

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2020 년 4 월 16 일 (16.04.2020) WIPO | PCT



(10) 국제공개번호
WO 2020/076036 A1

(51) 국제특허분류:

H04N 19/11 (2014.01) H04N 19/119 (2014.01)
H04N 19/70 (2014.01) H04N 19/105 (2014.01)
H04N 19/593 (2014.01) H04N 19/186 (2014.01)
H04N 19/176 (2014.01) H04N 19/82 (2014.01)

(21) 국제출원번호: PCT/KR2019/013137

(22) 국제출원일: 2019 년 10 월 7 일 (07.10.2019)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:

10-2018-0119455 2018 년 10 월 7 일 (07.10.2018) KR
10-2018-0121530 2018 년 10 월 12 일 (12.10.2018) KR
10-2018-0126681 2018 년 10 월 23 일 (23.10.2018) KR
10-2018-0137708 2018 년 11 월 9 일 (09.11.2018) KR
10-2019-0020241 2019 년 2 월 21 일 (21.02.2019) KR
10-2019-0045260 2019 년 4 월 18 일 (18.04.2019) KR
10-2019-0052222 2019 년 5 월 3 일 (03.05.2019) KR

(71) 출원인: 주식회사 윌러스표준기술연구소 (WILUS INSTITUTE OF STANDARDS AND TECHNOLOGY INC.) [KR/KR]; 13595 경기도 성남시 분당구 황새울로 216, 5층, Gyeonggi-do (KR). (주)휴맥스 (HUMAX CO.,

LTD.) [KR/KR]; 17040 경기도 용인시 처인구 영문로 2, Gyeonggi-do (KR).

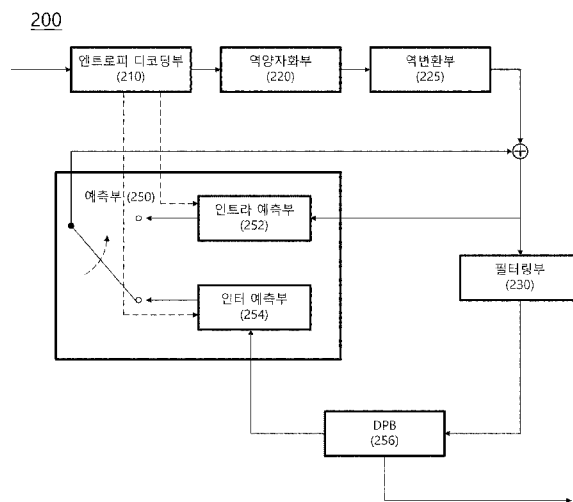
(72) 발명자: 김동철 (KIM, Dongcheol); 16323 경기도 수원시 장안구 하พล로46번길 17, 307동 506호, Gyeonggi-do (KR). 고건중 (KO, Geonjung); 06514 서울시 서초구 잠원로 157, 120동 1006호, Seoul (KR). 정재홍 (JUNG, Jaehong); 04775 서울시 성동구 돌레11길 1, 101동 1704호, Seoul (KR). 손주형 (SON, Juhung); 16024 경기도 의왕시 내손순환로 7, 308동 1402호, Gyeonggi-do (KR). 박진삼 (KWAK, Jinsam); 14071 경기도 안양시 동안구 귀인로 213, 102동 1704호, Gyeonggi-do (KR).

(74) 대리인: 임국일 (LIM, Kukil); 13595 경기도 성남시 분당구 황새울로 216, 5층, Gyeonggi-do (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD,

(54) Title: METHOD AND DEVICE FOR PROCESSING VIDEO SIGNAL USING MPM CONFIGURATION METHOD FOR MULTIPLE REFERENCE LINES

(54) 발명의 명칭: 복수개의 참조 라인을 위한 MPM 구성 방법을 이용한 비디오 신호 처리 방법 및 장치



210 ... Entropy decoding unit
220 ... Dequantization unit
225 ... Inverse conversion unit
230 ... Filtering unit
250 ... Prediction unit
252 ... Intra prediction unit
254 ... Inter prediction unit

(57) Abstract: Disclosed are a video signal processing method and device for encoding or decoding video signals. The video signal processing method comprises the steps of: obtaining information, from a received bitstream, indicating whether an MPM is used in a current block; when at least one of the intra prediction mode of the left block of the current block or the intra prediction mode of the upper block of the current block is an angular mode, generating an MPM list using only the angular mode; determining the intra prediction mode on the basis of the MPM list; and predicting the current block on the basis of the determined intra prediction mode.

(57) 요약서: 비디오 신호를 인코딩하거나 디코딩하는 비디오 신호 처리 방법 및 장치가 개시된다. 비디오 신호 처리 방법은 수신된 비트스트림으로부터 현재 블록의 MPM 사용 여부를 나타내는 정보를 획득하는 단계, 현재 블록의 좌측 블록의 인트라 예측 모드 또는 현재 블록의 상측 블록의 인트라 예측 모드 중 적어도 하나가 각도 모드인 경우, 각도 모드만을 이용하여 MPM 리스트를 생성하는 단계 MPM 리스트에 기초하여 인트라 예측 모드를 결정하는 단계 및 결정된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 포함하는 것을 특징으로 한다.



SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역
내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE,
LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM,
ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유
럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK,
MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI
(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML,
MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제21조(3))

명세서

발명의 명칭: 복수개의 참조 라인을 위한 MPM 구성 방법을 이용한 비디오 신호 처리 방법 및 장치

기술분야

- [1] 본 발명은 비디오 신호의 처리 방법 및 장치에 관한 것으로, 보다 상세하게는 비디오 신호를 인코딩하거나 디코딩하는 비디오 신호 처리 방법 및 장치에 관한 것이다.

배경기술

- [2] 압축 부호화란 디지털화한 정보를 통신 회선을 통해 전송하거나, 저장 매체에 적합한 형태로 저장하기 위한 일련의 신호 처리 기술을 의미한다. 압축 부호화의 대상에는 음성, 영상, 문자 등의 대상이 존재하며, 특히 영상을 대상으로 압축 부호화를 수행하는 기술을 비디오 영상 압축이라고 일컫는다. 비디오 신호에 대한 압축 부호화는 공간적인 상관관계, 시간적인 상관관계, 확률적인 상관관계 등을 고려하여 잉여 정보를 제거함으로써 이루어진다. 그러나 최근의 다양한 미디어 및 데이터 전송 매체의 발전으로 인해, 더욱 고효율의 비디오 신호 처리 방법 및 장치가 요구되고 있다.

발명의 상세한 설명

기술적 과제

- [3] 본 발명의 목적은 비디오 신호의 코딩 효율을 높이하고자 함에 있다. 구체적으로, 본 발명은 변환 블록에 적합한 변환 커널을 사용하여 코딩 효율을 높이기 위한 목적을 가지고 있다.

과제 해결 수단

- [4] 상기와 같은 과제를 해결하기 위해, 본 발명은 다음과 같은 비디오 신호 처리 장치 및 비디오 신호 처리 방법을 제공한다.
- [5] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법은 수신된 비트스트림으로부터 현재 블록의 참조 샘플 라인 인덱스를 획득하는 단계, 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계, 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 획득하는 단계, planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 인트라 예측 모드를 결정하는 단계 및 결정된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 포함하는 것을 특징으로 한다.
- [6] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 참조 샘플 라인 인덱스가 0이 아닌 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 획득하지 않고, planar 모드를 사용하는지 여부를 나타내는 정보가 비트스트림으로부터 획득되지 않는 경우,

planar 모드를 사용하는지 여부를 나타내는 정보는 planar 모드를 사용하지 않음을 나타내는 것을 특징으로 한다.

- [7] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법은 현재 블록이 코딩 트리 유닛의 최상측에 포함되는지 여부를 결정하는 단계 및 현재 블록이 코딩 트리 유닛의 최상측에 포함되지 않은 경우, 참조 샘플 라인 인덱스를 비트스트림으로부터 획득하는 단계를 포함하는 것을 특징으로 한다.
- [8] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법은 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계, 참조 샘플 라인 인덱스가 0인 경우, MPM(Most Probable Modes) 사용여부를 나타내는 정보를 비트스트림으로부터 획득하는 단계 및 MPM을 사용여부를 나타내는 정보가 MPM을 사용함을 나타내고, 참조 샘플 라인 인덱스가 0인 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 획득하는 단계를 포함하는 것을 특징으로 한다.
- [9] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법은 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하는 것을 나타내는지 여부를 결정하는 단계, planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하지 않는 것을 나타내는 경우, MPM인덱스를 비트스트림으로부터 획득하는 단계, 현재 블록의 미리 복원된 좌측 블록 또는 미리 복원된 상측 블록 중 적어도 하나에 기초하여, 적어도 하나의 각도 모드를 포함하는 MPM리스트를 생성하는 단계, MPM인덱스에 기초하여 MPM리스트에서 하나의 인트라 예측 모드를 선택하는 단계 및 선택된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 포함하는 것을 특징으로 한다.
- [10] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 장치는 프로세서 및 메모리를 포함하고, 프로세서는 메모리에 저장된 명령어에 기초하여, 수신된 비트스트림으로부터 현재 블록의 참조 샘플 라인 인덱스를 획득하는 단계, 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계, 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 획득하는 단계, planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 인트라 예측 모드를 결정하는 단계 및 결정된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 수행하는 것을 특징으로 한다.
- [11] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 장치는 프로세서는 메모리에 저장된 명령어에 기초하여, 참조 샘플 라인 인덱스가 0이 아닌 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 획득하지 않고, 현재 블록의 planar 모드를 사용하는지 여부를 나타내는 정보가 비트스트림으로부터 획득되지 않는 경우, planar 모드를

- 사용하는지 여부를 나타내는 정보는 planar 모드를 사용하지 않음을 나타내는 것을 특징으로 한다.
- [12] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 장치는 프로세서는 메모리에 저장된 명령어에 기초하여, 현재 블록이 코딩 트리 유닛의 최상측에 포함되는지 여부를 결정하는 단계 및 현재 블록이 코딩 트리 유닛의 최상측에 포함되지 않은 경우, 참조 샘플 라인 인덱스를 비트스트림으로부터 획득하는 단계를 수행하는 것을 특징으로 한다.
- [13] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 장치는 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계, 참조 샘플 라인 인덱스가 0인 경우, MPM(Most Probable Modes) 사용여부를 나타내는 정보를 비트스트림으로부터 획득하는 단계 및 MPM을 사용여부를 나타내는 정보가 MPM을 사용함을 나타내고, 참조 샘플 라인 인덱스가 0인 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 획득하는 단계를 수행하는 것을 특징으로 한다.
- [14] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 장치는 프로세서는 메모리에 저장된 명령어에 기초하여, planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하는 것을 나타내는지 여부를 결정하는 단계, planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하지 않는 것을 나타내는 경우, MPM인덱스를 비트스트림으로부터 획득하는 단계, 현재 블록의 미리 복원된 좌측 블록 또는 미리 복원된 상측 블록 중 적어도 하나에 기초하여, 적어도 하나의 각도 모드를 포함하는 MPM리스트를 생성하는 단계, MPM인덱스에 기초하여 MPM리스트에서 하나의 인트라 예측 모드를 선택하는 단계 및 선택된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 수행하는 것을 특징으로 한다.
- [15] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 인코딩하는 방법은, 인트라 예측에 기초하여 비디오 신호를 인코딩하는 방법에 있어서, 현재 블록이 MPM(Most Probable Modes)을 사용하고, 현재 블록의 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보를 생성하는 단계 및 planar 모드를 사용하는지 여부를 나타내는 정보를 포함하는 비트스트림을 생성하는 단계를 포함하는 것을 특징으로 한다.
- [16] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 인코딩하는 장치는, 프로세서 및 메모리를 포함하고, 프로세서는 메모리에 저장된 명령어에 기초하여, 현재 블록이 MPM(Most Probable Modes)을 사용하고, 현재 블록의 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보를 생성하는 단계 및 planar 모드를 사용하는지 여부를 나타내는 정보를 포함하는 비트스트림을 생성하는 단계를 수행하는 것을 특징으로 한다.
- [17] 비밀시적 컴퓨터가 읽을 수 있는 매체에 저장된 비디오 비트스트림을 포함하는

장치에 있어서, 비트스트림은 현재 블록이 MPM(Most Probable Modes)을 사용하고, 현재 블록의 참조 샘플 라인 인덱스가 0인 경우, 인코더에 의해 생성된 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 생성되는 것을 특징으로 한다.

- [18] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법은 수신된 비트스트림으로부터 현재 블록의 MPM 사용여부를 나타내는 정보를 획득하는 단계, 현재 블록의 좌측 블록의 인트라 예측 모드 또는 현재 블록의 상측 블록의 인트라 예측 모드 중 적어도 하나가 각도 모드인 경우, 각도 모드만을 이용하여 MPM리스트를 생성하는 단계, MPM리스트에 기초하여 인트라 예측 모드를 결정하는 단계 및 결정된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 포함하는 것을 특징으로 한다.
- [19] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 MPM리스트를 생성하는 단계는 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 동일하고, 좌측 블록의 인트라 예측 모드가 각도 모드인 경우, 좌측 블록의 인트라 예측 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드 및 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.
- [20] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 MPM리스트를 생성하는 단계는, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 동일하지 않고, 좌측 블록의 인트라 예측 모드 또는 상측 블록의 인트라 예측 모드 중 어느 하나만 각도 모드인 경우, 각도 모드인 블록의 인트라 예측 모드, 각도 모드인 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 각도 모드인 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 각도 모드인 블록의 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드 및 각도 모드인 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.
- [21] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 MPM리스트를 생성하는 단계는, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 동일하지 않고, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 각도 모드인 경우, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 크기 비교에 기초하여 MPM리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.

- [22] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 **MPM**리스트를 생성하는 단계는, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드를 **MPM**리스트에 포함하는 단계 및 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 상측 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 상측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스에 대응되는 각도 모드, 상측 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드, 또는 상측 블록의 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드 중 적어도 하나의 인트라 예측 모드에 기초하여 **MPM**리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.
- [23] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 **MPM**리스트를 생성하는 단계는, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 1과 같은 경우, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 **MPM**리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.
- [24] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 **MPM**리스트를 생성하는 단계는, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 62보다 크거나 같은 경우, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 2만큼 큰 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 **MPM**리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.
- [25] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 **MPM**리스트를 생성하는 단계는, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 2와 같은 경우, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은

인덱스보다 1 만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1 만큼 작은 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1 만큼 큰 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.

- [26] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 MPM리스트를 생성하는 단계는, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 1과 같지 않고, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 62보다 작고, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 2와 같지 않은 경우, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1 만큼 작은 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1 만큼 큰 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1 만큼 작은 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 포함하는 것을 특징으로 한다.
- [27] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스는 하기와 같은 식에 의하여 획득되고, 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스 = ((인트라 예측 모드의 인덱스+offset)%(offset+3))+2, 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스는 하기와 같은 식에 의하여 획득되고, 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스 = ((인트라 예측 모드의 인덱스-1)%(offset+3))+2, 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스는 하기와 같은 식에 의하여 획득되고, 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스 = ((인트라 예측 모드의 인덱스+offset-1)%(offset+3))+2, 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스는 하기와 같은 식에 의하여 획득되고, 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스 = ((인트라 예측 모드의 인덱스)%(offset+3))+2, offset은 61인 것을 특징으로 한다.
- [28] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법은 수신된 비트스트림에 기초하여 현재 블록의 인트라 예측 모드의 인덱스를 획득하는 단계, 비트스트림으로부터 현재 블록의 참조 샘플 라인 인덱스를 획득하는 단계, 인트라 예측 모드의 인덱스 또는 참조 샘플 라인 인덱스 중 적어도 하나에 기초하여 복수의 참조 샘플들 및 추가 참조 샘플을 생성하는 단계를 포함하는 것을 특징으로 한다.
- [29] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는

방법의 복수의 참조 샘플들 및 추가 참조 샘플을 생성하는 단계는, 인트라 예측 모드의 인덱스가 34보다 크거나 같고, 인트라 예측 모드의 인덱스에 매핑되는 인트라 예측 모드의 각도가 0보다 크거나 같은 경우, 현재 블록에 인접한 미리 복원된 상측 블록에 기초하여 복수의 제 1 참조 샘플들 및 제 1 추가 참조 샘플을 생성하는 단계 및 인트라 예측 모드의 인덱스가 34보다 작고, 인트라 예측 모드의 각도가 0보다 크거나 같은 경우, 현재 블록에 인접한 미리 복원된 좌측 블록에 기초하여 복수의 제 2 참조 샘플들 및 제 2 추가 참조 샘플을 생성하는 단계를 포함하는 것을 특징으로 한다.

- [30] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 복수의 제 1 참조 샘플들 및 제 1 추가 참조 샘플을 생성하는 단계는, 미리 복원된 상측 블록에 포함된 샘플들 중 복수의 제 1 참조 샘플들의 위치에 대응되는 샘플들에 기초하여 복수의 제 1 참조 샘플들을 획득하는 단계 및 복수의 제 1 참조 샘플들 중 최우측 위치의 샘플을 패딩하여 제 1 추가 참조 샘플을 획득하는 단계를 포함하고, 복수의 제 2 참조 샘플들 및 제 2 추가 참조 샘플을 생성하는 단계는, 미리 복원된 좌측 블록에 포함된 샘플들 중 복수의 제 2 참조 샘플들의 위치에 대응되는 샘플들에 기초하여 복수의 제 2 참조 샘플들을 획득하는 단계; 및 복수의 제 2 참조 샘플들 중 최하측 위치의 샘플을 패딩하여 제 2 추가 참조 샘플을 획득하는 단계를 포함하는 것을 특징으로 한다.
- [31] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 복수의 참조 샘플들 및 추가 참조 샘플을 생성하는 단계는, 참조 샘플 라인 인덱스 또는 현재 블록의 너비 및 높이의 비율에 기초하여 추가 참조 샘플의 개수를 결정하는 단계를 포함하는 것을 특징으로 한다.
- [32] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 추가 참조 샘플의 개수는, $\text{추가 참조 샘플의 개수} = \text{참조 샘플 라인 인덱스} \times \text{현재 블록의 너비 및 높이의 비율} + \text{offset}$ 인 것을 특징으로 한다.
- [33] 본 개시의 일 실시예에 따른 인트라 예측에 기초하여 비디오 신호를 처리하는 방법의 offset은 1인 것을 특징으로 한다.

발명의 효과

- [34] 본 발명의 실시예에 따르면, 비디오 신호의 코딩 효율이 높아질 수 있다. 또한, 본 발명의 일 실시예에 따르면, 인트라 예측 모드에 기초하여 현재 블록을 보다 정확하게 예측할 수 있다.

[35]

도면의 간단한 설명

- [36] 도 1은 본 발명의 일 실시예에 따른 비디오 신호 인코딩 장치의 개략적인 블록도이다.
- [37] 도 2는 본 발명의 일 실시예에 따른 비디오 신호 디코딩 장치의 개략적인 블록도이다.

- [38] 도 3은 픽처 내에서 코딩 트리 유닛이 코딩 유닛들로 분할되는 실시예를 도시한다.
- [39] 도 4는 쿼드 트리 및 멀티-타입 트리의 분할을 시그널링하는 방법의 일 실시예를 도시한다.
- [40] 도 5 및 도 6은 본 발명의 실시예에 따른 인트라 예측 방법을 더욱 구체적으로 도시한다.
- [41] 도 7은 본 발명의 일 실시예에 따른 인터 예측 방법을 도시한다.
- [42] 도 8은 인코더가 레지듀얼 신호를 변환하는 방법을 구체적으로 나타내는 도면이다.
- [43] 도 9는 인코더 및 디코더가 변환 계수를 역변환하여 레지듀얼 신호를 획득하는 방법을 구체적으로 나타내는 도면이다.
- [44] 도 10은 본 개시의 일 실시예에 따른 Intra Subpartitions(ISP) mode의 개념을 나타낸 도면이다.
- [45] 도 11은 본 개시의 일 실시예에 따른 복수개의 참조 샘플 라인을 사용하여 예측하는 인트라(화면내) 예측 모드 및 MPM구성을 위한 이웃 블록의 위치를 나타낸 도면이다.
- [46] 도 12는 본 개시의 일 실시예에 따른 Coding Unit의 선택스 구조를 나타낸 도면이다.
- [47] 도 13은 본 개시의 일 실시예에 따른 intra_mode_planar_flag를 사용하여 화면내 예측 모드의 시그널링을 나타내는 선택스 구조를 나타낸 도면이다.
- [48] 도 14는 본 개시의 일 실시예에 따른 intra_mode_planar_flag와 ISP 모드 플래그 파싱 조건을 사용하여 화면내 예측 모드의 시그널링을 나타내는 선택스 구조를 나타낸 도면이다.
- [49] 도 15는 본 개시의 일 실시예에 따른 intra_mode_planar_flag와 참조샘플 라인 인덱스 조건을 사용하여 화면내 예측 모드 시그널링과 관련된 선택스 구조를 나타낸 도면이다.
- [50] 도 16은 본 개시의 일 실시예에 따른 MPM 시그널링에 관련된 선택스 구조를 나타낸 것으로 참조 라인 인덱스가 우선 파싱/시그널링 되는 것을 고려한 선택스 구조를 나타낸 도면이다.
- [51] 도 17은 본 개시의 일 실시예에 따른 MPM 시그널링에 관련된 선택스 구조를 나타낸 것으로 참조 라인 인덱스 값에 상관없이 Planar flag를 파싱하는 것을 고려한 선택스 구조를 나타낸 도면이다.
- [52] 도 18은 본 개시의 일 실시예에 따른 Planar 플래그와 MPM 사용여부를 나타내는 정보를 시그널링/파싱하는 선택스 구조를 나타낸 도면이다.
- [53] 도 19는 본 개시의 일 실시예에 따른 Planar 플래그에 기반한 참조 샘플 라인 인덱스 와 및 MPM 사용여부를 나타내는 정보에 대한 시그널링/파싱을 나타내는 선택스 구조 도면이다.
- [54] 도 20은 본 개시의 일 실시예에 따른 코딩 유닛 선택스 구조에서

intra_luma_planar_flag를 나타낸 도면이다.

- [55] 도 21은 본 개시의 일 실시예에 따른 화면내 예측에 사용된 복수개의 참조 샘플 라인 (multi-reference line intra prediction (MRL))을 나타낸 도면이다.
- [56] 도 22는 본 개시의 일 실시예에 따른 현재 예측 블록에 이웃하는 블록에서 사용된 예측 모드를 고려하여 MPM 리스트를 만드는데 필요한 이웃 블록의 위치를 나타내는 도면이다.
- [57] 도 23은 본 개시의 일 실시예에 따라 Most Probable Mode(MPM)에 +1, -1 등의 offset들을 적용하여 새로운 MPM을 유도하는 방법을 나타낸 도면이다.
- [58] 도 24는 본 개시의 일 실시예에 따라 Most Probable Mode(MPM)에 +1, -1 등의 offset들을 적용하여 새로운 MPM을 유도하는 방법을 non-square블록에 적용한 예를 나타낸 도면이다.
- [59] 도 25는 본 개시의 일 실시예에 따라 MRL를 적용한 직사각형 블록에 대한 참조 샘플을 설명하기 위한 도면이다.
- [60] 도 26은 본 개시의 일 실시예에 따라 MRL를 적용하는 직사각형 블록에 대한 참조 샘플을 설명하기 위한 도면이다.
- [61] 도 27은 본 개시의 일 실시예에 따라 MRL를 적용하는 직사각형 블록에 대한 참조 샘플을 설명하기 위한 도면이다.

발명의 실시를 위한 형태

- [62] 본 명세서에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도, 관례 또는 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한 특정 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 발명의 설명 부분에서 그 의미를 기재할 것이다. 따라서 본 명세서에서 사용되는 용어는, 단순한 용어의 명칭이 아닌 그 용어가 가진 실질적인 의미와 본 명세서의 전반에 걸친 내용을 토대로 해석되어야 함을 밝혀두고자 한다.
- [63] 본 명세서에서 일부 용어들은 다음과 같이 해석될 수 있다. 코딩은 경우에 따라 인코딩 또는 디코딩으로 해석될 수 있다. 본 명세서에서 비디오 신호의 인코딩(부호화)을 수행하여 비디오 신호 비트스트림을 생성하는 장치는 인코딩 장치 혹은 인코더로 지칭되며, 비디오 신호 비트스트림의 디코딩(복호화)을 수행하여 비디오 신호를 복원하는 장치는 디코딩 장치 혹은 디코더로 지칭된다. 또한, 본 명세서에서 비디오 신호 처리 장치는 인코더 및 디코더를 모두 포함하는 개념의 용어로 사용된다. 정보(information)는 값(values), 파라미터(parameter), 계수(coefficients), 성분(elements) 등을 모두 포함하는 용어로서, 경우에 따라 의미는 달리 해석될 수 있으므로 본 발명은 이에 한정되지 아니한다. '유닛'은 영상 처리의 기본 단위 또는 픽처의 특정 위치를 지칭하는 의미로 사용되며, 루마(luma) 성분 및 크로마(chroma) 성분 중 적어도 하나를 포함하는 이미지 영역을 가리킨다. 또한, '블록'은 루마 성분 및 크로마

성분들(즉, Cb 및 Cr) 중 특정 성분을 포함하는 이미지 영역을 가리킨다. 다만, 실시예에 따라서 '유닛', '블록', '파트션' 및 '영역' 등의 용어는 서로 혼용하여 사용될 수 있다. 또한, 본 명세서에서 유닛은 코딩 유닛, 예측 유닛, 변환 유닛을 모두 포함하는 개념으로 사용될 수 있다. 픽처는 필드 혹은 프레임을 가리키며, 실시예에 따라 상기 용어들은 서로 혼용하여 사용될 수 있다.

[64] 도 1은 본 발명의 일 실시예에 따른 비디오 신호 인코딩 장치(100)의 개략적인 블록도이다. 도 1을 참조하면, 본 발명의 인코딩 장치(100)는 변환부(110), 양자화부(115), 역양자화부(120), 역변환부(125), 필터링부(130), 예측부(150) 및 엔트로피 코딩부(160)를 포함한다.

[65] 변환부(110)는 입력 받은 비디오 신호와 예측부(150)에서 생성된 예측 신호의 차이인 레지듀얼 신호를 변환하여 변환 계수 값을 획득한다. 예를 들어, 이산 코사인 변환(Discrete Cosine Transform, DCT), 이산 사인 변환(Discrete Sine Transform, DST) 또는 웨이블릿 변환(Wavelet Transform) 등이 사용될 수 있다. 이산 코사인 변환 및 이산 사인 변환은 입력된 픽처 신호를 블록 형태로 나누어 변환을 수행하게 된다. 변환에 있어서 변환 영역 내의 값들의 분포와 특성에 따라서 코딩 효율이 달라질 수 있다. 양자화부(115)는 변환부(110)에서 출력된 변환 계수 값을 양자화한다.

[66] 코딩 효율을 높이기 위하여 픽처 신호를 그대로 코딩하는 것이 아니라, 예측부(150)를 통해 이미 코딩된 영역을 이용하여 픽처를 예측하고, 예측된 픽처에 원본 픽처와 예측 픽처 간의 레지듀얼 값을 더하여 복원 픽처를 획득하는 방법이 사용된다. 인코더와 디코더에서 미스매치가 발생되지 않도록 하기 위해, 인코더에서 예측을 수행할 때에는 디코더에서도 사용 가능한 정보를 사용해야 한다. 이를 위해, 인코더에서는 부호화한 현재 블록을 다시 복원하는 과정을 수행한다. 역양자화부(120)에서는 변환 계수 값을 역양자화하고, 역변환부(125)에서는 역양자화된 변환 계수 값을 이용하여 레지듀얼 값을 복원한다. 한편, 필터링부(130)는 복원된 픽처의 품질 개선 및 부호화 효율 향상을 위한 필터링 연산을 수행한다. 예를 들어, 더블록킹 필터, 샘플 적응적 오프셋(Sample Adaptive Offset, SAO) 및 적응적 루프 필터 등이 포함될 수 있다. 필터링을 거친 픽처는 출력되거나 참조 픽처로 이용하기 위하여 복호 픽처 버퍼(Decoded Picture Buffer, DPB, 156)에 저장된다.

[67] 코딩 효율을 높이기 위하여 픽처 신호를 그대로 코딩하는 것이 아니라, 예측부(150)를 통해 이미 코딩된 영역을 이용하여 픽처를 예측하고, 예측된 픽처에 원 픽처와 예측 픽처 간의 레지듀얼 값을 더하여 복원 픽처를 획득하는 방법이 사용된다. 인트라 예측부(152)에서는 현재 픽처 내에서 화면내 예측을 수행하며, 인터 예측부(154)에서는 복호 픽처 버퍼(156)에 저장된 참조 픽처를 이용하여 현재 픽처를 예측한다. 인트라 예측부(152)는 현재 픽처 내의 복원된 영역들로부터 화면내 예측을 수행하여, 화면내 부호화 정보를 엔트로피 코딩부(160)에 전달한다. 인터 예측부(154)는 다시 모션 추정부(154a) 및 모션

보상부(154b)를 포함하여 구성될 수 있다. 모션 추정부(154a)에서는 복원된 특정 영역을 참조하여 현재 영역의 모션 벡터값을 획득한다. 모션 추정부(154a)에서는 참조 영역의 위치 정보(참조 프레임, 모션 벡터 등) 등을 엔트로피 코딩부(160)로 전달하여 비트스트림에 포함될 수 있도록 한다. 모션 추정부(154a)에서 전달된 모션 벡터값을 이용하여 모션 보상부(154b)에서는 화면간 모션 보상을 수행한다.

[68] 예측부(150)는 인트라 예측부(152)와 인터 예측부(154)를 포함한다. 인트라 예측부(152)는 현재 픽처 내에서 인트라(intra) 예측을 수행하며, 인터 예측부(154)는 복호 픽처 버퍼(156)에 저장된 참조 픽처를 이용하여 현재 픽처를 예측하는 인터(inter) 예측을 수행한다. 인트라 예측부(152)는 현재 픽처 내의 복원된 샘플들로부터 인트라 예측을 수행하여, 인트라 부호화 정보를 엔트로피 코딩부(160)에 전달한다. 인트라 부호화 정보는 인트라 예측 모드, MPM(Most Probable Mode) 플래그, MPM 인덱스 중 적어도 하나를 포함할 수 있다. 인트라 부호화 정보는 참조 샘플에 관한 정보를 포함할 수 있다. 인터 예측부(154)는 모션 추정부(154a) 및 모션 보상부(154b)를 포함하여 구성될 수 있다. 모션 추정부(154a)는 복원된 참조 픽처의 특정 영역을 참조하여 현재 영역의 모션 벡터값을 획득한다. 모션 추정부(154a)는 참조 영역에 대한 모션 정보 세트(참조 픽처 인덱스, 모션 벡터 정보 등)를 엔트로피 코딩부(160)로 전달한다. 모션 보상부(154b)는 모션 추정부(154a)에서 전달된 모션 벡터값을 이용하여 모션 보상을 수행한다. 인터 예측부(154)는 참조 영역에 대한 모션 정보를 포함하는 인터 부호화 정보를 엔트로피 코딩부(160)에 전달한다.

[69] 추가적인 실시예에 따라, 예측부(150)는 인트라 블록 카피(block copy, BC) 예측부(미도시)를 포함할 수 있다. 인트라 BC 예측부는 현재 픽처 내의 복원된 샘플들로부터 인트라 BC 예측을 수행하여, 인트라 BC 부호화 정보를 엔트로피 코딩부(160)에 전달한다. 인트라 BC 예측부는 현재 픽처 내의 특정 영역을 참조하여 현재 영역의 예측에 이용되는 참조 영역을 나타내는 블록 벡터값을 획득한다. 인트라 BC 예측부는 획득된 블록 벡터값을 이용하여 인트라 BC 예측을 수행할 수 있다. 인트라 BC 예측부는 인트라 BC 부호화 정보를 엔트로피 코딩부(160)로 전달한다. 인트라 BC 부호화 정보는 블록 벡터 정보를 포함할 수 있다.

[70] 위와 같은 픽처 예측이 수행될 경우, 변환부(110)는 원본 픽처와 예측 픽처 간의 레지듀얼 값을 변환하여 변환 계수 값을 획득한다. 이때, 변환은 픽처 내에서 특정 블록 단위로 수행될 수 있으며, 특정 블록의 크기는 기 설정된 범위 내에서 가변할 수 있다. 양자화부(115)는 변환부(110)에서 생성된 변환 계수 값을 양자화하여 엔트로피 코딩부(160)로 전달한다.

[71] 엔트로피 코딩부(160)는 양자화된 변환 계수를 나타내는 정보, 인트라 부호화 정보, 및 인터 부호화 정보 등을 엔트로피 코딩하여 비디오 신호 비트스트림을 생성한다. 엔트로피 코딩부(160)에서는 가변 길이 코딩(Variable Length Coding, VLC) 방식과 산술 코딩(arithmetic coding) 방식 등이 사용될 수 있다. 가변 길이

코딩(VLC) 방식은 입력되는 심볼들을 연속적인 코드워드로 변환하는데, 코드워드의 길이는 가변적일 수 있다. 예를 들어, 자주 발생하는 심볼들을 짧은 코드워드로, 자주 발생하지 않은 심볼들은 긴 코드워드로 표현하는 것이다. 가변 길이 코딩 방식으로서 컨텍스트 기반 적응형 가변 길이 코딩(Context-based Adaptive Variable Length Coding, CAVLC) 방식이 사용될 수 있다. 산술 코딩은 연속적인 데이터 심볼들을 하나의 소수로 변환하는데, 산술 코딩은 각 심볼을 표현하기 위하여 필요한 최적의 소수 비트를 얻을 수 있다. 산술 코딩으로서 컨텍스트 기반 적응형 산술 부호화(Context-based Adaptive Binary Arithmetic Code, CABAC)가 이용될 수 있다. 예를 들어, 엔트로피 코딩부(160)는 양자화된 변환 계수를 나타내는 정보를 이진화할 수 있다. 또한, 엔트로피 코딩부(160)는 이진화된 정보를 산술 코딩하여 비트스트림을 생성할 수 있다.

- [72] 상기 생성된 비트스트림은 NAL(Network Abstraction Layer) 유닛을 기본 단위로 캡슐화 된다. NAL 유닛은 부호화된 정수 개의 코딩 트리 유닛(coding tree unit)을 포함한다. 비디오 디코더에서 비트스트림을 복호화하기 위해서는 먼저 비트스트림을 NAL 유닛 단위로 분리한 후, 분리된 각각의 NAL 유닛을 복호화해야 한다. 한편, 비디오 신호 비트스트림의 복호화를 위해 필요한 정보들은 픽처 파라미터 세트(Picture Parameter Set, PPS), 시퀀스 파라미터 세트(Sequence Parameter Set, SPS), 비디오 파라미터 세트(Video Parameter Set, VPS) 등과 같은 상위 레벨 세트의 RBSP(Raw Byte Sequence Payload)를 통해 전송될 수 있다.
- [73] 한편, 도 1의 블록도는 본 발명의 일 실시예에 따른 인코딩 장치(100)를 나타낸 것으로서, 분리하여 표시된 블록들은 인코딩 장치(100)의 엘리먼트들을 논리적으로 구별하여 도시한 것이다. 따라서 전술한 인코딩 장치(100)의 엘리먼트들은 디바이스의 설계에 따라 하나의 칩으로 또는 복수의 칩으로 장착될 수 있다. 일 실시예에 따르면, 전술한 인코딩 장치(100)의 각 엘리먼트의 동작은 프로세서(미도시)에 의해 수행될 수 있다.
- [74] 도 2는 본 발명의 일 실시예에 따른 비디오 신호 디코딩 장치(200)의 개략적인 블록도이다. 도 2를 참조하면 본 발명의 디코딩 장치(200)는 엔트로피 디코딩부(210), 역양자화부(220), 역변환부(225), 필터링부(230) 및 예측부(250)를 포함한다.
- [75] 엔트로피 디코딩부(210)는 비디오 신호 비트스트림을 엔트로피 디코딩하여, 각 영역에 대한 변환 계수 정보, 인트라 부호화 정보, 인터 부호화 정보 등을 추출한다. 예를 들어, 엔트로피 디코딩부(210)는 비디오 신호 비트스트림으로부터 특정 영역의 변환 계수 정보에 대한 이진화 코드를 획득할 수 있다. 또한, 엔트로피 디코딩부(210)는 이진화 코드를 역 이진화하여 양자화된 변환 계수를 획득한다. 역양자화부(220)는 양자화된 변환 계수를 역양자화하고, 역변환부(225)는 역양자화된 변환 계수를 이용하여 레지듀얼 값을 복원한다. 비디오 신호 처리 장치(200)는 역변환부(225)에서 획득된 레지듀얼 값을

- 예측부(250)에서 획득된 예측값과 합산하여 원래의 화소값을 복원한다.
- [76] 한편, 필터링부(230)는 픽처에 대한 필터링을 수행하여 화질을 향상시킨다. 여기에는 블록 왜곡 현상을 감소시키기 위한 디블록킹 필터 및/또는 픽처 전체의 왜곡 제거를 위한 적응적 루프 필터 등이 포함될 수 있다. 필터링을 거친 픽처는 출력되거나 다음 픽처에 대한 참조 픽처로 이용하기 위하여 복호 픽처 버퍼(DPB, 256)에 저장된다.
- [77] 예측부(250)는 인트라 예측부(252) 및 인터 예측부(254)를 포함한다. 예측부(250)는 전술한 엔트로피 디코딩부(210)를 통해 복호화된 부호화 타입, 각 영역에 대한 변환 계수, 인트라/인터 부호화 정보 등을 활용하여 예측 픽처를 생성한다. 복호화가 수행되는 현재 블록을 복원하기 위해서, 현재 블록이 포함된 현재 픽처 또는 다른 픽처들의 복호화된 영역이 이용될 수 있다. 복원에 현재 픽처만을 이용하는, 즉 인트라 예측 또는 인트라 BC 예측을 수행하는 픽처(또는, 타일/슬라이스)를 인트라 픽처 또는 I 픽처(또는, 타일/슬라이스), 인트라 예측, 인터 예측 및 인트라 BC 예측을 모두 수행할 수 있는 픽처(또는, 타일/슬라이스)를 인터 픽처(또는, 타일/슬라이스)라고 한다. 인터 픽처(또는, 타일/슬라이스) 중 각 블록의 샘플값들을 예측하기 위하여 최대 하나의 모션 벡터 및 참조 픽처 인덱스를 이용하는 픽처(또는, 타일/슬라이스)를 예측 픽처(predictive picture) 또는 P 픽처(또는, 타일/슬라이스)라고 하며, 최대 두 개의 모션 벡터 및 참조 픽처 인덱스를 이용하는 픽처(또는, 타일/슬라이스)를 쌍예측 픽처(Bi-predictive picture) 또는 B 픽처(또는, 타일/슬라이스)라고 한다. 다시 말해서, P 픽처(또는, 타일/슬라이스)는 각 블록을 예측하기 위해 최대 하나의 모션 정보 세트를 이용하고, B 픽처(또는, 타일/슬라이스)는 각 블록을 예측하기 위해 최대 두 개의 모션 정보 세트를 이용한다. 여기서, 모션 정보 세트는 하나 이상의 모션 벡터와 하나의 참조 픽처 인덱스를 포함한다.
- [78] 인트라 예측부(252)는 인트라 부호화 정보 및 현재 픽처 내의 복원된 샘플들을 이용하여 예측 블록을 생성한다. 전술한 바와 같이, 인트라 부호화 정보는 인트라 예측 모드, MPM(Most Probable Mode) 플래그, MPM 인덱스 중 적어도 하나를 포함할 수 있다. 인트라 예측부(252)는 현재 블록의 좌측 및/또는 상측에 위치한 복원된 샘플들을 참조 샘플들로 이용하여 현재 블록의 샘플 값들을 예측한다. 본 개시에서, 복원된 샘플들, 참조 샘플들 및 현재 블록의 샘플들은 픽셀들을 나타낼 수 있다. 또한, 샘플 값(sample value)들은 픽셀 값들을 나타낼 수 있다.
- [79] 일 실시예에 따르면, 참조 샘플들은 현재 블록의 주변 블록에 포함된 샘플들일 수 있다. 예를 들어, 참조 샘플들은 현재 블록의 좌측 경계에 인접한 샘플들 및/또는 상측 경계에 인접한 샘플들일 수 있다. 또한, 참조 샘플들은 현재 블록의 주변 블록의 샘플들 중 현재 블록의 좌측 경계로부터 기 설정된 거리 이내의 라인 상에 위치하는 샘플들 및/또는 현재 블록의 상측 경계로부터 기 설정된 거리 이내의 라인 상에 위치하는 샘플들일 수 있다. 이때, 현재 블록의 주변

블록은 현재 블록에 인접한 좌측(L) 블록, 상측(A) 블록, 하좌측(Below Left, BL) 블록, 상우측(Above Right, AR) 블록 또는 상좌측(Above Left, AL) 블록 중 적어도 하나를 포함할 수 있다.

- [80] 인터 예측부(254)는 복호 픽처 버퍼(256)에 저장된 참조 픽처 및 인터 부호화 정보를 이용하여 예측 블록을 생성한다. 인터 부호화 정보는 참조 블록에 대한 현재 블록의 모션 정보 세트(참조 픽처 인덱스, 모션 벡터 정보 등)를 포함할 수 있다. 인터 예측에는 L0 예측, L1 예측 및 쌍예측(Bi-prediction)이 있을 수 있다. L0 예측은 L0 픽처 리스트에 포함된 1개의 참조 픽처를 이용한 예측이고, L1 예측은 L1 픽처 리스트에 포함된 1개의 참조 픽처를 이용한 예측을 의미한다. 이를 위해서는 1세트의 모션 정보(예를 들어, 모션 벡터 및 참조 픽처 인덱스)가 필요할 수 있다. 쌍예측 방식에서는 최대 2개의 참조 영역을 이용할 수 있는데, 이 2개의 참조 영역은 동일한 참조 픽처에 존재할 수도 있고, 서로 다른 픽처에 각각 존재할 수도 있다. 즉, 쌍예측 방식에서는 최대 2세트의 모션 정보(예를 들어, 모션 벡터 및 참조 픽처 인덱스)가 이용될 수 있는데, 2개의 모션 벡터가 동일한 참조 픽처 인덱스에 대응될 수도 있고 서로 다른 참조 픽처 인덱스에 대응될 수도 있다. 이때, 참조 픽처들은 시간적으로 현재 픽처 이전이나 이후 모두에 표시(또는 출력)될 수 있다. 일 실시예에 따라, 쌍예측 방식에서는 사용되는 2개의 참조 영역은 L0 픽처 리스트 및 L1 픽처 리스트 각각에서 선택된 영역일 수 있다.
- [81] 인터 예측부(254)는 모션 벡터 및 참조 픽처 인덱스를 이용하여 현재 블록의 참조 블록을 획득할 수 있다. 상기 참조 블록은 참조 픽처 인덱스에 대응하는 참조 픽처 내에 존재한다. 또한, 모션 벡터에 의해서 특정된 블록의 샘플 값 또는 이의 보간(interpolation)된 값이 현재 블록의 예측자(predictor)로 이용될 수 있다. 서브펠(sub-pel) 단위의 픽셀 정확도를 갖는 모션 예측을 위하여 이를 테면, 루마 신호에 대하여 8-탭 보간 필터가, 크로마 신호에 대하여 4-탭 보간 필터가 사용될 수 있다. 다만, 서브펠 단위의 모션 예측을 위한 보간 필터는 이에 한정되지 않는다. 이와 같이 인터 예측부(254)는 이전에 복원된 픽처로부터 현재 유닛의 텍스처를 예측하는 모션 보상(motion compensation)을 수행한다. 이때, 인터 예측부는 모션 정보 세트를 이용할 수 있다.
- [82] 추가적인 실시예에 따라, 예측부(250)는 인트라 BC 예측부(미도시)를 포함할 수 있다. 인트라 BC 예측부는 현재 픽처 내의 복원된 샘플들을 포함하는 특정 영역을 참조하여 현재 영역을 복원할 수 있다. 인트라 BC 예측부는 엔트로피 디코딩부(210)로부터 현재 영역에 대한 인트라 BC 부호화 정보를 획득한다. 인트라 BC 예측부는 현재 픽처 내의 특정 영역을 지시하는 현재 영역의 블록 벡터값을 획득한다. 인트라 BC 예측부는 획득된 블록 벡터값을 이용하여 인트라 BC 예측을 수행할 수 있다. 인트라 BC 부호화 정보는 블록 벡터 정보를 포함할 수 있다.
- [83] 상기 인트라 예측부(252) 또는 인터 예측부(254)로부터 출력된 예측값, 및

역변환부(225)로부터 출력된 레지듀얼 값이 더해져서 복원된 비디오 픽처가 생성된다. 즉, 비디오 신호 디코딩 장치(200)는 예측부(250)에서 생성된 예측 블록과 역변환부(225)로부터 획득된 레지듀얼을 이용하여 현재 블록을 복원한다.

- [84] 한편, 도 2의 블록도는 본 발명의 일 실시예에 따른 디코딩 장치(200)를 나타낸 것으로서, 분리하여 표시된 블록들은 디코딩 장치(200)의 엘리먼트들을 논리적으로 구별하여 도시한 것이다. 따라서 전술한 디코딩 장치(200)의 엘리먼트들은 디바이스의 설계에 따라 하나의 칩으로 또는 복수의 칩으로 장착될 수 있다. 일 실시예에 따르면, 전술한 디코딩 장치(200)의 각 엘리먼트의 동작은 프로세서(미도시)에 의해 수행될 수 있다.
- [85] 도 3은 픽처 내에서 코딩 트리 유닛(Coding Tree Unit, CTU)이 코딩 유닛들(Coding Units, CUs)로 분할되는 실시예를 도시한다. 비디오 신호의 코딩 과정에서, 픽처는 코딩 트리 유닛(CTU)들의 시퀀스로 분할될 수 있다. 코딩 트리 유닛은 루마(luma) 샘플들의 $N \times N$ 블록과, 이에 대응하는 크로마(chroma) 샘플들의 2개의 블록들로 구성된다. 코딩 트리 유닛은 복수의 코딩 유닛들로 분할될 수 있다. 코딩 트리 유닛은 분할되지 않고 리프 노드가 될 수도 있다. 이 경우, 코딩 트리 유닛 자체가 코딩 유닛이 될 수 있다. 코딩 유닛은 상기에서 설명한 비디오 신호의 처리 과정, 즉 인트라/인터 예측, 변환, 양자화 및/또는 엔트로피 코딩 등의 과정에서 픽처를 처리하기 위한 기본 단위를 가리킨다. 하나의 픽처 내에서 코딩 유닛의 크기 및 모양은 일정하지 않을 수 있다. 코딩 유닛은 정사각형 혹은 직사각형의 모양을 가질 수 있다. 직사각형 코딩 유닛(혹은, 직사각형 블록)은 수직 코딩 유닛(혹은, 수직 블록)과 수평 코딩 유닛(혹은, 수평 블록)을 포함한다. 본 명세서에서, 수직 블록은 높이가 너비보다 큰 블록이며, 수평 블록은 너비가 높이보다 큰 블록이다. 또한, 본 명세서에서 정사각형이 아닌(non-square) 블록은 직사각형 블록을 가리킬 수 있지만, 본 발명은 이에 한정되지 않는다.
- [86] 도 3을 참조하면, 코딩 트리 유닛은 먼저 쿼드 트리(Quad Tree, QT) 구조로 분할된다. 즉, 쿼드 트리 구조에서 $2N \times 2N$ 크기를 가지는 하나의 노드는 $N \times N$ 크기를 가지는 네 개의 노드들로 분할될 수 있다. 본 명세서에서 쿼드 트리는 4진(quaternary) 트리로도 지칭될 수 있다. 쿼드 트리 분할은 재귀적으로 수행될 수 있으며, 모든 노드들이 동일한 깊이로 분할될 필요는 없다.
- [87] 한편, 전술한 쿼드 트리의 리프 노드(leaf node)는 멀티-타입 트리(Multi-Type Tree, MTT) 구조로 더욱 분할될 수 있다. 본 발명의 실시예에 따르면, 멀티 타입 트리 구조에서는 하나의 노드가 수평 혹은 수직 분할의 2진(binary, 바이너리) 혹은 3진(ternary, 터너리) 트리 구조로 분할될 수 있다. 즉, 멀티-타입 트리 구조에는 수직 바이너리 분할, 수평 바이너리 분할, 수직 터너리 분할 및 수평 터너리 분할의 4가지 분할 구조가 존재한다. 본 발명의 실시예에 따르면, 상기 각 트리 구조에서 노드의 너비 및 높이는 모두 2의 거듭제곱 값을 가질 수 있다.

예를 들어, 바이너리 트리(Binary Tree, BT) 구조에서, $2N \times 2N$ 크기의 노드는 수직 바이너리 분할에 의해 2개의 $N \times 2N$ 노드들로 분할되고, 수평 바이너리 분할에 의해 2개의 $2N \times N$ 노드들로 분할될 수 있다. 또한, 터너리 트리(Ternary Tree, TT) 구조에서, $2N \times 2N$ 크기의 노드는 수직 터너리 분할에 의해 $(N/2) \times 2N$, $N \times 2N$ 및 $(N/2) \times 2N$ 의 노드들로 분할되고, 수평 바이너리 분할에 의해 $2N \times (N/2)$, $2N \times N$ 및 $2N \times (N/2)$ 의 노드들로 분할될 수 있다. 이러한 멀티-타입 트리 분할은 재귀적으로 수행될 수 있다.

- [88] 멀티-타입 트리의 리프 노드는 코딩 유닛이 될 수 있다. 코딩 유닛에 대한 분할이 지시되지 않거나 코딩 유닛이 최대 변환 길이에 비해 크지 않은 경우, 해당 코딩 유닛은 더 이상의 분할 없이 예측 및 변환의 단위로 사용된다. 한편, 전술한 쿼드 트리 및 멀티-타입 트리에서 다음의 파라미터들 중 적어도 하나가 사전에 정의되거나 PPS, SPS, VPS 등과 같은 상위 레벨 세트의 RBSP를 통해 전송될 수 있다. 1) CTU 크기: 쿼드 트리의 루트 노드(root node) 크기, 2) 최소 QT 크기(MinQtSize): 허용된 최소 QT 리프 노드 크기, 3) 최대 BT 크기(MaxBtSize): 허용된 최대 BT 루트 노드 크기, 4) 최대 TT 크기(MaxTtSize): 허용된 최대 TT 루트 노드 크기, 5) 최대 MTT 깊이(MaxMttDepth): QT의 리프 노드로부터의 MTT 분할의 최대 허용 깊이, 6) 최소 BT 크기(MinBtSize): 허용된 최소 BT 리프 노드 크기, 7) 최소 TT 크기(MinTtSize): 허용된 최소 TT 리프 노드 크기.
- [89] 도 4는 쿼드 트리 및 멀티-타입 트리의 분할을 시그널링하는 방법의 일 실시예를 도시한다. 전술한 쿼드 트리 및 멀티-타입 트리의 분할을 시그널링하기 위해 기 설정된 플래그들이 사용될 수 있다. 도 4를 참조하면, 쿼드 트리 노드의 분할 여부를 지시하는 플래그 'qt_split_flag', 멀티-타입 트리 노드의 분할 여부를 지시하는 플래그 'mtt_split_flag', 멀티-타입 트리 노드의 분할 방향을 지시하는 플래그 'mtt_split_vertical_flag' 또는 멀티-타입 트리 노드의 분할 모양을 지시하는 플래그 'mtt_split_binary_flag' 중 적어도 하나가 사용될 수 있다.
- [90] 본 발명의 실시예에 따르면, 코딩 트리 유닛은 쿼드 트리의 루트 노드이며, 쿼드 트리 구조로 우선 분할될 수 있다. 쿼드 트리 구조에서는 각각의 노드 'QT_node' 별로 'qt_split_flag'가 시그널링된다. 'qt_split_flag'의 값이 1일 경우 해당 노드는 4개의 정사각형 노드들로 분할되며, 'qt_split_flag'의 값이 0일 경우 해당 노드는 쿼드 트리의 리프 노드 'QT_leaf_node'가 된다.
- [91] 각각의 쿼드 트리 리프 노드 'QT_leaf_node'는 멀티-타입 트리 구조로 더 분할될 수 있다. 멀티-타입 트리 구조에서는 각각의 노드 'MTT_node' 별로 'mtt_split_flag'가 시그널링된다. 'mtt_split_flag'의 값이 1일 경우 해당 노드는 복수의 직사각형 노드들로 분할되며, 'mtt_split_flag'의 값이 0일 경우 해당 노드는 멀티-타입 트리의 리프 노드 'MTT_leaf_node'가 된다. 멀티-타입 트리 노드 'MTT_node'가 복수의 직사각형 노드들로 분할될 경우(즉, 'mtt_split_flag'의 값이 1일 경우), 노드 'MTT_node'를 위한 'mtt_split_vertical_flag' 및 'mtt_split_binary_flag'가 추가로 시그널링될 수 있다. 'mtt_split_vertical_flag'의

값이 1일 경우 노드 'MTT_node'의 수직 분할이 지시되며, 'mtt_split_vertical_flag'의 값이 0일 경우 노드 'MTT_node'의 수평 분할이 지시된다. 또한, 'mtt_split_binary_flag'의 값이 1일 경우 노드 'MTT_node'는 2개의 직사각형 노드들로 분할되며, 'mtt_split_binary_flag'의 값이 0일 경우 노드 'MTT_node'는 3개의 직사각형 노드들로 분할된다.

- [92] 코딩을 위한 픽처 예측(모션 보상)은 더 이상 나누어지지 않는 코딩 유닛(즉 코딩 유닛 트리의 리프 노드)을 대상으로 이루어진다. 이러한 예측을 수행하는 기본 단위를 이하에서는 예측 유닛(prediction unit) 또는 예측 블록(prediction block)이라고 한다.
- [93] 이하, 본 명세서에서 사용되는 유닛이라는 용어는 예측을 수행하는 기본 단위인 상기 예측 유닛을 대체하는 용어로 사용될 수 있다. 다만, 본 발명이 이에 한정되는 것은 아니며, 더욱 광의적으로는 상기 코딩 유닛을 포함하는 개념으로 이해될 수 있다.
- [94] 도 5 및 도 6은 본 발명의 실시예에 따른 인트라 예측 방법을 더욱 구체적으로 도시한다. 전술한 바와 같이, 인트라 예측부는 현재 블록의 좌측 및/또는 상측에 위치한 복원된 샘플들을 참조 샘플들로 이용하여 현재 블록의 샘플 값들을 예측한다.
- [95] 먼저, 도 5는 인트라 예측 모드에서 현재 블록의 예측을 위해 사용되는 참조 샘플들의 일 실시예를 도시한다. 일 실시예에 따르면, 참조 샘플들은 현재 블록의 좌측 경계에 인접한 샘플들 및/또는 상측 경계에 인접한 샘플들일 수 있다. 도 5에 도시된 바와 같이, 현재 블록의 크기가 WXH 이고 현재 블록에 인접한 단일 참조 라인(line)의 샘플들이 인트라 예측에 사용될 경우, 현재 블록의 좌측 및/또는 상측에 위치한 최대 $2W+2H+1$ 개의 주변 샘플들을 사용하여 참조 샘플들이 설정될 수 있다.
- [96] 또한, 참조 샘플로 사용될 적어도 일부의 샘플이 아직 복원되지 않은 경우, 인트라 예측부는 참조 샘플 패딩 과정을 수행하여 참조 샘플을 획득할 수 있다. 또한, 인트라 예측부는 인트라 예측의 오차를 줄이기 위해 참조 샘플 필터링 과정을 수행할 수 있다. 즉, 주변 샘플들 및/또는 참조 샘플 패딩 과정에 의해 획득된 참조 샘플들에 필터링을 수행하여 필터링된 참조 샘플들을 획득할 수 있다. 인트라 예측부는 이와 같이 획득된 참조 샘플들을 이용하여 현재 블록의 샘플들을 예측한다. 인트라 예측부는 필터링되지 않은 참조 샘플들 또는 필터링된 참조 샘플들을 이용하여 현재 블록의 샘플들을 예측한다. 본 개시에서, 주변 샘플들은 적어도 하나의 참조 라인 상의 샘플들을 포함할 수 있다. 예를 들어, 주변 샘플들은 현재 블록의 경계에 인접한 라인 상의 인접 샘플들을 포함할 수 있다.
- [97] 다음으로, 도 6은 인트라 예측에 사용되는 예측 모드들의 일 실시예를 도시한다. 인트라 예측을 위해, 인트라 예측 방향을 지시하는 인트라 예측 모드 정보가 시그널링될 수 있다. 인트라 예측 모드 정보는 인트라 예측 모드 세트를

구성하는 복수의 인트라 예측 모드들 중 어느 하나를 지시한다. 현재 블록이 인트라 예측 블록일 경우, 디코더는 비트스트림으로부터 현재 블록의 인트라 예측 모드 정보를 수신한다. 디코더의 인트라 예측부는 추출된 인트라 예측 모드 정보에 기초하여 현재 블록에 대한 인트라 예측을 수행한다.

- [98] 본 발명의 실시예에 따르면, 인트라 예측 모드 세트는 인트라 예측에 사용되는 모든 인트라 예측 모드들(예, 총 67개의 인트라 예측 모드들)을 포함할 수 있다. 더욱 구체적으로, 인트라 예측 모드 세트는 평면 모드, DC 모드 및 복수의(예, 65개의) 각도 모드들(즉, 방향 모드들)을 포함할 수 있다. 각각의 인트라 예측 모드는 기 설정된 인덱스(즉, 인트라 예측 모드 인덱스)를 통해 지시될 수 있다. 예를 들어, 도 6에 도시된 바와 같이 인트라 예측 모드 인덱스 0은 평면 모드를 지시하고, 인트라 예측 모드 인덱스 1은 DC 모드를 지시한다. 또한, 인트라 예측 모드 인덱스 2 내지 66은 서로 다른 각도 모드들을 각각 지시할 수 있다. 각도 모드들은 기 설정된 각도 범위 이내의 서로 다른 각도들을 각각 지시한다. 예를 들어, 각도 모드는 시계 방향으로 45도에서 -135도 사이의 각도 범위(즉, 제1 각도 범위) 이내의 각도를 지시할 수 있다. 상기 각도 모드는 12시 방향을 기준으로 정의될 수 있다. 이때, 인트라 예측 모드 인덱스 2는 수평 대각(Horizontal Diagonal, HDIA) 모드를 지시하고, 인트라 예측 모드 인덱스 18은 수평(Horizontal, HOR) 모드를 지시하고, 인트라 예측 모드 인덱스 34는 대각(Diagonal, DIA) 모드를 지시하고, 인트라 예측 모드 인덱스 50은 수직(Vertical, VER) 모드를 지시하며, 인트라 예측 모드 인덱스 66은 수직 대각(Vertical Diagonal, VDIA) 모드를 지시한다.
- [99] 이하, 도 7을 참조하여 본 발명의 일 실시예에 따른 인터 예측 방법에 대해 설명하도록 한다. 본 개시에서, 인터 예측 방법은 병진 운동(translation motion)에 최적화된 일반 인터 예측 방법 및 어파인(affine) 모델 기반의 인터 예측 방법을 포함할 수 있다. 또한, 모션 벡터는 일반 인터 예측 방법에 따른 모션 보상을 위한 일반 모션 벡터 및 어파인 모션 보상을 위한 컨트롤 포인트 모션 벡터(control point motion vector) 중 적어도 하나를 포함할 수 있다.
- [100] 도 7은 본 발명의 일 실시예에 따른 인터 예측 방법을 도시한다. 전술한 바와 같이, 디코더는 복호화된 다른 픽처의 복원된 샘플들을 참조하여 현재 블록을 예측할 수 있다. 도 7을 참조하면, 디코더는 현재 블록(701)의 모션 정보 세트에 기초하여 참조 픽처(720) 내의 참조 블록(702)을 획득한다. 이때, 모션 정보 세트는 참조 픽처 인덱스 및 모션 벡터(703)를 포함할 수 있다. 참조 픽처 인덱스는 참조 픽처 리스트에서 현재 블록의 인터 예측을 위한 참조 블록이 포함된 참조 픽처(720)를 지시한다. 일 실시예에 따라, 참조 픽처 리스트는 전술한 L0 픽처 리스트 또는 L1 픽처 리스트 중 적어도 하나를 포함할 수 있다. 모션 벡터(703)는 현재 픽처(710) 내에서 현재 블록(701)의 좌표 값과 참조 픽처(720) 내에서 참조 블록(702)의 좌표 값 간의 오프셋을 나타낸다. 디코더는 참조 블록(702)의 샘플 값들에 기초하여 현재 블록(701)의 예측자를 획득하고,

상기 예측자를 이용하여 현재 블록(701)을 복원한다.

- [101] 구체적으로, 인코더는 복원 순서가 앞선 픽처들에서 현재 블록과 유사한 블록을 탐색하여 전술한 참조 블록을 획득할 수 있다. 예를 들어, 인코더는 기 설정된 탐색 영역 내에서 현재 블록과 샘플 값 차이의 합이 최소가 되는 참조 블록을 탐색할 수 있다. 이때, 현재 블록과 참조 블록의 샘플들 간의 유사도를 측정하기 위해, SAD (Sum Of Absolute Difference) 또는 SATD (Sum of Hadamard Transformed Difference) 중 적어도 하나가 사용될 수 있다. 여기에서, SAD는 두 블록에 포함된 샘플 값들의 차이 각각의 절대값을 모두 더한 값일 수 있다. 또한, SATD는 두 블록에 포함된 샘플 값들의 차이를 하다마드 변환(Hadamard Transform)하여 획득된 하다마드 변환 계수의 절대값을 모두 더한 값일 수 있다.
- [102] 한편, 현재 블록은 하나 이상의 참조 영역을 이용하여 예측될 수도 있다. 전술한 바와 같이, 현재 블록은 2개 이상의 참조 영역을 이용하는 쌍예측 방식을 통해 인터 예측될 수 있다. 일 실시예에 따라, 디코더는 현재 블록의 2개의 모션 정보 세트에 기초하여 2개의 참조 블록을 획득할 수 있다. 또한, 디코더는 획득된 2개의 참조 블록 각각의 샘플 값들에 기초하여 현재 블록의 제1 예측자 및 제2 예측자를 획득할 수 있다. 또한, 디코더는 제1 예측자 및 제2 예측자를 이용하여 현재 블록을 복원할 수 있다. 예를 들어, 디코더는 제1 예측자 및 제2 예측자의 샘플 별 평균에 기초하여 현재 블록을 복원할 수 있다.
- [103] 전술한 바와 같이, 현재 블록의 모션 보상을 위해, 하나 이상의 모션 정보 세트가 시그널링될 수 있다. 이때, 복수의 블록 각각의 모션 보상을 위한 모션 정보 세트 간의 유사성이 이용될 수 있다. 예를 들어, 현재 블록의 예측에 사용되는 모션 정보 세트는 기 복원된 다른 샘플들 중 어느 하나의 예측에 사용된 모션 정보 세트로부터 유도될 수 있다. 이를 통해, 인코더 및 디코더는 시그널링 오버헤드를 감소시킬 수 있다.
- [104] 예를 들어, 현재 블록의 모션 정보 세트와 동일 또는 유사한 모션 정보 세트에 기초하여 예측되었을 가능성이 있는 복수의 후보 블록들이 존재할 수 있다. 디코더는 해당 복수의 후보 블록들을 기초로 머지 후보 리스트(merge candidate list)를 생성할 수 있다. 여기에서, 머지 후보 리스트는 현재 블록보다 먼저 복원된 샘플들 중에서, 현재 블록의 모션 정보 세트와 관련된 모션 정보 세트에 기초하여 예측되었을 가능성이 있는 샘플에 대응하는 후보들을 포함할 수 있다. 인코더와 디코더는 미리 정의된 규칙에 따라 현재 블록의 머지 후보 리스트를 구성할 수 있다. 이때, 인코더와 디코더가 각각 구성한 머지 후보 리스트는 서로 동일할 수 있다. 예를 들어, 인코더 및 디코더는 현재 픽처 내에서 현재 블록의 위치에 기초하여 현재 블록의 머지 후보 리스트를 구성할 수 있다. 인코더 및 디코더가 현재 블록의 머지 후보 리스트를 구성하는 방법에 대해서는 도 9를 통해 후술하도록 한다. 본 개시에서, 특정 블록의 위치는 특정 블록을 포함하는 픽처 내에서 특정 블록의 좌상단(top-left) 샘플의 상대적인 위치를 나타낸다.
- [105] 한편, 코딩 효율을 높이기 위하여 전술한 레지듀얼 신호를 그대로 코딩하는

것이 아니라, 레지듀얼 신호를 변환하여 획득된 변환 계수 값을 양자화하고, 양자화된 변환 계수를 코딩하는 방법이 사용될 수 있다. 전술한 바와 같이, 변환부는 레지듀얼 신호를 변환하여 변환 계수 값을 획득할 수 있다. 이때, 특정 블록의 레지듀얼 신호는 현재 블록의 전 영역에 분산되어 있을 수 있다. 이에 따라, 레지듀얼 신호에 대한 주파수 영역 변환을 통해 저주파 영역에 에너지를 집중시켜 코딩 효율을 향상시킬 수 있다. 이하에서는, 레지듀얼 신호가 변환 또는 역변환되는 방법에 대해 구체적으로 설명하도록 한다.

[106] 도 8은 인코더가 레지듀얼 신호를 변환하는 방법을 구체적으로 나타내는 도면이다. 전술한 바와 같이, 공간 영역의 레지듀얼 신호는 주파수 영역으로 변환될 수 있다. 인코더는 획득된 레지듀얼 신호를 변환하여 변환 계수를 획득할 수 있다. 먼저, 인코더는 현재 블록에 대한 레지듀얼 신호를 포함하는 적어도 하나의 레지듀얼 블록을 획득할 수 있다. 레지듀얼 블록은 현재 블록 또는 현재 블록으로부터 분할된 블록들 중 어느 하나일 수 있다. 본 개시에서, 레지듀얼 블록은 현재 블록의 레지듀얼 샘플들을 포함하는 레지듀얼 어레이(array) 또는 레지듀얼 매트릭스(matrix)로 지칭될 수 있다. 또한, 본 개시에서 레지듀얼 블록은 변환 유닛 또는 변환 블록의 크기와 동일한 크기의 블록을 나타낼 수 있다.

[107] 다음으로, 인코더는 변환 커널을 사용하여 레지듀얼 블록을 변환할 수 있다. 레지듀얼 블록에 대한 변환에 사용되는 변환 커널은 수직 변환 및 수평 변환의 분리 가능한 특성을 가지는 변환 커널일 수 있다. 이 경우, 레지듀얼 블록에 대한 변환은 수직 변환 및 수평 변환으로 분리되어 수행될 수 있다. 예를 들어, 인코더는 레지듀얼 블록의 수직 방향으로 변환 커널을 적용하여 수직 변환을 수행할 수 있다. 또한, 인코더는 레지듀얼 블록의 수평 방향으로 변환 커널을 적용하여 수평 변환을 수행할 수 있다. 본 개시에서, 변환 커널은 변환 매트릭스, 변환 어레이, 변환 함수, 변환과 같이 레지듀얼 신호의 변환에 사용되는 파라미터 세트를 지칭하는 용어로 사용될 수 있다. 일 실시예에 따라, 변환 커널은 복수의 사용 가능한 커널들 중 어느 하나일 수 있다. 또한, 수직 변환 및 수평 변환 각각에 대해 서로 다른 변환 타입에 기반한 변환 커널이 사용될 수도 있다.

[108] 인코더는 레지듀얼 블록으로부터 변환된 변환 블록을 양자화부로 전달하여 양자화할 수 있다. 이때, 변환 블록은 복수의 변환 계수들을 포함할 수 있다. 구체적으로, 변환 블록은 2차원 배열된 복수의 변환 계수들로 구성될 수 있다. 변환 블록의 크기는 레지듀얼 블록과 마찬가지로 현재 블록 또는 현재 블록으로부터 분할된 블록 중 어느 하나와 동일할 수 있다. 양자화부로 전달된 변환 계수들은 양자화된 값으로 표현될 수 있다.

[109] 또한, 인코더는 변환 계수가 양자화되기 전에 추가적인 변환을 수행할 수 있다. 도 8에 도시된 바와 같이, 전술한 변환 방법은 1차 변환(primary transform)으로 지칭되고, 추가적인 변환은 2차 변환(secondary transform)으로 지칭될 수 있다.

2차 변환은 레지듀얼 블록 별로 선택적일 수 있다. 일 실시예에 따라, 인코더는 1차 변환만으로 저주파 영역에 에너지를 집중시키기 어려운 영역에 대해 2차 변환을 수행하여 코딩 효율을 향상시킬 수 있다. 예를 들어, 레지듀얼 값들이 레지듀얼 블록의 수평 또는 수직 방향 이외의 방향에서 크게 나타나는 블록에 대해 2차 변환이 추가될 수 있다. 인트라 예측된 블록의 레지듀얼 값들은 인트라 예측된 블록의 레지듀얼 값들에 비해 수평 또는 수직 방향 이외의 방향으로 변화할 확률이 높을 수 있다. 이에 따라, 인코더는 인트라 예측된 블록의 레지듀얼 신호에 대해 2차 변환을 추가적으로 수행할 수 있다. 또한, 인코더는 인트라 예측된 블록의 레지듀얼 신호에 대해 2차 변환을 생략할 수 있다.

[110] 다른 예로, 현재 블록 또는 레지듀얼 블록의 크기에 따라, 2차 변환 수행 여부가 결정될 수 있다. 또한, 현재 블록 또는 레지듀얼 블록의 크기에 따라 크기가 서로 다른 변환 커널이 사용될 수 있다. 예를 들어, 너비 또는 높이 중 짧은 변의 길이가 제1 기 설정된 길이 보다 짧은 블록에 대해서는 8X8 2차 변환이 적용될 수 있다. 또한, 너비 또는 높이 중 짧은 변의 길이가 제2 기 설정된 길이 보다 긴 블록에 대해서는 4X4 2차 변환이 적용될 수 있다. 이때, 제1 기 설정된 길이는 제2 기 설정된 길이 보다 큰 값일 수 있으나, 본 개시가 이에 제한되는 것은 아니다. 또한, 2차 변환은 1차 변환과 달리 수직 변환 및 수평 변환으로 분리되어 수행되지 않을 수 있다. 이러한 2차 변환은 저대역 비-분리 변환(Low Frequency Non-Separable Transform, LFNST)으로 지칭될 수 있다.

[111] 또한, 특정 영역의 비디오 신호의 경우, 급격한 밝기 변화로 인해 주파수 변환을 수행하여도 고주파 대역 에너지가 줄어들지 않을 수 있다. 이에 따라, 양자화에 의한 압축 성능이 저하될 수 있다. 또한, 레지듀얼 값이 드물게 존재하는 영역에 대해 변환을 수행하는 경우, 인코딩 시간 및 디코딩 시간이 불필요하게 증가할 수 있다. 이에 따라, 특정 영역의 레지듀얼 신호에 대한 변환은 생략될 수 있다. 특정 영역의 레지듀얼 신호에 대한 변환 수행 여부는 특정 영역의 변환과 관련된 신택스 요소에 의해 결정될 수 있다. 예를 들어, 상기 신택스 요소는 변환 스킵 정보(transform skip information)를 포함할 수 있다. 변환 스킵 정보는 변환 스킵 플래그(transform skip flag)일 수 있다. 레지듀얼 블록에 대한 변환 스킵 정보가 변환 스킵을 나타내는 경우, 해당 레지듀얼 블록에 대한 변환이 수행되지 않는다. 이 경우, 인코더는 해당 영역의 변환이 수행되지 않은 레지듀얼 신호를 곧바로 양자화할 수 있다. 도 8을 참조하여 설명된 인코더의 동작들은 도 1의 변환부를 통해 수행될 수 있다.

[112] 전술한 변환 관련 신택스 요소들은 비디오 신호 비트스트림으로부터 파싱된 정보일 수 있다. 디코더는 비디오 신호 비트스트림을 엔트로피 디코딩하여 변환 관련 신택스 요소들을 획득할 수 있다. 또한, 인코더는 변환 관련 신택스 요소들을 엔트로피 코딩하여 비디오 신호 비트스트림을 생성할 수 있다.

[113] 도 9은 인코더 및 디코더가 변환 계수를 역변환하여 레지듀얼 신호를 획득하는 방법을 구체적으로 나타내는 도면이다. 이하 설명의 편의를 위해, 인코더 및

디코더 각각의 역변환부를 통해 역변환 동작이 수행되는 것으로 설명한다. 역변환부는 역양자화된 변환 계수를 역변환하여 레지듀얼 신호를 획득할 수 있다. 먼저, 역변환부는 특정 영역의 변환 관련 신택스 요소로부터 해당 영역에 대한 역변환이 수행되는지 검출할 수 있다. 일 실시예에 따라, 특정 변환 블록에 대한 변환 관련 신택스 요소가 변환 스킵을 나타내는 경우, 해당 변환 블록에 대한 역변환이 생략될 수 있다. 이 경우, 변환 블록에 대해 전송한 1차 역변환 및 2차 역변환이 모두 생략될 수 있다. 또한, 역양자화된 변환 계수는 레지듀얼 신호로 사용될 수 있다. 예를 들어, 디코더는 역양자화된 변환 계수를 레지듀얼 신호로 사용하여 현재 블록을 복원할 수 있다.

[114] 다른 일 실시예에 따라, 특정 변환 블록에 대한 변환 관련 신택스 요소가 변환 스킵을 나타내지 않을 수 있다. 이 경우, 역변환부는 2차 변환에 대한 2차 역변환 수행 여부를 결정할 수 있다. 예를 들어, 변환 블록이 인트라 예측된 블록의 변환 블록인 경우, 변환 블록에 대한 2차 역변환이 수행될 수 있다. 또한, 변환 블록에 대응하는 인트라 예측 모드에 기초하여 해당 변환 블록에 사용되는 2차 변환 커널이 결정될 수 있다. 다른 예로, 변환 블록의 크기에 기초하여 2차 역변환 수행 여부가 결정될 수도 있다. 2차 역변환은 역양자화 과정 이후 1차 역변환이 수행되기 전에 수행될 수 있다.

[115] 역변환부는 역양자화된 변환 계수 또는 2차 역변환된 변환 계수에 대한 1차 역변환을 수행할 수 있다. 1차 역변환의 경우, 1차 변환과 마찬가지로 수직 변환 및 수평 변환으로 분리되어 수행될 수 있다. 예를 들어, 역변환부는 변환 블록에 대한 수직 역변환 및 수평 역변환을 수행하여 레지듀얼 블록을 획득할 수 있다. 역변환부는 변환 블록의 변환에 사용된 변환 커널에 기초하여 변환 블록을 역변환할 수 있다. 예를 들어, 인코더는 복수의 사용 가능한 변환 커널들 중 현재 변환 블록에 적용된 변환 커널을 지시하는 정보를 명시적 또는 묵시적으로 시그널링할 수 있다. 디코더는 시그널링된 변환 커널을 나타내는 정보를 이용하여 복수의 사용 가능한 변환 커널들 중 변환 블록의 역변환에 사용될 변환 커널을 선택할 수 있다. 역변환부는 변환 계수에 대한 역변환을 통해 획득된 레지듀얼 신호를 이용하여 현재 블록을 복원할 수 있다.

[116] 도 10은 본 개시의 일 실시예에 따른 Intra Subpartitions(ISP) mode의 개념을 나타낸 도면이다.

[117] 도 10에서 코딩 블록의 인트라 예측 모드(1010)를 참조하면, 하나의 코딩 블록 전체가 예측될 수 있다. 하지만 Intra Subpartition(ISP)에 의하면 코딩 블록은 적어도 하나의 블록으로 분할될 수 있다.

[118] 예를 들어, 도 10에서 인트라 서브 파티션 예측 모드(1020)를 참조하면, 하나의 코딩 블록은 복수개의 변환 블록으로 나누어질 수 있다. 코딩 블록의 크기에 따라 2 개 혹은 4개로 변환 블록을 가질 수 있다. 또한, 코딩 블록에 따라 가로 방향 혹은 세로 방향으로 그 변환 블록이 나누어 질 수 있다. ISP모드는 참조 라인 인덱스 0 일 때만 동작하며 MPM 인덱스로만 해당 화면내 예측 모드를 알려 줄

수 있다. 또한, MPM 리스트는 DC모드를 제외하고 구성될 수 있다. 본 개시에서 화면내 예측 모드는 인트라 예측 모드와 동일한 의미이다.

[119] 도 11은 본 개시의 일 실시예에 따른 복수개의 참조 샘플 라인을 사용하여 예측하는 화면내 예측 모드 및 MPM구성을 위한 이웃 블록의 위치를 나타낸 도면이다.

[120] 도 11에서 그림(1110)은 현재 코딩 블록과 복수개의 참조 샘플 라인과의 관계를 나타낸다. 본 개시에서 현재 코딩 블록은 단순히 현재 블록으로 지칭될 수 있다. 현재 코딩 블록은 복수개의 참조 라인 중 하나를 사용하여 화면내 예측을 수행할 수 있다. 선택 요소 `intra_luma_ref_idx[ x0 ][ y0 ]` 를 이용하여 참조 샘플 라인 인덱스를 알려 줄 수 있다. 선택 요소 `intra_luma_ref_idx[ x0 ][ y0 ]` 는 루마 컴포넌트에 한해 적용될 수 있으며 참조 샘플 라인 인덱스가 0보다 큰 경우는 MPM 인덱스로 화면내 예측 모드를 알려 줄 수 있다.

[121] 예를 들어 참조 샘플 라인 인덱스가 0인 경우 현재 코딩 블록은 Reference line 0을 이용하여 인트라 예측될 수 있다. 참조 샘플 라인 인덱스가 1인 경우 현재 코딩 블록은 Reference line 1을 이용하여 인트라 예측될 수 있다. 참조 샘플 라인 인덱스가 2인 경우 현재 코딩 블록은 Reference line 2을 이용하여 인트라 예측될 수 있다. 참조 샘플 라인 인덱스가 3인 경우 현재 코딩 블록은 Reference line 3을 이용하여 인트라 예측될 수 있다.

[122] reference line 2는 사용되지 않을 수 있다. 이 경우, 참조 샘플 라인 인덱스가 2인 경우 현재 코딩 블록은 Reference line 3을 이용하여 인트라 예측될 수 있다. 하지만 이에 한정되는 것은 아니고, 참조 샘플 라인 인덱스가 3인 경우 현재 코딩 블록은 Reference line 3을 이용하여 인트라 예측 될 수 있다.

[123] 도11에서 그림(1120)은 MPM(Most Probable Modes) 구성방법을 나타낸다. MPM은 현재 코딩 유닛의 이웃 블록의 화면내(인트라) 예측 모드 기반으로 구성될 수 있다. 이웃블록은 해당 도면처럼 2곳 혹은 그 이상의 복수개를 참조할 수 있다.

[124] MPM은 그림(1120)의 A와 L의 위치에서의 화면내 모드 설정 방법에 따라 결정된 모드 기반으로 구성될 수 있다. A의 위치는 (CbW-1, -1) 일수 있고 L의 위치는 (-1, CbH-1)일 수 있다. 여기서 CbW는 현재 코딩블록의 너비일 수 있으며, CbH는 현재 코딩블록의 높이 일 수 있다. 이때, 참조 샘플 라인 인덱스 값, ISP 모드 적용 여부도 반영되어 그 구성 방법이 다를 수 있다. MPM리스트 수는 6개 인 것을 예로 들었지만 그 수가 달라도 적용 가능하다.

[125] 일 예로 현재 블록의 이웃 블록의 화면내 모드가 각도 모드가 아닌 PLANAR, DC 모드이거나, 둘 중 한가지 모드로 같은 경우 다음과 같이 MPM 리스트가 구성될 수 있다.

[126] 제1 MPM 구성 세트로 `IntraLumaRefLineIdx[ xCb ][ yCb ]` 가 0과 같고 `IntraSubPartitionsSplitType` 이 `ISP_NO_SPLIT` 인 경우로 reference line 0을 사용할 수 있다.

[127] 또한, ISP 모드가 적용되지 않는 경우일 때, 제1 MPM 구성 세트는 다음과 같이 구성될 수 있다. IntraLumaRefLineIdx[xCb][yCb]은 현재 코딩 블록의 참조 샘플 라인 인덱스를 지시하는 지시자일 수 있다. IntraSubPartitionsSplitType 은 ISP의 분할 방향을 지시하는 지시자일 수 있다. ISP_NO_SPLIT 은 ISP 가 적용되지 않음을 나타내는 지시자 일 수 있다. candModeList[]은 MPM 리스트의 후보를 저장하는 변수로 배열 값을 갖는다. 화면내 예측 모드는 각도 모드와 각도 모드가 아닌 경우로 나뉘 수 있는데 INTRA_PLANAR 와 INTRA_DC는 각도 모드가 아닌 경우를 지시한다. 각도모드는 "INTRA_ANGULAR+숫자" 형태로 표시될 수 있다. 숫자는 정해진 규칙에 의한 각도 모드의 순서/각도를 의미하는 숫자 동일 일 수 있다. 예를 들어 INTRA_ANGULAR50, INTRA_ANGULAR18, INTRA_ANGULAR46 등과 같이 표시될 수 있다. 아래 수도코드는 제 1 MPM 구성 세트를 형성하는 과정을 나타낸다.

[128]

[129] If IntraLumaRefLineIdx[xCb][yCb] is equal to 0 and IntraSubPartitionsSplitType is equal to ISP_NO_SPLIT, the following applies:

[130] candModeList[0] = candIntraPredModeA

[131] candModeList[1] = (candModeList[0] == INTRA_PLANAR) ? INTRA_DC : INTRA_PLANAR

[132] candModeList[2] = INTRA_ANGULAR50

[133] candModeList[3] = INTRA_ANGULAR18

[134] candModeList[4] = INTRA_ANGULAR46

[135] candModeList[5] = INTRA_ANGULAR54

[136]

[137] 다음은 제2 MPM 구성 세트로 IntraLumaRefLineIdx[xCb][yCb] 이 0 아닌, 1 혹은 2 일 수 있으며 사용되는 라인 인덱스는 두번째 혹은 네번째 참조 샘플 라인일 수 있다. 이 경우는 ISP모드는 적용되지 않을 수 있다. 제 2 MPM 구성 세트는 다음과 같이 구성될 수 있다.

[138] Otherwise, if IntraLumaRefLineIdx[xCb][yCb] is not equal to 0, the following applies:

[139] candModeList[0] = INTRA_ANGULAR50

[140] candModeList[1] = INTRA_ANGULAR18

[141] candModeList[2] = INTRA_ANGULAR2

[142] candModeList[3] = INTRA_ANGULAR34

[143] candModeList[4] = INTRA_ANGULAR66

[144] candModeList[5] = INTRA_ANGULAR26

[145]

[146] 다음은 제3 MPM 구성 세트로 IntraSubPartitionsSplitType 이 ISP_HOR_SPLIT인 경우로, 현재 코딩 블록에 ISP 모드가 적용되며 변환 블록이 수평방향으로

나누어지게 될 때 사용되는 구성 방법이다. ISP 모드는 참조 샘플 라인 인덱스 값이 0보다 크면 적용되지 않을 수 있다. 아래 수도코드는 제 3 MPM 구성 세트를 형성하는 과정을 나타낸다.

[147]

[148] Otherwise, if IntraSubPartitionsSplitType is equal to ISP_HOR_SPLIT, the following applies:

[149] candModeList[0] = INTRA_PLANAR

[150] candModeList[1] = INTRA_ANGULAR18

[151] candModeList[2] = INTRA_ANGULAR25

[152] candModeList[3] = INTRA_ANGULAR10

[153] candModeList[4] = INTRA_ANGULAR65

[154] candModeList[5] = INTRA_ANGULAR50

[155]

[156] 다음은 제4 MPM 구성 세트로 IntraSubPartitionsSplitType 이 ISP_VER_SPLIT인 경우로, 현재 코딩 블록에 ISP 모드가 적용되며 변환 블록이 수직방향으로 나누어지게 될 때 사용되는 구성 방법이다. ISP 모드는 참조 샘플 라인 인덱스 값이 0보다 크면 적용되지 않을 수 있다. 아래 수도코드는 제 4 MPM 구성 세트를 형성하는 과정을 나타낸다.

[157]

[158] Otherwise, if IntraSubPartitionsSplitType is equal to ISP_VER_SPLIT, the following applies:

[159] candModeList[0] = INTRA_PLANAR

[160] candModeList[1] = INTRA_ANGULAR50

[161] candModeList[2] = INTRA_ANGULAR43

[162] candModeList[3] = INTRA_ANGULAR60

[163] candModeList[4] = INTRA_ANGULAR3

[164] candModeList[5] = INTRA_ANGULAR18

[165]

[166] 본 발명에서는 참조 샘플 라인 인덱스 값, ISP적용 여부, 이웃블록인 L과 A의 예측 모드를 고려하여 간단한 형태의 MPM구성 방법 및 시그널링 방법을 다룬다.

[167] 먼저 MPM구성 방법을 설명하면 이하와 같다. 현재 블록은 이웃 블록의 화면내 예측 모드와 유사한 모드를 가질 가능성이 높아, 인코더 또는 디코더는 이를 기반으로 MPM구성 할 수 있다. MPM구성은 이웃 블록 L과 A의 모드에 따라 3가지 경우로 구분할 수 있다.

[168] 제1 이웃블록 모드 경우로 L과 A 모두 각도 모드가 아닌 PLANAR 혹은 DC 모드일 수 있다.

[169] 제1-1 MPM 구성방법은 다음과 같을 수 있다. 구성 요소는 다음과 같을 수 있고

그 순서는 다르게 구성할 수 있다. 제1-1 MPM 구성방법은 참조 샘플 인덱스, ISP적용 여부 및 ISP의 분할방향 모드 등에 따라 최적화된 순서를 구성하여 사용할 수 있다.

[170] candModeList[0] = INTRA_ANGULAR50

[171] candModeList[1] = INTRA_ANGULAR18

[172] candModeList[2] = INTRA_ANGULAR3

[173] candModeList[3] = INTRA_ANGULAR65

[174] candModeList[4] = INTRA_ANGULAR34

[175] candModeList[5] = INTRA_ANGULAR25

[176]

[177] 제1-2 MPM구성방법은 다음과 같을 수 있다. 구성 요소는 다음과 같을 수 있고 그 순서는 다르게 구성될 수 있다. 제1-2 MPM구성방법은 참조 샘플 인덱스, ISP적용 여부 및 ISP의 분할방향 모드 등에 따라 최적화된 순서를 구성하여 사용할 수 있다.

[178] candModeList[0] = INTRA_ANGULAR50

[179] candModeList[1] = INTRA_ANGULAR18

[180] candModeList[2] = INTRA_ANGULAR2

[181] candModeList[3] = INTRA_ANGULAR66

[182] candModeList[4] = INTRA_ANGULAR34

[183] candModeList[5] = INTRA_ANGULAR25

[184]

[185] 제1-3 MPM구성방법은 제1-1 MPM 구성방법에서 INTRA_ANGULAR34를 DC_IDX로 변경된 조합으로 구성할 수 있다. 제1-3 MPM구성방법의 구성 순서는 다르게 구성될 수 있다. 제1-3 MPM구성방법은 참조 샘플 인덱스, ISP적용 여부 및 ISP의 분할방향 모드 등에 따라 최적화된 순서를 사용할 수 있다.

[186] 제1-4 MPM구성방법은 제1-2 MPM 구성방법에서 INTRA_ANGULAR34를 DC_IDX로 변경된 조합일 수 있다. 제1-4 MPM구성방법의 구성 순서는 다르게 구성될 수 있다. 제1-4 MPM구성방법은 참조 샘플 인덱스, ISP적용 여부 및 ISP의 분할방향 모드 등에 따라 최적화된 순서를 구성하여 사용할 수 있다.

[187] 제2 이웃블록 모드 경우로 L과 A에서 적어도 하나는 DC 모드이고 하나는 각도 모드일 수 있다.

[188] 제2-1 MPM 구성방법은 다음과 같을 수 있고 그 순서는 다르게 구성될 수 있다. maxAB는 이웃 블록 L과 A의 화면내 예측 모드 중 큰 인덱스 값을 나타낸다. 이 경우는 하나의 각도 모드와 하나의 DC 모드인 경우로 각도 모드 인덱스가 DC 모드 인덱스 값보다 큰 경우이다. maxAB로 결정된 모드 인덱스에 오프셋 값을 적용하여 아래 예시처럼 복수개의 후보들을 만들 수 있다. 아래에서는 4개의 추가 모드를 생성하는 것을 예로 들었다. 오프셋 값은 -1, +1, -2, +2를 적용한 것을 예로 들었으나 이에 한정하지 않는다. 예를 들어 -1, +1, -3, +3 일수도 있고,

-2, +2, -3, +3일 수도 있다.

[189] candModeList [0] = maxAB

[190] candModeList [1] = DC_IDX

[191] candModeList [2] = 2+((maxAB+61)%64) //-1

[192] candModeList [3] = 2+((maxAB-1)%64) //+1

[193] candModeList [4] = 2+((maxAB+60)%64) //-2

[194] candModeList [5] = 2+((maxAB)%64) //+2

[195]

[196] 제3 이웃 블록 모드 경우로 L과 A이 둘 다 각도 모드일 수 있다. L과 A의 각도 모드는 같은 각도 모드이거나 서로 다른 각도 모드일 수 있다.

[197] 제3-1 MPM 구성방법은 다음과 같을 수 있고 그 순서는 다르게 구성될 수 있다. maxAB는 이웃 블록 L과 A의 화면내 예측 모드 중 큰 인덱스 값을 나타낸다. minAB는 이웃 블록 L과 A의 화면내 예측 모드 중 작은 인덱스 값을 나타낸다. 이웃블록 L과 A 모드는 각도 모드이기 때문에 오프셋 값을 적용하여 새로운 모드를 생성하는 경우 서로 같아 질 수 있기 때문에 maxAB-minAB의 차이 값이 특정 값 안에 있는 경우와 아닌 경우로 구분하여 오프셋 값도 다르게 설정될 수 있다. 아래 예시에는 두 모드의 차이 값이 3과 61 사이에 있는 경우 오프셋 값을 -1, +1, -2로 설정하고 그 이외의 경우는 -3, +3, -4를 설정하는 것으로 예를 들었다. 모드간 차이 값의 범위를 설정하는 구간과 오프셋 값은 위의 예로 한정하지 않는다. 변수 leftIntraDir 은 왼쪽 이웃블록(L)의 화면내 예측 모드를 지시할 수 있다. 변수 aboveIntraDir 은 위쪽 이웃블록(A)의 화면내 예측 모드를 지시할 수 있다.

[198] candModeList [0] = leftIntraDir

[199] candModeList [1] = aboveIntraDir

[200] candModeList [2] = DC_IDX

[201] If maxAB-minAB is in the range of 3 to 61

[202] candModeList [3] = 2+((maxAB+61)%64) //-1: 오프셋

[203] candModeList [4] = 2+((maxAB-1)%64) //+1: 오프셋

[204] candModeList [5] = 2+((maxAB+60)%64) //-2: 오프셋

[205] Else

[206] candModeList [3] = 2+((maxAB+59)%64) //-3: 오프셋

[207] candModeList [4] = 2+((maxAB+1)%64) //+3: 오프셋

[208] candModeList [5] = 2+((maxAB+58)%64) //-4: 오프셋

[209]

[210] 도 12는 본 개시의 일 실시예에 따른 Coding Unit의 선택 구조를 나타낸 도면이다.

[211] 현재 코딩 유닛(x0, y0)의 CuPredMode[x0][y0] 값이 MODE_INTRA이고 treeType 이 SINGLE_TREE OR DUAL_TREE_LUMA이면, 디코더는

`intra_luma_ref_idx[ x0 ][ y0 ]`, `intra_subpartitions_mode_flag[ x0 ][ y0 ]`,
`intra_subpartitions_split_flag[ x0 ][ y0 ]`, `intra_luma_mpm_flag[ x0 ][ y0 ]`,
`intra_luma_mpm_idx[ x0 ][ y0 ]` 또는 `intra_luma_mpm_remainder[ x0 ][ y0 ]` 의
 정보를 비트 스트림으로부터 획득하거나 기 설정된 방법에 따라 유추한 값을
 사용할 수 있다. `CuPredMode[ x0 ][ y0 ]` 값은 인트라, 인터, Intra Block Copy
 모드를 나타낼 수 있다, `treeType`은 루마와 크로마가 같은 구조로 분할되는지
 여부를 나타낼 수 있다. `intra_luma_ref_idx[ x0 ][ y0 ]`은 참조 샘플 라인을 지시할
 수 있다. `intra_subpartitions_mode_flag[ x0 ][ y0 ]`은 코딩 유닛이 복수개의 변환
 블록으로 분할 되어 예측되는 모드 여부를 지시할 수 있다.

`intra_subpartitions_split_flag[ x0 ][ y0 ]`은 코딩 유닛이 복수개의 변환 블록으로
 분할되어 예측 될 때 분할 방향을 수평 혹은 수직방향인지를 지시 할 수 있다.
`intra_luma_mpm_flag[ x0 ][ y0 ]`은 코딩 유닛의 모드 시그널링 방법으로 MPM
 리스트를 사용했는지 여부를 알려주는 것을 지시할 수 있다.

`intra_luma_mpm_idx[ x0 ][ y0 ]`는 mpm 리스트 중 몇 번째 인지를 알려주는
 지시자 일 수 있다. `intra_luma_mpm_remainder[ x0 ][ y0 ]`은 MPM 리스트
 모드들을 제외한 나머지 모드 들에서의 인덱스 값을 지시 할 수 있다.

[212] 각각의 선택스 구성 요소는 서로 다른 이진 부호화 방법으로 변환/코딩 될 수
 있다. 예를 들어, 선택스 구성요소는 context-adaptive arithmetic entropy-coded
 syntax element(ae(v)) 또는 fixed-pattern bit string(f(n))으로 부호화되어 있을 수
 있다.

[213] 도 13은 본 개시의 일 실시예에 따른 `intra_mode_planar_flag`를 사용하여 화면내
 예측 모드의 시그널링을 나타내는 선택스 구조를 나타낸 도면이다.

[214] 도 13은 `intra_mode_planar_flag[x0][y0]`를 이용한 선택스 구조이다.

`intra_mode_planar_flag[x0][y0]`는 현재 코딩 유닛(현재 블록)의 화면내 예측
 모드가 Planar 모드 인지 아닌지를 나타내는 정보이다.

`intra_mode_planar_flag[x0][y0]`가 1인 경우 Planar 모드가 사용됨을 나타낼 수
 있다. 또한 `intra_mode_planar_flag[x0][y0]`가 0인 경우 planar 모드가 사용되지
 않음을 나타낼 수 있다. 디코더는 비트스트림으로부터

`intra_mode_planar_flag[x0][y0]` 를 획득할 수 있다.

[215] planar 모드를 사용하는지 여부를 나타내는

정보(`intra_mode_planar_flag[x0][y0]`)가 0과 같고 $(y0 \% CtbSizeY) > 0$ 을 만족하는
 경우에 디코더는 `intra_luma_ref_idx[x0][y0]`를 파싱할 수 있다. planar 모드를
 사용하는지 여부를 나타내는 정보(`intra_mode_planar_flag[x0][y0]`)가 1인 경우
 디코더는 참조 샘플 라인 인덱스가 0임을 유추할 수 있고 참조샘플 라인
 인덱스가 0보다 큰 경우 Planar 모드가 사용되지 않을 수 있다. 라인 인덱스가
 0보다 크다는 것은 라인 인덱스가 0이 아님을 나타낼 수 있다.

[216] 또한, 참조 샘플 라인 인덱스가 0이 아닌 경우, 디코더는 planar 모드를 사용하지
 않으므로, 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보를

비트스트림으로부터 획득하지 않을 수 있다. 또한, 현재 블록의 planar 모드를 사용하는지 여부를 나타내는 정보가 비트스트림으로부터 획득되지 않는 경우, planar 모드를 사용하는지 여부를 나타내는 정보는 planar 모드를 사용하지 않음을 나타낼 수 있다. 또한, 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보를 planar 모드를 사용하지 않음으로 설정할 수 있다. 디코더 및 인코더는 planar 모드를 사용하는지 여부를 나타내는 정보를 따로 송수신하지 않으므로 디코딩 및 인코딩 효율을 높일 수 있다.

[217] 또한, 디코더는 ISP관련 플래그 2개를(intra_subpartitions_mode_flag[x0][y0], intra_subpartitions_split_flag[x0][y0]) 파싱할 수 있다. intra_luma_mpm_flag[x0][y0]는 다음 조건 (intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0 && intra_mode_Planar_flag[x0][y0] == 0)을 모두 만족하는 경우 파싱될 수 있다. 조건이 만족되지 않는 경우, intra_luma_mpm_flag[x0][y0]는 유추될 수 있다. 디코더는 intra_luma_mpm_flag[x0][y0] 값에 따라 intra_luma_mpm_idx[x0][y0]를 파싱할지 안 할지가 결정할 수 있다.

[218] 도 14는 본 개시의 일 실시예에 따른 intra_mode_planar_flag와 ISP 모드 플래그 파싱 조건을 사용하여 화면내 예측 모드의 시그널링을 나타내는 선택스 구조를 나타낸 도면이다.

[219] 도 14는 intra_mode_planar_flag[x0][y0]와 ISP 모드에서 PLANAR 와 DC 모드를 사용하지 않는 경우를 고려한 선택스 구조이다. ISP 모드에서 PLANAR 와 DC 모드를 사용하지 않기에 intra_mode_planar_flag[x0][y0]가 참인 경우, 디코더는 intra_subpartitions_mode_flag[x0][y0]을 0의 값으로 유추 할 수 있다. 디코더는 다음 조건들을 만족 하는 경우 intra_subpartitions_mode_flag[x0][y0]를 파싱할 수 있다.

[220]

[221] (intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY || cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY) && Intra_mode_Planar_flag[x0][y0] == 0)

[222]

[223] 디코더는 다음과 같은 조건인 경우 intra_luma_mpm_flag[x0][y0]를 파싱할 수 있다.

[224]

[225] (intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0 && intra_mode_Planar_flag[x0][y0] == 0) 혹은 ((intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0) || intra_mode_Planar_flag[x0][y0] == 0) 혹은 (intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)

[226]

- [227] ISP 모드에서 PLANAR 와 DC모드를 모두 사용하지 않는 경우 종래에 정의된 ISP모드일 때 사용되는 MPM구성 방법들에서 PLANAR모드를 제외하고 새로운 각도모드로 대신 구성할 수 있다. 혹은 본 발명에서 제안한 MPM구성 방법들에서 PLANAR 모드를 제외하고 새로운 각도 모드를 추가하여 구성할 수 있다. 디코더는 MPM리스트를 각도모드만으로 채울 수 있다. 또한, 도 13 또는 도 14에서 설명된 사항은 도 15 및 도 16에도 적용될 수 있다.
- [228] 도 15는 본 개시의 일 실시예에 따른 intra_mode_planar_flag와 참조샘플 라인 인덱스 조건을 사용하여 화면내 예측 모드 시그널링과 관련된 선택스 구조를 나타낸 도면이다.
- [229] 상기 도 12, 13에서 이미 설명한 내용은 생략한다. 도 15는 intra_mode_planar_flag[x0][y0]와 ISP 모드에서 DC 모드를 사용하지 않는 경우를 고려한 선택스 구조일 수 있다.
- [230] 디코더는 수신된 비트스트림으로부터 현재 블록의 참조 샘플 라인 인덱스(intra_luma_ref_idx[x0][y0])를 획득하는 단계를 수행할 수 있다. 디코더는 참조 샘플 라인 인덱스(intra_luma_ref_idx[x0][y0])가 0인지 여부를 결정(intra_luma_ref_idx[x0][y0]==0)하는 단계를 수행할 수 있다. 디코더는 참조 샘플 라인 인덱스가 0인 경우에만 planar 모드를 사용하는지 여부를 나타내는 정보(intra_mode_Planar_flag[x0][y0])를 파싱할 수 있다. 파싱한다는 것은 디코더가 수신한 비트스트림으로부터 정보를 획득함을 의미할 수 있다.
- [231] 참조 샘플 라인 인덱스가 0이 아닌 경우 planar 모드가 사용되지 않을 수 있다. 따라서 참조 샘플 라인 인덱스가 0이 아닌 경우, 비트스트림은 planar 모드를 사용하는지 여부를 나타내는 정보를 포함하지 않을 수 있다. 또한, 디코더는 참조 샘플 라인 인덱스가 0이 아닌 경우 planar 모드를 사용하는지 여부를 나타내는 정보를 planar 모드를 사용하지 않음으로 설정할 수 있다.
- [232] 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 인트라 예측 모드를 결정하는 단계를 수행할 수 있다. 인트라 예측 모드는 planar 모드, DC모드 또는 각도 모드 중 하나일 수 있다. 또한 디코더는 결정된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 수행할 수 있다. planar 모드를 사용하는지 여부를 나타내는 정보는 'planar 모드를 사용'하거나 'planar 모드를 사용하지 않음'을 나타낼 수 있다. 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용함을 나타내는 경우, 현재 블록을 planar 모드에 기초하여 예측할 수 있다. 또한 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하지 않음을 나타내는 경우, planar 모드가 아닌 예측 모드에 기초하여 현재 블록을 예측할 수 있다.
- [233] intra_luma_mpm_flag[x0][y0]는 다음과 같은 조건인 경우 파싱될 수 있다. (intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0 && intra_mode_Planar_flag[x0][y0]==0). 즉, 디코더는 intra_mode_Planar_flag[x0][y0]==0인 경우 intra_luma_mpm_flag[x0][y0]를

파싱할 수 있다.

- [234] 위에서는 MPM(Most Probable Modes) 사용여부를 나타내는 정보가 planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 획득되는 것으로 설명하였으나, 이에 한정되는 것은 아니다. 디코더는 소정의 조건에 기초하여 MPM(Most Probable Modes) 사용여부를 나타내는 정보를 수신된 비트스트림으로부터 획득할 수 있다. 또한 디코더는 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0])가 MPM을 사용함을 나타내는 것으로 결정한 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 파싱할 수 있다. 디코더는 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0])가 MPM을 사용하지 않음을 나타내는 것으로 결정한 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 획득하지 않고, planar 모드를 사용하는지 여부를 나타내는 정보를 사용하지 않음으로 설정할 수 있다. 도 13 또는 도 14의 설명과 중복되는 설명은 생략한다.
- [235] 또한, 제1 이웃블록의 인트라 모드는 상술한 바와 같이 정의된 MPM구성 방법을 사용하고 제2 이웃블록의 인트라 모드와 제3 이웃블록의 인트라 모드는 제2-1 MPM 구성방법 과 제3-1 MPM 구성방법으로 동작 할 수 있다. 예를 들어 도11에서 기술한 내용에서 제1 이웃블록의 인트라 모드는 정의된 제1 MPM 구성 세트부터 제4 MPM 구성 세트까지 사용하고 제2 이웃블록의 인트라 모드와 제3 이웃블록의 인트라 모드는 제2-1 MPM 구성방법 과 제3-1 MPM 구성방법을 사용할 수 있다.
- [236] 또한, 제1 이웃블록의 인트라 모드는 "참조 샘플 라인 인덱스가 0이고 ISP 모드를 사용하지 않는 경우"와 "참조 샘플 라인 인덱스가 0이 아니고 ISP 모드를 사용하지 않는 경우"만 고려할 수 있다. 또한, 제2 이웃블록의 인트라 모드와 제3 이웃블록의 인트라 모드는 제2-1 MPM 구성방법 과 제3-1 MPM 구성방법으로 동작 할 수 있다. 예를 들어 도11에서 기술한 내용처럼 제1 이웃블록의 인트라 모드가 위의 두가지 경우로 나누어 MPM구성을 하고 제2 이웃블록의 인트라 모드와 제3 이웃블록의 인트라 모드는 제2-1 MPM 구성방법 과 제3-1 MPM 구성방법으로 동작 할 수 있다.
- [237] 또한 디코더는 참조 샘플 라인 인덱스가 0이고 ISP모드를 사용하지 않는 경우 도 11에서 기술한 제1 MPM 구성 방법을 사용할 수 있다. 참조 샘플 라인 인덱스가 0보다 큰 경우와 참조 샘플 라인 인덱스가 0이고 ISP모드를 사용하는 경우, 디코더는 도8에서 기술된 제1-1 MPM 구성방법, 제1-2 MPM 구성방법, 제1-3 MPM 구성방법, 제1-4 MPM 구성방법들 중 하나를 사용할 수 있다.
- [238] 상기 내용은 디코더 관점에서 기술되었으나 인코더에서도 동일하게 동작될 수 있다.
- [239] 도 16은 본 개시의 일 실시예에 따른 MPM 시그널링에 관련된 선택스 구조를 나타낸 것으로 참조 라인 인덱스가 우선 파싱/시그널링 되는 것을 고려한 선택스 구조를 나타낸 도면이다.

- [240] 도 16에서 CUPredMode[x0][y0] 가 MODE_INTRA라이고 pcm_flag[x0][y0]가 0인 일반적인 화면내 예측 모드인 경우, 디코더는 $y0\%CtbSizeY > 0$ 가 참인지 여부를 결정할 수 있다. $y0\%CtbSizeY > 0$ 는 현재 블록이 코딩 트리 유닛의 최상측에 포함되는지 여부를 결정하는 단계를 나타낼 수 있다. $y0$ 는 현재 블록의 좌상측 픽셀의 y축 좌표이다. CtbSizeY는 코딩 트리 유닛의 크기를 나타낼 수 있다. 또한 $(y0\%CtbSizeY > 0 == 1)$ 는 현재 블록이 코딩 트리 유닛의 최상측에 포함되지 않은 경우를 의미할 수 있다.
- [241] 현재 블록이 코딩 트리 유닛의 최상측에 포함되지 않은 경우, 디코더는 참조 샘플 라인 인덱스(intra_luma_ref_idx[x0][y0])를 파싱할 수 있다. 파싱한다는 것은 수신된 비트스트림으로부터 선택 요소를 획득하는 것을 의미한다.
- [242] 참조 샘플 라인 인덱스가 0이면 planar 모드를 사용하는지 여부를 나타내는 정보(intra_mode_Planar_flag[x0][y0])를 파싱해서 0 인지 1인지를 확인하고 1이면 화면내 예측 모드는 Planar 모드가 된다. 혹은 0 이면 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0]) 파싱 후 intra_luma_mpm_idx[x0][y0] 혹은 intra_luma_mpm_remainder[x0][y0]정보 기반으로 화면내 예측 모드를 획득/확인할 수 있다.
- [243] 참조 샘플 라인 인덱스(Intra_luma_ref_idx[x0][y0])를 먼저 시그널링/획득하는 경우, 디코더는 참조 샘플 라인 인덱스(Intra_luma_ref_idx[x0][y0])가 0인지 여부를 결정하는 단계를 수행할 수 있다. 참조 샘플 라인 인덱스가 0보다 큰 값을 지시하는 경우 현재 블록의 참조 샘플 라인이 가장 가까이 인접한 참조 샘플 라인이 아닐 수 있어 Planar 모드를 사용하지 않을 수 있다. 이런 경우는 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보를 0으로 infer할수 있어 시그널링 및 파싱하여 정보 획득하는 과정을 수행하지 않고도 그 값을 알 수 있다. 이는 Planar 모드 대신 MPM 혹은 MPM이 아닌 방법의 변수 기반으로 예측 모드 정보를 시그널링/획득할 수 있다.
- [244] 본 개시의 일 실시예에서, 인코더 또는 디코더는 MPM 사용여부를 나타내는 정보를 사용할 수이다. 또한 인코더는 참조 샘플 라인 인덱스(Intra_luma_ref_idx[x0][y0])가 0인 경우, MPM 사용여부를 나타내는 정보를 비트스트림으로 생성할 수 있다. 디코더는 참조 샘플 라인 인덱스(Intra_luma_ref_idx[x0][y0])가 0인 경우, MPM 사용여부를 나타내는 정보를 비트스트림으로부터 획득할 수 있다.
- [245] 디코더는 MPM 사용여부를 나타내는 정보가 MPM을 사용함을 나타내고, 참조 샘플 라인 인덱스가 0인 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 상기 비트스트림으로부터 획득하는 단계를 수행할 수 있다. 디코더는 MPM 사용여부를 나타내는 정보가 MPM을 사용하지 않음을 나타내거나, 참조 샘플 라인 인덱스가 0이 아닌 경우, planar 모드를 사용하는지 여부를 나타내는 정보를 상기 비트스트림으로부터 획득하지 않을 수 있다. planar 모드를 사용하는지 여부를 나타내는 정보가 비트스트림으로부터 획득되지 않는 경우, planar 모드를

사용하는지 여부를 나타내는 정보는 planar 모드를 사용하지 않음을 나타낼 수 있다.

- [246] 또한, 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보(intra_mode_Planar_flag[x0][y0])를 파싱 후에는 intra_subpartitions_mode_flag[x0][y0]와 intra_mode_Planar_flag[x0][y0]와 intra_suppartitions_split_flag[x0][y0]를 파싱 할 수 있다. MPM 사용여부를 나타내는 정보(Intra_luma_mpm_flag[x0][y0])는 intra_luma_ref_idx[x0][y0]와 intra_subpartitions_mode_flag[x0][y0]와 intra_luma_Planar_flag[x0][y0]를 기반하여 파싱 될 수 있다. 예를 들어 intra_luma_ref_idx[x0][y0]가 참조 샘플 라인 0번째를 사용함을 지시하고 intra_subpartitions_mode_flag[x0][y0]가 ISP모드를 사용하지 않는 것을 지시하고 intra_luma_Planar_flag[x0][y0]가 Planar 모드가 아님을 지시하는 경우 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0]) MPM 사용여부를 나타내는 정보를 파싱 할 수 있다. 또한, intra_luma_Planar_flag[x0][y0]가 0을 지시하고 intra_luma_ref_idx[x0][y0]가 0보다 큰 값을 지시하면 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0])는 1로 infer할 수 있다. 또한, intra_luma_Planar_flag[x0][y0]가 0을 지시하고 intra_luma_ref_idx[x0][y0]가 0을 지시하고 intra_suppartitions_mode_flag[x0][y0]가 1을 지시하면 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0]) MPM 사용여부를 나타내는 정보는 1로 infer할 수 있다.

- [247] intra_suppartitions_mode_flag[x0][y0]가 1을 지시하면 ISP 모드가 적용되었다는 것을 의미할 수 있다. 혹은 intra_luma_Planar_flag[x0][y0]가 0을 지시하고 intra_suppartitions_mode_flag[x0][y0]가 1을 지시하면 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0])는 1로 infer할 수 있다. ISP 모드는 참조 샘플 라인 인덱스가 0인 경우로 현재 예측 블록에 가장 인접한 참조 샘플 라인을 사용하는 경우에만 적용될 수 있기에 상기 planar 모드를 사용하는지 여부를 나타내는 정보와 같은 지시자를 생략할 수 있다. 모든 참조 샘플 라인 인덱스에서 적용되는 경우에도 상기 planar 모드를 사용하는지 여부를 나타내는 정보 지시자를 생략할 수 있다. 이렇게 mpm flag를 infer하는 방법은 Planar 플래그가 먼저 시그널링/파싱되고 MPM 사용여부를 나타내는 정보가 뒤에 오는 선택스 구조인 상기 도 13에서 도 17에도 공통으로 적용할 수 있다. 그러한 내용은 시맨틱에서 별도로 기술될 수 있다.

- [248] CUPredMode[x0][y0]는 예측 모드를 지시하며 Intra, Intra, IBC를 지시할 수 있다. intra_luma_ref_idx[x0][y0]는 참조 샘플 라인 인덱스를 지시하며 0, 1, 2일 수 있으며 현재 예측 블록의 이웃 참조 샘플 라인 0 번째 1번째, 3번째를 의미할 수 있다. intra_mode_Planar_flag[x0][y0]는 planar 모드를 사용하는지 여부를 나타내는 정보 일수 있다. intra_luma_mpm_flag[x0][y0]는 MPM 사용여부를 나타내는 정보 또는 화면내(인트라) 예측 모드가 MPM(Most Probable Mode)

리스트에 포함되는지 여부를 지시하는 변수 일수 있다.

`intra_subpartitions_mode_flag[x0][y0]`는 ISP모드 여부를 지시하는 변수 일수 있다. `intra_suppartitions_split_flag[x0][y0]`는 ISP모드를 사용하는 경우 수직/수평방향에 해당하는 분할 모드를 지시하는 변수 일수 있다. `intra_luma_mpm_idx[x0][y0]`는 MPM 리스트내의 위치를 지시하는 변수 일수 있다.

`intra_luma_mpm_remainder[x0][y0]`는 MPM리스트에 포함되지 않는 화면내 예측 모드들 중 하나를 지시하는 변수 일수 있다. 상기 모든 발명에서 같은 변수명으로 사용되는 경우 일반적으로 같은 의미로 해석할 수 있다.

[249] 도 17은 본 개시의 일 실시예에 따른 MPM 시그널링에 관련된 선택스 구조를 나타낸 것으로 참조 라인 인덱스 값에 상관없이 Planar flag를 파싱하는 것을 고려한 선택스 구조를 나타낸 도면이다.

[250] 도 17에서 `CUPredMode[x0][y0]`가 `MODE_INTRA`라이고 `pcm_flag[x0][y0]`가 0인 일반적인 화면내 예측 모드인 경우, `intra_luma_ref_idx[x0][y0]` 파싱하고, `intra_mode_Planar_flag[x0][y0]`를 파싱할 수 있다. `intra_mode_Planar_flag`는 `intra_luma_ref_idx` 값과 별개로 시그널링 될 수 있다. 이는 `intra_luma_ref_idx`가 지시하는 값이 현재 예측 블록에 가장 가까이 인접한 참조 샘플 라인이 아닌 경우에도 Planar 모드를 사용할 수 있음을 의미할 수 있다.

`Intra_mode_Planar_flag[x0][y0]`가 지시하는 값이 0인지 1인지를 확인하고 1이면 화면내 예측 모드는 Planar 모드가 된다. 0이면 `intra_luma_mpm_flag[x0][y0]` 파싱 후 `intra_luma_mpm_idx[x0][y0]` 혹은 `intra_luma_mpm_remainder[x0][y0]` 정보 기반으로 화면내 예측 모드를 획득/확인할 수 있다.

`intra_mode_Planar_flag[x0][y0]` 파싱 후에는 `intra_subpartitions_mode_flag[x0][y0]`와 `intra_mode_Planar_flag[x0][y0]`와 `intra_suppartitions_split_flag[x0][y0]`를 파싱할 수 있다. `Intra_luma_mpm_flag[x0][y0]`는 `intra_luma_ref_idx[x0][y0]`와 `intra_subpartitions_mode_flag[x0][y0]`와 `intra_luma_Planar_flag[x0][y0]`를 기반하여 파싱 될 수 있다. 예를 들어 `intra_luma_ref_idx[x0][y0]`가 참조 샘플 라인 0번째를 사용함을 지시하고 `intra_subpartitions_mode_flag[x0][y0]`가 ISP모드를 사용하지 않는 것을 지시하고 `intra_luma_Planar_flag[x0][y0]`가 Planar 모드가 아님을 지시하는 경우 `intra_luma_mpm_flag[x0][y0]`를 파싱 할 수 있다.

[251]

[252] 도 18은 본 개시의 일 실시예에 따른 Planar 플래그와 MPM 사용여부를 나타내는 정보를 시그널링/파싱하는 선택스 구조를 나타낸 도면이다.

[253] 도 18에서 `CUPredMode[x0][y0]`가 `MODE_INTRA`라이고 `pcm_flag[x0][y0]`가 0인 일반적인 화면내 예측 모드인 경우, 디코더는 참조 샘플 라인 인덱스(`intra_luma_ref_idx[x0][y0]`)를 파싱하고, 참조 샘플 라인 인덱스가 0을 지시하면 planar 모드를 사용하는지 여부를 나타내는 정보(`intra_mode_Planar_flag[x0][y0]`)를 파싱 할 수 있다. planar 모드를 사용하는지 여부를 나타내는 정보(`Intra_mode_Planar_flag[x0][y0]`)가 지시하는

값이 0 인지 1인지를 확인하고 1이면 화면내 예측 모드는 Planar 모드가 된다. 그 값이 0 이면 MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0]) 파싱 후 intra_luma_mpm_idx[x0][y0] 혹은 intra_luma_mpm_remainder[x0][y0] 정보 기반으로 화면내 예측 모드를 획득/확인할 수 있다.

- [254] 예를 들어, MPM 사용여부를 나타내는 정보(intra_luma_mpm_flag[x0][y0])가 MPM을 사용하지 않음을 나타내는 경우, 디코더는 인트라 예측 모드를 나타내는 인덱스(intra_luma_mpm_remainder[x0][y0])를 비트스트림으로부터 획득하는 단계를 수행할 수 있다. 디코더는 인트라 예측 모드를 나타내는 인덱스에 기초하여 인트라 예측을 수행하여 상기 현재 블록을 예측하는 단계를 수행할 수 있다. 디코더는 MPM에 사용된 인트라 모드를 제외한 인트라 모드 중에서 intra_luma_mpm_remainder[x0][y0]에 기초하여 하나의 인트라 모드를 선택할 수 있다. 디코더는 선택된 인트라 모드에 기초하여 현재 블록을 예측할 수 있다.

- [255] intra_mode_Planar_flag[x0][y0] 파싱 후에는 intra_subpartitions_mode_flag[x0][y0] 와 intra_suppartitions_split_flag[x0][y0]를 파싱 할 수 있다. Intra_luma_mpm_flag[x0][y0]는 intra_luma_Planar_flag[x0][y0] 값에 기반하여 파싱 될 수 있다. 예를 들어 intra_luma_Planar_flag[x0][y0]가 Planar 모드가 아님을 지시하는 경우 intra_luma_mpm_flag[x0][y0]를 파싱 할 수 있다. 참조 샘플 인덱스가 0이 아닌 경우와 ISP모드를 사용하는 경우에는 MPM 사용여부를 나타내는 정보가 1인 값으로 infer하는 것 대신 해당 각각의 경우에 대해서도 Planar 플래그가 0을 지시하여 Planar모드가 아님을 나타내면 MPM 사용여부를 나타내는 정보를 시그널링/파싱 할 수 있다.

- [256] 본 개시의 일 실시예에서, MPM 방법의 MPM 사용여부를 나타내는 정보를 포함할 수 있다. 디코더는 참조 샘플 라인 인덱스(Intra_luma_ref_idx[x0][y0])가 0인 경우, MPM 사용여부를 나타내는 정보를 비트스트림으로부터 획득할 수 있다. 또한 인코더는 디코더는 참조 샘플 라인 인덱스(Intra_luma_ref_idx[x0][y0])가 0인 경우, MPM 사용여부를 나타내는 정보를 비트스트림으로 생성하지 않을 수 있다. 디코더는 MPM 사용여부를 나타내는 정보에 기초하여 인트라 예측을 수행하여 현재 블록을 예측하는 단계를 수행할 수 있다. 디코더 및 인코더는 Planar 모드를 사용하는지 여부를 나타내는 정보를 생략함으로써, 디코딩 및 인코딩의 효율을 높일 수 있다.

- [257] 도 19는 본 개시의 일 실시예에 따른 Planar 플래그에 기반한 참조 샘플 라인 인덱스 와 및 MPM 사용여부를 나타내는 정보에 대한 시그널링/파싱을 나타내는 선택스 구조 도면이다.

- [258] 도19에서 CUPredMode[x0][y0] 가 MODE_INTRA라이고 pcm_flag[x0][y0]가 0 인 일반적인 화면내 예측 모드인 경우, intra_mode_Planar_flag[x0][y0]를 파싱하고 그 값이 0을 지시하면 intra_luma_ref_idx[x0][y0] 파싱할 수 있다. intra_mode_Planar_flag[x0][y0]이 1을 지시하면 intra_luma_ref_idx[x0][y0]은 0으로 infer 할수 있어 별도의 시그널링/파싱이 필요하지 않을 수 있다.

Intra_mode_Planar_flag[x0][y0]가 지시하는 값이 0 인지 1인지를 확인하고 1이면 화면내 예측 모드는 Planar 모드 일수 있다. 혹은 0 이면 intra_luma_mpm_flag[x0][y0] 파싱 후 intra_luma_mpm_idx[x0][y0] 혹은 intra_luma_mpm_remainder[x0][y0]정보 기반으로 화면내 예측 모드를 획득/확인할 수 있다. intra_mode_Planar_flag[x0][y0] 파싱 후에는 intra_subpartitions_mode_flag[x0][y0] 와 intra_suppartitions_split_flag[x0][y0]를 파싱 할 수 있다. Intra_luma_mpm_flag[x0][y0]는 intra_luma_Planar_flag[x0][y0] 값에 기반하여 파싱 될 수 있다. 예를 들어 intra_luma_Planar_flag[x0][y0]가 Planar 모드가 아님을 지시하는 경우 intra_luma_mpm_flag[x0][y0]를 파싱 할 수 있다. 참조 샘플 인덱스가 0이 아닌 경우와 ISP모드를 사용하는 경우에는 MPM 사용여부를 나타내는 정보가 1인 값으로 infer하는 것 대신 해당 각각의 경우에 대해서도 Planar 플래그가 0을 지시하여 Planar모드가 아님을 나타내면 MPM 사용여부를 나타내는 정보를 시그널링/파싱 할 수 있다.

[259] 상기 제안된 방법 들에서 MPM list를 구성하는 방법은 MRL, ISP, MRL과 ISP가 적용되지 않은 일반적인 상황 모두 같은 방법으로 MPM list를 구성하고 기 설정된 MPM list 수로 구성될 수 있다. 기 설정된 MPM 구성 개수는 6개일 수 있다. 하지만 6개로 한정하지 않는다. 이렇게 구성된 MPM list에서 Planar mode만 별도의 flag로 시그널링 될 수 있고 intra_luma_ref_idx 값에 의해 0으로 infer될 수 있다. Planar mode를 제외한 나머지 MPM list는 intra_luma_mpm_flag와 intra_luma_mpm_idx를 통해 시그널링 될 수 있다. Intra_mode_Planar_flag와 ISP관련 플래그 2개와의 선택스 구조상의 위치는 서로 간에 의존성이 없이 독립적으로 정보 구성 및 획득될 수 있다.

[260] 도 20은 본 개시의 일 실시예에 따른 코딩 유닛 선택스 구조에서 intra_luma_planar_flag를 나타낸 도면이다.

[261] 도 20의 선택스 구조 및 시그널링/파싱 방법은 도 12의 설명과 큰 틀에서는 같을 수 있고 MPM flag가 참인 경우의 동작은 다음과 같을 수 있다.

[262] 실시예20-1

```
[263] If(intra_luma_mpm_flag[x0][y0]){
[264]   If(intra_luma_ref_idx[x0][y0]==0)
[265]     Intra_luma_planar_flag[x0][y0]
[266]   If( Intra_luma_planar_flag[x0][y0]==0)
[267]     Intra_luma_mpm_idx[x0][y0]
[268]   }else{
[269]     Intra_luma_mpm_remainder[x0][y0]
[270]   }
```

[271] 상기 실시예20-1에서 MPM 사용여부를 나타내는

정보(intra_luma_mpm_flag[x0][y0])가 참(예를 들어 값이 1과 같은 경우)이면 참조 샘플 라인 인덱스(intra_luma_ref_idx[x0][y0]) 값이 0과 같은 지 확인하고 같으면

planar 모드를 사용하는지 여부를 나타내는

정보(`intra_luma_planar_flag[x0][y0]`)의 값을 확인할 수 있다. planar 모드를 사용하는지 여부를 나타내는 정보(`intra_luma_planar_flag[x0][y0]`)가 지시하는 값이 참이면 Intra Luma Planar Mode를 지시할 수 있다.

[272] 이때, 상기 Planar 모드는 MPM list내에서 첫번째일 수 있다. MPM list의 구성 개수는 6개 혹은 기 설정된 최대 크기가 될 수 있다. 추가적으로 Intra Subpartitions (ISP) mode를 사용한 Planar Mode인 경우와 아닌 경우로 구분하여 context model을 사용하여 `intra_luma_planar_flag`를 Entropy coding 할 수 있다.

[273] 다른 경우로 `intra_luma_ref_idx[x0][y0]>0` 인 경우에는 planar 모드를 사용하는지 여부를 나타내는 정보(`Intra_luma_planar_flag[x0][y0]`)는 0의 값으로 유추/infer 할 수 있다. 여기서 planar 모드를 사용하는지 여부를 나타내는 정보가 0인 경우 planar 모드를 사용하지 않음을 나타낼 수 있다. 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보를 비트스트림으로부터 파싱하지 않고, planar 모드를 사용하는지 여부를 나타내는 정보를 planar 모드를 사용하지 않음으로 결정할 수 있다.

[274] 디코더는 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하는 것을 나타내는지 여부를 결정하는 단계를 수행할 수 있다. planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하지 않는 것을 나타내는 경우, 디코더는 MPM인덱스(`intra_luma_mpm_idx[x0][y0]`)를 비트스트림으로부터 획득하는 단계를 수행할 수 있다. 디코더는 MPM인덱스(`intra_luma_mpm_idx[x0][y0]`)를 비트스트림으로부터 파싱하여 해당 화면내(인트라) 예측 모드를 결정할 수 있다. planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하지 않는 것을 나타내는 경우, 디코더는 MPM 리스트에 각도 모드를 적어도 하나 추가할 수 있다.

[275] 디코더는 현재 블록의 미리 복원된 좌측 블록 또는 미리 복원된 상측 블록 중 적어도 하나에 기초하여, 인트라 예측에 포함된 적어도 하나의 각도 모드를 포함하는 MPM리스트를 생성하는 단계를 수행할 수 있다. MPM리스트는 적어도 하나의 후보 인트라 예측 모드를 포함할 수 있다. MPM리스트는 적어도 하나의 각도 모드를 포함할 수 있다. 또한 디코더는 MPM인덱스에 기초하여 상기 MPM리스트에서 하나의 각도 모드를 선택하는 단계를 수행할 수 있다.

[276] MPM인덱스(`Intra_luma_mpm_idx`)는 bypass 코딩 될 수 있다. MPM 리스트가 인트라 모드를 6개 포함하는 경우 MPM인덱스(`Intra_luma_mpm_idx[x0][y0]`)는 두번째부터 여섯번째까지 중 하나를 지시할 수 있다. 디코더는 MPM인덱스에 기초하여 MPM 리스트의 두번째 모드 부터 여섯번째 모드 중 하나의 인트라 예측 모드를 선택할 수 있다. 즉, MPM인덱스(`Intra_luma_mpm_idx[x0][y0]`)는 MPM에 포함된 5개의 후보 중 하나를 선택할 수 있다. 5개의 후보 중 적어도 하나는 각도 모드일 수 있다. 이미 설명한 바와 같이 MPM 리스트는 6개로 한정되지 않는다. 예를 들어 MPM리스트는 5개의 인트라 모드를 포함할 수 있다.

- [277] 디코더는 MPM인덱스에 기초하여 MPM리스트에서 하나의 인트라 예측 모드를 선택할 수 있다. 디코더는 선택된 인트라 예측 모드에 기초하여 인트라 예측을 수행하여 현재 블록을 예측하는 단계를 수행할 수 있다.
- [278] MPM 사용여부를 나타내는 정보(`intra_luma_mpm_flag[x0][y0]`)가 MPM을 사용하지 않음을 나타내는 경우, 디코더는 인트라 예측 모드를 나타내는 인덱스(`Intra_luma_mpm_remainder[x0][y0]`)를 상기 비트스트림으로부터 획득하는 단계를 수행할 수 있다. 여기서 MPM 사용여부를 나타내는 정보(`intra_luma_mpm_flag[x0][y0]`)가 MPM을 사용하지 않음을 나타낸다는 것은 MPM 사용여부를 나타내는 정보가 거짓 또는 0과 같은 값을 가진다는 것을 나타낼 수 있다. 디코더는 인트라 예측 모드를 나타내는 인덱스(`Intra_luma_mpm_remainder[x0][y0]`)에 기초하여 인트라 예측을 수행하여 현재 블록을 예측하는 단계를 수행할 수 있다.
- [279] 다른 방법으로 MPM 사용여부를 나타내는 정보가 참일 때 디코더는 `planar` 모드를 사용하는지 여부를 나타내는 정보를 시그널링/파싱할 수 있다. 또한 `planar` 모드를 사용하는지 여부를 나타내는 정보가 참이 아니면 디코더는 MPM인덱스(`intra_luma_mpm_idx[x0][y0]`)로 화면내 예측 모드를 알 수 있다. 여기서 MPM 사용여부를 나타내는 정보가 참이라는 것은 MPM 사용여부를 나타내는 정보가 'MPM 사용'을 나타냄을 의미한다. 또한 `planar` 모드를 사용하는지 여부를 나타내는 정보가 참이라는 것은 `planar` 모드를 사용하는지 여부를 나타내는 정보가 'planar 모드 사용'을 나타냄을 의미한다.
- [280] 이러한 동작은 실시예 20-1과 같을 수 있다. 다른 점은 MPM인덱스(`intra_luma_mpm_idx[x0][y0]`)가 기 설정된 최대 크기를 가질 수 있다. 기 설정된 값을 0보다 큰 정수 값일 수 있고 본 예시에서는 6인 것으로 한다. MPM인덱스(`intra_luma_mpm_idx[x0][y0]`)는 최대 크기 6개에 대해 `bypass coding`될 수 있다. 하지만 이에 한정되는 것은 아니며 MPM인덱스는 다양한 값을 가질 수 있다.
- [281] 상기 실시예 20-1과 달리 `intra_luma_planar_flag[x0][y0]`를 시그널링/파싱하는 조건을 다음과 같이 세분화할 수 있다.
- [282] 실시예 20-2
- [283] `If(intra_luma_mpm_flag[x0][y0]){`
- [284] `If(intra_luma_ref_idx[x0][y0]==0 && intra_subpartitions_mode_flag[x0][y0]==0)`
- [285] `Intra_luma_planar_flag[x0][y0]`
- [286] `Else If(intra_luma_ref_idx[x0][y0]==0 && intra_subpartitions_mode_flag[x0][y0]==1)`
- [287] `Intra_luma_planar_flag[x0][y0]`
- [288] `If(Intra_luma_planar_flag[x0][y0]==0)`
- [289] `Intra_luma_mpm_idx[x0][y0]`
- [290] `}else{`

[291] Intra_luma_mpm_remainder[x0][y0]

[292] }

[293] 상기 실시예 20-2는 intra_luma_planar_flag[x0][y0]가 entropy coding 될 때 참조 샘플 라인 인덱스가 0인 경우와 참조 샘플 라인 인덱스가 0이면서 ISP모드를 사용하는 경우에 따라 context model을 달리하는 것에 기초한 선택스 구조로 나타낸 것이다. 실시예 20-1과 달리 intra_luma_planar_flag의 시그널링/과싱 단계에 ISP여부를 동시에 확인하여 해당 정보 값과 context model을 구분할 수 있다.

[294] 상기 내용은 디코더 관점에서 기술되었으나 인코더에서도 동일하게 동작될 수 있다. 상기 내용중 과싱이라는 용어는 비트 스트림으로부터 정보를 획득하는 과정에 중점 설명하였으나 인코더 측면에서는 비트 스트림에 해당 정보를 구성하는 것으로 해석할 수 있다. 따라서 과싱이라는 용어를 디코더 동작으로만 한정하지 않고 인코더에서는 비트 스트림을 구성하는 행위로까지 해석할 수 있다.

[295] 예를 들어, 인트라 예측에 기초하여 비디오 신호를 인코딩하는 방법에 있어서, 인코더는 현재 블록이 MPM(Most Probable Modes)을 사용하고, 상기 현재 블록의 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보를 생성하는 단계를 수행할 수 있다. 인코더는 상기 planar 모드를 사용하는지 여부를 나타내는 정보를 포함하는 비트스트림을 생성하는 단계를 수행할 수 있다. 인코더는 프로세서 및 메모리를 포함하고, 상기 프로세서는 상기 메모리에 저장된 명령어에 기초하여 동작을 수행할 수 있다.

[296] 또한 비일시적 컴퓨터가 읽을 수 있는 매체에 저장된 비디오 비트스트림을 포함하는 장치에 있어서, 인코더는 현재 블록이 MPM(Most Probable Modes)을 사용하고, 상기 현재 블록의 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보를 생성하고, planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 비트스트림을 생성할 수 있다. 비트스트림은 비일시적인 컴퓨터가 읽을 수 있는 매체에 저장될 수 있다.

[297] 도 21은 본 개시의 일 실시예에 따른 화면내 예측에 사용된 복수개의 참조 샘플 라인 (multi-reference line intra prediction (MRL))을 나타낸 도면이다.

[298] 인코더 또는 디코더는 현재 블록과 가장 가까운 참조샘플 라인을 기본적으로 사용하며, 추가적으로 단일 혹은 복수개의 참조 샘플 라인을 화면내 예측에 사용할 수 있다. 이때, 사용되는 참조샘플 라인수가 정의되며 모두 사용하는 경우는 추가 정보전달은 생략한다. 하지만, 일부를 사용하는 경우는 사용되는 참조 샘플 라인 정보 지시자를 인코더는 디코더에 알려 주어야 한다.

[299] 화면내(인트라) 모든 예측 모드에 사용될 수 있고 별도의 모드는 제외하고 사용될 수 있다. 인코더와 디코더의 복잡도/연산량 감소를 위해 특정 모드에서만 수행될 수 있다. 예를 들어 MPM(Most probable mode) 리스트에서만 복수개의 참조 샘플 라인을 사용하여 화면내 예측을 수행할 수 있다. 또한, DC, planar 모드를 제외한 각도 모드에만 수행될 수도 있다. 이 경우, 인코더는 예측에

사용되는 참조 샘플 라인을 지시하는 참조 샘플 라인 인덱스(MRL index)를 먼저 적용한 후에 MPM인덱스를 비트스트림에 포함시킬 수 있다. 디코더에서는 MRL 인덱스를 확인하면 MPM 사용여부를 나타내는 정보(MPM flag)가 true임을 알 수 있고 MPM 인덱스로 해당 모드를 알 수 있다.

- [300] 또한 인코더는 MPM 사용여부를 나타내는 정보 및 MPM 인덱스를 비트스트림에 포함할 수 있다. 디코더는 비트스트림으로부터 현재 블록의 MPM 사용여부를 나타내는 정보(MPM flag)를 획득하는 단계를 수행할 수 있다. 또한 디코더는 MPM 인덱스에 기초하여 MPM리스트 중에서 인트라 예측 모드를 결정할 수 있다.
- [301] 도 22는 본 개시의 일 실시예에 따른 현재 예측 블록에 이웃하는 블록에서 사용된 예측 모드를 고려하여 MPM 리스트를 만드는데 필요한 이웃 블록의 위치를 나타내는 도면이다.
- [302] 도 22에서처럼 이웃하는 블록들 중에서 왼쪽 블록(L)과 위 블록(A)만 고려하고 그 위치 또한 도면22에서처럼 특정 위치로 고정될 수 있다. L은 크기가 $W \times H$ 인 현재 블록의 좌 상단 좌표를 기준으로 $(-1, H-1)$, A는 $(W-1, -1)$ 의 위치가 될 수 있다.
- [303] 이웃한 블록에 포함된 좌측 블록과 상측 블록은 현재 블록 이전에 미리 복원된 블록일 수 있다. 좌측 블록 또는 상측 블록은 복원을 수행하는 과정에서 인트라 예측 모드에 기초하여 예측될 수 있다. 디코더는 L 과 A의 예측 모드를 고려하여 3가지 경우의 수로 구분하여 MPM리스트를 구성할 수 있다. 이때, MRL을 MPM리스트에서만 사용하게 되면 이를 고려한 MPM구성 방법이 필요하다. MRL을 사용하는 경우와 사용하지 않는 경우 MPM구성 방법이 달라져야 한다. MRL은 Planar와 DC모드를 제외하고 사용될 수 있으므로, MRL이 사용되었을 때 상기 두개의 모드가 MPM 리스트에 포함되지 않도록 MPM 리스트 구성방법이 달라져야 한다. MPM 리스트의 크기는 3 내지 6이 될 수 있고, 특정 MPM 크기에 한정되지 않으며, 본 발명에서는 MPM List를 6으로 가정하여 발명의 예를 설명한다.
- [304] 인코더 또는 디코더가 MRL 적용 여부에 따라 MPM 리스트 구성을 다르게 설정할 때, MRL 적용을 하지 않는 경우의 MPM 리스트를 MPM_LIST_NO_MRL이라고 하자. MRL을 적용하는 경우 MPM_LIST_MRL은 MPM_LIST_NO_MRL과 다음과 같은 방식을 적용하여 다르게 구성될 수 있다. 인코더 또는 디코더가 MPM_LIST_NO_MRL에서 Planar 또는 DC가 포함된 MPM list에서 해당 모드를 삭제하여, MPM list의 구성을 줄일 수 있다. 예를 들어, 인코더 또는 디코더가 planar 및 DC가 포함된 6 MPM list에서 planar 및 DC를 삭제하여 4개의 MPM list로 구성할 수 있다. 이와 같이 구성하는 경우, 인코더 또는 디코더는 MPM_LIST_NO_MRL의 MPM 리스트의 index를, 삭제된 planar 또는 DC를 고려하여, 이전 값으로 설정할 수 있다. MPM_LIST_NO_MRL을 구성하는 방법에서 PLANAR mode 와 DC mode가 MPM 리스트에 채워지게 되면

이들을 포함하지 않고 해당 인덱스에 다음 순서의 MPM을 채우도록 한다. 이는, 기존의 MPM 리스트를 만드는 방법에 Planar와 DC 모드가 모두 혹은 하나만 포함되는 방식이면 인코더 또는 디코더는 해당 인덱스에 Planar/DC를 포함하지 않고, 다음 순번의 MPM이 채워지게 되고 전체 MPM 개수에서 해당 두 모드에 대한 2개 혹은 1개가 빠진 형태로 전체 MPM 크기도 정해진다. 이는 정해진 MPM 총 개수에서 1개 혹은 2개가 빠지게 되어 MPM 인덱스를 시그널링하는 관점에서 정보량을 줄일 수 있다. 예를 들어, Truncated unary(TU)방식으로 이진화를 하는 경우 6개에서 4개로 크기가 줄어들면 1비트를 줄일 수 있다.

- [305] MPM_LIST_NO_MRL에서 Planar 또는 DC가 포함된 MPM 리스트에서 해당 모드를 각도 모드(angular mode)로 변경하여 추가할 수 있다. 즉, MPM리스트는 각도 모드만을 포함할 수 있다. Non-angular mode로서 planar 또는 DC 모드를 대체하여, 새로운 각도 모드를 추가하고, 해당 모드는 MPM_LIST_NO_MRL에서 정의된 각도 모드를 이용하여 추가적인 offset을 적용하여 구성할 수 있다. 해당 offset 값은 ((-/+) or (+/-)) 1,2,3,4 등 MPM 리스트의 각도 모드(angular mode) 값에 근사한 각도 모드(angular mode)로 설정할 수 있다. 또는, remaining mode로부터 채울 수 있다. 인코더 또는 디코더는 특정 위치를 고정하거나 인덱스가 작은/큰 순서로 1개 혹은 2개를 선택하여 채울 수 있다. Remaining mode가 추가 구분되어 selected mode 와 non-selected mode로 구분되는 경우 selected mode에서 특정 위치에서 1개 혹은 2개를 선택하여 채울 수 있다. 인덱스가 작은 순서 혹은 큰 순서로 정할 수 있다.
- [306] L, A와 같이 인접 블록의 각도 모드(angular mode)를 중심으로 MPM_LIST_NO_MRL 대비 Planar, DC 등 non-angular mode의 값을 L, A의 특정 위치를 우선하여 각도 모드(angular mode) 값을 활용하여 추가 구성할 수 있다. 예를 들어, L, A가 모두 각도 모드(angular mode)를 가지고 있고, L을 A보다 우선하여 적용하는 경우, Planar 또는 DC mode를 대체하여 새로운 각도 모드(angular mode)는 L에서 정의한 각도 모드(angular mode)의 인접 mode로서 offset을 적용하여 구성할 수 있다. 반대의 경우도 가능하다.
- [307] Angular mode 중 특정 angular mode 값을 우선하여, MPM_LIST_NO_MRL의 Planar/DC에 대해 특정 angular mode를 활용하여 대체할 수 있다. 예를 들어, VER_IDX 또는 HOR_IDX와 같이 특정 angular mode를 우선하여, 해당 mode에 offset(0을 포함)을 적용하여, Planar/DC를 대체할 수 있다. 미리 포함된 MPM list와 같은 경우 다른 offset값을 적용하여 그 값을 대신 채울 수 있다.
- [308] Current block의 shape에 따라서 planar/DC를 특정 angular mode 또는 L, A 중 특정 위치를 우선하여, 대체할 수 있다. 예를 들어 현재 블록이 non-square인 경우, 긴변에 위치한 L 또는 A의 각도 모드(angular mode)를 활용하여 대체하거나, VER/HOR의 angular mode를 활용하여 대체할 수 있다. 예를 들어, 가로가 큰 block 구성의 경우, A 또는 VER의 각도 모드(angular mode)를 우선하여, 해당 값에 offset (0을포함)을 적용하여 대체할 수 있다. 다른 예로, 현재 블록이

non-square인 경우, 짧은 변에 위치한 L 또는 A 의 각도 모드(angular mode)를 활용하여 대체할 수 있다.

[309] 위와 같은 MRL 적용 여부에 따른 MPM 리스트 구성 방법은 하나 또는 복수의 방식을 혼합하여 적용할 수 있으며, MPM 리스트의 구성은 6개와 같이 고정되어 사용되거나, 가변적으로 적용하는 것이 가능하다. 또한, MPM 리스트로 가정하여 설명하고 있으나, non-MPM list의 구성에도 같은 방식으로 확장하여 적용할 수 있다.

[310] 다음은 MRL을 고려한 MPM 리스트 구성 방법으로 L과 A에 사용된 예측 모드를 고려하여 3가지 경우로 구분하여 구성한다.

[311] 경우1. L과 A에서 각도 모드가 하나도 없을 때 혹은 이웃하는 L 과 A 가 둘다 존재하지 않거나 혹은 둘 중 하나는 존재하지 않고 나머지 하나가 각도모드가 아닌 경우, MPM 리스트를 구성하는 예;

[312] mpm[0] = VER_IDX ;

[313] mpm[1] = HOR_IDX ;

[314] mpm[2] = VER_IDX - 4;

[315] mpm[3] = VER_IDX + 4;

[316] mpm[4] = HOR_IDX-4;

[317] mpm[5] = HOR_IDX+4;

[318]

[319] 경우2 : L과 A에서 각도 모드가 하나 있는 경우, MPM 리스트를 구성하는 예
[320] 다음과 같은 순서 기반으로 생성하도록 한다.

[321] 순서: {left mode, above mode, left mode -1, left mode + 1, left mode -2, left mode +2, vertical mode, horizontal mode}

[322] 이때, L 과 A가 같은 각도 모드이면 상기 순서에서 상측 블록 모드(above mode)를 순서에서 제외하도록 한다. 이때 좌측 블록 모드(left mode)는 각도 모드(AngMode)일 수 있다.

[323] leftIntraDir = left mode

[324] aboveIntraDir = above mode

[325] AngMode is the available angular mode after checking left and above mode.

[326]

[327]

[328] offset = 62;

[329] mod = offset + 3;

[330] mpm[0] = AngMode;

[331] mpm[1] = ((AngMode + offset) % mod) + 2; // -1 mode of angular mode

[332] mpm[2] = ((AngMode - 1) % mod) + 2; // +1 mode of the angular mode;

[333]

[334] mpm[3] = ((AngMode + offset-1) % mod) + 2; // -2 mode of the angular mode

- [335] $\text{mpm}[4] = ((\text{AngMode}) \% \text{mod}) + 2; // +2 \text{ mode of the angular mode}$
- [336] $\text{mpm}[5] = (\text{AngMode} > 48 \ \&\& \ \text{AngMode} < 52) ? \text{HOR_IDX} : \text{VER_IDX};$
- [337] 경우3. L 과 A 가 서로 다른 각도 모드 일 수 있다.
- [338] 이때, 순서는 다음과 같을 수 있다.
- [339] 순서1 : {left mode, above mode, left mode -1, left mode + 1, above mode -1, above mode + 1, left mode -2, left mode + 2,}
- [340] 혹은 순서2: {left mode, above mode, left mode -1, left mode + 1, above mode -1, above mode + 1, left mode -2, above mode + 2}
- [341] 혹은 순서3: {left mode, above mode, left mode -1, left mode + 1, above mode -1, above mode + 1, Vertical mode, Horizontal mode}
- [342] 순서1을 기준으로 좌측 블록 모드(left mode)의 인덱스에 -1을 더한 값이 상측 블록의 인트라 예측 모드(above mode)와 같으면 상측 블록의 인트라 예측 모드(above mode)의 인덱스에 -1을 더한 값이 좌측 블록의 인트라 예측 모드(left mode)의 인덱스에 +1을 더한 값보다 우선할 수 있다. 즉, 현재 단계의 계산된 모드가 이전 단계에 이미 계산되어 MPM 리스트에 포함된 단계와 같으면 L인지 A의 모드인지를 확인하고, 예를들어 L모드이면 L모드에서의 순서상 다음차례의 값을 적용한다. 혹은 상기 순서 1과 같이 미리 정해진 순서대로 계산하여 채우되어 이미 계산된 MPM 리스트 중 하나와 같으면 다음 순서를 계산하여 채우도록 한다.
- [343] $\text{mpm}[0] = \text{leftIntraDir};$
- [344] $\text{mpm}[1] = \text{aboveIntraDir};$
- [345] $\text{mpm}[2] \sim \text{mpm}[5]$ 는 상기 MPM구성 순서1~3중에 한가지 방법으로 그 값을 결정하되 이미 채워진 MPM과 같을 경우 그 다음 계산 방법을 이용하여 서로 중복되지 않도록 구성한다.
- [346]
- [347] MRL 사용시 MPM 구성의 다른 예
- [348] MPM_set_0 는 상기 기술한 경우1 과 같은 순서로 MPM을 구성한다.
- [349] MPM_set_1 : 좌측 블록의 인트라 예측 모드(Left mode)와 상측 블록의 인트라 예측 모드(above mode)가 같고 각도 모드이거나 둘 중 하나만 각도 모드이면 다음과 같이 구성한다. 그 각도 모드를 AngMode라 한다.
- [350] $\text{mpm}[0] = \text{AngMode}$
- [351] $\text{mpm}[1] = 2 + ((\text{AngMode} + 62) \% 65) // A-1$
- [352] $\text{mpm}[2] = 2 + ((\text{AngMode} - 1) \% 65) // A+1$
- [353] $\text{mpm}[3] = 2 + ((\text{AngMode} + 61) \% 65) // A-2$
- [354] $\text{mpm}[4] = 2 + ((\text{AngMode}) \% 65) // A+2$
- [355] $\text{mpm}[5] = 2 + ((\text{AngMode} + 60) \% 65) // A-3$
- [356]
- [357] 상술한 바와 같이 디코더는 수신된 비트스트림으로부터 현재 블록의 MPM

사용여부를 나타내는 정보를 획득하는 단계를 수행할 수 있다. 디코더는 MPM 사용여부를 나타내는 정보에 기초하여 MPM 리스트를 생성할 것을 결정할 수 있다. MPM리스트는 MPM_set_1과 같이 생성될 수 있다. 위의 MPM_set_1은 offset이 62이고, mod는 65인 경우의 예시이다. 하지만 이에 한정되는 것은 아니다. offset은 61일 수 있다. 또한 mod는 64일 수 있다.

[358] 또한 디코더는 현재 블록의 좌측 블록의 인트라 예측 모드 또는 현재 블록의 상측 블록의 인트라 예측 모드 중 적어도 하나가 각도 모드인 경우, 각도 모드만을 이용하여 MPM리스트를 생성하는 단계를 수행할 수 있다. MPM리스트는 MPM_set_1과 같이 생성될 수 있다. AngMode는 좌측 블록의 인트라 예측 모드(Left mode) 또는 상측 블록의 인트라 예측 모드(above mode)일 수 있다. MPM_set_1은 좌측 블록의 인트라 예측 모드(Left mode) 또는 상측 블록의 인트라 예측 모드(above mode)가 동일한 경우이므로, 좌측 블록의 인트라 예측 모드(Left mode) 또는 상측 블록의 인트라 예측 모드(above mode) 중 하나를 사용하여 MPM리스트를 결정할 수 있다.

[359] 상술한 MPM_set_1과 같이 상측 블록의 인트라 예측 모드 또는 좌측 블록의 인트라 예측 모드 중 적어도 하나가 각도 모드이면 MPM 리스트는 각도모드만 포함할 수 있다. 즉 MPM 리스트는 DC 모드 및 planar모드를 포함하지 않을 수 있다. 디코더는 생성된 MPM리스트에 기초하여 인트라 예측 모드를 결정하는 단계를 수행할 수 있다. 또한 디코더는 결정된 인트라 예측 모드에 기초하여 현재 블록을 예측하는 단계를 수행할 수 있다.

[360] 본 개시의 일 실시예에 따라 MPM 리스트를 생성하는 과정은 이하와 같다. 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 동일하고, 좌측 블록의 인트라 예측 모드가 각도 모드인 경우, 디코더는 좌측 블록의 인트라 예측 모드(AngMode), 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스($2 + ((\text{AngMode} + \text{offset}) \% \text{mod}))$)에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스($2 + ((\text{AngMode} - 1) \% \text{mod}))$)에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스($2 + ((\text{AngMode} + \text{offset} - 1) \% \text{mod}))$)에 대응되는 각도 모드 및 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스($2 + (\text{AngMode} \% \text{mod}))$)에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 수행할 수 있다. 여기서 offset은 61이고 mod는 offset+3일 수 있다.

[361] 또한, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 동일하지 않고, 좌측 블록의 인트라 예측 모드 또는 상측 블록의 인트라 예측 모드 중 어느 하나만 각도 모드인 경우, 디코더는 각도 모드인 블록의 인트라 예측 모드(AngMode), 각도 모드인 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스($2 + ((\text{AngMode} + \text{offset}) \% \text{mod}))$)에 대응되는 각도 모드, 각도 모드인 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스($2 + ((\text{AngMode} - 1) \% \text{mod}))$)에 대응되는 각도 모드, 각도 모드인 블록의 인트라 예측 모드의

인덱스보다 2만큼 작은 인덱스($2 + ((\text{AngMode} + \text{offset} - 1) \% \text{mod})$)에 대응되는 각도 모드 및 각도 모드인 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스($2 + (\text{AngMode} \% \text{mod})$)에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 수행할 수 있다. 여기서 offset은 61이고 mod는 offset+3일 수 있다. 또한 디코더는 좌측 블록의 인트라 예측 모드 또는 상측 블록의 인트라 예측 모드 중 각도 모드인 예측 모드를 결정하기 위하여 다음과 같은 식을 이용할 수 있다.

- [362] 각도 모드인 블록의 인트라 예측 모드의 인덱스 = $\text{Max}(\text{좌측 블록의 인트라 예측 모드의 인덱스}, \text{상측 블록의 인트라 예측 모드의 인덱스})$
- [363] planar 모드의 인덱스는 1이고 DC 모드의 인덱스는 0이며, 각도모드는 1보다 큰 인덱스를 가지고, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드 중 하나만 각도모드라면, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 값을 각도 모드이기 때문이다.
- [364] MPM_set_2: 좌측 블록의 인트라 예측 모드(Left mode)와 상측 블록의 인트라 예측 모드(above mode)가 같지 않으며 둘다 1보다 크면 다음과 같이 구성할 수 있다. 여기서 인트라 예측 모드가 둘다 1보다 크다는 것은, 인트라 예측 모드가 DC 모드 planar 모드가 아니라는 것을 의미한다.
- [365] $\text{mpm}[0] = \text{leftIntraDir};$
- [366] $\text{mpm}[1] = \text{aboveIntraDir};$
- [367] leftIntraDir과 aboveIntraDir의 인덱스 크기를 비교하여, 그중 큰 값을 leftIntraDir로 가정하여 기술하면 다음과 같다.
- [368] $\text{mpm}[2] = (\text{leftIntraDir} + 62) \% 65 + 2; // \text{leftIntraDir} - 1;$
- [369] $\text{mpm}[3] = (\text{leftIntraDir} - 1) \% 65 + 2; // \text{leftIntraDir} + 1;$
- [370] $\text{mpm}[4] = (\text{leftIntraDir} + 61) \% 65 + 2; // \text{leftIntraDir} - 2;$
- [371] $\text{mpm}[5] = (\text{leftIntraDir}) \% 65 + 2; // \text{leftIntraDir} + 2;$
- [372] 상기 mpm순서를 채울 때 미리 채워진 것과 같으면 -1, +1 과 같은 offset값의 절대값이 1씩 혹은 2씩 혹은 기 설정된 크기만큼 증가될 수 있다.
- [373] leftIntraDir과 aboveIntraDir의 인덱스 크기를 비교하여, 그중 크거나 작은 값을 기준으로 위와 같이 offset값을 적용하여 mpm[2]~mpm[5]까지 구성할 수 있다.
- [374] 또한, 본 개시의 일 실시예에 따르면, 좌측 블록의 인트라 예측 모드(Left mode) 및 상측 블록의 인트라 예측 모드(above mode)가 동일하지 않고, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 각도 모드인 경우, 디코더는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 크기 비교에 기초하여 MPM리스트를 생성하는 단계를 수행할 수 있다. 예를 들어 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 값 또는 큰 값을 이용하여 MPM리스트를 생성할 수 있다.
- [375] 보다 구체적으로 좌측 블록의 인트라 예측 모드(Left mode) 및 상측 블록의

인트라 예측 모드(above mode)가 동일하지 않고, 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드가 각도 모드인 경우, 디코더는 좌측 블록의 인트라 예측 모드 및 상측 블록의 인트라 예측 모드를 MPM리스트에 포함하는 단계를 수행할 수 있다. 또한 디코더는 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 상측 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 상측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스에 대응되는 각도 모드, 상측 블록의 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드, 또는 상측 블록의 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드 중 적어도 하나의 인트라 예측 모드에 기초하여 MPM리스트를 생성하는 단계를 수행할 수 있다.

[376] 위와 같이 디코더는 좌측 블록의 인트라 예측 모드의 인덱스 또는 상측 블록의 인트라 예측 모드의 인덱스에서 1만큼 크거나 작은 값을 MPM 리스트에 포함하거나, 좌측 블록의 인트라 예측 모드의 인덱스 또는 상측 블록의 인트라 예측 모드의 인덱스에서 2만큼 크거나 작은 값을 MPM 리스트에 포함하는 경우, MPM리스트에 중복되는 각도모드가 포함될 가능성이 있다. 즉, 좌측 블록의 인트라 예측 모드의 인덱스 또는 상측 블록의 인트라 예측 모드의 인덱스의 차이가 1, 2이거나, 62보다 큰 경우, MPM 리스트에 중복되는 각도 모드가 포함될 가능성이 있다. 따라서 디코더는 아래 설명하는 바와 같은 방식으로 MPM 리스트를 생성할 수 있다.

[377] 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 1과 같은 경우, 디코더는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 2만큼 작은 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 수행할 수 있다.

[378] 또한, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 62보다 크거나 같은 경우, 디코더는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1만큼 작은 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의

인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 2 만큼 큰 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 수행할 수 있다.

[379] 또한 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 2와 같은 경우, 디코더는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1 만큼 큰 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1 만큼 작은 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1 만큼 큰 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 수행할 수 있다.

[380] 또한, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 1과 같지 않고, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 62보다 작고, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스의 차이가 2와 같지 않은 경우, 디코더는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1 만큼 작은 인덱스에 대응되는 각도 모드, 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 작은 인덱스보다 1 만큼 큰 인덱스에 대응되는 각도 모드, 또는 좌측 블록의 인트라 예측 모드의 인덱스 및 상측 블록의 인트라 예측 모드의 인덱스 중 큰 인덱스보다 1 만큼 작은 인덱스에 대응되는 각도 모드 중 적어도 하나에 기초하여 MPM리스트를 생성하는 단계를 수행하는 것을 특징으로 한다.

[381] MRL 사용시 MPM 구성의 다른 예2로 기존의 MPM List 구성 방법에서 Planar and/or DC를 제외하고 정하는 방법은 이하와 같다.

[382] Left와 Above가 모두 각도 모드가 아닌 경우,

[383] $mpm[0] = VER_IDX;$

[384] $mpm[1] = HOR_IDX;$

[385] $mpm[2] = VER_IDX - 4;$

[386] $mpm[3] = VER_IDX + 4;$

[387] Left와 Above에 적어도 하나의 각도 모드가 있는 경우 아래와 같이 구성할 수 있다. 이 경우 MPM리스트에 삽입되는 모드들의 순서는 이하와 같이 주어질 수 있다.

[388] {left mode, above mode, planar, DC, left mode -1, left mode + 1, vertical mode, horizontal mode}

[389] 위 순서에서 먼저 있을 수록 MPM리스트에 먼저 포함됨을 의미할 수 있다. 또한, left mode는 좌측 블록의 인트라 예측 모드, above mode는 상측 블록의

- 인트라 예측 모드를 나타낸다. 또한 *verticla mode*는 각도 모드의 하나로써, 수직방향 모드를 의미한다. 또한 *horizontal mode*는 수평 방향 모드를 의미한다.
- [390] 좌측 블록의 인트라 예측 모드와 상측 블록의 인트라 예측 모드가 동일한 경우 특별한 케이스가 발생할 수 있다. 이때, 상측 블록의 인트라 예측 모드는 MPM리스트에 포함되지 않을 수 있다.
- [391] 예를 들어 이하와 같이 가정하자.
- [392] `leftIntraDir = left mode`
- [393] `aboveIntraDir = above mode`
- [394] `AngMode`는 좌측 인트라 예측 모드 및 상측 인트라 예측 모드를 확인한 후 사용가능한 각도 모드일 수 있다.
- [395] `availNonAngMode`는 좌측 인트라 예측 모드 및 상측 인트라 예측 모드를 확인한 후 사용가능한 비-각도 모드일 수 있다. 만약 비-각도 모드가 사용가능하지 않다면 `availNonAngMode = -1`일 수 있다.
- [396] 위와 같은 가정에 기초하여 MPM리스트는 이하와 같이 생성될 수 있다.
- [397] `offset = 62;`
- [398] `mod = offset + 3;`
- [399] `mpm[0] = AngMode;`
- [400] `mpm[1] = ((AngMode + offset) % mod) + 2; // -1 mode of angular mode`
- [401] `mpm[2] = ((AngMode - 1) % mod) + 2; // +1 mode of the angular mode`
- [402] `mpm[3] = (AngMode > 48 && AngMode < 52) ? HOR_IDX: VER_IDX;`
- [403]
- [404] 좌측 블록의 인트라 예측 모드(*Left mode*)와 상측 블록의 인트라 예측 모드(*Above mode*)가 서로 다른 각도 모드인 경우 MPM리스트는 아래와 같이 구성될 수 있다.
- [405] 정의에 따라, 좌측 블록의 인트라 예측 모드(*Left mode*)와 상측 블록의 인트라 예측 모드(*Above mode*) 모두가 각도 모드이고 서로 다르다. MPM리스트에 삽입되는 모드들의 순서는 이하와 같이 주어질 수 있다.
- [406] {*left mode*, *above mode*, *planar*, *DC*, *left mode* -1, *left mode* + 1, *above mode* -1, *above mode* + 1}
- [407] 위 순서에서 먼저 있을 수록 MPM리스트에 먼저 포함됨을 의미할 수 있다.
- [408] 이때, 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응하는 각도 모드가 상측 블록의 인트라 예측 모드와 같은 경우가 발생할 수 있다. 이 경우 상측 블록의 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스에 대응하는 각도 모드가 좌측 블록의 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스에 대응하는 각도 모드보다 먼저 MPM리스트에 포함될 수 있다.
- [409] 예를 들어 이하와 같은 과정에 의하여 MPM리스트가 생성될 수 있다.
- [410] `leftIntraDir = left mode`
- [411] `aboveIntraDir = above mode`

```

[412]    mode = leftIntraDir;
[413]    angMinus = ((mode + offset) % mod) + 2; // angular -1 ;
[414]    angPlus = ((mode - 1) % mod) + 2; // angular +1 ;
[415]    mpm[0] = leftIntraDir;
[416]    mpm[1] = aboveIntraDir;
[417]    if (angMinus == aboveIntraDir)
[418]    {
[419]    mpm[2] = ((aboveIntraDir + offset) % mod) + 2; // above; angular -1 ;
[420]    mpm[3] = angPlus;
[421]    }
[422]    else
[423]    {
[424]    mpm[2] = angMinus;
[425]    if (aboveIntraDir == angPlus) mpm[3] = ((aboveIntraDir - 1) % mod) + 2;
[426]    else mpm[3] = angPlus;
[427]    }
[428]
[429]    MRL 사용시 MPM 구성의 다른 예3으로 기존의 MPM List 구성 방법에서 Planar
and/or DC를 제외하고 정하는 방법을 수도 코드로 나타내면 이하와 같다.
[430]    1. The candModeList[ x ] with x = 0..5 is derived as follows:
[431]    - CandModeList[0]=candIntraPredModeA CandModeList[1]!=candIntraPredModeA
[432]    CandModeList[2]=INTRA_ANGULAR50
[433]    CandModeList[3]=INTRA_ANGULAR18
[434]    CandModeList[4]=INTRA_ANGULAR46
[435]    CandModeList[5]=INTRA_ANGULAR54
[436]    - If candIntraPredModeB is equal to candIntraPredModeA, the following applies:
[437]    - If candIntraPredModeA is larger than 1, candModeList[ x ] with x = 0..5 is derived
as follows:
[438]    candModeList[ 0 ] = candIntraPredModeA
[439]    CandModeList[1]= 2 + ( ( candIntraPredModeA + 62 ) % 65 )
[440]    CandModeList[2]= 2 + ( ( candIntraPredModeA - 1 ) % 65 )
[441]    CandModeList[3]= 2 + ( ( candIntraPredModeA + 61 ) % 65 )
[442]
[443]    - Otherwise (candIntraPredModeB is not equal to candIntraPredModeA), the
following applies:
[444]    - Set the variable biggerIdx equals to biggerIdx= candModeList[ 0 ] >
candModeList[ 1 ] ? 0 : 1
[445]    - If both of candIntraPredModeA and candIntraPredModeB are larger than 1,

```

candModeList[x] with $x = 2..5$ is derived as follows:

[446] candModeList[0] = candIntraPredModeA

[447] candModeList[1] = candIntraPredModeB

[448] - If candModeList[biggerIdx]- candModeList[biggerIdx] equals to neither 64 nor 1, the following applies:

[449] candModeList[2]=2 + ((candModeList[biggerIdx] + 62) % 65)

[450] candModeList[3]=2 + ((candModeList[biggerIdx] - 1) % 65)

[451] - Otherwise, the following applies:

[452] candModeList[2]=2 + ((candModeList[biggerIdx] + 61) % 65)

[453] candModeList[3]=2 + (candModeList[biggerIdx] % 65)

[454] 도 23은 본 개시의 일 실시예에 따라 Most Probable Mode(MPM)에 +1, -1 등의 offset들을 적용하여 새로운 MPM을 유도하는 방법을 나타낸 도면이다.

[455] MPM을 구성하는 방법 중 주변블록에서 사용했던 각도 모드 기반으로 offset1을 적용하여 새로운 MPM를 만들 수 있다. 여기서 offset1은 -1, +1, -2, +2..등으로 크기와 순서는 방법에 따라 다를 수 있다.

[456] 예를 들면 다음과 같은 방법으로 구성할 수 있다.

[457] 예1:

[458] Offset2=61, Mod=offset2+3

[459] AngMod: 각도 모드

[460] 예1-1:((Angmod+offset)%Mod)+2 : 각도 모드에 -1 적용한 것과 같음

[461] 예1-2: ((Angmod-1)%Mod)+2 : 각도 모드에 +1 적용한 것과 같음

[462] 예1-3: ((Angmod+offset -1)%Mod)+2 : 각도 모드에 -2 적용한 것과 같음

[463] 예1-4: (Angmod%Mod)+2 : 각도 모드에 +2 적용한 것과 같음

[464] 예1-5: ((Angmod+offset -2)%Mod)+2 : 각도 모드에 -3 적용한 것과 같음

[465]

[466] 위의 식을 다시 정리하면 이하와 같다.

[467] 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스는 하기와 같은 식에 의하여 획득될 수 있다.

[468] 인트라 예측 모드의 인덱스보다 1만큼 작은 인덱스 = ((인트라 예측 모드의 인덱스+offset)%(offset+3))+2

[469] 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스는 하기와 같은 식에 의하여 획득될 수 있다.

[470] 인트라 예측 모드의 인덱스보다 1만큼 큰 인덱스 = ((인트라 예측 모드의 인덱스-1)%(offset+3))+2

[471] 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스는 하기와 같은 식에 의하여 획득될 수 있다.

[472] 인트라 예측 모드의 인덱스보다 2만큼 작은 인덱스 = ((인트라 예측 모드의 인덱스+offset-1)%(offset+3))+2

[473] 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스는 하기와 같은 식에 의하여 획득될 수 있다.

[474] 인트라 예측 모드의 인덱스보다 2만큼 큰 인덱스 = ((인트라 예측 모드의 인덱스)%(offset+3))+2

[475] 여기서 offset은 61일 수 있다.

[476]

[477] 각도 모드가 65개인 경우 각도 모드 인덱스는 2 부터 66까지 사용하여 시그널링 가능한 경우, 위 예1에서 각도모드 66 에 1을 증가하게 되면 각도 모드 3으로 유도된다. 도23의 a에서 확인할 수 있다. 예1에서 각도모드가 2인 경우 -1을 적용하게 되면 각도모드 65가 되며 도23의 b에서 확인할 수 있다. 현재 각도 모드에서 특정 수를 더하여(각도모드+offset) 66보다 커지게 되면 예1에서 의도한 대로 동작하지 않고 예1의 일부 경우는 결과값에 +1을 한 형태가 될 수 있다. 또한, 현재 각도 모드에 특정 수를 빼서(각도모드2 -offset) 2보다 작아지게 되면 예1에서 의도한 대로 동작하지 않고 일부 경우는 예1의 결과 값에 -1을 한 형태가 될 수 있다.

[478] 예2:

[479] Offset3=62, Mod=offset3+3

[480] AngMod : 각도 모드

[481] AngMod: 각도 모드

[482] 예2-1:((Angmod+offset)%Mod)+2 : 각도 모드에 -1 적용한 것과 같음

[483] 예2-2: ((Angmod-1)%Mod)+2 : 각도 모드에 +1 적용한 것과 같음

[484] 예2-3: ((Angmod+offset -1)%Mod)+2 : 각도 모드에 -2 적용한 것과 같음

[485] 예2-4: (Angmod%Mod)+2 : 각도 모드에 +2 적용한 것과 같음

[486] 예2-5: ((Angmod+offset -2)%Mod)+2 : 각도 모드에 -3 적용한 것과 같음

[487] 상기 예2는 예1에서 offset2의 값 61대신 offset3값 62를 사용한 예이다.

이경우에, 각도모드+ offset값이 66보다 크면 예1의 경우와 달리 예2에서 의도한 대로 동작 하는 것을 도 23의 c에서 확인할 수 있다. 또한, 각도모드-offset값이 2보다 작은 경우에도 예2에서 의도한 대로 동작하는 것을 도7의 d에서 확인할 수 있다.

[488]

[489] 도 24는 본 개시의 일 실시예에 따라 Most Probable Mode(MPM)에 +1, -1등의 offset들을 적용하여 새로운 MPM을 유도하는 방법을 non-square블록에 적용한 예를 나타낸 도면이다.

[490] 정사각형 블록이 아닌 경우 추가된 wide angle mode를 사용할 수 있다. 기존 각도 모드 2~66번보다 큰 절대 각도를 사용할 수 있다. 이 블록에 MPM구성 방법이 상기 예1과 같으면 다음과 같이 구성될 수 있다. 주변 블록의 각도 모드가 1개인 경우 그 모드가 2인 경우다.

[491] Offset2=61, Mod=offset2+3

- [492] $\text{AngMod} = 2$,
- [493] 예 1-1: $((\text{Angmod} + \text{offset}) \% \text{Mod}) + 2 = 65$, 각도 모드에 -1 적용한 것과 같음
- [494] 예 1-2: $((\text{Angmod} - 1) \% \text{Mod}) + 2 = 3$, 각도 모드에 +1 적용한 것과 같음
- [495] 예 1-3: $((\text{Angmod} + \text{offset} - 1) \% \text{Mod}) + 2 = 64$, 각도 모드에 -2 적용한 것과 같음
- [496] 예 1-4: $(\text{Angmod} \% \text{Mod}) + 2 = 4$, 각도 모드에 +2 적용한 것과 같음
- [497] 예 1-5: $((\text{Angmod} + \text{offset} - 2) \% \text{Mod}) + 2 = 63$, 각도 모드에 -3 적용한 것과 같음
- [498] 각도모드 2로부터 순서대로 유도된 각도모드는 65, 3, 64, 4, 63이 되며 6MPM은 아래와 같은 각도 모드들 중 적어도 하나를 포함할 수 있다.
- [499] $6 \text{ MPM} = [2, 65, 3, 64, 4, 63]$
- [500]
- [501] 도 8의 a에서 확인할 수 있으며, 이 경우 모드2는 모드 67, 모드 3은 68, 모드 4는 69로 확장된 각도 모드를 지시한다. 따라서, 직사각형 블록에서 상기 6MPM은 [67, 65, 68, 64, 69, 63]과 같게 된다. 이는 확장된 각도 모드를 사용하는 경우, 예1의 의도대로 기준 각도모드 67로부터 유도되는 -1, +1, -2, +2, +3이 적용되는 대신 -2, +1, -3, +2, -4가 적용되게 된다. 이는 기준 각도 모드의 주변 각도를 최대한 사용하려는 의도에서 벗어나게 된다.
- [502] 또다른 예로, 각도 모드가 66인 경우 예1을 예로 들면 다음과 같다.
- [503] $\text{Offset2} = 61, \text{Mod} = \text{offset2} + 3$
- [504] $\text{AngMod} = 66$,
- [505] 예 1-1: $((\text{Angmod} + \text{offset}) \% \text{Mod}) + 2 = 65$, 각도 모드에 -1 적용한 것과 같음
- [506] 예 1-2: $((\text{Angmod} - 1) \% \text{Mod}) + 2 = 3$, 각도 모드에 +2 적용한 것과 같음
- [507] 예 1-3: $((\text{Angmod} + \text{offset} - 1) \% \text{Mod}) + 2 = 64$, 각도 모드에 -2 적용한 것과 같음
- [508] 예 1-4: $(\text{Angmod} \% \text{Mod}) + 2 = 4$, 각도 모드에 +3 적용한 것과 같음
- [509] 예 1-5: $((\text{Angmod} + \text{offset} - 2) \% \text{Mod}) + 2 = 63$, 각도 모드에 -3 적용한 것과 같음
- [510] 각도모드 66으로부터 순서대로 유도된 각도모드는 65, 3, 64, 4, 63이 되며 6MPM은 아래의 각도 모드들 중 적어도 하나를 포함할 수 있다.
- [511] $6\text{MPM} = [66, 65, 3, 64, 4, 63]$
- [512] 도 8의 b처럼 세로로 직사각형 블록의 경우 wide angle을 사용하게 되면 모드 66은 모드 -1을, 65는 -2, 64는 -3, 63은 -4를 지시한다. 따라서 해당 블록에서의 6MPM은 아래의 각도 모드들 중 적어도 하나를 포함할 수 있다.
- [513] $6\text{MPM} = [-1, -2, 3, -3, 4, -4]$
- [514] 해당 블록의 wide angle 모드 -1에 -1, +1, -2, +2, +3 값을 적용하는 대신, -1, 3, -2, +3, -3을 적용하게 된다. 이는 각도 모드 -1의 주변 각도 모드를 최대한 사용하려는 의도에서 벗어나게 된다.
- [515] 이를 위해, 정사각형 블록 유무에 따라 상기 예1의 방법 혹은 예2의 방법을 사용하여 MPM을 구성하도록 한다. 정사각형 블록에서는 예1의 방법을 적용하고 정사각형 블록이 아닌 경우 예2의 방법으로 사용하도록 한다.
- [516] 도 25는 본 개시의 일 실시예에 따라 MRL를 적용한 직사각형 블록에 대한 참조

샘플을 설명하기 위한 도면이다.

- [517] 이미 설명한 바와 같이 디코더는 수신된 비트스트림에 기초하여 현재 블록의 인트라 예측 모드의 인덱스를 획득하는 단계를 수행할 수 있다. 보다 구체적으로 디코더는 비트스트림으로부터 선택스 엘리먼트를 획득하고 선택스 엘리먼트에 기초하여 현재 블록의 인트라 예측 모드의 인덱스를 획득할 수 있다.
- [518] 또한 디코더는 수신된 비트스트림으로부터 상기 현재 블록의 참조 샘플 라인 인덱스를 획득하는 단계를 수행할 수 있다. 디코더는 비트스트림을 파싱하여 참조 샘플 라인 인덱스를 획득할 수 있다.
- [519] 또한, 디코더는 인트라 예측 모드의 인덱스 또는 상기 참조 샘플 라인 인덱스 중 적어도 하나에 기초하여 복수의 참조 샘플들 및 추가 참조 샘플을 생성하는 단계를 수행할 수 있다.
- [520] 도 25는 W:H=2:1인 직사각형 경우를 나타낸다. 상기 도 21에서 현재 블록의 너비 및 폭의 길이가 동일한 것에 비해 도 25는 현재 블록의 너비 및 높이의 길이가 서로 다를 수 있다.
- [521] 도 25에서 Segment B, Segment C, Segment D 또는 Segment E는 복수의 참조 샘플들을 의미할 수 있다. 복수의 참조 샘플들의 개수는 참조 샘플 라인 인덱스와 관련 없이 결정될 수 있다. 예를 들어 복수의 참조 샘플들의 개수는, 현재 블록의 너비, 높이에 기초하여 결정될 수 있다.
- [522] 도 25에서 Segment A 또는 Segment F는 추가 참조 샘플을 의미할 수 있다. 복수의 참조 샘플들의 개수는 인트라 예측 모드의 인덱스 또는 참조 샘플 라인 인덱스 중 적어도 하나에 기초하여 결정될 수 있다. 예를 들어 복수의 참조 샘플들의 개수는, 현재 블록의 너비, 높이에 기초하여 결정될 수 있다. 이에 대하여 아래에서 보다 자세히 설명한다.
- [523] 도 25의 Segment A 및 Segment F는 도 21의 Segment A 및 Segment F에 비하여 샘플 수에서 차이가 발생할 수 있다. W:H의 비율의 차이가 클수록 차이가 커질 수 있다. 수평 또는 수직 방향 직사각형 예측 블록에 45도 각도보다 큰 각도 모드가 적용되는 경우 정사각형 예측 블록과 달리 추가 참조 샘플이 발생할 수 있다.
- [524] 인코더 또는 디코더는 참조 샘플 라인 인덱스 또는 상기 현재 블록의 너비 및 높이의 비율에 기초하여 추가 참조 샘플의 개수를 결정하는 단계를 수행할 수 있다.
- [525] 예를 들어, Segment A는 참조 샘플 라인 인덱스(MRL index) 만큼 추가 참조 샘플이 생성될 수 있다. 보다 구체적으로 Segment A의 추가 참조 샘플의 개수는 $\text{MRL index} + \text{offset}$ 로 정의될 수 있다. Offset은 미리 정의한 샘플 수를 사용할 수 있다. 도 25에는 MRL_index가 2개의 참조 샘플이 3인 경우, $3 + \text{offset}$ 만큼 추가 참조 샘플이 생성될 수 있다.
- [526] Segment F에는 $(\text{MRL_index} \times \text{whRatio})$ 만큼의 추가 참조 샘플이 생성될 수 있다. 여기서 MRL_index는 참조 샘플 라인 인덱스를 나타내고 whRatio는 현재

블록의 너비 및 높이의 비율을 나타낼 수 있다. 인코더 또는 디코더는 추가 참조 샘플의 개수를 참조 샘플 라인 인덱스 x 현재 블록의 너비 및 높이의 비율 + offset와 같이 결정할 수 있다. 또한 인코더 또는 디코더는 Segment F에 $(MRL_index \times whRatio + (MRL_index \text{ or } Offset))$ 만큼 추가 참조 샘플을 생성할 수 있다. 여기서 offset은 2 또는 3일 수 있다. 즉, $((2 \text{ or } 3) + MRL_index \times whRatio)$ 만큼 추가 참조 샘플이 생성될 수 있다. 변수 whRatio는 $\text{Min}(\text{Abs}(\text{Log2}(nTbW / nTbH)), 2)$ 일 수 있다. 하지만 이에 한정되는 것은 아니고 whRatio는 $\text{max}(nTbW / nTbH, 1)$ 일 수 있다.

- [527] 도 25는 수평 직사각형 예측 블록 형태여서 Segment A보다 Segment F가 더 많은 샘플수가 필요할 수 있다. 이와 다르게 수직방향 직사각형 예측 블록 형태는 Segment F보다 Segment A보다 더 많은 샘플수가 필요할 수 있다.
- [528] 도 26은 본 개시의 일 실시예에 따라 MRL를 적용하는 직사각형 블록에 대한 참조 샘플을 설명하기 위한 도면이다.
- [529] 도 26은 W:H=1:2인 직사각형 인 경우를 나타낸다. 도 25와 달리 도 26은 수직방향 직사각형 예측 블록을 나타낸다. 디코더 또는 인코더는 도 26의 Segment A에 도 25의 Segment F에 적용했던 방법들을 적용할 수 있다.
- [530] 도 27은 본 개시의 일 실시예에 따라 MRL를 적용하는 직사각형 블록에 대한 참조 샘플을 설명하기 위한 도면이다.
- [531] 도 27은 W:H=1:4 직사각형인 경우를 나타낸다. 디코더 또는 인코더는 도 27의 1:4의 블록의 Segment A에 도 25의 Segment F에 적용된 방법을 적용할 수 있다. 또한 디코더 또는 인코더는 도 27의 Segment F에 도 25의 Segment A에 적용된 방법을 적용할 수 있다. 수직/수평 방향 직사각형 예측 블록의 너비 대 높이 비율을 한정하지 않고 모든 비율에 도 25 내지 도 27에서 설명된 방법에 기초하여 추가 참조 샘플을 생성할 수 있다.
- [532] 이하에서는 복수의 참조 샘플들 및 추가 참조 샘플을 생성하는 단계에 대하여 보다 자세히 설명한다.
- [533] 인코더 또는 디코더는 인트라 예측 모드의 인덱스가 34보다 크거나 같고, 인트라 예측 모드의 인덱스에 매핑되는 인트라 예측 모드의 각도가 0보다 크거나 같은 경우, 현재 블록에 인접한 미리 복원된 상측 블록에 기초하여 복수의 제 1 참조 샘플들 및 제 1 추가 참조 샘플을 생성하는 단계를 수행할 수 있다.
- [534] 인트라 예측 모드의 인덱스는 인트라 예측 모드에 대응되는 인덱스를 의미한다. 이미 설명한 바와 같이 인트라 예측 모드는 67개가 있을 수 있다. 또한 인트라 예측 모드는 현재 블록의 크기에 기초하여 확장될 수 있다.
- [535] 또한 인트라 예측 모드의 각도는 인트라 예측 모드의 인덱스에 매핑되는 구성이다. 인코더 또는 디코더는 소정의 테이블에 기초하여 인트라 예측 모드의 인덱스를 인트라 예측 모드의 각도에 매핑시킬 수 있다. 인트라 예측 모드의 인덱스는 인트라 예측 모드에 일대일 대응될 수 있다. 하지만 인트라 예측

모드의 각도는 인트라 예측 모드와 일대다로 대응될 수 있다. 즉, 인트라 예측 모드의 각도는 서로 다른 인트라 예측 모드를 나타낼 수 있다. 예를 들어 인트라 예측 모드의 인덱스가 18인 경우 및 인트라 예측 모드의 인덱스가 50인 경우 인트라 예측 모드의 각도는 0일 수 있다.

- [536] 복수의 제 1 참조 샘플들은 참조 샘플들에 포함될 수 있다. 제 1 추가 참조 샘플은 추가 샘플에 포함될 수 있다. 복수의 제 1 참조 샘플들 및 제 1 추가 참조 샘플은 현재 블록의 좌상측, 상측 또는 우상측 중 적어도 하나에 위치한 미리 복원된 블록에 기초하여 획득될 수 있다.
- [537] 인트라 예측 모드의 인덱스가 34보다 작고, 인트라 예측 모드의 각도가 0보다 크거나 같은 경우, 인코더 또는 디코더는 현재 블록에 인접한 미리 복원된 좌측 블록에 기초하여 복수의 제 2 참조 샘플들 및 제 2 추가 참조 샘플을 생성하는 단계를 수행할 수 있다.
- [538] 복수의 제 2 참조 샘플들은 복수의 참조 샘플들에 포함될 수 있다. 또한 제 2 추가 참조 샘플은 추가 참조 샘플에 포함될 수 있다. 복수의 제 2 참조 샘플들 및 제 2 추가 참조 샘플은 현재 블록의 좌하측, 좌측 또는 좌상측 중 적어도 하나에 위치한 미리 복원된 블록에 기초하여 획득될 수 있다.
- [539] 보다 구체적으로 인코더 또는 디코더는 미리 복원된 상측 블록에 포함된 샘플들 중 복수의 제 1 참조 샘플들의 위치에 대응되는 샘플들에 기초하여 복수의 제 1 참조 샘플들을 획득하는 단계를 수행할 수 있다. 제 1 참조 샘플들의 위치에 대응되는 샘플들은 현재 블록의 좌상측, 상측 또는 우상측 중 적어도 하나의 미리 복원된 블록에 포함되는 샘플들일 수 있다.
- [540] 현재 블록의 좌상측 픽셀의 위치를 (0,0)으로 했을 때, 제 1 참조 샘플들의 위치에 대응되는 미리복원된 샘플들의 위치는 $(-1 - \text{refIdx} + x, -1 - \text{refIdx})$ 을 포함할 수 있다. 여기서 x 는 0 내지 $n\text{TbW} + \text{refIdx} + 1$ 일 수 있다. refIdx 는 참조 샘플 라인 인덱스일 수 있다. $n\text{TbW}$ 는 현재 블록의 너비를 의미할 수 있다.
- [541] 또한 제 1 참조 샘플들의 위치에 대응되는 미리 복원된 샘플들의 위치는 $(-1 - \text{refIdx} + x, -1 - \text{refIdx})$ 을 더 포함할 수 있다. 여기서 x 는 $n\text{TbW} + 2 + \text{refIdx}$ 내지 $\text{refW} + \text{refIdx}$ 일 수 있다. refIdx 는 참조 샘플 라인 인덱스일 수 있다. $n\text{TbW}$ 는 현재 블록의 너비를 의미할 수 있다. refW 는 제 1 참조 샘플의 너비를 의미할 수 있다.
- [542] 또한, 인코더 또는 디코더는 복수의 제 1 참조 샘플들 중 최우측 위치의 샘플을 패딩하여 제 1 추가 참조 샘플을 획득하는 단계를 수행할 수 있다. 제 1 참조 샘플들 중 최우측 위치 샘플의 위치는 $(-1 + \text{refW}, -1 - \text{refIdx})$ 을 포함할 수 있다. 제 1 추가 참조 샘플이 복수인 경우 모두 동일한 값을 가질 수 있다. 인코더 및 디코더는 참조 샘플 라인 인덱스, 현재 블록의 너비 및 높이의 비율 및 offset에 기초하여 제 1 추가 참조 샘플의 개수를 결정할 수 있다. 이에 대해서는 이미 상술한 바 있으므로 중복되는 설명은 생략한다.
- [543] 인코더 또는 디코더는 미리 복원된 좌측 블록에 포함된 샘플들 중 복수의 제 2

참조 샘플들의 위치에 대응되는 샘플들에 기초하여 복수의 제 2 참조 샘플들을 획득하는 단계를 수행할 수 있다. 제 2 참조 샘플들의 위치에 대응되는 샘플들은 현재 블록의 좌하측, 좌측 또는 좌상측 중 적어도 하나의 미리 복원된 블록에 포함되는 샘플들일 수 있다.

- [544] 현재 블록의 좌상측 픽셀의 위치를 (0,0)으로 했을 때, 제 2 참조 샘플들의 위치에 대응되는 미리복원된 샘플들의 위치는 $(-1 - \text{refIdx}, -1 - \text{refIdx} + x)$ 를 포함할 수 있다. 여기서 x 는 0 내지 $n\text{TbH} + \text{refIdx} + 1$ 을 포함할 수 있다. 일 수 있다. refIdx 는 참조 샘플 라인 인덱스일 수 있다. $n\text{TbH}$ 는 현재 블록의 높이를 의미할 수 있다.
- [545] 또한 제 2 참조 샘플들의 위치에 대응되는 미리 복원된 샘플들의 위치는 $(-1 - \text{refIdx}, -1 - \text{refIdx} + x)$ 를 더 포함할 수 있다. 여기서 x 는 $n\text{TbH} + 2 + \text{refIdx}$ 내지 $\text{refH} + \text{refIdx}$ 일 수 있다. refIdx 는 참조 샘플 라인 인덱스일 수 있다. $n\text{TbH}$ 는 현재 블록의 높이를 의미할 수 있다. refH 는 제 2 참조 샘플의 높이를 의미할 수 있다.
- [546] 인코더 및 디코더는 복수의 제 2 참조 샘플들 중 최하측 위치의 샘플을 패딩하여 제 2 추가 참조 샘플을 획득하는 단계를 수행할 수 있다. 제 2 참조 샘플들 중 최하측 위치 샘플의 위치는 $(-1 - \text{refIdx}, -1 + \text{refH})$ 을 포함할 수 있다. 제 2 추가 참조 샘플이 복수인 경우 모두 동일한 값을 가질 수 있다. 인코더 및 디코더는 참조 샘플 라인 인덱스, 현재 블록의 너비 및 높이의 비율 및 offset에 기초하여 제 2 추가 참조 샘플의 개수를 결정할 수 있다. 이에 대해서는 이미 상술한 바 있으므로 중복되는 설명은 생략한다.
- [547]
- [548] 다음은 상기 도 25~27에서 제안한 방법을 적용하는 경우에 대하여 아래에서 보다 자세히 설명한다.
- [549] 인코더 또는 디코더는 예측 샘플들의 값인 $\text{predSamples}[x][y]$ (with $x = 0..n\text{TbW} - 1, y = 0..n\text{TbH} - 1$)을 아래와 같이 생성할 수 있다.
- [550] 만약 인트라 예측 모드의 인덱스가 34보다 크거나 같으면, 인코더 또는 디코더는 아래의 과정을 수행한다.
- [551] 1. The reference sample array $\text{ref}[x]$ is specified as follows:
- [552] - The following applies:
- [553] $\text{ref}[x] = p[-1 - \text{refIdx} + x][-1 - \text{refIdx}]$, with $x = 0..n\text{TbW} + \text{refIdx}$
- [554] - If intraPredAngle is less than 0, the main reference sample array is extended as follows:
- [555] - When $(n\text{TbH} * \text{intraPredAngle}) \gg 5$ is less than -1,
- [556] $\text{ref}[x] = p[-1 - \text{refIdx}][-1 - \text{refIdx} + ((x * \text{invAngle} + 128) \gg 8)]$,
- [557] with $x = -1..(n\text{TbH} * \text{intraPredAngle}) \gg 5$ (8 35)
- [558] - Otherwise,
- [559] $\text{ref}[x] = p[-1 - \text{refIdx} + x][-1 - \text{refIdx}]$, with $x = n\text{TbW} + \text{refIdx} + 1..refW$
- [560] - refIdx 가 0보다 큰 때, 추가 참조 샘플들 $\text{ref}[\text{refW} + x]$ with $x = 1..(3 * \text{refIdx})$

)은 이하와 같이 패딩될 수 있다.

[561] 직사각형 블록을 고려하는 경우 패딩되는 샘플 개수는 $(3 * \text{refIdx})$ 대신 $W/H > 1$ 을 만족하면 상기 도 25의 Segment F에 적용한 방법들 중 하나를 적용하고, $W/H < 1$ 을 만족하면 도 25의 Segment A에 적용한 방법들 중 하나를 적용할 수 있다. 인코더 또는 디코더는 아래와 같이 패딩을 수행할 수 있다.

[562] $\text{ref}[\text{refW} + x] = p[-1 - \text{refIdx} + \text{refW}][-1 - \text{refIdx}]$

[563]

[564] 2. The values of the prediction samples $\text{predSamples}[x][y]$, with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ are derived as follows:

[565] - The index variable $iIdx$ and the multiplication factor $iFact$ are derived as follows:

[566] $iIdx = ((y + 1 + \text{refIdx}) * \text{intraPredAngle}) \gg 5$

[567] $iFact = ((y + 1 + \text{refIdx}) * \text{intraPredAngle}) \& 31$

[568] - Depending on the value of $iFact$, the following applies:

[569] - If $iFact$ is not equal to 0, the value of the prediction samples $\text{predSamples}[x][y]$ is derived as follows:

[570] $\text{predSamples}[x][y] =$

[571] $((32 - iFact) * \text{ref}[x + iIdx + 1] + iFact * \text{ref}[x + iIdx + 2] + 16) \gg 5$

[572] - Otherwise, the value of the prediction samples $\text{predSamples}[x][y]$ is derived as follows:

[573] $\text{predSamples}[x][y] = \text{ref}[x + iIdx + 1]$

[574]

[575] 하지만 위와 같은 방식에 한정되는 것은 아니다. 만약 인트라 예측 모드의 인덱스가 34보다 크거나 같으면, 인코더 또는 디코더는 아래의 과정을 수행한다.

[576] 1-1. The reference sample array $\text{ref}[x]$ is specified as follows:

[577] - The following applies:

[578] $\text{ref}[x] = p[-1 - \text{refIdx} + x][-1 - \text{refIdx}]$, with $x = 0..nTbW + \text{refIdx} + 1$

[579] - If intraPredAngle is less than 0, the main reference sample array is extended as follows:

[580] $\text{ref}[x] = p[-1 - \text{refIdx}][-1 - \text{refIdx} + \text{Min}((x * \text{invAngle} + 256) \gg 9, nTbH)]$, with $x = -nTbH..1$

[581] - Otherwise,

[582] $\text{ref}[x] = p[-1 - \text{refIdx} + x][-1 - \text{refIdx}]$, with $x = nTbW + 2 + \text{refIdx}.. \text{refW} + \text{refIdx}$

[583] - refIdx 가 0보다 큰 때, 추가 참조 샘플들 $\text{ref}[\text{refW} + \text{refIdx} + x]$ with $x = 1..(\text{Max}(1, nTbW / nTbH) * \text{refIdx} + 1)$ 은 이하와 같이 패딩될 수 있다.

[584] 직사각형 블록을 고려하는 경우 패딩되는 샘플 개수는 $(\text{Max}(1, nTbW / nTbH) * \text{refIdx} + 1)$ 일 수 있다. 인코더 또는 디코더는 아래와 같이 패딩을 수행할 수 있다.

[585] $\text{ref}[\text{refW} + \text{refIdx} + x] = p[-1 + \text{refW}][-1 - \text{refIdx}]$

[586]

[587] 2-1. The values of the prediction samples $\text{predSamples}[x][y]$, with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ are derived as follows:

[588] - The index variable $iIdx$ and the multiplication factor $iFact$ are derived as follows:

[589] $iIdx = (((y + 1 + refIdx) * intraPredAngle) >> 5) + refIdx$

[590] $iFact = (((y + 1 + refIdx) * intraPredAngle) \& 31)$

[591] If $cIdx$ is equal to 0, the following applies:

[592] The interpolation filter coefficients $fT[j]$ with $j = 0..3$ are derived as follows:

[593] $fT[j] = \text{filterFlag} ? fG[iFact][j] : fC[iFact][j]$

[594] The value of the prediction samples $\text{predSamples}[x][y]$ is derived as follows:

[595] $\text{predSamples}[x][y] = \text{Clip1Y}(((\sum_{i=0}^3 fT[i] * ref[x + iIdx + i])^{+32}) >> 6)$

[596] Otherwise ($cIdx$ is not equal to 0), depending on the value of $iFact$, the following applies:

[597] If $iFact$ is not equal to 0, the value of the prediction samples $\text{predSamples}[x][y]$ is derived as follows:

[598] $\text{predSamples}[x][y] =$

[599] $((32 - iFact) * ref[x + iIdx + 1] + iFact * ref[x + iIdx + 2] + 16) >> 5$

[600] Otherwise, the value of the prediction samples $\text{predSamples}[x][y]$ is derived as follows:

[601] $\text{predSamples}[x][y] = ref[x + iIdx + 1]$

[602]

[603] 위와는 다르게 만약 인트라 예측 모드의 인덱스가 34보다 작으면, 인코더 또는 디코더는 아래의 과정을 수행한다.

[604] 1. The reference sample array $ref[x]$ is specified as follows:

[605] - The following applies:

[606] $ref[x] = p[-1 - refIdx][-1 - refIdx + x]$, with $x = 0..nTbH + refIdx$

[607] - If $intraPredAngle$ is less than 0, the main reference sample array is extended as follows:

[608] - When $(nTbW * intraPredAngle) >> 5$ is less than -1,

[609] $ref[x] = p[-1 - refIdx + ((x * invAngle + 128) >> 8)][-1 - refIdx]$,

[610] with $x = -1..(nTbW * intraPredAngle) >> 5$

[611] - Otherwise,

[612] $ref[x] = p[-1 - refIdx][-1 - refIdx + x]$, with $x = nTbH + refIdx + 1..refH$

[613] - $refIdx$ 가 0보다 큰 때, 추가 참조 샘플들 $ref[refH + x]$ with $x = 1..(3 * refIdx)$ 은 아래와 같이 패딩될 수 있다.

[614] 직사각형 블록을 고려하는 경우 패딩되는 샘플 수는 $(3 * refIdx)$ 대신 $W/H > 1$

을 만족하면 상기 도 25의 Segment A에 적용한 방법들 중 하나를 적용하고, $W/H < 1$ 을 만족하면 도 25의 Segment F에 적용한 방법들 중 하나를 적용한다. 인코더 또는 디코더는 이하와 같이 패딩을 수행할 수 있다.

[615] $\text{ref}[\text{refH} + x] = p[-1 - \text{refIdx}][-1 - \text{refIdx} + \text{refH}]$

[616]

[617] 하지만 이에 한정되는 것은 아니다. 만약 인트라 예측 모드의 인덱스가 34보다 작으면, 인코더 또는 디코더는 아래의 과정을 수행한다.

[618] 1-1. The reference sample array $\text{ref}[x]$ is specified as follows:

[619] - The following applies:

[620] $\text{ref}[x] = p[-1 - \text{refIdx}][-1 - \text{refIdx} + x]$, with $x = 0..nTbH + \text{refIdx} + 1$

[621] - If intraPredAngle is less than 0, the main reference sample array is extended as follows:

[622] $\text{ref}[x] = p[-1 - \text{refIdx} + \text{Min}((x * \text{invAngle} + 256) >> 9, nTbW)][-1 - \text{refIdx}]$,
with $x = -nTbW..-1$

[623] - Otherwise,

[624] $\text{ref}[x] = p[-1 - \text{refIdx}][-1 - \text{refIdx} + x]$, with $x = nTbH + 2 + \text{refIdx}.. \text{refH} + \text{refIdx}$

[625] - refIdx 가 0보다 큰 때, 추가 참조 샘플들 $\text{ref}[\text{refH} + x]$ with $x = 1..(\text{Max}(1, nTbW / nTbH) * \text{refIdx} + 1)$ 은 아래와 같이 패딩될 수 있다.

[626] 직사각형 블록을 고려하는 경우 패딩되는 샘플 수는 $(\text{Max}(1, nTbW / nTbH) * \text{refIdx} + 1)$ 일 수 있다. 인코더 또는 디코더는 이하와 같이 패딩을 수행할 수 있다.

[627] $\text{ref}[\text{refH} + \text{refIdx} + x] = p[-1 + \text{refH}][-1 - \text{refIdx}]$

[628]

[629] 상술한 본 발명의 실시예들은 다양한 수단을 통해 구현될 수 있다. 예를 들어, 본 발명의 실시예들은 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다.

[630] 하드웨어에 의한 구현의 경우, 본 발명의 실시예들에 따른 방법은 하나 또는 그 이상의 ASICs(Application Specific Integrated Circuits), DSPs(Digital Signal Processors), DSPDs(Digital Signal Processing Devices), PLDs(Programmable Logic Devices), FPGAs(Field Programmable Gate Arrays), 프로세서, 컨트롤러, 마이크로 컨트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[631] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 실시예들에 따른 방법은 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차 또는 함수 등의 형태로 구현될 수 있다. 소프트웨어 코드는 메모리에 저장되어 프로세서에 의해 구동될 수 있다. 상기 메모리는 프로세서의 내부 또는 외부에 위치할 수 있으며, 이미 공지된 다양한 수단에 의해 프로세서와 데이터를 주고받을 수 있다.

[632] 일부 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고,

휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조 또는 프로그램 모듈과 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 매커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.

- [633] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로 해석해야 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.
- [634] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

청구범위

- [청구항 1] 인트라 예측에 기초하여 비디오 신호를 처리하는 방법에 있어서,
수신된 비트스트림으로부터 현재 블록의 참조 샘플 라인 인덱스를
획득하는 단계;
상기 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계;
상기 참조 샘플 라인 인덱스가 0인 경우, 상기 현재 블록이 planar 모드를
사용하는지 여부를 나타내는 정보를 상기 비트스트림으로부터 획득하는
단계;
상기 planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 인트라
예측 모드를 결정하는 단계; 및
상기 결정된 인트라 예측 모드에 기초하여 상기 현재 블록을 예측하는
단계를 포함하는 것을 특징으로 하는 비디오 신호 처리 방법.
- [청구항 2] 제 1 항에 있어서,
상기 참조 샘플 라인 인덱스가 0이 아닌 경우, 상기 planar 모드를
사용하는지 여부를 나타내는 정보를 상기 비트스트림으로부터 획득하지
않고, 상기 planar 모드를 사용하는지 여부를 나타내는 정보가 상기
비트스트림으로부터 획득되지 않는 경우, 상기 planar 모드를 사용하는지
여부를 나타내는 정보는 상기 planar 모드를 사용하지 않음을 나타내는
것을 특징으로 하는 비디오 신호 처리 방법.
- [청구항 3] 제 1 항에 있어서,
상기 현재 블록이 코딩 트리 유닛의 최상측에 포함되는지 여부를
결정하는 단계; 및
상기 현재 블록이 상기 코딩 트리 유닛의 최상측에 포함되지 않은 경우,
상기 참조 샘플 라인 인덱스를 상기 비트스트림으로부터 획득하는
단계를 포함하는 것을 특징으로 하는 비디오 신호 처리 방법.
- [청구항 4] 제 3 항에 있어서,
상기 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계;
상기 참조 샘플 라인 인덱스가 0인 경우, MPM(Most Probable Modes)
사용여부를 나타내는 정보를 상기 비트스트림으로부터 획득하는 단계;
및
상기 MPM을 사용여부를 나타내는 정보가 MPM을 사용함을 나타내고,
상기 참조 샘플 라인 인덱스가 0인 경우, 상기 planar 모드를 사용하는지
여부를 나타내는 정보를 상기 비트스트림으로부터 획득하는 단계를
포함하는 것을 특징으로 하는 비디오 신호 처리 방법.
- [청구항 5] 제 1 항에 있어서,
상기 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를
사용하는 것을 나타내는지 여부를 결정하는 단계;

상기 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를 사용하지 않는 것을 나타내는 경우, MPM인덱스를 상기 비트스트림으로부터 획득하는 단계;
 상기 현재 블록의 미리 복원된 좌측 블록 또는 미리 복원된 상측 블록 중 적어도 하나에 기초하여, 적어도 하나의 각도 모드를 포함하는 MPM리스트를 생성하는 단계;
 상기 MPM인덱스에 기초하여 상기 MPM리스트에서 하나의 인트라 예측 모드를 선택하는 단계; 및
 상기 선택된 인트라 예측 모드에 기초하여 상기 현재 블록을 예측하는 단계를 포함하는 것을 특징으로 하는 비디오 신호 처리 방법.

[청구항 6] 인트라 예측에 기초하여 비디오 신호를 처리하는 장치로써,
 프로세서 및 메모리를 포함하고, 상기 프로세서는 상기 메모리에 저장된 명령어에 기초하여,
 수신된 비트스트림으로부터 현재 블록의 참조 샘플 라인 인덱스를 획득하는 단계;
 상기 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계;
 상기 참조 샘플 라인 인덱스가 0인 경우, 상기 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보를 상기 비트스트림으로부터 획득하는 단계;
 상기 planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 인트라 예측 모드를 결정하는 단계; 및
 상기 결정된 인트라 예측 모드에 기초하여 상기 현재 블록을 예측하는 단계를 수행하는 것을 특징으로 하는 비디오 신호 처리 장치.

[청구항 7] 제 6 항에 있어서,
 상기 프로세서는 상기 메모리에 저장된 명령어에 기초하여,
 참조 샘플 라인 인덱스가 0이 아닌 경우, 상기 planar 모드를 사용하는지 여부를 나타내는 정보를 상기 비트스트림으로부터 획득하지 않고, 상기 현재 블록의 상기 planar 모드를 사용하는지 여부를 나타내는 정보가 상기 비트스트림으로부터 획득되지 않는 경우, 상기 planar 모드를 사용하는지 여부를 나타내는 정보는 상기 planar 모드를 사용하지 않음을 나타내는 것을 특징으로 하는 비디오 신호 처리 장치.

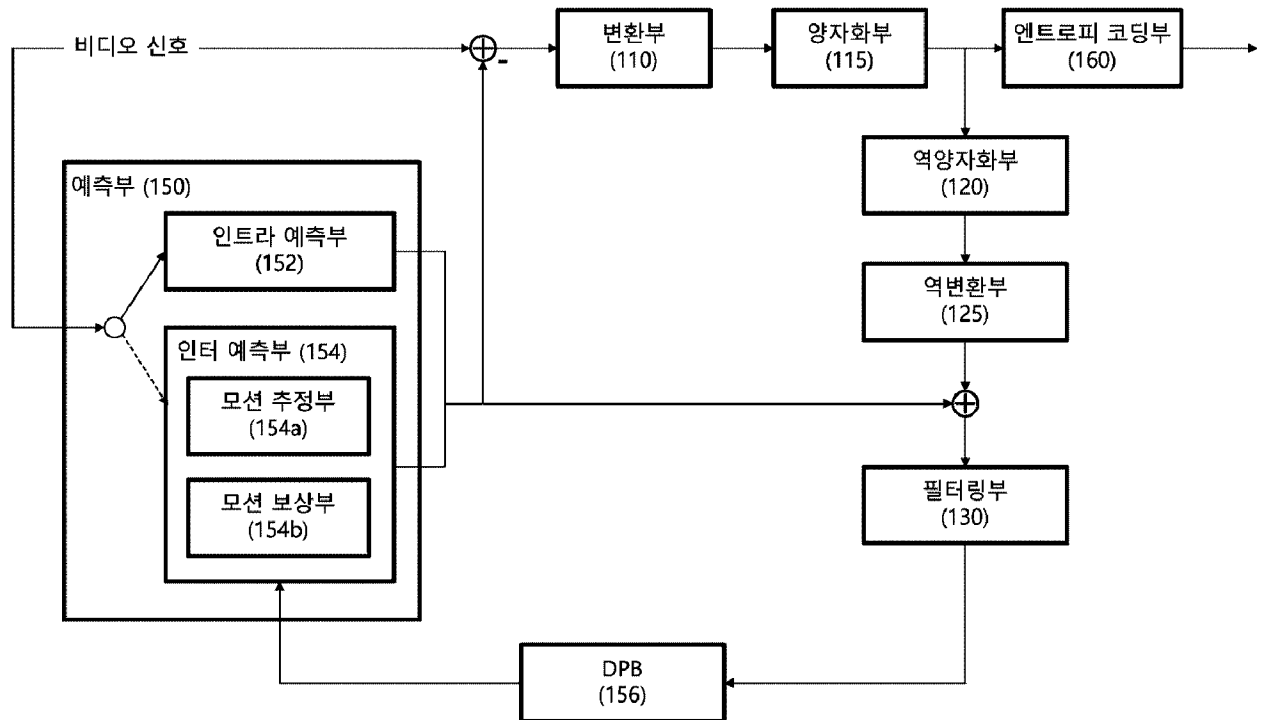
[청구항 8] 제 6 항에 있어서,
 상기 프로세서는 상기 메모리에 저장된 명령어에 기초하여,
 상기 현재 블록이 코딩 트리 유닛의 최상측에 포함되는지 여부를 결정하는 단계; 및
 상기 현재 블록이 상기 코딩 트리 유닛의 최상측에 포함되지 않은 경우, 상기 참조 샘플 라인 인덱스를 상기 비트스트림으로부터 획득하는 단계를 수행하는 것을 특징으로 하는 비디오 신호 처리 방법.

- [청구항 9] 제 8 항에 있어서,
 상기 참조 샘플 라인 인덱스가 0인지 여부를 결정하는 단계;
 상기 참조 샘플 라인 인덱스가 0인 경우, MPM(Most Probable Modes)
 사용여부를 나타내는 정보를 상기 비트스트림으로부터 획득하는 단계;
 및
 상기 MPM을 사용여부를 나타내는 정보가 MPM을 사용함을 나타내고,
 상기 참조 샘플 라인 인덱스가 0인 경우, 상기 planar 모드를 사용하는지
 여부를 나타내는 정보를 상기 비트스트림으로부터 획득하는 단계를
 수행하는 것을 특징으로 하는 비디오 신호 처리 장치.
- [청구항 10] 제 6 항에 있어서,
 상기 프로세서는 상기 메모리에 저장된 명령어에 기초하여,
 상기 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를
 사용하는 것을 나타내는지 여부를 결정하는 단계;
 상기 planar 모드를 사용하는지 여부를 나타내는 정보가 planar 모드를
 사용하지 않는 것을 나타내는 경우, MPM인덱스를 상기
 비트스트림으로부터 획득하는 단계;
 상기 현재 블록의 미리 복원된 좌측 블록 또는 미리 복원된 상측 블록 중
 적어도 하나에 기초하여, 적어도 하나의 각도 모드를 포함하는
 MPM리스트를 생성하는 단계;
 상기 MPM인덱스에 기초하여 상기 MPM리스트에서 하나의 인트라 예측
 모드를 선택하는 단계; 및
 상기 선택된 인트라 예측 모드에 기초하여 상기 현재 블록을 예측하는
 단계를 수행하는 것을 특징으로 하는 비디오 신호 처리 방법.
- [청구항 11] 인트라 예측에 기초하여 비디오 신호를 인코딩하는 방법에 있어서,
 현재 블록이 MPM(Most Probable Modes)을 사용하고, 상기 현재 블록의
 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지
 여부를 나타내는 정보를 생성하는 단계; 및
 상기 planar 모드를 사용하는지 여부를 나타내는 정보를 포함하는
 비트스트림을 생성하는 단계를 포함하는 것을 특징으로 하는 비디오
 신호 인코딩 방법.
- [청구항 12] 인트라 예측에 기초하여 비디오 신호를 인코딩하는 장치로써,
 프로세서 및 메모리를 포함하고, 상기 프로세서는 상기 메모리에 저장된
 명령어에 기초하여,
 현재 블록이 MPM(Most Probable Modes)을 사용하고, 상기 현재 블록의
 참조 샘플 라인 인덱스가 0인 경우, 현재 블록이 planar 모드를 사용하는지
 여부를 나타내는 정보를 생성하는 단계; 및
 상기 planar 모드를 사용하는지 여부를 나타내는 정보를 포함하는
 비트스트림을 생성하는 단계를 수행하는 것을 특징으로 하는 비디오

- 신호를 인코딩 장치.
- [청구항 13] 비일시적 컴퓨터가 읽을 수 있는 매체에 저장된 비디오 비트스트림을 포함하는 장치에 있어서,
상기 비트스트림은,
현재 블록이 MPM(Most Probable Modes)을 사용하고, 상기 현재 블록의 참조 샘플 라인 인덱스가 0인 경우, 인코더에 의해 생성된 현재 블록이 planar 모드를 사용하는지 여부를 나타내는 정보에 기초하여 생성되는 것을 특징으로 하는 장치.

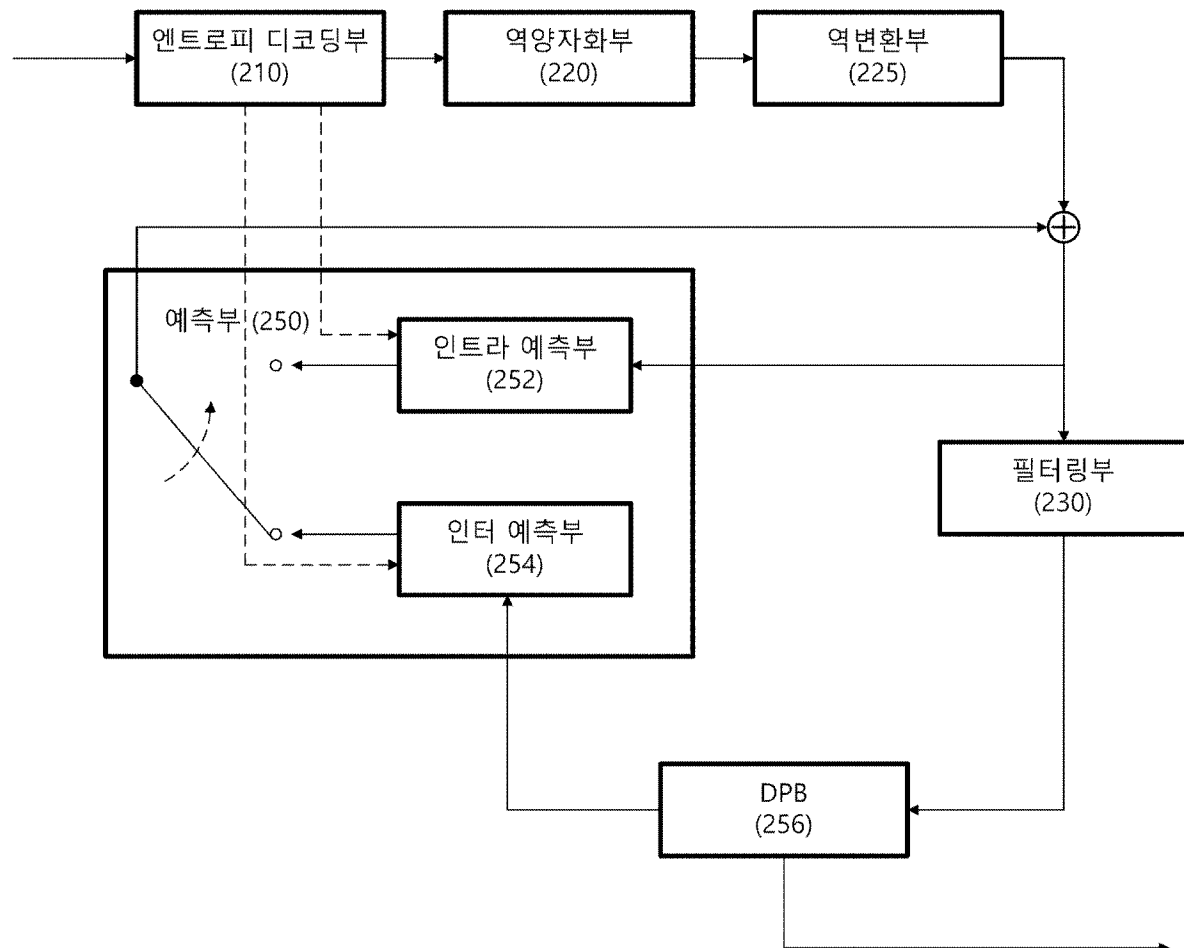
[도1]

100

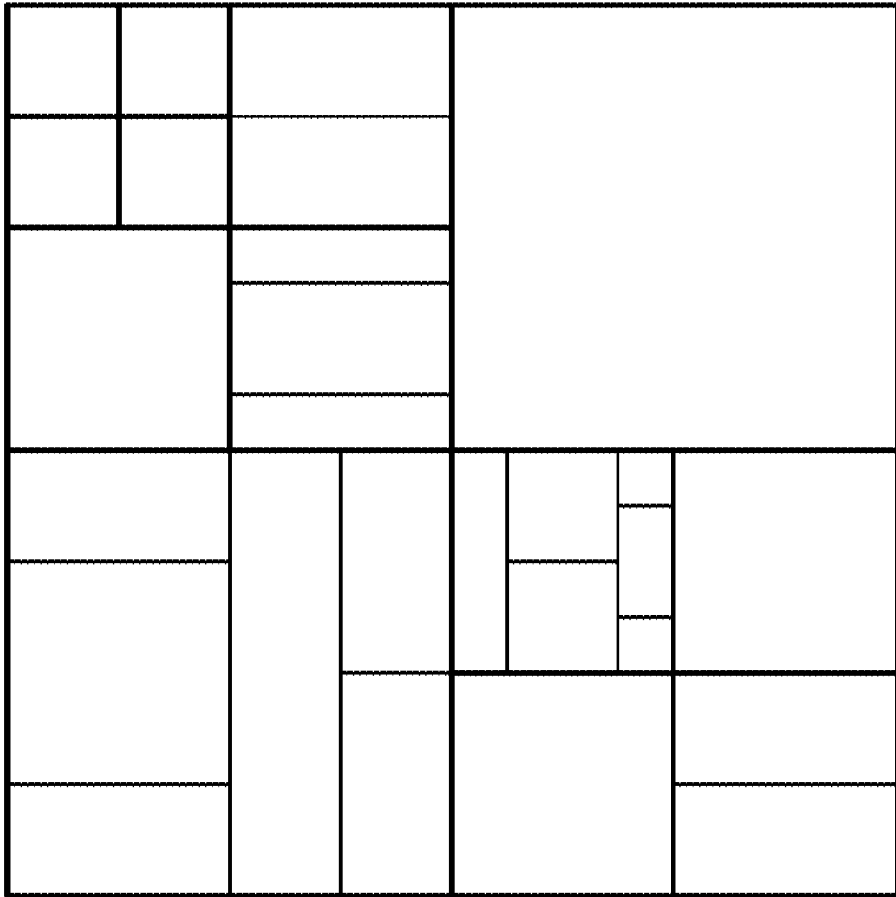


[도2]

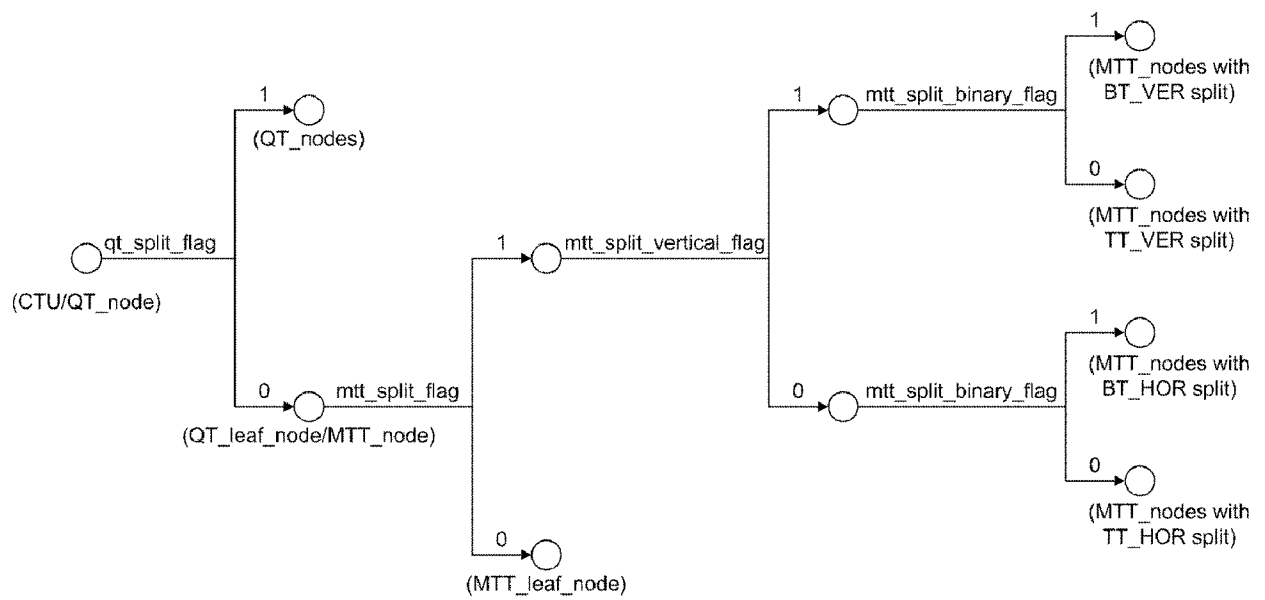
200



[도3]



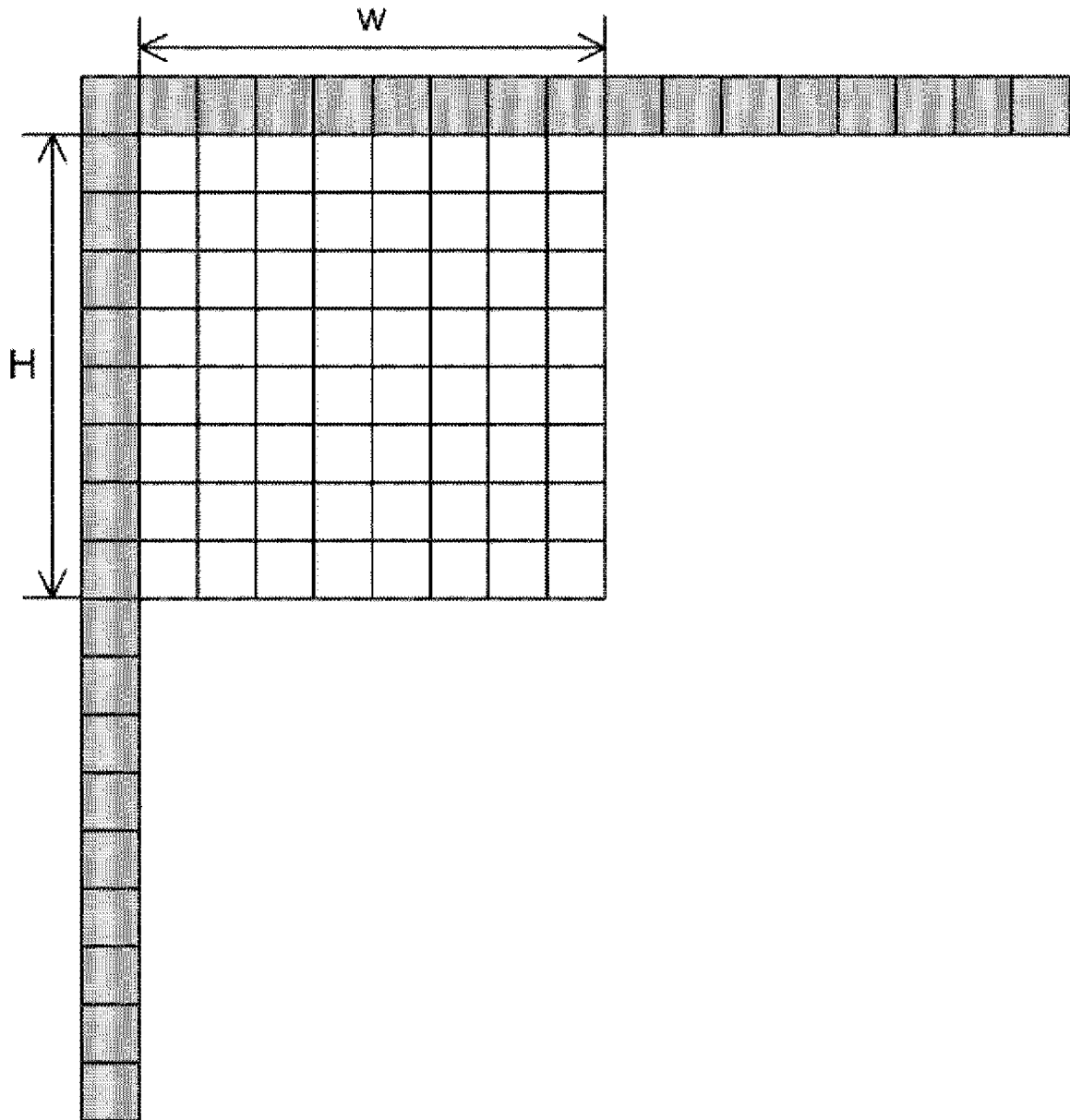
[도4]



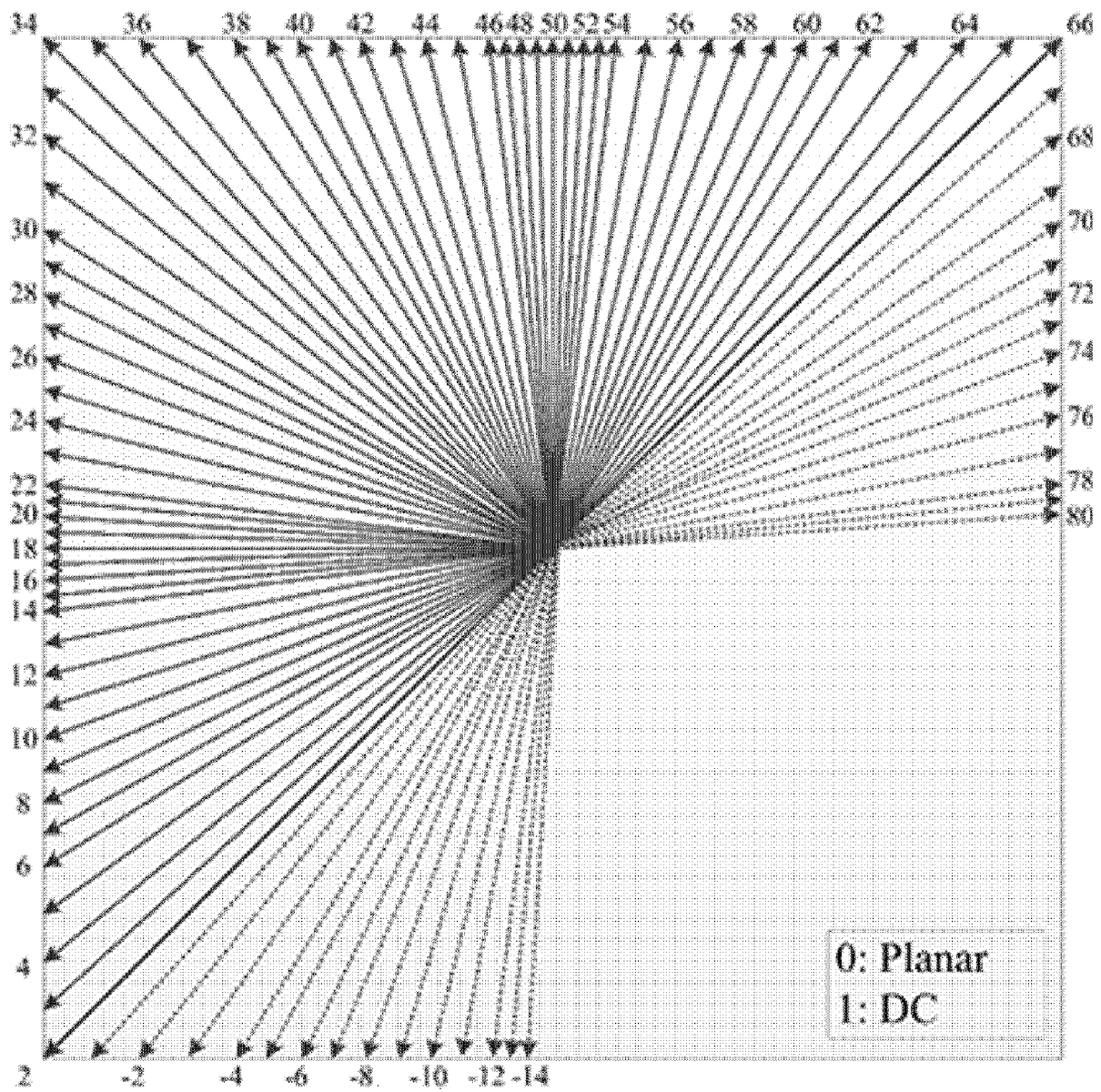
[도5]

■ 참조 샘플

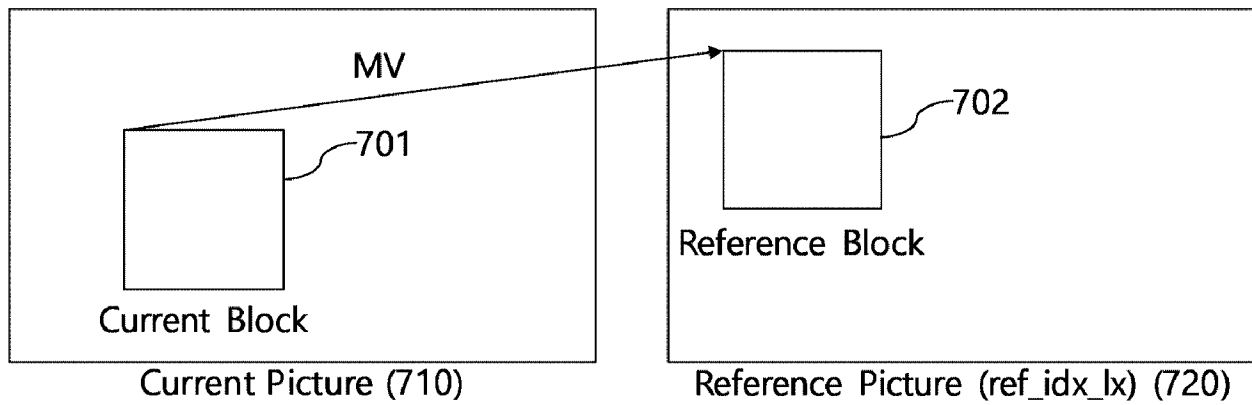
□ 현재 유닛의 샘플



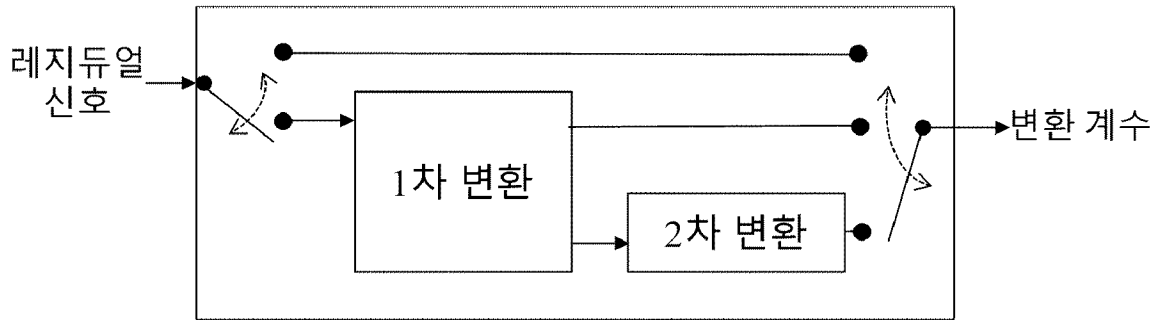
[도6]



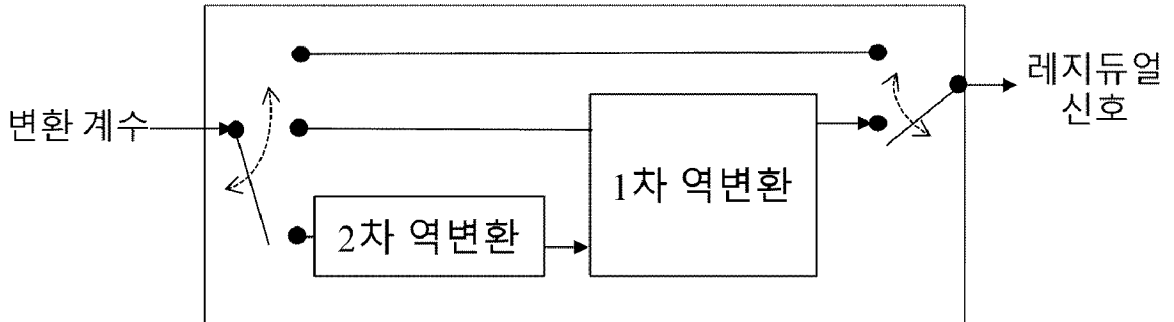
[도7]



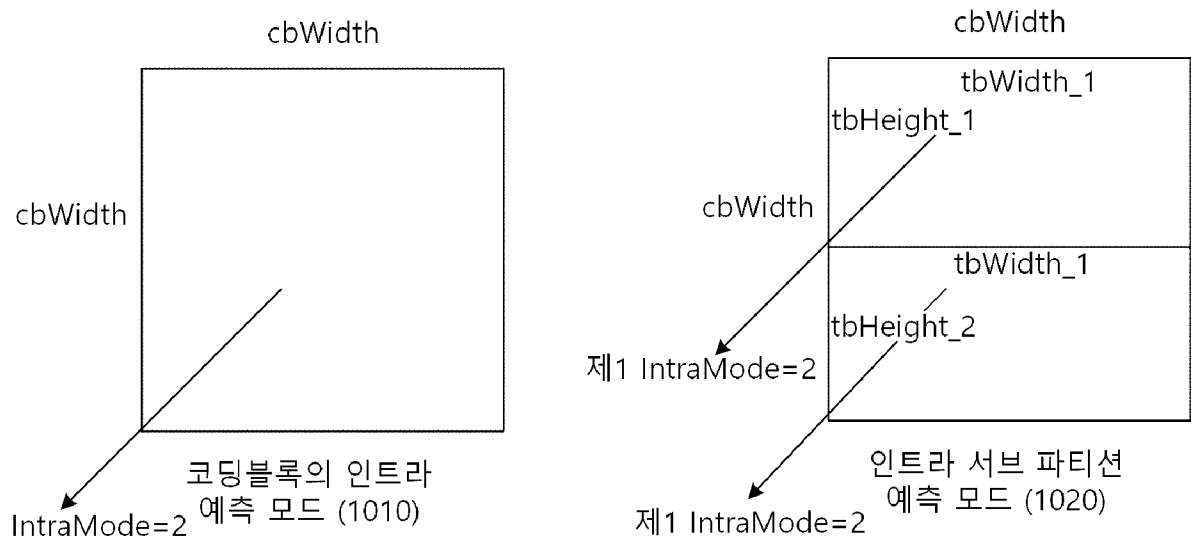
[도8]



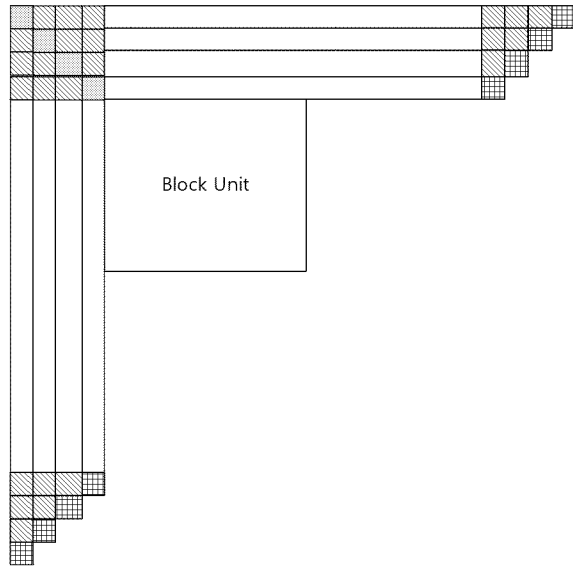
[도9]



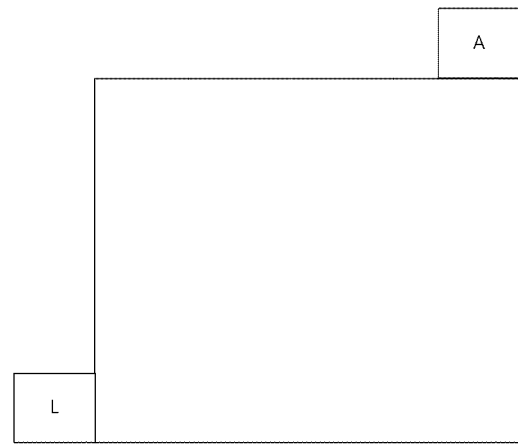
[도10]



[도11]



Multiple reference simple line (1110)



Neighboring modes for MPM generation (1120)

[도 12]

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	
if(tile_group_type != I) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx [x0][y0]	ae(v)
if (intra_luma_ref_idx[x0][y0] = = 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag [x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] = = 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] = = 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	

Coding unit Syntax 구조

[도13]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_mode_Planar_flag[x0][y0]	ae(v)
If(Intra_mode_Planar_flag[x0][y0] == 0 && (y0 % CtbSizeY) > 0))	
intra_luma_ref_idx[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag[x0][y0]	
if(intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	ae(v)
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0 && intra_mode_Planar_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	
...	

Coding unit Syntax 구조

[도 14]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_mode_Planar_flag[x0][y0]	ae(v)
if(Intra_mode_Planar_flag[x0][y0] == 0 && (y0 % CtbSizeY) > 0)	
intra_luma_ref_idx[x0][y0]	ae(v)
if((intra_luma_ref_idx[x0][y0] = = 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY) && Intra_mode_Planar_flag[x0][y0] == 0)	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] = = 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] = = 0 && intra_subpartitions_mode_flag[x0][y0] = = 0 && intra_mode_Planar_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	
...	

Coding unit Syntax 구조

[도15]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx [x0][y0]	ae(v)
if (intra_luma_ref_idx[x0][y0] = = 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag [x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] = = 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag [x0][y0]	
If(intra_luma_ref_idx[x0][y0] == 0)	
intra_mode_Planar_flag [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] = = 0 && intra_subpartitions_mode_flag[x0][y0] = = 0 && intra_mode_Planar_flag[x0][y0]==0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	
...	

Coding unit Syntax 구조

[도16]

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	
if(tile_group_type != I) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag	&&
cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY &&	
cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx[x0][y0]	ae(v)
intra_mode_Planar_flag[x0][y0]	ae(v)
if((intra_luma_ref_idx[x0][y0] == 0	&&
(cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) &&	
(cbWidth * cbHeight > MinTbSizeY * MinTbSizeY)	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1	&&
cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0	&&
intra_subpartitions_mode_flag[x0][y0] == 0 && intra_mode_Planar_flag[x0][y0] ==	
0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	
if(tile_group_type != I) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag ;	&&
cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY &&	&&
cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx [x0][y0]	ae(v)
intra_mode_Planar_flag [x0][y0]	ae(v)
if((intra_luma_ref_idx[x0][y0] == 0 &&	&&
(cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) &&	&&
(cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag [x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1 &&	&&
cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 &&	&&
intra_subpartitions_mode_flag[x0][y0] == 0 && intra_mode_Planar_flag[x0][y0] ==	
0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	

[도18]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(tile_group_type != I) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_mode_Planar_flag [x0][y0]	ae(v)
if (intra_luma_ref_idx[x0][y0] = = 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag [x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] = = 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag [x0][y0]	ae(v)
if(intra_mode_Planar_flag[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	

[도19]

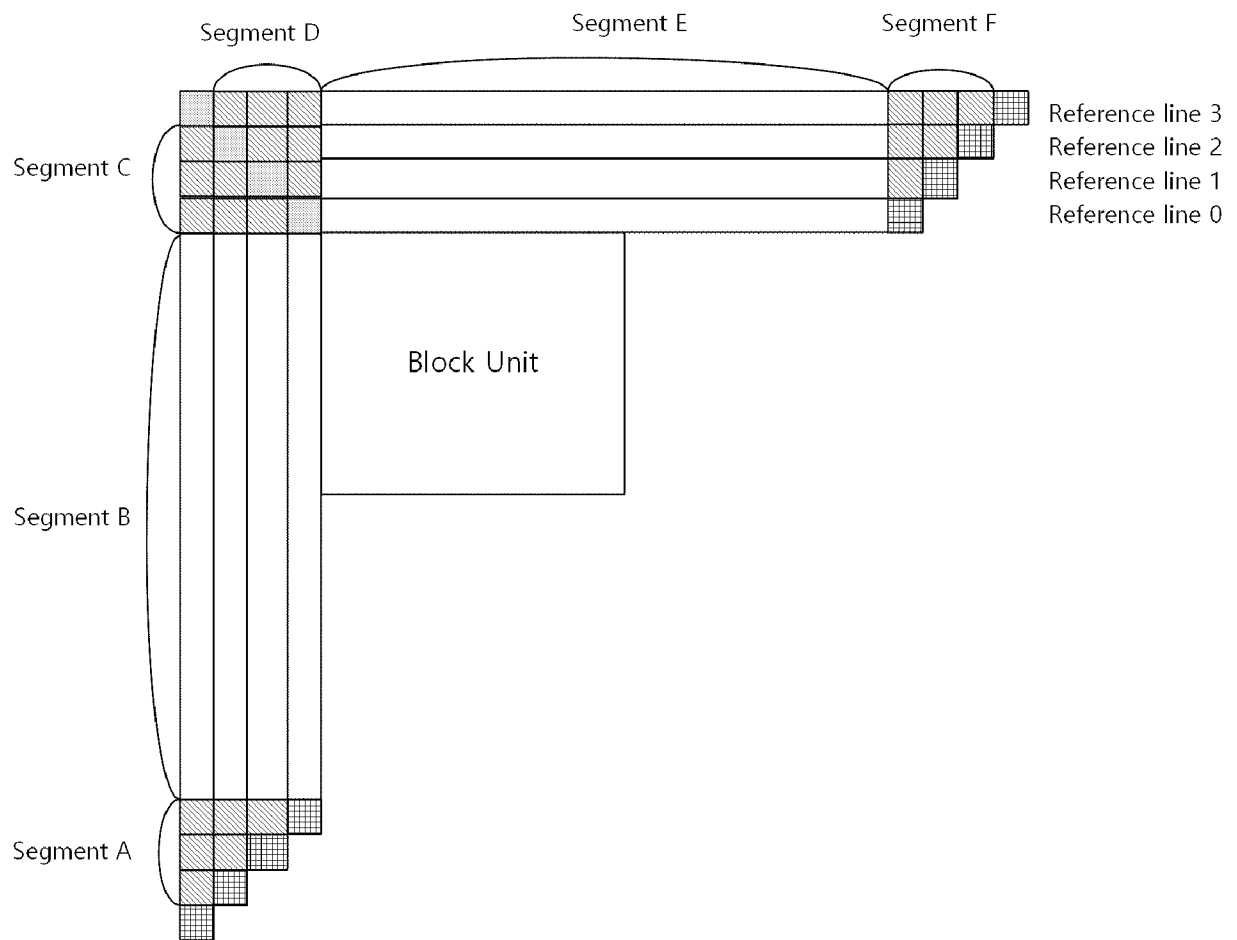
	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	
if(tile_group_type != 1) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_mode_Planar_flag[x0][y0]	ae(v)
if(intra_luma_Planar_flag[x0][y0] == 0 && y0 % CtbSizeY > 0)	
intra_luma_ref_idx[x0][y0]	ae(v)
if (intra_luma_ref_idx[x0][y0] = = 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] = = 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_mode_Planar_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	

[도20]

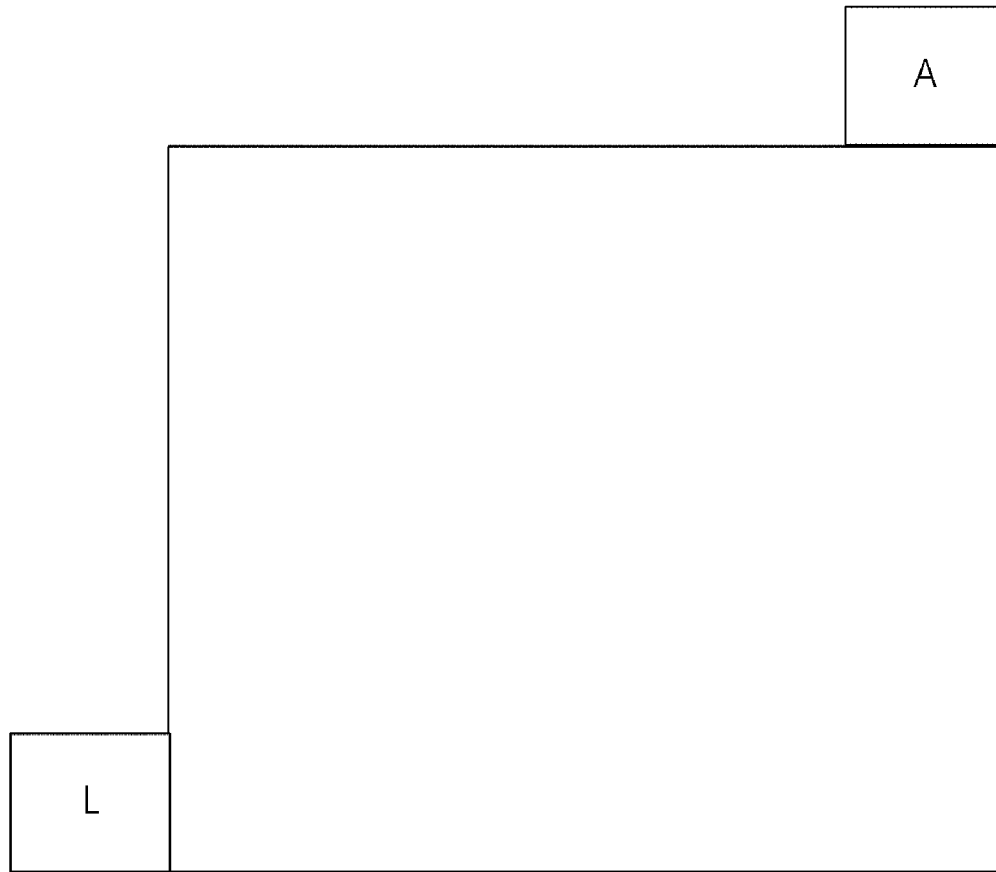
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(tile_group_type != I) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag [x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]) {	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_planar_flag [x0][y0]	ae(v)
if(intra_luma_planar_flag[x0][y0] == 0)	
intra_luma_mpm_idx [x0][y0]	
} else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */	

Coding unit Syntax 구조

[도21]

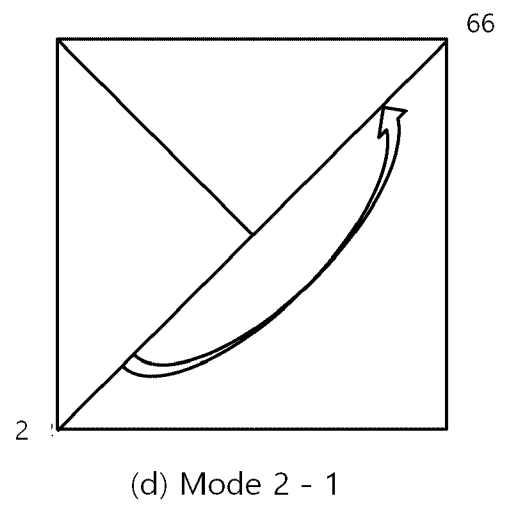
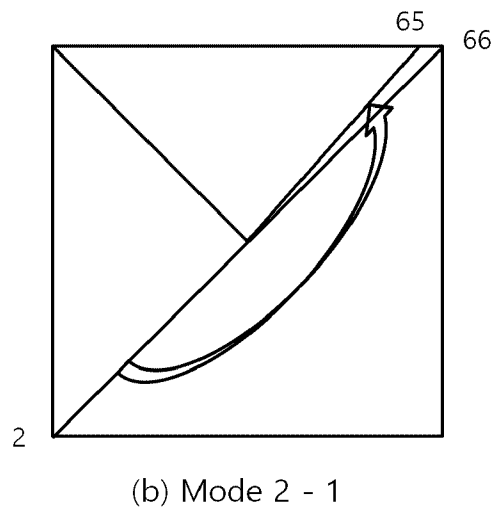
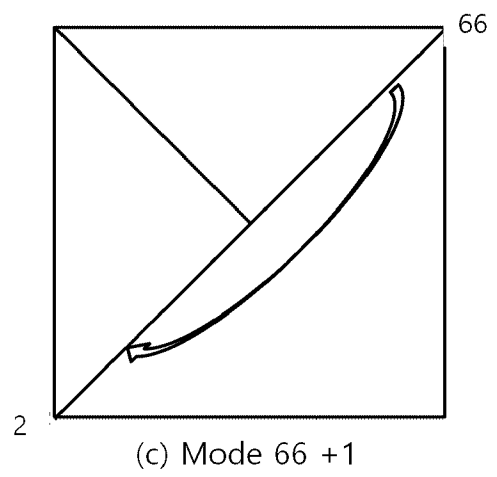
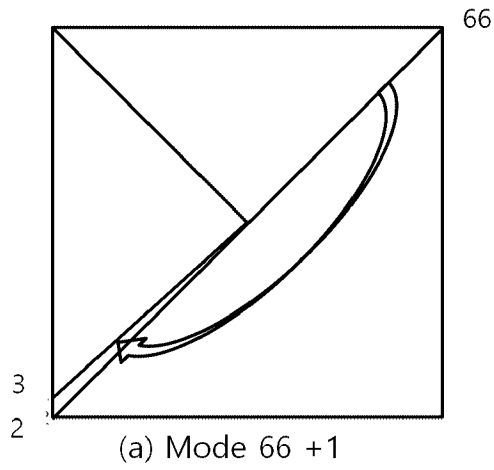


[도22]

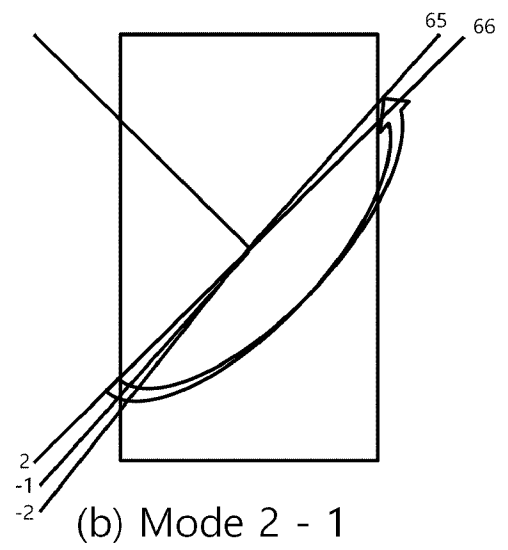
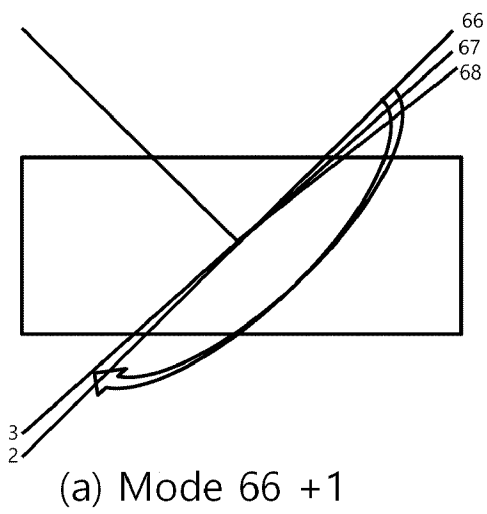


Neighboring modes for MPM generation

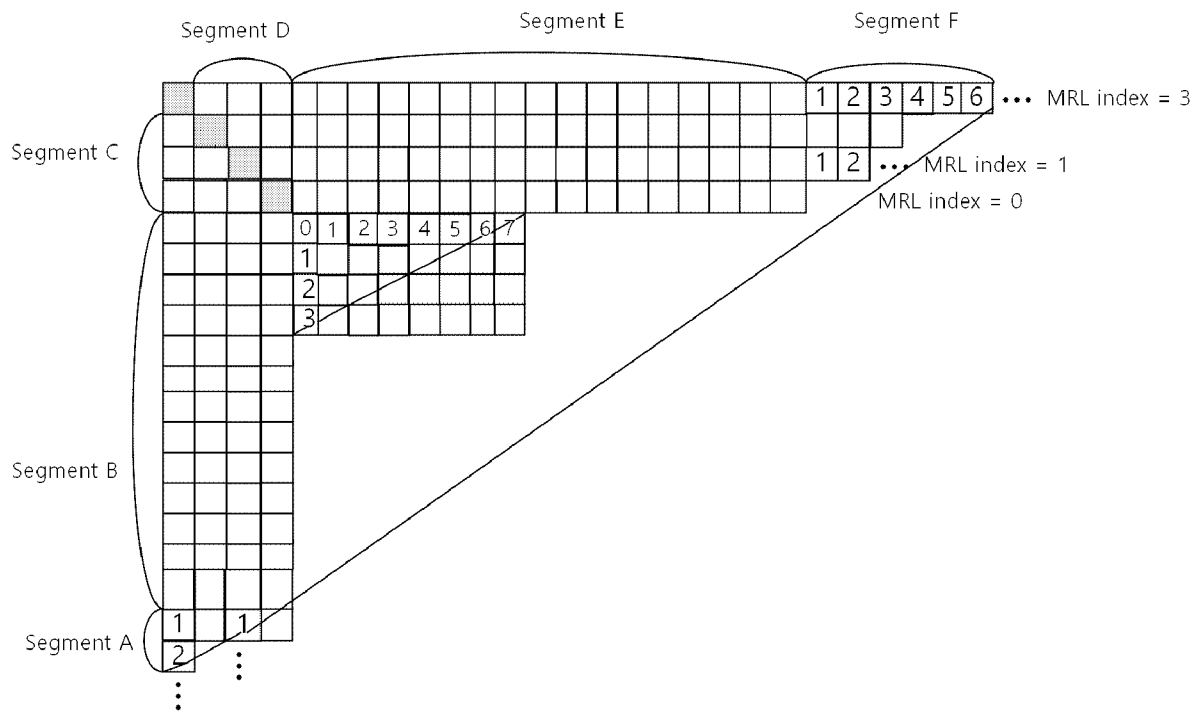
[도23]



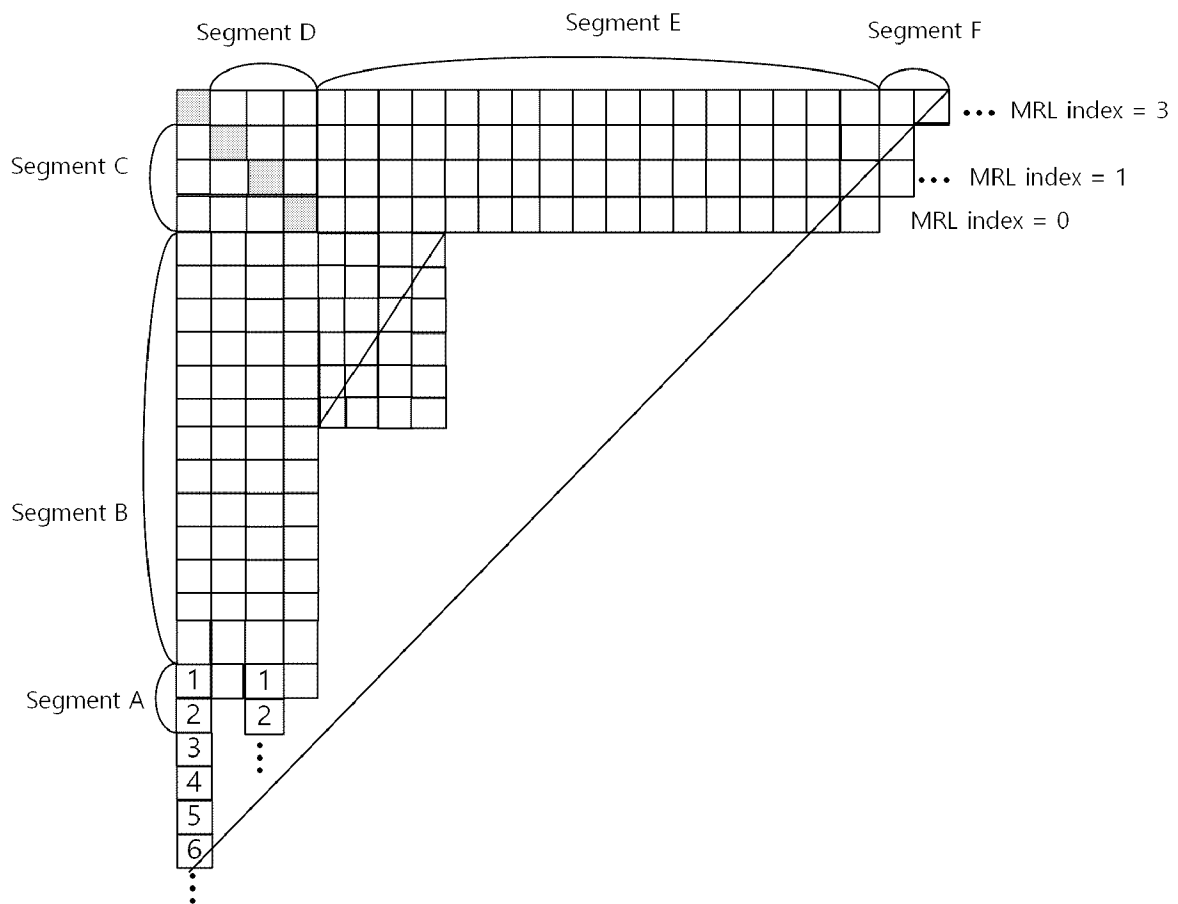
[도24]



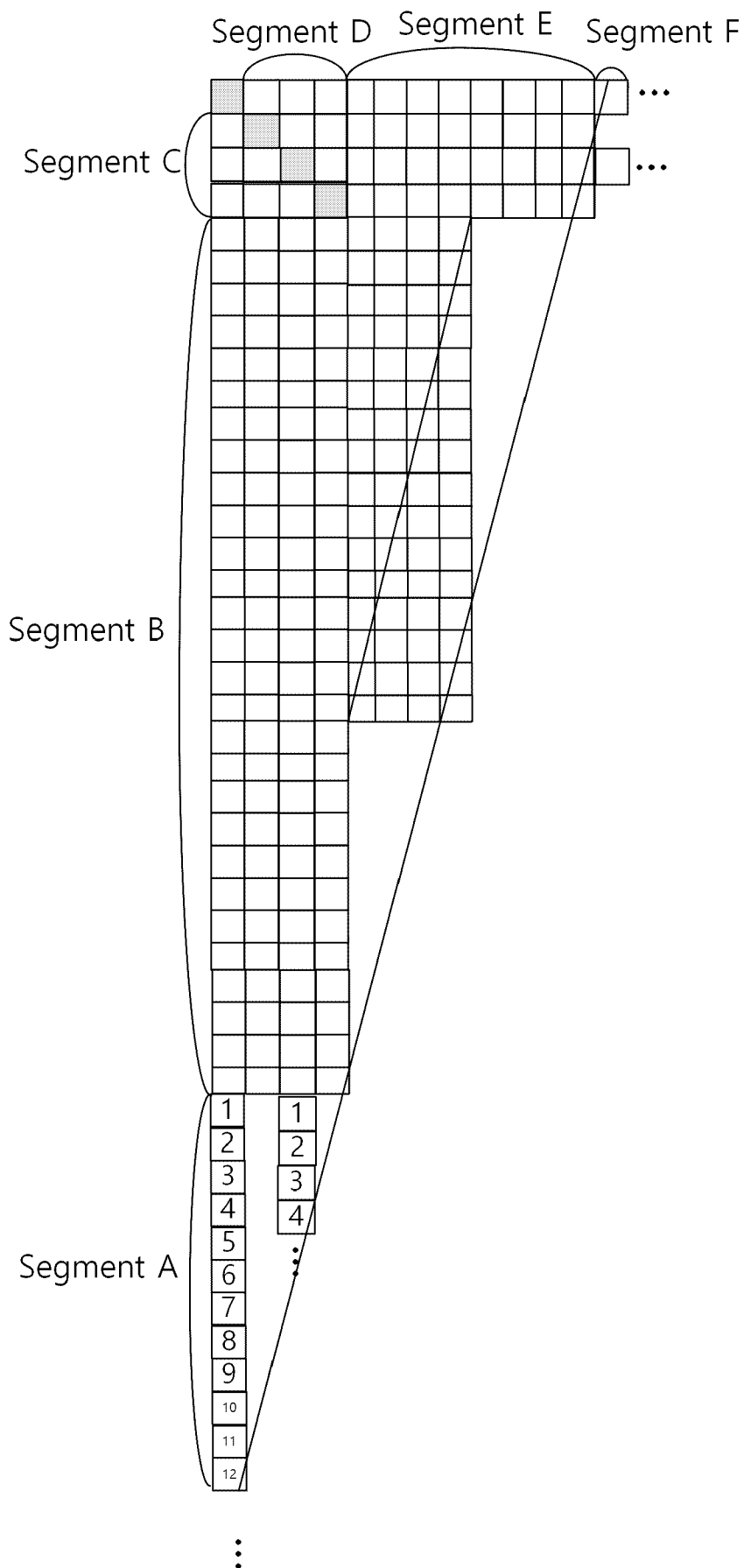
[도25]



[도26]



[도27]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2019/013137

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/11(2014.01)i, H04N 19/70(2014.01)i, H04N 19/593(2014.01)i, H04N 19/176(2014.01)i, H04N 19/119(2014.01)i, H04N 19/105(2014.01)i, H04N 19/186(2014.01)i, H04N 19/82(2014.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N 19/11; H03M 7/48; H04N 19/124; H04N 19/184; H04N 7/34; H04N 19/70; H04N 19/593; H04N 19/176; H04N 19/119; H04N 19/105; H04N 19/186; H04N 19/82

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models: IPC as above

Japanese utility models and applications for utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & Keywords: MPM, MRL, Planar

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JIANG, Hui-yu et al. CE3-related: Advanced MPM based on intra reference line selection scheme. JVET-K0175-v3. Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. 11th Meeting. pages 1-4, 14 July 2018 See pages 1-2.	1-13
A	KR 10-2018-0014109 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 07 February 2018 See paragraph [0010]; and claim 1.	1-13
A	KR 10-2013-0045152 A (INTEAPP CO., LTD.) 03 May 2013 See paragraphs [0015]-[0029]; and claim 1.	1-13
A	WO 2016-198169 A1 (GURULOGIC MICROSYSTEMS OY.) 15 December 2016 See claims 1-5.	1-13
A	KR 10-2014-0071429 A (INFOBRIDGE PTE. LTD.) 11 June 2014 See paragraphs [0010]-[0016]; and claim 1.	1-13



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

17 JANUARY 2020 (17.01.2020)

Date of mailing of the international search report

17 JANUARY 2020 (17.01.2020)

Name and mailing address of the ISA/KR



Korean Intellectual Property Office
Government Complex Daejeon Building 4, 189, Cheongsu-ro, Seo-gu,
Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2019/013137

Patent document cited in search report	Publication date	Patent family member	Publication date
KR 10-2018-0014109 A	07/02/2018	CN 107105236 A	29/08/2017
		CN 107105237 A	29/08/2017
		EP 3442234 A1	13/02/2019
		JP 2014-501092 A	16/01/2014
		KR 10-2019-0006043 A	16/01/2019
		US 2018-0376138 A1	27/12/2018
		WO 2012-087035 A2	28/06/2012
KR 10-2013-0045152 A	03/05/2013	CN 107105262 A	29/08/2017
		EP 2950535 A2	02/12/2015
		HK 1214445 A1	22/07/2016
		JP 2015-146629 A	13/08/2015
		KR 10-2013-0045148 A	03/05/2013
		MX 2014003707 A	28/07/2014
		US 2016-0044326 A1	11/02/2016
		US 2018-0084268 A1	22/03/2018
		WO 2013-062191 A1	02/05/2013
WO 2016-198169 A1	15/12/2016	EP 3469711 A1	17/04/2019
		GB 2539239 A	14/12/2016
		KR 10-2018-0016556 A	14/02/2018
		RU 2017146242 A	10/07/2019
		US 2018-0167084 A1	14/06/2018
KR 10-2014-0071429 A	11/06/2014	CN 106254882 A	21/12/2016
		CY 1120151 T1	12/12/2018
		DK 2752004 T3	28/05/2018
		EP 3579556 A1	11/12/2019
		ES 2670014 T3	29/05/2018
		HK 1214702 A1	29/07/2016
		HR P20180864 T1	13/07/2018
		HU E037573 T2	28/09/2018
		JP 2016-007016 A	14/01/2016
		KR 10-2014-0074347 A	17/06/2014
		RS 57159 B1	31/07/2018
		SI 2752004 T1	31/05/2018
		TW 201904289 A	16/01/2019
		US 2019-0141327 A1	09/05/2019
		US 2019-0342553 A1	07/11/2019
		WO 2013-064098 A1	10/05/2013

A. 발명이 속하는 기술분류(국제특허분류(IPC))

H04N 19/11(2014.01)i, H04N 19/70(2014.01)i, H04N 19/593(2014.01)i, H04N 19/176(2014.01)i, H04N 19/119(2014.01)i, H04N 19/105(2014.01)i, H04N 19/186(2014.01)i, H04N 19/82(2014.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)

H04N 19/11; H03M 7/48; H04N 19/124; H04N 19/184; H04N 7/34; H04N 19/70; H04N 19/593; H04N 19/176; H04N 19/119; H04N 19/105; H04N 19/186; H04N 19/82

조사된 기술분야에 속하는 최소문헌 이외의 문헌

한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC
일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))

eKOMPASS(특허청 내부 검색시스템) & 키워드: MPM, MRL, Planar

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	HUI-YU JIANG 등, 'CE3-related: Advanced MPM based on intra reference line selection scheme', JVET-K0175-v3, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 11th Meeting, 페이지 1-4, 2018.07.14 페이지 1-2 참조.	1-13
A	KR 10-2018-0014109 A (한국전자통신연구원) 2018.02.07 단락 [0010]; 및 청구항 1 참조.	1-13
A	KR 10-2013-0045152 A ((주)인터랩) 2013.05.03 단락 [0015]-[0029]; 및 청구항 1 참조.	1-13
A	WO 2016-198169 A1 (GURULOGIC MICROSYSTEMS OY) 2016.12.15 청구항 1-5 참조.	1-13
A	KR 10-2014-0071429 A (인포브릿지 피티이 엘티디) 2014.06.11 단락 [0010]-[0016]; 및 청구항 1 참조.	1-13

☐ 추가 문헌이 C(계속)에 기재되어 있습니다.

☒ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

“A” 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌

“D” 본 국제출원에서 출원인이 인용한 문헌

“E” 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후 “X”에 공개된 선출원 또는 특허 문헌

“L” 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌

“O” 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌

“P” 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

“T” 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌

“X” 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.

“Y” 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.

“&” 동일한 대응특허문헌에 속하는 문헌

국제조사의 실제 완료일

2020년 01월 17일 (17.01.2020)

국제조사보고서 발송일

2020년 01월 17일 (17.01.2020)

ISA/KR의 명칭 및 우편주소



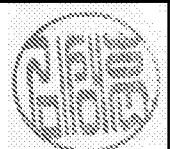
대한민국 특허청
(35208) 대전광역시 서구 청사로 189,
4동 (둔산동, 정부대전청사)

팩스 번호 +82-42-481-8578

심사관

김성훈

전화번호 +82-42-481-8710



국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
KR 10-2018-0014109 A	2018/02/07	CN 107105236 A CN 107105237 A EP 3442234 A1 JP 2014-501092 A KR 10-2019-0006043 A US 2018-0376138 A1 WO 2012-087035 A2	2017/08/29 2017/08/29 2019/02/13 2014/01/16 2019/01/16 2018/12/27 2012/06/28
KR 10-2013-0045152 A	2013/05/03	CN 107105262 A EP 2950535 A2 HK 1214445 A1 JP 2015-146629 A KR 10-2013-0045148 A MX 2014003707 A US 2016-0044326 A1 US 2018-0084268 A1 WO 2013-062191 A1	2017/08/29 2015/12/02 2016/07/22 2015/08/13 2013/05/03 2014/07/28 2016/02/11 2018/03/22 2013/05/02
WO 2016-198169 A1	2016/12/15	EP 3469711 A1 GB 2539239 A KR 10-2018-0016556 A RU 2017146242 A US 2018-0167084 A1	2019/04/17 2016/12/14 2018/02/14 2019/07/10 2018/06/14
KR 10-2014-0071429 A	2014/06/11	CN 106254882 A CY 1120151 T1 DK 2752004 T3 EP 3579556 A1 ES 2670014 T3 HK 1214702 A1 HR P20180864 T1 HU E037573 T2 JP 2016-007016 A KR 10-2014-0074347 A RS 57159 B1 SI 2752004 T1 TW 201904289 A US 2019-0141327 A1 US 2019-0342553 A1 WO 2013-064098 A1	2016/12/21 2018/12/12 2018/05/28 2019/12/11 2018/05/29 2016/07/29 2018/07/13 2018/09/28 2016/01/14 2014/06/17 2018/07/31 2018/05/31 2019/01/16 2019/05/09 2019/11/07 2013/05/10