될 때, 이웃 블록은 이용불가능한 (또는 인터-코딩된) 것으로 간주될 수도 있고 및/또는 그러한 이웃 블록의 인트라 모드의 값은 비-ALWIP 코딩된 블록의 MPM 도출를 위한 특정 값 (예를 들어, 평면 모드, DC 모드, 또는 다른 인트라 예측 모드) 인 것으로 추론될 수도 있다. 설명을 쉽게 하기 위해, 본 개시는 특정 값을 디폴트 값으로 지칭할 수도 있다. 이러한 방식으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 ALWIP 모드들을 비-ALWIP-인트라 예측 모드들에 매핑하기 위해 매핑 표를 저장할 필요성을 회피할 수도 있고 매핑 표에 대한 룩업 동작들을 회피할 수도 있다. 이는 비디오 인코더 (200) 및 비디오 디코더 (300) 의 저장 요건들을 감소시킬 수도 있고, 이는 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 감소시킬 수도 있다.

[0238] 따라서, 본 개시의 제 3 기법에 따라, 비디오 코더 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더 (300)) 는 ALWIP 를 사용하여 코딩되지 않은 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 도출 프로세스를 수행할 수도 있다. MPM 도출 프로세스를 수행하는 것의 일부로서, 비디오 코더는 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정할 수도 있다. 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 비디오 코더는 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정할 수도 있다. 그 후 비디오 코더는 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함할 수도 있다.

이웃 블록이 ALWIP-코딩되지 않은 경우, 비디오 코더는 하나 이상의 MPM들이 이웃 블록의 인트라 예측 모드를 포함한다고 결정할 수도 있다. 비디오 코더는 또 다른 인트라 예측 모드 또는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩 (예를 들어, 인코딩 또는 디코딩) 할 수도 있다.

[0239] 또한, 일부 예들에서, 크로마 블록이 DM 모드 (즉, 크로마 블록의 인트라 예측 모드가 병치된 루마 블록으로부터 상속됨) 및 병치된 루마 블록 (또는 특히 크로마 블록의 DM 모드를 도출하는데 사용된 루마 블록) 이 ALWIP 로 코딩될 때, 병치된 블록의 인트라 모드는 디폴트 값 (예를 들어, 평면) 인 것으로 결정, 예를 들어 추론될 수도 있다. 이러한 방식으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 ALWIP 모드들을 비-ALWIP-인트라 예측 모드들에 매핑하기 위해 매핑 표를 저장할 필요성을 회피할 수도 있고 매핑 표에 대한 룩업 동작들을 회피할 수도 있다. 이는 비디오 인코더 (200) 및 비디오 디코더 (300) 의 저장 요건들을 감소시킬 수도 있고, 이는 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 감소시킬 수도 있다.

[0240] 따라서, 크로마 블록이 DM 모드로 코딩되는 일부 예들에서, 비디오 코더 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더(300)) 는 병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정할 수도 있으며, 여기서 병치된 루마 블록은 크로마 블록과 병치된다. 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여, 비디오 코더는 병치된 루마 블록의 인트라 예측 모드가 또 다른 인트라 예측 모드 또는 평면 모드와 같은, 디폴트 값이라고 결정, 예를 들어 추론할 수도 있다. 비디오 코더는 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 크로마 블록을 코딩 (예를 들어, 인코딩 또는 디코딩) 할 수도 있다. 예를 들어, 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 크로마 블록을 인코딩하기 위해, 비디오 인코더 (200) 는 인트라 예측 모드를 사용하여 예측 블록을 생성할 수도 있다. 비디오 인코더 (200) 는 그 후 예측 블록 및 크로마 블록의 샘플들에 기초하여 크로마 블록에 대한 잔차 데이터를 생성할 수도 있다. 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 크로마 블록을 디코딩하기 위해, 비디오 디코더 (300) 는 인트라 예측 모드를 사용하여 예측 블록을 생성할 수도 있다. 비디오 디코더 (300) 는 그 후 크로마 블록에 대한 잔차 데이터 및 예측 블록에 기초하여 크로마 블록을 복원할 수도 있다.

[0241] 본 개시의 제 4 기법에 따라, 비-ALWIP 이웃 블록이 ALWIP 블록의 MPM 도출에서 사용될 때, 이웃 블록은 이용불가능한 (또는 인터-코딩된) 것으로 간주될 수도 있고 및/또는 이웃 모드 값은 ALWIP 블록의 MPM 도출를 위한 디폴트 값 에 할당될 수도 있다. 즉, ALWIP-코딩된 블록에 대한 ALWIP-MPM들 및 이웃 블록이 ALWIP 를 사용하여 코딩되지 않는다고 결정할 때, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 이웃 블록이 이용불가능하다고 고려할 수도 있거나 이웃 블록이 디폴트 인트라 예측 모드를 갖는다고 가정할 수도 있다. 이는 비디오 인코더 (200) 및 비디오 디코더 (300) 가 비-ALWIP 방향성 인트라 예측 모드들을 ALWIP 모드들로 변환하기 위해 매핑 데이터를 저장할 필요성을 회피 또는 감소시킬 수도 있다. 이러한 매핑 데이터를 저장할 필요성을 회피 또는 감소시키면 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 감소시킬 수도 있다.

[0242] 일부 예들에서, 이웃 블록이 조합된 인트라/인터 모드 (CIIP) 로 코딩되는 경우, 이웃 블록은 이용가능한 것으로 간주되고 ALWIP 블록의 MPM 도출를 위해 또 다른 적절한 비-ALWIP 모드, 또는 평면 모드를 할당받을 수도 있다. 대안으로, CIIP 모드는 코딩 효율 이익들을 최적화하기 위해 적절한 ALWIP 모드에 직접 매핑될 수도 있다. 할당은 블록 치수 등에 의존할 수도 있다. 이웃 블록이 인트라 블록 카피 모드 (IBC) 로 코딩되는 경우, 블록은 이용가능한 것으로 간주되고 ALWIP 블록의 MPM 도출를 위해, 다른 적절한 비-ALWIP 모드, 또는 평면 모드를 할당받을 수도 있다. 대안으로, IBC 모드는 코딩 효율 이익들을 최적화하기 위해 적절한 ALWIP 모드에 직접 매핑될 수도 있다. 할당은 블록 치수 등에 의존할 수도 있다. CIIP 모드들 및 IBC 모드들을 평면 모드와 같은 특정 모드에 직접 (예를 들어, 1 대 1 관계로) 매핑하면 더 복잡한 데이터를 저장할 필요성을 회피 또는 감소시킬 수도 있다. 이러한 매핑 데이터를 저장할 필요성을 회피 또는 감소시키면 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 감소시킬 수도 있다.

[0243] 본 개시의 제 5 기법에 따라, 4 또는 0.25 의 폭-대-높이 비율을 갖는 블록들에 대해 ALWIP 를 디스에이블하기 위한 제약이 도입될 수도 있다. 즉, 비디오 코딩 표준은 4 또는 0.25 의 폭 대 높이 비율을 갖는 블록이 ALWIP 를 사용하여 코딩되지 않을 수도 있다는 것을 제공할 수도 있다. 이러한 폭 대 높이 비율을 갖는 블록들을 코딩하지 않으면 비디오 인코더 (200) 및 비디오 디코더 (300) 가 저장해야 할 수도 있는 ALWIP 파라미터들의 수를 감소시킴으로써, 비디오 인코더 (200) 및 비디오 디코더 (300) 의 비용 및 복잡성을 잠재적으로 감소시킬 수도 있다.

[0244] ALWIP 에 대한 MPM 리스트 도출은 현재 블록의 상단-우측 (예를 들어, 상단-우측 샘플 바로 위) 및/또는 하단-좌측 샘플들 (예를 들어, 하단-좌측 샘플의 바로 좌측) 에 인접하는 이웃 블록들로부터의 모드들을 사용한다.

일부 예들에서, ALWIP 에 대한 MPM들을 도출하기 위해 사용된 이웃 위치는 정규 인트라 예측을 위한 MPM들을 도출하는데 사용된 이웃 위치와 정렬된다. 보다 일반적으로, 현재 블록의 이웃에 있는 임의의 하나 이상의 블록들은 ALWIP 에 대한 MPM 도출을 위해 사용될 수도 있다. 도 13 은 본 개시의 하나 이상의 기법들에 따른, 현재 블록 (554) 에 대한 ALWIP MPM들을 도출하는데 사용된 예시의 이웃 블록들 (550, 552) 을 도시하는 개념적 다이어그램이다. 도 13 의 예에서, 블록 A (550) 는 현재 블록 (554) 의 상단-우측 샘플에 인접한 (상단-) 블록이다. 블록 L (552) 은 현재 블록 (554) 의 하단-좌측 샘플에 인접한 (좌측-) 블록이다.

[0245] JVET-N0217 로부터의 다음의 사양 텍스트는 본 개시의 제 4 기법과 관련하여 설명된 기법들을 구현하도록 편집된다. 변경들은 태그들 <!> …</!> 로 표시된다.

[0246] **ALWIP 블록의 MPM 도출을 위한 CIIP 코딩된 블록에 모드 할당**

[0247] **아핀 선형 가중 인트라 예측 모드에 대한 도출 프로세스**

[0248] 이 프로세스에 대한 입력은:

[0249] - 현재 픽처의 상단 좌측 샘플에 대한 현재 루마 코딩 블록의 상단 좌측 샘플을 특정하는 루마 위치 (xCb, yCb),

[0250] - 루마 샘플들에서 현재 코딩 블록의 폭을 특정하는 변수 cbWidth,

[0251] - 루마 샘플들에서 현재 코딩 블록의 높이를 특정하는 변수 cbHeight,

[0252] 이 프로세스에서, 아핀 선형 가중 예측 모드 IntraPredModeY[ xCb ][ yCb ] 가 도출된다.

[0253] IntraPredModeY[ xCb ][ yCb ] 는 다음의 순서화된 단계들에 의해 도출된다:

[0254] 1. 이웃 위치들 ( xNbA, yNbA ) 및 ( xNbB, yNbB ) 는 각각 ( xCb - 1, yCb ) 및 ( xCb, yCb - 1 ) 와 동일하게 설정된다.

[0255] 2. X 가 A 또는 B 로 대체되는 것에 대해, 변수 candLwipModeX 는 다음과 같이 도출된다:

[0256] - 조항 6.4.X 에 특정된 바와 같은 블록에 대한 가용성 도출 프로세스 [Ed. (BB): 이웃 블록 가용성 체크 프로세스 tbd] 는 입력들로서 ( xCb, yCb ) 와 동일하게 설정된 위치( xCurr, yCurr ) 및 ( xNbX, yNbX ) 동일하게 설정된 이웃 위치( xNbY, yNbY ) 로 호출되고, 출력들이 이용가능한X 에 할당된다.

[0257] - 후보 아핀 선형 가중 인트라 예측 모드 candLwipModeX 는 다음과 같이 도출된다:

[0258] - 다음 조건들 중 하나 이상이 참이면, candLwipModeX 는 -1 과 동일하게 설정된다:

[0259] - 변수 availableX 는 FALSE 이다.

[0260] - CuPredMode[ xNbX ][ yNbX ]는 MODE_INTRA 와 동일하지 않고 ciip_flag[ xNbX ][ yNbX ] 는 1 이 아니다.

[0261] - pcm_flag[ xNbX ][ yNbX ]는 1 이다.

[0262] - X 는 B 와 같고 yCb 1 은 ( ( yCb >> CtbLog2SizeY ) << CtbLog2SizeY ) 보다 작다.

[0263] - 그렇지 않으면, 다음이 적용된다:

[0264] - 8.4.X.1 절에 특정된 블록에 대한 사이즈 타입 도출 프로세스는 입력으로서 루마 샘플들에서의 현재 코딩 블록의 폭 cbWidth 과 루마 샘플들에서의 현재 코딩 블록 높이 cbHeight 로 호출되고, 출력은 변수 sizeId 에 할당된다.

[0265] - intra_lwip_flag[ xNbX ][ yNbX ] 이 1 과 동일한 경우, 8.4.X.1 절에 특정된 바와 같이 블록에 대한 사이즈 타입 도출 프로세스는 입력으로서 루마 샘플들에서의 이웃 코딩 블록의 폭 nbWidthX 과 루마 샘플들에서의 이웃 코딩 블록의 높이 nbHeightX 로 호출되고, 출력은 변수 sizeIdX 에 할당된다.

[0266] - sizeId 가 sizeIdX 와 동일하면, candLwipModeX 는 IntraPredModeY[ xNbX ][ yNbX ] 와 동일하게 설정된다.

[0267] - 그렇지 않으면, candLwipModeX 는 1 과 동일하게 설정된다.

[0268] - <!> 그렇지 않고, intra_lwip_flag[ xNbX ][ yNbX ] 가 0 과 같고 ciip_flag ciip_flag[ xNbX ][ yNbX ] 가 1 과 같으면, IntraPredModeY[ xNbX ][ yNbX ] 가 INTRA_PLANAR 와 동일하게 설정되고 candLwipModeX 는 표 8-X1 에서 특정된 바와 같이 IntraPredModeY[ xNbX ][ yNbX ] 및 sizeId 를 사용하여

도출된다.</!>

[0269] - 그렇지 않으면, candLwipModeX 는 표 8-X1 에 특정된 바와 같이 IntraPredModeY[ xNbX ][ yNbX ] 및 sizeId 를 사용하여 도출된다.

[0270] 3. candLwipModeList[ x ](x = 0..2) 는, 표 8-X2 에 특정된 바와 같이 lwipMpmCand[ sizeId ] 를 사용하여 다음과 같이 도출된다:

[0271] - candLwipModeA 및 candLwipModeB 는 모두 1 과 동일하며, 다음이 적용된다:

$$candLwipModeList[\ 0\ ] = lwipMpmCand[\ sizeId\ ][\ 0\ ] \quad (8\text{-}X1)$$

$$candLwipModeList[\ 1\ ] = lwipMpmCand[\ sizeId\ ][\ 1\ ] \quad (8\text{-}X2)$$

[0272] $$candLwipModeList[\ 2\ ] = lwipMpmCand[\ sizeId\ ][\ 2\ ] \quad (8\text{-}X3)$$

[0273] - 그렇지 않으면, 다음이 적용된다:

[0274] - candLwipModeA 가 candLwipModeB 와 동일하거나, candLwipModeA 또는 candLwipModeB 가 1 과 동일하면, 다음이 적용된다:

$$candLwipModeList[\ 0\ ] = (\ candLwipModeA\ !=-1\ )\ ?\ candLwipModeA\ :\ candLwipModeB \quad (8\text{-}X4)$$

[0275]

[0276] - candLwipModeList[ 0 ] 이 lwipMpmCand[ sizeId ][ 0 ] 와 동일하면, 다음이 적용된다:

$$candLwipModeList[\ 1\ ] = lwipMpmCand[\ sizeId\ ][\ 1\ ] \quad (8\text{-}X5)$$

[0277] $$candLwipModeList[\ 2\ ] = lwipMpmCand[\ sizeId\ ][\ 2\ ] \quad (8\text{-}X6)$$

[0278] - 그렇지 않으면, 다음이 적용된다:

$$candLwipModeList[\ 1\ ] = lwipMpmCand[\ sizeId\ ][\ 0\ ] \quad (8\text{-}X7)$$

$$candLwipModeList[\ 2\ ] = (\ candLwipModeList[\ 0\ ] != lwipMpmCand[\ sizeId\ ][\ 1\ ]\ )\ ?\ lwipMpmCand[\ sizeId\ ][\ 1\ ]\ :\ lwipMpmCand[\ sizeId\ ][\ 2\ ] \quad (8\text{-}X8)$$

[0279]

[0280] - 그렇지 않으면, 다음이 적용된다:

$$candLwipModeList[\ 0\ ] = candLwipModeA \quad (8\text{-}X9)$$

$$candLwipModeList[\ 1\ ] = candLwipModeB \quad (8\text{-}X10)$$

[0281]

[0282] - candLwipModeA 및 candLwipModeB 가 모두 lwipMpmCand[ sizeId ][ 0 ] 와 동일하지 않으면, 다음이 적용된다:

[0283] candLwipModeList[ 2 ] = lwipMpmCand[ sizeId ][ 0 ] (8-X11)

[0284] - 그렇지 않으면, 다음이 적용된다:

[0285] - candLwipModeA 및 candLwipModeB 가 모두 lwipMpmCand[ sizeId ][ 1 ] 와 동일하지 않으면, 다음이 적용된다:

[0286] candLwipModeList[ 2 ] = lwipMpmCand[ sizeId ][ 1 ] (8-X12)

[0287] - 그렇지 않으면, 다음이 적용된다:

[0288] candLwipModeList[ 2 ] = lwipMpmCand[ sizeId ][ 2 ] (8-X13)

[0289] 4. IntraPredModeY[ xCb ][ yCb ] 는 다음의 절차를 적용함으로써 도출된다:

[0290] - intra_lwip_mpm_flag[ xCb ][ yCb ] 이 1 과 동일하면, IntraPredModeY[ xCb ][ yCb ] 는 candLwipModeList[ intra_lwip_mpm_idx[ xCb ][ yCb ] ] 와 동일하게 설정된다.

[0291] - 그렇지 않으면, IntraPredModeY[ xCb ][ yCb ] 가 다음의 순서화된 단계들을 적용함으로써 도출된다:

[0292] 1. candLwipModeList[ i ] 이 candLwipModeList[ j ] 보다 크고 i = 0..1 에 대해 그리고 각각의 i 에 대해, j = ( i + 1 )..2 일 때, 양자의 값들은 다음과 같이 스와핑된다:

[0293] ( candLwipModeList[ i ], candLwipModeList[ j ] ) = Swap( candLwipModeList[ i ], candLwipModeList[ j ] ) (8-X14)

[0294] 2. IntraPredModeY[ xCb ][ yCb ] 는 다음의 순서화된 단계들에 의해 도출된다:

[0295] i. IntraPredModeY[ xCb ][ yCb ] 는 intra_lwip_mpm_remainder[ xCb ][ yCb ] 와 동일하게 설정된다.

[0296] ii. i 가 0 내지 2 포함인 경우, IntraPredModeY[ xCb ][ yCb ] 가 candLwipModeList[ i ] 이상일 때, IntraPredModeY[ xCb ][ yCb ] 의 값은 1 씩 증분된다.

[0297] 변수 IntraPredModeY[ x ][ y ] (x = xCb..xCb + cbWidth - 1 및 y = yCb..yCb + cbHeight - 1) 은 IntraPredModeY[ xCb ][ yCb ] 와 동일하게 설정된다.

[0298] **8.4.X.1 예측 블록 사이즈 타입에 대한 도출 프로세스**

[0299] 이 프로세스에 대한 입력은:

[0300] - 루마 샘플들에서 현재 코딩 블록의 폭을 특정하는 변수 cbWidth,

[0301] - 루마 샘플들에서 현재 코딩 블록의 높이를 특정하는 변수 cbHeight,

[0302] 이러한 프로세스의 출력은 변수 sizeID 이다.

[0303] 변수 sizeId 는 다음과 같이 도출된다:

[0304] - cbWidth 및 cbHeight 양자 모두가 4 이면, sizeId 는 0 과 동일하게 설정된다.

[0305] - 그렇지 않고, cbWidth 및 cbHeight 양자 모두가 8 이하이면, sizeId 는 1 과 동일하게 설정된다.

[0306] - 그렇지 않으면, sizeId 는 2 와 동일하게 설정된다.

[0307] 표 8-X1 - 인트라 예측과 아핀 선형 가중 인트라 예측 모드들 사이의 매핑 사양

| IntraPredModeY[ xNbX ][ yNbX ] | 블록 사이즈 타입 사이즈ID | | |
|---|---|---|---|
| | **0** | **1** | **2** |
| **0** | 17 | 0 | 5 |
| **1** | 17 | 0 | 1 |
| **2, 3** | 17 | 10 | 3 |
| **4, 5** | 9 | 10 | 3 |
| **6,7** | 9 | 10 | 3 |
| **8, 9** | 9 | 10 | 3 |
| **10, 11** | 9 | 10 | 0 |
| **12, 13** | 17 | 4 | 0 |
| **14, 15** | 17 | 6 | 0 |
| **16, 17** | 17 | 7 | 4 |
| **18, 19** | 17 | 7 | 4 |
| **20, 21** | 17 | 7 | 4 |
| **22, 23** | 17 | 5 | 5 |
| **24, 25** | 17 | 5 | 1 |
| **26, 27** | 5 | 0 | 1 |
| **28, 29** | 5 | 0 | 1 |
| **30, 31** | 5 | 3 | 1 |
| **32, 33** | 5 | 3 | 1 |
| **34, 35** | 34 | 12 | 6 |
| **36, 37** | 22 | 12 | 6 |
| **38, 39** | 22 | 12 | 6 |
| **40, 41** | 22 | 12 | 6 |
| **42, 43** | 22 | 14 | 6 |
| **44, 45** | 34 | 14 | 10 |
| **46, 47** | 34 | 14 | 10 |
| **48, 49** | 34 | 16 | 9 |
| **50, 51** | 34 | 16 | 9 |
| **52, 53** | 34 | 16 | 9 |

[0308]

| | | | |
|---|---|---|---|
| **54, 55** | 34 | 15 | 9 |
| **56, 57** | 34 | 13 | 9 |
| **58, 59** | 26 | 1 | 8 |
| **60, 61** | 26 | 1 | 8 |
| **62, 63** | 26 | 1 | 8 |
| **64, 65** | 26 | 1 | 8 |
| **66** | 26 | 1 | 8 |

[0309]

[0310] **표 8-X2 - 아핀 선형 가중 인트라 예측 후보 모드들의 사양**

| | 후보 모드 | | |
|---|---|---|---|
| | **0** | **1** | **2** |
| lwipMpmCand[ 0 ] | 17 | 34 | 5 |
| lwipMpmCand[ 1 ] | 0 | 7 | 16 |
| lwipMpmCand[ 2 ] | 1 | 4 | 6 |

[0311]

[0312] 본 개시의 제 5 기법을 구현하기 위한 JVET-N0217 의 사양 텍스트에 대한 변경 예는 하기 <!> …</!> 태그들로 표기된다.

[0313] **아핀 선형 가중 인트라 예측 모드에 대한 도출 프로세스**

[0314] 이 프로세스에 대한 입력은:

[0315] - 현재 픽처의 상단 좌측 샘플에 대한 현재 루마 코딩 블록의 상단 좌측 샘플을 특정하는 루마 위치 (xCb, yCb),

[0316] - 루마 샐플들에서 현재 코딩 블록의 폭을 특정하는 변수 cbWidth,

[0317] - 루마 샘플들에서 현재 코딩 블록의 높이를 특정하는 변수 cbHeight,

[0318] 이 프로세스에서, 아핀 선형 가중 예측 모드 IntraPredModeY[ xCb ][ yCb ] 가 도출된다.

[0319] IntraPredModeY[ xCb ][ yCb ] 는 다음의 순서화된 단계들에 의해 도출된다:

[0320] 1. 이웃 위치들 ( xNbA, yNbA ) 및 ( xNbB, yNbB ) 는 각각 ( xCb - 1, yCb <!>+ cbHeight - 1</!> ) 및 ( xCb <!>+ cbWidth - 1</!>, yCb - 1 ) 와 동일하게 설정된다.

[0321] 2. X 가 A 또는 B 로 대체되는 것에 대해, 변수 candLwipModeX 는 다음과 같이 도출된다:

[0322] - 조항 6.4.X 에 특정된 바와 같은 블록에 대한 가용성 도출 프로세스 [Ed. (BB): 이웃 블록 가용성 체크 프로세스 tbd] 는 입력들로서 ( xCb, yCb ) 와 동일하게 설정된 위치( xCurr, yCurr ) 및 ( xNbX, yNbX ) 동일하게 설정된 이웃 위치( xNbY, yNbY ) 로 호출되고, 출력들이 이용가능한X 에 할당된다.

[0323] - 후보 아핀 선형 가중 인트라 예측 모드 candLwipModeX 는 다음과 같이 도출된다:

[0324] - 다음 조건들 중 하나 이상이 참이면, candLwipModeX 는 -1 과 동일하게 설정된다:

[0325] - 변수 availableX 는 FALSE 이다.

[0326] - CuPredMode[ xNbX ][ yNbX ]는 MODE_INTRA 와 동일하지 않고 ciip_flag[ xNbX ][ yNbX ] 는 1 이 아니다.

[0327] - pcm_flag[ xNbX ][ yNbX ]는 1 이다.

[0328] - X 는 B 와 같고 yCb - 1 은 ( ( yCb >> CtbLog2SizeY ) << CtbLog2SizeY ) 보다 작다.

[0329] - 그렇지 않으면, 다음이 적용된다:

[0330] - 8.4.X.1 절에 특정된 블록에 대한 사이즈 타입 도출 프로세스는 입력으로서 루마 샘플들에서의 현재 코딩 블록의 폭 cbWidth 과 루마 샘플들에서의 현재 코딩 블록 높이 cbHeight 로 호출되고, 출력은 변수 sizeId 에 할당된다.

[0331] - intra_lwip_flag[ xNbX ][ yNbX ] 이 1 과 동일한 경우, 8.4.X.1 절에 특정된 바와 같이 블록에 대한 사이즈 타입 도출 프로세스는 입력으로서 루마 샘플들에서의 이웃 코딩 블록의 폭 nbWidthX 과 루마 샘플들에서의 이웃 코딩 블록의 높이 nbHeightX 로 호출되고, 출력은 변수 sizeIdX 에 할당된다.

[0332] - sizeId 가 sizeIdX 와 동일하면, candLwipModeX 는 IntraPredModeY[ xNbX ][ yNbX ] 와 동일하게 설정된다.

[0333] - 그렇지 않으면, candLwipModeX 는 1 과 동일하게 설정된다.

[0334] - 그렇지 않으면, candLwipModeX 는 표 8-X1 에 특정된 바와 같이 IntraPredModeY[ xNbX ][ yNbX ] 및

sizeId 를 사용하여 도출된다.

[0335] 3. candLwipModeList[ x ](x = 0..2) 는, 표 8-X2 에 특정된 바와 같이 lwipMpmCand[ sizeId ] 를 사용하여 다음과 같이 도출된다:

[0336] - candLwipModeA 및 candLwipModeB 는 모두 1 과 동일하며, 다음이 적용된다:

candLwipModeList[ 0 ] = lwipMpmCand[ sizeId ][ 0 ] (8-X1)

candLwipModeList[ 1 ] = lwipMpmCand[ sizeId ][ 1 ] (8-X2)

candLwipModeList[ 2 ] = lwipMpmCand[ sizeId ][ 2 ] (8-X3)

[0337]

[0338] - 그렇지 않으면, 다음이 적용된다:

[0339] - candLwipModeA 가 candLwipModeB 와 동일하거나, candLwipModeA 또는 candLwipModeB 가 1 과 동일하면, 다음이 적용된다:

candLwipModeList[ 0 ] =
( candLwipModeA != −1 ) ? candLwipModeA : candLwipModeB
(8-X4)

[0340]

[0341] - candLwipModeList[ 0 ] 이 lwipMpmCand[ sizeId ][ 0 ] 와 동일하면, 다음이 적용된다:

candLwipModeList[ 1 ] = lwipMpmCand[ sizeId ][ 1 ] (8-X5)

candLwipModeList[ 2 ] = lwipMpmCand[ sizeId ][ 2 ] (8-X6)

[0342]

[0343] - 그렇지 않으면, 다음이 적용된다:

candLwipModeList[ 1 ] = lwipMpmCand[ sizeId ][ 0 ] (8-X7)

[0344]

candLwipModeList[ 2 ] =
( candLwipModeList[ 0 ] != lwipMpmCand[ sizeId ][ 1 ] ) ?
lwipMpmCand[ sizeId ][ 1 ] : lwipMpmCand[ sizeId ][ 2 ]
(8-X8)

[0345]

[0346] - 그렇지 않으면, 다음이 적용된다:

candLwipModeList[ 0 ] = candLwipModeA
(8-X9)

candLwipModeList[ 1 ] = candLwipModeB
(8-X10)

[0347]

[0348] - candLwipModeA 및 candLwipModeB 가 모두 lwipMpmCand[ sizeId ][ 0 ] 와 동일하지 않으면, 다음이 적용된다:

candLwipModeList[ 2 ] = lwipMpmCand[ sizeId ][ 0 ] (8-X11)

[0349]

[0350] - 그렇지 않으면, 다음이 적용된다

[0351] - candLwipModeA 및 candLwipModeB 가 모두 lwipMpmCand[ sizeId ][ 1 ] 와 동일하지 않으면, 다음이 적용

된다:

[0352] $$candLwipModeList[\,2\,] = lwipMpmCand[\,sizeId\,][\,1\,] \quad (8\text{-}X12)$$

[0353] - 그렇지 않으면, 다음이 적용된다:

[0354] $$candLwipModeList[\,2\,] = lwipMpmCand[\,sizeId\,][\,2\,] \quad (8\text{-}X13)$$

[0355] 4. IntraPredModeY[ xCb ][ yCb ] 는 다음의 절차를 적용함으로써 도출된다:

[0356] - intra_lwip_mpm_flag[ xCb ][ yCb ] 이 1 과 동일하면, IntraPredModeY[ xCb ][ yCb ] 는 candLwipModeList[ intra_lwip_mpm_idx[ xCb ][ yCb ] ] 와 동일하게 설정된다.

[0357] - 그렇지 않으면, IntraPredModeY[ xCb ][ yCb ] 가 다음의 순서화된 단계들을 적용함으로써 도출된다:

[0358] 1. candLwipModeList[ i ] 이 candLwipModeList[ j ] 보다 크고 i = 0..1 에 대해 그리고 각각의 i 에 대해, j = ( i + 1 )..2 일 때, 양자의 값들은 다음과 같이 스와핑된다:

[0359] $$(\,candLwipModeList[\,i\,], candLwipModeList[\,j\,]\,) = Swap(\,candLwipModeList[\,i\,], candLwipModeList[\,j\,]\,) \quad (8\text{-}X14)$$

[0360] 2. IntraPredModeY[ xCb ][ yCb ] 는 다음의 순서화된 단계들에 의해 도출된다:

[0361] i. IntraPredModeY[ xCb ][ yCb ] 는 intra_lwip_mpm_remainder[ xCb ][ yCb ] 와 동일하게 설정된다.

[0362] ii. i 가 0 내지 2 포함인 경우, IntraPredModeY[ xCb ][ yCb ] 가 candLwipModeList[ i ] 이상일 때, IntraPredModeY[ xCb ][ yCb ] 의 값은 1 씩 증분된다.

[0363] 변수 IntraPredModeY[ x ][ y ](x = xCb..xCb + cbWidth - 1 및 y = yCb..yCb + cbHeight - 1) 은 IntraPredModeY[ xCb ][ yCb ] 와 동일하게 설정된다.

[0364] 표 8-X1 - 인트라 예측과 아핀 선형 가중 인트라 예측 모드들 사이의 매핑 사양

| IntraPredModeY[ xNbX ][ yNbX ] | 블록 사이즈 타입 사이즈ID | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| 0 | 17 | 0 | 5 |
| 1 | 17 | 0 | 1 |
| 2, 3 | 17 | 10 | 3 |
| 4, 5 | 9 | 10 | 3 |
| 6,7 | 9 | 10 | 3 |
| 8, 9 | 9 | 10 | 3 |
| 10, 11 | 9 | 10 | 0 |
| 12, 13 | 17 | 4 | 0 |
| 14, 15 | 17 | 6 | 0 |
| 16, 17 | 17 | 7 | 4 |
| 18, 19 | 17 | 7 | 4 |
| 20, 21 | 17 | 7 | 4 |
| 22, 23 | 17 | 5 | 5 |
| 24, 25 | 17 | 5 | 1 |
| 26, 27 | 5 | 0 | 1 |
| 28, 29 | 5 | 0 | 1 |
| 30, 31 | 5 | 3 | 1 |
| 32, 33 | 5 | 3 | 1 |
| 34, 35 | 34 | 12 | 6 |
| 36, 37 | 22 | 12 | 6 |
| 38, 39 | 22 | 12 | 6 |
| 40, 41 | 22 | 12 | 6 |
| 42, 43 | 22 | 14 | 6 |
| 44, 45 | 34 | 14 | 10 |
| 46, 47 | 34 | 14 | 10 |
| 48, 49 | 34 | 16 | 9 |
| 50, 51 | 34 | 16 | 9 |
| 52, 53 | 34 | 16 | 9 |

[0365]

| | | | |
|---|---|---|---|
| 54, 55 | 34 | 15 | 9 |
| 56, 57 | 34 | 13 | 9 |
| 58, 59 | 26 | 1 | 8 |
| 60, 61 | 26 | 1 | 8 |
| 62, 63 | 26 | 1 | 8 |
| 64, 65 | 26 | 1 | 8 |
| 66 | 26 | 1 | 8 |

[0366]

[0367] **표 8-X2 - 아핀 선형 가중 인트라 예측 후보 모드들의 사양**

| | 후보 모드 | | |
|---|---|---|---|
| | **0** | **1** | **2** |
| lwipMpmCand[ 0 ] | 17 | 34 | 5 |
| lwipMpmCand[ 1 ] | 0 | 7 | 16 |
| lwipMpmCand[ 2 ] | 1 | 4 | 6 |

[0368]

[0369] 본 개시는 일반적으로 신택스 엘리먼트들과 같은, 소정의 정보의 "시그널링" 을 언급할 수도 있다. 용어 " 시그널링" 은 일반적으로 인코딩된 비디오 데이터를 디코딩하는데 사용된 신택스 엘리먼트들 및/또는 다른 데이터에 대한 값들의 통신을 지칭할 수도 있다. 즉, 비디오 인코더 (200) 는 비트스트림에서 신택스 엘리먼트들에 대한 값들을 시그널링할 수도 있다. 일반적으로, 시그널링은 비트스트림에서 값을 생성하는 것을 지칭한다. 위에 언급된 바와 같이, 소스 디바이스 (102) 는 목적지 디바이스 (116) 에 의한 나중 취출을 위해 저장 디바이스 (112) 에 신택스 엘리먼트를 저장할 때 발생할 수 있는 것과 같은, 실질적으로 실시간으로 또는 비실시간으로, 비트스트림을 목적지 디바이스 (116) 로 전송할 수도 있다.

[0370] 도 14a 및 도 14b 는 예시의 쿼드트리 이진 트리 (QTBT) 구조 (130), 및 대응하는 코딩 트리 유닛 (CTU) (132) 을 도시하는 개념적 다이어그램들이다. 실선들은 쿼드트리 분할을 나타내고, 점선들은 이진 트리 분할을 나타낸다. 이진 트리의 각각의 분할된 (즉, 비-리프) 노드에서, 어느 분할 타입 (즉, 수평 또는 수직) 이 사용되는지를 나타내기 위해 하나의 플래그가 시그널링되며, 여기서 0 은 수평 분할을 표시하고 1 은 수직 분할을 표시한다. 쿼드트리 스플릿에 대해, 스플릿 타입을 표시할 필요는 없는데, 이는 쿼드트리 노드들이 동일한 사이즈를 가진 4 개의 서브-블록들로 수평으로 및 수직으로 블록을 스플릿하기 때문이다. 이에 따라, QTBT 구조 (130)(즉, 실선들) 의 영역 트리 레벨 (즉, 제 1 레벨) 에 대한 신택스 엘리먼트들 (이를 테면, 스플릿 정보) 및 QTBT 구조 (130) (즉, 점선들) 의 예측 트리 레벨 (즉, 제 2 레벨) 에 대한 신택스 엘리먼트들 (예컨대, 스플릿 정보) 을, 비디오 인코더 (200) 가 인코딩할 수도 있고, 비디오 디코더 (300) 가 디코딩할 수도 있다. QTBT 구조 (130) 의 종단 리프 노드들에 의해 표현된 CU들에 대해, 예측 및 변환 데이터와 같은 비디오 데이터를, 비디오 인코더 (200) 가 인코딩할 수도 있고, 비디오 디코더 (300) 가 디코딩할 수도 있다.

[0371] 일반적으로, 도 14b 의 CTU (132) 는 제 1 및 제 2 레벨들에서 QTBT 구조 (130) 의 노드들에 대응하는 블록들의 사이즈들을 정의하는 파라미터들과 연관될 수도 있다. 이들 파라미터들은 CTU 사이즈 (샘플들에서 CTU (132) 의 사이즈를 나타냄), 최소 쿼드트리 사이즈 (최소 허용된 쿼드트리 리프 노드 사이즈를 나타내는 MinQTSize), 최대 이진 트리 사이즈 (최대 허용된 이진 트리 루트 노드 사이즈를 나타내는 MaxBTSize), 최대 이진 트리 깊이 (최대 허용된 이진 트리 깊이를 나타내는 MaxBTDepth), 및 최소 이진 트리 사이즈 (최소 허용된 이진 트리 리프 노드 사이즈를 나타내는 MinBTSize) 를 포함할 수도 있다.

[0372] CTU 에 대응하는 QTBT 구조의 루트 노드는 QTBT 구조의 제 1 레벨에서 4 개의 자식 노드들을 가질 수도 있고, 이들의 각각은 쿼드트리 파티셔닝에 따라 파티셔닝될 수도 있다. 즉, 제 1 레벨의 노드들은 리프 노드들 (자식 노드들이 없음) 이거나 또는 4 개의 자식 노드들을 갖는다. QTBT 구조 (130) 의 예는 그러한 노드들을 브랜치들에 대한 실선들을 갖는 부모 노드 및 자식 노드들을 포함하는 것으로서 나타낸다. 제 1 레벨의 노드들이 최대 허용된 이진 트리 루트 노드 사이즈 (MaxBTSize) 보다 더 크지 않으면, 그들은 개개의 이진 트리들에 의해 추가로 파티셔닝될 수 있다. 하나의 노드의 이진 트리 분할은 스플릿으로부터 발생하는 노드들이 최소 허용된 이진 트리 리프 노드 사이즈 (MinBTSize) 또는 최대 허용된 이진 트리 깊이 (MaxBTDepth) 에 도달할 때까지 반복될 수 있다. QTBT 구조 (130) 의 예는 그러한 노드들을 브랜치들에 대한 점선들을 갖는 것으로서 나타낸다. 이진 트리 리프 노드는 임의의 추가 파티셔닝 없이, 예측 (예를 들어, 인트라-픽처 또는 인터-픽처 예측) 및 변환을 위해 사용되는 코딩 유닛 (CU) 으로 지칭된다. 위에 논의된 바와 같이, CU들은 또한, "비디오 블록들" 또는 "블록들" 로 지칭될 수도 있다.

[0373] QTBT 파티셔닝 구조의 일 예에서, CTU 사이즈는 128x128 (루마 샘플들 및 2 개의 대응하는 64x64 크로마 샘플들) 로서 설정되고, MinQTSize 는 16x16 으로서 설정되고, MaxBTSize 는 64x64 로서 설정되고, (폭 및 높이 양자 모두에 대한) MinBTSize 는 4 로서 설정되고, 그리고 MaxBTDepth 는 4 로서 설정된다. 쿼드트리 파티셔닝은 쿼드트리 리프 노드들을 생성하기 위해 먼저 CTU 에 적용된다. 쿼드트리 리프 노드들은 16x16 (즉, MinQTSize) 으로부터 128x128 (즉, CTU 사이즈) 까지의 사이즈들을 가질 수도 있다. 쿼드트리 리프 노

드가 128x128 인 경우, 사이즈가 MaxBTSize (즉, 이 예에서는 64x64) 를 초과하기 때문에 그것은 바이너리 트리에 의해 추가로 스플릿되지 않을 것이다. 그렇지 않으면, 쿼드트리 노드는 바이너리 트리에 의해 추가로 파티셔닝될 것이다. 따라서, 쿼드트리 리프 노드는 또한 이진 트리에 대한 루트 노드이고 이진 트리 깊이를 0 으로서 갖는다. 이진 트리 깊이가 MaxBTDepth (이 예에서는 4) 에 도달할 때, 추가의 스플릿이 허용되지 않는다. 이진 트리 노드가 MinBTSize (이 예에서는 4) 와 동일한 폭을 가질 때, 그것은 추가의 수직 스플릿이 허용되지 않음을 암시한다. 유사하게, 높이가 MinBTSize 와 동일한 이진 트리 노드는 그 이진 트리 노드에 대해 추가의 수평 스플릿이 허용되지 않음을 암시한다. 위에 언급된 바와 같이, 이진 트리의 리프 노드들은 CU들로 지칭되고, 추가 파티셔닝 없이 예측 및 변환에 따라 추가로 프로세싱된다.

[0374] 도 15 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코더 (200) 를 도시하는 블록 다이어그램이다. 도 15 는 설명의 목적으로 제공되며 본 개시에 폭넓게 예시되고 기재되는 바와 같이 기법들을 제한하는 것으로 고려되지 않아야 한다. 설명의 목적으로, 본 개시는 HEVC 비디오 코딩 표준 및 개발 중인 H.266 비디오 코딩 표준과 같은 비디오 코딩 표준들의 콘텍스트에서 비디오 인코더 (200) 를 기재한다. 그러나, 본 개시의 기법들은 이들 비디오 코딩 표준들에 한정되지 않으며, 일반적으로 비디오 인코딩 및 디코딩에 적용가능하다.

[0375] 도 15 의 예에서, 비디오 인코더 (200) 는 비디오 데이터 메모리 (230), 모드 선택 유닛 (202), 잔차 생성 유닛 (204), 변환 프로세싱 유닛 (206), 양자화 유닛 (208), 역 양자화 유닛 (210), 역 변환 프로세싱 유닛 (212), 복원 유닛 (214), 필터 유닛 (216), 디코딩된 픽처 버퍼 (DPB)(218), 및 엔트로피 인코딩 유닛 (220) 을 포함한다. 비디오 데이터 메모리 (230), 모드 선택 유닛 (202), 잔차 생성 유닛 (204), 변환 프로세싱 유닛 (206), 양자화 유닛 (208), 역 양자화 유닛 (210), 역 변환 프로세싱 유닛 (212), 복원 유닛 (214), 필터 유닛 (216), DPB (220), 및 엔트로피 인코딩 유닛 (220) 중 임의의 것 또는 전부가 하나 이상의 프로세서들에서 또는 프로세싱 회로부에서 구현될 수도 있다. 더욱이, 비디오 인코더 (200) 는 이들 및 다른 기능들을 수행하기 위해 부가 또는 대안의 프로세서들 또는 프로세싱 회로부를 포함할 수도 있다.

[0376] 비디오 데이터 메모리 (230) 는 비디오 인코더 (200) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 인코더 (200) 는 예를 들어, 비디오 소스 (104)(도 1) 로부터 비디오 데이터 메모리 (230) 에 저장된 비디오 데이터를 수신할 수도 있다. DPB (218) 는 비디오 인코더 (200) 에 의한 후속 비디오 데이터의 예측에 사용하기 위해 참조 비디오 데이터를 저장하는 참조 픽처 메모리로서 작용할 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 다양한 메모리 디바이스들, 예컨대 동기식 랜덤 액세스 메모리 (synchronous dynamic random access memory; SDRAM), 자기저항 RAM (magnetoresistive RAM; MRAM), 저항 RAM (resistive RAM; RRAM) 을 포함하는 DRAM, 또는 다른 타입들의 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (230) 는 도시된 바와 같이, 비디오 인코더 (200) 의 다른 컴포넌트들과 온-칩이거나 그 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0377] 본 개시에서, 비디오 데이터 메모리 (230) 에 대한 참조는 이처럼 구체적으로 기재되지 않으면 비디오 인코더 (200) 내부의 메모리 또는 이처럼 구체적으로 기재되지 않으면 비디오 인코더 (200) 외부의 메모리로 제한되는 것으로 해석되지 않아야 한다. 오히려, 비디오 데이터 메모리 (230) 에 대한 참조는 비디오 인코더 (200) 가 인코딩을 위해 수신하는 비디오 데이터 (예를 들어, 인코딩될 현재 블록에 대한 비디오 데이터) 를 저장하는 참조 메모리로서 이해되어야 한다. 도 1 의 메모리 (106) 는 또한 비디오 인코더 (200) 의 다양한 유닛들으로부터의 출력들의 일시적 저장을 제공할 수도 있다.

[0378] 도 15 의 다양한 유닛들은 비디오 인코더 (200) 에 의해 수행되는 동작들의 이해를 돕기 위해 도시된다. 이 유닛들은 고정 기능 회로들, 프로그램가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하며, 수행될 수 있는 동작들에 대해 미리설정된다. 프로그램가능 회로들은 다양한 태스크들을 수행하도록 프로그램될 수 있는 회로들을 지칭하고, 수행될 수 있는 동작들에서 유연한 기능성을 제공한다. 예를 들어, 프로그램가능 회로들은, 프로그램가능 회로들이 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 소프트웨어 명령들을 (예를 들어, 파라미터들을 수신하거나 파라미터들을 출력하기 위해) 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작 타입들은 일반적으로 불변이다. 일부 예들에서, 유닛들 중 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그램가능) 일 수도 있고, 일부 예들에 있어서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.

[0379] 비디오 인코더 (200) 는 프로그램가능 회로들로부터 형성된, 산술 로직 유닛 (arithmetic logic unit; ALU) 들, 기본 기능 유닛 (elementary function unit; EFU) 들, 디지털 회로들, 아날로그 회로들, 및/또는 프로그램가능 코어들을 포함할 수도 있다. 비디오 인코더 (200) 의 동작들이 프로그램가능 회로들에 의해 실행되는 소프트웨어를 사용하여 수행되는 예들에서, 메모리 (106)(도 1) 는 비디오 인코더 (200) 가 수신하고 실행하는 소프트웨어의 오브젝트 코드를 저장할 수도 있거나 또는 비디오 인코더 (200) 내의 다른 메모리 (미도시) 가 이러한 명령들을 저장할 수도 있다 .

[0380] 비디오 데이터 메모리 (230) 는 수신된 비디오 데이터를 저장하도록 구성된다. 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 비디오 데이터의 픽처를 취출하고 그 비디오 데이터를 잔차 생성 유닛 (204) 및 모드 선택 유닛 (202) 에 제공할 수도 있다. 비디오 데이터 메모리 (230) 에서의 비디오 데이터는 인코딩될 원시 비디오 데이터일 수도 있다.

[0381] 모드 선택 유닛 (202) 은 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라-예측 유닛 (226) 을 포함한다. 모드 선택 유닛 (202) 은 다른 예측 모드들에 따라 비디오 예측을 수행하기 위해 부가적인 기능 유닛들을 포함할 수도 있다. 예를 들어, 모드 선택 유닛 (202) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 추정 유닛 (222) 및/또는 모션 보상 유닛 (224) 의 일부일 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다. 예를 들어, 도 15 의 예에서, 인트라 예측 유닛 (226) 은 ALWIP 를 수행하도록 구성된 ALWIP 유닛 (227) 을 포함한다.

[0382] 모드 선택 유닛 (202) 은 일반적으로 인코딩 파라미터들의 조합들 및 그러한 조합들에 대한 결과의 레이트-왜곡 값들을 테스트하기 위해 다중 인코딩 패스들을 조정한다. 인코딩 파라미터들은 CU들로의 CTU들의 파티셔닝, CU들에 대한 예측 모드들, CU들의 잔차 데이터에 대한 변환 타입들, CU들의 잔차 데이터에 대한 양자화 파라미터들 등을 포함할 수도 있다. 모드 선택 유닛 (202) 은 궁극적으로 다른 테스트된 조합들보다 우수한 레이트-왜곡 값들을 갖는 인코딩 파라미터들의 조합을 선택할 수도 있다.

[0383] 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 취출된 픽처를 일련의 CTU들로 파티셔닝하고, 슬라이스 내에 하나 이상의 CTU들을 캡슐화할 수도 있다. 모드 선택 유닛 (202) 은 상술한 HEVC 의 쿼드-트리 구조 또는 QTBT 구조와 같은, 트리 구조에 따라 픽처의 CTU 를 파티셔닝할 수도 있다. 상술한 바와 같이, 비디오 인코더 (200) 는 트리 구조에 따라 CTU 를 파티셔닝하는 것으로부터 하나 이상의 CU들을 형성할 수도 있다. 이러한 CU 는 또한 일반적으로 "비디오 블록" 또는 "블록" 으로 지칭될 수도 있다.

[0384] 일반적으로, 모드 선택 유닛 (202) 은 또한 현재 블록 (예를 들어, 현재 CU, 또는 HEVC 에서, PU 및 TU 의 오버랩 부분) 에 대한 예측 블록을 생성하기 위해 그의 컴포넌트들 (예를 들어, 모션 추정 유닛 (222), 모션 보상 유닛 (224) 및 인트라-예측 유닛 (226)) 을 제어한다. 현재 블록의 인터-예측을 위해, 모션 추정 유닛 (222) 은 모션 탐색을 수행하여 하나 이상의 참조 픽처들 (예를 들어, DPB (218) 에 저장된 하나 이상의 이전에 코딩된 픽처들) 에서 하나 이상의 근접하게 매칭하는 참조 블록들을 식별할 수도 있다. 특히, 모션 추정 유닛 (222) 은, 예를 들어 절대차의 합 (SAD), 제곱차의 합 (SSD), 평균 절대차 (MAD), 평균 제곱차 (MSD) 등에 따라, 잠재적 참조 블록이 현재 블록에 얼마나 유사한지를 나타내는 값을 계산할 수도 있다. 모션 추정 유닛 (222) 은 일반적으로 고려되는 참조 블록과 현재 블록 사이의 샘플 별 차이들을 사용하여 이러한 계산들을 수행할 수도 있다. 모션 추정 유닛 (222) 은 현재 블록과 가장 근접하게 매칭하는 참조 블록을 표시하는, 이러한 계산들로부터 야기되는 최저 값을 갖는 참조 블록을 식별할 수도 있다.

[0385] 모션 추정 유닛 (222) 은 현재 픽처에서의 현재 블록의 포지션에 대한 참조 픽처들에서의 참조 블록들의 포지션들을 정의하는 하나 이상의 모션 벡터들 (MV들) 을 형성할 수도 있다. 모션 추정 유닛 (222) 은 그 후 모션 벡터들을 모션 보상 유닛 (224) 에 제공할 수도 있다. 예를 들어, 단방향 인터-예측에 대해, 모션 추정 유닛 (222) 은 단일 모션 벡터를 제공할 수도 있는 반면, 양방향 인터-예측에 대해, 모션 추정 유닛 (222) 은 2 개의 모션 벡터들을 제공할 수도 있다. 그 후, 모션 보상 유닛 (224) 은 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 예를 들어, 모션 보상 유닛 (224) 은 모션 벡터를 사용하여 참조 블록의 데이터를 취출할 수도 있다. 다른 예로서, 모션 벡터가 분수 샘플 정밀도를 갖는다면, 모션 보상 유닛 (224) 은 하나 이상의 보간 필터들에 따라 예측 블록에 대한 값들을 보간할 수도 있다. 또한, 양방향 인터-예측에 대해, 모션 보상 유닛 (224) 은 개개의 모션 벡터에 의해 식별된 2 개의 참조 블록들에 대한 데이터를 취출하고, 예를 들어 샘플 별 평균화 또는 가중된 평균화를 통해 취출된 데이터를 결합할 수도 있다.

[0386] 또 다른 예로서, 인트라-예측 또는 인트라-예측 코딩에 대해, 인트라-예측 유닛 (226) 은 현재 블록에 이웃하는 샘플들로부터 예측 블록을 생성할 수도 있다. 예를 들어, 방향성 모드들에 대해, 인트라-예측 유닛 (226)

은 일반적으로 이웃하는 샘플들의 값들을 수학적으로 결합하고 현재 블록에 걸쳐 정의된 방향에서 이들 계산된 값들을 파퓰레이트하여 예측 블록을 생성할 수도 있다. 또 다른 예로서, DC 모드에 대해, 인트라-예측 유닛 (226) 은 현재 블록에 대한 이웃하는 샘플들의 평균을 계산하고 예측 블록을 생성하여 예측 블록의 각각의 샘플에 대해 이러한 결과의 평균을 포함할 수도 있다.

[0387] 본 개시의 제 3 기법의 일 예에 따라, 인트라 예측 유닛 (226) 은 비디오 데이터의 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 유도 프로세스를 수행할 수도 있다. 이 예에서, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. MPM 도출 프로세스를 수행하는 것의 일부로서, 인트라 예측 유닛 (226) 은 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정할 수도 있다. 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 인트라 예측 유닛 (226) 은 이웃 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정할 수도 있다. 인트라 예측 유닛 (226) 은 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함할 수도 있다. 비디오 인코더 (200) 는 또 다른 인트라 예측 모드 또는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 인코딩할 수도 있다. 예를 들어, 현재 블록을 인코딩하는 비디오 인코더 (200) 의 일부로서, 인트라 예측 유닛 (226) 은 현재 블록에 대한 예측 블록을 생성하기 위해 또 다른 인트라 예측 모드 또는 MPM 중 하나를 사용할 수도 있다.

[0388] 모드 선택 유닛 (202) 은 예측 블록을 잔차 생성 유닛 (204) 에 제공한다. 잔차 생성 유닛 (204) 은 비디오 데이터 메모리 (230) 로부터의 현재 블록의 원시의, 코딩되지 않은 버전 및 모드 선택 유닛 (202) 으로부터의 예측 블록을 수신한다. 잔차 생성 유닛 (204) 은 현재 블록과 예측 블록 사이의 샘플 별 차이를 계산한다. 결과의 샘플 별 차이는 현재 블록에 대한 잔차 블록을 정의한다. 일부 예들에서, 잔차 생성 유닛 (204) 은 또한 잔차 차분 펄스 코드 변조 (residual differential pulse code modulation; RDPCM) 를 사용하여 잔차 블록을 생성하기 위해 잔차 블록에서의 샘플 값들 사이의 차이를 결정할 수도 있다. 일부 예들에서, 잔차 생성 유닛 (204) 은 이진 감산을 수행하는 하나 이상의 감산 회로들을 사용하여 형성될 수도 있다.

[0389] 모드 선택 유닛 (202) 이 CU들을 PU들로 파티셔닝하는 예들에서, 각각의 PU 는 루마 PU 및 대응하는 크로마 PU들과 연관될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다양한 사이즈를 갖는 PU들을 지원할 수도 있다. 위에 표시된 바와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있고 PU 의 사이즈는 PU 의 루마 예측 유닛의 사이즈를 지칭할 수도 있다. 특정 CU 의 사이즈가 2Nx2N 임을 가정하면, 비디오 인코더 (200) 는 인트라-예측을 위해 2Nx2N 또는 NxN 의 PU 사이즈들을 지원하고, 인터-예측을 위해 2Nx2N, 2NxN, Nx2N, NxN, 기타 등등의 대칭적인 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 또한, 인터 예측을 위해 2NxnU, 2NxnD, nLx2N, 및 nRx2N 의 PU 크기들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

[0390] 모드 선택 유닛이 CU 를 PU들로 추가로 파티셔닝하지 않는 예들에서, 각각의 CU 는 루마 코딩 블록 및 대응하는 크로마 코딩 블록들과 연관될 수도 있다. 위에서와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 2N×2N, 2N×N 또는 N×2N 의 CU 사이즈를 지원할 수도 있다.

[0391] 인트라-블록 카피 모드 코딩, 아핀 모드 코딩 및 선형 모델 (LM) 모드 코딩과 같은 다른 비디오 코딩 기법들에 대해, 몇몇 예들에서와 같이, 모드 선택 유닛 (202) 은 코딩 기술과 연관된 개개의 유닛들을 통해, 인코딩될 현재 블록에 대한 예측 블록을 생성한다. 팔레트 모드 코딩과 같은 일부 예에서, 모드 선택 유닛 (202) 은 예측 블록을 생성하지 않을 수도 있고, 대신에 선택된 팔레트에 기초하여 블록을 복원하는 방식을 표시하는 신택스 엘리먼트들을 생성할 수도 있다. 이러한 모드들에서, 모드 선택 유닛 (202) 은 이들 신택스 엘리먼트들을 인코딩될 엔트로피 인코딩 유닛 (220) 에 제공할 수도 있다.

[0392] 상술한 바와 같이, 잔차 생성 유닛 (204) 은 현재 블록 및 대응하는 예측 블록에 대해 비디오 데이터를 수신한다. 잔차 생성 유닛 (204) 은 그 후 현재 블록에 대한 잔차 블록을 생성한다. 잔차 블록을 생성하기 위해, 잔차 생성 유닛 (204) 은 현재 블록과 예측 블록 사이의 샘플 별 차이들을 계산한다.

[0393] 변환 프로세싱 유닛 (206) 은 잔차 블록에 하나 이상의 변환들을 적용하여 변환 계수들의 블록 (본 명세서에서는 "변환 계수 블록" 으로 지칭됨) 을 생성한다. 변환 프로세싱 유닛 (206) 은 다양한 변환들을 잔차 블록에 적용하여 변환 계수 블록을 형성할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (206) 은 이산 코사인 변환 (DCT), 방향성 변환, Karhunen-Loeve 변환 (KLT) 또는 개념적으로 유사한 변환을 잔차 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 대한 다중 변환들, 예를 들어 1차 변환 및 2차 변환, 예컨대 회전 변환을 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록

에 변환들을 적용하지 않는다.

[0394] 양자화 유닛 (208) 은 양자화된 변환 계수 블록을 생성하기 위해 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 유닛 (208) 은 현재 블록과 연관된 양자화 파라미터 (QP) 값에 따라 변환 계수 블록의 변환 계수들을 양자화할 수도 있다. 비디오 인코더 (202) 는 (예를 들어, 모드 선택 유닛 (202) 을 통해) CU 와 연관된 QP 값을 조정함으로써 현재 블록과 연관된 변환 계수 블록들에 적용된 양자화도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서 양자화된 변환 계수들은 변환 프로세싱 유닛 (206) 에 의해 생성된 원래의 변환 계수들보다 더 낮은 정확도를 가질 수도 있다.

[0395] 역 양자화 유닛 (210) 및 역 변환 프로세싱 유닛 (212) 은 각각 양자화된 변환 계수 블록에 역 양자화 및 역 변환들을 적용하여, 변환 계수 블록으로부터 잔차 블록을 복원할 수도 있다. 복원 유닛 (214) 은 모드 선택 유닛 (202) 에 의해 생성된 예측 블록 및 복원된 잔차 블록에 기초하여 (잠재적으로 어느 정도의 왜곡을 가짐에도 불구하고) 현재 블록에 대응하는 복원된 블록을 생성할 수도 있다 . 예를 들어, 복원 유닛 (214) 은 복원된 잔차 블록의 샘플들을, 모드 선택 유닛 (202) 에 의해 생성된 예측 블록으로부터의 대응하는 샘플들에 가산하여 복원된 블록을 생성할 수도 있다.

[0396] 필터 유닛 (216) 은 복원된 블록에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (216) 은 CU들의 에지들을 따라 블록크니스 아티팩트 (blockiness artifacts) 를 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (216) 의 동작들은 일부 예들에서 스킵될 수도 있다.

[0397] 비디오 인코더 (200) 는 DPB (218) 에 복원된 블록들을 저장한다. 예를 들어, 필터 유닛 (312) 의 동작들이 수행되지 않은 예들에서, 복원 유닛 (310) 은 복원된 블록들을 DPB (314) 에 저장할 수도 있다. 필터 유닛 (312) 의 동작들이 수행되는 예들에서, 필터 유닛 (312) 은 필터링된 복원된 블록들을 DPB (314) 에 저장할 수도 있다. 예를 들어, 필터 유닛 (216) 의 동작들이 필요하지 않은 예들에서, 복원 유닛 (214) 은 복원된 블록들을 DPB (218) 에 저장할 수도 있다. 필터 유닛 (216) 의 동작들이 필요한 예들에서, 필터 유닛 (216) 은 필터링된 복원된 블록들을 DPB (218) 에 저장할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 복원된 (및 잠재적으로 필터링된) 블록들로부터 형성된 DPB (218) 로부터 참조 픽처를 취출하여, 후속하여 인코딩된 픽처들의 블록들을 인터-예측할 수도 있다. 또한, 인트라-예측 유닛 (226) 은 현재 픽처에서의 다른 블록들을 인트라-예측하기 위해 현재 픽처의 DPB (218) 에서 복원된 블록들을 사용할 수도 있다.

[0398] 일반적으로, 엔트로피 인코딩 유닛 (220) 은 비디오 인코더 (200) 의 다른 기능성 컴포넌트들로부터 취출된 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 양자화 유닛 (208) 으로부터 양자화된 변환 계수 블록들을 엔트로피 인코딩할 수도 있다. 또 다른 예로서, 엔트로피 인코딩 유닛 (220) 은 모드 선택 유닛 (202) 으로부터 예측 신택스 엘리먼트들 (예를 들어, 인트라-예측에 대한 인트라-모드 정보 또는 인터-예측에 대한 모션 정보) 를 엔트로피 인코딩할 수도 있다. 엔트로피 인코딩 유닛 (220) 은 엔트로피 인코딩된 데이터를 생성하기 위해, 비디오 데이터의 또 다른 예인, 신택스 엘리먼트들에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 컨텍스트 적응 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 신택스 기반 컨텍스트 적응 바이너리 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-골롬 인코딩 동작, 또는 다른 타입의 엔트로피 인코딩 동작을 데이터에 대해 수행할 수도 있다. 일부 예들에서, 엔트로피 인코딩 유닛 (220) 은 신택스 엘리먼트들이 엔트로피 인코딩되지 않은 바이패스 모드에서 동작할 수도 있다.

[0399] 비디오 인코더 (200) 는 픽처 또는 슬라이스의 블록들을 복원하는데 필요한 엔트로피 인코딩된 신택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 특히, 엔트로피 인코딩 유닛 (220) 이 비트스트림을 출력할 수도 있다.

[0400] 상술한 동작들은 블록과 관련하여 설명된다. 이러한 설명은 루마 코딩 블록 및/또는 크로마 코딩 블록들에 대한 동작들인 것으로 이해되어야 한다. 상술한 바와 같이, 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 CU 의 루마 및 크로마 성분들이다. 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 PU 의 루마 및 크로마 성분들이다.

[0401] 일부 예들에서, 루마 코딩 블록에 대해 수행되는 동작들은 크로마 코딩 블록에 대해 반복될 필요가 없다. 하나의 예로서, 크로마 블록들에 대한 모션 벡터 (MV) 및 참조 픽처를 식별하기 위해 루마 코딩 블록에 대한 MV 및 참조 픽처를 식별하는 동작들이 반복될 필요는 없다. 오히려, 루마 코딩 블록에 대한 MV 는 크로마 블록

들에 대한 MV 를 결정하도록 스케일링될 수도 있고, 참조 픽처는 동일할 수도 있다. 다른 예로서, 인트라-예측 프로세스는 루마 코딩 블록들 및 크로마 코딩 블록들에 대해 동일할 수도 있다.

[0402] 비디오 인코더 (200) 는 비디오 데이터를 저장하도록 구성된 메모리, 및 회로부에서 구현되고 본 개시의 ALWIP 기법들을 수행하도록 구성된 하나 이상의 프로세싱 유닛들을 포함하는 비디오 데이터를 인코딩하도록 구성된 디바이스의 예를 나타낸다. 예를 들어, 비디오 인코더 (200) 는 비디오 데이터를 저장하도록 구성된 메모리 및 회로부에서 구현되고 비디오 데이터의 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 도출 프로세스를 수행하도록 구성된 하나 이상의 프로세싱 유닛들을 포함하는 디바이스의 예를 나타낼 수도 있으며, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. 이 예에서, 하나 이상의 프로세서들은 MPM 도출 프로세스를 수행하는 것의 일부로서, 하나 이상의 프로세서들이 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하도록 구성된다. 또한, 하나 이상의 프로세서들은 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 이웃 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정하도록 구성된다.

하나 이상의 프로세서들은 또한 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함하도록 구성될 수도 있다. 하나 이상의 프로세서들은 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 인코딩할 수도 있다.

[0403] 게다가, 블록이 DM 모드로 코딩된 크로마 블록인 일부 예들에서, 비디오 인코더 (200) 는 비디오 데이터를 저장하도록 구성된 메모리 및 병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정하도록 구성된 하나 이상의 프로세서들을 포함하는 디바이스의 예를 나타낸다. 이러한 예들에서, 병치된 루마 블록은 블록과 함께 병치된다. 또한, 이러한 예들에서, 하나 이상의 프로세서들은 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여, 병치된 루마 블록의 인트라 예측 모드가 디폴트 값이라고 결정, 예를 들어 추론할 수도 있다. 즉, 하나 이상의 프로세서들은 병치된 루마 블록의 인트라 예측 모드가 평면 인트라 예측 모드와 같은 디폴트 인트라 예측 모드인 것으로 결정하거나 그렇지 않으면 이를 처리할 수도 있다. 하나 이상의 프로세서들은 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 블록을 인코딩하도록 구성될 수도 있다.

[0404] 도 16 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더 (300) 를 도시하는 블록 다이어그램이다. 도 16 은 설명의 목적을 위해 제공되고 본 개시에 폭넓게 예시되고 설명된 기법들에 대해 제한하지 않는다. 설명의 목적으로, 본 개시는 JEM, VVC 및 HEVC 의 기법들에 따른 비디오 디코더 (300) 를 기재한다.

그러나, 본 개시의 기법들은 다른 비디오 코딩 표준들로 구성되는 비디오 코딩 디바이스들에 의해 수행될 수도 있다.

[0405] 도 16 의 예에서, 비디오 디코더 (300) 는 코딩된 픽처 버퍼 (CPB)(320), 엔트로피 디코딩 유닛 (302), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (310), 복원 유닛 (310), 필터 유닛 (312), 및 디코딩된 픽처 버퍼 (DPB)(314) 를 포함한다. CBP 메모리 (230), 엔트로피 디코딩 유닛 (320), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 복원 유닛 (310), 필터 유닛 (216) 및 DPB (314) 중 임의의 것 또는 전부가 하나 이상의 프로세서들에서 또는 프로세싱 회로부에서 구현될 수도 있다. 더욱이, 비디오 디코더 (300) 는 이들 및 다른 기능들을 수행하기 위해 부가 또는 대안의 프로세서들 또는 프로세싱 회로부를 포함할 수도 있다.

[0406] 예측 프로세싱 유닛 (304) 은 모션 보상 유닛 (316) 및 인트라-예측 유닛 (318) 을 포함한다. 예측 프로세싱 유닛 (304) 은 다른 예측 모드들에 따라 예측을 수행하기 위해 부가 유닛들을 포함할 수도 있다. 예를 들어, 예측 프로세싱 유닛 (304) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 보상 유닛 (316) 의 일부를 형성할 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다. 예를 들어, 도 16 의 예에서, 인트라 예측 유닛 (318) 은 ALWIP 프로세스를 수행하는 ALWIP 유닛 (319) 을 포함한다. 다른 예들에서, 비디오 디코더 (300) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0407] CPB 메모리 (320) 는 비디오 디코더 (300) 의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. CPB 메모리 (320) 에 저장된 비디오 데이터는, 예를 들어 컴퓨터 판독가능 매체 (110)(도 1) 로부터 획득될 수도 있다. CPB 메모리 (320) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터 (예를 들어, 신택스 엘리먼트들) 를 저장하는 CPB 를 포함할 수도 있다. 또한, CPB 메모리 (320) 는 비디오 디코더 (300) 의 다양한 유닛들로부터의 출력들을 나타내는 일시적인 데이터와 같은, 코딩된 픽처의 신택스 엘리먼트들 이외의 비디오 데이터를 저장할 수도 있다. DPB (314) 는 일반적으로, 인코딩된 비디오 비트스트림의 후속 데이터 또는 픽처들을 디코딩할 때, 참조 비디오 데이터로서 비디오 디코더 (300) 가 출력하고 및/또는 사용할 수도 있는 디코딩된 픽처들을 저장한다. CPB 메모리

(320) 및 DPB (314) 는 다양한 메모리 디바이스들, 예컨대 동기식 동적 랜덤 액세스 메모리 (SDRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM) 을 포함하는 DRAM, 또는 다른 타입들의 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. CPB 메모리 (320) 및 DPB (314) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, CPB 메모리 (320) 는 비디오 디코더 (300) 의 다른 컴포넌트들과 온-칩이거나 그 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0408] 부가적으로 또는 대안으로, 일부 예들에서, 비디오 디코더 (300) 는 메모리 (120)(도 1) 로부터 코딩된 비디오 데이터를 취출할 수도 있다. 즉, 메모리 (120) 는 CPB 메모리 (320) 로 위에서 논의된 바와 같이 데이터를 저장할 수도 있다. 마찬가지로, 메모리 (120) 는 비디오 디코더 (300) 의 기능성의 일부 또는 전부가 비디오 디코더 (300) 의 프로세싱 회로부에 의해 실행되는 소프트웨어에서 구현될 때, 비디오 디코더 (300) 에 의해 실행될 명령들을 저장할 수도 있다.

[0409] 도 16 에 나타낸 다양한 유닛들은 비디오 인코더 (300) 에 의해 수행되는 동작들의 이해를 돕기 위해 도시된다. 이 유닛들은 고정 기능 회로들, 프로그램가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 도 15 와 유사하게, 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하며, 수행될 수 있는 동작들에 대해 미리설정된다. 프로그램가능 회로들은 다양한 태스크들을 수행하도록 프로그램될 수 있는 회로들을 지칭하고, 수행될 수 있는 동작들에서 유연한 기능성을 제공한다. 예를 들어, 프로그램가능 회로들은, 프로그램가능 회로들이 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 소프트웨어 명령들을 (예를 들어, 파라미터들을 수신하거나 파라미터들을 출력하기 위해) 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작 타입들은 일반적으로 불변이다. 일부 예들에서, 유닛들 중 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그램가능) 일 수도 있고, 일부 예들에 있어서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.

[0410] 비디오 디코더 (300) 는 프로그램가능 회로들로부터 형성된, ALU들, EFU들, 디지털 회로들, 아날로그 회로들, 및/또는 프로그램가능 코어들을 포함할 수도 있다. 비디오 디코더 (300) 의 동작들이 프로그램가능 회로들 상에서 실행하는 소프트웨어에 의해 수행되는 예들에서, 온-칩 또는 오프-칩 메모리는 비디오 디코더 (300) 가 수신하고 실행하는 소프트웨어의 명령들 (예를 들어, 오브젝트 코드) 을 저장할 수도 있다.

[0411] 엔트로피 디코딩 유닛 (302) 은 인코딩된 비디오 데이터를 CPB 로부터 수신하고, 그 비디오 데이터를 엔트로피 디코딩하여 신택스 엘리먼트들을 재생할 수도 있다. 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 복원 유닛 (310), 및 필터 유닛 (312) 은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0412] 일반적으로, 비디오 디코더 (300) 는 블록 별 (block-by-block ) 단위로 픽처를 복원한다. 비디오 디코더 (300) 는 개별적으로 (현재 복원되고 있는, 즉 디코딩되는 블록이 "현재 블록" 으로 지칭될 수도 있는 경우) 각각의 블록에 대해 복원 동작을 수행할 수도 있다.

[0413] 엔트로피 디코딩 유닛 (302) 은 양자화 파라미터 (QP) 및/또는 변환 모드 표시(들)과 같은 변환 정보 뿐만 아니라, 양자화된 변환 계수 블록의 양자화된 변환 계수들을 정의하는 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 역 양자화 유닛 (306) 은 양자화된 변환 계수 블록과 연관된 QP 를 사용하여, 양자화도 및 유사하게, 적용할 역 양자화 유닛 (306) 에 대한 역 양자화도를 결정할 수도 있다. 역 양자화 유닛 (306) 은 예를 들어, 양자화된 변환 계수들을 역 양자화하기 위해 비트단위 (bitwise) 좌측-시프트 동작을 수행할 수도 있다. 따라서, 역 양자화 유닛 (306) 은 변환 계수들을 포함하는 변환 계수 블록을 형성할 수도 있다.

[0414] 역 양자화 유닛 (306) 이 변환 계수 블록을 형성한 후, 역 변환 프로세싱 유닛 (308) 은 현재 블록과 연관된 잔차 블록을 생성하기 위해 변환 계수 블록에 하나 이상의 역 변환들을 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (308) 은 역 DCT, 역 정수 변환, 역 Karhunen-Loeve 변환 (KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 변환 계수 블록에 적용할 수도 있다.

[0415] 또한, 예측 프로세싱 유닛 (304) 은 엔트로피 디코딩 유닛 (302) 에 의해 엔트로피 디코딩된 예측 정보 신택스 엘리먼트들에 따라 예측 블록을 생성한다. 예를 들어, 예측 정보 신택스 엘리먼트들이 현재 블록이 인터-예측된 것을 표시하면, 모션 보상 유닛 (316) 은 예측 블록을 생성할 수도 있다. 이 경우, 예측 정보 신택스 엘리먼트들은 참조 블록을 취출할 DPB (314) 에서의 참조 픽처뿐만 아니라 현재 픽처에서의 현재 블록의 위치에 대한 참조 픽처에서의 참조 블록의 위치를 식별하는 모션 벡터를 표시할 수도 있다. 모션 보상 유닛 (316) 은 일반적으로 모션 보상 유닛 (224)(도 15) 과 관련하여 설명된 것과 실질적으로 유사한 방식으로 인터-예측

프로세스를 수행할 수도 있다.

[0416] 다른 예로서, 예측 정보 신택스 엘리먼트가 현재 블록이 인트라-예측되는 것을 표시하면, 인트라-예측 유닛 (318) 은 예측 정보 신택스 엘리먼트들에 의해 표시된 인트라-예측 모드에 따라 예측 블록을 생성할 수도 있다. 다시, 인트라-예측 유닛 (318) 은 일반적으로 인트라-예측 유닛 (226)(도 15) 과 관련하여 설명된 것과 실질적으로 유사한 방식으로 인트라-예측 프로세스를 수행할 수도 있다. 인트라-예측 유닛 (318) 은 DPB (314) 로부터 현재 블록에 이웃하는 샘플들의 데이터를 취출할 수도 있다.

[0417] 본 개시의 제 3 기법의 일 예에 따라, 인트라 예측 유닛 (318) 은 비디오 데이터의 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 유도 프로세스를 수행할 수도 있다. 이 예에서, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. MPM 도출 프로세스를 수행하는 것의 일부로서, 인트라 예측 유닛 (318) 은 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정할 수도 있다. 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 인트라 예측 유닛 (318) 은 이웃 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정할 수도 있다. 인트라 예측 유닛 (318) 은 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함할 수도 있다. 비디오 디코더 (300) 는 또 다른 인트라 예측 모드 또는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 디코딩할 수도 있다. 예를 들어, 현재 블록을 코딩하는 비디오 디코더 (300) 의 일부로서, 인트라 예측 유닛 (318) 은 현재 블록에 대한 예측 블록을 생성하기 위해 또 다른 인트라 예측 모드 또는 MPM들 중 하나를 사용할 수도 있다.

[0418] 복원 유닛 (310) 은 예측 블록 및 잔차 블록을 사용하여 현재 블록을 복원한다. 예를 들어, 복원 유닛 (310) 은 잔차 픽셀 블록의 샘플들을 예측 블록의 대응하는 샘플들에 가산하여 현재 블록을 복원할 수도 있다.

[0419] 필터 유닛 (312) 은 복원된 블록에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (312) 은 복원된 블록들의 에지들을 따라 블록크니스 아티팩트를 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (312) 의 동작들이 모든 예들에서 반드시 수행되지는 않는다.

[0420] 비디오 디코더 (300) 는 DPB (314) 에 복원된 블록들을 저장할 수도 있다. 위에 논의된 바와 같이, DPB (314) 는 예측 프로세싱 유닛 (304) 에 인트라-예측을 위한 현재 픽처의 샘플들 및 후속 모션 보상을 위해 이전에 디코딩된 픽처들와 같은 참조 정보를 제공할 수도 있다. 또한, 비디오 디코더 (300) 는 도 1 의 디스플레이 디바이스 (118) 와 같은 디스플레이 디바이스 상으로의 후속 프리젠테이션을 위해 DPB 로부터 디코딩된 픽처들을 출력할 수도 있다.

[0421] 이러한 방식으로, 비디오 디코더 (300) 는 비디오 데이터를 저장하도록 구성된 메모리, 및 회로부에서 구현되고 본 개시의 ALWIP 기법들을 수행하도록 구성된 하나 이상의 프로세싱 유닛들을 포함하는 비디오 디코딩 디바이스의 예를 나타낸다. 예를 들어, 비디오 디코더 (300) 는 비디오 데이터를 저장하도록 구성된 메모리 및 회로부에서 구현되고 비디오 데이터의 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 도출 프로세스를 수행하도록 구성된 하나 이상의 프로세싱 유닛들을 포함하는 디바이스의 예를 나타낼 수도 있으며, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. 이 예에서, 하나 이상의 프로세서들은 MPM 도출 프로세스를 수행하는 것의 일부로서, 하나 이상의 프로세서들이 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하도록 구성된다. 또한, 하나 이상의 프로세서들은 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 이웃 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정하도록 구성된다. 하나 이상의 프로세서들은 또한 하나 이상의 MPM들 중 하나로서 디폴트에 대응하는 인트라 예측 모드를 포함할 수도 있다. 하나 이상의 프로세서들은 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 디코딩할 수도 있다.

[0422] 게다가, 블록이 DM 모드로 코딩된 크로마 블록인 일부 예들에서, 비디오 인코더 (200) 는 비디오 데이터를 저장하도록 구성된 메모리 및 병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정하도록 구성된 하나 이상의 프로세서들을 포함하는 디바이스의 예를 나타낸다. 이러한 예들에서, 병치된 루마 블록은 블록과 함께 병치된다. 또한, 이러한 예들에서, 하나 이상의 프로세서들은 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여, 병치된 루마 블록의 인트라 예측 모드가 디폴트 값이라고 결정, 예를 들어 추론할 수도 있다. 즉, 하나 이상의 프로세서들은 병치된 루마 블록의 인트라 예측 모드가 평면 인트라 예측 모드와 같은 디폴트 인트라 예측 모드인 것으로 결정하거나 그렇지 않으면 이를 처리할 수도 있다. 하나 이상의 프로세서들은 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 블록을 디코딩하도록 구성될 수도 있다.

[0423] 도 17 은 현재 블록을 인코딩하기 위한 예시의 방법을 도시하는 플로우챠트이다. 현재 블록은 현재 CU 를

포함할 수도 있다. 비디오 인코더 (200)(도 1 및 도 15) 와 관련하여 설명되지만, 도 17 과 유사한 방법을 수행하도록 다른 디바이스들이 구성될 수도 있음을 이해해야 한다.

[0424] 본 예에서, 비디오 인코더 (200) 는 초기에 현재 블록을 예측한다 (350). 예를 들어, 비디오 인코더 (200) 는 현재 블록에 대한 예측 블록을 형성할 수도 있다. 일부 예들에서, 비디오 인코더 (200)(예를 들어, 비디오 인코더 (200) 의 인트라 예측 유닛 (226)) 는 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 도출 프로세스를 수행한다. 이 예에서, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. MPM들은 인트라 예측 모드를 포함할 수도 있고, 일부 예들에서, MPM들의 각각은 인트라 예측 모드이다. 게다가, 이 예에서, MPM 도출 프로세스를 수행하는 것의 일부로서, 비디오 인코더 (200) 는 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정할 수도 있고, 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정한다. 비디오 인코더 (200) 는 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함할 수도 있다. 비디오 인코더 (200) 는 그 후 현재 블록에 대한 예측 블록을 형성하기 위해 또 다른 인트라 예측 모드 또는 MPM들 중 하나를 사용할 수도 있다.

[0425] 비디오 인코더 (200) 는 그 후 현재 블록에 대한 잔차 블록을 계산할 수도 있다 (352). 잔차 블록을 계산하기 위해, 비디오 인코더 (200) 는 원래의, 인코딩되지 않은 블록과 현재 블록에 대한 예측 블록 사이의 차이를 계산할 수도 있다. 비디오 인코더 (200) 는 그 후 잔차 블록의 변환 계수들을 변환하고 양자화할 수도 있다 (354).

[0426] 다음으로, 비디오 인코더 (200) 는 잔차 블록의 양자화된 변환 계수들을 스캔할 수도 있다 (356). 스캔 동안 또는 스캔에 후속하여, 비디오 인코더 (200) 는 변환 계수들을 엔트로피 인코딩할 수도 있다 (358). 예를 들어, 비디오 인코더 (200) 는 CAVLC 또는 CABAC 를 사용하여 변환 계수들을 인코딩할 수도 있다. 비디오 인코더 (200) 는 그 후 블록의 엔트로피 인코딩된 데이터를 출력할 수도 있다 (360).

[0427] 도 18 은 비디오 데이터의 현재 블록을 디코딩하기 위한 예시의 방법을 도시하는 플로우챠트이다. 현재 블록은 현재 CU 를 포함할 수도 있다. 비디오 디코더 (300)(도 1 및 도 16) 와 관련하여 설명되지만, 도 18 과 유사한 방법을 수행하도록 다른 디바이스들이 구성될 수도 있음을 이해해야 한다.

[0428] 비디오 디코더 (300) 는 현재 블록에 대한 엔트로피 코딩된 데이터, 이를 테면, 엔트로피 코딩된 예측 정보 및 현재 블록에 대응하는 잔차 블록의 변환 계수들에 대한 엔트로피 코딩된 데이터를 수신할 수도 있다 (370). 비디오 디코더 (300) 는 현재 블록에 대한 예측 정보를 결정하고 잔차 블록의 변환 계수들을 재생하기 위해 엔트로피 코딩된 데이터를 엔트로피 디코딩할 수도 있다 (372).

[0429] 비디오 디코더 (300) 는 현재 블록에 대한 예측 블록을 계산하기 위해, 예를 들어 현재 블록에 대한 예측 정보에 의해 표시된 바와 같이 인트라- 또는 인터-예측 모드를 사용하여, 현재 블록을 예측할 수도 있다 (374). 일부 예들에서, 비디오 디코더 (300)(예를 들어, 비디오 디코더 (300) 의 인트라 예측 유닛 (318)) 는 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 도출 프로세스를 수행한다. 이 예에서, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. MPM들은 인트라 예측 모드를 포함할 수도 있고, 일부 예들에서, MPM들의 각각은 인트라 예측 모드이다. 게다가, 이 예에서, MPM 도출 프로세스를 수행하는 것의 일부로서, 비디오 디코더 (300) 는 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정할 수도 있고, 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 이웃 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정한다. 비디오 디코더 (300) 는 하나 이상의 MPM들 중 하나로서 디폴트 값 (예를 들어, 평면 모드) 에 대응하는 인트라 예측 모드를 포함할 수도 있다. 비디오 디코더 (300) 는 그 후 현재 블록에 대한 예측 블록을 형성하기 위해 또 다른 인트라 예측 모드 또는 MPM들 중 하나를 사용할 수도 있다.

[0430] 비디오 디코더 (300) 는 양자화된 변환 계수들의 블록을 생성하기 위해 복원된 변환 계수들을 역 스캔할 수도 있다 (376). 비디오 디코더 (300) 는 그 후 변환 계수들을 역 양자화하고 변환 계수들에 하나 이상의 역 변환들을 적용하여 잔차 블록을 생성할 수도 있다 (378). 비디오 디코더 (300) 는 예측 블록 및 잔차 블록을 조합함으로써 현재 블록을 디코딩할 수도 있다 (380).

[0431] 도 19 는 본 개시의 하나 이상의 기법들에 따른, 비디오 데이터를 코딩하는 예시의 방법을 도시하는 플로우챠트이다. 도 19 의 방법은 비디오 코더, 예컨대 비디오 인코더 (200) 또는 비디오 디코더 (300) 에 의해 수행될 수도 있다.

[0432] 도 19 의 예에서, 비디오 코더는 비디오 데이터의 현재 블록에 대한 하나 이상의 MPM들을 도출하는 MPM 도출 프로세스를 수행할 수도 있다 (400). 일부 예들에서, 비디오 인코더 (200) 의 인트라 예측 유닛 (226)(도 15)

은 MPM 도출 프로세스를 수행한다. 일부 예들에서, 비디오 디코더 (300) 의 인트라 예측 유닛 (318)(도 16) 은 MPM 도출 프로세스를 수행한다. 도 19 의 예에서, 현재 블록은 ALWIP 를 사용하여 코딩되지 않는다. MPM들은 인트라 예측 모드를 포함할 수도 있고, 일부 예들에서, MPM들의 각각은 인트라 예측 모드이다. 비디오 코더가 비디오 인코더 (예를 들어, 비디오 인코더 (200)) 인 예들에서, 비디오 인코더는 현재 블록을 포함하는 픽처를 캡처하는 카메라를 포함하는 디바이스 (예를 들어, 소스 디바이스(102)) 에 포함될 수도 있다.

[0433] 또한, 도 19 의 예에서, MPM 도출 프로세스를 수행하는 것은 비디오 코더가 현재 블록의 이웃 블록이 ALWIP-코딩된 이웃 블록인지 여부를 결정하는 것을 포함한다 (402). 이웃 블록은 현재 블록의 상측 이웃 블록 또는 좌측 이웃 블록일 수도 있다. 일부 예들에서, 비디오 코더는 현재 픽처의 블록들에 대한 alwip_flags 를 저장할 수도 있고 이웃 블록이 ALWIP 를 사용하여 코딩되는지 여부를 결정하기 위해 이웃 블록의 저장된 alwip_flags 를 사용할 수도 있다.

[0434] 이웃 블록이 ALWIP-코딩된 이웃 블록인 것에 기초하여, 비디오 코더는 이웃 블록의 인트라 예측 모드의 값이 디폴트 값이라고 결정할 수도 있다 (404). 예를 들어, 비디오 코더는 이웃 블록의 인트라 모드의 값이 평면 모드를 표시하는 특정 값인 것으로 결정 (예를 들어, 추론) 할 수도 있다. 즉, 디폴트 값에 대응하는 인트라 예측 모드는 평면 모드일 수도 있다. 다른 예들에서, 디폴트 값에 대응하는 인트라 예측 모드는 또 다른 인트라 예측 모드일 수도 있다. 비디오 코더 (200) 는 그 후 현재 블록에 대한 하나 이상의 MPM들 중 하나로서 디폴트 값에 대응하는 인트라 예측 모드를 포함할 수도 있다 (406).

[0435] 비디오 코더는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 코딩할 수도 있다 (408). 예를 들어, 비디오 코더가 비디오 인코더 (200) 와 같은 비디오 인코더인 예들에서, 비디오 인코더는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 인코딩할 수도 있다. 그러한 예들에서, 비디오 인코더는 현재 블록에 대한 예측 블록을 생성하기 위해 MPM 을 사용할 수도 있다. 그 후 비디오 인코더는 현재 블록에 대한 잔차 데이터를 생성하기 위해 예측 블록 및 현재 블록의 샘플들을 사용할 수도 있다. 비디오 인코더는 그 후 예를 들어, 도 15 와 관련하여 본 개시물의 다른 곳에서 설명된 바와 같이 현재 블록에 대한 잔차 데이터를 프로세싱할 수도 있다. 비디오 코더가 비디오 디코더 (300) 와 같은 비디오 디코더인 예들에서, 비디오 디코더는 현재 블록에 대한 MPM들 중 하나에 기초하여 현재 블록을 디코딩할 수도 있다. 그러한 예들에서, 비디오 디코더는 현재 블록에 대한 예측 블록을 생성하기 위해 MPM 을 사용할 수도 있다. 비디오 디코더는 현재 블록에 대한 잔차 데이터 및 예측 블록을 사용하여 현재 블록을 복원할 수도 있다. 게다가, 일부 예들에서, 비디오 디코더를 포함하는 디바이스는 현재 블록을 포함하는 픽처를 디스플레이하는 디스플레이를 포함할 수도 있다.

[0436] 도 20 은 본 개시의 하나 이상의 기법들에 따른, 크롬 블록을 코딩하는 예시의 방법을 도시하는 플로우챠트이다. 도 20 의 예에서, 비디오 코더 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더 (300)) 는 DM 모드를 사용하여 크로마 블록을 코딩하고 있을 수도 있다.

[0437] 특히, 비디오 코더는 병치된 루마 블록이 ALWIP 로 코딩되는지 여부를 결정할 수도 있다 (450). 병치된 루마 블록은 크로마 블록과 함께 병치된다. 병치된 루마 블록이 ALWIP 로 코딩되는 것에 기초하여, 비디오 코더는 병치된 루마 블록의 인트라 예측 모드의 값이 평면 모드에 대응하는 값과 같은, 디폴트 값이라고 결정할 수도 있다.

[0438] 그 후 비디오 코더는 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 크로마 블록을 코딩할 수도 있다 (454). 예를 들어, 비디오 코더가 비디오 인코더 (200) 와 같은 비디오 인코더인 예들에서, 비디오 인코더는 디폴트 값에 대응하는 인트라 예측 모드를 사용하여 크로마 블록에 대한 예측 블록을 생성할 수도 있다. 그 후 비디오 인코더는 크로마 블록에 대한 잔차 데이터를 생성하기 위해 크로마 블록의 샘플들 및 예측 블록을 사용할 수도 있다. 비디오 인코더는 그 후 예를 들어, 도 15 와 관련하여 본 개시의 다른 곳에서 설명된 바와 같이 크로마 블록에 대한 잔차 데이터를 프로세싱할 수도 있다. 비디오 코더가 비디오 디코더 (300) 와 같은 비디오 디코더인 예들에서, 비디오 디코더는 디폴트 값에 대응하는 인트라 예측 모드에 기초하여 크로마 블록을 디코딩할 수도 있다. 그러한 예들에서, 비디오 디코더는 크로마 블록에 대한 예측 블록을 생성하기 위해 디폴트 값에 대응하는 인트라 예측 모드를 사용할 수도 있다. 비디오 디코더는 그 후 크로마 블록에 대한 잔차 데이터 및 예측 블록을 사용하여 크로마 블록을 복원할 수도 있다. 게다가, 일부 예들에서, 비디오 디코더를 포함하는 디바이스는 크로마 블록을 포함하는 픽처를 디스플레이하는 디스플레이를 포함할 수도 있다.

[0439] 다음의 단락들은 본 개시의 하나 이상의 기법들에 따른 예들의 비제한적 세트이다.

[0440] 실시예 1. 비디오 데이터를 코딩하는 방법으로서, 방법은 현재 블록이 아핀 선형 가중 인트라 예측 (ALWIP) 로 코딩되는지 여부를 표시하는 ALWIP 신택스 엘리먼트에 대한 컨텍스트를 결정하는 단계로서, 상측 이웃 블록은 상측 이웃 블록이 현재 블록과 상이한 코딩 트리 유닛 (CTU) 에 있는 것에 기초하여 컨텍스트를 결정하는데 이용불가능한 것으로 결정되며, 상측 이웃 블록은 현재 블록의 상측 이웃인, 상기 컨텍스트를 결정하는 단계; 및 컨텍스트에 기초하여 ALWIP 신택스 엘리먼트를 코딩하는 단계를 포함한다.

[0441] 실시예 2. 비디오 데이터를 코딩하는 방법으로서, 방법은 비디오 데이터의 블록들의 세트를 코딩하는 단계를 포함하며, 임계치 폭보다 작은 폭 또는 임계치 높이보다 작은 높이를 갖는 블록들에 대한 것보다 아핀 선형 가중 인트라 예측 (ALWIP) 을 디스에이블하는 제약이 코딩에 부과된다.

[0442] 실시예 3. 비디오 데이터를 코딩하는 방법으로서, 방법은 비디오 데이터의 현재 블록에 대한 이웃 블록은 그 이웃 블록이 아핀 선형 가중 인트라 예측 (ALWIP) 코딩되고 그 이웃 블록이 비디오 데이터의 비-ALWIP 코딩된 블록의 최확 모드 (MPM) 도출로 고려되는 것에 기초하여 이용불가능한 것으로 간주된다고 결정하는 단계; 이용가능한 이웃 블록들에 기초하여 현재 블록에 대한 MPM들을 결정하는 단계; 및 MPM들 중 하나에 기초하여 현재 블록을 코딩하는 단계를 포함한다.

[0443] 실시예 4. 비디오 데이터를 코딩하는 방법으로서, 방법은 비디오 데이터의 현재 블록에 대한 이웃 블록은 그 이웃 블록이 아핀 선형 가중 인트라 예측 (ALWIP) 코딩되고 그 이웃 블록이 비디오 데이터의 ALWIP 코딩된 블록의 최확 모드 (MPM) 도출에서 사용되는 것에 기초하여 이용불가능한 것으로 간주된다고 결정하는 단계; 및 현재 블록을 코딩하기 위해 ALWIP 를 사용하는 단계를 포함한다.

[0444] 실시예 5. 비디오 데이터를 코딩하는 방법으로서, 방법은 비디오 데이터의 현재 블록의 이웃에 있는 하나 이상의 블록들에 기초하여 아핀 선형 가중 인트라 예측 (ALWIP) 을 위한 최확 모드 (MPM) 리스트를 도출하는 단계; 및 현재 블록을 코딩하기 위해 ALWIP 를 수행하도록 MPM 리스트로부터 MPM 을 사용하는 단계를 포함한다.

[0445] 실시예 6. 실시예 5 의 방법에서는, 현재 블록의 이웃에 있는 하나 이상의 블록들이 현재 블록의 상단 우측 샘플 위에 있고 이에 인접하는 제 1 이웃 블록 및 현재 블록의 하단 좌측 샘플의 좌측에 있고 이에 인접하는 제 1 이웃 블록을 포함한다.

[0446] 실시예 7. 방법은 실시예들 1 내지 6 중 하나 이상의 임의의 조합의 방법들을 포함한다.

[0447] 실시예 8. 실시예들 1 내지 7 중 임의의 것의 방법에서는, 코딩하는 단계가 디코딩하는 단계를 포함한다.

[0448] 실시예 9. 실시예들 1 내지 7 중 임의의 것의 방법에서는, 코딩하는 단계가 인코딩하는 단계를 포함한다.

[0449] 실시예 10. 비디오 데이터를 코딩하기 위한 디바이스로서, 디바이스는 실시예들 1 내지 9 중 임의의 것의 방법을 수행하기 위한 하나 이상의 수단들을 포함한다.

[0450] 실시예 11. 실시예 10 의 디바이스에서는, 하나 이상의 수단들이 회로부에서 구현된 하나 이상의 프로세서들을 포함한다.

[0451] 실시예 12. 실시예들 10 및 11 중 임의의 것의 디바이스는 비디오 데이터를 저장하기 위한 메모리를 더 포함한다.

[0452] 실시예 13. 실시예들 10 내지 12 중 임의의 것의 디바이스는 디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함한다.

[0453] 실시예 14. 실시예들 10 내지 13 중 임의의 것의 디바이스에서는, 디바이스가 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스 또는 셋톱 박스 중 하나 이상을 포함한다.

[0454] 실시예 15. 실시예들 10 내지 14 중 임의의 것의 디바이스에서는, 디바이스가 비디오 디코더를 포함한다.

[0455] 실시예 16. 실시예들 10 내지 15 중 임의의 것의 디바이스에서는, 디바이스가 비디오 인코더를 포함한다.

[0456] 실시예 17. 실행될 때 하나 이상의 프로세서들로 하여금 실시예들 1 내지 7 중 임의의 것의 방법을 수행하게 하는 명령들이 저장된 컴퓨터 판독가능 저장 매체.

[0457] 실시예 18. 비디오 데이터를 코딩하기 위한 디바이스로서, 디바이스는 실시예들 1 내지 7 중 임의의 것의 방법들을 수행하는 수단을 포함한다.

[0458] 예시에 의존하여, 본 명세서에서 설명된 기법들 중 임의의 것의 소정의 액트들 또는 이벤트들은 상이한 시퀀스

로 수행될 수 있고, 전체적으로 부가되거나 병합되거나 또는 제거될 수도 있음 (예를 들어, 설명된 모든 액트들 또는 이벤트들이 그 기법들의 실시를 위해 필수적인 것은 아님) 이 인식되어야 한다. 더욱이, 소정의 예들에서, 액트들 또는 이벤트들은 순차적인 것보다는, 예를 들어 다중-스레딩된 프로세싱, 인터럽트 프로세싱, 또는 다중의 프로세서들을 통해 동시에 수행될 수도 있다.

[0459] 하나 이상의 예들에 있어서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합에서 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 컴퓨터 판독가능 매체 상의 하나 이상의 명령들 또는 코드로서 저장되거나 이를 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예를 들어, 통신 프로토콜에 따라, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체는 일반적으로, (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품이 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0460] 한정이 아닌 예로서, 그러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장부, 자기 디스크 저장부 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 이용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 커넥션이 컴퓨터 판독가능 매체로 적절히 명명된다. 예를 들어, 동축 케이블, 광섬유 케이블, 꼬임쌍선, 디지털 가입자 라인 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신된다면, 동축 케이블, 광섬유 케이블, 꼬임쌍선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 캐리어파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않지만 대신 비일시적인 유형의 저장 매체들로 지향됨이 이해되어야 한다. 본원에서 이용된 디스크 (disk) 와 디스크 (disc) 는, 컴팩트 디스크(CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크, 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 통상 자기적으로 데이터를 재생하는 반면, 디스크(disc) 들은 레이저들을 이용하여 광학적으로 데이터를 재생한다. 또한, 상기의 조합은 컴퓨터 판독 가능 매체의 범위 내에 포함되어야 한다.

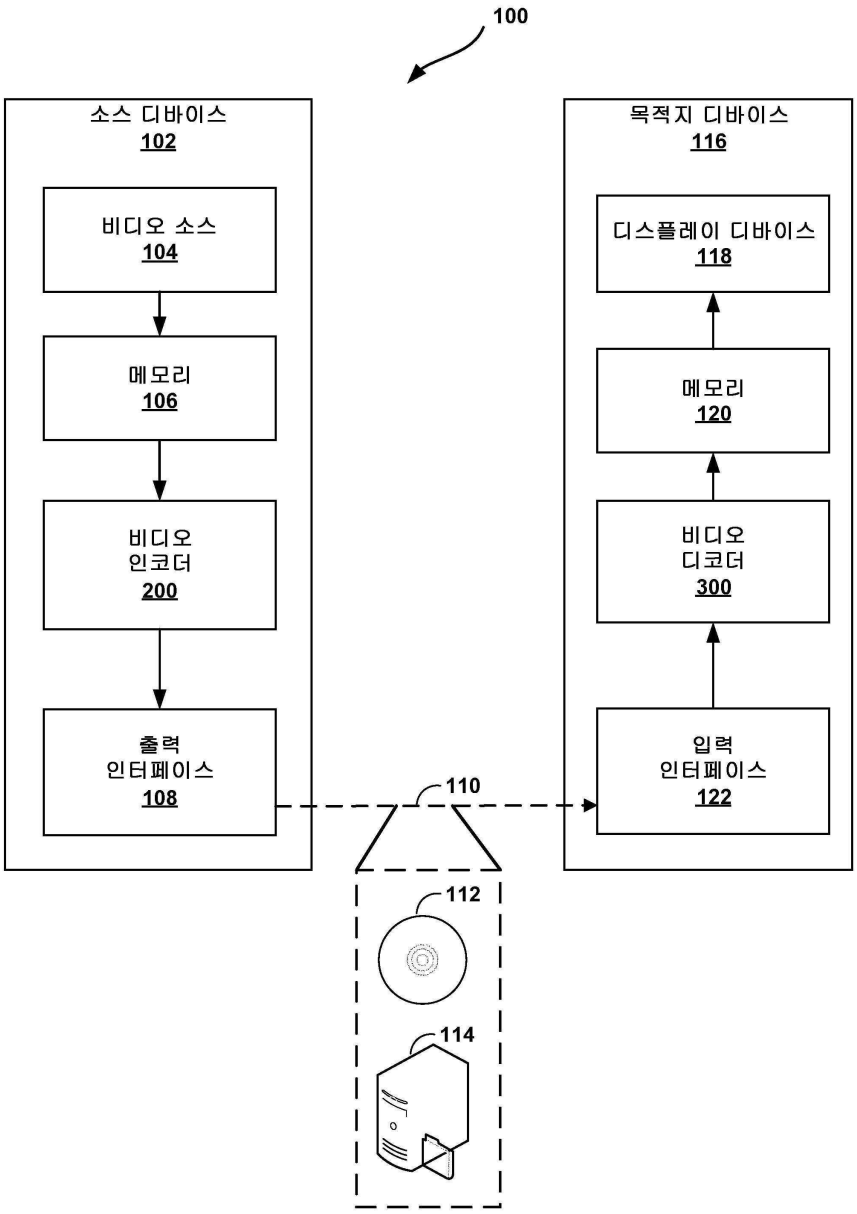
[0461] 명령들은 하나 이상의 프로세서, 이를 테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적 회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 또는 다른 등가의 집적 또는 이산 로직 회로부에 의해 실행될 수도 있다. 이에 따라, 본 명세서에서 사용된 바와 같은 용어들 "프로세서" 및 "프로세싱 회로" 는 전술한 구조들 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 추가로, 일부 양태들에서, 본 명세서에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 통합될 수도 있다. 또한, 그 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0462] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들어, 칩 세트) 를 포함하여, 광범위하게 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들 또는 유닛들이, 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적인 양태들을 강조하기 위하여 본 개시에 설명되었지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하는 것은 아니다. 오히려, 상술한 바와 같이, 다양한 유닛들이 코덱 하드웨어 유닛에 결합될 수도 있거나, 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상술한 하나 이상의 프로세서들을 포함하는 상호동작 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

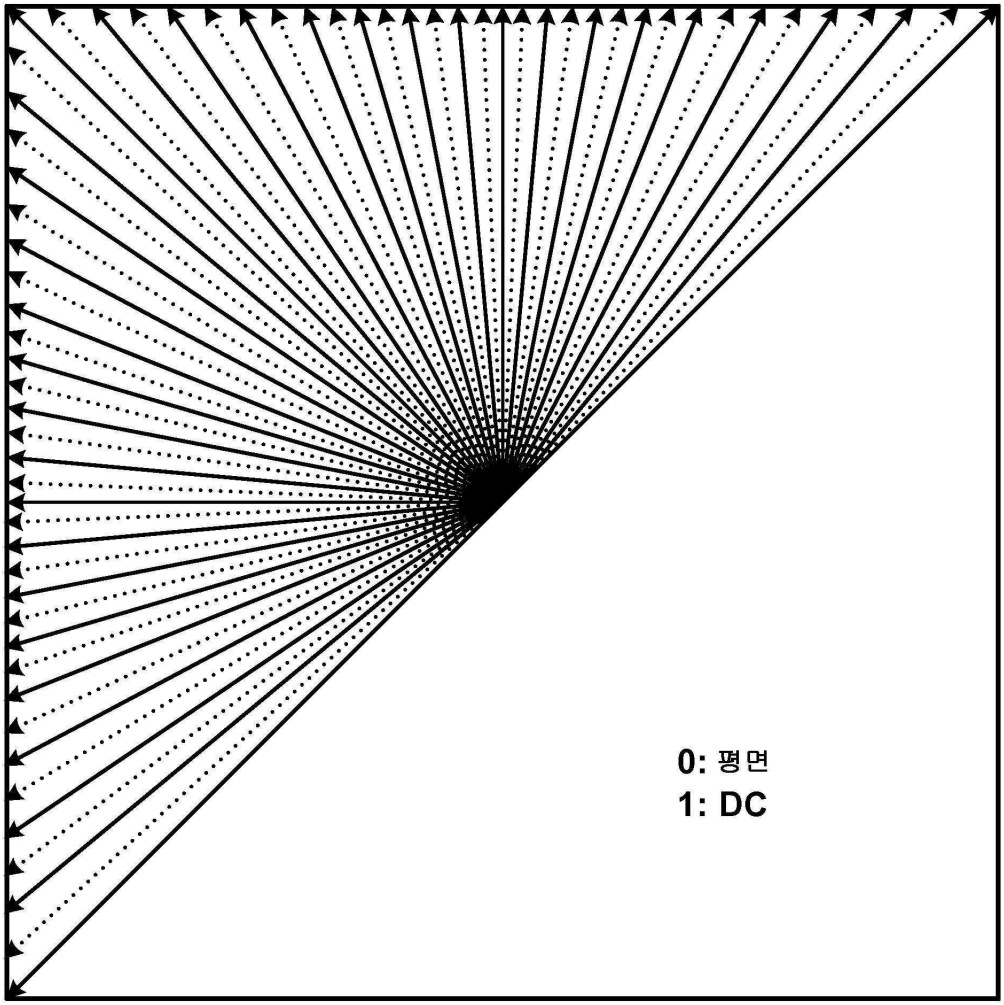
[0463] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

## 도면

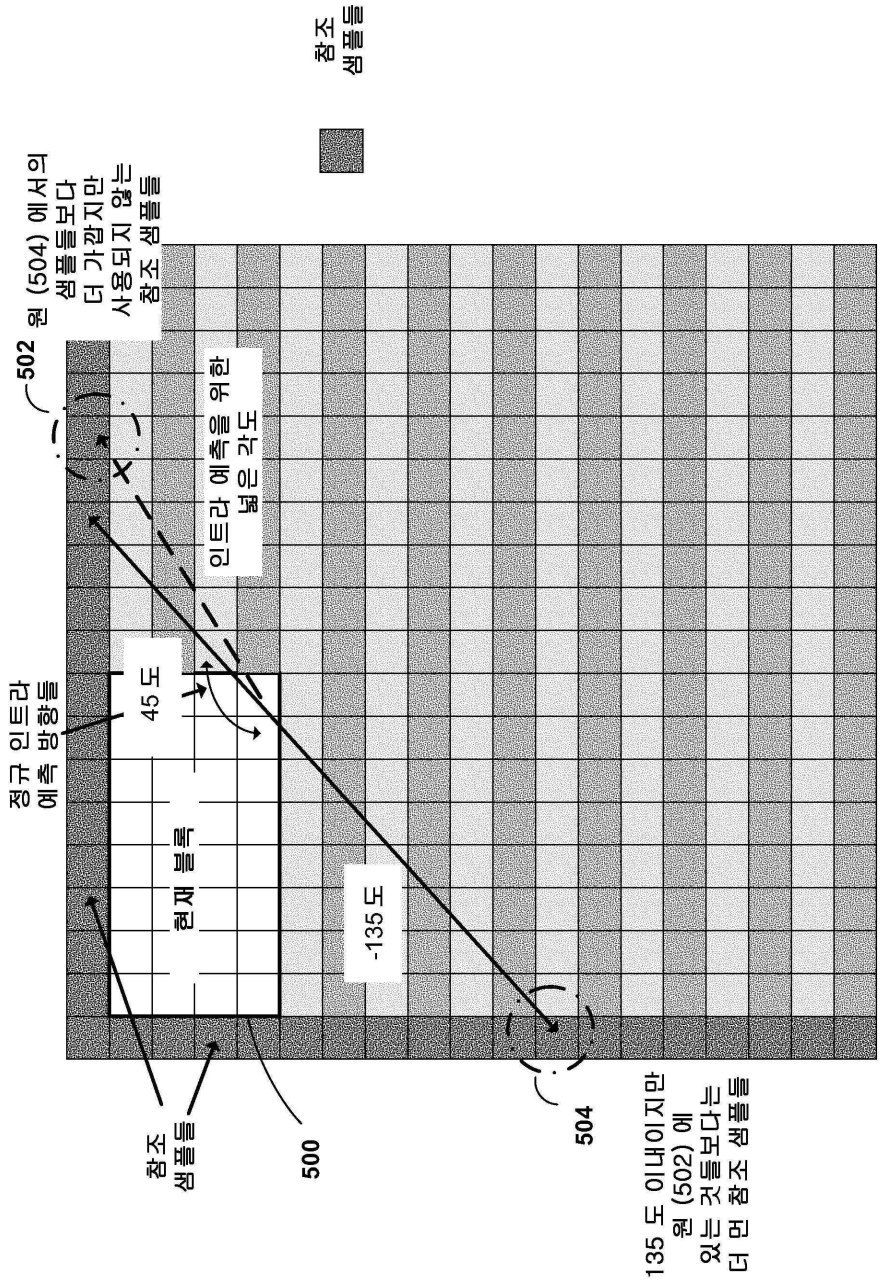
### 도면1


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

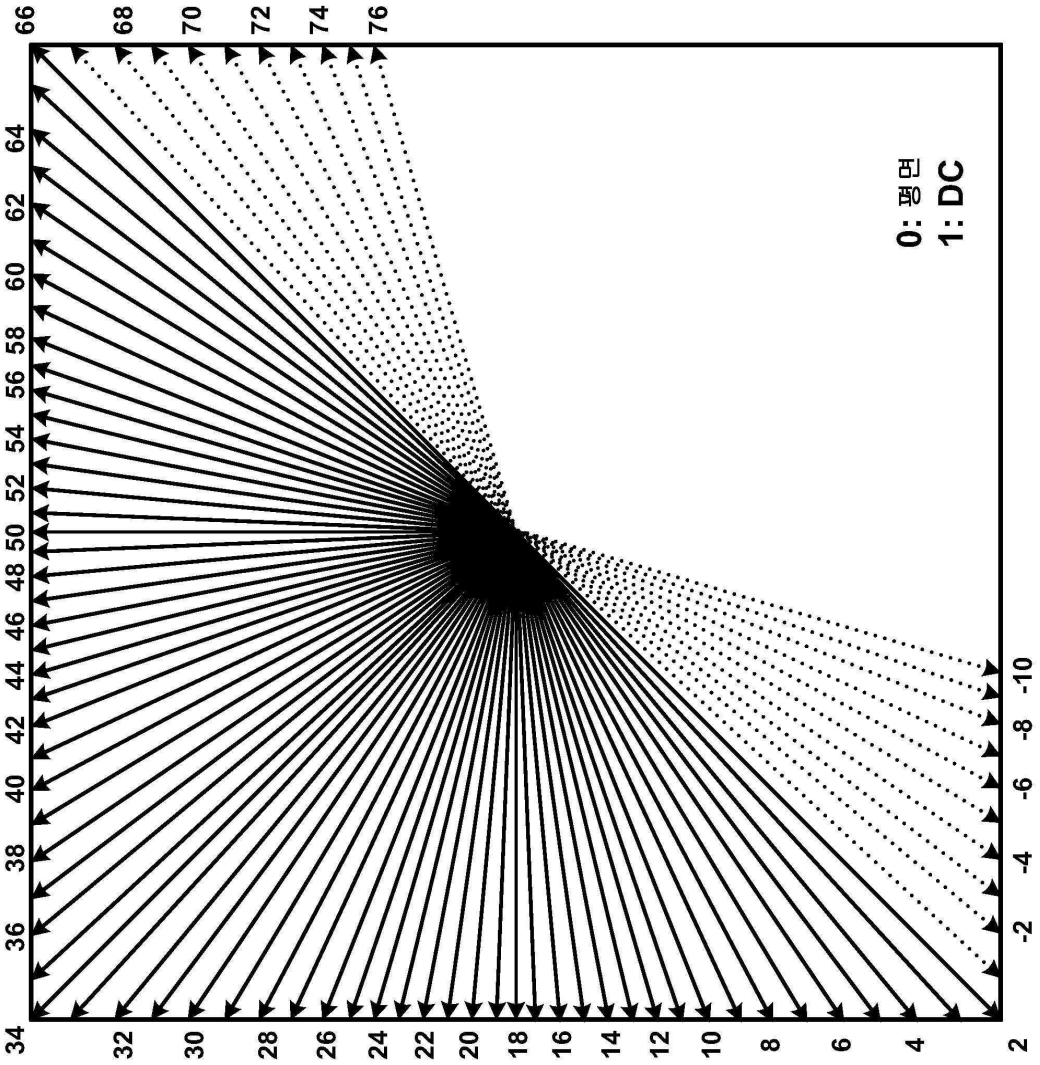
도면2



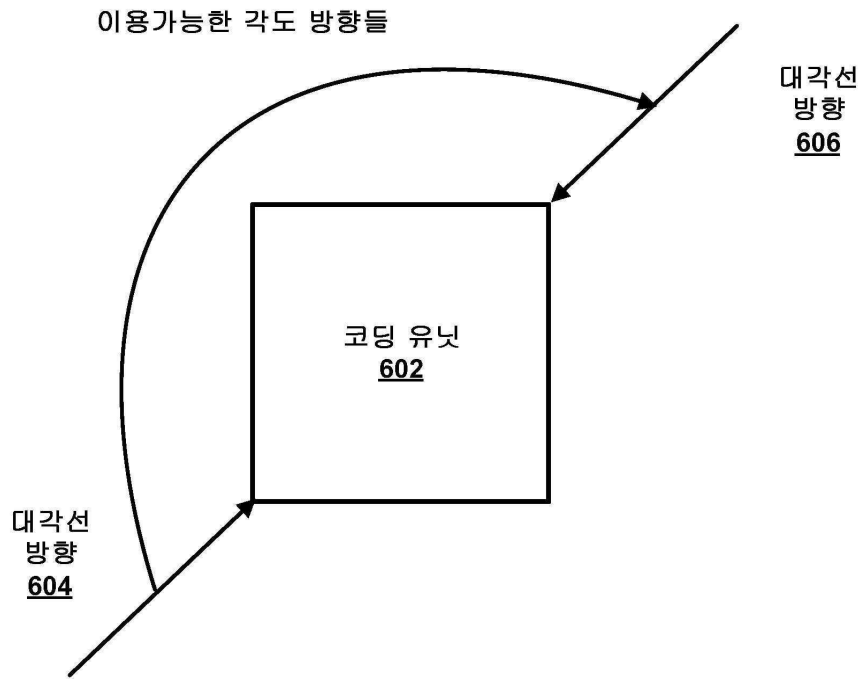
도면3


참조 샘플들
502
원 (504) 에서의 샘플들보다 더 가깝지만 사용되지 않는 참조 샘플들
인트라 예측을 위한 넓은 각도
정규 인트라 예측 방향들
45 도
현재 블록
-135 도
참조 샘플들
500
504
135 도 이내이지만 원 (502) 에 있는 것들보다는 더 먼 참조 샘플들

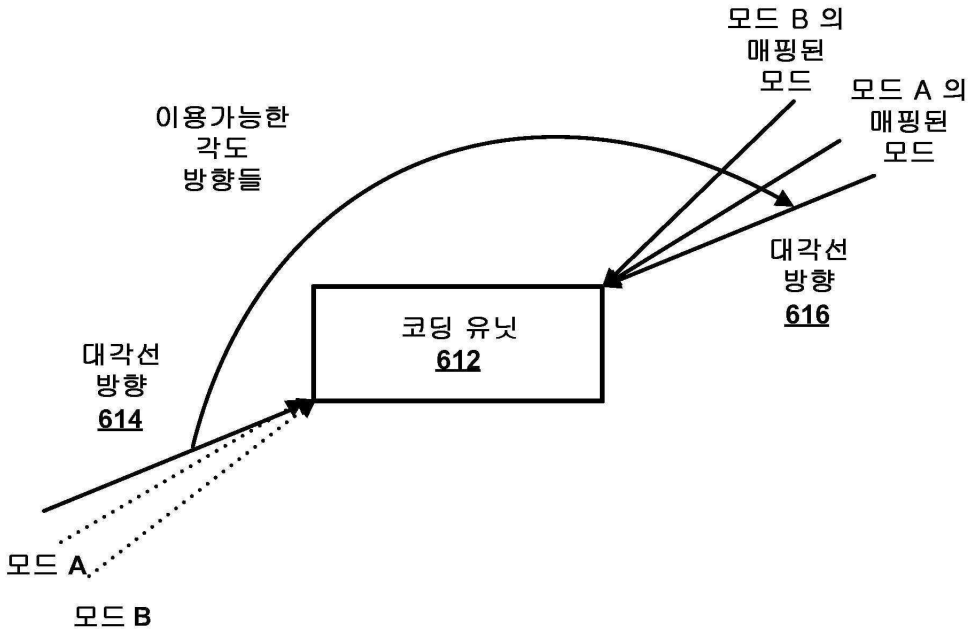
도면4


66
68
70
72
74
76
64
62
60
58
56
54
52
50
48
46
44
42
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
-2
-4
-6
-8
-10
0: 평면
1: DC

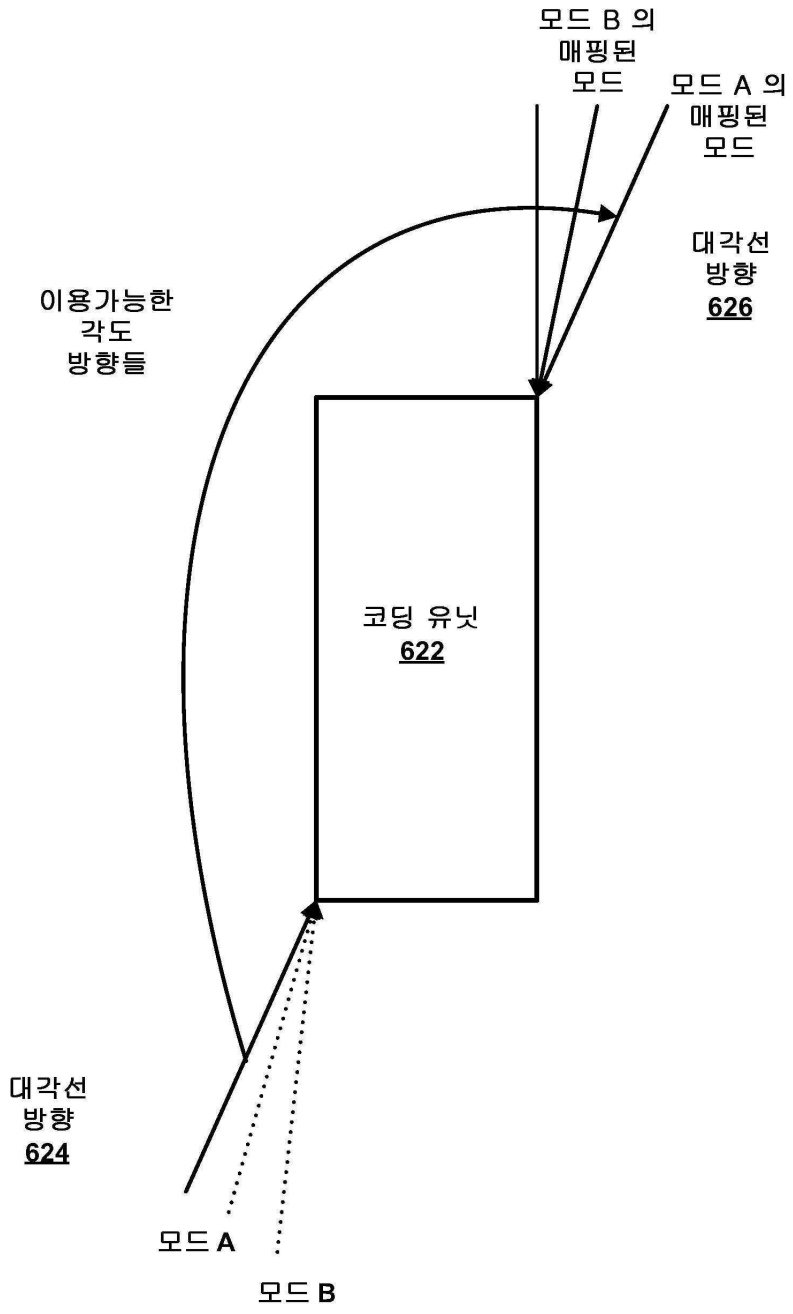
도면5a


이용가능한 각도 방향들
대각선
방향
606
코딩 유닛
602
대각선
방향
604

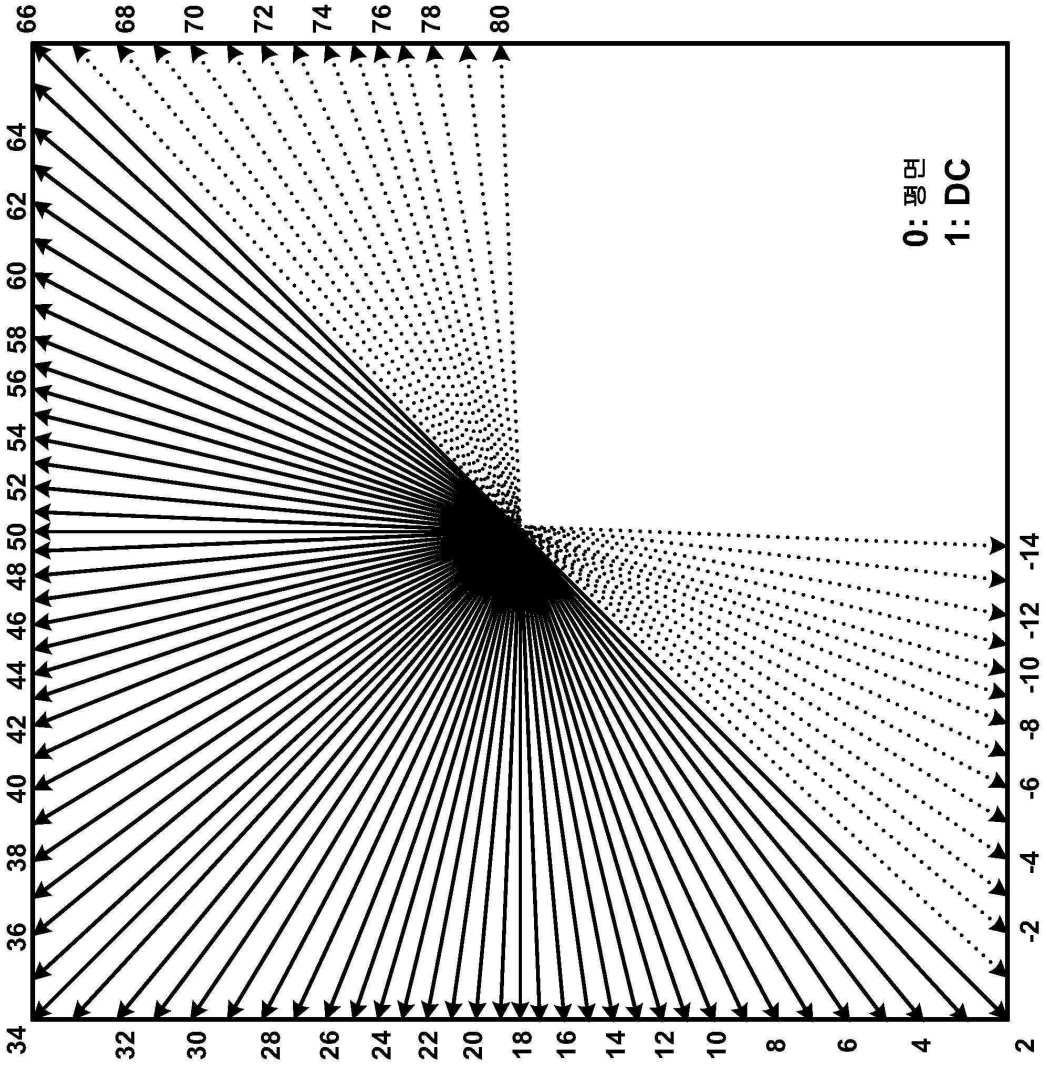
도면5b


모드 B 의
매핑된
모드
모드 A 의
매핑된
모드
이용가능한
각도
방향들
대각선
방향
616
코딩 유닛
612
대각선
방향
614
모드 A
모드 B

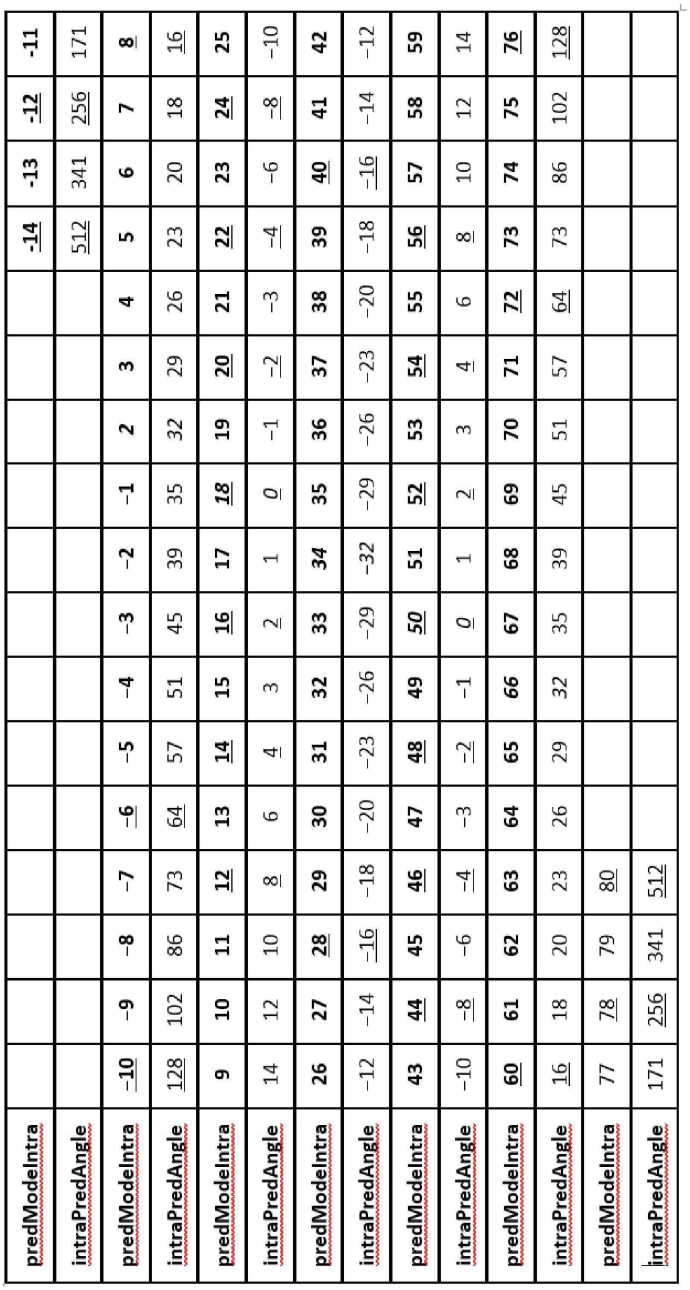
도면5c


모드 B 의
매핑된
모드
모드 A 의
매핑된
모드
대각선
방향
626
이용가능한
각도
방향들
코딩 유닛
622
대각선
방향
624
모드 A
모드 B

도면6

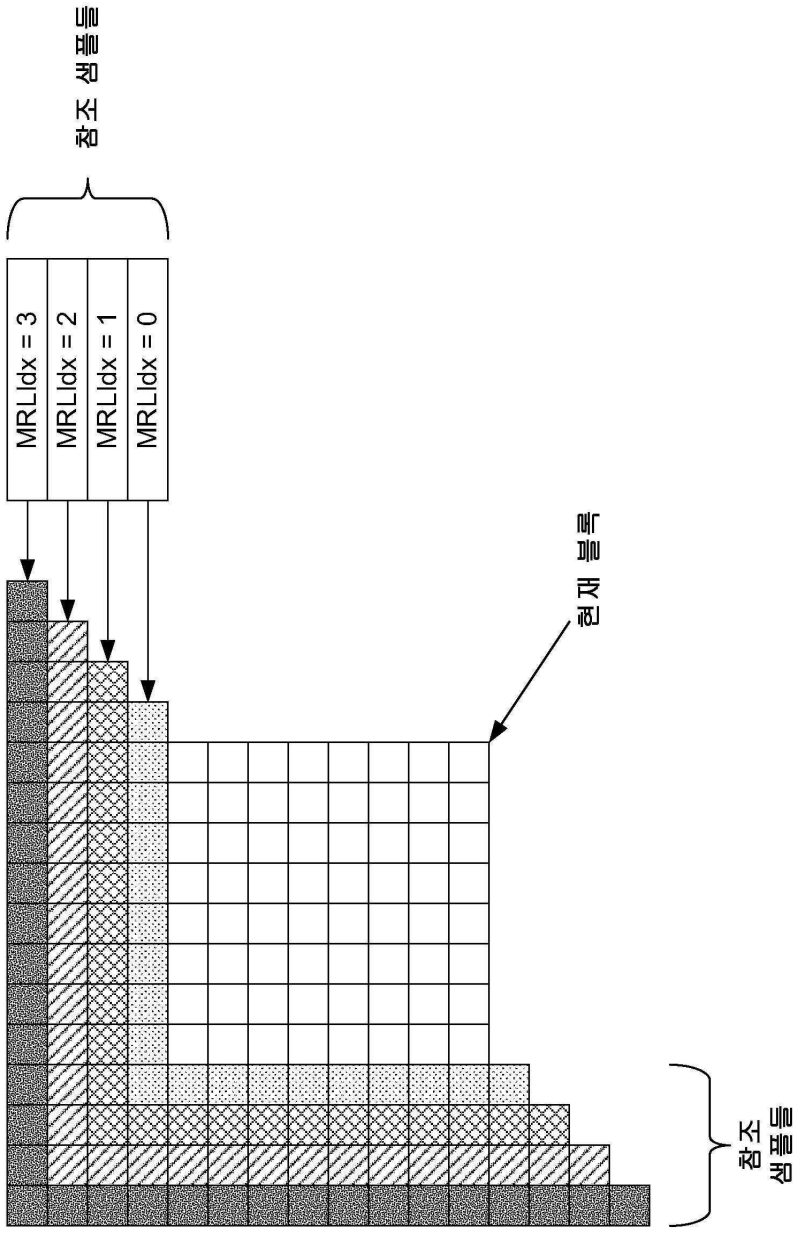

0: 평면
1: DC
66
68
70
72
74
76
78
80
64
62
60
58
56
54
52
50
48
46
44
42
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
-2
-4
-6
-8
-10
-12
-14

도면7

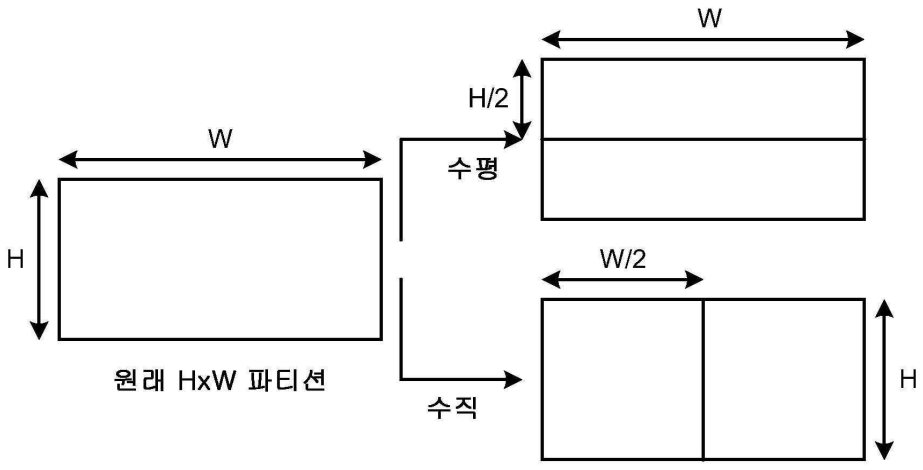


| predModeIntra | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| intraPredAngle | | | | | | | | | | | | | | | | | | | | | |
| predModeIntra | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | | | |
| intraPredAngle | 128 | 102 | 86 | 73 | 64 | 57 | 51 | 45 | 39 | 35 | 32 | 29 | 26 | 23 | 20 | 18 | 16 | | | | |
| predModeIntra | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | | | | |
| intraPredAngle | 14 | 12 | 10 | 8 | 6 | 4 | 3 | 2 | 1 | 0 | -1 | -2 | -3 | -4 | -6 | -8 | -10 | | | | |
| predModeIntra | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | | | | |
| intraPredAngle | -12 | -14 | -16 | -18 | -20 | -23 | -26 | -29 | -32 | -29 | -26 | -23 | -20 | -18 | -16 | -14 | -12 | | | | |
| predModeIntra | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | | | | |
| intraPredAngle | -10 | -8 | -6 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 6 | 8 | 10 | 12 | 14 | | | | |
| predModeIntra | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | | | | |
| intraPredAngle | 16 | 18 | 20 | 23 | 26 | 29 | 32 | 35 | 39 | 45 | 51 | 57 | 64 | 73 | 86 | 102 | 128 | | | | |
| predModeIntra | 77 | 78 | 79 | 80 | | | | | | | | | | | | | | | | | |
| intraPredAngle | 171 | 256 | 341 | 512 | | | | | | | | | | | | | | | | | |

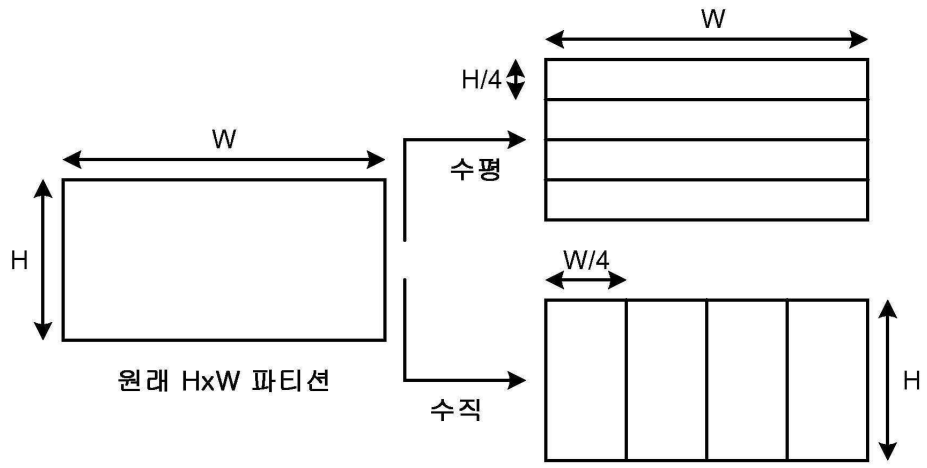
도면8


참조 샘플들
MRLIdx = 3
MRLIdx = 2
MRLIdx = 1
MRLIdx = 0
현재 블록
참조 샘플들

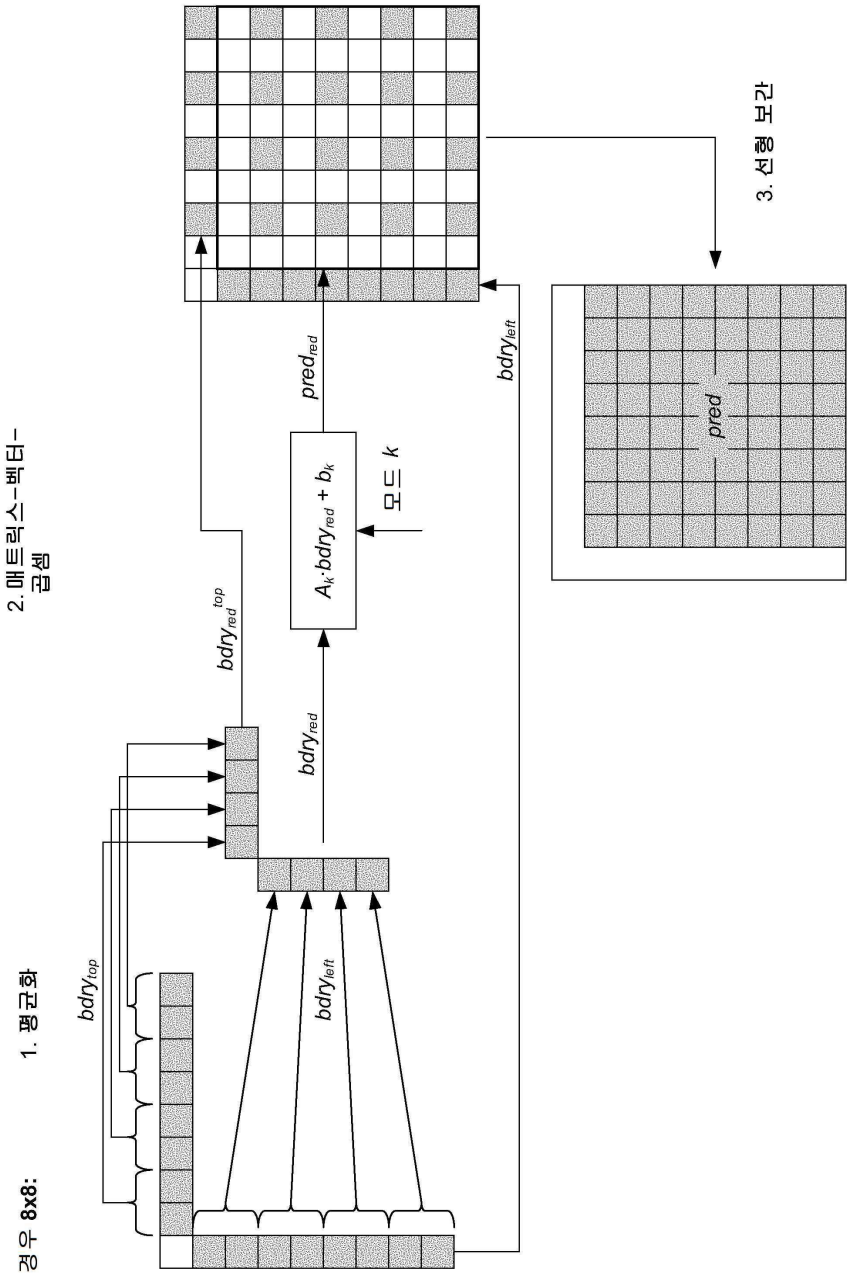
도면9


W
H/2
수평
W
H
원래 HxW 파티션
W/2
수직
H

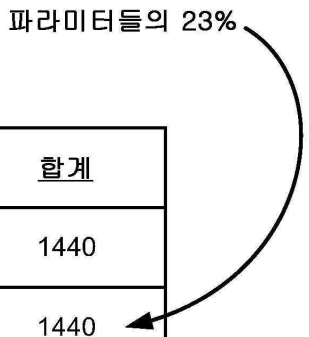
도면10



도면11

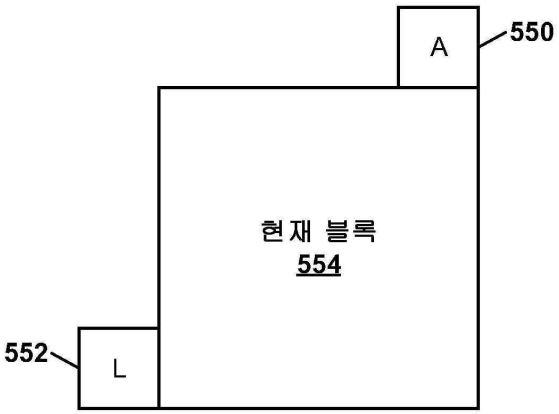

경우 8x8:
1. 평균화
2. 매트릭스-벡터-곱셈
3. 선형 보간
$bdry_{top}$
$bdry_{left}$
$bdry_{red}$
$bdry_{red}^{top}$
$A_k \cdot bdry_{red} + b_k$
모드 $k$
$pred_{red}$
$bdry_{left}$
$pred$

도면12

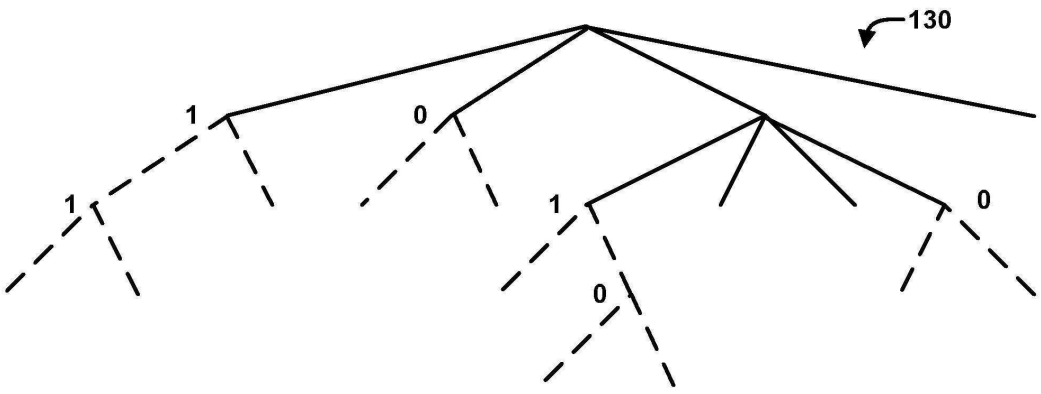



| 블록들 | 매트릭스 파라미터들 | 오프셋 파라미터들 | 합계 |
|---|---|---|---|
| 4x4 | 1152 | 288 | 1440 |
| 4x8,8x4,8x8 | 1280 | 160 | 1440 |
| 그외 | 3072 | 384 | 3456 |
| 합계 | | | 6336 |

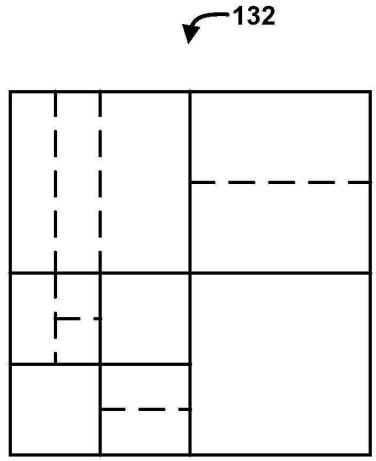
도면13



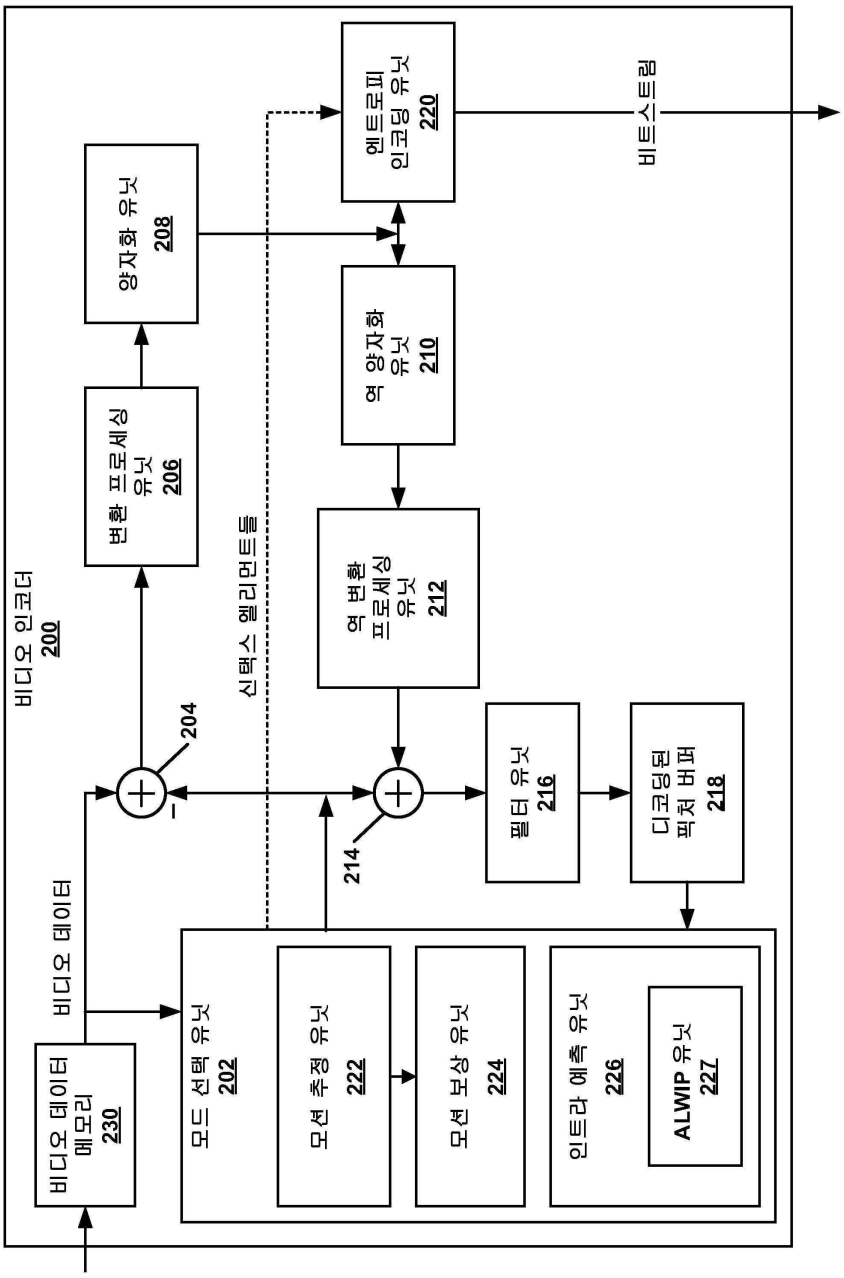

도면14a



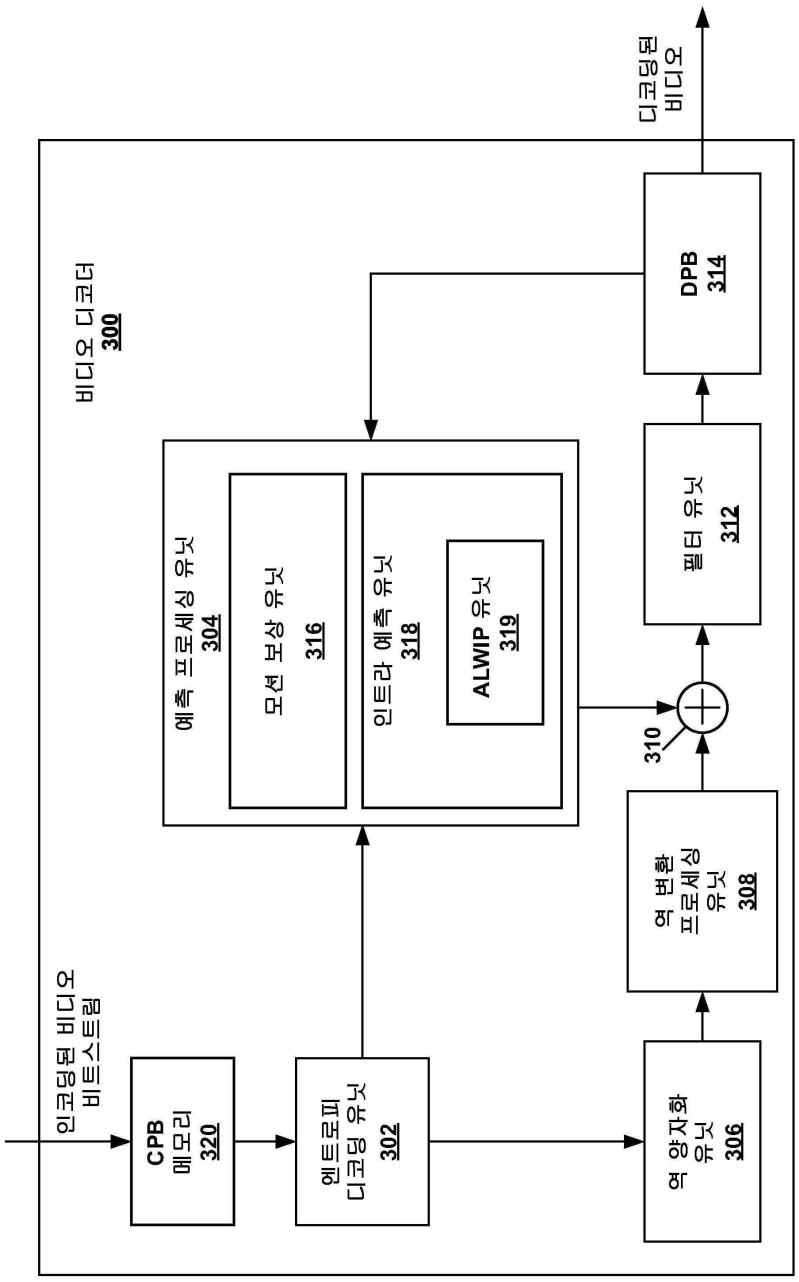
도면14b



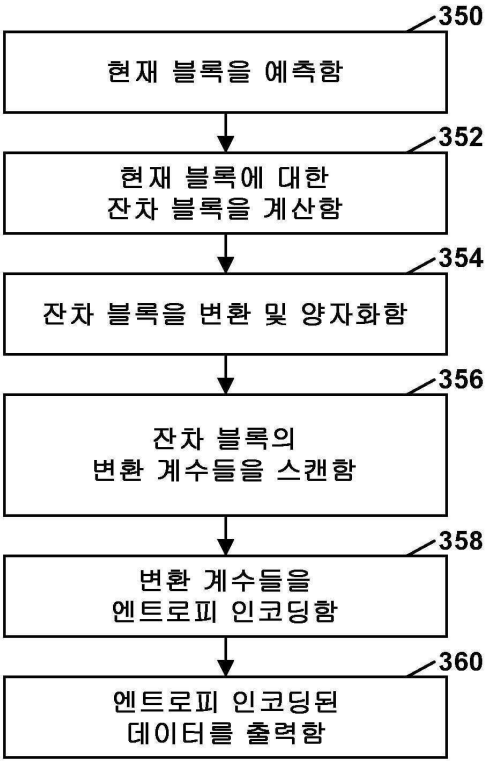
도면15


비디오 인코더
200
비디오 데이터
비디오 데이터 메모리
230
모드 선택 유닛
202
모션 추정 유닛
222
모션 보상 유닛
224
인트라 예측 유닛
226
ALWIP 유닛
227
204
214
신택스 엘리먼트들
변환 프로세싱 유닛
206
양자화 유닛
208
역 양자화 유닛
210
역 변환 프로세싱 유닛
212
필터 유닛
216
디코딩된 픽처 버퍼
218
엔트로피 인코딩 유닛
220
비트스트림

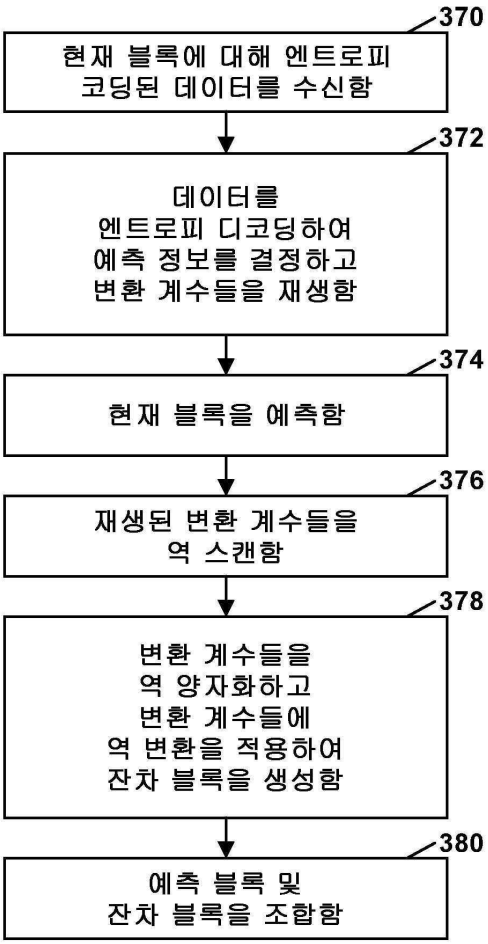
도면16


디코딩된 비디오
비디오 디코더 300
DPB 314
필터 유닛 312
예측 프로세싱 유닛 304
모션 보상 유닛 316
인트라 예측 유닛 318
ALWIP 유닛 319
310
역 변환 프로세싱 유닛 308
역 양자화 유닛 306
엔트로피 디코딩 유닛 302
CPB 메모리 320
인코딩된 비디오 비트스트림

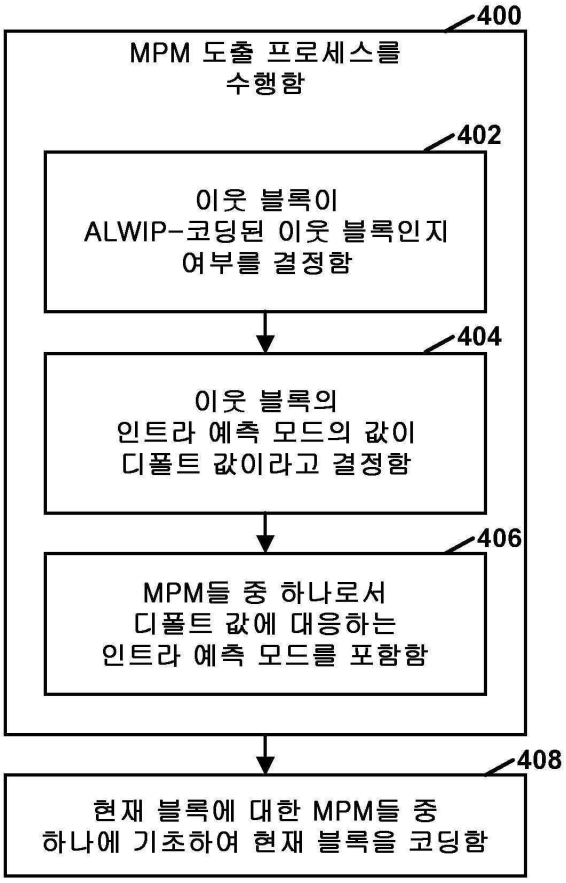
도면17


350
현재 블록을 예측함
352
현재 블록에 대한
잔차 블록을 계산함
354
잔차 블록을 변환 및 양자화함
356
잔차 블록의
변환 계수들을 스캔함
358
변환 계수들을
엔트로피 인코딩함
360
엔트로피 인코딩된
데이터를 출력함

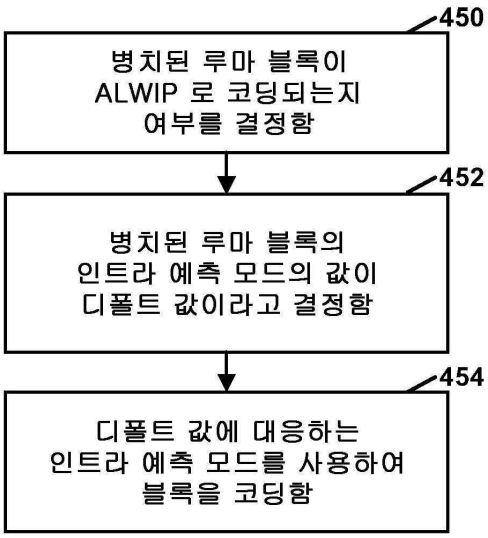
도면18


370
현재 블록에 대해 엔트로피 코딩된 데이터를 수신함
372
데이터를 엔트로피 디코딩하여 예측 정보를 결정하고 변환 계수들을 재생함
374
현재 블록을 예측함
376
재생된 변환 계수들을 역 스캔함
378
변환 계수들을 역 양자화하고 변환 계수들에 역 변환을 적용하여 잔차 블록을 생성함
380
예측 블록 및 잔차 블록을 조합함

도면19


400
MPM 도출 프로세스를
수행함
402
이웃 블록이
ALWIP-코딩된 이웃 블록인지
여부를 결정함
404
이웃 블록의
인트라 예측 모드의 값이
디폴트 값이라고 결정함
406
MPM들 중 하나로서
디폴트 값에 대응하는
인트라 예측 모드를 포함함
408
현재 블록에 대한 MPM들 중
하나에 기초하여 현재 블록을 코딩함

도면20


450
병치된 루마 블록이
ALWIP 로 코딩되는지
여부를 결정함
452
병치된 루마 블록의
인트라 예측 모드의 값이
디폴트 값이라고 결정함
454
디폴트 값에 대응하는
인트라 예측 모드를 사용하여
블록을 코딩함