



공개특허 10-2022-0016833



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0016833  
(43) 공개일자 2022년02월10일

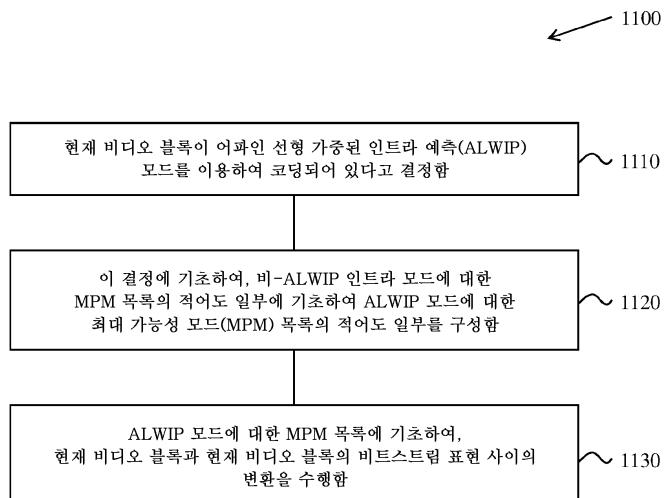
- (51) 국제특허분류(Int. Cl.)  
*HO4N 19/11* (2014.01) *HO4N 19/105* (2014.01)  
*HO4N 19/119* (2014.01) *HO4N 19/176* (2014.01)  
*HO4N 19/186* (2014.01) *HO4N 19/59* (2014.01)  
*HO4N 19/593* (2014.01)
- (52) CPC특허분류  
*HO4N 19/11* (2015.01)  
*HO4N 19/105* (2015.01)
- (21) 출원번호 10-2021-7038590
- (22) 출원일자(국제) 2020년06월05일  
심사청구일자 없음
- (85) 번역문제출일자 2021년11월25일
- (86) 국제출원번호 PCT/CN2020/094513
- (87) 국제공개번호 WO 2020/244610  
국제공개일자 2020년12월10일
- (30) 우선권주장  
PCT/CN2019/090205 2019년06월05일 중국(CN)
- (71) 출원인  
베이징 바이트댄스 네트워크 테크놀로지 컴퍼니, 리미티드  
중국, 베이징 100041, 스징산 디스트릭트, 스싱로드, 넘버 30, 넘버 3빌딩, 2층, 룸 비-0085  
바이트댄스 아이엔씨  
미국, 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
- (72) 발명자  
명, 즈핀  
중국 100080 베이징 하이뎬 디스트릭트 즈춘 로드 넘버 63 차이나 세틀라이트 커뮤니케이션즈 타워 진르터우탸오 포스트 오피스  
장, 카이  
미국 90066 캘리포니아주 로스엔젤레스 스위트 넘버 137 6층 웨스트 제퍼슨 블러바드 12655  
(뒷면에 계속)
- (74) 대리인  
양영준, 임규빈, 백만기

전체 청구항 수 : 총 47 항

## (54) 발명의 명칭 행렬 기반의 인트라 예측을 위한 컨텍스트 결정

**(57) 요 약**

비디오 코딩을 위한 행렬 기반의 인트라 예측 방법들을 포함하는, 디지털 비디오 코딩을 위한 디바이스들, 시스템들 및 방법들이 설명된다. 한 대표적인 양태에서, 비디오 처리를 위한 방법은, 비디오의 루마 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 코딩되어 있는지에 관한 제1 결정을 수행하는 단계; 루마 비디오 블록이 비디오의 현재 크로마 비디오 블록에 대한 크로마 인트라 모드를 결정하는데 적용가능하다는 제2 결정을 수행하는 단계; 제1 결정 및 제2 결정에 기초하여, 현재 크로마 비디오 블록에 이용될 크로마 인트라 모드에 관한 제3 결정을 수행하는 단계; 및 제3 결정에 기초하여, 현재 크로마 비디오 블록과 현재 크로마 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

**대 표 도 - 도11**

(52) CPC특허분류

*HO4N 19/119* (2015.01)

*HO4N 19/176* (2015.01)

*HO4N 19/186* (2015.01)

*HO4N 19/59* (2015.01)

*HO4N 19/593* (2015.01)

(72) 발명자

장, 리

미국 90066 캘리포니아주 로스엔젤레스 스위트 넘  
버 137 6층 웨스트 제퍼슨 블러바드 12655

류, 홍빈

중국 100080 베이징 하이뎬 디스트릭트 즈춘 로드  
넘버 63 차이나 세틀라이트 커뮤니케이션즈 타워  
진루터우타오 포스트 오피스

---

취, 지정

미국 90066 캘리포니아주 로스엔젤레스 스위트 넘  
버 137 6층 웨스트 제퍼슨 블러바드 12655

## 명세서

### 청구범위

#### 청구항 1

비디오 처리를 위한 방법으로서,

비디오의 루마 비디오 블록이 행렬 기반의 인트라 예측(matrix based intra prediction)(MIP) 모드를 이용하여 코딩되어 있는지에 관한 제1 결정을 수행하는 단계 – 이 모드에서, 상기 루마 비디오 블록의 예측 블록은 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 백터 꼽셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –;

상기 루마 비디오 블록이 상기 비디오의 현재 크로마 비디오 블록에 대한 크로마 인트라 모드를 결정하는데 적용가능하다는 제2 결정을 수행하는 단계 – 상기 제2 결정을 수행하는 단계는 상기 현재 크로마 비디오 블록과 연관된 현재 픽처의 상단 좌측 루마 샘플에 관한 상기 현재 크로마 비디오 블록의 상단 좌측 샘플에 기초하고, 상기 제2 결정을 수행하는 단계는 현재 비디오 유닛의 상기 루마 비디오 블록 및/또는 컬러 포맷에 관한 상기 현재 크로마 비디오 블록의 치수들에 기초함 –;

상기 제1 결정 및 상기 제2 결정에 기초하여, 상기 현재 크로마 비디오 블록에 이용될 상기 크로마 인트라 모드에 관한 제3 결정을 수행하는 단계; 및

상기 제3 결정에 기초하여, 상기 현재 크로마 비디오 블록과 상기 현재 크로마 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계

를 포함하는, 방법.

#### 청구항 2

제1항에 있어서, 상기 크로마 인트라 모드는 직접 모드 또는 도출된 모드를 포함하는, 방법.

#### 청구항 3

제1항에 있어서,

상기 현재 크로마 비디오 블록의 상단 좌측 샘플은 상기 루마 비디오 블록 내에 위치하고,

상기 현재 크로마 비디오 블록의 제1 폭 및 상기 현재 크로마 비디오 블록의 제1 높이는 각각, 상기 루마 비디오 블록의 제2 폭 및 상기 루마 비디오 블록의 제2 높이보다 작거나 같은, 방법.

#### 청구항 4

제1항에 있어서,

상기 현재 픽처의 상단 좌측 루마 샘플에 관한 상기 현재 크로마 비디오 블록의 상단 좌측 샘플은  $(xCb, yCb)$ 이고,

상기 루마 비디오 블록 내에 위치한  $(xCb + offsetX, yCb + offsetY)$ 로 표기되는 위치의 예측 모드는 상기 현재 크로마 비디오 블록에 대한 상기 크로마 인트라 모드를 결정하는데 적용가능하며,

상기  $offsetX$ 는 수평 방향의 오프셋이고,

상기  $offsetY$ 는 수직 방향의 오프셋인, 방법.

#### 청구항 5

제4항에 있어서, 상기  $offsetX$  및 상기  $offsetY$  양쪽 모두가 0인 것은 불허되는, 방법.

#### 청구항 6

제4항에 있어서, 상기 offsetX는 상기 현재 크로마 비디오 블록의 폭을 2로 나눈 것과 동일한, 방법.

#### 청구항 7

제4항에 있어서, 상기 offsetX는 상기 루마 비디오 블록의 폭을 2로 나눈 것과 동일한, 방법.

#### 청구항 8

제4항에 있어서, 상기 offsetX는 상기 현재 크로마 비디오 블록의 폭을 2로 나눈 결과로부터 1을 뺀 것과 동일한, 방법.

#### 청구항 9

제4항에 있어서, 상기 offsetX는 상기 현재 크로마 비디오 블록의 폭을 2로 나눈 결과로부터 1을 더한 것과 동일한, 방법.

#### 청구항 10

제4항에 있어서, 상기 offsetY는 상기 현재 크로마 비디오 블록의 높이를 2로 나눈 것과 동일한, 방법.

#### 청구항 11

제4항에 있어서, 상기 offsetY는 상기 루마 비디오 블록의 높이를 2로 나눈 것과 동일한, 방법.

#### 청구항 12

제4항에 있어서, 상기 offsetY는 상기 현재 크로마 비디오 블록의 높이를 2로 나눈 결과로부터 1을 뺀 것과 동일한, 방법.

#### 청구항 13

제4항에 있어서, 상기 offsetY는 상기 현재 크로마 비디오 블록의 높이를 2로 나눈 결과로부터 1을 더한 것과 동일한, 방법.

#### 청구항 14

비디오 처리를 위한 방법으로서,

비디오의 루마 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 코딩되어 있다는 제1 결정을 수행하는 단계 – 이 모드에서, 상기 루마 비디오 블록의 예측 블록은 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 꼽셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –;

상기 제1 결정에 기초하여, 도출된 모드(DM)가 상기 루마 비디오 블록과 연관된 현재 크로마 비디오 블록에 대한 크로마 인트라 모드로서 이용되는지에 관한 규칙에 따른 제2 결정을 수행하는 단계; 및

상기 제2 결정에 기초하여, 상기 현재 크로마 비디오 블록과 상기 현재 크로마 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계

를 포함하는, 방법.

#### 청구항 15

제14항에 있어서, 상기 규칙은 상기 루마 비디오 블록이 상기 MIP 모드를 이용하여 코딩되어 있는 것에 응답하여 상기 DM이 상기 크로마 인트라 모드로서 이용됨을 명시하는, 방법.

#### 청구항 16

제14항에 있어서, 상기 비트스트림 표현에서의 상기 크로마 인트라 모드의 시그널링은 스kip되는, 방법.

#### 청구항 17

제14항에 있어서,

상기 변환을 수행하는 단계는 상기 비트스트림 표현에서 상기 현재 크로마 비디오 블록에 대해 상기 DM에 대한 시그널링을 포함하고,

상기 규칙은 미리정의된 인트라 예측 모드가 상기 현재 크로마 비디오 블록에 적용됨을 명시하는, 방법.

#### 청구항 18

제17항에 있어서, 상기 미리정의된 인트라 예측 모드는 Planar 모드인, 방법.

#### 청구항 19

제17항에 있어서, 상기 미리정의된 인트라 예측 모드는 스크린 콘텐츠와 연관된 코딩된 정보에 기초하는, 방법.

#### 청구항 20

제19항에 있어서, 상기 미리정의된 인트라 예측 모드는 상기 스크린 콘텐츠가 카메라 캡처된 스크린 콘텐츠임을 상기 코딩된 정보가 나타내는 것에 응답하여 Planar 모드인, 방법.

#### 청구항 21

제19항에 있어서, 상기 미리정의된 인트라 예측 모드는 상기 스크린 콘텐츠가 비-카메라 캡처된 콘텐츠임을 상기 코딩된 정보가 나타내는 것에 응답하여 수평 예측 모드인, 방법.

#### 청구항 22

제14항에 있어서, 상기 규칙은 상기 루마 비디오 블록이 상기 MIP 모드를 이용하여 코딩되어 있는 것에 응답하여 상기 DM이 상기 크로마 인트라 모드로서 이용되지 않음을 명시하는, 방법.

#### 청구항 23

제22항에 있어서, 상기 DM은 상기 비트스트림 표현에서 시그널링되지 않는, 방법.

#### 청구항 24

제23항에 있어서, 4개의 크로마 모드는 상기 크로마 인트라 모드에 대해 허용되고, 상기 루마 비디오 블록이 상기 MIP 모드를 이용하여 코딩되어 있는 것에 응답하여 그리고 교차-성분 선형 모델 예측(cross-component linear model prediction)(CCLM)이 상기 현재 크로마 비디오 블록에 대해 인에이블되어 있지 않음을 플래그가 나타내는 것에 응답하여 상기 현재 크로마 비디오 블록에 대해 상기 비트스트림 표현에서 시그널링되는, 방법.

#### 청구항 25

제24항에 있어서, 상기 비트스트림 표현에서 상기 크로마 인트라 모드를 시그널링하기 위해 최대 2개의 비트가 이용되는, 방법.

#### 청구항 26

제23항에 있어서, 7개의 크로마 모드는 상기 크로마 인트라 모드에 대해 허용되고, 상기 루마 비디오 블록이 상기 MIP 모드를 이용하여 코딩되어 있는 것에 응답하여 그리고 교차-성분 선형 모델 예측(CCLM)이 상기 현재 크로마 비디오 블록에 대해 인에이블되어 있음을 플래그가 나타내는 것에 응답하여 상기 현재 크로마 비디오 블록에 대해 상기 비트스트림 표현에서 시그널링되는, 방법.

#### 청구항 27

제26항에 있어서, 상기 비트스트림 표현에서 상기 크로마 인트라 모드를 시그널링하기 위해 최대 4개의 비트가 이용되는, 방법.

#### 청구항 28

비디오 처리를 위한 방법으로서,

비디오의 현재 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드로 코딩되어 있다고 결정하는 단계 – 이 모드에서, 상기 현재 비디오 블록의 예측 블록은 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 꼽셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –;

상기 결정에 기초하여, 규칙에 기초한 최대 가능성 모드(most probable mode)(MPM) 목록을 생성하는 단계 – 상기 규칙은 상기 MPM 목록이 상기 현재 비디오 블록의 이웃 비디오 블록에 대해 생성된 또 다른 MPM 목록과는 독립적으로 생성됨을 명시함 –; 및

상기 MPM 목록을 이용하여 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계

를 포함하는, 방법.

#### 청구항 29

제28항에 있어서, 상기 MPM 목록은 미리정의된 최대 가능성 모드들을 포함하는, 방법.

#### 청구항 30

제29항에 있어서, 상기 미리정의된 최대 가능성 모드들은 상기 현재 비디오 블록의 치수들에 관계없이 {M0, M1, M2}를 포함하는, 방법.

#### 청구항 31

제29항에 있어서, 상기 미리정의된 최대 가능성 모드들은 상기 현재 비디오 블록의 치수들에 기초하는, 방법.

#### 청구항 32

제31항에 있어서, 상기 미리정의된 최대 가능성 모드들은 상기 현재 비디오 블록의 블록 크기 유형을 나타내는 식별자에 기초하는, 방법.

#### 청구항 33

제32항에 있어서, 상기 블록 크기 유형을 나타내는 식별자는 음이 아닌 값인, 방법.

#### 청구항 34

제32항에 있어서, 상기 식별자는 0, 1 또는 2를 포함하는, 방법.

#### 청구항 35

제32항에 있어서, 상기 미리정의된 최대 가능성 모드들은 블록 크기 유형의 식별자가 소정의 값과 동일한 것에 응답하여 {17, 34, 5}인, 방법.

#### 청구항 36

제35항에 있어서, 상기 소정의 값은 0인, 방법.

#### 청구항 37

제32항에 있어서, 상기 미리정의된 최대 가능성 모드들은 상기 블록 크기 유형의 식별자가 소정의 값과 동일한 것에 응답하여 {0, 7, 16}인, 방법.

#### 청구항 38

제37항에 있어서, 상기 소정의 값은 1인, 방법.

#### 청구항 39

제32항에 있어서, 상기 미리정의된 최대 가능성 모드들은 상기 블록 크기 유형의 식별자가 소정의 값과 동일한 것에 응답하여 {1, 4, 6}인, 방법.

**청구항 40**

제39항에 있어서, 상기 소정의 값은 2인, 방법.

**청구항 41**

비디오 인코딩을 위한 방법으로서,

비디오의 현재 비디오 블록을 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 인코딩하는 단계 – 이 모드에서, 상기 현재 비디오 블록의 예측 블록은 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정되고, 규칙은 상기 MIP 모드에서 이용되는 행렬의 수와 상기 현재 비디오 블록의 치수들 사이의 관계를 기술함 –; 및

상기 현재 비디오 블록의 인코딩된 표현을 상기 현재 비디오 블록의 비트스트림 표현에 추가하는 단계  
를 포함하는, 방법.

**청구항 42**

비디오 디코딩을 위한 방법으로서,

비디오의 현재 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 비트스트림 표현으로 인코딩되어 있다고 결정하는 단계 – 이 모드에서, 상기 현재 비디오 블록의 예측 블록은 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정되고, 규칙은 상기 MIP 모드에서 이용되는 행렬의 수와 상기 현재 비디오 블록의 치수들 사이의 관계를 기술함 –; 및

상기 비트스트림 표현을 파싱 및 디코딩함으로써 상기 현재 비디오 블록의 디코딩된 표현을 생성하는 단계  
를 포함하는, 방법.

**청구항 43**

제41항 또는 제42항에 있어서,

상기 규칙은 상기 행렬의 수가 상기 현재 비디오 블록의 치수들에 기초하지 않음을 명시하고,

상기 행렬의 수는 이전에 저장되고,

상기 행렬의 수는 11, 19 또는 35인, 방법.

**청구항 44**

제41항 또는 제42항에 있어서,

상기 규칙은 상기 행렬의 수가 상기 현재 비디오 블록의 치수들에 기초하지 않음을 명시하고,

상기 MIP 모드에 대한 시그널링은 상기 현재 비디오 블록에 대한 상기 비트스트림 표현에 포함되고,

상기 MIP 모드에 대한 상기 시그널링은 상기 현재 비디오 블록의 치수들에 의존하지 않는, 방법.

**청구항 45**

제41항 또는 제42항에 있어서, 상기 규칙은 상기 행렬의 수가 상기 현재 비디오 블록의 치수들에 기초함을 명시하는, 방법.

**청구항 46**

프로세서 및 명령어들이 저장된 비일시적 메모리를 포함하는 비디오 시스템의 장치로서, 상기 명령어들은 상기 프로세서에 의한 실행시 상기 프로세서로 하여금 제1항 내지 제45항 중 어느 하나의 방법을 구현하게 하는, 장치.

**청구항 47**

비일시적 컴퓨터 관독가능한 매체에 저장된 컴퓨터 프로그램 제품으로서, 상기 컴퓨터 프로그램 제품은 제1항 내지 제45항 중 어느 하나의 방법을 실행하기 위한 프로그램 코드를 포함하는, 컴퓨터 프로그램 제품.

## 발명의 설명

### 기술 분야

[0001] 관련 출원의 상호참조

[0002] 적용 가능한 특허법 및/또는 과리 협약에 따른 규칙들 하에서, 본 출원은 2019년 6월 5일 출원된 국제 특허 출원 번호 PCT/CN2019/090205호에 대한 우선권을 주장한다. 법에 따른 모든 목적을 위해, 앞서 언급된 출원의 전체 개시내용은 본 출원의 개시내용의 일부로서 참조에 의해 포함된다.

[0003] 기술 분야

[0004] 본 특허 문서는, 비디오 코딩 기술들, 디바이스들 및 시스템들에 관한 것이다.

### 배경 기술

[0005] 비디오 압축에서의 진보에도 불구하고, 디지털 비디오는 여전히 인터넷 및 기타의 디지털 통신 네트워크에서 가장 큰 대역폭을 이용한다. 비디오를 수신하고 디스플레이할 수 있는 접속된 사용자 디바이스들의 수가 증가함에 따라, 디지털 비디오 이용에 대한 대역폭 수요는 계속 증가할 것으로 예상된다.

### 발명의 내용

[0006] 디지털 비디오 코딩과 관련된 디바이스들, 시스템들 및 방법들, 구체적으로는, 비디오 코딩을 위한 행렬 기반의 인트라 예측 방법들이 설명된다. 설명된 방법들은 기존의 비디오 코딩 표준들(예를 들어, 고효율 비디오 코딩(HEVC)) 및 미래의 비디오 코딩 표준들(예를 들어, VVC(Versatile Video Coding)) 또는 코덱들 양쪽 모두에 적용될 수 있다.

[0007] 한 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 예시적인 방법은, 비디오의 루마 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 코딩되어 있는지에 관한 제1 결정을 수행하는 단계 – 이 모드에서, 루마 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 백터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –; 루마 비디오 블록이 비디오의 현재 크로마 비디오 블록에 대한 크로마 인트라 모드를 결정하는데 적용 가능하다는 제2 결정을 수행하는 단계 – 상기 제2 결정을 수행하는 단계는, 현재 크로마 비디오 블록과 연관된 현재 픽처의 상단 좌측 루마 샘플에 관한 현재 크로마 비디오 블록의 상단 좌측 샘플에 기초하고, 상기 제2 결정을 수행하는 단계는 현재 비디오 유닛의 루마 비디오 블록 및/또는 컬러 포맷에 관한 현재 크로마 비디오 블록의 치수들에 기초함 –; 상기 제1 결정 및 상기 제2 결정에 기초하여, 상기 현재 크로마 비디오 블록에 이용될 상기 크로마 인트라 모드에 관한 제3 결정을 수행하는 단계; 및 상기 제3 결정에 기초하여, 상기 현재 크로마 비디오 블록과 상기 현재 크로마 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0008] 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 예시적인 방법은, 비디오의 루마 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 코딩되어 있다는 제1 결정을 수행하는 단계 – 이 모드에서, 루마 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 백터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –; 상기 제1 결정에 기초하여, 도출된 모드(derived mode)(DM)가 상기 루마 비디오 블록과 연관된 현재 크로마 비디오 블록에 대한 크로마 인트라 모드로서 이용되는지에 관한 규칙에 따른 제2 결정을 수행하는 단계; 및 상기 제2 결정에 기초하여, 현재 크로마 비디오 블록과 상기 현재 크로마 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0009] 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 예시적인 방법은, 비디오의 현재 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드로 코딩되어 있다고 결정하는 단계 – 이 모드에서, 현재 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 백터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –; 상기 결정에 기초하여, 규칙에 기초한 최대 가능성 모드(most probable mode)(MPM) 목록을 생성하는 단계 – 상기 규

칙은 상기 MPM 목록이 상기 현재 비디오 블록의 이웃 비디오 블록에 대해 생성된 또 다른 MPM 목록과는 독립적으로 생성됨을 명시함 –; 및 상기 MPM 목록을 이용하여 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0010] 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 예시적인 방법은, 비디오의 현재 비디오 블록을 행렬 기반의 인트라 예측 모드(matrix based intra prediction)(MIP) 모드를 이용하여 인코딩하는 단계 – 이 모드에서, 현재 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정되고, 규칙은 상기 MIP 모드에서 이용되는 행렬의 수와 상기 현재 비디오 블록의 치수들 사이의 관계를 기술함 –; 상기 현재 비디오 블록의 인코딩된 표현을 상기 현재 비디오 블록의 비트스트림 표현에 추가하는 단계를 포함한다.

[0011] 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 예시적인 방법은, 비디오의 현재 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 비트스트림 표현으로 인코딩되어 있다고 결정하는 단계 – 이 모드에서, 현재 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정되고, 규칙은 상기 MIP 모드에서 이용되는 행렬의 수와 상기 현재 비디오 블록의 치수들 사이의 관계를 기술함 –; 및 상기 비트스트림 표현을 파싱 및 디코딩함으로써 상기 현재 비디오 블록의 디코딩된 표현을 생성하는 단계를 포함한다.

[0012] 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 이 예시적인 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(affine linear weighted intra prediction)(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 상기 결정에 기초하여, 비ALWIP 인트라 모드에 대한 MPM 목록의 적어도 일부에 기초하여 ALWIP 모드에 대한 최고 가능성 모드(MPM) 목록의 적어도 일부를 구성하는 단계; 및 상기 ALWIP 모드에 대한 상기 MPM 목록에 기초하여, 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0013] 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 이 예시적인 방법은, 현재 비디오 블록의 루마 성분이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 상기 결정에 기초하여, 크로마 인트라 모드를 추론하는 단계, 상기 크로마 인트라 모드에 기초하여, 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0014] 역시 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 이 예시적인 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 및 상기 결정에 기초하여, 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0015] 역시 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 이 예시적인 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드와는 상이한 코딩 모드를 이용하여 코딩되어 있다고 결정하는 단계, 및 상기 결정에 기초하여, 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0016] 역시 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 이 예시적인 방법은, 현재 비디오 블록에 대해, 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 제1 예측을 생성하는 단계; 상기 제1 예측에 기초하여, 위치 의존 인트라 예측 조합(position dependent intra prediction combination)(PDPC)을 이용하여 제2 예측을 생성하는 단계; 및 상기 제2 예측에 기초하여, 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0017] 역시 또 다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하는데 이용될 수 있다. 이 예시적인 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; ALWIP 모드에 기초하여, 상기 현재 비디오 블록의 복수의 서브블록을 예측하는 단계; 및 상기 예측에 기초하여, 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0018] 역시 또 다른 대표적인 양태에서, 비디오 처리 방법이 개시된다. 이 방법은, 현재 비디오 블록에 대한 규칙에

기초하여, 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안 어파인 선형 가중된 인트라 예측(ALWIP) 모드의 이용을 나타내는 플래그의 컨텍스트를 결정하는 단계; ALWIP 모드에 기초하여, 상기 현재 비디오 블록의 복수의 서브블록을 예측하는 단계; 및 상기 예측에 기초하여, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0019] 역시 또 다른 대표적인 양태에서, 비디오 처리 방법이 개시된다. 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 및 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안, 상기 ALWIP 모드와 연관된 업샘플링 프로세스에서 현재 비디오 블록의 샘플들에 관해 적어도 2개의 필터링 스테이지를 수행하는 단계를 포함하고, 상기 적어도 2개의 필터링 스테이지 중 제1 필터링 스테이지에서의 샘플들의 제1 정밀도는 상기 적어도 2개의 필터링 스테이지 중 제2 필터링 스테이지에서의 샘플들의 제2 정밀도와는 상이하다.

[0020] 역시 또 다른 양태에서, 비디오 처리 방법이 개시된다. 이 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 및 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안, 상기 ALWIP 모드와 연관된 업샘플링 프로세스에서 현재 비디오 블록의 샘플들에 관해 적어도 2개의 필터링 스테이지를 수행하는 단계를 포함하고, 상기 업샘플링 프로세스는, 수직 및 수평 업샘플링 양쪽 모두가 수행되는 경우 고정된 순서로 수행된다.

[0021] 역시 또 다른 양태에서, 비디오 처리 방법이 개시된다. 이 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 및 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안, 상기 ALWIP 모드와 연관된 업샘플링 프로세스에서 현재 비디오 블록의 샘플들에 관해 적어도 2개의 필터링 스테이지를 수행하는 단계를 포함하고, 상기 변환은 상기 업샘플링 프로세스 이전에 전치 동작(transposing operation)을 수행하는 단계를 포함한다.

[0022] 역시 또 다른 대표적인 양태에서, 전술된 방법은 프로세서-실행가능한 코드의 형태로 구현되어 컴퓨터-판독가능 프로그램 매체에 저장된다.

[0023] 역시 또 다른 대표적인 양태에서, 전술된 방법을 수행하도록 구성되거나 동작가능한 디바이스가 개시된다. 디바이스는, 이 방법을 구현하도록 프로그래밍된 프로세서를 포함할 수 있다.

[0024] 역시 또 다른 대표적인 양태에서, 비디오 디코더 장치는 여기서 설명된 방법을 구현할 수 있다.

[0025] 개시된 기술의 상기 및 기타의 양태들 및 피처들은, 도면들, 상세한 설명 및 청구항들에서 더 상세하게 설명된다.

### 도면의 간단한 설명

[0026] 도 1은 33개의 인트라 예측 방향의 한 예를 도시한다.

도 2는 67개의 인트라 예측 모드의 한 예를 나타낸다.

도 3은 선형 모델의 가중치들의 도출에 이용된 샘플들의 위치들의 한 예를 도시한다.

도 4는 예측 블록에 이웃한 4개의 기준 라인의 한 예를 도시한다.

도 5a 및 도 5b는 블록 크기에 따른 서브파티션들의 예들을 도시한다.

도 6은  $4 \times 4$  블록들에 대한 ALWIP의 한 예를 도시한다.

도 7은  $8 \times 8$  블록들에 대한 ALWIP의 한 예를 도시한다.

도 8은  $8 \times 4$  블록들에 대한 ALWIP의 한 예를 도시한다.

도 9는  $16 \times 16$  블록들에 대한 ALWIP의 한 예를 도시한다.

도 10은 MPM 목록 구성에서 이용하는 이웃 블록들의 한 예를 도시한다.

도 11은 개시된 기술에 따른 행렬 기반의 인트라 예측을 위한 한 예시적인 방법의 플로차트를 도시한다.

도 12는 개시된 기술에 따른 행렬 기반의 인트라 예측을 위한 또 다른 예시적인 방법의 플로차트를 도시한다.

도 13은 개시된 기술에 따른 행렬 기반의 인트라 예측을 위한 역시 또 다른 예시적인 방법의 플로차트를 도시한

다.

도 14는 개시된 기술에 따른 행렬 기반의 인트라 예측을 위한 역시 또 다른 예시적인 방법의 플로차트를 도시한다.

도 15는 본 문서에 설명된 시각적 미디어 디코딩 또는 시각적 미디어 인코딩 기술을 구현하기 위한 하드웨어 플랫폼의 한 예의 블록도이다.

도 16은 이웃 블록들의 한 예를 도시한다.

도 17은 제안된 감축된 경계 샘플들 생성의 한 예이다.

도 18은 원래의 복원된 이웃 샘플들을 이용한 제안된 업샘플링의 한 예를 도시한다.

도 19는 비디오 디코더의 한 예를 나타내는 블록도이다.

도 20은 여기서 개시된 다양한 기술이 구현될 수 있는 한 예시적인 비디오 처리 시스템을 도시하는 블록도이다.

도 21은 본 개시내용의 기술들을 이용할 수 있는 한 예시적인 비디오 코딩 시스템을 나타내는 블록도이다.

도 22는 비디오 인코더의 한 예를 나타내는 블록도이다.

도 23 내지 도 27은 개시된 기술에 따른 행렬 기반의 인트라 예측을 위한 추가적인 예시적인 방법들의 플로차트를 도시한다.

## 발명을 실시하기 위한 구체적인 내용

[0027]

고해상도 비디오에 대한 수요가 증가함에 따라, 비디오 코딩 방법들 및 기술들은 현대 기술에서 유비쿼터스이다. 비디오 코덱은 전형적으로 디지털 비디오를 압축하거나 압축해제하는 전자 회로 또는 소프트웨어를 포함하고, 더 높은 코딩 효율성을 제공하기 위해 지속적으로 개선되고 있다. 비디오 코덱은 압축되지 않은 비디오를 압축된 포맷으로 또는 그 반대로 변환한다. 비디오 품질, (비트 레이트에 의해 결정되는) 비디오를 나타내는데 이용되는 데이터의 양, 인코딩 및 디코딩 알고리즘들의 복잡성, 데이터 손실들 및 오류들에 대한 민감도, 편집 용이성, 랜덤 액세스, 및 종단간 지연(레이턴시) 사이에는 복잡한 관계들이 존재한다. 압축된 포맷은 대개, 표준 비디오 압축 명세, 예를 들어 HEVC(High Efficiency Video Coding) 표준(H.265 또는 MPEG-H Part 2라고도 알려짐), 마무리 예정인 VVC(Versatile Video Coding) 표준, 또는 기타의 현재 및/또는 미래의 비디오 코딩 표준들을 준수한다.

[0028]

개시된 기술의 실시예들은 런타임 성능을 개선하기 위해 기존의 비디오 코딩 표준들(예를 들어, HEVC, H.265) 및 미래의 표준들에 적용될 수 있다. 섹션 제목들은 본 문서에서 설명의 가독성을 향상시키기 위해 이용되며 논의 또는 실시예들(및/또는 구현들)을 각각의 섹션들로만 어떤 식으로든 제한하지 않는다.

[0029]

### 1 HEVC에 대한 간략한 검토

[0030]

#### 1.1 HEVC/H.265에서의 인트라 예측

[0031]

인트라 예측은, 고려된 컬러 채널에서 이전에 복원된 샘플들을 이용하여 주어진 TB(transform block; 변환 블록)에 대한 샘플들을 생성하는 것을 포함한다. 인트라 예측 모드는 루마 및 크로마 채널들에 대해 별개로 시그널링되며, 크로마 채널 인트라 예측 모드는 'DM\_CHROMA' 모드를 통해 루마 채널 인트라 예측 모드에 선택적으로 의존한다. 인트라 예측 모드는 PB(prediction block; 예측 블록) 레벨에서 시그널링되지만, CU에 대한 잔차 퀘드 트리 계층구조에 따라 TB 레벨에서 인트라 예측 프로세스가 적용됨으로써, 1개 TB의 코딩이 CU 내의 다음 TB의 코딩에 영향을 미치는 것을 허용하므로, 기준 값들로서 이용되는 샘플까지의 거리를 감축시킨다.

[0032]

HEVC는, 35개의 인트라 예측 모드 – DC 모드, 평면 모드 및 33개의 방향성, 또는 '각도' 인트라 예측 모드를 포함한다. 33개의 각도 인트라 예측 모드가 도 1에 나와 있다.

[0033]

크로마 컬러 채널과 연관된 PB들의 경우, 인트라 예측 모드는, 평면, DC, 수평, 수직, 'DM\_CHROMA' 모드 또는 때로는 대각선 모드 '34'로서 명시된다.

[0034]

크로마 포맷들 4:2:2 및 4:2:0의 경우, 크로마 PB는 2개 또는 4개의 (각각) 루마 PB와 중첩될 수 있다; 이 경우 DM\_CHROMA에 대한 루마 방향은 이를 루마 PB들의 상단 좌측에서 취해진다는 점에 유의한다.

[0035]

DM\_CHROMA 모드는, 루마 컬러 채널 PB의 인트라 예측 모드가 크로마 컬러 채널 PB들에 적용됨을 나타낸다. 이

것은 비교적 일반적이기 때문에, intra\_chroma\_pred\_mode의 최고-가능성-모드 코딩 방식은 이 모드가 선택되도록 편향된다.

## [0036] 2. VVC에서 인트라 예측의 예들

### [0037] 2.1 67개의 인트라 예측 모드를 이용한 인트라 모드 코딩

자연스런 비디오에서 나타나는 임의의 엣지 방향들을 캡처하기 위해, 방향성 인트라 모드의 수는, HEVC에서 이 용될 때, 33에서 65로 확장된다. 추가적인 방향 모드들은 도 2에서 빨간색 점선 화살표들로 표시되며, 평면 및 DC 모드는 동일하게 유지된다. 이를 더 조밀한 방향성 인트라 예측 모드들은, 모든 블록 크기에 대해, 및 루마 및 크로마 인트라 예측 양쪽 모두에 대해 적용된다.

## [0039] 2.2 교차-성분 선형 모델(CCLM)의 예들

일부 실시예에서, 및 교차-성분 중복성을 감축시키기 위해, 다음과 같이 선형 모델을 이용함으로써 동일한 CU의 복원된 루마 샘플들에 기초하여 크로마 샘플들이 예측되는 JEM에서 교차-성분 선형 모델(CCLM) 예측 모드(LM이라고도 함)가 이용된다:

$$pred_C(i, j) = \alpha \cdot rec_L'(i, j) + \beta \quad (1)$$

여기서,  $pred_C(i, j)$  는 CU에서 예측된 크로마 샘플을 나타내고,  $rec_L'(i, j)$  는 동일한 CU의 다운샘플링된 복원된 루마 샘플을 나타낸다. 선형 모델 파라미터  $\alpha$  및  $\beta$  는, 다운샘플링된 이웃 루마 샘플 세트 내부의 최소 샘플 값과 최대 샘플 값을 갖는 루마 샘플, 및 그들의 대응하는 크로마 샘플들인, 2개의 샘플로부터의 루마 값들과 크로마 값들 사이의 관계로부터 도출된다. 도 3은 CCLM 모드에 포함된 현재 블록의 샘플과 좌측 및 위쪽 샘플들의 위치의 한 예를 도시한다.

[0043] 이 파라미터 계산은 디코딩 프로세스의 일부로 수행되고, 인코더 검색 동작과 같지 않다. 그 결과,  $\alpha$  및  $\beta$  값을 디코더에 전달하기 위해 어떠한 신박스도 이용되지 않는다.

[0044] 크로마 인트라 모드 코딩의 경우, 크로마 인트라 모드 코딩을 위해 총 8개의 인트라 모드가 허용된다. 이를 모드는, 5개의 전통적인 인트라 모드와 3개의 교차-성분 선형 모델 모드(CCLM, LM\_A 및 LM\_L)를 포함한다. 크로마 모드 코딩은 대응하는 루마 블록의 인트라 예측 모드에 직접적으로 의존한다. I개 슬라이스에서 루마 및 크로마 성분들에 대한 별개의 블록 분할 구조가 인에이블되기 때문에, 하나의 크로마 블록은 복수의 루마 블록에 대응할 수 있다. 따라서, 크로마 DM 모드의 경우, 현재 크로마 블록의 중심 위치를 커버하는 대응하는 루마 블록의 인트라 예측 모드가 직접 상속된다.

## [0045] 2.3 복수의 기준 라인(MRL) 인트라 예측

[0046] 복수의 기준 라인(MRL) 인트라 예측은 인트라 예측을 위해 더 많은 기준 라인을 이용한다. 도 4에서, 4개의 기준 라인의 예가 나와 있고, 여기서, 세그먼트 A 및 F의 샘플들은 복원된 이웃 샘플들로부터 가져온 것이 아니라, 세그먼트 B 및 E로부터 가장 가까운 샘플들로 각각 페팅된다. HEVC 인트라-픽치 예측은 가장 가까운 기준 라인(즉, 기준 라인 0)을 이용한다. MRL에서, 2개의 추가적인 라인(기준 라인 1 및 기준 라인 3)이 이용된다. 선택된 기준 라인의 인덱스(mrl\_idx)는 시그널링되고 인트라 예측자를 생성하는데 이용된다. 0보다 큰 기준 라인 idx의 경우, MPM 목록에는 추가적인 기준 라인 모드들만 포함하고 나머지 모드 없이 신호 mpm 인덱스만 포함한다.

## [0047] 2.4 인트라 서브파티션들(ISP)

[0048] ISP(Intra Sub-Partitions) 툴은 루마 인트라 예측된 블록들을 블록 크기에 따라 수직 또는 수평으로 2개 또는 4개의 서브파티션으로 분할한다. 예를 들어, ISP에 대한 최소 블록 크기는 4x8(또는 8x4)이다. 블록 크기는 4x8(또는 8x4)보다 크다면, 대응하는 블록은 4개의 서브파티션으로 분할된다. 도 5는 2개의 가능성의 예들을 도시한다. 모든 서브파티션은 적어도 16개의 샘플을 갖는 조건을 충족한다.

[0049] 각각의 서브파티션에 대해, 예측 신호에 잔차 신호를 추가함으로써 복원된 샘플이 획득된다. 여기서, 엔트로피 디코딩, 역양자화, 역변환 등의 프로세스들에 의해 잔차 신호가 생성된다. 따라서, 각각의 서브파티션의 복원된 샘플 값들은 다음 서브파티션의 예측을 생성하는데 이용가능하고, 각각의 서브파티션은 반복적으로

처리된다. 또한 처리될 첫 번째 서브파티션은 CU의 상단 좌측 샘플을 포함하고 계속해서 아래쪽(수평 분할) 또는 우측(수직 분할)으로 이어지는 서브파티션이다. 그 결과, 서브파티션 예측 신호들을 생성하는데 이용되는 기준 샘플들은 라인들의 좌측과 위쪽들에만 위치한다. 모든 서브파티션은 동일한 인트라 모드를 공유한다.

## [0050] 2.5 어파인 선형 가중된 인트라 예측(ALWIP 또는 행렬 기반의 인트라 예측)

[0051] 어파인 선형 가중된 인트라 예측(ALWIP, 일명, 행렬 기반의 인트라 예측(MIP))은 JVET-N0217에서 제안된다.

[0052] JVET-N0217에서, 2개의 테스트가 수행된다. 테스트 1에서, ALWIP는 8K 바이트의 메모리 제한과 샘플당 최대 4개의 곱셈으로 설계된다. 테스트 2는 테스트 1과 유사하지만, 메모리 요건 및 모델 아키텍처 측면에서 설계를 더욱 단순화한다.

[0053] ○ 모든 블록 형상에 대한 단일 세트의 행렬들 및 오프셋 벡터들.

[0054] ○ 모든 블록 형상에 대한 19개로의 모드 수 감축.

[0055] ○ 5760 10비트 값들, 즉, 7.20 칼로바이트로의 메모리 요건의 감축.

[0056] ○ 예측된 샘플들의 선형 보간은 첫 번째 테스트에서와 같이 반복 보간을 대체하여 방향당 단일의 단계에서 실행된다.

### [0057] 2.5.1 JVET-N0217의 테스트 1

[0058] 폭  $W$  및 높이  $H$  의 직사각형 블록의 샘플을 예측하기 위해, 어파인 선형 가중된 인트라 예측(ALWIP)은 블록 좌측의  $H$ 개의 복원된 이웃 경계 샘플들의 한 라인과 블록 위의  $W$  복원된 이웃 경계 샘플들의 한 라인을 입력으로서 취한다. 복원된 샘플들이 이용가능하지 않다면, 이들은 종래의 인트라 예측에서와 같이 생성된다.

[0059] 예측 신호의 생성은 다음과 같은 3개 단계에 기초한다:

[0060] 경계 샘플들 중에서,  $W = H = 4$  의 경우 4개 샘플, 기타 모든 경우 8개 샘플이 평균화에 의해 추출된다.

[0061] 오프셋의 추가가 뒤따르는 행렬 벡터 곱셈은 평균된 샘플들을 입력으로서 이용하여 실행된다. 결과는 원래 블록의 서브샘플링된 샘플 세트에 관한 감축된 예측 신호이다.

[0062] 나머지 위치들에서의 예측 신호는, 각각의 방향에서의 단일 단계 선형 보간인 선형 보간에 의해 서브샘플링된 세트에 관한 예측 신호로부터 생성된다.

[0063] 예측 신호를 생성하는데 필요한 행렬들 및 오프셋 벡터들은 행렬들의 3개 세트  $S_0, S_1, S_{2\triangle}$ 로부터 취해진다.

세트  $S_0$  는, 각각 16개의 행 및 4개의 열과 각각 크기가 16인 18개의 오프셋 벡터  $b_0^i, i \in \{0, \dots, 17\}$  를 갖는 18개의 행렬  $A_0^i, i \in \{0, \dots, 17\}$  로 구성된다. 그 세트의 행렬들 및 오프셋 벡터들은 크기  $4 \times 4$ 인 블록들에 이용된다. 세트  $S_1$  은 10개의 행렬  $A_1^i, i \in \{0, \dots, 9\}$  로 구성되며, 각각의 행렬은 16개의 행 및 8개의

열과, 각각 크기가 16인 10개의 오프셋 벡터  $b_1^i, i \in \{0, \dots, 9\}$  를 갖는다. 그 세트의 행렬들 및 오프셋 벡터들은 크기를  $4 \times 8$ ,  $8 \times 4$ , 및  $8 \times 8$ 인 블록들에 이용된다. 마지막으로, 세트  $S_2$ 는 6개의 행렬  $A_2^i, i \in \{0, \dots, 5\}$ 로 구성되고, 각각의 행렬은 64개의 행 및 8개의 열과, 크기가 64인 6개의 오프셋 벡터  $b_2^i, i \in \{0, \dots, 5\}$  를 갖는다. 그 세트의 행렬들 및 오프셋 벡터들 또는 이를 행렬들 및 오프셋 벡터들의 부분들은 다른 모든 블록 형상들에 이용된다.

[0064] 행렬 벡터 곱의 계산에 요구되는 곱셈의 총 수는 항상  $4 \times W \times H$  보다 작거나 같다. 즉, ALWIP 모드에 대해 샘플당 최대 4개의 곱셈이 요구된다.

### [0065] 2.5.2 경계의 평균화

[0066] 제1 단계에서 입력 경계들  $bdry^{top}$  및  $bdry^{left}$  는 더 작은 경계들  $bdry_{red}^{top}$  및  $bdry_{red}^{left}$ 로 감축된다.

여기서,  $bdry_{red}^{top}$  및  $bdry_{red}^{left}$  양쪽 모두는  $4 \times 4$  블록의 경우 2개의 샘플로 구성되고, 다른 모든 경우 양쪽 모두는 4개의 샘플로 구성된다.

[0067]  $4 \times 4$  블록의 경우,  $0 \leq i < 2$ 에 대해, 다음과 같이 정의한다

$$bdry_{red}^{top}[i] = \left( \left( \sum_{j=0}^1 bdry^{top}[i \cdot 2 + j] \right) + 1 \right) \gg 1$$

[0068] 그리고, 유사하게  $bdry_{red}^{left}$  을 정의한다.

[0070] 그렇지 않고, 블록 폭  $W$ 가  $W = 4 \cdot 2^k$ 로 주어지면,  $0 \leq i < 4$ 에 대해, 다음과 같이 정의한다

$$bdry_{red}^{top}[i] = \left( \left( \sum_{j=0}^{2^k-1} bdry^{top}[i \cdot 2^k + j] \right) + (1 \ll (k-1)) \right) \gg k$$

[0071] 그리고, 유사하게  $bdry_{red}^{left}$  을 정의한다.

[0073] 2개의 감축된 경계  $bdry_{red}^{top}$  및  $bdry_{red}^{left}$  는 감축된 경계 벡터  $bdry_{red}$ 에 연결되어, 형상  $4 \times 4$ 의 블록들의 경우 크기가 4이고 다른 모든 형상의 블록들에 대한 크기가 8이다.  $mode$ 가 ALWIP 모드를 지칭한다면, 이 연결은 다음과 같이 정의된다:

$$bdry_{red} = \begin{cases} [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } W = H = 4 \text{ and } mode < 18 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } W = H = 4 \text{ and } mode \geq 18 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) = 8 \text{ and } mode < 10 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) = 8 \text{ and } mode \geq 10 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) > 8 \text{ and } mode < 6 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) > 8 \text{ and } mode \geq 6. \end{cases}$$

[0075] 마지막으로, 서브샘플링된 예측 신호의 보간을 위해, 큰 블록들에서, 평균된 경계의 제2 버전이 필요하다. 즉,  $\min(W, H) > 8$ 이고  $W \geq H$ 이면,  $W = 8 * 2^l$ 로 쓸 수 있고,  $0 \leq i < 8$ 에 대해, 다음과 같이 정의 한다

$$bdry_{redII}^{top}[i] = \left( \left( \sum_{j=0}^{2^l-1} bdry^{top}[i \cdot 2^l + j] \right) + (1 \ll (l-1)) \right) \gg l.$$

[0077]  $\min(W, H) > 8$ 이고  $H > W$ 이면,  $bdry_{redII}^{left}$  을 정의한다.

### 2.5.3 행렬 벡터 곱셈에 의한 감축된 예측 신호의 생성

[0079] 감축된 입력 벡터  $bdry_{red}$ 로부터, 감축된 예측 신호  $pred_{red}$ 가 생성된다. 후자의 신호는 폭  $W_{red}$  및

높이  $H_{red}$  의 다운샘플링된 블록에 관한 신호이다. 여기서,  $W_{red}$  및  $H_{red}$ 는 다음과 같이 정의된다:

$$W_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(W, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(H, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

[0080] [0081] 감축된 예측 신호  $pred_{red}$ 는, 행렬 벡터 곱을 계산하고 오프셋을 더함으로써 계산된다:

$$pred_{red} = A \cdot bdry_{red} + b$$

[0083] [0084]  $A$ 는,  $W_{red} \cdot H_{red}$  개의 행과,  $W = H = 4$ 인 경우 4개의 열, 다른 경우에는 8개의 열을 갖는 행렬이다.  $b$ 는 크기  $W_{red} \cdot H_{red}$ 의 벡터이다.

[0084] 행렬  $A$  및 벡터  $b$ 는 오프셋  $S_0, S_1, S_2$  중 하나로부터 다음과 같이 취해진다. 인덱스  $idx = idx(W, H)$ 을 다음과 같이 정의한다:

$$idx(W, H) = \begin{cases} 0 & \text{for } W = H = 4 \\ 1 & \text{for } \max(W, H) = 8 \\ 2 & \text{for } \max(W, H) > 8. \end{cases}$$

[0086]  $m$ 을 다음과 같이 둔다:

$$m = \begin{cases} mode & \text{for } W = H = 4 \text{ and } mode < 18 \\ mode - 17 & \text{for } W = H = 4 \text{ and } mode \geq 18 \\ mode & \text{for } \max(W, H) = 8 \text{ and } mode < 10 \\ mode - 9 & \text{for } \max(W, H) = 8 \text{ and } mode \geq 10 \\ mode & \text{for } \max(W, H) > 8 \text{ and } mode < 6 \\ mode - 5 & \text{for } \max(W, H) > 8 \text{ and } mode \geq 6. \end{cases}$$

[0088] 그 다음,  $idx \leq 1$  또는  $idx = 2$  및  $\min(W, H) > 4$ 이면,  $A = A_{idx}^m$  및  $b = b_{idx}^m$  둔다.

[0089]  $idx = 2$  및  $\min(W, H) = 4$ 의 경우,  $A$ 를,  $W = 4$  경우 다운샘플링된 블록의 홀수 x-좌표에 대응하고,  $H = 4$ 인 경우에는 다운샘플링된 블록의 홀수 y-좌표에 대응하는,  $A_{idx}^m$ 의 모든 행을 생략함으로서 발생하는 행렬이라고 하자.

[0089] 마지막으로, 감축된 예측 신호는 다음과 같은 경우들에서 그 전치(transpose)로 대체된다:

- $W = H = 4$  and  $mode \geq 18$
- $\max(W, H) = 8$  and  $mode \geq 10$
- $\max(W, H) > 8$  and  $mode \geq 6$

[0090] [0091]  $pred_{red}$ 의 계산에 필요한 곱셈의 수는  $W = H = 4$ 의 경우에 4인데, 그것은 이 경우에  $A$ 가 4개의 열과

16개의 행을 갖기 때문이다. 다른 모든 경우에는,  $A$ 는 8개의 열과  $W_{red} \cdot H_{red}$  개의 행을 가지며, 이를 경우

$8 \cdot W_{red} \cdot H_{red} \leq 4 \cdot W \cdot H$   
에 개의 곱셈이 요구된다는 것을 즉시 확인한다, 즉, 또한 이를 경우에,  $pred_{red}$

을 계산하기 위해 샘플당 최대 4개의 곱셈이 필요하다.

#### [0092] 2.5.4 전체 ALWIP 프로세스의 예시

[0093] 평균화, 행렬 벡터 곱셈 및 선형 보간의 전체 프로세스가 도 6 내지 도 9에서 상이한 형상들에 대해 예시되어 있다. 나머지 형상들은 도시된 경우들 중 하나인 것으로 취급된다는 점에 유의한다.

[0094] 1.  $4 \times 4$  블록이 주어지면, ALWIP는 경계의 각각의 축을 따라 2개의 평균을 취한다. 결과적인 4개의 입력 샘플은 행렬 벡터 곱셈에 들어간다. 행렬들은 세트  $S_0$ 로부터 취해진다. 오프셋을 추가하면, 16개의 최종 예측 샘플이 나온다. 예측 신호의 생성에 대해 선형 보간이 필요하지 않다. 따라서, 샘플당 총  $(4 \cdot 16)/(4 \cdot 4) = 4$  개의 곱셈이 수행된다.

[0095] 2.  $8 \times 8$  블록이 주어지면, ALWIP는 경계의 각각의 축을 따라 4개의 평균을 취한다. 결과적인 8개의 입력 샘플은 행렬 벡터 곱셈에 들어간다. 행렬들은 세트  $S_1$ 로부터 취해진다. 이것은 예측 블록의 홀수 위치들에서 16 개의 샘플을 생성한다. 따라서, 샘플당 총  $(8 \cdot 16)/(8 \cdot 8) = 2$  개의 곱셈이 수행된다. 오프셋을 추가한 후, 이들 샘플들은 감축된 상단 경계를 이용하여 수직으로 보간된다. 원래의 좌측 경계를 이용하여 수평 보간이 뒤따른다.

[0096] 3.  $8 \times 4$  블록이 주어지면, ALWIP는 경계의 수평 축을 따라 4개의 평균 및 좌측 경계에서 4개의 원래 경계 값을 취한다. 결과적인 8개의 입력 샘플은 행렬 벡터 곱셈에 들어간다. 행렬들은 세트  $S_1$ 로부터 취해진다. 이것은 예측 블록의 홀수 수평 및 각각의 수직 위치들에서 16개의 샘플을 생성한다. 따라서, 샘플당 총  $(8 \cdot 16)/(8 \cdot 4) = 4$  개의 곱셈이 수행된다. 오프셋을 추가한 후, 이들 샘플들은 원래의 좌측 경계를 이용하여 수평으로 보간된다.

[0097] 4.  $16 \times 16$  블록이 주어지면, ALWIP는 경계의 각각의 축을 따라 4개의 평균을 취한다. 결과적인 8개의 입력 샘플은 행렬 벡터 곱셈에 들어간다. 행렬들은 세트  $S_2$ 로부터 취해진다. 이것은 예측 블록의 홀수 위치들에서 64개의 샘플을 생성한다. 따라서, 샘플당 총  $(8 \cdot 64)/(16 \cdot 16) = 2$  개의 곱셈이 수행된다. 오프셋을 추가한 후, 이들 샘플들은 상단 경계의 8개의 평균을 이용하여 수직으로 보간된다. 원래의 좌측 경계를 이용하여 수평 보간이 뒤따른다. 이 경우, 보간 프로세스는 어떠한 곱셈도 추가하지 않는다. 따라서, ALWIP 예측을 계산하기 위해 샘플당 총 2개의 곱셈이 요구된다.

[0098] 더 큰 형상들의 경우, 절차는 본질적으로 동일하며 샘플당 곱셈 수가 4개보다 작은지를 체크하는 것이 쉽다.

[0099]  $W > 8$ 인  $W \times 8$  블록들의 경우, 샘플들이 홀수 수평 및 각각의 수직 위치들에 제공되므로 수평 보간만이 필요하다.

[0100] 마지막으로,  $W > 8$ 인  $W \times 4$  블록들의 경우,  $A_{kbe}$ 를, 다운샘플링된 블록의 수평 축을 따라 홀수 엔트리에 대응하는 모든 행을 생략함으로써 발생하는 행렬이라고 하자. 따라서, 출력 크기는 32이고 다시 수평 보간만 수행된다.

[0101] 전치된 사례들은 그에 따라 취급된다.

#### [0102] 2.5.5 단일 단계 선형 보간

[0103]  $\max(W, H) \geq 8$ 인  $W \times H$  블록의 경우, 예측 신호는 선형 보간에 의해  $W_{red} \times H_{red}$ 에 의해 감축된 예측 신호  $pred_{red}$ 로부터 발생한다. 블록 형상에 따라, 선형 보간은, 수직, 수평 또는 양쪽 방향에서 이루어진다. 선형 보간이 양쪽 방향에서 적용되는 경우,

이면 수평 방향에서 먼저 적용되고, 그렇지 않으면 수직 방향에서 먼저 적용된다.

[0104] 일반성을 잃지 않고  $\max(W, H) \geq 8$ 이고  $W \geq H$ 인  $W \times H$  블록을 고려하자. 그 다음, 1차원 선형 보간

이 다음과 같이 수행된다. 일반성을 잃지 않고, 수직 방향의 선형 보간을 설명하는 것으로 충분하다. 먼저, 감축된 예측 신호는 경계 신호에 의해 상단으로 확장된다. 수직 업샘플링 인자  $U_{ver} = H/H_{red}$  를 정의하고  $U_{ver} = 2^{u_{ver}} > 1$  라고 기재한다. 그 다음, 확장된 감축된 예측 신호를 다음과 같이 정의한다

$$[0105] \quad pred_{red}[x][-1] = \begin{cases} bdry_{red}^{top}[x] & \text{for } W = 8 \\ bdry_{redII}^{top}[x] & \text{for } W > 8. \end{cases}$$

[0106] 그 다음, 이 확장된 감축된 예측 신호로부터, 수직 선형 보간된 예측 신호는 다음과 같이 생성된다

$$[0107] \quad pred_{red}^{ups,ver}[x][U_{ver} \cdot y + k] = ((U_{ver} - k - 1) \cdot pred_{red}[x][y - 1] + (k + 1) \cdot$$

$$[0108] \quad pred_{red}[x][y] + \frac{U_{ver}}{2}) \gg u_{ver}$$

$$[0109] \quad 0 \leq x < W_{red} \quad 0 \leq y < H_{red} \quad 0 \leq k < U_{ver} \\ , \quad \text{및} \quad \text{에 대해.}$$

### 2.5.6 제안된 인트라 예측 모드들의 신호화

[0111] 인트라 모드의 각각의 CU(Coding Unit)에 대해, ALWIP 모드가 대응하는 PU(Prediction Unit)에 적용될지의 여부를 나타내는 플래그가 비트스트림에서 전송된다. 후자 인덱스의 신호화는 JVET-M0043에서와 동일한 방식으로 MRL과 조율된다. ALWIP 모드가 적용되는 경우, ALWIP 모드의 인덱스  $predmode$  는 3 MPMS를 갖는 MPM-목록을 이용하여 시그널링된다.

[0112] 여기서, MPM들의 도출은 위쪽과 좌측 PU의 인트라 모드들을 이용하여 다음과 같이 수행된다. 각각의 종래의 인트라 예측 모드  $predmode_{Angular}$  에 ALWIP 모드를 할당하는 3개의 고정 테이블  $map\_angular\_to\_alwip_{idx}$ ,  $idx \in \{0,1,2\}$  이 있다.

$$[0113] \quad predmode_{ALWIP} = map\_angular\_to\_alwip_{idx}[predmode_{Angular}].$$

[0114] 폭  $W$  과 높이  $H$  의 각각의 PU에 대해, 인덱스를 정의한다

$$[0115] \quad idx(PU) = idx(W, H) \in \{0,1,2\}$$

[0116] 이 인덱스는 섹션 2.5.3에서와 같이 ALWIP 파라미터들이 3개 세트 중 어느 것으로부터 선택되는지를 나타낸다.

[0117] 위쪽 예측 유닛  $PU_{above}$  이 이용가능하고 현재 PU와 동일한 CTU에 속하고 인트라 모드에 있는 경우,  $idx(PU) = idx(PU_{above})$  이고 ALWIP가 ALWIP-모드  $predmode_{ALWIP}^{above}$  와 함께  $PU_{above}$ 에 적용되면, 다음과 같이 둘수 있다

$$[0118] \quad mode_{ALWIP}^{above} = predmode_{ALWIP}^{above}.$$

[0119] 위쪽 PU가 이용가능하고 현재 PU와 동일한 CTU에 속하며, 인트라 모드에 있고 종래의 인트라 예측 모드  $predmode_{Angular}^{above}$  가 위쪽 PU에 적용되는 경우, 다음과 같이 둘 수 있다

$$[0120] \quad mode_{ALWIP}^{above} = map\_angular\_to\_alwip_{idx(PU_{above})}[predmode_{Angular}^{above}].$$

[0121] 다른 모든 경우에는, 다음과 같이 둘 수 있다

$$mode_{ALWIP}^{above} = -1,$$

[0123] 이것은 이 모드가 이용가능하지 않음을 의미한다. 동일한 방식으로 그러나 좌측 PU가 현재 PU와 동일한 CTU에 속할 필요가 있다는 제한없이, 모드  $mode_{ALWIP}^{left}$ 를 도출한다.

[0124] 마지막으로, 3개의 고정된 디폴트 목록  $list_{idx}, idx \in \{0,1,2\}$ 이 제공되며, 각각의 목록은 3개의 구별되는 ALWIP 모드를 포함한다. 디폴트 목록  $list_{idx(PU)}$ 과 모드들  $mode_{ALWIP}^{above}$  및  $mode_{ALWIP}^{left}$ 로부터, -1을 디폴트 값들로 대체하고 반복을 제거함으로써 3개의 구별되는 MPM을 구성한다.

[0125] ALWIP MPM 목록 구성에 이용되는 좌측 이웃 블록 및 위쪽 이웃 블록은, 도 10에 도시된 바와 같이, A1과 B1이다.

#### 2.5.7 종래의 루마 및 크로마 인트라 예측 모드들에 대한 적응형 MPM 목록 도출

[0127] 제안된 ALWIP 모드들은 다음과 같이 종래의 인트라 예측 모드들 MPM 기반의 코딩과 조율된다. 종래의 인트라 예측 모드들에 대한 루마 및 크로마 MPM 목록 도출 프로세스들은, 고정 테이블들  $map_{alwip\_to\_angular}_{idx}, idx \in \{0,1,2\}$ 를 이용하고, 주어진 PU에 관한 ALWIP 모드  $predmode_{ALWIP}$ 를 종래의 인트라 예측 모드들 중 하나에 맵핑한다.

$$predmode_{Angular} = map_{alwip\_to\_angular}_{idx(PU)}[predmode_{ALWIP}]$$

[0129] 루마 MPM 목록 도출의 경우, ALWIP 모드  $predmode_{ALWIP}$ 를 이용하는 이웃 루마 블록과 조우할 때마다, 이 블록은 종래의 인트라 예측 모드  $predmode_{Angular}$ 를 이용하는 것처럼 취급된다. 크로마 MPM 목록 도출의 경우, 현재 루마 블록이 LWIP 모드를 이용할 때마다, ALWIP 모드를 종래의 인트라 예측 모드로 변환하기 위해 동일한 맵핑이 이용된다.

#### 2.5.8 대응하는 수정된 연구 초안

[0131] 일부 실시예에서, 이 섹션에서 설명된 바와 같이, intra\_lwip\_flag, intra\_lwip\_mpm\_flag, intra\_lwip\_mpm\_idx 및 intra\_lwip\_mpm\_remainder와 관련된 부분들이 개시된 기술의 실시예들에 기초하여 연구 초안에 추가되었다.

[0132] 일부 실시예에서, 이 섹션에서 설명된 바와 같이, <begin> 및 <end> 태그들은 개시된 기술의 실시예들에 기초한 연구 초안에 대한 추가 및 수정을 나타내는데 이용된다.

#### 선택스 테이블들

#### 코딩 유닛 선택스

coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {	기술자
if( tile_group_type != I    sps_abc_enabled_flag ) {	
if( treeType != DUAL_TREE_CHROMA )	
cu_skip_flag[ x0 ][ y0 ]	ae(v)
if( cu_skip_flag[ x0 ][ y0 ] == 0 && tile_group_type != I )	
pred_mode_flag	ae(v)

[0135]

if( (( tile_group_type == I && cu_skip_flag[ x0 ][ y0 ] == 0 )    ( tile_group_type != I && CuPredMode[ x0 ][ y0 ] != MODE_INTRA ) ) && sps_ibc_enabled_flag )	
<b>pred_mode_ibc_flag</b>	ae(v)
}	
if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {	
if( sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY )	
<b>pcm_flag[ x0 ][ y0 ]</b>	ae(v)
if( pcm_flag[ x0 ][ y0 ] ) {	
while( !byte_aligned() )	
<b>pcm_alignment_zero_bit</b>	f(l)
pcm_sample( cbWidth, cbHeight, treeType )	
} else {	
if( treeType == SINGLE_TREE    treeType == DUAL_TREE_LUMA ) {	
if( Abs( Log2( cbWidth ) - Log2( cbHeight ) ) <= 2 )	
<b>intra_lwip_flag[ x0 ][ y0 ]</b>	ae(v)
if( intra_lwip_flag[ x0 ][ y0 ] ) {	
<b>intra_lwip_mpm_flag[ x0 ][ y0 ]</b>	ae(v)
if( intra_lwip_mpm_flag[ x0 ][ y0 ] )	
<b>intra_lwip_mpm_idx[ x0 ][ y0 ]</b>	ae(v)
else	
<b>intra_lwip_mpm_remainder[ x0 ][ y0 ]</b>	ae(v)
} else {	
if( ( y0 % CtbSizeY ) > 0 )	
<b>intra_luma_ref_idx[ x0 ][ y0 ]</b>	ae(v)
if( intra_luma_ref_idx[ x0 ][ y0 ] == 0 && ( cbWidth <= MaxTbSizeY    cbHeight <= MaxTbSizeY ) && ( cbWidth * cbHeight > MinTbSizeY * MinTbSizeY ))	
<b>intra_subpartitions_mode_flag[ x0 ][ y0 ]</b>	ae(v)
if( intra_subpartitions_mode_flag[ x0 ][ y0 ] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY )	
<b>intra_subpartitions_split_flag[ x0 ][ y0 ]</b>	ae(v)

[0136]

if( intra_luma_ref_idx[ x0 ][ y0 ] == 0 &&	
intra_subpartitions_mode_flag[ x0 ][ y0 ] == 0 )	
<b>intra_luma_mpm_flag[ x0 ][ y0 ]</b>	ae(v)
if( intra_luma_mpm_flag[ x0 ][ y0 ] )	
<b>intra_luma_mpm_idx[ x0 ][ y0 ]</b>	ae(v)
else	
<b>intra_luma_mpm_remainder[ x0 ][ y0 ]</b>	ae(v)
}	
}	
if( treeType == SINGLE_TREE    treeType == DUAL_TREE_CHROMA )	
<b>intra_chroma_pred_mode[ x0 ][ y0 ]</b>	ae(v)
}	
} else if( treeType != DUAL_TREE_CHROMA ) { /* MODE_INTER or MODE_IBC */	
...	

[0137]

### 시맨틱스

[0138]

<begin>**intra\_lwip\_flag[ x0 ][ y0 ]**가 1인 것은, 루마 샘플에 대한 인트라 예측 유형이 어파인 선형 가중된 인트라 예측임을 명시한다. **intra\_lwip\_flag[ x0 ][ y0 ]**가 0인 것은, 루마 샘플들에 대한 인트라 예측 유형이 어파인 선형 가중된 인트라 예측이 아님을 명시한다.

[0139]

**intra\_lwip\_flag[ x0 ][ y0 ]**가 없는 것은, 0인 것으로 추론된다.

[0140]

선택스 요소들 **intra\_lwip\_mpm\_flag[ x0 ][ y0 ]**, **intra\_lwip\_mpm\_idx[ x0 ][ y0 ]** 및 **intra\_lwip\_mpm\_remainder [ x0 ][ y0 ]**는 루마 샘플들에 대한 어파인 선형 가중된 인트라 예측 모드를 명시한다. 어레이 인덱스들 x0, y0은, 픽처의 상단 좌측 루마 샘플에 관한 고려되는 코딩 블록의 상단 좌측 루마 샘플의 위치 (x0, y0)를 명시한다. **intra\_lwip\_mpm\_flag[ x0 ][ y0 ]**가 1일 때, 어파인 선형 가중된 인트라 예측 모드는 조항 8.4.X에 따라 이웃 인트라 예측된 코딩 유닛으로부터 추론된다.

[0141]

**intra\_lwip\_mpm\_flag[ x0 ][ y0 ]**가 없는 것은, 1이라고 추론된다.<end>

[0142]

**intra\_subpartitions\_split\_flag[ x0 ][ y0 ]**은 인트라 서브파티션 분할 유형이 수평인지 수직인지를 명시한다. **intra\_subpartitions\_split\_flag[ x0 ][ y0 ]**가 없으면, 다음과 같이 추론된다:

[0143]

- **intra\_lwip\_flag[ x0 ][ y0 ]**가 1이면, **intra\_subpartitions\_split\_flag[ x0 ][ y0 ]**가 0인 것으로 추론된다.

[0144]

- 그렇지 않으면, 다음이 적용된다:

[0145]

- **cbHeight**가 MaxTbSizeY보다 크다면, **intra\_subpartitions\_split\_flag[ x0 ][ y0 ]**가 0인 것으로 추론된다.

[0146]

- 그렇지 않으면(**cbWidth**가 MaxTbSizeY보다 큼), **intra\_subpartitions\_split\_flag[ x0 ][ y0 ]**가 1인 것으로 추론된다.

[0147]

### 디코딩 프로세스

[0148]

8.4.1 인트라 예측 모드에서 코딩된 코딩 유닛에 대한 일반 디코딩 프로세스

[0149]

이 프로세스에 대한 입력들은 다음과 같다:

[0150]

- 현재 픽처의 상단 좌측 루마 샘플에 관해 현재 코딩 블록의 상단 좌측 샘플을 명시하는 루마 위치 (xCb, yCb),

[0151]

- 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수 cbWidth,

[0152]

- 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수 cbHeight,

[0153]

- 단일 또는 이중 트리가 이용되는지를 명시하고 이중 트리가 이용되는 경우 현재 트리가 루마 또는 크로마 성분들에 대응하는지를 명시하는 변수 treeType.

- [0155] 이 프로세스의 출력은 인루프 필터링 이전의 수정된 복원된 픽처이다.
- [0156] 조항 8.7.1에 명시된 양자화 파라미터들에 대한 도출 프로세스는, 루마 위치( $xCb$ ,  $yCb$ ), 루마 샘플들에서 현재 코딩 블록의 폭  $cbWidth$  및 루마 샘플들에서 현재 코딩 블록 높이  $cbHeight$ , 및 변수  $treeType$ 을 입력들로서 취하여 기동된다.
- [0157]  $treeType \neq SINGLE\_TREE$ 와 같거나  $treeType \neq DUAL\_TREE\_LUMA$ 와 같을 때, 루마 샘플들에 대한 디코딩 프로세스는 다음과 같이 명시된다:
- $pcm\_flag[xCb][yCb]$ 가 1이라면, 복원된 픽쳐는 다음과 같이 수정된다:
- [0159]  $S_x[xCb + i][yCb + j] =$
- [0160]  $pcm\_sample\_luma[(cbHeight * j) + i] \ll (BitDepth_Y - PcmBitDepth_Y)$ , (8-6)
- [0161] with  $i = 0..cbWidth - 1$ ,  $j = 0..cbHeight - 1$
- [0162] - 그렇지 않으면, 다음이 적용된다:
- [0163] 1. 루마 인트라 예측 모드는 다음과 같이 도출된다:
- [0164] -  $intra\_lwp\_flag[xCb][yCb]$ 가 1과 같다면, 현재 코딩의 폭인 루마 위치( $xCb$ ,  $yCb$ ), 루마 샘플들에서의 현재 코딩 블록의 폭  $cbWidth$  및 루마 샘플들에서 현재 코딩 블록의 높이  $cbHeight$ 를 입력으로서 취하여 조항 8.4.X에 명시된 어파인 선형 가중된 인트라 예측 모드에 대한 도출 프로세스가 기동된다.
- [0165] - 그렇지 않으면, 루마 위치( $xCb$ ,  $yCb$ ), 루마 샘플들에서 현재 코딩 블록의 폭  $cbWidth$  및 루마 샘플들에서 현재 코딩 블록의 높이  $cbHeight$ 를 입력으로서 취하여 조항 8.4.2에 명시된 루마 인트라 예측 모드에 대한 도출 프로세스가 기동된다.
- [0166] 2. 루마 위치( $xCb$ ,  $yCb$ ), 트리 유형  $treeType$ ,  $cbWidth$ 와 동일하게 설정된 변수  $nTbW$ ,  $cbHeight$ 와 동일하게 설정된 변수  $nTbH$ ,  $IntraPredModeY[xCb][yCb]$ 와 동일하게 설정된 변수  $predModeIntra$ , 및 0으로 동일하게 설정된 변수  $cIdx$ 를 입력으로서 취하여 조항 8.4.4.1에 명시된 인트라 블록들에 대한 일반 디코딩 프로세스가 기동되고, 출력은 인루프 필터링 이전의 수정된 복원된 픽쳐이다.
- [0167] ...
- [0168] <begin>
- [0169] 8.4.X 어파인 선형 가중된 인트라 예측 모드에 대한 도출 프로세스
- [0170] 이 프로세스에 대한 입력은 다음과 같다:
- 현재 픽처의 상단 좌측 루마 샘플에 관해 현재 루마 코딩 블록의 상단 좌측 샘플을 명시하는 루마 위치 ( $xCb$ ,  $yCb$ ),
  - 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수  $cbWidth$ ,
  - 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수  $cbHeight$ .
- [0174] 이 프로세스에서 어파인 선형 가중된 인트라 예측 모드  $IntraPredModeY[xCb][yCb]$ 가 도출된다.
- [0175]  $IntraPredModeY[xCb][yCb]$ 는 다음과 같은 순서화된 단계들에 의해 도출된다:
1. 이웃 위치들 ( $xNbA$ ,  $yNbA$ ) 및 ( $xNbB$ ,  $yNbB$ )은 각각 ( $xCb-1$ ,  $yCb$ ) 및 ( $xCb$ ,  $yCb-1$ )와 동일하게 설정된다.
  2.  $X$ 가  $A$  또는  $B$ 로 대체되는 경우, 변수  $candLwipModeX$ 는 다음과 같이 도출된다:
- [0178] - 조항 6.4.X [Ed. (BB): 이웃 블록 가용성 체킹 프로세스  $tbd$ ]에 명시된 블록에 대한 가용성 도출 프로세스는, ( $xCb$ ,  $yCb$ )와 동일하게 설정된 위치( $xCurr$ ,  $yCurr$ )와 ( $xNbX$ ,  $yNbX$ )와 동일하게 설정된 이웃 위치( $xNbY$ ,  $yNbY$ )를 입력으로서 취하여 기동되고, 출력은  $availableX$ 에 할당된다.
- [0179] - 후보 어파인 선형 가중된 인트라 예측 모드  $candLwipModeX$ 는 다음과 같이 도출된다:

- [0180] - 다음과 같은 조건들 중 하나 이상이 참이라면, *candLwipModeX*는 -1로 설정된다.
- [0181] - 변수 *availableX*는 FALSE이다.
- [0182] - *CuPredMode[ xNbX ][ yNbX ]*는 *MODE\_INTRA*와 동일하지 않고 *mh\_intra\_flag[ xNbX ][ yNbX ]*는 1과 동일하지 않다.
- [0183] - *pcm\_flag[ xNbX ][ yNbX ]*는 1이다.
- [0184] - *X*는 *B*와 동일하고 *yCb*-1은  $((yCb >> CtbLog2SizeY) << CtbLog2SizeY)$ 보다 작다.
- [0185] - 그렇지 않으면, 다음이 적용된다:
  - 조항 8.4.X.1에 명시된 블록에 대한 크기 유형 도출 프로세스는, 루마 샘플들에서 현재 코딩 블록 폭 *cbWidth* 와 루마 샘플들에서 현재 코딩 블록의 높이 *cbHeight*를 입력으로서 취하여 기동되고, 출력은 변수 *sizeId*에 할당된다.
  - *intra\_lwip\_flag[ xNbX ][ yNbX ]*가 1이면, 조항 8.4.X.1에 명시된 블록에 대한 크기 유형 도출 프로세스는, 루마 샘플들에서 이웃 코딩 블록의 폭 *nbWidthX*와 루마 샘플들에서 이웃 코딩 블록의 높이 *nbHeightX*를 입력으로서 취하여 기동되고, 출력은 변수 *sizeIdX*에 할당된다.
- [0188] - *sizeId*가 *sizeIdX*이면, *candLwipModeX*는 *IntraPredModeY[ xNbX ][ yNbX ]*로 설정된다.
- [0189] - 그렇지 않으면, *candLwipModeX*가 -1로 설정된다.
- [0190] - 그렇지 않으면, 표 8-X1에 명시된 대로 *IntraPredModeY[ xNbX ][ yNbX ]* 및 *sizeId*를 이용하여 *candLwipModeX*가 도출된다.
- [0191] 3.  $x=0..2$ 인 *candLwipModeList[ x ]*는, 표 8-X2에 명시된 대로 *IwipMpmCand[ sizeId ]*를 이용하여 다음과 같이 도출된다:
  - *candLwipModeA* 및 *candLwipModeB* 양쪽 모두가 -1이라면, 다음이 적용된다:
    - candLwipModeList[ 0 ] = IwipMpmCand[ sizeId ][ 0 ]*(8-X1)
    - candLwipModeList[ 1 ] = IwipMpmCand[ sizeId ][ 1 ]*(8-X2)
    - candLwipModeList[ 2 ] = IwipMpmCand[ sizeId ][ 2 ]*(8-X3)
  - 그렇지 않으면, 다음이 적용된다:
    - *candLwipModeA*가 *candLwipModeB*거나 *candLwipModeA* 또는 *candLwipModeB*가 -1이라면, 다음이 적용된다:
      - candLwipModeList[ 0 ] = ( candLwipModeA != -1 ) ? candLwipModeA : candLwipModeB*(8-X4)
    - *candLwipModeList[ 0 ]*가 *IwipMpmCand[ sizeId ][ 0 ]*이면, 다음이 적용된다:
      - candLwipModeList[ 1 ] = IwipMpmCand[ sizeId ][ 1 ]*(8-X5)
      - candLwipModeList[ 2 ] = IwipMpmCand[ sizeId ][ 2 ]*(8-X6)
- [0203] - 그렇지 않으면, 다음이 적용된다:
  - candLwipModeList[ 1 ] = IwipMpmCand[ sizeId ][ 0 ]*(8-X7)
  - candLwipModeList[ 2 ] = ( candLwipModeList[ 0 ] != IwipMpmCand[ sizeId ][ 1 ] ) ? IwipMpmCand[ sizeId ][ 1 ] : IwipMpmCand[ sizeId ][ 2 ]*(8-X8)
- [0206] - 그렇지 않으면, 다음이 적용된다:
  - candLwipModeList[ 0 ] = candLwipModeA*(8-X9)
  - candLwipModeList[ 1 ] = candLwipModeB*(8-X10)
- [0209] - *candLwipModeA* 및 *candLwipModeB* 양쪽 모두가 *IwipMpmCand[ sizeId ][ 0 ]*이 아니면, 다음이 적용된다:
  - candLwipModeList[ 2 ] = IwipMpmCand[ sizeId ][ 0 ]*(8-X11)

- [0211] - 그렇지 않으면, 다음이 적용된다:
- [0212] -  $candLwipModeA$  및  $candLwipModeB$  양쪽 모두가  $IwipMpmCand[ sizeId ][ 1 ]$ 이 아니면, 다음이 적용된다:
- [0213]  $candLwipModeList[ 2 ] = IwipMpmCand[ sizeId ][ 1 ]$ (8-X12)
- [0214] - 그렇지 않으면, 다음이 적용된다:
- [0215]  $candLwipModeList[ 2 ] = IwipMpmCand[ sizeId ][ 2 ]$ (8-X13)
- [0216] 4.  $IntraPredModeY[ xCb ][ yCb ]$ 는 다음 절차를 적용하여 도출된다:
- [0217] -  $intra_lwip_mpm_flag[ xCb ][ yCb ]$ 가 1이면,  $IntraPredModeY[ xCb ][ yCb ]$ 가  $candLwipModeList[ intra_lwip_mpm_idx[ xCb ][ yCb ] ]$ 로 설정된다.
- [0218] - 그렇지 않으면,  $IntraPredModeY[ xCb ][ yCb ]$ 는 다음과 같은 순서화된 단계들을 적용하여 도출된다:
- [0219] 1.  $i = 0..1$ 이고 각각의  $i, j = (i + 1)..2$ 에 대해  $candLwipModeList[ i ]$ 가  $candLwipModeList[ j ]$ 보다 크면, 양쪽 값들은 다음과 같이 스왑된다:
- $$( candLwipModeList[ i ], candLwipModeList[ j ] ) = Swap( candLwipModeList[ i ], candLwipModeList[ j ] )$$
- (8-X14)
- [0220] 2.  $IntraPredModeY[ xCb ][ yCb ]$ 는 다음과 같은 순서화된 단계들에 의해 도출된다:
- [0221] i.  $IntraPredModeY[ xCb ][ yCb ]$ 는  $intra_lwip_mpm_remainder[ xCb ][ yCb ]$ 와 동일하게 설정된다.
- [0222] ii.  $i$ 가 0 내지 2인 경우,  $IntraPredModeY[ xCb ][ yCb ]$ 가  $candLwipModeList[ i ]$ 보다 크거나 같다면,  $IntraPredModeY[ xCb ][ yCb ]$ 값이 1씩 증가된다.
- [0223]  $x = xCb..xCb + cbWidth-1$  및  $y = yCb..yCb + cbHeight-1$ 인 변수  $IntraPredModeY[ x ][ y ]$ 는  $IntraPredModeY[ xCb ][ yCb ]$ 와 동일하게 설정된다.
- [0224] 8.4.X.1 예측 블록 크기 유형에 대한 도출 프로세스
- [0225] a) 프로세스에 대한 입력은 다음과 같다:
- [0226] - 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수  $cbWidth$ ,
- [0227] - 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수  $cbHeight$ .
- [0228] b) 프로세스의 출력은 변수  $sizeId$ 이다.
- [0229] 변수  $sizeId$ 는 다음과 같이 도출된다:
- [0230] -  $cbWidth$ 와  $cbHeight$  양쪽 모두가 4이면,  $sizeId$ 는 0으로 설정된다.
- [0231] - 그렇지 않고  $cbWidth$ 와  $cbHeight$  양쪽 모두가 8보다 작거나 같으면  $sizeId$ 는 1로 설정된다.
- [0232] - 그렇지 않으면,  $sizeId$ 는 2로 설정된다.

[0234]

표 8-X1 – 인트라 예측과 어파인 선형 가중된 인트라 예측 모드들 사이의 맵핑 명세

IntraPredModeY[ xNbX ][ yNbX ]	블록 크기 유형 sizeId		
	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
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

[0235]

표 8-X2 – 어파인 선형 가중된 인트라 예측 후보 모드들의 명세

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

[0237]

[0238] &lt;end&gt;

[0239] 8.4.2. 루마 인트라 예측 모드의 도출 프로세스

[0240] ① 프로세스에 대한 입력은 다음과 같다:

[0241] - 현재 뷰처의 상단 좌측 루마 샘플에 관해 현재 루마 코딩 블록의 상단 좌측 샘플을 명시하는 루마 위치 ( $xCb$ ,  $yCb$ ),

- [0242] - 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수 *cbWidth*,
- [0243] - 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수 *cbHeight*.
- [0244] 이 프로세스에서, 루마 인트라 예측 모드 *IntraPredModeY[ xCb ][ yCb ]*가 도출된다.
- [0245] 표 8-1은 인트라 예측 모드 *IntraPredModeY[ xCb ][ yCb ]*에 대한 값 및 연관된 이름들을 명시한다.
- [0246] 표 8-1 – 인트라 예측 모드 및 연관된 이름들의 명세

인트라 예측 모드	연관된 이름
0	<i>INTRA_PLANAR</i>
1	<i>INTRA_DC</i>
2..66	<i>INTRA_ANGULAR2..INTRA_ANGULAR66</i>
81..83	<i>INTRA_LT_CCLM, INTRA_L_CCLM, INTRA_T_CCLM</i>

- [0247]
- [0248] 유의사항 –: 인트라 예측 모드들 *INTRA\_LT\_CCLM*, *INTRA\_L\_CCLM* 및 *INTRA\_T\_CCLM*은 크로마 성분들에만 적용 가능하다.
- [0249] *IntraPredModeY[ xCb ][ yCb ]*는 다음과 같은 순서화된 단계들에 의해 도출된다:
- [0250] 1. 이웃 위치들 (*xNbA*, *yNbA*) 및 (*xNbB*, *yNbB*)는 각각 (*xCb-1*, *yCb + cbHeight-1*) 및 (*xCb + cbWidth-1*, *yCb-1*)와 동일하게 설정된다.
  - [0251] 2. *X*가 *A* 또는 *B*로 대체되는 경우, 변수 *candIntraPredModeX*는 다음과 같이 도출된다:
    - [0252] - 조항 <begin> 6.4.X [Ed. (BB): 이웃 블록 가용성 체킹 프로세스 *tbd*] <end>에 명시된 블록에 대한 가용성 도출 프로세스는, (*xCb*, *yCb*)와 동일하게 설정된 위치(*xCurr*, *yCurr*)와 (*xNbX*, *yNbX*)와 동일하게 설정된 이웃 위치 (*xNbY*, *yNbY*)를 입력들로서 취하여 기동되고, 출력은 *availableX*에 할당된다.
  - [0253] - 후보 인트라 예측 모드 *candIntraPredModeX*는 다음과 같이 도출된다:
  - [0254] - 다음과 같은 조건들 중 하나 이상이 참이라면, *candIntraPredModeX*는 *INTRA\_PLANAR*과 동일하게 설정된다.
  - [0255] - 변수 *availableX*는 *FALSE*이다.
    - [0256] - *CuPredMode[ xNbX ][ yNbX ]*는 *MODE\_INTRA*가 아니고 *clip\_flag[ xNbX ][ yNbX ]*는 1이 아니다.
    - [0257] - *pcm\_flag[ xNbX ][ yNbX ]*는 1이다.
    - [0258] - *X*는 B이고 *yCb-1*은 ((*yCb* >> *CtbLog2SizeY*) << *CtbLog2SizeY*)보다 작다.
    - [0259] - 그렇지 않으면 *candIntraPredModeX*는 다음과 같이 도출된다:
      - [0260] - *intra\_lwip\_flag[ xCb ][ yCb ]*가 1이면, *candIntraPredModeX*는 다음과 같은 순서화된 단계들에 의해 도출된다:
        - i. 조항 8.4.X.1에 명시된 블록에 대한 크기 유형 도출 프로세스는, 루마 샘플들에서 현재 코딩 블록 폭 *cbWidth*와 루마 샘플들에서 현재 코딩 블록의 높이 *cbHeight*를 입력으로서 취하여 기동되고, 출력은 변수 *sizeId*에 할당된다.
        - ii. *candIntraPredModeX*는 표 8-X3에 명시된 *IntraPredModeY[ xNbX ][ yNbX ]* 및 *sizeId*를 이용하여 도출된다.
          - [0263] - 그렇지 않으면 *candIntraPredModeX*는 *IntraPredModeY[ xNbX ][ yNbX ]*와 동일하게 설정된다.
      - [0264] 3. 변수들 *ispDefaultMode1* 및 *ispDefaultMode2*는 다음과 같이 정의된다:
        - [0265] - *IntraSubPartitionsSplitType*이 *ISP\_HOR\_SPLIT*이면, *ispDefaultMode1*은 *INTRA\_ANGULAR18*과 동일하게 설정되고 *ispDefaultMode2*는 *INTRA\_ANGULAR5*와 동일하게 설정된다.
          - [0266] - 그렇지 않으면 *ispDefaultMode1*은 *INTRA\_ANGULAR50*로 설정되고 *ispDefaultMode2*는 *INTRA\_ANGULAR63*과 동일하

계 설정된다.

[0267] ...

[0268] 표 8-X3 – 어파인 선형 가중된 인트라 예측 및 인트라 예측 모드들 사이의 맵핑 명세

<i>IntraPredModeY[ xNbX ][ yNbX ]</i>	블록 크기 유형 <i>sizeId</i>		
	0	1	2
0	0	0	1
1	18	1	1
2	18	0	1
3	0	1	1
4	18	0	18
5	0	22	0
6	12	18	1
7	0	18	0
8	18	1	1
9	2	0	50
10	18	1	0
11	12	0	
12	18	1	
13	18	0	
14	1	44	
15	18	0	
16	18	50	
17	0	1	
18	0	0	
19	50		
20	0		
21	50		
22	0		
23	56		
24	0		
25	50		
26	66		
27	50		
28	56		
29	50		
30	50		
31	1		
32	50		
33	50		
34	50		

[0269]

#### 8.4.3 크로마 인트라 예측 모드를 위한 도출 프로세스

o) 프로세스에 대한 입력은 다음과 같다:

- 현재 블록의 상단 좌측 루마 샘플에 관해 현재 크로마 코딩 블록의 상단 좌측 샘플을 명시하는 루마 위치 (*xCb*, *yCb*),

- 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수 *cbWidth*,

- 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수 *cbHeight*.

o) 프로세스에서 크로마 인트라 예측 모드 *IntraPredModeC[ xCb ][ yCb ]*가 도출된다.

대응하는 루마 인트라 예측 모드 *lumaIntraPredMode*는 다음과 같이 도출된다:

- [0277] - *intra\_lwip\_flag[ xCb ][ yCb ]*가 1이면, *lumaIntraPredMode*는 다음과 같은 순서화된 단계들에 의해 도출된다:
- [0278] i. 조항 8.4.X.1에 명시된 블록에 대한 크기 유형 도출 프로세스는, 루마 샘플들에서 현재 코딩 블록 폭 *cbWidth*와 루마 샘플들에서 현재 코딩 블록의 높이 *cbHeight*를 입력으로서 취하여 기동되고, 출력은 변수 *sizeId*에 할당된다.
- [0279] ii. 루마 인트라 예측 모드는 *IntraPredModeY[ xCb + cbWidth/2 ][ yCb + cbHeight/2 ]* 및 표 8-X3에 명시된 *sizeId*를 이용하여 도출되고 *candIntraPredModeX* 값을 *lumaIntraPredMode*에 할당한다.
- [0280] - 그렇지 않으면, *lumaIntraPredMode*는 *IntraPredModeY[ xCb + cbWidth / 2 ][ yCb + cbHeight / 2 ]*로 설정된다.
- [0281] 크로마 인트라 예측 모드 *IntraPredModeC[ xCb ][ yCb ]*는 *intra\_chroma\_pred\_mode[ xCb ][ yCb ]* 및 표 8-2 및 표 8-3에 명시된 *lumaIntraPredMode*를 이용하여 도출된다.
- [0282] ...
- [0283] **xxx. 인트라 샘플 예측**
- [0284] <begin>
- [0285] i) 프로세스에 대한 입력들은 다음과 같다:
- [0286] - 현재 빙처의 상단 좌측 샘플에 관해 현재 변환 블록의 상단 좌측 샘플을 명시하는 샘플 위치 (*xTbCmp*, *yTbCmp*),
- [0287] - 인트라 예측 모드를 명시하는 변수 *predModeIntra*,
- [0288] - 변환 블록 폭을 명시하는 변수 *nTbW*,
- [0289] - 변환 블록 높이를 명시하는 변수 *nTbH*,
- [0290] - 코딩 블록 폭을 명시하는 변수 *nCbW*,
- [0291] - 코딩 블록 높이를 명시하는 변수 *nCbH*,
- [0292] - 현재 블록의 컬러 성분을 명시하는 변수 *cIdx*.
- [0293] ii) 프로세스의 출력들은  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들 *predSamples[ x ][ y ]*이다.
- [0294] 예측된 샘플 *predSamples[ x ][ y ]*는 다음과 같이 도출된다:
- [0295] - *intra\_lwip\_flag[ xTbCmp ][ yTbCmp ]*가 1이고 *cIdx*가 0이면, 조항 8.4.4.2.X1에 명시된 어파인 선형 가중된 인트라 샘플 예측 프로세스가, 위치 (*xTbCmp*, *yTbCmp*), 인트라 예측 모드 *predModeIntra*, 변환 블록 폭 *nTbW* 및 높이 *nTbH*를 입력들로서 취하여 기동되고, 출력은 *predSamples*이다.
- [0296] - 그렇지 않은 경우, 조항 8.4.4.2.X1절에 명시된 일반 인트라 샘플 예측 프로세스가, 위치 (*xTbCmp*, *yTbCmp*), 인트라 예측 모드 *predModeIntra*, 변환 블록 폭 *nTbW* 및 높이 *nTbH*, 코딩 블록 폭 *nCbW* 및 높이 *nCbH*, 변수 *cIdx*를 입력들로서 취하여 기동되고, 출력은 *predSamples*이다.
- [0297] **8.4.4.2.X1 어파인 선형 가중된 인트라 샘플 예측**
- [0298] i) 프로세스에 대한 입력들은 다음과 같다:
- [0299] - 현재 빙처의 상단 좌측 샘플에 관해 현재 변환 블록의 상단 좌측 샘플을 명시하는 샘플 위치 (*xTbCmp*, *yTbCmp*),
- [0300] - 인트라 예측 모드를 명시하는 변수 *predModeIntra*,
- [0301] - 변환 블록 폭을 명시하는 변수 *nTbW*,
- [0302] - 변환 블록 높이를 명시하는 변수 *nTbH*.
- [0303] ii) 프로세스의 출력들은  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들 *predSamples[ x ][ y ]*이다.

[0304] 조항 8.4.X.1에 명시된 블록에 대한 크기 유형 도출 프로세스는, 변환 블록 폭  $nTbW$  및 변환 블록 높이  $nTbH$ 를 입력으로서 취하여 기동되고, 출력은 변수  $sizeId$ 에 할당된다.

[0305] 변수들  $numModes$ ,  $boundarySize$ ,  $predW$ ,  $predH$  및  $predC$ 는 표 8-X4에 명시된  $sizeId$ 를 이용하여 도출된다.

**표 8-X4 - sizeId에 의존하는 모드 수, 경계 샘플 크기 및 예측 크기들의 명세**

$sizeId$	$numModes$	$boundarySize$	$predW$	$predH$	$predC$
0	35	2	4	4	4
1	19	4	4	4	4
2	11	4	$\text{Min}(nTbW, 8)$	$\text{Min}(nTbH, 8)$	8

[0307]

플래그  $isTransposed$ 는 다음과 같이 도출된다:

[0309]  $isTransposed = ( \text{predModeIntra} > ( numModes / 2 ) ) ? 1 : 0(8-X15)$

[0310] 플래그들  $needUpsBdryHor$  및  $needUpsBdryVer$ 는 다음과 같이 도출된다:

[0311]  $needUpsBdryHor = ( nTbW > predW ) ? \text{TRUE} : \text{FALSE}(8-X16)$

[0312]  $needUpsBdryVer = ( nTbH > predH ) ? \text{TRUE} : \text{FALSE}(8-X17)$

[0313] 변수들  $upsBdryW$  및  $upsBdryH$ 는 다음과 같이 도출된다:

[0314]  $upsBdryW = ( nTbH > nTbW ) ? nTbW : predW(8-X18)$

[0315]  $upsBdryH = ( nTbH > nTbW ) ? predH : nTbH(8-X19)$

[0316] 변수들  $IwipW$  및  $IwipH$ 는 다음과 같이 도출된다:

[0317]  $IwipW = ( isTransposed = = 1 ) ? predH : predW(8-X20)$

[0318]  $IwipH = ( isTransposed = = 1 ) ? predW : predH(8-X21)$

[0319]  $x = 0..nTbW-1$ 인 기준 샘플들  $refT[x]$  및  $y = 0..nTbH-1$ 인  $refL[y]$ 의 생성의 경우, 조항 8.4.4.2.X2에 명시된 기준 샘플 도출 프로세스는, 샘플 위치( $xTbCmp$ ,  $yTbCmp$ ), 변환 블록 폭  $nTbW$ , 변환 블록 높이  $nTbH$ 를 입력들로서 취하여 기동되고,  $x = 0..nTbW - 1$ 인 상단 및 좌측 기준 샘플들  $refT[x](x=0..nTbW-1)$  및  $y = 0..nTbH - 1$ 인  $refL[y]$ 를 각각 출력들로서 취한다.

[0320]  $x = 0..2 * boundarySize-1$ 인 경계 샘플들  $p[x]$ 의 생성의 경우, 다음이 적용된다.:

[0321] - 조항 8.4.4.2.X3에 명시된 경계 감축 프로세스는, 상단 기준 샘플들에 대해, 블록 크기  $nTbW$ , 기준 샘플들  $refT$ , 경계 크기  $boundarySize$ , 업샘플링 경계 플래그  $needUpsBdryVer$ , 및 업샘플링 경계 크기  $upsBdryW$ 를 입력으로서 취하여 기동되고,  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들  $redT[x]$ 와  $x = 0..upsBdryW - 1$ 인 업샘플링 경계 샘플들  $upsBdryT[x]$ 을 출력들로서 취한다.

[0322] - 조항 8.4.4.2.X3에 명시된 경계 감축 프로세스는, 좌측 기준 샘플들에 대해, 블록 크기  $nTbH$ , 기준 샘플들  $refL$ , 경계 크기  $boundarySize$ , 업샘플링 경계 플래그  $needUpsBdryHor$ , 및 업샘플링 경계 크기  $upsBdryH$ 를 입력들로서 취하여 기동되고,  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들  $redL[x]$  및  $x = 0..upsBdryH-1$ 인 업샘플링 경계 샘플들  $upsBdryL[x]$ 을 출력들로서 취한다.

[0323] - 감축된 상단 및 좌측 경계 샘플들  $redT$  및  $redL$ 은 다음과 같이 경계 샘플 어레이  $p$ 에 할당된다:

[0324] -  $isTransposed$ 가 1이라면,  $p[x]$ 는  $x = 0..boundarySize-1$ 인  $redL[x]$ 와 동일하게 설정되고,  $p[x + boundarySize]$ 는  $x = 0..boundarySize - 1$ 인  $redT[x]$ 와 동일하게 설정된다.

[0325] - 그렇지 않으면,  $p[x]$ 는  $x = 0..boundarySize-1$ 인  $redT[x]$ 로 설정되고  $p[x + boundarySize]$ 는  $x = 0..boundarySize-1$ 인  $redL[x]$ 와 동일하게 설정된다.

[0326]  $predModeIntra$ 에 따른 인트라 샘플 예측 프로세스의 경우, 다음과 같은 순서화된 단계들이 적용된다:

[0327] 1. 어파인 선형 가중된 샘플들,  $x = 0..IwipW-1$ ,  $y = 0..IwipH-1$ 인  $predLwip[x][y]$ 는 다음과 같이 도출된

다:

- 변수 `modeId`는 다음과 같이 도출된다:

[0329]  $modeId = predModeIntra - (isTransposed == 1) ? (numModes / 2) : 0$  (8-X22)

- 가중치 행렬인,  $x = 0..2 * boundarySize - 1$ ,  $y = 0..predC * predC - 1$ 인  $mWeight[x][y]$ 는 표 8-XX[TBD: 가중치 행렬들 추가]에 명시된 `sizeId` 및 `modeId`를 이용하여 도출된다.

- $y = 0..predC * predC - 1$ 인 편향 벡터 `vBias[y]`는 표 8-XX[TBD: 편향 벡터들 추가]에 명시된 `sizeId` 및 `modeId`를 이용하여 도출된다.

- 변수 `sW`는 표 8-X5에 명시된 `sizeId` 및 `modeId`를 이용하여 도출된다.

- 이파인 선형 가중된 샘플들,  $x = 0..lwipW - 1$ ,  $y = 0..lwipH - 1$ 인 `predLwip[x][y]`는 다음과 같이 도출된다:

[0334]  $oW = 1 \ll (sW - 1)$  (8-X23)

[0335]  $sB = BitDepth_y - 1$  (8-X24)

[0336]  $incW = (predC > lwipW) ? 2 : 1$  (8-X25)

[0337]  $incH = (predC > lwipH) ? 2 : 1$  (8-X26)

[0338]  $predLwip[x][y] = (\sum_{i=0}^{2 * boundarySize - 1} mWeight[i][y * incH * predC + x * incW] * p[i])$

[0339]  $(vBias[y * incH * predC + x * incW] \ll sB) + oW) \gg sW$  (8-X27)

- $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ 인 예측된 샘플들 `predSamples[x][y]`는 다음과 같이 도출된다:

- `isTransposed`가 1일 때,  $x = 0..predW - 1$ ,  $y = 0..predH - 1$ 인 `predLwip[x][y]`는 `predLwip[y][x]`와 동일하게 설정된다.

- `needUpsBdryVer`가 TRUE이거나 `needUpsBdryHor`가 TRUE이면, 조항 8.4.4.2.X4에 명시된 예측 업샘플링 프로세스는, 입력 블록 폭 `predW`, 입력 블록 높이 `predH`, 이파인 선형 가중된 샘플 `predLwip`, 변환 블록 폭 `nTbW`, 변환 블록 높이 `nTbH`, 업샘플링 경계 폭 `upsBdryW`, 업샘플링 경계 높이 `upsBdryH`, 상단 업샘플링 경계 샘플들 `upsBdryT`, 및 좌측 업샘플링 경계 샘플들 `upsBdryL`을 입력들로서 취하여 기동되고, 출력은 예측된 샘플 어레이 `predSamples`이다.

- 그렇지 않으면,  $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ 인 `predSamples[x][y]`는 `predLwip[x][y]`와 동일하게 설정된다.

#### 표 8-X5 - `sizeId` 및 `modeId`에 의존하는 가중치 시프트 `sW`의 명세

<code>sizeId</code>	<code>modeId</code>																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	8	9	8	8	8	8	9	8							
2	8	8	8	8	8	8											

[0345]

#### 8.4.4.2.X2 기준 샘플 도출 프로세스

- 이 프로세스에 대한 입력들은 다음과 같다:

- 현재 꼭처의 상단 좌측 루마 샘플에 관해 현재 변환 블록의 상단 좌측 루마 샘플을 명시하는 샘플 위치 ( $xTbY$ ,  $yTbY$ ),

- 변환 블록 폭을 명시하는 변수 `nTbW`,

- 변환 블록 높이를 명시하는 변수 `nTbH`.

- 이 프로세스의 출력들은 각각  $x = 0..nTbW - 1$ 인 상단 및 좌측 기준 샘플 `refT[x]` 및  $y = 0..nTbH - 1$ 인 `refL[y]`

]이다.

[0352]  $x = 0..nTbW-1$ 인 이웃 샘플들  $refT[x]$  및  $y = 0..nTbH-1$ 인  $refL[y]$ 는 인루프 필터 프로세스 이전에 구성된 샘플들이며 다음과 같이 도출된다:

- 상단 및 좌측 이웃 루마 위치들 ( $xNbT, yNbT$ ) 및 ( $xNbL, yNbL$ )는 다음과 같이 명시된다:

$$(xNbT, yNbT) = (xTbY + x, yTbY - 1) \quad (8-X28)$$

$$(xNbL, yNbL) = (xTbY - 1, yTbY + y) \quad (8-X29)$$

- 조항 6.4.X [Ed. (BB): 이웃 블록 가용성 체킹 프로세스  $tbd$ ]에 명시된 블록에 대한 가용성 도출 프로세스는, ( $xTbY, yTbY$ )와 동일하게 설정된 현재 루마 위치 ( $xCurr, yCurr$ )와 상단 이웃 루마 위치 ( $xNbT, yNbT$ )를 입력들로서 취하여 기동되고, 출력은  $x = 0..nTbW - 1$ 인  $availTop[x]$ 에 할당된다.

- 조항 6.4.X [Ed. (BB): 이웃 블록 가용성 체킹 프로세스  $tbd$ ]에 명시된 블록에 대한 가용성 도출 프로세스는, ( $xTbY, yTbY$ )와 동일하게 설정된 현재 루마 위치 ( $xCurr, yCurr$ )와 좌측 이웃 루마 위치 ( $xNbL, yNbL$ )를 입력들로서 취하여 기동되고, 출력은  $y = 0..nTbH - 1$ 인  $availLeft[y]$ 에 할당된다.

-  $x = 0..nTbW - 1$ 인 상단 기준 샘플들  $refT[x]$ 은 다음과 같이 도출된다:

-  $x = 0..nTbW-1$ 인 모든  $availTop[x]$ 가 TRUE이면, 위치 ( $xNbT, yNbT$ )의 샘플은  $x = 0..nTbW-1$ 인  $refT[x]$ 에 할당된다.

- 그렇지 않고,  $availTop[0]$ 이 FALSE이면,  $x = 0..nTbW-1$ 인 모든  $refT[x]$ 가  $1 \ll (BitDepth_Y - 1)$ 로 설정된다.

- 그렇지 않으면,  $x = 0..nTbW-1$ 인 기준 샘플들  $refT[x]$ 는 다음과 같은 순서화된 단계들에 따라 도출된다:

1. 변수  $lastT$ 는,  $x = 1..nTbW-1$ 인 시퀀스  $availTop[x]$  내의 첫 번째 요소의 위치  $x$ 로 설정된다.

2. 모든  $x = 0..lastT - 1$ 에 대해, 위치 ( $xNbT, yNbT$ )의 샘플은  $refT[x]$ 에 할당된다.

3. 모든  $x = lastT..nTbW - 1$ 에 대해,  $refT[x]$ 는  $refT[lastT - 1]$ 로 설정된다.

-  $x = 0..nTbH-1$ 인 좌측 기준 샘플들  $refL[y]$ 는 다음과 같이 도출된다:

-  $y = 0..nTbH-1$ 인 모든  $availLeft[y]$ 가 TRUE이면, 위치 ( $xNbL, yNbL$ )의 샘플은  $y = 0..nTbH-1$ 인  $refL[y]$ 에 할당된다.

- 그렇지 않고,  $availLeft[0]$ 이 FALSE이면,  $y = 0..nTbH - 1$ 인 모든  $refL[y]$ 는  $1 \ll (BitDepth_Y - 1)$ 로 설정된다.

- 그렇지 않으면,  $y = 0..nTbH - 1$ 인 기준 샘플들  $refL[y]$ 는 다음과 같은 순서화된 단계들에 따라 도출된다:

1. 변수  $lastL$ 은 FALSE와 동일한  $y = 1..nTbH - 1$ 인 시퀀스  $availLeft[y]$ 에서 첫 번째 요소의 위치  $y$ 로 설정된다.

2. 모든  $y = 0..lastL-1$ 에 대해, 위치 ( $xNbL, yNbL$ )의 샘플은  $refL[y]$ 에 할당된다.

3. 모든  $y = lastL..nTbH-1$ 에 대해,  $refL[y]$ 는  $refL[lastL-1]$ 로 설정된다.

## 경계 감축 프로세스의 명세

i) 프로세스에 대한 입력들은 다음과 같다:

- 변환 블록 크기를 명시하는 변수  $nTbX$ ,

-  $x = 0..nTbX-1$ 인 기준 샘플들  $refX[x]$ ,

- 다운샘플링된 경계 크기를 명시하는 변수  $boundarySize$ ,

- 업샘플링에 대해 중간 경계 샘플들이 요구되는지를 명시하는 플래그  $needUpsBdryX$ ,

- 업샘플링에 대한 경계 크기를 명시하는 변수  $upsBdrySize$ .

- [0379] ① 프로세스의 출력들은  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들  $redX[x]$  및  $x = 0..upsBdrySize-1$ 인 업샘플링 경계 샘플들  $upsBdryX[x]$ 이다.
- [0380]  $x = 0..upsBdrySize-1$ 인 업샘플링 경계 샘플들  $upsBdryX[x]$ 는 다음과 같이 도출된다:
- $needUpsBdryX \neq \text{TRUE}$ 이고  $upsBdrySize \neq nTbX$ 보다 작다면, 다음이 적용된다:
- [0382]  $uDwn = nTbX / upsBdrySize(8-X30)$
- [0383]  $upsBdryX[x] = (\sum_{i=0}^{uDwn-1} refX[x * uDwn + i] + (1 << (\log_2(uDwn) - 1))) >> \log_2(uDwn)$  (8-X31)
- [0384] - 그렇지 않으면 ( $upsBdrySize \geq nTbX$ 와 같음),  $upsBdryX[x]$ 는  $refX[x]$ 로 설정된다.
- [0385]  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들  $redX[x]$ 는 다음과 같이 도출된다:
- $boundarySize \geq upsBdrySize$ 보다 작다면, 다음이 적용된다:
- [0387]  $bDwn = upsBdrySize / boundarySize(8-X32)$
- [0388]  $refX[x] = (\sum_{i=0}^{bDwn-1} upsBdryX[x * bDwn + i] + (1 << (\log_2(bDwn) - 1))) >> \log_2(bDwn)$  (8-X33)
- [0389] - 그렇지 않으면 ( $boundarySize \geq upsBdrySize$ 와 같음),  $refX[x]$ 는  $upsBdryX[x]$ 로 설정된다.
- [0390] **8.4.4.2.X4 예측 업샘플링 프로세스의 명세**
- [0391] ① 프로세스에 대한 입력들은 다음과 같다:
- 입력 블록 폭을 명시하는 변수  $predW$ ,
  - 입력 블록 높이를 명시하는 변수  $predH$ ,
  - $x = 0..predW-1$ ,  $y = 0..predH-1$ 인 어파인 선형 가중된 샘플들  $predLwip[x][y]$ ,
  - 변환 블록 폭을 명시하는 변수  $nTbW$ ,
  - 변환 블록 높이를 명시하는 변수  $nTbH$ ,
  - 업샘플링 경계 폭을 명시하는 변수  $upsBdryW$ ,
  - 업샘플링 경계 높이를 명시하는 변수  $upsBdryH$ ,
  - $x = 0..upsBdryW-1$ 인 상단 업샘플링 경계 샘플들  $upsBdryT[x]$ ,
  - $x = 0..upsBdryH-1$ 인 좌측 업샘플링 경계 샘플들  $upsBdryL[x]$ .
- [0401] ② 프로세스의 출력들은  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들  $predSamples[x][y]$ 이다.
- [0402] 저밀도 예측된 샘플들  $predSamples[m][n]$ 은 다음과 같이  $x = 0..predW-1$ ,  $y = 0..predH-1$ 인  $predLwip[x][y]$ 로부터 도출된다:
- [0403]  $upHor = nTbW / predW(8-X34)$
- [0404]  $upVer = nTbH / predH(8-X35)$
- [0405]  $predSamples[(x+1)*upHor-1][(y+1)*upVer-1] = predLwip[x][y]$  (8-X36)
- [0406]  $x = 0..upsBdryW-1$ 인 상단 경계 샘플들  $upsBdryT[x]$ 는 다음과 같아  $predSamples[m][-1]$ 에 할당된다:
- [0407]  $predSamples[(x+1)*(nTbW/upsBdryW)-1][-1] = upsBdryT[x]$  (8-X37)
- [0408]  $y = 0..upsBdryH-1$ 인 좌측 경계 샘플들  $upsBdryL[y]$ 는 다음과 같아  $predSamples[-1][n]$ 에 할당된다:
- [0409]  $predSamples[-1][(y+1)*(nTbH/upsBdryH)-1] = upsBdryL[y]$  (8-X38)

- [0410]  $x = 0..nTbW-1, y = 0..nTbH-1$ 인 예측된 샘플들  $predSamples[x][y]$ 는 다음과 같이 도출된다:
- $nTbH \neq nTbW$ 보다 크다면, 다음과 같은 순서화된 단계들이 적용된다:
- $upHor > 1$ 보다 크면,  $m = 0..predW-1, n = 1..predH-1$  모든 저밀도 위치들 ( $xHor, yHor$ ) = ( $m * upHor - 1, n * upVer - 1$ )에 대한 수평 업샘플링이 다음과 같이  $dX = 1..upHor-1$ 과 함께 적용된다:
- $$predSamples[xHor + dX][yHor] = ((upHor - dX) * predSamples[xHor][yHor]) \\ dX * predSamples[xHor + upHor][yHor] / upHor$$
- [0413] (8-X39)
- $m = 0..nTbW-1, n = 0..predH-1$  모든 저밀도 위치들 ( $xVer, yVer$ ) = ( $m, n * upVer-1$ )에 대한 수직 업샘플링은 다음과 같이  $dY = 1..upVer-1$ 과 함께 적용된다:
- $$predSamples[xVer][yVer + dY] = ((upVer - dY) * predSamples[xVer][yVer]) + \\ dY * predSamples[xVer][yVer + upVer] / upVer$$
- [0415] (8-X40)
- 그렇지 않으면, 다음과 같은 순서화된 단계들가 적용된다:
- $upVer > 1$ 보다 크면,  $m = 1..predW, n = 0..predH-1$  모든 저밀도 위치들 ( $xVer, yVer$ ) = ( $m * upHor - 1, n * upVer - 1$ )에 대한 수직 업샘플링이 (8-X40)에 명시된 대로  $dY = 1..upVer-1$ 과 함께 적용된다.
  - $m = 0..predW-1, n = 0..nTbH-1$  모든 저밀도 위치들 ( $xHor, yHor$ ) = ( $m * upHor-1, n$ )에 대한 수평 업샘플링은 (8-X39)에 명시된 대로  $dX = 1..upHor-1$ 과 함께 적용된다.
- [0419] <end>
- [0420] 표 9-9 - 선택스 요소들 및 연관된 2진화들
- | 선택스 구조               | 선택스 요소                                | 2진화       |   |
|----------------------|---------------------------------------|-----------|---|
|                      |                                       | 프로세스      | 입력 파라미터들  |
| <i>coding_unit()</i> | <i>cu_skip_flag[]</i>                 | <i>FL</i> | <i>cMax = 1</i>   |
|                      | <i>pred_mode_ibc_flag</i>             | <i>FL</i> | <i>cMax = 1</i>   |
|                      | <i>pred_mode_flag</i>                 | <i>FL</i> | <i>cMax = 1</i>   |
|                      | <i>&lt;begin&gt;intra_lwip_flag[]</i> | <i>FL</i> | <i>cMax = 1</i>   |
|                      | <i>intra_lwip_mpm_flag[]</i>          | <i>FL</i> | <i>cMax = 1</i>   |
|                      | <i>intra_lwip_mpm_idx[]</i>           | <i>TR</i> | <i>cMax = 2, cRiceParam = 0</i>   |
|                      | <i>intra_lwip_mpm_remainder[]</i>     | <i>FL</i> | <i>cMax = (cbWidth == 4 &amp;&amp; cbHeight == 4) ? 31 : ((cbWidth &lt;= 8 &amp;&amp; cbHeight &lt;= 8) ? 15 : 7)</i> |
|                      | ...                                   |           |   |

[0421]

[0422] 표 9-15 - 컨텍스트 코딩된 빈들을 선택스 요소들로의  $ctxInc$ 의 할당

선택스 요소	$binIdx$					
	0	1	2	3	4	$\geq 5$
...	terminate	na	na	na	na	na
$intra\_lwip\_flag[J][J]$	$(Abs(Log2(cbWidth) - Log2(cbHeight)) > 1) ? 3 : \{0,1,2\}$ (clause 9.5.4.2.2)	na	na	na	na	na
$intra\_lwip\_mpm\_flag[J][J]$	0	na	na	na	na	na
$intra\_lwip\_mpm\_idx[J][J]$	bypass	bypass	na	na	na	na
$intra\_lwip\_mpm\_remainde[r][J][J]$	bypass	bypass	bypass	bypass	bypass	na

[0423]

[0424] 표 9-16 - 좌측 및 위쪽 선택스 요소들을 이용한  $ctxInc$ 의 명세

선택스 요소	$condL$	$condA$	$ctxSetIdx$
...			
$intra\_lwip\_flag[x0][y0]$	$intra\_lwip\_flag[xNbL][yNbL]$	$intra\_lwip\_flag[xNbA][yNbA]$	0
...			

[0425]

[0426] <end>

[0427] ALWIP의 요약

[0428] 폭  $W$  및 높이  $H$ 의 직사각형 블록의 샘플을 예측하기 위해, 어파인 선형 가중된 인트라 예측(ALWIP)은 블록 좌측의 H개의 복원된 이웃 경계 샘플들의 한 라인과 블록 위의  $W$  복원된 이웃 경계 샘플들의 한 라인을 입력으로서 취한다. 복원된 샘플들이 이용가능하지 않다면, 이들은 종래의 인트라 예측에서와 같이 생성된다. ALWIP는 루마 인트라 블록에만 적용된다. 크로마 인트라 블록의 경우, 종래의 인트라 코딩 모드들이 적용된다.

[0429] 예측 신호의 생성은 다음과 같은 3개 단계에 기초한다:

[0430] 1. 경계 샘플들로부터,  $W=H=4$ 인 경우 4개의 샘플, 그 외의 경우 8개의 샘플이 평균화함으로써 추출된다.

[0431] 2. 오프셋의 추가가 뒤따르는 행렬 벡터 곱셈은 평균된 샘플들을 입력으로서 이용하여 실행된다. 결과는 원래 블록의 서브샘플링된 샘플 세트에 관한 감축된 예측 신호이다.

[0432] 3. 나머지 위치들에서의 예측 신호는, 각각의 방향에서의 단일 단계 선형 보간인 선형 보간에 의해 서브샘플링된 세트에 관한 예측 신호로부터 생성된다.

[0433] ALWIP 모드가 적용되는 경우, ALWIP 모드의 인덱스  $predmode$ 는 3 MPMS를 갖는 MPM-목록을 이용하여 시그널링된다. 여기서, MPM들의 도출은 위쪽과 좌측 PU의 인트라 모드들을 이용하여 다음과 같이 수행된다. 각각의 종래의 인트라 예측 모드  $predmode_{Angular}$ 에 ALWIP 모드를 할당하는 3개의 고정 테이블  $map\_angular\_to\_alwip_idx$ ,  $idx \in \{0,1,2\}$  이 있다.

[0434]  $\text{predmode}_{\text{ALWIP}} = \text{map\_angular\_to\_alwip}_{idx}[\text{predmode}_{\text{Angular}}]$ .

[0435] 폭  $W$  및 높이  $H$ 의 각각의 PU에 대해, 인덱스를 정의한다

[0436]  $idx(PU) = idx(W, H) \in \{0, 1, 2\}$

[0437] 이 인덱스는 ALWIP 파라미터들이 3개 세트 중 어느 것으로부터 선택되는지를 나타낸다.

[0438] 위쪽 예측 유닛  $PU_{above}$  이 이용가능하고 현재 PU와 동일한 CTU에 속하고 인트라 모드에 있는 경우,  $idx(PU) = idx(PU_{above})$  이고 ALWIP가 ALWIP-모드  $\text{predmode}_{\text{ALWIP}}^{above}$  와 함께  $PU_{above}$ 에 적용되면, 다음과 같이 둘 수 있다

[0439]  $\text{mode}_{\text{ALWIP}}^{above} = \text{predmode}_{\text{ALWIP}}^{above}$ .

[0440] 위쪽 PU가 이용가능하고 현재 PU와 동일한 CTU에 속하며, 인트라 모드에 있고 종래의 인트라 예측된 모드  $\text{predmode}_{\text{Angular}}^{above}$  가 위쪽 PU에 적용되는 경우, 다음과 같이 둘 수 있다

[0441]  $\text{mode}_{\text{ALWIP}}^{above} = \text{map\_angular\_to\_alwip}_{idx(=PU_{above})}[\text{predmode}_{\text{Angular}}^{above}]$ .

[0442] 다른 모든 경우에는, 다음과 같이 둘 수 있다

[0443]  $\text{mode}_{\text{ALWIP}}^{above} = -1$

[0444] 이것은 이 모드가 이용가능하지 않음을 의미한다. 동일한 방식으로 그러나 좌측 PU가 현재 PU와 동일한 CTU에 속할 필요가 있다는 제한없이, 모드  $\text{mode}_{\text{ALWIP}}^{left}$  를 도출한다.

[0445] 마지막으로, 3개의 고정된 디폴트 목록  $list_{idx}, idx \in \{0, 1, 2\}$  이 제공되며, 각각의 목록은 3개의 구별되는 ALWIP 모드를 포함한다. 디폴트 목록  $list_{idx(PU)}$  과 모드를  $\text{mode}_{\text{ALWIP}}^{above}$  및  $\text{mode}_{\text{ALWIP}}^{left}$ 로부터, -1을 디폴트 값들로 대체하고 반복을 제거함으로써 3개의 구별되는 MPM을 구성한다.

[0446] 루마 MPM 목록 도출의 경우, ALWIP 모드  $\text{predmode}_{\text{ALWIP}}$  를 이용하는 이웃 루마 블록과 조우할 때마다, 이 블록은 종래의 인트라 예측 모드  $\text{predmode}_{\text{Angular}}$  를 이용하는 것처럼 취급된다.

[0447]  $\text{predmode}_{\text{Angular}} = \text{map\_alwip\_to\_angular}_{idx(PU)}[\text{predmode}_{\text{ALWIP}}]$

### [0448] 3 VVC에서의 변환

#### [0449] 3.1 복수의 변환 선택 (MTS)

[0450] HEVC에서 이용된 DCT-II에 추가하여, MTS(Multiple Transform Selection) 방식은 인터 및 인트라 코딩된 블록들 양쪽 모두의 잔차 코딩에 이용된다. 이것은 DCT8/DST7로부터 선택된 복수의 변환을 이용한다. 새로 도입된 변환 행렬들은 DST-VII 및 DCT-VIII이다.

#### [0451] 3.2 JVET-N0193에서 제안된 감축된 2차 변환(RST)

[0452] 감축된 2차 변환(reduced secondary transform)(RST)은, 4x4 및 8x8 블록들에 대해 각각 16x16 및 16x64 비분리 변환을 적용한다. 1차 순방향 및 역변환들은 여전히 2개의 1-D 수평/수직 변환 패스들과 동일한 방식으로 수행된다. 2차 순방향 및 역변환들은 1차 변환들과는 별개의 프로세스 단계이다. 인코더의 경우, 1차 순방향 변환이 먼저 수행된 다음, 2차 순방향 변환 및 양자화, CABAC 비트 인코딩이 뒤따른다. 디코더, CABAC 비트 디

코딩 및 역양자화의 경우, 2차 역변환이 먼저 수행된 다음, 1차 역변환이 뒤따른다. RST는 인트라 슬라이스와 인터 슬라이스들 양쪽 모두에서 인트라 코딩된 TU들에만 적용된다.

### [0453] 3.3 JVET-N0185의 인트라 모드 코딩을 위한 단일화된 MPM 목록

MRL(Multiple Reference Line) 및 ISP(Intra sub-partition) 코딩 툴들이 적용되는지의 여부와 관계없이 인트라 블록들에 대해 단일화된 6-MPM 목록이 제안된다. MPM 목록은 VTM4.0에서와 같이 좌측과 위쪽 이웃 블록들의 인트라 모드들에 기초하여 구성된다. 좌측의 모드를 Left, 위쪽 블록의 모드를 Above라고 가정하면, 단일화된 MPM 목록은 다음과 같이 구성된다:

● 이웃 블록이 이용가능하지 않으면, 그 인트라 모드는 디폴트로 Planar로 설정된다.

● 양쪽 모드를 Left 및 Above가 비-각도 모드들인 경우:

a. MPM 목록 → {Planar, DC, V, H, V-4, V+4}

● Left 및 Above 모드들 중 하나가 각도 모드이고, 다른 하나가 비각도형인 경우:

a. 모드 Max를 Left 및 Above 중 더 큰 모드로서 설정

b. MPM 목록 → {Planar, Max, DC, Max -1, Max +1, Max -2}

● Left 및 Above 양쪽 모두가 각도형인 경우:

a. 모드 Max를 Left 및 Above 중 더 큰 모드로서 설정

b. Left와 Above 모드의 차이가 2 내지 62(양쪽 끝값 포함) 범위에 있는 경우

i. MPM 목록 → {Planar, Left, Above, DC, Max -1, Max +1}

c. 그렇지 않으면

i. MPM 목록 → {Planar, Left, Above, DC, Max -2, Max +2}

● Left 및 Above 양쪽 모두가 각도형인 경우:

a. MPM 목록 → {Planar, Left, Left -1, Left +1, DC, Left -2}

게다가, MPM 인덱스 코드워드의 첫 번째 빈(bin)은 CABAC 컨텍스트 코딩된다. 현재 인트라 블록이 MRL 인에이블, ISP 인에이블, 또는 일반 인트라 블록인지에 대응하여, 총 3개의 컨텍스트가 이용된다.

단일화된 MPM 목록 구성에서 이용되는 좌측 이웃 블록 및 위쪽 이웃 블록은 도 10에 도시된 바와 같이 A2 및 B2이다.

하나의 MPM 플래그가 먼저 코딩된다. 블록이 MPM 목록 내의 모드 중 하나로 코딩되면, MPM 인덱스가 추가로 코딩된다. 그렇지 않으면, (MPM들을 제외한) 나머지 모드들에 대한 인덱스가 코딩된다.

### [0472] 4 기준 구현들의 단점들의 예들

JVET-N0217에서의 ALWIP의 설계는 다음과 같은 문제점을 갖는다:

1) 2019년 3월 JVET 회의에서, MRL 모드, ISP 모드 및 일반 인트라 모드에 대해 단일화된 6-MPM 목록 생성이 채택되었다. 그러나, 어파인 선형 가중된 예측 모드는 MPM 목록 구성을 복잡하게 만드는 상이한 3-MPM 목록 구성 을 이용한다. 복잡한 MPM 목록 구성은, 특히 4x4 샘플 등의 작은 블록들의 경우, 디코더의 처리량을 손상시킬 수 있다.

2) ALWIP는 블록의 루마 성분에만 적용된다. ALWP 코딩된 블록의 크로마 성분의 경우, 크로마 모드 인덱스가 코딩되어 디코더로 전송되므로, 불필요한 시그널링이 초래할 수 있다.

3) 다른 코딩 툴들과 ALWIP의 상호작용들이 고려되어야 한다.

4)  $\text{upsBdryX}[x] = (\sum_{i=0}^{uDwn-1} \text{refX}[x * uDwn + i] + (1 \ll (Log2(uDwn) - 1))) \gg Log2(uDwn)$  (8-X31)에서

upsBdryX를 계산할 때,  $\text{Log2}(uDwn) - 1$ 이 -1인 반면, -1과의 좌측 시프트는 미정의되는 것이 가능하다.

5) 예측 샘플들을 업샘플링할 때, 어떠한 반올림도 적용되지 않는다.

[0479] 6) 디블록킹 프로세스에서, ALWIP 코딩된 블록들은 일반 인트라 블록들로서 취급된다.

[0480] 7) ALWIP 플래그 (예를 들어, intra\_lwip\_flag)를 코딩하는데 있어서 너무 많은 컨텍스트(예를 들어, 4)가 이용된다.

[0481] 8) 수직 업샘플링과 수평 업샘플링이 양쪽 모두가 요구될 때, 업샘플링 순서는 블록 형상에 의존한다. 이것은 하드웨어 친화적이지 않다.

[0482] 9) 업샘플링에 대해 선형 보간 필터가 이용되므로, 비효율적일 수 있다.

## 5 행렬 기반의 인트라 코딩을 위한 예시적인 방법들

[0484] 현재 개시된 기술의 실시예들은 기존 구현들의 단점들을 극복함으로써, 더 높은 코딩 효율을 갖지만 더 낮은 계산 복잡성을 갖는 비디오 코딩을 제공한다. 본 문서에서 설명된 바와 같이, 비디오 코딩을 위한 행렬 기반의 인트라 예측된 방법들은, 기존의 및 미래의 비디오 코딩 표준들 양쪽 모두를 향상시킬 수 있고, 다양한 구현들에 대해 설명된 다음과 같은 예들에서 명료화된다. 아래에 제공된 개시된 기술의 예들은 일반적인 개념을 설명하는 것이고, 제한하는 것으로서 해석될 것을 의미하지 않는다. 한 예에서, 반대로 명시적으로 나타내지 않는 한, 이들 예에서 설명된 다양한 퍼처들은 결합될 수 있다.

[0485] 이하의 논의에서, 인트라 예측 모드란, (DC, 평면, CCLM 및 기타 가능한 인트라 예측 모드를 포함한) 각도 인트라 예측 모드를 지칭한다; 반면, 인트라 모드란 일반 인트라 모드 또는 MRL, ISP 또는 ALWIP를 지칭한다.

[0486] 이하의 논의에서, "다른 인트라 모드들"이란, ALWIP를 제외한, 일반 인트라 모드, MRL 또는 ISP 등의, 하나 또는 복수의 인트라 모드를 지칭할 수 있다.

[0487] 이하의 논의에서, SatShift(x, n)는 다음과 같이 정의된다

$$SatShift(x, n) = \begin{cases} (x + offset0) \gg n & \text{if } x \geq 0 \\ -((-x + offset1) \gg n) & \text{if } x < 0 \end{cases}$$

[0488] Shift(x, n)는 Shift(x, n) = (x + offset0) >> n으로서 정의된다.

[0489] 한 예에서, offset0 및/또는 offset1은 (1<<n)>>1 또는 (1<<(n-1))로 설정된다. 또 다른 예에서, offset0 및/또는 offset1은 0으로 설정된다.

[0490] 또 다른 예에서, offset0=offset1= ((1<<n)>>1)-1 또는 ((1<<(n-1)))-1이다.

[0491] Clip3(min, max, x)는 다음과 같이 정의된다

$$Clip3(Min, Max, x) = \begin{cases} Min & \text{if } x < Min \\ Max & \text{if } x > Max \\ x & \text{Otherwise} \end{cases}$$

## ALWIP에 대한 MPM 목록 구성

[0492] 1. ALWIP에 대한 MPM 목록의 전체 또는 일부는, (일반 인트라 모드, MRL 또는 ISP 등의) 비-ALWIP 인트라 모드에 대한 MPM 목록을 구성하는 전체 또는 부분적 절차에 따라 구성될 수 있다는 것이 제안된다.

[0493] a. 한 예에서, ALWIP에 대한 MPM 목록의 크기는, 비-ALWIP 인트라 모드에 대한 MPM 목록의 크기와 동일할 수 있다.

[0494] i. 예를 들어, MPM 목록의 크기는, ALWIP 및 비-ALWIP 인트라 모드들 양쪽 모두에 대해 6이다.

[0495] b. 한 예에서, ALWIP에 대한 MPM 목록은 비-ALWIP 인트라 모드에 대한 MPM 목록으로부터 도출될 수 있다.

[0496] i. 한 예에서, 비-ALWIP 인트라 모드에 대한 MPM 목록이 먼저 구성될 수 있다. 그 후, 이들의 일부 또는 전부는, ALWIP 코딩된 블록들에 대한 MPM 목록에 추가로 더해질 수 있는 MPM들로 변환될 수 있다.

[0497] 1) 대안으로서, 또한, 변환된 MPM을 ALWIP 코딩된 블록들에 대한 MPM 목록에 추가할 때, 프루닝(pruning)이 적용될 수 있다.

- [0501] 2) 디폴트 모드들은 ALWIP 코딩된 블록들에 대한 MPM 목록에 추가될 수 있다.
- [0502] a. 한 예에서, 디폴트 모드들은 비-ALWIP 인트라 모드의 MPM 목록으로부터 변환된 것들 이전에 추가될 수 있다.
- [0503] b. 대안으로서, 비-ALWIP 인트라 모드의 MPM 목록으로부터 변환된 것들 이후에 디폴트 모드들이 추가될 수 있다.
- [0504] c. 대안으로서, 비-ALWIP 인트라 모드의 MPM 목록으로부터 변환된 것들과 인터리빙된 방식으로 디폴트 모드들이 추가될 수 있다.
- [0505] d. 한 예에서, 디폴트 모드들은 모든 종류의 블록들에 대해 동일하도록 고정될 수 있다.
- [0506] e. 대안으로서, 디폴트 모드들은, 이웃 블록들의 가용성, 이웃 블록들의 모드 정보, 블록 치수 등의, 코딩된 정보에 따라 결정될 수 있다.
- [0507] i.i. 한 예에서, 비-ALWIP 인트라 모드에 대한 MPM 목록 내의 하나의 인트라 예측 모드는, ALWIP에 대한 MPM 목록 내에 놓여질 때, 그 대응하는 ALWIP 인트라 예측 모드로 변환될 수 있다.
- [0508] 1) 대안으로서, 비-ALWIP 인트라 모드들에 대한 MPM 목록 내의 모든 인트라 예측 모드들은, ALWIP에 대한 MPM 목록을 구성하는데 이용되기 이전에 대응하는 ALWIP 인트라 예측 모드들로 변환될 수 있다.
- [0509] 2) 대안으로서, (이웃 블록들로부터의 인트라 예측 모드들 및 Planar 및 DC 등의 디폴트 인트라 예측 모드들을 포함할 수 있는) 모든 후보 인트라 예측 모드들은, 비-ALWIP 인트라 모드들에 대한 MPM 목록이 ALWIP에 대한 MPM 목록을 도출하는데 추가로 이용될 수 있는 경우, 비-ALWIP 인트라 모드들에 대한 MPM 목록을 구성하는데 이용되기 전에, 대응하는 ALWIP 인트라 예측 모드들로 변환될 수 있다.
- [0510] 3) 한 예에서, 2개의 변환된 ALWIP 인트라 예측 모드가 비교될 수 있다.
- [0511] a. 한 예에서, 이들이 동일하다면, 이들 중 하나만이 ALWIP에 대한 MPM 목록 내에 둘 수 있다.
- [0512] b. 한 예에서, 이들이 동일하다면, 이들 중 하나만이 비-ALWIP 인트라 모드들에 대한 MPM 목록 내에 둘 수 있다.
- [0513] i.iii. 한 예에서, 비-ALWIP 인트라 모드에 대한 MPM 목록 내의 S개의 인트라 예측 모드들 중 K개는 ALWIP 모드에 대한 MPM 목록으로서 선택될 수 있다. 예를 들어, K는 3이고 S는 6이다.
- [0514] 1) 한 예에서, 비-ALWIP 인트라 모드들에 대한 MPM 목록 내의 처음 K개의 인트라 예측 모드가 ALWIP 모드에 대한 MPM 목록으로서 선택될 수 있다.
- [0515] 2. ALWIP에 대한 MPM 목록을 도출하는데 이용되는 하나 또는 복수의 이웃 블록은, (일반 인트라 모드, MRL 또는 ISP 등의) 비-ALWIP 인트라 모드들에 대한 MPM 목록을 도출하는데에도 이용될 수 있다는 것이 제안된다.
- [0516] a. 한 예에서, ALWIP에 대한 MPM 목록을 도출하는데 이용되는 현재 블록 좌측의 이웃 블록은 비-ALWIP 인트라 모드들에 대한 MPM 목록을 도출하는데 이용되는 것과 동일해야 한다.
- [0517] i. 현재 블록의 상단 좌측 코너가 ( $x_{Cb}$ ,  $y_{Cb}$ )이고 현재 블록의 폭과 높이가 W 및 H라고 가정하면, 한 예에서, ALWIP 및 비-ALWIP 인트라 모드들 양쪽 모두에 대한 MPM 목록을 도출하는데 이용되는 좌측 이웃 블록은 위치 ( $x_{Cb}-1$ ,  $y_{Cb}$ )를 커버할 수 있다. 한 대안적인 예에서, ALWIP 및 비-ALWIP 인트라 모드들 양쪽 모두에 대한 MPM 목록을 도출하는데 이용되는 좌측 이웃 블록은 위치 ( $x_{Cb}-1$ ,  $y_{Cb}+H-1$ )를 커버할 수 있다.
- [0518] ii. 예를 들어, 단일화된 MPM 목록 구성에 이용되는 좌측 이웃 블록과 위쪽 이웃 블록은 도 10에 도시된 바와 같이 A2 및 B2이다.
- [0519] b. 한 예에서, ALWIP에 대한 MPM 목록을 도출하는데 이용되는 현재 블록 위쪽의 이웃 블록은 비-ALWIP 인트라 모드들에 대한 MPM 목록을 도출하는데 이용되는 것과 동일해야 한다.
- [0520] i. 현재 블록의 상단 좌측 코너가 ( $x_{Cb}$ ,  $y_{Cb}$ )이고 현재 블록의 폭과 높이가 W 및 H라고 가정하면, 한 예에서, ALWIP 및 비-ALWIP 인트라 모드들 양쪽 모두에 대한 MPM 목록을 도출하는데 이용되는 위쪽 이웃 블록은 위치 ( $x_{Cb}$ ,  $y_{Cb}-1$ )를 커버할 수 있다. 한 대안적인 예에서, ALWIP 및 비-ALWIP 인트라 모드들 양쪽 모두에 대한 MPM 목록을 도출하는데 이용되는 위쪽 이웃 블록은 위치 ( $x_{Cb}+W-1$ ,  $y_{Cb}-1$ )를 커버할 수 있다.

- [0521] i.i. 예를 들어, 단일화된 MPM 목록 구성에 이용되는 좌측 이웃 블록과 위쪽 이웃 블록은 도 10에 도시된 바와 같이 A1 및 B1이다.
- [0522] 3. ALWIP에 대한 MPM 목록이 현재 블록의 폭 및/또는 높이에 따라 상이한 방식들로 구성될 수 있다는 것이 제안된다.
- [0523] a. 한 예에서, 상이한 블록 치수들에 대해 상이한 이웃 블록들이 액세스될 수 있다.
- [0524] 4. ALWIP에 대한 MPM 목록과 비-ALWIP 인트라 모드들에 대한 MPM 목록은 동일한 절차로 그러나 상이한 파라미터들로 구성될 수 있다는 것이 제안된다.
- [0525] a. 한 예에서, 비-ALWIP 인트라 모드들의 MPM 목록 구성 절차에서 S개의 인트라 예측 모드 중 K개가 ALWIP 모드에서 이용되는 MPM 목록에 대해 도출될 수 있다. 예를 들어, K는 3이고 S는 6이다.
- [0526] i. 한 예에서, MPM 목록 구성 절차에서 처음 K개의 인트라 예측 모드는 ALWIP 모드에서 이용되는 MPM 목록에 대해 도출될 수 있다.
- [0527] b. 한 예에서, MPM 목록 내의 제1 모드는 상이할 수 있다.
- [0528] i. 예를 들어, 비-ALWIP 인트라 모드들에 대한 MPM 목록 내의 첫 번째 모드는 Planar일 수 있지만, ALWIP에 대해서는 MPM 목록 내의 모드 X0일 수 있다.
- [0529] 1) 한 예에서, X0은 Planar로부터 변환된 ALWIP 인트라 예측 모드일 수 있다.
- [0530] c. 한 예에서, MPM 목록 내의 스터핑 모드(stuffing mode)들은 상이할 수 있다.
- [0531] i. 예를 들어, 비-ALWIP 인트라 모드들에 대한 MPM 목록의 처음 3개의 스터핑 모드는, DC, Vertical 및 Horizontal일 수 있지만, ALWIP에 대해서는 MPM 목록의 모드 내의 X1, X2, X3일 수 있다.
- [0532] 1) 한 예에서, X1, X2, X3은 상이한 sizeId에 대해 상이할 수 있다.
- [0533] ii. 한 예에서, 스터핑 모드의 수는 상이할 수 있다.
- [0534] d. 한 예에서, MPM 목록 내의 이웃 모드들은 상이할 수 있다.
- [0535] i. 예를 들어, 이웃 블록들의 일반 인트라 예측 모드들은 비-ALWIP 인트라 모드들에 대한 MPM 목록을 구성하는데 이용된다. 그리고 이들은 ALWIP 모드에 대한 MPM 목록을 구성하기 위해 ALWIP 인트라 예측 모드들로 변환된다.
- [0536] e. 한 예에서, MPM 목록 내의 시프트된 모드들은 상이할 수 있다.
- [0537] i. 예를 들어, X가 일반 인트라 예측 모드고 K0이 정수인 X+K0이 비-ALWIP 인트라 모드에 대한 MPM 목록에 포함될 수 있다. 그리고 Y가 ALWIP 인트라 예측 모드이고 K1이 정수인 Y+K1이 ALWIP에 대한 MPM 목록에 포함될 수 있고, 여기서 K0은 K1과는 상이할 수 있다.
- [0538] 1) 한 예에서, K1은 폭과 높이에 의존할 수 있다.
- [0539] 5. 비-ALWIP 인트라 모드들로 현재 블록에 대한 MPM 목록을 구성할 때 이웃 블록은 ALWIP로 코딩된 경우 이용불가능한 것으로 취급되는 것이 제안된다.
- [0540] a. 대안으로서, 비-ALWIP 인트라 모드들로 현재 블록에 대한 MPM 목록을 구성할 때 이웃 블록은 ALWIP로 코딩된 경우 (Planar 등의) 미리정의된 인트라 예측 모드로 코딩된 것으로 취급된다.
- [0541] 6. ALWIP 모드로 현재 블록에 대한 MPM 목록을 구성할 때 이웃 블록은 비-ALWIP 인트라 모드로 코딩된 경우 이용불가능한 것으로 취급되는 것이 제안한다.
- [0542] a. 대안으로서, ALWIP 모드로 현재 블록에 대한 MPM 목록을 구성할 때 이웃 블록은 비-ALWIP 인트라 모드들로 코딩된 경우 미리정의된 ALWIP 인트라 예측 모드 X로 코딩된 것으로 취급된다.
- [0543] i. 한 예에서, X는 폭 및/또는 높이 등의 블록 치수들에 의존할 수 있다.
- [0544] 7. 라인 버퍼로부터 ALWIP 플래그의 저장을 제거하는 것이 제안된다.
- [0545] a. 한 예에서, 액세스될 2번째 블록이 현재 블록과 비교하여 상이한 LCU/CTU 행/영역에 위치해 있을 때, 2번

째 블록이 ALWIP로 코딩되어 있는지에 대한 조건부 체크는 스kip된다.

[0546] b. 한 예에서, 액세스할 2번째 블록이 현재 블록과 비교하여 상이한 LCU/CTU 행/영역에 있는 경우, 2번째 블록은, 일반 인트라 코딩된 블록으로서 취급되는 등의, 비-ALWIP 모드와 동일한 방식으로 취급된다.

[0547] 8. ALWIP 플래그를 인코딩할 때, K개( $K \geq 0$ )보다 많지 않은 컨텍스트들만 이용될 수 있다.

[0548] a. 한 예에서,  $K = 1$ 이다.

[0549] 9. ALWIP 모드와 연관된 모드 인덱스를 직접 저장하는 것 대신에 ALWIP 코딩된 블록들의 변환된 인트라 예측 모드를 저장하는 것이 제안된다.

[0550] a. 한 예에서, 하나의 ALWIP 코딩된 블록과 연관된 디코딩된 모드 인덱스는, 예를 들어 섹션 2.5.7에 설명된 바와 같은 [map\\_alwip\\_to\\_angular](#)에 따르는 등에 의해, 일반 인트라 모드에 맵핑된다.

[0551] b. 대안으로서, 또한, ALWIP 플래그의 저장이 완전히 제거된다.

[0552] c. 대안으로서, 또한, ALWIP 모드의 저장이 완전히 제거된다.

[0553] d. 대안으로서, 또한, 하나의 이웃/현재 블록이 ALWIP 플래그로 코딩되어 있는지에 대한 조건 체크는 스kip될 수 있다.

[0554] e. 대안으로서, 또한, ALWIP 코딩된 블록들에 대해 할당된 모드들의 변환 및 하나의 액세스된 블록과 연관된 일반 인트라 예측은 스kip될 수 있다.

#### 상이한 컬러 성분들에 관한 ALWIP

[0556] 10. 대응하는 루마 블록이 ALWIP 모드로 코딩된 경우 추론된 크로마 인트라 모드 (예를 들어, DM 모드)가 항상 적용될 수 있다는 것이 제안된다.

[0557] a. 한 예에서, 대응하는 루마 블록이 ALWIP 모드로 코딩된 경우 시그널링 없이 크로마 인트라 모드는 DM 모드인 것으로 추론된다.

[0558] b. 한 예에서, 대응하는 루마 블록은 주어진 위치(예를 들어, 현재 크로마 블록의 상단 좌측, 현재 크로마 블록의 중심)에 위치한 크로마 샘플의 대응하는 샘플을 커버하는 것일 수 있다.

[0559] c. 한 예에서, DM 모드는, (ALWIP) 모드를 일반 인트라 모드 중 하나로 맵핑하는 등을 통해, 대응하는 루마 블록의 인트라 예측 모드에 따라 도출될 수 있다.

[0560] 11. 크로마 블록의 대응하는 루마 블록이 ALWIP 모드로 코딩될 때, 수개의 DM 모드가 도출될 수 있다.

[0561] 12. 크로마 모드 결정 프로세스 (예를 들어, 직접 모드/도출 모드)에서, ALWIP 코딩되어 있는지 및/또는 그 ALWIP 모드를 결정하는데 이용되는 대응하는 루마 블록은, 코딩된 크로마 블록(들)의 치수들 및/또는 컬러 포맷과 함께 상단 좌측 위치에 따라 결정될 수 있다.

[0562] a. 코딩된 크로마 블록(들)의 상단 좌측 위치가 ( $x_{Cb}$ ,  $y_{Cb}$ )이고, 코딩된 크로마 블록(들)의 폭과 높이가 각각  $CbWidth$  및  $CbHeight$ 라고 가정하고, 여기서 모든 위치와 길이는 루마 샘플 유닛 내에 있다.

[0563] b. 한 예에서, 대응하는 루마 블록은  $offsetX$  및  $offsetY$  양쪽 모두가 0인 것이 불허되는 위치 ( $x_{Cb}+offsetY$ ,  $y_{Cb}+offsetY$ )를 커버하는 루마 블록 (예를 들어, 코딩 블록)으로서 선택될 수 있다.

[0564] i. 한 예에서  $offsetX = (CbWidth/2)$  또는  $(CbWidth/2-1)$  또는  $(CbWidth/2 + 1)$

[0565] ii. 한 예에서,  $offsetY = (CbHeight/2)$  또는  $(CbHeight/2 - 1)$  또는  $(CbHeight/2+1)$ .

[0566] 13. 하나의 대응하는 루마 블록이 ALWIP 모드로 코딩된 경우 특별 모드가 크로마 블록에 할당되는 것이 제안된다.

[0567] a. 한 예에서, 특별 모드는 ALWIP 코딩된 블록들과 연관된 인트라 예측 모드에 관계없이 주어진 일반 인트라 예측 모드인 것으로 정의된다.

[0568] b. 한 예에서, 인트라 예측의 상이한 방식들이 이 특별 모드에 할당될 수 있다.

[0569] c. 대안으로서, 루마 블록이 ALWIP 모드로 코딩되면, 크로마 DM 모드에 대한 연관된 일반 인트라 모드는 식

별된 특별 모드인 것으로 항상 추론될 수 있다.

[0570] d. 대안으로서, 또한, 크로마 모드들의 시그널링은 스kip될 수 있다.

[0571] e. 대안으로서, 크로마 블록(들)의 대응하는 루마 블록이 ALWIP 모드로 코딩되고 크로마 블록(들)에 대해 DM 모드가 시그널링될 때, 미리정의된 인트라 예측 모드가 크로마 블록(들)에 대해 적용된다.

[0572] i. 예를 들어, 미리정의된 인트라 예측 모드는 Planar 모드일 수 있다.

[0573] ii. 예를 들어, 미리정의된 인트라 예측 모드는 스크린 콘텐츠의 표시 등의 코딩된 정보에 의존할 수 있다.

[0574] 1) 한 예에서, 스크린 콘텐츠의 표시가 콘텐츠가 카메라 캡처 콘텐츠임을 나타내는 경우, Planar 모드로 설정될 수 있다.

[0575] 2) 한 예에서, 스크린 콘텐츠의 표시가 콘텐츠가 스크린 콘텐츠임을 나타내는 경우, 수평 예측 모드로 설정될 수 있다.

[0576] f. 한 예에서, 크로마 블록(들)의 대응하는 루마 블록이 ALWIP 모드로 코딩될 때, DM 모드는 크로마 블록(들)에 대해 절대 이용되지 않을 수 있다.

[0577] i. 예를 들어, DM 모드 관련 선택스는 시그널링되지 않을 수 있다.

[0578] 1) 한 예에서, 대응하는 루마 블록이 ALWIP 모드이고 sps\_cclm\_enabled\_flag가 거짓인 경우, 4개의 크로마 모드가 크로마 블록에 대해 시그널링되는 것이 허용될 수 있다.

[0579] a. 예를 들어, 이 경우, 인트라 크로마 모드들의 2진화를 위해 최대 2개의 빈이 요구될 수 있다.

[0580] 2) 한 예에서, 대응하는 루마 블록이 ALWIP 모드이고 sps\_cclm\_enabled\_flag가 참인 경우, 7개의 크로마 모드가 크로마 블록에 대해 시그널링되는 것이 허용될 수 있다.

[0581] a. 예를 들어, 이 경우, 인트라 크로마 모드들의 2진화를 위해 최대 4개의 빈이 요구될 수 있다.

[0582] 14. ALWIP는 또한, 크로마 성분에도 적용될 수 있다고 제안된다.

[0583] a. 한 예에서, 행렬 및/또는 편향 벡터는 상이한 컬러 성분들에 대해 상이할 수 있다.

[0584] b. 한 예에서, 행렬 및/또는 편향 벡터는 Cb 및 Cr에 대해 공동으로 미리정의될 수 있다

[0585] i. 한 예에서, Cb 및 Cr 성분은 연결될 수 있다.

[0586] ii. 한 예에서, Cb 및 Cr 성분은 인터리빙될 수 있다.

[0587] c. 한 예에서, 크로마 성분은 대응하는 루마 블록과 동일한 ALWIP 인트라 예측 모드를 공유할 수 있다.

[0588] i. 한 예에서, 대응하는 루마 블록이 ALWIP 모드를 적용하고 크로마 블록이 DM 모드로 코딩된 경우 동일한 ALWIP 인트라 예측 모드가 크로마 성분에 적용된다.

[0589] ii. 한 예에서, 동일한 ALWIP 인트라 예측 모드가 크로마 성분에 적용되고 그 후의 선형 보간은 스kip될 수 있다.

[0590] iii. 한 예에서, 동일한 ALWIP 인트라 예측 모드가 서브샘플링된 행렬 및/또는 편향 벡터와 함께 크로마 성분에 적용된다.

[0591] d. 한 예에서, 상이한 성분에 대한 ALWIP 인트라 예측 모드의 수는 상이할 수 있다.

[0592] i. 예를 들어, 크로마 성분들에 대한 ALWIP 인트라 예측 모드의 수는 동일한 블록 폭 및 높이에 대한 루마 성분에 대한 것보다 작을 수 있다.

### ALWIP의 적용가능성

[0594] 15. ALWIP가 적용가능한지가 시그널링할 수 있다는 것이 제안된다.

[0595] a. 예를 들어, 이것은, (예를 들어, SPS에서) 시퀀스 레벨, (예를 들어, PPS 또는 픽처 헤더에서) 픽처

레벨, (예를 들어, 슬라이스 헤더에서) 슬라이스 레벨, (예를 들어, 타일 그룹 헤더에서) 타일 그룹 레벨, 타일 레벨, CTU 행 레벨 또는 CTU 레벨에서 시그널링될 수 있다.

[0596] b. 예를 들어, `intra_lwip_flag`는 시그널링되지 않고 ALWIP가 적용될 수 없다면 0으로 추론될 수도 있다.

[0597] 16. ALWIP가 적용 가능한지는 블록 폭 (W) 및/또는 높이 (H)에 따라 달라질 수 있다는 것이 제안된다.

[0598] c. 예를 들어,  $W \geq T_1$  (또는  $W > T_1$ ) 및  $H \geq T_2$  (또는  $H > T_2$ )인 경우 ALWIP가 적용되지 않을 수 있다. 예를 들어  $T_1=T_2=32$ ;

[0599] i. 예를 들어,  $W \leq T_1$  (또는  $W < T_1$ ) 및  $H \leq T_2$  (또는  $H < T_2$ )인 경우 ALWIP가 적용되지 않을 수 있다. 예를 들어  $T_1=T_2=32$ ;

[0600] d. 예를 들어,  $W \geq T_1$  (또는  $W > T_1$ ) 또는  $H \geq T_2$  (또는  $H > T_2$ )인 경우 ALWIP가 적용되지 않을 수 있다. 예를 들어  $T_1=T_2=32$ ;

[0601] e. 예를 들어,  $W + H \geq T$  (또는  $W * H \geq T$ )인 경우 ALWIP가 적용되지 않을 수 있다. 예를 들어  $T = 256$ ;

[0603] i. 예를 들어  $W + H \leq T$  (또는  $W + H < T$ )인 경우 ALWIP가 적용되지 않을 수 있다. 예를 들어  $T = 256$ ;

[0604] f. 예를 들어  $W * H \geq T$  (또는  $W * H > T$ )인 경우 ALWIP가 적용되지 않을 수 있다. 예를 들어  $T = 256$ ;

[0605] i. 예를 들어  $W * H \leq T$  (또는  $W * H < T$ )인 경우 ALWIP가 적용되지 않을 수 있다. 예를 들어  $T = 256$ ;

[0606] g. 예를 들어, `intra_lwip_flag`는 시그널링되지 않고 ALWIP가 적용될 수 없다면 0으로 추론될 수 있다.

### ALWIP의 계산 문제들

[0608] 17. ALWIP에 수반된 임의의 시프트 동작은 S만큼 숫자를 좌측으로 시프트하거나 우측으로 시프트할 수 있다는 것이 제안된다, 여기서 S는 0보다 크거나 같아야 한다.

[0609] a. 한 예에서, 우측 시프트 동작은 S가 0과 같거나 0보다 클 때 상이할 수 있다.

[0610] i. 한 예에서 `upsBdryX[ x ]`는 다음과 같이 계산되어야 한다

uDwn>1일 때는

$$\text{upsBdryX}[x] = (\sum_{i=0}^{uDwn-1} \text{refX}[x * uDwn + i] + (1 << (\text{Log2}(uDwn) - 1))) >> \text{Log2}(uDwn), \quad uDwn = 1\text{일}$$

때는  $\text{upsBdryX}[x] = \sum_{i=0}^{uDwn-1} \text{refX}[x * uDwn + i]$

[0612] b. 한 예에서 `upsBdryX[ x ]`는 다음과 같이 계산되어야 한다

$$\text{upsBdryX}[x] = (\sum_{i=0}^{uDwn-1} \text{refX}[x * uDwn + i] +$$

$$(1 << \text{Log2}(uDwn) >> 1)) >> \text{Log2}(uDwn)$$

[0614] 18. ALWIP의 업샘플링 프로세스에서 결과들을 0에 가깝게 또는 0에서 멀리 반올림해야 한다고 제안된다.

[0615] a. 한 예에서,

$$\text{predSamples}[xHor + dX][yHor] = ((\text{upHor} - dX) * \text{predSamples}[xHor][yHor]$$

$$+ dX * \text{predSamples}[xHor + upHor][yHor] + \text{offsetHor}) / \text{upHor} \quad (8-X39)$$

[0617] 및

$$\text{predSamples}[xVer][yVer + dY] = ((\text{upVer} - dY) * \text{predSamples}[xVer][yVer] +$$

$$dY * \text{predSamples}[xVer][yVer + upVer] + \text{offsetVer}) / \text{upVer} \quad (8-X40)$$

- [0619] 여기서 offsetHor 및 offsetVer는 정수들이다. 예를 들어, offsetHor = upHor/2 및 offsetVer=upVer/2이다.
- [0620] **다른 코딩 틀들과의 상호작용**
- [0621] 19. CIIP-코딩된 블록에 대해 ALWIP가 이용될 수 있다는 것이 제안된다.
- [0622] a. 한 예에서, CIIP-코딩된 블록에서, ALWIP 인트라 예측 모드 또는 Planar 등의 일반 인트라 예측 모드가 인트라 예측 신호를 생성하는데 이용되는지가 명시적으로 시그널링될 수 있다.
- [0623] b. 한 예에서, ALWIP 인트라 예측 모드 또는 Planar 등의 일반 인트라 예측 모드가 인트라 예측 신호를 생성하는데 이용될 수 있는지가 묵시적으로 추론될 수 있다.
- [0624] i. 한 예에서, ALWIP 인트라 예측 모드는 CIIP 코딩된 블록에서 결코 이용되지 않을 수 있다.
- [0625] 1) 대안으로서, CIIP 코딩된 블록에서 일반 인트라 예측은 절대 이용되지 않을 수 있다.
- [0626] ii. 한 예에서, 이웃 블록들의 정보로부터, 인트라 예측된 신호를 생성하기 위해 ALWIP 인트라 예측 모드 또는 Planar 등의 일반 인트라 예측 모드가 이용되는지가 추론될 수 있다.
- [0627] 20. CCLM 모드에서 이웃 루마 샘플들을 다운샘플링하는데 이용되는 절차의 전체 또는 일부가 ALWIP 모드에서 이웃 샘플들을 다운샘플링하는데 이용될 수 있다는 것이 제안된다.
- [0628] a. 대안으로서, ALWIP 모드에서 이웃 루마 샘플들을 다운샘플링하는데 이용되는 절차의 전체 또는 일부는 CCLM 모드에서 이웃 샘플들을 다운샘플링하는데 이용될 수 있다.
- [0629] b. 다운샘플링 절차는 CCLM 프로세스 및 ALWIP 프로세스에서 이용될 때 상이한 파라미터들/인수들과 함께 기동될 수 있다.
- [0630] c. 한 예에서, CCLM 프로세스의 (이웃 루마 위치들의 선택, 다운샘플링 필터들 등의) 다운샘플링 방법은 ALWIP 프로세스에서 이용될 수 있다.
- [0631] d. 이웃 루마 샘플들을 다운샘플링하는데 이용되는 절차는, 적어도 다운샘플링된 위치들의 선택, 다운샘플링 필터들, 반올림 및 클리핑 동작들을 포함한다.
- [0632] 21. ALWIP 모드로 코딩된 블록은 RST 또는/및 2차 변환 또는/및 회전 변환 또는/및 비분리형 2차 변환 (NSST)을 적용할 수 없다는 것이 제안된다.
- [0633] a. 한 예에서, 이러한 제약이 적용될 수 있는지의 여부는 블록들의 치수 정보에 의존할 수 있으며, 예를 들어 (15)에 설명된 조건과 동일하다.
- [0634] b. 대안으로서, RST 또는/및 2차 변환 또는/및 회전 변환 또는/및 NSST가 적용될 때 ALWIP 모드는 불허될 수 있다.
- [0635] c. 대안으로서, ALWIP 모드로 코딩된 블록은 RST 또는/및 2차 변환 또는/및 회전 변환 또는/및 비분리 2차 변환 (NSST)을 적용할 수 있다.
- [0636] i. 한 예에서, 변환 행렬의 선택은 ALWIP 인트라 예측 모드에 의존할 수 있다.
- [0637] ii. 한 예에서, 변환 행렬의 선택은 ALWIP 인트라 예측 모드로부터 변환되는 일반 인트라 예측 모드에 의존할 수 있다.
- [0638] iii. 한 예에서, 변환 행렬의 선택은 ALWIP 인트라 예측 모드로부터 변환되는 일반 인트라 예측 모드에 따른 분류에 의존할 수 있다.
- [0639] 22. ALWIP 모드로 코딩된 블록은 BDPCM (Block-based DPCM) 또는 Residue RDPCM을 적용할 수 없음이 제안된다.
- [0640] a. 대안으로서, BDPCM 또는 RDPCM이 적용되는 경우 ALWIP 모드가 불허될 수 있다.
- [0641] 23. ALWIP 모드로 코딩된 블록은 변환으로서 DCT-II만 이용할 수 있다는 것이 제안된다.
- [0642] a. 한 예에서, 변환 행렬 인덱스들의 시그널링은 항상 스kip된다.
- [0643] b. 대안으로서, ALWIP 모드로 코딩된 블록에 이용되는 변환이 명시적으로 시그널링되는 대신 묵시적으로 도

출될 수 있다는 것이 제안된다. 예를 들어, 변환은 JVET-M0303에서 제안된 방식에 따라 선택될 수 있다.

[0644] c. 대안으로서, ALWIP 모드로 코딩된 블록은 변환 스킵만을 이용할 수 있다는 것이 제안된다.

[0645] i. 대안으로서, 또한 ALWIP가 이용될 때, 변환 스킵 이용 표시의 시그널링이 스킵된다.

[0646] d. 한 예에서, (인에이블됨/디스에이블됨, 예측 모드 인덱스 등의) ALWIP 모드 정보는 변환 행렬의 표시들 후에 조건부로 시그널링될 수 있다.

[0647] i. 한 예에서, (변환 스킵 또는 DCT-II 등의) 주어진 변환 행렬에 대해, ALWIP 모드 정보의 표시들이 시그널링될 수 있다.

[0648] ii. 대안으로서, 또한, ALWIP 모드 정보의 표시들은 일부 미리정의된 변환 행렬들에 대해 스kip될 수 있다.

[0649] 24. ALWIP 모드로 코딩된 블록은, 선택된 변환이 모드-의존적일 때 ALWIP 인트라 예측 모드로부터 변환된 일반 인트라 예측으로 코딩된 것으로 간주되는 것이 제안된다.

[0650] 25. ALWIP 모드는 변환 스킵을 이용하지 않을 수 있다.

[0651] a. 예를 들어, 이 경우 변환 스킵 이용의 표시를 추가로 시그널링할 필요가 없다.

[0652] b. 대안으로서, 변환 스킵이 적용될 때 ALWIP 모드가 불허될 수 있다.

[0653] i. 예를 들어, 이 경우 변환 스킵이 적용될 때 ALWIP 모드 정보를 시그널링할 필요가 없다.

[0654] 26. 디블로킹 필터, SAO(sample adaptive offset), ALF(adaptive loop filter) 등의 필터링 프로세스에서, 필터들을 선택하는 방법 및/또는 샘플들을 필터링할지는 ALWIP를 이용하여 결정될 수 있다.

[0655] 27. 필터링되지 않은 이웃 샘플들은 ALWIP 모드에서 이용될 수 있다.

[0656] a. 대안으로서, 필터링된 이웃 샘플들은 ALWIP 모드에서 이용될 수 있다.

[0657] b. 한 예에서, 필터링된 이웃 샘플들은 다운샘플링을 위해 이용될 수 있고 필터링되지 않은 이웃 샘플들은 업샘플링을 위해 이용될 수 있다.

[0658] c. 한 예에서, 필터링되지 않은 이웃 샘플들은 다운샘플링을 위해 이용될 수 있고 필터링된 이웃 샘플들은 업샘플링을 위해 이용될 수 있다.

[0659] d. 한 예에서, 필터링된 좌측 이웃 샘플들은 업샘플링에서 이용될 수 있고 필터링되지 않은 위쪽 이웃 샘플들은 업샘플링에서 이용될 수 있다.

[0660] e. 한 예에서, 필터링되지 않은 좌측 이웃 샘플들은 업샘플링에서 이용될 수 있고 필터링된 위쪽 이웃 샘플들은 업샘플링에서 이용될 수 있다.

[0661] f. 한 예에서, 필터링된 또는 필터링되지 않은 이웃 샘플들이 이용되는지는 ALWIP 모드에 의존할 수 있다.

[0662] i. 한 예에서, ALWIP 모드는 전통적인 인트라 예측 모드로 변환될 수 있고, 필터링된 또는 필터링되지 않은 이웃 샘플들이 이용되는지는 변환된 전통적인 인트라 예측 모드에 의존할 수 있다. 예를 들어, 이러한 결정은 종래의 인트라 예측 모드들과 동일하다.

[0663] ii. 대안으로서, 필터링된 또는 필터링되지 않은 이웃 샘플들이 ALWIP 모드에 이용되는지는 시그널링될 수 있다.

[0664] g. 한 예에서, 필터링된 샘플들은 전통적인 인트라 예측 모드들과 동일하게 생성될 수 있다.

[0665] 28. 어떤 행렬들 또는/및 오프셋 벡터들이 이용되는지는 재성형 (일명 LMCS, 크로마 스케일링이 있는 루마 맵핑) 정보에 의존할 수 있다.

[0666] a. 한 예에서, 재성형이 온 및 오프일 때 상이한 행렬들 또는/및 오프셋 벡터들이 이용될 수 있다.

[0667] b. 한 예에서, 상이한 재성형 파라미터들에 대해 상이한 행렬들 또는/및 오프셋 벡터들이 이용될 수 있다.

[0668] c. 한 예에서, ALWIP는 항상 원래 도메인에서 수행될 수 있다.

[0669] i. 예를 들어, 이웃 샘플들은 ALWIP에서 이용되기 전에 원래 도메인(재성형이 적용된 경우)에 맵핑된다.

- [0670] 29. 재성형이 적용될 때 ALWIP가 디스에이블될 수 있다.
- [0671] a. 대안으로서, ALWIP가 인에이블된 경우 재성형이 디스에이블될 수 있다.
- [0672] b. 한 예에서, 재성형이 적용될 때 HDR (높은 동적 범위) 콘텐츠에 대해 ALWIP는 디스에이블될 수 있다.
- [0673] 30. ALWIP에서 이용되는 행렬들은 샘플 비트 깊이에 의존할 수 있다.
- [0674] a. 대안으로서, 또한, ALWIP에서 이용되는 오프셋 값들은 샘플 비트 깊이에 의존할 수 있다.
- [0675] b. 대안으로서, 행렬 파라미터들 및 오프셋 값들은 N비트 샘플들 ( $M \leq N$ )에 대해 M비트 정밀도로 저장될 수 있다, 예를 들어, 행렬 파라미터들 및 오프셋 값들은 10비트 샘플에 대해 8비트 정밀도로 저장될 수 있다.
- [0676] c. 샘플 비트 깊이는 루마 등의 컬러 성분에 대한 입력 어레이의 비트 깊이일 수 있다.
- [0677] d. 샘플 비트 깊이는 루마 등의 컬러 성분에 대한 내부 어레이/복원된 샘플의 비트 깊이일 수 있다.
- [0678] 31. 명시된 블록 크기에 대한 행렬 파라미터들 및/또는 오프셋 값들은 다른 블록 크기들에 대한 행렬 파라미터들 및/또는 오프셋 값들로부터 도출될 수 있다.
- [0679] 32. 한 예에서, 8x8 블록의 16x8 행렬은 4x4 블록의 16x4 행렬로부터 도출될 수 있다.
- [0680] 33. ALWIP에 의해 생성된 예측은, 추가로 이용될 예측 신호를 획득하기 위해 처리될 중간 신호로서 취급될 수 있는 것이 제안된다.
- [0681] a. 한 예에서, 위치 의존 인트라 예측 조합(Position Dependent Intra Prediction Combination)(PDPC)은 추가로 이용될 예측 신호를 생성하기 위해 ALWIP에 의해 생성된 예측에 적용될 수 있다.
- [0682] i. 한 예에서, PDPC는, 블록이 Planar 또는 DC 등의 특정한 일반 인트라 예측 모드로 코딩되는 것과 동일한 방식으로, ALWIP 코딩된 블록에 관해 이루어진다.
- [0683] ii. 한 예에서, PDPC는, ALWIP 인트라 예측 모드로부터 변환되는 일반 인트라 예측 모드로 코딩된 블록과 동일한 방식으로 ALWIP 코딩된 블록에 관해 이루어진다.
- [0684] iii. 한 예에서, PDPC는 ALWIP 코딩된 블록에 관해 조건부로 적용된다.
- [0685] 1) 예를 들어, ALWIP 인트라 예측 모드로부터 변환되는 일반 인트라 예측 모드에 관해 PDPC가 적용되는 경우에만 ALWIP 코딩된 블록에 PDPC가 적용된다.
- [0686] b. 한 예에서, ALWIP에 의해 생성된 경계 샘플들 예측은, 추가로 이용될 예측 신호를 생성하기 위해 이웃 샘플들로 필터링될 수 있다.
- [0687] i. 한 예에서, 경계 샘플들에 관한 필터링은, 블록이 Planar 또는 DC 등의 특정한 일반 인트라 예측 모드로 코딩되는 것과 동일한 방식으로 ALWIP 코딩된 블록에 관해 이루어진다.
- [0688] ii. 한 예에서, 경계 샘플들에 관한 필터링은, ALWIP 인트라 예측 모드로부터 변환되는 일반 인트라 예측 모드로 코딩된 블록과 동일한 방식으로 ALWIP 코딩된 블록에 관해 이루어진다.
- [0689] iii. 한 예에서, 경계 샘플에 관한 필터링은 ALWIP 코딩된 블록에 관해 조건부로 적용된다.
- [0690] 1) 예를 들어, 경계 샘플들에 관한 필터링은, ALWIP 인트라 예측 모드로부터 변환되는 일반 인트라 예측 모드에서 경계 샘플들에 대한 필터링이 적용되는 경우에만 ALWIP 코딩된 블록에 관해 적용된다.
- [0691] 34. ALWIP의 업샘플링 프로세스에서 쌍선형 보간 필터 이외의 보간 필터가 이용될 수 있다는 것이 제안된다.
- [0692] a. 한 예에서, 4-탭 보간 필터들은 ALWIP의 업샘플링 프로세스에서 이용될 수 있다.
- [0693] i. 예를 들어, 크로마 성분들에 대한 모션 보상을 수행하는데 이용되는 VVC에서의 4-탭 보간 필터들은 ALWIP의 업샘플링 프로세스에서 이용될 수 있다.
- [0694] ii. 예를 들어, 각도 인트라 예측을 수행하는데 이용되는 VVC에서의 4-탭 보간 필터들은 ALWIP의 업샘플링 프로세스에서 이용될 수 있다.
- [0695] iii. 예를 들어, 루마 성분에 대한 모션 보상을 수행하는데 이용되는 VVC에서의 8-탭 보간 필터들은 ALWIP의 업샘플링 프로세스에서 이용될 수 있다.

- [0696] 35. ALWIP 모드로 코딩된 블록 내의 샘플들은 상이한 방식들로 예측될 수 있다.
- [0697] a. 한 예에서,  $W*H$  블록에 대해,  $sW*sH$  ALWIP를 적용함으로써 블록 내의  $sW*sH$  서브블록의 예측이 생성될 수 있다.
- [0698] i. 한 예에서,  $W*H$  블록에 대해,  $W/2*H/2$  ALWIP를 적용함으로써 그 상단 좌측  $W/2*H/2$  블록의 예측이 생성될 수 있다.
- [0699] ii. 한 예에서,  $W*H$  블록에 대해,  $W/2*H$  ALWIP를 적용함으로써 그 좌측  $W/2*H$  블록의 예측이 생성될 수 있다.
- [0700] iii. 한 예에서,  $W*H$  블록에 대해,  $W*H/2$  ALWIP를 적용함으로써 그 상단  $W*H/2$  블록의 예측이 생성될 수 있다.
- [0701] iv. 한 예에서,  $sW*sH$  서브블록은 이용가능한 좌측 또는/및 위쪽 이웃 샘플들을 가질 수 있다.
- [0702] b. 한 예에서, 서브블록의 위치를 결정하는 방법은 블록들의 치수에 의존할 수 있다.
- [0703] i. 예를 들어,  $W \geq H$ 인 경우,  $W/2*H$  ALWIP를 적용함으로써 그 좌측  $W/2*H$  블록의 예측이 생성될 수 있다.
- [0704] ii. 예를 들어,  $H \geq W$ 인 경우,  $W*H/2$  ALWIP를 적용함으로써 그 상단  $W*H/2$  블록의 예측이 생성될 수 있다.
- [0705] iii. 예를 들어,  $W$ 가  $H$ 와 같을 때,  $W/2*H/2$  ALWIP를 적용함으로써 그 상단 좌측  $W/2*H/2$  블록의 예측이 생성될 수 있다.
- [0706] c. 한 예에서, 또한,  $W*H$  ALWIP를 적용함으로써 나머지 샘플들 (예를 들어, 샘플들은  $sW*sH$  서브블록에 속하지 않음)의 예측이 생성될 수 있다.
- [0707] i. 대안으로서, (예를 들어, 변환된 인트라 예측 모드를 인트라 모드로서 이용한) 종래의 인트라 예측을 적용함으로써 나머지 샘플들의 예측이 생성될 수 있다.
- [0708] ii. 또한,  $sW*sH$  서브블록 내의 샘플들에 대해서는 계산이 스kip될 수 있다.
- [0709] 36. ALWIP 모드에서 코딩된 블록 내의 샘플들은 (예를 들어, 크기  $sW*sH$ 를 갖는) 서브블록 레벨에서 예측될 수 있다.
- [0710] a. 한 예에서,  $sW*sH$  ALWIP는, (예를 들어, 경계 서브블록에 대해) 이웃 복원된 샘플들 또는/및 (예를 들어, 내부 서브블록들에 대해) 이웃 예측된 샘플들을 이용하여 각각의 서브블록에 적용될 수 있다.
- [0711] b. 한 예에서, 서브블록들은 래스터-스캔 순서로 예측될 수 있다.
- [0712] c. 한 예에서, 서브블록들은 지그재그 순서로 예측될 수 있다.
- [0713] d. 한 예에서, 서브블록의 폭 (높이)는  $sWMax$  ( $shMax$ )보다 크지 않을 수 있다.
- [0714] e. 한 예에서, 폭 또는 높이, 또는 폭과 높이 양쪽 모두를 갖는 블록이 양쪽 모두 임계값  $L$ 보다 크거나 (또는 동일할 때), 블록은 복수의 서브블록으로 분할될 수 있다.
- [0715] f. 임계값  $L$ 은 미리정의되거나 SPS/PPS/픽처/슬라이스/타일 그룹/타일 레벨에서 시그널링될 수 있다.
- [0716] i. 대안으로서, 임계값들은, 블록 크기, 픽처 유형, 시간 계층 인덱스 등의, 소정의 코딩된 정보에 의존할 수 있다.
- [0717] 37. (인접한 또는 인접하지 않은) 이웃 샘플들은 ALWIP에서 이용되기 전에 필터링된다는 것이 제안된다.
- [0718] a. 대안으로서, 이웃 샘플들은 ALWIP에서 이용되기 전에 필터링되지 않는다.
- [0719] b. 대안으로서, 이웃 샘플들은 ALWIP에서 이용되기 전에 조건부로 필터링된다.
- [0720] i. 예를 들어, 이웃 샘플들은, ALWIP 인트라 예측 모드가 하나 또는 일부 특정한 값들과 동일한 경우에만 ALWIP에서 이용되기 전에 필터링된다.
- [0721] 38. ALWIP 플래그를 코딩할 때, 산술 코딩에서 ALWIP 플래그에 대한 컨텍스트를 도출하는 방법은 현재 블록의

모든 치수에 대해 동일하다는 것이 제안된다.

[0722] a. 한 예에서, 산술 코딩에서 ALWIP 플래그에 대한 컨텍스트를 도출하는 방법은, ( $\text{Abs}(\text{Log2(cbWidth)} - \text{Log2(cbHeight)})$ )가 1보다 크거나 작을 때 동일하고, 여기서 CbWidth 및 CbHeight는 각각 현재 블록의 폭과 높이이다.

[0723] b. 한 예에서, 산술 코딩에서 ALWIP 플래그에 대한 컨텍스트의 도출은 이웃 블록들의 ALWIP 정보 및/또는 이웃 블록들의 가용성에만 의존한다.

[0724] i. 한 예에서, 복수의 이웃 블록 ALWIP 정보 (예를 들어, intra\_lwip\_flag) 및/또는 이웃 블록들의 가용성이 직접 이용된다. 예를 들어, 좌측 및 위쪽 이웃 블록들의 ALWIP 플래그들 및/또는 좌측 및 이웃 블록들의 가용성을 산술 코딩에서 ALWIP 플래그에 대한 컨텍스트를 도출하는데 이용된다. 한 예가 표 5에 도시되어 있다. 대안으로서, 또한, 컨텍스트 인덱스 오프셋  $\text{ctxInc} = (\text{condL} \&& \text{availableL}) + (\text{condA} \&& \text{availableA}) + \text{ctxSetIdx} * 3$ 이다.

표-5 – 좌측 및 위쪽 신팩스 요소들을 이용한 ctxInc의 명세

신팩스 요소	condL	condA	ctxSetI dx
intra_lwip_flag[ x 0 ][ y0 ]	intra_lwip_flag[ xNbL ][ yNbL ]	intra_lwip_flag[ xNbA ][ yNbA ]	0

[0726]

[0727] ii. 한 예에서, 이웃 블록의 ALWIP 정보 중 하나(예를 들어, intra\_lwip\_flag)는 산술 코딩에서 ALWIP 플래그에 대한 컨텍스트를 도출하는데 이용되고, 이웃 블록은 좌측 이웃 블록일 수 있다. 한 예가 표 6에 도시되어 있다. 대안으로서, 또한, 컨텍스트 인덱스 오프셋  $\text{ctxInc} = (\text{condL} \&& \text{availableL}) + \text{ctxSetIdx} * 3$ 이다.

표-6 – 좌측 및 위쪽 신팩스 요소들을 이용한 ctxInc의 명세

신팩스 요소	condL	condA	ctxSetI dx
intra_lwip_flag[ x 0 ][ y0 ]	intra_lwip_flag[ xNbL ][ yNbL ]		0

[0729]

[0730] iii. 한 예에서, 이웃 블록의 ALWIP 플래그 정보 중 하나(예를 들어, intra\_lwip\_flag)는 산술 코딩에서 ALWIP 플래그에 대한 컨텍스트를 도출하는데 이용되고, 이웃 블록은 위쪽 이웃 블록일 수 있다. 한 예가 표 7에 도시되어 있다. 대안으로서, 또한, 컨텍스트 인덱스 오프셋  $\text{ctxInc} = (\text{condA} \&& \text{availableA}) + \text{ctxSetIdx} * 3$ 이다.

표-7 – 좌측 및 위쪽 신팩스 요소들을 이용한 ctxInc의 명세

신팩스 요소	condL	condA	ctxSetI dx
intra_lwip_flag[ x 0 ][ y0 ]		intra_lwip_flag[ xNbA ][ yNbA ]	0

[0732]

[0733] c. 한 예에서, 산술 코딩에서 ALWIP 플래그를 코딩하기 위해 하나의 고정 컨텍스트가 이용된다.

[0734] d. 한 예에서, ALWIP 플래그는 산술 코딩에서 바이패스 코딩된다.

[0735] e. 대안으로서, K개 컨텍스트들이 산술 코딩에서 ALWIP 플래그를 코딩하는데 이용될 수 있다. 이용되는 컨텍스트는 블록의 치수 (예를 들어, W로 표시된 폭 및 H로 표시된 높이)에 의존할 수 있다.

[0736] i. 한 예에서 K는 2와 같다.  $W > N * H$  또는  $H > N * W$  (예를 들어,  $N = 2$ )일 때, 첫 번째 컨텍스트가 이용되며, 그렇지 않으면 2번째 컨텍스트가 이용된다.

- [0737] 39. N개 ( $N \geq 0$ ) 컨텍스트들은 산술 코딩에서 ALWIP 플래그 (예를 들어, `intra_lwip_flag`)를 코딩하는데 이용될 수 있다는 것이 제안된다.
- [0738] a. 한 예에서, N은 3과 같다. ALWIP 플래그 및/또는 2개의 이웃하는/하거나 이웃하지 않는 블록들의 가용성은, 산술 코딩에서 ALWIP 플래그에 대해 이용된 컨텍스트를 도출하는데 이용될 수 있다.
- [0739] i. 한 예에서, 2개의 이웃 블록은 위쪽 (예를 들어, 도 10에서 B1) 블록 및 좌측 (예를 들어, 도 10에서 A1) 블록을 포함할 수 있다.
- [0740] ii. 한 예에서, 2개의 이웃 블록은 위쪽 블록 및 좌측 아래 (예를 들어, 도 10의 A2) 블록을 포함할 수 있다.
- [0741] iii. 한 예에서, 2개의 이웃 블록은 위쪽 블록 및 위쪽-우측 (예를 들어, 도 10의 B2) 블록을 포함할 수 있다.
- [0742] iv. 한 예에서, 2개의 이웃 블록은 위쪽-우측 (예를 들어, 도 10의 B2) 블록 및 좌측 (예를 들어, 도 10의 A1) 블록을 포함할 수 있다.
- [0743] v. 한 예에서, 2개의 이웃 블록은 우측 위쪽(예를 들어, 도 10에서 B2) 블록 및 좌측 아래 (예를 들어, 도 10에서 A2) 블록을 포함할 수 있다.
- [0744] vi. 한 예에서, 2개의 이웃 블록은 좌측 블록 (예를 들어, 도 10에서 A1) 및 좌측 아래 (예를 들어, 도 10에서 A2) 블록을 포함할 수 있다.
- [0745] vii. 한 예에서, 이웃 블록은 도 10과 상이하게 정의될 수 있다. 한 예가 도 16에 설명되어 있다. 2개의 이웃 블록은 {우측-위쪽, 위쪽, 좌측-위쪽, 좌측, 좌측-아래} 블록들 중 임의의 2개를 포함할 수 있다. 예를 들어, 2개의 이웃 블록은 {B0, B1, B2, A0, A1}의 블록들 중 임의의 2개를 포함할 수 있다.
- [0746] b. 한 예에서 N은 2와 같다. ALWIP 플래그 및/또는 1개의 이웃하는/하거나 이웃하지 않는 블록의 가용성은, 산술 코딩에서 ALWIP 플래그에 대해 이용된 컨텍스트를 도출하는데 이용될 수 있다.
- [0747] i. 한 예에서, 이웃 블록은 {우측-위쪽, 위쪽, 좌측-위쪽, 좌측, 좌측-아래} 중 어느 하나일 수 있다. 이웃 블록의 한 예가 도 10에 설명되어 있다.
- [0748] ii. 한 예에서, 이웃 블록은 {우측-위쪽, 위쪽, 좌측-위쪽, 좌측, 좌측-아래} 블록 중 어느 하나일 수 있다. 이웃 블록의 한 예가 도 16에 설명되어 있다.
- [0749] c. 한 예에서, 산술 코딩에서 ALWIP 플래그를 코딩하기 위해 하나의 고정된 컨텍스트가 이용될 수 있다.
- [0750] d. 한 예에서, ALWIP 플래그는 산술 코딩에서 바이패스 코딩될 수 있다. 도 16은 이웃 블록들의 한 예를 도시한다.
- [0751] 40. 업샘플링 경계 샘플들을 계산하지 않고 감축된 경계 샘플들이 생성될 수 있다는 것이 제안된다.
- [0752] a. 한 예에서, 업샘플링 경계 샘플 위치들에 위치한 기준 샘플들은 예측 업샘플링 프로세스에 직접 이용된다.
- [0753] i. 한 예에서, 업샘플링 경계 샘플들은 복수의 이웃한 기준 샘플들을 평균화함으로써 계산되지 않을 수 있다.
- [0754] b. 한 예에서, 감축된 경계 샘플들은 기준 샘플 및 다운스케일링 인자로부터 직접 계산될 수 있다.
- [0755] i. 한 예에서, 다운스케일링 인자는 변환 블록 크기 및 다운샘플링된 경계 크기에 의해 계산될 수 있다.
- [0756] 41. 행렬 곱셈에 이용되는 감축된 경계 샘플들은 한 스테이지에서 생성될 수 있다는 것이 제안된다.
- [0757] a. 한 예에서, 이들은 한 스테이지에서 원래의 복원된 이웃 샘플들로부터 직접 생성될 수 있다 (VVC WD5는 섹션 2.2.1.5.4.4에 설명된 바와 같이 ALWIP 감축된 경계 샘플들을 생성하기 위해 2-스테이지 다운샘플링을 이용한다는 점에 유의한다), 여기서, 원래의 복원된 이웃 샘플들은 추가 처리없이 이웃 샘플들을 디코딩할 수 있다. 예를 들어, 원래의 복원된 이웃 샘플들은 각도 인터 예측 샘플들을 생성하는데 이용될 수 있다.
- [0758] b. 한 예에서, 감축된 경계 샘플들은 현재 블록의 상단 이웃 행들 및/또는 좌측 이웃 열들에 위치한 원래의 복원된 샘플들로부터 생성될 수 있다.

- [0759] i. 예를 들어, N개의 감축된 경계 샘플이 현재 블록에 (주어진 순서로) 이웃한 M개의 원래의 복원된 샘플로부터 생성될 필요가 있다고 가정하면, 각각의 K개의 연속적인 원래의 복원된 이웃 샘플들이 하나의 출력 감축된 경계 샘플을 얻는데 이용될 수 있다.
- [0760] 1) 한 예에서,  $K = M/N$ .
- [0761] a. 대안으로서,  $K = (M+N/2)/N$ .
- [0762] 2) 한 예에서, 하나의 출력 감축된 경계 샘플은 K개의 연속적인 원래의 복원된 이웃 샘플들의 평균으로서 도출될 수 있다.
- [0763] 3) 한 예에서, 하나의 출력 감축된 경계 샘플은 K개의 연속적인 원래의 복원된 이웃 샘플들의 각종 평균으로서 도출될 수 있다.
- [0764] c. 한 예에서, 좌측 감축된 경계 샘플들은 현재 블록의 좌측 이웃 열들에 위치한 원래의 복원된 샘플들로부터 생성될 수 있는 반면, 상단 감축된 샘플들은 현재 블록의 상단 이웃 행들에 위치한 원래의 복원된 샘플들로부터 생성될 수 있다.
- [0765] i. 예를 들어, 도 17에 도시된 바와 같이, 각각  $\text{boundary}_{\text{left}}$  및  $\text{boundary}_{\text{top}}$ 으로 표시된 좌측 경계 및 상단 경계의 4개의 감축된 경계 샘플들은, 현재  $16 \times 16$  ALWIP 블록의 좌측/상단 원래의 이웃 복원된 샘플(도면에서  $16 \times 16$  블록에 이웃하는 회색 격자들로 마킹됨)에 의해 생성된다.
- [0766] d. 감축된 경계 샘플들을 생성하는 방법은 블록 치수들/코딩된 정보(예를 들어, 인트라 예측 모드, 변환 유형 등)에 의존할 수 있다.
- [0767] e. 한 예에서, 앞서 언급된 방법은 감축된 경계 샘플들을 생성할 것을 요구하는 모든 크기의 ALWIP 블록(예를 들어,  $4 \times 4$  ALWIP 블록들로부터  $64 \times 64$  ALWIP 블록들까지)에 적용될 수 있다
- [0768] f. 한 예에서, 현재 블록의 좌측 이웃 열들 및 현재 블록의 상단 이웃 행들에 대한 감축된 경계 샘플들을 위한 생성 프로세스는 상이한 방식들로 수행될 수 있다.
- [0769] i. 예를 들어,  $8 \times 4$  ALWIP 블록의 경우, 미리정의된 감축된 경계 샘플들의 수는, 상단에 4개, 좌측에 4개이며,  $8 \times 4$  ALWIP 블록의 상단 행에 위치한 8개의 이웃 샘플들을 이용하여 상단에 4개의 감축된 경계 샘플들을 생성하는 반면,  $8 \times 4$  ALWIP 블록의 좌측 열에 있는 4개의 이웃 샘플들은 좌측의 4개의 감축된 경계 샘플들로서 직접 복사된다.
- [0770] 42. 감축된 예측 블록으로부터 최종 예측 블록을 생성하기 위해 업샘플링 프로세스에서 (현재 블록에 이웃하거나 이웃하지 않는) 원래의 복원된 이웃 샘플들의 전부 또는 일부를 이용하는 것이 제안된다.
- [0771] a. 한 예에서, 원래의 복원된 이웃 샘플들은 현재 블록의 상단 이웃 행들 및/또는 좌측 이웃 열들에 위치할 수 있다. 한 예가 도 18에 도시되어 있으며, 여기서  $64 \times 64$  최종 예측 블록은,  $8 \times 8$  감축된 예측 블록 +  $64 \times 64$  블록의 원래의 복원된 이웃 샘플들로부터 업샘플링함으로써 생성된다.
- [0772] i. 대안으로서, 더욱이, 감축된 경계 샘플은 감축된 예측 블록을 획득하기 위한 행렬 곱셈에만 이용될 수 있지만, 최종 예측 블록을 생성하기 위한 업샘플링 프로세스에서는 이용되지 않을 수 있다. 예를 들어, K개의 감축된 경계 샘플은  $M \times N$  감축된 예측 블록을 생성하기 위해 ALWIP의 행렬 곱셈에 입력될 수 있지만, 업샘플링 프로세스에서 최종 예측 블록을 생성하는데 이용되지 않을 수 있다. 예를 들어  $K=8$ 이고  $M \times N$ 은  $8 \times 8$ 이다.
- [0773] b. 한 예에서, 선택된 원래의 복원된 이웃 샘플들은 감축된 예측 블록으로부터 최종 예측 블록을 생성하기 위해 업샘플링 프로세스에서 이용될 수 있다.
- [0774] i. 예를 들어, 현재 블록 좌측의 모든 원래의 복원된 이웃 샘플들이 선택될 수 있다.
- [0775] ii. 예를 들어, 현재 블록 위의 모든 원래의 복원된 이웃 샘플들이 선택될 수 있다.
- [0776] iii. 예를 들어, 현재 블록 좌측의 각각의 M개의 연속적인 원래의 복원된 이웃 샘플들 중 K개가 선택될 수 있다, 예를 들어  $K = 1$ ,  $M = 2/4/8$ .
- [0777] 1) 예를 들어, 각각의 M개의 연속적인 이웃들 중 마지막 K개의 원래의 복원된 이웃 샘플들이 선택될 수 있다.

- [0778] 2) 예를 들어, 각각의 M개의 연속적인 이웃들 중 처음 K개의 원래의 복원된 이웃 샘플들이 선택될 수 있다.
- [0779] iv. 예를 들어, 현재 블록 위쪽의 각각의 M개의 연속적인 원래의 복원된 이웃 샘플들 중 K개가 선택될 수 있다, 예를 들어  $K = 1, M = 2/4/8$ .
- [0780] 1) 예를 들어, 각각의 M개의 연속적인 이웃들 중 마지막 K개의 원래의 복원된 이웃 샘플들이 선택될 수 있다.
- [0781] 2) 예를 들어, 각각의 M개의 연속적인 이웃들 중 처음 K개의 원래의 복원된 이웃 샘플들이 선택될 수 있다.
- [0782] v. 예를 들어, 선택은 블록 폭 및 높이에 의존할 수 있다. blkW 및 blkH가 각각 ALWIP 블록의 폭과 높이를 나타낸다고 가정한다. 그리고 (blkX, blkY)는 블록의 상단 좌측 위치를 나타낸다.
- [0783] 1) 예를 들어, blkW가 blkH보다 크거나 같으면, 현재 블록 좌측의 모든 원래의 복원된 이웃 샘플들이 선택될 수 있거나, 및/또는 M으로 표시된, 현재 블록 위쪽의 선택된 원래의 복원된 이웃 샘플들의 수는 blkW에 의존할 수 있다.
- [0784] a. 한 예에서, 현재 블록 위쪽의 k번째 선택된 샘플들은 위치( $blkX + (k+1)*blkW/M-1, blkY-1$ )에 있을 수 있고, 여기서, k는 0에서 M-1이다.
- [0785] b. 예를 들어,  $blkW \leq 8$ 이면,  $M = 4$ 이다.
- [0786] c. 예를 들어,  $blkW > 8$ 이면,  $M = 8$ 이다.
- [0787] d. 대안으로서, blkW와 blkH 사이의 관계에 상관없이, 현재 블록 좌측의 모든 원래의 복원된 이웃 샘플이 선택될 수 있거나, 및/또는 현재 블록 위쪽의 M개의 원래의 복원된 이웃 샘플들이 선택될 수 있고, 여기서 M은 상기 규칙들에 의해 결정된다.
- [0788] 2) 예를 들어, blkW가 blkH보다 작은 경우, 현재 블록 위쪽의 모든 원래의 복원된 이웃 샘플들이 선택될 수 있거나, 및/또는 M으로 표시된, 현재 블록 좌측의 선택된 원래의 복원된 이웃 샘플들의 수는 blkH에 의존할 수 있다.
- [0789] a. 한 예에서, 현재 블록 좌측의 k번째 선택된 샘플들은 위치 ( $blkX-1, blkY+(k+1)*blkH/M-1$ )에 있을 수 있고, 여기서 k는 0에서 M-1이다.
- [0790] b. 예를 들어,  $blkH \leq 8$ 이면  $M = 4$ 이다.
- [0791] c. 예를 들어,  $blkH > 8$ 이면  $M = 8$ 이다.
- [0792] d. 대안으로서, blkW와 blkH 사이의 관계에 상관없이, 현재 블록 위쪽의 모든 원래의 복원된 이웃 샘플이 선택될 수 있거나, 및/또는 현재 블록 좌측의 M개의 원래의 복원된 이웃 샘플들이 선택될 수 있고, 여기서 M은 상기 규칙들에 의해 결정된다.
- [0793] c. 한 예에서, ALWIP 업샘플링에 이용되는 이웃 샘플들은 최종 예측 블록을 생성하는데 이용되기 전에 추가로 수정될 수 있다(예를 들어, 필터링될 수 있다, 여기서 필터는 N=2 또는 3 등의 N-탭 필터일 수 있음).
- [0794] i. 한 예에서, 이웃 샘플들 필터링 프로세스는 ALWIP 모드에 따라 적응적으로 적용될 수 있다.
- [0795] d. 최종 예측 블록을 생성하는 방법(예를 들어, 선형 보간)은 블록 치수/코딩된 정보(예를 들어, 인트라 예측 방향, 변환 유형 등)에 의존할 수 있다.
- [0796] 43. 한 예에서, 샘플들은 ALWIP의 업샘플링 프로세스의 상이한 필터링 스테이지들에서 상이한 정밀도들을 가질 수 있다. "샘플들"은 업샘플링 프로세스 전후의 예측 샘플들 또는 임의의 중간 샘플들을 지칭할 수 있다.
- [0797] a. 한 예에서, 샘플들은 제1 필터링 스테이지에서 수평으로 제1 치수를 따라 업샘플링되고; 그 다음, 샘플들은 ALWIP의 업샘플링 프로세스의 제2 필터링 스테이지에서 수직으로 제2 치수를 따라 업샘플링된다.
- [0798] i. 대안으로서, 샘플들은 제1 필터링 스테이지에서 수직으로 제1 치수를 따라 업샘플링되고; 그 다음, 샘플들은 ALWIP의 업샘플링 프로세스의 제2 필터링 스테이지에서 제2 치수를 따라 수평으로 업샘플링된다.
- [0799] b. 한 예에서, 제1 필터링 스테이지에서 우측 시프팅 또는 분할이 없는 출력 업샘플링 결과들은 제2 필터링

스테이지에 대한 입력 샘플로서 이용될 수 있다.

[0800] i. 한 예에서, 제2 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해  $Shift1$ 만큼 우측 시프트되거나  $Dem1$ 로 나누어질 수 있다.

[0801] ii. 한 예에서, 제1 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해  $Shift2$ 만큼 우측 시프트되거나  $Dem2$ 로 나누어질 수 있다.

[0802] 1) 한 예에서,  $Shift1=2 \times Shift2$ ;  $Dem1 = Dem2 \times Dem2$ .

[0803] iii. 한 예에서, 제2 필터링 스테이지에 입력되지만 제1 필터링 스테이지에서 출력된 업샘플링 결과들이 아닌 샘플들은, 제2 필터링 스테이지에 입력되기 전에  $Shift3$ 만큼 좌측 시프트되거나  $Dem3$ 으로 곱해질 수 있다.

[0804] 1) 한 예에서,  $Shift3 = Shift1$ ;  $Dem3 = Dem2$ .

[0805] c. 한 예에서, 제1 필터링 스테이지의 출력 업샘플링 결과들은 제2 필터링 스테이지에 대한 입력 샘플들로서 이용되기 전에  $Shift1$ 만큼 우측 시프트되거나  $Dem1$ 로 나누어질 수 있다.

[0806] i. 한 예에서, 제2 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해  $Shift2$ 만큼 우측으로 시프트되거나  $Dem2$ 로 나누어질 수 있고, 여기서  $Shift2$ 는  $Shift1$ 과 같지 않을 수 있다, 예를 들어  $Shift2 > Shift1$ ;  $Dem2$ 는  $Dem1$ 과 같지 않을 수 있다, 예를 들어,  $Dem2 > Dem1$ .

[0807] ii. 한 예에서, 제1 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해  $Shift3$ 만큼 우측으로 시프트되거나  $Dem3$ 으로 나누어질 수 있고, 여기서  $Shift3$ 은  $Shift1$ 과 같을 수 있다;  $Dem3$ 은  $Dem1$ 과 같지 않을 수 있다.

[0808] 1) 한 예에서,  $Shift3=Shift1+Shift2$ .

[0809] iii. 한 예에서, 제2 필터링 스테이지에 입력되지만 제1 필터링 스테이지에서 출력된 업샘플링 결과들이 아닌 샘플들은 제2 필터링 스테이지에 입력되기 전에 좌측 시프트되거나 인자에 의해 곱해질 수 있다.

[0810] d. 한 예에서, 제1 필터링 스테이지의 출력 업샘플링 결과들은 제2 필터링 스테이지에 대한 입력 샘플들로서 이용되기 전에  $Shift1$ 만큼 좌측 시프트되거나  $Dem1$ 에 의해 곱해질 수 있다.

[0811] i. 한 예에서, 제2 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해 우측 시프트되거나 인자에 의해 나누어질 수 있다.

[0812] ii. 한 예에서, 제1 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해 우측 시프트되거나 인자에 의해 나누어질 수 있다.

[0813] iii. 한 예에서, 제2 필터링 스테이지에 입력되지만 제1 필터링 스테이지에서 출력된 업샘플링 결과들이 아닌 샘플들은 제2 필터링 스테이지에 입력되기 전에  $Shift2$ 만큼 좌측 시프트되거나  $Dem2$ 에 의해 곱해질 수 있고, 여기서  $Shift2$ 는  $Shift1$ 과 같지 않을 수 있다, 예를 들어,  $Shift2 > Shift1$ ;  $Dem2$ 는  $Dem1$ 과 같지 않을 수 있다, 예를 들어,  $Dem2 > Dem1$ .

[0814] e. 한 예에서, 제1 필터링 스테이지에 입력된 샘플들은 제1 필터링 스테이지에 대한 입력 샘플들로서 이용되기 전에  $Shift1$ 만큼 좌측 시프트되거나  $Dem1$ 에 의해 곱해질 수 있다.

[0815] i. 한 예에서, 제2 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해 우측 시프트되거나 인자에 의해 나누어질 수 있다.

[0816] ii. 한 예에서, 제1 필터링 스테이지의 출력 업샘플링 필터링 결과들은 최종 업샘플링된 결과들을 도출하기 위해 우측 시프트되거나 인자에 의해 나누어질 수 있다.

[0817] iii. 한 예에서, 제2 필터링 스테이지에 입력되지만 제1 필터링 스테이지에서 출력된 업샘플링 결과들이 아닌 샘플들은 제2 필터링 스테이지에 입력되기 전에  $Shift2$ 만큼 좌측 시프트되거나  $Dem2$ 에 의해 곱해질 수 있고, 여기서  $Shift2$ 는  $Shift1$ 과 같지 않을 수 있다, 예를 들어,  $Shift2 > Shift1$ ;  $Dem2$ 는  $Dem1$ 과 같지 않을 수 있다, 예를 들어,  $Dem2 > Dem1$ .

[0818] 44. ALWIP에서의 업샘플링은 수직 업샘플링과 수평 업샘플링이 양쪽 모두가 요구되는 경우 고정된 순서로 수행될 수 있다는 것이 제안된다.

- [0819] a. 한 예에서, 수평 업샘플링이 먼저 수행되고, 수직 업샘플링이 2번째로 수행될 수 있다.
- [0820] b. 한 예에서, 수직 업샘플링이 먼저 수행되고, 수평 업샘플링이 2번째로 수행될 수 있다.
- [0821] 45. 한 예에서, 업샘플링 전의 ALWIP에서의 예측 샘플들은 블록 치수들에 따라 전치될 수 있다.
- [0822] a. 한 예에서,  $W \times H$  블록이 먼저  $H \times W$  블록으로 전치될 수 있고, 그 다음 업샘플링이 적용될 수 있다.
- [0823] b. 대안으로서, 또한, 업샘플링 프로세스 후, 업샘플링된 샘플들은 역전된 방식으로 전치될 수 있다.
- [0824] 46. ALWIP에서 업샘플링을 위해 쌍선형 필터 대신에 대안적인 보간 필터들이 이용될 수 있다는 것이 제안된다.
- [0825] a. 한 예에서, (4-tap, 6-tap, 8-tap 등) 가우시안 필터가 이용될 수 있다.
- [0826] b. 한 예에서, (4-tap, 6-tap, 8-tap 등) 큐빅 필터가 이용될 수 있다.
- [0827] c. 한 예에서, 크로마 샘플들에 대한 모션 보상에 이용되는 보간 필터들이 이용될 수 있다.
- [0828] d. 한 예에서, 루마 샘플들에 대한 모션 보상에 이용되는 보간 필터들(6-탭, 8-탭 등)이 이용될 수 있다.
- [0829] e. 어떤 보간 필터가 이용되는지는 블록 치수에 의존할 수 있다.
- [0830] f. 어떤 보간 필터가 이용되는지는 업샘플링 비율에 의존할 수 있다.
- [0831] g. 어떤 보간 필터가 이용되는지는 ALWIP의 예측 모드에 의존할 수 있다.
- [0832] h. 어떤 보간 필터가 이용되는지는 얼마나 많은 샘플이 업샘플링에 이용가능한지에 의존할 수 있다.
- [0833] i. 예를 들어, 하나의 행(또는 열)에 4개의 가용 샘플이 있는 경우(이웃 기준 샘플들은 제외), 4탭 보간 필터가 이용될 수 있다.
- [0834] ii. 예를 들어, 하나의 행(또는 열)에 8개의 가용 샘플이 있는 경우(이웃 기준 샘플들은 제외), 4탭 또는 8탭 보간 필터가 이용될 수 있다.
- [0835] 47. ALWIP 모드로 코딩된 블록에 대한 MPM들은 이웃 블록들과는 독립적으로 구성될 수 있는 것이 제안된다.
- [0836] a. 한 예에서, ALWIP 모드로 코딩된 블록에 대한 MPM은 미리정의된다.
- [0837] i. 예를 들어, 미리정의된 MPM들은 모든 블록 치수에 대해 {M0, M1, M2}이다. 예를 들어, M0=0, M1=1 및 M2=2.
- [0838] ii. 예를 들어, ALWIP 모드로 코딩된 블록에 대해 미리정의된 MPM들은 블록 치수들에 의존할 수 있다. 예를 들어, ALWIP 모드로 코딩된 블록에 대해 미리정의된 MPM들은 섹션 2.2.1.5에 정의된 sizeId에 의존할 수 있다.
- [0839] 1) 예를 들어, sizeId가 0일 때 MPM들은 순서대로 {17, 34, 5}이다.
- [0840] 2) 예를 들어, sizeId가 1일 때 MPM들은 순서대로 {0, 7, 16}이다.
- [0841] 3) 예를 들어, sizeId가 2일 때 MPM들은 순서대로 {1, 4, 6}이다.
- [0842] iii. 예를 들어, ALWIP 모드로 코딩된 블록에 대해 미리정의된 MPM들은 처음 K개 행렬들과 연관되도록 고정된다.
- [0843] 1) 한 예에서, K = 3이다.
- [0844] 2) 한 예에서, K는 블록 치수들에 의존할 수 있다.
- [0845] 48. 블록 치수들에 관계없이 ALWIP에 이용되는 행렬들의 수를 단일화할 것이 제안된다.
- [0846] a. 한 예에서, S개의 행렬이 모든 블록 치수에 대해 저장될 수 있다. 예를 들어, S=11, S=19, 또는 S=35.
- [0847] b. 한 예에서, ALWIP 모드 시그널링은 모든 블록 치수에 대해 동일한 방식으로 이루어진다.
- [0848] c. 대안으로서, 또한, 행렬들은 상이한 블록 치수에 대해 상이할 수 있다.
- [0849] 5. 실시예들

[0850] 새로 추가된 부분은 굵은 기울임꼴로 강조되고 삭제된 부분들은 밑줄이 그어진 기울임꼴 텍스트로 강조된다.

### 5.1 한 예

[0852] ALWIP 플래그를 코딩하기 위해 3개의 컨텍스트가 이용된다.

[0853] 표 9-15 – 컨텍스트 코딩된 빈들을 갖는 신택스 요소들로의 ctxInc의 할당

선택스 요소	<i>binIdx</i>					
	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>&gt;= 5</i>
...	<i>terminate</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>

[0854]

표 9-15 – 컨텍스트 코딩된 빈들을 갖는 신택스 요소들로의 ctxInc의 할당

선택스 요소	<i>binIdx</i>					
	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>&gt;= 5</i>
<i>intra_lwip_flag[ ][ ]</i>	<i>(Abs( Log2(cbWid th) – Log2(cbHeight) ) ≥ 1) ? 3 : ( 0,1,2 (clause 9.5.4.2.2) )</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_flag[ ][ ]</i>	<i>( 0,1,2 (clause 9.5.4.2.2) )</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_mpm_flag[ ][ ]</i>	<i>0</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_mpm_idx[ ][ ]</i>	<i>bypass</i>	<i>bypass</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_mpm_remainer[ ][ ]</i>	<i>bypass</i>	<i>bypass</i>	<i>bypass</i>	<i>bypass</i>	<i>bypass</i>	<i>na</i>

[0856]

### 5.2 한 예

[0858] ALWIP 플래그를 코딩하기 위해 하나의 고정 컨텍스트가 이용된다.

[0859] 표 9-15 – 컨텍스트 코딩된 빈들을 갖는 신택스 요소들로의 ctxInc의 할당

선택스 요소	<i>binIdx</i>					
	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>&gt;= 5</i>
...	<i>terminate</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>

[0860]

표 9-15 – 컨텍스트 코딩된 빙들을 갖는 십екс 요소들로의 ctxInc의 할당

십екс 요소	binIdx					
	0	1	2	3	4	>= 5
<i>intra_lwip_flag[ ][ ]</i>	$(Abs(Log2(cbWid) - Log2(cbHeight)) > 1) ? 3 : (0,1,2)$ <i>(clause 9.5.4.2.2)</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_flag[ ][ ]</i>	0	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_mpm_flag[ ][ ]</i>	0	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_mpm_idx[ ][ ]</i>	bypass	bypass	<i>na</i>	<i>na</i>	<i>na</i>	<i>na</i>
<i>intra_lwip_mpm_remainer[ ][ ]</i>	bypass	bypass	bypass	bypass	bypass	<i>na</i>

[0862]

### 5.3 한 예

경계 감축 프로세스를 한-단계로 수행한다.

아래 실시예들은 채택된 JVET-N0220-proposal-test-CE3-4.1\_v2에 기초한다.

8.4.4.2.X1 어파인 선형 가중된 인트라 샘플 예측

#### 8.4.4.2.X3 경계 감축 프로세스의 명세

이 프로세스에 대한 입력들은 다음과 같다:

- 변환 블록 크기를 명시하는 변수 nTbX,

-  $x = 0..nTbX-1$ 인 기준 샘플들 refX[ x ],

- 다운샘플링된 경계 크기를 명시하는 변수 boundarySize,

- 업샘플링에 대해 중간 경계 샘플들이 요구되는지를 명시하는 플래그 needUpsBdryX,

- 업샘플링에 대한 경계 크기를 명시하는 변수 upsBdrySize.

이 프로세스의 출력들은  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들 redX[ x ] 및  $x = 0..upsBdrySize-1$ 인 업샘플링 경계 샘플들 upsBdryX[ x ]이다.

$x = 0..upsBdrySize-1$ 인 업샘플링 경계 샘플들 upsBdryX[ x ]는 다음과 같이 도출된다:

- needUpsBdryX가 TRUE이고 upsBdrySize가 nTbX보다 작다면, 다음이 적용된다:

$uDwn = nTbX / upsBdrySize$  (8-X30)

$$upsBdryX[x] = refX[x * uDwn]$$

[0878]

$$\text{upsBdryX}[x] = (\sum_{i=0}^{uDwn-1} \text{refX}[x * uDwn + i] + \\ (1 << (\text{Log2}(uDwn) - 1))) >> \text{Log2}(uDwn) \quad (8-X31)$$

[0879] - 그렇지 않으면(upsBdrySize가 nTbX와 같음), upsBdryX[ x ]는 refX[ x ]로 설정된다.

[0880]  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들 redX[ x ]는 다음과 같이 도출된다:

[0881] - boundarySize가 upsBdrySize nTbX보다 작다면, 다음이 적용된다:

$$\text{bDwn} = \frac{\text{upsBdrySize}}{\text{nTbX}} \text{nTbX} / \text{boundarySize} \quad (8-X32)$$

$$\text{redX}[x] = (\sum_{i=0}^{\text{bDwn}-1} \text{upsBdryX} \text{refX}[x * \text{bDwn} + i] + \\ (1 << (\text{Log2}(\text{bDwn}) - 1))) >> \text{Log2}(\text{bDwn}) \quad (8-X33)$$

[0882] 방정식 8-X33에서 항 upsBdryX가 삭제되었다.

[0883] - 그렇지 않으면(boundarySize가 upsBdrySize nTbX와 동일) redX[ x ]가 upsBdryX[x] refX[x]로 설정된다.

#### 5.4 한 예

[0884] ALWIP의 업샘플링 프로세스에서 상이한 필터링 스테이지들에서 상이한 정밀도들로 예측 샘플들을 도출한다.

[0885] 아래 실시예들은 채택된 JVET-N0217-proposal-test-CE3-4.1\_v2에 기초한다.

##### 8.4.4.2.X4 예측 업샘플링 프로세스의 명세

[0886] 이 프로세스에 대한 입력들은 다음과 같다:

[0887] - 입력 블록 폭을 명시하는 변수 predW,

[0888] - 입력 블록 높이를 명시하는 변수 predH,

[0889] -  $x = 0..predW-1$ ,  $y = 0..predH-1$ 인 어파인 선형 가중된 샘플들 predLwip[ x ][ y ],

[0890] - 변환 블록 폭을 명시하는 변수 nTbW,

[0891] - 변환 블록 높이를 명시하는 변수 nTbH,

[0892] - 업샘플링 경계 폭을 명시하는 변수 upsBdryW,

[0893] - 업샘플링 경계 높이를 명시하는 변수 upsBdryH,

[0894] -  $x = 0..upsBdryW-1$ 인 상단 업샘플링 경계 샘플들 upsBdryT[ x ],

[0895] -  $x = 0..upsBdryH-1$ 인 좌측 업샘플링 경계 샘플 upsBdryL[ x ].

[0896] 이 프로세스의 출력들은  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들 predSamples[ x ][ y ]이다.

[0897] 저밀도 예측된 샘플들 predSamples[ m ][ n ]는 다음과 같이  $x = 0..predW-1$ ,  $y = 0..predH-1$ 인 predLwip[ x ][ y ]로부터 도출된다:

[0898] upHor = nTbW / predW (8-X34)

[0899] upVer = nTbH / predH (8-X35)

$$\text{predSamples}[ (x + 1) * \text{upHor} - 1 ][ (y + 1) * \text{upVer} - 1 ] = \text{predLwip}[x][y] \quad (8-X36)$$

[0900]  $x = 0..upsBdryW-1$ 인 상단 경계 샘플들 upsBdryT[ x ]는 다음과 같이 predSamples[ m ][ -1 ]에 할당된다:

$$\text{predSamples}[ (x + 1) * (nTbW / upsBdryW) - 1 ][ -1 ] = \text{upsBdryT}[x] \quad (8-X37)$$

[0901]  $y = 0..upsBdryH-1$ 인 좌측 경계 샘플들 upsBdryL[ y ]은 다음과 같이 predSamples[ -1 ][ n ]에 할당된다:

[0909]  $\text{predSamples}[-1][(\text{y} + 1) * (\text{nTbH} / \text{upsBdryH}) - 1] = \text{upsBdryL}[\text{y}]$  (8-X38)

[0910]  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들  $\text{predSamples}[x][y]$ 는 다음과 같이 도출된다:

[0911] -  $nTbH$ 가  $nTbW$ 보다 크다면, 다음과 같은 순서화된 단계들이 적용된다:

[0912] 1.  $upHor$ 가 1보다 크면,  $m = 0..predW-1$ ,  $n = 1..predH$ 인 모든 저밀도 위치들 ( $xHor$ ,  $yHor$ ) = ( $m * upHor - 1$ ,  $n * upVer - 1$ )에 대한 수평 업샘플링이 다음과 같이  $dX = 1..upHor-1$ 과 함께 적용된다:

$\text{predSamples}[xHor + dX][yHor] = ((upHor - dX) * \text{predSamples}[xHor][yHor]) +$

[0913]  $dX * \text{predSamples}[xHor + upHor][yHor]) / upHor$  (8-X39)

[0914] 2.  $m = 0..nTbW-1$ ,  $n = 0..predH-1$ 인 모든 저밀도 위치들 ( $xVer$ ,  $yVer$ ) = ( $m, n * upVer-1$ )에 대한 수직 업샘플링은 다음과 같이  $dY = 1..upVer-1$ 과 함께 적용된다:

[0915]  $yVer$ 가 -1이면  $\text{predSamples}[xVer][yVer] = \text{predSamples}[xVer][yVer] \ll \log2(upHor)$

$\text{predSamples}[xVer][yVer + dY] = ((upVer - dY) * \text{predSamples}[xVer][yVer]) +$   
 $dY * \text{predSamples}[xVer][yVer + upVer]) / upVer + (1 \ll (\log2(upHor) + \log2(upVer) - 1)) \gg$   
 $(\log2(upHor) + \log2(upVer))$

[0916] (8-X40)

[0917] - 그렇지 않으면, 다음과 같은 순서화된 단계들이 적용된다:

[0918] 1.  $upVer$ 가 1보다 크면,  $m = 1..predW$ ,  $n = 0..predH-1$ 인 모든 저밀도 위치들 ( $xVer$ ,  $yVer$ ) = ( $m * upHor - 1$ ,  $n * upVer - 1$ )에 대한 수직 업샘플링이 (8-X40) (8-X41)에 명시된 대로  $dY = 1..upVer-1$ 과 함께 적용된다.

$\text{predSamples}[xVer][yVer + dY] = ((upVer - dY) * \text{predSamples}[xVer][yVer]) +$

[0919]  $dY * \text{predSamples}[xVer][yVer + upVer])$  (8-X41)

[0920] 2.  $m = 0..predW-1$ ,  $n = 0..nTbH-1$ 인 모든 저밀도 위치들 ( $xHor$ ,  $yHor$ ) = ( $m * upHor-1$ ,  $n$ )에 대한 수평 업샘플링은 (8-X39)에 명시된 대로  $dX = 1..upHor-1$ 과 함께 다음과 같이 적용된다.

[0921] If  $xHor$  is equal to -1,  $\text{predSamples}[xHor][yHor] = \text{predSamples}[xHor][yHor] \ll \log2(upVer)$

[0922]  $\text{predSamples}[xHor + dX][yHor] = ((upHor - dX) * \text{predSamples}[xHor][yHor]) +$   
 $dX * \text{predSamples}[xHor + upHor][yHor] + (1 \ll (\log2(upHor) + \log2(upVer) - 1)) \gg (\log2(upHor) +$

[0923]  $\log2(upVer))$  (8-X42)

## 5.5 글머리 기호 40에 대응하는 예들

[0925] 블록 치수들이  $W \times H$ 라고 가정한다.  $x = Sx, Sx+Kx, Sx+2Kx, Sx+3Kx, \dots, y = Sy, Sy+Ky, Sy+2Ky, Sy+3Ky \dots$ 인 샘플들  $P(x, y)$ 가 업샘플링 프로세스에 입력되어  $x=0, 1, 2 \dots W-1, y=0, 1, 2, \dots H-1$ 인 업샘플링된 샘플들  $S(x, y)$ 를 도출한다.  $Kx$  및  $Ky$ 는 각각 수평 및 수직 방향을 따른 단계 크기들이다.  $(Sx, Sy)$ 는 시작 위치이다.

[0926] 제1 스테이지에서 1-D 업샘플링이 수평으로 수행되고 제2 스테이지에서 1-D 업샘플링이 수직으로 수행된다고 가정한다.

[0927] 한 예에서, 우측 시프팅이 없는 제1 스테이지의 출력 결과들은 다음과 같이 도출될 수 있다

[0928]  $S' (Sx+Kx-1, Sy) = F1 * P(Sx, Sy) + F2 * P(Sx+Kx, Sy).$

[0929]  $S' (Sx+Kx-1, Sy+Ky) = F1 * P(Sx, Sy+Ky) + F2 * P(Sx+Kx, Sy+Ky).$

[0930]  $F1, F2$ 는 2-탭 필터에 대한 계수들이고  $F1+F2 = 2^N$ 이다.

[0931] 그 다음 제2 스테이지에서의 출력 결과는 다음과 같이 도출될 수 있다

[0932]  $S' (Sx+Kx-1, Sy+1) = F3 * S' (Sx+Kx-1, Sy) + F4 * S' (Sx+Kx-1, Sy+Ky).$

[0933] F3, F4는 2탭 필터에 대한 계수들이며  $F3+F4 = 2^N$ 이다.

[0934] 그 다음, 최종 업샘플링된 샘플 값은 다음과 같이 도출될 수 있다:

[0935]  $S(Sx+Kx-1, Sy+1) = \text{Shift}(S' (Sx+Kx-1, Sy+1), 2N);$

[0936]  $S(Sx+Kx-1, Sy) = \text{Shift}(S' (Sx+Kx-1, Sy), N);$

[0937]  $S(Sx+Kx-1, Sy+Ky) = \text{Shift}(S' (Sx+Kx-1, Sy+Ky), N);$

## 5.6 한 예

[0939] 한 스테이지에서 감축된 경계 샘플들을 도출하고 업샘플링을 위한 기준 베피를 생성한다.

[0940] 아래 실시예들은 채택된 JVET-N0217-proposal-test-CE3-4.1\_v2에 기초한다.

[0941] 8.4.4.2.X1 어파인 선형 가중된 인트라 샘플 예측

### 8.4.4.2.X3 경계 감축 프로세스의 명세

[0943] ...

[0944]  $x = 0 \dots \text{upsBdrySize}-1$ 인 업샘플링 경계 샘플들  $\text{upsBdryX}[x]$ 는 다음과 같이 도출된다:

[0945] -  $\text{needUpsBdryX}$ 가 TRUE이고  $\text{upsBdrySize}$ 가  $nTbX$ 보다 작다면, 다음이 적용된다:

[0946]  $uDwn = nTbX / \text{upsBdrySize}$  (8-X30)

$$\text{upsBdryX}[x] = (\sum_{i=0}^{uDwn-1} \text{refX}[x * uDwn + i] + (1 << (\text{Log2}(uDwn) - 1))) >> \text{Log2}(uDwn)$$

$$\text{upsBdryX}[x] = \text{refX}[(x + 1) * uDwn - 1]$$

(8-X31)

- 그렇지 않으면( $\text{upsBdrySize}$ 가  $nTbX$ 와 같음),  $\text{upsBdryX}[x]$ 는  $\text{refX}[x]$ 로 설정된다.

[0949]  $x = 0 \dots \text{boundarySize}-1$ 인 감축된 경계 샘플들  $\text{redX}[x]$ 는 다음과 같이 도출된다:

[0951] -  $\text{boundarySize}$ 가  $\text{upsBdrySize}$   $nTbX$ 보다 작다면, 다음이 적용된다:

[0952]  $bDwn = \text{upsBdrySize} nTbX / \text{boundarySize}$  (8-X32)

$$\text{redX}[x] = (\sum_{i=0}^{bDwn-1} \text{upsBdryX} \text{refX}[x * bDwn + i] + (1 << (\text{Log2}(bDwn) - 1))) >> \text{Log2}(bDwn)$$

[0953] 방정식 8-X33에서 항  $\text{upsBdryX}$ 가 삭제되었다.

[0955] - 그렇지 않으면( $\text{boundarySize}$ 가  $\text{upsBdrySize}$   $nTbX$ 와 같음)  $\text{redX}[x]$ 가  $\text{upsBdryX} \text{refX}[x]$ 로 설정된다.

## 5.7 한 예

[0957] ALWIP(일명, 행렬 기반의 인트라 예측 또는 MIP)의 고정 차수 업샘플링에 대한 예들이 여기에 제시된다. 텍스트는 JVET-N1001-v6에 기초한다.

### 5.7.1 먼저 수평 업샘플링, 그 다음, 수직 업샘플링

#### 8.4.5.2.1 행렬 기반의 인트라 샘플 예측

[0959] i) 프로세스에 대한 입력들은 다음과 같다:

[0961] - 현재 빙처의 상단 좌측 샘플에 관해 현재 변환 블록의 상단 좌측 샘플을 명시하는 샘플 위치 ( $xTbCmp$ ,  $yTbCmp$ ),

[0962] - 인트라 예측 모드를 명시하는 변수  $\text{predModeIntra}$ ,

[0963] - 변환 블록 폭을 명시하는 변수  $nTbW$ ,

[0964] - 변환 블록 높이를 명시하는 변수 nTbH.

[0965] 이 프로세스의 출력들은  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들 predSamples[ x ][ y ]이다.

[0966] 변수들 numModes, boundarySize, predW, predH 및 predC는 표 8-7에 명시된 MipSizeId[ xTbCmp ][ yTbCmp ]를 이용하여 도출된다.

[0967] 표 8-7 – 예측 모드들 numModes, 경계 크기 boundarySize, 및 MipSizeId를 이용한 예측 크기들 predW, predH 및 predC의 수의 명세

MipSizeId	numModes	boundarySize	predW	predH	predC
0	35	2	4	4	4
1	19	4	4	4	4
2	11	4	Min( nTbW, 8 )	Min( nTbH, 8 )	8

[0968]

플래그 isTransposed는 다음과 같이 도출된다:

isTransposed = ( predModeIntra > ( numModes / 2 ) ) ? TRUE : FALSE (8-56)

플래그들 needUpsBdryHor 및 needUpsBdryVer는 다음과 같이 도출된다:

needUpsBdryHor = ( nTbW > predW ) ? TRUE : FALSE (8-57)

needUpsBdryVer = ( nTbH > predH ) ? TRUE : FALSE (8-58)

변수들 upsBdryW 및 upsBdryH는 다음과 같이 도출된다:

upsBdryW = ( nTbH > nTbW ) ? nTbW: predW (8-59)

upsBdryH = ( nTbH > nTbW ) ? predH: nTbH (8-60)

upsBdryW = nTbW(8-59)

upsBdryH = predH(8-60)

변수들 mipW 및 mipH는 다음과 같이 도출된다:

mipW = isTransposed ? predH : predW (8-61)

mipH = isTransposed ? predW : predH (8-62)

x = 0..nTbW-1인 기준 샘플들 refT[ x ] 및 y = 0..nTbH-1인 refL[ y ]의 생성의 경우, 조항 8.4.5.2.2에 명시된 MIP 기준 샘플 도출 프로세스는, 샘플 위치(xTbCmp, yTbCmp), 변환 블록 폭 nTbW, 변환 블록 높이 nTbH를 입력들로서 취하여 기동되고, x = 0..nTbW - 1인 상단 및 좌측 기준 샘플들 refT[x](x=0..nTbW-1 및 y = 0..nTbH - 1인 refL[ y ])를 각각 출력들로서 취한다.

x = 0..2 \* boundarySize-1인 경계 샘플들 p[ x ]의 생성의 경우, 다음이 적용된다:

- 조항 8.4.5.2.3에 명시된 MIP 경계 다운샘플링 프로세스는, 상단 기준 샘플들에 대해, 블록 크기 nTbW, x = 0..nTbW - 1인 기준 샘플들 refT[ x ], 경계 크기 boundarySize, 업샘플링 경계 플래그 needUpsBdryHor, 및 업샘플링 경계 크기 upsBdryW를 입력들로서 취하여 기동되고, x = 0..boundarySize-1인 감축된 경계 샘플들 refT[ x ]와 x = 0..upsBdryW - 1인 업샘플링 경계 샘플들 upsBdryT[ x ]을 출력들로서 취한다.

- 조항 8.4.5.2.3에 명시된 MIP 경계 다운샘플링 프로세스는, 좌측 기준 샘플들에 대해, 블록 크기 nTbH, y = 0..nTbH - 1인 기준 샘플들 refL[ y ], 경계 크기 boundarySize, 업샘플링 경계 플래그 needUpsBdryVer, 및 업샘플링 경계 크기 upsBdryH를 입력들로서 취하여 기동되고, x= 0..boundarySize-1인 감축된 경계 샘플들 refL[ x ] 및 x = 0..upsBdryH-1인 업샘플링 경계 샘플들 upsBdryL[ x ]을 출력들로서 취한다.

- 감축된 상단 및 좌측 경계 샘플들 redT 및 redL은 다음과 같이 경계 샘플 어레이 p에 할당된다:

- isTransposed가 1이라면, p[ x ]는 x = 0..boundarySize-1인 redL[ x ]로 설정되고, p[ x + boundarySize ]는 x = 0..boundarySize - 1인 redT[ x ]로 설정된다.

- [0988] - 그렇지 않으면,  $p[x]$ 는  $x = 0..boundarySize-1$ 인  $redT[x]$ 로 설정되고  $p[x + boundarySize]$ 는  $x = 0..boundarySize-1$ 인  $redL[x]$ 로 설정된다.
- [0989] predModeIntra에 따른 인트라 샘플 예측 프로세스의 경우, 다음과 같은 순서화된 단계들이 적용된다:
- [0990] 3.  $x = 0..mipW-1$ ,  $y = 0..mipH-1$ 인 행렬 기반의 인트라 예측 샘플들 predMip[ $x$ ][ $y$ ]는 다음과 같이 도출된다:
- [0991] - 변수 modeId는 다음과 같이 도출된다:
- [0992]  $modeId = predModeIntra - (isTransposed ? numModes / 2 : 0)$  (8-63)
- [0993] -  $x = 0..2 * boundarySize-1$ ,  $y = 0..predC * predC-1$ 인 가중치 행렬 mWeight[ $x$ ][ $y$ ]는 표 8-XX에 명시된 MipSizeId[ $xTbCmp$ ][ $yTbCmp$ ] 및 modeId를 이용하여 도출된다[Ed. (BB): 10비트가 아닌 가중치 솔루션이 채택되면 가중치 행렬들을 추가]
- [0994] -  $y=0..predC * predC-1$ 인 편향 벡터 vBias[ $y$ ]는 표 8-XX에 명시된 sizeId 및 modeId를 이용하여 도출된다[Ed. (BB): 10비트가 아닌 가중치 솔루션이 채택되면 편향 벡터들을 추가].
- [0995] - 변수 sW는 표 8-8에 명시된 대로 MipSizeId[ $xTbCmp$ ][ $yTbCmp$ ] 및 modeId를 이용하여 도출된다.
- [0996] -  $x = 0..mipW-1$ ,  $y = 0..mipH-1$ 인 행렬 기반의 인트라 예측 샘플들 predMip[ $x$ ][ $y$ ]는 다음과 같이 도출된다:
- [0997]  $oW = 1 \ll (sW - 1)$  (8-64)
- [0998]  $sB = BitDepthY - 1$  (8-65)
- [0999]  $incW = (predC > mipW) ? 2 : 1$  (8-66)
- [1000]  $incH = (predC > mipH) ? 2 : 1$  (8-67)
- $$\text{predMip}[x][y] = (\sum_{i=0}^{2 * boundarySize-1} mWeight[i][y * incH * predC + x * incW] * p[i]) + (vBias[y * incH * predC + x * incW] \ll sB) + oW$$
- [1001] . . . (8-68)
- [1002] 4. isTransposed가 TRUE이면,  $x = 0..predH-1$ ,  $y = 0..predW-1$ 인 predH x predW 어레이 predMip[ $x$ ][ $y$ ]가 다음과 같이 전치된다:
- [1003]  $\text{predTemp}[y][x] = \text{predMip}[x][y]$  (8-69)
- [1004]  $\text{predMip} = \text{predTemp}$  (8-70)
- [1005] 5.  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들 predSamples[ $x$ ][ $y$ ]는 다음과 같이 도출된다:
- [1006] - needUpsBdryVer가 TRUE이거나 needUpsBdryHor가 TRUE이면, 조항 8.4.5.2.4에 명시된 MIP 예측 업샘플링 프로세스는, 입력 블록 폭 predW, 입력 블록 높이 predH,  $x = 0..predW - 1$ ,  $y = 0..predH - 1$ 인 행렬 기반의 인트라 예측 샘플들 predMip[ $x$ ][ $y$ ], 변환 블록 폭 nTbW, 변환 블록 높이 nTbH, 업샘플링 경계 폭 upsBdryW, 업샘플링 경계 높이 upsBdryH, 상단 업샘플링 경계 샘플들 upsBdryT, 및 좌측 업샘플링 경계 샘플들 upsBdryL을 입력들로서 취하여 기동되고, 출력은 예측된 샘플 어레이 predSamples이다.
- [1007] - 그렇지 않으면,  $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ 인 predSamples[ $x$ ][ $y$ ]는 predMip[ $x$ ][ $y$ ]로 설정된다.
- [1008] 6.  $x = 0..nTbW-1 = 0..nTbH-1$ 인 예측된 샘플들 predSamples[ $x$ ][ $y$ ]는 다음과 같이 클리핑된다:
- [1009]  $\text{predSamples}[x][y] = \text{Clip}_Y(\text{predSamples}[x][y])$  (8-71)

표 8-8 – MipSizeId 및 modelId에 의존하는 가중치 시프트 Sw 명세

MipSizeId	modelId																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	8	9	8	8	8	8	9	8							
2	8	8	8	8	8	8											

## 8.4.5.2.4 MIP 예측 업샘플링 프로세스

이) 프로세스에 대한 입력들은 다음과 같다:

- 입력 블록 폭을 명시하는 변수 predW,

- 입력 블록 높이를 명시하는 변수 predH,

 $-x = 0..predW - 1, y = 0..predH - 1$ 인 행렬 기반의 인트라 예측된 샘플들 predMip[ x ][ y ],

- 변환 블록 폭을 명시하는 변수 nTbW,

- 변환 블록 높이를 명시하는 변수 nTbH,

- 업샘플링 경계 폭을 명시하는 변수 upsBdryW,

- 업샘플링 경계 높이를 명시하는 변수 upsBdryH,

 $-x = 0..upsBdryW - 1$ 인 상단 업샘플링 경계 샘플들 upsBdryT[ x ], $-x = 0..upsBdryH - 1$ 인 좌측 업샘플링 경계 샘플들 upsBdryL[ x ].이) 프로세스의 출력들은  $x = 0..nTbW - 1, y = 0..nTbH - 1$ 인 예측된 샘플들 predSamples[ x ][ y ]이다.저밀도 예측된 샘플들 predSamples[ m ][ n ]는 다음과 같이  $x = 0..predW - 1, y = 0..predH - 1$ 인 predMip[ x ][ y ]로부터 도출된다:

$$upHor = nTbW / predW \quad (8-78)$$

$$upVer = nTbH / predH \quad (8-79)$$

$$predSamples[ (x + 1) * upHor - 1 ][ (y + 1) * upVer - 1 ] = predMip[ x ][ y ] \quad (8-80)$$

 $x = 0..upsBdryW - 1$ 인 상단 경계 샘플들 upsBdryT[ x ]는 다음과 같이 predSamples[ m ][ -1 ]에 할당된다:

$$predSamples[ (x + 1) * (nTbW / upsBdryW) - 1 ][ -1 ] = upsBdryT[ x ] \quad (8-81)$$

 $y = 0..upsBdryH - 1$ 인 좌측 경계 샘플들 upsBdryL[ y ]은 다음과 같이 predSamples[ -1 ][ n ]에 할당된다:

$$predSamples[ -1 ][ (y + 1) * (nTbH / upsBdryH) - 1 ] = upsBdryL[ y ] \quad (8-82)$$

 $x = 0..nTbW - 1, y = 0..nTbH - 1$ 인 예측된 샘플들 predSamples[ x ][ y ]는 다음과 같이 도출된다:-  $nTbH > nTbW$ 보다 크다면, 다음과 같은 순서화된 단계들이 적용된다:

1.  $upHor$ 가 1보다 크면,  $m = 0..predW - 1, n = 1..predH - 1$ 인 모든 저밀도 위치들 ( $xHor, yHor$ ) = ( $m * upHor - 1, n * upVer - 1$ )에 대한 수평 업샘플링이 다음과 같이  $dX = 1..upHor - 1$ 와 함께 적용된다:

$$\begin{aligned} sum &= (upHor - dX) * predSamples[ xHor ][ yHor ] + \\ & dX * predSamples[ xHor + upHor ][ yHor ] \end{aligned} \quad (8-83)$$

$$predSamples[ xHor + dX ][ yHor ] = (sum + upHor / 2 - (sum < 0 ? 1 : 0)) / upHor \quad (8-84)$$

2.  $m = 0..nTbW - 1, n = 0..predH - 1$ 인 모든 저밀도 위치들 ( $xVer, yVer$ ) = ( $m, n * upVer - 1$ )에 대한 수직

업샘플링은 다음과 같이  $dY = 1..upVer-1$ 과 함께 적용된다:

```
sum = ( upVer - dY ) * predSamples[ xVer ][ yVer ] +
dY * predSamples[ xVer ][ yVer + upVer ](8-85)
```

[1038]  $\text{predSamples}[ \text{xVer} ][ \text{yVer} + dY ] = ( \text{sum} + \text{upVer}/2 - ( \text{sum} < 0 ? 1 : 0 ) ) / \text{upVer}$  (8-86)

[1040] - 그렇지 않으면, 다음과 같은 순서화된 단계들이 적용된다:

[1041] 1.  $upVer \geq 1$ 보다 크면,  $m = 1..predW$ ,  $n = 0..predH-1$ 인 모든 저밀도 위치들 ( $xVer$ ,  $yVer$ ) = ( $m * upHor - 1$ ,  $n * upVer - 1$ )에 대한 수직 업샘플링이 다음과 같이  $dY = 1..upVer-1$ 과 함께 적용된다:

$sum = (upVer - dY) * predSamples[ xVer ][ yVer ] +$

$dY * predSamples[ xVer ][ yVer + upVer ](8-87)$

[1043]  $\text{predSamples}[ \text{xVer} ][ \text{yVer} + dY ] = ( \text{sum} + \text{upVer}/2 - ( \text{sum} < 0 ? 1 : 0 ) ) / \text{upVer}$  (8-88)

[1044] 2.  $m = 0..predW-1$ ,  $n = 0..nTbH-1$ 인 모든 저밀도 위치들 ( $xHor$ ,  $yHor$ ) = ( $m * upHor-1$ ,  $n$ )에 대한 수평 업샘플링은 다음과 같이  $dX = 1..upHor-1$ 과 함께 적용된다:

$sum = (upHor - dX) * predSamples[ xHor ][ yHor ] +$

$dX * predSamples[ xHor + upHor ][ yHor ]$  (8-89)

[1046]  $\text{predSamples}[ \text{xHor} + dX ][ \text{yHor} ] = ( \text{sum} + \text{upHor}/2 - ( \text{sum} < 0 ? 1 : 0 ) ) / \text{upHor}$  (8-90)

## 5.7.2 먼저 수직 업샘플링, 그 다음 수평 업샘플링

### 8.4.5.2.1 행렬 기반의 인트라 샘플 예측

[1049] 이) 프로세스에 대한 입력들은 다음과 같다:

[1050] - 현재 픽처의 상단 좌측 샘플에 관해 현재 변환 블록의 상단 좌측 샘플을 명시하는 샘플 위치 ( $xTbCmp$ ,  $yTbCmp$ ),

[1051] - 인트라 예측 모드를 명시하는 변수  $\text{predModeIntra}$ ,

[1052] - 변환 블록 폭을 명시하는 변수  $nTbW$ ,

[1053] - 변환 블록 높이를 명시하는 변수  $nTbH$ .

[1054] 이) 프로세스의 출력들은  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들  $\text{predSamples}[ x ][ y ]$ 이다.

[1055] 변수들  $\text{numModes}$ ,  $\text{boundarySize}$ ,  $\text{predW}$ ,  $\text{predH}$  및  $\text{predC}$ 는 표 8-7에 명시된  $\text{MipSizeId}[ \text{xTbCmp} ][ \text{yTbCmp} ]$ 를 이용하여 도출된다.

[1056] 표 8-7 – 예측 모드들  $\text{numModes}$ , 경계 크기  $\text{boundarySize}$ , 및  $\text{MipSizeId}$ 를 이용한 예측 크기를  $\text{predW}$ ,  $\text{predH}$  및  $\text{predC}$ 의 수의 명세

MipSizeId	numModes	boundarySize	predW	predH	predC
0	35	2	4	4	4
1	19	4	4	4	4
2	11	4	Min( nTbW, 8 )	Min( nTbH, 8 )	8

[1058] 플래그  $\text{isTransposed}$ 는 다음과 같이 도출된다:

[1059]  $\text{isTransposed} = ( \text{predModeIntra} > ( \text{numModes} / 2 ) ) ? \text{TRUE} : \text{FALSE}$  (8-56)

[1060] 플래그들  $\text{needUpsBdryHor}$  및  $\text{needUpsBdryVer}$ 는 다음과 같이 도출된다:

[1061]  $\text{needUpsBdryHor} = ( \text{nTbW} > \text{predW} ) ? \text{TRUE} : \text{FALSE}$  (8-57)

- [1062] needUpsBdryVer = ( nTbH > predH ) ? TRUE : FALSE (8-58)
- [1063] 변수들 upsBdryW 및 upsBdryH는 다음과 같이 도출된다:
- [1064]  $upsBdryW = ( nTbH > nTbW ) ? nTbW : predW(8-59)$
- [1065]  $upsBdryH = ( nTbH > nTbW ) ? predH : nTbH(8-60)$
- [1066] upsBdryW = predW (8-59)
- [1067] upsBdryH = nTbH (8-60)
- [1068] 변수들 mipW 및 mipH는 다음과 같이 도출된다:
- [1069] mipW = isTransposed ? predH : predW (8-61)
- [1070] mipH = isTransposed ? predW : predH (8-62)
- [1071]  $x = 0..nTbW-1$ 인 기준 샘플들 refT[ x ] 및  $y = 0..nTbH-1$ 인 refL[ y ]의 생성의 경우, 조항 8.4.5.2.2에 명시된 MIP 기준 샘플 도출 프로세스는, 샘플 위치(xTbCmp, yTbCmp), 변환 블록 폭 nTbW, 변환 블록 높이 nTbH를 입력들로서 취하여 기동되고,  $x = 0..nTbW - 1$ 인 상단 및 좌측 기준 샘플들 refT[x]( $x=0..nTbW-1$  및  $y = 0..nTbH - 1$ ) refL[ y ]를 각각 출력들로서 취한다.
- [1072]  $x = 0..2 * boundarySize-1$ 인 경계 샘플들 p[ x ]의 생성의 경우, 다음과 같이 적용된다:
- 조항 8.4.5.2.3에 명시된 MIP 경계 다운샘플링 프로세스는, 상단 기준 샘플들에 대해, 블록 크기 nTbW,  $x = 0..nTbW - 1$ 인 기준 샘플들 refT[ x ], 경계 크기 boundarySize, 업샘플링 경계 플래그 needUpsBdryHor, 및 업샘플링 경계 크기 upsBdryW를 입력들로서 취하여 기동되고,  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들 redT[ x ]와  $x = 0..upsBdryW - 1$ 인 업샘플링 경계 샘플들 upsBdryT[ x ]을 출력들로서 취한다.
- [1074] - 조항 8.4.5.2.3에 명시된 MIP 경계 다운샘플링 프로세스는, 좌측 기준 샘플들에 대해, 블록 크기 nTbH,  $y = 0..nTbH - 1$ 인 기준 샘플들 refL[ y ], 경계 크기 boundarySize, 업샘플링 경계 플래그 needUpsBdryVer, 및 업샘플링 경계 크기 upsBdryH를 입력들로서 취하여 기동되고,  $x = 0..boundarySize-1$ 인 감축된 경계 샘플들 redL[ x ] 및  $x = 0..upsBdryH - 1$ 인 업샘플링 경계 샘플들 upsBdryL[ x ]을 출력들로서 취한다.
- [1075] - 감축된 상단 및 좌측 경계 샘플들 redT 및 redL은 다음과 같이 경계 샘플 어레이 p에 할당된다:
- [1076] - isTransposed가 1이라면, p[ x ]는  $x = 0..boundarySize-1$ 인 redL[ x ]로 설정되고, p[ x + boundarySize ]는  $x = 0..boundarySize - 1$ 인 redT[ x ]로 설정된다.
- [1077] - 그렇지 않으면, p[ x ]는  $x = 0..boundarySize-1$ 인 redT[ x ]로 설정되고 p[ x + boundarySize ]는  $x = 0..boundarySize-1$ 인 redL[ x ]로 설정된다.
- [1078] predModeIntra에 따른 인트라 샘플 예측 프로세스의 경우, 다음과 같은 순서화된 단계들이 적용된다:
- [1079] 7.  $x = 0..mipW-1$ ,  $y = 0..mipH-1$ 인 행렬 기반의 인트라 예측 샘플들 predMip[ x ][ y ]는 다음과 같이 도출된다:
- 변수 modeId는 다음과 같이 도출된다:
- [1081] modeId = predModeIntra - ( isTransposed ? numModes / 2 : 0 ) (8-63)
- [1082] -  $x = 0..2 * boundarySize-1$ ,  $y = 0..predC * predC-1$ 인 가중치 행렬 mWeight[ x ][ y ]는 표 8-XX에 명시된 MipSizeId[ xTbCmp ][ yTbCmp ] 및 modeId를 이용하여 도출된다[Ed. (BB): 10비트가 아닌 가중치 솔루션이 채택되면 가중치 행렬들을 추가]
- [1083] -  $y=0..predC * predC-1$ 인 편향 벡터 vBias[ y ]는 표 8-XX에 명시된 sizeId 및 modeId를 이용하여 도출된다[Ed. (BB): 10비트가 아닌 가중치 솔루션이 채택되면 편향 벡터들을 추가].
- [1084] - 변수 sW는 표 8-8에 명시된 대로 MipSizeId[ xTbCmp ][ yTbCmp ] 및 modeId를 이용하여 도출된다.
- [1085] -  $x = 0..mipW-1$ ,  $y = 0..mipH-1$ 인 행렬 기반의 인트라 예측 샘플들 predMip[ x ][ y ]는 다음과 같이 도출된다:

- [1086]  $oW = 1 \ll (sW - 1)$  (8-64)
- [1087]  $sB = BitDepth_y - 1$  (8-65)
- [1088]  $incW = (\text{predC} > \text{mipW}) ? 2 : 1$  (8-66)
- [1089]  $incH = (\text{predC} > \text{mipH}) ? 2 : 1$  (8-67)
- [1090]  $\text{predMip}[x][y] = ((\sum_{i=0}^{2 * \text{boundarySize} - 1} \text{mWeight}[i][y * incH * \text{predC} + x * incW] * p[i]) + (vBias[y * incH * \text{predC} + x * incW] \ll sB) + oW) \gg sW$  (8-68)
- [1091] 8. `isTransposed`가 TRUE이면,  $x = 0..predH-1$ ,  $y = 0..predW-1$ 인 `predH` x `predW` 어레이 `predMip[x][y]`가 다음과 같이 전치된다:
- [1092]  $\text{predTemp}[y][x] = \text{predMip}[x][y]$  (8-69)
- [1093]  $\text{predMip} = \text{predTemp}$  (8-70)
- [1094] 9.  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들 `predSamples[x][y]`는 다음과 같이 도출된다:
- [1095] - `needUpsBdryVer`가 TRUE이거나 `needUpsBdryHor`가 TRUE이면, 조항 8.4.5.2.4에 명시된 MIP 예측 업샘플링 프로세스는, 입력 블록 폭 `predW`, 입력 블록 높이 `predH`,  $x = 0..predW - 1$ ,  $y = 0..predH - 1$ 인 행렬 기반의 인트라 예측 샘플들 `predMip[x][y]`, 변환 블록 폭 `nTbW`, 변환 블록 높이 `nTbH`, 업샘플링 경계 폭 `upsBdryW`, 업샘플링 경계 높이 `upsBdryH`, 상단 업샘플링 경계 샘플들 `upsBdryT`, 및 좌측 업샘플링 경계 샘플들 `upsBdryL`을 입력들로서 취하여 기동되고, 출력은 예측된 샘플 어레이 `predSamples`이다.
- [1096] - 그렇지 않으면,  $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ 인 `predSamples[x][y]`는 `predMip[x][y]`로 설정된다.
- [1097] 10.  $x = 0..nTbW-1 = 0..nTbH-1$ 인 예측된 샘플들 `predSamples[x][y]`는 다음과 같이 클리핑된다:
- [1098]  $\text{predSamples}[x][y] = \text{Clip}_Y(\text{predSamples}[x][y])$  (8-71)

표 8-8 - `MipSizeId` 및 `modeId`에 의존하는 가중치 시프트 `sW` 명세

	modeId																	
<code>MipSizeId</code>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	
1	8	8	8	9	8	8	8	8	9	8								
2	8	8	8	8	8	8												

[1100]

#### 8.4.5.2.4 MIP 예측 업샘플링 프로세스

[1101] a) 프로세스에 대한 입력들은 다음과 같다:

[1102] - 입력 블록 폭을 명시하는 변수 `predW`,[1103] - 입력 블록 높이를 명시하는 변수 `predH`,[1104] -  $x = 0..predW - 1$ ,  $y = 0..predH - 1$ 인 행렬 기반의 인트라 예측된 샘플들 `predMip[x][y]`,[1105] - 변환 블록 폭을 명시하는 변수 `nTbW`,[1106] - 변환 블록 높이를 명시하는 변수 `nTbH`,[1107] - 업샘플링 경계 폭을 명시하는 변수 `upsBdryW`,[1108] - 업샘플링 경계 높이를 명시하는 변수 `upsBdryH`,[1109] -  $x = 0..upsBdryW - 1$ 인 상단 업샘플링 경계 샘플들 `upsBdryT[x]`,[1110] -  $x = 0..upsBdryW - 1$ 인 좌측 업샘플링 경계 샘플들 `upsBdryL[x]`,

- [1111] -  $x = 0..upsBdryH-1$ 인 좌측 업샘플링 경계 샘플들  $upsBdryL[x]$ .
- [1112] 이 프로세스의 출력들은  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들  $predSamples[x][y]$ 이다.
- [1113] 저밀도 예측된 샘플들  $predSamples[m][n]$ 는 다음과 같이  $x = 0..predW-1$ ,  $y = 0..predH-1$ 인  $predMip[x][y]$ 로부터 도출된다:
- [1114]  $upHor = nTbW / predW$  (8-78)
- [1115]  $upVer = nTbH / predH$  (8-79)
- [1116]  $predSamples[(x+1)*upHor-1][(y+1)*upVer-1] = predMip[x][y]$  (8-80)
- [1117]  $x = 0..upsBdryW-1$  상단 경계 샘플들  $upsBdryT[x]$ 는 다음과 같이  $predSamples[m][-1]$ 에 할당된다:
- [1118]  $predSamples[(x+1)*(nTbW/upsBdryW)-1][-1] = upsBdryT[x]$  (8-81)
- [1119]  $y = 0..upsBdryH-1$ 인 좌측 경계 샘플들  $upsBdryL[y]$ 은 다음과 같이  $predSamples[-1][n]$ 에 할당된다:
- [1120]  $predSamples[-1][(y+1)*(nTbH/upsBdryH)-1] = upsBdryL[y]$  (8-82)
- [1121]  $x = 0..nTbW-1$ ,  $y = 0..nTbH-1$ 인 예측된 샘플들  $predSamples[x][y]$ 는 다음과 같이 도출된다:
- [1122] -  $nTbH > nTbW$ 보다 크다면. 다음과 같은 순서화된 단계들이 적용된다:
- [1123] 1.  $upHor$ 가 1보다 크면,  $m = 0..predW-1$ ,  $n = 1..predH-1$ 인 모든 저밀도 위치들 ( $xHor$ ,  $yHor$ ) = ( $m * upHor - 1, n * upVer - 1$ )에 대한 수평 업샘플링이 다음과 같이  $dX = 1..upHor-1$ 과 함께 적용된다:  
 $sum = (upHor - dX) * predSamples[xHor][yHor] +$   
 $dX * predSamples[xHor + upHor][yHor]$  (8-83)
- [1124]  $predSamples[xHor + dX][yHor] = (sum + upHor/2 - (sum < 0 ? 1 : 0)) / upHor$  (8-84)
- [1125] 2.  $m = 0..nTbW-1$ ,  $n = 0..predH-1$ 인 모든 저밀도 위치들 ( $xVer$ ,  $yVer$ ) = ( $m, n * upVer-1$ )에 대한 수직 업샘플링은 다음과 같이  $dY = 1..upVer-1$ 과 함께 적용된다:  
 $sum = (upVer - dY) * predSamples[xVer][yVer] +$   
 $dY * predSamples[xVer + upVer][yVer]$  (8-85)
- [1126]  $predSamples[xVer][yVer + dY] = (sum + upVer/2 - (sum < 0 ? 1 : 0)) / upVer$  (8-86)
- [1127] - 그렇지 않으면. 다음과 같은 순서화된 단계들가 적용된다:
- [1128] 1.  $upVer$ 가 1보다 크면,  $m = 1..predW$ ,  $n = 0..predH-1$ 인 모든 저밀도 위치들 ( $xVer$ ,  $yVer$ ) = ( $m * upHor - 1, n * upVer - 1$ )에 대한 수직 업샘플링이 다음과 같이  $dY = 1..upVer-1$ 과 함께 적용된다:  
 $sum = (upVer - dY) * predSamples[xVer][yVer] +$   
 $dY * predSamples[xVer + upVer][yVer]$  (8-87)
- [1129]  $predSamples[xVer][yVer + dY] = (sum + upVer/2 - (sum < 0 ? 1 : 0)) / upVer$  (8-88)
- [1130] 2.  $m = 0..predW-1$ ,  $n = 0..nTbH-1$ 인 모든 저밀도 위치들 ( $xHor$ ,  $yHor$ ) = ( $m * upHor-1, n$ )에 대한 수평 업샘플링은 다음과 같이  $dX = 1..upHor-1$ 과 함께 적용된다:  
 $sum = (upHor - dX) * predSamples[xHor][yHor] +$   
 $dX * predSamples[xHor + upHor][yHor]$  (8-89)
- [1131]  $predSamples[xHor + dX][yHor] = (sum + upHor/2 - (sum < 0 ? 1 : 0)) / upHor$  (8-90)

- [1136] 5.8 MPM 코딩을 위한 JVET-N1001-v7에 기초한 예시적인 연구 초안
- [1137] 8.4.2 MIP 모드에 대한 도출 프로세스
- [1138] 이 프로세스에 대한 입력은 다음과 같다:
- 현재 픽처의 상단 좌측 루마 샘플에 관해 현재 루마 코딩 블록의 상단 좌측 샘플을 명시하는 루마 위치 ( $x_{Cb}$ ,  $y_{Cb}$ ),
  - 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수  $cbWidth$ ,
  - 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수  $cbHeight$ .
- [1140] 이 프로세스에서, 행렬 기반의 인트라 예측 모드  $IntraPredModeY[x_{Cb}][y_{Cb}]$ 가 도출된다.  $IntraPredModeY[x_{Cb}][y_{Cb}]$ 는 다음과 같은 순서화된 단계들에 의해 도출된다:
- [1141] 5. 이웃 위치들 ( $xNbA$ ,  $yNbA$ ) 및 ( $xNbB$ ,  $yNbB$ )은 각각 ( $x_{Cb}-1$ ,  $y_{Cb}$ ) 및 ( $x_{Cb}$ ,  $y_{Cb}-1$ )와 동일하게 설정된다.
- [1142] 6.  $X$ 가  $A$  또는  $B$ 로 대체되는 경우, 변수  $candMipModeX$ 는 다음과 같이 도출된다:
- [1143] - 조항 6.4.X [Ed. (BB): 이웃 블록 가용성 체킹 프로세스 tbd]에 명시된 블록에 대한 가용성 도출 프로세스는, ( $x_{Cb}$ ,  $y_{Cb}$ )와 동일하게 설정된 위치( $x_{Curr}$ ,  $y_{Curr}$ )와 ( $xNbX$ ,  $yNbX$ )와 동일하게 설정된 이웃 위치( $xNbY$ ,  $yNbY$ )를 입력으로서 취하여 기동되고, 출력은  $availableX$ 에 할당된다.
- [1144] - 후보 MIP 모드  $candMipModeX$ 는 다음과 같이 도출된다:
- [1145] - 다음과 같은 조건들 중 하나 이상이 참이라면,  $candMipModeX$ 는 -1로 설정된다.
- [1146] - 변수  $availableX$ 는 FALSE이다.
- [1147] -  $CuPredMode[xNbX][yNbX]$ 는 MODE\_INTRA가 아니고  $clip_flag[xNbX][yNbX]$ 는 1이 아니다.
- [1148] -  $pcm_flag[xNbX][yNbX]$ 는 1이다.
- [1149] -  $X$ 는  $B$ 이고  $y_{Cb}-1$ 은  $((y_{Cb} >> CtbLog2SizeY) << CtbLog2SizeY)$ 보다 작다.
- [1150] - 그렇지 않으면, 다음이 적용된다:
- [1151] -  $intra_mip_flag[xNbX][yNbX]$ 가 1이면, 다음이 적용된다:
- [1152]  $MipSizeId[x_{Cb}][y_{Cb}]$  가  $MipSizeId[xNbX][yNbX]$ 이면,  $candMipModeX$ 는  $IntraPredModeY[xNbX][yNbX]$ 로 설정된다.
- [1153] - 그렇지 않으면,  $candMipModeX$ 가 -1로 설정된다.
- [1154] - 그렇지 않으면, 표 8-4에 명시된 대로  $IntraPredModeY[xNbX][yNbX]$  및  $MipSizeId[x_{Cb}][y_{Cb}]$ 를 이용하여  $candMipModeX$ 가 도출된다.
- [1155] 7.  $x = 0..2$ 인  $candMipModeList[x]$ 는 표 8-2에 명시된 대로  $mipMpmCand[sizeId]$ 를 이용하여 다음과 같이 도출된다:
- [1156] -  $candMipModeA$  및  $candMipModeB$  양쪽 모두가 -1이라면, 다음이 적용된다:
- [1157]  $candMipModeList[0] = mipMpmCand[sizeId][0]$  (8-10)
- [1158]  $candMipModeList[1] = mipMpmCand[sizeId][1]$  (8-11)
- [1159]  $candMipModeList[2] = mipMpmCand[sizeId][2]$  (8-12)
- [1160] - 그렇지 않으면, 다음이 적용된다:
- [1161] -  $candMipModeA$ 가  $candMipModeB$ 거나  $candMipModeA$  또는  $candMipModeB$ 가 -1이라면, 다음이 적용된다:
- [1162]  $candMipModeList[0] = (candMipModeA != -1) ? candMipModeA : candMipModeB$  (8-13)

- [1165] - candMipModeList[ 0 ]가 mipMpmCand[ sizeId ][ 0 ]이면, 다음이 적용된다:
- [1166] `candMipModeList[ 1 ] = mipMpmCand[ sizeId ][ 1 ](8-14)`
- [1167] `candMipModeList[ 2 ] = mipMpmCand[ sizeId ][ 2 ](8-15)`
- [1168] - 그렇지 않으면, 다음이 적용된다:
- [1169] `candMipModeList[ 1 ] = mipMpmCand[ sizeId ][ 0 ](8-16)`
- `candMipModeList[ 2 ] =`
- `(candMipModeList[ 0 ] != mipMpmCand[ sizeId ][ 1 ]) ?`
- `mipMpmCand[ sizeId ][ 1 ] : mipMpmCand[ sizeId ][ 2 ]`
- [1170] `(8-17)`
- [1171] - 그렇지 않으면, 다음이 적용된다:
- [1172] `candMipModeList[ 0 ] = candMipModeA(8-18)`
- [1173] `candMipModeList[ 1 ] = candMipModeB(8-19)`
- [1174] - candMipModeA 및 candMipModeB 양쪽 모두가 mipMpmCand[ sizeId ][ 0 ]이 아니면, 다음이 적용된다:
- [1175] `candMipModeList[ 2 ] = mipMpmCand[ sizeId ][ 0 ](8-20)`
- [1176] - 그렇지 않으면, 다음이 적용된다:
- [1177] - candMipModeA 및 candMipModeB 양쪽 모두가 mipMpmCand[ sizeId ][ 1 ]이 아니면, 다음이 적용된다:
- [1178] `candMipModeList[ 2 ] = mipMpmCand[ sizeId ][ 1 ](8-21)`
- [1179] - 그렇지 않으면, 다음이 적용된다:
- [1180] `candMipModeList[ 2 ] = mipMpmCand[ sizeId ][ 2 ](8-22)`
- [1181] 8. IntraPredModeY[ xCb ][ yCb ]는 다음 절차를 적용하여 도출된다:
- [1182] - intra\_mip\_mpm\_flag[ xCb ][ yCb ]가 1이면, IntraPredModeY[ xCb ][ yCb ]가 candMipModeList[ intra\_mip\_mpm\_idx[ xCb ][ yCb ] ]로 설정된다.
- [1183] - 그렇지 않으면, IntraPredModeY[ xCb ][ yCb ]는 다음과 같은 순서화된 단계들을 적용하여 도출된다:
- [1184] 3. i = 0..1이고 각각의 i, j = ( i + 1 )..2에 대해 candMipModeList[ i ]가 candMipModeList[ j ]보다 크면, 양쪽 값들은 다음과 같이 스왑된다:
- ( candMipModeList[ i ], candMipModeList[ j ] ) =  
Swap( candMipModeList[ i ], candMipModeList[ j ] )      (8-23)
- [1186] 4. IntraPredModeY[ xCb ][ yCb ]는 다음과 같은 순서화된 단계들에 의해 도출된다:
- [1187] i. IntraPredModeY[ xCb ][ yCb ]는 intra\_mip\_mpm\_remainder[ xCb ][ yCb ]와 동일하게 설정된다.
- [1188] ii. i가 0 내지 2인 경우, IntraPredModeY[ xCb ][ yCb ]가 candMipModeList[ i ]보다 크거나 같다면, IntraPredModeY[ xCb ][ yCb ]값이 1씩 증가된다.
- [1189] x = xCb..xCb + cbWidth-1 및 y = yCb..yCb + cbHeight-1인 변수 IntraPredModeY[ x ][ y ]는 IntraPredModeY[ xCb ][ yCb ]와 동일하도록 설정된다.

[1190]

표 8-1 - 인트라 예측과 MIP 모드들 사이의 맵핑 명세

<u>IntraPredModeY/[xNbX]/[yNbX]</u>	<u>MipSizeld</u>		
	<u>0</u>	<u>1</u>	<u>2</u>
<u>0</u>	<u>17</u>	<u>0</u>	<u>5</u>
<u>1</u>	<u>17</u>	<u>0</u>	<u>1</u>
<u>2,3</u>	<u>17</u>	<u>10</u>	<u>3</u>
<u>4,5</u>	<u>2</u>	<u>10</u>	<u>3</u>
<u>6,7</u>	<u>2</u>	<u>10</u>	<u>3</u>
<u>8,9</u>	<u>2</u>	<u>10</u>	<u>3</u>
<u>10,11</u>	<u>2</u>	<u>10</u>	<u>0</u>
<u>12,13</u>	<u>17</u>	<u>4</u>	<u>0</u>
<u>14,15</u>	<u>17</u>	<u>6</u>	<u>0</u>
<u>16,17</u>	<u>17</u>	<u>7</u>	<u>4</u>
<u>18,19</u>	<u>17</u>	<u>7</u>	<u>4</u>
<u>20,21</u>	<u>17</u>	<u>7</u>	<u>4</u>
<u>22,23</u>	<u>17</u>	<u>5</u>	<u>5</u>
<u>24,25</u>	<u>17</u>	<u>5</u>	<u>1</u>
<u>26,27</u>	<u>5</u>	<u>0</u>	<u>1</u>
<u>28,29</u>	<u>5</u>	<u>0</u>	<u>1</u>
<u>30,31</u>	<u>5</u>	<u>3</u>	<u>1</u>
<u>32,33</u>	<u>5</u>	<u>3</u>	<u>1</u>
<u>34,35</u>	<u>34</u>	<u>12</u>	<u>6</u>
<u>36,37</u>	<u>22</u>	<u>12</u>	<u>6</u>
<u>38,39</u>	<u>22</u>	<u>12</u>	<u>6</u>
<u>40,41</u>	<u>22</u>	<u>12</u>	<u>6</u>
<u>42,43</u>	<u>22</u>	<u>14</u>	<u>6</u>
<u>44,45</u>	<u>34</u>	<u>14</u>	<u>10</u>
<u>46,47</u>	<u>34</u>	<u>14</u>	<u>10</u>
<u>48,49</u>	<u>34</u>	<u>16</u>	<u>9</u>
<u>50,51</u>	<u>34</u>	<u>16</u>	<u>9</u>
<u>52,53</u>	<u>34</u>	<u>16</u>	<u>9</u>
<u>54,55</u>	<u>34</u>	<u>15</u>	<u>9</u>
<u>56,57</u>	<u>34</u>	<u>13</u>	<u>9</u>
<u>58,59</u>	<u>26</u>	<u>1</u>	<u>8</u>
<u>60,61</u>	<u>26</u>	<u>1</u>	<u>8</u>
<u>62,63</u>	<u>26</u>	<u>1</u>	<u>8</u>
<u>64,65</u>	<u>26</u>	<u>1</u>	<u>8</u>
<u>66</u>	<u>26</u>	<u>1</u>	<u>8</u>

[1191]

표 8-2 - MIP 후보 모드들 mipMpmCand[ sizeId ][ x ]의 명세

sizeId	후보 모드 x		
	0	1	2
0	17	34	5
1	0	7	16
2	1	4	6

[1193]

## i. 루마 인트라 예측 모드의 도출 프로세스

o) 프로세스에 대한 입력은 다음과 같다:

- 현재 픽처의 상단 좌측 루마 샘플에 관해 현재 루마 코딩 블록의 상단 좌측 샘플을 명시하는 루마 위치 (xCb, yCb),

- 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수 cbWidth,

- [1198] - 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수 *cbHeight*.
- [1199] 이 프로세스에서, 루마 인트라 예측 모드 *IntraPredModeY[ xCb ][ yCb ]*가 도출된다.
- [1200] 표 8-3은 인트라 예측 모드 *IntraPredModeY[ xCb ][ yCb ]*에 대한 값 및 연관된 이름들을 명시한다.
- [1201] 표 8-3 – 인트라 예측 모드 및 연관된 이름들의 명세

인트라 예측 모드	연관된 이름
0	INTRAL_PLANAR
1	INTRAL_DC
2..66	INTRAL_ANGULAR2..INTRAL_ANGULAR66
81..83	INTRAL_LT_CCLM, INTRAL_L_CCLM, INTRAL_T_CCLM

- [1202]
- [1203] 유의사항 –: 인트라 예측 모드들 INTRAL\_LT\_CCLM, INTRAL\_L\_CCLM 및 INTRAL\_T\_CCLM은 크로마 성분들에만 적용 가능하다.
- [1204] *IntraPredModeY[ xCb ][ yCb ]*는 다음과 같이 도출된다:
- [1205] - *BdpcmFlag[ xCb ][ yCb ]*가 1이거나 *intra\_luma\_not\_planar\_flag[ xCb ][ yCb ]*가 0이면, *IntraPredModeY[ xCb ][ yCb ]*가 INTRAL\_PLANAR로 설정된다.
- [1206] - 그렇지 않으면 (*intra\_luma\_not\_planar\_flag[ xCb ][ yCb ]*가 1임), 다음과 같은 순서화된 단계들이 적용된다:
- [1207] 9. 이웃 위치들 (*xNbA, yNbA*) 및 (*xNbB, yNbB*)는 각각 (*xCb-1, yCb + cbHeight-1*) 및 (*xCb + cbWidth-1, yCb-1*)와 동일하게 설정된다.
- [1208] 10. X가 A 또는 B로 대체되는 경우, 변수 *candIntraPredModeX*는 다음과 같이 도출된다:
- [1209] - 조항 6.4.X [Ed. (BB): 이웃 블록 가용성 체킹 프로세스 tbd]에 명시된 블록에 대한 가용성 도출 프로세스는, (*xCb, yCb*)와 동일하게 설정된 위치(*xCurr, yCurr*)와 (*xNbX, yNbX*)와 동일하게 설정된 이웃 위치(*xNbY, yNbY*)를 입력으로서 취하여 기동되고, 출력은 *availableX*에 할당된다.
- [1210] - 후보 인트라 예측 모드 *candIntraPredModeX*는 다음과 같이 도출된다:
- [1211] - 다음과 같은 조건들 중 하나 이상이 참이라면, *candIntraPredModeX*는 INTRAL\_PLANAR로 설정된다.
- [1212] - 변수 *availableX*는 FALSE이다.
- [1213] - *CuPredMode[ xNbX ][ yNbX ]*는 *MODE\_INTRA*가 아니고 *ciip\_flag[ xNbX ][ yNbX ]*는 1이 아니다.
- [1214] - *pcm\_flag[ xNbX ][ yNbX ]*는 1이다.
- [1215] - X는 B이고 *yCb-1*은 ((*yCb >> CtbLog2SizeY*) << *CtbLog2SizeY*)보다 작다.
- [1216] - *intra\_mip\_flag[ xCb ][ yCb ]*는 1이다.
- [1217] - 그렇지 않으면 *candIntraPredModeX*는 다음과 같이 도출된다:
- [1218] - *intra\_mip\_flag[ xCb ][ yCb ]*가 1인 경우, *candIntraPredModeX*는 표 8-4에 명시된 *IntraPredModeY[ xNbX ][ yNbX ]* 및 *MipSizeId[ xCb ][ yCb ]*를 이용하여 도출된다.
- [1219] - 그렇지 않으면 *candIntraPredModeX*는 *IntraPredModeY[ xNbX ][ yNbX ]*로 설정된다.
- [1220] 11. *x = 0..4*인 *candModeList[ x ]*는 다음과 같이 도출된다:
- [1221] - *candIntraPredModeB*가 *candIntraPredModeA*와 같고 *candIntraPredModeA*가 INTRAL\_DC보다 크다면, *x=0..4*인 *candModeList[x]*는 다음과 같이 도출된다:
- [1222] *candModeList[ 0 ] = candIntraPredModeA* (8-24)

[1223] candModeList[ 1 ] = 2 + ( ( candIntraPredModeA + 61 ) % 64 ) (8-25)

[1224] candModeList[ 2 ] = 2 + ( ( candIntraPredModeA - 1 ) % 64 ) (8-26)

[1225] candModeList[ 3 ] = INTRA\_DC (8-27)

[1226] candModeList[ 4 ] = 2 + ( ( candIntraPredModeA + 60 ) % 64 ) (8-28)

[1227] - 그렇지 않고 candIntraPredModeB가 candIntraPredModeA와 같지 않고 candIntraPredModeA 또는 candIntraPredModeB가 INTRA\_DC보다 크다면, 다음이 적용된다:

[1228] - 변수들 minAB 및 maxAB는 다음과 같이 도출된다:

[1229] minAB = Min( candIntraPredModeA, candIntraPredModeB ) (8-29)

[1230] maxAB = Max( candIntraPredModeA, candIntraPredModeB ) (8-30)

[1231] - candIntraPredModeA와 candIntraPredModeB 양쪽 모두가 INTRA\_DC보다 큰 경우,  $x = 0..4$ 인 candModeList[ x ]는 다음과 같이 도출된다:

[1232] candModeList[ 0 ] = candIntraPredModeA (8-31)

[1233] candModeList[ 1 ] = candIntraPredModeB (8-32)

[1234] candModeList[ 2 ] = INTRA\_DC (8-33)

[1235] - maxAB - minAB가 2 내지 62(양쪽 끝값 포함) 범위이면, 다음이 적용된다:

[1236] candModeList[ 3 ] = 2 + ( ( maxAB + 61 ) % 64 ) (8-34)

[1237] candModeList[ 4 ] = 2 + ( ( maxAB - 1 ) % 64 ) (8-35)

[1238] - 그렇지 않으면, 다음이 적용된다:

[1239] candModeList[ 3 ] = 2 + ( ( maxAB + 60 ) % 64 ) (8-36)

[1240] candModeList[ 4 ] = 2 + ( ( maxAB ) % 64 ) (8-37)

[1241] - 그렇지 않으면(candIntraPredModeA 또는 candIntraPredModeB가 INTRA\_DC보다 큼),  $x = 0..4$ 인 candModeList[ x ]는 다음과 같이 도출된다:

[1242] candModeList[ 0 ] = maxAB (8-38)

[1243] candModeList[ 1 ] = INTRA\_DC (8-39)

[1244] candModeList[ 2 ] = 2 + ( ( maxAB + 61 ) % 64 ) (8-40)

[1245] candModeList[ 3 ] = 2 + ( ( maxAB - 1 ) % 64 ) (8-41)

[1246] candModeList[ 4 ] = 2 + ( ( maxAB + 60 ) % 64 ) (8-42)

[1247] - 그렇지 않으면, 다음이 적용된다:

[1248] candModeList[ 0 ] = INTRA\_DC (8-43)

[1249] candModeList[ 1 ] = INTRA\_ANGULAR50 (8-44)

[1250] candModeList[ 2 ] = INTRA\_ANGULAR18 (8-45)

[1251] candModeList[ 3 ] = INTRA\_ANGULAR46 (8-46)

[1252] candModeList[ 4 ] = INTRA\_ANGULAR54 (8-47)

[1253] 12. IntraPredModeY[ xCb ][ yCb ]는 다음 절차를 적용하여 도출된다:

[1254] - intra\_luma\_mpm\_flag[ xCb ][ yCb ]가 1이면, IntraPredModeY[ xCb ][ yCb ]가 candModeList[ intra\_luma\_mpm\_idx[ xCb ][ yCb ] ]로 설정된다.

[1255] - 그렇지 않으면, IntraPredModeY[ xCb ][ yCb ]는 다음과 같은 순서화된 단계들을 적용하여 도출된다:

[1256] 5. candModeList[ i ]가,  $i = 0..3$  및 각각의  $i, j = (i + 1)..4$ 에 대해 candModeList[ j ]보다 크면, 양쪽 값들은 다음과 같이 교환된다:

( candModeList[ i ], candModeList[ j ] ) = Swap( candModeList[ i ],  
candModeList[ j ] ) (8-48)

[1258] 6. IntraPredModeY[ xCb ][ yCb ]는 다음과 같은 순서화된 단계들에 의해 도출된다:

[1259] i. IntraPredModeY[ xCb ][ yCb ]는 intra\_luma\_mpm\_remainder[ xCb ][ yCb ]로 설정된다.

[1260] ii. IntraPredModeY[ xCb ][ yCb ]값이 1씩 증가한다.

[1261] iii. i가 0 내지 4인 경우, IntraPredModeY[ xCb ][ yCb ]가 candModeList[ i ]보다 크거나 같다면, IntraPredModeY[ xCb ][ yCb ]값이 1씩 증가된다.

[1262]  $x = xCb..xNbX + cbWidth-1$  및  $y = yCb..yNbX + cbHeight-1$ 인 변수 IntraPredModeY[ x ][ y ]는 IntraPredModeY[ xCb ][ yCb ]로 설정된다.

[1263] 표 8-4 – MIP와 인트라 예측 모드들 사이의 맵핑 명세

<i>IntraPredModeY[ xNbX ][ yNbX ]</i>	<i>MipSizeId</i>		
	<u>0</u>	<u>1</u>	<u>2</u>
<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>
<u>1</u>	<u>18</u>	<u>1</u>	<u>1</u>
<u>2</u>	<u>18</u>	<u>0</u>	<u>1</u>
<u>3</u>	<u>0</u>	<u>1</u>	<u>1</u>
<u>4</u>	<u>18</u>	<u>0</u>	<u>18</u>
<u>5</u>	<u>0</u>	<u>22</u>	<u>0</u>
<u>6</u>	<u>12</u>	<u>18</u>	<u>1</u>
<u>7</u>	<u>0</u>	<u>18</u>	<u>0</u>
<u>8</u>	<u>18</u>	<u>1</u>	<u>1</u>
<u>9</u>	<u>2</u>	<u>0</u>	<u>50</u>
<u>10</u>	<u>18</u>	<u>1</u>	<u>0</u>
<u>11</u>	<u>12</u>	<u>0</u>	
<u>12</u>	<u>18</u>	<u>1</u>	
<u>13</u>	<u>18</u>	<u>0</u>	
<u>14</u>	<u>1</u>	<u>44</u>	
<u>15</u>	<u>18</u>	<u>0</u>	
<u>16</u>	<u>18</u>	<u>50</u>	
<u>17</u>	<u>0</u>	<u>1</u>	
<u>18</u>	<u>0</u>	<u>0</u>	
<u>19</u>	<u>50</u>		
<u>20</u>	<u>0</u>		
<u>21</u>	<u>50</u>		
<u>22</u>	<u>0</u>		
<u>23</u>	<u>56</u>		
<u>24</u>	<u>0</u>		
<u>25</u>	<u>50</u>		
<u>26</u>	<u>66</u>		
<u>27</u>	<u>50</u>		
<u>28</u>	<u>56</u>		
<u>29</u>	<u>50</u>		
<u>30</u>	<u>50</u>		
<u>31</u>	<u>1</u>		
<u>32</u>	<u>50</u>		
<u>33</u>	<u>50</u>		
<u>34</u>	<u>50</u>		

[1264]

- [1265] 8.4.4 크로마 인트라 예측 모드를 위한 도출 프로세스
- [1266] 이) 프로세스에 대한 입력은 다음과 같다:
- 현재 픽처의 상단 좌측 루마 샘플에 관해 현재 크로마 코딩 블록의 상단 좌측 샘플을 명시하는 루마 위치( $xCb$ ,  $yCb$ ),
  - 루마 샘플들에서 현재 코딩 블록의 폭을 명시하는 변수  $cbWidth$ ,
  - 루마 샘플들에서 현재 코딩 블록의 높이를 명시하는 변수  $cbHeight$ .
- [1268] 이) 프로세스에서 크로마 인트라 예측 모드  $IntraPredModeC[xCb][yCb]$ 가 도출된다.
- [1269] 대응하는 루마 인트라 예측 모드  $lumaIntraPredMode$ 는 다음과 같이 도출된다:
- [1270] -  $intra_mip_flag[xCb + cbWidth / 2][yCb + cbHeight / 2]$ 가 1이라면,  $lumaIntraPredMode$ 는 INTRA\_PLANAR로 설정된다. 표 8-4에 명시된  $IntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2]$  및  $sizeId$ 를 이용하여 도출되고  $candIntraPredModeX$  값을  $lumaIntraPredMode$ 에 할당.
- [1271] - 그렇지 않으면,  $lumaIntraPredMode$ 는  $IntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2]$ 로 설정된다.
- [1272] 크로마 인트라 예측 모드  $IntraPredModeC[xCb][yCb]$ 는  $intra_chroma_pred_mode[xCb][yCb]$  및 표 8-5 및 표 8-6에 명시된  $lumaIntraPredMode$ 를 이용하여 도출된다.
- [1273] 전술된 예들은 후술되는 방법들, 예를 들어, 비디오 인코더 및/또는 디코더에서 구현될 수 있는 방법들(1100 내지 1400과 2300 내지 2700)에 정황에서 포함될 수 있다.
- [1274] 도 11은 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 방법(1100)은, 단계 1110에서, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계를 포함한다.
- [1275] 방법(1100)은, 단계 1120에서, 이 결정에 기초하여, 비-ALWIP 인트라 모드에 대한 MPM 목록의 적어도 일부에 기초하여 ALWIP 모드에 대한 MPM(most probable mode) 목록의 적어도 일부를 구성하는 단계를 포함한다.
- [1276] 방법(1100)은, 단계 1130에서, ALWIP 모드에 대한 MPM 목록에 기초하여, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.
- [1277] 일부 실시예에서, ALWIP 모드의 MPM 목록의 크기는 비-ALWIP 인트라 모드에 대한 MPM 목록의 크기와 동일하다. 한 예에서, ALWIP 모드의 MPM 목록 크기는 6이다.
- [1278] 일부 실시예에서, 방법(1100)은 ALWIP 모드에 대한 MPM 목록에 디폴트 모드들을 삽입하는 단계를 더 포함한다. 한 예에서, 디폴트 모드들은 비-ALWIP 인트라 모드에 대한 MPM 목록에 기초하는 ALWIP 모드에 대한 MPM 목록 부분 앞에 삽입된다. 또 다른 예에서, 디폴트 모드들은 비-ALWIP 인트라 모드에 대한 MPM 목록에 기초하는 ALWIP 모드에 대한 MPM 목록 부분에 이어 삽입된다. 역시 또 다른 예에서, 디폴트 모드들은 비-ALWIP 인트라 모드에 대한 MPM 목록에 기초하는 ALWIP 모드에 대한 MPM 목록 부분과 인터리빙된 방식으로 삽입된다.
- [1279] 일부 실시예에서, ALWIP 모드에 대한 MPM 목록 및 비-ALWIP 인트라 모드에 대한 MPM 목록을 구성하는 것은 하나 이상의 이웃 블록에 기초한다.
- [1280] 일부 실시예에서, ALWIP 모드에 대한 MPM 목록 및 비-ALWIP 인트라 모드에 대한 MPM 목록을 구성하는 것은 현재 비디오 블록의 높이 또는 폭에 기초한다.
- [1281] 일부 실시예에서, ALWIP 모드에 대한 MPM 목록을 구성하는 것은, 비-ALWIP 인트라 모드에 대한 MPM 목록을 구성하는데 이용되는 제2 세트의 파라미터들과는 상이한 제1 세트의 파라미터들에 기초한다.
- [1282] 일부 실시예에서, 방법(1100)은, 현재 비디오 블록의 이웃 블록이 ALWIP 모드로 코딩되었다고 결정하는 단계, 및 비-ALWIP 인트라 모드에 대한 MPM 목록을 구성하는데 있어서, 이웃 블록을 이용가능하지 않은 것으로 지정하는 단계를 더 포함한다.
- [1283] 일부 실시예에서, 방법(1100)은, 현재 비디오 블록의 이웃 블록이 비-ALWIP 인트라 모드로 코딩되었다고 결정하는 단계, 및 ALWIP 모드에 대한 MPM 목록을 구성하는데 있어서, 이웃 블록을 이용가능하지 않은 것으로 지정하

는 단계를 더 포함한다.

- [1286] 일부 실시예에서, 비-ALWIP 인트라 모드는, 일반 인트라 모드, 복수의 기준 라인(MRL) 인트라 예측 모드 또는 인트라 서브파티션(ISP) 툴에 기초한다.
- [1287] 도 12는 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 방법(1200)은, 단계 1210에서, 현재 비디오 블록의 루마 성분이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계를 포함한다.
- [1288] 방법(1200)은, 단계 1220에서, 이 결정에 기초하여 크로마 인트라 모드를 추론하는 단계를 포함한다.
- [1289] 방법(1200)은, 단계 1230에서, 크로마 인트라 모드에 기초하여, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.
- [1290] 일부 실시예에서, 루마 성분은 크로마 성분의 미리결정된 크로마 샘플을 커버한다. 한 예에서, 미리결정된 크로마 샘플은 크로마 성분의 상단 좌측 샘플 또는 중앙 샘플이다.
- [1291] 일부 실시예에서, 추론된 크로마 인트라 모드는 DM 모드이다.
- [1292] 일부 실시예에서, 추론된 크로마 인트라 모드는 ALWIP 모드이다.
- [1293] 일부 실시예에서, ALWIP 모드는 현재 비디오 블록의 하나 이상의 크로마 성분에 적용된다.
- [1294] 일부 실시예에서, ALWIP 모드의 상이한 행렬 또는 편향 벡터들은 현재 비디오 블록의 상이한 컬러 성분들에 적용된다. 한 예에서, 상이한 행렬 또는 편향 벡터들은 Cb 및 Cr 성분들에 대해 공동으로 미리정의된다. 또 다른 예에서, Cb 및 Cr 성분들은 연결된다. 또 다른 예에서, Cb 및 Cr 성분은 인터리빙된다.
- [1295] 도 13은 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 방법(1300)은, 단계 1310에서, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계를 포함한다.
- [1296] 방법(1300)은, 단계 1320에서, 이 결정에 기초하여, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.
- [1297] 일부 실시예에서, 이 결정은, 시퀀스 파라미터 세트(SPS), 픽처 파라미터 세트(PPS), 슬라이스 헤더, 타일 그룹 헤더, 타일 헤더, 코딩 트리 유닛(CTU) 행 또는 CTU 영역에서의 시그널링에 기초한다.
- [1298] 일부 실시예에서, 결정은 현재 비디오 블록의 높이(H) 또는 폭(W)에 기초한다. 한 예에서,  $W > T1$  또는  $H > T2$ 이다. 또 다른 예에서,  $W \geq T1$  또는  $H \geq T2$ 이다. 역시 또 다른 예에서,  $W < T1$  또는  $H < T2$ 이다. 역시 또 다른 예에서,  $W \leq T1$  또는  $H \leq T2$ 이다. 역시 또 다른 예에서,  $T1 = 32$  및  $T2 = 32$ 이다.
- [1299] 일부 실시예에서, 결정은 현재 비디오 블록의 높이(H) 또는 폭(W)에 기초한다. 한 예에서,  $W + H \leq T$ 이다. 또 다른 예에서,  $W + H \geq T$ 이다. 역시 또 다른 예에서,  $W \times H \leq T$ 이다. 역시 또 다른 예에서,  $W \times H \geq T$ 이다. 역시 또 다른 예에서,  $T = 256$ 이다.
- [1300] 도 14는 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 방법(1400)은, 단계 1410에서, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드와는 상이한 코딩 모드를 이용하여 코딩되어 있다고 결정하는 단계를 포함한다.
- [1301] 방법(1400)은, 단계 1420에서, 이 결정에 기초하여, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.
- [1302] 일부 실시예에서, 코딩 모드는 조합된 인트라 및 인터 예측(CIIP) 모드이고, 방법(1400)은 ALWIP 모드와 일반 인트라 예측 모드 사이의 선택을 수행하는 단계를 더 포함한다. 한 예에서, 선택을 수행하는 단계는, 현재 비디오 블록의 비트스트림 표현에서의 명시적 시그널링에 기초한다. 또 다른 예에서, 선택을 수행하는 단계는 미리결정된 규칙에 기초한다. 역시 또 다른 예에서, 미리결정된 규칙은 현재 비디오 블록이 CIIP 모드를 이용하여 코딩되어 있을 때 항상 ALWIP 모드를 선택한다. 역시 또 다른 예에서, 미리결정된 규칙은 현재 비디오 블록이 CIIP 모드를 이용하여 코딩되어 있을 때 항상 일반 인트라 예측 모드를 선택한다.
- [1303] 일부 실시예에서, 코딩 모드는 교차-성분 선형 모델(cross-component linear model)(CCLM) 예측 모드이다. 한 예에서, ALWIP 모드에 대한 다운샘플링 절차는 CCLM 예측 모드에 대한 다운샘플링 절차에 기초한다. 또 다른

예에서, ALWIP 모드에 대한 다운샘플링 절차는 제1 세트의 파라미터들에 기초하고, CCLM 예측 모드에 대한 다운샘플링 절차는 제1 세트의 파라미터들과는 상이한 제2 세트의 파라미터들에 기초한다. 또 다른 예에서, ALWIP 모드 또는 CCLM 예측 모드에 대한 다운샘플링 절차는, 다운샘플링된 위치들의 선택, 다운샘플링 필터들의 선택, 반올림 동작 또는 클리핑 동작 중 적어도 하나를 포함한다.

[1304] 일부 실시예에서, 방법(1400)은, 감축된 2차 변환(RST), 2차 변환, 회전 변환 또는 비분 2차 변환(Non-Separable Secondary Transform)(NSST) 중 하나 이상을 적용하는 단계를 더 포함한다.

[1305] 일부 실시예에서, 방법(1400)은 블록 기반의 차분 필스 코딩 변조(differential pulse coded modulation)(DPCM) 또는 잔차 DPCM을 적용하는 단계를 더 포함한다.

[1306] 일부 실시예에서, 비디오 처리 방법은, 현재 비디오 블록에 대한 규칙에 기초하여, 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안 어파인 선형 가중된 인트라 예측(ALWIP) 모드의 이용을 나타내는 플래그의 컨텍스트를 결정하는 단계; ALWIP 모드에 기초하여, 상기 현재 비디오 블록의 복수의 서브블록을 예측하는 단계; 및 상기 예측에 기초하여, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다. 규칙은 선행적 기술을 이용하여 목시적으로 명시될 수 있거나 코딩된 비트스트림에서 시그널링될 수 있다. 이 방법의 다른 예들과 양태들은 섹션 4의 항목 37 및 38에 추가로 설명되어 있다.

[1307] 일부 실시예에서, 비디오 처리 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 및 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안, 상기 ALWIP 모드와 연관된 업샘플링 프로세스에서 현재 비디오 블록의 샘플들에 관해 적어도 2개의 필터링 스테이지를 수행하는 단계를 포함하고, 상기 적어도 2개의 필터링 스테이지 중 제1 필터링 스테이지에서의 샘플들의 제1 정밀도는 상기 적어도 2개의 필터링 스테이지 중 제2 필터링 스테이지에서의 샘플들의 제2 정밀도와는 상이하다.

[1308] 한 예에서, 현재 비디오 블록의 샘플들은, 예측 샘플들, 업샘플링 프로세스 이전의 중간 샘플들 또는 업샘플링 프로세스 이후의 중간 샘플들이다. 또 다른 예에서, 샘플들은 제1 필터링 스테이지에서 수평으로 제1 치수에서 업샘플링되고, 여기서 샘플들은 제2 필터링 스테이지에서 수직으로 제2 치수에서 업샘플링된다. 역시 또 다른 예에서, 샘플들은 제1 필터링 스테이지에서 수직으로 제1 치수에서 업샘플링되고, 여기서 샘플들은 제2 필터링 스테이지에서 수평으로 제2 치수에서 업샘플링된다.

[1309] 한 예에서, 제1 필터링 스테이지의 출력은 우측 시프트되거나 나누어져 처리된 출력을 생성하고, 여기서, 처리된 출력은 제2 필터링 스테이지에 대한 입력이다. 또 다른 예에서, 제1 필터링 스테이지의 출력은 좌측 시프트되거나 곱해져 처리된 출력을 생성하고, 여기서 처리된 출력은 제2 필터링 스테이지에 대한 입력이다. 이 방법의 다른 예들과 양태들은 섹션 4의 항목 40에 추가로 설명되어 있다.

[1310] 섹션 4의 항목 41 내지 항목 43에서 추가로 설명된 바와 같이, 비디오 처리 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안, ALWIP 모드와 연관된 업샘플링 프로세스에서의 현재 비디오 블록의 샘플들에 관해 적어도 2개의 필터링 스테이지를 수행하는 단계를 포함하고, 여기서, 업샘플링 프로세스는 수직 및 수평 업샘플링 양쪽 모두가 수행되는 경우에 대해 고정된 순서로 수행된다. 섹션 4의 항목 41 내지 43에서 추가로 설명된 바와 같이, 또 다른 방법은, 현재 비디오 블록이 어파인 선형 가중된 인트라 예측(ALWIP) 모드를 이용하여 코딩되어 있다고 결정하는 단계; 및 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환 동안, 상기 ALWIP 모드와 연관된 업샘플링 프로세스에서 현재 비디오 블록의 샘플들에 관해 적어도 2개의 필터링 스테이지를 수행하는 단계를 포함하고, 상기 변환은 상기 업샘플링 프로세스 이전에 전치 동작을 수행하는 단계를 포함한다.

[1311] 전술된 방법들의 추가적인 피처들은 섹션 4의 항목 41 내지 43에 설명되어 있다.

## 6. 개시된 기술의 예시적인 구현들

[1313] 도 15는 비디오 처리 장치(1500)의 블록도이다. 장치(1500)는 여기서 설명된 방법들 중 하나 이상을 구현하는데 이용될 수 있다. 장치(1500)는, 스마트폰, 태블릿, 컴퓨터, 사물 인터넷(IoT) 수신기 등으로 구현될 수 있다. 장치(1500)는, 하나 이상의 프로세서(1502), 하나 이상의 메모리(1504) 및 비디오 처리 하드웨어(1506)를 포함할 수 있다. 프로세서(들)(1502)는 본 문서에 설명된 하나 이상의 (방법들 1100 내지 1400 및 2300 내지 2700을 포함한 그러나 이것으로 제한되지 않는) 방법을 구현하도록 구성될 수 있다. 메모리(메모리들)(1504)는

여기서 설명된 방법들 및 기술들을 구현하는데 이용되는 데이터 및 코드를 저장하는데 이용될 수 있다. 비디오 처리 하드웨어(1506)는 본 문서에 설명된 일부 기술을 하드웨어 회로로 구현하는데 이용될 수 있다.

[1314] 일부 실시예에서, 비디오 코딩 방법들은 도 15와 관련하여 설명된 바와 같이 하드웨어 플랫폼 상에서 구현되는 장치를 이용하여 구현될 수 있다.

[1315] 개시된 기술의 일부 실시예는, 비디오 처리 툴 또는 모드를 가능케하기 위한 결정 또는 판정을 내리는 것을 포함한다. 한 예에서, 비디오 처리 툴 또는 모드가 인에이블되면, 인코더는 비디오 블록의 처리에서 툴 또는 모드를 이용하거나 구현하지만, 툴 또는 모드의 이용에 기초하여 결과 비트스트림을 반드시 수정할 필요는 없다. 즉, 비디오 블록으로부터 비디오의 비트스트림 표현으로의 변환은, 결정 또는 판정에 기초하여 인에이블될 때 비디오 처리 툴 또는 모드를 이용할 것이다. 또 다른 예에서, 비디오 처리 툴 또는 모드가 인에이블될 때, 디코더는 비디오 처리 툴 또는 모드에 기초하여 비트스트림이 수정되었다는 것을 알고 비트스트림을 처리할 것이다. 즉, 비디오의 비트스트림 표현으로부터 비디오 블록으로의 변환은, 결정 또는 판정에 기초하여 인에이블된 비디오 처리 툴 또는 모드를 이용하여 수행될 것이다.

[1316] 개시된 기술의 일부 실시예는, 비디오 처리 툴 또는 모드를 디스에이블하기 위한 결정 또는 판정을 내리는 것을 포함한다. 한 예에서, 비디오 처리 툴 또는 모드가 디스에이블되면, 인코더는 비디오 블록을 비디오의 비트스트림 표현으로 변환하는데 있어서 툴 또는 모드를 이용하지 않을 것이다. 또 다른 예에서, 비디오 처리 툴 또는 모드가 디스에이블될 때, 디코더는 결정 또는 판정에 기초하여 디스에이블된 비디오 처리 툴 또는 모드를 이용하여 비트스트림이 수정되지 않았다는 것을 알고 비트스트림을 처리할 것이다.

[1317] 도 21은 본 개시내용의 기술들을 이용할 수 있는 한 예시적인 비디오 코딩 시스템(100)을 나타내는 블록도이다. 도 21에 도시된 바와 같이, 비디오 코딩 시스템(100)은 소스 디바이스(110) 및 목적지 디바이스(120)를 포함할 수 있다. 소스 디바이스(110)는 비디오 인코딩 디바이스라고 지칭될 수 있는 인코딩된 비디오 데이터를 생성한다. 목적지 디바이스(120)는 비디오 디코딩 디바이스라고 지칭될 수도 있는 소스 디바이스(110)에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수 있다. 소스 디바이스(110)는, 비디오 소스(112), 비디오 인코더(114), 및 입력/출력(I/O) 인터페이스(116)를 포함할 수 있다.

[1318] 비디오 소스(112)는, 비디오 캡처 디바이스 등의 소스, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하는 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽 시스템, 또는 이러한 소스들의 조합을 포함할 수 있다. 비디오 데이터는 하나 이상의 픽처를 포함할 수 있다. 비디오 인코더(114)는 비디오 소스(112)로부터의 비디오 데이터를 인코딩하여 비트스트림을 생성한다. 비트스트림은 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함할 수 있다. 비트스트림은 코딩된 픽처들 및 연관된 데이터를 포함할 수 있다. 코딩된 픽처는 픽처의 코딩된 표현이다. 연관된 데이터는, 시퀀스 파라미터 세트들, 픽처 파라미터 세트들, 및 기타의 선택스 구조들을 포함할 수 있다. I/O 인터페이스(116)는 변조기/복조기(모뎀) 및/또는 전송기를 포함할 수 있다. 인코딩된 비디오 데이터는 네트워크(130a)를 통해 I/O 인터페이스(116)를 이용하여 목적지 디바이스(120)에 직접 전송될 수 있다. 인코딩된 비디오 데이터는 또한, 목적지 디바이스(120)에 의한 액세스를 위해 저장 매체/서버(130b)에 저장될 수 있다.

[1319] 목적지 디바이스(120)는, I/O 인터페이스(126), 비디오 디코더(124), 및 디스플레이 디바이스(122)를 포함할 수 있다.

[1320] I/O 인터페이스(126)는 수신기 및/또는 모뎀을 포함할 수 있다. I/O 인터페이스(126)는 소스 디바이스(110) 또는 저장 매체/서버(130b)로부터 인코딩된 비디오 데이터를 취득할 수 있다. 비디오 디코더(124)는 인코딩된 비디오 데이터를 디코딩할 수 있다. 디스플레이 디바이스(122)는 디코딩된 비디오 데이터를 사용자에게 디스플레이할 수 있다. 디스플레이 디바이스(122)는 목적지 디바이스(120)와 통합될 수 있거나, 외부 디스플레이 디바이스와 인터페이스하도록 구성된 목적지 디바이스(120)의 외부에 있을 수 있다.

[1321] 비디오 인코더(114) 및 비디오 디코더(124)는, HEVC(High Efficiency Video Coding) 표준, VVC(Versatile Video Coding) 표준, 및 기타의 현재 및/또는 추가 표준 등의 비디오 압축 표준에 따라 동작할 수 있다.

[1322] 도 22는 도 21에 나타낸 시스템(100)의 비디오 인코더(114)일 수 있는 비디오 인코더(200)의 한 예를 나타내는 블록도이다.

[1323] 비디오 인코더(200)는 본 개시내용의 기술들 중 임의의 것 또는 전부를 수행하도록 구성될 수 있다. 도 22의 예에서, 비디오 인코더(200)는 복수의 기능 컴포넌트를 포함한다. 본 개시내용에서 설명된 기술들은 비디오 인코더(200)의 다양한 컴포넌트들 사이에서 공유될 수 있다. 일부 예에서, 프로세서는 본 개시내용에서 설명된 기

술들 중 임의의 것 또는 모든 기술들을 수행하도록 구성될 수도 있다.

[1324] 비디오 인코더(200)의 기능 컴포넌트들은, 파티션 유닛(201), 모드 선택 유닛(203), 모션 추정 유닛(204), 모션 보상 유닛(205) 및 인트라 예측 유닛(206)을 포함할 수 있는 예측 유닛(202), 잔차 생성 유닛(207), 변환 유닛(208), 양자화 유닛(209), 역양자화 유닛(210), 역변환 유닛(211), 복원 유닛(212), 베퍼(213), 및 엔트로피 인코딩 유닛(214)을 포함할 수 있다.

[1325] 다른 예들에서, 비디오 인코더(200)는, 더 많거나 더 적거나 또는 상이한 기능 컴포넌트들을 포함할 수 있다. 한 예에서, 예측 유닛(202)은 인트라 블록 복사(IBC) 유닛을 포함할 수 있다. IBC 유닛은 적어도 하나의 기준 픽처가 현재 비디오 블록이 위치해 있는 픽처인 IBC 모드에서 예측을 수행할 수 있다.

[1326] 또한, 모션 추정 유닛(204) 및 모션 보상 유닛(205) 등의 일부 컴포넌트는 고도로 통합될 수 있지만, 설명의 목적을 위해 도 22의 예에서는 별개로 표시된다.

[1327] 파티션 유닛(201)은 픽처를 하나 이상의 비디오 블록으로 파티셔닝할 수 있다. 비디오 인코더(200) 및 비디오 디코더(300)는 다양한 비디오 블록 크기를 지원할 수 있다.

[1328] 모드 선택 유닛(203)은, 예를 들어, 여러 결과들에 기초하여, 코딩 모드들 중 하나, 즉, 인트라 또는 인터 코딩 모드를 선택할 수 있고, 결과적인 인트라-또는 인터-코딩된 블록을 잔차 생성 유닛(207)에 제공하여 잔차 블록 데이터를 생성하고 복원 유닛(212)에 제공하여 기준 픽처로서 이용하기 위해 인코딩된 블록을 복원할 수 있다. 일부 예에서, 모드 선택 유닛(203)은 예측이 인터 예측 신호 및 인트라 예측 신호에 기초하는 CIIP(intra and inter predication) 모드의 조합을 선택할 수 있다. 모드 선택 유닛(203)은 또한, 인터 예측의 경우 블록에 대하여 모션 벡터에 대한 해상도(예를 들어, 서브픽셀 또는 정수 픽셀 정밀도)를 선택할 수 있다.

[1329] 현재 비디오 블록에 관한 인터 예측을 수행하기 위해, 모션 추정 유닛(204)은, 베퍼(213)로부터의 하나 이상의 기준 프레임을 현재 비디오 블록과 비교함으로써 현재 비디오 블록에 대한 모션 정보를 생성할 수 있다. 모션 보상 유닛(205)은, 현재 비디오 블록과 연관된 픽처 이외의 베퍼(213)로부터의 픽처들의 모션 정보 및 디코딩된 샘플들에 기초하여 현재 비디오 블록에 대한 예측된 비디오 블록을 결정할 수 있다.

[1330] 모션 추정 유닛(204) 및 모션 보상 유닛(205)은, 예를 들어 현재 비디오 블록이, I 슬라이스, P 슬라이스, 또는 B 슬라이스에 있는지에 따라 현재 비디오 블록에 대해 상이한 동작들을 수행할 수 있다.

[1331] 일부 예에서, 모션 추정 유닛(204)은 현재 비디오 블록에 대한 일방 예측을 수행할 수 있고, 모션 추정 유닛(204)은 현재 비디오 블록에 대한 기준 비디오 블록에 대해 목록 0 또는 목록 1의 기준 픽처들을 검색할 수도 있다. 그 다음, 모션 추정 유닛(204)은, 기준 비디오 블록, 및 현재 비디오 블록과 기준 비디오 블록 사이의 공간적 변위를 나타내는 모션 벡터를 포함하는 목록 0 또는 목록 1의 기준 픽처를 나타내는 기준 인덱스를 생성할 수 있다. 모션 추정 유닛(204)은, 기준 인덱스, 예측 방향 표시자, 및 모션 벡터를, 현재 비디오 블록의 모션 정보로서 출력할 수 있다. 모션 보상 유닛(205)은, 현재 비디오 블록의 모션 정보가 나타내는 기준 비디오 블록에 기초하여 현재 블록의 예측된 비디오 블록을 생성할 수 있다.

[1332] 다른 예들에서, 모션 추정 유닛(204)은 현재 비디오 블록에 대한 양방향 예측을 수행할 수 있고, 모션 추정 유닛(204)은 현재 비디오 블록에 대한 기준 비디오 블록에 대해 목록 0에서 기준 픽처들을 검색할 수 있고, 또한, 현재 비디오 블록에 대한 또 다른 기준 비디오 블록에 대해 목록 1에서 기준 픽처들을 검색할 수도 있다. 그 다음, 모션 추정 유닛(204)은, 기준 비디오 블록들과 현재 비디오 블록 사이의 공간적 변위들을 나타내는 모션 벡터들 및 기준 비디오 블록들을 포함하는 목록 0 및 목록 1의 기준 픽처들을 나타내는 기준 인덱스들을 생성할 수 있다. 모션 추정 유닛(204)은, 현재 비디오 블록의 모션 정보로서 현재 비디오 블록의 기준 인덱스들 및 모션 벡터들을 출력할 수 있다. 모션 보상 유닛(205)은, 현재 비디오 블록의 모션 정보가 나타내는 기준 비디오 블록들에 기초하여 현재 비디오 블록의 예측된 비디오 블록을 생성할 수 있다.

[1333] 일부 예에서, 모션 추정 유닛(204)은 디코더의 디코딩 처리를 위한 모션 정보의 전체 세트를 출력할 수 있다.

[1334] 일부 예에서, 모션 추정 유닛(204)은 현재 비디오에 대한 모션 정보의 전체 세트를 출력하지 않을 수도 있다. 오히려, 모션 추정 유닛(204)은 또 다른 비디오 블록의 모션 정보를 참조하여 현재 비디오 블록의 모션 정보를 시그널링할 수 있다. 예를 들어, 모션 추정 유닛(204)은 현재 비디오 블록의 모션 정보가 이웃 비디오 블록의 모션 정보와 충분히 유사하다고 결정할 수 있다.

[1335] 한 예에서, 모션 추정 유닛(204)은, 현재 비디오 블록과 연관된 선택스 구조에서, 현재 비디오 블록이 또 다른

비디오 블록과 동일한 모션 정보를 갖는다는 것을 비디오 디코더(300)에 나타내는 값을 표시할 수 있다.

[1336] 또 다른 예에서, 모션 추정 유닛(204)은, 현재 비디오 블록과 연관된 신택스 구조에서, 또 다른 비디오 블록 및 모션 벡터 차이(MVD)를 식별할 수 있다. 모션 벡터 차이는, 현재 비디오 블록의 모션 벡터와 표시된 비디오 블록의 모션 벡터 사이의 차이를 나타낸다. 비디오 디코더(300)는 표시된 비디오 블록의 모션 벡터와 모션 벡터 차이를 이용하여 현재 비디오 블록의 모션 벡터를 결정할 수 있다.

[1337] 위에서 논의된 바와 같이, 비디오 인코더(200)는 모션 벡터를 예측적으로 시그널링할 수 있다. 비디오 인코더(200)에 의해 구현될 수 있는 예측 시그널링 기술들의 2개의 예는 AMVP(advanced motion vector predication) 및 병합 모드 시그널링을 포함한다.

[1338] 인트라 예측 유닛(206)은 현재 비디오 블록에 관해 인트라 예측을 수행할 수 있다. 인트라 예측 유닛(206)이 현재 비디오 블록에 대해 인트라 예측을 수행할 때, 인트라 예측 유닛(206)은 동일한 픽처 내의 다른 비디오 블록들의 디코딩된 샘플들에 기초하여 현재 비디오 블록에 대한 예측 데이터를 생성할 수 있다. 현재 비디오 블록에 대한 예측 데이터는 예측된 비디오 블록 및 다양한 신택스 요소를 포함할 수 있다.

[1339] 잔차 생성 유닛(207)은 현재 비디오 블록으로부터 현재 비디오 블록의 예측된 비디오 블록(들)을 차감(예를 들어, 빼기 부호로 표시)함으로써 현재 비디오 블록에 대한 잔차 데이터를 생성할 수 있다. 현재 비디오 블록의 잔차 데이터는 현재 비디오 블록 내의 샘플들의 상이한 샘플 성분들에 대응하는 잔차 비디오 블록들을 포함할 수 있다.

[1340] 다른 예들에서, 현재 비디오 블록에 대한 잔차 데이터가 없을 수 있고, 예를 들어 스킵 모드에서, 잔차 생성 유닛(207)은 차감 동작을 수행하지 않을 수 있다.

[1341] 변환 처리 유닛(208)은, 현재 비디오 블록과 연관된 잔차 비디오 블록에 하나 이상의 변환을 적용함으로써 현재 비디오 블록에 대한 하나 이상의 변환 계수 비디오 블록을 생성할 수 있다.

[1342] 변환 처리 유닛(208)이 현재 비디오 블록과 연관된 변환 계수 비디오 블록을 생성한 후, 양자화 유닛(209)은 현재 비디오 블록과 연관된 하나 이상의 양자화 파라미터(QP) 값에 기초하여 현재 비디오 블록과 연관된 변환 계수 비디오 블록을 양자화할 수 있다.

[1343] 역양자화 유닛(210) 및 역변환 유닛(211)은 변환 계수 비디오 블록에 역양자화 및 역변환을 각각 적용하여, 변환 계수 비디오 블록으로부터 잔차 비디오 블록을 복원할 수 있다. 복원 유닛(212)은, 버퍼(213)에 저장하기 위한 현재 블록과 연관된 복원된 비디오 블록을 생성하기 위해 예측 유닛(202)에 의해 생성된 하나 이상의 예측된 비디오 블록으로부터의 대응하는 샘플들에 복원된 잔차 비디오 블록을 추가할 수 있다.

[1344] 복원 유닛(212)이 비디오 블록을 복원한 후, 루프 필터링 동작이 수행되어 비디오 블록에서 비디오 블록화 아티팩트들을 감소시킬 수 있다.

[1345] 엔트로피 인코딩 유닛(214)은 비디오 인코더(200)의 다른 기능 컴포넌트들로부터 데이터를 수신할 수 있다. 엔트로피 인코딩 유닛(214)이 데이터를 수신할 때, 엔트로피 인코딩 유닛(214)은 하나 이상의 엔트로피 인코딩 동작들 수행하여 엔트로피 인코딩된 데이터를 생성하고 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수 있다.

[1346] 도 19는 도 21에 나타낸 시스템(100)의 비디오 디코더(114)일 수 있는 비디오 디코더(300)의 한 예를 나타내는 블록도이다.

[1347] 비디오 디코더(300)는 본 개시내용의 기술들 중 임의의 것 또는 전부를 수행하도록 구성될 수 있다. 도 19의 예에서, 비디오 디코더(300)는 복수의 기능 컴포넌트를 포함한다. 본 개시내용에서 설명된 기술들은 비디오 디코더(300)의 다양한 컴포넌트들 사이에서 공유될 수 있다. 일부 예에서, 프로세서는 본 개시내용에서 설명된 기술들 중 임의의 것 또는 모든 기술들을 수행하도록 구성될 수도 있다.

[1348] 도 19의 예에서, 비디오 디코더(300)는, 엔트로피 디코딩 유닛(301), 모션 보상 유닛(302), 인트라 예측 유닛(303), 역양자화 유닛(304), 역변환 유닛(305), 및 복원 유닛(306) 및 버퍼(307)를 포함한다. 비디오 디코더(300)는, 일부 예에서, 비디오 인코더(200)(도 22)에 관하여 설명된 인코딩 패스와는 대체로 역 관계를 이루는 디코딩 패스를 수행할 수 있다.

[1349] 엔트로피 디코딩 유닛(301)은 인코딩된 비트스트림을 회수할 수 있다. 인코딩된 비트스트림은 엔트로피 코딩된 비디오 데이터(예를 들어, 비디오 데이터의 인코딩된 블록들)를 포함할 수 있다. 엔트로피 디코딩 유닛(301)은

엔트로피 코딩된 비디오 데이터를 디코딩할 수 있고, 엔트로피 디코딩된 비디오 데이터로부터, 모션 보상 유닛(302)은, 모션 벡터들, 모션 벡터 정밀도, 기준 픽처 목록 인덱스들, 및 기타의 모션 정보를 포함하는 모션 정보를 결정할 수 있다. 모션 보상 유닛(302)은, 예를 들어, AMVP 및 병합 모드를 수행함으로써 이러한 정보를 결정할 수 있다.

[1350] 모션 보상 유닛(302)은, 아마도 보간 필터들에 기초한 보간을 수행하여, 모션 보상된 블록들을 생성할 수 있다. 서브픽셀 정밀도로 이용될 보간 필터들에 대한 식별자들이 신팩스 요소들에 포함될 수 있다.

[1351] 모션 보상 유닛(302)은, 기준 블록의 정수이하 픽셀들에 대한 보간된 값들을 계산하기 위해 비디오 블록의 인코딩 동안에 비디오 인코더(20)에 의해 이용되는 보간 필터들을 이용할 수 있다. 모션 보상 유닛(302)은 수신된 신팩스 정보에 따라 비디오 인코더(200)에 의해 이용되는 보간 필터들을 결정할 수 있고 보간 필터들을 이용하여 예측 블록들을 생성할 수 있다.

[1352] 모션 보상 유닛(302)은, 인코딩된 비디오 시퀀스의 프레임(들) 및/또는 슬라이스(들)을 인코딩하는데 이용되는 블록들의 크기를 결정하기 위해 신팩스 정보, 인코딩된 비디오 시퀀스의 픽처의 각각의 매크로블록이 어떻게 파티션화되어 있는지를 기술하는 파티션 정보, 각각의 파티션이 어떻게 인코딩되어 있는지를 나타내는 모드들, 각각의 인터-인코딩된 블록에 대한 하나 이상의 기준 프레임(및 기준 프레임 목록들), 및 인코딩된 비디오 시퀀스를 디코딩하기 위한 기타의 정보 중 일부를 이용한다.

[1353] 인트라 예측 유닛(303)은 공간적으로 인접한 블록들로부터 예측 블록을 형성하기 위해 예를 들어 비트스트림에서 수신된 인트라 예측 모드들을 이용할 수 있다. 역양자화 유닛(303)은, 비트스트림에서 제공되고 엔트로피 디코딩 유닛(301)에 의해 디코딩되는 양자화된 비디오 블록 계수들을 역양자화, 즉, 양자화해제한다. 역변환 유닛(303)은 역변환을 적용한다.

[1354] 복원 유닛(306)은 디코딩된 블록들을 형성하기 위해 모션 보상 유닛(202) 또는 인트라 예측 유닛(303)에 의해 생성된 대응하는 예측 블록들과 잔차 블록들을 합산할 수 있다. 원한다면, 블록현상 아티팩트를 제거하기 위하여 디블록킹 필터도 역시 적용되어 디코딩된 블록들을 필터링한다. 그 다음, 디코딩된 비디오 블록들은 베퍼(307)에 저장되며, 베퍼(307)는 후속 모션 보상/인트라 예측을 위한 기준 블록들을 제공하고, 또한, 디스플레이 디바이스 상의 프리젠테이션을 위해 디코딩된 비디오를 생성한다.

[1355] 본 문서에서, "비디오 처리"라는 용어는, 비디오 인코딩, 비디오 디코딩, 비디오 압축 또는 비디오 압축해제를 지칭할 수 있다. 예를 들어, 비디오 압축 알고리즘들은, 비디오의 픽셀 표현으로부터 대응하는 비트스트림 표현으로 또는 그 반대로 변환하는 동안 적용될 수 있다. 현재 비디오 블록의 비트스트림 표현 또는 코딩된 표현은, 예를 들어, 신팩스에 의해 정의된 바와 같이, 비트스트림 내의 상이한 위치들에 함께 배치되거나 분산되어 있는 비트들에 대응할 수 있다. 예를 들어, 비디오 블록은, 변환되고 코딩된 오차 잔차 값들의 측면에서 및 비트스트림의 헤더들 및 기타의 필드들 내의 비트들을 이용하여 인코딩될 수 있다. 또한, 변환 동안, 디코더는 상기의 솔루션들에서 설명된 바와 같이, 결정에 기초하여 일부 필드가 존재하거나 없을 수 있다는 것을 아는 상태에서 비트스트림을 파싱할 수 있다. 유사하게, 인코더는 소정의 신팩스 필드들이 포함되어 있는지의 여부를 결정할 수 있고, 그에 따라 코딩된 표현으로부터 신팩스 필드들을 포함시키거나 제외함으로써 코딩된 표현을 생성할 수 있다.

[1356] 도 20은 여기서 개시된 다양한 기술들이 구현될 수 있는 한 예시적인 비디오 처리 시스템(2000)을 보여주는 블록도이다. 다양한 구현은 시스템(2000)의 컴포넌트들 중 일부 또는 전부를 포함할 수 있다. 시스템(2000)은 비디오 콘텐츠를 수신하기 위한 입력(2002)을 포함할 수 있다. 비디오 콘텐츠는, 원시 또는 압축되지 않은 포맷, 예를 들어 8 또는 10비트 다중 컴포넌트 픽셀 값들로 수신될 수 있거나, 압축되거나 인코딩된 포맷일 수 있다. 입력(2002)은, 네트워크 인터페이스, 주변 버스 인터페이스, 또는 저장 인터페이스를 나타낼 수 있다. 네트워크 인터페이스의 예는, Ethernet, 수동 광 네트워크(pассивный оптический интерфейс)(PON) 등의 유선 인터페이스들, 및 Wi-Fi 또는 셀룰러 인터페이스들 등의 무선 인터페이스들을 포함한다.

[1357] 시스템(2000)은, 본 문서에서 설명된 다양한 코딩 또는 인코딩 방법들을 구현할 수 있는 코딩 컴포넌트(2004)를 포함할 수 있다. 코딩 컴포넌트(2004)는, 비디오의 코딩된 표현을 생성하기 위해 입력(2002)으로부터 코딩 컴포넌트(2004)의 출력으로 비디오의 평균 비트레이트를 감소시킬 수 있다. 따라서 코딩 기술들은, 때때로, 비디오 압축 또는 비디오 트랜스코딩 기술들이라고도 한다. 코딩 컴포넌트(2004)의 출력은, 컴포넌트(2006)에 의해 표현되는 바와 같이, 접속된 통신을 통해 저장되거나 전송될 수 있다. 입력(2002)에서 수신된 비디오의 저장되거나 전달된 비트스트림(또는 코딩된) 표현은, 디스플레이 인터페이스(2010)로 전송되는 픽셀 값들 또는 디스플

레이 가능한 비디오를 생성하기 위해 컴포넌트(2008)에 의해 이용될 수 있다. 비트스트림 표현으로부터 사용자가 볼 수 있는 비디오를 생성하는 프로세스는 때때로 비디오 압축해제라고도 불린다. 더욱이, 소정의 비디오 처리 동작들은 "코딩" 동작들 또는 툴들이라고 지칭되지만, 코딩 툴들 또는 동작들은 인코더에서 이용되고, 코딩의 결과들을 역전시키는 대응하는 디코딩 툴들 또는 동작들은 디코더에 의해 수행될 것이라는 것을 이해할 것이다.

[1358] 주변기기 버스 인터페이스 또는 디스플레이 인터페이스의 예들은, USB(universal serial bus) 또는 HDMI(high definition multimedia interface) 또는 Displayport 등을 포함할 수 있다. 저장 인터페이스들의 예들은, SATA(serial advanced technology attachment), PCI, IDE 인터페이스 등을 포함한다. 본 문서에서 설명된 기술들은, 디지털 데이터 처리 및/또는 비디오 디스플레이를 수행할 수 있는 모바일 전화들, 랩톱들, 스마트폰들, 또는 기타의 디바이스들 등의 다양한 전자 디바이스에 구현될 수 있다.

[1359] 일부 실시예에서, ALWIP 모드 또는 MIP 모드는, 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작(또는 평균화 동작)을 수행한 후, 행렬 벡터 곱셈 동작을 수행하고, 선택적으로(또는 선택사항으로서) 업샘플링 동작(또는 선형 보간 동작)을 수행함으로써, 현재 비디오 블록의 예측 블록을 계산하는데 이용된다. 일부 실시예에서, ALWIP 모드 또는 MIP 모드는, 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작(또는 평균화 동작)을 수행한 후, 행렬 벡터 곱셈 동작을 수행함으로써, 현재 비디오 블록의 예측 블록을 계산하는데 이용된다. 일부 실시예에서, ALWIP 모드 또는 MIP 모드는 또한, 행렬 벡터 곱셈 동작을 수행한 후에 업샘플링 동작(또는 선형 보간 동작)을 수행할 수 있다.

[1360] 도 23은 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 이 방법(2300)은, 비디오의 루마 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 코딩되어 있는지에 관한 제1 결정을 수행하는 단계(2302) – 이 모드에서, 루마 비디오 블록의 예측 블록은 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –; 루마 비디오 블록이 비디오의 현재 크로마 비디오 블록에 대한 크로마 인트라 모드를 결정하는데 적용가능하다는 제2 결정을 수행하는 단계(2304) – 상기 제2 결정을 수행하는 단계는, 현재 크로마 비디오 블록과 연관된 현재 픽처의 상단 좌측 루마 샘플에 관한 현재 크로마 비디오 블록의 상단 좌측 샘플에 기초하고, 상기 제2 결정을 수행하는 단계는 현재 비디오 유닛의 루마 비디오 블록 및/또는 컬러 포맷에 관한 현재 크로마 비디오 블록의 치수들에 기초함 –; 상기 제1 결정 및 상기 제2 결정에 기초하여, 상기 현재 크로마 비디오 블록에 이용될 상기 크로마 인트라 모드에 관한 제3 결정을 수행하는 단계(2306); 및 상기 제3 결정에 기초하여, 상기 현재 크로마 비디오 블록과 상기 현재 크로마 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계(2308)를 포함한다.

[1361] 방법(2300)에 대한 일부 실시예에서, 크로마 인트라 모드는 직접 모드 또는 도출된 모드를 포함한다. 방법(2300)에 대한 일부 실시예에서, 현재 크로마 비디오 블록의 상단 좌측 샘플은 루마 비디오 블록 내에 위치하고, 현재 크로마 비디오 블록의 제1 폭 및 현재 크로마 비디오 블록의 제1 높이는, 각각, 루마 비디오 블록의 제2 폭 및 루마 비디오 블록의 제2 높이보다 작거나 같다. 방법(2300)에 대한 일부 실시예에서, 현재 픽처의 상단 좌측 루마 샘플에 관한 현재 크로마 비디오 블록의 상단 좌측 샘플은  $(x_{Cb}, y_{Cb})$ 이고, 루마 비디오 블록 내에 위치한  $(x_{Cb} + \text{offsetX}, y_{Cb} + \text{offsetY})$ 로 표기되는 위치의 예측 모드는 현재 크로마 비디오 블록에 대한 크로마 인트라 모드를 결정하는데 적용가능하며,  $\text{offsetX}$ 는 수평 방향의 오프셋이고,  $\text{offsetY}$ 는 수직 방향의 오프셋이다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetX}$  및  $\text{offsetY}$  양쪽 모두가 0인 것은 불허된다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetX}$ 는 현재 크로마 비디오 블록의 폭을 2로 나눈 것과 동일하다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetX}$ 는 루마 비디오 블록의 폭을 2로 나눈 것과 동일하다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetX}$ 는 현재 크로마 비디오 블록의 폭을 2로 나눈 결과로부터 1을 뺀 것과 동일하다.

[1362] 방법(2300)에 대한 일부 실시예에서,  $\text{offsetX}$ 는 현재 크로마 비디오 블록의 폭을 2로 나눈 결과에 1을 더한 것과 동일하다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetY}$ 는 현재 크로마 비디오 블록의 높이를 2로 나눈 것과 동일하다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetY}$ 는 루마 비디오 블록의 높이를 2로 나눈 것과 동일하다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetY}$ 는 현재 크로마 비디오 블록의 높이를 2로 나눈 결과로부터 1을 뺀 것과 동일하다. 방법(2300)에 대한 일부 실시예에서,  $\text{offsetY}$ 는 현재 크로마 비디오 블록의 높이를 2로 나눈 결과에 1을 더한 것과 동일하다.

[1363] 도 24는 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 이 방법(2400)은, 비디오의 루마 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 코딩된다는 제1 결정을 수행하는 단계(2402) – 이

모드에서, 루마 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –; 상기 제1 결정에 기초하여, 도출된 모드(derived mode)(DM)가 상기 루마 비디오 블록과 연관된 현재 크로마 비디오 블록에 대한 크로마 인트라 모드로서 이용되는지에 관한 규칙에 따른 제2 결정을 수행하는 단계(2404); 및 상기 제2 결정에 기초하여, 현재 크로마 비디오 블록과 상기 현재 크로마 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계(2406)를 포함한다.

[1364] 방법(2400)에 대한 일부 실시예에서, 규칙은, 루마 비디오 블록이 MIP 모드를 이용하여 코딩되는 것에 응답하여 DM이 크로마 인트라 모드로서 이용되는 것을 명시한다. 방법(2400)에 대한 일부 실시예에서, 비트스트림 표현에서 크로마 인트라 모드의 시그널링은 스kip된다. 방법(2400)에 대한 일부 실시예에서, 변환을 수행하는 단계는, 비트스트림 표현에서 현재 크로마 비디오 블록에 대해 DM에 대한 시그널링을 포함하고, 규칙은 미리정의된 인트라 예측 모드가 현재 크로마 비디오 블록에 적용된다는 것을 명시한다. 방법(2400)에 대한 일부 실시예에서, 미리정의된 인트라 예측 모드는 Planar 모드이다. 방법(2400)에 대한 일부 실시예에서, 미리정의된 인트라 예측 모드는 스크린 콘텐츠와 연관된 코딩된 정보에 기초한다. 방법(2400)에 대한 일부 실시예에서, 미리정의된 인트라 예측 모드는, 스크린 콘텐츠가 카메라 캡처된 스크린 콘텐츠임을 코딩된 정보가 나타내는 것에 응답하여 Planar 모드이다.

[1365] 방법(2400)에 대한 일부 실시예에서, 미리정의된 인트라 예측 모드는 스크린 콘텐츠가 비-카메라 캡처된 콘텐츠임을 코딩된 정보가 나타내는 것에 응답하여 수평 예측 모드이다. 방법(2400)에 대한 일부 실시예에서, 규칙은, 루마 비디오 블록이 MIP 모드를 이용하여 코딩되어 있는 것에 응답하여 DM이 크로마 인트라 모드로서 이용되지 않음을 명시한다. 방법(2400)에 대한 일부 실시예에서, DM은 비트스트림 표현에서 시그널링되지 않는다. 방법(2400)에 대한 일부 실시예에서, 4개의 크로마 모드가 크로마 인트라 모드에 대해 허용되고, 루마 비디오 블록이 MIP 모드를 이용하여 코딩되어 있는 것에 응답하여 및 교차-성분 선형 모델 예측(CCLM)이 현재 크로마 비디오 블록에 대해 인에이블되어 있지 않음을 플래그가 나타내는 것에 응답하여 현재 크로마 비디오 블록에 대해 비트스트림 표현에서 시그널링된다. 방법(2400)에 대한 일부 실시예에서, 비트스트림 표현에서 크로마 인트라 모드를 시그널링하기 위해 최대 2개의 비트가 이용된다. 방법(2400)에 대한 일부 실시예에서, 7개의 크로마 모드가 크로마 인트라 모드에 대해 허용되고, 루마 비디오 블록이 MIP 모드를 이용하여 코딩되어 있는 것에 응답하여 및 교차-성분 선형 모델 예측(CCLM)이 현재 크로마 비디오 블록에 대해 인에이블되어 있음을 플래그가 나타내는 것에 응답하여 현재 크로마 비디오 블록에 대해 비트스트림 표현에서 시그널링된다. 방법(2400)에 대한 일부 실시예에서, 비트스트림 표현에서 크로마 인트라 모드를 시그널링하기 위해 최대 4개의 비트가 이용된다.

[1366] 도 25는 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 이 방법(2500)은, 비디오의 현재 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드로 코딩되어 있다고 결정하는 단계(2502) – 이 모드에서, 현재 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정됨 –; 상기 결정에 기초하여, 규칙에 기초한 최대 가능성 모드(MPM) 목록을 생성하는 단계(2504) – 상기 규칙은 상기 MPM 목록이 상기 현재 비디오 블록의 이웃 비디오 블록에 대해 생성된 또 다른 MPM 목록과는 독립적으로 생성됨을 명시함 –; 및 상기 MPM 목록을 이용하여 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계(2506)를 포함한다.

[1367] 방법(2500)에 대한 일부 실시예에서, MPM 목록은 미리정의된 최대 가능성 모드들을 포함한다. 방법(2500)에 대한 일부 실시예에서, 미리정의된 최대 가능성 모드들은 현재 비디오 블록의 치수들에 관계없이 {M0, M1, M2}를 포함한다. 방법(2500)에 대한 일부 실시예에서, 미리정의된 최대 가능성 모드는 현재 비디오 블록의 치수들에 기초한다. 방법(2500)에 대한 일부 실시예에서, 미리정의된 최대 가능성 모드는 현재 비디오 블록의 블록 크기 유형을 나타내는 식별자에 기초한다. 방법(2500)에 대한 일부 실시예에서, 블록 크기 유형을 나타내는 식별자는 음이 아닌 값이다. 방법(2500)에 대한 일부 실시예에서, 식별자는 0, 1, 또는 2를 포함한다. 방법(2500)에 대한 일부 실시예에서, 미리정의된 최대 가능성 모드들은 블록 크기 유형의 식별자가 소정의 값과 동일한 것에 응답하여 {17, 34, 5}이다. 방법(2500)에 대한 일부 실시예에서, 소정의 값은 0이다. 방법(2500)에 대한 일부 실시예에서, 미리정의된 최대 가능성 모드들은 블록 크기 유형의 식별자가 소정의 값과 동일한 것에 응답하여 {0, 7, 16}이다. 방법(2500)에 대한 일부 실시예에서, 소정의 값은 1이다. 방법(2500)에 대한 일부 실시예에서, 미리정의된 최대 가능성 모드들은 블록 크기 유형의 식별자가 소정의 값과 동일한 것에 응답하여 {1, 4, 6}이다. 방법(2500)에 대한 일부 실시예에서, 소정의 값은 2이다.

- [1368] 도 26은 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 이 방법(2600)은, 비디오의 현재 비디오 블록을 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 인코딩하는 단계(2602) – 이 모드에서, 현재 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정되고, 규칙은 상기 MIP 모드에서 이용되는 행렬의 수와 상기 현재 비디오 블록의 치수들 사이의 관계를 기술함 –; 상기 현재 비디오 블록의 인코딩된 표현을 상기 현재 비디오 블록의 비트스트림 표현에 추가하는 단계(2604)를 포함한다.
- [1369] 도 27은 비디오 처리를 위한 한 예시적인 방법의 플로차트를 도시한다. 이 방법(2700)은, 비디오의 현재 비디오 블록이 행렬 기반의 인트라 예측(MIP) 모드를 이용하여 비트스트림 표현으로 인코딩되어 있다고 결정하는 단계(2702) – 이 모드에서, 현재 비디오 블록의 예측 블록은, 상기 비디오의 이전에 코딩된 샘플들에 관해, 경계 다운샘플링 동작, 후속하는 행렬 벡터 곱셈 동작, 및 선택적으로 후속하는 업샘플링 동작을 수행함으로써 결정되고, 규칙은 상기 MIP 모드에서 이용되는 행렬의 수와 상기 현재 비디오 블록의 치수들 사이의 관계를 기술함 –; 및 상기 비트스트림 표현을 파싱 및 디코딩함으로써 상기 현재 비디오 블록의 디코딩된 표현을 생성하는 단계(2704)를 포함한다.
- [1370] 방법(들)(2600 및/또는 2700)에 대한 일부 실시예에서, 규칙은, 행렬의 수가 현재 비디오 블록의 치수들에 기초하지 않고, 행렬의 수는 이전에 저장되고, 행렬의 수가 11, 19 또는 35임을 명시한다. 방법(들)(2600 및/또는 2700)에 대한 일부 실시예에서, 규칙은, 행렬의 수가 현재 비디오 블록의 치수들에 기초하지 않음을 명시하고, MIP 모드에 대한 시그널링은 현재 비디오 블록에 대한 비트스트림 표현에 포함되고, MIP 모드에 대한 시그널링은 현재 비디오 블록의 치수들에 의존하지 않음을 명시한다. 방법(들)(2600 및/또는 2700)에 대한 일부 실시예에서, 규칙은, 행렬의 수가 현재 비디오 블록의 치수들에 기초함을 명시한다.
- [1371] 본 특허 문서에서 설명된 임의의 방법에 대한 일부 실시예에서, 변환을 수행하는 것은 비트스트림 표현이 현재 비디오 블록으로부터 생성되는 인코딩 동작을 포함한다. 본 특허 문서에서 설명된 임의의 방법에 대한 일부 실시예에서, 변환을 수행하는 것은 현재 비디오 블록이 비트스트림 표현으로부터 생성되는 디코딩 동작을 포함한다.
- [1372] 전술한 내용으로부터, 본 개시된 기술의 특정한 실시예들이 예시의 목적으로 여기서 설명되었지만, 본 발명의 범위를 벗어나지 않고 다양한 수정이 이루어질 수 있다는 것이 이해될 것이다. 따라서, 본 개시된 기술은 첨부된 청구항들에 의한 경우를 제외하고는 제한되지 않는다.
- [1373] 본 특허 문서에서 설명된 주제와 기능적 동작들의 구현은, 본 명세서에서 개시된 구조들 및 그들의 구조적 균등물을 포함한, 다양한 시스템들, 디지털 전자 회로, 컴퓨터 소프트웨어, 펌웨어, 또는 하드웨어로, 또는 이들 중 하나 이상의 조합으로 구현될 수 있다. 본 명세서에서 설명된 주제의 구현은, 하나 이상의 컴퓨터 프로그램 제품, 즉, 데이터 처리 장치에 의해 실행되거나 데이터 처리 장치의 동작을 제어하기 위한 유형의 및 비일시적인 컴퓨터 판독가능한 매체에 인코딩된 컴퓨터 프로그램 명령어들의 하나 이상의 모듈로서 구현될 수 있다. 컴퓨터 판독가능한 매체는, 머신 판독가능한 저장 디바이스, 머신 판독가능한 저장 기판, 메모리 디바이스, 머신 판독가능한 전파된 신호에 영향을 미치는 물질의 조성, 또는 이를 중 하나 이상의 조합일 수 있다. 용어 "데이터 처리 유닛 또는"데이터 처리 장치"는, 예로서 프로그램가능한 프로세서, 컴퓨터, 또는 복수의 프로세서나 컴퓨터를 포함한, 데이터를 처리하기 위한 모든 장치, 디바이스, 및 머신을 포함한다. 장치는, 하드웨어 외에도, 해당 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드, 예를 들어, 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 체제, 또는 이들 중 하나 이상의 조합으로 구성된 코드를 포함할 수 있다.
- [1374] (프로그램, 소프트웨어, 소프트웨어 애플리케이션, 스크립트, 또는 코드라고도 알려진) 컴퓨터 프로그램은, 컴파일형 또는 인터프리터형 언어를 포함한 임의의 형태의 프로그래밍 언어로 작성될 수 있고-, 단독형 프로그램이나, 모듈, 컴포넌트, 서브루틴, 또는 컴퓨팅 환경에서 이용하기에 적합한 기타의 유닛을 포함한, 임의 형태로 배치될 수 있다. 컴퓨터 프로그램은 반드시 파일 시스템 내의 파일에 대응할 필요는 없다. 프로그램은 다른 프로그램이나 데이터를 보유하고 있는 파일(예를 들어, 마크업 언어 문서에 저장된 하나 이상의 스크립트)의 일부로서, 해당 프로그램에 전용된 단일 파일에, 또는 복수의 조율된 파일들(예를 들어, 하나 이상의 모듈, 서브프로그램, 또는 코드의 부분들을 저장하고 있는 파일들)에 저장될 수 있다. 컴퓨터 프로그램은, 하나의 컴퓨터에서, 또는 적어도 하나의 사이트에 위치하거나 복수의 사이트에 걸쳐 분산되고 통신 네트워크를 통해 상호접속된 복수의 컴퓨터에서 실행되도록 배치될 수 있다.
- [1375] 본 명세서에서 설명된 프로세스들 및 로직 흐름들은, 하나 이상의 입력 데이터에 관해 동작하고 출력을 생성함으로써 하나 이상의 컴퓨터 프로그램을 실행하여 기능을 수행하는 하나 이상의 프로그램가능한 프로세서에 의해

수행될 수 있다. 프로세스들과 로직 흐름들은 또한, 특별 목적 로직 회로, 예를 들어, FPGA(필드 프로그래머블 게이트 어레이) 또는 ASIC(application specific integrated circuit; 주문형 집적 회로)에 의해 수행될 수 있고, 또한 장치가 이들로서 구현될 수 있다.

[1376] 컴퓨터 프로그램을 실행하기에 적합한 프로세서로는, 예로서, 범용 마이크로프로세서 및 특별 목적 마이크로프로세서 양쪽 모두, 및 임의의 종류의 디지털 컴퓨터의 임의의 하나 이상의 프로세서가 포함된다. 일반적으로, 프로세서는 판독 전용 메모리나 랜덤 액세스 메모리 또는 양쪽 모두로부터 명령어와 데이터를 수신할 것이다. 컴퓨터의 본질적 요소들은, 명령어를 수행하기 위한 프로세서와, 명령어 및 데이터를 저장하기 위한 하나 이상의 메모리 디바이스이다. 일반적으로, 컴퓨터는 또한, 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스, 예를 들어, 자기, 광자기 디스크, 또는 광 디스크를 포함하거나, 이들로부터 데이터를 수신, 이들에게 데이터를 전송, 또는 양쪽 모두를 위해 동작적으로 결합될 수 있다. 그러나, 컴퓨터는 이러한 디바이스를 가질 필요는 없다. 컴퓨터 프로그램 명령어 및 데이터를 저장하기에 적합한 컴퓨터 판독가능한 매체는, 예로서 반도체 메모리 디바이스들, 예를 들어 EPROM, EEPROM 및 플래시 메모리 디바이스를 포함한, 모든 형태의 비휘발성 메모리, 매체 및 메모리 디바이스를 포함한다. 프로세서와 메모리는 특별 목적 로직 회로에 의해 보충되거나, 이에 병합될 수 있다.

[1377] 도면들과 함께, 본 명세서는, 단지 예시인 것으로 간주되어야 하며, 여기서, 예시란 예를 의미한다. 명세서에서 사용될 때 "또는"의 사용은, 문맥상 명시적으로 달리 나타내지 않는 한, "및/또는"을 포함하는 것을 의도한다.

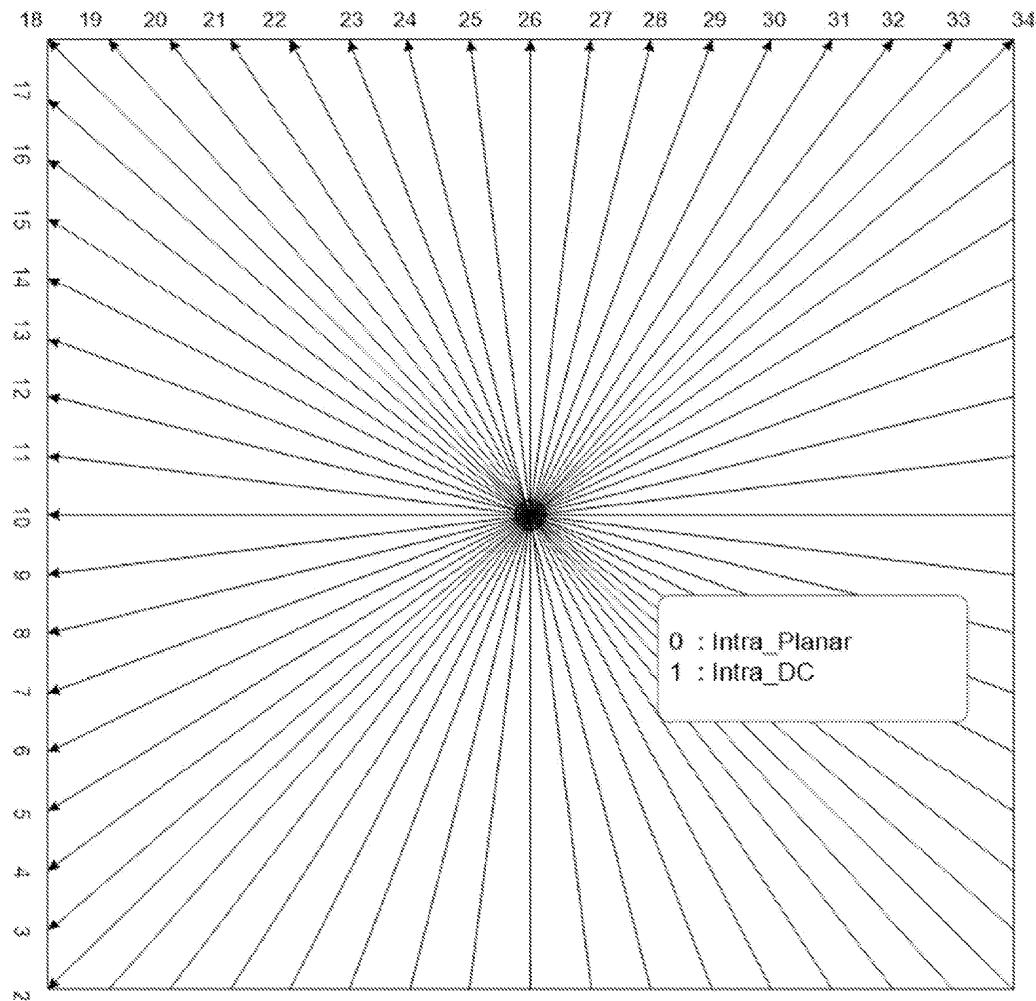
[1378] 본 특허 문서가 많은 특정한 사항들을 포함하지만, 이들은 본 발명의 범위의 또는 청구 범위에 관한 제한으로서 해석되어서는 안 되고, 특정한 발명의 특정한 실시예를 고유의 피처들의 설명으로서 해석되어야 한다. 별개의 실시예들의 정황에서 본 특허 문서에서 설명된 소정 피처들은 또한, 단일의 실시예에서 조합하여 구현될 수 있다. 역으로, 단일 실시예의 정황에서 설명된 다양한 피처들은 또한, 복수의 실시예에서 별개로 구현되거나 임의의 적절한 하위조합(subcombination)으로 구현될 수 있다. 게다가, 피처들이 상기에서 소정 조합으로 작용하는 것으로 설명되거나 심지어 그와 같이 처음 청구되더라도, 청구된 조합으로부터의 하나 이상의 피처들은 일부 경우에는 그 조합으로부터 삭제될 수도 있고, 청구된 조합은 하위조합이나 하위조합의 변형에 관한 것일 수도 있다.

[1379] 유사하게, 동작들이 도면에서 특정 순서로 도시되더라도, 이것은, 바람직한 결과를 달성하기 위해 이러한 동작들이 도시된 특정 순서로 또는 순차적 순서로 수행될 것을 요구하거나, 도시된 모든 동작들이 수행될 것을 요구하는 것으로 이해되어서는 안 된다. 또한, 본 특허 문헌에서 설명된 실시예들에서의 다양한 시스템 컴포넌트들의 분리는 모든 실시예에서 이러한 분리를 요구하는 것으로 이해되어서는 안 된다.

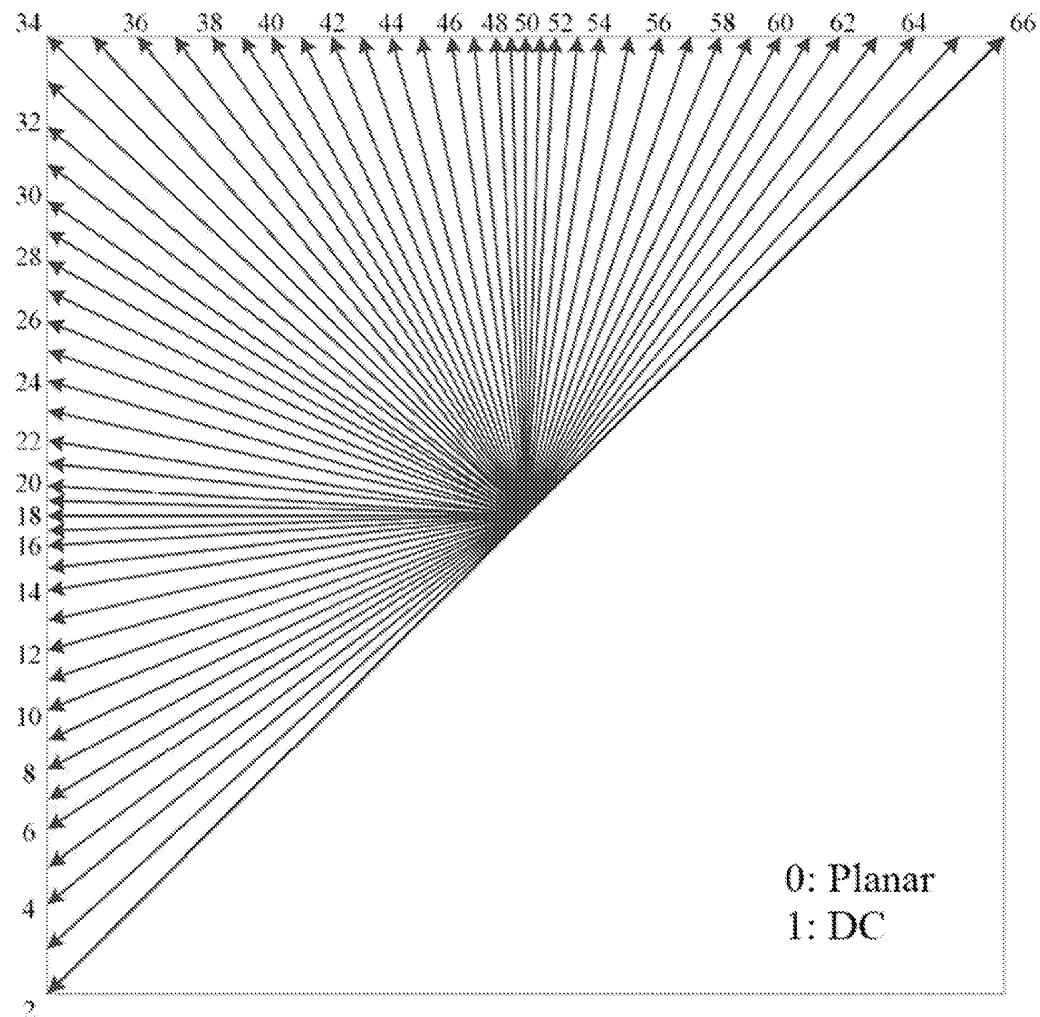
[1380] 단지 몇 개의 구현 및 예들만이 설명되었지만, 본 특허 문서에서 설명되고 예시된 내용에 기초하여 다른 구현, 개선 및 변형이 이루어질 수 있다.

도면

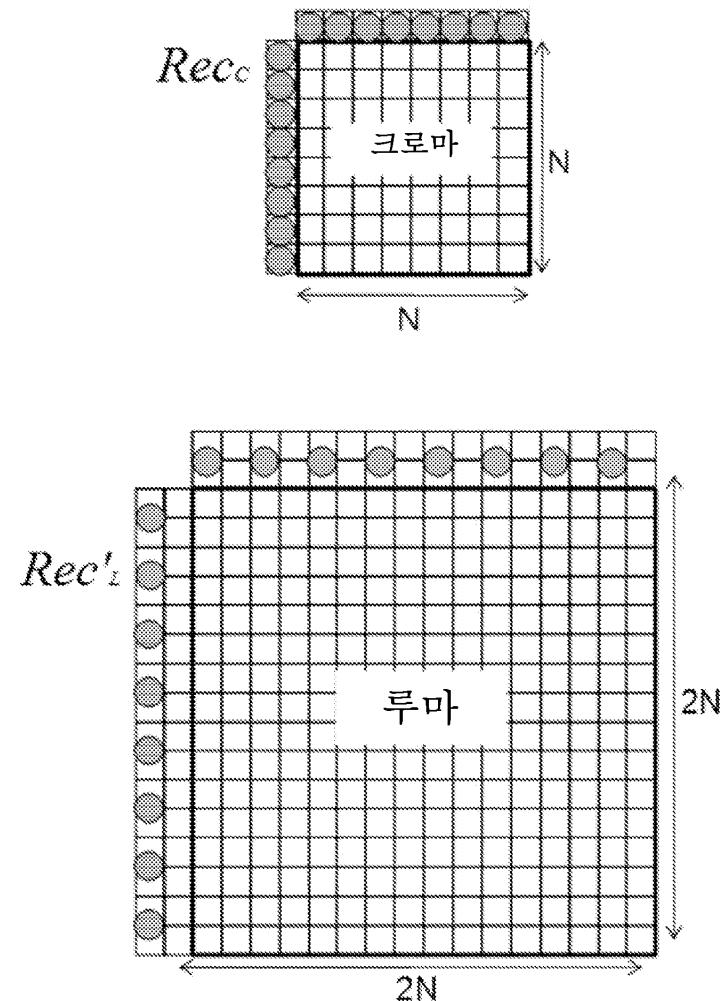
도면1



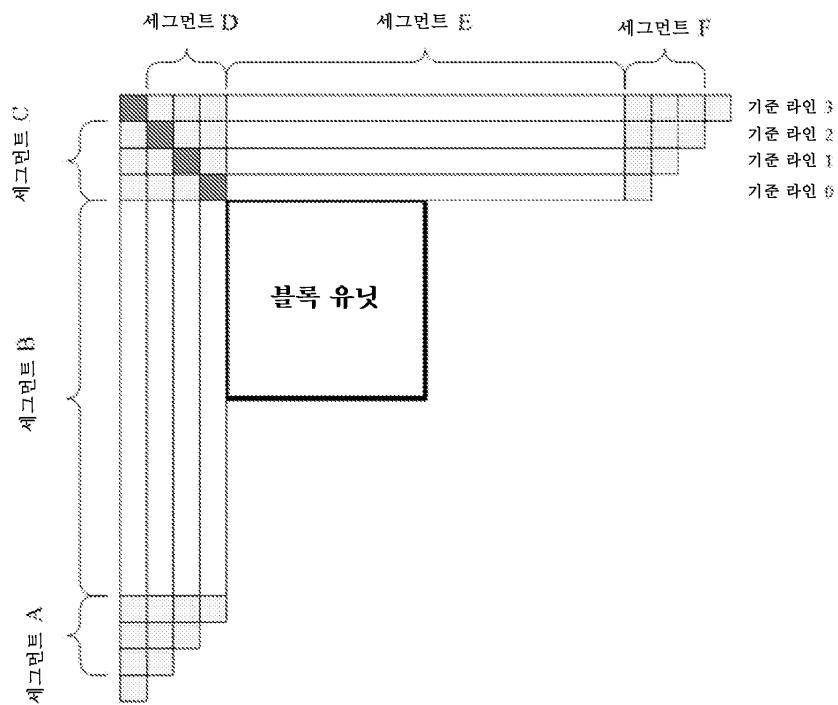
도면2



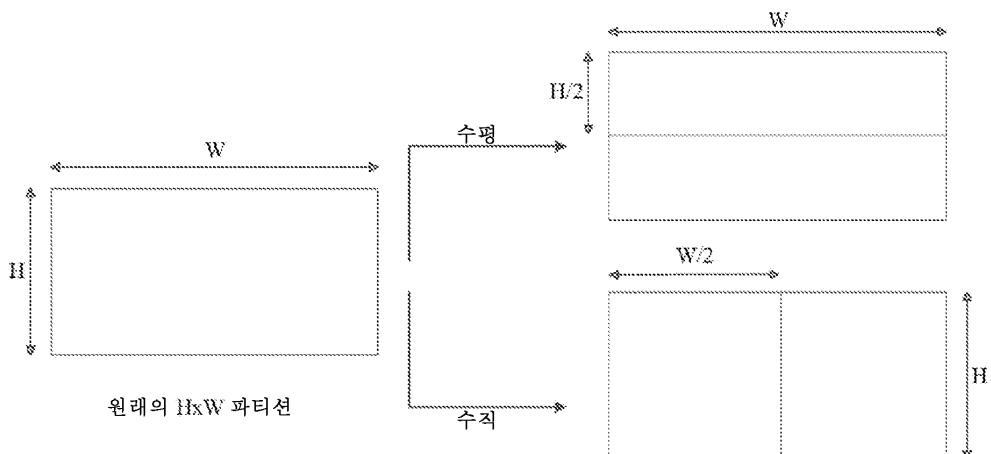
도면3



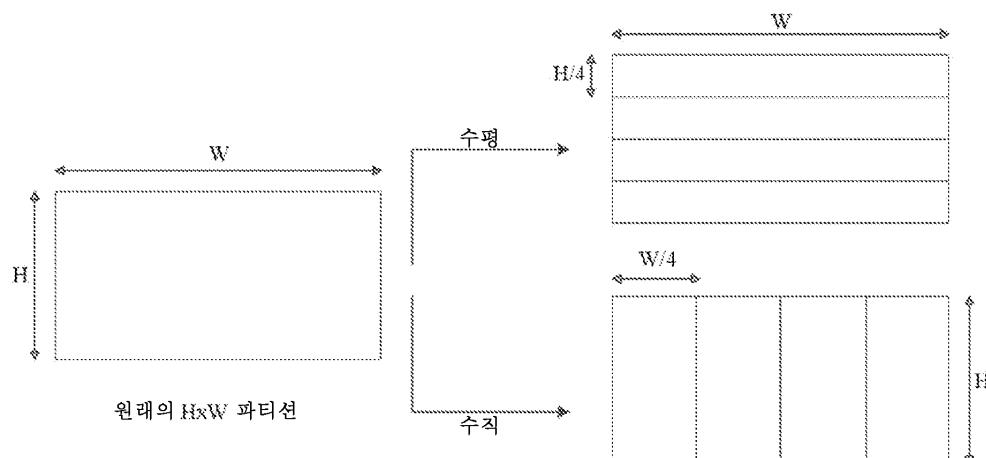
## 도면4



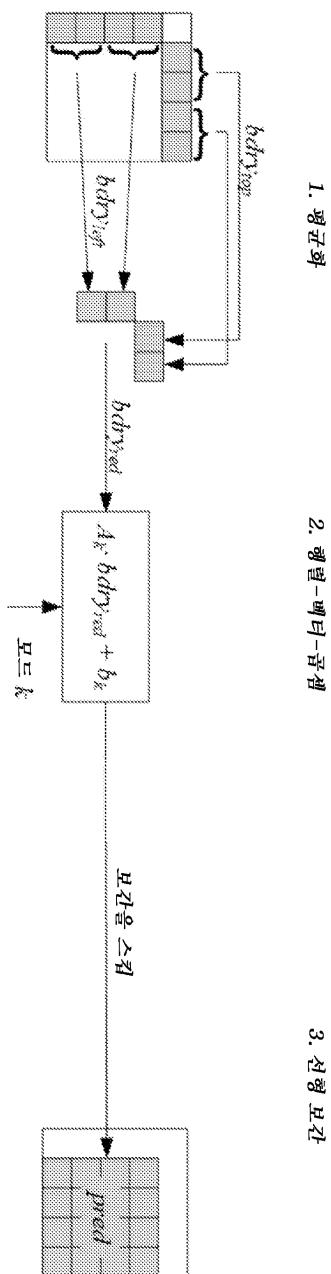
## 도면5a



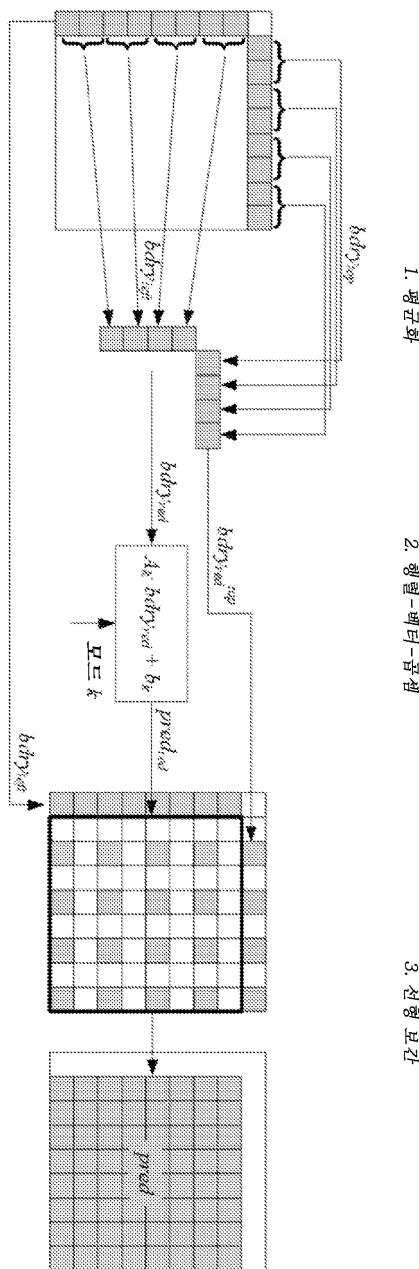
## 도면5b



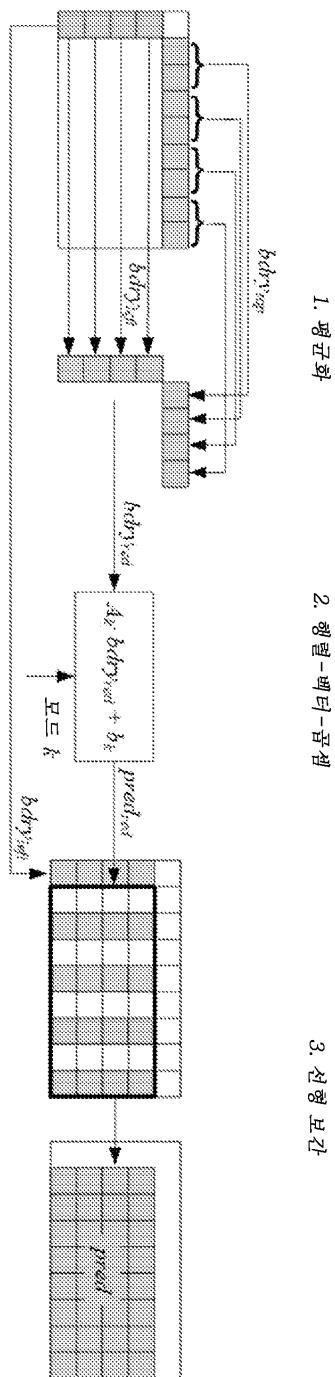
도면6



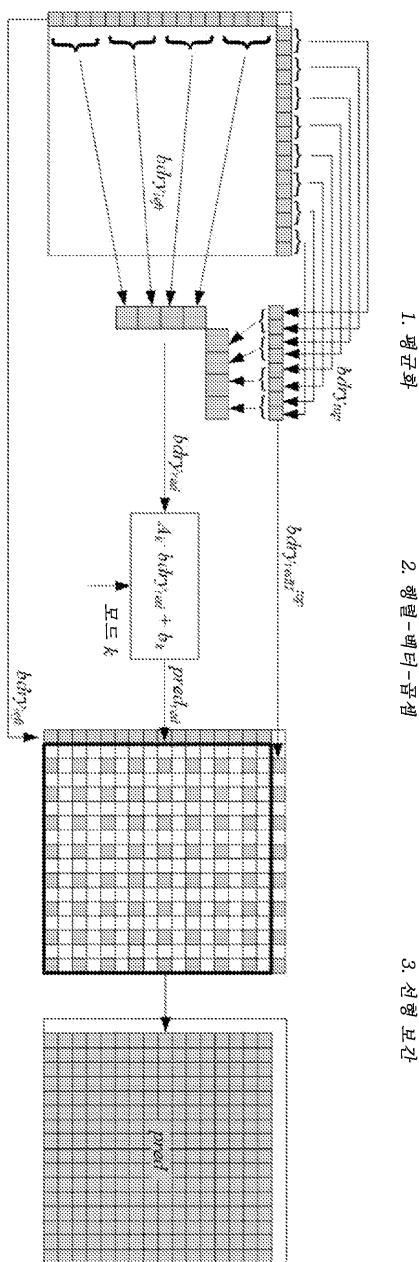
도면7

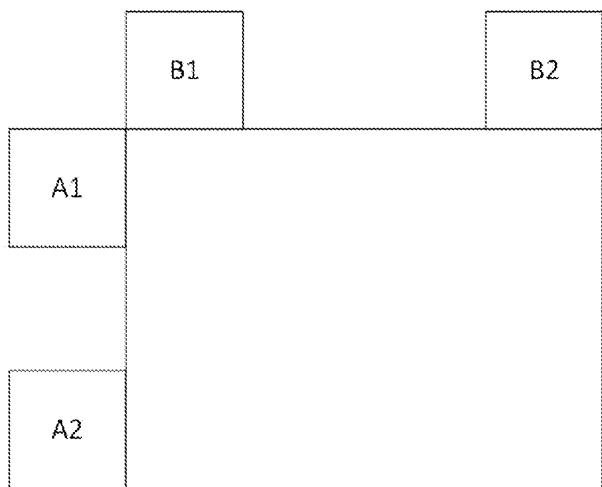
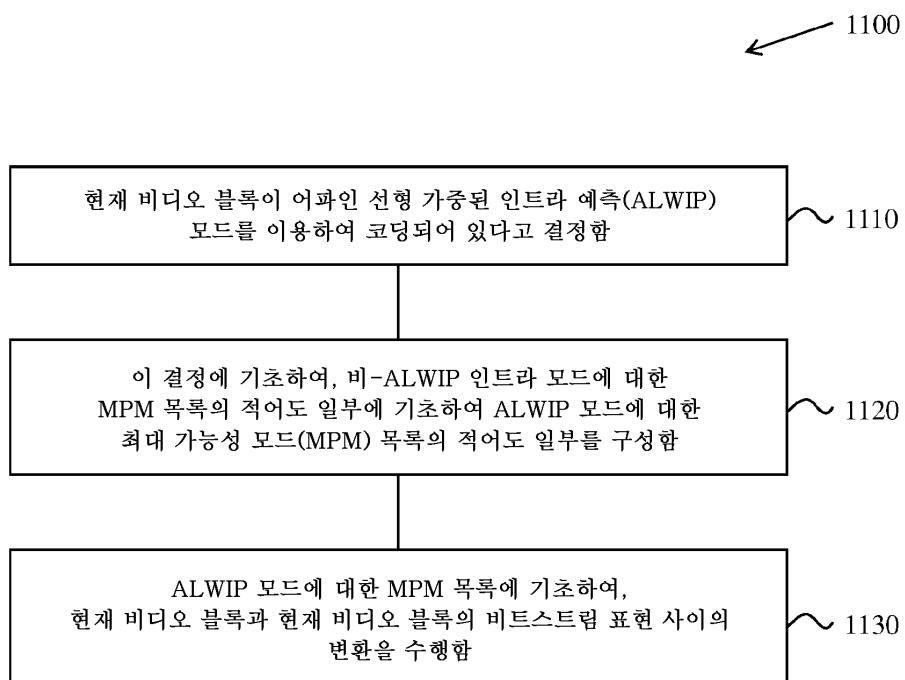


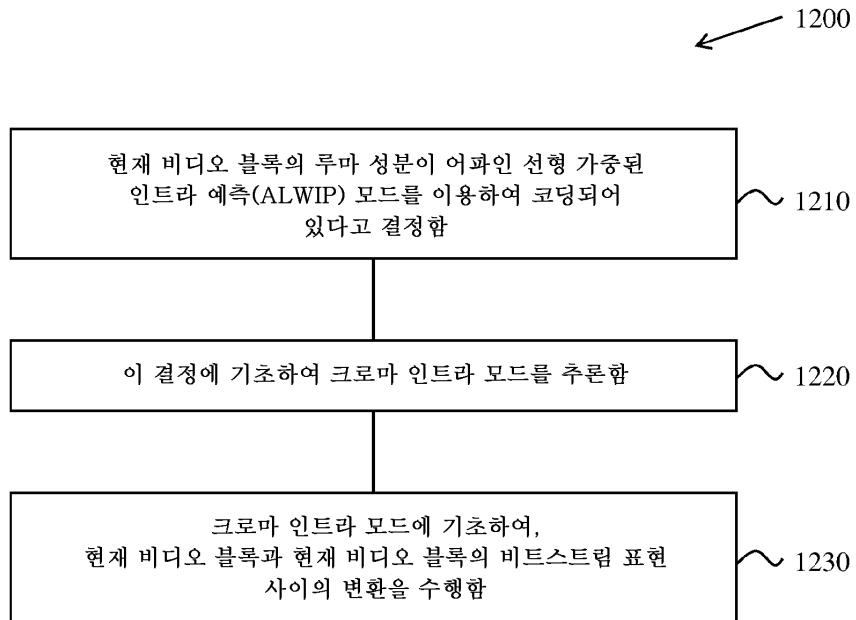
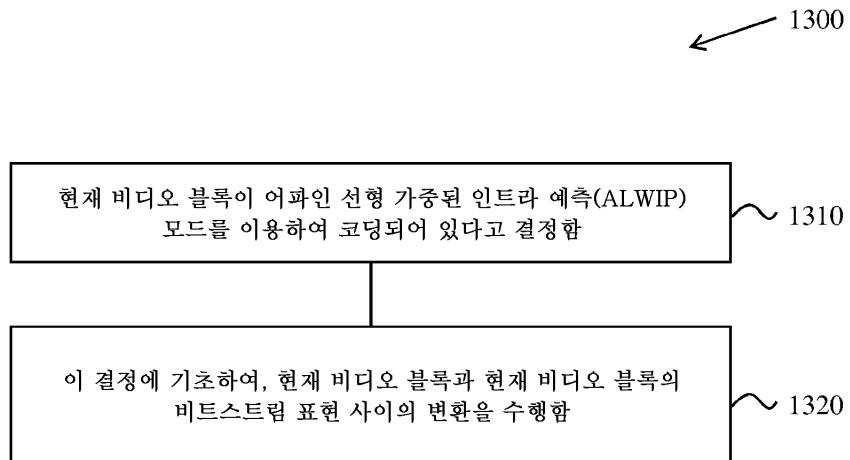
도면8

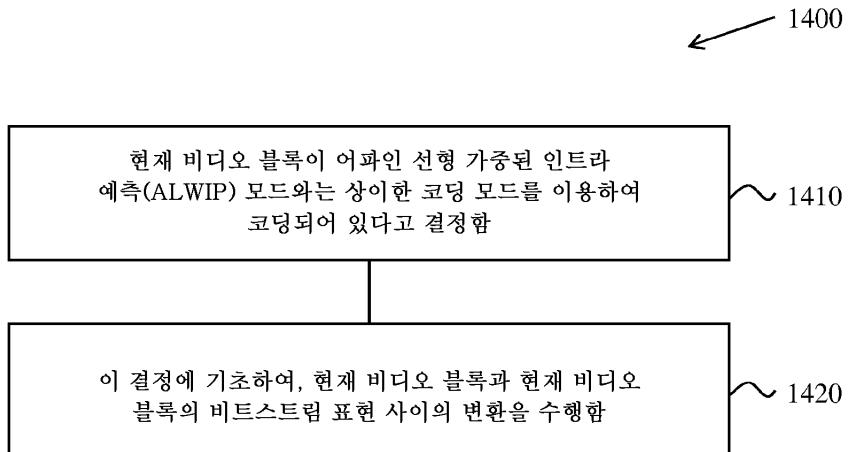
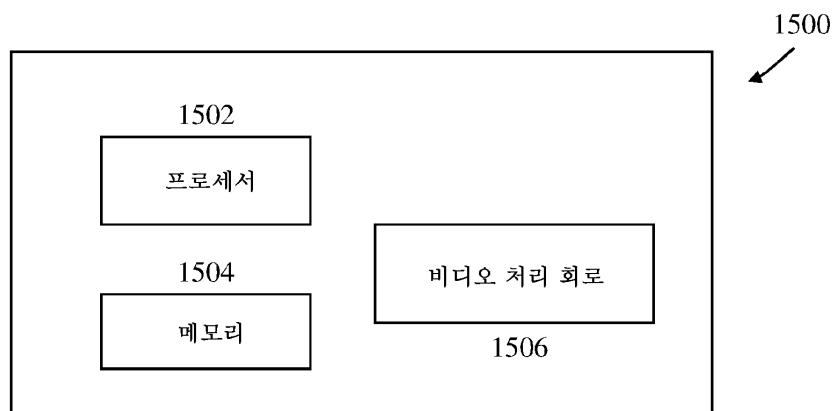


도면9

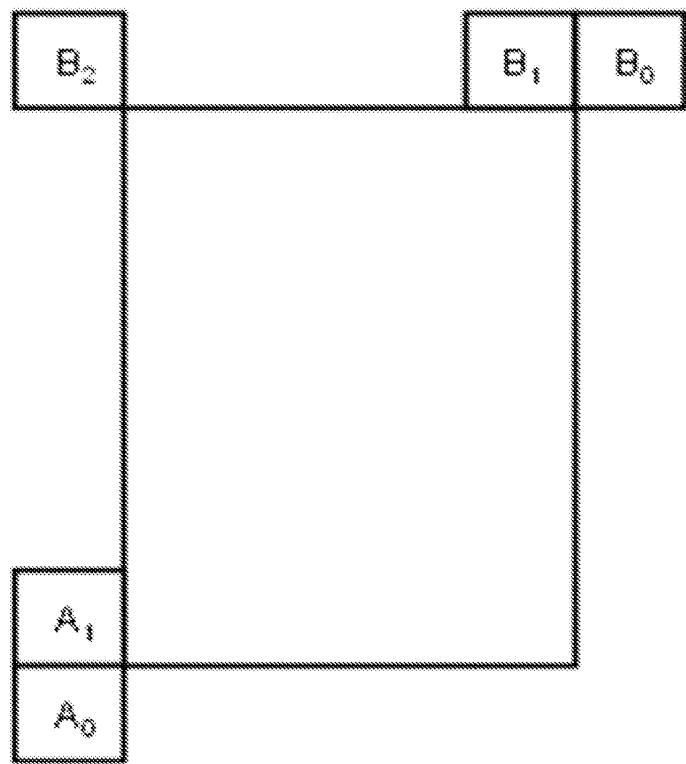


**도면10****도면11**

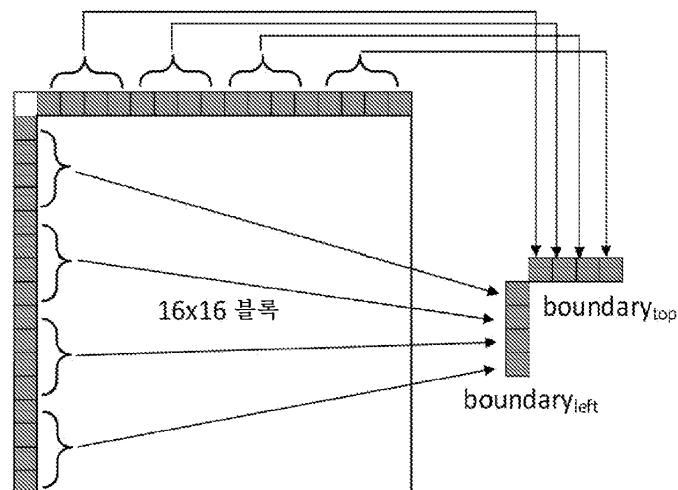
**도면12****도면13**

**도면14****도면15**

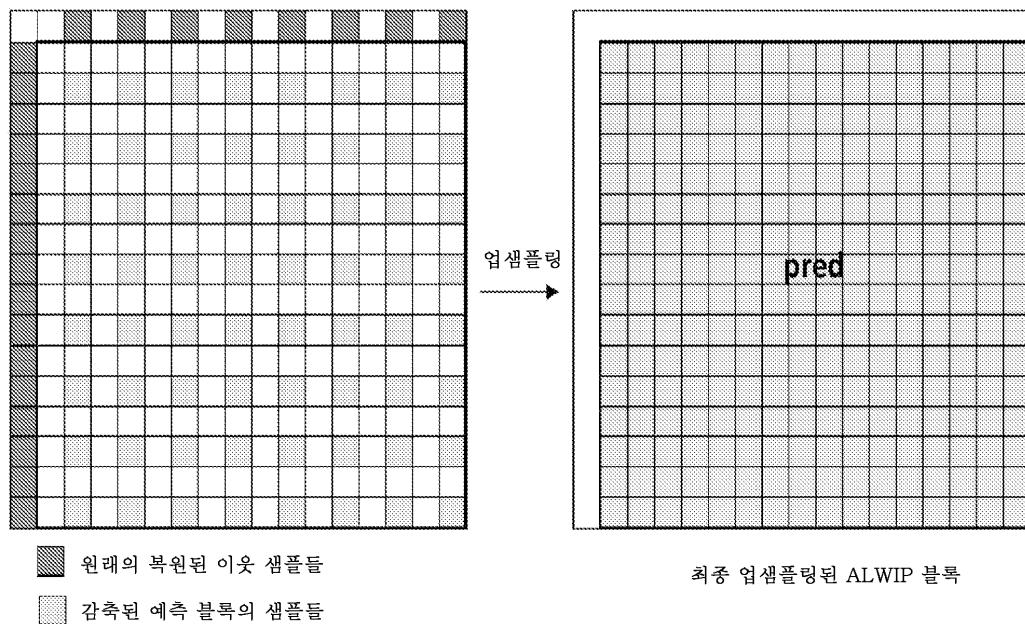
도면16



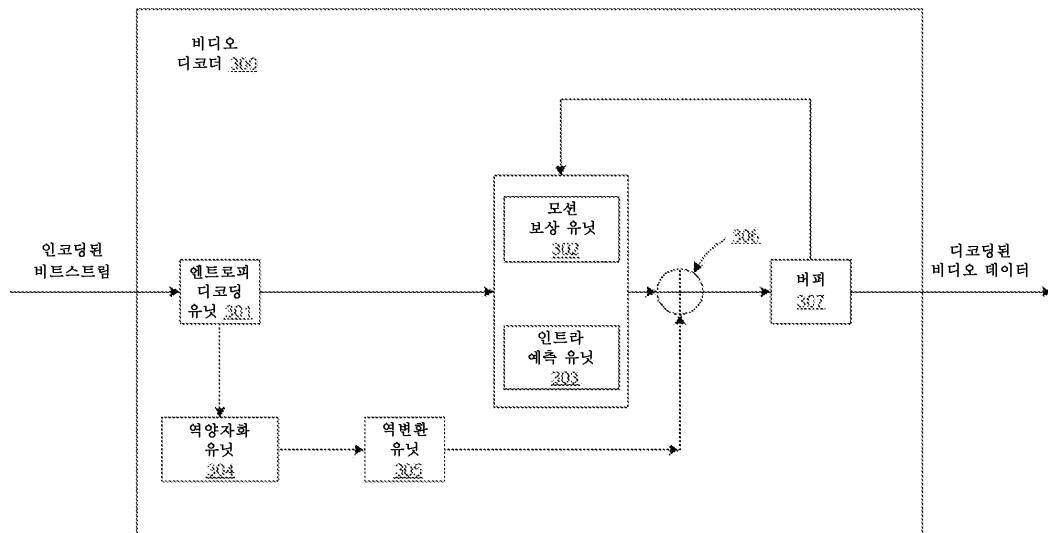
도면17



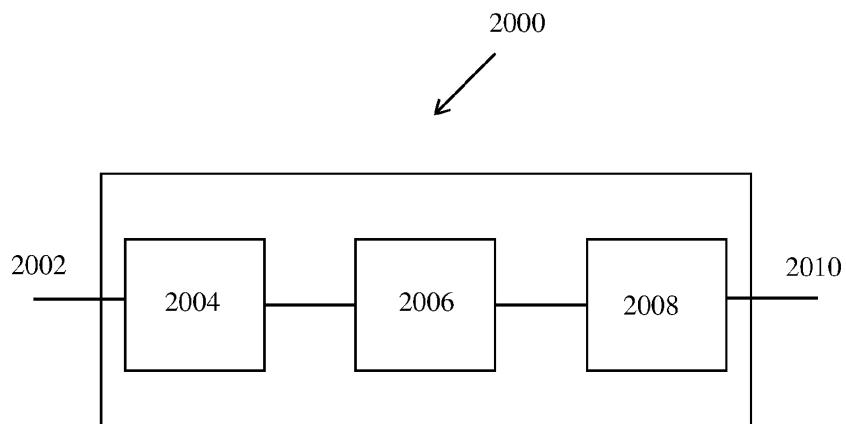
도면18



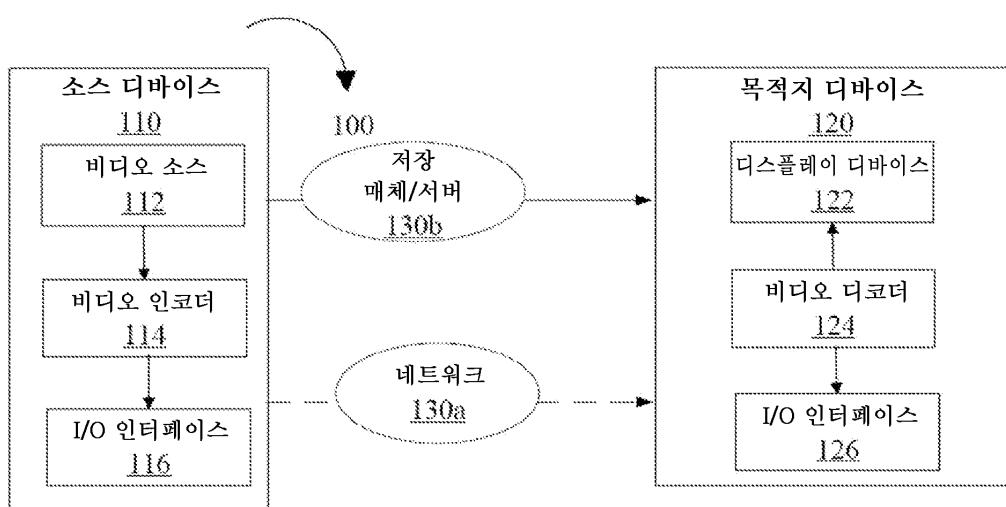
도면19



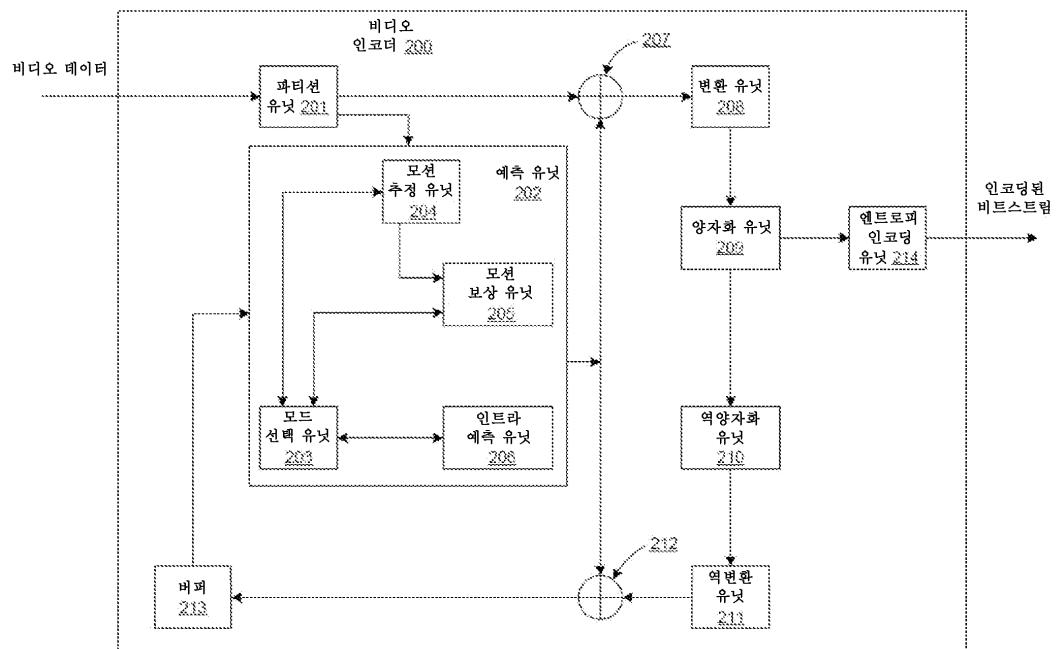
도면20



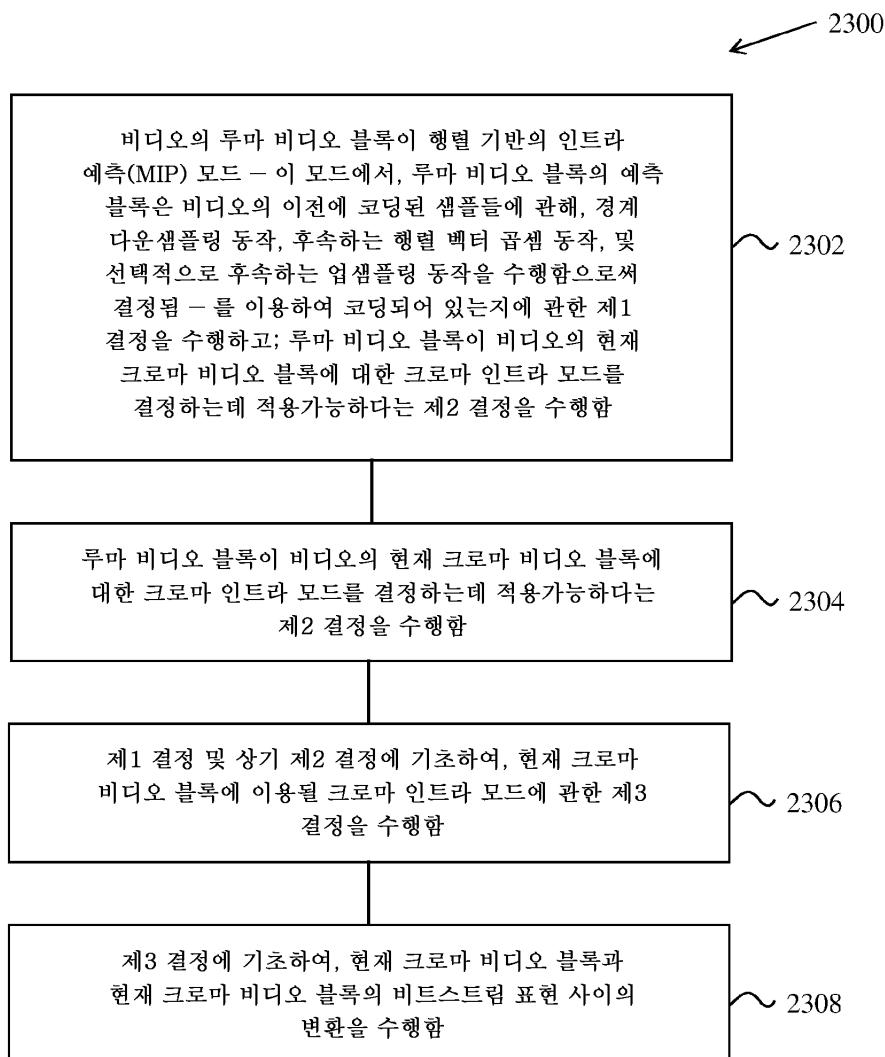
도면21



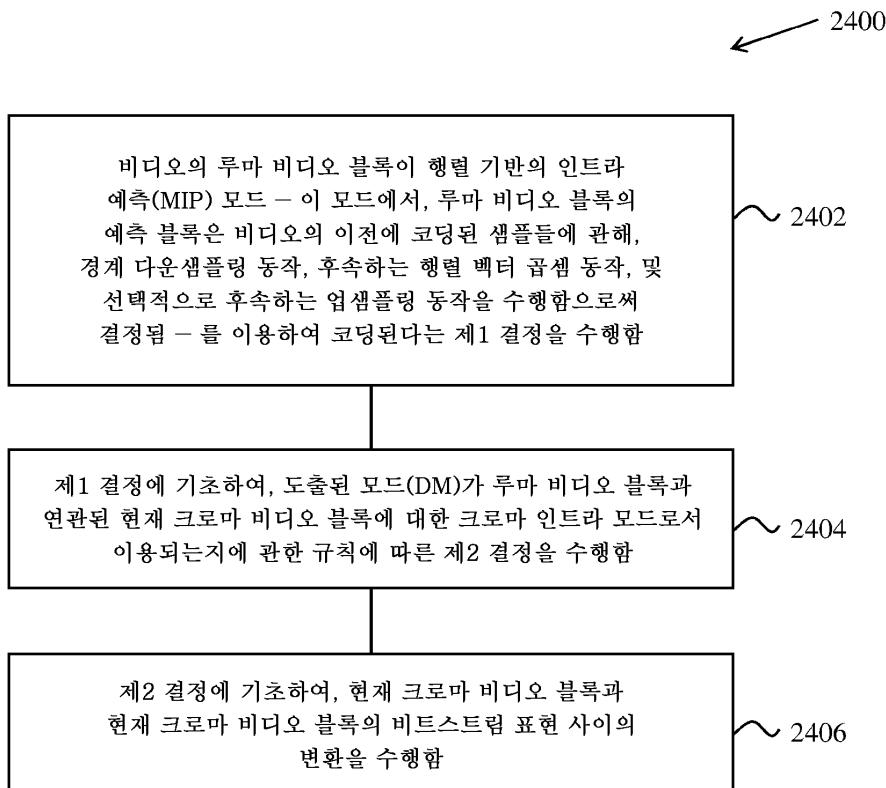
## 도면22



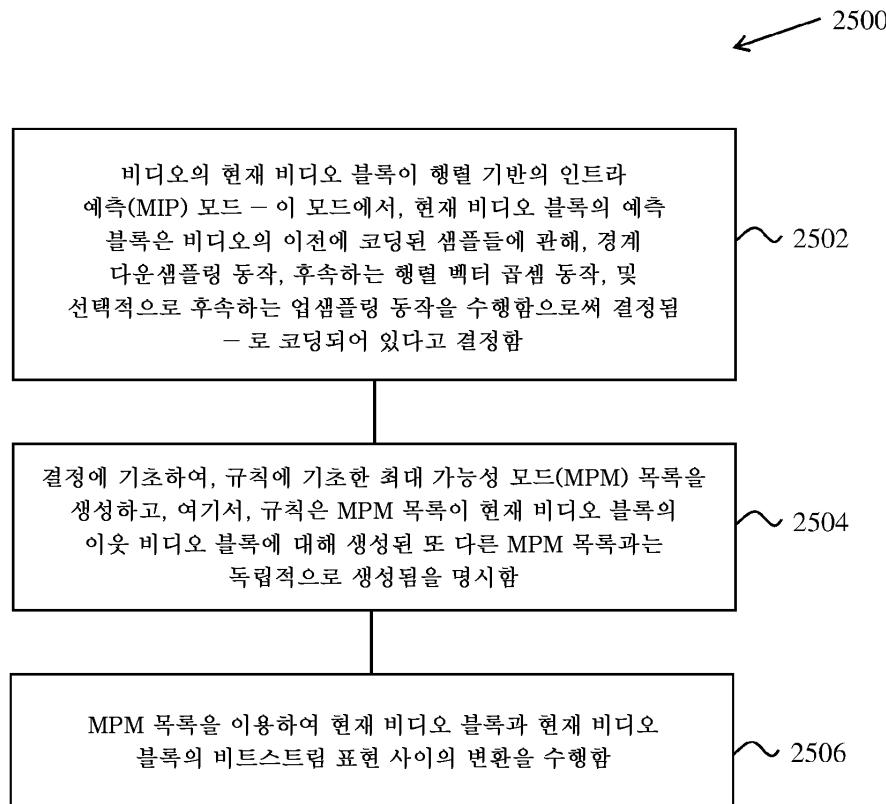
## 도면23



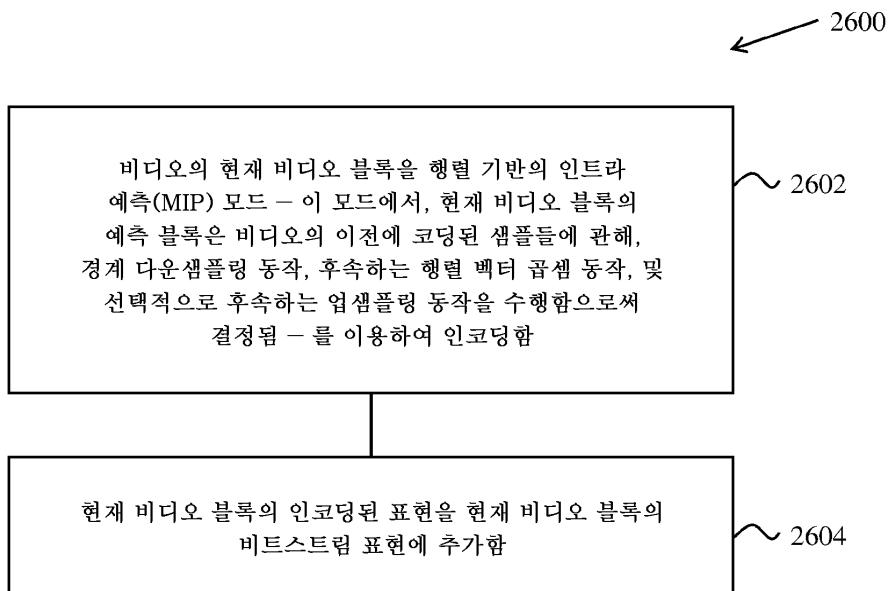
## 도면24



## 도면25



## 도면26



도면27

