



(51) 국제특허분류:
H04N 19/117 (2014.01) H04N 19/176 (2014.01)
H04N 19/82 (2014.01) H04N 19/70 (2014.01)
H04N 19/119 (2014.01)

(21) 국제출원번호: PCT/KR2020/016558

(22) 국제출원일: 2020년 11월 23일 (23.11.2020)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2019-0151653 2019년 11월 22일 (22.11.2019) KR

(71) 출원인: 한국전자통신연구원 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) [KR/KR]; 34129 대전시 유성구 가정로 218, Daejeon (KR).

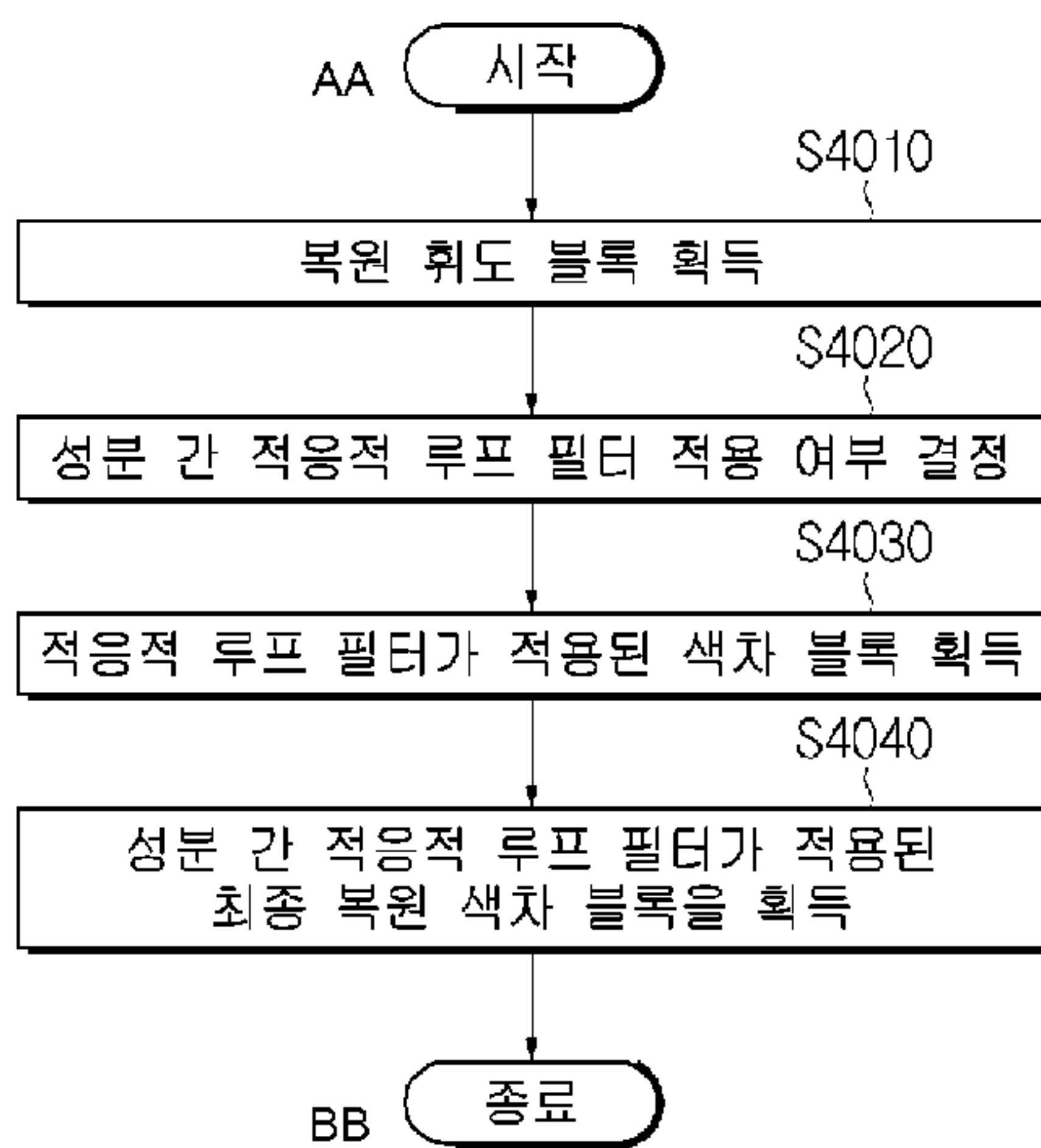
(72) 발명자: 임성창 (LIM, Sung Chang); 34189 대전시 유성구 도안동로 523 202동 1801호, Daejeon (KR). 강정원 (KANG, Jung Won); 34076 대전시 유성구 지족로 362 303동 303호, Daejeon (KR). 김수웅 (KIM, Soo Woong); 30150 세종시 남세종로 384 902동 802호, Sejong (KR). 방건 (BANG, Gun); 34076 대전시 유성구 지족로 362 303동 303호, Daejeon (KR). 이진호 (LEE, Jin Ho); 34075 대전시 유성구 지족동로 124 102동 1904호, Daejeon (KR). 이하현 (LEE, Ha Hyun); 02142 서울시 중랑구 동일로102길 34-8 402호, Seoul (KR).

(74) 대리인: 성병기 (SUNG, Byung Kee); 06651 서울시 서초구 사임당로 32 12층 마루특허법률사무소, Seoul (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,

(54) Title: ADAPTIVE IN-LOOP FILTERING METHOD AND DEVICE

(54) 발명의 명칭: 적응적 루프내 필터링 방법 및 장치



S4010 ... Acquire reconstructed luma block
S4020 ... Determine whether to apply cross-component adaptive loop filter
S4030 ... Acquire chroma block to which adaptive loop filter has been applied
S4040 ... Acquire final reconstructed chroma block to which cross-component adaptive loop filter has been applied
AA ... Start
BB ... End

(57) Abstract: The present invention relates to an image decoding method, and an image decoding method according to one embodiment of the present invention comprises the steps of: acquiring a reconstructed luma block of the current block; determining whether to apply a cross-component adaptive loop filter to the current block; acquiring a reconstructed chroma block of the current block to which the adaptive loop filter has been applied, if the cross-component adaptive loop filter is to be applied to the current block; and using the reconstructed luma block and the reconstructed chroma block to which the adaptive loop filter has been applied to acquire the final reconstructed chroma block of the current block to which the cross-component adaptive loop filter has been applied.

EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, 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, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 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), 유럽 (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))

(57) 요약서: 본 발명은 영상 복호화 방법에 관한 것으로서, 본 발명의 일 실시예에 따른 영상 복호화 방법은 현재 블록의 복원 휘도 블록을 획득하는 단계, 상기 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component adaptive loop filter)의 적용 여부를 결정하는 단계, 상기 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우, 적응적 루프 필터(adaptive loop filter)가 적용된 상기 현재 블록의 복원 색차 블록을 획득하는 단계 및 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의 최종 복원 색차 블록을 획득하는 단계를 포함하는, 영상 복호화 방법.

명세서

발명의 명칭: 적응적 루프내 필터링 방법 및 장치

기술분야

- [1] 본 발명은 영상 부호화/복호화 방법, 장치 및 비트스트림을 저장한 기록 매체에 관한 것이다. 구체적으로, 본 발명은 루프내 필터링을 기반으로 하는 영상 부호화/복호화 방법 및 장치에 관한 것이다.

배경기술

- [2] 최근 HD(High Definition) 영상 및 UHD(Ultra High Definition) 영상과 같은 고해상도, 고품질의 영상에 대한 수요가 다양한 응용 분야에서 증가하고 있다. 영상 데이터가 고해상도, 고품질이 될수록 기존의 영상 데이터에 비해 상대적으로 데이터량이 증가하기 때문에 기존의 유무선 광대역 회선과 같은 매체를 이용하여 영상 데이터를 전송하거나 기존의 저장 매체를 이용해 저장하는 경우, 전송 비용과 저장 비용이 증가하게 된다. 영상 데이터가 고해상도, 고품질화 됨에 따라 발생하는 이러한 문제들을 해결하기 위해서는 더 높은 해상도 및 화질을 갖는 영상에 대한 고효율 영상 부호화(encoding)/복호화(decoding) 기술이 요구된다.
- [3] 영상 압축 기술로 현재 픽처의 이전 또는 이후 픽처로부터 현재 픽처에 포함된 화소값을 예측하는 화면 간 예측 기술, 현재 픽처 내의 화소 정보를 이용하여 현재 픽처에 포함된 화소값을 예측하는 화면내 예측 기술, 잔여 신호의 에너지를 압축하기 위한 변환 및 양자화 기술, 출현 빈도가 높은 값에 짧은 부호를 할당하고 출현 빈도가 낮은 값에 긴 부호를 할당하는 엔트로피 부호화 기술 등 다양한 기술이 존재하고 이러한 영상 압축 기술을 이용해 영상 데이터를 효과적으로 압축하여 전송 또는 저장할 수 있다.

발명의 상세한 설명

기술적 과제

- [4] 본 발명은 적응적 루프내 필터링을 이용한 영상 부호화/복호화 방법 및 장치를 제공하는 것을 목적으로 한다.
- [5] 본 발명은 성분 간 적응적 루프내 필터링을 이용한 영상 부호화/복호화 방법 및 장치를 제공하는 것을 목적으로 한다.
- [6] 또한, 본 발명은 본 발명의 영상 부호화/복호화 방법 또는 장치에 의해 생성된 비트스트림을 저장한 기록 매체를 제공하는 것을 목적으로 한다.

기술적 해결방법

- [7] 본 개시의 일 실시예에 따른 영상 복호화 방법은, 현재 블록의 복원 휘도 블록을 획득하는 단계, 상기 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component adaptive loop filter)의 적용 여부를 결정하는 단계, 상기 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우, 적응적 루프

필터(adaptive loop filter)가 적용된 상기 현재 블록의 복원 색차 블록을 획득하는 단계 및 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의 최종 복원 색차 블록을 획득하는 단계를 포함할 수 있다.

- [8] 상기 성분 간 적응적 루프 필터의 적용 여부를 결정하는 단계는, 상기 현재 블록에 대한 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보를 획득하는 단계를 포함하되, 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는 시퀀스 파라미터 세트(sequence parameter set)를 통해 시그널링될 수 있다.
- [9] 상기 영상 복호화 방법은, 상기 현재 블록에 대해 상기 성분 간 적응적 루프 필터의 적용이 가능한 경우, 상기 현재 블록의 Cb 성분에 대한 성분 간 적응적 루프 필터의 적용 여부를 지시하는 정보를 획득하는 단계를 더 포함할 수 있다.
- [10] 상기 Cb 성분에 대한 성분 간 루프 필터의 적용 여부를 지시하는 정보는, 픽처 헤더(picture header)를 통해 시그널링될 수 있다.
- [11] 상기 영상 복호화 방법은, 상기 Cb 성분에 대해 성분 간 루프 필터가 적용되는 경우, 상기 Cb 성분이 참조하는 적응 파라미터 세트 지시자를 획득하는 단계를 더 포함할 수 있다.
- [12] 상기 영상 복호화 방법은, 상기 현재 블록에 대해 상기 성분 간 적응적 루프 필터의 적용이 가능한 경우, 상기 현재 블록의 Cr 성분에 대한 성분 간 적응적 루프 필터의 적용 여부를 지시하는 정보를 획득하는 단계를 더 포함하는, 영상 복호화 방법.
- [13] 상기 Cr 성분에 대한 성분 간 루프 필터의 적용 여부를 지시하는 정보는, 픽처 헤더(picture header)를 통해 시그널링될 수 있다.
- [14] 상기 영상 복호화 방법은, 상기 Cr 성분에 대해 성분 간 루프 필터가 적용되는 경우, 상기 Cr 성분이 참조하는 적응 파라미터 세트 지시자를 획득하는 단계를 더 포함할 수 있다.
- [15] 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보가 시그널링되지 않는 경우, 상기 현재 블록에 대해서는 성분 간 적응적 루프 필터의 적용이 불가능한 것으로 결정될 수 있다.
- [16] 상기 영상 복호화 방법은, 상기 현재 블록의 복원 색차 블록을 획득하는 단계 및 상기 복원 색차 블록에 대해 적응적 루프 필터를 적용할지 여부를 결정하는 단계를 더 포함할 수 있다.
- [17] 상기 복원 색차 블록에 대해 적응적 루프 필터의 적용 여부를 결정하는 단계는, 상기 복원 색차 블록에 대한 상기 적응적 루프 필터의 적용 가능 여부를 지시하는 정보를 획득하는 단계를 포함하되, 상기 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는 픽처 헤더(picture header)를 통해 시그널링될 수 있다.
- [18] 상기 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는, Cb 성분 및 Cr 성분 각각에 대해 시그널링될 수 있다.

- [19] 상기 영상 복호화 방법은, 상기 복원 색차 블록에 적용되는 적응적 루프 필터의 종류를 지시하는 정보를 획득하는 단계를 더 포함하되, 상기 적응적 루프 필터의 종류는 선형적(linear) 적응적 루프 필터 및 비선형적(non-linear) 적응적 루프 필터를 포함할 수 있다.
- [20] 상기 복원 색차 블록에 적용되는 적응적 루프 필터의 종류를 지시하는 정보는 단일 구문 요소일 수 있다.
- [21] 상기 최종 복원 색차 블록은, 상기 복원 휘도 블록에 상기 성분 간 적응적 루프 필터를 적용하여 획득된 성분 간 루프 필터가 적용된 휘도 블록과 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 유도될 수 있다.
- [22] 본 개시의 일 실시예에 따른 영상 부호화 방법은, 현재 블록의 복원 휘도 블록을 획득하는 단계; 상기 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component adaptive loop filter)의 적용 여부를 결정하는 단계, 상기 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우, 적응적 루프 필터(adaptive loop filter)가 적용된 상기 현재 블록의 복원 색차 블록을 획득하는 단계 및 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의 최종 복원 색차 블록을 획득하는 단계를 포함할 수 있다.
- [23] 상기 영상 부호화 방법은, 상기 현재 블록에 대한 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보를 부호화하는 단계를 더 포함하되, 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는 시퀀스 파라미터 세트(sequence parameter set)를 통해 부호화될 수 있다.
- [24] 상기 최종 복원 색차 블록은, 상기 복원 휘도 블록에 상기 성분 간 적응적 루프 필터를 적용하여 획득된 성분 간 루프 필터가 적용된 휘도 블록과 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 유도될 수 있다.
- [25] 본 개시의 일 실시예에 따른 영상 복호화 장치에 수신되고 현재 픽처에 포함된 현재 블록을 복원하는데 이용되는 비트스트림을 저장한 컴퓨터 판독가능한 기록매체는, 현재 블록의 복원 휘도 블록을 획득하는 단계, 상기 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component adaptive loop filter)의 적용 여부를 결정하는 단계, 상기 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우, 적응적 루프 필터(adaptive loop filter)가 적용된 상기 현재 블록의 복원 색차 블록을 획득하는 단계 및 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의 최종 복원 색차 블록을 획득하는 단계를 포함하는 영상 부호화 방법에 의해 생성될 수 있다.
- [26] 상기 영상 부호화 방법은, 상기 현재 블록에 대한 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보를 부호화하는 단계를 더 포함하되, 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는 시퀀스 파라미터 세트(sequence parameter set)를 통해 부호화될 수 있다.

발명의 효과

- [27] 본 발명에 따르면, 적응적 루프내 필터링을 이용한 영상 부호화/복호화 방법 및 장치가 제공될 수 있다.
- [28] 본 발명에 따르면, 성분간 적응적 루프내 필터링을 이용한 영상 부호화/복호화 방법 및 장치가 제공될 수 있다.
- [29] 또한, 본 발명에 따르면, 영상 부호화기/복호화기의 계산 복잡도, 메모리 접근 대역폭을 감소시키기 위해 부표본 기반 블록 분류를 이용한 루프내 필터링 방법 및 장치가 제공될 수 있다.
- [30] 또한, 본 발명에 따르면, 영상 부호화기/복호화기의 계산 복잡도, 메모리 요구량, 메모리 접근 대역폭을 감소시키기 위해 다중 필터 형태를 이용한 루프내 필터링 방법 및 장치가 제공될 수 있다.
- [31] 또한, 본 발명에 따르면, 본 발명의 영상 부호화/복호화 방법 또는 장치에 의해 생성된 비트스트림을 저장한 기록 매체가 제공될 수 있다.
- [32] 또한, 본 발명에 따르면, 영상의 부호화 및/또는 복호화 효율을 향상시킬 수 있다.

도면의 간단한 설명

- [33] 도 1은 본 발명이 적용되는 부호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.
- [34] 도 2는 본 발명이 적용되는 복호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.
- [35] 도 3은 영상을 부호화 및 복호화할 때의 영상의 분할 구조를 개략적으로 나타내는 도면이다.
- [36] 도 4는 화면 내 예측 과정의 실시예를 설명하기 위한 도면이다.
- [37] 도 5는 화면 간 예측 과정의 실시예를 설명하기 위한 도면이다.
- [38] 도 6은 변환 및 양자화의 과정을 설명하기 위한 도면이다.
- [39] 도 7은 화면 내 예측에 이용 가능한 참조 샘플들을 설명하기 위한 도면이다.
- [40] 도 8a 및 도 8b는 본 발명의 일 실시예에 따른 영상 복호화 방법 및 부호화 방법을 나타내는 흐름도이다.
- [41] 도 9는 부표본에 기반하여 가로, 세로, 제1 대각, 제2 대각 방향에 따른 기울기 값을 결정하는 일 예이다.
- [42] 도 10 내지 도 12는 본 발명의 일 실시예에 따른 특정 샘플 위치에서 가로, 세로, 제1 대각, 제2 대각 방향에 따른 기울기 값을 결정하는 일 예이다.
- [43] 도 13은 본 발명의 일 실시예에 따른 마름모 형태의 필터를 나타내는 도면이다.
- [44] 도 14는 본 발명의 일 실시예에 따른 필터 탭 수가 5x5인 필터를 나타내는 도면이다.
- [45] 도 15a 및 도 15b는 본 발명의 일 실시예에 따른 다양한 필터 형태를 나타내는 도면이다.

- [46] 도 16은 적응적 루프내 필터링에 필요한 시퀀스 파라미터 세트 선택의 일 실시 예를 나타낸다.
- [47] 도 17은 적응적 루프내 필터링에 필요한 적응 파라미터 세트 선택의 일 실시 예를 나타낸다.
- [48] 도 18은 적응 파라미터 세트의 적응 파라미터 타입의 일 실시 예를 나타낸다.
- [49] 도 19는 적응적 루프내 필터링에 필요한 슬라이스 헤더 선택의 일 실시 예를 나타낸다.
- [50] 도 20a 내지 도 20d는 적응적 루프내 필터링에 필요한 적응적 루프내 필터 데이터 선택의 일 실시 예를 나타낸다.
- [51] 도 21은 적응적 루프내 필터링에 필요한 부호화 트리 블록 선택의 일 실시 예를 나타낸다.
- [52] 도 22a 및 도 22b는 고정된 필터 계수의 일 실시 예를 나타낸다.
- [53] 도 23은 적응적 루프내 필터 계수 클래스와 필터의 매핑 관계의 일 실시 예를 나타낸다.
- [54] 도 24는 적응적 루프내 필터 데이터 선택의 절삭 색인에 대한 일 실시 예를 나타낸다.
- [55] 도 25는 도 24의 적응적 루프내 필터 데이터 선택의 절삭 색인에 대한 자세한 설명을 나타낸다.
- [56] 도 26은 적응적 루프내 필터 데이터 선택의 절삭 색인에 대한 또 다른 일 실시 예를 나타낸다.
- [57] 도 27은 도 26의 적응적 루프내 필터 데이터 선택의 절삭 색인에 대한 자세한 설명을 나타낸다.
- [58] 도 28은 적응적 루프내 필터링에 필요한 슬라이스 헤더 선택의 또 다른 일 실시 예를 나타낸다.
- [59] 도 29a 및 도 29b는 적응적 루프내 필터링에 필요한 적응적 루프내 필터 데이터 선택의 또 다른 일 실시 예를 나타낸다.
- [60] 도 30a 및 도 30b는 적응적 루프내 필터링에 필요한 부호화 트리 블록 선택의 또 다른 일 실시 예를 나타낸다.
- [61] 도 31은 도 28의 슬라이스 헤더 선택에 대한 자세한 설명을 나타낸다.
- [62] 도 32 및 33은 도 29a 및 도 29b의 적응적 루프내 필터 데이터 선택에 대한 자세한 설명을 나타낸다.
- [63] 도 34 및 35는 도 29a 및 도 29b의 적응적 루프내 필터 데이터 선택에 대한 자세한 설명을 나타낸다.
- [64] 도 36 및 도 37은 도 31의 부호화 트리 블록 선택의 자세한 설명을 나타낸다.
- [65] 도 38은 일 실시 예에 따른 적응적 루프내 필터를 이용한 비디오 복호화 방법을 도시한다.
- [66] 도 39는 일 실시 예에 따른 적응적 루프내 필터를 이용한 비디오 부호화 방법을 도시한다.

- [67] 도 40은 본 개시의 또 다른 일 실시예에 따른 영상 복호화 방법을 설명하기 위한 도면이다.
- [68] 도 41은 본 개시의 또 다른 일 실시예에 따른 영상 부호화 방법을 설명하기 위한 도면이다.
- [69] 도 42는 본 개시의 또 다른 일 실시예에 따른 시퀀스 파라미터 세트(sequence parameter set)를 설명하기 위한 도면이다.
- [70] 도 43은 본 개시의 또 다른 일 실시예에 따른 적응 파라미터 세트(adaptation parameter set)를 설명하기 위한 도면이다.
- [71] 도 44는 본 개시의 또 다른 일 실시예에 따른 적응 파라미터를 설명하기 위한 도면이다.
- [72] 도 45는 본 개시의 또 다른 일 실시예에 따른 픽처 헤더(picture header)를 설명하기 위한 도면이다.
- [73] 도 46은 본 개시의 또 다른 일 실시예에 따른 슬라이스 헤더(slice header)를 설명하기 위한 도면이다.
- [74] 도 47은 본 개시의 또 다른 일 실시예에 따른 ALF 데이터(data)를 설명하기 위한 도면이다.
- [75] 도 48은 본 개시의 또 다른 일 실시예에 따른 CTU(Coding Tree Unit)을 설명하기 위한 도면이다.
- [76] 도 49 및 도 50은 본 개시의 또 다른 일 실시예에 따른 ALF 데이터를 설명하기 위한 다른 도면이다.
- [77] 도 51 및 도 52는 본 개시의 또 다른 일 실시예에 따른 ALF 데이터를 설명하기 위한 또 다른 도면이다.

발명의 실시를 위한 최선의 형태

- [78] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세한 설명에 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다. 도면에서의 요소들의 형상 및 크기 등은 보다 명확한 설명을 위해 과장될 수 있다. 후술하는 예시적 실시예들에 대한 상세한 설명은, 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 실시예를 실시할 수 있기에 충분하도록 상세히 설명된다. 다양한 실시예들은 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예에 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 실시예의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이

이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 예시적 실시예들의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다.

[79] 본 발명에서 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[80] 본 발명의 어떤 구성 요소가 다른 구성 요소에 “연결되어” 있다거나 “접속되어” 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있으나, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 “직접 연결되어” 있다거나 “직접 접속되어” 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

[81] 본 발명의 실시예에 나타나는 구성부들은 서로 다른 특징적인 기능들을 나타내기 위해 독립적으로 도시되는 것으로, 각 구성부들이 분리된 하드웨어나 하나의 소프트웨어 구성단위로 이루어짐을 의미하지 않는다. 즉, 각 구성부는 설명의 편의상 각각의 구성부로 나열하여 포함한 것으로 각 구성부 중 적어도 두 개의 구성부가 합쳐져 하나의 구성부로 이루어지거나, 하나의 구성부가 복수 개의 구성부로 나뉘어져 기능을 수행할 수 있고 이러한 각 구성부의 통합된 실시예 및 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.

[82] 본 발명에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 발명에서, “포함하다” 또는 “가지다” 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다. 즉, 본 발명에서 특정 구성을 “포함”한다고 기술하는 내용은 해당 구성 이외의 구성을 배제하는 것이 아니며, 추가적인 구성이 본 발명의 실시 또는 본 발명의 기술적 사상의 범위에 포함될 수 있음을 의미한다.

[83] 본 발명의 일부의 구성 요소는 본 발명에서 본질적인 기능을 수행하는 필수적인 구성 요소는 아니고 단지 성능을 향상시키기 위한 선택적 구성 요소일 수 있다. 본 발명은 단지 성능 향상을 위해 사용되는 구성 요소를 제외한 본 발명의 본질을 구현하는데 필수적인 구성부만을 포함하여 구현될 수 있고, 단지

성능 향상을 위해 사용되는 선택적 구성 요소를 제외한 필수 구성 요소만을 포함한 구조도 본 발명의 권리범위에 포함된다.

- [84] 이하, 도면을 참조하여 본 발명의 실시 형태에 대하여 구체적으로 설명한다. 본 명세서의 실시예를 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 명세서의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략하고, 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.
- [85] 이하에서 영상은 동영상(video)을 구성하는 하나의 픽처(picture)를 의미할 수 있으며, 동영상 자체를 나타낼 수도 있다. 예를 들면, "영상의 부호화 및/또는 복호화"는 "동영상의 부호화 및/또는 복호화"를 의미할 수 있으며, "동영상을 구성하는 영상들 중 하나의 영상의 부호화 및/또는 복호화"를 의미할 수도 있다.
- [86] 이하에서, 용어들 "동영상" 및 "비디오"는 동일한 의미로 사용될 수 있으며, 서로 교체되어 사용될 수 있다.
- [87] 이하에서, 대상 영상은 부호화의 대상인 부호화 대상 영상 및/또는 복호화의 대상인 복호화 대상 영상일 수 있다. 또한, 대상 영상은 부호화 장치로 입력된 입력 영상일 수 있고, 복호화 장치로 입력된 입력 영상일 수 있다. 여기서, 대상 영상은 현재 영상과 동일한 의미를 가질 수 있다.
- [88] 이하에서, 용어들 "영상", "픽처", "프레임(frame)" 및 "스크린(screen)"은 동일한 의미로 사용될 수 있으며, 서로 교체되어 사용될 수 있다.
- [89] 이하에서, 대상 블록은 부호화의 대상인 부호화 대상 블록 및/또는 복호화의 대상인 복호화 대상 블록일 수 있다. 또한, 대상 블록은 현재 부호화 및/또는 복호화의 대상인 현재 블록일 수 있다. 예를 들면, 용어들 "대상 블록" 및 "현재 블록"은 동일한 의미로 사용될 수 있으며, 서로 교체되어 사용될 수 있다.
- [90] 이하에서, 용어들 "블록" 및 "유닛"은 동일한 의미로 사용될 수 있으며, 서로 교체되어 사용될 수 있다. 또는 "블록"은 특정한 유닛을 나타낼 수 있다.
- [91] 이하에서, 용어들 "영역(region)" 및 "세그먼트(segment)"는 서로 교체되어 사용될 수 있다.
- [92] 이하에서, 특정한 신호는 특정한 블록을 나타내는 신호일 수 있다. 예를 들면, 원(original) 신호는 대상 블록을 나타내는 신호일 수 있다. 예측(prediction) 신호는 예측 블록을 나타내는 신호일 수 있다. 잔여(residual) 신호는 잔여 블록(residual block)을 나타내는 신호일 수 있다.
- [93] 실시예들에서, 특정된 정보, 데이터, 플래그(flag), 색인(index) 및 요소(element), 속성(attribute) 등의 각각은 값을 가질 수 있다. 정보, 데이터, 플래그, 색인 및 요소, 속성 등의 값 "0"은 논리 거짓(logical false) 또는 제1 기정의된(predefined) 값을 나타낼 수 있다. 말하자면, 값 "0", 거짓, 논리 거짓 및 제1 기정의된 값은 서로 대체되어 사용될 수 있다. 정보, 데이터, 플래그, 색인 및 요소, 속성 등의 값 "1"은 논리 참(logical true) 또는 제2 기정의된 값을 나타낼 수 있다. 말하자면, 값 "1", 참, 논리 참 및 제2 기정의된 값은 서로 대체되어 사용될 수 있다.

- [94] 행, 열 또는 색인(index)을 나타내기 위해 i 또는 j 등의 변수가 사용될 때, i 의 값은 0 이상의 정수일 수 있으며, 1 이상의 정수일 수도 있다. 말하자면, 실시예들에서 행, 열 및 색인 등은 0에서부터 카운트될 수 있으며, 1에서부터 카운트될 수 있다.
- [95]
- [96] 용어 설명
- [97] 부호화기(Encoder): 부호화(Encoding)를 수행하는 장치를 의미한다. 즉, 부호화 장치를 의미할 수 있다.
- [98] 복호화기(Decoder): 복호화(Decoding)를 수행하는 장치를 의미한다. 즉, 복호화 장치를 의미할 수 있다.
- [99] 블록(Block): 샘플(Sample)의 $M \times N$ 배열이다. 여기서 M 과 N 은 양의 정수 값을 의미할 수 있으며, 블록은 흔히 2차원 형태의 샘플 배열을 의미할 수 있다. 블록은 유닛을 의미할 수 있다. 현재 블록은 부호화 시 부호화의 대상이 되는 부호화 대상 블록, 복호화 시 복호화의 대상이 되는 복호화 대상 블록을 의미할 수 있다. 또한, 현재 블록은 부호화 블록, 예측 블록, 잔여 블록, 변환 블록 중 적어도 하나일 수 있다.
- [100] 샘플(Sample): 블록을 구성하는 기본 단위이다. 비트 깊이 (bit depth, B_d)에 따라 0부터 $2^{B_d} - 1$ 까지의 값으로 표현될 수 있다. 본 발명에서 샘플은 화소 또는 픽셀과 같은 의미로 사용될 수 있다. 즉, 샘플, 화소, 픽셀은 서로 같은 의미를 가질 수 있다.
- [101] 유닛(Unit): 영상 부호화 및 복호화의 단위를 의미할 수 있다. 영상의 부호화 및 복호화에 있어서, 유닛은 하나의 영상을 분할한 영역일 수 있다. 또한, 유닛은 하나의 영상을 세분화된 유닛으로 분할하여 부호화 또는 복호화 할 때 그 분할된 단위를 의미할 수 있다. 즉, 하나의 영상은 복수의 유닛들로 분할될 수 있다. 영상의 부호화 및 복호화에 있어서, 유닛 별로 기정의된 처리가 수행될 수 있다. 하나의 유닛은 유닛에 비해 더 작은 크기를 갖는 하위 유닛으로 더 분할될 수 있다. 기능에 따라서, 유닛은 블록(Block), 매크로블록(Macroblock), 부호화 트리 유닛(Coding Tree Unit), 부호화 트리 블록(Coding Tree Block), 부호화 유닛(Coding Unit), 부호화 블록(Coding Block), 예측 유닛(Prediction Unit), 예측 블록(Prediction Block), 잔여 유닛(Residual Unit), 잔여 블록(Residual Block), 변환 유닛(Transform Unit), 변환 블록(Transform Block) 등을 의미할 수 있다. 또한, 유닛은 블록과 구분하여 지칭하기 위해 휘도(Luma) 성분 블록과 그에 대응하는 색차(Chroma) 성분 블록 그리고 각 블록에 대한 구문 요소를 포함한 것을 의미할 수 있다. 유닛은 다양한 크기와 형태를 가질 수 있으며, 특히 유닛의 형태는 정사각형뿐만 아니라 직사각형, 사다리꼴, 삼각형, 오각형 등 2차원으로 표현될 수 있는 기하학적 도형을 포함할 수 있다. 또한, 유닛 정보는 부호화 유닛, 예측 유닛, 잔여 유닛, 변환 유닛 등을 가리키는 유닛의 타입, 유닛의 크기, 유닛의 깊이, 유닛의 부호화 및 복호화 순서 등 중 적어도 하나 이상을 포함할 수 있다.

- [102] 부호화 트리 유닛(Coding Tree Unit): 하나의 휘도 성분(Y) 부호화 트리 블록과 관련된 두 색차 성분(Cb, Cr) 부호화 트리 블록들로 구성된다. 또한, 상기 블록들과 각 블록에 대한 구문 요소를 포함한 것을 의미할 수도 있다. 각 부호화 트리 유닛은 부호화 유닛, 예측 유닛, 변환 유닛 등의 하위 유닛을 구성하기 위하여 쿼드트리(quad tree), 이진트리(binary tree), 3분할트리(ternary tree) 등 하나 이상의 분할 방식을 이용하여 분할될 수 있다. 입력 영상의 분할처럼 영상의 복/부호화 과정에서 처리 단위가 되는 샘플 블록을 지칭하기 위한 용어로 사용될 수 있다. 여기서, 쿼드트리는 4분할트리(quarterternary tree)를 의미할 수 있다.
- [103] 부호화 블록의 크기가 소정의 범위 내에 속하는 경우에는 쿼드트리로만 분할이 가능할 수 있다. 여기서, 소정의 범위는 쿼드트리만으로 분할이 가능한 부호화 블록의 최대 크기 및 최소 크기 중 적어도 하나로 정의될 수 있다. 쿼드트리 형태의 분할이 허용되는 부호화 블록의 최대/최소 크기를 나타내는 정보는 비트스트림을 통해 시그널링될 수 있고, 해당 정보는 시퀀스, 픽처 파라미터, 타일 그룹, 또는 슬라이스(세그먼트) 중 적어도 하나의 단위로 시그널링될 수 있다. 또는, 부호화 블록의 최대/최소 크기는 부호화기/복호화기에 기-설정된 고정된 크기일 수도 있다. 예를 들어, 부호화 블록의 크기가 256x256 내지 64x64에 해당하는 경우에는 쿼드트리로만 분할이 가능할 수 있다. 또는 부호화 블록의 크기가 최대 변환 블록의 크기 보다 큰 경우에는 쿼드트리로만 분할이 가능할 수 있다. 이때, 상기 분할되는 블록은 부호화 블록 또는 변환 블록 중 적어도 하나일 수 있다. 이러한 경우에 부호화 블록의 분할을 나타내는 정보(예컨대, split_flag)는 쿼드트리 분할 여부를 나타내는 플래그일 수 있다. 부호화 블록의 크기가 소정의 범위 내에 속하는 경우에는 이진트리 또는 3분할트리로만 분할이 가능할 수 있다. 이 경우, 쿼드트리에 관한 상기 설명은 이진트리 또는 3분할트리에 대해서도 동일하게 적용될 수 있다.
- [104] 부호화 트리 블록(Coding Tree Block): Y 부호화 트리 블록, Cb 부호화 트리 블록, Cr 부호화 트리 블록 중 어느 하나를 지칭하기 위한 용어로 사용될 수 있다.
- [105] 주변 블록(Neighbor block): 현재 블록에 인접한 블록을 의미할 수 있다. 현재 블록에 인접한 블록은 현재 블록에 경계가 맞닿은 블록 또는 현재 블록으로부터 소정의 거리 내에 위치한 블록을 의미할 수 있다. 주변 블록은 현재 블록의 꼭지점에 인접한 블록을 의미할 수 있다. 여기에서, 현재 블록의 꼭지점에 인접한 블록이란, 현재 블록에 가로로 인접한 이웃 블록에 세로로 인접한 블록 또는 현재 블록에 세로로 인접한 이웃 블록에 가로로 인접한 블록일 수 있다. 주변 블록은 복원된 주변 블록을 의미할 수도 있다.
- [106] 복원된 주변 블록(Reconstructed Neighbor Block): 현재 블록 주변에 공간적(Spatial)/시간적(Temporal)으로 이미 부호화 또는 복호화된 주변 블록을 의미할 수 있다. 이때, 복원된 주변 블록은 복원된 주변 유닛을 의미할 수 있다. 복원된 공간적 주변 블록은 현재 픽처 내의 블록이면서 부호화 및/또는 복호화를 통해 이미 복원된 블록일 수 있다. 복원된 시간적 주변 블록은 참조 영상 내에서

현재 픽처의 현재 블록과 대응하는 위치의 복원된 블록 또는 그 주변 블록일 수 있다.

- [107] 유닛 깊이(Depth): 유닛이 분할된 정도를 의미할 수 있다. 트리 구조(Tree Structure)에서 가장 상위 노드(Root Node)는 분할되지 않은 최초의 유닛에 대응할 수 있다. 가장 상위 노드는 루트 노드로 칭해질 수 있다. 또한, 가장 상위 노드는 최소의 깊이 값을 가질 수 있다. 이 때, 가장 상위 노드는 레벨(Level) 0의 깊이를 가질 수 있다. 레벨 1의 깊이를 갖는 노드는 최초의 유닛이 한 번 분할됨에 따라 생성된 유닛을 나타낼 수 있다. 레벨 2의 깊이를 갖는 노드는 최초의 유닛이 두 번 분할됨에 따라 생성된 유닛을 나타낼 수 있다. 레벨 n의 깊이를 갖는 노드는 최초의 유닛이 n번 분할됨에 따라 생성된 유닛을 나타낼 수 있다. 리프 노드(Leaf Node)는 가장 하위의 노드일 수 있으며, 더 분할될 수 없는 노드일 수 있다. 리프 노드의 깊이는 최대 레벨일 수 있다. 예를 들면, 최대 레벨의 기정의된 값은 3일 수 있다. 루트 노드는 깊이가 가장 얇고, 리프 노드는 깊이가 가장 깊다고 할 수 있다. 또한, 유닛을 트리 구조로 표현했을 때 유닛이 존재하는 레벨이 유닛 깊이를 의미할 수 있다.
- [108] 비트스트림(Bitstream): 부호화된 영상 정보를 포함하는 비트의 열을 의미할 수 있다.
- [109] 파라미터 세트(Parameter Set): 비트스트림 내의 구조 중 헤더(header) 정보에 해당한다. 비디오 파라미터 세트(video parameter set), 시퀀스 파라미터 세트(sequence parameter set), 픽처 파라미터 세트(picture parameter set), 적응 파라미터 세트(adaptation parameter set) 중 적어도 하나가 파라미터 세트에 포함될 수 있다. 또한, 파라미터 세트는 타일 그룹(tile group) 헤더, 슬라이스(slice) 헤더 및 타일(tile) 헤더 정보를 포함할 수도 있다. 또한, 상기 타일 그룹은 여러 타일을 포함하는 그룹을 의미할 수 있으며, 슬라이스와 동일한 의미일 수 있다.
- [110] 파싱(Parsing): 비트스트림을 엔트로피 복호화하여 구문 요소(Syntax Element)의 값을 결정하는 것을 의미하거나, 엔트로피 복호화 자체를 의미할 수 있다.
- [111] 심볼(Symbol): 부호화/복호화 대상 유닛의 구문 요소, 부호화 파라미터(coding parameter), 변환 계수(Transform Coefficient)의 값 등 중 적어도 하나를 의미할 수 있다. 또한, 심볼은 엔트로피 부호화의 대상 또는 엔트로피 복호화의 결과를 의미할 수 있다.
- [112] 예측 모드(Prediction Mode): 화면 내 예측으로 부호화/복호화되는 모드 또는 화면 간 예측으로 부호화/복호화되는 모드를 지시하는 정보일 수 있다.
- [113] 예측 유닛(Prediction Unit): 화면 간 예측, 화면 내 예측, 화면 간 보상, 화면 내 보상, 움직임 보상 등 예측을 수행할 때의 기본 단위를 의미할 수 있다. 하나의 예측 유닛은 더 작은 크기를 가지는 복수의 파티션(Partition) 또는 복수의 하위 예측 유닛들로 분할 될 수도 있다. 복수의 파티션들 또한 예측 또는 보상의 수행에 있어서의 기본 단위일 수 있다. 예측 유닛의 분할에 의해 생성된 파티션

또한 예측 유닛일 수 있다.

[114] 예측 유닛 파티션(Prediction Unit Partition): 예측 유닛이 분할된 형태를 의미할 수 있다.

[115] 참조 영상 리스트(Reference Picture List): 화면 간 예측 또는 움직임 보상에 사용되는 하나 이상의 참조 영상들을 포함하는 리스트를 의미할 수 있다. 참조 영상 리스트의 종류는 LC (List Combined), L0 (List 0), L1 (List 1), L2 (List 2), L3 (List 3) 등이 있을 수 있으며, 화면 간 예측에는 1개 이상의 참조 영상 리스트들이 사용될 수 있다.

[116] 화면 간 예측 지시자(Inter Prediction Indicator): 현재 블록의 화면 간 예측 방향(단방향 예측, 쌍방향 예측 등)을 의미할 수 있다. 또는, 현재 블록의 예측 블록을 생성할 때 사용되는 참조 영상의 개수를 의미할 수 있다. 또는, 현재 블록에 대해 화면 간 예측 또는 움직임 보상을 수행할 때 사용되는 예측 블록의 개수를 의미할 수 있다.

[117] 예측 리스트 활용 플래그(prediction list utilization flag): 특정 참조 영상 리스트 내 적어도 하나의 참조 영상을 이용하여 예측 블록을 생성하는지 여부를 나타낸다. 예측 리스트 활용 플래그를 이용하여 화면 간 예측 지시자를 도출할 수 있고, 반대로 화면 간 예측 지시자를 이용하여 예측 리스트 활용 플래그를 도출할 수 있다. 예를 들어, 예측 리스트 활용 플래그가 제1 값인 0을 지시하는 경우, 해당 참조 영상 리스트 내 참조 영상을 이용하여 예측 블록을 생성하지 않는 것을 나타낼 수 있고, 제2 값인 1을 지시하는 경우, 해당 참조 영상 리스트를 이용하여 예측 블록을 생성할 수 있는 것을 나타낼 수 있다.

[118] 참조 영상 색인(Reference Picture Index): 참조 영상 리스트에서 특정 참조 영상을 지시하는 색인을 의미할 수 있다.

[119] 참조 영상(Reference Picture): 화면 간 예측 또는 움직임 보상을 위해서 특정 블록이 참조하는 영상을 의미할 수 있다. 또는, 참조 영상은 화면 간 예측 또는 움직임 보상을 위해 현재 블록이 참조하는 참조 블록을 포함하는 영상일 수 있다. 이하, 용어 "참조 픽처" 및 "참조 영상"은 동일한 의미로 사용될 수 있으며, 서로 교체되어 사용될 수 있다.

[120] 움직임 벡터(Motion Vector): 화면 간 예측 또는 움직임 보상에 사용되는 2차원 벡터일 수 있다. 움직임 벡터는 부호화/복호화 대상 블록과 참조 블록 사이의 오프셋을 의미할 수 있다. 예를 들어, (mvX, mvY)는 움직임 벡터를 나타낼 수 있다. mvX는 수평(horizontal) 성분, mvY는 수직(vertical) 성분을 나타낼 수 있다.

[121] 탐색 영역(Search Range): 탐색 영역은 화면 간 예측 중 움직임 벡터에 대한 탐색이 이루어지는 2차원의 영역일 수 있다. 예를 들면, 탐색 영역의 크기는 MxN일 수 있다. M 및 N은 각각 양의 정수일 수 있다.

[122] 움직임 벡터 후보(Motion Vector Candidate): 움직임 벡터를 예측할 때 예측 후보가 되는 블록 또는 그 블록의 움직임 벡터를 의미할 수 있다. 또한, 움직임 벡터 후보는 움직임 벡터 후보 리스트에 포함될 수 있다.

- [123] 움직임 벡터 후보 리스트(Motion Vector Candidate List): 하나 이상의 움직임 벡터 후보들을 이용하여 구성된 리스트를 의미할 수 있다.
- [124] 움직임 벡터 후보 색인(Motion Vector Candidate Index): 움직임 벡터 후보 리스트 내의 움직임 벡터 후보를 가리키는 지시자를 의미할 수 있다. 움직임 벡터 예측기(Motion Vector Predictor)의 색인(index)일 수 있다.
- [125] 움직임 정보(Motion Information): 움직임 벡터, 참조 영상 색인, 화면 간 예측 지시자 뿐만 아니라 예측 리스트 활용 플래그, 참조 영상 리스트 정보, 참조 영상, 움직임 벡터 후보, 움직임 벡터 후보 색인, 머지 후보, 머지 색인 등 중 적어도 하나를 포함하는 정보를 의미할 수 있다.
- [126] 머지 후보 리스트(Merge Candidate List): 하나 이상의 머지 후보들을 이용하여 구성된 리스트를 의미할 수 있다.
- [127] 머지 후보(Merge Candidate): 공간적 머지 후보, 시간적 머지 후보, 조합된 머지 후보, 조합 양예측 머지 후보, 제로 머지 후보 등을 의미할 수 있다. 머지 후보는 화면 간 예측 지시자, 각 리스트에 대한 참조 영상 색인, 움직임 벡터, 예측 리스트 활용 플래그, 화면 간 예측 지시자 등의 움직임 정보를 포함할 수 있다.
- [128] 머지 색인(Merge Index): 머지 후보 리스트 내 머지 후보를 가리키는 지시자를 의미할 수 있다. 또한, 머지 색인은 공간적/시간적으로 현재 블록과 인접하게 복원된 블록들 중 머지 후보를 유도한 블록을 지시할 수 있다. 또한, 머지 색인은 머지 후보가 가지는 움직임 정보 중 적어도 하나를 지시할 수 있다.
- [129] 변환 유닛(Transform Unit): 변환, 역변환, 양자화, 역양자화, 변환 계수 부호화/복호화와 같이 잔여 신호(residual signal) 부호화/복호화를 수행할 때의 기본 단위를 의미할 수 있다. 하나의 변환 유닛은 분할되어 더 작은 크기를 가지는 복수의 하위 변환 유닛들로 분할될 수 있다. 여기서, 변환/역변환은 1차 변환/역변환 및 2차 변환/역변환 중 적어도 하나 이상을 포함할 수 있다.
- [130] 스케일링(Scaling): 양자화된 레벨에 인수를 곱하는 과정을 의미할 수 있다. 양자화된 레벨에 대한 스케일링의 결과로 변환 계수를 생성할 수 있다. 스케일링을 역양자화(dequantization)라고도 부를 수 있다.
- [131] 양자화 매개변수(Quantization Parameter): 양자화에서 변환 계수를 이용하여 양자화된 레벨(quantized level)을 생성할 때 사용하는 값을 의미할 수 있다. 또는, 역양자화에서 양자화된 레벨을 스케일링하여 변환 계수를 생성할 때 사용하는 값을 의미할 수도 있다. 양자화 매개변수는 양자화 스텝 크기(step size)에 매핑된 값일 수 있다.
- [132] 잔여 양자화 매개변수(Delta Quantization Parameter): 예측된 양자화 매개변수와 부호화/복호화 대상 유닛의 양자화 매개변수의 차분(difference) 값을 의미할 수 있다.
- [133] 스캔(Scan): 유닛, 블록 또는 행렬 내 계수의 순서를 정렬하는 방법을 의미할 수 있다. 예를 들어, 2차원 배열을 1차원 배열 형태로 정렬하는 것을 스캔이라고 한다. 또는, 1차원 배열을 2차원 배열 형태로 정렬하는 것도 스캔 또는 역

스캔(Inverse Scan)이라고 부를 수 있다.

- [134] 변환 계수(Transform Coefficient): 부호화기에서 변환을 수행하고 나서 생성된 계수 값을 의미할 수 있다. 또는, 복호화기에서 엔트로피 복호화 및 역양자화 중 적어도 하나를 수행하고 나서 생성된 계수 값을 의미할 수도 있다. 변환 계수 또는 잔여 신호에 양자화를 적용한 양자화된 레벨 또는 양자화된 변환 계수 레벨도 변환 계수의 의미에 포함될 수 있다.
- [135] 양자화된 레벨(Quantized Level): 부호화기에서 변환 계수 또는 잔여 신호에 양자화를 수행하여 생성된 값을 의미할 수 있다. 또는, 복호화기에서 역양자화를 수행하기 전 역양자화의 대상이 되는 값을 의미할 수도 있다. 유사하게, 변환 및 양자화의 결과인 양자화된 변환 계수 레벨도 양자화된 레벨의 의미에 포함될 수 있다.
- [136] нену로 변환 계수(Non-zero Transform Coefficient): 값의 크기가 0이 아닌 변환 계수 또는 값의 크기가 0이 아닌 변환 계수 레벨 또는 양자화된 레벨을 의미할 수 있다.
- [137] 양자화 행렬(Quantization Matrix): 영상의 주관적 화질 또는 객관적 화질을 향상시키기 위해서 양자화 또는 역양자화 과정에서 이용하는 행렬을 의미할 수 있다. 양자화 행렬을 스케일링 리스트(scaling list)라고도 부를 수 있다.
- [138] 양자화 행렬 계수(Quantization Matrix Coefficient): 양자화 행렬 내의 각 원소(element)를 의미할 수 있다. 양자화 행렬 계수를 행렬 계수(matrix coefficient)라고도 할 수 있다.
- [139] 기본 행렬(Default Matrix): 부호화기와 복호화기에서 미리 정의되어 있는 소정의 양자화 행렬을 의미할 수 있다.
- [140] 비 기본 행렬(Non-default Matrix): 부호화기와 복호화기에서 미리 정의되지 않고, 사용자에게 의해서 시그널링되는 양자화 행렬을 의미할 수 있다.
- [141] 통계값(statistic value): 연산 가능한 특정 값들을 가지는 변수, 부호화 파라미터, 상수 등 적어도 하나에 대한 통계값은 해당 특정 값들의 평균값, 합값, 가중평균값, 가중합값, 최소값, 최대값, 최빈값, 중간값, 보간값 중 적어도 하나 이상일 수 있다.
- [142] 도 1은 본 발명이 적용되는 부호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.
- [143] 부호화 장치(100)는 인코더, 비디오 부호화 장치 또는 영상 부호화 장치일 수 있다. 비디오는 하나 이상의 영상들을 포함할 수 있다. 부호화 장치(100)는 하나 이상의 영상들을 순차적으로 부호화할 수 있다.
- [144] 도 1을 참조하면, 부호화 장치(100)는 움직임 예측부(111), 움직임 보상부(112), 인트라 예측부(120), 스위치(115), 감산기(125), 변환부(130), 양자화부(140), 엔트로피 부호화부(150), 역양자화부(160), 역변환부(170), 가산기(175), 필터부(180) 및 참조 픽처 버퍼(190)를 포함할 수 있다.
- [145] 부호화 장치(100)는 입력 영상에 대해 인트라 모드 및/또는 인터 모드로

부호화를 수행할 수 있다. 또한, 부호화 장치(100)는 입력 영상에 대한 부호화를 통해 부호화된 정보를 포함하는 비트스트림을 생성할 수 있고, 생성된 비트스트림을 출력할 수 있다. 생성된 비트스트림은 컴퓨터 판독가능한 기록 매체에 저장될 수 있거나, 유/무선 전송 매체를 통해 스트리밍될 수 있다. 예측 모드로 인트라 모드가 사용되는 경우 스위치(115)는 인트라로 전환될 수 있고, 예측 모드로 인터 모드가 사용되는 경우 스위치(115)는 인터로 전환될 수 있다. 여기서 인트라 모드는 화면 내 예측 모드를 의미할 수 있으며, 인터 모드는 화면 간 예측 모드를 의미할 수 있다. 부호화 장치(100)는 입력 영상의 입력 블록에 대한 예측 블록을 생성할 수 있다. 또한, 부호화 장치(100)는 예측 블록이 생성된 후, 입력 블록 및 예측 블록의 차분(residual)을 사용하여 잔여 블록을 부호화할 수 있다. 입력 영상은 현재 부호화의 대상인 현재 영상으로 칭해질 수 있다. 입력 블록은 현재 부호화의 대상인 현재 블록 또는 부호화 대상 블록으로 칭해질 수 있다.

- [146] 예측 모드가 인트라 모드인 경우, 인트라 예측부(120)는 현재 블록의 주변에 이미 부호화/복호화된 블록의 샘플을 참조 샘플로서 이용할 수 있다. 인트라 예측부(120)는 참조 샘플을 이용하여 현재 블록에 대한 공간적 예측을 수행할 수 있고, 공간적 예측을 통해 입력 블록에 대한 예측 샘플들을 생성할 수 있다. 여기서 인트라 예측은 화면 내 예측을 의미할 수 있다.
- [147] 예측 모드가 인터 모드인 경우, 움직임 예측부(111)는, 움직임 예측 과정에서 참조 영상으로부터 입력 블록과 가장 매치가 잘 되는 영역을 검색할 수 있고, 검색된 영역을 이용하여 움직임 벡터를 도출할 수 있다. 이때, 상기 영역으로 탐색 영역을 사용할 수 있다. 참조 영상은 참조 픽처 버퍼(190)에 저장될 수 있다. 여기서, 참조 영상에 대한 부호화/복호화가 처리되었을 때 참조 픽처 버퍼(190)에 저장될 수 있다.
- [148] 움직임 보상부(112)는 움직임 벡터를 이용하는 움직임 보상을 수행함으로써 현재 블록에 대한 예측 블록을 생성할 수 있다. 여기서 인터 예측은 화면 간 예측 또는 움직임 보상을 의미할 수 있다.
- [149] 상기 움직임 예측부(111)와 움직임 보상부(112)는 움직임 벡터의 값이 정수 값을 가지지 않을 경우에 참조 영상 내의 일부 영역에 대해 보간 필터(Interpolation Filter)를 적용하여 예측 블록을 생성할 수 있다. 화면 간 예측 또는 움직임 보상을 수행하기 위해 부호화 유닛을 기준으로 해당 부호화 유닛에 포함된 예측 유닛의 움직임 예측 및 움직임 보상 방법이 스킵 모드(Skip Mode), 머지 모드(Merge Mode), 향상된 움직임 벡터 예측(Advanced Motion Vector Prediction; AMVP) 모드, 현재 픽처 참조 모드 중 어떠한 방법인지 여부를 판단할 수 있고, 각 모드에 따라 화면 간 예측 또는 움직임 보상을 수행할 수 있다.
- [150] 감산기(125)는 입력 블록 및 예측 블록의 차분을 사용하여 잔여 블록을 생성할 수 있다. 잔여 블록은 잔여 신호로 칭해질 수도 있다. 잔여 신호는 원 신호 및 예측 신호 간의 차이(difference)를 의미할 수 있다. 또는, 잔여 신호는 원신호 및

예측 신호 간의 차이를 변환(transform)하거나, 양자화하거나, 또는 변환 및 양자화함으로써 생성된 신호일 수 있다. 잔여 블록은 블록 단위의 잔여 신호일 수 있다.

- [151] 변환부(130)는 잔여 블록에 대해 변환(transform)을 수행하여 변환 계수(transform coefficient)를 생성할 수 있고, 생성된 변환 계수를 출력할 수 있다. 여기서, 변환 계수는 잔여 블록에 대한 변환을 수행함으로써 생성된 계수 값일 수 있다. 변환 생략(transform skip) 모드가 적용되는 경우, 변환부(130)는 잔여 블록에 대한 변환을 생략할 수도 있다.
- [152] 변환 계수 또는 잔여 신호에 양자화를 적용함으로써 양자화된 레벨(quantized level)이 생성될 수 있다. 이하, 실시예들에서는 양자화된 레벨도 변환 계수로 칭해질 수 있다.
- [153] 양자화부(140)는 변환 계수 또는 잔여 신호를 양자화 매개변수에 따라 양자화함으로써 양자화된 레벨을 생성할 수 있고, 생성된 양자화된 레벨을 출력할 수 있다. 이때, 양자화부(140)에서는 양자화 행렬을 사용하여 변환 계수를 양자화할 수 있다.
- [154] 엔트로피 부호화부(150)는, 양자화부(140)에서 산출된 값들 또는 부호화 과정에서 산출된 부호화 파라미터(Coding Parameter) 값들 등에 대하여 확률 분포에 따른 엔트로피 부호화를 수행함으로써 비트스트림(bitstream)을 생성할 수 있고, 비트스트림을 출력할 수 있다. 엔트로피 부호화부(150)는 영상의 샘플에 관한 정보 및 영상의 복호화를 위한 정보에 대한 엔트로피 부호화를 수행할 수 있다. 예를 들면, 영상의 복호화를 위한 정보는 구문 요소(syntax element) 등을 포함할 수 있다.
- [155] 엔트로피 부호화가 적용되는 경우, 높은 발생 확률을 갖는 심볼(symbol)에 적은 수의 비트가 할당되고 낮은 발생 확률을 갖는 심볼에 많은 수의 비트가 할당되어 심볼이 표현됨으로써, 부호화 대상 심볼들에 대한 비트열의 크기가 감소될 수 있다. 엔트로피 부호화부(150)는 엔트로피 부호화를 위해 지수 골롬(exponential Golomb), CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary Arithmetic Coding)과 같은 부호화 방법을 사용할 수 있다. 예를 들면, 엔트로피 부호화부(150)는 가변 길이 부호화(Variable Length Coding/Code; VLC) 테이블을 이용하여 엔트로피 부호화를 수행할 수 있다. 또한 엔트로피 부호화부(150)는 대상 심볼의 이진화(binization) 방법 및 대상 심볼/빈(bin)의 확률 모델(probability model)을 도출한 후, 도출된 이진화 방법, 확률 모델, 문맥 모델(Context Model)을 사용하여 산술 부호화를 수행할 수도 있다.
- [156] 엔트로피 부호화부(150)는 변환 계수 레벨(양자화된 레벨)을 부호화하기 위해 변환 계수 스캐닝(Transform Coefficient Scanning) 방법을 통해 2차원의 블록 형태(form) 계수를 1차원의 벡터 형태로 변경할 수 있다.
- [157] 부호화 파라미터(Coding Parameter)는 구문 요소와 같이 부호화기에서

부호화되어 복호화기로 시그널링되는 정보(플래그, 색인 등)뿐만 아니라, 부호화 과정 또는 복호화 과정에서 유도되는 정보를 포함할 수 있으며, 영상을 부호화하거나 복호화할 때 필요한 정보를 의미할 수 있다. 예를 들어, 유닛/블록 크기, 유닛/블록 깊이, 유닛/블록 분할 정보, 유닛/블록 형태, 유닛/블록 분할 구조, 쿼드트리 형태의 분할 여부, 이진트리 형태의 분할 여부, 이진트리 형태의 분할 방향(가로 방향 또는 세로 방향), 이진트리 형태의 분할 형태(대칭 분할 또는 비대칭 분할), 3분할트리 형태의 분할 여부, 3분할트리 형태의 분할 방향(가로 방향 또는 세로 방향), 3분할트리 형태의 분할 형태(대칭 분할 또는 비대칭 분할), 복합형트리 형태의 분할 여부, 복합형트리 형태의 분할 방향(가로 방향 또는 세로 방향), 복합형트리 형태의 분할 형태(대칭 분할 또는 비대칭 분할), 복합형트리 형태의 분할 트리(이진트리 또는 3분할 트리), 예측 모드(화면 내 예측 또는 화면 간 예측), 화면 내 휘도 예측 모드/방향, 화면 내 색차 예측 모드/방향, 화면 내 분할 정보, 화면 간 분할 정보, 부호화 블록 분할 플래그, 예측 블록 분할 플래그, 변환 블록 분할 플래그, 참조 샘플 필터링 방법, 참조 샘플 필터 탭, 참조 샘플 필터 계수, 예측 블록 필터링 방법, 예측 블록 필터 탭, 예측 블록 필터 계수, 예측 블록 경계 필터링 방법, 예측 블록 경계 필터 탭, 예측 블록 경계 필터 계수, 화면 내 예측 모드, 화면 간 예측 모드, 움직임 정보, 움직임 벡터, 움직임 벡터 차분, 참조 영상 색인, 화면 간 예측 방향, 화면 간 예측 지시자, 예측 리스트 활용 플래그, 참조 영상 리스트, 참조 영상, 움직임 벡터 예측 색인, 움직임 벡터 예측 후보, 움직임 벡터 후보 리스트, 머지 모드 사용 여부, 머지 색인, 머지 후보, 머지 후보 리스트, 스킵(skip) 모드 사용 여부, 보간 필터 종류, 보간 필터 탭, 보간 필터 계수, 움직임 벡터 크기, 움직임 벡터 표현 정확도, 변환 종류, 변환 크기, 1차 변환 사용 여부 정보, 2차 변환 사용 여부 정보, 1차 변환 색인, 2차 변환 색인, 잔여 신호 유무 정보, 부호화 블록 패턴(Coded Block Pattern), 부호화 블록 플래그(Coded Block Flag), 양자화 매개변수, 잔여 양자화 매개변수, 양자화 행렬, 화면 내 루프 필터 적용 여부, 화면 내 루프 필터 계수, 화면 내 루프 필터 탭, 화면 내 루프 필터 모양/형태, 디블록킹 필터 적용 여부, 디블록킹 필터 계수, 디블록킹 필터 탭, 디블록킹 필터 강도, 디블록킹 필터 모양/형태, 적응적 샘플 오프셋 적용 여부, 적응적 샘플 오프셋 값, 적응적 샘플 오프셋 카테고리, 적응적 샘플 오프셋 종류, 적응적 루프내 필터 적용 여부, 적응적 루프내 필터 계수, 적응적 루프내 필터 탭, 적응적 루프내 필터 모양/형태, 이진화/역이진화 방법, 문맥 모델 결정 방법, 문맥 모델 업데이트 방법, 레귤러 모드 수행 여부, 바이패스 모드 수행 여부, 문맥 빈, 바이패스 빈, 중요 계수 플래그, 마지막 중요 계수 플래그, 계수 그룹 단위 부호화 플래그, 마지막 중요 계수 위치, 계수 값이 1보다 큰지에 대한 플래그, 계수 값이 2보다 큰지에 대한 플래그, 계수 값이 3보다 큰지에 대한 플래그, 나머지 계수 값 정보, 부호(sign) 정보, 복원된 휘도 샘플, 복원된 색차 샘플, 잔여 휘도 샘플, 잔여 색차 샘플, 휘도 변환 계수, 색차 변환 계수, 휘도 양자화된 레벨, 색차 양자화된 레벨, 변환 계수

레벨 스캐닝 방법, 복호화기 측면 움직임 벡터 탐색 영역의 크기, 복호화기 측면 움직임 벡터 탐색 영역의 형태, 복호화기 측면 움직임 벡터 탐색 횟수, CTU 크기 정보, 최소 블록 크기 정보, 최대 블록 크기 정보, 최대 블록 깊이 정보, 최소 블록 깊이 정보, 영상 디스플레이/출력 순서, 슬라이스 식별 정보, 슬라이스 타입, 슬라이스 분할 정보, 타일 그룹 식별 정보, 타일 그룹 타입, 타일 그룹 분할 정보, 타일 식별 정보, 타일 타입, 타일 분할 정보, 픽처 타입, 입력 샘플 비트 심도, 복원 샘플 비트 심도, 잔여 샘플 비트 심도, 변환 계수 비트 심도, 양자화된 레벨 비트 심도, 휘도 신호에 대한 정보, 색차 신호에 대한 정보 중 적어도 하나의 값 또는 조합된 형태가 부호화 파라미터에 포함될 수 있다.

- [158] 여기서, 플래그 또는 색인을 시그널링(signaling)한다는 것은 인코더에서는 해당 플래그 또는 색인을 엔트로피 부호화(Entropy Encoding)하여 비트스트림(Bitstream)에 포함하는 것을 의미할 수 있고, 디코더에서는 비트스트림으로부터 해당 플래그 또는 색인을 엔트로피 복호화(Entropy Decoding)하는 것을 의미할 수 있다.
- [159] 부호화 장치(100)가 인터 예측을 통한 부호화를 수행할 경우, 부호화된 현재 영상은 이후에 처리되는 다른 영상에 대한 참조 영상으로서 사용될 수 있다. 따라서, 부호화 장치(100)는 부호화된 현재 영상을 다시 복원 또는 복호화할 수 있고, 복원 또는 복호화된 영상을 참조 영상으로 참조 픽처 버퍼(190)에 저장할 수 있다.
- [160] 양자화된 레벨은 역양자화부(160)에서 역양자화(dequantization)될 수 있고, 역변환부(170)에서 역변환(inverse transform)될 수 있다. 역양자화 및/또는 역변환된 계수는 가산기(175)를 통해 예측 블록과 합해질 수 있다, 역양자화 및/또는 역변환된 계수와 예측 블록을 합함으로써 복원 블록(reconstructed block)이 생성될 수 있다. 여기서, 역양자화 및/또는 역변환된 계수는 역양자화 및 역변환 중 적어도 하나 이상이 수행된 계수를 의미하며, 복원된 잔여 블록을 의미할 수 있다.
- [161] 복원 블록은 필터부(180)를 거칠 수 있다. 필터부(180)는 디블록킹 필터(deblocking filter), 샘플 적응적 오프셋(Sample Adaptive Offset; SAO), 적응적 루프내 필터(Adaptive Loop Filter; ALF) 등 적어도 하나를 복원 샘플, 복원 블록 또는 복원 영상에 적용할 수 있다. 필터부(180)는 루프내 필터(in-loop filter)로 칭해질 수도 있다.
- [162] 디블록킹 필터는 블록들 간의 경계에서 발생한 블록 왜곡을 제거할 수 있다. 디블록킹 필터를 수행할지 여부를 판단하기 위해 블록에 포함된 몇 개의 열 또는 행에 포함된 샘플을 기초로 현재 블록에 디블록킹 필터 적용할지 여부를 판단할 수 있다. 블록에 디블록킹 필터를 적용하는 경우 필요한 디블록킹 필터링 강도에 따라 서로 다른 필터를 적용할 수 있다.
- [163] 샘플 적응적 오프셋을 이용하여 부호화 에러를 보상하기 위해 샘플 값에 적정 오프셋(offset) 값을 더할 수 있다. 샘플 적응적 오프셋은 디블록킹을 수행한

영상에 대해 샘플 단위로 원본 영상과의 오프셋을 보정할 수 있다. 영상에 포함된 샘플을 일정한 수의 영역으로 구분한 후 오프셋을 수행할 영역을 결정하고 해당 영역에 오프셋을 적용하는 방법 또는 각 샘플의 에지 정보를 고려하여 오프셋을 적용하는 방법을 사용할 수 있다.

- [164] 적응적 루프내 필터는 복원 영상 및 원래의 영상을 비교한 값에 기반하여 필터링을 수행할 수 있다. 영상에 포함된 샘플을 소정의 그룹으로 나눈 후 해당 그룹에 적용될 필터를 결정하여 그룹마다 차별적으로 필터링을 수행할 수 있다. 적응적 루프내 필터를 적용할지 여부에 관련된 정보는 부호화 유닛(Coding Unit, CU) 별로 시그널링될 수 있고, 각각의 블록에 따라 적용될 적응적 루프내 필터의 모양 및 필터 계수는 달라질 수 있다.
- [165] 필터부(180)를 거친 복원 블록 또는 복원 영상은 참조 픽처 버퍼(190)에 저장될 수 있다. 필터부(180)를 거친 복원 블록은 참조 영상의 일부일 수 있다. 말하자면, 참조 영상은 필터부(180)를 거친 복원 블록들로 구성된 복원 영상일 수 있다. 저장된 참조 영상은 이후 화면 간 예측 또는 움직임 보상에 사용될 수 있다.
- [166] 도 2는 본 발명이 적용되는 복호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.
- [167] 복호화 장치(200)는 디코더, 비디오 복호화 장치 또는 영상 복호화 장치일 수 있다.
- [168] 도 2를 참조하면, 복호화 장치(200)는 엔트로피 복호화부(210), 역양자화부(220), 역변환부(230), 인트라 예측부(240), 움직임 보상부(250), 가산기(255), 필터부(260) 및 참조 픽처 버퍼(270)를 포함할 수 있다.
- [169] 복호화 장치(200)는 부호화 장치(100)에서 출력된 비트스트림을 수신할 수 있다. 복호화 장치(200)는 컴퓨터 판독가능한 기록 매체에 저장된 비트스트림을 수신하거나, 유/무선 전송 매체를 통해 스트리밍되는 비트스트림을 수신할 수 있다. 복호화 장치(200)는 비트스트림에 대하여 인트라 모드 또는 인터 모드로 복호화를 수행할 수 있다. 또한, 복호화 장치(200)는 복호화를 통해 복원된 영상 또는 복호화된 영상을 생성할 수 있고, 복원된 영상 또는 복호화된 영상을 출력할 수 있다.
- [170] 복호화에 사용되는 예측 모드가 인트라 모드인 경우 스위치가 인트라로 전환될 수 있다. 복호화에 사용되는 예측 모드가 인터 모드인 경우 스위치가 인터로 전환될 수 있다.
- [171] 복호화 장치(200)는 입력된 비트스트림을 복호화하여 복원된 잔여 블록(reconstructed residual block)을 획득할 수 있고, 예측 블록을 생성할 수 있다. 복원된 잔여 블록 및 예측 블록이 획득되면, 복호화 장치(200)는 복원된 잔여 블록과 및 예측 블록을 더함으로써 복호화 대상이 되는 복원 블록을 생성할 수 있다. 복호화 대상 블록은 현재 블록으로 칭해질 수 있다.
- [172] 엔트로피 복호화부(210)는 비트스트림에 대한 확률 분포에 따른 엔트로피 복호화를 수행함으로써 심볼들을 생성할 수 있다. 생성된 심볼들은 양자화된

레벨 형태의 심볼을 포함할 수 있다. 여기에서, 엔트로피 복호화 방법은 상술된 엔트로피 부호화 방법의 역과정일 수 있다.

- [173] 엔트로피 복호화부(210)는 변환 계수 레벨(양자화된 레벨)을 복호화하기 위해 변환 계수 스캐닝 방법을 통해 1차원의 벡터 형태 계수를 2차원의 블록 형태로 변경할 수 있다.
- [174] 양자화된 레벨은 역양자화부(220)에서 역양자화될 수 있고, 역변환부(230)에서 역변환될 수 있다. 양자화된 레벨은 역양자화 및/또는 역변환이 수행된 결과로서, 복원된 잔여 블록으로 생성될 수 있다. 이때, 역양자화부(220)는 양자화된 레벨에 양자화 행렬을 적용할 수 있다.
- [175] 인트라 모드가 사용되는 경우, 인트라 예측부(240)는 복호화 대상 블록 주변의 이미 복호화된 블록의 샘플 값을 이용하는 공간적 예측을 현재 블록에 대해 수행함으로써 예측 블록을 생성할 수 있다.
- [176] 인터 모드가 사용되는 경우, 움직임 보상부(250)는 움직임 벡터 및 참조 픽처 버퍼(270)에 저장되어 있는 참조 영상을 이용하는 움직임 보상을 현재 블록에 대해 수행함으로써 예측 블록을 생성할 수 있다. 상기 움직임 보상부(250)는 움직임 벡터의 값이 정수 값을 가지지 않을 경우에 참조 영상 내의 일부 영역에 대해 보간 필터를 적용하여 예측 블록을 생성할 수 있다. 움직임 보상을 수행하기 위해 부호화 유닛을 기준으로 해당 부호화 유닛에 포함된 예측 유닛의 움직임 보상 방법이 스킵 모드, 머지 모드, AMVP 모드, 현재 픽처 참조 모드 중 어떠한 방법인지 여부를 판단할 수 있고, 각 모드에 따라 움직임 보상을 수행할 수 있다.
- [177] 가산기(255)는 복원된 잔여 블록 및 예측 블록을 가산하여 복원 블록을 생성할 수 있다. 필터부(260)는 디블록킹 필터, 샘플 적응적 오프셋 및 적응적 루프내 필터 등 적어도 하나를 복원 블록 또는 복원 영상에 적용할 수 있다. 필터부(260)는 복원 영상을 출력할 수 있다. 복원 블록 또는 복원 영상은 참조 픽처 버퍼(270)에 저장되어 인터 예측에 사용될 수 있다. 필터부(260)를 거친 복원 블록은 참조 영상의 일부일 수 있다. 말하자면, 참조 영상은 필터부(260)를 거친 복원 블록들로 구성된 복원 영상일 수 있다. 저장된 참조 영상은 이후 화면 간 예측 또는 움직임 보상에 사용될 수 있다.
- [178] 도 3은 영상을 부호화 및 복호화할 때의 영상의 분할 구조를 개략적으로 나타내는 도면이다. 도 3은 하나의 유닛이 복수의 하위 유닛으로 분할되는 실시예를 개략적으로 나타낸다.
- [179] 영상을 효율적으로 분할하기 위해, 부호화 및 복호화에 있어서, 부호화 유닛(Coding Unit; CU)이 사용될 수 있다. 영상 부호화/복호화의 기본 단위로서 부호화 유닛이 사용될 수 있다. 또한, 영상 부호화/복호화 시 화면 내 예측 모드 및 화면 간 예측 모드가 구분되는 단위로 부호화 유닛을 사용할 수 있다. 부호화 유닛은 예측, 변환, 양자화, 역변환, 역양자화, 또는 변환 계수의 부호화/복호화의 과정을 위해 사용되는 기본 단위일 수 있다.

- [180] 도 3을 참조하면, 영상(300)은 최대 부호화 유닛(Largest Coding Unit; LCU) 단위로 순차적으로 분할되고, LCU 단위로 분할 구조가 결정된다. 여기서, LCU는 부호화 트리 유닛(Coding Tree Unit; CTU)과 동일한 의미로 사용될 수 있다. 유닛의 분할은 유닛에 해당하는 블록의 분할을 의미할 수 있다. 블록 분할 정보에는 유닛의 깊이(depth)에 관한 정보가 포함될 수 있다. 깊이 정보는 유닛이 분할되는 회수 및/또는 정도를 나타낼 수 있다. 하나의 유닛은 트리 구조(tree structure)를 기초로 깊이 정보를 가지고 계층적으로 복수의 하위 유닛들로 분할될 수 있다. 말하자면, 유닛 및 상기의 유닛의 분할에 의해 생성된 하위 유닛은 노드 및 상기의 노드의 자식 노드에 각각 대응할 수 있다. 각각의 분할된 하위 유닛은 깊이 정보를 가질 수 있다. 깊이 정보는 CU의 크기를 나타내는 정보일 수 있고, 각 CU마다 저장될 수 있다. 유닛 깊이는 유닛이 분할된 회수 및/또는 정도를 나타내므로, 하위 유닛의 분할 정보는 하위 유닛의 크기에 관한 정보를 포함할 수도 있다.
- [181] 분할 구조는 CTU(310) 내에서의 부호화 유닛(Coding Unit; CU)의 분포를 의미할 수 있다. 이러한 분포는 하나의 CU를 복수(2, 4, 8, 16 등을 포함하는 2 이상의 양의 정수)의 CU들로 분할할지 여부에 따라 결정할 수 있다. 분할에 의해 생성된 CU의 가로 크기 및 세로 크기는 각각 분할 전의 CU의 가로 크기의 절반 및 세로 크기의 절반이거나, 분할된 개수에 따라 분할 전의 CU의 가로 크기보다 작은 크기 및 세로 크기보다 작은 크기를 가질 수 있다. CU는 복수의 CU로 재귀적으로 분할될 수 있다. 재귀적 분할에 의해, 분할된 CU의 가로 크기 및 세로 크기 중 적어도 하나의 크기가 분할 전의 CU의 가로 크기 및 세로 크기 중 적어도 하나에 비해 감소될 수 있다. CU의 분할은 기정의된 깊이 또는 기정의된 크기까지 재귀적으로 이루어질 수 있다. 예컨대, CTU의 깊이는 0일 수 있고, 최소 부호화 유닛(Smallest Coding Unit; SCU)의 깊이는 기정의된 최대 깊이일 수 있다. 여기서, CTU는 상술된 것과 같이 최대의 부호화 유닛 크기를 가지는 부호화 유닛일 수 있고, SCU는 최소의 부호화 유닛 크기를 가지는 부호화 유닛일 수 있다. CTU(310)로부터 분할이 시작되고, 분할에 의해 CU의 가로 크기 및/또는 세로 크기가 줄어들 때마다 CU의 깊이는 1씩 증가한다. 예를 들면, 각각의 깊이 별로, 분할되지 않는 CU는 $2N \times 2N$ 크기를 가질 수 있다. 또한, 분할되는 CU의 경우, $2N \times 2N$ 크기의 CU가 $N \times N$ 크기를 가지는 4개의 CU들로 분할될 수 있다. N 의 크기는 깊이가 1씩 증가할 때마다 절반으로 감소할 수 있다.
- [182] 또한, CU가 분할되는지 여부에 대한 정보는 CU의 분할 정보를 통해 표현될 수 있다. 분할 정보는 1비트의 정보일 수 있다. SCU를 제외한 모든 CU는 분할 정보를 포함할 수 있다. 예를 들면, 분할 정보의 값이 제1 값이면, CU가 분할되지 않을 수 있고, 분할 정보의 값이 제2 값이면, CU가 분할될 수 있다.
- [183] 도 3을 참조하면, 깊이가 0인 CTU는 64×64 블록일 수 있다. 0은 최소 깊이일 수 있다. 깊이가 3인 SCU는 8×8 블록일 수 있다. 3은 최대 깊이일 수 있다. 32×32 블록 및 16×16 블록의 CU는 각각 깊이 1 및 깊이 2로 표현될 수 있다.

- [184] 예를 들어, 하나의 부호화 유닛이 4개의 부호화 유닛으로 분할 될 경우, 분할된 4개의 부호화 유닛의 가로 및 세로 크기는 분할되기 전 부호화 유닛의 가로 및 세로 크기와 비교하여 각각 절반의 크기를 가질 수 있다. 일 예로, 32x32 크기의 부호화 유닛이 4개의 부호화 유닛으로 분할 될 경우, 분할된 4개의 부호화 유닛은 각각 16x16의 크기를 가질 수 있다. 하나의 부호화 유닛이 4개의 부호화 유닛으로 분할 될 경우, 부호화 유닛은 쿼드트리(quad-tree) 형태로 분할(쿼드트리 분할, quad-tree partition)되었다고 할 수 있다.
- [185] 예를 들어, 하나의 부호화 유닛이 2개의 부호화 유닛으로 분할 될 경우, 분할된 2개의 부호화 유닛의 가로 또는 세로 크기는 분할되기 전 부호화 유닛의 가로 또는 세로 크기와 비교하여 절반의 크기를 가질 수 있다. 일 예로, 32x32 크기의 부호화 유닛이 2개의 부호화 유닛으로 세로로 분할 될 경우, 분할된 2개의 부호화 유닛은 각각 16x32의 크기를 가질 수 있다. 일 예로, 8x32 크기의 부호화 유닛이 2개의 부호화 유닛으로 가로로 분할 될 경우, 분할된 2개의 부호화 유닛은 각각 8x16의 크기를 가질 수 있다. 하나의 부호화 유닛이 2개의 부호화 유닛으로 분할 될 경우, 부호화 유닛은 이진트리(binary-tree) 형태로 분할(이진트리 분할, binary-tree partition)되었다고 할 수 있다.
- [186] 예를 들어, 하나의 부호화 유닛이 3개의 부호화 유닛으로 분할 될 경우, 분할되기 전 부호화 유닛의 가로 또는 세로 크기를 1:2:1의 비율로 분할함으로써, 3개의 부호화 유닛으로 분할할 수 있다. 일 예로, 16x32 크기의 부호화 유닛이 3개의 부호화 유닛으로 가로로 분할 될 경우, 분할된 3개의 부호화 유닛은 상측부터 각각 16x8, 16x16 및 16x8의 크기를 가질 수 있다. 일 예로, 32x32 크기의 부호화 유닛이 3개의 부호화 유닛으로 세로로 분할 될 경우, 분할된 3개의 부호화 유닛은 좌측부터 각각 8x32, 16x32 및 8x32의 크기를 가질 수 있다. 하나의 부호화 유닛이 3개의 부호화 유닛으로 분할 될 경우, 부호화 유닛은 3분할트리(ternary-tree) 형태로 분할(3분할트리 분할, ternary-tree partition)되었다고 할 수 있다.
- [187] 도 3의 CTU(320)는 쿼드트리 분할, 이진트리 분할 및 3분할트리 분할이 모두 적용된 CTU의 일 예이다.
- [188] 전술한 바와 같이, CTU를 분할하기 위해, 쿼드트리 분할, 이진트리 분할 및 3분할트리 분할 중 적어도 하나가 적용될 수 있다. 각각의 분할은 소정의 우선 순위에 기초하여 적용될 수 있다. 예컨대, CTU에 대해 쿼드트리 분할이 우선적으로 적용될 수 있다. 더 이상 쿼드트리 분할될 수 없는 부호화 유닛은 쿼드트리의 리프 노드에 해당될 수 있다. 쿼드트리의 리프 노드에 해당하는 부호화 유닛은 이진트리 및/또는 3분할트리의 루트 노드가 될 수 있다. 즉, 쿼드트리의 리프 노드에 해당하는 부호화 유닛은 이진트리 분할되거나 3분할트리 분할되거나 또는 더 이상 분할되지 않을 수 있다. 이 때, 쿼드트리의 리프 노드에 해당하는 부호화 유닛을 이진트리 분할하거나 3분할트리 분할하여 생성된 부호화 유닛에 대해서는 다시 쿼드트리 분할이 수행되지 않도록

함으로써, 블록의 분할 및/또는 분할 정보의 시그널링을 효과적으로 수행할 수 있다.

- [189] 쿼드트리 of 각 노드에 해당하는 부호화 유닛의 분할은 쿼드 분할 정보를 이용하여 시그널링될 수 있다. 제1값(예컨대, '1')을 갖는 쿼드 분할 정보는 해당 부호화 유닛이 쿼드트리 분할됨을 지시할 수 있다. 제2값(예컨대, '0')을 갖는 쿼드 분할 정보는 해당 부호화 유닛이 쿼드트리 분할되지 않음을 지시할 수 있다. 쿼드 분할 정보는 소정의 길이(예컨대, 1비트)를 갖는 플래그일 수 있다.
- [190] 이진트리 분할과 3분할트리 분할 사이에는 우선순위가 존재하지 않을 수 있다. 즉, 쿼드트리의 리프 노드에 해당하는 부호화 유닛은 이진트리 분할되거나 3분할트리 분할될 수 있다. 또한, 이진트리 분할 또는 3분할트리 분할에 의해 생성된 부호화 유닛은 다시 이진트리 분할 또는 3분할트리 분할되거나 또는 더 이상 분할되지 않을 수 있다.
- [191] 이진트리 분할과 3분할트리 분할 사이에 우선순위가 존재하지 않는 경우의 분할은 복합형트리 분할(multi-type tree partition)이라고 호칭할 수 있다. 즉, 쿼드트리의 리프 노드에 해당하는 부호화 유닛은 복합형트리(multi-type tree)의 루트 노드가 될 수 있다. 복합형트리의 각 노드에 해당하는 부호화 유닛의 분할은 복합형트리의 분할 여부 정보, 분할 방향 정보 및 분할 트리 정보 중 적어도 하나를 이용하여 시그널링될 수 있다. 상기 복합형트리의 각 노드에 해당하는 부호화 유닛의 분할을 위해 순차적으로 분할 여부 정보, 분할 방향 정보 및 분할 트리 정보가 시그널링될 수도 있다.
- [192] 제1값(예컨대, '1')을 갖는 복합형트리의 분할 여부 정보는 해당 부호화 유닛이 복합형트리 분할됨을 지시할 수 있다. 제2값(예컨대, '0')을 갖는 복합형트리의 분할 여부 정보는 해당 부호화 유닛이 복합형트리 분할되지 않음을 지시할 수 있다.
- [193] 복합형트리의 각 노드에 해당하는 부호화 유닛이 복합형트리 분할되는 경우, 해당 부호화 유닛은 분할 방향 정보를 더 포함할 수 있다. 분할 방향 정보는 복합형트리 분할의 분할 방향을 지시할 수 있다. 제1값(예컨대, '1')을 갖는 분할 방향 정보는 해당 부호화 유닛이 세로 방향으로 분할됨을 지시할 수 있다. 제2값(예컨대, '0')을 갖는 분할 방향 정보는 해당 부호화 유닛이 가로 방향으로 분할됨을 지시할 수 있다.
- [194] 복합형트리의 각 노드에 해당하는 부호화 유닛이 복합형트리 분할되는 경우, 해당 부호화 유닛은 분할 트리 정보를 더 포함할 수 있다. 분할 트리 정보는 복합형트리 분할을 위해 사용된 트리를 지시할 수 있다. 제1값(예컨대, '1')을 갖는 분할 트리 정보는 해당 부호화 유닛이 이진트리 분할됨을 지시할 수 있다. 제2값(예컨대, '0')을 갖는 분할 트리 정보는 해당 부호화 유닛이 3분할트리 분할됨을 지시할 수 있다.
- [195] 분할 여부 정보, 분할 트리 정보 및 분할 방향 정보는 각각 소정의 길이(예컨대, 1비트)를 갖는 플래그일 수 있다.

- [196] 쿼드 분할 정보, 복합형트리의 분할 여부 정보, 분할 방향 정보 및 분할 트리 정보 중 적어도 하나는 엔트로피 부호화/복호화될 수 있다. 상기 정보들의 엔트로피 부호화/복호화를 위해, 현재 부호화 유닛에 인접한 주변 부호화 유닛의 정보가 이용될 수 있다. 예컨대, 좌측 부호화 유닛 및/또는 상측 부호화 유닛의 분할 형태(분할 여부, 분할 트리 및/또는 분할 방향)는 현재 부호화 유닛의 분할 형태와 유사할 확률이 높다. 따라서, 주변 부호화 유닛의 정보에 기초하여, 현재 부호화 유닛의 정보의 엔트로피 부호화/복호화를 위한 컨텍스트 정보를 유도할 수 있다. 이때, 주변 부호화 유닛의 정보에는 해당 부호화 유닛의 쿼드 분할 정보, 복합형트리의 분할 여부 정보, 분할 방향 정보 및 분할 트리 정보 중 적어도 하나가 포함될 수 있다.
- [197] 다른 실시예로서, 이진트리 분할과 3분할트리 분할 중, 이진트리 분할이 우선적으로 수행될 수 있다. 즉, 이진트리 분할이 먼저 적용되고, 이진트리의 리프 노드에 해당하는 부호화 유닛을 3분할트리의 루트 노드로 설정할 수도 있다. 이 경우, 3분할트리의 노드에 해당하는 부호화 유닛에 대해서는 쿼드트리 분할 및 이진트리 분할이 수행되지 않을 수 있다.
- [198] 쿼드트리 분할, 이진트리 분할 및/또는 3분할트리 분할에 의해 더 이상 분할되지 않는 부호화 유닛은 부호화, 예측 및/또는 변환의 단위가 될 수 있다. 즉, 예측 및/또는 변환을 위해 부호화 유닛이 더 이상 분할되지 않을 수 있다. 따라서, 부호화 유닛을 예측 유닛 및/또는 변환 유닛으로 분할하기 위한 분할 구조, 분할 정보 등이 비트스트림에 존재하지 않을 수 있다.
- [199] 다만, 분할의 단위가 되는 부호화 유닛의 크기가 최대 변환 블록의 크기보다 큰 경우, 해당 부호화 유닛은 최대 변환 블록의 크기와 같거나 또는 작은 크기가 될 때까지 재귀적으로 분할될 수 있다. 예컨대, 부호화 유닛의 크기가 64x64이고, 최대 변환 블록의 크기가 32x32인 경우, 상기 부호화 유닛은 변환을 위해, 4개의 32x32 블록으로 분할될 수 있다. 예컨대, 부호화 유닛의 크기가 32x64이고, 최대 변환 블록의 크기가 32x32인 경우, 상기 부호화 유닛은 변환을 위해, 2개의 32x32 블록으로 분할될 수 있다. 이 경우, 변환을 위한 부호화 유닛의 분할 여부는 별도로 시그널링되지 않고, 상기 부호화 유닛의 가로 또는 세로와 최대 변환 블록의 가로 또는 세로의 비교에 의해 결정될 수 있다. 예컨대, 부호화 유닛의 가로가 최대 변환 블록의 가로보다 큰 경우, 부호화 유닛은 세로로 2등분될 수 있다. 또한, 부호화 유닛의 세로가 최대 변환 블록의 세로보다 큰 경우, 부호화 유닛은 가로로 2등분될 수 있다.
- [200] 부호화 유닛의 최대 및/또는 최소 크기에 관한 정보, 변환 블록의 최대 및/또는 최소 크기에 관한 정보는 부호화 유닛의 상위 레벨에서 시그널링되거나 결정될 수 있다. 상기 상위 레벨은 예컨대, 시퀀스 레벨, 픽처 레벨, 타일 레벨, 타일 그룹 레벨, 슬라이스 레벨 등일 수 있다. 예컨대, 부호화 유닛의 최소 크기는 4x4로 결정될 수 있다. 예컨대, 변환 블록의 최대 크기는 64x64로 결정될 수 있다. 예컨대, 변환 블록의 최소 크기는 4x4로 결정될 수 있다.

- [201] 쿼드트리의 리프 노드에 해당하는 부호화 유닛의 최소 크기(쿼드트리 최소 크기)에 관한 정보 및/또는 복합형트리의 루트 노드에서 리프 노드에 이르는 최대 깊이(복합형트리 최대 깊이)에 관한 정보는 부호화 유닛의 상위 레벨에서 시그널링되거나 결정될 수 있다. 상기 상위 레벨은 예컨대, 시퀀스 레벨, 픽처 레벨, 슬라이스 레벨, 타일 그룹 레벨, 타일 레벨 등일 수 있다. 상기 쿼드트리 최소 크기에 관한 정보 및/또는 상기 복합형트리 최대 깊이에 관한 정보는 화면 내 슬라이스와 화면 간 슬라이스의 각각에 대해 시그널링되거나 결정될 수 있다.
- [202] CTU의 크기와 변환 블록의 최대 크기에 대한 차분 정보는 부호화 유닛의 상위 레벨에서 시그널링되거나 결정될 수 있다. 상기 상위 레벨은 예컨대, 시퀀스 레벨, 픽처 레벨, 슬라이스 레벨, 타일 그룹 레벨, 타일 레벨 등일 수 있다. 이진트리의 각 노드에 해당하는 부호화 유닛의 최대 크기(이진트리 최대 크기)에 관한 정보는 부호화 트리 유닛의 크기와 상기 차분 정보를 기반으로 결정될 수 있다. 3분할트리의 각 노드에 해당하는 부호화 유닛의 최대 크기(3분할트리 최대 크기)는 슬라이스의 타입에 따라 다른 값을 가질 수 있다. 예컨대, 화면 내 슬라이스인 경우, 3분할트리 최대 크기는 32x32일 수 있다. 또한, 예컨대, 화면 간 슬라이스인 경우, 3분할 트리 최대 크기는 128x128일 수 있다. 예컨대, 이진트리의 각 노드에 해당하는 부호화 유닛의 최소 크기(이진트리 최소 크기) 및/또는 3분할트리의 각 노드에 해당하는 부호화 유닛의 최소 크기(3분할트리 최소 크기)는 부호화 블록의 최소 크기로 설정될 수 있다.
- [203] 또 다른 예로, 이진트리 최대 크기 및/또는 3분할트리 최대 크기는 슬라이스 레벨에서 시그널링되거나 결정될 수 있다. 또한, 이진트리 최소 크기 및/또는 3분할트리 최소 크기는 슬라이스 레벨에서 시그널링되거나 결정될 수 있다.
- [204] 전술한 다양한 블록의 크기 및 깊이 정보에 기초하여, 쿼드 분할 정보, 복합형트리의 분할 여부 정보, 분할 트리 정보 및/또는 분할 방향 정보 등이 비트스트림에 존재하거나 존재하지 않을 수 있다.
- [205] 예컨대, 부호화 유닛의 크기가 쿼드트리 최소 크기보다 크지 않으면, 상기 부호화 유닛은 쿼드 분할 정보를 포함하지 않고, 해당 쿼드 분할 정보는 제2값으로 추론될 수 있다.
- [206] 예컨대, 복합형트리의 노드에 해당하는 부호화 유닛의 크기(가로 및 세로)가 이진트리 최대 크기(가로 및 세로) 및/또는 3분할트리 최대 크기(가로 및 세로)보다 큰 경우, 상기 부호화 유닛은 이진트리 분할 및/또는 3분할트리 분할되지 않을 수 있다. 그에 따라, 상기 복합형트리의 분할 여부 정보는 시그널링되지 않고, 제2값으로 추론될 수 있다.
- [207] 또는, 복합형트리의 노드에 해당하는 부호화 유닛의 크기(가로 및 세로)가 이진트리 최소 크기(가로 및 세로)와 동일하거나, 부호화 유닛의 크기(가로 및 세로)가 3분할트리 최소 크기(가로 및 세로)의 두 배와 동일한 경우, 상기 부호화 유닛은 이진트리 분할 및/또는 3분할트리 분할되지 않을 수 있다. 그에 따라, 상기 복합형트리의 분할 여부 정보는 시그널링되지 않고, 제2값으로 추론될 수

있다. 왜냐하면, 상기 부호화 유닛을 이진트리 분할 및/또는 3분할트리 분할할 경우, 이진트리 최소 크기 및/또는 3분할트리 최소 크기보다 작은 부호화 유닛이 생성되기 때문이다.

[208] 또는, 이진트리 분할 또는 3분할트리 분할은 가상의 파이프라인 데이터 유닛의 크기(이하, 파이프라인 버퍼 크기)에 기초하여 제한될 수 있다. 예컨대, 이진트리 분할 또는 3분할트리 분할에 의해, 부호화 유닛이 파이프라인 버퍼 크기에 적합하지 않은 서브 부호화 유닛으로 분할될 경우, 해당 이진트리 분할 또는 3분할트리 분할은 제한될 수 있다. 파이프라인 버퍼 크기는 최대 변환 블록의 크기(예컨대, 64X64)일 수 있다. 예컨대, 파이프라인 버퍼 크기가 64X64일 때, 아래의 분할은 제한될 수 있다.

[209] - NxM(N 및/또는 M은 128) 부호화 유닛에 대한 3분할트리 분할

[210] - 128xN($N \leq 64$) 부호화 유닛에 대한 수평 방향 이진트리 분할

[211] - Nx128($N \leq 64$) 부호화 유닛에 대한 수직 방향 이진트리 분할

[212] 또는, 복합형트리의 노드에 해당하는 부호화 유닛의 복합형트리 내의 깊이가 복합형트리 최대 깊이와 동일한 경우, 상기 부호화 유닛은 이진트리 분할 및/또는 3분할트리 분할되지 않을 수 있다. 그에 따라, 상기 복합형트리의 분할 여부 정보는 시그널링되지 않고, 제2값으로 추론될 수 있다.

[213] 또는, 복합형트리의 노드에 해당하는 부호화 유닛에 대해 수직 방향 이진트리 분할, 수평 방향 이진트리 분할, 수직 방향 3분할트리 분할 및 수평 방향 3분할트리 분할 중 적어도 하나가 가능한 경우에만, 상기 복합형트리의 분할 여부 정보를 시그널링할 수 있다. 그렇지 않은 경우, 상기 부호화 유닛은 이진트리 분할 및/또는 3분할트리 분할되지 않을 수 있다. 그에 따라, 상기 복합형트리의 분할 여부 정보는 시그널링되지 않고, 제2값으로 추론될 수 있다.

[214] 또는, 복합형트리의 노드에 해당하는 부호화 유닛에 대해 수직 방향 이진트리 분할과 수평 방향 이진트리 분할이 모두 가능하거나, 수직 방향 3분할트리 분할과 수평 방향 3분할트리 분할이 모두 가능한 경우에만, 상기 분할 방향 정보를 시그널링할 수 있다. 그렇지 않은 경우, 상기 분할 방향 정보는 시그널링되지 않고, 분할이 가능한 방향을 지시하는 값으로 추론될 수 있다.

[215] 또는, 복합형트리의 노드에 해당하는 부호화 유닛에 대해 수직 방향 이진트리 분할과 수직 방향 3분할트리 분할이 모두 가능하거나, 수평 방향 이진트리 분할과 수평 방향 3분할트리 분할이 모두 가능한 경우에만, 상기 분할 트리 정보를 시그널링할 수 있다. 그렇지 않은 경우, 상기 분할 트리 정보는 시그널링되지 않고, 분할이 가능한 트리를 지시하는 값으로 추론될 수 있다.

[216] 도 4는 화면 내 예측 과정의 실시예를 설명하기 위한 도면이다.

[217] 도 4의 중심으로부터 외곽으로의 화살표들은 화면 내 예측 모드들의 예측 방향들을 나타낼 수 있다.

[218] 화면 내 부호화 및/또는 복호화는 현재 블록의 주변 블록의 참조 샘플을 이용하여 수행될 수 있다. 주변 블록은 복원된 주변 블록일 수 있다. 예를 들면,

화면 내 부호화 및/또는 복호화는 복원된 주변 블록이 포함하는 참조 샘플의 값 또는 부호화 파라미터를 이용하여 수행될 수 있다.

[219] 예측 블록은 화면 내 예측의 수행의 결과로 생성된 블록을 의미할 수 있다. 예측 블록은 CU, PU 및 TU 중 적어도 하나에 해당할 수 있다. 예측 블록의 단위는 CU, PU 및 TU 중 적어도 하나의 크기일 수 있다. 예측 블록은 2x2, 4x4, 16x16, 32x32 또는 64x64 등의 크기를 갖는 정사각형의 형태의 블록일 수 있고, 2x8, 4x8, 2x16, 4x16 및 8x16 등의 크기를 갖는 직사각형 모양의 블록일 수도 있다.

[220] 화면 내 예측은 현재 블록에 대한 화면 내 예측 모드에 따라 수행될 수 있다. 현재 블록이 가질 수 있는 화면 내 예측 모드의 개수는 기정의된 고정된 값일 수 있으며, 예측 블록의 속성에 따라 다르게 결정된 값일 수 있다. 예를 들면, 예측 블록의 속성은 예측 블록의 크기 및 예측 블록의 형태 등을 포함할 수 있다.

[221] 화면 내 예측 모드의 개수는 블록의 크기에 관계없이 N개로 고정될 수 있다. 또는, 예를 들면, 화면 내 예측 모드의 개수는 3, 5, 9, 17, 34, 35, 36, 65, 또는 67 등일 수 있다. 또는, 화면 내 예측 모드의 개수는 블록의 크기 및/또는 색 성분(color component)의 타입에 따라 상이할 수 있다. 예를 들면, 색 성분이 휘도(luma) 신호인지 아니면 색차(chroma) 신호인지에 따라 화면 내 예측 모드의 개수가 다를 수 있다. 예컨대, 블록의 크기가 커질수록 화면 내 예측 모드의 개수는 많아질 수 있다. 또는 휘도 성분 블록의 화면 내 예측 모드의 개수는 색차 성분 블록의 화면 내 예측 모드의 개수보다 많을 수 있다.

[222] 화면 내 예측 모드는 비방향성 모드 또는 방향성 모드일 수 있다. 비방향성 모드는 DC 모드 또는 플래너(Planar) 모드일 수 있으며, 방향성 모드(angular mode)는 특정한 방향 또는 각도를 가지는 예측 모드일 수 있다. 상기 화면 내 예측 모드는 모드 번호, 모드 값, 모드 숫자, 모드 각도, 모드 방향 중 적어도 하나로 표현될 수 있다. 화면 내 예측 모드의 개수는 상기 비방향성 및 방향성 모드를 포함하는 하나 이상의 M개 일 수 있다. 현재 블록을 화면 내 예측하기 위해 복원된 주변 블록에 포함되는 샘플들이 현재 블록의 참조 샘플로 이용 가능한지 여부를 검사하는 단계가 수행될 수 있다. 현재 블록의 참조 샘플로 이용할 수 없는 샘플이 존재할 경우, 복원된 주변 블록에 포함된 샘플들 중 적어도 하나의 샘플 값을 복사 및/또는 보간한 값을 이용하여 참조 샘플로 이용할 수 없는 샘플의 샘플 값으로 대체한 후, 현재 블록의 참조 샘플로 이용할 수 있다.

[223] 도 7은 화면 내 예측에 이용 가능한 참조 샘플들을 설명하기 위한 도면이다.

[224] 도 7에 도시된 바와 같이, 현재 블록의 화면 내 예측을 위해, 참조 샘플 라인 0 내지 참조 샘플 라인 3 중 적어도 하나가 이용될 수 있다. 도 7에 있어서, 세그먼트 A와 세그먼트 F의 샘플들은 복원된 이웃 블록으로부터 가져오는 대신 각각 세그먼트 B와 세그먼트 E의 가장 가까운 샘플들로 패딩될 수 있다. 현재 블록의 화면 내 예측을 위해 이용될 참조 샘플 라인을 지시하는 인덱스 정보가 시그널링될 수 있다. 현재 블록의 상단 경계가 CTU의 경계인 경우, 참조 샘플

- 라인 0만 이용 가능할 수 있다. 따라서 이 경우, 상기 인덱스 정보는 시그널링되지 않을 수 있다. 참조 샘플 라인 0 이외에 다른 참조 샘플 라인이 이용되는 경우, 후술하는 예측 블록에 대한 필터링은 수행되지 않을 수 있다.
- [225] 화면 내 예측 시 화면 내 예측 모드 및 현재 블록의 크기 중 적어도 하나에 기반하여 참조 샘플 또는 예측 샘플 중 적어도 하나에 필터를 적용할 수 있다.
- [226] 플래너 모드의 경우, 현재 블록의 예측 블록을 생성할 때, 예측 대상 샘플의 예측 블록 내 위치에 따라, 현재 샘플의 상단 및 좌측 참조 샘플, 현재 블록의 우상단 및 좌하단 참조 샘플의 가중합을 이용하여 예측 대상 샘플의 샘플값을 생성할 수 있다. 또한, DC 모드의 경우, 현재 블록의 예측 블록을 생성할 때, 현재 블록의 상단 및 좌측 참조 샘플들의 평균 값을 이용할 수 있다. 또한, 방향성 모드의 경우 현재 블록의 상단, 좌측, 우상단 및/또는 좌하단 참조 샘플을 이용하여 예측 블록을 생성할 수 있다. 예측 샘플 값 생성을 위해 실수 단위의 보간을 수행할 수도 있다.
- [227] 색 성분간 화면내 예측의 경우, 제1 색 성분의 대응 복원 블록에 기초하여 제2 색 성분의 현재 블록에 대한 예측 블록을 생성할 수 있다. 예컨대, 제1 색 성분은 휘도 성분, 제2 색 성분은 색차 성분일 수 있다. 색 성분간 화면내 예측을 위해, 제1 색 성분과 제2 색 성분 간의 선형 모델의 파라미터가 템플릿에 기초하여 유도될 수 있다. 템플릿은 현재 블록의 상단 및/또는 좌측 주변 샘플 및 이에 대응하는 제1 색 성분의 복원 블록의 상단 및/또는 좌측 주변 샘플을 포함할 수 있다. 예컨대, 선형 모델의 파라미터는 템플릿내의 샘플들 중 최대값을 갖는 제1 색 성분의 샘플값과 이에 대응하는 제2 색 성분의 샘플값, 템플릿내의 샘플들 중 최소값을 갖는 제1 색 성분의 샘플값과 이에 대응하는 제2 색 성분의 샘플값을 이용하여 유도될 수 있다. 선형 모델의 파라미터가 유도되면, 대응 복원 블록을 선형 모델에 적용하여 현재 블록에 대한 예측 블록을 생성할 수 있다. 영상 포맷에 따라, 제1 색 성분의 복원 블록의 주변 샘플 및 대응 복원 블록에 대해서도 샘플링이 수행될 수 있다. 예컨대, 제2 색 성분의 1개의 샘플이 제1 색 성분의 4개의 샘플들에 대응되는 경우, 제1 색 성분의 4개의 샘플들을 서브 샘플링하여, 1개의 대응 샘플을 계산할 수 있다. 이 경우, 선형 모델의 파라미터 유도 및 색 성분간 화면 내 예측은 서브 샘플링된 대응 샘플에 기초하여 수행될 수 있다. 색 성분간 화면내 예측의 수행 여부 및/또는 템플릿의 범위는 화면내 예측 모드로서 시그널링될 수 있다.
- [228] 현재 블록은 가로 또는 세로 방향으로 2개 또는 4개의 서브 블록들로 분할될 수 있다. 분할된 서브 블록들은 순차적으로 복원될 수 있다. 즉, 서브 블록에 대해 화면 내 예측이 수행되어 서브 예측 블록이 생성될 수 있다. 또한, 서브 블록에 대해 역양자화 및/또는 역변환이 수행되어 서브 잔차 블록이 생성될 수 있다. 서브 예측 블록을 서브 잔차 블록에 더해서 복원된 서브 블록이 생성될 수 있다. 복원된 서브 블록은 후순위 서브 블록의 화면 내 예측을 위한 참조 샘플로서 이용될 수 있다. 서브 블록은 소정 개수(예컨대, 16개) 이상의 샘플들을 포함하는

블록일 수 있다. 따라서, 예컨대, 현재 블록이 8x4 블록 또는 4x8 블록의 경우, 현재 블록은 2개의 서브 블록들로 분할될 수 있다. 또한, 현재 블록이 4x4 블록인 경우, 현재 블록은 서브 블록들로 분할될 수 없다. 현재 블록이 그 외의 크기를 갖는 경우, 현재 블록은 4개의 서브 블록들로 분할될 수 있다. 상기 서브 블록 기반의 화면 내 예측의 수행 여부 및/또는 분할 방향(가로 또는 세로)에 관한 정보가 시그널링될 수 있다. 상기 서브 블록 기반의 화면 내 예측은 참조 샘플 라인 0을 이용하는 경우에만 수행되도록 제한될 수 있다. 상기 서브 블록 기반의 화면 내 예측이 수행되는 경우, 후술하는 예측 블록에 대한 필터링은 수행되지 않을 수 있다.

- [229] 화면내 예측된 예측 블록에 필터링을 수행하여 최종 예측 블록을 생성할 수 있다. 상기 필터링은 필터링 대상 샘플, 좌측 참조 샘플, 상단 참조 샘플 및/또는 좌상단 참조 샘플에 소정의 가중치를 적용함으로써 수행될 수 있다. 상기 필터링에 이용되는 가중치 및/또는 참조 샘플(범위, 위치 등)은 블록 크기, 화면내 예측 모드 및 필터링 대상 샘플의 예측 블록 내 위치 중 적어도 하나에 기초하여 결정될 수 있다. 상기 필터링은 소정의 화면내 예측 모드(예컨대, DC, planar, 수직, 수평, 대각 및/또는 인접 대각 모드)의 경우에만 수행될 수 있다. 인접 대각 모드는 대각 모드에 k 를 가감한 모드일 수 있다. 예컨대, k 는 8 이하의 양의 정수일 수 있다.
- [230] 현재 블록의 화면 내 예측 모드는 현재 블록의 주변에 존재하는 블록의 화면 내 예측 모드로부터 예측하여 엔트로피 부호화/복호화할 수 있다. 현재 블록과 주변 블록의 화면 내 예측 모드가 동일하면 소정의 플래그 정보를 이용하여 현재 블록과 주변 블록의 화면 내 예측 모드가 동일하다는 정보를 시그널링할 수 있다. 또한, 복수 개의 주변 블록의 화면 내 예측 모드 중 현재 블록의 화면 내 예측 모드와 동일한 화면 내 예측 모드에 대한 지시자 정보를 시그널링 할 수 있다. 현재 블록과 주변 블록의 화면 내 예측 모드가 상이하면 주변 블록의 화면 내 예측 모드를 기초로 엔트로피 부호화/복호화를 수행하여 현재 블록의 화면 내 예측 모드 정보를 엔트로피 부호화/복호화할 수 있다.
- [231] 도 5는 화면 간 예측 과정의 실시예를 설명하기 위한 도면이다.
- [232] 도 5에 도시된 사각형은 영상을 나타낼 수 있다. 또한, 도 5에서 화살표는 예측 방향을 나타낼 수 있다. 각 영상은 부호화 타입에 따라 I 픽처(Intra Picture), P 픽처(Predictive Picture), B 픽처(Bi-predictive Picture) 등으로 분류될 수 있다.
- [233] I 픽처는 화면 간 예측 없이 화면 내 예측을 통해 부호화/복호화될 수 있다. P 픽처는 단방향(예컨대, 순방향 또는 역방향)에 존재하는 참조 영상만을 이용하는 화면 간 예측을 통해 부호화/복호화될 수 있다. B 픽처는 쌍방향(예컨대, 순방향 및 역방향)에 존재하는 참조 영상들을 이용하는 화면 간 예측을 통해 부호화/복호화 될 수 있다. 또한, B 픽처인 경우, 쌍방향에 존재하는 참조 영상들을 이용하는 화면 간 예측 또는 순방향 및 역방향 중 일 방향에 존재하는 참조 영상을 이용하는 화면 간 예측을 통해 부호화/복호화될 수 있다. 여기에서,

쌍방향은 순방향 및 역방향일 수 있다. 여기서, 화면 간 예측이 사용되는 경우, 부호화기에서는 화면 간 예측 또는 움직임 보상을 수행할 수 있고, 복호화기에서는 그에 대응하는 움직임 보상을 수행할 수 있다.

[234] 아래에서, 실시예에 따른 화면 간 예측에 대해 구체적으로 설명된다.

[235] 화면 간 예측 또는 움직임 보상은 참조 영상 및 움직임 정보를 이용하여 수행될 수 있다.

[236] 현재 블록에 대한 움직임 정보는 부호화 장치(100) 및 복호화 장치(200)의 각각에 의해 화면 간 예측 중 도출될 수 있다. 움직임 정보는 복원된 주변 블록의 움직임 정보, 콜 블록(collocated block; col block)의 움직임 정보 및/또는 콜 블록에 인접한 블록을 이용하여 도출될 수 있다. 콜 블록은 이미 복원된 콜 픽처(collocated picture; col picture) 내에서 현재 블록의 공간적 위치에 대응하는 블록일 수 있다. 여기서, 콜 픽처는 참조 영상 리스트에 포함된 적어도 하나의 참조 영상 중에서 하나의 픽처일 수 있다.

[237] 움직임 정보의 도출 방식은 현재 블록의 예측 모드에 따라 다를 수 있다. 예를 들면, 화면 간 예측을 위해 적용되는 예측 모드로서, AMVP 모드, 머지 모드, 스킵 모드, 움직임 벡터 차분을 가진 머지 모드, 서브 블록 머지 모드, 삼각 분할 모드, 인터 인트라 결합 예측 모드, 어파인 인터 모드 등이 있을 수 있다. 여기서 머지 모드를 움직임 병합 모드(motion merge mode)라고 지칭할 수 있다.

[238] 예를 들면, 예측 모드로서, AMVP가 적용되는 경우, 복원된 주변 블록의 움직임 벡터, 콜 블록의 움직임 벡터, 콜 블록에 인접한 블록의 움직임 벡터, (0, 0) 움직임 벡터 중 적어도 하나를 움직임 벡터 후보로 결정하여 움직임 벡터 후보 리스트(motion vector candidate list)를 생성할 수 있다. 생성된 움직임 벡터 후보 리스트를 이용하여 움직임 벡터 후보를 유도할 수 있다. 유도된 움직임 벡터 후보를 기반으로 현재 블록의 움직임 정보를 결정할 수 있다. 여기서, 콜 블록의 움직임 벡터 또는 콜 블록에 인접한 블록의 움직임 벡터를 시간적 움직임 벡터 후보(temporal motion vector candidate)라 지칭할 수 있고, 복원된 주변 블록의 움직임 벡터를 공간적 움직임 벡터 후보(spatial motion vector candidate)라 지칭할 수 있다.

[239] 부호화 장치(100)는 현재 블록의 움직임 벡터 및 움직임 벡터 후보 간의 움직임 벡터 차분(MVD: Motion Vector Difference)을 계산할 수 있고, MVD를 엔트로피 부호화할 수 있다. 또한, 부호화 장치(100)는 움직임 벡터 후보 색인을 엔트로피 부호화하여 비트스트림을 생성할 수 있다. 움직임 벡터 후보 색인은 움직임 벡터 후보 리스트에 포함된 움직임 벡터 후보 중에서 선택된 최적의 움직임 벡터 후보를 지시할 수 있다. 복호화 장치(200)는 움직임 벡터 후보 색인을 비트스트림으로부터 엔트로피 복호화하고, 엔트로피 복호화된 움직임 벡터 후보 색인을 이용하여 움직임 벡터 후보 리스트에 포함된 움직임 벡터 후보 중에서 복호화 대상 블록의 움직임 벡터 후보를 선택할 수 있다. 또한, 복호화 장치(200)는 엔트로피 복호화된 MVD 및 움직임 벡터 후보의 합을 통해 복호화

대상 블록의 움직임 벡터를 도출할 수 있다.

- [240] 한편, 부호화 장치(100)는 계산된 MVD의 해상도 정보를 엔트로피 부호화할 수 있다. 복호화 장치(200)는 MVD 해상도 정보를 이용하여 엔트로피 복호화된 MVD의 해상도를 조정할 수 있다.
- [241] 한편, 부호화 장치(100)는 어파인 모델에 기반하여 현재 블록의 움직임 벡터 및 움직임 벡터 후보 간의 움직임 벡터 차분(MVD: Motion Vector Difference)을 계산할 수 있고, MVD를 엔트로피 부호화할 수 있다. 복호화 장치(200)는 엔트로피 복호화된 MVD 및 어파인 제어 움직임 벡터 후보의 합을 통해 복호화 대상 블록의 어파인 제어 움직임 벡터를 도출하여 서브 블록 단위로 움직임 벡터를 유도할 수 있다.
- [242] 비트스트림은 참조 영상을 지시하는 참조 영상 색인 등을 포함할 수 있다. 참조 영상 색인은 엔트로피 부호화되어 비트스트림을 통해 부호화 장치(100)로부터 복호화 장치(200)로 시그널링될 수 있다. 복호화 장치(200)는 유도된 움직임 벡터와 참조 영상 색인 정보에 기반하여 복호화 대상 블록에 대한 예측 블록을 생성할 수 있다.
- [243] 움직임 정보의 도출 방식의 다른 예로, 머지 모드가 있다. 머지 모드란 복수의 블록들에 대한 움직임의 병합을 의미할 수 있다. 머지 모드는 현재 블록의 움직임 정보를 주변 블록의 움직임 정보로부터 유도하는 모드를 의미할 수 있다. 머지 모드가 적용되는 경우, 복원된 주변 블록의 움직임 정보 및/또는 콜 블록의 움직임 정보를 이용하여 머지 후보 리스트(merge candidate list)를 생성할 수 있다. 움직임 정보는 1) 움직임 벡터, 2) 참조 영상 색인, 및 3) 화면 간 예측 지시자 중 적어도 하나를 포함할 수 있다. 예측 지시자는 단방향 (L0 예측, L1 예측) 또는 쌍방향일 수 있다.
- [244] 머지 후보 리스트는 움직임 정보들이 저장된 리스트를 나타낼 수 있다. 머지 후보 리스트에 저장되는 움직임 정보는, 현재 블록에 인접한 주변 블록의 움직임 정보(공간적 머지 후보(spatial merge candidate)) 및 참조 영상에서 현재 블록에 대응되는(collocated) 블록의 움직임 정보(시간적 머지 후보(temporal merge candidate)), 이미 머지 후보 리스트에 존재하는 움직임 정보들의 조합에 의해 생성된 새로운 움직임 정보, 현재 블록 이전에 부호화/복호화된 블록의 움직임 정보(히스토리 기반 머지 후보(history-based merge candidate)) 및 제로 머지 후보 중 적어도 하나일 수 있다.
- [245] 부호화 장치(100)는 머지 플래그(merge flag) 및 머지 색인(merge index) 중 적어도 하나를 엔트로피 부호화하여 비트스트림을 생성한 후 복호화 장치(200)로 시그널링할 수 있다. 머지 플래그는 블록 별로 머지 모드를 수행할지 여부를 나타내는 정보일 수 있고, 머지 색인은 현재 블록에 인접한 주변 블록들 중 어떤 블록과 머지를 할 것인가에 대한 정보일 수 있다. 예를 들면, 현재 블록의 주변 블록들은 현재 블록의 좌측 인접 블록, 상단 인접 블록 및 시간적 인접 블록 중 적어도 하나를 포함할 수 있다.

- [246] 한편, 부호화 장치(100)는 머지 후보의 움직임 정보 중 움직임 벡터를 보정하기 위한 보정 정보를 엔트로피 부호화하여 복호화 장치(200)로 시그널링할 수 있다. 복호화 장치(200)는 머지 색인에 의해 선택된 머지 후보의 움직임 벡터를 보정 정보에 기초하여 보정할 수 있다. 여기서, 보정 정보는 보정 여부 정보, 보정 방향 정보 및 보정 크기 정보 중 적어도 하나를 포함할 수 있다. 위와 같이, 시그널링되는 보정 정보를 기초로 머지 후보의 움직임 벡터를 보정하는 예측 모드를 움직임 벡터 차분을 가진 머지 모드로 칭할 수 있다.
- [247] 스킵 모드는 주변 블록의 움직임 정보를 그대로 현재 블록에 적용하는 모드일 수 있다. 스킵 모드가 사용되는 경우, 부호화 장치(100)는 어떤 블록의 움직임 정보를 현재 블록의 움직임 정보로서 이용할 것인지에 대한 정보를 엔트로피 부호화하여 비트스트림을 통해 복호화 장치(200)에 시그널링할 수 있다. 이때, 부호화 장치(100)는 움직임 벡터 차분 정보, 부호화 블록 플래그 및 변환 계수 레벨(양자화된 레벨) 중 적어도 하나에 관한 구문 요소를 복호화 장치(200)에 시그널링하지 않을 수 있다.
- [248] 서브 블록 머지 모드(subblock merge mode)는, 부호화 블록(CU)의 서브 블록 단위로 움직임 정보를 유도하는 모드를 의미할 수 있다. 서브 블록 머지 모드가 적용되는 경우, 참조 영상에서 현재 서브 블록에 대응되는(collocated) 서브 블록의 움직임 정보 (서브블록 기반 시간적 머지 후보(Sub-block based temporal merge candidate)) 및/또는 어파인 제어 포인트 움직임 벡터 머지 후보(affine control point motion vector merge candidate)를 이용하여 서브 블록 머지 후보 리스트(subblock merge candidate list)가 생성될 수 있다.
- [249] 기하 분할 모드(geometric partitioning mode)는, 현재 블록을 소정의 방향으로 분할하여 각각의 움직임 정보를 유도하고, 유도된 각각의 움직임 정보를 이용하여 각각의 예측 샘플을 유도하고, 유도된 각각의 예측 샘플을 가중합하여 현재 블록의 예측 샘플을 유도하는 모드를 의미할 수 있다.
- [250] 인터 인트라 결합 예측 모드는, 화면 간 예측으로 생성된 예측 샘플과 화면 내 예측으로 생성된 예측 샘플을 가중합하여 현재 블록의 예측 샘플을 유도하는 모드를 의미할 수 있다.
- [251] 복호화 장치(200)는 도출된 움직임 정보를 자체적으로 보정할 수 있다. 복호화 장치(200)는 도출된 움직임 정보가 지시하는 참조 블록을 기준으로 기정의된 구역 탐색하여 최소의 SAD를 갖는 움직임 정보를 보정된 움직임 정보로 유도할 수 있다.
- [252] 복호화 장치(200)는 광학적 흐름(Optical Flow)을 이용하여 화면 간 예측을 통해 유도된 예측 샘플을 보상할 수 있다.
- [253] 도 6은 변환 및 양자화의 과정을 설명하기 위한 도면이다.
- [254] 도 6에 도시된 바와 같이 잔여 신호에 변환 및/또는 양자화 과정을 수행하여 양자화된 레벨이 생성될 수 있다. 상기 잔여 신호는 원본 블록과 예측 블록(화면 내 예측 블록 또는 화면 간 예측 블록) 간의 차분으로 생성될 수 있다. 여기에서,

예측 블록은 화면 내 예측 또는 화면 간 예측에 의해 생성된 블록일 수 있다. 여기서, 변환은 1차 변환 및 2차 변환 중 적어도 하나를 포함할 수 있다. 잔여 신호에 대해서 1차 변환을 수행하면 변환 계수가 생성될 수 있고, 변환 계수에 2차 변환을 수행하여 2차 변환 계수를 생성할 수 있다.

- [255] 1차 변환(Primary Transform)은 기-정의된 복수의 변환 방법 중 적어도 하나를 이용하여 수행될 수 있다. 일 예로, 기-정의된 복수의 변환 방법은 DCT(Discrete Cosine Transform), DST(Discrete Sine Transform) 또는 KLT(Karhunen-Loeve Transform) 기반 변환 등을 포함할 수 있다. 1차 변환이 수행 후 생성되는 변환 계수에 2차 변환(Secondary Transform)을 수행할 수 있다. 1차 변환 및/또는 2차 변환시에 적용되는 변환 방법은 현재 블록 및/또는 주변 블록의 부호화 파라미터 중 적어도 하나에 따라 결정될 수 있다. 또는 변환 방법을 지시하는 변환 정보가 시그널링될 수도 있다. DCT 기반 변환은 예컨대, DCT2, DCT-8 등을 포함할 수 있다. DST 기반 변환은 예컨대, DST-7을 포함할 수 있다.

[256]

- [257] 1차 변환 및/또는 2차 변환이 수행된 결과 또는 잔여 신호에 양자화를 수행하여 양자화된 레벨을 생성할 수 있다. 양자화된 레벨은 화면 내 예측 모드 또는 블록 크기/형태 중 적어도 하나를 기준으로 우상단 대각 스캔, 수직 스캔, 수평 스캔 중 적어도 하나에 따라 스캐닝(scanning) 될 수 있다. 예를 들어, 우상단(up-right) 대각 스캐닝을 이용하여 블록의 계수를 스캔함으로써 1차원 벡터 형태로 변경시킬 수 있다. 변환 블록의 크기 및/또는 화면 내 예측 모드에 따라 우상단 대각 스캔 대신 2차원의 블록 형태 계수를 열 방향으로 스캔하는 수직 스캔, 2차원의 블록 형태 계수를 행 방향으로 스캔하는 수평 스캔이 사용될 수도 있다. 스캐닝된 양자화된 레벨은 엔트로피 부호화되어 비트스트림에 포함될 수 있다.

- [258] 복호화기에서는 비트스트림을 엔트로피 복호화하여 양자화된 레벨을 생성할 수 있다. 양자화된 레벨은 역 스캐닝(Inverse Scanning)되어 2차원의 블록 형태로 정렬될 수 있다. 이때, 역 스캐닝의 방법으로 우상단 대각 스캔, 수직 스캔, 수평 스캔 중 적어도 하나가 수행될 수 있다.

- [259] 양자화된 레벨에 역양자화를 수행할 수 있고, 2차 역변환 수행 여부에 따라 2차 역변환을 수행할 수 있고, 2차 역변환이 수행된 결과에 1차 역변환 수행 여부에 따라 1차 역변환을 수행하여 복원된 잔여 신호가 생성될 수 있다.

- [260] 화면 내 예측 또는 화면 간 예측을 통해 복원된 휘도 성분에 대해 인루프 필터링 전에 동적 범위(dynamic range)의 역매핑(inverse mapping)이 수행될 수 있다. 동적 범위는 16개의 균등한 조각(piece)으로 분할될 수 있고, 각 조각에 대한 매핑 함수가 시그널링될 수 있다. 상기 매핑 함수는 슬라이스 레벨 또는 타일 그룹 레벨에서 시그널링될 수 있다. 상기 역매핑을 수행하기 위한 역매핑 함수는 상기 매핑 함수에 기초하여 유도될 수 있다. 인루프 필터링, 참조 픽처의 저장 및 움직임 보상은 역매핑된 영역에서 수행되며, 화면 간 예측을 통해 생성된 예측 블록은 상기 매핑 함수를 이용한 매핑에 의해 매핑된 영역으로 전환된 후, 복원

블록의 생성에 이용될 수 있다. 그러나, 화면 내 예측은 매핑된 영역에서 수행되므로, 화면 내 예측에 의해 생성된 예측 블록은 매핑/역매핑 없이, 복원 블록의 생성에 이용될 수 있다.

- [261] 현재 블록이 색차 성분의 잔차 블록인 경우, 매핑된 영역의 색차 성분에 대해 스케일링을 수행함으로써 상기 잔차 블록은 역매핑된 영역으로 전환될 수 있다. 상기 스케일링의 가용 여부는 슬라이스 레벨 또는 타일 그룹 레벨에서 시그널링될 수 있다. 상기 스케일링은 루마 성분에 대한 상기 매핑이 가용하고 휘도 성분의 분할과 색차 성분의 분할이 동일한 트리 구조를 따르는 경우에만 상기 스케일링이 적용될 수 있다. 상기 스케일링은 상기 색차 블록에 대응하는 휘도 예측 블록의 샘플값의 평균에 기초하여 수행될 수 있다. 이 때, 현재 블록이 화면 간 예측을 사용하는 경우, 상기 휘도 예측 블록은 매핑된 휘도 예측 블록을 의미할 수 있다. 휘도 예측 블록의 샘플값의 평균이 속하는 조각(piece)의 인덱스를 이용하여, 룩업테이블을 참조함으로써, 상기 스케일링에 필요한 값을 유도할 수 있다. 최종적으로 상기 유도된 값을 이용하여 상기 잔차 블록을 스케일링함으로써, 상기 잔차 블록은 역매핑된 영역으로 전환될 수 있다. 이후의 색차 성분 블록의 복원, 화면 내 예측, 화면 간 예측, 인루프 필터링 및 참조 픽처의 저장은 역매핑된 영역에서 수행될 수 있다.
- [262] 상기 휘도 성분 및 색차 성분의 매핑/역매핑이 가용한지 여부를 나타내는 정보는 시퀀스 파라미터 셋을 통해 시그널링될 수 있다.
- [263] 현재 블록의 예측 블록은 현재 블록과 현재 픽처 내 참조 블록 사이의 위치 이동(displacement)을 나타내는 블록 벡터에 기초하여 생성될 수 있다. 이와 같이, 현재 픽처를 참조하여 예측 블록을 생성하는 예측 모드를 화면내 블록 카피(Intra Block Copy, IBC) 모드라고 명명할 수 있다. IBC 모드는 $M \times N$ ($M \leq 64$, $N \leq 64$) 부호화 유닛에 적용될 수 있다. IBC 모드는 스킵 모드, 머지 모드, AMVP 모드 등을 포함할 수 있다. 스킵 모드 또는 머지 모드의 경우, 머지 후보 리스트가 구성되고, 머지 인덱스가 시그널링되어 하나의 머지 후보가 특정될 수 있다. 상기 특정된 머지 후보의 블록 벡터가 현재 블록의 블록 벡터로서 이용될 수 있다. 머지 후보 리스트는 공간적 후보, 히스토리에 기반한 후보, 두개 후보의 평균에 기반한 후보 또는 제로 머지 후보 등 적어도 하나 이상을 포함할 수 있다. AMVP 모드의 경우, 차분 블록 벡터가 시그널링될 수 있다. 또한, 예측 블록 벡터는 현재 블록의 좌측 이웃 블록 및 상단 이웃 블록으로부터 유도될 수 있다. 어느 이웃 블록을 이용할지에 관한 인덱스는 시그널링될 수 있다. IBC 모드의 예측 블록은 현재 CTU 또는 좌측 CTU에 포함되고, 기 복원된 영역내의 블록으로 한정될 수 있다. 예를 들어, 현재 블록의 예측 블록은, 현재 블록이 속한 64×64 블록보다 부호화/복호화 순서상 앞선 3개의 64×64 블록 영역내에 위치하도록 블록 벡터의 값이 제한될 수 있다. 이와 같이 블록 벡터의 값을 제한함으로써, IBC 모드 구현에 따른 메모리 소비와 장치의 복잡도를 경감할 수 있다.

[264]

[265] 이하에서는, 도 8 내지 도 15를 참고하여, 본 발명의 일 실시 예에 따른 부표본 기반의 블록 분류를 이용한 루프내 필터링 방법에 대해 후술한다.

[266] 본 발명에서 루프내 필터링(in-loop filtering)은 디블록킹 필터링(deblocking filtering), 샘플 적응적 오프셋(sample adaptive offset), 양방향 필터링(bilateral filtering), 적응적 루프내 필터링(adaptive in-loop filtering) 중 적어도 하나를 포함할 수 있다.

[267] 화면내/화면간 예측 블록과 복원된 잔여 블록을 합하여 생성된 복원 영상에 상기 디블록킹 필터링과 상기 샘플 적응적 오프셋 중 적어도 하나를 수행하여 복원 영상 내 존재하는 블록킹 현상(blocking artifact)과 링잉 현상(ringing artifact)을 효과적으로 감소시킬 수 있다. 하지만, 디블록킹 필터는 블록의 경계에 대해서 세로 방향 필터링 및 가로 방향 필터링을 수행하여 블록들 사이의 경계에서 발생하는 블록킹 현상을 감소시키는데 목적이 있으나, 디블록킹 필터는 블록 경계 필터링 시에 원본 영상과 복원 영상과의 왜곡을 최소화하지는 못하는 단점이 있다. 또한, 샘플 적응적 오프셋은 링잉 현상 감소를 위해 화소 단위로 인접한 화소들과의 화소 값 비교를 통해 특정 화소에 오프셋을 더해주거나, 화소 값이 특정한 범위에 속하는 화소들에 대해서 오프셋을 더해주는 방법이며, 샘플 적응적 오프셋은 율-왜곡 최적화를 이용하여 원본 영상과 복원 영상과의 왜곡을 일부 최소화하나, 원본 영상과 복원 영상 사이에 왜곡 차이가 클 경우에는 왜곡 최소화 관점에서 한계가 존재한다.

[268] 상기 양방향 필터링은 필터링이 수행되는 영역 내 중심 샘플과 영역 내 다른 샘플들 간의 거리와 샘플들의 값의 차이에 따라 필터 계수가 결정되어 필터링을 수행하는 것을 의미할 수 있다.

[269] 상기 적응적 루프내 필터링은 복원 영상에 원본 영상과 복원 영상 간의 왜곡을 최소화하는 필터를 적용함으로써 원본 영상과 복원 영상의 왜곡을 최소화하는 필터링을 의미할 수 있다.

[270] 본 발명에서 특별한 언급이 없는 한 루프내 필터링은 적응적 루프내 필터링을 의미할 수 있다. 또한, 적응적 루프내 필터는 적응적 루프 필터와 동일한 의미일 수 있다.

[271] 본 발명에서 필터링은 샘플, 블록, CU, PU, TU, CTU, 슬라이스, 타일, 타일 그룹, 영상(픽처), 시퀀스 중 적어도 하나의 단위에 필터를 적용하는 것을 의미할 수 있다. 상기 필터링은 블록 분류, 필터링 수행, 필터 정보 부호화/복호화 중 적어도 하나를 포함할 수 있다.

[272] 본 발명에서 CU(Coding Unit, 부호화 유닛), PU(Prediction Unit, 예측 유닛), TU(Transform Unit, 변환 유닛), CTU(Coding Tree Unit, 부호화 트리 유닛)는 각각 CB(Coding Block, 부호화 블록), PB(Prediction Block, 예측 블록), TB(Transform Block, 변환 블록), CTB(Coding Tree Block, 부호화 트리 블록)와 동일한 의미일 수 있다.

- [273] 본 발명에서 블록은 CU, PU, TU, CB, PB, TB 등 부호화/복호화 과정 내에서 사용되는 블록 또는 유닛 단위 중 적어도 하나를 의미할 수 있다.
- [274] 상기 루프내 필터링은 양방향 필터링, 디블록킹 필터링, 샘플 적응적 오프셋, 적응적 루프내 필터링 순서대로 복원 영상에 적용되어 복호 영상을 생성할 수 있다. 하지만, 루프내 필터링에 포함되는 필터링들 간 순서를 변경하여 복원 영상에 적용하여 복호 영상을 생성할 수 있다.
- [275] 일 예로, 디블록킹 필터링, 샘플 적응적 오프셋, 적응적 루프내 필터링 순서대로 복원 영상에 루프내 필터링을 적용할 수 있다.
- [276] 다른 예로, 양방향 필터링, 적응적 루프내 필터링, 디블록킹 필터링, 샘플 적응적 오프셋 순서대로 복원 영상에 루프내 필터링을 적용할 수 있다.
- [277] 또 다른 예로, 적응적 루프내 필터링, 디블록킹 필터링, 샘플 적응적 오프셋 순서대로 복원 영상에 루프내 필터링을 적용할 수 있다.
- [278] 또 다른 예로, 적응적 루프내 필터링, 샘플 적응적 오프셋, 디블록킹 필터링 순서대로 복원 영상에 루프내 필터링을 적용할 수 있다.
- [279] 본 발명에서 복호 영상은 화면내 예측 블록 또는 화면간 예측 블록에 복원된 잔여 블록을 더하여 생성되는 복원된 블록으로 구성된 복원 영상에 루프내 필터링 또는 후처리 필터링을 수행한 결과 영상을 의미할 수 있다. 또한, 본 발명에서 복호 샘플/블록/CTU/영상 등과 복원 샘플/블록/CTU/영상 등의 의미는 서로 다르지 않고 동일한 것을 의미할 수도 있다.
- [280] 상기 적응적 루프내 필터링은 복원 영상에 수행되어 복호 영상을 생성할 수 있다. 또는 적응적 루프내 필터링은 디블록킹 필터링, 샘플 적응적 오프셋, 양방향 필터링 중 적어도 하나 이상이 수행된 복호 영상에 수행될 수 있다. 또한, 적응적 루프내 필터링은 적응적 루프내 필터링이 수행된 복호 영상에 수행될 수 있다. 이 경우, 적응적 루프내 필터링을 복원 또는 복호 영상에 N차례 반복적으로 수행할 수 있다. 이때, N은 양의 정수이다.
- [281] 상기 루프내 필터링 방법 중 적어도 하나의 필터링이 수행된 복호 영상에 루프내 필터링을 수행할 수 있다. 예를 들어, 루프내 필터링 방법 중 적어도 하나가 수행된 복호 영상에 루프내 필터링 방법 중 적어도 다른 하나의 필터링이 수행될 경우, 상기 루프내 필터링 방법 중 적어도 다른 하나에 사용되는 파라미터가 변경되고, 파라미터가 변경된 루프내 필터링 방법이 상기 복호 영상에 수행될 수 있다. 이때, 상기 파라미터는 부호화 파라미터, 필터 계수, 필터 탭 수(필터 길이), 필터 형태, 필터 종류, 필터링 수행 횟수, 필터 강도, 임계값 중 적어도 하나를 포함할 수 있다.
- [282] 상기 필터 계수는 필터를 구성하는 계수를 의미할 수 있고, 복원 샘플에 곱해지는 마스크(mask) 형태에서 특정 마스크 위치에 해당하는 계수 값을 의미할 수 있다.
- [283] 상기 필터 탭 수는 필터의 길이를 의미하고, 필터가 특정 1가지 방향에 대해 대칭적(symmetric) 특성을 가지면 실제 부호화/복호화 대상이 되는 필터 계수는

반으로 줄어 들 수 있다. 또한, 상기 필터 탭은 가로 방향에서 필터의 넓이를 의미하거나, 세로 방향에서 필터의 높이를 의미할 수 있다. 또한, 상기 필터 탭은 2차원 필터 형태에서 가로 방향으로 필터의 넓이와 세로 방향으로 필터의 높이를 함께 의미할 수 있다. 또한, 필터는 특정 2가지 이상 방향에 대해 대칭적 특징을 가질 수 있다.

- [284] 상기 필터 형태는 필터가 마스크 형태를 가질 경우, 필터 형태는 마름모(diamond/rhombus), 직사각형(non-square), 정사각형(square), 사다리꼴(trapezoid), 대각(diagonal), 눈송이(snowflake), 샵(sharp), 클로버(clover), 십자가(cross), 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 등 2차원으로 표현될 수 있는 기하학적 도형 또는 이들의 결합된 형태를 의미할 수 있다. 또는, 필터 형태는 3차원 필터를 2차원으로 투사(projection)시킨 형태일 수 있다.
- [285] 상기 필터 종류는 위너 필터(Wiener filter), 저역 통과 필터(low-pass filter), 고역 통과 필터(high-pass filter), 선형 필터(linear filter), 비선형 필터(non-linear filter), 양방향 필터(bilateral filter) 등의 필터를 의미할 수 있다.
- [286] 본 발명에서는 상기 다양한 필터 종류 중 위너 필터를 위주로 설명하나, 이에 한정되는 것은 아니고 상기 다양한 필터 종류에 대해서 본 발명의 실시예 또는 실시예의 조합이 적용될 수 있다.
- [287] 적응적 루프내 필터링의 필터 종류로 위너 필터(Wiener filter)를 이용할 수 있다. 위너 필터는 최적의 선형 필터(optimal linear filter)로 영상 내 잡음(noise), 블러링(blurring), 왜곡(distortion)을 효과적으로 제거하여 부호화 효율을 향상하는데 그 목적이 있다. 이때, 위너 필터는 원본 영상과 복원/복호 영상 사이의 왜곡을 최소화하도록 설계될 수 있다.
- [288] 상기 필터링 방법 중 적어도 하나는 부호화 과정 또는 복호화 과정 내에서 수행될 수 있다. 이때, 부호화 과정 또는 복호화 과정은 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 단위 중 적어도 하나의 단위로 부호화 과정 또는 복호화 과정이 수행되는 것을 의미할 수 있다. 상기 필터링 방법 중 적어도 하나는 슬라이스, 타일, 타일 그룹, 픽처 등의 단위로 부호화 과정 또는 복호화 과정 내에서 수행될 수 있다. 예를 들어, 위너 필터는 적응적 루프내 필터링 형태로 부호화 과정 또는 복호화 과정 내에서 수행될 수 있다. 즉, 적응적 루프내 필터링에서 루프내의 의미는 부호화 과정 또는 복호화 과정 내에서 필터링이 수행된다는 의미이다. 적응적 루프내 필터링이 수행될 경우, 적응적 루프내 필터링이 수행된 복호 영상은 차후 부호화/복호화되는 영상의 참조 영상으로 사용될 수 있다. 이때, 차후 부호화/복호화되는 영상에서는 적응적 루프내 필터링이 수행된 복호 영상을 참조하여 화면간 예측 또는 움직임 보상을 수행할 수 있으므로 적응적 루프내 필터링이 수행된 복호 영상의 부호화 효율이 향상될 뿐만 아니라, 상기 차후 부호화/복호화되는 영상의 부호화 효율도 향상될 수 있다.
- [289] 또한, 상기 필터링 방법 중 적어도 하나는 CTU 단위 또는 블록 단위의 부호화

과정 또는 복호화 과정 내에서 수행될 수 있다. 예를 들어, 위너 필터는 적응적 루프내 필터링 형태로 CTU 단위 또는 블록 단위의 부호화 과정 또는 복호화 과정 내에서 수행될 수 있다. 즉, 적응적 루프내 필터링에서 루프내의 의미는 CTU 단위 또는 블록 단위의 부호화 과정 또는 복호화 과정 내에서 필터링이 수행된다는 의미이다. CTU 단위 또는 블록 단위에서 적응적 루프내 필터링이 수행될 경우, 적응적 루프내 필터링이 수행된 복호 CTU 또는 복호 블록은 차후 부호화/복호화되는 CTU 또는 블록의 참조 CTU 또는 참조 블록으로 사용될 수도 있다. 이때, 차후 부호화/복호화되는 CTU 또는 블록에서는 적응적 루프내 필터링이 수행된 복호 CTU 또는 복호 블록을 참조하여 화면내 예측 등을 수행할 수 있으므로 적응적 루프내 필터링이 수행된 복호 CTU 또는 복호 블록의 부호화 효율이 향상될 뿐만 아니라, 상기 차후 부호화/복호화되는 CTU 또는 블록의 부호화 효율도 향상될 수 있다.

- [290] 또한, 상기 필터링 방법 중 적어도 하나는 후처리 필터(post filter) 형태로 복호화 과정 후에 수행될 수 있다. 예를 들어, 위너 필터는 후처리 필터 형태로 복호화 과정 후에 수행될 수 있다. 위너 필터가 복호화 과정 후에 수행될 경우, 영상 출력(디스플레이) 전에 복원/복호 영상에 위너 필터가 수행될 수 있다. 후처리 필터링이 수행될 경우, 후처리 필터링이 수행된 복호 영상은 차후 부호화/복호화되는 영상의 참조 영상으로 사용될 수 없다.
- [291] 적응적 루프내 필터링은 블록 기반의 필터 적응을 사용할 수 있다. 여기서, 블록 기반의 필터 적응이란 다수 개의 필터 중에 블록 별로 사용되는 필터가 적응적으로 선택된다는 것을 의미할 수 있다. 상기 블록 기반의 필터 적응은 블록 분류를 의미할 수 있다.
- [292] 도 8a는 본 발명의 일 실시 예에 따른 영상 복호화 방법을 나타내는 흐름도이다.
- [293] 도 8a를 참조하면, 복호화기는 부호화 유닛에 대한 필터 정보를 복호화할 수 있다(S701).
- [294] 상기 필터 정보는 부호화 유닛에 대해서만 한정되는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등에 대한 필터 정보를 의미할 수 있다.
- [295] 한편, 상기 필터 정보는, 필터링 수행 여부, 필터 계수 값, 필터 개수, 필터 탭 수(필터 길이) 정보, 필터 형태 정보, 필터 종류 정보, 블록 분류 색인에 대한 고정된 필터 사용 여부 정보 및 필터의 대칭 형태에 관한 정보 중 적어도 하나를 포함할 수 있다.
- [296] 한편, 상기 필터 형태 정보는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 형태 중 적어도 하나의 정보를 포함할 수 있다.
- [297] 한편, 상기 필터 계수 값은 블록 분류 단위로 분류된 블록에 기반하여 기하학적 변환된 필터 계수 값을 포함할 수 있다.
- [298] 한편, 필터의 대칭 형태에 관한 정보는 점 대칭, 가로 대칭, 세로 대칭 및 대각

대칭 형태 중 적어도 하나 이상의 정보를 포함할 수 있다.

- [299] 또한, 복호화기는 상기 부호화 유닛을 블록 분류 단위로 분류할 수 있다(S702). 또한, 복호화기는 상기 블록 분류 단위로 분류된 부호화 유닛에 블록 분류 색인을 할당할 수 있다.
- [300] 상기 블록 분류는 부호화 유닛 단위로 수행되는 것만 한정하는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등의 단위에서 수행될 수 있다.
- [301] 한편, 상기 블록 분류 색인은 방향성 정보와 활동성 정보를 이용하여 결정될 수 있다.
- [302] 한편, 상기 방향성 정보 및 활동성 정보 중 적어도 하나는 세로, 가로, 제1 대각, 제2 대각 방향 중 적어도 하나에 대한 기울기(gradient) 값에 기반하여 결정될 수 있다.
- [303] 한편, 상기 기울기 값은 상기 블록 분류 단위에서의 1D 라플라시안 연산을 이용하여 획득될 수 있다.
- [304] 한편, 상기 1D 라플라시안 연산은 상기 연산의 수행 위치가 서브샘플링된 1D 라플라시안 연산일 수 있다.
- [305] 한편, 상기 기울기 값은 시간적 계층 식별자(temporal layer identifier)에 기반하여 결정될 수 있다.
- [306] 또한, 복호화기는 상기 필터 정보를 이용하여 상기 블록 분류 단위로 분류된 부호화 유닛을 필터링할 수 있다(S703).
- [307] 상기 필터링의 대상은 부호화 유닛으로 한정되는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등에서 필터링이 수행될 수 있다.
- [308] 도 8b는 본 발명의 일 실시 예에 따른 영상 부호화 방법을 나타내는 흐름도이다.
- [309] 도 8b를 참조하면, 부호화기는 부호화 유닛을 블록 분류 단위로 분류할 수 있다(S801). 또한, 부호화기는 상기 블록 분류 단위로 분류된 부호화 유닛에 블록 분류 색인을 할당할 수 있다.
- [310] 상기 블록 분류는 부호화 유닛 단위로 수행되는 것만 한정하는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등의 단위에서 수행될 수 있다.
- [311] 한편, 상기 블록 분류 색인은 방향성 정보와 활동성 정보를 이용하여 결정될 수 있다.
- [312] 한편, 상기 방향성 정보 및 활동성 정보 중 적어도 하나는 세로, 가로, 제1 대각, 제2 대각 방향 중 적어도 하나에 대한 기울기 값에 기반하여 결정될 수 있다.
- [313] 한편, 상기 기울기 값은 상기 블록 분류 단위에서의 1D 라플라시안 연산을 이용하여 획득될 수 있다.
- [314] 한편, 상기 1D 라플라시안 연산은 상기 연산의 수행 위치가 서브샘플링된 1D 라플라시안 연산일 수 있다.

- [315] 한편, 상기 기울기 값은 시간적 계층 식별자에 기반하여 결정될 수 있다.
- [316] 또한, 부호화기는 상기 부호화 유닛에 대한 필터 정보를 이용하여 상기 블록 분류 단위로 분류된 부호화 유닛을 필터링할 수 있다(S802).
- [317] 상기 필터링의 대상은 부호화 유닛으로 한정되는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등에서 필터링이 수행될 수 있다.
- [318] 한편, 상기 필터 정보는, 필터링 수행 여부, 필터 계수 값, 필터 개수, 필터 탭 수(필터 길이) 정보, 필터 형태 정보, 필터 종류 정보, 블록 분류 색인에 대한 고정된 필터 사용 여부 정보 및 필터의 대칭 형태에 관한 정보 중 적어도 하나를 포함할 수 있다.
- [319] 한편, 상기 필터 형태 정보는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 형태 중 적어도 하나의 정보를 포함할 수 있다.
- [320] 한편, 상기 필터 계수 값은 블록 분류 단위로 분류된 블록에 기반하여 기하학적 변환된 필터 계수 값을 포함할 수 있다.
- [321] 또한, 부호화기는 상기 필터 정보를 부호화할 수 있다(S803).
- [322] 상기 필터 정보는 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등에 대한 필터 정보를 의미할 수 있다.
- [323] 부호화기에서는 적응적 루프내 필터링은 블록 분류(block classification) 단계, 필터링(filtering) 수행 단계, 필터 정보(filtering information) 부호화 단계로 구성될 수 있다.
- [324] 한편, 복호화기에서는 필터 정보 복호화 단계, 블록 분류 단계, 필터링 수행 단계 등을 포함할 수 있다.
- [325] 본 발명에서 중복되는 설명을 피하기 위해, 필터 정보 부호화 단계와 필터 정보 복호화 단계를 합하여 필터 정보 부호화/복호화 단계로 후술한다.
- [326] 이하에서는, 블록 분류 단계에 대해서 후술한다.
- [327] 복원 영상 내 NxM 블록 단위(또는, 블록 분류 단위)로 블록 분류 색인(block classification index)을 할당하여, 각 블록을 L개만큼 분류할 수 있다. 여기서, 복원/복호 영상 뿐만 아니라, 복원/복호 슬라이스, 복원/복호 타일 그룹, 복원/복호 타일, 복원/복호 CTU, 복원/복호 블록 등 중 적어도 하나에 대해 블록 분류 색인을 할당할 수 있다.
- [328] 여기서, N, M, L은 양의 정수일 수 있다. 예를 들어, N과 M은 각각 2, 4, 8, 16, 32 중 하나일 수 있고, L은 4, 8, 16, 20, 25, 32 중 하나일 수 있다. 만약, N과 M이 각각 1이면 블록 단위로 블록 분류가 아닌 샘플 단위의 샘플 분류가 수행될 수 있다. 또한, N과 M이 서로 다른 양의 정수 값을 가지면, NxM 블록 단위가 비정방형 형태를 가질 수 있다. 또한, N과 M이 서로 같은 양의 정수 값을 가질 수 있다.
- [329] 예를 들어, 복원 영상을 2x2 블록 단위로 총 25가지의 블록 분류 색인을 할당할 수 있다. 예를 들어, 복원 영상을 4x4 블록 단위로 총 25가지의 블록 분류 색인을

할당할 수 있다.

[330] 상기 블록 분류 색인은 0부터 L-1의 값을 가질 수 있고, 또는, 1부터 L의 값을 가질 수도 있다.

[331] 상기 블록 분류 색인 C는 방향성(directionality) D 값과 활동성(activity) A 값을 양자화 한 Aq 값 중 적어도 하나를 기반으로 하며, 수학식 1과 같이 나타낼 수 있다.

[332] [수학식 1]

$$[333] \quad \mathbf{C} = 5\mathbf{D} + \mathbf{A}_q$$

[334] 수학식 1에서 상수 값 5는 일 예이며, J 값을 사용할 수 있다. 이때, J는 L보다 작은 값을 가지는 양의 정수일 수 있다.

[335] 본 발명의 일 실시 예에 따를 때, 2x2 블록 분류의 경우, 1D 라플라시안(Laplacian) 연산을 이용하여 세로, 가로, 제1 대각(135도 대각), 제2 대각(45도 대각) 방향에 따른 각각의 기울기(gradient) 값의 합 gv, gh, gd1, gd2는 수학식 2 내지 수학식 5와 같이 나타낼 수 있다. 상기 기울기 값의 합을 이용하여 상기 D 값과 A 값을 도출할 수 있다. 상기 기울기 값의 합은 일 실시 예이며, 기울기 값의 통계 값이 기울기 값의 합을 대체하여 사용될 수 있다.

[336] [수학식 2]

$$[337] \quad g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|$$

[338] [수학식 3]

$$[339] \quad g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|$$

[340] [수학식 4]

$$[341] \quad g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)|$$

[342] [수학식 5]

$$[343] \quad g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$$

[344] 수학식 2 내지 수학식 5에서, i와 j 각각은 2x2 블록 내 좌상단 가로 위치와 세로 위치를 의미하고, R(i,j)는 (i,j) 위치에서의 복원된 샘플 값을 의미할 수 있다.

[345] 또한, 수학식 2 내지 수학식 5에서, k와 l 각각은 방향에 따른 샘플 단위 1D 라플라시안 연산의 결과 V_{k,l}, H_{k,l}, D1_{k,l}, D2_{k,l}의 합이 계산되는 가로 범위와 세로 범위를 의미할 수 있다. 상기 각각의 방향에 따른 샘플 단위 1D 라플라시안 연산의 결과는 각각의 방향에 따른 샘플 단위 기울기(gradient) 값을 의미할 수 있다. 즉, 1D 라플라시안 연산의 결과는 기울기 값을 의미할 수 있다. 상기 1D 라플라시안 연산은 세로, 가로, 제1 대각, 제2 대각 방향 각각에 대해 수행될 수 있으며, 해당 방향에 대한 기울기 값을 나타낼 수 있다. 또한, 상기 1D

라플라시안 연산은 세로, 가로, 제1 대각, 제2 대각 방향 각각에 대해 수행된 결과가 각각 $V_{k,l}$, $H_{k,l}$, $D1_{k,l}$, $D2_{k,l}$ 를 의미할 수 있다.

- [346] 일 예로, k 와 l 은 서로 같은 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 같을 수 있다.
- [347] 다른 예로, k 와 l 은 서로 다른 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 다를 수 있다.
- [348] 또 다른 예로, k 는 $i-2$ 에서 $i+3$ 의 범위, l 은 $j-2$ 에서 $j+3$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 6×6 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다.
- [349] 또 다른 예로, k 는 $i-1$ 에서 $i+2$ 의 범위, l 은 $j-1$ 에서 $j+2$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 4×4 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다.
- [350] 또 다른 예로, k 는 i 에서 $i+1$ 의 범위, l 은 j 에서 $j+1$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 2×2 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기와 동일할 수 있다.
- [351] 또 다른 예로, 1D 라플라시안 연산의 결과의 합이 계산되는 범위는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.
- [352] 또 다른 예로, 블록 분류 단위의 형태는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.
- [353] 또 다른 예로, 상기 1D 라플라시안 연산의 합이 계산되는 범위는 $S \times T$ 의 크기를 가질 수 있다. 이때, S 와 T 는 0을 포함한 양의 정수일 수 있다.
- [354] 한편, 상기 제1 대각을 나타내는 $D1$ 과 상기 제2 대각을 나타내는 $D2$ 는 상기 제1 대각을 나타내는 $D0$ 와 상기 제2 대각을 나타내는 $D1$ 을 의미할 수도 있다.
- [355] 본 발명의 일 실시 예에 따를 때, 4×4 블록 분류의 경우, 1D 라플라시안 연산을 이용하여 세로, 가로, 제1 대각, 제2 대각 방향에 따른 각각의 기울기 값의 합 g_v , g_h , $gd1$, $gd2$ 는 수학식 6 내지 수학식 9와 같이 나타낼 수 있다. 상기 기울기 값의 합을 이용하여 상기 D 값과 A 값을 도출할 수 있다. 상기 기울기 값의 합은 일 실시예이며, 기울기 값의 통계 값이 기울기 값의 합을 대체하여 사용될 수 있다.
- [356] [수학식 6]
- [357]
$$g_v = \sum_{k=i-2}^{i+5} \sum_{l=j-2}^{j+5} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|$$
- [358] [수학식 7]

- [359] $g_h = \sum_{k=i-2}^{i+5} \sum_{l=j-2}^{j+5} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|$
- [360] [수학식 8]
- [361] $g_{d1} = \sum_{k=i-2}^{i+5} \sum_{l=j-2}^{j+5} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)|$
- [362] [수학식 9]
- [363] $g_{d2} = \sum_{k=i-2}^{i+5} \sum_{l=j-2}^{j+5} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$
- [364] 수학식 6 내지 수학식 9에서, i와 j 각각은 4x4 블록 내 좌상단 가로 위치와 세로 위치를 의미하고, R(i,j)는 (i,j) 위치에서의 복원된 샘플 값을 의미할 수 있다.
- [365] 또한, 수학식 6 내지 수학식 9에서, k와 l 각각은 방향에 따른 샘플 단위 1D 라플라시안 연산의 결과 $V_{k,l}$, $H_{k,l}$, $D1_{k,l}$, $D2_{k,l}$ 의 합이 계산되는 가로 범위와 세로 범위를 의미할 수 있다. 상기 각각의 방향에 따른 샘플 단위 1D 라플라시안 연산의 결과는 각각의 방향에 따른 샘플 단위 기울기(gradient) 값을 의미할 수 있다. 즉, 1D 라플라시안 연산의 결과는 기울기 값을 의미할 수 있다. 상기 1D 라플라시안 연산은 세로, 가로, 제1 대각, 제2 대각 방향 각각에 대해 수행될 수 있으며, 해당 방향에 대한 기울기 값을 나타낼 수 있다. 또한, 상기 1D 라플라시안 연산은 세로, 가로, 제1 대각, 제2 대각 방향 각각에 대해 수행된 결과가 각각 $V_{k,l}$, $H_{k,l}$, $D1_{k,l}$, $D2_{k,l}$ 를 의미할 수 있다.
- [366] 일 예로, k와 l은 서로 같은 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 같을 수 있다.
- [367] 다른 예로, k와 l은 서로 다른 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 다를 수 있다.
- [368] 또 다른 예로, k는 i-2에서 i+5의 범위, l은 j-2에서 j+5의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 8x8 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다.
- [369] 또 다른 예로, k는 i에서 i+3의 범위, l은 j에서 j+3의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 4x4 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기와 동일할 수 있다.
- [370] 또 다른 예로, 1D 라플라시안 연산의 결과의 합이 계산되는 범위는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.
- [371] 또 다른 예로, 상기 1D 라플라시안 연산의 합이 계산되는 범위는 SxT의 크기를 가질 수 있다. 이때, S와 T는 0을 포함한 양의 정수일 수 있다.
- [372] 또 다른 예로, 블록 분류 단위의 형태는 마름모, 직사각형, 정사각형, 사다리꼴,

대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.

- [373] 본 발명의 일 실시 예에 따를 때, 4x4 블록 분류의 경우, 1D 라플라시안 연산을 이용하여 세로, 가로, 제1 대각, 제2 대각 방향에 따른 각각의 기울기 값의 합 g_v , g_h , g_{d1} , g_{d2} 를 수학식 10 내지 수학식 13과 같이 나타낼 수 있다. 상기 기울기 값은 블록 분류의 계산 복잡도를 감소시키기 위해 부표본(또는, 서브샘플링(subsampling))에 기반하여 나타낼 수 있다. 상기 기울기 값의 합을 이용하여 상기 D 값과 A 값을 도출할 수 있다. 상기 기울기 값의 합은 일 실시예이며, 기울기 값의 통계 값이 기울기 값의 합을 대체하여 사용될 수 있다.

[374] [수학식 10]

$$[375] \quad g_v = \sum_k \sum_l V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|, \quad k = i-2, i, i+2, i+4, l = j-2, \dots, j+5$$

[376] [수학식 11]

$$[377] \quad g_h = \sum_k \sum_l H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|, \quad k = i-2, \dots, i+5, l = j-2, j, j+2, j+4$$

[378] [수학식 12]

$$[379] \quad g_{d1} = \sum_k \sum_l m_{k,l} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)|, \quad k = i-2, \dots, i+5, l = j-2, \dots, j+5$$

$$m_{k,l} = \begin{cases} 1 & \text{if both } k \text{ and } l \text{ are even} \\ 1 & \text{if both } k \text{ and } l \text{ are odd} \\ 0 & \text{otherwise} \end{cases}$$

[380] [수학식 13]

$$[381] \quad g_{d2} = \sum_k \sum_l n_{k,l} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|, \quad k = i-2, \dots, i+5, l = j-2, \dots, j+5$$

$$n_{k,l} = \begin{cases} 1 & \text{if } k \text{ is even and } l \text{ is odd} \\ 1 & \text{if } k \text{ is odd and } l \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

- [382] 수학식 10 내지 수학식 13에서, i와 j 각각은 4x4 블록 내 좌상단 가로 위치와 세로 위치를 의미하고, R(i,j)는 (i,j) 위치에서의 복원된 샘플 값을 의미할 수 있다.

- [383] 또한, 수학식 10 내지 수학식 13에서, k와 l 각각은 방향에 따른 샘플 단위 1D 라플라시안 연산의 결과 $V_{k,l}$, $H_{k,l}$, $D1_{k,l}$, $D2_{k,l}$ 의 합이 계산되는 가로 범위와 세로 범위를 의미할 수 있다. 상기 각각의 방향에 따른 샘플 단위 1D 라플라시안 연산의 결과는 각각의 방향에 따른 샘플 단위 기울기(gradient) 값을 의미할 수 있다. 즉, 1D 라플라시안 연산의 결과는 기울기 값을 의미할 수 있다. 상기 1D 라플라시안 연산은 세로, 가로, 제1 대각, 제2 대각 방향 각각에 대해 수행될 수 있으며, 해당 방향에 대한 기울기 값을 나타낼 수 있다. 또한, 상기 1D 라플라시안 연산은 세로, 가로, 제1 대각, 제2 대각 방향 각각에 대해 수행된 결과가 각각 $V_{k,l}$, $H_{k,l}$, $D1_{k,l}$, $D2_{k,l}$ 를 의미할 수 있다.

- [384] 일 예로, k와 l은 서로 같은 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 같을 수 있다.

- [385] 다른 예로, k와 l은 서로 다른 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 다를 수 있다.

- [386] 또 다른 예로, k는 i-2에서 i+5의 범위, l은 j-2에서 j+5의 범위를 가져서, 1D

라플라시안 연산의 합이 계산되는 범위가 8×8 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다.

- [387] 또 다른 예로, k 는 i 에서 $i+3$ 의 범위, l 은 j 에서 $j+3$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 4×4 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기와 동일할 수 있다.
- [388] 또 다른 예로, 1D 라플라시안 연산의 결과의 합이 계산되는 범위는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.
- [389] 또 다른 예로, 상기 1D 라플라시안 연산의 합이 계산되는 범위는 $S \times T$ 의 크기를 가질 수 있다. 이때, S 와 T 는 0을 포함한 양의 정수일 수 있다.
- [390] 또 다른 예로, 블록 분류 단위의 형태는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.
- [391] 본 발명의 일 실시 예에 따를 때, 상기 부표본에 기반하여 기울기 값을 계산하는 방법은 계산하고자 하는 기울기 값의 방향에 따라, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 해당 방향에 위치한 샘플에 1D 라플라시안 연산을 수행하여 기울기 값을 계산할 수 있다. 여기서, 1D 라플라시안 연산의 합이 계산되는 범위 내 샘플들 중 적어도 하나에서 수행된 1D 라플라시안 연산 결과에 대해 통계 값을 계산하여 기울기 값의 통계 값을 산출할 수 있다. 이때, 통계값은 합(sum), 가중합 또는 평균값 중 어느 하나일 수 있다.
- [392] 일 예로, 가로 방향 기울기 값의 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 행 내 모든 샘플의 위치에서는 1D 라플라시안 연산을 수행하되, 1D 라플라시안 연산의 합이 계산되는 범위 내 각 행을 P 행씩 건너뛰면서 가로 방향 기울기 값을 계산할 수 있다. 여기서, P 는 양의 정수이다.
- [393] 다른 예로, 세로 방향 기울기 값의 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 열 내 모든 샘플의 위치에서는 1D 라플라시안 연산을 수행하되, 1D 라플라시안 연산의 합이 계산되는 범위 내 각 열을 P 열씩 건너뛰면서 세로 방향 기울기 값을 계산할 수 있다. 여기서, P 는 양의 정수이다.
- [394] 또 다른 예로, 제1 대각 방향 기울기 값 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 가로 방향 및 세로 방향 중 적어도 하나의 방향에 대해 P 열 및 Q 행 중 적어도 하나에 대해 건너뛰면서 존재하는 샘플의 위치에서 1D 라플라시안 연산을 수행하여 제1 대각 방향 기울기 값을 계산할 수 있다. 여기서, P 와 Q 는 0을 포함한 양의 정수이다.
- [395] 또 다른 예로, 제2 대각 방향 기울기 값 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 가로 방향 및 세로 방향 중 적어도 하나의 방향에 대해 P 열

및 Q행 중 적어도 하나에 대해 건너뛰면서 존재하는 샘플의 위치에서 1D 라플라시안 연산을 수행하여 제2 대각 방향 기울기 값을 계산할 수 있다. 여기서, P와 Q는 0을 포함한 양의 정수이다.

[396] 본 발명의 일 실시 예에 따를 때, 상기 부표본에 기반하여 기울기 값을 계산하는 방법은 1D 라플라시안 연산의 합이 계산되는 범위 내에 위치한 적어도 하나의 샘플에 1D 라플라시안 연산을 수행하여 기울기 값을 계산할 수 있다. 여기서, 1D 라플라시안 연산의 합이 계산되는 범위 내 샘플들 중 적어도 하나에서 수행된 1D 라플라시안 연산 결과에 대해 통계 값을 계산하여 기울기 값의 통계 값을 산출할 수 있다. 이때, 통계값은 합(sum), 가중합 또는 평균값 중 어느 하나일 수 있다.

[397] 일 예로, 기울기 값 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 행 내 모든 샘플의 위치에서는 1D 라플라시안 연산을 수행하되, 1D 라플라시안 연산의 합이 계산되는 범위 내 각 행을 P행씩 건너뛰면서 기울기 값을 계산할 수 있다. 여기서, P는 양의 정수이다.

[398] 다른 예로, 기울기 값 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 열 내 모든 샘플의 위치에서는 1D 라플라시안 연산을 수행하되, 1D 라플라시안 연산의 합이 계산되는 범위 내 각 열을 P열씩 건너뛰면서 기울기 값을 계산할 수 있다. 여기서, P는 양의 정수이다.

[399] 또 다른 예로, 기울기 값 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 가로 방향 및 세로 방향 중 적어도 하나의 방향에 대해 P열 및 Q행 중 적어도 하나에 대해 건너뛰면서 존재하는 샘플의 위치에서 1D 라플라시안 연산을 수행하여 기울기 값을 계산할 수 있다. 여기서, P와 Q는 0을 포함한 양의 정수이다.

[400] 또 다른 예로, 기울기 값 계산 시, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 가로 방향 및 세로 방향에 대해 P열 및 Q행에 대해 건너뛰면서 존재하는 샘플의 위치에서 1D 라플라시안 연산을 수행하여 기울기 값을 계산할 수 있다. 여기서, P와 Q는 0을 포함한 양의 정수이다.

[401] 한편, 상기 기울기는 가로 방향 기울기, 세로 방향 기울기, 제1 대각 방향 기울기, 제2 대각 방향 기울기 중 적어도 하나를 포함할 수 있다.

[402] 도 9는 부표본에 기반하여 가로, 세로, 제1 대각, 제2 대각 방향에 따른 기울기 값을 결정하는 일 예이다.

[403] 도 9의 예와 같이, 4x4 블록 분류의 경우, 부표본에 기반하여 세로, 가로, 제1 대각, 제2 대각 방향에 따른 각각의 기울기 값의 합 gv, gh, gd1, gd2 들 중 적어도 하나를 계산할 수 있다. 여기서, V, H, D1, D2는 샘플 단위에서 세로, 가로, 제1 대각, 제2 대각 방향에 따른 각각의 1D 라플라시안 연산의 결과를 나타낸다. 즉, V, H, D1, D2 위치에서 각각의 세로, 가로, 제1 대각, 제2 대각 방향에 따른 1D 라플라시안 연산이 수행될 수 있다. 또한, 1D 라플라시안 연산이 수행되는 위치가 서브샘플링될 수 있다. 도 11에서, 블록 분류 색인 C는 음영이 표시된 4x4

블록 단위로 할당될 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다. 여기서, 가는 실선에 포함된 사각형은 각 복원 샘플 위치를 나타내고, 굵은 실선은 1D 라플라시안 연산의 합이 계산되는 범위를 나타낸다.

- [404] 본 발명의 일 실시 예에 따를 때, 부표본에 기반하여 $N \times M$ 블록 단위 내 특정 위치에 존재하는 샘플에 대해 1D 라플라시안 연산을 수행하여 기울기 값을 계산할 수 있다. 이때, 특정 위치는 블록 내에서 절대적인 위치, 상대적인 위치 중 적어도 하나일 수 있다. 여기서, 1D 라플라시안 연산의 합이 계산되는 범위 내 샘플들 중 적어도 하나에서 수행된 1D 라플라시안 연산 결과에 대해 통계 값을 계산하여 기울기 값의 통계 값을 산출할 수 있다. 이때, 통계값은 합(sum), 가중합 또는 평균값 중 어느 하나일 수 있다.
- [405] 일 예로, 절대적인 위치인 경우, $N \times M$ 블록 내 좌상단 위치를 의미할 수 있다.
- [406] 다른 예로, 절대적인 위치인 경우, $N \times M$ 블록 내 우하단 위치를 의미할 수 있다.
- [407] 또 다른 예로, 상대적인 위치인 경우, $N \times M$ 블록 내 가운데 위치를 의미할 수 있다.
- [408] 본 발명의 일 실시 예에 따를 때, 부표본에 기반하여 $N \times M$ 블록 단위 내 R 개의 샘플에 대해 1D 라플라시안 연산을 수행하여 기울기 값을 계산할 수 있다. 이때, R 은 0을 포함한 양의 정수이다. 또한, 상기 R 은 N 과 M 의 곱보다 같거나 작을 수 있다. 여기서, 1D 라플라시안 연산의 합이 계산되는 범위 내 샘플들 중 적어도 하나에서 수행된 1D 라플라시안 연산 결과에 대해 통계 값을 계산하여 기울기 값의 통계 값을 산출할 수 있다. 이때, 통계값은 합(sum), 가중합 또는 평균값 중 어느 하나일 수 있다.
- [409] 일 예로, R 이 1인 경우, $N \times M$ 블록 내에서 1개의 샘플에 대해서만 1D 라플라시안 연산을 수행할 수 있다.
- [410] 다른 예로, R 이 2인 경우, $N \times M$ 블록 내에서 2개의 샘플에 대해서만 1D 라플라시안 연산을 수행할 수 있다.
- [411] 또 다른 예로, R 이 4인 경우, $N \times M$ 블록 내에서 4개의 샘플에 대해서만 1D 라플라시안 연산을 수행할 수 있다.
- [412] 본 발명의 일 실시 예에 따를 때, 부표본에 기반하여 1D 라플라시안 연산의 합이 계산되는 범위인 $N \times M$ 내 R 개의 샘플에 대해 1D 라플라시안 연산을 수행하여 기울기 값을 계산할 수 있다. 이때, R 은 0을 포함한 양의 정수이다. 또한, 상기 R 은 N 과 M 의 곱보다 같거나 작을 수 있다. 여기서, 1D 라플라시안 연산의 합이 계산되는 범위 내 샘플들 중 적어도 하나에서 수행된 1D 라플라시안 연산 결과에 대해 통계 값을 계산하여 기울기 값의 통계 값을 산출할 수 있다. 이때, 통계값은 합(sum), 가중합 또는 평균값 중 어느 하나일 수 있다.
- [413] 일 예로, R 이 1인 경우, 1D 라플라시안 연산의 합이 계산되는 범위인 $N \times M$ 내에서 1개의 샘플에 대해서만 1D 라플라시안 연산을 수행할 수 있다.
- [414] 다른 예로, R 이 2인 경우, 1D 라플라시안 연산의 합이 계산되는 범위인 $N \times M$

- 내에서 2개의 샘플에 대해서만 1D 라플라시안 연산을 수행할 수 있다.
- [415] 또 다른 예로, R 이 4인 경우, 1D 라플라시안 연산의 합이 계산되는 범위인 $N \times M$ 내에서 4개의 샘플에 대해서만 1D 라플라시안 연산을 수행할 수 있다.
- [416] 도 10 내지 도 12는 특정 샘플 위치에서 가로, 세로, 제1 대각, 제2 대각 방향에 따른 기울기 값을 결정하는 일 예이다. 상기 특정 샘플 위치는 블록 분류 단위 내에서 서브샘플링된 샘플 위치일 수 있고, 1D 라플라시안 연산의 합이 계산되는 범위 내에서 서브샘플링된 샘플 위치일 수 있다. 또한, 상기 특정 샘플 위치는 블록 별로 동일할 수 있다. 반대로, 상기 특정 샘플 위치는 블록 별로 상이할 수 있다. 또한, 상기 특정 샘플 위치는 계산하고자 하는 1D 라플라시안 연산의 방향에 관계없이 동일할 수 있다. 반대로, 상기 특정 샘플 위치는 계산하고자 하는 1D 라플라시안 연산의 방향에 따라 상이할 수 있다.
- [417] 도 10의 예와 같이, 4×4 블록 분류의 경우, 특정 샘플 위치에서 기울기 값의 합 g_v, g_h, g_{d1}, g_{d2} 들 중 적어도 하나를 계산할 수 있다. 여기서, $V, H, D1, D2$ 는 샘플 단위에서 세로, 가로, 제1 대각, 제2 대각 방향에 따른 각각의 1D 라플라시안 연산의 결과를 나타낸다. 즉, $V, H, D1, D2$ 위치에서 각각의 세로, 가로, 제1 대각, 제2 대각 방향에 따른 1D 라플라시안 연산이 수행될 수 있다. 또한, 1D 라플라시안 연산이 수행되는 위치가 서브샘플링될 수 있다. 도 10에서, 블록 분류 색인 C 는 음영이 표시된 4×4 블록 단위로 할당될 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다. 여기서, 가는 실선에 포함된 사각형은 각 복원 샘플 위치를 나타내고, 굵은 실선은 1D 라플라시안 연산의 합이 계산되는 범위를 나타낸다.
- [418] 도 10의 예와 같이, 1D 라플라시안 연산의 방향에 관계없이, 1D 라플라시안 연산이 수행되는 특정 샘플 위치가 동일할 수 있다. 또한, 도 10의 예와 같이 1D 라플라시안 연산이 수행되는 샘플 위치의 패턴은 체커판(checkboard) 패턴 또는 quincunx 패턴이라고 할 수 있다. 또한, 1D 라플라시안 연산이 수행되는 샘플 위치는 1D 라플라시안 연산의 합이 계산되는 범위 또는 블록 분류 단위 또는 블록 단위 내에서 가로 방향(x 축 방향) 및 세로 방향(y 축 방향)으로 모두 짝수 값을 가지거나, 모두 홀수 값을 가지는 위치를 의미할 수 있다.
- [419] 도 11의 예와 같이, 4×4 블록 분류의 경우, 특정 샘플 위치에서 기울기 값의 합 g_v, g_h, g_{d1}, g_{d2} 들 중 적어도 하나를 계산할 수 있다. 여기서, $V, H, D1, D2$ 는 샘플 단위에서 세로, 가로, 제1 대각, 제2 대각 방향에 따른 각각의 1D 라플라시안 연산의 결과를 나타낸다. 즉, $V, H, D1, D2$ 위치에서 각각의 세로, 가로, 제1 대각, 제2 대각 방향에 따른 1D 라플라시안 연산이 수행될 수 있다. 또한, 1D 라플라시안 연산이 수행되는 위치가 서브샘플링될 수 있다. 도 11에서, 블록 분류 색인 C 는 음영이 표시된 4×4 블록 단위로 할당될 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다. 여기서, 가는 실선에 포함된 사각형은 각 복원 샘플 위치를 나타내고, 굵은 실선은 1D 라플라시안 연산의 합이 계산되는 범위를 나타낸다.

- [420] 도 11의 예와 같이, 1D 라플라시안 연산의 방향에 관계없이, 1D 라플라시안 연산이 수행되는 특정 샘플 위치가 동일할 수 있다. 또한, 도 11의 예와 같이 1D 라플라시안 연산이 수행되는 샘플 위치의 패턴은 체커판(checkboard) 패턴 또는 quincunx 패턴이라고 할 수 있다. 또한, 1D 라플라시안 연산이 수행되는 샘플 위치는 1D 라플라시안 연산의 합이 계산되는 범위 또는 블록 분류 단위 또는 블록 단위 내에서 가로 방향(x축 방향) 및 세로 방향(y축 방향)으로 각각 짝수 값 및 홀수 값을 가지거나, 각각 홀수 값 및 짝수 값을 가지는 위치를 의미할 수 있다.
- [421] 도 12의 예와 같이, 4x4 블록 분류의 경우, 특정 샘플 위치에서 기울기 값의 합 g_v, g_h, g_{d1}, g_{d2} 들 중 적어도 하나를 계산할 수 있다. 여기서, V, H, D1, D2는 샘플 단위에서 세로, 가로, 제1 대각, 제2 대각 방향에 따른 각각의 1D 라플라시안 연산의 결과를 나타낸다. 즉, V, H, D1, D2 위치에서 각각의 세로, 가로, 제1 대각, 제2 대각 방향에 따른 1D 라플라시안 연산이 수행될 수 있다. 또한, 1D 라플라시안 연산이 수행되는 위치가 서브샘플링될 수 있다. 도 12에서, 블록 분류 색인 C는 음영이 표시된 4x4 블록 단위로 할당될 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기보다 클 수 있다. 여기서, 가는 실선에 포함된 사각형은 각 복원 샘플 위치를 나타내고, 굵은 실선은 1D 라플라시안 연산의 합이 계산되는 범위를 나타낸다.
- [422] 본 발명의 일 실시 예에 따를 때, 세로, 가로, 제1 대각, 제2 대각 방향에 대한 각각의 기울기 값의 합 g_v, g_h, g_{d1}, g_{d2} 들 중 적어도 2개 이상의 값을 서로 비교할 수 있다.
- [423] 일 예로, 상기 기울기 값의 합들을 계산한 이후, 세로 방향 기울기의 합 g_v 값과 가로 방향 기울기의 합 g_h 값을 비교하여, 가로 방향 및 세로 방향에 대한 기울기 값의 합의 최대값 $g_{h,v}^{max}$ 과 최소값 $g_{h,v}^{min}$ 을 수학적 14의 예와 같이 유도할 수 있다.
- [424] [수학적 14]
- [425]
$$g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v),$$
- [426] 이때, 세로 방향 기울기의 합 g_v 값과 가로 방향 기울기의 합 g_h 값을 비교하기 위해, 수학적 15의 예와 같이 상기 기울기 값의 합들의 크기를 서로 비교할 수 있다.
- [427] [수학적 15]
- [428]
$$g_{h,v}^{max} = \begin{cases} g_h & \text{if } g_h \geq g_v \\ g_v & \text{if } g_v > g_h \end{cases}, \quad g_{h,v}^{min} = \begin{cases} g_h & \text{if } g_h \leq g_v \\ g_v & \text{if } g_v < g_h \end{cases}$$
- [429] 다른 예로, 제1 대각 방향 기울기의 합 g_{d1} 값과 제2 대각 방향 기울기의 합 g_{d2} 값을 비교하여, 제1 대각 방향 및 제2 대각 방향에 대한 기울기 값의 합의 최대값 $g_{d0,d1}^{max}$ 과 최소값 $g_{d0,d1}^{min}$ 을 수학적 16의 예와 같이 유도할 수 있다.

[430] [수학식 16]

$$[431] \quad g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}),$$

[432] 이때, 제1 대각 방향 기울기의 합 g_{d1} 값과 제2 대각 방향 기울기의 합 g_{d2} 값을 비교하기 위해, 수학식 17의 예와 같이 상기 기울기 값의 합들의 크기를 서로 비교할 수 있다.

[433] [수학식 17]

$$[434] \quad g_{d0,d1}^{max} = \begin{cases} g_{d0} & \text{if } g_{d0} \geq g_{d1} \\ g_{d1} & \text{if } g_{d1} > g_{d0} \end{cases}, \quad g_{d0,d1}^{min} = \begin{cases} g_{d0} & \text{if } g_{d0} \leq g_{d1} \\ g_{d1} & \text{if } g_{d1} < g_{d0} \end{cases}$$

[435] 본 발명의 일 실시 예에 따를 때, 방향성 D 값을 산출하기 위해, 상기 최대값들 및 상기 최소값들을 아래와 같이 2개의 임계값 t_1 및 t_2 와 비교할 수 있다.

[436] 방향성 D 값은 0을 포함하는 양의 정수일 수 있다. 일 예로, 방향성 D 값은 0에서 4 사이의 값을 가질 수 있다. 다른 예로, 방향성 D 값은 0에서 2 사이의 값을 가질 수 있다.

[437] 또한, 방향성 D 값은 해당 영역의 특징에 따라 결정될 수 있다. 예를 들어, 방향성 D 값이 0인 경우 텍스처(texture) 영역, 방향성 D 값이 1인 경우 강한 수평/수직 방향성, 방향성 D 값이 2인 경우 약한 수평/수직 방향성, 방향성 D 값이 3인 경우 강한 제1/제2 대각 방향성, 방향성 D 값이 4인 경우 약한 제1/제2 대각 방향성을 의미할 수 있다. 상기 방향성 D 값의 결정은 하기 단계 1 내지 단계 4에 의해 수행될 수 있다.

[438]

[439] 단계 1: $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ 와 $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ 가 만족하면 D 값을 0으로 설정

[440] 단계 2: $g_{h,v}^{max} / g_{h,v}^{min} > g_{d0,d1}^{max} / g_{d0,d1}^{min}$ 가 만족하면 단계 3으로 진행, 그렇지 않으면 단계 4로 진행

[441] 단계 3: $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ 가 만족하면 D 값을 2로 설정, 그렇지 않으면 D 값을 1로 설정

[442] 단계 4: $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ 가 만족하면 D 값을 4로 설정, 그렇지 않으면 D 값을 3으로 설정

[443]

[444] 여기서, 임계값 t_1 및 t_2 는 양의 정수이다. t_1 및 t_2 는 같은 값을 가질 수도 있고 서로 다른 값을 가질 수도 있다. 일 예로, t_1 및 t_2 는 각각 2, 9일 수 있다. 다른 예로, t_1 및 t_2 는 각각 1일 수 있다. 또 다른 예로, t_1 및 t_2 는 각각 1, 9일 수 있다.

[445] 활동성 A 값은, 2x2 블록 분류의 경우, 수학식 18의 예와 같이 나타낼 수 있다.

[446] [수학식 18]

$$[447] \quad A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}).$$

- [448] 일 예로, k 와 l 은 서로 같은 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 같을 수 있다.
- [449] 다른 예로, k 와 l 은 서로 다른 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 다를 수 있다.
- [450] 또 다른 예로, k 는 $i-2$ 에서 $i+3$ 의 범위, l 은 $j-2$ 에서 $j+3$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 6×6 크기일 수 있다.
- [451] 또 다른 예로, k 는 $i-1$ 에서 $i+2$ 의 범위, l 은 $j-1$ 에서 $j+2$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 4×4 크기일 수 있다.
- [452] 또 다른 예로, k 는 i 에서 $i+1$ 의 범위, l 은 j 에서 $j+1$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 2×2 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기와 동일할 수 있다.
- [453] 또 다른 예로, 1D 라플라시안 연산의 결과의 합이 계산되는 범위는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.
- [454] 또한, 활동성 A 값은 4×4 블록 분류의 경우, 수학적 식 19의 예와 같이 나타낼 수 있다.
- [455] [수학적 식 19]
- [456]
$$A = \sum_{k=i-2}^{i+5} \sum_{l=j-2}^{j+5} (V_{k,l} + H_{k,l}).$$
- [457] 일 예로, k 와 l 은 서로 같은 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 같을 수 있다.
- [458] 다른 예로, k 와 l 은 서로 다른 범위를 가질 수 있다. 즉, 1D 라플라시안 연산의 합이 계산되는 범위의 가로 길이와 세로 길이가 다를 수 있다.
- [459] 또 다른 예로, k 는 $i-2$ 에서 $i+5$ 의 범위, l 은 $j-2$ 에서 $j+5$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 8×8 크기일 수 있다.
- [460] 또 다른 예로, k 는 i 에서 $i+3$ 의 범위, l 은 j 에서 $j+3$ 의 범위를 가져서, 1D 라플라시안 연산의 합이 계산되는 범위가 4×4 크기일 수 있다. 이때, 1D 라플라시안 연산의 합이 계산되는 범위가 블록 분류 단위의 크기와 동일할 수 있다.
- [461] 또 다른 예로, 1D 라플라시안 연산의 결과의 합이 계산되는 범위는 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나의 2차원 기하학적 형태일 수 있다.
- [462] 한편, 상기 활동성 A 값은 양자화되어 I 과 J 사이 범위의 양자화한 활동성 A_q 값을 가질 수 있다. 여기서, I 와 J 는 각각 0을 포함한 양의 정수일 수 있고, 각각 0과 4일 수 있다.

- [463] 양자화한 활동성 A_q 값은 소정의 방식을 이용하여 결정할 수 있다.
- [464] 일 예로, 양자화한 활동성 A_q 값은 수학적 식 20의 예와 같이 나타낼 수 있다. 이 경우, 양자화한 활동성 A_q 값은 특정 최소값 X 및 특정 최대값 Y 의 범위 안에 포함될 수 있다.
- [465] [수학적 식 20]
- [466]
$$A_q = (WA) \gg R = \frac{WA}{2^R}$$
- [467] 수학적 식 20에서 양자화한 활동성 A_q 값은 상기 활동성 A 값에 특정 상수 W 값을 곱하고, R 만큼 우측 쉬프트 연산을 수행하여 산출될 수 있다. 이때, X , Y , W , R 은 0을 포함한 양의 정수이다. 일 예로, W 는 24일 수 있고, R 은 13일 수 있다. 다른 예로, W 는 64일 수 있고, R 은 $(3 + N\text{비트})$ 일 수 있다. 예를 들어, N 은 양의 정수일 수 있고, 8 또는 10일 수 있다. 또 다른 예로, W 는 32일 수 있고, R 은 $(3 + N\text{비트})$ 일 수 있다. 예를 들어, N 은 양의 정수일 수 있고, 8 또는 10일 수 있다.
- [468] 다른 예로, 상기 양자화한 활동성 A_q 값은 룩 업 테이블(LUT, Look Up Table) 방식을 이용하여, 활동성 A 값과 양자화한 활동성 A_q 값 사이의 매핑 관계를 설정할 수 있다. 즉, 활동성 A 값에 연산을 수행하고 상기 테이블을 이용하여 양자화한 활동성 A_q 값을 산출할 수 있다. 이때, 상기 연산은 곱셈, 나눗셈 연산, 우측 쉬프트 연산, 좌측 쉬프트 연산, 덧셈, 뺄셈 중 적어도 하나를 포함할 수 있다.
- [469] 한편, 색차 성분의 경우 상기 블록 분류 과정을 수행하지 않고, K 개의 필터로 각각의 색차 성분에 대해 필터링을 수행할 수 있다. 여기서, K 는 0을 포함한 양의 정수일 수 있다. 또는 K 는 0 내지 7 중 하나일 수 있다. 또한, 색차 성분의 경우 상기 블록 분류 과정을 수행하지 않고, 대응하는 위치의 휘도 성분에서 결정한 블록 분류 색인을 이용하여 필터링을 수행할 수 있다. 또한, 색차 성분의 경우 색차 성분에 대한 필터 정보가 시그널링 되지 않고 고정된 형태의 필터가 이용될 수 있다.
- [470] 또 다른 예로, 상기 블록 분류 색인을 결정하는 방법들 중 적어도 하나 이상의 조합을 이용해서 블록 분류 색인을 결정할 수 있다.
- [471] 이하에서는, 필터링 수행 단계에 대해서 후술한다.
- [472] 본 발명의 일 실시 예에 따를 때, 상기 결정된 블록 분류 색인에 대응하는 필터를 이용하여 복원/복호 영상 내 샘플 또는 블록에 필터링을 수행할 수 있다. 상기 필터링 수행 시 L 가지의 필터 중 하나의 필터가 선택될 수 있다. 상기 L 은 0을 포함한 양의 정수일 수 있다.
- [473] 일 예로, 상기 블록 분류 단위로 L 가지의 필터 중 하나의 필터가 선택되어 복원/복호 영상에 복원/복호 샘플 단위로 필터링을 수행할 수 있다.
- [474] 다른 예로, 상기 블록 분류 단위로 L 가지의 필터 중 하나의 필터가 선택되어 복원/복호 영상에 블록 분류 단위로 필터링을 수행할 수 있다.
- [475] 또 다른 예로, 상기 블록 분류 단위로 L 가지의 필터 중 하나의 필터가 선택되어

복원/복호 영상에 CU 단위로 필터링을 수행할 수 있다.

[476] 또 다른 예로, 상기 블록 분류 단위로 L가지의 필터 중 하나의 필터가 선택되어 복원/복호 영상에 블록 단위로 필터링을 수행할 수 있다.

[477] 또 다른 예로, 상기 블록 분류 단위로 L가지의 필터 중 U개의 필터가 선택되어 복원/복호 영상에 복원/복호 샘플 단위로 필터링을 수행할 수 있다. 이때, U는 양의 정수이다.

[478] 또 다른 예로, 상기 블록 분류 단위로 L가지의 필터 중 U개의 필터가 선택되어 복원/복호 영상에 블록 분류 단위로 필터링을 수행할 수 있다. 이때, U는 양의 정수이다.

[479] 또 다른 예로, 상기 블록 분류 단위로 L가지의 필터 중 U개의 필터가 선택되어 복원/복호 영상에 CU 단위로 필터링을 수행할 수 있다. 이때, U는 양의 정수이다.

[480] 또 다른 예로, 상기 블록 분류 단위로 L가지의 필터 중 U개의 필터가 선택되어 복원/복호 영상에 블록 단위로 필터링을 수행할 수 있다. 이때, U는 양의 정수이다.

[481] 한편, 상기 L가지의 필터들을 필터 세트(filter set)라고 표현할 수 있다.

[482] 본 발명의 일 실시 예에 따를 때, 상기 L가지의 필터는 필터 계수, 필터 탭 수(필터 길이), 필터 형태 및 필터 종류 중 적어도 하나가 서로 다를 수 있다.

[483] 일 예로, 상기 L가지의 필터는 블록, CU, PU, TU, CTU, 슬라이스, 타일, 타일 그룹, 영상(픽처), 시퀀스 중 적어도 하나의 단위에서 필터 계수, 필터 탭 수(필터 길이), 필터 형태, 필터 종류 중 적어도 하나가 서로 같을 수 있다.

[484] 다른 예로, 상기 L가지의 필터는 블록, CU, PU, TU, CTU, 슬라이스, 타일, 타일 그룹, 영상(픽처), 시퀀스 중 적어도 하나의 단위에서 필터 계수, 필터 탭 수(필터 길이), 필터 형태, 필터 종류 중 적어도 하나가 서로 다를 수 있다.

[485] 또한, 상기 필터링은 샘플, 블록, CU, PU, TU, CTU, 슬라이스, 타일, 타일 그룹, 영상(픽처), 시퀀스 중 적어도 하나의 단위로 동일한 필터 또는 서로 다른 필터를 이용해서 수행될 수 있다.

[486] 또한, 상기 필터링은 샘플, 블록, CU, PU, TU, CTU, 슬라이스, 타일, 타일 그룹, 영상(픽처), 시퀀스 중 적어도 하나의 단위에서 필터링을 수행할지에 대한 필터링 수행 여부 정보에 따라 수행될 수 있다. 상기 필터링 수행 여부 정보는 샘플, 블록, CU, PU, TU, CTU, 슬라이스, 타일, 타일 그룹, 영상(픽처), 시퀀스 중 적어도 하나의 단위에서 부호화기에서 복호화기로 시그널링되는 정보일 수 있다.

[487] 본 발명의 일 실시 예에 따를 때, 상기 필터링을 위한 필터로 마름모(diamond, rhombus)의 필터 형태를 가지며, 필터 탭 수가 서로 다른 N가지 필터를 사용할 수 있다. 여기서, N은 양의 정수일 수 있다. 예를 들어, 필터 탭수가 5x5, 7x7, 9x9인 마름모 형태는 도 13과 같이 나타낼 수 있다.

[488] 도 13은 본 발명의 일 실시 예에 따른 마름모 형태의 필터를 나타내는 도면이다.

[489] 도 13을 참조하면, 필터 탭수가 5x5, 7x7, 9x9인 마름모 형태의 3가지 필터 중

어떤 필터를 사용할지에 대한 정보를 부호화기에서 복호화기로 시그널링하기 위해, 픽처/타일/타일 그룹/슬라이스/시퀀스 단위에서 필터 색인(filter index)을 엔트로피 부호화/복호화할 수 있다. 즉, 비트스트림 내 시퀀스 파라미터 세트(sequence parameter set), 픽처 파라미터 세트(picture parameter set), 슬라이스 헤더(slice header), 슬라이스 데이터(slice data), 타일 헤더(tile header), 타일 그룹 헤더(tile group header) 등에 필터 색인을 엔트로피 부호화/복호화할 수 있다.

- [490] 본 발명의 일 실시 예에 따를 때, 부호화기/복호화기에서 필터 탭수가 1개로 고정되어 있을 경우, 상기 필터 색인을 엔트로피 부호화/복호화하지 않고, 해당 필터를 이용해서 부호화기/복호화기에서 필터링을 수행할 수 있다. 여기서, 상기 필터 탭수는 휘도 성분인 경우 7x7 마름모 형태이고, 색차 성분인 경우 5x5 마름모 형태일 수 있다.
- [491] 본 발명의 일 실시 예에 따를 때, 상기 3가지의 마름모 형태 필터 중 적어도 하나는 휘도 성분 및 색차 성분 중 적어도 하나의 복원/복호 샘플 필터링에 사용될 수 있다.
- [492] 일 예로, 휘도 성분 복원/복호 샘플에는 도 33의 3가지 마름모 형태 필터 중 적어도 하나의 필터가 복원/복호 샘플 필터링에 사용될 수 있다.
- [493] 다른 예로, 색차 성분 복원/복호 샘플에는 도 33의 5x5 마름모 형태 필터가 복원/복호 샘플 필터링에 사용될 수 있다.
- [494] 또 다른 예로, 색차 성분 복원/복호 샘플에는 상기 색차 성분에 대응하는 휘도 성분에서 선택된 필터를 이용하여 필터링이 수행될 수 있다.
- [495] 본 발명의 일 실시 예에 따를 때, 상기 필터링을 위한 필터로 필터 탭 수는 모두 5x5로 동일하지만, 필터 형태가 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 등 중 적어도 하나를 포함하는 필터가 사용될 수 있다. 예를 들어, 필터 탭수가 5x5이며, 필터 형태가 정사각형, 팔각형, 눈송이, 마름모 형태는 도 34와 같이 나타낼 수 있다.
- [496] 여기서, 필터 탭 수는 5x5로 한정되지 않고, 3x3, 4x4, 5x5, 6x6, 7x7, 8x8, 9x9, 5x3, 7x3, 9x3, 7x5, 9x5, 9x7, 11x7 등 HxV의 필터 탭을 가지는 적어도 하나의 필터가 사용될 수 있다. 여기서, H와 V는 양의 정수일 수 있고, 서로 같은 값 또는 서로 다른 값을 가질 수도 있다. 또한, H와 V 중 적어도 하나는 부호화기/복호화기에서 기정의된 값일 수 있고, 부호화기에서 복호화기로 시그널링되는 값일 수 있다. 또한, H와 V 중 하나의 값을 이용하여 다른 하나의 값이 정의될 수 있다. 또한, H와 V의 값을 이용하여 H 또는 V의 최종 값이 정의될 수 있다.
- [497] 한편, 도 14의 예와 같은 필터들 중 어떤 필터를 사용할지에 대한 정보를 부호화기에서 복호화기로 시그널링하기 위해, 픽처/타일/타일 그룹/슬라이스/시퀀스 단위에서 필터 색인을 엔트로피 부호화/복호화할 수 있다. 즉, 비트스트림 내 시퀀스 파라미터 세트, 픽처 파라미터 세트, 슬라이스 헤더,

슬라이스 데이터, 타일 헤더, 타일 그룹 헤더 등에 필터 색인을 엔트로피 부호화/복호화할 수 있다.

- [498] 한편, 도 14의 예와 같은 정사각형, 팔각형, 눈송이, 마름모 형태 필터 중 적어도 하나는 휘도 성분 및 색차 성분 중 적어도 하나의 복원/복호 샘플 필터링에 사용될 수 있다.
- [499] 본 발명의 일 실시 예에 따를 때, 상기 필터링을 위한 필터로서 필터 형태가 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 중 적어도 하나를 포함하는 필터가 사용될 수 있다. 예를 들어, 도 35a 및/또는 도 35b의 예와 같이 정사각형, 팔각형, 눈송이, 마름모, 육각형, 직사각형, 십자가, 삼, 클로버, 대각 형태 등을 가질 수 있다.
- [500] 일 예로, 도 15a 및/또는 도 15b에서 필터 세로 길이가 5인 필터들 중 적어도 하나를 이용해서 필터 세트를 구성한 후 필터링을 수행할 수 있다.
- [501] 다른 예로, 도 15a 및/또는 도 15b에서 필터 세로 길이가 3인 필터들 중 적어도 하나를 이용해서 필터 세트를 구성한 후 필터링을 수행할 수 있다.
- [502] 또 다른 예로, 도 15a 및/또는 도 15b에서 필터 세로 길이가 5 및 3인 필터들 중 적어도 하나를 이용해서 필터 세트를 구성한 후 필터링을 수행할 수 있다.
- [503] 한편, 도 15a 및/또는 도 15b에서 필터의 형태는 필터 세로 길이를 3 또는 5를 기준으로 설계한 것이지만, 이에 한정되지 않으며, 필터 세로 길이가 M 인 경우로 설계하여 필터링에 사용할 수 있다. 여기서, M 은 양의 정수이다.
- [504] 한편, 도 15a 및/또는 도 15b의 예와 같은 필터들 중 H 만큼의 필터 세트를 구성하여 어떤 필터를 사용할지에 대한 정보를 부호화기에서 복호화기로 시그널링하기 위해, 픽처/타일/타일 그룹/슬라이스/시퀀스 단위에서 필터 색인을 엔트로피 부호화/복호화할 수 있다. 여기서, H 는 양의 정수이다. 즉, 비트스트림 내 시퀀스 파라미터 세트, 픽처 파라미터 세트, 슬라이스 헤더, 슬라이스 데이터, 타일 헤더, 타일 그룹 헤더 등에 필터 색인을 엔트로피 부호화/복호화할 수 있다.
- [505] 한편, 상기 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 형태 필터 중 적어도 하나는 휘도 성분 및 색차 성분 중 적어도 하나의 복원/복호 샘플 필터링에 사용될 수 있다.
- [506] 한편, 도 15a 및/또는 도 15b의 각 필터 형태 내 숫자는 필터 계수 색인을 나타내며, 필터 중심으로 필터 계수 색인이 대칭 형태를 가진다. 즉, 도 15a 및/또는 도 15b의 필터들을 점 대칭 필터 형태라고 할 수 있다.
- [507] 한편, 상기 마름모, 직사각형, 정사각형, 사다리꼴, 대각, 눈송이, 삼, 클로버, 십자가, 삼각형, 오각형, 육각형, 팔각형, 십각형, 십이각형 형태 필터 중 적어도 하나는 휘도 성분 및 색차 성분 중 적어도 하나의 복원/복호 샘플 필터링에 사용될 수 있다.
- [508] 본 발명의 일 실시 예에 따를 때, 복원/복호 샘플 $R(i,j)$ 에 대해 상기 필터링을

수행하여, 필터링된 복호 샘플 $R'=(i,j)$ 를 생성할 수 있다. 상기 필터링된 복호 샘플은 수학식 21의 예와 같이 나타낼 수 있다.

[509] [수학식 21]

$$[510] \quad R'(i,j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k,l) \times R(i+k, j+l)$$

[511] 수학식 21에서 L은 가로 방향 혹은 세로 방향에 따른 필터 탭 수(필터 길이)이며, $f(k,l)$ 는 필터 계수이다.

[512] 한편, 상기 필터링 수행 시 필터링된 복호 샘플 $R'=(i,j)$ 에 오프셋 Y 값을 더해줄 수 있다. 상기 오프셋 Y 값은 엔트로피 부호화/복호화될 수 있다. 또한, 오프셋 Y 값은 현재 복원/복호 샘플 값과 주변 복원/복호 샘플들의 값 중 적어도 하나 이상의 통계값을 이용해서 산출될 수 있다. 또한, 상기 오프셋 Y 값은 현재 복원/복호 샘플과 주변 복원/복호 샘플 중 적어도 하나의 부호화 파라미터에 기반하여 결정될 수 있다. 여기서, Y는 0을 포함하는 정수이다.

[513] 한편, 상기 필터링된 복호 샘플은 N비트 내로 표현되도록 절삭을 수행할 수 있다. 여기서, N은 양의 정수이다. 예를 들어, 복원/복호 샘플에 필터링을 수행하여 생성된 필터링된 복호 샘플에 10비트로 절삭할 경우, 최종 복호 샘플 값은 0~1023 사이의 값을 가질 수 있다.

[514] 본 발명의 일 실시 예에 따를 때, 색차 성분의 경우 휘도 성분의 필터링 정보에 기반하여 필터링이 수행될 수 있다.

[515] 일 예로, 색차 성분의 복원 영상 필터링은 휘도 성분의 복원 영상 필터링이 수행되는 경우에만 수행될 수 있다. 여기서, 색차 성분의 복원 영상 필터링은 U(Cr) 및 V(Cb) 성분 중 적어도 하나 이상에 수행될 수 있다.

[516] 다른 예로, 색차 성분의 경우 대응하는 휘도 성분의 필터 계수, 필터 탭 수, 필터 형태, 필터링 수행 여부 정보 중 적어도 하나를 이용하여 필터링을 수행할 수 있다.

[517] 본 발명의 일 실시 예에 따를 때, 상기 필터링 수행 시, 현재 샘플 주변에 가용하지 않는 샘플이 존재하는 경우, 패딩을 수행한 후 패딩된 샘플을 이용하여 필터링이 수행될 수 있다. 상기 패딩은 상기 가용하지 않는 샘플에 대해 인접하며 가용한 샘플 값을 상기 가용하지 않는 샘플 값에 복사하는 방식을 의미할 수 있다. 또는, 상기 가용하지 않는 샘플에 대해 인접하며 가용한 샘플 값에 기반하여 획득된 샘플 값 또는 통계 값을 이용할 수 있다. 상기 패딩은 P열 개 및 R행 개만큼 반복하여 수행할 수 있다. 여기서, P와 R은 양의 정수일 수 있다.

[518] 여기서, 가용하지 않는 샘플은 CTU, CTB, 슬라이스, 타일, 타일 그룹, 픽처 경계 밖에 존재하는 샘플을 의미할 수 있다. 또한, 가용하지 않는 샘플은 현재 샘플이 속한 CTU, CTB, 슬라이스, 타일, 타일 그룹, 픽처 중 적어도 하나와 서로 다른 CTU, CTB, 슬라이스, 타일, 타일 그룹, 픽처 중 적어도 하나에 속한 샘플을

의미할 수 있다.

- [519] 일 실시 예에 따르면, 픽처 경계 밖에 위치한 샘플에 대한 패딩 방법과, 픽처 경계 안에 존재하지만 적응적 루프내 필터링이 적용되는 소정의 영역 밖에 위치한 샘플에 대한 패딩 방법은 상이할 수 있다. 픽처 경계 밖에 위치한 샘플에 대하여, 픽처의 경계에 위치한 가용 샘플의 값을 복사하여, 픽처의 경계 밖에 위치한 샘플의 값이 결정될 수 있다. 픽처 경계 안에 존재하지만 적응적 루프내 필터링이 적용되는 소정의 영역 밖에 위치한 샘플에 대하여, 가용하지 않은 샘플의 패딩 값은 픽처의 경계를 기준으로 가용한 샘플의 값을 미러링함으로써 결정될 수 있다.
- [520] 일 실시 예에 따르면, 픽처 경계 밖에 위치한 샘플에 대한 패딩은 수평 방향으로만 수행되고, 수직 방향으로는 수행되지 않을 수 있다. 반대로, 픽처 경계 밖에 위치한 샘플에 대한 패딩은 수직 방향으로만 수행되고, 수평 방향으로는 수행되지 않을 수 있다.
- [521] 또한, 필터링 수행 시, 소정의 샘플을 이용하지 않을 수 있다.
- [522] 일 예로, 상기 필터링 수행 시, 패딩된 샘플을 이용하지 않을 수도 있다.
- [523] 다른 예로, 상기 필터링 수행 시, 현재 샘플 주변에 가용하지 않는 샘플이 존재하는 경우, 상기 가용하지 않는 샘플을 필터링에 이용하지 않을 수 있다.
- [524] 또 다른 예로, 상기 필터링 수행 시, 현재 샘플 주변에 존재하는 샘플이 CTU 또는 CTB 경계 밖에 위치하는 경우, 상기 주변에 존재하는 샘플을 필터링에 이용하지 않을 수 있다.
- [525] 또한, 상기 필터링 수행 시, 디블록킹 필터링, 적응적 샘플 오프셋, 적응적 루프내 필터링 중 적어도 하나가 적용된 샘플을 사용할 수 있다.
- [526] 또한, 상기 필터링 수행 시, 현재 샘플 주변에 존재하는 샘플이 CTU 또는 CTB 경계 밖에 위치하는 경우, 디블록킹 필터링, 적응적 샘플 오프셋, 적응적 루프내 필터링 중 적어도 하나를 적용하지 않을 수 있다.
- [527] 또한, 상기 필터링에 사용되는 샘플에 CTU 또는 CTB 경계 밖 등에 위치하여 가용하지 않는 샘플이 포함될 경우, 해당 가용하지 않은 샘플에 디블록킹 필터링, 적응적 샘플 오프셋, 적응적 루프내 필터링 중 적어도 하나를 적용하지 않고 필터링에 사용할 수 있다.
- [528] 본 발명의 일 실시 예에 따를 때, 상기 필터링 수행 시, CU, PU, TU, 블록, 블록 분류 단위, CTU, CTB 중 적어도 하나의 경계 주변에 존재하는 샘플들 중 적어도 하나에 대해 필터링을 수행할 수 있다. 이때, 상기 경계는 수직 방향 경계, 수평 방향 경계, 대각 방향 경계 중 적어도 하나를 포함할 수 있다. 또한, 상기 경계 주변에 존재하는 샘플들은 경계를 기준으로 U행, U열, U 샘플들 중 적어도 하나일 수 있다. 여기서, U는 양의 정수일 수 있다.
- [529] 본 발명의 일 실시 예에 따를 때, 상기 필터링 수행 시, CU, PU, TU, 블록, 블록 분류 단위, CTU, CTB 중 적어도 하나의 경계 주변에 존재하는 샘플들을 제외한 블록 내부에 존재하는 샘플들 중 적어도 하나에 대해 필터링을 수행할 수 있다.

이때, 상기 경계는 수직 방향 경계, 수평 방향 경계, 대각 방향 경계 중 적어도 하나를 포함할 수 있다. 또한, 상기 경계 주변에 존재하는 샘플들은 경계를 기준으로 U행, U열, U 샘플들 중 적어도 하나일 수 있다. 여기서, U는 양의 정수일 수 있다.

[530] 또 다른 예로, 상기 필터링 방법들 중 적어도 하나 이상의 조합을 이용해서 필터링을 수행할 수 있다.

[531] 이하에서는, 필터 정보 부호화/복호화 단계에 대해서 후술한다.

[532] 본 발명의 일 실시 예에 따를 때, 필터 정보는 비트스트림 내 슬라이스 헤더와 슬라이스 데이터 내 첫번째 CTU 구문 요소(syntax element) 사이에서 엔트로피 부호화/복호화될 수 있다.

[533] 또한, 필터 정보는 비트스트림 내 시퀀스 파라미터 세트, 픽처 파라미터 세트, 슬라이스 헤더, 슬라이스 데이터, 타일 헤더, 타일 그룹 헤더, CTU, CTB 등에서 엔트로피 부호화/복호화될 수 있다.

[534] 한편, 상기 필터 정보에는 휘도 성분 필터링 수행 여부 정보, 색차 성분 필터링 수행 여부, 필터 계수 값, 필터 개수, 필터 탭 수(필터 길이) 정보, 필터 형태 정보, 필터 종류 정보, 슬라이스/타일/타일 그룹/픽처/CTU/CTB/블록/CU 단위 필터링 수행 여부 정보, CU 단위 필터링 수행 여부 정보 수 정보, CU 최대 깊이 필터링 정보, CU 단위 필터링 수행 여부 정보, 이전 참조 영상 필터 사용 여부 정보, 이전 참조 영상 필터 색인, 블록 분류 색인에 대한 고정된 필터 사용 여부 정보, 고정된 필터에 대한 색인 정보, 필터 병합 정보, 휘도 성분과 색차 성분에 다른 필터를 사용하는지 여부 정보, 필터의 대칭 형태에 관한 정보 중 적어도 하나 이상이 포함될 수 있다.

[535] 여기서, 상기 필터 탭 수는 필터의 가로 길이, 필터의 세로 길이, 필터의 제1 대각 길이, 필터의 제2 대각 길이, 필터의 가로와 세로 길이, 필터 내 필터 계수의 개수 중 적어도 하나일 수 있다.

[536] 한편, 필터 정보에는 최대 L개의 휘도 필터가 포함될 수 있다. 여기서, L은 양의 정수이며, 25일 수 있다. 또한, 필터 정보에는 최대 L개의 색차 필터가 포함될 수 있다. 여기서, L은 0을 포함한 양의 정수이며, 0 내지 8일 수 있다. 상기 최대 L개의 색차 필터에 대한 정보는 파라미터 세트 또는 헤더에 포함될 수 있다. 상기 파라미터 세트는 적응 파라미터 세트일 수 있다. 상기 최대 L개의 색차 필터에 대한 정보는 색차 ALF 개수 정보를 의미할 수 있고, 예를 들어, `alf_chroma_num_alt_filters_minus1`의 형태로 부호화/복호화될 수 있다.

[537] 한편, 하나의 필터에는 최대 K개의 휘도 필터 계수가 포함될 수 있다. 여기서, K는 양의 정수이며, 13일 수 있다. 또한, 필터 정보에는 최대 K개의 색차 필터 계수가 포함될 수 있다. 여기서, K는 양의 정수이며, 7일 수 있다.

[538] 예를 들어, 필터의 대칭 형태에 관한 정보는 필터 형태가 점 대칭 형태, 가로 대칭 형태, 세로 대칭 형태 또는 이들의 결합 형태에 관한 정보일 수 있다.

[539] 한편, 필터 계수들 중 일부의 계수들만 시그널링 할 수 있다. 예를 들어, 필터가

대칭 형태인 경우, 필터의 대칭 형태에 관한 정보와 대칭되는 필터 계수 세트 중 하나의 세트만 시그널링 할 수 있다. 또한, 예를 들어, 필터 중심의 필터 계수는 암시적으로 도출될 수 있기 때문에 시그널링 되지 않을 수 있다.

- [540] 본 발명의 일 실시 예에 따를 때, 상기 필터 정보 중 필터 계수 값은 부호화기에서 양자화될 수 있고, 양자화된 필터 계수 값이 엔트로피 부호화될 수 있다. 마찬가지로, 복호화기에서 양자화된 필터 계수 값이 엔트로피 복호화될 수 있고, 양자화된 필터 계수 값이 역양자화되어 필터 계수 값으로 복원될 수도 있다. 상기 필터 계수 값은 고정된 M 비트(bit)로 표현될 수 있는 값 범위 내로 양자화되고, 역양자화될 수도 있다. 또한, 상기 필터 계수 중 적어도 하나는 서로 다른 비트로 양자화되고, 역양자화될 수도 있다. 반대로, 상기 필터 계수 중 적어도 하나는 서로 동일한 비트로 양자화되고, 역양자화될 수도 있다. 또한, 상기 M 비트는 양자화 매개변수에 따라 결정될 수 있다. 또한, 상기 M 비트는 부호화기 및 복호화기에서 기정의된 상수 값일 수 있다. 여기서, M 은 양의 정수일 수고 8 또는 10일 수 있다. 또한, 상기 M 비트는 부호화기/복호화기에서 샘플을 표현하는데 필요한 비트 수 보다 작거나 같을 수 있다. 예를 들어, 샘플을 표현하는데 필요한 비트 수가 10이면, 상기 M 은 8일 수 있다. 필터 내 필터 계수들 중 제1 필터 계수는 $-2M$ 에서 $2M-1$ 까지의 값을 가질 수 있고, 제2 필터 계수는 0에서 $2M-1$ 까지의 값을 가질 수 있다. 여기서, 제1 필터 계수는 필터 계수들 중 중심 필터 계수를 제외한 나머지 필터 계수를 의미할 수 있고, 제2 필터 계수는 필터 계수들 중 중심 필터 계수를 의미할 수 있다.

- [541] 또한, 상기 필터 정보 중 필터 계수 값은 부호화기 및 복호화기 중 적어도 하나에서 절삭(clipping)될 수 있고, 절삭에 관한 최소값 및 최대값 중 적어도 하나가 엔트로피 부호화/복호화될 수 있다. 상기 필터 계수 값은 최소값 및 최대값의 범위 내로 절삭될 수 있다. 또한, 상기 최소값 및 최대값 중 적어도 하나는 필터 계수마다 서로 다른 값을 가질 수 있다. 반대로, 상기 최소값 및 최대값 중 적어도 하나는 필터 계수마다 서로 동일한 값을 가질 수 있다. 또한, 상기 최소값 및 최대값 중 적어도 하나는 양자화 매개변수에 따라 결정될 수 있다. 또한 상기 최소값 및 최대값 중 적어도 하나는 부호화기 및 복호화기에서 기정의된 상수 값일 수 있다.

- [542] 본 발명의 일 실시 예에 따를 때, 휘도 성분 및 색차 성분 중 적어도 하나에 대한 필터링 수행 여부는 필터링 수행 여부 정보(플래그)를 이용할 수 있다.

- [543] 일 예로, 휘도 성분 및 색차 성분 중 적어도 하나에 대한 필터링 수행 여부는 CTU/CTB/CU/블록 단위의 필터링 수행 여부 정보(플래그)를 이용할 수 있다. 예를 들어, CTB 단위의 필터링 수행 여부 정보가 제1 값인 경우 해당 CTB에 대해 필터링을 수행할 수 있고, 제2 값인 경우 해당 CTB에 대해 필터링을 수행하지 않을 수 있다. 이때, 각 CTB 별로 필터링 수행 여부 정보가 엔트로피 부호화/복호화될 수 있다. 또 다른 예로, CU의 최대 깊이 또는 최소 크기에 관한 정보(CU 최대 깊이 필터링 정보)를 추가로 엔트로피 부호화/복호화하여, 해당

최대 깊이 또는 최소 크기까지만 CU 단위 필터링 수행 여부 정보를 엔트로피 부호화/복호화할 수 있다.

- [544] 예를 들어, CU 단위의 플래그는 블록 구조(block structure)에 따라 정방향 블록 분할과 비정방향 블록 분할이 가능한 경우, 정방향 블록 분할의 깊이까지만 엔트로피 부호화/복호화될 수 있다. 또한, CU 단위의 플래그는 추가적으로 비정방향 블록 분할의 깊이까지 엔트로피 부호화/복호화될 수 있다.
- [545] 다른 예로, 휘도 성분 및 색차 성분 중 적어도 하나에 대한 필터링 수행 여부 정보는 블록 단위의 플래그를 이용할 수 있다. 예를 들어, 블록 단위의 플래그가 제1 값인 경우 해당 블록에 대해 필터링을 수행할 수 있고, 제2 값인 경우 해당 블록에 대해 필터링을 수행하지 않을 수 있다. 상기 블록 단위의 크기는 $N \times M$ 이며, N 과 M 은 양의 정수일 수 있다.
- [546] 또 다른 예로, 휘도 성분 및 색차 성분 중 적어도 하나에 대한 필터링 수행 여부 정보는 CTU 단위의 플래그를 이용할 수 있다. 예를 들어, CTU 단위의 플래그가 제1 값인 경우 해당 CTU에 대해 필터링을 수행할 수 있고, 제2 값인 경우 해당 CTU에 대해 필터링을 수행하지 않을 수 있다. 상기 CTU 단위의 크기는 $N \times M$ 이며, N 과 M 은 양의 정수일 수 있다.
- [547] 또 다른 예로, 휘도 성분 및 색차 성분 중 적어도 하나에 대한 필터링 수행 여부는 픽처/슬라이스/타일 그룹/타일 종류에 따라 결정될 수 있으며, 휘도 성분 및 색차 성분 중 적어도 하나에 대한 필터링 수행 여부 정보는 픽처/슬라이스/타일 그룹/타일 단위의 플래그를 이용할 수 있다.
- [548] 본 발명의 일 실시 예에 따를 때, 상기 필터링은 고정된 필터(fixed filter) 세트를 이용할 수 있다.
- [549] 화면내 예측 영상(I 영상/슬라이스/타일 그룹/타일)에서는 시간적 필터 계수 예측을 사용할 수 없지만, 각 블록 분류 색인에 따라 최대 16 종류의 고정된 필터 세트들 중 적어도 하나의 필터를 필터링에 사용할 수 있다. 고정된 필터 세트 사용 여부를 부호화기에서 복호화기로 시그널링하기 위해, 각 블록 분류 색인에 대한 고정된 필터 사용 여부 정보가 엔트로피 부호화/복호화될 수 있고, 고정된 필터를 사용할 경우 고정된 필터에 대한 색인 정보도 엔트로피 부호화/복호화할 수 있다. 만약 특정 블록 분류 색인에 대해 고정된 필터를 사용할 경우에도 필터 계수는 엔트로피 부호화/복호화될 수 있고, 엔트로피 부호화/복호화되는 필터 계수와 고정된 필터 계수를 이용하여 복원 영상을 필터링할 수 있다.
- [550] 또한, 상기 화면간 예측 영상(B/P-영상/슬라이스/타일 그룹/타일)에서도 상기 고정된 필터 세트를 이용할 수 있다.
- [551] 또한, 상기 적응적 루프내 필터링은 필터 계수의 엔트로피 부호화/복호화 없이 고정된 필터로 수행될 수 있다. 여기서, 고정된 필터는 부호화기와 복호화기에서 미리 정의한 필터 세트를 의미할 수 있다. 이때, 필터 계수의 엔트로피 부호화/복호화 없이, 부호화기와 복호화기에서 미리 정의한 필터 세트 중 어떤 필터 또는 어떤 필터 세트를 사용할지에 대한 고정된 필터에 대한 색인 정보를

엔트로피 부호화/복호화할 수 있다. 상기 블록 분류 단위, 블록 단위, CU 단위, CTU 단위, 슬라이스, 타일, 타일 그룹, 픽처 단위 중 적어도 하나의 단위에서 필터 계수 값, 필터 탭 수(필터 길이), 필터 형태 중 적어도 하나가 다른 고정된 필터로 필터링을 수행할 수 있다.

[552] 본 발명의 일 실시 예에 따를 때, 기정의된 고정된 필터로부터 필터 계수 예측이 금지될 수 있다.

[553] 본 발명의 일 실시 예에 따를 때, 곱셈 연산을 쉬프트 연산으로 대체할 수 있다. 우선, 휘도 및/또는 색차 블록의 필터링을 수행하기 위해 사용되는 필터 계수들을 2개의 그룹으로 나눌 수 있다. 예를 들어, 제1 그룹 {L0, L1, L2, L3, L4, L5, L7, L8, L9, L10, L14, L15, L16, L17} 및 그 나머지 계수들을 포함하는 제2 그룹으로 나눌 수 있다. 제1 그룹에는 {-64, -32, -16, -8, -4, 0, 4, 8, 16, 32, 64}의 계수 값들을 가지도록 제한될 수 있다. 이때, 제1 그룹에 포함된 필터 계수들과 복원/복호 샘플에 대한 곱셈 연산은 단일 비트 쉬프트 연산으로 구현될 수 있다. 따라서, 제1 그룹에 포함된 필터 계수들은 시그널링 오버헤드를 줄이기 위해, 이진화 전에 각각 비트 쉬프트 연산이 수행된 값에 매핑될 수 있다.

[554] 본 발명에 따른 컴퓨터 판독가능한 기록 매체는 부호화 유닛을 블록 분류 단위로 분류하는 단계, 상기 부호화 유닛에 대한 필터 정보를 이용하여 상기 블록 분류 단위로 분류된 부호화 유닛을 필터링하는 단계 및 상기 필터 정보를 부호화하는 단계를 포함하는 영상 부호화 방법으로 생성된 비트스트림을 저장할 수 있다. 상기 블록 분류는 부호화 유닛 단위로 수행되는 것만 한정하는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등의 단위에서 수행될 수 있다. 또한, 상기 필터링의 대상은 부호화 유닛으로 한정되는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등에서 필터링이 수행될 수 있다. 또한, 상기 필터 정보는 부호화 유닛에 대해서만 한정되는 것은 아니고, 슬라이스, 타일, 타일 그룹, 픽처, 시퀀스, CTU, 블록, CU, PU, TU 등에 대한 필터 정보를 의미할 수 있다.

[555]

[556] 아래는 적응적 루프내 필터링을 부호화기/복호화기에서 구현하기 위해 필요한 구문 요소 정보, 구문 요소 정보의 의미, 부호화/복호화 과정의 예를 나타낸다. 본 명세서에서 선택스는 상기 구문 요소를 의미할 수 있다.

[557] 도 16 내지 도 21는 적응적 루프내 필터링에 필요한 구문 요소 정보의 예를 나타낸다. 적응적 루프내 필터링에 필요한 구문 요소 중 적어도 하나가 파라미터 세트, 헤더, 브릭, CTU, 또는 CU 중 적어도 하나에서 엔트로피 부호화/복호화될 수 있다.

[558] 이때, 파라미터 세트, 헤더, 브릭, CTU, 또는 CU 는 비디오 파라미터 세트(video parameter set), 디코딩 파라미터 세트(decoding parameter set), 시퀀스 파라미터 세트(sequence parameter set), 적응 파라미터 세트(adaptation parameter set), 픽처 파라미터 세트(picture parameter set), 픽처 헤더(picture header), 서브 픽처

헤더(sub-picture header), 슬라이스 헤더(slice header), 타일 그룹 헤더(tile group header), 타일 헤더(tile header), 브릭(brick), CTU(coding tree unit), 또는 CU(coding unit) 등 중 적어도 하나일 수 있다.

- [559] 여기서, 상기 시그널링되는 파라미터 세트, 헤더, 브릭, CTU, 또는 CU 단위에서 상기 적응적 루프내 필터링에 대한 구문 요소를 이용하여 적응적 루프내 필터링에 사용될 수 있다.
- [560] 예를 들어, 상기 적응적 루프내 필터링에 대한 구문 요소가 시퀀스 파라미터 세트에서 엔트로피 부호화/복호화될 경우, 시퀀스 단위에서 동일한 구문 요소 값을 가지는 적응적 루프내 필터링에 대한 구문 요소를 이용하여 적응적 루프내 필터링을 수행할 수 있다.
- [561] 또 다른 예로, 상기 적응적 루프내 필터링에 대한 구문 요소가 픽처 파라미터 세트에서 엔트로피 부호화/복호화될 경우, 픽처 단위에서 동일한 구문 요소 값을 가지는 적응적 루프내 필터링에 대한 구문 요소를 이용하여 적응적 루프내 필터링을 수행할 수 있다.
- [562] 또 다른 예로, 상기 적응적 루프내 필터링에 대한 구문 요소가 픽처 헤더에서 엔트로피 부호화/복호화될 경우, 픽처 단위에서 동일한 구문 요소 값을 가지는 적응적 루프내 필터링에 대한 구문 요소를 이용하여 적응적 루프내 필터링을 수행할 수 있다.
- [563] 또 다른 예로, 상기 적응적 루프내 필터링에 대한 구문 요소가 슬라이스 헤더에서 엔트로피 부호화/복호화될 경우, 슬라이스 단위에서 동일한 구문 요소 값을 가지는 적응적 루프내 필터링에 대한 구문 요소를 이용하여 적응적 루프내 필터링이 수행될 수 있다.
- [564] 또 다른 예로, 상기 적응적 루프내 필터링에 대한 구문 요소가 적응 파라미터 세트에서 엔트로피 부호화/복호화될 경우, 동일한 적응 파라미터 세트를 참조하는 단위에서 동일한 구문 요소 값을 가지는 적응적 루프내 필터링에 대한 구문 요소를 이용하여 적응적 루프내 필터링이 수행될 수 있다.
- [565] 적응 파라미터 세트는 서로 다른 픽처, 서브 픽처, 슬라이스, 타일 그룹, 타일, 또는 브릭에서 참조하여 공유될 수 있는 파라미터 세트를 의미할 수 있다. 또한, 픽처 내 서브 픽처, 슬라이스, 타일 그룹, 타일, 또는 브릭에서는 서로 다른 적응 파라미터 세트를 참조하여, 적응 파라미터 세트 내 정보를 사용할 수 있다.
- [566] 또한, 적응 파라미터 세트는 픽처 내 서브 픽처, 슬라이스, 타일 그룹, 타일, 또는 브릭에서는 서로 다른 적응 파라미터 세트의 식별자를 사용하여 서로 다른 적응 파라미터 세트를 참조할 수 있다.
- [567] 또한, 적응 파라미터 세트는 서브 픽처 내 슬라이스, 타일 그룹, 타일, 또는 브릭에서는 서로 다른 적응 파라미터 세트의 식별자를 사용하여 서로 다른 적응 파라미터 세트를 참조할 수 있다.
- [568] 또한, 적응 파라미터 세트는 슬라이스 내 타일, 또는 브릭에서는 서로 다른 적응 파라미터 세트의 식별자를 사용하여 서로 다른 적응 파라미터 세트를 참조할 수

있다.

[569] 또한, 적응 파라미터 세트는 타일 내 브릭에서는 서로 다른 적응 파라미터 세트의 식별자를 사용하여 서로 다른 적응 파라미터 세트를 참조할 수 있다.

[570] 상기 적응 파라미터 세트 식별자는 적응 파라미터 세트에 부여된 식별번호를 의미할 수 있다.

[571] 상기 시퀀스의 파라미터 세트 또는 헤더에 적응 파라미터 세트 식별자에 대한 정보가 포함될 수 있다. 그리고 해당 적응 파라미터 세트 식별자에 대응하는 적응 파라미터 세트가 해당 시퀀스에서 사용될 수 있다.

[572] 상기 픽처의 파라미터 세트 또는 헤더에 적응 파라미터 세트 식별자에 대한 정보가 포함될 수 있다. 그리고 해당 적응 파라미터 세트 식별자에 대응하는 적응 파라미터 세트가 해당 픽처에서 사용될 수 있다.

[573] 상기 서브 픽처의 파라미터 세트 또는 헤더에 적응 파라미터 세트 식별자에 대한 정보가 포함될 수 있다. 그리고 해당 적응 파라미터 세트 식별자에 대응하는 적응 파라미터 세트가 해당 서브 픽처에서 사용될 수 있다.

[574] 상기 타일의 파라미터 세트 또는 헤더에 적응 파라미터 세트 식별자에 대한 정보가 포함될 수 있다. 그리고 해당 적응 파라미터 세트 식별자에 대응하는 적응 파라미터 세트가 해당 타일에서 사용될 수 있다.

[575] 상기 슬라이스의 파라미터 세트 또는 헤더에 적응 파라미터 세트 식별자에 대한 정보가 포함될 수 있다. 그리고 해당 적응 파라미터 세트 식별자에 대응하는 적응 파라미터 세트가 해당 슬라이스에서 사용될 수 있다.

[576] 상기 브릭의 파라미터 세트 또는 헤더에 적응 파라미터 세트 식별자에 대한 정보가 포함될 수 있다. 그리고 해당 적응 파라미터 세트 식별자에 대응하는 적응 파라미터 세트가 해당 브릭에서 사용될 수 있다.

[577] 상기 픽처는 하나 이상의 타일 행과 하나 이상의 타일 열로 분할될 수 있다.

[578] 상기 서브 픽처는 픽처 내에서 하나 이상의 타일 행과 하나 이상의 타일 열로 분할될 수 있다. 상기 서브 픽처는 픽처 내에서 직사각형/정사각형 형태를 가지는 영역이며, 하나 이상의 CTU를 포함할 수 있다. 또한, 하나의 서브 픽처 내에는 적어도 하나 이상의 타일/브릭/슬라이스가 포함될 수 있다.

[579] 상기 타일은 픽처 내에서 직사각형/정사각형 형태를 가지는 영역이며, 하나 이상의 CTU를 포함할 수 있다. 또한, 타일은 하나 이상의 브릭으로 분할될 수 있다.

[580] 상기 브릭은 타일 내에서 하나 이상의 CTU 행을 의미할 수 있다. 타일은 하나 이상의 브릭으로 분할될 수 있고, 각 브릭은 적어도 하나 이상의 CTU 행을 가질 수 있다. 또한 브릭은 추가적으로 분할되지 않는 타일을 의미할 수 있다.

[581] 상기 슬라이스는 픽처 내에서 하나 이상의 타일을 포함할 수 있고, 타일 내 하나 이상의 브릭을 포함할 수 있다.

[582] 도 16의 예와 같이, `sps_alf_enabled_flag`는 적응적 루프내 필터링이 시퀀스 단위에서 수행되는지에 대한 정보를 의미할 수 있다.

- [583] 예를 들어, `sps_alf_enabled_flag`가 제1 값(예: 0)인 경우 시퀀스 단위에서 적응적 루프내 필터링이 수행되지 않을 수 있다. 그리고 `sps_alf_enabled_flag`가 제2 값(예: 1)인 경우 시퀀스 단위에서 적응적 루프내 필터링이 수행될 수 있다.
- [584] `sps_alf_enabled_flag`가 비트스트림 내에 존재하지 않을 경우 `sps_alf_enabled_flag`는 제1 값(예: 0)으로 추론(infer)될 수 있다.
- [585] 도 17의 예와 같이, `adaptation_parameter_set_id`는 다른 구문 요소에 의해 참조되는 적응 파라미터 세트의 식별자(identifier)를 의미할 수 있다. `adaptation_parameter_set_id`가 비트스트림 내에 존재하지 않을 경우 `adaptation_parameter_set_id`는 제1 값(예: 0)으로 추론될 수 있다.
- [586] 도 18의 예와 같이, `aps_params_type`는 적응 파라미터 세트 내에 존재하는 적응 파라미터 세트 타입 정보를 의미할 수 있다. 또한, `aps_params_type`은 적응 파라미터 세트에 포함된 부호화 정보의 타입을 의미할 수 있다. 예를 들어, `aps_params_type`가 제1 값(예: 0)인 경우, 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값은 적응적 루프내 필터링에 관한 파라미터(ALF 타입)를 의미할 수 있다. `aps_params_type`가 제2 값(예: 1)인 경우, 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값은 색차 스케일링 및 휘도 매핑에 관한 파라미터(색차 스케일링 및 휘도 매핑 타입)를 의미할 수 있다. `aps_params_type`가 제3 값(예: 2)인 경우, 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값은 양자화 행렬 세트에 관한 파라미터(양자화 행렬 타입)를 의미할 수 있다. 여기서, SL은 양자화 행렬을 의미하는 스케일링 리스트(scaling list)를 의미할 수 있다. `aps_params_type`가 비트스트림 내에 존재하지 않을 경우 `aps_params_type`는 제1 값(예: 0), 제2 값(예: 1), 및 제3 값(예: 2)이 아닌 다른 값으로 추론될 수 있다.
- [587] 이전에 시그널링된 적응 파라미터 세트의 적응 파라미터 세트 식별자와 동일한 적응 파라미터 세트 식별자가 현재 적응 파라미터 세트에 대하여 새롭게 시그널링될 수 있다. 또한, 이전에 시그널링된 적응 파라미터 세트의 적응 파라미터 세트 식별자 및 적응 파라미터 타입과 동일한 적응 파라미터 세트 식별자 및 적응 파라미터 타입을 가지는 적응 파라미터 세트에 대하여 새롭게 시그널링될 수 있다. 이 때, 이전에 시그널링된 적응 파라미터 세트의 데이터/컨텐츠/구문 요소 값은 새롭게 시그널링되는 적응 파라미터 세트의 데이터/컨텐츠/구문 요소 값으로 대체될 수 있다. 상기 대체되는 과정은 적응 파라미터 세트 업데이트(update) 과정을 의미한다.
- [588] 즉, 부호화기/복호화기에서는 이전에 시그널링된 적응 파라미터 세트의 데이터/컨텐츠/구문 요소 값을 참조하여, 적응적 루프내 필터링, 색차 스케일링 및 휘도 매핑(LMCS; luma mapping with chroma scaling), 양자화 행렬을 이용한 양자화/역양자화 중 적어도 하나가 수행될 수 있다. 이전에 시그널링된 적응 파라미터 세트의 적응 파라미터 세트 식별자와 동일한 적응 파라미터 세트 식별자가 현재 적응 파라미터 세트에 대하여 새롭게 시그널링된 시점부터,

부호화기/복호화기에서는 새롭게 시그널링된 시점의 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값을 참조하여, 적응적 루프내 필터링, 색차 스케일링 및 휘도 매핑(LMCS; luma mapping with chroma scaling), 양자화 행렬을 이용한 양자화/역양자화 중 적어도 하나가 수행될 수 있다.

- [589] 또한, 부호화기/복호화기에서는 이전에 시그널링된 적응 파라미터 세트의 데이터/컨텐츠/구문 요소 값을 참조하여, 적응적 루프내 필터링, 색차 스케일링 및 휘도 매핑(LMCS; luma mapping with chroma scaling), 양자화 행렬을 이용한 양자화/역양자화 중 적어도 하나가 수행될 수 있다. 이전에 시그널링된 적응 파라미터 세트의 적응 파라미터 세트 식별자 및 적응 파라미터 타입과 동일한 적응 파라미터 세트 식별자 및 적응 파라미터 타입을 가지는 적응 파라미터 세트에 대하여 새롭게 시그널링된 시점부터, 부호화기/복호화기에서는 새롭게 시그널링된 시점의 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값을 참조하여, 적응적 루프내 필터링, 색차 스케일링 및 휘도 매핑(LMCS; luma mapping with chroma scaling), 양자화 행렬을 이용한 양자화/역양자화 중 적어도 하나가 수행될 수 있다.
- [590] 일 실시 예에 따르면, 적응 파라미터 세트에 색차 성분에 관련된 부호화 정보가 포함되어 있는지 여부를 나타내는 색차 성분 존재 정보(aps_chroma_present_flag)가 적응 파라미터 세트에 포함되어 부호화/복호화될 수 있다. 색차 성분 존재 정보가 적응 파라미터 세트에 색차 성분에 관련된 부호화 정보가 포함되어 있음을 나타내는 경우, 적응 파라미터 세트는 적응적 루프내 필터링, 색차 스케일링 및 휘도 매핑, 또는 양자화 행렬의 색차 성분에 대한 부호화 정보를 포함할 수 있다. 그렇지 않을 경우, 적응 파라미터 세트는 적응적 루프내 필터링, 색차 스케일링 및 휘도 매핑, 또는 양자화 행렬의 색차 성분에 대한 부호화 정보를 포함하지 않을 수 있다.
- [591] 상기 색차 성분 존재 정보에 따라, 색차 성분의 양자화 행렬이 존재하는지 여부가 결정될 수 있다. 또한 상기 색차 성분 존재 정보에 따라, 색차 성분의 적응적 루프내 필터 정보 존재하는지 여부가 결정될 수 있다. 또한 상기 색차 성분 존재 정보에 따라, 색차 성분의 색차 스케일링 및 휘도 매핑 정보가 존재하는지 여부가 결정될 수 있다.
- [592] 일 실시 예에 따르면, 현재 비디오, 시퀀스, 픽처, 또는 슬라이스의 크로마 포맷이 4:0:0인 경우(모노크롬(monochrome)인 경우), 상기 색차 성분 존재 정보는 색차 성분이 존재하지 않음을 나타낼 수 있다. 따라서, 크로마 포맷이 4:0:0인 경우, 픽처 헤더 또는 슬라이스 헤더는 색차 성분에 대한 적응 파라미터 세트 식별자를 포함하지 않을 수 있다. 즉, 크로마 포맷이 4:0:0인 경우, 픽처 헤더 또는 슬라이스 헤더는 색차 성분에 대한 양자화 행렬 정보, 적응적 루프내 필터 정보, 색차 스케일링 및 휘도 매핑(LMCS, Luma Mapping with Chroma Scaling) 정보를 포함하지 않을 수 있다.
- [593] chroma_format_idc 또는 ChromaArrayType은 크로마 포맷을 의미할 수 있다.

상기 크로마 포맷은 색차 성분의 형식을 의미할 수 있다.

[594] 예를 들어, chroma_format_idc가 제1 값(예: 0)인 경우, 크로마 포맷(chroma format)은 4:0:0 으로 설정될 수 있다. 현재 픽처의 크로마 포맷이 4:0:0인 경우, 현재 픽처는 색차 성분이 없는 모노크롬(monochrome)으로 결정될 수 있다.

[595] 또한, chroma_format_idc가 제2 값(예: 1)인 경우, 크로마 포맷은 4:2:0으로 설정될 수 있다. chroma_format_idc가 제3 값(예: 2)인 경우, 크로마 포맷은 4:2:2으로 설정될 수 있다. chroma_format_idc가 제4 값(예: 3)인 경우, 크로마 포맷은 4:4:4으로 설정될 수 있다.

[596] 또한, 크로마 포맷이 4:0:0인 경우, 현재 픽처 또는 현재 슬라이스에 참조하는 적응 파라미터 세트 내의 색차 성분 존재 정보는 색차 성분에 관한 부호화 정보가 존재하지 않는 것을 의미하는 제1 값(예: 0)으로 결정될 수 있다.

[597] 또한, 크로마 포맷이 4:0:0이 아닌 경우, 현재 픽처 또는 현재 슬라이스에 참조하는 적응 파라미터 세트 내의 색차 성분 존재 정보는 색차 성분에 관한 부호화 정보가 존재하는 것을 의미하는 제2 값(예: 1)으로 결정될 수 있다.

[598] 또한, 크로마 포맷이 4:0:0인 경우, 현재 픽처 또는 현재 슬라이스에 참조하는 적응 파라미터 세트 내의 색차 성분 존재 정보는 색차 성분에 관한 양자화 행렬 정보가 존재하지 않는 것을 의미하는 제1 값(예: 0)으로 결정될 수 있다.

[599] 또한, 크로마 포맷이 4:0:0이 아닌 경우, 현재 픽처 또는 현재 슬라이스에 참조하는 적응 파라미터 세트 내의 색차 성분 존재 정보는 색차 성분에 관한 양자화 행렬 정보가 존재하는 것을 의미하는 제2 값(예: 1)으로 결정될 수 있다.

[600] 도 19의 예와 같이, slice_alf_enabled_flag는 적응적 루프내 필터링이 슬라이스 단위에서 Y, Cb, Cr 성분 중 적어도 하나에 대해 수행되는지에 대한 정보를 의미할 수 있다.

[601] 예를 들어, slice_alf_enabled_flag가 제1 값(예: 0)인 경우 슬라이스 단위에서 Y, Cb, Cr 성분 모두에 대해 적응적 루프내 필터링이 수행되지 않을 수 있다. 그리고, slice_alf_enabled_flag가 제2 값(예: 1)인 경우 슬라이스 단위에서 Y, Cb, Cr 성분 중 적어도 하나에 대해 적응적 루프내 필터링이 수행될 수 있다.

[602] slice_alf_enabled_flag가 비트스트림 내에 존재하지 않을 경우, slice_alf_enabled_flag는 제1 값(예: 0)으로 추론될 수 있다.

[603] slice_num_alf_aps_ids_luma는 슬라이스에서 참조하는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 개수를 의미하는 슬라이스 휘도 ALF 세트 개수 정보일 수 있다. 여기서, ALF 세트는 복수의 적응적 루프내 필터(ALF, Adaptive Loop Filter)들로 이루어진 필터 집합을 의미할 수 있다. 이때, 휘도 ALF는 휘도 성분에 대한 적응적 루프내 필터를 의미할 수 있다.

[604] 상기 slice_num_alf_aps_ids_luma는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예를 들어 6일 수 있다.

[605] slice_num_alf_aps_ids_luma가 비트스트림 내에 존재하지 않을 경우 slice_num_alf_aps_ids_luma는 제1 값(예: 0)으로 추론될 수 있다.

- [606] 또한, 부호화기/복호화기에서 기설정된 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수와 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수는 서로 동일할 수 있다. 또한, 부호화기/복호화기에서 기설정된 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수와 양자화 행렬 세트에 대한 적응 파라미터 세트의 최대 개수는 서로 동일할 수 있다. 또한, 부호화기/복호화기에서 기설정된 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수와 양자화 행렬 세트에 대한 적응 파라미터 세트의 최대 개수는 서로 동일할 수 있다.
- [607] 또한, 부호화기/복호화기에서 기설정된 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수 및 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수는 서로 상이할 수 있다. 또한, 부호화기/복호화기에서 기설정된 양자화 행렬 세트에 대한 적응 파라미터 세트의 최대 개수 및 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수는 서로 상이할 수 있다. 또한, 부호화기/복호화기에서 기설정된 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수 및 양자화 행렬 세트에 대한 적응 파라미터 세트의 최대 개수는 서로 상이할 수 있다.
- [608] 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수, 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수, 및 양자화 행렬 세트에 대한 적응 파라미터 세트의 최대 개수의 합이 부호화기/복호화기에서 가질 수 있는 적응 파라미터 세트의 최대 개수일 수 있다.
- [609] 상기 적응적 루프내 필터링에 대한 적응 파라미터 세트의 개수는 화면내 슬라이스인 경우 최대 K개를 가질 수 있고, 화면간 슬라이스인 경우 최대 L개를 가질 수 있다. 여기서, K와 L은 양의 정수일 수 있고, 예를 들어 K는 1, L은 6일 수 있다.
- [610] 상기 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 개수는 화면내 슬라이스인 경우 최대 K개를 가질 수 있고, 화면간 슬라이스인 경우 최대 L개를 가질 수 있다. 여기서, K와 L은 양의 정수일 수 있고, 예를 들어 K는 1, L은 6일 수 있다.
- [611] 상기 양자화 행렬 세트에 대한 적응 파라미터 세트의 개수는 화면내 슬라이스인 경우 최대 K개를 가질 수 있고, 화면간 슬라이스인 경우 최대 L개를 가질 수 있다. 여기서, K와 L은 양의 정수일 수 있고, 예를 들어 K는 1, L은 6일 수 있다.
- [612] 또는, 상기 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 개수는 슬라이스의 종류에 관계없이 최대 J개를 가질 수 있다. 여기서, J는 1 내지 8의 양의 정수일 수 있다. 예를 들어 J는 4일 수 있다.
- [613] 또는, 상기 적응적 루프내 필터에 대한 적응 파라미터 세트의 개수는 슬라이스의 종류에 관계없이 최대 J개를 가질 수 있다. 여기서, J는 1 내지 8의 양의 정수일 수 있다. 예를 들어 J는 8일 수 있다.

- [614] 또는, 상기 양자화 행렬 세트에 대한 적응 파라미터 세트의 개수는 슬라이스의 종류에 관계없이 최대 J 개를 가질 수 있다. 여기서, J 는 1 내지 8의 양의 정수일 수 있다. 예를 들어 J 는 8일 수 있다.
- [615] 일 실시 예에 따르면, 상기 적응 파라미터 세트 식별자는 N 비트의 양의 정수 값을 가질 수 있다. 상기 N 값은 1보다 큰 양의 정수이다. 예를 들어, N 값은 5일 수 있다. 따라서, 상기 적응 파라미터 세트 식별자는 0부터 31 중 하나의 값을 나타낼 수 있다. 즉, 최대 32개의 적응 파라미터 세트가 정의될 수 있다.
- [616] 상기 32개의 적응 파라미터 세트 중에서 8개의 적응 파라미터 세트는 양자화 행렬에 대한 정보를 나타낼 수 있다. 그리고 다른 8개의 적응 파라미터 세트는 적응적 루프내 필터에 관한 정보를 나타낼 수 있다. 그리고 다른 4개의 적응 파라미터 세트는 색차 스케일링 및 휘도 매핑에 관한 정보를 나타낼 수 있다.
- [617] 양자화 행렬에 대한 적응 파라미터 세트 식별자는 0에서 N 까지의 값을 가질 수 있다. 이때, N 은 양의 정수이고, 7일 수 있다.
- [618] 또한, 적응적 루프내 필터에 대한 적응 파라미터 세트 식별자는 0에서 N 까지의 값을 가질 수 있다. 이때, N 은 양의 정수이고, 7일 수 있다.
- [619] 또한, 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트 식별자는 0에서 N 까지의 값을 가질 수 있다. 이때, N 은 양의 정수이고, 3일 수 있다.
- [620] 상기 양자화 행렬, 적응적 루프내 필터, 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트 식별자는 적응 파라미터 세트 타입에 따라 고정 길이 부호(fixed-length code)로 부호화될 수 있다.
- [621] 일 실시 예에 따르면, 하나의 픽처에 포함된 슬라이스들에 의하여 참조되는 적응 파라미터 세트의 최대 개수는 하나 이상일 수 있다. 예를 들어, 하나의 픽처에 포함된 슬라이스들은 최대 하나의 양자화 행렬에 대한 적응 파라미터 세트, 최대 하나의 색차 스케일링 및 휘도 매핑에 관한 적응 파라미터 세트, 최대 N 개의 적응적 루프내 필터에 대한 적응 파라미터 세트를 참조할 수 있다. 상기 N 은 2 이상의 정수이다. 예를 들어, N 은 8 또는 9일 수 있다.
- [622] $\text{slice_alf_aps_id_luma}[i]$ 는 슬라이스에서 참조하는 i 번째 적응적 루프내 필터링에 대한 적응 파라미터 세트를 나타내는 슬라이스 휘도 ALF 세트 식별자를 의미할 수 있다.
- [623] $\text{slice_alf_aps_id_luma}[i]$ 가 비트스트림 내에 존재하지 않을 경우 $\text{slice_alf_aps_id_luma}[i]$ 는 제1 값(예: 0)으로 추론될 수 있다.
- [624] 여기서, $\text{slice_alf_aps_id_luma}[i]$ 와 동일한 $\text{adaptation_parameter_set_id}$ 를 가지는 적응 파라미터 세트의 시간적 계층 식별자(Temporal Layer Identifier)는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다. 본 명세서에서, $\text{adaptation_parameter_set_id}$ 는 적응 파라미터 세트 식별자를 의미할 수 있다.
- [625] 하나의 픽처 내에서 2개 이상의 서브 픽처/슬라이스/타일 그룹/타일/브릭이 동일한 $\text{adaptation_parameter_set_id}$ 를 가지고, 상기 $\text{adaptation_parameter_set_id}$ 를 가지는 2개 이상의 적응적 루프내 필터링에 대한 적응 파라미터 세트가 존재할

경우, 동일한 adaptation_parameter_set_id를 가지는 2개 이상의 적응적 루프내 필터링에 대한 적응 파라미터 세트는 동일한 데이터/콘텐츠/구문 요소 값을 가질 수 있다.

- [626] 화면내 슬라이스인 경우, slice_alf_aps_id_luma[i]는 화면내 슬라이스를 포함하는 픽처 또는 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [627] 일 실시 예에 따르면, slice_alf_aps_id_luma[i]는 휘도 성분에 대한 적응적 루프내 필터 세트가 포함된 적응 파라미터 세트만을 참조할 수 있다.
- [628] 일 실시 예에 따르면, slice_alf_aps_id_luma[i]는 픽처 헤더 또는 슬라이스 헤더에서 참조하는 적응 파라미터 세트들 중에서 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조할 수 있다. 그리고 픽처 헤더 또는 슬라이스 헤더는 픽처 또는 슬라이스에 적용될 수 있는 적응적 루프내 필터링에 대한 적응 파라미터 세트를 나타내는 부호화 정보를 포함할 수 있다. 여기서, 상기 부호화 정보가 나타내는 adaptation_parameter_set_id를 가지는 적응 파라미터 세트의 시간적 계층 식별자(Temporal Layer Identifier)는 현재 픽처의 시간적 계층 식별자보다 작거나 같을 수 있다.
- [629] 소정의 적응 파라미터 세트의 적응 파라미터 세트 타입이 적응적 루프내 필터링에 관한 파라미터(ALF 타입)이며 적응 파라미터 세트 식별자(adaptation_parameter_set_id)가 현재 픽처 또는 픽처 헤더에서 참조하는 적응 파라미터 세트의 식별자인 alf_aps_id_luma[i]와 동일한 경우, 소정의 적응 파라미터 세트의 시간적 계층 식별자는 현재 픽처의 시간적 계층 식별자보다 작거나 같을 수 있다.
- [630] slice_alf_chroma_idc가 제1 값(예: 0)인 경우, Cb 및 Cr 성분에 적응적 루프내 필터링이 수행되지 않을 수 있다.
- [631] 또한, slice_alf_chroma_idc가 제2 값(예: 1)인 경우, Cb 성분에 적응적 루프내 필터링이 수행될 수 있다.
- [632] 또한, slice_alf_chroma_idc가 제3 값(예: 2)인 경우, Cr 성분에 적응적 루프내 필터링이 수행될 수 있다.
- [633] 또한, slice_alf_chroma_idc가 제4 값(예: 3)인 경우, Cb 및 Cr 성분에 적응적 루프내 필터링이 수행될 수 있다.
- [634] slice_alf_chroma_idc가 비트스트림 내에 존재하지 않을 경우 slice_alf_chroma_idc는 제1 값(예: 0)으로 추론될 수 있다.
- [635] 일 실시 예에 따르면, 픽처 헤더 또는 슬라이스 헤더는 픽처에 포함된 슬라이스 또는 슬라이스에 Cb 및 Cr 성분에 적응적 루프내 필터링이 수행되는지 여부를 나타내는 부호화 정보를 포함할 수 있다. 그리고 상기 픽처 헤더 또는 슬라이스 헤더의 부호화 정보가 허용하는 색차 성분 (Cb, Cr)에 대하여, slice_alf_chroma_idc는 슬라이스에 색차 성분의 허용 여부를 나타내는 색차 ALF 적용 정보일 수 있다. 예를 들어, 상기 픽처 헤더 또는 슬라이스 헤더의 Cb 성분의

적응적 루프내 필터링만이 허용될 경우, slice_alf_chroma_idc는 슬라이스에 Cb 성분의 적응적 루프내 필터링이 허용되는지만 나타낼 수 있다. 반대로 상기 픽처 헤더 또는 슬라이스 헤더의 Cr 성분의 적응적 루프내 필터링만이 허용될 경우, slice_alf_chroma_idc는 슬라이스에 Cr 성분의 적응적 루프내 필터링이 허용되는지만 나타낼 수 있다. 그리고 Cb 성분 및 Cr 성분이 모두 허용되지 않을 경우, slice_alf_chroma_idc는 부호화/복호화/획득되지 않고, 제1 값(예: 0)으로 추론될 수 있다.

[636] 일 실시 예에 따르면, slice_alf_chroma_idc 대신, 각각 Cb 성분에 적응적 루프내 필터링이 수행되는지 나타내는 slice_alf_cb_flag와 Cr 성분에 적응적 루프내 필터링이 수행되는지 나타내는 slice_alf_cr_flag가 부호화/복호화/획득될 수 있다. slice_alf_cb_flag가 제1 값(예: 0)인 경우, Cb 성분에 적응적 루프내 필터링이 수행되지 않을 수 있다. 그리고 slice_alf_cb_flag가 제2 값(예: 1)인 경우, Cb 성분에 적응적 루프내 필터링이 수행될 수 있다. slice_alf_cr_flag가 제1 값(예: 0)인 경우, Cr 성분에 적응적 루프내 필터링이 수행되지 않을 수 있다. 그리고 slice_alf_cr_flag가 제2 값(예: 1)인 경우, Cr 성분에 적응적 루프내 필터링이 수행될 수 있다. slice_alf_cb_flag 및 slice_alf_cr_flag가 비트스트림 내에 존재하지 않을 경우, slice_alf_cb_flag 및 slice_alf_cr_flag는 각각 제1 값(예: 0)으로 추론될 수 있다.

[637] 일 실시 예에 따르면, 픽처 헤더 또는 슬라이스 헤더는 픽처에 포함된 슬라이스 또는 슬라이스에 Cb 성분에 적응적 루프내 필터링이 수행되는지 여부를 나타내는 부호화 정보를 포함할 수 있다. 그리고 상기 픽처 헤더 또는 슬라이스 헤더의 부호화 정보가 Cb 성분의 적응적 루프내 필터링을 허용할 경우, slice_alf_cb_flag는 부호화/복호화/획득될 수 있다. 그리고 slice_alf_cb_flag에 따라, 픽처 또는 슬라이스에 Cb 성분의 필터링 허용 여부가 결정될 수 있다. 만약 상기 픽처 헤더 또는 슬라이스 헤더의 부호화 정보가 Cb 성분의 적응적 루프내 필터링을 허용하지 않을 경우, slice_alf_cb_flag는 부호화/복호화/획득되지 않고, slice_alf_cb_flag는 제1 값(예: 0)으로 추론될 수 있다.

[638] 마찬가지로, 픽처 헤더 또는 슬라이스 헤더는 픽처에 포함된 슬라이스 또는 슬라이스에 Cr 성분에 적응적 루프내 필터링이 수행되는지 여부를 나타내는 부호화 정보를 포함할 수 있다. 그리고 상기 픽처 헤더 또는 슬라이스 헤더의 부호화 정보가 Cr 성분의 적응적 루프내 필터링을 허용할 경우, slice_alf_cr_flag는 부호화/복호화/획득될 수 있다. 그리고 slice_alf_cr_flag에 따라, 픽처 또는 슬라이스에 Cr 성분의 필터링 허용 여부가 결정될 수 있다. 만약 상기 픽처 헤더 또는 슬라이스 헤더의 부호화 정보가 Cr 성분의 적응적 루프내 필터링을 허용하지 않을 경우, slice_alf_cr_flag는 부호화/복호화/획득되지 않고, slice_alf_cr_flag는 제1 값(예: 0)으로 추론될 수 있다.

[639] 상기 slice_alf_cb_flag 및 slice_alf_cr_flag는 슬라이스에서 시그널링되는 색차 성분에 적응적 루프내 필터링이 수행되는지 여부에 대한 정보의 예이며,

픽처에서 색차 성분에 적응적 루프내 필터링이 수행되는지 여부에 대한 정보가 시그널링될 경우, 상기 slice_alf_cb_flag 및 slice_alf_cr_flag는 ph_alf_cb_flag 및 ph_alf_cr_flag 형태로 변경되어 사용될 수 있다.

[640] slice_alf_aps_id_chroma는 슬라이스의 색차 성분에서 참조하는 적응 파라미터 세트의 식별자를 의미할 수 있다. 즉, slice_alf_aps_id_chroma는 슬라이스에서 참조하는 적응적 루프내 필터링에 대한 적응 파라미터 세트를 나타내는 슬라이스 색차 ALF 세트 식별자를 의미할 수 있다.

[641] 상기 slice_alf_aps_id_chroma는 slice_alf_cb_flag와 slice_alf_cr_flag 중 적어도 하나가 제2 값(예: 1)인 경우에 부호화/복호화될 수 있다.

[642] slice_alf_aps_id_chroma가 비트스트림 내에 존재하지 않을 경우 slice_alf_aps_id_chroma는 제1 값(예: 0)으로 추론될 수 있다.

[643] 여기서, slice_alf_aps_id_chroma와 동일한 adaptation_parameter_set_id를 가지는 적응 파라미터 세트의 시간적 계층 식별자는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다.

[644] 화면내 슬라이스인 경우, slice_alf_aps_id_chroma 또는 slice_alf_aps_id_chroma[i]는 화면내 슬라이스를 포함하는 픽처 또는 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.

[645] 일 실시 예에 따르면, slice_alf_aps_id_chroma[i]는 색차 성분에 대한 적응적 루프내 필터 세트가 포함된 적응 파라미터 세트만을 참조할 수 있다.

[646] 또한, slice_alf_aps_id_chroma 대신 slice_alf_aps_id_chroma[i]를 사용할 수 있다. 즉, 색차 성분에 대한 적응적 루프내 필터링 시 적응적 루프내 필터 정보를 포함하는 상기 2개 이상의 적응 파라미터 세트 중 하나의 적응 파라미터 세트가 선택될 수 있다. 해당 적응 파라미터 세트의 적응적 루프내 필터 정보(slice_alf_aps_id_chroma[i])가 색차 성분에 대한 적응적 루프내 필터링에 사용될 수 있다.

[647] slice_alf_aps_id_chroma[i]는 슬라이스에서 참조하는 i번째 적응적 루프내 필터링에 대한 적응 파라미터 세트의 식별자를 의미할 수 있다. 즉, slice_alf_aps_id_chroma[i]는 슬라이스에서 참조하는 i번째 적응적 루프내 필터링에 대한 적응 파라미터 세트를 나타내는 슬라이스 색차 ALF 세트 식별자를 의미할 수 있다.

[648] slice_alf_aps_id_chroma[i]가 비트스트림 내에 존재하지 않을 경우 slice_alf_aps_id_chroma[i]는 제1 값(예: 0)으로 추론될 수 있다.

[649] 여기서, slice_alf_aps_id_chroma[i]와 동일한 adaptation_parameter_set_id를 가지는 적응 파라미터 세트의 시간적 계층 식별자는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다.

[650] 하나의 픽처 내에서 2개 이상의 서브 픽처/슬라이스/타일 그룹/타일/브릭이 동일한 adaptation_parameter_set_id를 가지고, 상기 adaptation_parameter_set_id를

가지는 2개 이상의 적응적 루프내 필터링에 대한 적응 파라미터 세트가 존재할 경우, 동일한 adaptation_parameter_set_id를 가지는 2개 이상의 적응적 루프내 필터링에 대한 적응 파라미터 세트는 동일한 데이터/콘텐츠/구문 요소 값을 가질 수 있다.

- [651] 화면내 슬라이스인 경우, slice_alf_aps_id_chroma[i]는 화면내 슬라이스를 포함하는 픽처 또는 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [652] 소정의 적응 파라미터 세트의 적응 파라미터 세트 타입이 적응적 루프내 필터링에 관한 파라미터(ALF 타입)이며 적응 파라미터 세트 식별자(adaptation_parameter_set_id)가 현재 픽처 또는 픽처 헤더에서 참조하는 적응 파라미터 세트의 식별자인 alf_aps_id_chroma 또는 alf_aps_id_chroma[i]와 동일한 경우, 소정의 적응 파라미터 세트의 시간적 계층 식별자는 현재 픽처의 시간적 계층 식별자보다 작거나 같을 수 있다.
- [653] 도 20a 내지 20d의 예와 같이, alf_luma_filter_signal_flag는 휘도 성분에 대한 적응적 루프내 필터 세트가 적응 파라미터 세트에 포함되는지 여부를 나타내는 휘도 ALF 시그널링 플래그를 의미할 수 있다. 또한, alf_luma_filter_signal_flag는 휘도 성분에 대한 적응적 루프내 필터 세트가 부호화/복호화되는지 여부를 나타내는 휘도 ALF 시그널링 플래그를 의미할 수 있다.
- [654] 예를 들어, alf_luma_filter_signal_flag가 제1 값(예: 0)인 경우 휘도 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화되지 않을 수 있다. alf_luma_filter_signal_flag가 제2 값(예: 1)인 경우 휘도 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화될 수 있다.
- [655] alf_luma_filter_signal_flag가 비트스트림 내에 존재하지 않을 경우 alf_luma_filter_signal_flag는 제1 값(예: 0)으로 추론될 수 있다.
- [656] 일 실시 예에 따라, 적응 파라미터 세트에서 부호화/복호화/획득된 색차 성분 존재 정보가 적응 파라미터 세트에 색차 성분에 대한 정보가 포함되어 있지 않다고 나타낸 경우, alf_luma_filter_signal_flag가 적응적 루프내 필터 데이터에 포함되지 않고, 제2 값(예: 1)으로 추론될 수 있다. 즉, alf_luma_filter_signal_flag가 적응적 루프내 필터 데이터에서 부호화/복호화되지 않을 수 있다.
- [657] 그리고 상기 색차 성분 존재 정보가 적응 파라미터 세트에 색차 성분에 대한 정보가 포함되어 있다고 나타낸 경우, alf_luma_filter_signal_flag가 적응적 루프내 필터 데이터에 포함될 수 있다. 즉, alf_luma_filter_signal_flag가 적응적 루프내 필터 데이터에서 부호화/복호화될 수 있다.
- [658] 소정의 적응 파라미터 세트의 적응 파라미터 세트 타입이 적응적 루프내 필터링에 관한 파라미터(ALF 타입)이며 적응 파라미터 세트 식별자(adaptation_parameter_set_id)가 slice_alf_aps_id_luma[i]와 동일한 경우, 휘도 ALF 시그널링 플래그(alf_luma_filter_signal_flag)는 제2 값(예: 1)로 결정될 수 있다. 즉, 적응 파라미터 세트에 적응적 루프내 필터링에 관한 파라미터가

존재하고, 현재 픽처 또는 현재 슬라이스에서 해당 적응 파라미터 세트를 참조할 경우, 휘도 성분에 대한 적응적 루프내 필터 정보가 시그널링되므로, 휘도 ALF 시그널링 플래그(`alf_luma_filter_signal_flag`)는 제2 값(예: 1)을 가질 수 있다.

[659] `alf_chroma_filter_signal_flag`는 색차 성분에 대한 적응적 루프내 필터 세트가 적응 파라미터 세트에 포함되는지 여부를 나타내는 색차 ALF 시그널링 플래그를 의미할 수 있다. 또한, `alf_chroma_filter_signal_flag`는 색차 성분에 대한 적응적 루프내 필터 세트가 부호화/복호화되는지 여부를 나타내는 색차 ALF 시그널링 플래그를 의미할 수 있다.

[660] 예를 들어, `alf_chroma_filter_signal_flag`가 제1 값(예: 0)인 경우 색차 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화되지 않을 수 있다 그리고, `alf_chroma_filter_signal_flag`가 제2 값(예: 1)인 경우, 색차 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화될 수 있다.

[661] `alf_chroma_filter_signal_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_chroma_filter_signal_flag`는 제1 값(예: 0)으로 추론될 수 있다.

[662] 소정의 적응 파라미터 세트의 적응 파라미터 세트 타입이 적응적 루프내 필터링에 관한 파라미터(ALF 타입)이며 적응 파라미터 세트 식별자(`adaptation_parameter_set_id`)가 `slice_alf_aps_id_chroma` 또는 `slice_alf_aps_id_chroma[i]`와 동일한 경우, 색차 ALF 시그널링 플래그(`alf_chroma_filter_signal_flag`)는 제2 값(예: 1)로 결정될 수 있다. 즉, 적응 파라미터 세트에 적응적 루프내 필터링에 관한 파라미터가 존재하고, 현재 픽처 또는 현재 슬라이스에서 해당 적응 파라미터 세트를 참조할 경우, 색차 성분에 대한 적응적 루프내 필터 정보가 시그널링되므로, 색차 ALF 시그널링 플래그(`alf_chroma_filter_signal_flag`)는 제2 값(예: 1)을 가질 수 있다.

[663] 적응적 루프내 필터 세트에 포함되는 서로 다른 적응적 루프내 필터의 개수의 최대값인 `NumAlfFilters`은 N일 수 있다. 여기서, N은 양의 정수일 수 있고, 예를 들어 25일 수 있다.

[664] `alf_luma_clip_flag`는 휘도 성분에 대해 선형(linear) 적응적 루프내 필터링이 수행되는지 또는 비선형(non-linear) 적응적 루프내 필터링이 수행되는지를 의미하는 휘도 절삭 플래그일 수 있다.

[665] 예를 들어, `alf_luma_clip_flag`가 제1 값(예: 0)인 경우 휘도 성분에 대해 선형 적응적 루프내 필터링이 수행될 수 있다. 그리고, `alf_luma_clip_flag`가 제2 값(예: 1)인 경우 휘도 성분에 대해 비선형 적응적 루프내 필터링이 수행될 수 있다.

[666] `alf_luma_clip_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_clip_flag`는 제1 값(예: 0)으로 추론될 수 있다.

[667] `alf_luma_num_filters_signalled_minus1`는 시그널링되는 휘도 ALF의 개수를 나타내는 휘도 시그널링 ALF 개수 정보를 의미할 수 있다. 그리고 `alf_luma_num_filters_signalled_minus1`에 +1한 값은 휘도 시그널링 ALF의 개수를 의미할 수 있다.

- [668] 상기 `alf_luma_num_filters_signalled_minus1`는 0에서 `NumAlfFilters - N`까지의 값을 가질 수 있다. 여기서, `N`은 양의 정수일 수 있고, 예를 들어 1일 수 있다.
- [669] `alf_luma_num_filters_signalled_minus1`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_num_filters_signalled_minus1`는 0의 값으로 추론될 수 있다.
- [670] `alf_luma_coeff_delta_idx[ filtIdx ]`는 `filtIdx`에 대응되는 휘도 적응적 루프내 필터가 참조하는 휘도 시그널링 적응적 루프내 필터의 색인을 나타낼 수 있다. `alf_luma_coeff_delta_idx[ filtIdx ]`는 휘도 ALF 델타 색인을 의미할 수 있다. 상기 휘도 ALF 델타 색인은 휘도 성분에 대한 필터 계수 차분 색인을 의미할 수 있다.
- [671] 상기 `filtIdx`는 0에서 `NumAlfFilters - N`까지의 값을 가질 수 있다. 여기서, `N`은 양의 정수일 수 있고, 예를 들어 1일 수 있다.
- [672] `alf_luma_coeff_delta_idx[ filtIdx ]`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_coeff_delta_idx[ filtIdx ]`는 0의 값으로 추론될 수 있다.
- [673] `alf_luma_use_fixed_filter_flag`는 적응적 루프내 필터 계수를 시그널링할 때 고정된 필터를 사용하는지를 의미할 수 있다.
- [674] 예를 들어, `alf_luma_use_fixed_filter_flag`가 제1 값(예: 0)인 경우 적응적 루프내 필터 계수를 시그널링할 때 고정된 필터가 사용되지 않을 수 있다. 그리고, `alf_luma_use_fixed_filter_flag`가 제2 값(예: 1)인 경우, 적응적 루프내 필터 계수를 시그널링할 때 고정된 필터가 사용될 수 있다.
- [675] `alf_luma_use_fixed_filter_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_use_fixed_filter_flag`는 제1 값(예: 0)으로 추론될 수 있다.
- [676] `alf_luma_fixed_filter_set_idx`는 고정된 필터 세트 색인을 의미할 수 있다.
- [677] 상기 `alf_luma_fixed_filter_set_idx`는 0에서 `N`까지의 값을 가질 수 있다. 여기서, `N`은 양의 정수일 수 있고, 예를 들어 15일 수 있다.
- [678] `alf_luma_fixed_filter_set_idx`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_fixed_filter_set_idx`는 0의 값으로 추론될 수 있다.
- [679] `alf_luma_fixed_filter_pred_present_flag`는 `alf_luma_fixed_filter_pred_flag[ i ]`가 비트스트림 내에 존재하는지 여부를 의미할 수 있다.
- [680] 예를 들어, `alf_luma_fixed_filter_pred_present_flag`가 제1 값(예: 0)인 경우 `alf_luma_fixed_filter_pred_flag[ i ]`가 비트스트림 내에 존재하지 않을 수 있다. 그리고, `alf_luma_fixed_filter_pred_present_flag`가 제2 값(예: 1)인 경우 `alf_luma_fixed_filter_pred_flag[ i ]`가 비트스트림 내에 존재할 수 있다.
- [681] 또한, `alf_luma_fixed_filter_pred_present_flag`는 휘도 성분에 대한 적응적 루프내 필터 계수 클래스(종류) 전체가 고정된 필터로부터 예측되어 시그널링 되는지를 의미할 수 있다. 예를 들어, `alf_luma_fixed_filter_pred_present_flag`가 제1 값(예: 0)인 경우, 휘도 성분에 대한 적응적 루프내 필터 계수 클래스(종류) 전체가 고정된 필터로부터 예측하여 시그널링 되지 않을 수 있다. `alf_luma_fixed_filter_pred_present_flag`가 제2 값(예: 1)인 경우 휘도 성분에 대한 적응적 루프내 필터 계수 클래스(종류) 중 적어도 하나를 고정된 필터로부터

예측하여 시그널링될 수 있다.

[682] `alf_luma_fixed_filter_pred_present_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_fixed_filter_pred_present_flag`는 제1 값(예: 0)으로 추론될 수 있다.

[683] `alf_luma_fixed_filter_pred_flag[i]`는 i 번째 적응적 루프내 필터 계수 클래스(종류)가 고정된 필터로부터 예측되어 시그널링되는지를 의미할 수 있다.

[684] 예를 들어, `alf_luma_fixed_filter_pred_flag[i]`가 제1 값(예: 0)인 경우 i 번째 적응적 루프내 필터 계수 클래스(종류)가 고정된 필터로부터 예측되지 않을 수 있다. 그리고, `alf_luma_fixed_filter_pred_flag[i]`가 제2 값(예: 1)인 경우 i 번째 적응적 루프내 필터 계수 클래스(종류)가 고정된 필터로부터 예측될 수 있다.

[685] `alf_luma_fixed_filter_pred_flag[i]`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_fixed_filter_pred_flag[i]`는 제2 값(예: 1)으로 추론될 수 있다.

[686] 일 실시 예에 따르면, 도 60a 내지 60d과 다르게, 적응적 루프내 필터 데이터 선택스는 고정된 필터에 관한 정보(`alf_luma_use_fixed_filter_flag`, `alf_luma_fixed_filter_set_idx`, `alf_luma_fixed_filter_pred_present_flag`, `alf_luma_fixed_filter_pred_flag`)를 포함하지 않을 수 있다. 따라서 고정된 필터에 관한 정보는 시그널링되지 않을 수 있다. 그러므로 부호화기/복호화기에서 미리 설정된 고정된 필터가 적응적 루프내 필터링에 사용될 수 있다. 또한, 고정된 필터로부터 필터 계수를 예측하여 시그널링되지 않을 수 있다. 부호화기/복호화기에서 미리 설정된 고정된 필터만 적응적 루프내 필터링에 사용되고, 고정된 필터로부터 예측된 필터 계수는 적응적 루프내 필터링에 사용되지 않을 수 있다.

[687] `alf_luma_coeff_delta_flag`는 `alf_luma_coeff_delta_prediction_flag`가 시그널링되지 않는지를 의미할 수 있다.

[688] 예를 들어, `alf_luma_coeff_delta_flag`가 제1 값(예: 0)인 경우 `alf_luma_coeff_delta_prediction_flag`가 시그널링될 수 있고, `alf_luma_coeff_delta_flag`가 제2 값(예: 1)인 경우 `alf_luma_coeff_delta_prediction_flag`가 시그널링되지 않을 수 있다.

[689] `alf_luma_coeff_delta_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_coeff_delta_flag`는 제2 값(예: 1)으로 추론될 수 있다.

[690] `alf_luma_coeff_delta_prediction_flag`는 시그널링되는 휘도 성분에 대한 적응적 루프내 필터 계수가 이전에 시그널링된 휘도 성분에 대한 적응적 루프내 필터 계수로부터 예측되는지를 의미할 수 있다.

[691] 예를 들어, `alf_luma_coeff_delta_prediction_flag`가 제1 값(예: 0)인 경우 시그널링되는 휘도 성분에 대한 적응적 루프내 필터 계수가 이전에 시그널링된 휘도 성분에 대한 적응적 루프내 필터 계수로부터 예측되지 않을 수 있다. 그리고, `alf_luma_coeff_delta_prediction_flag`가 제2 값(예: 1)인 경우 시그널링되는 휘도 성분에 대한 적응적 루프내 필터 계수가 이전에 시그널링된 휘도 성분에 대한 적응적 루프내 필터 계수로부터 예측될 수 있다.

- [692] `alf_luma_coeff_delta_prediction_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_coeff_delta_prediction_flag`는 제1 값(예: 0)으로 추론될 수 있다.
- [693] `alf_luma_min_eg_order_minus1`에 +1한 값은 휘도 성분에 대한 적응적 루프내 필터 계수를 시그널링할 때 사용되는 지수-골롬 부호(exponential Golomb code)의 최소 차수(order)를 의미할 수 있다.
- [694] `alf_luma_eg_order_increase_flag[i]`는 휘도 성분에 대한 적응적 루프내 필터 계수를 시그널링할 때 사용되는 지수-골롬 부호의 차수가 1만큼 증가하는지 여부를 의미할 수 있다.
- [695] `alf_luma_coeff_delta_abs[sfIdx][j]`의 값을 엔트로피 부호화/복호화하기 위해 사용되는 지수-골롬 부호의 차수 `expGoOrderY[i]`는 아래와 같이 유도될 수 있다.
- [696]
$$\text{expGoOrderY}[i] = (i == 0 ? \text{alf_luma_min_eg_order_minus1} + 1 : \text{expGoOrderY}[i - 1]) + \text{alf_luma_eg_order_increase_flag}[i]$$
- [697] `alf_luma_coeff_flag[sfIdx]`는 `sfIdx`에 의해 지시되는 휘도 성분에 대한 적응적 루프내 필터가 시그널링되는지를 의미할 수 있다.
- [698] 예를 들어, `alf_luma_coeff_flag[sfIdx]`가 제1 값(예: 0)인 경우 `sfIdx`에 의해 지시되는 휘도 성분에 대한 적응적 루프내 필터가 모두 0의 값으로 설정될 수 있다. 그리고, `alf_luma_coeff_flag[sfIdx]`가 제2 값(예: 1)인 경우 `sfIdx`에 의해 지시되는 휘도 성분에 대한 적응적 루프내 필터가 시그널링될 수 있다.
- [699] `alf_luma_coeff_flag[sfIdx]`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_coeff_flag[sfIdx]`는 제2 값(예: 1)으로 추론될 수 있다.
- [700] 일 실시 예에 따르면, 도 60a 내지 60d과 다르게, `alf_luma_coeff_flag[sfIdx]`는 적응적 루프내 필터 데이터 선택스에 포함되지 않을 수 있다. 즉, `alf_luma_coeff_flag[sfIdx]`는 적응적 루프내 필터 데이터 선택스에서 부호화/복호화되지 않을 수 있다. 따라서, `alf_luma_coeff_flag[sfIdx]`의 부호화/복호화/획득 없이, 적응적 루프내 필터의 계수 정보가 부호화/복호화/획득됨이 결정될 수 있다.
- [701] `alf_luma_coeff_delta_abs[sfIdx][j]`는 `sfIdx`에 의해 지시되는 휘도 성분에 대한 적응적 루프내 필터의 j번째 계수 차분(delta)의 절대값을 의미할 수 있다.
- [702] `alf_luma_coeff_delta_abs[sfIdx][j]`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_coeff_delta_abs[sfIdx][j]`는 0의 값으로 추론될 수 있다.
- [703] 지수-골롬 부호(이진화) `uek(v)`의 차수 k는 아래와 같이 유도될 수 있다.
- [704]
$$\text{golombOrderIdxY}[] = \{ 0, 0, 1, 0, 0, 1, 2, 1, 0, 0, 1, 2 \}$$
- [705]
$$k = \text{expGoOrderY}[\text{golombOrderIdxY}[j]]$$
- [706] 일 실시 예에 따라, `alf_luma_coeff_delta_abs[sfIdx][j]`의 엔트로피 부호화/복호화/이진화를 위한 지수-골롬 부호 `uek(v)`의 차수 k는 0으로 고정될 수 있다. 따라서, `alf_luma_coeff_delta_abs[sfIdx][j]`의 엔트로피 부호화/복호화/이진화를 위하여 0차 지수-골롬 부호 `ue(v)`가 적용될 수 있다.

alf_luma_coeff_delta_abs[sfIdx][j]의 값은 0을 포함한 양의 정수로, 0 내지 128의 범위 내에서 값을 가질 수 있다.

[707] alf_luma_coeff_delta_sign[sfIdx][j]는 sfIdx에 의해 지시되는 휘도 성분에 대한 적응적 루프내 필터의 j번째 계수 또는 계수 차분의 부호(sign)를 의미할 수 있다. 또한, alf_luma_coeff_delta_sign[sfIdx][j]는 아래와 같이 유도될 수 있다.

[708] alf_luma_coeff_delta_sign[sfIdx][j]가 제1 값(예: 0)인 경우, 대응하는 휘도 성분에 대한 적응적 루프내 필터 계수 또는 계수 차분은 양의 부호(positive sign)를 가질 수 있다.

[709] alf_luma_coeff_delta_sign[sfIdx][j]가 제2 값(예: 1)인 경우, 대응하는 휘도 성분에 대한 적응적 루프내 필터 계수 또는 계수 차분은 음의 부호(negative sign)를 가질 수 있다.

[710] alf_luma_coeff_delta_sign[sfIdx][j]가 비트스트림 내에 존재하지 않을 경우 alf_luma_coeff_delta_sign[sfIdx][j]는 제1 값(예: 0)으로 추론될 수 있다.

[711] 일 실시 예에 따르면, 적응적 루프내 필터의 필터 계수 차분 값 대신 필터 계수의 값이 직접 시그널링 될 수 있다. 예를 들어, 필터 계수의 절대 값 정보와 부호 정보가 적응적 루프내 필터 데이터 선택스에 포함될 수 있다. 즉, 필터 계수의 절대 값 정보와 부호 정보가 적응적 루프내 필터 데이터 선택스에서 부호화/복호화될 수 있다. 상기 필터 계수의 절대 값 정보의 부호화/복호화를 위하여, 0차 지수-골롬 부호 ue(v)가 사용될 수 있다. 필터 계수의 절대 값 정보의 값은 0을 포함한 양의 정수로, 0 내지 128의 범위 내에서 값을 가질 수 있다. 상기 적응적 루프내 필터의 필터 계수는 휘도 성분에 대한 적응적 루프내 필터의 필터 계수 및 색차 성분에 대한 적응적 루프내 필터의 필터 계수 중 적어도 하나를 포함할 수 있다.

[712] 휘도 시그널링 ALF인 filtCoeff[sfIdx][j]는 아래와 같이 유도될 수 있다. 이때, sfIdx는 0에서 alf_luma_num_filters_signalled_minus1까지의 값을 가질 수 있다. 또한, j는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예를 들어 11일 수 있다.

[713]
$$\text{filtCoeff}[\text{sfIdx}][j] = \text{alf_luma_coeff_delta_abs}[\text{sfIdx}][j] * (1 - 2 * \text{alf_luma_coeff_delta_sign}[\text{sfIdx}][j])$$

[714] alf_luma_coeff_delta_prediction_flag가 제1 값(예: 1)인 경우, filtCoeff[sfIdx][j]는 아래와 같이 유도될 수 있다. 이때, sfIdx는 1에서 alf_luma_num_filters_signalled_minus1까지의 값을 가질 수 있다. 또한, j는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예를 들어 11일 수 있다.

[715]
$$\text{filtCoeff}[\text{sfIdx}][j] += \text{filtCoeff}[\text{sfIdx} - 1][j]$$

[716] 일 실시 예에 따라, 도 60a 내지 60d과 다르게, alf_luma_coeff_delta_prediction_flag가 적응적 루프내 필터 데이터 선택스에 포함되지 않을 수 있다. 즉, alf_luma_coeff_delta_prediction_flag가 적응적 루프내

필터 데이터 신택스에서 부호화/복호화되지 않을 수 있다. 따라서, 적응적 루프내 필터의 휘도 성분 필터 계수 델타 값은 예측 부호화/복호화되는 대신, 직접 시그널링될 수 있다.

- [717] 휘도 성분에 대한 적응적 루프내 필터 계수 $\text{AlfCoeffL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j]$ 는 아래와 같이 유도될 수 있다. 이때, filtIdx 는 0에서 $\text{NumAlfFilters} - 1$ 까지의 값을 가질 수 있다. 또한, j 는 0에서 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 11일 수 있다.
- [718] $\text{AlfCoeffL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j] = \text{filtCoeff}[\text{alf_luma_coeff_delta_idx}[\text{filtIdx}][j]]$
- [719] $\text{alf_luma_use_fixed_filter_flag}$ 가 제2 값(예: 1)을 가지고, $\text{alf_luma_fixed_filter_pred_flag}[\text{filtIdx}]$ 가 제2 값(예: 1)을 가지는 경우, $\text{AlfCoeffL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j]$ 는 아래의 예와 같이 유도될 수 있다. 이때, filtIdx 는 0에서 $\text{NumAlfFilters} - 1$ 까지의 값을 가질 수 있다. 또한, j 는 0에서 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 11일 수 있다.
- [720] $\text{AlfCoeffL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j] += \text{AlfFixFiltCoeff}[\text{AlfClassToFiltMap}[\text{alf_luma_fixed_filter_set_idx}][\text{filtIdx}][j]]$
- [721] 고정된 필터 계수 $\text{AlfFixFiltCoeff}[i][j]$ 는 도 22a 및 22b의 예와 같이 유도될 수 있다. 이때, i 는 0에서 imax 까지의 값을 가질 수 있다. 또한, j 는 0에서 N 까지의 값을 가질 수 있다. 여기서, imax 는 양의 정수일 수 있고, 예를 들어 64일 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 11일 수 있다.
- [722] 적응적 루프내 필터 계수 클래스와 필터의 매핑 관계 $\text{AlfClassToFiltMap}[m][n]$ 는 도 23의 예와 같이 유도될 수 있다. 이때, m 은 0에서 mmax 까지의 값을 가질 수 있다. 또한, n 은 0에서 nmax 까지의 값을 가질 수 있다. 여기서, mmax 는 양의 정수일 수 있고, 예를 들어 15일 수 있다. 여기서, nmax 는 양의 정수일 수 있고, 예를 들어 24일 수 있다.
- [723] 상기 $\text{AlfCoeffL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j]$ 는 $-2M$ 에서 $2M - 1$ 까지의 값을 가질 수 있다. 이때, filtIdx 는 0에서 $\text{NumAlfFilters} - 1$ 까지의 값을 가질 수 있다. 또한, j 는 0에서 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 11일 수 있다. 이때, M 은 양의 정수일 수 있고, 예를 들어 7일 수 있다.
- [724] $\text{alf_luma_clip_min_eg_order_minus1}$ 에 +1한 값은 휘도 성분에 대한 적응적 루프내 필터 절삭 색인(clipping index)을 시그널링할 때 사용되는 지수-골롬 부호의 최소 차수를 의미할 수 있다. 상기 절삭 색인은 비선형 적응적 루프내 필터일 경우에 시그널링되거나 사용될 수 있다.
- [725] $\text{alf_luma_clip_eg_order_increase_flag}[i]$ 는 휘도 성분에 대한 적응적 루프내 필터 절삭 색인을 시그널링할 때 사용되는 지수-골롬 부호의 차수가 1만큼

증가하는지 여부를 의미할 수 있다.

[726] $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 를 엔트로피 부호화/복호화하기 위해 사용되는
지수-골롬 부호의 차수 $k\text{ClipY}[i]$ 는 아래와 같이 유도될 수 있다.

[727] $k\text{ClipY}[i] = (i == 0 ? \text{alf_luma_clip_min_eg_order_minus1} + 1 : k\text{ClipY}[i - 1]) +$
 $\text{alf_luma_clip_eg_order_increase_flag}[i]$

[728] $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 는 sfIdx 에 의해 지시되는 휘도 성분에 대한 적응적
루프내 필터의 j 번째 계수가 복원/복호 샘플에 곱해지기 전에 절삭에 사용되는
절삭 값(clipping value)에 대한 휘도 절삭 색인을 의미할 수 있다. 상기 절삭 값은
 $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 가 나타내는 휘도 절삭 색인 및 영상의 비트 템스에
의하여 결정될 수 있다. 여기서, 비트 템스는 시퀀스, 픽처, 슬라이스, 타일 그룹,
타일, CTU 등 중 적어도 하나의 단위에서 결정된 비트 템스를 의미할 수 있다.

[729] $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 가 비트스트림 내에 존재하지 않을 경우
 $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 는 제1 값(예: 0)으로 추론될 수 있다.

[730] 상기 $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 는 0에서 M 까지의 값을 가질 수 있다. 여기서,
 M 은 양의 정수일 수 있고, 예를 들어 3일 수 있다. 이때, sfIdx 는 0에서
 $\text{alf_luma_num_filters_signalled_minus1}$ 까지의 값을 가질 수 있다. 또한, j 는 0에서
 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 11일 수
있다.

[731] 지수-골롬 부호(이진화) $\text{uek}(v)$ 의 차수 k 는 아래와 같이 유도될 수 있다.

[732] $k = k\text{ClipY}[\text{golombOrderIdxY}[j]]$

[733] $\text{filterClips}[\text{sfIdx}][j]$ 는 아래와 같이 유도될 수 있다. 이때, sfIdx 는 0에서
 $\text{alf_luma_num_filters_signalled_minus1}$ 까지의 값을 가질 수 있다. 또한, j 는 0에서
 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 11일 수
있다.

[734] $\text{filterClips}[\text{sfIdx}][j] = \text{Round}(2(\text{BitDepthY} * (M + 1 - \text{alf_luma_clip_idx}[\text{sfIdx}][j]) / (M + 1)))$ 또는 $\text{filterClips}[\text{sfIdx}][j] = \text{Round}(2(\text{BitDepthY} - 8) * 2(8 * (M - \text{alf_luma_clip_idx}[\text{sfIdx}][j]) / M))$

[735] 여기서, BitDepthY 는 휘도 성분에 대한 입력 비트 템스/심도(depth)를 의미할 수
있다. 여기서, 비트 템스는 시퀀스, 픽처, 슬라이스, 타일 그룹, 타일, CTU 등 중
적어도 하나의 단위에서 결정된 비트 템스를 의미할 수 있다.

[736] 여기서, M 은 양의 정수일 수 있고, 예를 들어 3일 수 있다. 또한, M 은
 $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 가 가질 수 있는 최대값을 의미할 수 있다.

[737] 휘도 성분에 대한 적응적 루프내 필터 절삭 값 $\text{AlfClipL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j]$ 은 아래와 같이 유도될 수 있다. 이때,
 filtIdx 는 0에서 $\text{NumAlfFilters} - 1$ 까지의 값을 가질 수 있다. 또한, j 는 0에서
 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 11일 수
있다.

[738] $\text{AlfClipL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j] = \text{filterClips}[\text{sfIdx}][j]$

alf_luma_coeff_delta_idx[filtIdx][j]

[739] 상기 alf_luma_clip_idx[sfIdx][j]에 사용되는 지수-골롬 부호는 값이 가질 수 있는 범위가 클 경우 효율적으로 엔트로피 부호화/복호화할 수 있는 방법이다. 하지만, 상기 예와 같이 alf_luma_clip_idx[sfIdx][j]가 가질 수 있는 값의 범위(0에서 3까지)가 상대적으로 작은 경우, alf_luma_clip_idx[sfIdx][j]에 사용되는 지수-골롬 부호는 엔트로피 부호화/복호화 측면에서 비효율적일 수 있다.

[740] 따라서, alf_luma_clip_min_eg_order_minus1 및 alf_luma_clip_eg_order_increase_flag[i] 중 적어도 하나의 구문 요소를 사용하는 대신, alf_luma_clip_idx[sfIdx][j]가 가질 수 있는 값의 범위(0에서 3까지)가 상대적으로 작은 경우, tu(3), f(2), u(2), tb(3) 등 중 적어도 하나의 엔트로피 부호화/복호화 방법을 사용하여 alf_luma_clip_idx[sfIdx][j]를 엔트로피 부호화/복호화할 수 있다.

[741] alf_chroma_clip_flag는 색차 성분에 대해 선형 적응적 루프내 필터링이 수행되는지 또는 비선형 적응적 루프내 필터링이 수행되는지를 의미하는 색차 절삭 플래그일 수 있다. 즉, alf_chroma_clip_flag는 시그널링되는 색차 ALF마다 각각 시그널링되어 적용되지 않고, alf_chroma_clip_flag는 시그널링되는 전체 색차 ALF에 대해 한 번만 시그널링되어 적용될 수 있다. 이때, 색차 ALF는 색차 성분에 대한 적응적 루프내 필터를 의미할 수 있다.

[742] 예를 들어, alf_chroma_clip_flag가 제1 값(예: 0)인 경우, 색차 성분에 대해 선형 적응적 루프내 필터링이 수행될 수 있다. 그리고, alf_chroma_clip_flag가 제2 값(예: 1)인 경우 색차 성분에 대해 비선형 적응적 루프내 필터링이 수행될 수 있다.

[743] alf_chroma_clip_flag가 비트스트림 내에 존재하지 않을 경우 alf_chroma_clip_flag는 제1 값(예: 0)으로 추론될 수 있다.

[744] alf_chroma_min_eg_order_minus1에 +1한 값은 색차 성분에 대한 적응적 루프내 필터 계수를 시그널링할 때 사용되는 지수-골롬 부호의 최소 차수를 의미할 수 있다.

[745] alf_chroma_eg_order_increase_flag[i]는 색차 성분에 대한 적응적 루프내 필터 계수를 시그널링할 때 사용되는 지수-골롬 부호의 차수가 1만큼 증가하는지 여부를 의미할 수 있다.

[746] alf_chroma_coeff_abs[j]의 값을 엔트로피 부호화/복호화하기 위해 사용되는 지수-골롬 부호의 차수 expGoOrderC[i]는 아래와 같이 유도될 수 있다.

[747]
$$\text{expGoOrderC}[i] = (i == 0 ? \text{alf_chroma_min_eg_order_minus1} + 1 : \text{expGoOrderC}[i - 1]) + \text{alf_chroma_eg_order_increase_flag}[i]$$

[748] alf_chroma_coeff_abs[j]는 색차 성분에 대한 적응적 루프내 필터의 j번째 계수의 절대값을 의미할 수 있다.

[749] alf_chroma_coeff_abs[j]가 비트스트림 내에 존재하지 않을 경우

- alf_chroma_coeff_abs[j]는 0의 값으로 추론될 수 있다.
- [750] 상기 alf_chroma_coeff_abs[j]는 0에서 $2M - 1$ 까지의 값을 가질 수 있다. 이때, M은 양의 정수일 수 있고, 예를 들어 7일 수 있다.
- [751] 지수-골롬 부호(이진화) uek(v)의 차수 k는 아래와 같이 유도될 수 있다.
- [752] $\text{golombOrderIdxC}[] = \{ 0, 0, 1, 0, 0, 1 \}$
- [753] $k = \text{expGoOrderC}[\text{golombOrderIdxC}[j]]$
- [754] 일 실시 예에 따라, alf_chroma_coeff_abs[j]의 엔트로피 부호화/복호화/이진화를 위한 지수-골롬 부호 uek(v)의 차수 k는 0으로 고정될 수 있다. 따라서, alf_chroma_coeff_abs[j]의 엔트로피 부호화/복호화/이진화를 위하여 0차 지수-골롬 부호 ue(v)가 사용될 수 있다. alf_chroma_coeff_abs[j]의 값은 0을 포함한 양의 정수로, 0 내지 128의 범위 내의 값을 가질 수 있다.
- [755] alf_chroma_coeff_sign[j]는 색차 성분에 대한 적응적 루프내 필터의 j번째 계수 또는 계수 차분의 부호를 의미할 수 있다. 또한, alf_chroma_coeff_sign[j]는 아래와 같이 유도될 수 있다.
- [756] alf_chroma_coeff_sign[j]가 제1 값(예: 0)인 경우, 대응하는 색차 성분에 대한 적응적 루프내 필터 계수 또는 계수 차분은 양의 부호를 가질 수 있다.
- [757] alf_chroma_coeff_sign[j]가 제2 값(예: 1)인 경우, 대응하는 색차 성분에 대한 적응적 루프내 필터 계수 또는 계수 차분은 음의 부호를 가질 수 있다.
- [758] alf_chroma_coeff_sign[j]가 비트스트림 내에 존재하지 않을 우 alf_chroma_coeff_sign[j]는 제1 값(예: 0)으로 추론될 수 있다.
- [759]
- [760] 색차 성분에 대한 적응적 루프내 필터 계수(색차 시그널링 ALF) AlfCoeffC[adaptation_parameter_set_id][j]는 아래와 같이 유도될 수 있다. 이때, j는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예를 들어 5일 수 있다.
- [761] $\text{AlfCoeffC}[\text{adaptation_parameter_set_id}][j] = \text{alf_chroma_coeff_abs}[j] * (1 - 2 * \text{alf_chroma_coeff_sign}[j])$
- [762] 상기 AlfCoeffC[adaptation_parameter_set_id][j]는 $-2M - 1$ 에서 $2M - 1$ 까지의 값을 가질 수 있다. 이때, j는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예를 들어 5일 수 있다. 이때, M은 양의 정수일 수 있고, 예를 들어 7일 수 있다. 또는, 상기 AlfCoeffC[adaptation_parameter_set_id][j]는 $-2M$ 에서 $2M - 1$ 까지의 값을 가질 수 있다.
- [763] alf_chroma_clip_min_eg_order_minus1에 +1한 값은 색차 성분에 대한 적응적 루프내 필터 절삭 색인을 시그널링할 때 사용되는 지수-골롬 부호의 최소 차수를 의미할 수 있다. 상기 절삭 색인은 비선형 적응적 루프내 필터일 경우에 시그널링되거나 사용될 수 있다.
- [764] alf_chroma_clip_eg_order_increase_flag[i]는 색차 성분에 대한 적응적 루프내 필터 절삭 색인을 시그널링할 때 사용되는 지수-골롬 부호의 차수가 1만큼

증가하는지 여부를 의미할 수 있다.

[765] $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화하기 위해 사용되는
지수-골롬 부호의 차수 $k\text{ClipC}[i]$ 는 아래와 같이 유도될 수 있다.

[766] $k\text{ClipC}[i] = (i == 0 ? \text{alf_chroma_clip_min_eg_order_minus1} + 1 : k\text{ClipC}[i - 1])$
 $+ \text{alf_chroma_clip_eg_order_increase_flag}[i]$

[767] $\text{alf_chroma_clip_idx}[j]$ 는 색차 성분에 대한 적응적 루프내 필터의 j 번째 계수가
복원/복호 샘플에 곱해지기 전에 절삭에 사용되는 절삭 값에 대한 색차 절삭
색인을 의미할 수 있다. 상기 절삭 값은 $\text{alf_chroma_clip_idx}[j]$ 가 나타내는 색차
절삭 색인 및 영상의 비트 템스에 의하여 결정될 수 있다. 여기서, 비트 템스는
시퀀스, 픽처, 슬라이스, 타일 그룹, 타일, CTU 등 중 적어도 하나의 단위에서
결정된 비트 템스를 의미할 수 있다.

[768] $\text{alf_chroma_clip_idx}[j]$ 가 비트스트림 내에 존재하지 않을 경우
 $\text{alf_chroma_clip_idx}[j]$ 는 제1 값(예: 0)으로 추론될 수 있다.

[769] 상기 $\text{alf_chroma_clip_idx}[j]$ 는 0에서 M 까지의 값을 가질 수 있다. 여기서, M 은
양의 정수일 수 있고, 예를 들어 3일 수 있다. 이때, j 는 0에서 N 까지의 값을 가질
수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 5일 수 있다.

[770] 지수-골롬 부호(이진화) $\text{uek}(v)$ 의 차수 k 는 아래와 같이 유도될 수 있다.

[771] $k = k\text{ClipC}[\text{golombOrderIdxC}[j]]$

[772] 일 실시 예에 따르면, 색차 성분의 절삭 색인이 나타내는 절삭 값은 휘도 성분의
절삭 색인이 나타내는 절삭 값과 동일한 방식으로 결정될 수 있다. 즉, 색차
성분에 대한 절삭 값과 휘도 성분에 대한 절삭 값은 절삭 색인 및 비트 템스 중
적어도 하나에 기반하여 동일한 방식으로 결정될 수 있다. 예를 들어, 색차
성분의 절삭 색인이 3일 때의 절삭 값이 휘도 성분의 절삭 색인이 3일 때의 절삭
값과 동일하도록 결정될 수 있다. 따라서, 절삭 색인과 비트 템스가 동일할 때,
휘도 성분 샘플에 대한 절삭 프로세스와 색차 성분 샘플에 대한 절삭 프로세스는
동일한 절삭 값에 의하여 수행될 수 있다.

[773] 또한, 상기 절삭 값은 비트 템스에 기반하여 설정될 수 있다. 예를 들어, 절삭
값은 $2 \ll (\text{BitDepth} - N)$ 의 값으로 설정될 수 있다. 여기서, N 은 0을 포함한 양의
정수일 수 있다. 상기 절삭 색인 값에 따라 N 이 결정될 수 있다. 상기 절삭 색인이
증가할수록 N 도 증가할 수 있다. 여기서, 상기 BitDepth 는 비트 템스를 의미할 수
있다.

[774] 색차 성분에 대한 적응적 루프내 필터 절삭 값 $\text{AlfClipC}[\text{adaptation_parameter_set_id}][j]$
은 아래와 같이 유도될 수 있다. 이때, j 는 0에서
 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 5일 수
있다.

[775] $\text{AlfClipC}[\text{adaptation_parameter_set_id}][j] = \text{Round}(2(\text{BitDepthC} - 8) * 2(8 * (M - \text{alf_chroma_clip_idx}[j]) / M))$ 또는 $\text{AlfClipC}[\text{adaptation_parameter_set_id}][j] = \text{Round}(2(\text{BitDepthC} * (M + 1 - \text{alf_chroma_clip_idx}[j]) / (M + 1)))$

- [776] 여기서, BitDepthC는 색차 성분에 대한 입력 비트 템스/심도(depth)를 의미할 수 있다. 여기서, 비트 템스는 시퀀스, 픽처, 슬라이스, 타일 그룹, 타일, CTU 등 중 적어도 하나의 단위에서 결정된 비트 템스를 의미할 수 있다.
- [777] 여기서, M은 양의 정수일 수 있고, 예를 들어 3일 수 있다. 또한, M은 $\text{alf_chroma_clip_idx}[j]$ 가 가질 수 있는 최대값을 의미할 수 있다.
- [778] 상기 $\text{alf_chroma_clip_idx}[j]$ 에 사용되는 지수-골롬 부호는 값이 가질 수 있는 범위가 클 경우 효율적으로 엔트로피 부호화/복호화할 수 있는 방법이다. 하지만, 상기 예와 같이 $\text{alf_chroma_clip_idx}[j]$ 가 가질 수 있는 값의 범위(0에서 3까지)가 상대적으로 작은 경우, $\text{alf_chroma_clip_idx}[j]$ 에 사용되는 지수-골롬 부호는 엔트로피 부호화/복호화 측면에서 비효율적일 수 있다.
- [779] 따라서, $\text{alf_chroma_clip_min_eg_order_minus1}$ 및 $\text{alf_chroma_clip_eg_order_increase_flag}[i]$ 중 적어도 하나의 구문 요소를 사용하는 대신, $\text{alf_chroma_clip_idx}[j]$ 가 가질 수 있는 값의 범위(0에서 3까지)가 상대적으로 작은 경우, tu(3), f(2), u(2), tb(3) 등 중 적어도 하나의 엔트로피 부호화/복호화 방법을 사용하여 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화할 수 있다.
- [780] 도 21의 예와 같이, $\text{alf_ctb_flag}[cIdx][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 는 휘도 성분 위치 (xCtb, yCtb)에서 cIdx에 의해 지시되는 색 성분(Y, Cb, 또는 Cr) 부호화 트리 블록(coding tree block)에서 적응적 루프내 필터링이 사용되는지를 의미할 수 있다.
- [781] 예를 들어, $\text{alf_ctb_flag}[0][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 는 현재 부호화 트리 블록의 휘도 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제1 ALF 부호화 트리 블록 플래그일 수 있다. $\text{alf_ctb_flag}[0][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 가 제1 값(예: 0)인 경우 휘도 성분 위치 (xCtb, yCtb)에서 휘도 성분 부호화 트리 블록(coding tree block)에서 적응적 루프내 필터링이 사용되지 않을 수 있다. 그리고, $\text{alf_ctb_flag}[0][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 가 제2 값(예: 1)인 경우 휘도 성분 위치 (xCtb, yCtb)에서 휘도 성분 부호화 트리 블록(coding tree block)에서 적응적 루프내 필터링이 사용될 수 있다.
- [782] $\text{alf_ctb_flag}[0][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 가 비트스트림 내에 존재하지 않을 경우 $\text{alf_ctb_flag}[0][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 는 제1 값(예: 0)으로 추론될 수 있다.
- [783] 예를 들어, $\text{alf_ctb_flag}[1][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 는 현재 부호화 트리 블록의 Cb 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제2 ALF 부호화 트리 블록 플래그일 수 있다. $\text{alf_ctb_flag}[1][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$ 가 제1 값(예: 0)인 경우 휘도 성분 위치 (xCtb, yCtb)에서 Cb 부호화 트리 블록(coding tree block)에서 적응적 루프내 필터링이 사용되지 않을 수 있다. 그리고, $\text{alf_ctb_flag}[1][xCtb \gg \text{Log2CtbSize}][yCtb \gg \text{Log2CtbSize}]$

yCtb >> Log2CtbSize]가 제2 값(예: 1)인 경우 휘도 성분 위치 (xCtb, yCtb)에서 Cb 부호화 트리 블록(coding tree block)에서 적응적 루프내 필터링이 사용될 수 있다.

[784] alf_ctb_flag[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]가 비트스트림 내에 존재하지 않을 경우 alf_ctb_flag[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]는 제1 값(예: 0)으로 추론될 수 있다.

[785] 예를 들어, alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]는 현재 부호화 트리 블록의 Cr 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제3 ALF 부호화 트리 블록 플래그일 수 있다. alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]가 제1 값(예: 0)인 경우 휘도 성분 위치 (xCtb, yCtb)에서 Cr 부호화 트리 블록(coding tree block)에서 적응적 루프내 필터링이 사용되지 않을 수 있다. 그리고, alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]가 제2 값(예: 1)인 경우 휘도 성분 위치 (xCtb, yCtb)에서 Cr 부호화 트리 블록(coding tree block)에서 적응적 루프내 필터링이 사용될 수 있다.

[786] alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]가 비트스트림 내에 존재하지 않을 경우 alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]는 제1 값(예: 0)으로 추론될 수 있다.

[787] alf_ctb_use_first_aps_flag는 slice_alf_aps_id_luma[0]와 동일한 값을 가지는 adaptation_parameter_set_id가 지시하는 적응 파라미터 세트의 적응적 루프내 필터 정보를 사용하는지를 의미할 수 있다.

[788] 예를 들어, alf_ctb_use_first_aps_flag가 제1 값(예: 0)인 경우, slice_alf_aps_id_luma[0]와 동일한 값을 가지는 adaptation_parameter_set_id가 지시하는 적응 파라미터 세트의 적응적 루프내 필터 정보가 사용되지 않을 수 있다. 그리고, alf_ctb_use_first_aps_flag가 제2 값(예: 1)인 경우, slice_alf_aps_id_luma[0]와 동일한 값을 가지는 adaptation_parameter_set_id가 지시하는 적응 파라미터 세트의 적응적 루프내 필터 정보가 사용될 수 있다.

[789] alf_ctb_use_first_aps_flag가 비트스트림 내에 존재하지 않을 경우 alf_ctb_use_first_aps_flag는 제1 값(예: 0)으로 추론될 수 있다.

[790] 일 실시 예에 따르면, 도 21의 예와 다르게, alf_ctb_use_first_aps_flag는 적응적 루프내 필터 데이터 선택스에 포함되지 않을 수 있다. 즉, alf_ctb_use_first_aps_flag는 적응적 루프내 필터 데이터 선택스에서 부호화/복호화되지 않을 수 있다. 따라서, alf_ctb_use_first_aps_flag의 부호화/복호화/획득 없이, alf_luma_prev_filter_idx를 사용하여 현재 휘도 부호화 트리 블록에 적용되는 적응적 루프내 필터 세트가 결정될 수 있다.

[791] alf_use_aps_flag는 현재 부호화 트리 블록에 적응 파라미터 세트의 ALF 세트가 적용되는 여부를 나타내는 적응 파라미터 세트 적용 플래그를 의미할 수 있다.

[792] 예를 들어, alf_use_aps_flag가 제1 값(예: 0)인 경우 휘도 성분 부호화 트리

블록에서 고정된 필터 중 적어도 하나가 사용될 수 있다. 즉, 고정 ALF 세트 내의 적어도 하나의 고정된 필터가 현재 부호화 트리 블록에서 사용될 수 있다. 그리고, `alf_use_aps_flag`가 제2 값(예: 1)인 경우 휘도 성분 부호화 트리 블록에서 적어도 하나의 적응 파라미터 세트 내 적응적 루프내 필터 정보가 사용될 수 있다.

- [793] `alf_use_aps_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_use_aps_flag`는 제1 값(예: 0)으로 추론될 수 있다.
- [794] `alf_luma_prev_filter_idx_minus1`에 1을 더한 값은 이전 픽처들 중 적어도 하나의 픽처/서브 픽처/슬라이스/타일 그룹/브릭에서 사용한 적응 파라미터 세트 내 적응적 루프내 필터 정보 중 어떤 적응적 루프내 필터 정보를 사용하는지에 대한 색인을 의미할 수 있다.
- [795] 상기 `alf_luma_prev_filter_idx_minus1`는 0에서 `slice_num_alf_aps_ids_luma - N`까지의 값을 가질 수 있다. 여기서, `N`은 양의 정수일 수 있고, 예를 들어 1일 수 있다. 도 21에 따르면, `alf_ctb_use_first_aps_flag`가 현재 부호화 트리 블록에 적용되는 적응적 루프내 필터의 색인 값이 1인지 여부를 나타내기 때문에, `alf_luma_prev_filter_idx_minus1`는 색인 값이 2 이상인 적어도 2개의 적응적 루프내 필터 중에서 하나를 나타낸다. 따라서, 상기 `N`은 2로 결정될 수 있다. 그러므로 만약 `slice_num_alf_aps_ids_luma`가 2일 때, 색인 값이 2 이상인 적응적 루프내 필터가 1개이므로, `alf_luma_prev_filter_idx_minus1`는 시그널링되지 않을 수 있다.
- [796] `alf_luma_prev_filter_idx_minus1`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_prev_filter_idx_minus1`는 0의 값으로 추론될 수 있다.
- [797] 일 실시 예에 따르면, 도 21의 예와 다르게, `alf_ctb_use_first_aps_flag`가 부호화 트리 블록의 선택스 구조에 포함되지 않을 경우, `alf_luma_prev_filter_idx_minus1` 대신에 `alf_luma_prev_filter_idx`가 부호화 트리 블록의 선택스 구조에 포함될 수 있다. 즉, `alf_luma_prev_filter_idx`가 부호화 트리 블록의 선택스 구조에서 부호화/복호화될 수 있다. `alf_luma_prev_filter_idx`는 이전 픽처/서브 픽처/슬라이스/타일 그룹/브릭들 중 적어도 하나의 픽처/서브 픽처/슬라이스/타일 그룹/브릭에서 사용한 적응 파라미터 세트 내 적응적 루프내 필터 정보 중 어떤 적응적 루프내 필터 정보를 사용하는지에 대한 색인을 의미할 수 있다.
- [798] 상기 `alf_luma_prev_filter_idx`는 0에서 `slice_num_alf_aps_ids_luma - N`까지의 값을 가질 수 있다. 여기서, `N`은 양의 정수일 수 있고, 예를 들어 1일 수 있다. `alf_ctb_use_first_aps_flag`가 부호화 트리 블록의 선택스 구조에 포함되지 않을 경우, `alf_luma_prev_filter_idx`는 색인 값에 관계없이 적어도 2개의 적응적 루프내 필터 중에서 하나를 나타낸다. 따라서, 상기 `N`은 1로 결정될 수 있다. 그러므로 만약 `slice_num_alf_aps_ids_luma`가 1일 때, 적응적 루프내 필터의 총 개수가 1개이므로, `alf_luma_prev_filter_idx_minus1`는 시그널링되지 않을 수 있다.

- [799] alf_use_aps_flag 가 제2 값(예: 1)인 경우 및 $\text{slice_num_alf_aps_ids_luma}$ 가 2 이상인 경우 중 적어도 하나의 경우를 만족할 경우에, 상기 $\text{alf_luma_prev_filter_idx}$ 는 부호화/복호화될 수 있다.
- [800] $\text{alf_luma_prev_filter_idx}$ 가 비트스트림 내에 존재하지 않을 경우 $\text{alf_luma_prev_filter_idx}$ 는 0의 값으로 추론될 수 있다.
- [801] 휘도 성분 위치 (x_{Ctb} , y_{Ctb})에서 휘도 성분 부호화 트리 블록에서 사용하는 적응적 루프내 필터 색인 $\text{AlfCtbFiltSetIdxY}[x_{\text{Ctb}} \gg \text{Log2CtbSize}][y_{\text{Ctb}} \gg \text{Log2CtbSize}]$ 은 아래와 같이 유도될 수 있다.
- [802] $\text{alf_ctb_use_first_aps_flag}$ 가 제2 값(예: 1)인 경우, $\text{AlfCtbFiltSetIdxY}[x_{\text{Ctb}} \gg \text{Log2CtbSize}][y_{\text{Ctb}} \gg \text{Log2CtbSize}]$ 는 F의 값으로 결정될 수 있다. 여기서, F는 양의 정수일 수 있고, 16일 수 있다. 또한, F는 부호화기/복호화기에서 사용하는 고정된 필터의 최대 개수를 의미할 수 있다. 즉, F는 $\text{alf_luma_fixed_filter_idx}$ 가 가질 수 있는 최대값에 +1한 값일 수 있다.
- [803] $\text{alf_ctb_use_first_aps_flag}$ 가 제1 값(예: 0)이고 alf_use_aps_flag 가 제1 값(예: 0)인 경우, $\text{AlfCtbFiltSetIdxY}[x_{\text{Ctb}} \gg \text{Log2CtbSize}][y_{\text{Ctb}} \gg \text{Log2CtbSize}]$ 는 $\text{alf_luma_fixed_filter_idx}$ 로 결정될 수 있다.
- [804] 그 밖의 경우 ($\text{alf_ctb_use_first_aps_flag}$ 가 제1 값(예: 0)이고 alf_use_aps_flag 가 제2 값(예: 1)인 경우), $\text{AlfCtbFiltSetIdxY}[x_{\text{Ctb}} \gg \text{Log2CtbSize}][y_{\text{Ctb}} \gg \text{Log2CtbSize}]$ 는 $17 + \text{alf_luma_prev_filter_idx_minus1}$ 로 결정될 수 있다.
- [805] 또는, 휘도 성분 위치 (x_{Ctb} , y_{Ctb})에서 휘도 성분 부호화 트리 블록에서 사용하는 적응적 루프내 필터 색인 $\text{AlfCtbFiltSetIdxY}[x_{\text{Ctb}} \gg \text{Log2CtbSize}][y_{\text{Ctb}} \gg \text{Log2CtbSize}]$ 은 아래와 같이 유도될 수 있다.
- [806] alf_use_aps_flag 가 제1 값(예: 0)인 경우, $\text{AlfCtbFiltSetIdxY}[x_{\text{Ctb}} \gg \text{Log2CtbSize}][y_{\text{Ctb}} \gg \text{Log2CtbSize}]$ 는 $\text{alf_luma_fixed_filter_idx}$ 로 결정될 수 있다.
- [807] 그렇지 않은 경우(alf_use_aps_flag 가 제2 값(예: 1)인 경우), $\text{AlfCtbFiltSetIdxY}[x_{\text{Ctb}} \gg \text{Log2CtbSize}][y_{\text{Ctb}} \gg \text{Log2CtbSize}]$ 는 $F + \text{alf_luma_prev_filter_idx}$ 로 결정될 수 있다. 여기서, F는 양의 정수일 수 있고, 16일 수 있다. 또한, F는 부호화기/복호화기에서 사용하는 고정된 필터의 최대 개수를 의미할 수 있다. 즉, F는 $\text{alf_luma_fixed_filter_idx}$ 가 가질 수 있는 최대값에 +1한 값일 수 있다.
- [808] $\text{alf_luma_fixed_filter_idx}$ 는 휘도 성분 부호화 트리 블록에서 사용하는 고정된 필터 색인을 의미할 수 있다. 즉, 고정 ALF 세트 내의 적어도 하나의 고정된 필터가 현재 부호화 트리 블록에서 사용될 경우, $\text{alf_luma_fixed_filter_idx}$ 를 이용하여 고정 ALF 세트에서 고정된 필터를 선택할 수 있다.
- [809] 상기 $\text{alf_luma_fixed_filter_idx}$ 는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예를 들어 15일 수 있다.
- [810] $\text{alf_luma_fixed_filter_idx}$ 가 비트스트림 내에 존재하지 않을 경우 $\text{alf_luma_fixed_filter_idx}$ 는 0의 값으로 추론될 수 있다.
- [811] 도 21의 부호화 트리 블록의 선택스 구조는 추가적으로 적응 파라미터 세트에

포함된 복수의 색차 적응적 루프내 필터들 중 하나를 나타내는 색차 적응적 루프내 필터 색인 정보(alf_ctb_filter_alt_idx)를 더 포함할 수 있다. 상기 색차 적응적 루프내 필터 색인 정보는 색차 ALF 개수 정보가 1보다 큰 양의 정수일 경우에 시그널링될 수 있다.

[812] 예를 들어, alf_ctb_flag[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]가 Cb 적응적 루프내 필터가 현재 부호화 트리 블록에 사용됨을 나타내고, 적응 파라미터 세트에 2개 이상의 Cb 적응적 루프내 필터가 포함된 경우, 현재 부호화 트리 블록에서 사용되는 Cb 적응적 루프내 필터를 나타내는 Cb 적응적 루프내 필터 색인 정보(alf_ctb_filter_alt_idx)가 부호화 트리 블록의 신택스 구조에 포함될 수 있다. 즉, 현재 부호화 트리 블록에서 사용되는 Cb 적응적 루프내 필터를 나타내는 Cb 적응적 루프내 필터 색인 정보(alf_ctb_filter_alt_idx)가 부호화 트리 블록의 신택스 구조에서 부호화/복호화될 수 있다.

[813] 또한 alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]가 Cr 적응적 루프내 필터가 현재 부호화 트리 블록에 사용됨을 나타내고, 적응 파라미터 세트에 2개 이상의 Cr 적응적 루프내 필터가 포함된 경우, 현재 부호화 트리 블록에서 사용되는 Cr 적응적 루프내 필터를 나타내는 Cr 적응적 루프내 필터 색인 정보(alf_ctb_filter_alt_idx)가 부호화 트리 블록의 신택스 구조에 포함될 수 있다. 즉, 현재 부호화 트리 블록에서 사용되는 Cr 적응적 루프내 필터를 나타내는 Cr 적응적 루프내 필터 색인 정보(alf_ctb_filter_alt_idx)가 부호화 트리 블록의 신택스 구조에서 부호화/복호화될 수 있다.

[814] 상기 Cb 적응적 루프내 필터 색인 정보와 Cr 적응적 루프내 필터 색인 정보는 각각의 색차 성분 별로 시그널링될 수 있으며, 서로 다른 필터 색인 또는 서로 동일한 필터 색인을 지시할 수 있다.

[815]

[816] 이하, 적응적 루프내 필터 프로세스의 일 실시 예가 설명된다. 아래에서, 복원(reconstruction)은 복호(decoding)를 의미할 수 있다.

[817] 적응적 루프내 필터링 수행 전 복원된 휘도 성분 픽처 샘플 배열 recPictureL 이 입력될 수 있다. 그리고 ChromaArrayType가 제1 값(예: 0)이 아닌 경우, 적응적 루프내 필터 수행 전 복원된 색차 성분 픽처 샘플 배열 recPictureCb 및 recPictureCr 이 입력될 수 있다.

[818] 그리고 적응적 루프내 필터 프로세스에 따라, 적응적 루프내 필터링 수행 후 변경된 복원된 휘도 성분 픽처 샘플 배열 alfPictureL이 출력될 수 있다. 그리고 ChromaArrayType가 제1 값(예: 0)이 아닌 경우, 적응적 루프내 필터 수행 전 복원된 색차 성분 픽처 샘플 배열 alfPictureCb 및 alfPictureCr이 출력될 수 있다.

[819] 적응적 루프내 필터링 수행 후 변경된 복원된 휘도 성분 픽처 샘플 배열 alfPictureL은 적응적 루프내 필터링 수행 전 복원된 휘도 성분 픽처 샘플 배열 recPictureL로 초기화될 수 있다.

[820] ChromaArrayType가 제1 값(예: 0)이 아닌 경우, 적응적 루프내 필터 수행 전

복원된 픽처 샘플 배열 alfPictureCb 및 alfPictureCr는 적응적 루프내 필터 수행 전 복원된 픽처 샘플 배열 recPictureCb 및 recPictureCr로 초기화될 수 있다.

- [821] slice_alf_enabled_flag가 제2 값(예: 1)인 경우, 휘도 부호화 트리 블록 위치가 (rx, ry)인 모든 부호화 트리 유닛(coding tree unit)에 대해서 아래의 과정이 수행될 수 있다. 여기서, rx는 0에서 PicWidthInCtbs - 1까지의 값을 가질 수 있고, ry는 0에서 PicHeightInCtbs - 1까지의 값을 가질 수 있다.
- [822] alf_ctb_flag[0][rx][ry]가 제2 값(예: 1)인 경우, 휘도 성분에 대한 부호화 트리 블록 필터링 과정이 수행될 수 있다.
- [823] 휘도 성분에 대한 부호화 트리 블록 필터링 과정의 입력: recPictureL, alfPictureL, 휘도 성분 부호화 트리 블록 위치 (xCtb, yCtb)가 (rx << CtbLog2SizeY, ry << CtbLog2SizeY)로 설정될 수 있다.
- [824] 휘도 성분에 대한 부호화 트리 블록 필터링 과정에 의하여 alfPictureL가 출력될 수 있다.
- [825] ChromaArrayType가 제1 값(예: 0)이 아니고 alf_ctb_flag[1][rx][ry]가 제2 값(예: 1)인 경우 색차 성분 Cb에 대한 부호화 트리 블록 필터링 과정이 수행될 수 있다.
- [826] 색차 성분 Cb에 대한 부호화 트리 블록 필터링 과정의 입력은 recPictureCb, alfPictureCb, 및 색차 성분 Cb 부호화 트리 블록 위치 (xCtbC, yCtbC)가 ((rx << CtbLog2SizeY) / SubWidthC, (ry << CtbLog2SizeY) / SubHeightC)로 설정될 수 있다.
- [827] 색차 성분 Cb에 대한 부호화 트리 블록 필터링 과정에 의하여 alfPictureCb이 출력될 수 있다.
- [828] 일 실시 예에 따르면, 현재 부호화 트리 블록의 Cb 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 픽처 또는 현재 슬라이스에 적용되는 적응적 루프내 필터 세트를 포함하는 한 개 이상의 적응 파라미터 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 Cb 적응적 루프내 필터 세트(Cb ALF 세트)를 포함하는 적응 파라미터 세트를 나타내는 제2 ALF 부호화 트리 블록 식별자가 부호화/복호화/획득될 수 있다. 그리고 상기 제2 ALF 부호화 트리 블록 식별자에 따라, 상기 현재 부호화 트리 블록에 적용되는 Cb 적응적 루프내 필터 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다. 그리고 상기 적응 파라미터 세트에서 Cb 적응적 루프내 필터 세트가 부호화/복호화/획득될 수 있다.
- [829] ChromaArrayType가 제1 값(예: 0)이 아니고 alf_ctb_flag[2][rx][ry]가 제2 값(예: 1)인 경우 색차 성분 Cr에 대한 부호화 트리 블록 필터링 과정이 수행될 수 있다.
- [830] 색차 성분 Cr에 대한 부호화 트리 블록 필터링 과정의 입력: recPictureCr, alfPictureCr, 색차 성분 Cr 부호화 트리 블록 위치 (xCtbC, yCtbC)가 ((rx << CtbLog2SizeY) / SubWidthC, (ry << CtbLog2SizeY) / SubHeightC)로 설정될 수 있다.

- [831] 색차 성분 Cr에 대한 부호화 트리 블록 필터링 과정에 의하여 `alfPictureCr`이 출력될 수 있다.
- [832] 일 실시 예에 따르면, 현재 부호화 트리 블록의 Cr 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 픽처 또는 현재 슬라이스에 적용되는 적응적 루프내 필터 세트를 포함하는 한 개 이상의 적응 파라미터 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 Cr 적응적 루프내 필터 세트(Cr ALF 세트)를 포함하는 적응 파라미터 세트를 나타내는 제3 ALF 부호화 트리 블록 식별자가 부호화/복호화/획득될 수 있다. 그리고 상기 제3 ALF 부호화 트리 블록 식별자에 따라, 상기 현재 부호화 트리 블록에 적용되는 Cr 적응적 루프내 필터 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다. 그리고 상기 적응 파라미터 세트에서 Cr 적응적 루프내 필터 세트가 부호화/복호화/획득될 수 있다.
- [833] 이하, 휘도 성분에 대한 부호화 트리 블록 필터링 과정의 일 실시 예가 설명된다.
- [834] 부호화 트리 블록 필터링을 위하여, 적응적 루프내 필터링 수행 전 복원된 휘도 성분 픽처 샘플 배열 `recPictureL`, 적응적 루프내 필터링 수행 후 변경된 휘도 성분 픽처 샘플 배열 `alfPictureL` 및 현재 픽처의 좌상단 샘플을 기준으로 한 현재 휘도 성분 부호화 트리 블록의 좌상단 샘플의 위치(`xCtb`, `yCtb`)가 입력될 수 있다.
- [835] 그리고 부호화 트리 블록 필터링에 따라 변경된 휘도 성분 픽처 샘플 배열 `alfPictureL`이 출력될 수 있다.
- [836] 부호화 트리 블록 필터링을 위한 필터 색인 유도 과정은 이하와 같이 수행될 수 있다.
- [837] 필터 색인 유도 과정을 위하여, 적응적 루프내 필터링 수행 전 복원된 휘도 성분 픽처 샘플 배열 `recPictureL` 및 현재 픽처의 좌상단 샘플을 기준으로 한 현재 휘도 성분 부호화 트리 블록의 좌상단 샘플의 위치(`xCtb`, `yCtb`)가 입력될 수 있다.
- [838] 필터 색인 유도 과정에 따라, 분류 필터 색인 배열 `filtIdx`과 전치 색인 배열 `transposeIdx`가 출력될 수 있다.
- [839] 이때, `x` 및 `y`는 0부터 `CtbSizeY - 1`까지의 값을 가질 수 있다.
- [840] `alfPictureL[ x ][ y ]`을 유도하기 위해, `recPictureL[ x ][ y ]` 내 현재 휘도 성분 부호화 트리 블록 내 각 복원된 휘도 성분 샘플은 아래와 같이 필터링될 수 있다. 이때, `x` 및 `y`는 0부터 `CtbSizeY - 1`까지의 값을 가질 수 있다.
- [841] `filtIdx[ x ][ y ]`에 지시되는 필터에 대한 휘도 성분 필터 계수 배열 `f[ j ]`과 휘도 성분 절삭 값 배열 `c[ j ]`은 아래와 같이 유도될 수 있다. 여기서, `j`는 0에서 `N`까지의 값을 가질 수 있고, 예를 들어 `N`은 양의 정수이고, 예를 들어 11일 수 있다.
- [842] `AlfCtbFiltSetIdxY[ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]`가 `F`보다 작은 값을 가질 경우, 아래와 같이 수행될 수 있다. 여기서, `F`는 양의 정수일 수 있고, 16일 수 있다. 또한, `F`는 부호화기/복호화기에서 사용하는 고정된 필터의 최대 개수를 의미할 수 있다. 즉, `F`는 `alf_luma_fixed_filter_idx`가 가질 수 있는 최대값에

+1한 값일 수 있다.

[843] $i = \text{AlfCtbFiltSetIdxY}[x\text{Ctb} \gg \text{Log2CtbSize}][y\text{Ctb} \gg \text{Log2CtbSize}]$

[844] $f[j] = \text{AlfFixFiltCoeff}[\text{AlfClassToFiltMap}[i][\text{filtIdx}[x][y]][j]]$

[845] $c[j] = 2\text{BitdepthY}$ 또는 $c[j] = \text{AlfClipL}[i][\text{filtIdx}[x][y]][j]$

[846] 여기서, $c[j]$ 에 $\text{AlfClipL}[i][\text{filtIdx}[x][y]][j]$ 를 할당하는 것은 고정된 필터에 비선형 적응적 루프내 필터링 방법인 절삭 방법을 사용하는 것을 의미할 수 있다.

[847] $\text{AlfCtbFiltSetIdxY}[x\text{Ctb} \gg \text{Log2CtbSize}][y\text{Ctb} \gg \text{Log2CtbSize}]$ 가 F보다 크거나 같을 경우, 아래와 같이 수행될 수 있다.

[848] $i = \text{slice_alf_aps_id_luma}[\text{AlfCtbFiltSetIdxY}[x\text{Ctb} \gg \text{Log2CtbSize}][y\text{Ctb} \gg \text{Log2CtbSize}] - F]$

[849] $f[j] = \text{AlfCoeffL}[i][\text{filtIdx}[x][y]][j]$

[850] $c[j] = \text{AlfClipL}[i][\text{filtIdx}[x][y]][j]$

[851] 여기서, $f[j]$ 는 휘도 ALF를 의미할 수 있고, $c[j]$ 는 휘도 절삭 값을 의미할 수 있다.

[852] 휘도 성분 필터 계수 및 절삭 값에 대한 색인 idx 은 $\text{transposeIdx}[x][y]$ 에 따라 아래와 같이 유도될 수 있다.

[853] $\text{transposeIndex}[x][y]$ 가 제2 값(예: 1)인 경우, 아래와 같이 수행될 수 있다.

[854] $\text{idx}[] = \{ 9, 4, 10, 8, 1, 5, 11, 7, 3, 0, 2, 6 \}$

[855] $\text{transposeIndex}[x][y]$ 가 제3 값(예: 2)인 경우, 아래와 같이 수행될 수 있다.

[856] $\text{idx}[] = \{ 0, 3, 2, 1, 8, 7, 6, 5, 4, 9, 10, 11 \}$

[857] $\text{transposeIndex}[x][y]$ 가 제4 값(예: 3)인 경우, 아래와 같이 수행될 수 있다.

[858] $\text{idx}[] = \{ 9, 8, 10, 4, 3, 7, 11, 5, 1, 0, 2, 6 \}$

[859] 그렇지 않은 경우($\text{transposeIndex}[x][y]$ 가 제1 값(예: 0)인 경우), 아래와 같이 수행될 수 있다.

[860] $\text{idx}[] = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 \}$

[861] 각 대응하는 휘도 성분 샘플 (x, y) 에 대한 recPicture 내 위치 $(hx + i, vy + j)$ 는 아래와 같이 유도될 수 있다. 이때, i 및 j 는 $-N$ 에서 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 3일 수 있다. 또한, $(N*2)+1$ 은 휘도 성분 필터의 가로 크기 또는 세로 크기를 의미할 수 있다.

[862] $hx + i = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x\text{Ctb} + x + i)$

[863] $vy + j = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y\text{Ctb} + y + j)$

[864] 가상 경계를 적용할지 여부에 대한 변수 $\text{applyVirtualBoundary}$ 는 아래와 같이 유도될 수 있다.

[865] 아래의 조건 중 적어도 하나가 참(true)일 경우, $\text{applyVirtualBoundary}$ 는 0으로 설정될 수 있다.

[866] 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 경계일 경우, $\text{applyVirtualBoundary}$ 는 0으로 설정될 수 있다.

- [867] 현재 부호화 트리 블록의 하단 경계가 브릭의 하단 경계이며, loop_filter_across_bricks_enabled_flag가 제1 값(예: 0)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.
- [868] 여기서, loop_filter_across_bricks_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 브릭 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 브릭 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [869] 현재 부호화 트리 블록의 하단 경계가 슬라이스의 하단 경계이며, loop_filter_across_slices_enabled_flag가 제1 값(예: 0)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.
- [870] 여기서, loop_filter_across_slices_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 슬라이스 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 슬라이스 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [871] 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계이며, loop_filter_across_tiles_enabled_flag가 제1 값(예: 0)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.
- [872] 또는, 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계이며, loop_filter_across_tiles_enabled_flag가 제1 값(예: 0)인 경우, 적응적 루프내 필터가 적용되는 영역(타일)의 하단 경계의 y 좌표의 위치가 현재 부호화 트리 블록의 하단 경계의 y 좌표의 위치에 따라 결정될 수 있다.
- [873] 여기서, loop_filter_across_tiles_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 타일 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 타일 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [874] 일 실시 예에 따르면, 현재 부호화 트리 블록의 하단 경계가 슬라이스의 하단 경계인 경우에도, applyVirtualBoundary는 1로 설정될 수 있다.
- [875] 또한 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계인 경우에도, applyVirtualBoundary는 1로 설정될 수 있다. 또한 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계인 경우에도, applyVirtualBoundary는 1로 설정될 수 있다.
- [876] 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계이며, loop_filter_across_subpictures_enabled_flag가 제1 값(예: 0)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.
- [877] 또는, 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계이며, loop_filter_across_subpictures_enabled_flag가 제1 값(예: 0)인 경우, 적응적 루프내 필터가 적용되는 영역(서브 픽처)의 하단 경계의 y 좌표의 위치가 현재 부호화 트리 블록의 하단 경계의 y 좌표의 위치에 따라 결정될 수 있다.

[878] 여기서, loop_filter_across_subpictures_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 서브 픽처 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 서브 픽처 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.

[879] 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 가상 경계이며, pps_loop_filter_across_virtual_boundaries_disabled_flag가 제2 값(예: 1)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.

[880] 그렇지 않은 경우(상기 조건이 참(true)이 아닌 경우), applyVirtualBoundary는 1로 설정될 수 있다.

[881] 복원된 샘플 오프셋 r1, r2, 및 r3은 세로 방향 휘도 성분 샘플 위치 y와 applyVirtualBoundary에 따라 표 1과 같이 결정될 수 있다.

[882] [표 1]

조건	r1	r2	r3
(y == CtbSizeY - 5 y == CtbSizeY - 4) && (applyVirtualBoundary == 1)	0	0	0
(y == CtbSizeY - 6 y == CtbSizeY - 3) && (applyVirtualBoundary == 1)	1	1	1
(y == CtbSizeY - 7 y == CtbSizeY - 2) && (applyVirtualBoundary == 1)	1	2	2
그렇지 않은 경우(위의 조건에 해당하지 않은 경우)	1	2	3

[884] 현재 샘플 위치에 대한 샘플 값 curr는 아래와 같이 유도될 수 있다.

[885] curr = recPictureL[h_x, v_y]

[886] 현재 샘플에 대해 필터링이 수행된 값 sum은 아래와 같이 유도될 수 있다.

[887] [수학식 22]

[888] sum = f[idx[0]] * (Clip3(-c[idx[0]], c[idx[0]], recPictureL[h_x, v_{y+r3}] - curr) +
 Clip3(-c[idx[0]], c[idx[0]], recPictureL[h_x, v_{y-r3}] - curr)) +
 f[idx[1]] * (Clip3(-c[idx[1]], c[idx[1]], recPictureL[h_{x+1}, v_{y+r2}] - curr) +
 Clip3(-c[idx[1]], c[idx[1]], recPictureL[h_{x-1}, v_{y-r2}] - curr)) +
 f[idx[2]] * (Clip3(-c[idx[2]], c[idx[2]], recPictureL[h_x, v_{y+r2}] - curr) +
 Clip3(-c[idx[2]], c[idx[2]], recPictureL[h_x, v_{y-r2}] - curr)) +
 f[idx[3]] * (Clip3(-c[idx[3]], c[idx[3]], recPictureL[h_{x-1}, v_{y+r2}] - curr) +
 Clip3(-c[idx[3]], c[idx[3]], recPictureL[h_{x+1}, v_{y-r2}] - curr)) +
 f[idx[4]] * (Clip3(-c[idx[4]], c[idx[4]], recPictureL[h_{x+2}, v_{y+r1}] - curr) +
 Clip3(-c[idx[4]], c[idx[4]], recPictureL[h_{x-2}, v_{y-r1}] - curr)) +
 f[idx[5]] * (Clip3(-c[idx[5]], c[idx[5]], recPictureL[h_{x+1}, v_{y+r1}] - curr) +
 Clip3(-c[idx[5]], c[idx[5]], recPictureL[h_{x-1}, v_{y-r1}] - curr)) +
 f[idx[6]] * (Clip3(-c[idx[6]], c[idx[6]], recPictureL[h_x, v_{y+r1}] - curr) +
 Clip3(-c[idx[6]], c[idx[6]], recPictureL[h_x, v_{y-r1}] - curr)) +
 f[idx[7]] * (Clip3(-c[idx[7]], c[idx[7]], recPictureL[h_{x-1}, v_{y+r1}] - curr) +
 Clip3(-c[idx[7]], c[idx[7]], recPictureL[h_{x+1}, v_{y-r1}] - curr)) +
 f[idx[8]] * (Clip3(-c[idx[8]], c[idx[8]], recPictureL[h_{x-2}, v_{y+r1}] - curr) +
 Clip3(-c[idx[8]], c[idx[8]], recPictureL[h_{x+2}, v_{y-r1}] - curr)) +
 f[idx[9]] * (Clip3(-c[idx[9]], c[idx[9]], recPictureL[h_{x+3}, v_y] - curr) +
 Clip3(-c[idx[9]], c[idx[9]], recPictureL[h_{x-3}, v_y] - curr)) +
 f[idx[10]] * (Clip3(-c[idx[10]], c[idx[10]], recPictureL[h_{x+2}, v_y] - curr) +
 Clip3(-c[idx[10]], c[idx[10]], recPictureL[h_{x-2}, v_y] - curr)) +
 f[idx[11]] * (Clip3(-c[idx[11]], c[idx[11]], recPictureL[h_{x+1}, v_y] - curr) +
 Clip3(-c[idx[11]], c[idx[11]], recPictureL[h_{x-1}, v_y] - curr))
 sum = curr + ((sum + 64) >> 7)

[889] alfPictureL[xCtb + x][yCtb + y]는 아래와 같이 유도될 수 있다.

- [890] $\text{pcm_loop_filter_disabled_flag}$ 가 제2 값(예: 1)이며 $\text{pcm_flag}[x_{\text{Ctb}} + x][y_{\text{Ctb}} + y]$ 가 제2 값(예: 1)인 경우, 아래와 같이 수행될 수 있다.
- [891] $\text{alfPictureL}[x_{\text{Ctb}} + x][y_{\text{Ctb}} + y] = \text{recPictureL}[hx, vy]$
- [892] 그렇지 않은 경우($\text{pcm_loop_filter_disabled_flag}$ 가 제1 값(예: 0)이며 $\text{pcm_flag}[x_{\text{Ctb}} + x][y_{\text{Ctb}} + y]$ 가 제1 값(예: 0)인 경우), 아래와 같이 수행될 수 있다.
- [893] $\text{alfPictureL}[x_{\text{Ctb}} + x][y_{\text{Ctb}} + y] = \text{Clip3}(0, (1 \ll \text{BitDepthY}) - 1, \text{sum})$
- [894]
- [895] 이하, 색차 성분에 대한 부호화 트리 블록 필터링 과정의 일 실시 예가 설명된다.
- [896] 복원 픽처인 recPicture 과 적응적 루프내 필터링된 픽처인 alfPicture 이 입력될 수 있다. 그리고 현재 픽처의 좌상단 샘플을 기준으로 한 현재 색차 성분 부호화 트리 블록의 좌상단 샘플의 위치 ($x_{\text{CtbC}}, y_{\text{CtbC}}$)가 입력될 수 있다. 그리고 적응적 루프내 필터링된 픽처인 alfPicture 가 출력될 수 있다.
- [897] 현재 색차 성분 부호화 트리 블록의 가로 크기 ctbWidthC 와 세로 크기 ctbHeightC 는 아래와 같이 유도될 수 있다.
- [898] $\text{ctbWidthC} = \text{CtbSizeY} / \text{SubWidthC}$
- [899] $\text{ctbHeightC} = \text{CtbSizeY} / \text{SubHeightC}$
- [900] $\text{alfPicture}[x][y]$ 을 유도하기 위해, $\text{recPicture}[x][y]$ 내 현재 색차 성분 부호화 트리 블록 내 각 복원된 색차 성분 샘플은 아래와 같이 필터링될 수 있다. 이때, x 는 0부터 $\text{ctbWidthC} - 1$ 까지의 값을 가질 수 있다. 또한, y 는 0부터 $\text{ctbHeightC} - 1$ 까지의 값을 가질 수 있다.
- [901] 각 대응하는 색차 성분 샘플 (x, y)에 대한 recPicture 내 위치 ($hx + i, vy + j$)는 아래와 같이 유도될 수 있다. 이때, i 및 j 는 $-N$ 에서 N 까지의 값을 가질 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 2일 수 있다. 또한, $(N*2)+1$ 은 색차 성분 필터의 가로 크기 또는 세로 크기를 의미할 수 있다.
- [902] $hx + i = \text{Clip3}(0, \text{pic_width_in_luma_samples} / \text{SubWidthC} - 1, x_{\text{CtbC}} + x + i)$
- [903] $vy + j = \text{Clip3}(0, \text{pic_height_in_luma_samples} / \text{SubHeightC} - 1, y_{\text{CtbC}} + y + j)$
- [904]
- [905] 가상 경계를 적용할지 여부에 대한 변수 $\text{applyVirtualBoundary}$ 는 아래와 같이 유도될 수 있다.
- [906] 아래의 조건 중 적어도 하나가 참(true)일 경우, $\text{applyVirtualBoundary}$ 는 0으로 설정될 수 있다.
- [907] 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 경계일 경우, $\text{applyVirtualBoundary}$ 는 0으로 설정될 수 있다.
- [908] 현재 부호화 트리 블록의 하단 경계가 브릭의 하단 경계이며, $\text{loop_filter_across_bricks_enabled_flag}$ 가 제1 값(예: 0)인 경우, $\text{applyVirtualBoundary}$ 는 0으로 설정될 수 있다.
- [909] 여기서, $\text{loop_filter_across_bricks_enabled_flag}$ 는 루프내 필터링 방법들 중

적어도 하나가 브릭 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 브릭 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.

- [910] 현재 부호화 트리 블록의 하단 경계가 슬라이스의 하단 경계이며, loop_filter_across_slices_enabled_flag가 제1 값(예: 0)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.
- [911] 여기서, loop_filter_across_slices_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 슬라이스 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 슬라이스 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [912] 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계이며, loop_filter_across_tiles_enabled_flag가 제1 값(예: 0)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.
- [913] 또는, 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계이며, loop_filter_across_tiles_enabled_flag가 제1 값(예: 0)인 경우, 적응적 루프내 필터가 적용되는 영역(타일)의 하단 경계의 y 좌표의 위치가 현재 부호화 트리 블록의 하단 경계의 y 좌표의 위치에 따라 결정될 수 있다.
- [914] 여기서, loop_filter_across_tiles_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 타일 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 타일 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [915] 일 실시 예에 따르면, 현재 부호화 트리 블록의 하단 경계가 슬라이스의 하단 경계인 경우에도, applyVirtualBoundary는 1로 설정될 수 있다.
- [916] 또한 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계인 경우에도, applyVirtualBoundary는 1로 설정될 수 있다. 또한 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계인 경우에도, applyVirtualBoundary는 1로 설정될 수 있다.
- [917] 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계이며, loop_filter_across_subpictures_enabled_flag가 제1 값(예: 0)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.
- [918] 또는, 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계이며, loop_filter_across_subpictures_enabled_flag가 제1 값(예: 0)인 경우, 적응적 루프내 필터가 적용되는 영역(서브 픽처)의 하단 경계의 y 좌표의 위치가 현재 부호화 트리 블록의 하단 경계의 y 좌표의 위치에 따라 결정될 수 있다.
- [919] 여기서, loop_filter_across_subpictures_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 서브 픽처 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 서브 픽처 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.

[920] 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 가상 경계이며, pps_loop_filter_across_virtual_boundaries_disabled_flag가 제2 값(예: 1)인 경우, applyVirtualBoundary는 0으로 설정될 수 있다.

[921] 그렇지 않은 경우(상기 조건이 참(true)이 아닌 경우), applyVirtualBoundary는 1로 설정될 수 있다.

[922] 복원된 샘플 오프셋 r1 및 r2은 세로 방향 색차 성분 샘플 위치 y와 applyVirtualBoundary에 따라 표 4와 같이 결정될 수 있다.

[923] [표 2]

조건	r1	r2
(y == ctbHeightC - 2 y == ctbHeightC - 3) && (applyVirtualBoundary == 1)	0	0
(y == ctbHeightC - 1 y == ctbHeightC - 4) && (applyVirtualBoundary == 1)	1	1
그렇지 않은 경우(위의 조건에 해당하지 않은 경우)	1	2

[925] 현재 샘플 위치에 대한 샘플 값 curr는 아래와 같이 유도될 수 있다.

[926] curr = recPicture[hx, vy]

[927] 색차 성분 필터 계수 배열 f[j]과 색차 성분 절삭 값 배열 c[j]은 아래와 같이 유도될 수 있다. 여기서, j는 0에서 N까지의 값을 가질 수 있고, 예를 들어 N은 양의 정수이고, 예를 들어 5일 수 있다.

[928] f[j] = AlfCoeffC[slice_alf_aps_id_chroma][j]

[929] c[j] = AlfClipC[slice_alf_aps_id_chroma][j]

[930] 여기서, f[j]는 색차 ALF를 의미할 수 있고, c[j]는 색차 절삭 값을 의미할 수 있다.

[931]

[932] 현재 샘플에 대해 필터링이 수행된 값 sum은 수학적식 23과 같이 유도될 수 있다.

[933] [수학적식 23]

[934]
$$\begin{aligned} \text{sum} = & f[0] * (\text{Clip3}(-c[0], c[0], \text{recPicture}[h_x, v_{y+r2}] - \text{curr}) + \\ & \text{Clip3}(-c[0], c[0], \text{recPicture}[h_x, v_{y-r2}] - \text{curr})) + \\ & f[1] * (\text{Clip3}(-c[1], c[1], \text{recPicture}[h_{x+1}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[1], c[1], \text{recPicture}[h_{x-1}, v_{y-r1}] - \text{curr})) + \\ & f[2] * (\text{Clip3}(-c[2], c[2], \text{recPicture}[h_x, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[2], c[2], \text{recPicture}[h_x, v_{y-r1}] - \text{curr})) + \\ & f[3] * (\text{Clip3}(-c[3], c[3], \text{recPicture}[h_{x-1}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[3], c[3], \text{recPicture}[h_{x+1}, v_{y-r1}] - \text{curr})) + \\ & f[4] * (\text{Clip3}(-c[4], c[4], \text{recPicture}[h_{x+2}, v_y] - \text{curr}) + \\ & \text{Clip3}(-c[4], c[4], \text{recPicture}[h_{x-2}, v_y] - \text{curr})) + \\ & f[5] * (\text{Clip3}(-c[5], c[5], \text{recPicture}[h_{x+1}, v_y] - \text{curr}) + \\ & \text{Clip3}(-c[5], c[5], \text{recPicture}[h_{x-1}, v_y] - \text{curr})) \\ \text{sum} = & \text{curr} + (\text{sum} + 64) \gg 7 \end{aligned}$$

[935] alfPicture[xCtbC + x][yCtbC + y]는 아래와 같이 유도될 수 있다.

[936] pcm_loop_filter_disabled_flag가 제2 값(예: 1)이며 pcm_flag[(xCtbC + x) * SubWidthC][(yCtbC + y) * SubHeightC]가 제2 값(예: 1)인 경우, 아래와 같이

수행될 수 있다.

[937] $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] = \text{recPictureL}[hx, vy]$

[938] 그렇지 않은 경우($\text{pcm_loop_filter_disabled_flag}$ 가 제1 값(예: 0)이며 $\text{pcm_flag}[(x\text{CtbC} + x) * \text{SubWidthC}][(y\text{CtbC} + y) * \text{SubHeightC}]$ 가 제1 값(예: 0)인 경우), 아래와 같이 수행될 수 있다.

[939] $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, \text{sum})$

[940] 도 24, 및 도 25의 예와 같이, $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화하는데 $u(2)$ 엔트로피 부호화/복호화 방식을 사용할 수 있다. 이때, $u(2)$ 의 최대값은 N 일 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 3일 수 있다. 또한, 상기 $u(2)$ 는 $f(2)$ 와 동일한 의미를 가질 수 있다. 상기 $u(n)$ 와 $f(n)$ 방식 중 적어도 하나는 구문 요소를 n 비트 고정 길이 부호(fixed-length code)로 부호화/복호화하는 것을 의미할 수 있다. 여기서, n 은 양의 정수일 수 있다.

[941] 도 26, 및 도 27의 예와 같이, $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화하는데 $\text{tu}(v)$ 엔트로피 부호화/복호화 방식을 사용할 수 있다. 이때, $\text{tu}(v)$ 의 최대값(maxVal)은 N 일 수 있다. 여기서, N 은 양의 정수일 수 있고, 예를 들어 3일 수 있다. 또한, 상기 $\text{tu}(v)$ 의 최대값이 3일 경우에 $\text{tu}(3)$ 과 동일한 의미를 가질 수 있다.

[942] 상기 도 24 내지 도 27의 예의 방법들을 사용할 경우, 구문 요소(syntax element)들인 $\text{alf_luma_clip_min_eg_order_minus1}$, $\text{alf_luma_clip_eg_order_increase_flag}[i]$, $\text{alf_chroma_clip_min_eg_order_minus1}$, $\text{alf_chroma_clip_eg_order_increase_flag}[i]$ 와 구문 요소들인 $\text{alf_luma_clip_min_eg_order_minus1}$, $\text{alf_luma_clip_eg_order_increase_flag}[i]$, $\text{alf_chroma_clip_min_eg_order_minus1}$, $\text{alf_chroma_clip_eg_order_increase_flag}[i]$ 에 관한 구문 요소 의미(semantics)를 사용하지 않고, $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화를 할 수 있으므로, $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화하는데 필요한 부호화기/복호화기의 구현 복잡도가 감소할 수 있다.

[943] 또한, 지수-골롬 부호의 차수 $k\text{ClipY}[i]$, 지수-골롬 부호의 차수 $k\text{ClipC}[i]$, 지수-골롬 부호(이진화) $\text{uek}(v)$ 의 차수 k 를 유도하는데 필요한 부호화/복호화 과정도 수행되지 않을 수 있으므로, $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화하는데 필요한 부호화기/복호화기의 구현 복잡도가 감소할 수 있다.

[944] 또한, 상기 도 24 내지 도 27의 예의 방법들을 사용할 경우, 지수-골롬 부호를 이용하여 $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화할 때보다 더 적은 비트로 $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화할 수 있기 때문에, 영상

부호화기/복호화기의 부호화 효율이 향상될 수 있다.

- [945] 상기 $tu(v)$ 에 대한 엔트로피 복호화는 아래와 같이 수행되어, 구문 요소의 값을 의미하는 `codeNum`는 아래와 같이 계산될 수 있다. 여기서, 엔트로피 복호화는 파싱(parsing)을 의미할 수 있다. $tu(v)$ 에 대한 값의 범위는 0부터 최대값(maxVal)로 결정될 수 있다. 이때, 최대값은 양의 정수일 수 있고, 예를 들어 1보다 크거나 같을 수 있다.

```
[946]   codeNum = 0
[947]   keepGoing = 1
[948]   for(i = 0; i < maxVal && keepGoing; i++){
[949]       keepGoing = read_bits( 1 )
[950]       if( keepGoing )
[951]           codeNum ++ }
```

- [952] 도 28 내지 도 37은 본 발명의 일 실시예에 따른 적응적 루프내 필터링 방법, 장치 및 비트스트림을 저장한 기록 매체를 부호화기/복호화기에서 구현하기 위해 필요한 구문 요소(syntax element) 정보, 구문 요소 정보의 의미(semantics), 부호화/복호화 과정의 예를 나타낸다.

- [953] 도 38은 일 실시예에 따른 적응적 루프내 필터를 이용한 비디오 복호화 방법의 도시한다.

- [954] 단계 3802에서 복수의 적응적 루프내 필터(ALF, Adaptive Loop Filter)들로 구성된 ALF 세트를 포함하는 적응 파라미터 세트들 중 적어도 하나가 부호화/복호화/획득될 수 있다.

- [955] 일 실시예에 따르면, 적응 파라미터 세트는 색차 ALF 개수 정보를 포함하고, 상기 ALF 세트는 상기 색차 ALF 개수 정보가 나타내는 개수의 색차 ALF를 포함하는 것을 특징으로 하는 비디오 복호화 방법이 제공된다.

- [956] 일 실시예에 따르면, 상기 적응 파라미터 세트는, 휘도 성분에 대해 비선형 적응적 루프내 필터링이 수행되는지 여부를 나타내는 휘도 절삭 플래그, 및 색차 성분에 대해 비선형 적응적 루프내 필터링이 수행되는지 여부를 나타내는 색차 절삭 플래그를 포함할 수 있다.

- [957] 일 실시예에 따르면, 상기 적응 파라미터 세트는, 상기 휘도 절삭 플래그가 휘도 성분에 대해 비선형 적응적 루프내 필터링이 수행됨을 나타낼 때, 비선형 적응적 루프내 필터링을 위한 절삭 값을 나타내는 휘도 절삭 색인을 포함하고, 상기 색차 절삭 플래그가 색차 성분에 대해 비선형 적응적 루프내 필터링이 수행됨을 나타낼 때, 비선형 적응적 루프내 필터링을 위한 절삭 값을 나타내는 색차 절삭 색인을 포함할 수 있다.

- [958] 일 실시예에 따르면, 상기 휘도 절삭 색인 및 상기 색차 절삭 색인은 2비트의 고정 길이로 부호화될 수 있다.

- [959] 일 실시예에 따르면, 휘도 절삭 색인이 나타내는 값과 현재 시퀀스의 비트맵스에 따라, 휘도 성분에 대한 비선형 적응적 루프내 필터링을 위한 휘도 절삭

값이 결정되고, 상기 색차 절삭 색인이 나타내는 값과 상기 현재 시퀀스의 비트맵스에 따라, 색차 성분에 대한 비선형 적응적 루프내 필터링을 위한 색차 절삭 값이 결정되고, 상기 휘도 절삭 색인이 나타내는 값과 상기 색차 절삭 색인이 나타내는 값이 동일할 때, 상기 휘도 절삭 값과 상기 색차 절삭 값이 동일할 수 있다.

- [960] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 상기 적응 파라미터 세트에 부여된 식별번호를 나타내는 적응 파라미터 세트 식별자 및 상기 적응 파라미터 세트에 포함된 부호화 정보의 타입을 나타내는 적응 파라미터 세트 타입 정보를 포함할 수 있다.
- [961] 일 실시 예에 따르면, 상기 적응 파라미터 세트 타입 정보가 ALF 타입을 나타낼 때, ALF 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다.
- [962] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 상기 적응 파라미터 세트가 휘도 성분에 대한 ALF를 포함하는지 여부를 나타내는 휘도 ALF 시그널링 플래그 및 상기 적응 파라미터 세트가 색차 성분에 대한 ALF를 포함하는지 여부를 나타내는 색차 ALF 시그널링 플래그를 포함할 수 있다.
- [963] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 휘도 시그널링 ALF의 개수를 나타내는 휘도 시그널링 ALF 개수 정보를 포함하고, 상기 휘도 시그널링 ALF 개수 정보가 상기 휘도 시그널링 ALF가 1개보다 많음을 나타낼 때, 휘도 ALF 세트의 소정 개수의 휘도 ALF가 참조하는 휘도 시그널링 ALF의 색인을 나타내는 휘도 ALF 델타 색인을 포함할 수 있다.
- [964] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 하나 이상의 휘도 시그널링 ALF를 포함하고, 상기 휘도 ALF 델타 색인에 따라, 상기 하나 이상의 휘도 시그널링 ALF로부터 상기 소정 개수의 휘도 ALF가 결정될 수 있다.
- [965] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 색차 ALF의 개수를 나타내는 색차 ALF 개수 정보를 포함하고, 색차 ALF 개수 정보가 나타내는 개수만큼, 색차 ALF를 포함할 수 있다.
- [966] 단계 3804에서 적응 파라미터 세트들 중에서, 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는, 상기 현재 픽처 또는 슬라이스에 적용되는 적응 파라미터 세트들 중 적어도 하나가 결정될 수 있다.
- [967] 일 실시 예에 따르면, 상기 현재 픽처 또는 슬라이스의 휘도 ALF 세트 개수 정보가 부호화/복호화/획득될 수 있다. 그리고 상기 휘도 ALF 세트 개수 정보가 나타내는 개수의 휘도 ALF 세트 식별자가 부호화/복호화/획득될 수 있다.
- [968] 단계 3806에서 현재 픽처 또는 슬라이스에 적용되는 적응 파라미터 세트로부터, 상기 현재 픽처 또는 슬라이스에 포함된 현재 부호화 트리 블록(CTB, Coding Tree Block)에 적용되는 ALF 세트를 포함하는, 상기 현재 CTB에 적용되는 적응 파라미터 세트가 결정될 수 있다.
- [969] 일 실시 예에 따르면, 상기 현재 픽처 또는 슬라이스의 색차 ALF 적용 정보가 부호화/복호화/획득될 수 있다. 그리고 상기 색차 ALF 적용 정보가 Cb 성분 및 Cr

성분 중 적어도 하나에 ALF가 적용됨을 나타낼 때, 색차 ALF 세트 식별자가 부호화/복호화/획득될 수 있다.

[970] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 휘도 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제1 ALF 부호화 트리 블록 플래그가 부호화/복호화/획득되고, 상기 제1 ALF 부호화 트리 블록 플래그에 따라, 상기 현재 부호화 트리 블록의 휘도 샘플에 적응적 루프내 필터링이 적용되는지 여부를 결정될 수 있다. 그리고 상기 현재 부호화 트리 블록의 Cb 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제2 ALF 부호화 트리 블록 플래그가 부호화/복호화/획득되고, 상기 제2 ALF 부호화 트리 블록 플래그에 따라, 상기 현재 부호화 트리 블록의 Cb 샘플에 적응적 루프내 필터링이 적용되는지 여부를 결정될 수 있다. 그리고 상기 현재 부호화 트리 블록의 Cr 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제3 ALF 부호화 트리 블록 플래그가 부호화/복호화/획득되고, 상기 제3 ALF 부호화 트리 블록 플래그에 따라, 상기 현재 부호화 트리 블록의 Cr 샘플에 적응적 루프내 필터링이 적용되는지 여부가 결정될 수 있다.

[971] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 휘도 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 부호화 트리 블록에 적응 파라미터 세트의 ALF 세트가 적용되는 여부를 나타내는 적응 파라미터 세트 적용 플래그가 부호화/복호화/획득될 수 있다. 그리고 상기 적응 파라미터 세트 적용 플래그가 상기 현재 부호화 트리 블록에 적응 파라미터 세트의 ALF 세트가 적용됨을 나타낼 때, 상기 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는 한 개 이상의 적응 파라미터 세트로부터, 상기 현재 부호화 트리 블록에 적용되는 휘도 ALF 세트가 결정될 수 있다. 반대로, 상기 적응 파라미터 세트 적용 플래그가 상기 현재 부호화 트리 블록에 적응 파라미터 세트의 ALF 세트가 적용되지 않음을 나타낼 때, 휘도 샘플에 대한 고정 ALF 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 고정된 필터가 결정될 수 있다.

[972] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 Cb 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는 한 개 이상의 적응 파라미터 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 Cb ALF 세트를 포함하는 적응 파라미터 세트를 나타내는 제2 ALF 부호화 트리 블록 식별자가 부호화/복호화/획득될 수 있다. 그리고 상기 제2 ALF 부호화 트리 블록 식별자에 따라, 상기 현재 부호화 트리 블록에 적용되는 Cb ALF 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다.

[973] 또한 상기 현재 부호화 트리 블록의 Cr 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는 한 개 이상의 적응 파라미터 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 Cr ALF 세트를 포함하는 적응 파라미터 세트를 나타내는 제3 ALF 부호화 트리 블록 식별자가 부호화/복호화/획득될 수 있다. 그리고 상기 제3 ALF 부호화 트리

블록 식별자에 따라, 상기 현재 부호화 트리 블록에 적용되는 Cr ALF 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다.

- [974] 단계 3808에서, 상기 결정된 현재 CTB에 적용되는 적응 파라미터 세트의 ALF 세트에 기초하여, 상기 현재 부호화 트리 블록이 필터링될 수 있다.
- [975] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 기본 필터링 단위 블록에 대하여, 블록 분류 색인이 할당될 수 있다. 상기 블록 분류 색인은 방향성 정보와 활동성 정보를 이용하여 결정될 수 있다.
- [976] 일 실시 예에 따르면, 상기 방향성 정보 및 활동성 정보 중 적어도 하나는 세로, 가로, 제1 대각, 제2 대각 방향 중 적어도 하나에 대한 기울기(gradient) 값에 기반하여 결정될 수 있다.
- [977] 도 39는 일 실시 예에 따른 적응적 루프내 필터를 이용한 비디오 부호화 방법을 도시한다.
- [978] 단계 3902에서 복수의 ALF들로 구성된 ALF 세트를 포함하는 적응 파라미터 세트들 중 적어도 하나가 결정될 수 있다.
- [979] 일 실시 예에 따르면, 적응 파라미터 세트는 색차 ALF 개수 정보를 포함하고, 상기 ALF 세트는 상기 색차 ALF 개수 정보가 나타내는 개수의 색차 ALF를 포함할 수 있다.
- [980] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 휘도 성분에 대해 비선형 적응적 루프내 필터링이 수행되는지 여부를 나타내는 휘도 절삭 플래그, 및 색차 성분에 대해 비선형 적응적 루프내 필터링이 수행되는지 여부를 나타내는 색차 절삭 플래그를 포함할 수 있다.
- [981] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 상기 휘도 절삭 플래그가 휘도 성분에 대해 비선형 적응적 루프내 필터링이 수행됨을 나타낼 때, 비선형 적응적 루프내 필터링을 위한 절삭 값을 나타내는 휘도 절삭 색인을 포함하고, 상기 색차 절삭 플래그가 색차 성분에 대해 비선형 적응적 루프내 필터링이 수행됨을 나타낼 때, 비선형 적응적 루프내 필터링을 위한 절삭 값을 나타내는 색차 절삭 색인을 포함할 수 있다.
- [982] 일 실시 예에 따르면, 상기 휘도 절삭 색인 및 상기 색차 절삭 색인은 2비트의 고정 길이로 부호화될 수 있다.
- [983] 일 실시 예에 따르면, 휘도 절삭 색인이 나타내는 값과 현재 시퀀스의 비트 템스에 따라, 휘도 성분에 대한 비선형 적응적 루프내 필터링을 위한 휘도 절삭 값이 결정되고, 상기 색차 절삭 색인이 나타내는 값과 상기 현재 시퀀스의 비트 템스에 따라, 색차 성분에 대한 비선형 적응적 루프내 필터링을 위한 색차 절삭 값이 결정되고, 상기 휘도 절삭 색인이 나타내는 값과 상기 색차 절삭 색인이 나타내는 값이 동일할 때, 상기 휘도 절삭 값과 상기 색차 절삭 값이 동일할 수 있다.
- [984] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 상기 적응 파라미터 세트에 부여된 식별번호를 나타내는 적응 파라미터 세트 식별자 및 상기 적응 파라미터

세트에 포함된 부호화 정보의 타입을 나타내는 적응 파라미터 세트 타입 정보를 포함할 수 있다.

- [985] 일 실시 예에 따르면, 상기 적응 파라미터 세트 타입 정보가 ALF 타입을 나타낼 때, ALF 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다.
- [986] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 상기 적응 파라미터 세트가 휘도 성분에 대한 ALF를 포함하는지 여부를 나타내는 휘도 ALF 시그널링 플래그 및 상기 적응 파라미터 세트가 색차 성분에 대한 ALF를 포함하는지 여부를 나타내는 색차 ALF 시그널링 플래그를 포함할 수 있다.
- [987] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 휘도 시그널링 ALF의 개수를 나타내는 휘도 시그널링 ALF 개수 정보를 포함하고, 상기 휘도 시그널링 ALF 개수 정보가 상기 휘도 시그널링 ALF가 1개보다 많음을 나타낼 때, 휘도 ALF 세트의 소정 개수의 휘도 ALF가 참조하는 휘도 시그널링 ALF의 색인을 나타내는 휘도 ALF 델타 색인을 포함할 수 있다.
- [988] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 하나 이상의 휘도 시그널링 ALF를 포함하고, 상기 휘도 ALF 델타 색인에 따라, 상기 하나 이상의 휘도 시그널링 ALF로부터 상기 소정 개수의 휘도 ALF가 결정될 수 있다.
- [989] 일 실시 예에 따르면, 상기 적응 파라미터 세트는, 색차 ALF의 개수를 나타내는 색차 ALF 개수 정보를 포함하고, 색차 ALF 개수 정보가 나타내는 개수만큼, 색차 ALF를 포함할 수 있다.
- [990] 단계 3904에서 적응 파라미터 세트들 중에서, 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는, 상기 현재 픽처 또는 슬라이스에 적용되는 적응 파라미터 세트들 중 적어도 하나가 결정될 수 있다.
- [991] 일 실시 예에 따르면, 상기 현재 픽처 또는 슬라이스의 휘도 ALF 세트 개수가 결정될 수 있다. 그리고 상기 휘도 ALF 세트 개수 정보가 나타내는 개수의 휘도 ALF 세트 식별자가 결정될 수 있다.
- [992] 일 실시 예에 따르면, 상기 현재 픽처 또는 슬라이스의 색차 ALF 적용 정보가 결정될 수 있다. 상기 색차 ALF 적용 정보가 Cb 성분 및 Cr 성분 중 적어도 하나에 ALF가 적용됨을 나타낼 때, 색차 ALF 세트 식별자가 결정될 수 있다.
- [993] 단계 3906에서 현재 픽처 또는 슬라이스에 적용되는 적응 파라미터 세트로부터, 상기 현재 픽처 또는 슬라이스에 포함된 현재 CTB에 적용되는 ALF 세트를 포함하는, 상기 현재 CTB에 적용되는 적응 파라미터 세트가 결정될 수 있다.
- [994] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 휘도 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제1 ALF 부호화 트리 블록 플래그가 결정될 수 있다. 그리고, 상기 현재 부호화 트리 블록의 Cb 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 제2 ALF 부호화 트리 블록 플래그가 결정될 수 있다. 그리고, 상기 현재 부호화 트리 블록의 Cr 샘플에 적응적 루프내 필터링이 적용되는지 여부를 나타내는 상기 제3 ALF 부호화 트리 블록 플래그가

결정될 수 있다.

- [995] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 휘도 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 부호화 트리 블록에 적응 파라미터 세트의 ALF 세트가 적용되는 여부를 나타내는 적응 파라미터 세트 적용 플래그가 결정될 수 있다. 상기 적응 파라미터 세트 적용 플래그가 상기 현재 부호화 트리 블록에 적응 파라미터 세트의 ALF 세트가 적용됨을 나타낼 때, 상기 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는 한 개 이상의 적응 파라미터 세트로부터, 상기 현재 부호화 트리 블록에 적용되는 휘도 ALF 세트가 결정될 수 있다. 그리고 상기 적응 파라미터 세트 적용 플래그가 상기 현재 부호화 트리 블록에 적응 파라미터 세트의 ALF 세트가 적용되지 않음을 나타낼 때, 휘도 샘플에 대한 고정 ALF 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 고정된 필터가 결정될 수 있다.
- [996] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 Cb 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는 한 개 이상의 적응 파라미터 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 Cb ALF 세트를 포함하는 적응 파라미터 세트를 나타내는 제2 ALF 부호화 트리 블록 식별자가 결정될 수 있다. 상기 제2 ALF 부호화 트리 블록 식별자에 따라, 상기 현재 부호화 트리 블록에 적용되는 Cb ALF 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다.
- [997] 상기 현재 부호화 트리 블록의 Cr 샘플에 적응적 루프내 필터링이 적용될 때, 상기 현재 픽처 또는 슬라이스에 적용되는 ALF 세트를 포함하는 한 개 이상의 적응 파라미터 세트들 중, 상기 현재 부호화 트리 블록에 적용되는 Cr ALF 세트를 포함하는 적응 파라미터 세트를 나타내는 제3 ALF 부호화 트리 블록 식별자가 결정될 수 있다. 상기 제3 ALF 부호화 트리 블록 식별자에 따라, 상기 현재 부호화 트리 블록에 적용되는 Cr ALF 세트를 포함하는 적응 파라미터 세트가 결정될 수 있다.
- [998] 단계 3908에서 상기 결정된 현재 CTB에 적용되는 적응 파라미터 세트의 ALF 세트에 기초하여, 상기 현재 부호화 트리 블록이 필터링될 수 있다.
- [999] 일 실시 예에 따르면, 상기 현재 부호화 트리 블록의 기본 필터링 단위 블록에 대하여, 블록 분류 색인이 할당될 수 있다. 상기 블록 분류 색인은 방향성 정보와 활동성 정보를 이용하여 결정될 수 있다.
- [1000] 일 실시 예에 따르면, 상기 방향성 정보 및 활동성 정보 중 적어도 하나는 세로, 가로, 제1 대각, 제2 대각 방향 중 적어도 하나에 대한 기울기(gradient) 값에 기반하여 결정될 수 있다.
- [1001] 도 38 및 39의 실시 예는 예시에 불과하며, 도 38 및 39의 각 단계는 통상의 기술자에게 용이하게 변형 실시 가능하다. 또한, 도 38 및 39의 각 구성은 생략되거나 다른 구성으로 대체될 수 있다. 도 38의 비디오 복호화 방법은 도 2의 복호화기에서 수행될 수 있다. 그리고 도 39의 비디오 부호화 방법은 도 1의

부호화기에서 수행될 수 있다. 또한 하나 이상의 프로세서는 도 38 및 39의 각 단계를 구현한 명령들을 수행할 수 있다. 그리고 도 38 및 39의 각 단계를 구현한 명령들을 포함하는 프로그램 제품은 메모리 디바이스에 저장되거나, 온라인 상으로 유통될 수 있다.

- [1002] 상기 적응적 루프내 필터는 하나 이상의 적응적 루프내 필터 계수를 포함할 수 있다.
- [1003] 상기 휘도 시그널링 ALF의 개수는 적응적 루프내 필터 클래스(종류)를 의미할 수 있다. 상기 적응적 루프내 필터 클래스(종류)는 하나 이상의 적응적 루프내 필터 계수를 포함하며 필터링에 사용되는 필터 하나를 의미할 수 있다. 즉, 적응적 루프내 필터 클래스(종류) 별로 적응적 루프내 필터는 서로 상이할 수 있다.
- [1004] 상기 플래그, 인덱스(색인) 등의 구문 요소가 파라미터 세트, 헤더, CTU, CU, PU, TU, 블록 등 적어도 하나의 비트스트림 구조에 포함되어 있다는 것은 해당 플래그, 인덱스(색인) 등의 구문 요소가 엔트로피 부호화/복호화되는 것을 의미할 수 있다.
- [1005] 또한, 상기 플래그, 인덱스(색인) 등의 구문 요소가 시그널링 단위는 해당 파라미터 세트, 헤더, CTU, CU, PU, TU, 블록 단위로 한정되는 것은 아니며, 다른 파라미터 세트, 헤더 CTU, CU, PU, TU, 블록 단위에서 시그널링될 수 있다.
- [1006] 한편, 휘도 블록과 색차 블록에 대해 개별적으로 루프내 필터링을 수행할 수 있다. 색차 블록의 개별적 적응적 루프내 필터링 지원 여부에 대해 픽처/슬라이스/타일 그룹/타일CTU/CTB 레벨에서 제어 플래그를 시그널링할 수 있다. 휘도 블록과 색차 블록에 통합적으로 적응적 루프내 필터링을 수행하는 모드 또는 휘도 블록과 색차 블록에 대해 개별적으로 적응적 루프내 필터링 모드를 지원하는 플래그를 시그널링할 수 있다.
- [1007] 상기 적응적 루프내 필터링에 대한 구문 요소 중 휘도 성분에 대한 구문 요소는 휘도 ALF 및 휘도 ALF 세트를 부호화/복호화/획득하기 위해 사용될 수 있다. 또한, 상기 적응적 루프내 필터링에 대한 구문 요소 중 색차 성분에 대한 구문 요소는 색차 ALF 및 색차 ALF 세트를 부호화/복호화/획득하기 위해 사용될 수 있다.
- [1008] 상기 특정 단위에서 시그널링되는 적응적 루프내 필터링에 대한 구문 요소는 해당 단위에서 시그널링되는 적응적 루프내 필터링에 대한 구문 요소의 예이며, 이에 한정되지 않고, 시퀀스 파라미터 세트, 적응 파라미터 세트, 픽처 파라미터 세트, 픽처 헤더, 슬라이스 헤더 등의 중 적어도 하나의 단위에서 상기 적응적 루프내 필터링에 대한 구문 요소가 시그널링될 수 있으며, 시그널링되는 적응적 루프내 필터링에 대한 구문 요소의 이름은 변경되어 사용될 수 있다.
- [1009] 일 예로, 휘도 필터와 색차 필터에 대한 필터 계수 값은 휘도 필터와 색차 필터에서 서로 다른 이진화 방법을 사용하여 엔트로피 부호화/복호화될 수 있다.
- [1010] 다른 예로, 휘도 필터에 대한 필터 계수 값은 휘도 필터 내 계수들끼리 서로

다른 이진화 방법을 사용하여 엔트로피 부호화/복호화될 수 있다. 또한, 휘도 필터에 대한 필터 계수 값은 휘도 필터 내 계수들끼리 서로 동일한 이진화 방법을 사용하여 엔트로피 부호화/복호화될 수 있다.

- [1011] 또 다른 예로, 색차 필터에 대한 필터 계수 값은 색차 필터 내 계수들끼리 서로 다른 이진화 방법을 사용하여 엔트로피 부호화/복호화될 수 있다. 또한, 색차 필터에 대한 필터 계수 값은 색차 필터 내 계수들끼리 서로 동일한 이진화 방법을 사용하여 엔트로피 부호화/복호화될 수 있다.
- [1012] 또한, 상기 필터 정보 중 적어도 하나 이상을 엔트로피 부호화/복호화할 때, 주변 블록의 필터 정보 중 적어도 하나 또는 이전에 부호화/복호화된 필터 정보 중 적어도 하나 이상 또는 이전 영상에서 부호화/복호화된 필터 정보를 필터 정보에 대한 예측 값으로 사용하여 엔트로피 부호화/복호화할 수 있다.
- [1013] 또한, 상기 필터 정보 중 적어도 하나 이상을 엔트로피 부호화/복호화할 때, 서로 다른 성분의 필터 정보 중 적어도 하나를 필터 정보에 대한 예측 값으로 사용하여 엔트로피 부호화/복호화할 수 있다.
- [1014] 또한, 상기 필터 계수를 엔트로피 부호화/복호화할 때, 필터 내 다른 필터 계수들 중 적어도 하나를 예측 값으로 사용하여 엔트로피 부호화/복호화할 수 있다.
- [1015] 또한, 상기 필터 정보 엔트로피 부호화/복호화 방법 중 적어도 하나 이상의 조합을 이용해서 필터 정보를 엔트로피 부호화/복호화할 수 있다.
- [1016] 본 발명의 일 실시 예에 따를 때, 상기 적응적 루프내 필터링은 블록, CU, PU, TU, CB, PB, TB, CTU, CTB, 슬라이스, 타일, 타일 그룹, 픽처 단위 중 적어도 하나의 단위에서 수행될 수 있다. 상기 단위로 수행되는 경우, 블록 분류 단계, 필터링 수행 단계, 필터 정보 부호화/복호화 단계가 상기 블록, CU, PU, TU, CB, PB, TB, CTU, CTB, 슬라이스, 타일, 타일 그룹, 픽처 단위 중 적어도 하나의 단위로 수행되는 것을 의미할 수 있다.
- [1017] 본 발명의 일 실시 예에 따를 때, 상기 적응적 루프내 필터링은 디블록킹 필터, 샘플 적응적 오프셋, 양방향 필터링 중 적어도 하나의 수행 여부에 따라 수행 여부가 결정될 수 있다.
- [1018] 일 예로, 현재 영상 내 복원/복호 샘플 중 디블록킹 필터, 샘플 적응적 오프셋, 양방향 필터링 중 적어도 하나가 적용된 샘플에 대해서는 적응적 루프내 필터링을 수행할 수 있다.
- [1019] 다른 예로, 현재 영상 내 복원/복호 샘플 중 디블록킹 필터, 샘플 적응적 오프셋, 양방향 필터링 중 적어도 하나가 적용된 샘플에 대해서는 적응적 루프내 필터링을 수행하지 않을 수 있다.
- [1020] 또 다른 예로, 현재 영상 내 복원/복호 샘플 중 디블록킹 필터, 샘플 적응적 오프셋, 양방향 필터링 중 적어도 하나가 적용된 샘플의 경우, 블록 분류의 수행 없이 L개의 필터를 이용해서 현재 영상 내 복원/복호 샘플에 대해서 적응적 루프내 필터링을 수행할 수 있다. 여기서, L은 양의 정수이다.

- [1021] 한편, 현재 영상에 대해 적응적 루프내 필터링 시, 블록 분류의 수행 없이 L 개의 필터를 이용해서 현재 영상 내 복원/복호 샘플에 대해서 적응적 루프내 필터링을 수행할 수 있다. 여기서, L 은 양의 정수이다. 이때, 상기 시간적 계층 식별자에 관계없이, 블록 분류를 수행하지 않고 L 개의 필터를 이용해서 현재 영상 내 복원/복호 샘플에 대해 적응적 루프내 필터링을 수행할 수 있다.
- [1022] 한편, 현재 영상에 대해 적응적 루프내 필터링 시, 블록 분류의 수행 여부에 관계없이 L 개의 필터를 이용해서 현재 영상 내 복원/복호 샘플에 대해서 적응적 루프내 필터링을 수행할 수 있다. 여기서, L 은 양의 정수이다. 이때, 상기 시간적 계층 식별자 및 블록 분류의 수행 여부에 관계없이 L 개의 필터를 이용해서 현재 영상 내 복원/복호 샘플에 대해 적응적 루프내 필터링을 수행할 수 있다.
- [1023] 한편, 하나의 필터 형태를 이용해서 적응적 루프내 필터링을 수행할 수 있다. 이때, 블록 분류를 수행하지 않고, 하나의 필터 형태를 이용해서 현재 영상 내 복원/복호 샘플에 대해 적응적 루프내 필터링을 수행할 수 있다. 또는, 블록 분류의 수행 여부에 관계없이 하나의 필터 형태를 이용해서 현재 영상 내 복원/복호 샘플에 대해 적응적 루프내 필터링을 수행할 수 있다.
- [1024] 한편, 하나의 필터 탭 수를 이용해서 적응적 루프내 필터링을 수행할 수 있다. 이때, 블록 분류를 수행하지 않고, 하나의 필터 탭 수를 이용해서 현재 영상 내 복원/복호 샘플에 대해 적응적 루프내 필터링을 수행할 수 있다. 또는, 블록 분류의 수행 여부에 관계없이 하나의 필터 탭 수를 이용해서 현재 영상 내 복원/복호 샘플에 대해 적응적 루프내 필터링을 수행할 수 있다.
- [1025] 한편, 상기 적응적 루프내 필터링은 특정 단위 별로 수행될 수 있다. 예를 들어, 상기 특정 단위는 픽처, 슬라이스, 타일, 타일 그룹, CTU, CTB, CU, PU, TU, CB, PB, TB, $M \times N$ 크기의 블록 중 적어도 하나의 단위일 수 있다. 여기서, M 과 N 은 양의 정수일 수 있고, 서로 같은 값을 가지거나 서로 다른 값을 가질 수 있다. 또한, M 과 N 중 적어도 하나는 부호화기/복호화기에서 기정의된 값일 수 있고, 부호화기에서 복호화기로 시그널링되는 값일 수 있다.
- [1026]
- [1027] 이하, 40 내지 도 52를 참고하여, 본 개시의 또 다른 일 실시예에 따른 적응적 루프내 필터링 방법 및 장치에 대해서 설명한다.
- [1028] 도 40은 본 개시의 또 다른 일 실시예에 따른 영상 복호화 방법을 설명하기 위한 도면이다.
- [1029] 도 40을 참조하면, 본 개시의 일 실시예에 따른 영상 복호화 방법은, 현재 블록의 복원 휘도 블록을 획득하는 단계(S4010), 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component adaptive loop filter)의 적용 여부를 결정하는 단계(S4020), 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우, 적응적 루프 필터(adaptive loop filter)가 적용된 현재 블록의 복원 색차 블록을 획득하는 단계(S4030) 및 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된

현재 블록의 최종 복원 색차 블록을 획득하는 단계(S4040)을 포함할 수 있다.

[1030] 도 41은 본 개시의 또 다른 일 실시예에 따른 영상 부호화 방법을 설명하기 위한 도면이다.

[1031] 도 41을 참조하면, 본 개시의 일 실시예에 따른 영상 부호화 방법은, 현재 블록의 복원 휘도 블록을 획득하는 단계(S4110), 현재 블록에 대한 성분 간 적응적 루프 필터의 적용 여부를 결정하는 단계(S4120), 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우, 적응적 루프 필터가 적용된 현재 블록의 복원 색차 블록을 획득하는 단계(S4130) 및 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의 최종 복원 색차 블록을 획득하는 단계(S4140)을 포함할 수 있다.

[1032] 도 42는 본 개시의 또 다른 일 실시예에 따른 시퀀스 파라미터 세트(Sequence Parameter Set)를 설명하기 위한 도면이다.

[1033] 일 예로, 시퀀스 파라미터 세트는 구문 요소 `sps_alf_enabled_flag`를 포함할 수 있다. 이때, `sps_alf_enabled_flag`는 적응적 루프내 필터링이 시퀀스 단위에서 수행되는지에 여부를 지시할 수 있다. 예컨대, `sps_alf_enabled_flag`가 제 1 값(예컨대, 0)인 경우, 시퀀스 단위에서 적응적 루프내 필터링이 수행되지 않음이 지시될 수 있고, `sps_alf_enabled_flag`가 제 2 값(예컨대, 1)인 경우 시퀀스 단위에서 적응적 루프내 필터링이 수행됨이 지시될 수 있다.

[1034] 한편, `sps_alf_enabled_flag`가 비트스트림 내에 존재하지 않는 경우 `sps_alf_enabled_flag`는 제1 값으로 추론(infer)될 수 있다.

[1035] 이하의 설명에서 제 1 값은 0, 제 2 값은 1인 경우를 예를 들어 설명하나, 제 1 값 및 제 2 값은 이러한 예시에 한정되지 않으며, 제 1 값이 1, 제 2 값이 0을 가질 수도 있다. 또한 제 1 값 및 제 2 값은 0 및 1 외에 다른 값으로 정의될 수도 있다. 또한, 제 3 값은 2, 제 4 값은 3인 예를 설명하나 제 3 값 및 제 4 값 역시 이러한 예시에 한정되지 않는다.

[1036] 다른 예로, 시퀀스 파라미터 세트는 구문 요소 `sps_ccalf_enabled_flag`를 포함할 수 있다. 이때, `sps_ccalf_enabled_flag`는 성분 간 적응적 루프내 필터링(Cross-component adaptive loop filter)이 적용될 수 있는지 여부를 지시할 수 있다. 예컨대, `sps_ccalf_enabled_flag`가 제 1 값인 경우, 시퀀스 단위에서 성분 간 적응적 루프내 필터링이 적용되지 않을 수 있고, `sps_ccalf_enabled_flag`가 제 2 값인 경우 시퀀스 단위에서 성분 간 적응적 루프내 필터링이 수행될 수 있다.

[1037] 한편, `sps_ccalf_enabled_flag`가 비트스트림 내에 존재하지 않는 경우 `sps_ccalf_enabled_flag`는 제1 값으로 추론(infer)될 수 있다.

[1038] 이하의 설명에서, 성분 간 적응적 루프내 필터링은 휘도 성분과 색차 성분의 의존성에 기초하여, 색차 성분의 적응적 루프내 필터링의 효율을 향상시키는 기술을 의미할 수 있다. 구체적으로, 부호화기/복호화기는 휘도 성분에 대해 성분 간 적응적 루프내 필터링을 적용한 샘플과 색차 성분에 대해 적응적 루프내

필터링을 적용한 샘플을 이용하여 최종적으로 필터링된 색차 성분 필터링 결과 샘플을 획득할 수 있다. 성분 간 적응적 루프내 필터링을 수행하는 구체적인 방법에 대해서는 후술한다.

- [1039] 도 43은 본 개시의 또 다른 일 실시예에 따른 적응 파라미터 세트(Adaptation Parameter Set)를 설명하기 위한 도면이다.
- [1040] 일 예로, 적응 파라미터 세트는 구문 요소 `adaptation_parameter_set_id` (또는 `aps_adaptation_parameter_set_id`)를 포함할 수 있다. `adaptation_parameter_set_id`는 적응 파라미터 식별자를 의미할 수 있다.
- [1041] 한편, 구문 요소 `adaptation_parameter_set_id`가 비트스트림 내에 존재하지 않는 경우 `adaptation_parameter_set_id`는 제 1 값으로 추론될 수 있다.
- [1042] 도 44는 본 개시의 또 다른 일 실시예에 따른 적응 파라미터를 설명하기 위한 도면이다.
- [1043] 일 예로, 적응 파라미터 세트는 구문 요소 `aps_params_type`를 포함할 수 있다. 이때, 구문 요소 `aps_params_type`는 적응 파라미터 세트 내에 존재하는 적응 파라미터의 타입(type)을 지시할 수 있다.
- [1044] 예컨대, `aps_params_type`가 제 1 값인 경우, 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소들은 적응적 루프내 필터링에 관한 파라미터일 수 있다. 반면, `aps_params_type`가 제 2 값인 경우, 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소들은 색차 스케일링 및 휘도 매핑(LMCS; luma mapping with chroma scaling)에 관한 파라미터일 수 있다.
- [1045] 또 다른 예로, `aps_params_type`는 `adaptation_parameter_set_id`보다 먼저 시그널링될 수 있다.
- [1046] 한편, 구문 요소 `aps_params_type`가 비트스트림 내에 존재하지 않을 경우 `aps_params_type`는 제 1 값과 제 2 값이 아닌 다른 값으로 추론될 수 있다. 예컨대, 다른 값은 도 44의 예시와 같이 2 내지 7 중 하나의 값일 수 있다.
- [1047] 한편, 이전에 시그널링된 적응 파라미터 식별자와 적응 파라미터의 타입 중 적어도 하나와 동일한 적응 파라미터 식별자 및 적응 파라미터의 타입과 동일한 적응 파라미터의 타입이 새롭게 시그널링되는 경우, 이전에 시그널링된 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값은 새롭게 시그널링되는 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값으로 대체될 수 있다. 이때, 대체 과정은 업데이트(update) 과정을 의미할 수 있다.
- [1048] 즉, 부호화기/복호화기는 이전에 시그널링된 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값을 참조하여, 적응적 루프내 필터링, 색차 스케일링 및 휘도 매핑 중 적어도 하나를 수행할 수 있다. 이전에 시그널링된 적응 파라미터 식별자와 적응 파라미터의 타입 중 적어도 하나와 동일한 적응 파라미터 세트의 식별자 및 동일한 적응 파라미터의 타입이 새롭게 시그널링된 시점부터 부호화기/복호화기는 새롭게 시그널링된 시점의 적응 파라미터 세트 내 데이터/컨텐츠/구문 요소 값을 참조하여, 적응적 루프내 필터링, 색차

스케일링 및 휘도 매핑 중 적어도 하나를 수행할 수 있다.

[1049] 도 45는 본 개시의 또 다른 일 실시예에 따른 픽처 헤더(Picture Header)를 설명하기 위한 도면이다.

[1050] 일 예로, 픽처 헤더는 구문 요소 `pic_alf_enabled_flag`(또는 `ph_alf_enabled_flag`)를 포함할 수 있다. `pic_alf_enabled_flag`는 적응적 루프내 필터링이 픽처 내 모든 슬라이스 단위에서 Y, Cb, Cr 성분 중 적어도 하나에 대해 수행되는지 여부를 지시할 수 있다.

[1051] 예컨대, `pic_alf_enabled_flag`가 제 1 값인 경우 픽처 내 적어도 하나의 슬라이스 단위에서 Y, Cb, Cr 성분에 대해 적응적 루프내 필터링이 수행되지 않을 수 있고, `pic_alf_enabled_flag`가 제 2 값인 경우 픽처 내 모든 슬라이스 단위에서 Y, Cb, Cr 성분 중 적어도 하나에 대해 적응적 루프내 필터링이 수행될 수 있다.

[1052] 한편, `pic_alf_enabled_flag`가 비트스트림 내에 존재하지 않는 경우 `pic_alf_enabled_flag`는 제 1 값으로 추론될 수 있다.

[1053] 다른 예로, 픽처 헤더는 구문 요소 `pic_num_alf_aps_ids_luma`(또는 `ph_num_alf_aps_ids_luma`)를 포함할 수 있다. `pic_num_alf_aps_ids_luma`는 픽처 헤더에 연관된 슬라이스에서 참조하는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 개수를 의미할 수 있다.

[1054] 예컨대, `pic_num_alf_aps_ids_luma`는 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 6 혹은 8일 수 있다.

[1055] 한편, `pic_num_alf_aps_ids_luma`가 비트스트림 내에 존재하지 않는 경우 `pic_num_alf_aps_ids_luma`는 제 1 값으로 추론될 수 있다.

[1056] 또 다른 예로, 픽처 헤더는 구문 요소 `pic_alf_aps_id_luma`(또는 `ph_alf_aps_id_luma`)를 포함할 수 있다. `pic_alf_aps_id_luma[i]`는 픽처 헤더에 연관된 슬라이스에서 참조하는 i번째 적응적 루프내 필터링에 대한 적응 파라미터 세트 식별자를 의미할 수 있다.

[1057] 예컨대, `aps_params_type`가 ALF_APS이며 `adaptation_parameter_set_id`가 `pic_alf_aps_id_luma[i]`와 동일한 APS NAL 유닛의 `alf_luma_filter_signal_flag` 값은 제 2 값일 수 있다.

[1058] 또한, `sps_ccalf_enabled_flag`가 제 1 값이고, `aps_params_type`가 ALF_APS이며 `adaptation_parameter_set_id`가 `pic_alf_aps_id_luma[i]`와 동일한 APS NAL 유닛의 `alf_cc_cb_filter_signal_flag` 및 `alf_cc_cr_filter_signal_flag`의 값은 제 1 값일 수 있다.

[1059] 한편, `pic_alf_aps_id_luma[i]`가 비트스트림 내에 존재하지 않을 경우 `pic_alf_aps_id_luma[i]`는 제 1 값으로 추론될 수 있다.

[1060] 일 예로, `pic_alf_aps_id_luma[i]`와 동일한 `adaptation_parameter_set_id`를 가지는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 시간적 계층 식별자(Temporal Layer Identifier)는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다.

- [1061] 다른 예로, 동일한 adaptation_parameter_set_id를 가지는 하나의 픽처 내의 2개 이상의 서브 픽처/슬라이스/타일 그룹/타일의 적응적 루프내 필터링에 대한 적응 파라미터 세트가 존재하는 경우, 동일한 adaptation_parameter_set_id를 가지는 2개 이상의 적응 파라미터 세트들은 동일한 데이터/콘텐츠/구문 요소 값을 가질 수 있다.
- [1062] 또 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우, pic_alf_aps_id_luma[i]는 화면내 슬라이스를 포함하는 픽처 또는 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1063] 또 다른 예로, 픽처 헤더는 구문 요소 pic_alf_chroma_idc를 포함할 수 있다. 예컨대, pic_alf_chroma_idc가 제 1 값인 경우, Cb 및 Cr 성분에 적응적 루프내 필터링이 수행되지 않을 수 있다. pic_alf_chroma_idc가 제 2 값인 경우, Cb 성분에 적응적 루프내 필터링이 수행될 수 있다. pic_alf_chroma_idc가 제 3 값(예컨대, 2)인 경우, Cr 성분에 적응적 루프내 필터링이 수행될 수 있다. pic_alf_chroma_idc가 제 4 값(예컨대, 3)인 경우, Cb 및 Cr 성분에 적응적 루프내 필터링이 수행될 수 있다.
- [1064] 한편, pic_alf_chroma_idc가 비트스트림 내에 존재하지 않을 경우 pic_alf_chroma_idc는 제 1 값으로 추론될 수 있다.
- [1065] 또 다른 예로, 픽처 헤더는 구문 요소 ph_alf_cb_enabled_flag 및/또는 ph_alf_cr_enabled_flag를 포함할 수 있다. ph_alf_cb_enabled_flag는 Cb 성분에 대한 적응적 루프내 필터링이 수행될 수 있는지 여부를 지시할 수 있다. ph_alf_cr_enabled_flag는 Cr 성분에 대한 적응적 루프내 필터링이 수행될 수 있는지 여부를 지시할 수 있다.
- [1066] 예컨대, ph_alf_cb_enabled_flag이 제 1 값인 경우, Cb 성분에 적응적 루프내 필터링이 수행되지 않을 수 있고, ph_alf_cb_enabled_flag이 제 2 값인 경우, Cb 성분에 적응적 루프내 필터링이 수행될 수 있다. 한편, ph_alf_cr_enabled_flag이 제 1 값인 경우, Cr 성분에 적응적 루프내 필터링이 수행되지 않을 수 있고, ph_alf_cr_enabled_flag이 제 2 값인 경우, Cr 성분에 적응적 루프내 필터링이 수행될 수 있다.
- [1067] 한편, ph_alf_cb_enabled_flag이 비트스트림 내에 존재하지 않을 경우 ph_alf_cb_enabled_flag는 제 1 값으로 추론될 수 있다. 또한, ph_alf_cr_enabled_flag이 비트스트림 내에 존재하지 않을 경우 ph_alf_cr_enabled_flag는 제 1 값으로 추론될 수 있다.
- [1068] 또 다른 예로, 픽처 헤더는 구문 요소 pic_alf_aps_id_chroma(또는 ph_alf_aps_id_chroma)를 포함할 수 있다. pic_alf_aps_id_chroma는 픽처 헤더에 연관된 슬라이스의 색차 성분이 참조하는 적응 파라미터 세트 식별자를 의미할 수 있다.
- [1069] 예컨대, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가

pic_alf_aps_id_chroma와 동일한 APS NAL 유닛의 alf_chroma_filter_signal_flag 값은 제 2 값일 수 있다.

- [1070] 또한, sps_ccalf_enabled_flag가 제 1 값이고, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 pic_alf_aps_id_chroma 와 동일한 APS NAL 유닛의 alf_cc_cb_filter_signal_flag 및 alf_cc_cr_filter_signal_flag의 값은 제 1 값일 수 있다.
- [1071] 한편, pic_alf_aps_id_chroma가 비트스트림 내에 존재하지 않을 경우 pic_alf_aps_id_chroma는 제 1 값으로 추론될 수 있다.
- [1072] 일 예로, pic_alf_aps_id_chroma와 동일한 adaptation_parameter_set_id를 가지는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 시간적 계층 식별자는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다.
- [1073] 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우, pic_alf_aps_id_chroma는 화면내 슬라이스를 포함하는 픽처 혹은 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1074] 또 다른 예로, 픽처 헤더는 pic_alf_aps_id_chroma 대신 구문 요소 pic_alf_aps_id_chroma[i](또는 ph_alf_aps_id_chroma[i])를 포함할 수 있다. pic_alf_aps_id_chroma[i]는 픽처 헤더에 연관된 슬라이스에서 참조하는 i번째 적응적 루프내 필터링에 대한 적응 파라미터 세트 식별자를 의미할 수 있다. 색차 성분에 대한 적응적 루프내 필터링 시 적응적 루프내 필터 정보를 포함하는 2개 이상의 적응 파라미터 세트 중 하나의 적응 파라미터 세트를 선택하여, 색차 성분에 대한 적응적 루프내 필터링이 수행될 수 있다.
- [1075] 예컨대, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 pic_alf_aps_id_chroma[i]와 동일한 APS NAL 유닛의 alf_chroma_filter_signal_flag 값은 제 2 값일 수 있다.
- [1076] 한편, pic_alf_aps_id_chroma[i]가 비트스트림 내에 존재하지 않을 경우 pic_alf_aps_id_chroma[i]는 제1 값으로 추론될 수 있다.
- [1077] 일 예로, pic_alf_aps_id_chroma[i]와 동일한 adaptation_parameter_set_id를 가지는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 시간적 계층 식별자는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다.
- [1078] 다른 예로, 하나의 픽처 내, 동일한 adaptation_parameter_set_id를 가지는 2개 이상의 서브 픽처/슬라이스/타일 그룹/타일/블록의 적응 파라미터 세트가 존재하는 경우, 동일한 adaptation_parameter_set_id를 가지는 2개 이상의 적응적 적응 파라미터 세트들은 동일한 데이터/콘텐츠/구문 요소 값을 가질 수 있다.
- [1079] 또 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우, pic_alf_aps_id_chroma[i]는 화면내 슬라이스를 포함하는 픽처 또는 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1080] 또 다른 예로, 픽처 헤더는 구문 요소

- pic_cross_component_alf_cb_enabled_flag(또는 ph_alf_cc_cb_enabled_flag) 및/또는 pic_cross_component_alf_cb_aps_id(또는 ph_alf_cc_cb_aps_id)를 포함할 수 있다.
- [1081] 예컨대, pic_cross_component_alf_cb_enabled_flag가 제 1 값인 경우, 성분 간 Cb 필터가 Cb 색 성분에 적용되지 않을 수 있다. 또한, pic_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, 성분 간 Cb 필터가 Cb 색 성분에 적용될 수 있다.
- [1082] 한편, pic_cross_component_alf_cb_enabled_flag가 비트스트림 내에 존재하지 않을 경우 pic_cross_component_alf_cb_enabled_flag는 제 1 값으로 추론될 수 있다.
- [1083] pic_cross_component_alf_cb_aps_id는 픽처 헤더에 연관된 슬라이스의 Cb 색 성분이 참조하는 adaptation_parameter_set_id를 지시할 수 있다.
- [1084] 일 예로, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 pic_cross_component_alf_cb_aps_id와 동일한 APS NAL 유닛의 TemporalId는 픽처 헤더에 연관된 현재 슬라이스의 NAL 유닛의 TemporalId보다 작거나 같을 수 있다.
- [1085] 다른 예로, 픽처 헤더에 연관된 현재 슬라이스가 화면내 슬라이스인 경우, pic_cross_component_alf_cb_aps_id는 화면내 슬라이스를 포함하는 픽처 혹은 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1086] 또 다른 예로, pic_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, 현재 픽처 내 모든 슬라이스, 모든 타일, 혹은 모든 서브픽처에서 pic_cross_component_alf_cb_aps_id에 의해 참조되는 ALF APS는 모두 같을 수 있다. 한편, pic_cross_component_cb_filters_signalled_minus1에 +1한 값은 성분 간 Cb 필터의 개수를 지시할 수 있다. 이때, pic_cross_component_cb_filters_signalled_minus1은 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있으며, 예컨대 3일 수 있다.
- [1087] 또 다른 예로, pic_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, pic_cross_component_cb_filters_signalled_minus1에 대한 별도의 시그널링 없이 pic_cross_component_cb_filters_signalled_minus1는 픽처 헤더에 연관된 현재 슬라이스에서 참조하는 pic_cross_component_alf_cb_aps_id의 ALF APS 내의 alf_cross_component_cb_filters_signalled_minus1 값으로 설정될 수 있다.
- [1088] 또 다른 예로, pic_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, pic_cross_component_cb_filters_signalled_minus1는 픽처 헤더에 연관된 현재 슬라이스에서 참조하는 pic_cross_component_alf_cb_aps_id의 ALF APS 내의 alf_cross_component_cb_filters_signalled_minus1 값보다 작거나 같을 수 있다.
- [1089] 또 다른 예로, 픽처 헤더는 구문 요소 pic_cross_component_alf_cr_enabled_flag(또는 ph_alf_cc_cr_enabled_flag) 및/또는 pic_cross_component_alf_cr_aps_id(또는 ph_alf_cc_cr_aps_id)를 포함할 수 있다.

- [1090] 예컨대, `pic_cross_component_alf_cr_enabled_flag`가 제1 값인 경우, 성분 간 Cr 필터가 Cr 색 성분에 적용되지 않을 수 있다. 또한, `pic_cross_component_alf_cr_enabled_flag`가 제 2 값인 경우, 성분 간 Cr 필터가 Cr 색 성분에 적용됨을 지시할 수 있다.
- [1091] 한편, `pic_cross_component_alf_cr_enabled_flag`가 비트스트림 내에 존재하지 않을 경우 `pic_cross_component_alf_cr_enabled_flag`는 제 1 값으로 추론될 수 있다.
- [1092] `pic_cross_component_alf_cr_aps_id`는 픽처 헤더에 연관된 슬라이스의 Cr 색 성분이 참조하는 `adaptation_parameter_set_id`를 지시할 수 있다.
- [1093] 일 예로, `aps_params_type`가 ALF_APS이며 `adaptation_parameter_set_id`가 `pic_cross_component_alf_cr_aps_id`와 동일한 APS NAL 유닛의 TemporalId는 픽처 헤더에 연관된 현재 슬라이스의 NAL 유닛의 TemporalId와 작거나 같을 수 있다.
- [1094] 다른 예로, 픽처 헤더에 연관된 현재 슬라이스가 화면내 슬라이스인 경우, `pic_cross_component_alf_cr_aps_id`는 화면내 슬라이스를 포함하는 픽처 혹은 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1095] 또 다른 예로, `pic_cross_component_alf_cr_enabled_flag`가 제2 값인 경우, 현재 픽처 내 모든 슬라이스, 모든 타일, 혹은 모든 서브픽처에서 `pic_cross_component_alf_cr_aps_id`에 의해 참조되는 ALF APS는 모두 같을 수 있다. 한편, `pic_cross_component_cr_filters_signalled_minus1`에 +1한 값은 성분 간 Cr 필터의 개수를 지시할 수 있다. 이때, `pic_cross_component_cr_filters_signalled_minus1`은 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 3일 수 있다.
- [1096] 또 다른 예로, `pic_cross_component_alf_cr_enabled_flag`가 제 2 값인 경우, `pic_cross_component_cr_filters_signalled_minus1`에 대한 별도의 시그널링 없이 `pic_cross_component_cr_filters_signalled_minus1`는 픽처 헤더에 연관된 현재 슬라이스에서 참조하는 `pic_cross_component_alf_cr_aps_id`의 ALF APS 내의 `alf_cross_component_cr_filters_signalled_minus1` 값으로 설정될 수 있다.
- [1097] 또 다른 예로, `pic_cross_component_alf_cr_enabled_flag`가 제 2 값인 경우, `pic_cross_component_cr_filters_signalled_minus1`는 픽처 헤더에 연관된 현재 슬라이스에서 참조하는 `pic_cross_component_alf_cr_aps_id`의 ALF APS 내의 `alf_cross_component_cr_filters_signalled_minus1` 값보다 작거나 같을 수 있다.
- [1098] 앞선 설명에서, 픽처 헤더에 연관된 슬라이스는 해당 픽처 헤더 내 적어도 하나의 구문 요소를 참조하는 슬라이스를 의미하거나, 해당 픽처 헤더에 대응하는 픽처에 속한 슬라이스를 의미할 수 있다.
- [1099] 도 46은 본 개시의 또 다른 일 실시예에 따른 슬라이스 헤더(slice header)를 설명하기 위한 도면이다.
- [1100] 일 예로, 슬라이스 헤더는 구문 요소 `slice_alf_enabled_flag`(또는 `sh_alf_enabled_flag`)를 포함할 수 있다. `slice_alf_enabled_flag`는 적응적 루프내

필터링이 슬라이스 단위에서 Y, Cb, Cr 성분 중 적어도 하나에 대해 수행되는지 여부를 지시할 수 있다.

- [1101] 예컨대, slice_alf_enabled_flag가 제 1 값인 경우 슬라이스 단위에서 Y, Cb, Cr 성분 모두에 대해 적응적 루프내 필터링이 수행되지 않을 수 있다. 반면, slice_alf_enabled_flag가 제 2 값인 경우 슬라이스 단위에서 Y, Cb, Cr 성분 중 적어도 하나에 대해 적응적 루프내 필터링이 수행될 수 있다.
- [1102] 한편, slice_alf_enabled_flag가 비트스트림 내에 존재하지 않을 경우 slice_alf_enabled_flag는 제 1 값 혹은 pic_alf_enabled_flag으로 추론될 수 있다.
- [1103] 다른 예로, 슬라이스 헤더는 구문 요소 slice_num_alf_aps_ids_luma(또는 sh_num_alf_aps_ids_luma)를 포함할 수 있다. slice_num_alf_aps_ids_luma는 슬라이스에서 참조하는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 개수를 의미할 수 있다.
- [1104] 예컨대, slice_num_alf_aps_ids_luma는 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 6일 수 있다.
- [1105] 한편, slice_num_alf_aps_ids_luma가 비트스트림 내에 존재하지 않을 경우 slice_num_alf_aps_ids_luma는 제1 값 혹은 pic_num_alf_aps_ids_luma으로 추론될 수 있다.
- [1106] 부호화기/복호화기에서 기설정된 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수와 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수는 서로 동일할 수 있다. 이때, 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수와 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수의 합은 적응 파라미터 세트의 최대 개수일 수 있다.
- [1107] 부호화기/복호화기에서 기설정된 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수와 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수는 서로 상이할 수 있다. 이때, 적응적 루프내 필터링에 대한 적응 파라미터 세트의 최대 개수와 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 최대 개수의 합은 파라미터 세트의 최대 개수일 수 있다.
- [1108] 일 예로, 적응적 루프내 필터링에 대한 적응 파라미터 세트의 개수는 화면내 슬라이스인 경우 최대 K개 일 수 있고, 화면간 슬라이스인 경우 최대 L개 일 수 있다. 이때, K와 L은 양의 정수일 수 있다. 예컨대, K는 1, L은 6일 수 있다.
- [1109] 다른 예로, 상기 적응적 루프내 필터링에 대한 적응 파라미터 세트의 개수는 슬라이스의 종류에 관계없이 최대 J개일 수 있다. 여기서, J는 양의 정수일 수 있다. 예컨대, J는 6 또는 8일 수 있다.
- [1110] 또 다른 예로, 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 개수는 화면내 슬라이스인 경우 최대 K일 수 있고, 화면간 슬라이스인 경우 최대 L개 일 수 있다. 이때, K와 L은 양의 정수일 수 있다. 예컨대, K는 1, L은 6일 수 있다.
- [1111] 또 다른 예로, 색차 스케일링 및 휘도 매핑에 대한 적응 파라미터 세트의 개수는 슬라이스의 종류에 관계없이 최대 J개 일 수 있다. 이때, J는 양의 정수일 수 있다.

예컨대, J는 6일 수 있다.

- [1112] 또 다른 예로, 슬라이스 헤더는 구문 요소 slice_alf_aps_id_luma(또는 sh_alf_aps_id_luma)를 포함할 수 있다. slice_alf_aps_id_luma[i]는 슬라이스에서 참조하는 i번째 적응적 루프내 필터링에 대한 적응 파라미터 세트 식별자를 의미할 수 있다.
- [1113] 예컨대, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 slice_alf_aps_id_luma[i]와 동일한 APS NAL 유닛의 alf_luma_filter_signal_flag 값은 제 2 값일 수 있다.
- [1114] 한편, slice_alf_aps_id_luma[i]가 비트스트림 내에 존재하지 않을 경우 slice_alf_aps_id_luma[i]는 제1 값 혹은 pic_alf_aps_id_luma[i]으로 추론될 수 있다.
- [1115] 일 예로, slice_alf_aps_id_luma[i]와 동일한 adaptation_parameter_set_id를 가지는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 시간적 계층 식별자는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다.
- [1116] 또한, sps_ccalf_enabled_flag가 제 1 값이고, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 slice_alf_aps_id_luma[i]와 동일한 APS NAL 유닛의 alf_cc_cb_filter_signal_flag 및 alf_cc_cr_filter_signal_flag의 값은 제 1 값일 수 있다.
- [1117] 다른 예로, 동일한 adaptation_parameter_set_id를 가지는 하나의 픽처 내의 2개 이상의 서브 픽처/슬라이스/타일 그룹/타일/브릭의 적응적 루프내 필터링에 대한 적응 파라미터 세트가 존재하는 경우, 동일한 adaptation_parameter_set_id를 가지는 2개 이상의 적응 파라미터 세트는 동일한 데이터/콘텐츠/구문 요소 값을 가질 수 있다.
- [1118] 또 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우, slice_alf_aps_id_luma[i]는 화면내 슬라이스를 포함하는 픽처 또는 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1119] 또 다른 예로, 슬라이스 헤더는 구문 요소 slice_alf_chroma_idc를 포함할 수 있다. 예컨대, slice_alf_chroma_idc가 제1 값인 경우, Cb 및 Cr 성분에 적응적 루프내 필터링이 수행되지 않을 수 있다. slice_alf_chroma_idc가 제 2 값인 경우, Cb 성분에 적응적 루프내 필터링이 수행될 수 있다. slice_alf_chroma_idc가 제 3 값인 경우, Cr 성분에 적응적 루프내 필터링이 수행될 수 있다. slice_alf_chroma_idc가 제 4 값인 경우, Cb 및 Cr 성분에 적응적 루프내 필터링이 수행될 수 있다.
- [1120] 한편, slice_alf_chroma_idc가 비트스트림 내에 존재하지 않을 경우 slice_alf_chroma_idc는 제 1 값으로 추론될 수 있다.
- [1121] 또 다른 예로, 픽처 헤더는 구문 요소 sh_alf_cb_enabled_flag 및/또는 sh_alf_cr_enabled_flag를 포함할 수 있다. sh_alf_cb_enabled_flag는 Cb 성분에

대한 적응적 루프내 필터링이 수행될 수 있는지 여부를 지시할 수 있다.

sh_alf_cr_enabled_flag는 Cr 성분에 대한 적응적 루프내 필터링이 수행될 수 있는지 여부를 지시할 수 있다.

[1122] 예컨대, sh_alf_cb_enabled_flag이 제 1 값인 경우, Cb 성분에 적응적 루프내 필터링이 수행되지 않을 수 있고, sh_alf_cb_enabled_flag이 제 2 값인 경우, Cb 성분에 적응적 루프내 필터링이 수행될 수 있다. 한편, sh_alf_cr_enabled_flag이 제 1 값인 경우, Cr 성분에 적응적 루프내 필터링이 수행되지 않을 수 있고, sh_alf_cr_enabled_flag이 제 2 값인 경우, Cr 성분에 적응적 루프내 필터링이 수행될 수 있다.

[1123] 한편, sh_alf_cb_enabled_flag이 비트스트림 내에 존재하지 않을 경우 sh_alf_cb_enabled_flag는 제 1 값으로 추론될 수 있다. 또한, sh_alf_cr_enabled_flag이 비트스트림 내에 존재하지 않을 경우 sh_alf_cr_enabled_flag는 제 1 값으로 추론될 수 있다.

[1124] 또 다른 예로, 슬라이스 헤더는 구문 요소 slice_alf_aps_id_chroma(또는 sh_alf_aps_id_chroma)를 포함할 수 있다. slice_alf_aps_id_chroma는 슬라이스의 색차 성분에서 참조하는 적응 파라미터 세트의 식별자를 의미할 수 있다.

[1125] 예컨대, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 slice_alf_aps_id_chroma와 동일한 APS NAL 유닛의 alf_chroma_filter_signal_flag 값은 제 2 값일 수 있다.

[1126] 또한, sps_ccalf_enabled_flag가 제 1 값이고, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 slice_alf_aps_id_chroma와 동일한 APS NAL 유닛의 alf_cc_cb_filter_signal_flag 및 alf_cc_cr_filter_signal_flag의 값은 제 1 값일 수 있다.

[1127] 한편, slice_alf_aps_id_chroma가 비트스트림 내에 존재하지 않을 경우 slice_alf_aps_id_chroma는 제1 값 혹은 pic_alf_aps_id_chroma으로 추론될 수 있다.

[1128] 일 예로, slice_alf_aps_id_chroma와 동일한 adaptation_parameter_set_id를 가지는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 시간적 계층 식별자는 현재 슬라이스의 시간적 계층식별자보다 작거나 같을 수 있다.

[1129] 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우, slice_alf_aps_id_chroma는 화면내 슬라이스를 포함하는 픽처 혹은 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.

[1130] 또 다른 예로, 슬라이스 헤더는 slice_alf_aps_id_chroma 대신 구문 요소 slice_alf_aps_id_chroma[i]를 포함할 수 있다. slice_alf_aps_id_chroma[i]는 픽처 헤더에 연관된 슬라이스에서 참조하는 i번째 적응적 루프내 필터링에 대한 적응 파라미터 세트 식별자를 의미할 수 있다. 색차 성분에 대한 적응적 루프내 필터링 시 적응적 루프내 필터 정보를 포함하는 2개 이상의 적응 파라미터 세트 중 하나의 적응 파라미터 세트를 선택하여, 색차 성분에 대한 적응적 루프내 필터링이 수행될 수 있다.

- [1131] 예컨대, `aps_params_type`가 `ALF_APS`이며 `adaptation_parameter_set_id`가 `slice_alf_aps_id_chroma[i]`와 동일한 APS NAL 유닛의 `alf_chroma_filter_signal_flag` 값은 제 2 값 일 수 있다.
- [1132] 한편 `slice_alf_aps_id_chroma[i]`가 비트스트림 내에 존재하지 않을 경우 `slice_alf_aps_id_chroma[i]`는 제 1 값 또는 `pic_alf_aps_id_chroma[i]`으로 추론될 수 있다.
- [1133] 일 예로, `slice_alf_aps_id_chroma[i]`와 동일한 `adaptation_parameter_set_id`를 가지는 적응적 루프내 필터링에 대한 적응 파라미터 세트의 시간적 계층 식별자는 현재 슬라이스의 시간적 계층 식별자보다 작거나 같을 수 있다.
- [1134] 다른 예로, 하나의 픽처 내, 동일한 `adaptation_parameter_set_id`를 가지는 2개 이상의 서브 픽처/슬라이스/타일 그룹/타일/브릭의 적응 파라미터 세트가 존재하는 경우, 동일한 `adaptation_parameter_set_id`를 가지는 2개 이상의 적응적 루프내 필터링에 대한 적응 파라미터 세트는 동일한 데이터/콘텐츠/구문 요소 값을 가질 수 있다.
- [1135] 또 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우, `slice_alf_aps_id_chroma[i]`는 화면내 슬라이스를 포함하는 픽처 또는 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1136] 또 다른 예로, 슬라이스 헤더는 구문 요소 `slice_cross_component_alf_cb_enabled_flag`(또는 `sh_alf_cc_cb_enabled_flag`) 및/또는 `slice_cross_component_alf_cb_aps_id`(또는 `sh_alf_cc_cb_aps_id`)를 포함할 수 있다.
- [1137] 예컨대, `slice_cross_component_alf_cb_enabled_flag`가 제 1 값인 경우, 성분 간 Cb 필터가 Cb 색 성분에 적용되지 않을 수 있다.
`slice_cross_component_alf_cb_enabled_flag`가 제 2 값인 경우, 성분 간 Cb 필터가 Cb 색 성분에 적용될 수 있다.
- [1138] 한편, `slice_cross_component_alf_cb_enabled_flag`가 비트스트림 내에 존재하지 않을 경우 `slice_cross_component_alf_cb_enabled_flag`는 제 1 값 혹은 `pic_cross_component_alf_cb_enabled_flag`으로 추론될 수 있다.
- [1139] `slice_cross_component_alf_cb_aps_id`는 슬라이스의 Cb 색 성분이 참조하는 `adaptation_parameter_set_id`를 지시할 수 있다.
- [1140] 일 예로, `aps_params_type`가 `ALF_APS`이며 `adaptation_parameter_set_id`가 `slice_cross_component_alf_cb_aps_id`와 동일한 APS NAL 유닛의 `TemporalId`는 현재 슬라이스의 NAL 유닛의 `TemporalId`보다 작거나 같을 수 있다.
- [1141] 한편, `slice_cross_component_alf_cb_aps_id`가 비트스트림 내에 존재하지 않을 경우 `slice_cross_component_alf_cb_aps_id`는 제 1 값 또는 `pic_cross_component_alf_cb_aps_id`으로 추론될 수 있다.
- [1142] 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우,

slice_cross_component_alf_cb_aps_id는 화면내 슬라이스를 포함하는 픽처 혹은 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.

- [1143] 또 다른 예로, slice_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, 현재 픽처 내 모든 슬라이스, 모든 타일, 혹은 모든 서브픽처에서 slice_cross_component_alf_cb_aps_id에 의해 참조되는 ALF APS는 모두 같을 수 있다. 한편, slice_cross_component_cb_filters_signalled_minus1에 +1한 값은 성분 간 Cb 필터의 개수를 지시할 수 있다. 이때, slice_cross_component_cb_filters_signalled_minus1은 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있으며, 예컨대 3일 수 있다.
- [1144] 한편, slice_cross_component_cb_filters_signalled_minus1가 비트스트림 내에 존재하지 않을 경우 slice_cross_component_cb_filters_signalled_minus1는 제1 값 또는 pic_cross_component_cb_filters_signalled_minus1으로 추론될 수 있다.
- [1145] 또 다른 예로, slice_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, slice_cross_component_cb_filters_signalled_minus1에 대한 별도의 시그널링 없이 slice_cross_component_cb_filters_signalled_minus1는 현재 슬라이스에서 참조하는 slice_cross_component_alf_cb_aps_id의 ALF APS 내의 alf_cross_component_cb_filters_signalled_minus1 값으로 설정될 수 있다.
- [1146] 또 다른 예로, slice_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, slice_cross_component_cb_filters_signalled_minus1는 현재 슬라이스에서 참조하는 slice_cross_component_alf_cb_aps_id의 ALF APS 내의 alf_cross_component_cb_filters_signalled_minus1 값보다 작거나 같을 수 있다.
- [1147] 또 다른 예로, slice_cross_component_alf_cb_enabled_flag가 제 2 값인 경우, slice_cross_component_cb_filters_signalled_minus1는 현재 슬라이스에서 참조하는 픽처 헤더의 pic_cross_component_cb_filters_signalled_minus1 값보다 작거나 같을 수 있다.
- [1148] 또 다른 예로, 슬라이스 헤더는 구문 요소 slice_cross_component_alf_cr_enabled_flag(또는 sh_alf_cc_cr_enabled_flag) 및/또는 slice_cross_component_alf_cr_aps_id(또는 sh_alf_cc_cr_aps_id)를 포함할 수 있다.
- [1149] 예컨대, slice_cross_component_alf_cr_enabled_flag가 제 1 값인 경우, 성분 간 Cr 필터가 Cr 색 성분에 적용되지 않을 수 있다. slice_cross_component_alf_cr_enabled_flag가 제 2 값인 경우, 성분 간 Cr 필터가 Cr 색 성분에 적용될 수 있다.
- [1150] 한편, slice_cross_component_alf_cr_enabled_flag가 비트스트림 내에 존재하지 않을 경우 slice_cross_component_alf_cr_enabled_flag는 제 1 값 혹은 pic_cross_component_alf_cr_enabled_flag으로 추론될 수 있다.
- [1151] slice_cross_component_alf_cr_aps_id는 슬라이스의 Cr 색 성분이 참조하는

- adaptation_parameter_set_id를 지시할 수 있다.
- [1152] 일 예로, aps_params_type가 ALF_APS이며 adaptation_parameter_set_id가 slice_cross_component_alf_cr_aps_id와 동일한 APS NAL 유닛의 TemporalId는 현재 슬라이스의 NAL 유닛의 TemporalId보다 작거나 같을 수 있다.
- [1153] 한편, slice_cross_component_alf_cr_aps_id가 비트스트림 내에 존재하지 않을 경우 slice_cross_component_alf_cr_aps_id는 제 1 값 또는 pic_cross_component_alf_cr_aps_id으로 추론될 수 있다.
- [1154] 다른 예로, 현재 슬라이스가 화면내 슬라이스인 경우, slice_cross_component_alf_cr_aps_id는 화면내 슬라이스를 포함하는 픽처 혹은 화면내 픽처가 아닌 다른 픽처에 관한 적응적 루프내 필터링에 대한 적응 파라미터 세트를 참조하지 않을 수 있다.
- [1155] 또 다른 예로, slice_cross_component_alf_cr_enabled_flag가 제 2 값인 경우, 현재 픽처 내 모든 슬라이스, 모든 타일, 혹은 모든 서브픽처에서 slice_cross_component_alf_cr_aps_id에 의해 참조되는 ALF APS는 모두 같을 수 있다. 한편, slice_cross_component_cr_filters_signalled_minus1에 +1한 값은 성분 간 Cr 필터의 개수를 지시할 수 있다. 이때, slice_cross_component_cr_filters_signalled_minus1은 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있으며, 예컨대 3일 수 있다.
- [1156] 한편, slice_cross_component_cr_filters_signalled_minus1가 비트스트림 내에 존재하지 않을 경우 slice_cross_component_cr_filters_signalled_minus1는 제 1 값 또는 pic_cross_component_cr_filters_signalled_minus1으로 추론될 수 있다.
- [1157] 또 다른 예로, slice_cross_component_alf_cr_enabled_flag가 제 2 값인 경우, slice_cross_component_cr_filters_signalled_minus1에 대한 별도의 시그널링 없이 slice_cross_component_cr_filters_signalled_minus1는 현재 슬라이스에서 참조하는 slice_cross_component_alf_cr_aps_id의 ALF APS 내의 alf_cross_component_cr_filters_signalled_minus1 값으로 설정될 수 있다.
- [1158] 또 다른 예로, slice_cross_component_alf_cr_enabled_flag가 제 2 값인 경우, slice_cross_component_cr_filters_signalled_minus1는 현재 슬라이스에서 참조하는 slice_cross_component_alf_cr_aps_id의 ALF APS 내의 alf_cross_component_cr_filters_signalled_minus1 값보다 작거나 같을 수 있다.
- [1159] 또 다른 예로, slice_cross_component_alf_cr_enabled_flag가 제 2 값인 경우, slice_cross_component_cr_filters_signalled_minus1는 현재 슬라이스에서 참조하는 픽처 헤더의 pic_cross_component_cr_filters_signalled_minus1 값보다 작거나 같을 수 있다.
- [1160] 슬라이스 단위 구문 요소 중 적어도 하나는 픽처 단위 구문 요소로 대체되어 시그널링될 수 있다. 즉, 슬라이스 헤더에서 시그널링되는 슬라이스 단위 구문 요소 중 적어도 하나는 픽처 헤더에서 시그널링될 수 있다.
- [1161] 또한, 슬라이스 단위 구문 요소 중 적어도 하나와 픽처 단위 구문 요소 중

적어도 하나는 중복하여 시그널링될 수 있다. 즉, 적어도 하나의 구문 요소가 픽처 헤더에서 시그널링되더라도 시그널링된 구문 요소와 같은 역할을 하는 구문 요소가 슬라이스 헤더에서도 시그널링될 수 있다.

[1162] 도 47은 본 개시의 또 다른 일 실시예에 따른 ALF 데이터(data)를 설명하기 위한 도면이다.

[1163] 일 예로, ALF 데이터는 구문 요소 `alf_luma_filter_signal_flag`를 포함할 수 있다. `alf_luma_filter_signal_flag`는 휘도 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화되는지 여부를 지시할 수 있다. 예컨대, `alf_luma_filter_signal_flag`가 제 1 값인 경우, 휘도 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화되지 않을 수 있고, `alf_luma_filter_signal_flag`가 제 2 값인 경우 휘도 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화될 수 있다.

[1164] 한편, `alf_luma_filter_signal_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_luma_filter_signal_flag`는 제 1 값으로 추론될 수 있다.

[1165] 다른 예로, ALF 데이터는 구문 요소 `alf_chroma_filter_signal_flag`를 포함할 수 있다. `alf_chroma_filter_signal_flag`는 색차 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화되는지 여부를 지시할 수 있다. 예컨대, `alf_chroma_filter_signal_flag`가 제 1 값인 경우 색차 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화되지 않을 수 있고, `alf_chroma_filter_signal_flag`가 제 2 값인 경우 색차 성분에 대한 적응적 루프내 필터 세트가 엔트로피 부호화/복호화될 수 있다.

[1166] 한편, `ChromaArrayType`가 제 1 값인 경우, `alf_chroma_filter_signal_flag`은 제 1 값일 수 있다. 또한, `alf_luma_filter_signal_flag` 값과 `alf_chroma_filter_signal_flag` 값은 둘 다 동시에 0의 값을 가지지 않을 수 있다.

[1167] 한편, `alf_chroma_filter_signal_flag`가 비트스트림 내에 존재하지 않을 경우 `alf_chroma_filter_signal_flag`는 제 1 값으로 추론될 수 있다.

[1168] 또 다른 예로, 적응적 루프내 필터 세트에 포함되는 서로 다른 적응적 루프내 필터의 개수의 최대값인 `NumAlfFilters`은 N일 수 있다. 이때, N은 양의 정수일 수 있으며, 예컨대 25일 수 있다.

[1169] 또 다른 예로, ALF 데이터는 구문 요소 `alf_cross_component_cb_filter_signal_flag`(또는 `alf_cc_cb_filter_signal_flag`) 및/또는 `alf_cross_component_cr_filter_signal_flag`(또는 `alf_cc_cr_filter_signal_flag`)를 포함할 수 있다.

[1170] `alf_cross_component_cb_filter_signal_flag`는 성분 간 Cb 필터가 시그널링되는지 여부를 지시할 수 있다. 예컨대, `alf_cross_component_cb_filter_signal_flag`가 제 1 값인 경우, 성분 간 Cb 필터가 시그널링되지 않을 수 있다. `alf_cross_component_cb_filter_signal_flag`가 제 2 값인 경우, 성분 간 Cb 필터가 시그널링될 수 있다.

- [1171] 한편, ChromaArrayType가 제 1 값인 경우,
alf_cross_component_cb_filter_signal_flag는 제 1 값일 수 있다.
- [1172] alf_cross_component_cr_filter_signal_flag는 성분 간 Cr 필터가
시그널링되는지를 의미할 수 있다. 예컨대,
alf_cross_component_cr_filter_signal_flag가 제 1 값인 경우, 성분 간 Cr 필터가
시그널링되지 않을 수 있다. alf_cross_component_cr_filter_signal_flag가 제 2 값인
경우, 성분 간 Cr 필터가 시그널링될 수 있다.
- [1173] 한편, ChromaArrayType가 제 1 값인 경우,
alf_cross_component_cr_filter_signal_flag는 제 1 값일 수 있다.
- [1174] 또 다른 예로, ALF 데이터는 구문 요소 alf_luma_clip_flag를 포함할 수 있다.
alf_luma_clip_flag는 휘도 성분에 대해 선형(linear) 적응적 루프내 필터링이
수행되는지 혹은 비선형(non-linear) 적응적 루프내 필터링이 수행되는지를
지시할 수 있다. 예컨대, alf_luma_clip_flag가 제 1 값인 경우, 휘도 성분에 대해
선형 적응적 루프내 필터링이 수행될 수 있고, alf_luma_clip_flag가 제 2 값인
경우, 휘도 성분에 대해 비선형 적응적 루프내 필터링이 수행될 수 있다.
- [1175] 한편, alf_luma_clip_flag가 비트스트림 내에 존재하지 않을 경우
alf_luma_clip_flag는 제 1 값으로 추론될 수 있다.
- [1176] 또 다른 예로, ALF 데이터는 구문 요소 alf_luma_num_filters_signalled_minus1를
포함할 수 있다. alf_luma_num_filters_signalled_minus1에 +1한 값은 휘도 성분에
대한 적응적 루프내 필터 클래스(종류)의 개수를 의미할 수 있다. 예컨대,
alf_luma_num_filters_signalled_minus1는 0에서 NumAlfFilters - N까지의 값을
가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 1일 수 있다.
- [1177] 한편, alf_luma_num_filters_signalled_minus1가 비트스트림 내에 존재하지 않을
경우 alf_luma_num_filters_signalled_minus1는 0의 값으로 추론될 수 있다.
- [1178] 또 다른 예로, ALF 데이터는 구문 요소 alf_luma_coeff_delta_idx를 포함할 수
있다. alf_luma_coeff_delta_idx[filtIdx]는 filtIdx에 의해 지시되는 휘도 성분에
대한 적응적 루프내 필터 계수의 색인을 의미할 수 있다.
- [1179] 예컨대, filtIdx는 0에서 NumAlfFilters - N까지의 값을 가질 수 있다. 이때, N은
양의 정수일 수 있고, 예컨대 1일 수 있다.
- [1180] 한편, alf_luma_coeff_delta_idx[filtIdx]가 비트스트림 내에 존재하지 않을 경우
alf_luma_coeff_delta_idx[filtIdx]는 0의 값으로 추론될 수 있다.
- [1181] 또한, alf_luma_coeff_delta_idx[filtIdx]의 길이는 $\text{Ceil}(\text{Log}_2(\text{alf_luma_num_filters_signalled_minus1} + 1))$ 비트로 표현될 수 있다. 여기서, $\text{Ceil}(x)$ 는 x 보다 작거나 같은 가장 작은 정수 값을 반환하는 연산자일 수 있다.
- [1182] 또 다른 예로, ALF 데이터는 구문 요소 alf_luma_coeff_abs를 포함할 수 있다.
alf_luma_coeff_abs[sfIdx][j]는 sfIdx에 의해 지시되는 휘도 성분에 대한
적응적 루프내 필터의 j번째 계수의 절대값을 의미할 수 있다.
- [1183] 한편, alf_luma_coeff_abs[sfIdx][j]가 비트스트림 내에 존재하지 않을 경우

alf_luma_coeff_abs[sfIdx][j]는 0의 값으로 추론될 수 있다.

[1184] 또한, alf_luma_coeff_abs[sfIdx][j]에 대한 지수-콜롬 이진화 uek(v)의 차수 k는 N일 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 3일 수 있다.

[1185] 또 다른 예로, ALF 데이터는 구문 요소 alf_luma_coeff_sign를 포함할 수 있다. alf_luma_coeff_sign[sfIdx][j]는 sfIdx에 의해 지시되는 휘도 성분 에 대한 적응적 루프내 필터의 j번째 계수의 부호(sign)를 의미할 수 있다. 예컨대, alf_luma_coeff_sign[sfIdx][j]가 제 1 값인 경우, 대응하는 휘도 성분 에 대한 적응적 루프내 필터 계수는 양의 부호(positive sign)를 가질 수 있다. 반면, alf_luma_coeff_sign[sfIdx][j]가 제 2 값인 경우, 대응하는 휘도 성분 에 대한 적응적 루프내 필터 계수는 음의 부호(negative sign)를 가질 수 있다.

[1186] 한편, alf_luma_coeff_sign[sfIdx][j]가 비트스트림 내에 존재하지 않을 경우 alf_luma_coeff_sign[sfIdx][j]는 제1 값으로 추론될 수 있다.

[1187] sfIdx에 의해 지시되는 필터 계수 filtCoeff[sfIdx][j]는 다음의 수학적 식 24에 따라 유도될 수 있다. 이때, sfIdx는 0에서 alf_luma_num_filters_signalled_minus1까지의 값을 가질 수 있다. 또한, j는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예컨대 11일 수 있다.

[1188] [수학적 식 24]

[1189]
$$\text{filtCoeff}[\text{sfIdx}][j] = \text{alf_luma_coeff_abs}[\text{sfIdx}][j] * (1 - 2 * \text{alf_luma_coeff_sign}[\text{sfIdx}][j])$$

[1190] 휘도 성분 에 대한 적응적 루프내 필터 계수

AlfCoeff_L[adaptation_parameter_set_id][filtIdx][j]는 다음의 수학적 식 25에 따라 유도될 수 있다. 이때, filtIdx는 0에서 NumAlfFilters - 1까지의 값을 가질 수 있다. 또한, j는 0에서 N까지의 값을 가질 수 있다. 여기서, N은 양의 정수일 수 있고, 예컨대 11일 수 있다.

[1191] [수학적 식 25]

[1192]
$$\text{AlfCoeff_L}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j] = \text{filtCoeff}[\text{alf_luma_coeff_delta_idx}[\text{filtIdx}]][j]$$

[1193] 한편, AlfCoeff_L[adaptation_parameter_set_id][filtIdx][j]는 -2^M 에서 $2^M - 1$ 까지의 값을 가질 수 있다. 이때, filtIdx는 0에서 NumAlfFilters - 1까지의 값을 가질 수 있다. 또한, j는 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대, 11일 수 있다. 또한, M은 양의 정수일 수 있고, 예컨대 7일 수 있다.

[1194] 또 다른 예로, ALF 데이터는 구문 요소 alf_luma_clip_idx를 포함할 수 있다. alf_luma_clip_idx[sfIdx][j]는 sfIdx에 의해 지시되는 휘도 성분 에 대한 적응적 루프내 필터의 j번째 계수가 복원/복호 샘플에 곱해지기 전에 절삭에 사용되는 절삭 값(clipping value)에 대한 절삭 색인을 의미할 수 있다.

[1195] 한편, alf_luma_clip_idx[sfIdx][j]가 비트스트림 내에 존재하지 않을 경우

alf_luma_clip_idx[sfIdx][j]는 제 1 값으로 추론될 수 있다.

[1196] 예컨대, alf_luma_clip_idx[sfIdx][j]는 0에서 M까지의 값을 가질 수 있다. 이때, M은 양의 정수일 수 있고, 예컨대 3일 수 있다. 이때, sfIdx는 0에서 alf_luma_num_filters_signalled_minus1까지의 값을 가질 수 있다. 또한, j는 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대, 11일 수 있다.

[1197] 휘도 성분에 대한 적응적 루프내 필터 절삭 값

AlfClip_L[adaptation_parameter_set_id][filtIdx][j]은 다음의 표 3에 따라 유도될 수 있다. 이때, filtIdx는 0에서 NumAlfFilters - 1까지의 값을 가질 수 있다. 또한, j는 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 11일 수 있다. 표 3에서 BitDepth는 입력 샘플의 비트 심도를 의미하며, clipIdx는 alf_luma_clip_idx[alf_luma_coeff_delta_idx[filtIdx]][j]로 설정될 수 있다.

[1198] [표 3]

[1199]

BitDepth	clipIdx			
	0	1	2	3
8	255	50	10	2
9	511	100	20	4
10	1023	201	39	8
11	2047	402	79	15
12	4095	803	158	31
13	8191	1607	315	62
14	16383	3214	630	124
15	32767	6427	1261	247
16	65535	12855	2521	495

[1200] 또 다른 예로, ALF 데이터는 구문 요소 alf_chroma_num_alt_filters_minus1을 포함할 수 있다. alf_chroma_num_alt_filters_minus1에 +1한 값은 색차 성분에 대한 필터 개수를 의미할 수 있다. 예컨대, alf_chroma_num_alt_filters_minus1는 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 7, 8, 3, 또는 4일 수 있다.

[1201] 한편, alf_chroma_num_alt_filters_minus1가 비트스트림 내에 존재하지 않을 경우 alf_chroma_num_alt_filters_minus1는 0의 값으로 추론될 수 있다.

[1202] 또 다른 예로, ALF 데이터는 구문 요소 alf_chroma_clip_flag를 포함할 수 있다. alf_chroma_clip_flag는 색차 성분에 대해 선형 적응적 루프내 필터링이 수행되는지 또는 비선형 적응적 루프내 필터링이 수행되는지를 지시할 수 있다. 예컨대, alf_chroma_clip_flag가 제 1 값인 경우 색차 성분에 대해 선형 적응적 루프내 필터링이 수행될 수 있고, alf_chroma_clip_flag가 제 2 값인 경우 색차 성분에 대해 비선형 적응적 루프내 필터링이 수행될 수 있다.

[1203] 한편, alf_chroma_clip_flag가 비트스트림 내에 존재하지 않을 경우 alf_chroma_clip_flag는 제 1 값으로 추론될 수 있다.

- [1204] 한편, `alf_chroma_clip_flag`가 ALF 필터 별로 시그널링되는 경우, `alf_chroma_clip_flag`는 `alf_chroma_clip_flag[altIdx]`와 같은 행렬 형태로 시그널링될 수 있다. 반면, `alf_chroma_clip_flag`가 각 ALF 데이터 별로 1개씩 시그널링되는 경우, `alf_chroma_clip_flag`는 단일 구문 요소로 시그널링될 수 있다.
- [1205] 또 다른 예로, ALF 데이터는 구문 요소 `alf_chroma_coeff_abs`를 포함할 수 있다. `alf_chroma_coeff_abs[altIdx][j]`는 `altIdx`에 의해 지시되는 색차 성분에 대한 적응적 루프내 필터의 j 번째 계수의 절대값을 의미할 수 있다.
- [1206] 예컨대, 상기 `alf_chroma_coeff_abs[altIdx][j]`는 0에서 $2^M - 1$ 까지의 값을 가질 수 있다. 이때, M 은 양의 정수일 수 있고, 예컨대 7일 수 있다. 또한, `alf_chroma_coeff_abs[altIdx][j]`에 대한 지수-콜롬 이진화 `uek(v)`의 차수 k 는 N 일 수 있다. 이때, N 은 양의 정수일 수 있고, 예컨대, 3일 수 있다.
- [1207] 한편, `alf_chroma_coeff_abs[altIdx][j]`가 비트스트림 내에 존재하지 않을 경우 `alf_chroma_coeff_abs[altIdx][j]`는 0의 값으로 추론될 수 있다.
- [1208] 또 다른 예로, ALF 데이터는 구문 요소 `alf_chroma_coeff_sign`를 포함할 수 있다. `alf_chroma_coeff_sign[altIdx][j]`는 `altIdx`에 의해 지시되는 색차 성분에 대한 적응적 루프내 필터의 j 번째 계수의 부호를 의미할 수 있다. 예컨대, `alf_chroma_coeff_sign[altIdx][j]`가 제 1 값인 경우, 대응하는 색차 성분에 대한 적응적 루프내 필터 계수는 양의 부호를 가질 수 있다. 반면, `alf_chroma_coeff_sign[altIdx][j]`가 제 2 값인 경우, 대응하는 색차 성분에 대한 적응적 루프내 필터 계수는 음의 부호를 가질 수 있다.
- [1209] 한편, `alf_chroma_coeff_sign[altIdx][j]`가 비트스트림 내에 존재하지 않을 경우 `alf_chroma_coeff_sign[altIdx][j]`는 제1 값(예: 0)으로 추론될 수 있다.
- [1210] 일 예로, 색차 성분에 대한 적응적 루프내 필터 계수 `AlfCoeff_C[adaptation_parameter_set_id][altIdx][j]`는 다음의 수학적 식 26에 따라 유도될 수 있다. 이때, j 는 0에서 N 까지의 값을 가질 수 있다. 또한, N 은 양의 정수일 수 있고, 예컨대 5일 수 있다.
- [1211] [수학적 식 26]
- [1212]
$$\text{AlfCoeff_C}[\text{adaptation_parameter_set_id}][\text{altIdx}][j] = \text{alf_chroma_coeff_abs}[\text{altIdx}][j] * (1 - 2 * \text{alf_chroma_coeff_sign}[\text{altIdx}][j])$$
- [1213] `AlfCoeff_C[adaptation_parameter_set_id][altIdx][j]`는 -2^M 에서 $2^M - 1$ 까지의 값을 가질 수 있다. 이때, j 는 0에서 N 까지의 값을 가질 수 있다. 이때, N 은 양의 정수일 수 있고, 예컨대 5일 수 있다. 또한, M 은 양의 정수일 수 있고, 예컨대 7일 수 있다.
- [1214] 또 다른 예로, ALF 데이터는 구문 요소 `alf_chroma_clip_idx`를 포함할 수 있다. `alf_chroma_clip_idx[altIdx][j]`는 `altIdx`에 의해 지시되는 색차 성분에 대한 적응적 루프내 필터의 j 번째 계수가 복원/복호 샘플에 곱해지기 전에 절삭에 사용되는 절삭 값에 대한 절삭 색인을 의미할 수 있다.

- [1215] 한편, $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ 가 비트스트림 내에 존재하지 않을 경우 $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ 는 제 1 값으로 추론될 수 있다.
- [1216] 예컨대, $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ 는 0에서 M 까지의 값을 가질 수 있다. 이때, M 은 양의 정수일 수 있고, 예컨대 3일 수 있다. 또한, j 는 0에서 N 까지의 값을 가질 수 있다. 또한, N 은 양의 정수일 수 있고, 예컨대 5일 수 있다.
- [1217] 색차 성분에 대한 적응적 루프내 필터 절삭 값
 $\text{AlfClip_C}[\text{adaptation_parameter_set_id}][\text{altIdx}][j]$ 은 위에서 도시한 표 3에 따라 유도될 수 있다. 이때, j 는 0에서 N 까지의 값을 가질 수 있다. 또한, N 은 양의 정수일 수 있고, 예컨대 5일 수 있다.
- [1218] 표 3에서 BitDepth 는 입력 샘플의 비트 심도를 의미하며, clipIdx 는 $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ 로 설정될 수 있다.
- [1219] 또 다른 예로, ALF 데이터는 구문 요소 $\text{alf_cross_component_cb_filters_signalled_minus1}$ (또는 $\text{alf_cc_cb_filters_signalled_minus1}$)를 포함할 수 있다.
 $\text{alf_cross_component_cb_filters_signalled_minus1}$ 에 +1한 값은 현재 ALF APS에서 시그널링되는 성분 간 Cb 필터의 개수를 의미할 수 있다. 예컨대, $\text{alf_cross_component_cb_filters_signalled_minus1}$ 는 0에서 M 까지의 값을 가질 수 있다. 이때, M 은 양의 정수일 수 있고, 예컨대 3일 수 있다. 또한, 연산 복잡도를 감소시키기 위해 상기 M 은 3 미만의 정수일 수 있다.
- [1220] 또 다른 예로, ALF 데이터는 구문 요소 $\text{alf_cross_component_cb_coeff_plus32}$ 를 포함할 수 있다. $\text{alf_cross_component_cb_coeff_plus32}[k][j]$ 에 -32한 값은 시그널링되는 k 번째에서 성분 간 Cb 필터의 j 번째 필터 계수의 값을 의미할 수 있다.
- [1221] 한편, $\text{alf_cross_component_cb_coeff_plus32}[k][j]$ 가 비트스트림 내에 존재하지 않을 경우 $\text{alf_cross_component_cb_coeff_plus32}[k][j]$ 는 32의 값으로 추론될 수 있다.
- [1222] 한편, $\text{alf_cross_component_cb_coeff_plus32}[k][j]$ 에 대한 지수-골롬 이진화 $\text{uek}(v)$ 의 차수 k 는 N 일 수 있다. 이때, N 은 양의 정수일 수 있고, 예컨대 3일 수 있다.
- [1223] k 번째 성분 간 Cb 필터 계수
 $\text{CcAlfApsCoeff_Cb}[\text{adaptation_parameter_set_id}][k][j]$ 는 다음의 수학적 식 27에 따라 유도될 수 있다.
- [1224] [수학적 식 27]
- [1225] $\text{CcAlfApsCoeff_Cb}[\text{adaptation_parameter_set_id}][k][j] = \text{alf_cross_component_cb_coeff_plus32}[k][j] - 32$
- [1226] 이때, 상기 $\text{CcAlfApsCoeff_Cb}[\text{adaptation_parameter_set_id}][k][j]$ 는 -2^M 에서 $2^M - 1$ 까지의 값을 가질 수 있다. 이때, j 는 0에서 N 까지의 값을 가질 수 있다. 또한, N 은 양의 정수일 수 있고, 예컨대 7일 수 있다. 또한, M 은 양의

정수일 수 있고, 예컨대 5일 수 있다. 또한, 연산 복잡도를 감소시키기 위해 상기 M은 5 미만의 정수일 수 있다.

[1227] 또 다른 예로, ALF 데이터는 구문 요소

`alf_cross_component_cr_filters_signalled_minus1`(또는 `alf_cc_cr_filters_signalled_minus1`)를 포함할 수 있다.

`alf_cross_component_cr_filters_signalled_minus1`에 +1한 값은 현재 ALF APS에서 시그널링되는 성분 간 Cr 필터의 개수를 의미할 수 있다. 예컨대, `alf_cross_component_cr_filters_signalled_minus1`는 0에서 M까지의 값을 가질 수 있다. 이때, M은 양의 정수일 수 있고, 예컨대 3일 수 있다. 또한, 연산 복잡도를 감소시키기 위해 상기 M은 3 미만의 정수일 수 있다.

[1228] 또 다른 예로, ALF 데이터는 구문 요소 `alf_cross_component_cr_coeff_plus32`를 포함할 수 있다. `alf_cross_component_cr_coeff_plus32[ k ][ j ]`에 -32한 값은 시그널링되는 k번째에서 성분 간 Cr 필터의 j번째 필터 계수의 값을 의미할 수 있다.

[1229] 한편, `alf_cross_component_cr_coeff_plus32[ k ][ j ]`가 비트스트림 내에 존재하지 않을 경우 `alf_cross_component_cr_coeff_plus32[ k ][ j ]`는 32의 값으로 추론될 수 있다.

[1230] 한편, `alf_cross_component_cr_coeff_plus32[ k ][ j ]`에 대한 지수-콜롬 이진화 $\text{uek}(v)$ 의 차수 k는 N일 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 3일 수 있다.

[1231] k번째 성분 간 Cr 필터 계수

`CcAlfApsCoeff_Cr[ adaptation_parameter_set_id ][ k ][ j ]`는 다음의 수학적 식 28에 따라 유도될 수 있다.

[1232] [수학적 식 28]

[1233]
$$\text{CcAlfApsCoeff_Cr}[\text{adaptation_parameter_set_id}][k][j] = \text{alf_cross_component_cr_coeff_plus32}[k][j] - 32$$

[1234] 이때, 상기 `CcAlfApsCoeff_Cr[ adaptation_parameter_set_id ][ k ][ j ]`는 -2^M 에서 $2^M - 1$ 까지의 값을 가질 수 있다. 이때, j는 0에서 N까지의 값을 가질 수 있다. 또한, N은 양의 정수일 수 있고, 예컨대 7일 수 있다. 또한, M은 양의 정수일 수 있고, 예컨대 5일 수 있다. 또한, 연산 복잡도를 감소시키기 위해 상기 M은 5 미만의 정수일 수 있다.

[1235] 상술한 실시예들에서 정수 32 대신 N이 사용될 수도 있다. 이때, N은 양의 정수일 수 있다. 여기서, N은 필터 계수의 기본 값을 의미할 수 있다.

[1236] 도 48은 본 개시의 또 다른 일 실시예에 따른 CTU(Coding Tree Unit)를 설명하기 위한 도면이다.

[1237] 일 예로, CTU는 구문 요소 `alf_ctb_flag`를 포함할 수 있다.

`alf_ctb_flag[ cIdx ][ CtbAddrX ][ CtbAddrY ]`는 휘도 성분 위치

(`xCtb`, `yCtb`)에서 `cIdx`에 의해 지시되는 색 성분(Y, Cb, 혹은 Cr)의 부호화 트리

블록에 적응적 루프내 필터링이 적용되는지 여부를 지시할 수 있다.

- [1238] 예컨대, $\text{alf_ctb_flag}[cIdx][CtbAddrX][CtbAddrY]$ 가 제 1 값인 경우 휘도 성분 위치 (x_{Ctb} , y_{Ctb})에서 $cIdx$ 에 의해 지시되는 색 성분의 부호화 트리 블록에 적응적 루프내 필터링이 적용되지 않을 수 있다. 반면, $\text{alf_ctb_flag}[cIdx][CtbAddrX][CtbAddrY]$ 가 제 2 값인 경우 휘도 성분 위치 (x_{Ctb} , y_{Ctb})에서 $cIdx$ 에 의해 지시되는 색 성분의 부호화 트리 블록에 적응적 루프내 필터링이 적용될 수 있다.
- [1239] 한편, $\text{alf_ctb_flag}[cIdx][CtbAddrX][CtbAddrY]$ 가 비트스트림 내에 존재하지 않을 경우 $\text{alf_ctb_flag}[cIdx][CtbAddrX][CtbAddrY]$ 는 제 1 값으로 추론될 수 있다.
- [1240] 다른 예로, CTU는 구문 요소 alf_use_aps_flag 를 포함할 수 있다. alf_use_aps_flag 는 휘도 성분 부호화 트리 블록에서 고정된 필터 중 적어도 하나가 사용되는지를 의미할 수 있다. 예컨대, alf_use_aps_flag 가 제 1 값인 경우 휘도 성분 부호화 트리 블록에서 고정된 필터 중 적어도 하나가 사용될 수 있다. 반면, alf_use_aps_flag 가 제 2 값인 경우 휘도 성분 부호화 트리 블록에서 적어도 하나의 적응적 루프내 필터 정보가 사용될 수 있다.
- [1241] 한편, alf_use_aps_flag 가 비트스트림 내에 존재하지 않을 경우 alf_use_aps_flag 는 제 1 값(예: 0)으로 추론될 수 있다.
- [1242] 또 다른 예로, CTU는 구문 요소 $\text{alf_luma_prev_filter_idx}$ 를 포함할 수 있다. $\text{alf_luma_prev_filter_idx}$ 는 이전 픽처들 중 적어도 하나의 픽처/서브 픽처/슬라이스/타일 그룹에서 사용한 적응적 루프내 필터 정보 중 어떤 적응적 루프내 필터 정보를 사용하는지에 대한 색인을 의미할 수 있다. 또는, $\text{alf_luma_prev_filter_idx}$ 는 이전 필터들 중 어떤 필터를 현재 부호화 트리 블록에 사용할지에 대한 색인을 지시할 수 있다.
- [1243] 예컨대, $\text{alf_luma_prev_filter_idx}$ 는 0에서 $\text{slice_num_alf_aps_ids_luma} - N$ 또는 $\text{pic_num_alf_aps_ids_luma} - N$ 까지의 값을 가질 수 있다. 이때, N 은 양의 정수일 수 있고, 예컨대 1일 수 있다.
- [1244] 한편, $\text{alf_luma_prev_filter_idx}$ 가 비트스트림 내에 존재하지 않을 경우 $\text{alf_luma_prev_filter_idx}$ 는 0의 값으로 추론될 수 있다.
- [1245] 휘도 성분 위치 (x_{Ctb} , y_{Ctb})에서 휘도 성분 부호화 트리 블록에서 사용하는 적응적 루프내 필터 색인 $\text{AlfCtbFiltSetIdxY}[CtbAddrX][CtbAddrY]$ 은, alf_use_aps_flag 에 기초하여 유도될 수 있다. 예컨대, alf_use_aps_flag 가 제 1 값인 경우, $\text{AlfCtbFiltSetIdxY}[CtbAddrX][CtbAddrY]$ 는 $\text{alf_luma_fixed_filter_idx}$ 로 결정될 수 있다. 반면, alf_use_aps_flag 가 제 2 값인 경우 $\text{AlfCtbFiltSetIdxY}[CtbAddrX][CtbAddrY]$ 는 $16 + \text{alf_luma_prev_filter_idx}$ 로 결정될 수 있다.
- [1246] 또 다른 예로, CTU는 구문 요소 $\text{alf_luma_fixed_filter_idx}$ 를 포함할 수 있다. $\text{alf_luma_fixed_filter_idx}$ 는 휘도 성분 부호화 트리 블록에서 사용하는 고정된

필터 색인을 지시할 수 있다. 예컨대, `alf_luma_fixed_filter_idx`는 0에서 N까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 15일 수 있다.

[1247] 한편, 현재 블록의 색 성분이 Cb인 경우, N은

`alf_cross_component_cb_filters_signalled_minus1` 보다 작거나 같을 수 있다. 반면, 현재 블록의 색 성분이 Cr인 경우, N은

`alf_cross_component_cr_filters_signalled_minus1` 보다 작거나 같을 수 있다.

[1248] 한편, `alf_luma_fixed_filter_idx`가 비트스트림 내에 존재하지 않을 경우

`alf_luma_fixed_filter_idx`는 0의 값으로 추론될 수 있다.

[1249] 또 다른 예로, CTU는 구문 요소 `alf_ctb_filter_alt_idx`를 포함할 수 있다.

`alf_ctb_filter_alt_idx[ chromaIdx ][ CtbAddrX ][ CtbAddrY ]`는 현재 부호화 트리 블록에서 사용되는 색차 성분 필터의 색인을 지시할 수 있다. 이때,

`chromaIdx`가 0인 경우 Cb 성분을 의미할 수 있고, 1인 경우 Cr 성분을 의미할 수 있다. 예컨대, `alf_ctb_filter_alt_idx[ chromaIdx ][ CtbAddrX ][ CtbAddrY ]`는

0에서 N 혹은 `alf_chroma_num_alt_filters_minus1`까지의 값을 가질 수 있다. 이때, N은 양의 정수일 수 있고, 예컨대 3일 수 있다.

[1250] 한편, `alf_ctb_filter_alt_idx[ chromaIdx ][ CtbAddrX ][ CtbAddrY ]`가

비트스트림 내에 존재하지 않을 경우

`alf_ctb_filter_alt_idx[ chromaIdx ][ CtbAddrX ][ CtbAddrY ]`는 0의 값으로 추론될 수 있다.

[1251] 또 다른 예로, CTU는 구문 요소 `alf_ctb_cross_component_cb_idc`(또는

`alf_ctb_cc_cb_idc`)를 포함할 수 있다. `alf_ctb_cross_component_cb_idc[ CtbAddrX ][ CtbAddrY ]`는 현재 부호화 트리 블록에 대해 성분 간 Cb 필터가 적용되는지

여부를 지시할 수 있다. 예컨대, `alf_ctb_cross_component_cb_idc[ CtbAddrX ][ CtbAddrY ]`가 제 1 값인 경우, 현재 부호화 트리 블록에서 성분 간 Cb 필터가

적용되지 않을 수 있다. 반면, `alf_ctb_cross_component_cb_idc[ CtbAddrX ][ CtbAddrY ]`가 제 1 값이 아닌 경우, 현재 부호화 트리 블록에서 성분 간 Cb

필터가 적용될 수 있다. `alf_ctb_cross_component_cb_idc[ CtbAddrX ][ CtbAddrY ]`가 제 1 값이 아닌 경우, `alf_ctb_cross_component_cb_idc[ CtbAddrX ][ CtbAddrY ]`가 지시하는 필터 색인을 이용하여 성분 간 Cb 필터링이 수행될 수 있다.

[1252] 또 다른 예로, CTU는 구문 요소 `alf_ctb_cross_component_cr_idc`(또는

`alf_ctb_cc_cr_idc`)를 포함할 수 있다. `alf_ctb_cross_component_cr_idc[ CtbAddrX ][ CtbAddrY ]`는 현재 부호화 트리 블록에 대해 성분 간 Cr 필터가 적용되는지

여부를 지시할 수 있다. 예컨대, `alf_ctb_cross_component_cr_idc[ CtbAddrX ][ CtbAddrY ]`가 제 1 값인 경우, 현재 부호화 트리 블록에서 성분 간 Cr 필터가

적용되지 않을 수 있다. 반면, `alf_ctb_cross_component_cr_idc[ CtbAddrX ][ CtbAddrY ]`가 제 1 값이 아닌 경우, 현재 부호화 트리 블록에서 성분 간 Cr 필터가

적용될 수 있다. `alf_ctb_cross_component_cr_idc[ CtbAddrX ][ CtbAddrY ]`가 제 1 값이 아닌 경우, `alf_ctb_cross_component_cr_idc[ CtbAddrX ][ CtbAddrY ]`가

지시하는 필터 색인을 이용하여 성분 간 Cr 필터링이 수행될 수 있다.

[1253] 본 개시의 몇몇 실시예에 따른 적응적 루프내 필터는 하나 이상의 적응적 루프내 필터 계수를 이용하여 정의될 수 있다. 또한, 적응적 루프내 필터의 종류는 하나 이상의 적응적 루프내 필터 계수 또는 형태에 따라 정의되는 필터를 의미할 수 있다.

[1254]

[1255] 이하, 본 개시의 몇몇 실시예에 따른 적응적 루프내 필터(또는 적응적 루프 필터) 적용 방법에 대해서 설명한다.

[1256] 일 예로, 적응적 루프내 필터의 입력은 적응적 루프내 필터링이 수행되기 전의 복원 픽처 휘도 샘플 배열(recPicture_L)일 수 있다. 다른 예로, ChromaArrayType가 제 1 값이 아닌 경우, 적응적 루프내 필터의 입력은 적응적 루프내 필터링이 수행되기 전의 복원 픽처 Cb 샘플 배열(recPicture_Cb) 및 Cr 샘플 배열(recPicture_Cr)일 수 있다.

[1257] 일 예로, 적응적 루프내 필터의 출력은 적응적 루프내 필터링이 수행된 후의 변경된 복원 픽처 휘도 샘플 배열(alfPicture_L)일 수 있다. 다른 예로, ChromaArrayType가 제 1 값이 아닌 경우, 적응적 루프내 필터의 출력은 적응적 루프내 필터링이 수행된 후의 변경된 복원 픽처 Cb 샘플 배열(alfPicture_Cb) 및 Cr 샘플 배열(alfPicture_Cr)일 수 있다.

[1258] 다음으로, 적응적 루프내 필터 프로세스에 대해서 자세히 설명한다.

[1259] 먼저, alfPicture_L은 recPicture_L로 초기화될 수 있다. ChromaArrayType가 제 1 값이 아닌 경우, 적응적 루프 필터 수행 전 alfPicture_Cb 및 alfPicture_Cr는 recPicture_Cb 및 recPicture_Cr로 초기화될 수 있다.

[1260] pic_alf_enabled_flag 혹은 slice_alf_enabled_flag 가 제 2 값인 경우, 휘도 부호화 트리 블록 위치가 (rx, ry)인 모든 부호화 트리 유닛(coding tree unit)에 대해서 다음의 과정이 수행될 수 있다. 예컨대, rx는 0에서 PicWidthInCtbs - 1까지의 값을 가질 수 있고, ry는 0에서 PicHeightInCtbs - 1까지의 값을 가질 수 있다.

[1261] 일 예로, alf_ctb_flag[0][rx][ry]가 제 2 값인 경우, 휘도 성분에 대한 부호화 트리 블록 필터링 과정이 수행될 수 있다.

[1262] 휘도 성분에 대한 부호화 트리 블록 필터링 과정의 입력은 recPicture_L, alfPicture_L, 휘도 성분 부호화 트리 블록 위치 (xCtb, yCtb)일 수 있다. 이때, 휘도 성분 부호화 트리 블록 위치는 (rx << CtbLog2SizeY, ry << CtbLog2SizeY)로 결정될 수 있다.

[1263] 한편, 휘도 성분에 대한 부호화 트리 블록 필터링 과정의 출력은 alfPicture_L일 수 있다.

[1264] 다른 예로, ChromaArrayType가 제 1 값이 아니고 alf_ctb_flag[1][rx][ry]가 제 2 값인 경우 Cb 색차 성분에 대한 부호화 트리 블록 필터링 과정이 수행될 수 있다.

[1265] Cb 색차 성분에 대한 부호화 트리 블록 필터링 과정의 입력은 recPicture_Cb,

alfPicture_Cb, 색차 성분 부호화 트리 블록 위치 (xCtbC, yCtbC)일 수 있다. 이때 색차 성분 부호화 트리 블록의 위치는

$((rx \ll CtbLog2SizeY) / SubWidthC, (ry \ll CtbLog2SizeY) / SubHeightC)$ 로 결정될 수 있다.

[1266] 한편, Cb 색차 성분에 대한 부호화 트리 블록 필터링 과정의 출력은 alfPicture_Cb일 수 있다.

[1267] 또 다른 예로, ChromaArrayType가 제1 값이 아니고 alf_ctb_flag[2][rx][ry]가 제2 값인 경우 Cr 색차 성분에 대한 부호화 트리 블록 필터링 과정이 수행될 수 있다.

[1268] Cr 색차 성분에 대한 부호화 트리 블록 필터링 과정의 입력은 recPicture_Cr, alfPicture_Cr, 색차 성분 부호화 트리 블록 위치 (xCtbC, yCtbC)일 수 있다. 이때 색차 성분 부호화 트리 블록의 위치는

$((rx \ll CtbLog2SizeY) / SubWidthC, (ry \ll CtbLog2SizeY) / SubHeightC)$ 로 결정될 수 있다.

[1269] 또 다른 예로, ChromaArrayType가 제 1 값이 아니고 alf_ctb_cross_component_cb_idc[rx][ry]가 제 1 값이 아닌 경우, Cb 색차 성분에 대한 성분 간 필터링 과정이 수행될 수 있다.

[1270] Cb 색차 성분에 대한 성분 간 필터링 과정의 입력은 recPicture_L, alfPicture_Cb, 색차 성분 부호화 트리 블록 위치 (xCtbC, yCtbC)일 수 있다. 이때, 색차 성분 부호화 트리 블록의 위치는

$((rx \ll CtbLog2SizeY) / SubWidthC, (ry \ll CtbLog2SizeY) / SubHeightC)$ 로 설정될 수 있다. 또한, ccAlfWidth를

$(1 \ll \ll CtbLog2SizeY) / SubWidthC$ 로 설정될 수 있고, ccAlfHeight는

$(1 \ll \ll CtbLog2SizeY) / SubHeightC$ 로 설정될 수 있다. 또한, 성분 간 필터 계수 CcAlfCoeff[j]는

CcAlfCoeff_Cb[slice_cross_component_alf_cb_aps_id][alf_ctb_cross_component_cb_idc[rx][ry] - 1][j]로 설정될 수 있다.

[1271] 한편, Cb 색차 성분에 대한 성분 간 필터링 과정의 출력은 ccAlfPicture_Cb일 수 있다.

[1272] 또 다른 예로, ChromaArrayType가 제 1 값이 아니고 alf_ctb_cross_component_cr_idc[rx][ry]가 제 1 값이 아닌 경우, Cr 색차 성분에 대한 성분 간 필터링 과정이 수행될 수 있다.

[1273] Cr 색차 성분에 대한 성분 간 필터링 과정의 입력은 recPicture_L, alfPicture_Cr, 색차 성분 부호화 트리 블록 위치 (xCtbC, yCtbC)일 수 있다. 이때, 색차 성분 부호화 트리 블록의 위치는

$((rx \ll CtbLog2SizeY) / SubWidthC, (ry \ll CtbLog2SizeY) / SubHeightC)$ 로 설정될 수 있다. 또한, ccAlfWidth를

$(1 \ll \ll CtbLog2SizeY) / SubWidthC$ 로 설정될 수 있고, ccAlfHeight는

($1 \ll \ll \text{CtbLog2SizeY}$) / SubHeightC로 설정될 수 있다. 또한, 성분 간 필터 계수 $\text{CcAlfCoeff}[j]$ 는

$\text{CcAlfCoeff_Cr}[\text{slice_cross_component_alf_cr_aps_id}][\text{alf_ctb_cross_component_cr_idc}[rx][ry] - 1][j]$ 로 설정될 수 있다.

[1274] 한편, Cr 색차 성분에 대한 성분 간 필터링 과정의 출력은 ccAlfPicture_Cr 일 수 있다.

[1275] 색차 성분에 대한 성분 간 필터링 과정은 휘도 성분과 색차 성분 사이 왜곡을 최소화하기 위해 색차 성분에 필터링이 수행되는 것을 의미할 수 있다. 한편 본 개시에 따른 색 성분은 휘도, 색차 성분으로 제한되지 않는다. 예를 들어, G 성분과 R 혹은 B 성분 사이에 왜곡을 최소화하기 위해 R 혹은 B 성분에 대해 필터링이 수행될 수 있다. 또한, 영상의 다양한 속성이 분해된 형태인 난반사(diffuse) 성분 영상, 정반사(specular) 성분 영상, 명암(shading) 성분 영상, 직접 명암(direct shading) 성분 영상, 간접 명암(indirect shading) 성분 영상, 알베도(albedo) 성분 영상, 깊이(depth) 성분 영상, 법선 벡터(normal vector) 성분 영상 중 적어도 하나에도 본 개시에 따른 필터링 과정이 적용될 수 있다.

[1276] 이하, 상술한 휘도 성분에 대한 부호화 트리 블록 필터링 과정에 대해서 자세히 설명한다.

[1277] 일 예로, 휘도 성분에 대한 부호화 트리 블록 필터링 과정의 입력은 recPicture_L , alfPicture_L 및 현재 픽처의 좌상단 샘플을 기준으로 한 현재 휘도 성분 부호화 트리 블록의 좌상단 샘플 위치(x_{Ctb} , y_{Ctb}) 중 적어도 하나일 수 있다. 한편, 휘도 성분에 대한 부호화 트리 블록 필터링 과정의 출력은 alfPicture_L 일 수 있다.

[1278] 먼저, 필터 색인 유도 과정이 수행될 수 있다

[1279] 일 예로, 필터 색인 유도 과정의 입력은 부호화 트리 블록의 좌상단 샘플 위치(x_{Ctb} , y_{Ctb}) 및 recPicture_L 중 적어도 하나일 수 있다. 한편, 필터 색인 유도 과정의 출력은 $\text{filtIdx}[x][y]$, $\text{transposeIdx}[x][y]$ 일 수 있다. 이때, x 및 y 는 0부터 $\text{CtbSizeY} - 1$ 까지의 값을 가질 수 있다.

[1280] 다음으로, $\text{recPicture_L}[x][y]$ 내 현재 휘도 성분 부호화 트리 블록 내 각 복원 휘도 성분 샘플은 다음과 같이 필터링될 수 있다. 이때, x 및 y 는 0부터 $\text{CtbSizeY} - 1$ 까지의 값을 가질 수 있다.

[1281] $\text{filtIdx}[x][y]$ 에 의해 지시되는 필터에 대한 휘도 성분 필터 계수 배열 $f[j]$ 과 휘도 성분 절삭 값 배열 $c[j]$ 은 다음과 같이 유도될 수 있다. 이때, j 는 0에서 N 까지의 값을 가질 수 있고, 예컨대 N 은 11일 수 있다.

[1282] 일 예로, $\text{AlfCtbFiltSetIdxY}[\text{CtbAddrX}][\text{CtbAddrY}]$ 가 F 보다 작은 값을 가질 경우, 다음의 수학적 식 29에 따른 연산이 수행될 수 있다. 이때, F 는 양의 정수일 수 있고, 예컨대 16일 수 있다. 또한, F 는 부호화기/복호화기에서 사용하는 고정된 필터의 최대 개수를 의미할 수 있다. 즉, F 는 $\text{alf_luma_fixed_filter_idx}$ 가 가질 수 있는 최대값에 +1한 값일 수 있다.

[1283] [수학식 29]

[1284] $i = \text{AlfCtbFiltSetIdxY}[\text{CtbAddrX}][\text{CtbAddrY}]$

[1285] $f[j] = \text{AlfFixFiltCoeff}[\text{AlfClassToFiltMap}[i][\text{filtIdx}[x][y]][j]$

[1286] $c[j] = 2^{\text{BitdepthY}}$ 혹은 $c[j] = \text{AlfClipL}[i][\text{filtIdx}[x][y]][j]$

[1287] 여기서, $c[j]$ 에 $\text{AlfClipL}[i][\text{filtIdx}[x][y]][j]$ 를 할당하는 것은 고정된 필터에 비선형 적응적 루프내 필터링 방법인 절삭 방법을 사용하는 것을 의미할 수 있다.

[1288] 반면, $\text{AlfCtbFiltSetIdxY}[\text{CtbAddrX}][\text{CtbAddrY}]$ 가 F보다 크거나 같을 경우, 다음의 수학식 30에 따른 연산이 수행될 수 있다.

[1289] [수학식 30]

[1290] $i = \text{pic_alf_aps_id_luma}[\text{AlfCtbFiltSetIdxY}[\text{CtbAddrX}][\text{CtbAddrY}] - F]$
혹은

$i = \text{slice_alf_aps_id_luma}[\text{AlfCtbFiltSetIdxY}[\text{CtbAddrX}][\text{CtbAddrY}] - F]$

[1291] $f[j] = \text{AlfCoeff_L}[i][\text{filtIdx}[x][y]][j]$

[1292] $c[j] = \text{AlfClip_L}[i][\text{filtIdx}[x][y]][j]$

[1293] 다음으로, 휘도 성분 필터 계수 및 절삭 값에 대한 색인 idx은 $\text{transposeIdx}[x][y]$ 에 따라 아래와 같이 유도될 수 있다.

[1294] 일 예로, $\text{transposeIndex}[x][y]$ 가 제 2 값인 경우, $\text{idx}[]$ 는 { 9, 4, 10, 8, 1, 5, 11, 7, 3, 0, 2, 6 }로 결정될 수 있다.

[1295] 다른 예로, $\text{transposeIndex}[x][y]$ 가 제 3 값인 경우, $\text{idx}[]$ 는 { 0, 3, 2, 1, 8, 7, 6, 5, 4, 9, 10, 11 }로 결정될 수 있다.

[1296] 또 다른 예로, $\text{transposeIndex}[x][y]$ 가 제 4 값인 경우, $\text{idx}[]$ 는 { 9, 8, 10, 4, 3, 7, 11, 5, 1, 0, 2, 6 }로 결정될 수 있다.

[1297] 또 다른 예로, $\text{transposeIndex}[x][y]$ 가 제 2 내지 제 4 값 이외의 값을 가지는 경우, $\text{idx}[]$ 는 { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 }로 결정될 수 있다.

[1298] 한편, 대응하는 휘도 성분 샘플 (x, y)에 대한 recPicture 내 위치 ($h_{(x + i)}$, $v_{(y + j)}$)는 다음의 수학식 31에 따라 유도될 수 있다. 이때, i 및 j는 -N에서 N까지의 값을 가질 수 있다. 또한, N은 양의 정수일 수 있고, 예컨대 3일 수 있다. 또한, $(N*2)+1$ 은 휘도 성분 필터의 가로 크기 혹은 세로 크기를 의미할 수 있다.

[1299] [수학식 31]

[1300] $h_{(x + i)} = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x_{\text{Ctb}} + x + i)$

[1301] $v_{(y + j)} = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y_{\text{Ctb}} + y + j)$

[1302] 다음으로, 필터링에 대해 가상 경계를 적용할지 여부가 결정될 수 있다. 일 예로, 가상 경계를 적용할지 여부에 대한 변수 $\text{applyVirtualBoundary}$ 는 다음과 같이 유도될 수 있다.

[1303] 아래의 조건 중 적어도 하나가 참(true)일 경우, $\text{applyVirtualBoundary}$ 는 0으로 설정될 수 있다.

- [1304] 예컨대, 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 경계일 경우, 현재 부호화 트리 블록의 하단 경계가 슬라이스의 하단 경계이며, loop_filter_across_slices_enabled_flag가 제 1 값인 경우, 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계이며, loop_filter_across_tiles_enabled_flag가 제 1 값인 경우, 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계이며, loop_filter_across_subpictures_enabled_flag가 제 1 값인 경우 및 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 가상 경계이며, pps_loop_filter_across_virtual_boundaries_disabled_flag가 제 2 값인 경우 중 적어도 하나가 만족될 시, applyVirtualBoundary는 0으로 설정될 수 있다.
- [1305] 이때, loop_filter_across_slices_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 슬라이스 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 슬라이스 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [1306] 또한, loop_filter_across_tiles_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 타일 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 타일 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [1307] 또한, loop_filter_across_subpictures_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 서브 픽처 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 서브 픽처 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [1308] 한편 상술한 조건 중 적어도 하나가 만족되지 않는 경우, applyVirtualBoundary는 1로 설정될 수 있다.
- [1309] 다음으로, 복원된 샘플 오프셋 r1, r2, 및 r3은 세로 방향 휘도 성분 샘플 위치 y와 applyVirtualBoundary에 따라 다음의 표 4에 따라 결정될 수 있다.

[1310] [표 4]

[1311]

조건	r1	r2	r3
(y == CtbSizeY - 5 y == CtbSizeY - 4) && (applyVirtualBoundary == 1)	0	0	0
(y == CtbSizeY - 6 y == CtbSizeY - 3) && (applyVirtualBoundary == 1)	1	1	1
(y == CtbSizeY - 7 y == CtbSizeY - 2) && (applyVirtualBoundary == 1)	1	2	2
그렇지 않은 경우(위의 조건에 해당하지 않은 경우)	1	2	3

- [1312] 다음으로, 현재 샘플 위치에 대한 샘플값 curr은 다음의 수학식 32에 따라 유도될 수 있다.
- [1313] [수학식 32]
- [1314] $curr = recPicture_L[h_x, v_y]$
- [1315] 다음으로, 현재 샘플에 대해 필터링이 수행된 값 sum은 다음의 수학식 33에 따라 유도될 수 있다.
- [1316] [수학식 33]

[1317]
$$\begin{aligned} \text{sum} = & f[\text{idx}[0]] * (\text{Clip3}(-c[\text{idx}[0]], c[\text{idx}[0]], \text{recPicture}_L[h_x, v_{y+r3}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[0]], c[\text{idx}[0]], \text{recPicture}_L[h_x, v_{y-r3}] - \text{curr})) + \\ & f[\text{idx}[1]] * (\text{Clip3}(-c[\text{idx}[1]], c[\text{idx}[1]], \text{recPicture}_L[h_{x+1}, v_{y+r2}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[1]], c[\text{idx}[1]], \text{recPicture}_L[h_{x-1}, v_{y-r2}] - \text{curr})) + \\ & f[\text{idx}[2]] * (\text{Clip3}(-c[\text{idx}[2]], c[\text{idx}[2]], \text{recPicture}_L[h_x, v_{y+r2}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[2]], c[\text{idx}[2]], \text{recPicture}_L[h_x, v_{y-r2}] - \text{curr})) + \\ & f[\text{idx}[3]] * (\text{Clip3}(-c[\text{idx}[3]], c[\text{idx}[3]], \text{recPicture}_L[h_{x-1}, v_{y+r2}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[3]], c[\text{idx}[3]], \text{recPicture}_L[h_{x+1}, v_{y-r2}] - \text{curr})) + \\ & f[\text{idx}[4]] * (\text{Clip3}(-c[\text{idx}[4]], c[\text{idx}[4]], \text{recPicture}_L[h_{x+2}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[4]], c[\text{idx}[4]], \text{recPicture}_L[h_{x-2}, v_{y-r1}] - \text{curr})) + \\ & f[\text{idx}[5]] * (\text{Clip3}(-c[\text{idx}[5]], c[\text{idx}[5]], \text{recPicture}_L[h_{x+1}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[5]], c[\text{idx}[5]], \text{recPicture}_L[h_{x-1}, v_{y-r1}] - \text{curr})) + \\ & f[\text{idx}[6]] * (\text{Clip3}(-c[\text{idx}[6]], c[\text{idx}[6]], \text{recPicture}_L[h_x, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[6]], c[\text{idx}[6]], \text{recPicture}_L[h_x, v_{y-r1}] - \text{curr})) + \\ & f[\text{idx}[7]] * (\text{Clip3}(-c[\text{idx}[7]], c[\text{idx}[7]], \text{recPicture}_L[h_{x-1}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[7]], c[\text{idx}[7]], \text{recPicture}_L[h_{x+1}, v_{y-r1}] - \text{curr})) + \\ & f[\text{idx}[8]] * (\text{Clip3}(-c[\text{idx}[8]], c[\text{idx}[8]], \text{recPicture}_L[h_{x-2}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[8]], c[\text{idx}[8]], \text{recPicture}_L[h_{x+2}, v_{y-r1}] - \text{curr})) + \\ & f[\text{idx}[9]] * (\text{Clip3}(-c[\text{idx}[9]], c[\text{idx}[9]], \text{recPicture}_L[h_{x+3}, v_y] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[9]], c[\text{idx}[9]], \text{recPicture}_L[h_{x-3}, v_y] - \text{curr})) + \\ & f[\text{idx}[10]] * (\text{Clip3}(-c[\text{idx}[10]], c[\text{idx}[10]], \text{recPicture}_L[h_{x+2}, v_y] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[10]], c[\text{idx}[10]], \text{recPicture}_L[h_{x-2}, v_y] - \text{curr})) + \\ & f[\text{idx}[11]] * (\text{Clip3}(-c[\text{idx}[11]], c[\text{idx}[11]], \text{recPicture}_L[h_{x+1}, v_y] - \text{curr}) + \\ & \text{Clip3}(-c[\text{idx}[11]], c[\text{idx}[11]], \text{recPicture}_L[h_{x-1}, v_y] - \text{curr})) \\ \text{sum} = & \text{curr} + ((\text{sum} + 64) >> 7) \end{aligned}$$

[1318] 마지막으로, $\text{alfPicture}_L[xCtb + x][yCtb + y]$ 는 다음과 같이 유도될 수 있다.

[1319] 일 예로, $\text{pcm_loop_filter_disabled_flag}$ 가 제 2 값이며 $\text{pcm_flag}[xCtb + x][yCtb + y]$ 가 제2 값인 경우, 다음의 수학적 식 34에 따른 연산이 수행될 수 있다.

[1320] [수학적 식 34]

[1321] $\text{alfPicture}_L[xCtb + x][yCtb + y] = \text{recPicture}_L[h_x, v_y]$

[1322] 반면, $\text{pcm_loop_filter_disabled_flag}$ 가 제 1 값이며 $\text{pcm_flag}[xCtb + x][yCtb + y]$ 가 제1 값인 경우), 다음의 수학적 식 35에 따른 연산이 수행될 수 있다.

[1323] [수학적 식 35]

[1324] $\text{alfPicture}_L[xCtb + x][yCtb + y] = \text{Clip3}(0, (1 \ll \text{BitDepthY}) - 1, \text{sum})$

[1325]

[1326] 이하, 상술한 필터 색인 유도 과정에 대해서 자세히 설명한다.

[1327] 일 예로, 필터 색인 유도 과정의 입력은 부호화 트리 블록의 좌상단 샘플 위치($xCtb, yCtb$) 및 recPicture_L 중 적어도 하나일 수 있다.

[1328] 한편, 필터 색인 유도 과정의 출력은 분류 필터 색인 배열(classification filter index array) $\text{filtIdx}[x][y]$ 및 전치 색인 배열(transpose index array) $\text{transposeIdx}[x][y]$ 중 적어도 하나일 수 있다. 이때, x 및 y 는 0부터

CtbSizeY - 1까지의 값을 가질 수 있다.

- [1329] 먼저, 대응하는 휘도 성분 샘플 (x, y)에 대한 recPicture 내 위치 (h_(x + i), v_(y + j))는 다음의 수학적 식 36 내지 수학적 식 39에 따라 유도될 수 있다. 이때, i 및 j는 -M에서 N까지의 값을 가질 수 있다. 여기서, M과 N은 양의 정수일 수 있고, 예컨대 M은 2, N은 5일 수 있다. 또한, M+N+1은 1D 라플라시안 연산의 합이 계산되는 범위(가로 크기 및 세로 크기)를 의미할 수 있다.
- [1330] [수학적 식 36]
- [1331] $h_{(x + i)} = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x_{\text{Ctb}} + x + i)$
- [1332] yCtb + CtbSizeY가 pic_height_in_luma_samples보다 크거나 같을 경우, 다음의 수학적 식 37에 따른 연산이 수행될 수 있다.
- [1333] [수학적 식 37]
- [1334] $v_{(y + j)} = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y_{\text{Ctb}} + y + j)$
- [1335] y가 CtbSizeY - 4보다 작을 경우, 다음의 수학적 식 38에 따른 연산이 수행될 수 있다.
- [1336] [수학적 식 38]
- [1337] $v_{(y + j)} = \text{Clip3}(0, y_{\text{Ctb}} + \text{CtbSizeY} - 5, y_{\text{Ctb}} + y + j)$
- [1338] 위의 조건 중 적어도 하나가 만족되지 않는 경우 다음의 수학적 식 39에 따른 연산이 수행될 수 있다.
- [1339] [수학적 식 39]
- [1340] $v_{(y + j)} = \text{Clip3}(y_{\text{Ctb}} + \text{CtbSizeY} - 4, \text{pic_height_in_luma_samples} - 1, y_{\text{Ctb}} + y + j)$
- [1341] 다음으로, 분류 필터 색인 배열 filtIdx와 전치 색인 배열 transposeIdx이 유도될 수 있다.
- [1342] 먼저, filtH[x][y], filtV[x][y], filtD0[x][y] 및 filtD1[x][y]가 다음의 연산에 따라 유도될 수 있다. 이때, x 및 y는 -2부터 CtbSizeY + 1까지의 값을 가질 수 있다. x와 y가 모두 짝수일 경우 혹은 x와 y가 모두 홀수일 경우, 다음의 수학적 식 40에 따른 연산이 수행될 수 있다.
- [1343] [수학적 식 40]
- [1344]
$$\begin{aligned} \text{filtH}[x][y] &= \text{Abs}((\text{recPicture}[h_x, v_y] \ll 1) - \text{recPicture}[h_{x-1}, v_y] - \text{recPicture}[h_{x+1}, v_y]) \\ \text{filtV}[x][y] &= \text{Abs}((\text{recPicture}[h_x, v_y] \ll 1) - \text{recPicture}[h_x, v_{y-1}] - \text{recPicture}[h_x, v_{y+1}]) \\ \text{filtD0}[x][y] &= \text{Abs}((\text{recPicture}[h_x, v_y] \ll 1) - \text{recPicture}[h_{x-1}, v_{y-1}] - \text{recPicture}[h_{x+1}, v_{y+1}]) \\ \text{filtD1}[x][y] &= \text{Abs}((\text{recPicture}[h_x, v_y] \ll 1) - \text{recPicture}[h_{x+1}, v_{y-1}] - \text{recPicture}[h_{x-1}, v_{y+1}]) \end{aligned}$$
- [1345] 한편, 위 조건(x와 y가 모두 짝수일 경우 혹은 x와 y가 모두 홀수일 경우)을 만족하지 않는 경우, filtH[x][y], filtV[x][y], filtD0[x][y] 및 filtD1[x][y]은 0으로 설정될 수 있다.
- [1346] 다음으로, minY, maxY 및 ac가 유도될 수 있다.
- [1347] 먼저, (y << 2)가 (CtbSizeY - 8)와 동일하고, (yCtb + CtbSizeY)가 pic_height_in_luma_samples - 1보다 작을 경우, 다음의 연산이 수행될 수 있다.

- [1348] minY 는 $-M$ 로 설정될 수 있다. 이때, M 은 양의 정수일 수 있고, 예컨대 2일 수 있다. maxY 는 N 으로 설정될 수 있다. 이때, N 은 양의 정수일 수 있고, 예컨대 3일 수 있다. ac 는 96으로 설정될 수 있다.
- [1349] 이때, $M+N+1$ 은 1D 라플라시안 연산의 합이 계산되는 범위(세로 크기)를 의미할 수 있다.
- [1350] minY 가 -2이며 maxY 가 3일 경우는 minY 가 -2이며 maxY 가 5일 경우보다 1D 라플라시안 연산의 합이 계산되는 범위가 작아지므로, $\text{avgVar}[x][y]$ 값이 0에서 15 사이의 값으로 계산될 수 있도록 ac 값이 조정될 필요가 있다. 따라서, 이러한 경우 ac 는 64와 96 사이의 값으로 설정될 수 있으며, 예를 들어 ac 는 85 혹은 86일 수 있다.
- [1351] 즉, minY 가 -2이며 maxY 가 3일 경우에는 1D 라플라시안 연산의 합이 계산되는 범위(가로 크기 x 세로 크기)가 8×6 이 되며, minY 가 -2이며 maxY 가 5일 경우에는 1D 라플라시안 연산의 합이 계산되는 범위(가로 크기 x 세로 크기)가 8×8 이 되므로, 1D 라플라시안 연산의 합이 계산되는 범위가 8×8 일 때 사용되는 ac 인 64를 $4/3$ 만큼 곱을 해줄 필요가 있다. 64에 대해 $4/3$ 만큼 곱을 할 경우 85.33의 값이 나오므로 ac 값은 85 혹은 86으로 설정될 수 있다.
- [1352] 다음으로, $(y \ll 2)$ 가 $(\text{CtbSizeY} - 4)$ 와 동일하고, $(y\text{Ctb} + \text{CtbSizeY})$ 가 $\text{pic_height_in_luma_samples} - 1$ 보다 작을 경우, 다음의 연산이 수행될 수 있다.
- [1353] minY 는 M 로 설정될 수 있다. 이때, M 은 양의 정수일 수 있고, 예컨대 0일 수 있다. maxY 는 N 으로 설정될 수 있다. 이때, N 은 양의 정수일 수 있고, 예컨대 5일 수 있다. ac 는 96으로 설정될 수 있다.
- [1354] 여기서, $M+N+1$ 은 1D 라플라시안 연산의 합이 계산되는 범위(세로 크기)를 의미할 수 있다.
- [1355] minY 가 0이며 maxY 가 5일 경우는 minY 가 -2이며 maxY 가 5일 경우보다 1D 라플라시안 연산의 합이 계산되는 범위가 작아지므로, $\text{avgVar}[x][y]$ 값이 0에서 15 사이의 값으로 계산될 수 있도록 ac 값을 조정될 필요가 있다. 따라서, 이러한 경우 ac 는 64와 96 사이의 값으로 설정될 수 있으며, 예를 들어 ac 는 85 혹은 86일 수 있다.
- [1356] 즉, minY 가 0이며 maxY 가 5일 경우에는 1D 라플라시안 연산의 합이 계산되는 범위(가로 크기 x 세로 크기)가 8×6 이 되며, minY 가 -2이며 maxY 가 5일 경우에는 1D 라플라시안 연산의 합이 계산되는 범위(가로 크기 x 세로 크기)가 8×8 이 되므로, 1D 라플라시안 연산의 합이 계산되는 범위가 8×8 일 때 사용되는 ac 인 64에 대해 $4/3$ 만큼 곱을 해줄 필요가 있다. 64에 대해 $4/3$ 만큼 곱을 할 경우 85.33의 값이 나오므로 ac 값은 85 혹은 86으로 설정될 수 있다.
- [1357] 위 조건들이 만족되지 않는 경우, 다음의 연산이 수행될 수 있다.
- [1358] minY 는 $-M$ 로 설정될 수 있다. 이때, M 은 양의 정수일 수 있고, 예컨대 2일 수 있다. maxY 는 N 으로 설정될 수 있다. 이때, N 은 양의 정수일 수 있고, 예컨대 5일 수 있다. ac 는 64로 설정될 수 있다.

[1359] 여기서, $M+N+1$ 은 1D 라플라시안 연산의 합이 계산되는 범위(세로 크기)를 의미할 수 있다.

[1360] 다음으로, $\text{sumH}[x][y]$, $\text{sumV}[x][y]$, $\text{sumD0}[x][y]$, $\text{sumD1}[x][y]$ 및 $\text{sumOfHV}[x][y]$ 가 유도될 수 있다.

[1361] 일 예로, $\text{sumH}[x][y]$, $\text{sumV}[x][y]$, $\text{sumD0}[x][y]$, $\text{sumD1}[x][y]$ 및 $\text{sumOfHV}[x][y]$ 는 다음의 수학적 식 41에 따라 유도될 수 있다. 이때, x 및 y 는 0부터 $(\text{CtbSizeY} - 1) \gg 2$ 까지의 값을 가질 수 있다.

[1362] [수학적 식 41]

$$\begin{aligned} \text{sumH}[x][y] &= \sum_i \sum_j \text{filtH}[h_{(x \ll 2)+i} - x\text{Ctb}][v_{(y \ll 2)+j} - y\text{Ctb}] \\ \text{sumV}[x][y] &= \sum_i \sum_j \text{filtV}[h_{(x \ll 2)+i} - x\text{Ctb}][v_{(y \ll 2)+j} - y\text{Ctb}] \\ \text{sumD0}[x][y] &= \sum_i \sum_j \text{filtD0}[h_{(x \ll 2)+i} - x\text{Ctb}][v_{(y \ll 2)+j} - y\text{Ctb}] \\ \text{sumD1}[x][y] &= \sum_i \sum_j \text{filtD1}[h_{(x \ll 2)+i} - x\text{Ctb}][v_{(y \ll 2)+j} - y\text{Ctb}] \\ \text{sumOfHV}[x][y] &= \text{sumH}[x][y] + \text{sumV}[x][y] \end{aligned}$$

[1364] 이때, i 는 -2에서 5까지의 값을 가질 수 있고, j 는 minY 에서 maxY 까지의 값을 가질 수 있다.

[1365] 다음으로, $\text{dir1}[x][y]$, $\text{dir2}[x][y]$ 및 $\text{dirS}[x][y]$ 가 유도될 수 있다. 이때, x 및 y 는 0부터 $\text{CtbSizeY} - 1$ 까지의 값을 가질 수 있다.

[1366] 먼저, hv1 , hv0 및 dirHV 는 다음과 같이 유도될 수 있다.

[1367] 일 예로, $\text{sumV}[x \gg 2][y \gg 2]$ 가 $\text{sumH}[x \gg 2][y \gg 2]$ 보다 클 경우, 다음의 수학적 식 42에 따른 연산이 수행될 수 있다.

[1368] [수학적 식 42]

$$\text{hv1} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 1$$

[1372] 반면, $\text{sumV}[x \gg 2][y \gg 2]$ 가 $\text{sumH}[x \gg 2][y \gg 2]$ 보다 크지 않은 경우, 다음의 수학적 식 43에 따른 연산이 수행될 수 있다.

[1373] [수학적 식 43]

$$\text{hv1} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 3$$

[1377] 다음으로, $d1$, $d0$ 및 dirD 는 다음과 같이 유도될 수 있다.

[1378] 일 예로, $\text{sumD0}[x \gg 2][y \gg 2]$ 가 $\text{sumD1}[x \gg 2][y \gg 2]$ 보다 클 경우, 다음의 수학적 식 44에 따른 연산이 수행될 수 있다.

[1379] [수학적 식 44]

$$d1 = \text{sumD0}[x \gg 2][y \gg 2]$$

$$d0 = \text{sumD1}[x \gg 2][y \gg 2]$$

$$\text{dirD} = 0$$

[1383] 반면, $\text{sumD0}[x \gg 2][y \gg 2]$ 가 $\text{sumD1}[x \gg 2][y \gg 2]$ 보다 크지 않은 경우, 다음의 수학적식 45에 따른 연산이 수행될 수 있다.

[1384] [수학적식 45]

[1385] $d1 = \text{sumD1}[x \gg 2][y \gg 2]$

[1386] $d0 = \text{sumD0}[x \gg 2][y \gg 2]$

[1387] $\text{dirD} = 2$

[1388] 다음으로, Hvd1 및 hvd0 는 다음의 수학적식 46에 따라 유도될 수 있다.

[1389] [수학적식 46]

[1390] $\text{hvd1} = (d1 * \text{hv0} > \text{hv1} * d0) ? d1 : \text{hv1}$

[1391] $\text{hvd0} = (d1 * \text{hv0} > \text{hv1} * d0) ? d0 : \text{hv0}$

[1392] 다음으로, $\text{dirS}[x][y]$, $\text{dir1}[x][y]$ 및 $\text{dir2}[x][y]$ 는 다음의 수학적식 67에 따라 유도될 수 있다.

[1393] [수학적식 47]

[1394] $\text{dir1}[x][y] = (d1 * \text{hv0} > \text{hv1} * d0) ? \text{dirD} : \text{dirHV}$

[1395] $\text{dir2}[x][y] = (d1 * \text{hv0} > \text{hv1} * d0) ? \text{dirHV} : \text{dirD}$

[1396] $\text{dirS}[x][y] = (\text{hvd1} * 2 > 9 * \text{hvd0}) ? 2 : ((\text{hvd1} > 2 * \text{hvd0}) ? 1 : 0)$

[1397] 다음으로 $\text{avgVar}[x][y]$ 가 다음의 수학적식 48에 따라 유도될 수 있다. 이때, x 및 y 는 0부터 $\text{CtbSizeY} - 1$ 까지의 값을 가질 수 있다.

[1398] [수학적식 48]

[1399] $\text{varTab}[] = \{0, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4\}$

[1400] $\text{avgVar}[x][y] = \text{varTab}[\text{Clip3}(0, 15, (\text{sumOfHV}[x \gg 2][y \gg 2] * \text{ac}) \gg (3 + \text{BitDepthY}))]$

[1401] 마지막으로, 분류 필터 색인 배열 $\text{filtIdx}[x][y]$ 과 전치 색인 배열 $\text{transposeIdx}[x][y]$ 은 다음의 수학적식 49에 따라 유도될 수 있다. 이때, x 및 y 는 0부터 $\text{CtbSizeY} - 1$ 까지의 값을 가질 수 있다.

[1402] [수학적식 49]

[1403] $\text{transposeTable}[] = \{0, 1, 0, 2, 2, 3, 1, 3\}$

[1404] $\text{transposeIdx}[x][y] = \text{transposeTable}[\text{dir1}[x][y] * 2 + (\text{dir2}[x][y] \gg 1)]$

[1405] $\text{filtIdx}[x][y] = \text{avgVar}[x][y]$

[1406] 한편, $\text{dirS}[x][y]$ 가 제1 값이 아닌 경우, $\text{filtIdx}[x][y]$ 는 다음의 수학적식 50에 따라 변경될 수 있다.

[1407] [수학적식 50]

[1408] $\text{filtIdx}[x][y] += (((\text{dir1}[x][y] \& 0x1) \ll 1) + \text{dirS}[x][y]) * 5$

[1409]

[1410] 이하, 본 개시의 일 실시예에 따른 색차 성분에 대한 부호화 트리 블록 필터링(적응적 루프 필터) 과정에 대해서 설명한다.

- [1411] 일 예로, 색차 성분에 대한 부호화 트리 블록 필터링 과정의 입력은 `recPicture`, `alfPicture` 및 부호화 트리 블록의 좌상단 샘플 위치(`xCtb`, `yCtb`) 중 적어도 하나일 수 있다.
- [1412] 한편, 색차 성분에 대한 부호화 트리 블록 필터링 과정의 출력은 `alfPicture`일 수 있다.
- [1413] 먼저, 현재 색차 성분 부호화 트리 블록의 가로 크기 `ctbWidthC`와 세로 크기 `ctbHeightC`는 다음의 수학적 식 51에 따라 유도될 수 있다.
- [1414] [수학적 식 51]
- [1415]
$$\text{ctbWidthC} = \text{CtbSizeY} / \text{SubWidthC}$$
- [1416]
$$\text{ctbHeightC} = \text{CtbSizeY} / \text{SubHeightC}$$
- [1417] 다음으로, `recPicture[ x ][ y ]` 내 현재 색차 성분 부호화 트리 블록 내 각 복원 색차 샘플은 다음과 같이 필터링될 수 있다. 이때, `x`는 0부터 `ctbWidthC` - 1까지의 값을 가질 수 있다. 또한, `y`는 0부터 `ctbHeightC` - 1까지의 값을 가질 수 있다.
- [1418] 각 대응하는 색차 성분 샘플 (`x`, `y`)에 대한 `recPicture` 내 위치 (`h_(x + i)`, `v_(y + j)`)는 다음의 수학적 식 52에 따라 유도될 수 있다. 이때, `i` 및 `j`는 -`N`에서 `N`까지의 값을 가질 수 있다. 또한, `N`은 양의 정수일 수 있고, 예컨대 2일 수 있다. 또한, $(N*2)+1$ 은 색차 성분 필터의 가로 크기 혹은 세로 크기를 의미할 수 있다.
- [1419] [수학적 식 52]
- [1420]
$$h_{(x + i)} = \text{Clip3}(0, \text{pic_width_in_luma_samples} / \text{SubWidthC} - 1, x\text{CtbC} + x + i)$$
- [1421]
$$v_{(y + j)} = \text{Clip3}(0, \text{pic_height_in_luma_samples} / \text{SubHeightC} - 1, y\text{CtbC} + y + j)$$
- [1422] 다음으로, 필터링에 대해 가상 경계를 적용할지 여부가 결정될 수 있다. 일 예로, 가상 경계를 적용할지 여부에 대한 변수 `applyVirtualBoundary`는 다음과 같이 유도될 수 있다.
- [1423] 아래의 조건 중 적어도 하나가 참(true)일 경우, `applyVirtualBoundary`는 0으로 설정될 수 있다.
- [1424] 예컨대, 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 경계일 경우, 현재 부호화 트리 블록의 하단 경계가 슬라이스의 하단 경계이며, `loop_filter_across_slices_enabled_flag`가 제 1 값인 경우, 현재 부호화 트리 블록의 하단 경계가 타일의 하단 경계이며, `loop_filter_across_tiles_enabled_flag`가 제 1 값인 경우, 현재 부호화 트리 블록의 하단 경계가 서브 픽처의 하단 경계이며, `loop_filter_across_subpictures_enabled_flag`가 제 1 값인 경우 및 현재 부호화 트리 블록의 하단 경계가 픽처의 하단 가상 경계이며, `pps_loop_filter_across_virtual_boundaries_disabled_flag`가 제 2 값인 경우 중 적어도 하나가 만족될 시, `applyVirtualBoundary`는 0으로 설정될 수 있다.

- [1425] 이때, loop_filter_across_slices_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 슬라이스 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 슬라이스 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [1426] 또한, loop_filter_across_tiles_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 타일 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 타일 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [1427] 또한, loop_filter_across_subpictures_enabled_flag는 루프내 필터링 방법들 중 적어도 하나가 서브 픽처 경계를 넘어서 필터링될 수 있는지 여부를 의미할 수 있다. 즉, 서브 픽처 경계에 루프내 필터링 방법들 중 적어도 하나를 사용하여 필터링할 수 있는지 여부를 의미할 수 있다.
- [1428] 한편 상술한 조건 중 적어도 하나가 만족되지 않는 경우, applyVirtualBoundary는 1로 설정될 수 있다.
- [1429] 다음으로 복원된 샘플 오프셋 r1 및 r2은 세로 방향 색차 성분 샘플 위치 y와 applyVirtualBoundary에 따라 다음의 표 5에 따라 결정될 수 있다.

[1430] [표 5]

[1431]

조건	r1	r2
(y = = ctbHeightC - 2 y = = ctbHeightC - 3) && (applyVirtualBoundary = = 1)	0	0
(y = = ctbHeightC - 1 y = = ctbHeightC - 4) && (applyVirtualBoundary = = 1)	1	1
그렇지 않은 경우(위의 조건에 해당하지 않은 경우)	1	2

- [1432] 다음으로, 현재 샘플 위치에 대한 샘플값 curr은 다음의 수학적식 53에 따라 유도될 수 있다.
- [1433] [수학적식 53]
- [1434] curr = recPicture[h_x, v_y]
- [1435] 다음으로, 색차 성분 필터 계수 배열 f[j]과 색차 성분 절삭 값 배열 c[j]은 다음의 수학적식 54에 따라 유도될 수 있다. 이때, j는 0에서 N까지의 값을 가질 수 있고, 예컨대 N은 5일 수 있다.
- [1436] [수학적식 54]
- [1437] f[j] = AlfCoeff_C[pic_alf_aps_id_chroma][j] 혹은 f[j] = AlfCoeff_C[slice_alf_aps_id_chroma][j]
- [1438] c[j] = AlfClip_C[pic_alf_aps_id_chroma][j] 혹은 c[j] = AlfClip_C[slice_alf_aps_id_chroma][j]
- [1439] 다음으로, 현재 샘플에 대해 필터링이 수행된 값 sum은 다음의 수학적식 55에 따라 유도될 수 있다.
- [1440] [수학적식 55]

[1441]
$$\begin{aligned} \text{sum} = & f[0] * (\text{Clip3}(-c[0], c[0], \text{recPicture}[h_x, v_{y+r2}] - \text{curr}) + \\ & \text{Clip3}(-c[0], c[0], \text{recPicture}[h_x, v_{y-r2}] - \text{curr})) + \\ & f[1] * (\text{Clip3}(-c[1], c[1], \text{recPicture}[h_{x+1}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[1], c[1], \text{recPicture}[h_{x-1}, v_{y-r1}] - \text{curr})) + \\ & f[2] * (\text{Clip3}(-c[2], c[2], \text{recPicture}[h_x, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[2], c[2], \text{recPicture}[h_x, v_{y-r1}] - \text{curr})) + \\ & f[3] * (\text{Clip3}(-c[3], c[3], \text{recPicture}[h_{x-1}, v_{y+r1}] - \text{curr}) + \\ & \text{Clip3}(-c[3], c[3], \text{recPicture}[h_{x+1}, v_{y-r1}] - \text{curr})) + \\ & f[4] * (\text{Clip3}(-c[4], c[4], \text{recPicture}[h_{x+2}, v_y] - \text{curr}) + \\ & \text{Clip3}(-c[4], c[4], \text{recPicture}[h_{x-2}, v_y] - \text{curr})) + \\ & f[5] * (\text{Clip3}(-c[5], c[5], \text{recPicture}[h_{x+1}, v_y] - \text{curr}) + \\ & \text{Clip3}(-c[5], c[5], \text{recPicture}[h_{x-1}, v_y] - \text{curr})) \\ \text{sum} = & \text{curr} + (\text{sum} + 64) >> 7 \end{aligned}$$

[1442] 마지막으로, $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y]$ 는 다음과 같이 유도될 수 있다.

[1443] 일 예로, $\text{pcm_loop_filter_disabled_flag}$ 가 제 2 값이며 $\text{pcm_flag}[(x\text{CtbC} + x) * \text{SubWidthC}][(y\text{CtbC} + y) * \text{SubHeightC}]$ 가 제 2 값인 경우, 다음의 수학적식 56에 따른 연산이 수행될 수 있다.

[1444] [수학적식 56]

[1445] $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] = \text{recPicture}[h_x, v_y]$

[1446] 반면, $\text{pcm_loop_filter_disabled_flag}$ 가 제 1 값이며

$\text{pcm_flag}[(x\text{CtbC} + x) * \text{SubWidthC}][(y\text{CtbC} + y) * \text{SubHeightC}]$ 가 제 1 값인 경우 다음의 수학적식 57에 따른 연산이 수행될 수 있다.

[1447] [수학적식 57]

[1448] $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, \text{sum})$

[1449]

[1450] 이하, 본 개시의 일 실시예에 따른 성분 간 루프내 필터링(Cross-component adaptive loop filter) 방법에 대해서 설명한다

[1451] 일 예로, 색차 성분에 대한 성분 간 루프내 필터링 과정의 입력은 recPicture_L , 적응적 루프내 필터가 적용된 색차 샘플 배열 alfPicture_C , 현재 픽처의 좌상단 샘플을 기준으로 한 현재 색차 성분 부호화 트리 블록의 좌상단 샘플 위치($x\text{CtbC}$, $y\text{CtbC}$), 색차 블록의 가로 크기 ccAlfWidth , 색차 블록의 세로 크기 ccAlfHeight 및 성분 간 필터 계수 $\text{CcAlfCoeff}[j]$ 중 적어도 하나일 수 있다.

[1452] 한편, 색차 성분에 대한 성분 간 루프내 필터링 과정의 출력은 변경되고 필터링된 복원 색차 성분 픽처 샘플 배열 ccAlfPicture 일 수 있다.

[1453] 먼저, 현재 색차 성분 부호화 트리 블록의 위치 ($x\text{Ctb}$, $y\text{Ctb}$)는 다음의 수학적식 58에 따라 유도될 수 있다.

[1454] [수학식 58]

[1455] $x_{Ctb} = (((x_{CtbC} * SubWidthC) >> CtbLog2SizeY) << CtbLog2SizeY$

[1456] $y_{Ctb} = (((y_{CtbC} * SubHeightC) >> CtbLog2SizeY) << CtbLog2SizeY$

[1457] 다음으로, 현재 색차 블록 내 복원된 각 색차 샘플

$alfPicture_C[x_{CtbC} + x][y_{CtbC} + y]$ 은 다음과 같이 필터링될 수 있다.

이때, x 는 0부터 $ccAlfWidth - 1$ 까지의 값을 가질 수 있다. 또한, y 는 0부터 $ccAlfHeight - 1$ 까지의 값을 가질 수 있다.

[1458] 현재 색차 샘플 위치 $(x_{CtbC} + x, y_{CtbC} + y)$ 에 대응하는 휘도 위치 (xL, yL) 는 $(x_{CtbC} + x) * SubWidthC, (y_{CtbC} + y) * SubHeightC$ 로 설정될 수 있다.

[1459] 변수 $clipLeftPos, clipRightPos, clipTopPos$ 및 $clipBottomPos$ 는 ALF 경계 위치 유도 과정으로부터 계산될 수 있다.

[1460] 일 예로, 세로 휘도 샘플 위치 $yL, clipLeftPos$ 및 $clipRightPos$ 에 따라 세로 샘플 위치 오프셋 $yM1, yP1$ 및 $yP2$ 는 다음의 표 6에 기초하여 유도될 수 있다. 또한, 가로 휘도 샘플 위치 $xL, clipLeftPos$ 및 $clipRightPos$ 에 따라 가로 샘플 위치 오프셋 $xM1$ 및 $xP1$ 는 다음의 표 7에 기초하여 유도될 수 있다.

[1461] [표 6]

[1462]

조건	yM1	yP1	yP2
$yL == clipTopPos + 1$	-1	1	1
$yL == clipTopPos$	0	0	1
$yL == clipBottomPos - 1$	0	0	1
$yL == clipBottomPos - 2$	-1	1	1
그렇지 않은 경우	-1	1	2

[1463] [표 7]

[1464]

조건	xM1	xP1
$xL == clipLeftPos$	0	0
$xL == clipRightPos - 1$	0	0
$xL == clipRightPos - 2$	-1	1
그렇지 않은 경우	-1	1

[1465] 다음으로, 현재 샘플 위치에 대한 샘플값 $curr$ 은 다음의 수학식 59에 따라 유도될 수 있다.

[1466] [수학식 59]

[1467] $curr = alfPicture_C[x_{CtbC} + x, y_{CtbC} + y]$

[1468] 다음으로, 성분 간 필터 계수 배열 $f[j]$ 은 다음의 수학식 60에 따라 유도될 수 있다. 이때, j 는 0에서 N 까지의 값을 가질 수 있다, 예컨대 N 은 양의 정수일 수 있고, 7일 수 있다. 다른 예로, 계산 복잡도 감소를 위해 상기 N 을 7보다 작은 수로 하여 곱셈기의 개수를 감소시킬 수 있다.

[1469] [수학식 60]

[1470] $f[j] = CcAlfCoeff[j]$

[1471] 다음으로, 현재 샘플에 대해 필터링이 수행된 값 sum은 다음의 수학식 61에 따라 유도될 수 있다.

[1472] [수학식 61]

[1473]
$$\begin{aligned} \text{sum} = & f[0] * \text{recPicture}_L[h_x, v_{y+yM1}] + \\ & f[1] * \text{recPicture}_L[h_{x+xM1}, v_y] + \\ & f[2] * \text{recPicture}_L[h_x, v_y] + \\ & f[3] * \text{recPicture}_L[h_{x+xP1}, v_y] + \\ & f[4] * \text{recPicture}_L[h_{x+xM1}, v_{y+yP1}] + \\ & f[5] * \text{recPicture}_L[h_x, v_{y+yP1}] + \\ & f[6] * \text{recPicture}_L[h_{x+xP1}, v_{y+yP1}] + \\ & f[7] * \text{recPicture}_L[h_x, v_{y+yP2}] \\ \text{sum} = & \text{Clip3}(-(1 \ll (\text{BitDepth}_C - 1)), (1 \ll (\text{BitDepth}_C - 1)) - 1, \text{sum}) \\ \text{sum} = & \text{curr} + (\text{sum} + 64) \gg (7 + (\text{BitDepth}_Y - \text{BitDepth}_C)) \end{aligned}$$

[1474] 마지막으로, $ccAlfPicture[xCtbC + x][yCtbC + y]$ 는 다음의 수학식 62에 따라 유도될 수 있다.

[1475] [수학식 62]

[1476] $ccAlfPicture[xCtbC + x][yCtbC + y] = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, \text{sum})$

[1477] 본 개시에 따른 필터링에는 상술한 필터의 예 (3x4 다이아몬드 필터) 뿐만 아니라 다른 형태의 필터 중 적어도 하나가 이용될 수도 있다. 한편, 본 개시의 몇몇 실시예에서 필터링 수행 시 곱셈 연산 대신 쉬프트 연산을 수행하기 위해 상기 성분 간 필터 계수의 절대값이 2의 제곱승으로 이루어지게 제한될 수 있다.

[1478]

[1479] $alf_luma_clip_idx[sfIdx][j]$ 에 사용되는 지수-골롬 부호는 해당 구문 요소의 값의 범위가 클 경우, 효율적으로 엔트로피 부호화/복호화할 수 있는 방법일 수 있다. 하지만, 상술한 바와 같이 $alf_luma_clip_idx[sfIdx][j]$ 가 가질 수 있는 값의 범위가 상대적으로 작은 경우(예컨대 0 내지 3), 지수-골롬 부호는 엔트로피 부호화/복호화 측면에서 비효율적일 수 있다.

[1480] 따라서, $alf_luma_clip_min_eg_order_minus1$ 및 $alf_luma_clip_eg_order_increase_flag[i]$ 중 적어도 하나의 구문 요소를 사용하는 대신, $tu(3)$, $f(2)$, $u(2)$, $tb(3)$ 등 중 적어도 하나의 엔트로피 부호화/복호화 방법을 사용하여 $alf_luma_clip_idx[sfIdx][j]$ 를 엔트로피 부호화/복호화할 수 있다.

[1481] $alf_chroma_clip_idx[sfIdx][j]$ 에 사용되는 지수-골롬 부호는 해당 구문 요소의 값의 범위가 클 경우, 효율적으로 엔트로피 부호화/복호화할 수 있는 방법일 수 있다. 하지만, 상술한 바와 같이 $alf_chroma_clip_idx[sfIdx][j]$ 가 가질 수 있는 값의 범위가 상대적으로 작은 경우(예컨대 0 내지 3), 지수-골롬

부호는 엔트로피 부호화/복호화 측면에서 비효율적일 수 있다.

- [1482] 따라서, `alf_chroma_clip_min_eg_order_minus1` 및 `alf_chroma_clip_eg_order_increase_flag[ i ]` 중 적어도 하나의 구문 요소를 사용하는 대신, `tu(3)`, `f(2)`, `u(2)`, `tb(3)` 등 중 적어도 하나의 엔트로피 부호화/복호화 방법을 사용하여 `alf_chroma_clip_idx[ sfIdx ][ j ]`를 엔트로피 부호화/복호화할 수 있다.
- [1483] 도 49 및 도 50은 본 개시의 또 다른 일 실시예에 따른 ALF 데이터를 설명하기 위한 다른 도면이다.
- [1484] 도 49 및 도 50에 도시된 바와 같이, `alf_luma_clip_idx[ sfIdx ][ j ]`와 `alf_chroma_clip_idx[ j ]`를 엔트로피 부호화/복호화하는데 `u(2)` 엔트로피 부호화/복호화 방식을 사용할 수 있다. 이때, `u(2)`의 최대값은 `N`일 수 있다. 이때, `N`은 양의 정수일 수 있고, 예컨대 3일 수 있다. 또한, `u(2)`는 `f(2)`와 동일한 의미를 가질 수 있다.
- [1485] 도 51 및 도 52는 본 개시의 또 다른 일 실시예에 따른 ALF 데이터를 설명하기 위한 또 다른 도면이다.
- [1486] 도 51 및 도 52에 도시된 바와 같이, `alf_luma_clip_idx[ sfIdx ][ j ]`와 `alf_chroma_clip_idx[ j ]`를 엔트로피 부호화/복호화하는데 `tu(v)` 엔트로피 부호화/복호화 방식을 사용할 수 있다. 이때, `tu(v)`의 최대값(`maxVal`)은 `N`일 수 있다. 이때, `N`은 양의 정수일 수 있고, 예컨대 3일 수 있다. 또한, `tu(v)`의 최대값이 3일 경우에 `tu(v)`는 `tu(3)`과 동일한 의미를 가질 수 있다.
- [1487] 도 49 내지 도 52에 따른 실시예가 적용되는 경우, 구문 요소 `alf_luma_clip_min_eg_order_minus1`, `alf_luma_clip_eg_order_increase_flag[ i ]`, `alf_chroma_clip_min_eg_order_minus1`, `alf_chroma_clip_eg_order_increase_flag[ i ]`와 구문 요소 `alf_luma_clip_min_eg_order_minus1`, `alf_luma_clip_eg_order_increase_flag[ i ]`, `alf_chroma_clip_min_eg_order_minus1`, `alf_chroma_clip_eg_order_increase_flag[ i ]`의 의미(semantics)를 사용하지 않고, `alf_luma_clip_idx[ sfIdx ][ j ]`와 `alf_chroma_clip_idx[ j ]`를 엔트로피 부호화/복호화를 할 수 있으므로, `alf_luma_clip_idx[ sfIdx ][ j ]`와 `alf_chroma_clip_idx[ j ]`를 엔트로피 부호화/복호화하는데 필요한 부호화기/복호화기의 구현 복잡도가 감소될 수 있다.
- [1488] 또한, 지수-골롬 부호의 차수 `kClipY[ i ]`, 지수-골롬 부호의 차수 `kClipC[ i ]`, 지수-골롬 부호(이진화) `uek(v)`의 차수 `k`를 유도하는데 필요한 부호화/복호화 과정도 수행되지 않으므로, `alf_luma_clip_idx[ sfIdx ][ j ]`와 `alf_chroma_clip_idx[ j ]`를 엔트로피 부호화/복호화하는데 필요한 부호화기/복호화기의 구현 복잡도가 감소할 수 있다.
- [1489] 도 49 내지 도 52에 따른 실시예가 적용되는 경우, 지수-골롬 부호를 이용하여 `alf_luma_clip_idx[ sfIdx ][ j ]`와 `alf_chroma_clip_idx[ j ]`를 엔트로피

부호화/복호화할 때보다 더 적은 비트로 $\text{alf_luma_clip_idx}[\text{sfIdx}][j]$ 와 $\text{alf_chroma_clip_idx}[j]$ 를 엔트로피 부호화/복호화할 수 있으므로, 영상 부호화기/복호화기의 부호화 효율이 향상될 수 있다.

- [1490] 한편, $\text{tu}(v)$ 에 대한 엔트로피 복호화 및 구문 요소의 값을 의미하는 codeNum 의 획득은 다음의 표 8에 따라 수행될 수 있다. 여기서, 엔트로피 복호화는 파싱(parsing)을 의미할 수 있다. $\text{tu}(v)$ 에 대한 값의 범위는 0부터 최대값(maxVal)로 결정될 수 있다. 이때, 최대값은 양의 정수일 수 있고, 예를 들어 1보다 크거나 같을 수 있다.

[1491] [표 8]

[1492]

```
codeNum = 0
keepGoing = 1
for(i = 0; i < maxVal && keepGoing; i++){
    keepGoing = read_bits( 1 )
    if( keepGoing )
        codeNum ++
}
```

- [1493] 본 발명에 따른 실시예들에서 altIdx 와 sfIdx 는 동일한 의미를 가질 수 있다.

- [1494] 본 발명에 따른 실시예 중 적어도 하나는 휘도 영상, 색차 영상 중 적어도 하나에 적용될 뿐만 아니라, 영상의 다양한 속성이 분해된 형태인 난반사(diffuse) 성분 영상, 정반사(specular) 성분 영상, 명암(shading) 성분 영상, 직접 명암(direct shading) 성분 영상, 간접 명암(indirect shading) 성분 영상, 알베도(albedo) 성분 영상, 깊이(depth) 성분 영상, 법선 벡터(normal vector) 성분 영상 중 적어도 하나에 적용되어, 영상 간 중복성을 제거하고 영상의 부호화 효율을 향상시킬 수 있다.

[1495]

- [1496] 상기의 실시예들은 부호화기 및 복호화기에서 같은 방법으로 수행될 수 있다.

- [1497] 상기 실시예들 중 적어도 하나 혹은 적어도 하나의 조합을 이용해서 영상을 부호화/복호화할 수 있다.

- [1498] 상기 실시예를 적용하는 순서는 부호화기와 복호화기에서 상이할 수 있고, 상기 실시예를 적용하는 순서는 부호화기와 복호화기에서 동일할 수 있다.

- [1499] 휘도 및 색차 신호 각각에 대하여 상기 실시예를 수행할 수 있고, 휘도 및 색차 신호에 대한 상기 실시예를 동일하게 수행할 수 있다.

- [1500] 본 발명의 상기 실시예들이 적용되는 블록의 형태는 정방형(square) 형태 혹은 비정방형(non-square) 형태를 가질 수 있다.

- [1501] 상기 부호화기에서 엔트로피 부호화되고 복호화기에서 엔트로피 복호화되는 구문 요소(플래그, 인덱스 등) 중 적어도 하나는 아래의 이진화(binanzation), 역이진화(debinanzation), 엔트로피 부호화/복호화 방법 중 적어도 하나 이상을

이용할 수 있다.

- [1502] - 부호를 가지는 0차수 지수-골롬(0-th order Exp_Golomb) 이진화/역이진화 방법 (se(v))
- [1503] - 부호를 가지는 k차수 지수-골롬(k-th order Exp_Golomb) 이진화/역이진화 방법 (sek(v))
- [1504] - 부호를 가지지 않는 양의 정수에 대한 0차수 지수-골롬(0-th order Exp_Golomb) 이진화/역이진화 방법 (ue(v))
- [1505] - 부호를 가지지 않는 양의 정수에 대한 k차수 지수-골롬(k-th order Exp_Golomb) 이진화/역이진화 방법 (uek(v))
- [1506] - 고정 길이(Fixed-length) 이진화/역이진화 방법 (f(n))
- [1507] - 절삭된 라이스(Truncated Rice) 이진화/역이진화 방법 혹은 절삭된 단항(Truncated Unary) 이진화/역이진화 방법 (tu(v))
- [1508] - 절삭된 이진(Truncated Binary) 이진화/역이진화 방법 (tb(v))
- [1509] - 문맥 적응적 산술 부호화/복호화 방법 (ae(v))
- [1510] - 바이트 단위 비트 스트링 (b(8))
- [1511] - 부호를 가지는 정수 이진화/역이진화 방법 (i(n))
- [1512] - 부호를 가지지 않는 양의 정수 이진화/역이진화 방법 (u(n))
- [1513] - 단항(Unary) 이진화/역이진화 방법
- [1514] 본 발명의 상기 실시예들은 부호화 블록, 예측 블록, 변환 블록, 블록, 현재 블록, 부호화 유닛, 예측 유닛, 변환 유닛, 유닛, 현재 유닛 중 적어도 하나의 크기에 따라 적용될 수 있다. 여기서의 크기는 상기 실시예들이 적용되기 위해 최소 크기 및/또는 최대 크기로 정의될 수도 있고, 상기 실시예가 적용되는 고정 크기로 정의될 수도 있다. 또한, 상기 실시예들은 제1 크기에서는 제1의 실시예가 적용될 수도 있고, 제2 크기에서는 제2의 실시예가 적용될 수도 있다. 즉, 상기 실시예들은 크기에 따라 복합적으로 적용될 수 있다. 또한, 본 발명의 상기 실시예들은 최소 크기 이상 및 최대 크기 이하일 경우에만 적용될 수도 있다. 즉, 상기 실시예들을 블록 크기가 일정한 범위 내에 포함될 경우에만 적용될 수도 있다.
- [1515] 예를 들어, 현재 블록의 크기가 8x8 이상일 경우에만 상기 실시예들이 적용될 수 있다. 예를 들어, 현재 블록의 크기가 4x4일 경우에만 상기 실시예들이 적용될 수 있다. 예를 들어, 현재 블록의 크기가 16x16 이하일 경우에만 상기 실시예들이 적용될 수 있다. 예를 들어, 현재 블록의 크기가 16x16 이상이고 64x64 이하일 경우에만 상기 실시예들이 적용될 수 있다.
- [1516] 본 발명의 상기 실시예들은 시간적 계층(temporal layer)에 따라 적용될 수 있다. 상기 실시예들이 적용 가능한 시간적 계층을 식별하기 위해 별도의 식별자(identifier)가 시그널링되고, 해당 식별자에 의해 특정된 시간적 계층에 대해서 상기 실시예들이 적용될 수 있다. 여기서의 식별자는 상기 실시예가 적용 가능한 최하위 계층 및/또는 최상위 계층으로 정의될 수도 있고, 상기 실시예가

적용되는 특정 계층을 지시하는 것으로 정의될 수도 있다. 또한, 상기 실시예가 적용되는 고정된 시간적 계층이 정의될 수도 있다.

[1517] 예를 들어, 현재 영상의 시간적 계층이 최하위 계층일 경우에만 상기 실시예들이 적용될 수 있다. 예를 들어, 현재 영상의 시간적 계층 식별자가 1 이상인 경우에만 상기 실시예들이 적용될 수 있다. 예를 들어, 현재 영상의 시간적 계층이 최상위 계층일 경우에만 상기 실시예들이 적용될 수 있다.

[1518] 본 발명의 상기 실시예들이 적용되는 슬라이스 종류(slice type) 혹은 타일 그룹 종류가 정의되고, 해당 슬라이스 종류 혹은 타일 그룹 종류에 따라 본 발명의 상기 실시예들이 적용될 수 있다.

[1519] 상술한 실시예들에서, 방법들은 일련의 단계 또는 유닛으로서 순서도를 기초로 설명되고 있으나, 본 발명은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 당해 기술 분야에서 통상의 지식을 가진 자라면 순서도에 나타난 단계들이 배타적이지 않고, 다른 단계가 포함되거나, 순서도의 하나 또는 그 이상의 단계가 본 발명의 범위에 영향을 미치지 않고 삭제될 수 있음을 이해할 수 있을 것이다.

[1520] 상술한 실시예는 다양한 양태의 예시들을 포함한다. 다양한 양태들을 나타내기 위한 모든 가능한 조합을 기술할 수는 없지만, 해당 기술 분야의 통상의 지식을 가진 자는 다른 조합이 가능함을 인식할 수 있을 것이다. 따라서, 본 발명은 이하의 특허청구범위 내에 속하는 모든 다른 교체, 수정 및 변경을 포함한다고 할 것이다.

[1521] 이상 설명된 본 발명에 따른 실시예들은 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드 뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[1522] 이상에서 본 발명이 구체적인 구성요소 등과 같은 특정 사항들과 한정된

실시에 및 도면에 의해 설명되었으나, 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명이 상기 실시예들에 한정되는 것은 아니며, 본 발명이 속하는 기술분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형을 꾀할 수 있다.

- [1523] 따라서, 본 발명의 사상은 상기 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등하게 또는 등가적으로 변형된 모든 것들은 본 발명의 사상의 범주에 속한다고 할 것이다.

산업상 이용가능성

- [1524] 본 발명은 영상을 부호화/복호화 하는데 이용될 수 있다.

청구범위

- [청구항 1] 영상 복호화 방법에 있어서,
 현재 블록의 복원 휘도 블록을 획득하는 단계;
 상기 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component adaptive loop filter)의 적용 여부를 결정하는 단계;
 상기 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우,
 적응적 루프 필터(adaptive loop filter)가 적용된 상기 현재 블록의 복원 색차 블록을 획득하는 단계; 및
 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의 최종 복원 색차 블록을 획득하는 단계를 포함하는, 영상 복호화 방법.
- [청구항 2] 제 1항에 있어서,
 상기 성분 간 적응적 루프 필터의 적용 여부를 결정하는 단계는,
 상기 현재 블록에 대한 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보를 획득하는 단계를 포함하되,
 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는 시퀀스 파라미터 세트(sequence parameter set)를 통해 시그널링되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 3] 제 2항에 있어서,
 상기 현재 블록에 대해 상기 성분 간 적응적 루프 필터의 적용이 가능한 경우,
 상기 현재 블록의 Cb 성분에 대한 성분 간 적응적 루프 필터의 적용 여부를 지시하는 정보를 획득하는 단계를 더 포함하는, 영상 복호화 방법.
- [청구항 4] 제 3항에 있어서,
 상기 Cb 성분에 대한 성분 간 루프 필터의 적용 여부를 지시하는 정보는, 픽처 헤더(picture header)를 통해 시그널링되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 5] 제 3항에 있어서,
 상기 Cb 성분에 대해 성분 간 루프 필터가 적용되는 경우, 상기 Cb 성분이 참조하는 적응 파라미터 세트 지시자를 획득하는 단계를 더 포함하는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 6] 제 2항에 있어서,
 상기 현재 블록에 대해 상기 성분 간 적응적 루프 필터의 적용이 가능한 경우,
 상기 현재 블록의 Cr 성분에 대한 성분 간 적응적 루프 필터의 적용 여부를 지시하는 정보를 획득하는 단계를 더 포함하는, 영상 복호화 방법.
- [청구항 7] 제 6항에 있어서,

상기 Cr 성분에 대한 성분 간 루프 필터의 적용 여부를 지시하는 정보는, 픽처 헤더(picture header)를 통해 시그널링되는 것을 특징으로 하는, 영상 복호화 방법.

[청구항 8] 제 6항에 있어서,
상기 Cr 성분에 대해 성분 간 루프 필터가 적용되는 경우, 상기 Cr 성분이 참조하는 적응 파라미터 세트 지시자를 획득하는 단계를 더 포함하는 것을 특징으로 하는, 영상 복호화 방법.

[청구항 9] 제 2항에 있어서,
상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보가 시그널링되지 않는 경우,
상기 현재 블록에 대해서는 성분 간 적응적 루프 필터의 적용이 불가능한 것으로 결정되는 것을 특징으로 하는, 영상 복호화 방법.

[청구항 10] 제 1항에 있어서,
상기 현재 블록의 복원 색차 블록을 획득하는 단계; 및
상기 복원 색차 블록에 대해 적응적 루프 필터를 적용할지 여부를 결정하는 단계를 더 포함하는, 영상 복호화 방법.

[청구항 11] 제 10항에 있어서,
상기 복원 색차 블록에 대해 적응적 루프 필터의 적용 여부를 결정하는 단계는,
상기 복원 색차 블록에 대한 상기 적응적 루프 필터의 적용 가능 여부를 지시하는 정보를 획득하는 단계를 포함하되,
상기 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는 픽처 헤더(picture header)를 통해 시그널링되는 것을 특징으로 하는, 영상 복호화 방법.

[청구항 12] 제 11항에 있어서,
상기 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는, Cb 성분 및 Cr 성분 각각에 대해 시그널링되는 것을 특징으로 하는, 영상 복호화 방법.

[청구항 13] 제 11항에 있어서,
상기 복원 색차 블록에 적용되는 적응적 루프 필터의 종류를 지시하는 정보를 획득하는 단계를 더 포함하되,
상기 적응적 루프 필터의 종류는 선형적(linear) 적응적 루프 필터 및 비선형적(non-linear) 적응적 루프 필터를 포함하는 것을 특징으로 하는, 영상 복호화 방법.

[청구항 14] 제 13항에 있어서,
상기 복원 색차 블록에 적용되는 적응적 루프 필터의 종류를 지시하는 정보는 단일 구문 요소인 것을 특징으로 하는, 영상 복호화 방법.

[청구항 15] 제 1항에 있어서,

상기 최종 복원 색차 블록은,
 상기 복원 휘도 블록에 상기 성분 간 적응적 루프 필터를 적용하여 획득된
 성분 간 루프 필터가 적용된 휘도 블록과 적응적 루프 필터가 적용된 복원
 색차 블록을 이용하여 유도되는 것을 특징으로 하는, 영상 복호화 방법.

[청구항 16] 영상 부호화 방법에 있어서,
 현재 블록의 복원 휘도 블록을 획득하는 단계;
 상기 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component
 adaptive loop filter)의 적용 여부를 결정하는 단계;
 상기 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우,
 적응적 루프 필터(adaptive loop filter)가 적용된 상기 현재 블록의 복원
 색차 블록을 획득하는 단계; 및
 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차
 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의
 최종 복원 색차 블록을 획득하는 단계를 포함하는, 영상 부호화 방법.

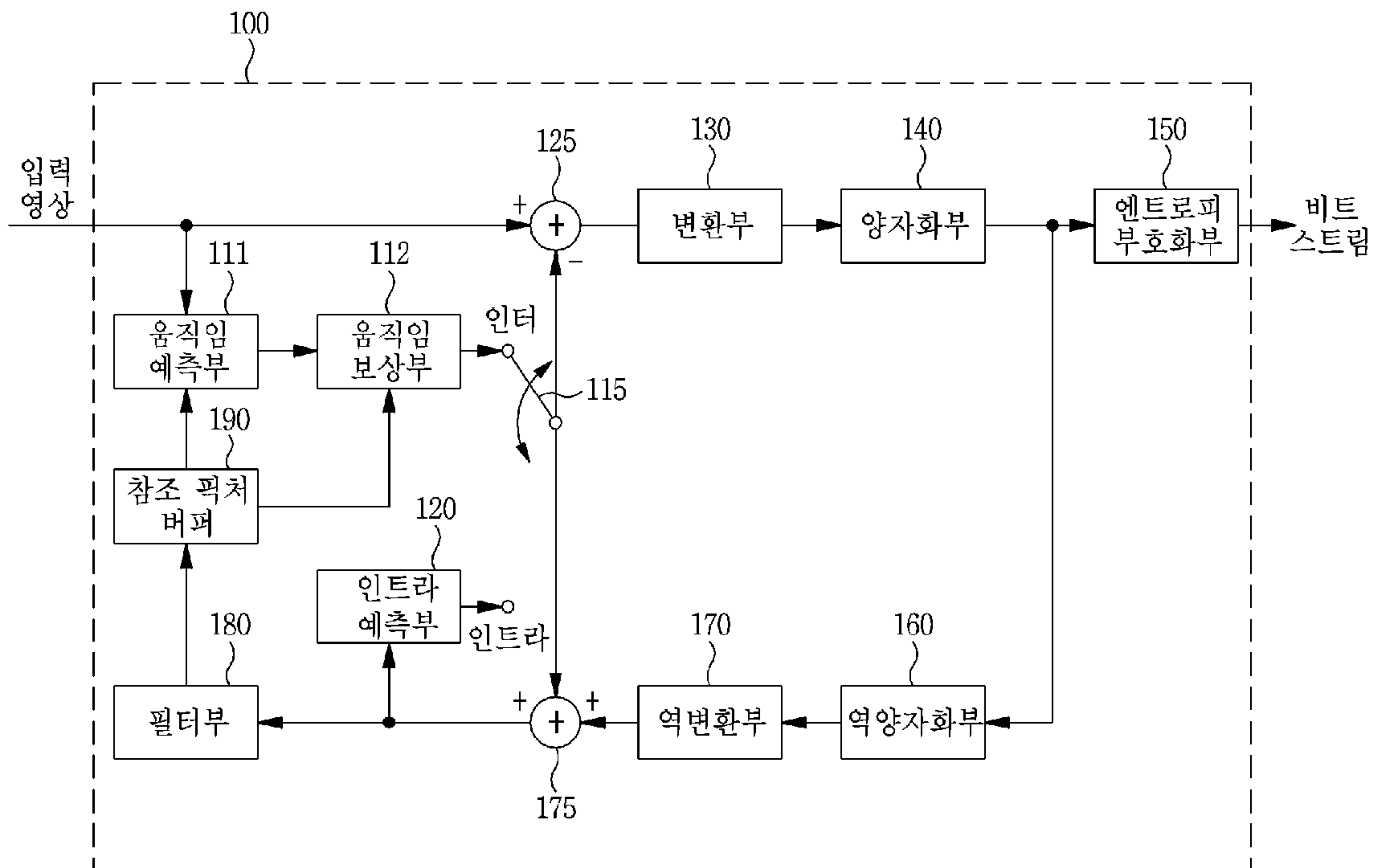
[청구항 17] 제 16항에 있어서,
 상기 현재 블록에 대한 상기 성분 간 적응적 루프 필터의 적용 가능
 여부를 지시하는 정보를 부호화하는 단계를 더 포함하되,
 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는
 시퀀스 파라미터 세트(sequence parameter set)를 통해 부호화되는 것을
 특징으로 하는, 영상 부호화 방법.

[청구항 18] 제 16항에 있어서,
 상기 최종 복원 색차 블록은,
 상기 복원 휘도 블록에 상기 성분 간 적응적 루프 필터를 적용하여 획득된
 성분 간 루프 필터가 적용된 휘도 블록과 적응적 루프 필터가 적용된 복원
 색차 블록을 이용하여 유도되는 것을 특징으로 하는, 영상 부호화 방법.

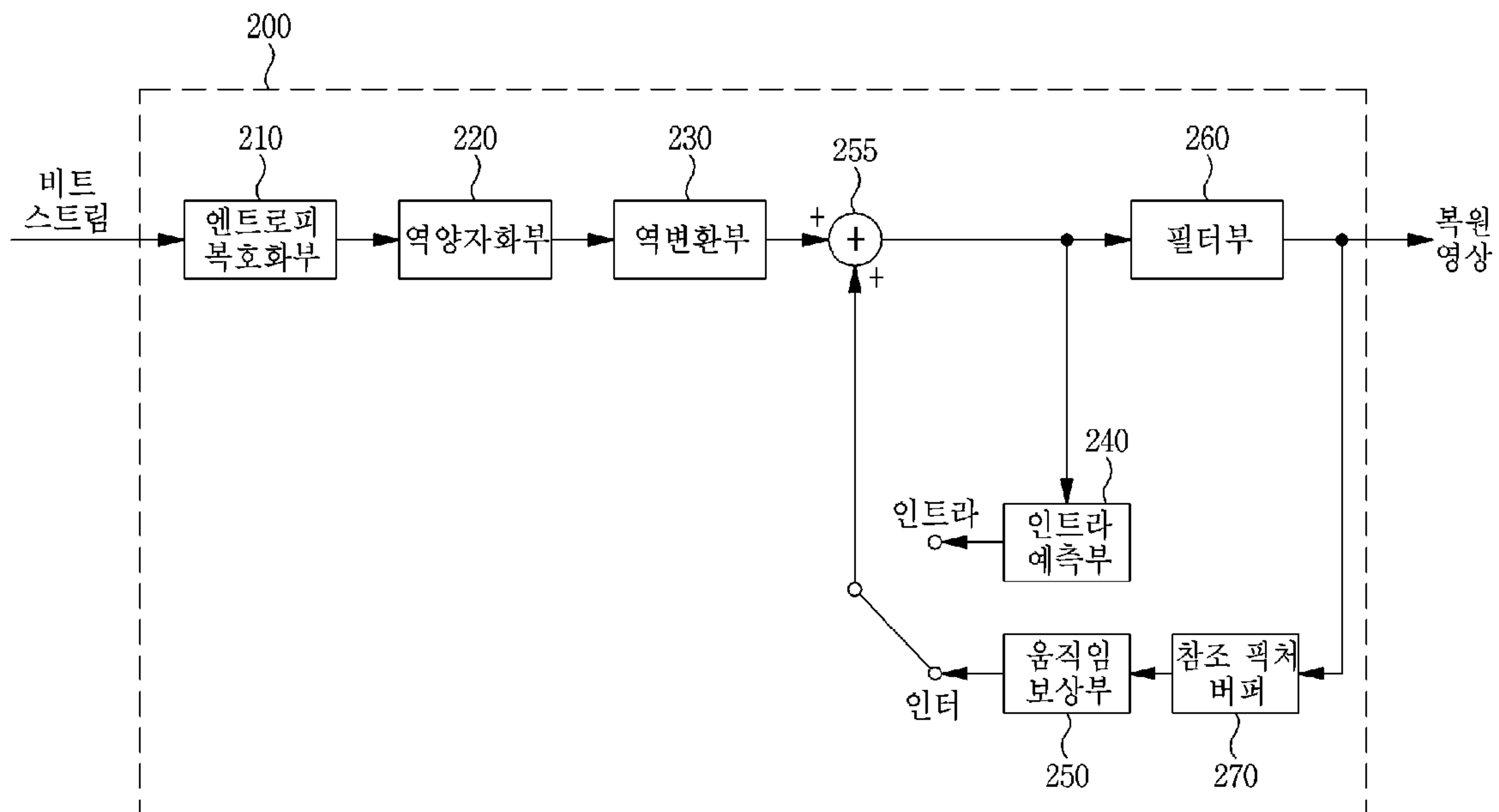
[청구항 19] 영상 복호화 장치에 수신되고 현재 픽처에 포함된 현재 블록을
 복원하는데 이용되는 비트스트림을 저장한 컴퓨터 판독가능한
 기록매체에 있어서,
 현재 블록의 복원 휘도 블록을 획득하는 단계;
 상기 현재 블록에 대한 성분 간 적응적 루프 필터(Cross-component
 adaptive loop filter)의 적용 여부를 결정하는 단계;
 상기 현재 블록에 대해 성분 간 적응적 루프 필터가 적용되는 경우,
 적응적 루프 필터(adaptive loop filter)가 적용된 상기 현재 블록의 복원
 색차 블록을 획득하는 단계; 및
 상기 복원 휘도 블록 및 상기 적응적 루프 필터가 적용된 복원 색차
 블록을 이용하여 상기 성분 간 적응적 루프 필터가 적용된 현재 블록의
 최종 복원 색차 블록을 획득하는 단계를 포함하는 영상 부호화 방법에
 의해 생성된 비트스트림을 저장하는 컴퓨터 판독 가능한 기록 매체.

[청구항 20] 제 19항에 있어서,
상기 현재 블록에 대한 상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보를 부호화하는 단계를 더 포함하되,
상기 성분 간 적응적 루프 필터의 적용 가능 여부를 지시하는 정보는 시퀀스 파라미터 세트(sequence parameter set)를 통해 부호화되는 것을 특징으로 하는, 컴퓨터 판독 가능한 기록 매체.

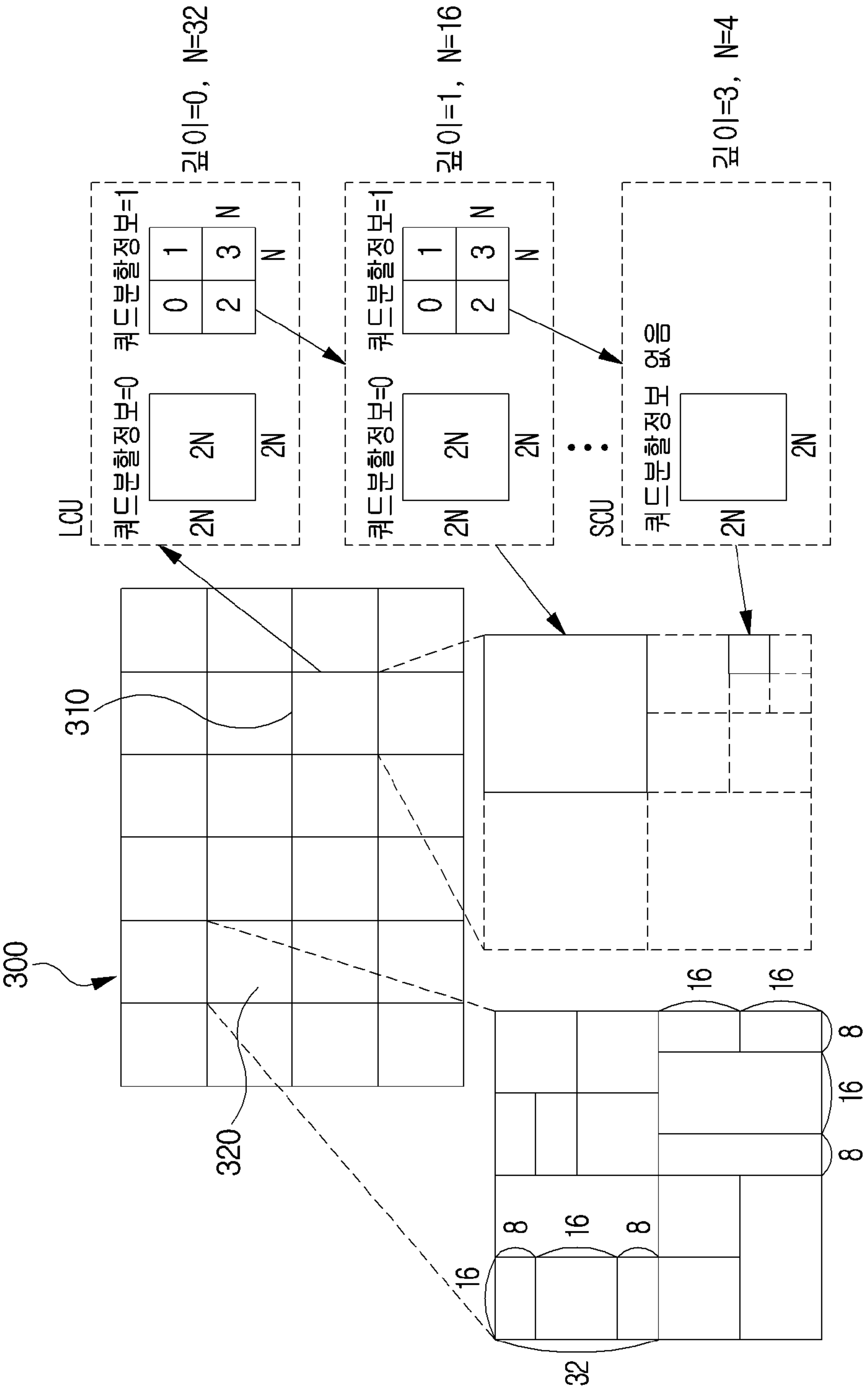
[도1]



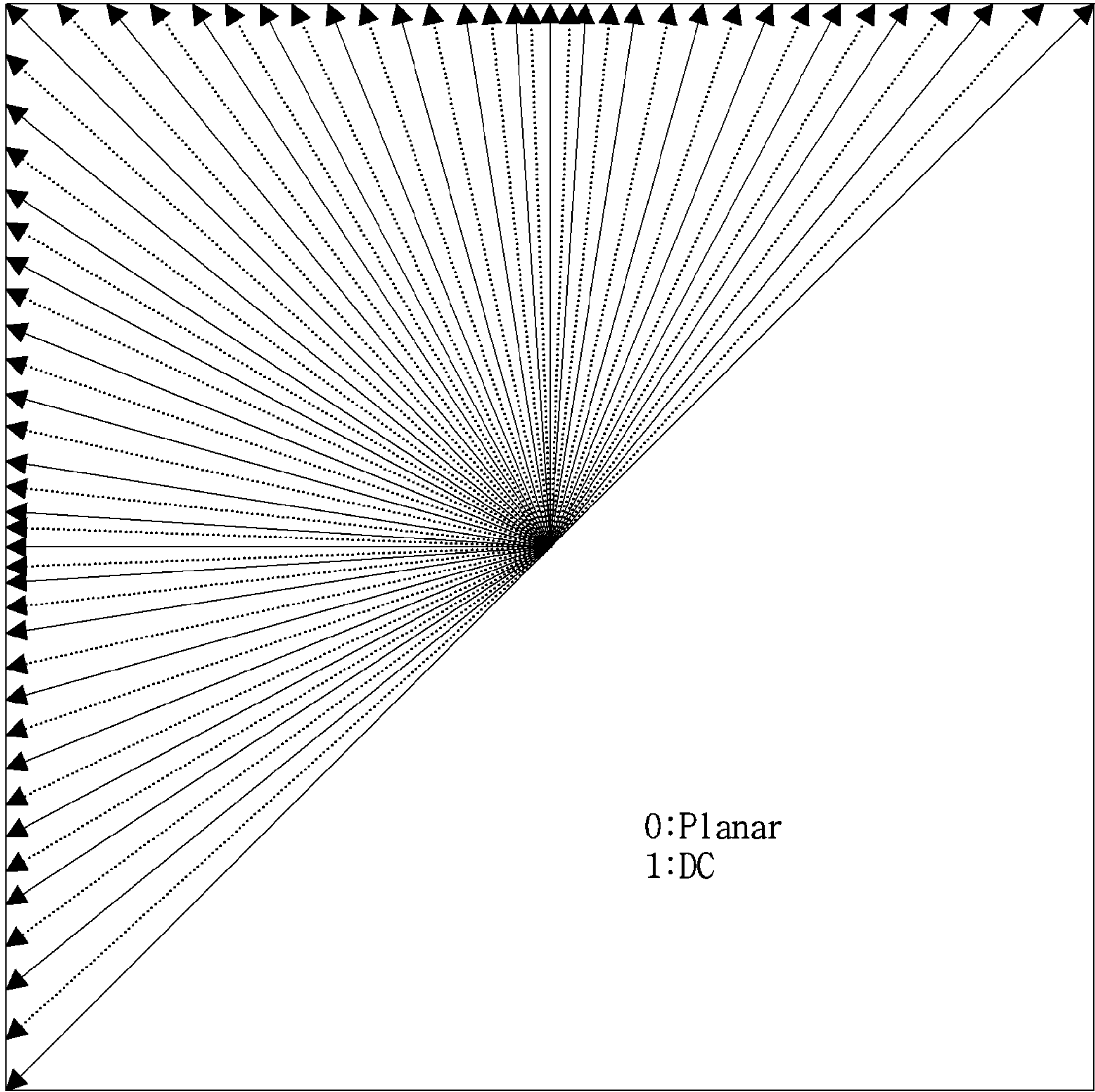
[도2]



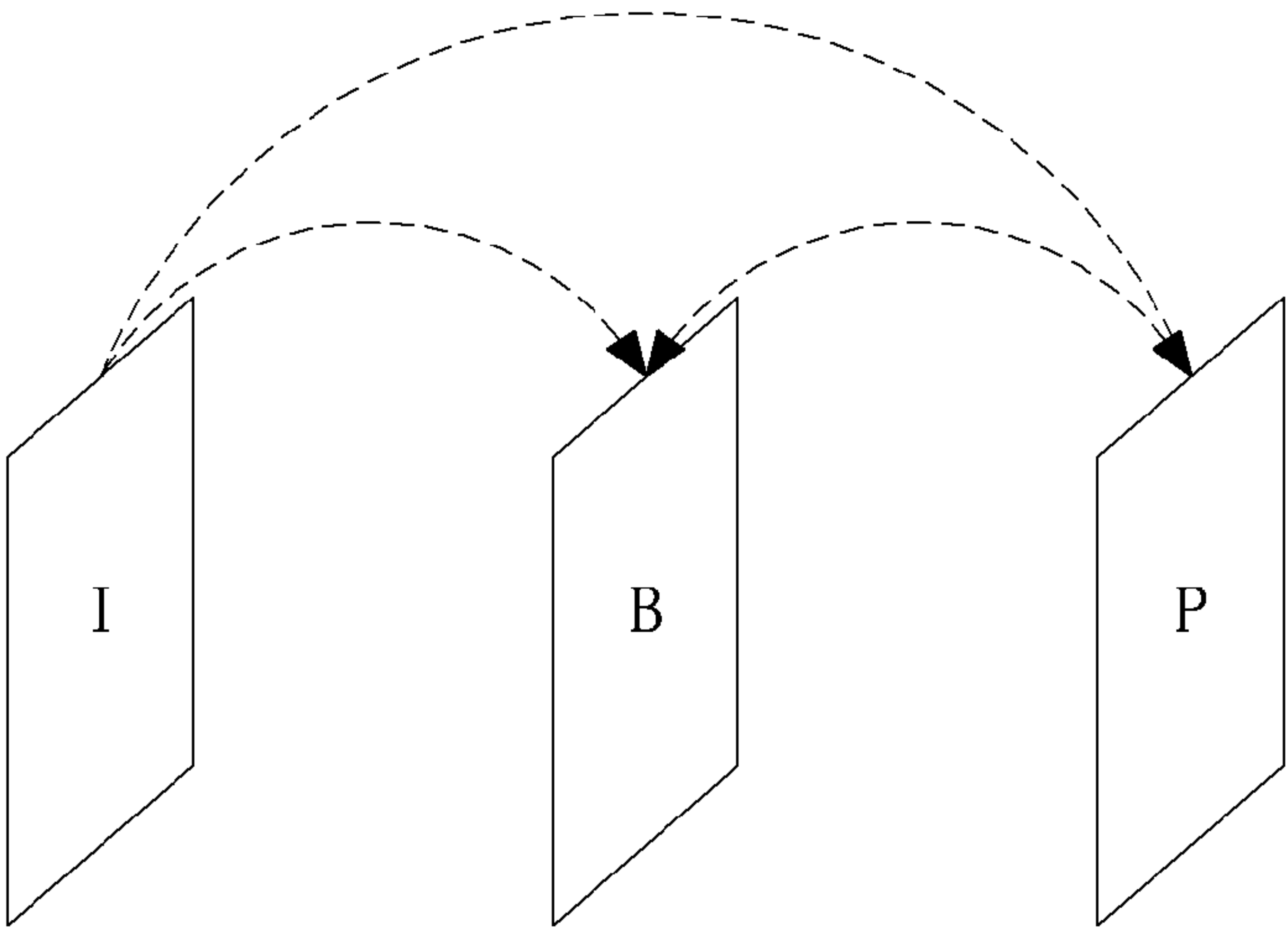
[도3]



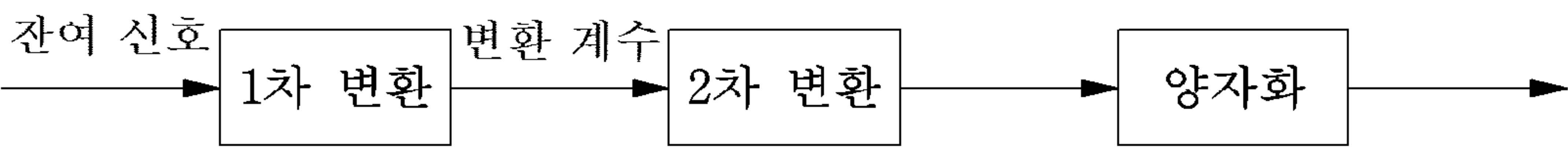
[도4]



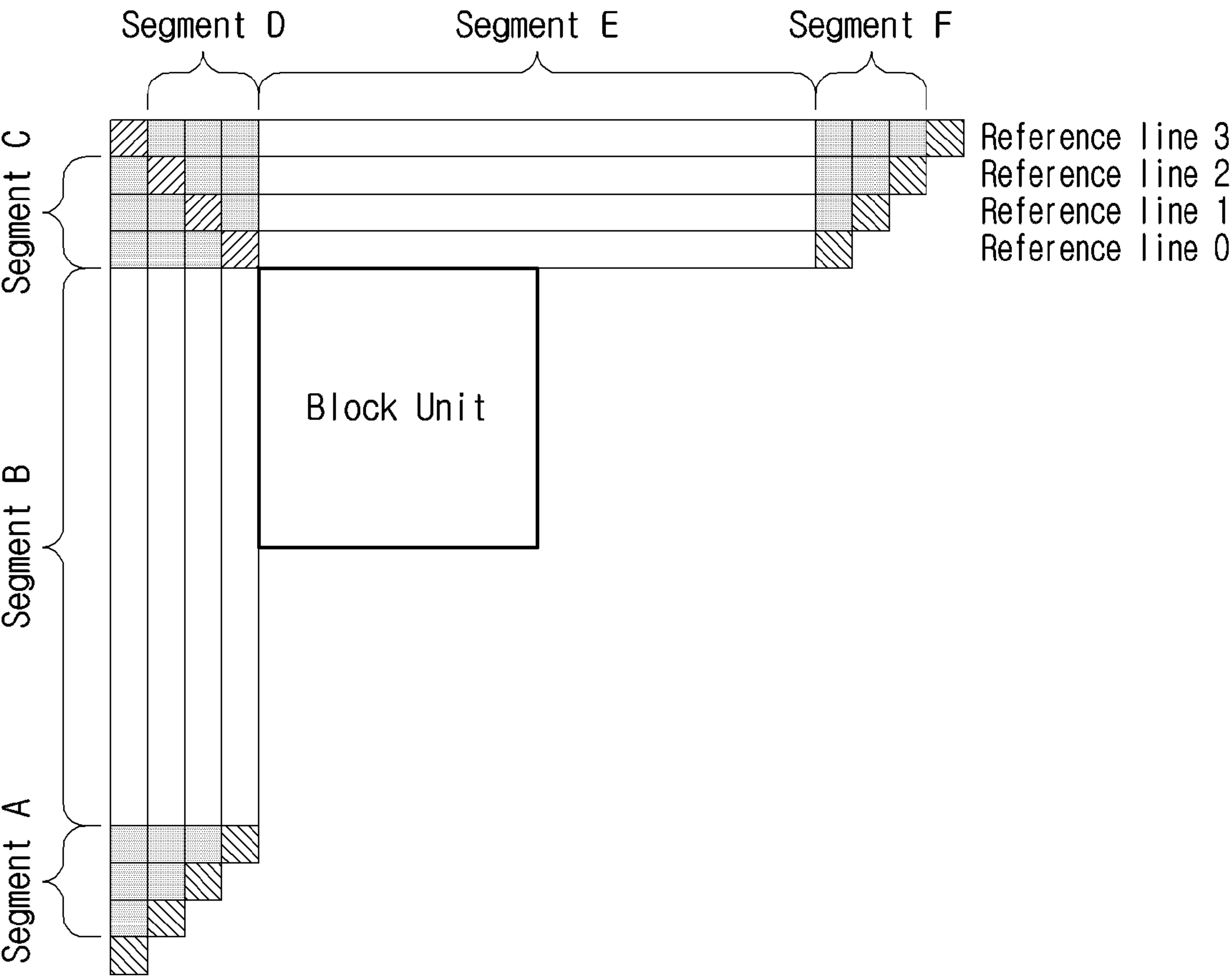
[도5]



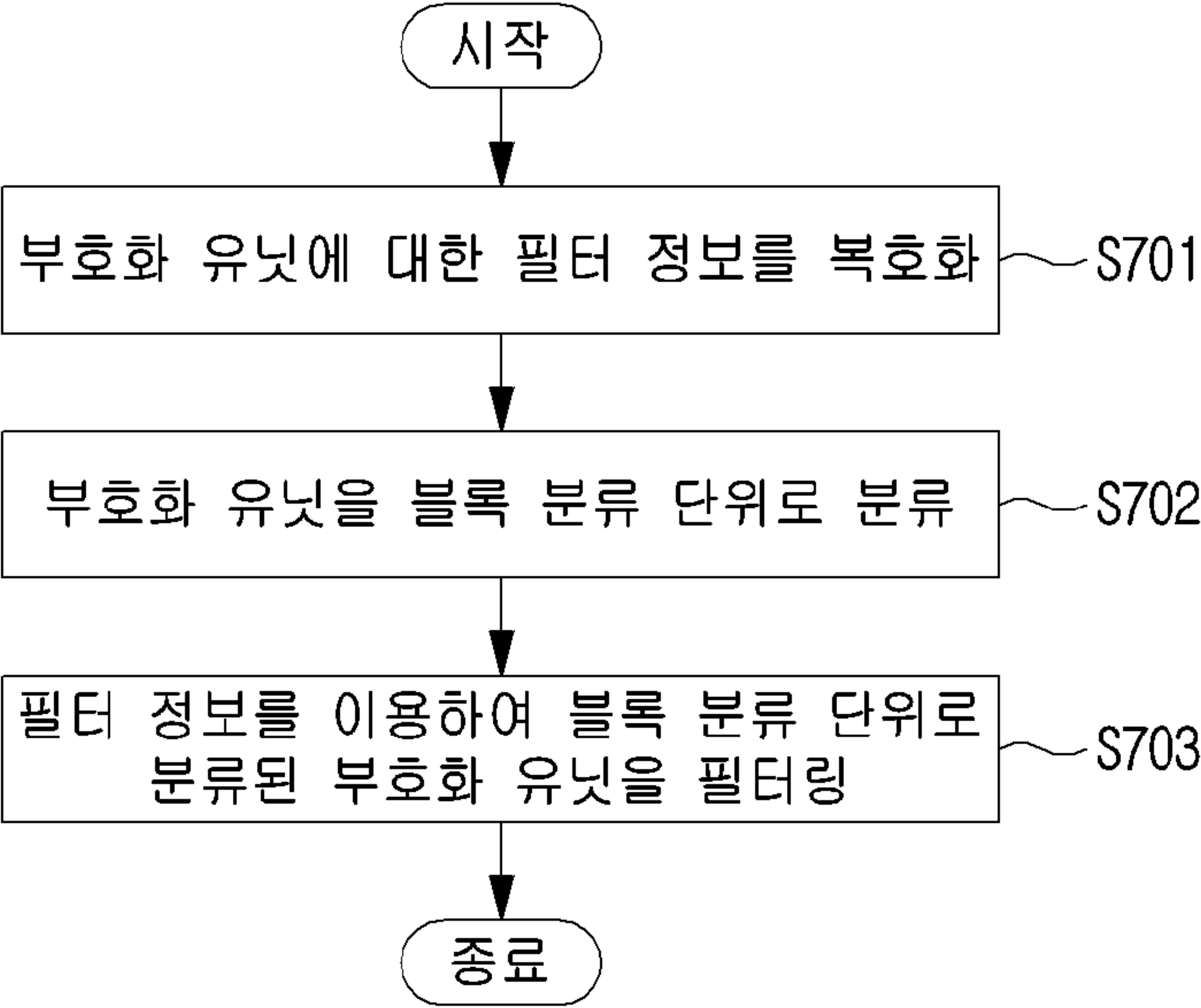
[도6]



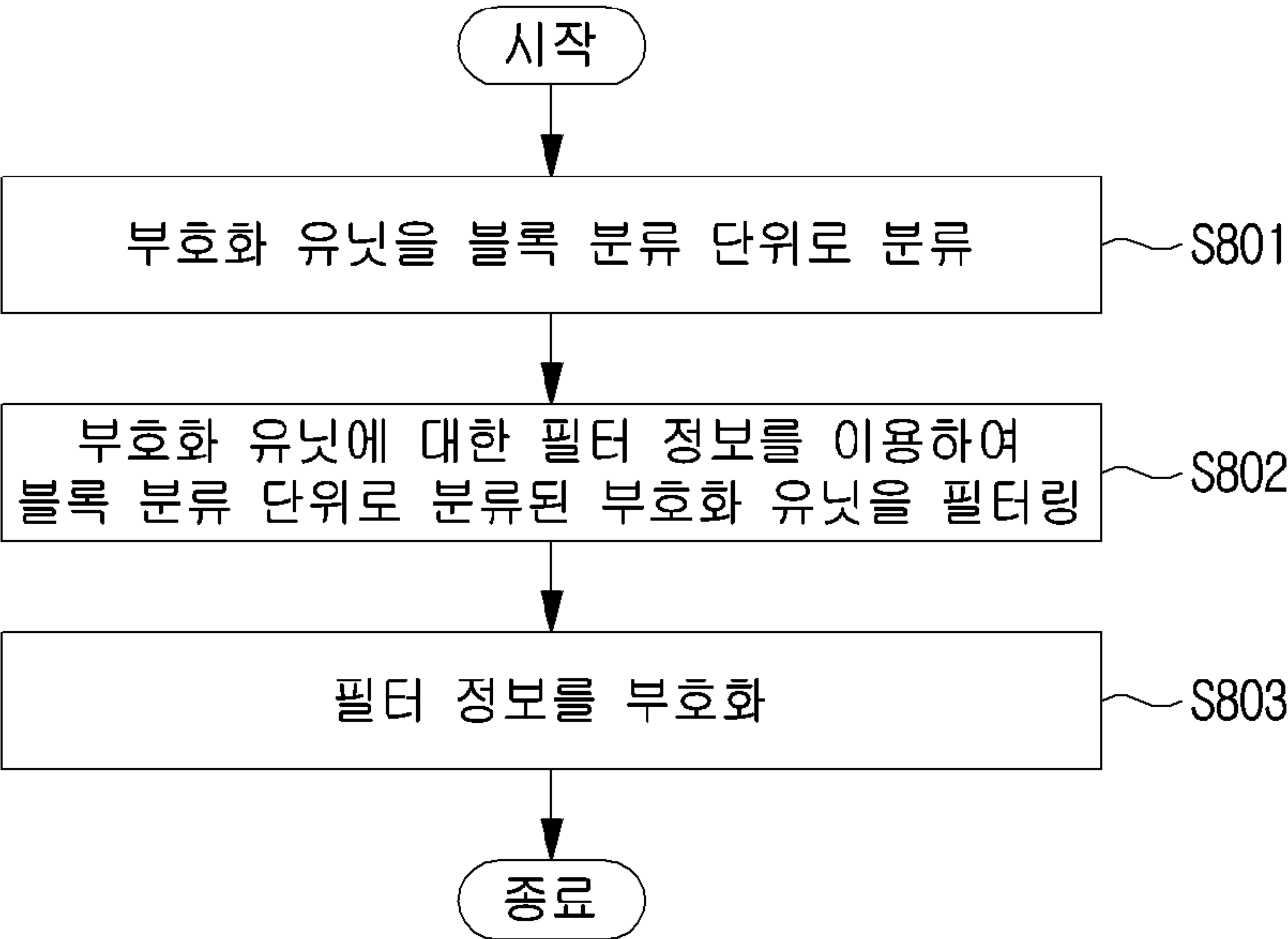
[도7]



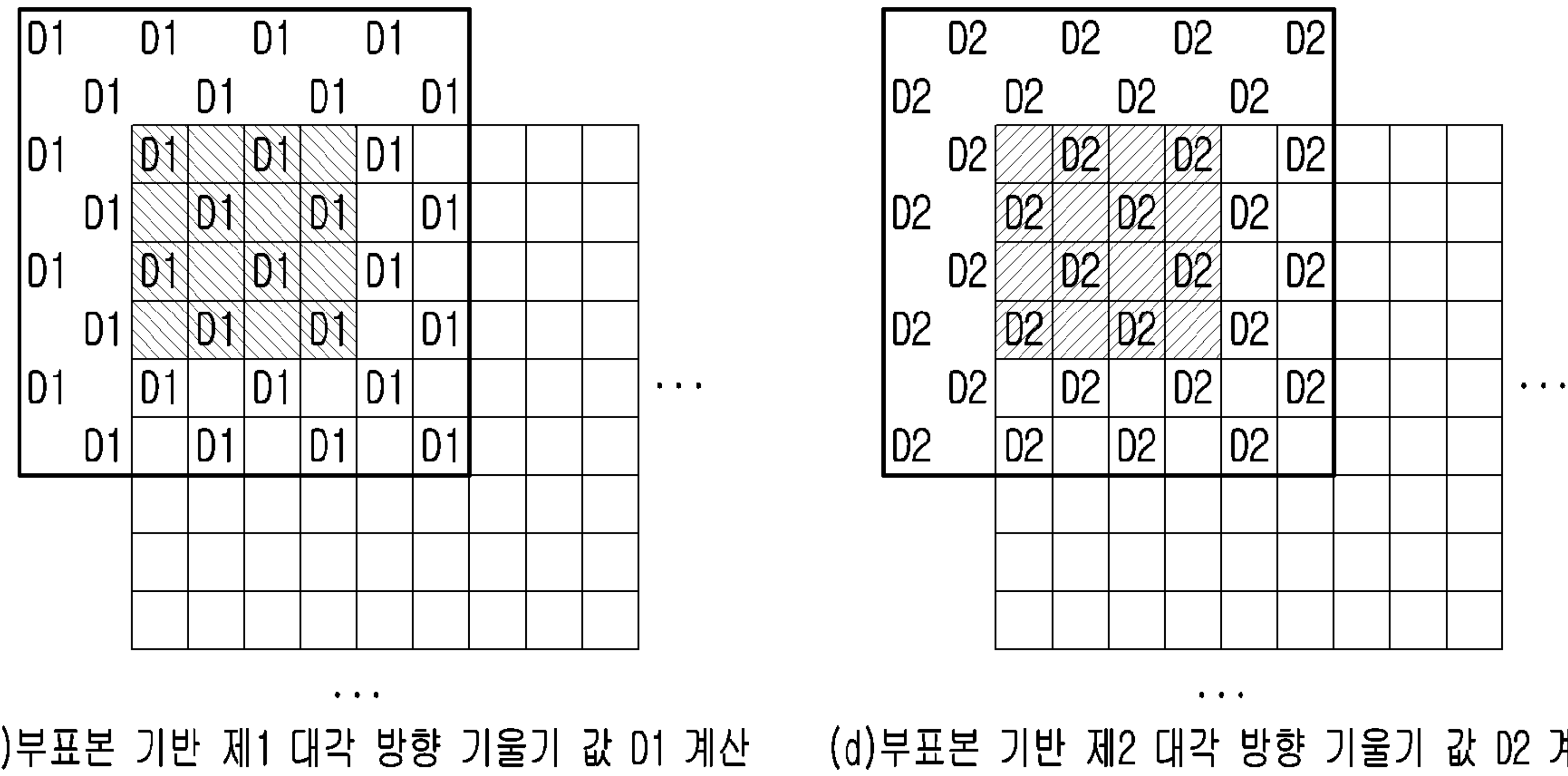
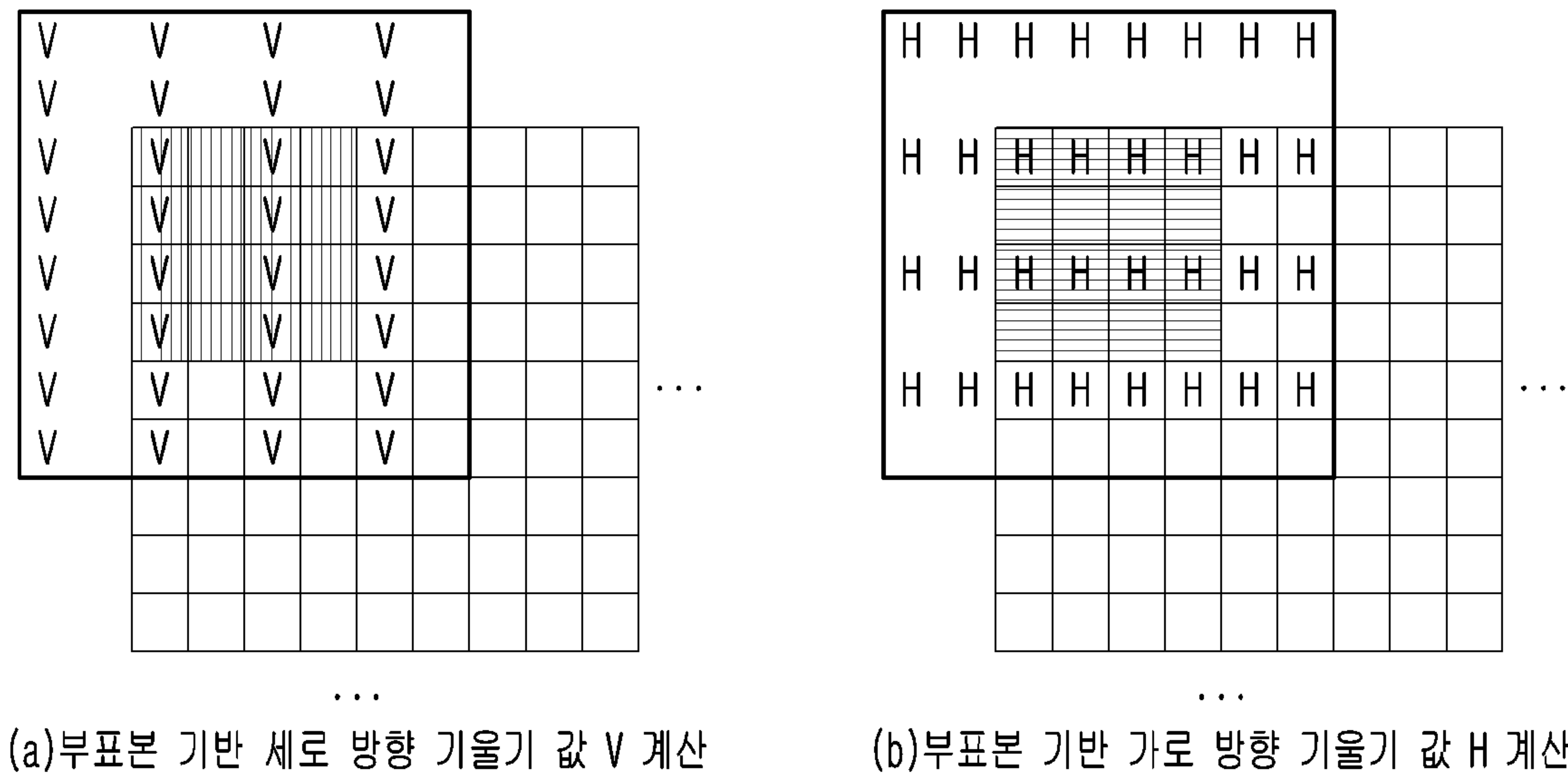
[도8a]



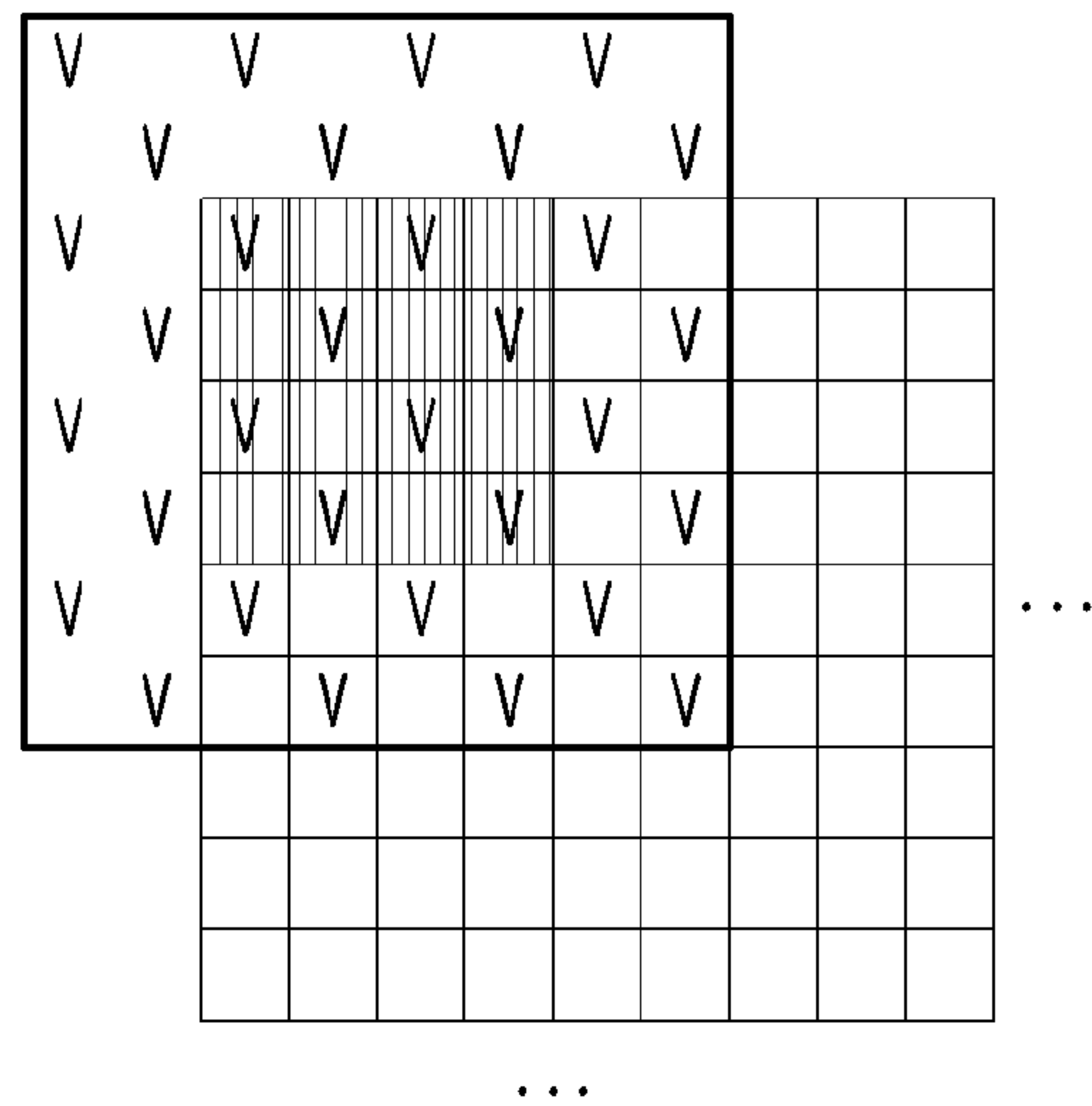
[도8b]



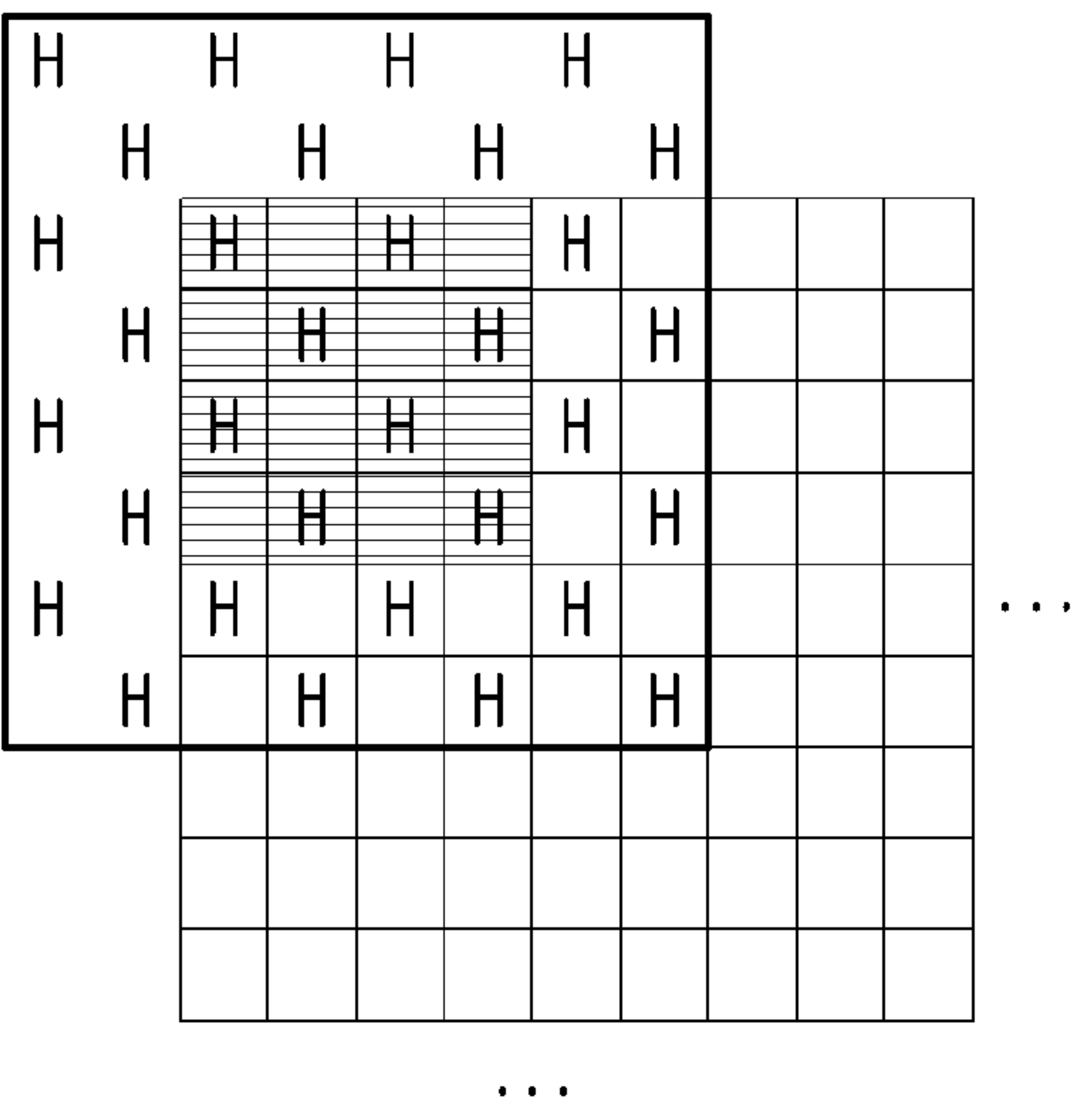
[도9]



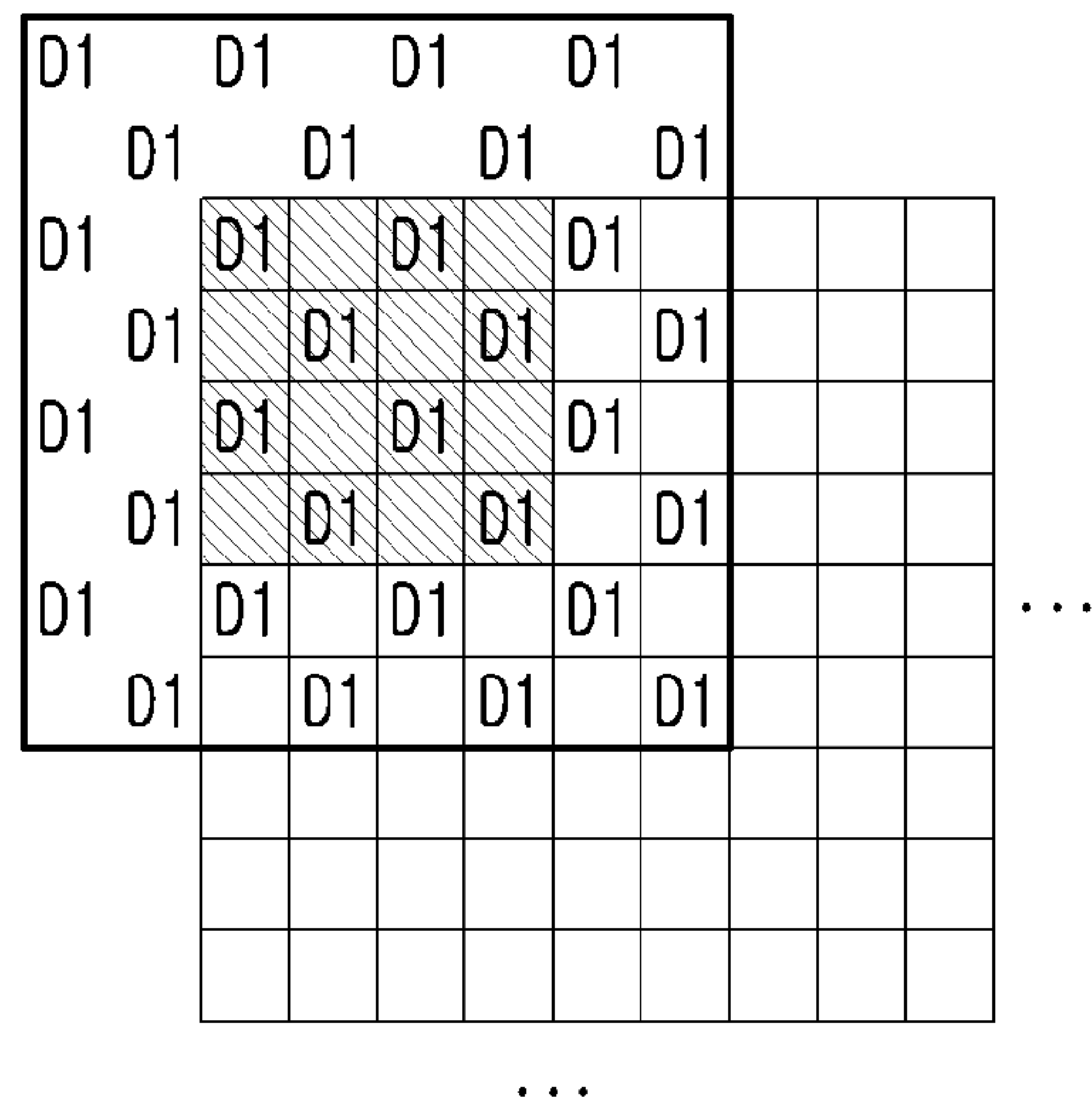
[도10]



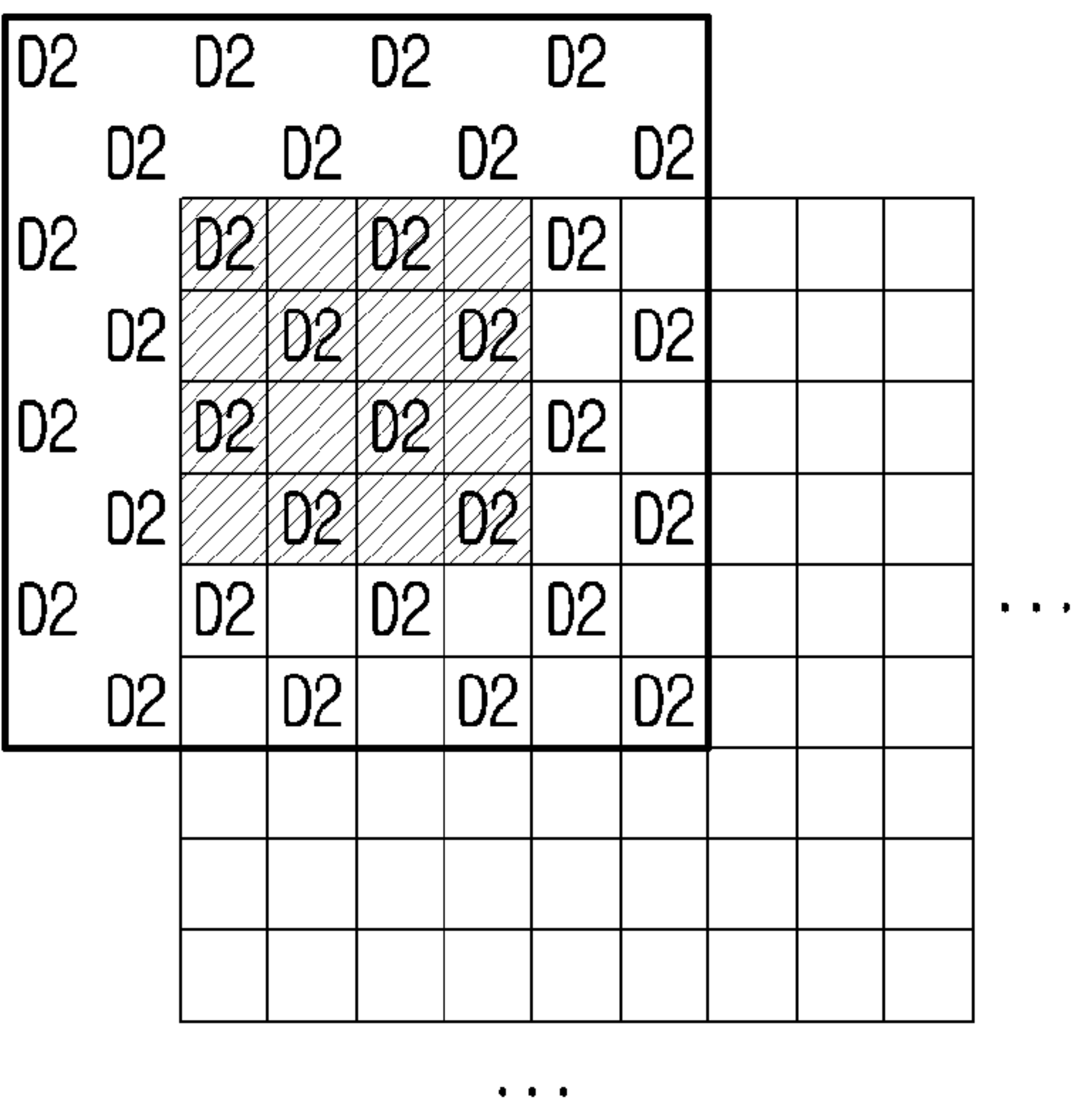
(a)부표본 기반 세로 방향 기울기 값 V 계산



(b)부표본 기반 가로 방향 기울기 값 H 계산

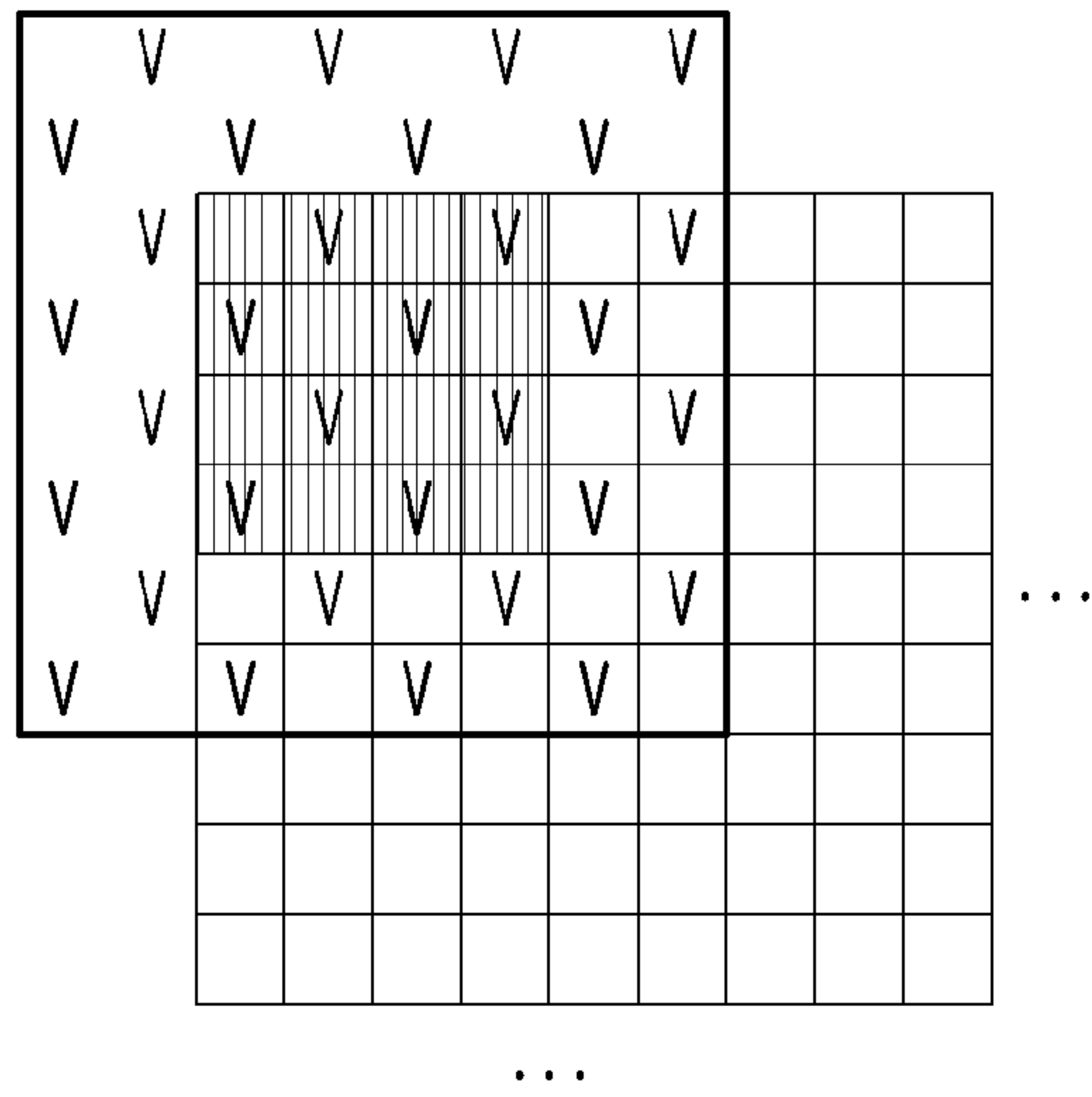


(c)부표본 기반 제1 대각 방향 기울기 값 D1 계산

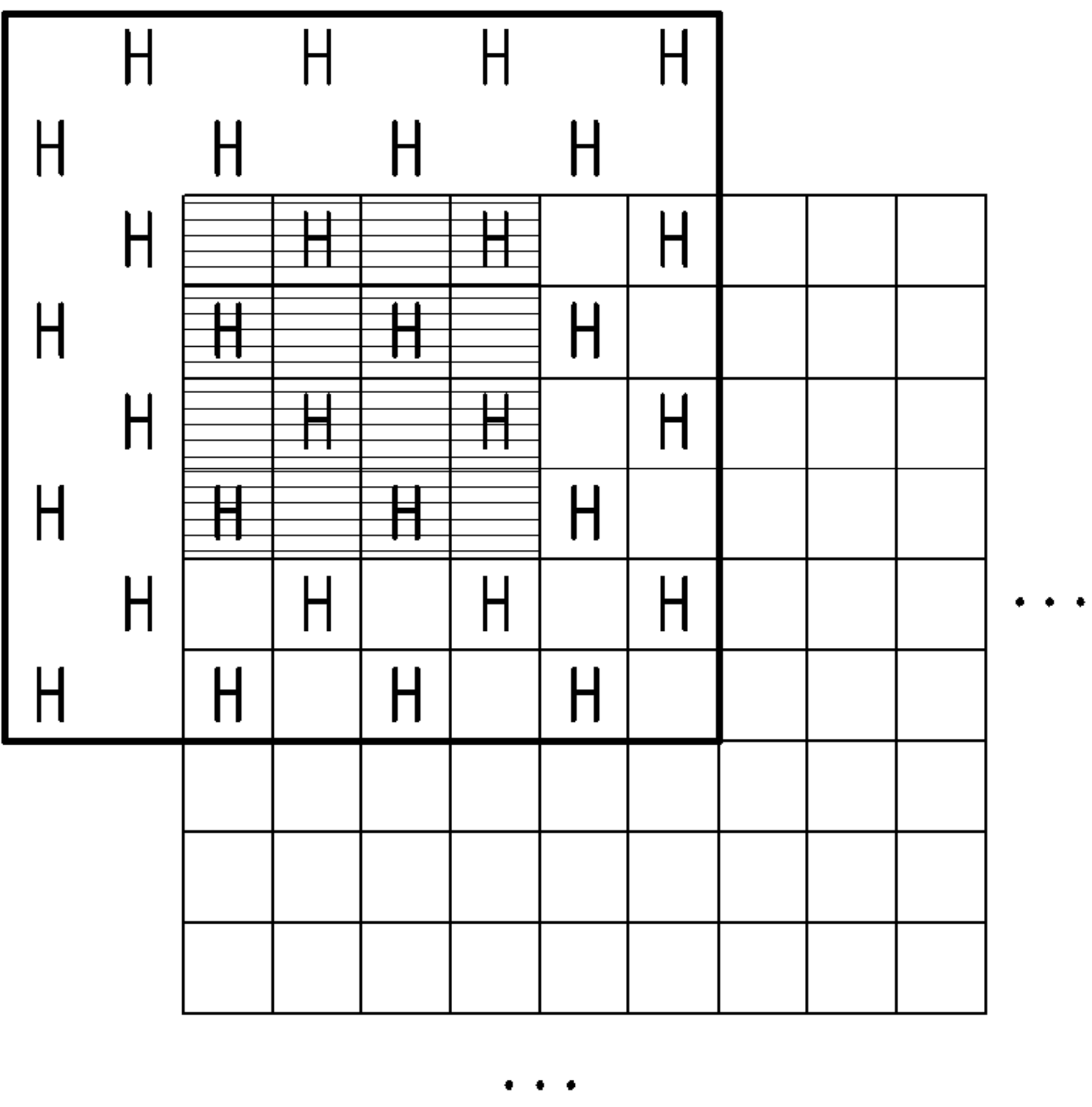


(d)부표본 기반 제2 대각 방향 기울기 값 D2 계산

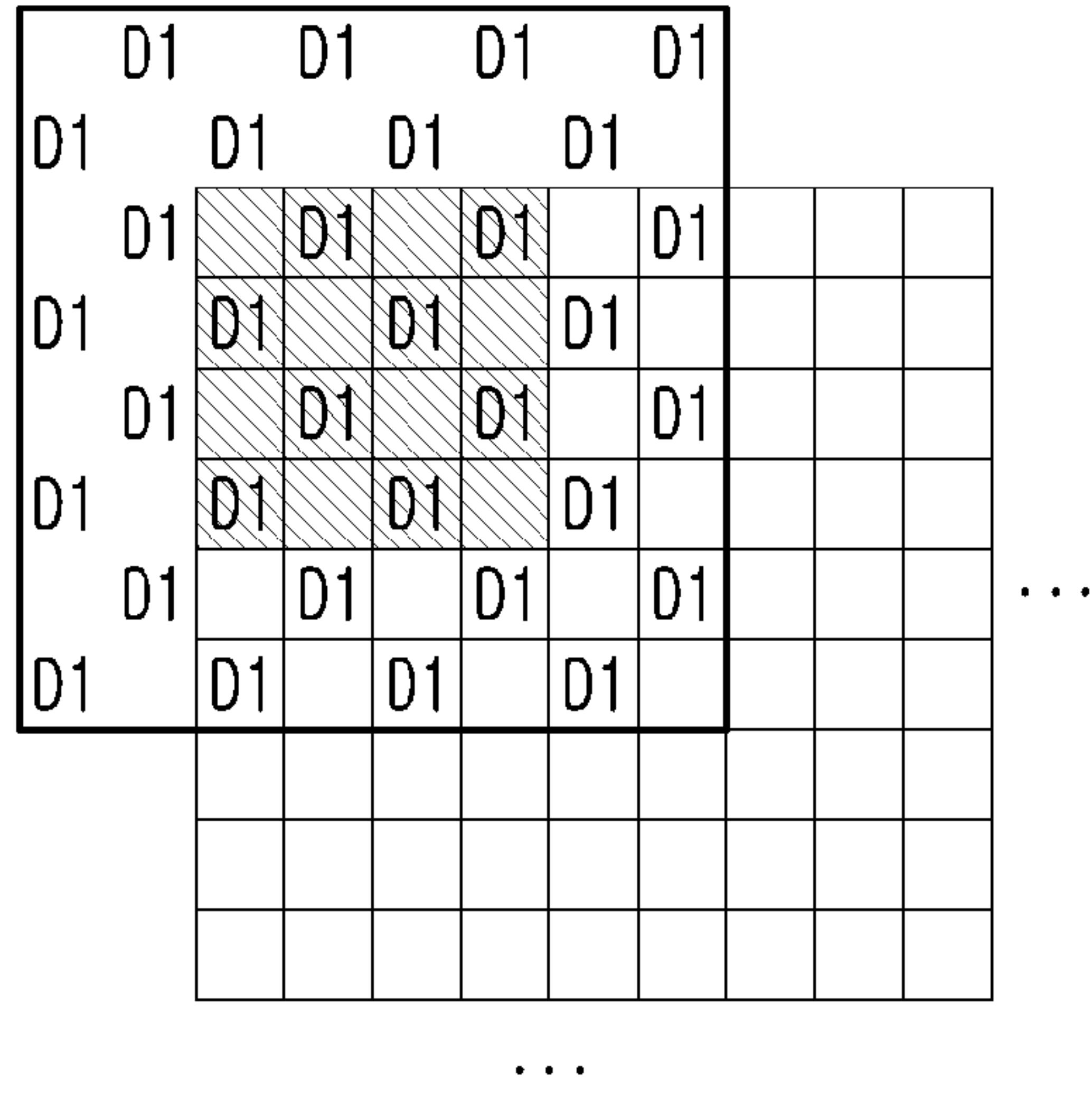
[도11]



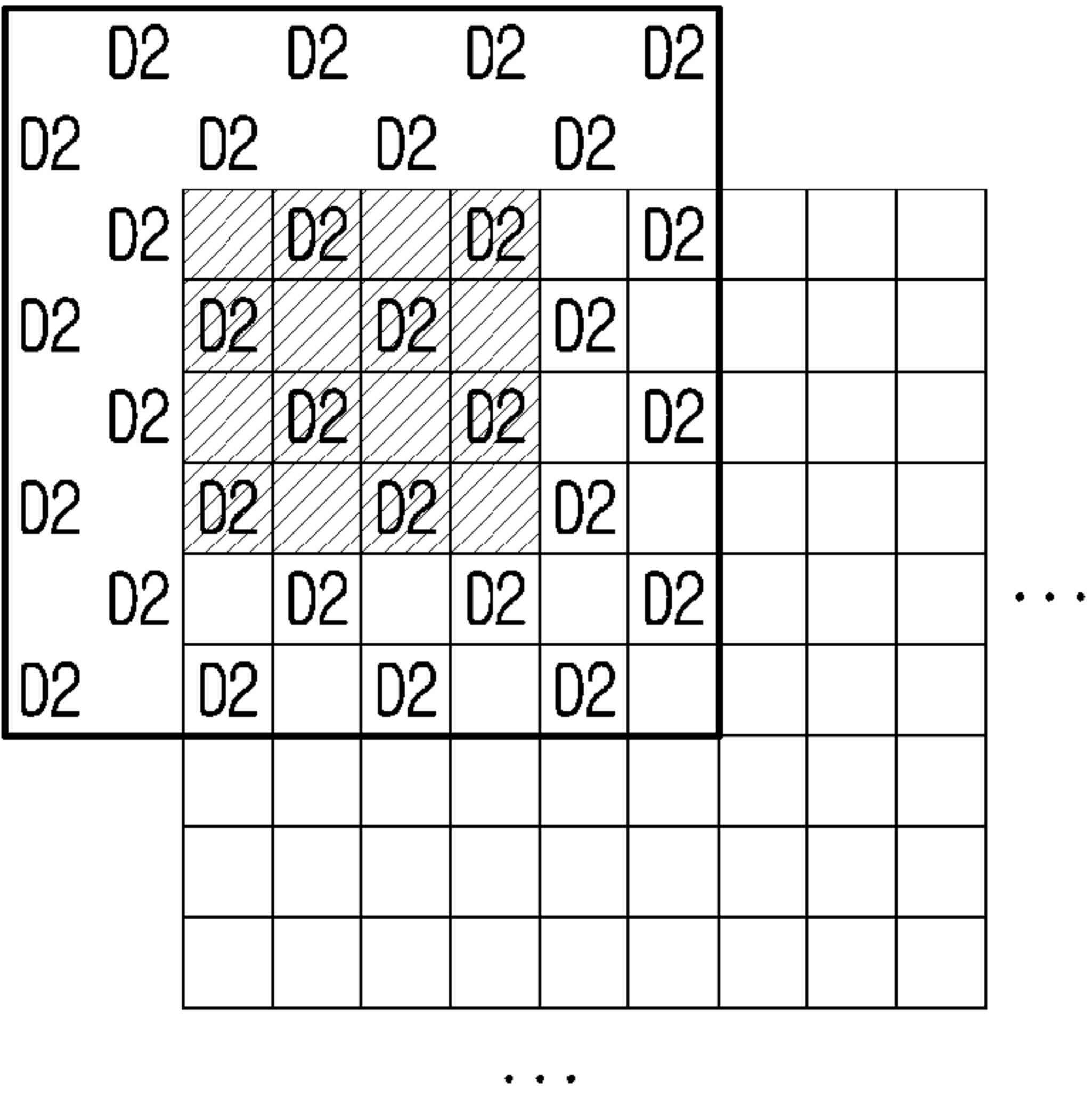
(a)부표본 기반 세로 방향 기울기 값 V 계산



(b)부표본 기반 가로 방향 기울기 값 H 계산

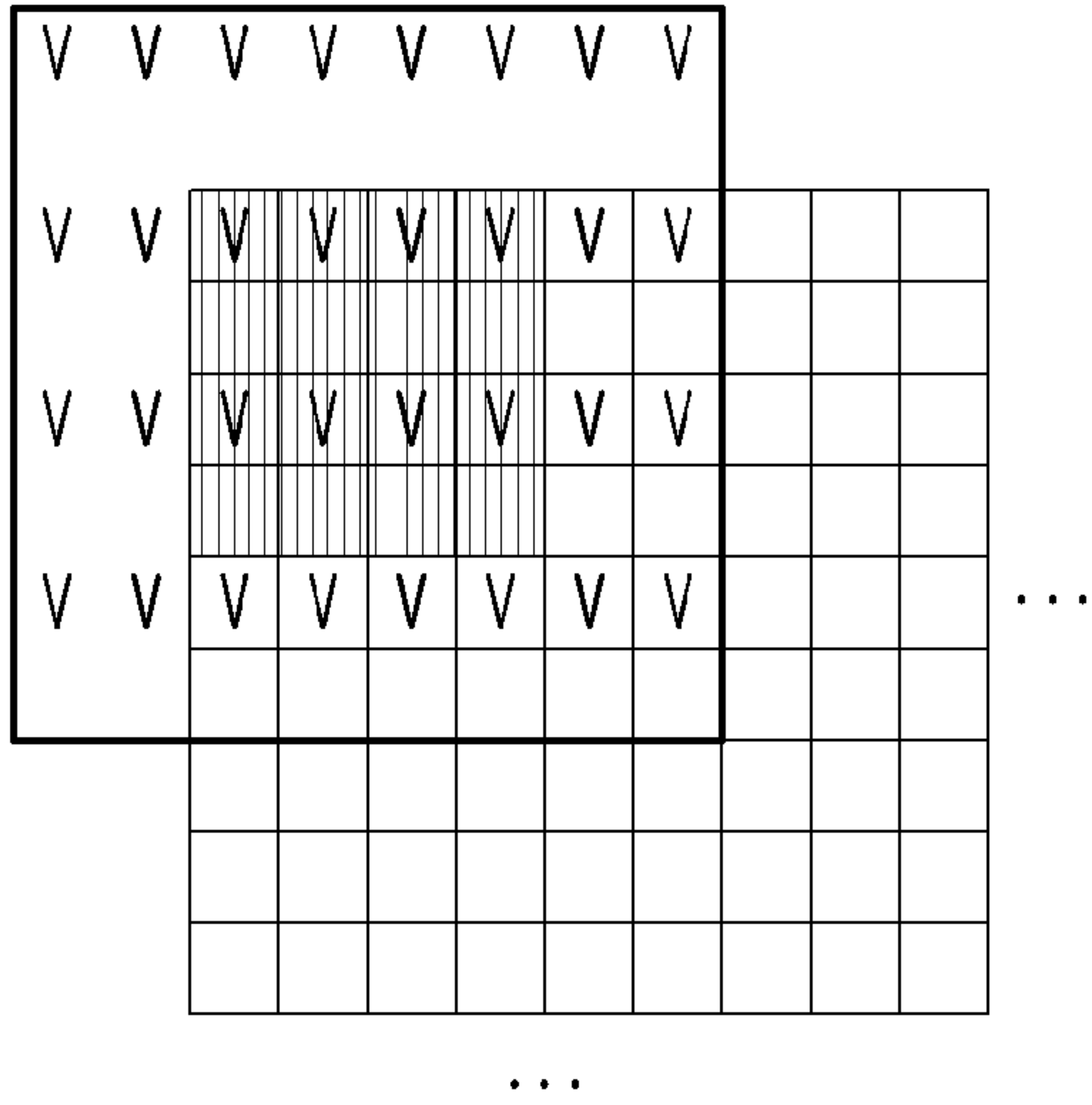


(c)부표본 기반 제1 대각 방향 기울기 값 D1 계산

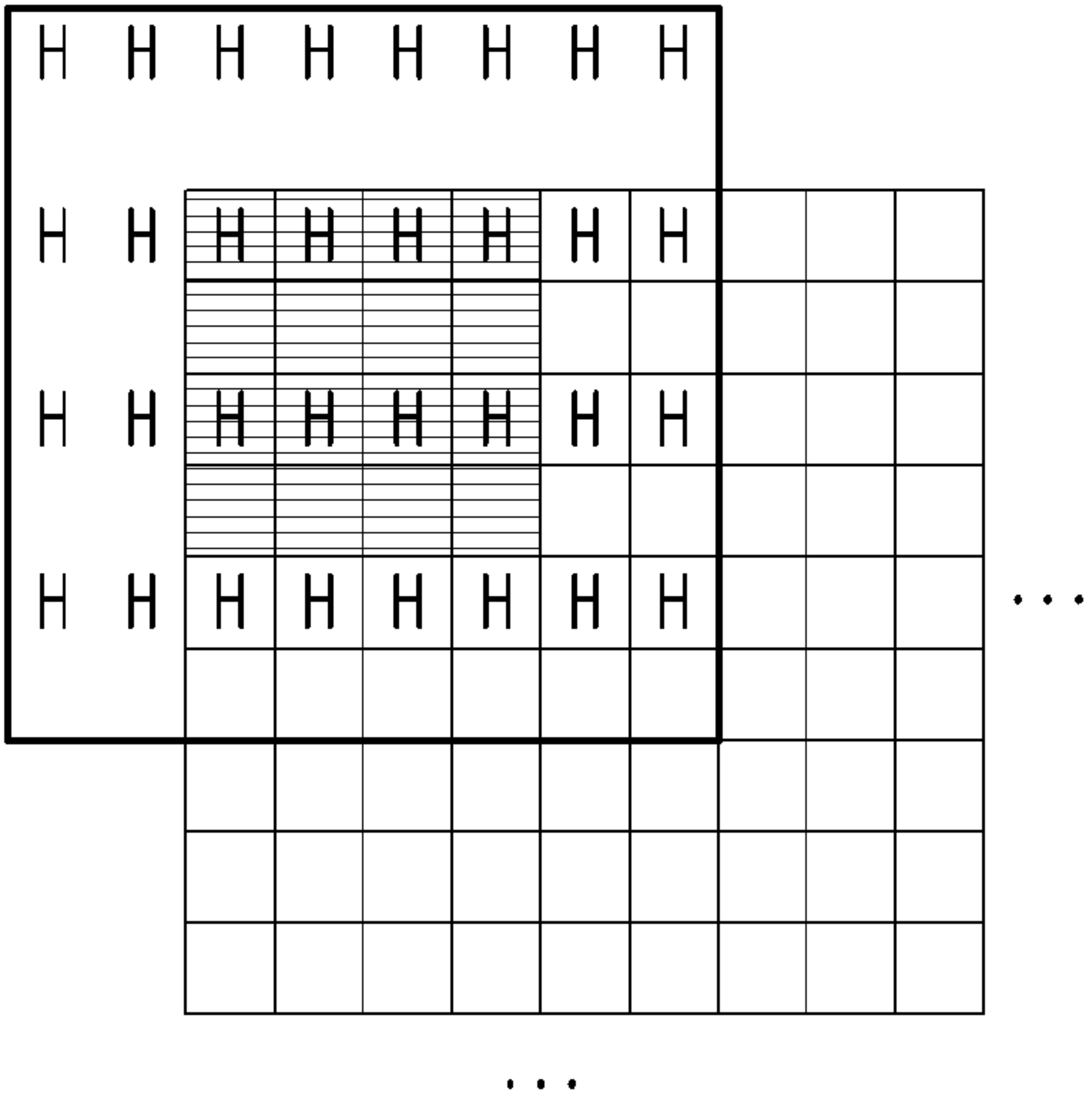


(d)부표본 기반 제2 대각 방향 기울기 값 D2 계산

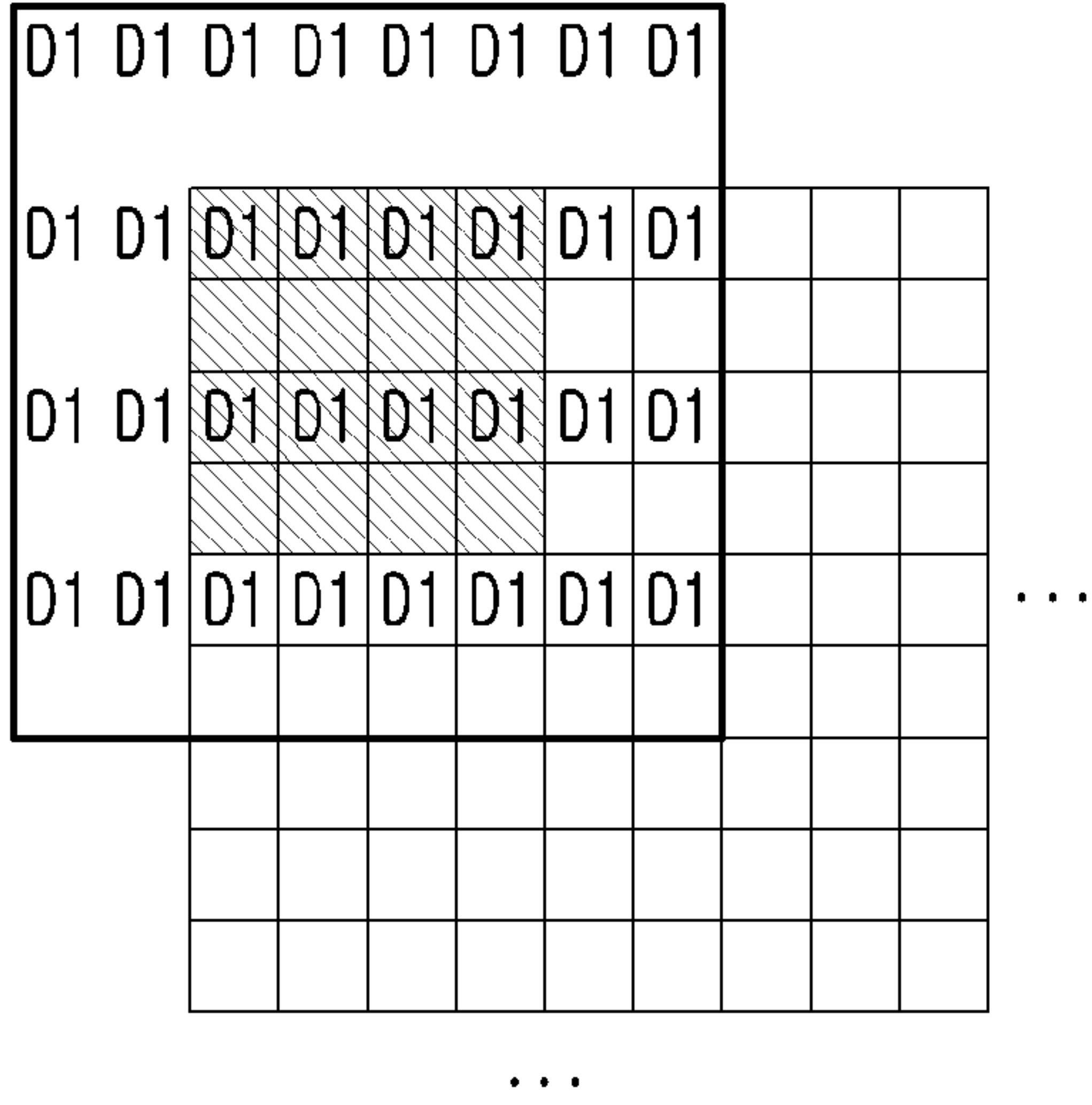
[도12]



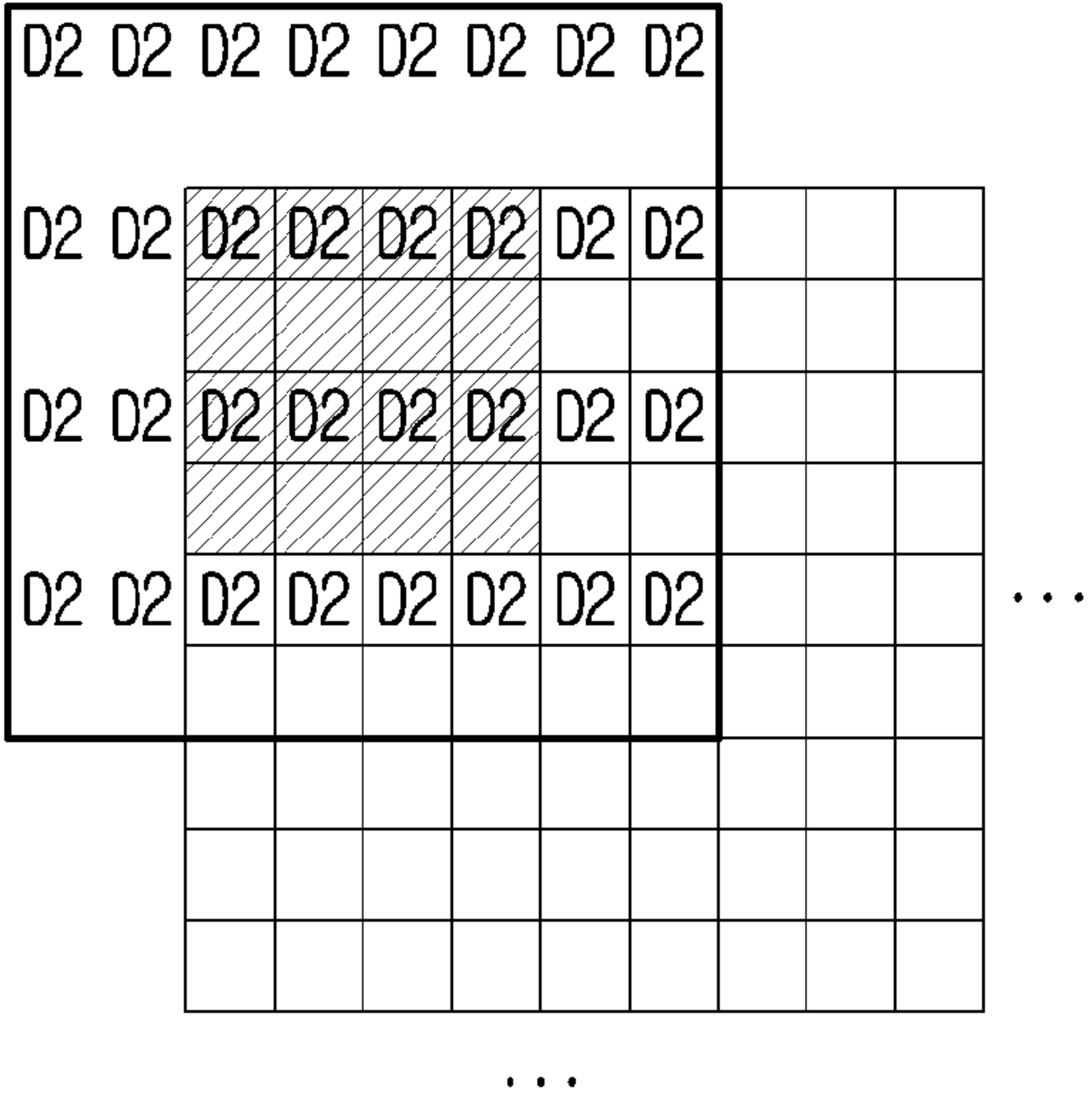
(a)부표본 기반 세로 방향 기울기 값 V 계산



(b)부표본 기반 가로 방향 기울기 값 H 계산

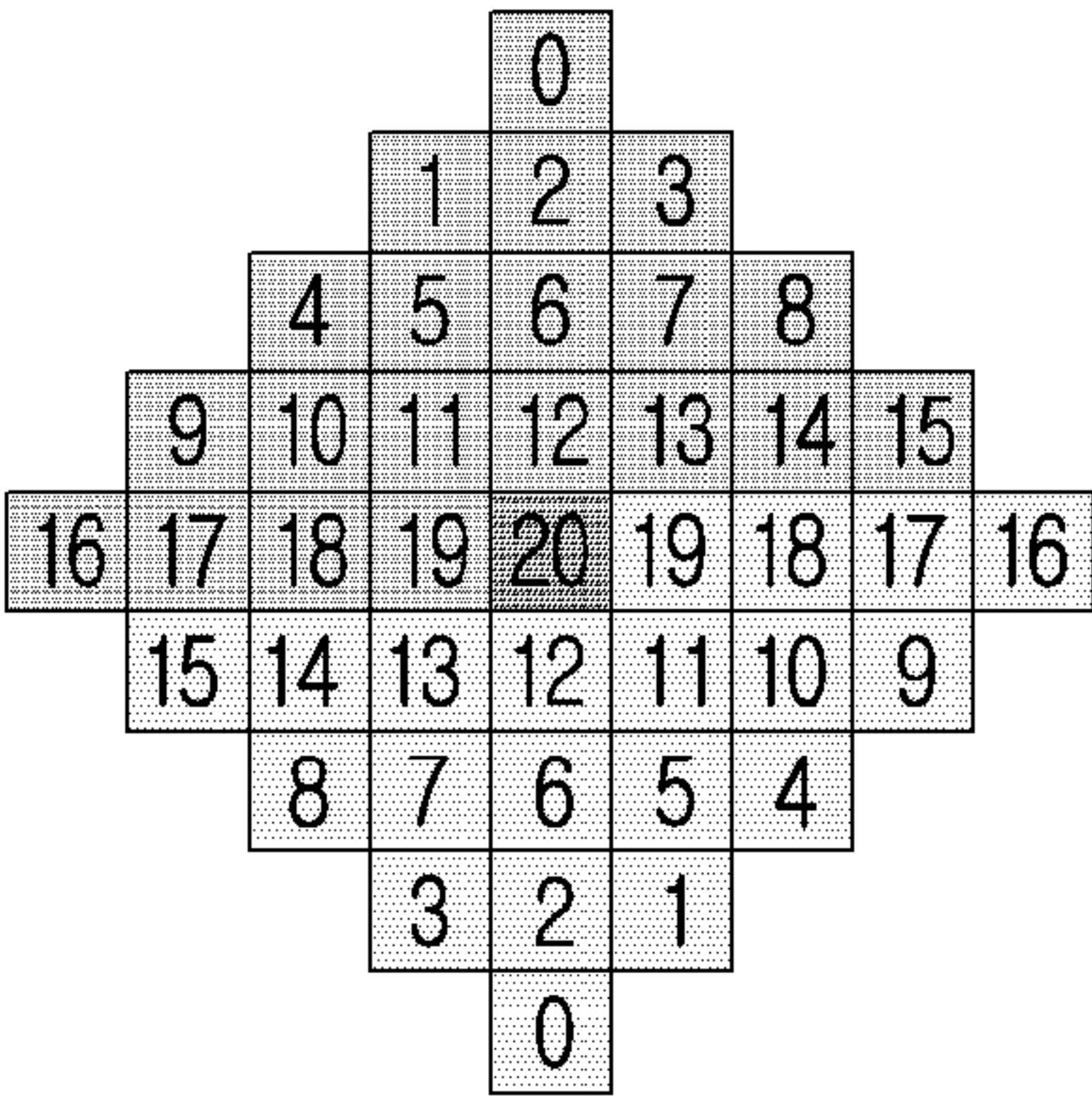


(c)부표본 기반 제1 대각 방향 기울기 값 D1 계산

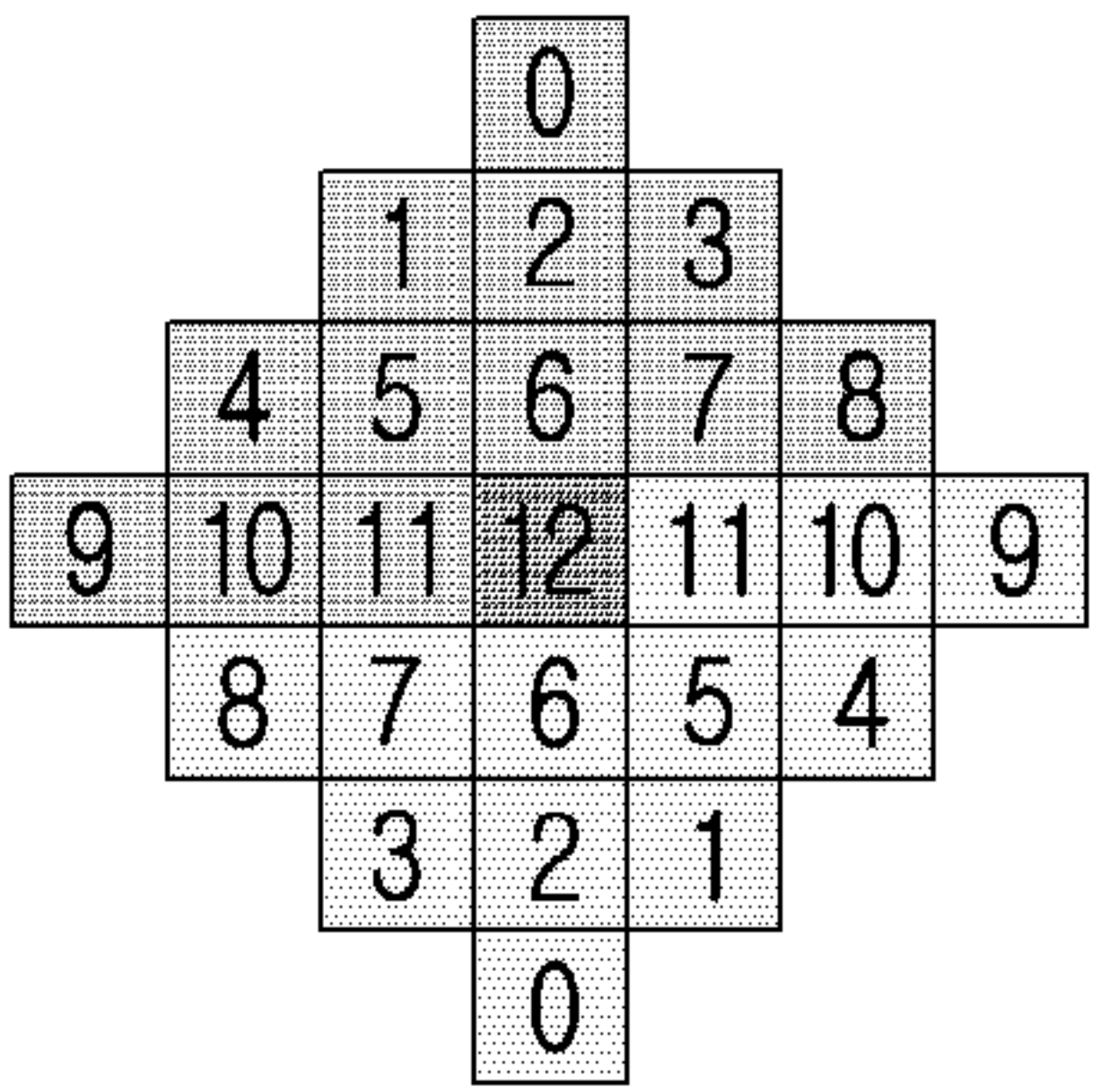


(d)부표본 기반 제2 대각 방향 기울기 값 D2 계산

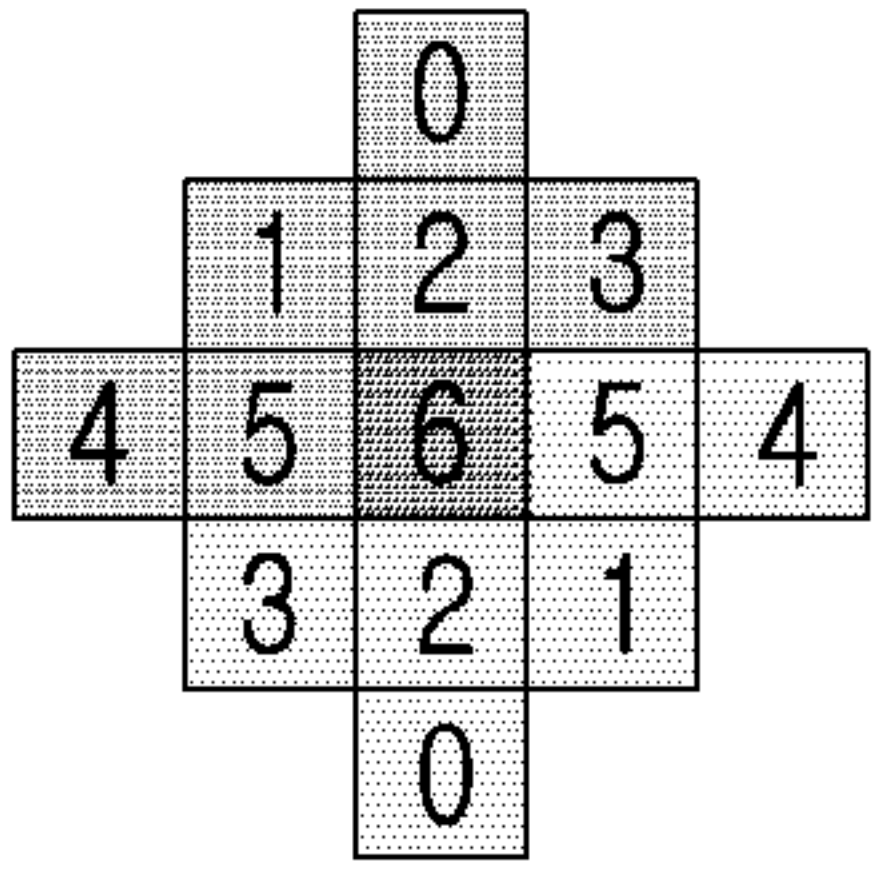
[도13]



(a) 9×9 마름모 형태 필터

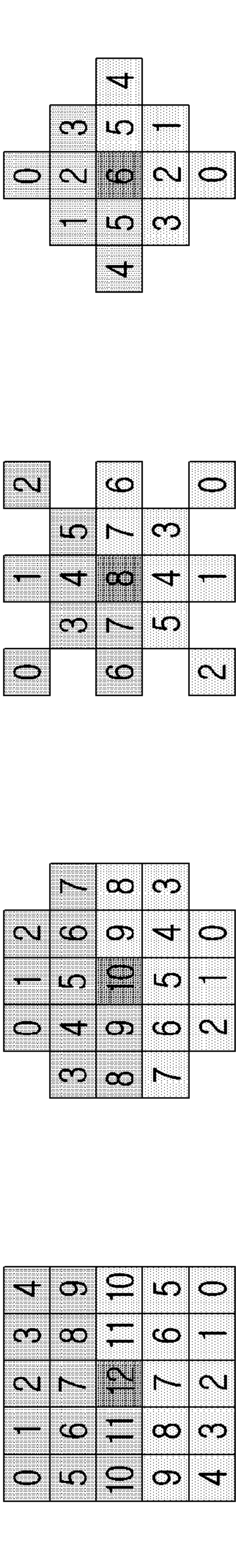


(b) 7×7 마름모 형태 필터

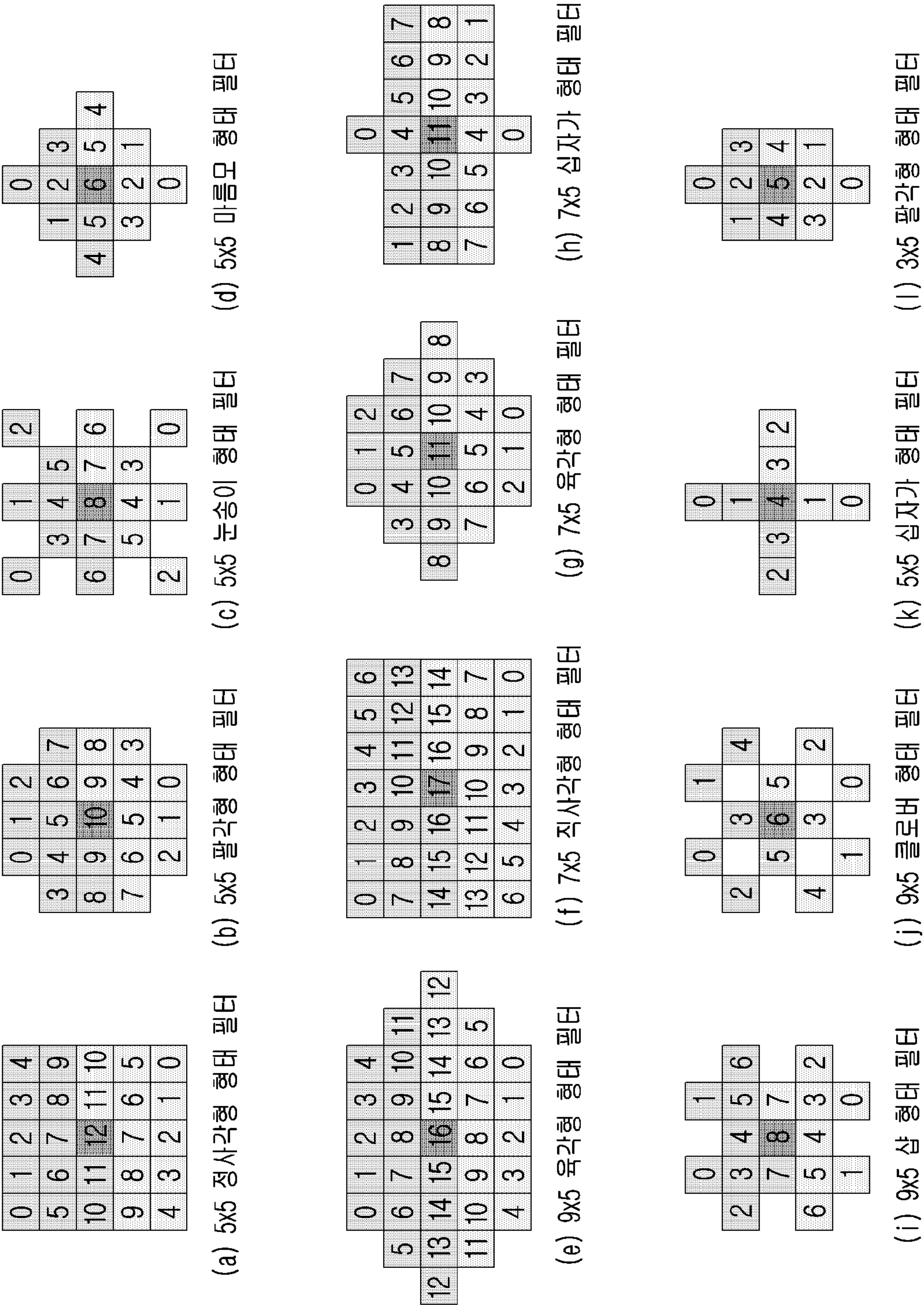


(c) 5×5 마름모 형태 필터

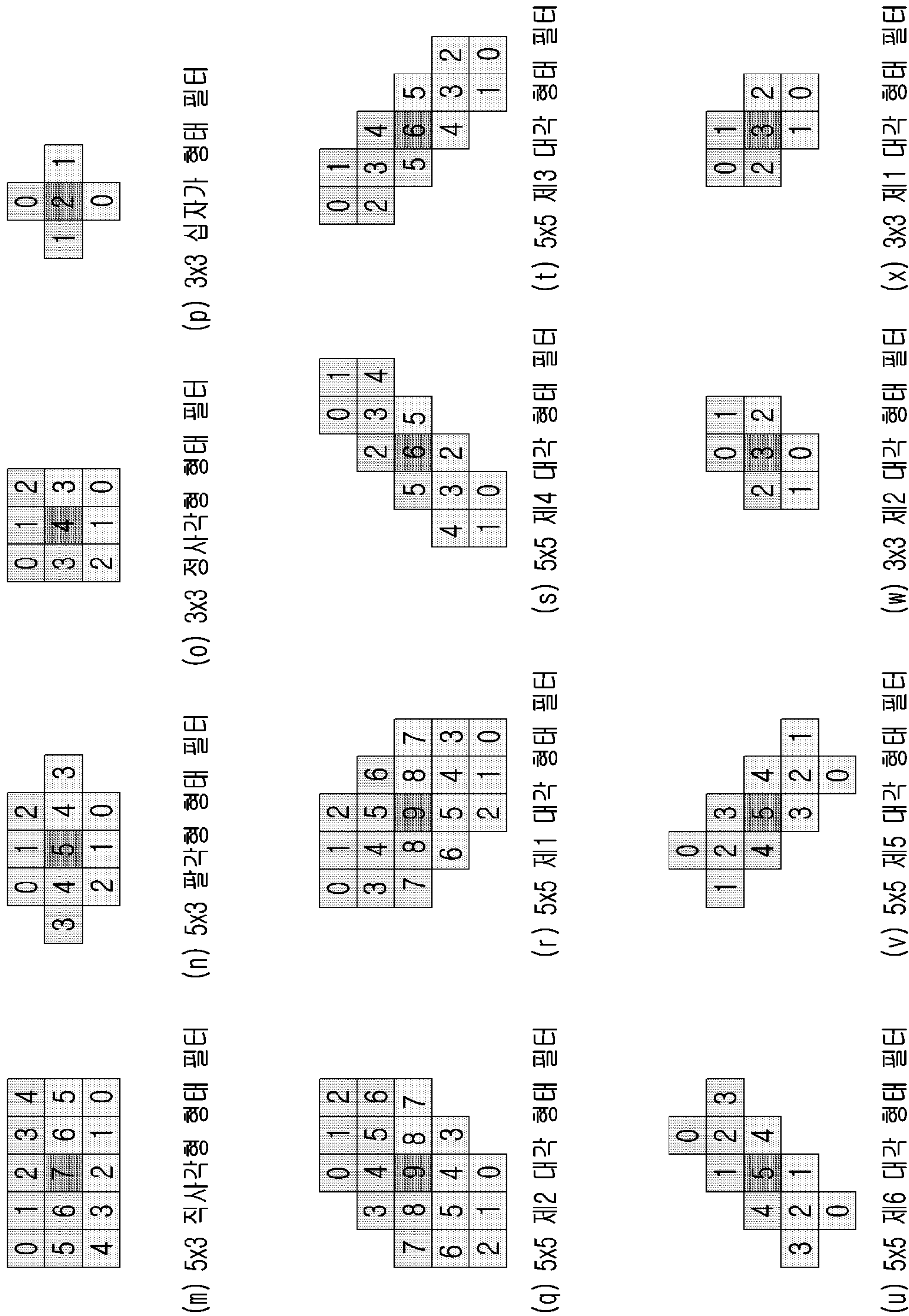
[도14]



[도15a]



[도15b]



[도16]

seq_parameter_set_rbsp() { sps_alf_enabled_flag	Descriptor u(1)
---	--------------------

[도17]

adaptation_parameter_set_rbsp() { adaptation_parameter_set_id	Descriptor u(5)
aps_params_type	u(3)
if(aps_params_type == ALF_APS)	
alf_data(adaptation_parameter_set_id)	
else if(aps_params_type == LMCS_APS)	
lmcs_data()	
}	

[도18]

aps_params_type	aps_params_type 이름	적용 파라미터의 타입
0	ALF_APS	ALF 파라미터 (적응적 루프내 필터링 파라미터)
1	LMCS_APS	LMCS 파라미터 (색차 스케일링 및 휘도 매핑 파라미터)
2..7	Reserved	Reserved

[도19]

slice header() {	Descriptor
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	tb(v)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(5)
slice_alf_chroma_idc	tu(v)
if(slice_alf_chroma_idc && (slice_type != I slice_num_alf_aps_ids_luma != 1))	
slice_alf_aps_id_chroma	u(5)
}	
}	

[도20a]

alf_data(adaptation_parameter_set_id) {	Descriptor
alf_luma_filter_signal_flag	u(1)
alf_chroma_filter_signal_flag	u(1)
if(alf_luma_filter_signal_flag) {	
alf_luma_clip_flag	u(1)
alf_luma_num_filters_signalled_minus1	tb(v)
if(alf_luma_num_filters_signalled_minus1 > 0) {	
for(filt_idx = 0; filt_idx < NumAlfFilters; filt_idx++)	
alf_luma_coeff_delta_idx[filt_idx]	tb(v)
}	
alf_luma_use_fixed_filter_flag	u(1)
if(alf_luma_use_fixed_filter_flag) {	
alf_luma_fixed_filter_set_idx	tb(v)
alf_luma_fixed_filter_pred_present_flag	u(1)
if(alf_luma_fixed_filter_pred_present_flag) {	
for(i = 0; i < NumAlfFilters; i++)	
alf_luma_fixed_filter_pred_flag[i]	u(1)
}	
}	

[도20b]

alf_luma_coeff_delta_flag	u(1)
if(alf_luma_coeff_delta_flag && alf_luma_num_filters_signalled_minus1 > 0)	
alf_luma_coeff_delta_prediction_flag	u(1)
alf_luma_min_eg_order_minus1	ue(v)
for(i = 0; i < 3; i++)	
alf_luma_eg_order_increase_flag[i]	u(1)
if(alf_luma_coeff_delta_flag){	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++){	
alf_luma_coeff_flag[sIdx]	u(1)
}	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++){	
if(alf_luma_coeff_flag[sIdx]){	
for(j = 0; j < 12; j++){	
alf_luma_coeff_delta_abs[sIdx][j]	uek(v)
if(alf_luma_coeff_delta_abs[sIdx][j]){	
alf_luma_coeff_delta_sign[sIdx][j]	u(1)
}	
}	
}	

[도20c]

if(alf_luma_clip_flag){	
alf_luma_clip_min_eg_order_minus1	ue(v)
for(i = 0; i < 3; i++)	
alf_luma_clip_eg_order_increase_flag[i]	u(1)
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++){	
if(alf_luma_coeff_flag[sIdx]){	
for(j = 0; j < 12; j++){	
if(filtCoeff[sIdx][j])	
alf_luma_clip_idx[sIdx][j]	uek(v)
}	
}	
}	
}	

[도20d]

if(alf_chroma_filter_signal_flag){	
alf_chroma_clip_flag	u(1)
alf_chroma_min_eg_order_minus1	us(v)
for(i=0; i<2; i++){	
alf_chroma_eg_order_increase_flag[i]	u(1)
for(j=0; j<6; j++){	
alf_chroma_coeff_abs[j]	uek(v)
if(alf_chroma_coeff_abs[j]>0)	
alf_chroma_coeff_sign[j]	u(1)
}	
}	
if(alf_chroma_clip_flag){	
alf_chroma_clip_min_eg_order_minus1	us(v)
for(i=0; i<2; i++){	
alf_chroma_clip_eg_order_increase_flag[i]	u(1)
for(j=0; j<6; j++){	
if(alf_chroma_coeff_abs[j])	
alf_chroma_clip_idx[j]	uek(v)
}	
}	
}	

[도21]

coding tree unit() {	Descriptor
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
if(slice sao luma flag slice sao chroma flag)	
sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)	
if(slice alf enabled flag){	
alf_ctb_flag[0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_ctb_flag[0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]) {	
if(slice num alf aps ids luma > 0)	
alf_ctb_use_first_aps_flag	ae(v)
if(!alf_ctb_use_first_aps_flag) {	
if(slice num alf aps ids luma > 1)	
alf_use_aps_flag	ae(v)
if(alf_use_aps_flag)	
alf_luma_prev_filter_idx_minus1	ae(v)
else	
alf_luma_fixed_filter_idx	ae(v)
}	
}	
if(slice alf chroma idc == 1 slice alf chroma idc == 3)	
alf_ctb_flag[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(slice alf chroma idc == 2 slice alf chroma idc == 3)	
alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
}	
if(slice type == I && qtbtt_dual_tree_intra_flag)	
dual_tree_implicit_qt_split(xCtb, yCtb, CtbSizeY, 0)	
else	
coding_tree(xCtb, yCtb, CtbSizeY, CtbSizeY, 1, 0, 0, 0, 0, 0, SINGLE_TREE)	
}	

[도22a]

AltFixFiltCoeff =

{

{ 0,	0,	2,	-3,	1,	-4,	1,	7,	-1,	1,	-1,	5}
{ 0,	0,	0,	0,	0,	-1,	0,	1,	0,	0,	-1,	2}
{ 0,	0,	0,	0,	0,	0,	0,	1,	0,	0,	0,	0}
{ 0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	-1,	1}
{ 2,	2,	-7,	-3,	0,	-5,	13,	22,	12,	-3,	-3,	17}
{-1,	0,	6,	-8,	1,	-5,	1,	23,	0,	2,	-5,	10}
{ 0,	0,	-1,	-1,	0,	-1,	2,	1,	0,	0,	-1,	4}
{ 0,	0,	3,	-11,	1,	0,	-1,	35,	5,	2,	-9,	9}
{ 0,	0,	8,	-8,	-2,	-7,	4,	4,	2,	1,	-1,	25}
{ 0,	0,	1,	-1,	0,	-3,	1,	3,	-1,	1,	-1,	8}
{ 0,	0,	3,	-3,	0,	-6,	5,	-1,	2,	1,	-4,	21}
{-7,	1,	5,	4,	-3,	5,	11,	13,	12,	-8,	11,	12}
{-5,	-3,	6,	-2,	-3,	8,	14,	15,	2,	-7,	11,	16}
{ 2,	-1,	-6,	-5,	-2,	-2,	20,	14,	-4,	0,	-3,	25}
{ 3,	1,	-8,	-4,	0,	-8,	22,	5,	-3,	2,	-10,	29}
{ 2,	1,	-7,	-1,	2,	-11,	23,	-5,	0,	2,	-10,	29}
{-6,	-3,	8,	9,	-4,	8,	9,	7,	14,	-2,	8,	9}
{ 2,	1,	-4,	-7,	0,	-8,	17,	22,	1,	-1,	-6,	23}
{ 3,	0,	-5,	-7,	0,	-7,	15,	18,	-5,	0,	-5,	27}
{ 2,	0,	0,	-7,	1,	-10,	13,	13,	-4,	2,	-7,	24}
{ 3,	3,	-13,	4,	-2,	-5,	9,	21,	25,	-2,	-3,	12}
{-5,	-2,	7,	-3,	-7,	9,	8,	9,	16,	-2,	15,	12}
{ 0,	-1,	0,	-7,	-5,	4,	11,	11,	8,	-6,	12,	22}
{ 3,	-2,	-3,	-8,	-4,	-1,	16,	15,	-2,	-3,	3,	26}
{ 2,	1,	-6,	-4,	-1,	-8,	16,	4,	-2,	1,	-7,	33}
{ 2,	1,	-4,	-2,	1,	-10,	17,	-2,	0,	2,	-11,	33}
{ 1,	-2,	7,	-15,	-16,	10,	8,	8,	20,	11,	14,	11}
{ 2,	2,	3,	-13,	-13,	4,	8,	12,	2,	-3,	16,	24}
{ 1,	4,	0,	-7,	-8,	-4,	9,	9,	-2,	-2,	8,	29}
{ 1,	1,	2,	-4,	-1,	-6,	6,	9,	-1,	-1,	-3,	30}

[도22b]

```

{-7, 3, 2, 10, -2, 3, 7, 11, 19, -7, 8, 10}
{ 0, -2, -5, -3, -2, 4, 20, 15, -1, -3, -1, 22}
{ 3, -1, -8, -4, -1, -4, 22, 8, -4, 2, -8, 28}
{ 0, 3, -14, 3, 0, 1, 19, 17, 8, -3, -7, 20}
{ 0, 2, -1, -8, 3, -6, 5, 21, 1, 1, -9, 18}
{-4, -2, 8, 20, -2, 2, 3, 5, 21, 4, 6, 1}
{ 2, -2, -3, -9, -4, 2, 14, 16, 3, -6, 8, 24}
{ 2, 1, 5, -16, -7, 2, 3, 11, 15, -3, 11, 22}
{ 1, 2, 3, -11, -2, -5, 4, 8, 9, -3, -2, 26}
{ 0, -1, 10, -9, -1, -8, 2, 3, 4, 0, 0, 29}
{ 1, 2, 0, -5, 1, -9, 9, 3, 0, 1, -7, 20}
{-2, 8, -6, -4, 3, -9, -8, 45, 14, 2, -13, 7}
{ 1, -1, 16, -19, -8, -4, -3, 2, 19, 0, 4, 30}
{ 1, 1, -3, 0, 2, -11, 15, -5, 1, 2, -9, 24}
{ 0, 1, -2, 0, 1, -4, 4, 0, 0, 1, -4, 7}
{ 0, 1, 2, -5, 1, -6, 4, 10, -2, 1, -4, 10}
{ 3, 0, -3, -6, -2, -6, 14, 8, -1, -1, -3, 31}
{ 0, 1, 0, -2, 1, -6, 5, 1, 0, 1, -5, 13}
{ 3, 1, 9, -19, -21, 9, 7, 6, 13, 5, 15, 21}
{ 2, 4, 3, -12, -13, 1, 7, 8, 3, 0, 12, 26}
{ 3, 1, -8, -2, 0, -6, 18, 2, -2, 3, -10, 23}
{ 1, 1, -4, -1, 1, -5, 8, 1, -1, 2, -5, 10}
{ 0, 1, -1, 0, 0, -2, 2, 0, 0, 1, -2, 3}
{ 1, 1, -2, -7, 1, -7, 14, 18, 0, 0, -7, 21}
{ 0, 1, 0, -2, 0, -7, 8, 1, -2, 0, -3, 24}
{ 0, 1, 1, -2, 2, -10, 10, 0, -2, 1, -7, 23}
{ 0, 2, 2, -11, 2, -4, -3, 39, 7, 1, -10, 9}
{ 1, 0, 13, -16, -5, -6, -1, 8, 6, 0, 6, 29}
{ 1, 3, 1, -6, -4, -7, 9, 6, -3, -2, 3, 33}
{ 4, 0, -17, -1, -1, 5, 26, 8, -2, 3, -15, 30}
{ 0, 1, -2, 0, 2, -8, 12, -6, 1, 1, -6, 16}
{ 0, 0, 0, -1, 1, -4, 4, 0, 0, 0, -3, 11}
{ 0, 1, 2, -8, 2, -6, 5, 15, 0, 2, -7, 9}
{ 1, -1, 12, -15, -7, -2, 3, 6, 6, -1, 7, 30}

```

},

[도23]

AllClassToFilterMap =

```

{
  { 8, 2, 2, 2, 3, 4, 52, 9, 9, 52, 4, 4, 5, 9, 2, 8, 10, 9, 1, 3, 39, 39, 10, 9, 52 },
  { 11, 12, 13, 14, 15, 30, 11, 17, 18, 19, 16, 20, 20, 4, 53, 21, 22, 23, 14, 25, 26, 26, 27, 28, 10 },
  { 16, 12, 31, 32, 14, 16, 30, 33, 53, 34, 35, 16, 20, 4, 7, 16, 21, 36, 18, 19, 21, 26, 37, 38, 39 },
  { 35, 11, 13, 14, 43, 35, 16, 4, 34, 62, 35, 35, 30, 56, 7, 35, 21, 38, 24, 40, 16, 21, 48, 57, 39 },
  { 11, 31, 32, 43, 44, 16, 4, 17, 34, 45, 30, 20, 20, 7, 5, 21, 22, 46, 40, 47, 26, 48, 53, 58, 10 },
  { 12, 13, 50, 51, 52, 11, 17, 53, 45, 9, 30, 4, 53, 19, 0, 22, 23, 25, 43, 44, 37, 27, 28, 10, 55 },
  { 30, 33, 62, 51, 44, 20, 42, 56, 34, 45, 20, 41, 41, 55, 5, 30, 56, 38, 40, 47, 21, 37, 42, 57, 8 },
  { 35, 11, 23, 32, 14, 35, 20, 4, 17, 18, 21, 20, 20, 20, 4, 16, 21, 36, 46, 25, 41, 26, 48, 49, 58 },
  { 22, 31, 59, 59, 3, 33, 32, 59, 59, 52, 4, 33, 17, 59, 56, 22, 36, 59, 59, 60, 22, 36, 59, 25, 56 },
  { 31, 28, 15, 60, 60, 22, 17, 19, 55, 55, 20, 20, 53, 19, 55, 22, 46, 25, 43, 60, 37, 28, 10, 55, 52 },
  { 12, 31, 32, 50, 51, 11, 33, 53, 19, 45, 16, 4, 4, 53, 5, 22, 36, 18, 25, 43, 26, 27, 27, 28, 10 },
  { 5, 2, 44, 52, 3, 4, 53, 45, 9, 3, 4, 56, 5, 0, 2, 5, 10, 47, 52, 3, 53, 39, 10, 9, 52 },
  { 12, 34, 44, 44, 3, 56, 56, 62, 45, 9, 56, 56, 7, 5, 0, 22, 38, 40, 47, 52, 48, 57, 39, 10, 9 },
  { 35, 11, 23, 14, 51, 35, 20, 41, 56, 62, 16, 20, 41, 56, 7, 16, 21, 38, 24, 40, 26, 26, 42, 57, 39 },
  { 33, 34, 51, 51, 52, 41, 41, 24, 62, 0, 41, 41, 56, 7, 5, 56, 38, 38, 40, 44, 37, 42, 57, 29, 10 },
  { 16, 31, 32, 15, 60, 30, 4, 17, 19, 25, 22, 20, 4, 53, 19, 21, 22, 46, 25, 55, 26, 48, 63, 58, 55 }
}

```

},

[도24]

alf_data(adaptation_parameter_set_id) {	Descriptor
...	
if(alf_luma_clip_flag) {	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++) {	
if(alf_luma_coeff_flag[sIdx]) {	
for(j = 0; j < 12; j++) {	
if(filtCoeff[sIdx][j])	
alf_luma_clip_idx[sIdx][j]	u(2)
}	
}	
}	
}	
}	
...	
}	
if(alf_chroma_clip_flag) {	
for(j = 0; j < 6; j++) {	
if(alf_chroma_coeff_abs[j])	
alf_chroma_clip_idx[j]	u(2)
}	
}	
...	

[도25]

7.4.6.3 Adaptive loop filter data semantics

...

alf_luma_clip_idx[sIdx][j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the signalled luma filter indicated by sIdx. When alf_luma_clip_idx[sIdx][j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of alf_luma_clip_idx[sIdx][j] with sIdx = 0..alf_luma_num_filters_signalled_minus1 and j = 0..11 shall be in the range of 0 to 3, inclusive.

...

alf_chroma_clip_idx[j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the chroma filter. When alf_chroma_clip_idx[j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of alf_chroma_clip_idx[j] with j = 0..5 shall be in the range of 0 to 3, inclusive.

...

[도26]

alf_data(adaptation_parameter_set_id) {	Descriptor
...	
if(alf_luma_clip_flag) {	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++) {	
if(alf_luma_coeff_flag[sIdx]) {	
for(j = 0; j < 12; j++) {	
if(filtCoeff[sIdx][j])	
alf_luma_clip_idx[sIdx][j]	tu(v)
}	
}	
}	
}	
}	
...	
}	
if(alf_chroma_clip_flag) {	
for(j = 0; j < 6; j++) {	
if(alf_chroma_coeff_abs[j])	
alf_chroma_clip_idx[j]	tu(v)
}	
}	
...	

[도27]

7.4.6.3 Adaptive loop filter data semantics

...

alf_luma_clip_idx[sflidx][j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the signalled luma filter indicated by sflidx. When **alf_luma_clip_idx[sflidx][j]** is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_luma_clip_idx[sflidx][j]** with sflidx = 0..alf_luma_num_filters_signalled_minus1 and j = 0..11 shall be in the range of 0 to 3, inclusive. The maximum value maxVal of the truncated unary binarization tu(v) is set equal to 3.

...

alf_chroma_clip_idx[j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the chroma filter. When **alf_chroma_clip_idx[j]** is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_chroma_clip_idx[j]** with j = 0..5 shall be in the range of 0 to 3, inclusive. The maximum value maxVal of the truncated unary binarization tu(v) is set equal to 3.

...

[도28]

General slice header syntax

slice_header() {	Descriptor
...	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	tb(v)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(5)
slice_alf_chroma_idc	tu(v)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(5)
}	
}	
...	
}	

[도29a]

Adaptive loop filter data syntax

	Descriptor
alf_data(adaptation_parameter_set_id) {	
alf_luma_filter_signal_flag	u(1)
alf_chroma_filter_signal_flag	u(1)
if(alf_luma_filter_signal_flag) {	
alf_luma_clip_flag	u(1)
alf_luma_num_filters_signalled_minus1	tb(v)
if(alf_luma_num_filters_signalled_minus1 > 0) {	
for(filIdx = 0; filIdx < NumAlfFilters; filIdx++)	
alf_luma_coeff_delta_idx[filIdx]	tb(v)
}	
alf_luma_coeff_signal_flag	u(1)
if(alf_luma_coeff_signal_flag) {	
for(sfilIdx = 0; sfilIdx <= alf_luma_num_filters_signalled_minus1; sfilIdx++)	
alf_luma_coeff_flag[sfilIdx]	u(1)
}	
for(sfilIdx = 0; sfilIdx <= alf_luma_num_filters_signalled_minus1; sfilIdx++) {	
if(alf_luma_coeff_flag[sfilIdx]) {	
for(j = 0; j < 12; j++) {	
alf_luma_coeff_abs[sfilIdx][j]	ue3(v)
if(alf_luma_coeff_abs[sfilIdx][j] > 0)	
alf_luma_coeff_sign[sfilIdx][j]	u(1)
}	
}	
}	
}	

[도29b]

if(alf_luma_clip_flag) {	
for(sfilIdx = 0; sfilIdx <= alf_luma_num_filters_signalled_minus1; sfilIdx++) {	
for(j = 0; j < 12; j++) {	
alf_luma_clip_idx[sfilIdx][j]	u(2)
}	
}	
}	
if(alf_chroma_filter_signal_flag) {	
alf_chroma_num_alts_minus1	tb(v)
for(altIdx = 0; altIdx <= alf_chroma_num_alts_minus1; altIdx++) {	
alf_chroma_clip_flag[altIdx]	u(1)
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[altIdx][j]	ue3(v)
if(alf_chroma_coeff_abs[altIdx][j] > 0)	
alf_chroma_coeff_sign[altIdx][j]	u(1)
}	
if(alf_chroma_clip_flag[altIdx]) {	
for(j = 0; j < 6; j++) {	
alf_chroma_clip_idx[altIdx][j]	u(2)
}	
}	
}	
}	
}	

[도30a]

Coding tree unit syntax

coding_tree_unit() {	Descriptor
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
if(slice_sao_luma_flag slice_sao_chroma_flag)	
sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)	
if(slice_alf_enabled_flag) {	
alf_ctb_flag[0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_ctb_flag[0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]) {	
if(slice_num_alf_aps_ids_luma > 0)	
alf_ctb_use_first_aps_flag	ae(v)
if(!alf_ctb_use_first_aps_flag) {	
if(slice_num_alf_aps_ids_luma > 1)	
alf_use_aps_flag	ae(v)
if(alf_use_aps_flag)	
alf_luma_prev_filter_idx_minus1	ae(v)
else	
alf_luma_fixed_filter_idx	ae(v)
}	
}	
}	

[도30b]

if(slice_alf_chroma_idc == 1 slice_alf_chroma_idc == 3) {	
alf_ctb_flag[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_ctb_flag[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	
&& aps_alf_chroma_num_alts_minus1 > 0)	
alf_ctb_filter_alt_idx[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize][0]	ae(v)
}	
if(slice_alf_chroma_idc == 2 slice_alf_chroma_idc == 3) {	
alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	
&& aps_alf_chroma_num_alts_minus1 > 0)	
alf_ctb_filter_alt_idx[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize][0]	ae(v)
}	
}	
if(slice_type == I && qtbt_dual_tree_intra_flag)	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbSizeY, 0)	
else	
coding_tree(xCtb, yCtb, CtbSizeY, CtbSizeY, 1, 0, 0, 0, 0, 0, SINGLE_TREE)	
}	

[31]

General slice header semantics

slice_alf_enabled_flag equal to 1 specifies that adaptive loop filter is enabled and may be applied to Y, Cb, or Cr colour component in a slice. **slice_alf_enabled_flag** equal to 0 specifies that adaptive loop filter is disabled for all colour components in a slice.

slice_num_alf_aps_ids_luma specifies the number of ALF APSs that the slice refers to. The value of **slice_num_alf_aps_ids_luma** shall be in the range of 0 to 6, inclusive.

The maximum value **maxVal** of the truncated binary binarization **tb(v)** is set equal to 6.

slice_alf_aps_id_luma[i] specifies the **adaptation_parameter_set_id** of the i-th ALF APS that the slice refers to. The **TemporalId** of the ALF APS NAL unit having **adaptation_parameter_set_id** equal to **slice_alf_aps_id_luma[i]** shall be less than or equal to the **TemporalId** of the coded slice NAL unit.

When multiple ALF APSs with the same value of **adaptation_parameter_set_id** are referred to by two or more slices of the same picture, the multiple ALF APSs with the same value of **adaptation_parameter_set_id** shall have the same content.

For intra slices and slices in an IRAP picture, **slice_alf_aps_id_luma[i]** shall not refer to an ALF APS associated with other pictures rather than the picture containing the intra slices or the IRAP picture.

slice_alf_chroma_idc equal to 0 specifies that the adaptive loop filter is not applied to Cb and Cr colour components. **slice_alf_chroma_idc** equal to 1 indicates that the adaptive loop filter is applied to the Cb colour component. **slice_alf_chroma_idc** equal to 2 indicates that the adaptive loop filter is applied to the Cr colour component. **slice_alf_chroma_idc** equal to 3 indicates that the adaptive loop filter is applied to Cb and Cr colour components. When **slice_alf_chroma_idc** is not present, it is inferred to be equal to 0.

The maximum value **maxVal** of the truncated unary binarization **tu(v)** is set equal to 3.

slice_alf_aps_id_chroma specifies the **adaptation_parameter_set_id** that the chroma component of the slice refers to. The **TemporalId** of the ALF APS NAL unit having **adaptation_parameter_set_id** equal to **slice_alf_aps_id_chroma** shall be less than or equal to the **TemporalId** of the coded slice NAL unit.

For intra slices and slices in an IRAP picture, **slice_alf_aps_id_chroma** shall not refer to an ALF APS associated with other pictures rather than the picture containing the intra slices or the IRAP picture.

[32]

Adaptive loop filter data semantics

alf_luma_filter_signal_flag equal to 1 specifies that a luma filter set is signalled. **alf_luma_filter_signal_flag** equal to 0 specifies that a luma filter set is not signalled.

alf_chroma_filter_signal_flag equal to 1 specifies that a chroma filter is signalled. **alf_chroma_filter_signal_flag** equal to 0 specifies that a chroma filter is not signalled.

The variable **NumAlfFilters** specifying the number of different adaptive loop filters is set equal to 25.

alf_luma_clip_flag equal to 0 specifies that linear adaptive loop filtering is applied on luma component. **alf_luma_clip_flag** equal to 1 specifies that non-linear adaptive loop filtering may be applied on luma component.

alf_luma_num_filters_signalled_minus1 plus 1 specifies the number of adaptive loop filter classes for which luma coefficients can be signalled. The value of **alf_luma_num_filters_signalled_minus1** shall be in the range of 0 to **NumAlfFilters** - 1, inclusive.

The maximum value **maxVal** of the truncated binary binarization **tb(v)** is set equal to **NumAlfFilters** - 1.

alf_luma_coeff_delta_idx[filtIdx] specifies the indices of the signalled adaptive loop filter luma coefficient deltas for the filter class indicated by **filtIdx** ranging from 0 to **NumAlfFilters** - 1. When **alf_luma_coeff_delta_idx[filtIdx]** is not present, it is inferred to be equal to 0.

The maximum value **maxVal** of the truncated binary binarization **tb(v)** is set equal to **alf_luma_num_filters_signalled_minus1**.

alf_luma_coeff_signal_flag equal to 1 indicates that **alf_luma_coeff_flag[sIdx]** is signalled. **alf_luma_coeff_signal_flag** equal to 0 indicates that **alf_luma_coeff_flag[sIdx]** is not signalled.

alf_luma_coeff_flag[sIdx] equal 1 specifies that the coefficients of the luma filter indicated by **sIdx** are signalled. **alf_luma_coeff_flag[sIdx]** equal to 0 specifies that all filter coefficients of the luma filter indicated by **sIdx** are set equal to 0. When not present, **alf_luma_coeff_flag[sIdx]** is set equal to 1.

alf_luma_coeff_abs[sIdx][j] specifies the absolute value of the j-th coefficient delta of the signalled luma filter indicated by **sIdx**. When **alf_luma_coeff_abs[sIdx][j]** is not present, it is inferred to be equal 0.

[도33]

alf_luma_coeff_sign[sIdx][j] specifies the sign of the j-th luma coefficient of the filter indicated by sIdx as follows:

- If **alf_luma_coeff_sign**[sIdx][j] is equal to 0, the corresponding luma filter coefficient has a positive value.
- Otherwise (**alf_luma_coeff_sign**[sIdx][j] is equal to 1), the corresponding luma filter coefficient has a negative value.

When **alf_luma_coeff_sign**[sIdx][j] is not present, it is inferred to be equal to 0.

The variable **filtCoeff**[sIdx][j] with sIdx = 0..alf_luma_num_filters_signalled_minus1, j = 0..11 is initialized as follows:

$$\text{filtCoeff}[sIdx][j] = \text{alf_luma_coeff_abs}[sIdx][j] * (1 - 2 * \text{alf_luma_coeff_sign}[sIdx][j]) \quad (7-68)$$

The luma filter coefficients **AlfCoeff**[adaptation_parameter_set_id] with elements **AlfCoeff**[adaptation_parameter_set_id][filtIdx][j], with filtIdx = 0..NumAlfFilters - 1 and j = 0..11 are derived as follows:

$$\text{AlfCoeff}[adaptation_parameter_set_id][filtIdx][j] = \text{filtCoeff}[\text{alf_luma_coeff_delta_idx}[filtIdx]][j] \quad (7-70)$$

The fixed filter coefficients **AlfFixFiltCoeff**[i][j] with i = 0..64, j = 0..11 and the class to filter mapping **AlfClassToFiltMap**[m][n] with m = 0..15 and n = 0..24 are derived as follows:

[도34]

$$\text{AlfClassToFiltMap} = \quad (7-73)$$

```
(
{ 8, 2, 2, 2, 3, 4, 53, 9, 9, 52, 4, 4, 5, 9, 2, 8, 10, 9, 1, 3, 39, 39, 10, 9, 52 }
{ 11, 12, 13, 14, 15, 30, 11, 17, 18, 19, 16, 20, 20, 4, 53, 21, 22, 23, 14, 25, 26, 26, 27, 28, 10 }
{ 16, 12, 31, 32, 14, 16, 30, 33, 53, 34, 35, 16, 20, 4, 7, 16, 21, 36, 18, 19, 21, 26, 37, 38, 39 }
{ 35, 11, 13, 14, 43, 35, 16, 4, 34, 62, 35, 35, 30, 56, 7, 35, 21, 38, 24, 40, 16, 21, 48, 57, 39 }
{ 11, 31, 32, 43, 44, 16, 4, 17, 34, 45, 30, 20, 20, 7, 5, 21, 22, 46, 40, 47, 26, 48, 63, 58, 10 }
{ 12, 13, 50, 51, 52, 11, 17, 53, 45, 9, 30, 4, 53, 19, 0, 22, 23, 25, 43, 44, 37, 27, 28, 10, 55 }
{ 30, 33, 62, 51, 44, 20, 41, 56, 34, 45, 20, 41, 41, 56, 5, 30, 56, 38, 40, 47, 11, 37, 42, 57, 8 }
{ 35, 11, 23, 32, 14, 35, 20, 4, 17, 19, 21, 20, 20, 20, 4, 16, 21, 36, 46, 25, 41, 26, 48, 49, 58 }
{ 12, 31, 59, 59, 3, 33, 33, 59, 59, 52, 4, 33, 17, 59, 55, 22, 36, 59, 59, 60, 22, 36, 59, 25, 55 }
{ 31, 25, 15, 60, 60, 22, 17, 19, 55, 55, 20, 20, 53, 19, 55, 22, 46, 25, 43, 60, 37, 28, 10, 55, 52 }
{ 12, 31, 32, 50, 51, 11, 33, 53, 19, 45, 16, 4, 4, 53, 5, 22, 36, 18, 25, 43, 26, 27, 27, 28, 10 }
{ 5, 2, 44, 52, 3, 4, 53, 45, 9, 3, 4, 56, 5, 0, 2, 5, 10, 47, 52, 3, 63, 39, 10, 9, 52 }
{ 12, 34, 44, 44, 3, 56, 56, 62, 45, 9, 56, 56, 7, 5, 0, 22, 38, 40, 47, 52, 48, 57, 39, 10, 9 }
{ 35, 11, 23, 14, 51, 35, 20, 41, 56, 62, 16, 20, 41, 56, 7, 16, 21, 38, 24, 40, 26, 26, 42, 57, 39 }
{ 33, 34, 51, 51, 52, 41, 41, 34, 62, 0, 41, 41, 56, 7, 5, 56, 38, 38, 40, 44, 37, 42, 57, 39, 10 }
{ 16, 31, 32, 15, 60, 30, 4, 17, 19, 25, 22, 20, 4, 53, 19, 21, 22, 46, 25, 55, 26, 48, 63, 58, 55 }
),
```

It is a requirement of bitstream conformance that the values of **AlfCoeff**[adaptation_parameter_set_id][filtIdx][j] with filtIdx = 0..NumAlfFilters - 1, j = 0..11 shall be in the range of -2^7 to $2^7 - 1$, inclusive.

alf_luma_clip_idx[sIdx][j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the signalled luma filter indicated by sIdx. When **alf_luma_clip_idx**[sIdx][j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_luma_clip_idx**[sIdx][j] with sIdx = 0..alf_luma_num_filters_signalled_minus1 and j = 0..11 shall be in the range of 0 to 3, inclusive.

filters_signalled_minus1, j = 0..11 are derived from Table 7.1 by using **BitDepthY** and **alf_luma_clip_idx**[sIdx][j] as input of the Table 7.1.

The luma filter clipping values **AlfClip**[adaptation_parameter_set_id] with elements **AlfClip**[adaptation_parameter_set_id][filtIdx][j], with filtIdx = 0..NumAlfFilters - 1 and j = 0..11 are derived as follows:

$$\text{AlfClip}[adaptation_parameter_set_id][filtIdx][j] = \text{filterClips}[\text{alf_luma_coeff_delta_idx}[filtIdx]][j] \quad (7-77)$$

alf_chroma_num_alts_minus1 specifies the number of alternative filters for Chroma components.

The maximum value **maxVal** of the truncated binary binarization **tb(v)** is set equal to 8.

alf_chroma_clip_flag[altIdx] equal to 0 specifies that linear adaptive loop filtering is applied on chroma components when using the Chroma filter with index altIdx ranging from 0 to **alf_chroma_num_alts_minus1**; **alf_chroma_clip_flag**[altIdx] equal to 1 specifies that non-linear adaptive loop filtering is applied on chroma component when using the Chroma filter with index altIdx. When not present, **alf_chroma_clip_flag**[altIdx] is inferred to be equal to 0.

[D35a]

alf_chroma_coeff_abs[altIdx][j] specifies the absolute value of the j-th chroma filter coefficient of the alternative Chroma filter with index altIdx, for altIdx ranging from 0 to alf_chroma_num_alts_minus1. When **alf_chroma_coeff_abs**[altIdx][j] is not present, it is inferred to be equal 0. It is a requirement of bitstream conformance that the values of **alf_chroma_coeff_abs**[altIdx][j] shall be in the range of 0 to $2^7 - 1$, inclusive.

alf_chroma_coeff_sign[altIdx][j] specifies the sign of the j-th chroma filter coefficient of the alternative Chroma filter with index altIdx, for altIdx ranging from 0 to alf_chroma_num_alts_minus1, as follows:

- If **alf_chroma_coeff_sign**[altIdx][j] is equal to 0, the corresponding chroma filter coefficient has a positive value.
- Otherwise (**alf_chroma_coeff_sign**[altIdx][j] is equal to 1), the corresponding chroma filter coefficient has a negative value.

When **alf_chroma_coeff_sign**[altIdx][j] is not present, it is inferred to be equal to 0.

The chroma filter coefficients **AlfCoeffC**[adaptation_parameter_set_id][altIdx] with elements **AlfCoeffC**[adaptation_parameter_set_id][altIdx][j], with altIdx = 0..alf_chroma_num_alts_minus1, j = 0..5 are derived as follows:

$$\text{AlfCoeffC}[\text{adaptation_parameter_set_id}][\text{altIdx}][j] = \text{alf_chroma_coeff_abs}[\text{altIdx}][j] * (1 - 2 * \text{alf_chroma_coeff_sign}[\text{altIdx}][j]) \quad (7-82)$$

It is a requirement of bitstream conformance that the values of **AlfCoeffC**[adaptation_parameter_set_id][altIdx][j] with altIdx = 0..alf_chroma_num_alts_minus1, j = 0..5 shall be in the range of $-2^7 - 1$ to $2^7 - 1$, inclusive.

alf_chroma_clip_idx[altIdx][j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the alternative Chroma filter with index altIdx, for altIdx ranging from 0 to alf_chroma_num_alts_minus1. When **alf_chroma_clip_idx**[altIdx][j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_chroma_clip_idx**[altIdx][j] with j = 0..5 shall be in the range of 0 to 3, inclusive.

The chroma filter clipping values **AlfClipC**[adaptation_parameter_set_id][altIdx] with elements **AlfClipC**[adaptation_parameter_set_id][altIdx][j], with altIdx = 0..alf_chroma_num_alts_minus1, j = 0..5 are derived from the Table 7.1 by using **BitDepthC** and **alf_chroma_clip_idx**[altIdx][j] as inputs of the Table 7.1:

[D35b]

Table 7.1 – Specification of the ALF clipping values

BitDepthY or BitDepthC	alf_chroma_clip_idx[altIdx][j] or alf_luma_clip_idx[sIdx][j]			
	0	1	2	3
8	4	16	64	255
9	5	23	108	511
10	6	32	181	1023
11	7	45	304	2047
12	8	64	512	4095
13	10	91	861	8191
14	11	128	1448	16383
15	13	181	2435	32767
16	16	256	4096	65535

[도36]

Coding tree unit semantics

The CTU is the root node of the coding tree structure.

The array `IsInSmr[ x ][ y ]` specifying whether the sample at (x, y) is located inside a shared merging candidate list region, is initialized as follows for $x = 0..CtbSizeY - 1$ and $y = 0..CtbSizeY - 1$:

$$IsInSmr[x][y] = FALSE \quad (7-96)$$

`alf_ctb_flag[ cIdx ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]` equal to 1 specifies that the adaptive loop filter is applied to the coding tree block of the colour component indicated by `cIdx` of the coding tree unit at luma location $(xCtb, yCtb)$. `alf_ctb_flag[ cIdx ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]` equal to 0 specifies that the adaptive loop filter is not applied to the coding tree block of the colour component indicated by `cIdx` of the coding tree unit at luma location $(xCtb, yCtb)$.

When `alf_ctb_flag[ cIdx ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]` is not present, it is inferred to be equal to 0.

[도37]

`alf_ctb_use_first_aps_flag` equal to 1 specifies that the filter information in APS with `adaptive_parameter_set_id` equal to `slice_alf_aps_id_luma[ 0 ]` is used. `alf_ctb_use_first_aps_flag` equal to 0 specifies that the luma CTB does not use the filter information in APS with `adaptive_parameter_set_id` equal to `slice_alf_aps_id_luma[ 0 ]`. When `alf_ctb_use_first_aps_flag` is not present, it is inferred to be equal to 0.

`alf_use_aps_flag` equal to 0 specifies that one of the fixed filter sets is applied to the luma CTB. `alf_use_aps_flag` equal to 1 specifies that a filter set from an APS is applied to the luma CTB. When `alf_use_aps_flag` is not present, it is inferred to be equal to 0.

`alf_luma_prev_filter_idx_minus1` plus 1 specifies the previous filter that is applied to the luma CTB. The value of `alf_luma_prev_filter_idx_minus1` shall be in a range of 0 to `slice_num_alf_aps_ids_luma - 2`, inclusive.

The variable `AlfCtbFiltSetIdxY[ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]` specifying the filter set index for the luma CTB at location $(xCtb, yCtb)$ is derived as follows:

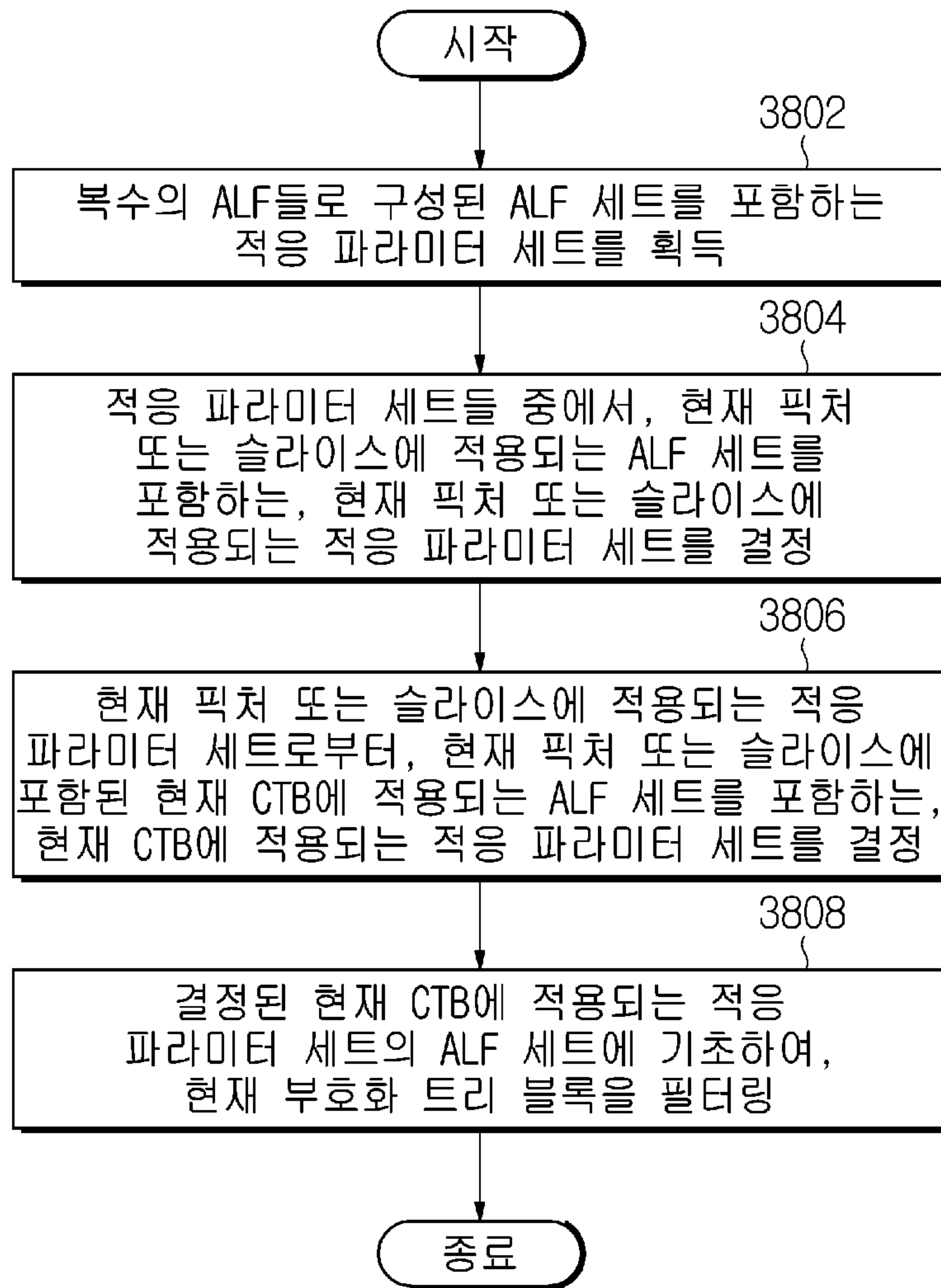
- If `alf_ctb_use_first_aps` is equal to 1, `AlfCtbFiltSetIdxY[ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]` is set equal to 16.
- Otherwise, if `alf_use_aps_flag` is equal to 0, `AlfCtbFiltSetIdxY[ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]` is set equal to `alf_luma_fixed_filter_idx`.
- Otherwise, `AlfCtbFiltSetIdxY[ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]` is set equal to `17 + alf_luma_prev_filter_idx_minus1`.

`alf_luma_fixed_filter_idx` specifies the fixed filter that is applied to the luma CTB. The value of `alf_luma_fixed_filter_idx` shall be in a range of 0 to 15, inclusive.

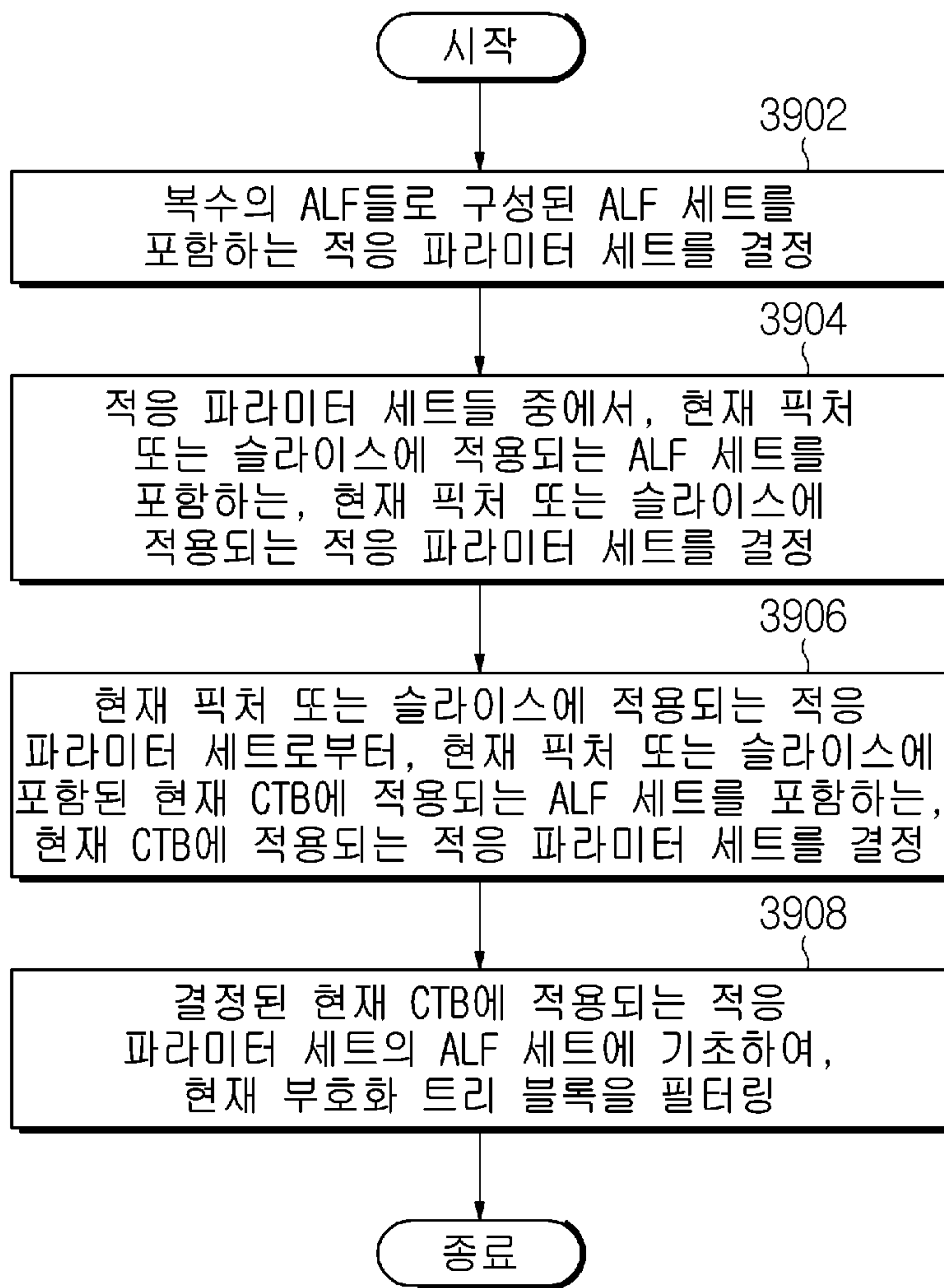
`alf_ctb_filter_alt_idx[ 1 ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ][ sIdx ]` specify the index of the alternative chroma filter to use when applying the adaptive loop filter on chroma component with index 1 of the coding tree unit at luma location $(xCtb, yCtb)$. When `alf_ctb_filter_alt_idx[ 1 ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ][ sIdx ]` is not present, it is inferred to be equal to zero.

`alf_ctb_filter_alt_idx[ 2 ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ][ sIdx ]` specify the index of the alternative chroma filter to use when applying the adaptive loop filter on chroma component with index 2 of the coding tree unit at luma location $(xCtb, yCtb)$. When `alf_ctb_filter_alt_idx[ 2 ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ][ sIdx ]` is not present, it is inferred to be equal to zero.

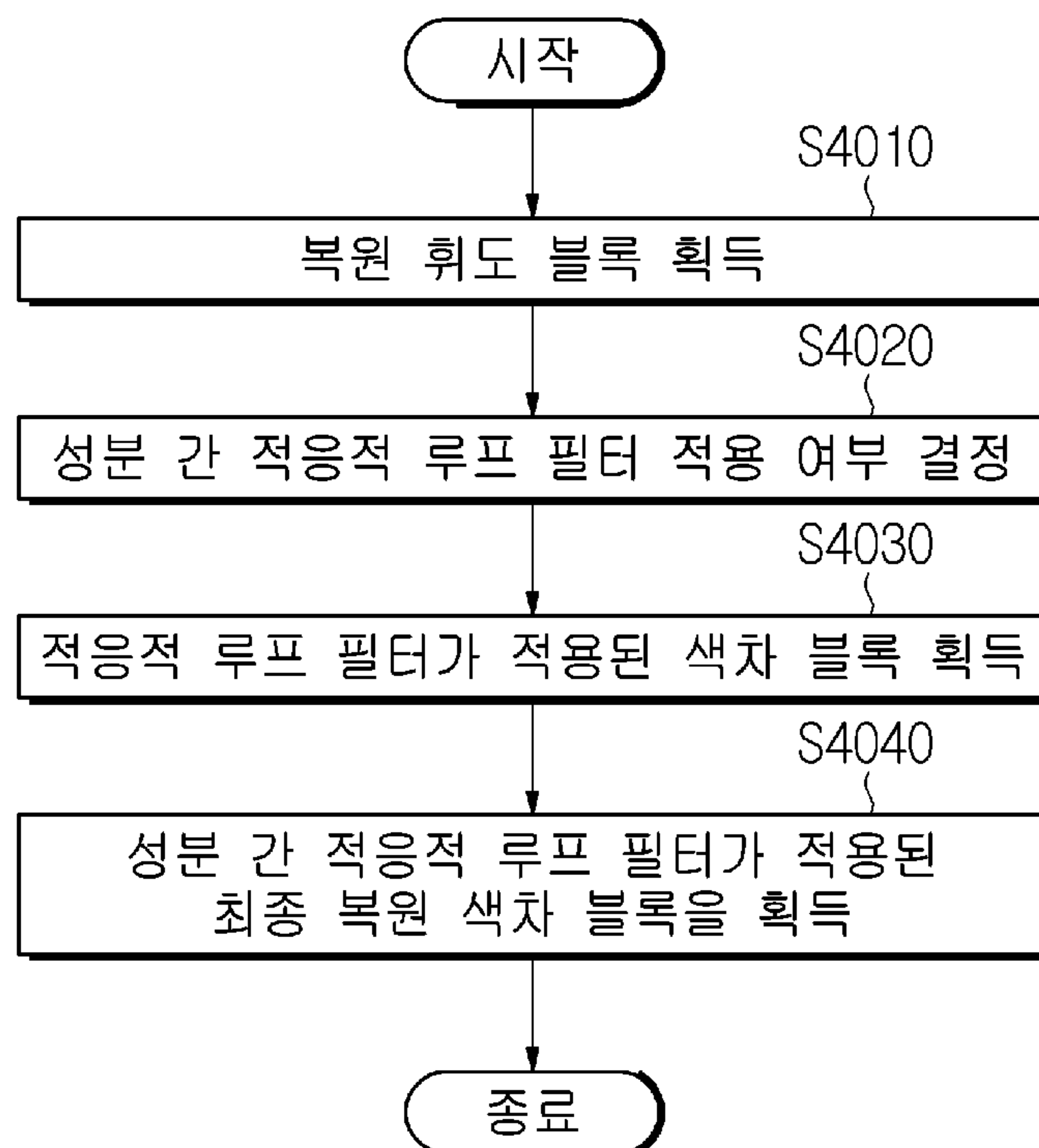
[도38]



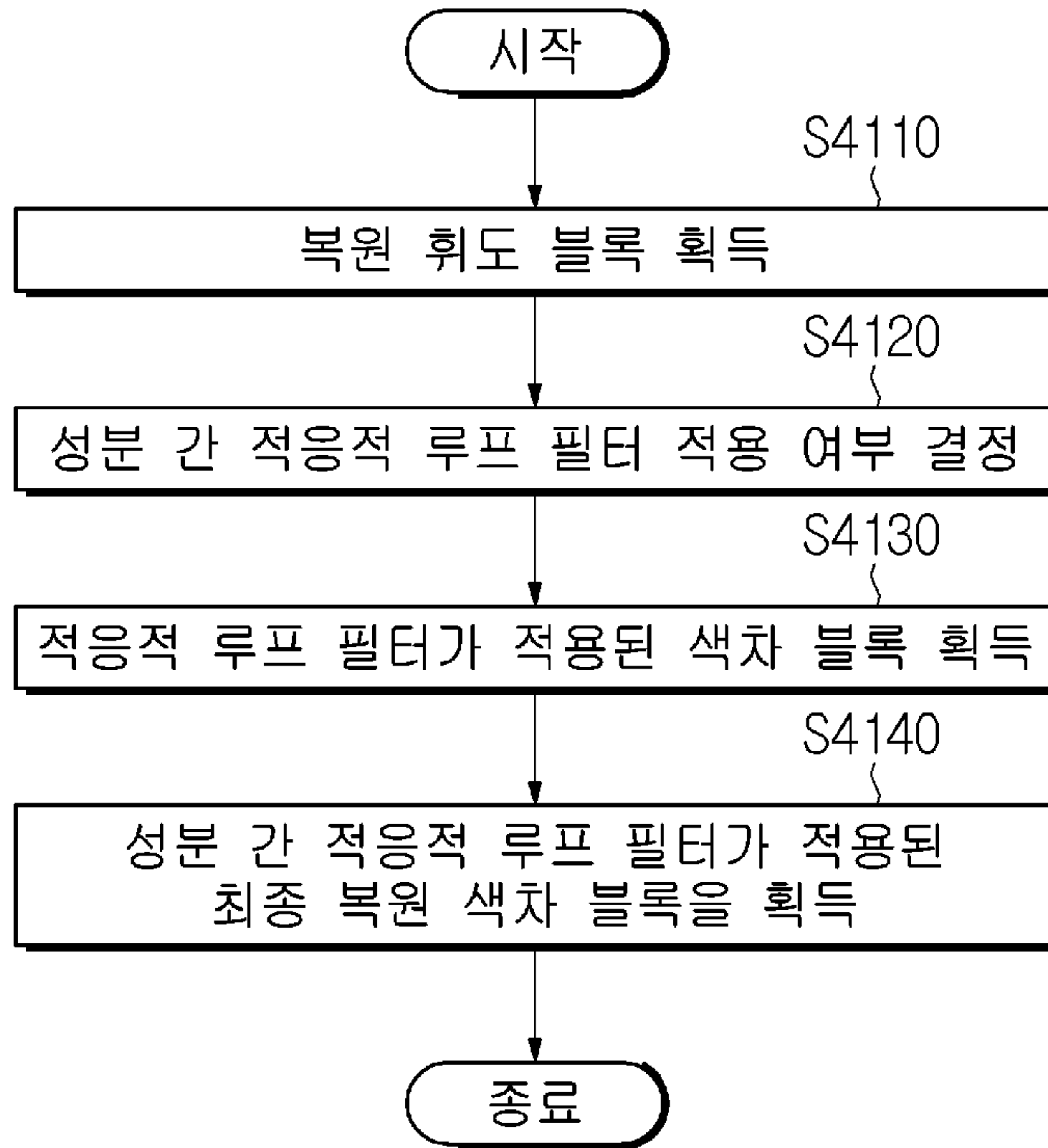
[도39]



[도40]



[도41]



[도42]

seq_parameter_set_rbsp() {	Descriptor
sps_alf_enabled_flag	u(1)

[도43]

adaptation_parameter_set_rbsp() {	Descriptor
adaptation_parameter_set_id	u(5)
aps_params_type	u(3)
if(aps_params_type == ALF_APS)	
alf_data(adaptation_parameter_set_id)	
else if(aps_params_type == LMCS_APS)	
lmcs_data()	
}	

[도44]

aps_params_type	aps_params_type 이름	적응 파라미터의 타입
0	ALF_APS	ALF 파라미터 (적응적 루프내 필터링 파라미터)
1	LMCS_APS	LMCS 파라미터 (색차 스케일링 및 휘도 매핑 파라미터)
2..7	Reserved	Reserved

[도45]

picture header rbsp() {	Descriptor
if(sps_alf_enabled_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
if(ChromaArrayType != 0)	
pic_cross_component_alf_cb_enabled_flag	u(1)
if(pic_cross_component_alf_cb_enabled_flag) {	
pic_cross_component_alf_cb_aps_id	u(3)
pic_cross_component_cb_filters_signalled_minus1	ue(v) 혹은 u(2)
}	
if(ChromaArrayType != 0)	
pic_cross_component_alf_cr_enabled_flag	u(1)
if(pic_cross_component_alf_cr_enabled_flag) {	
pic_cross_component_alf_cr_aps_id	u(3)
pic_cross_component_cr_filters_signalled_minus1	ue(v) 혹은 u(2)
}	
}	

[도46]

slice header() {	Descriptor
if(sps_alf_enabled_flag && !pic_alf_enabled_present_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
if(ChromaArrayType != 0)	
slice_cross_component_alf_cb_enabled_flag	u(1)
if(slice_cross_component_alf_cb_enabled_flag) {	
slice_cross_component_alf_cb_aps_id	u(3)
slice_cross_component_cb_filters_signalled_minus1	ue(v) 혹은 u(2)
}	
if(ChromaArrayType != 0)	
slice_cross_component_alf_cr_enabled_flag	u(1)
if(slice_cross_component_alf_cr_enabled_flag) {	
slice_cross_component_alf_cr_aps_id	u(3)
slice_cross_component_cr_filters_signalled_minus1	ue(v) 혹은 u(2)
}	
}	

[도47a]

alf data() {	Descriptor
alf luma filter signal flag	u(1)
alf chroma filter signal flag	u(1)
alf cross component cb filter signal flag	u(1)
alf cross component cr filter signal flag	u(1)
if(alf luma filter signal flag) {	
alf luma clip flag	u(1)
alf luma num filters signalled minus1	ue(v)
if(alf luma num filters signalled minus1 > 0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf luma coeff delta idx[filtIdx]	u(v)
}	
for(sfIdx = 0; sfIdx <= alf luma num filters signalled minus1; sfIdx++) {	
for (j = 0; j < 12; j++) {	
alf luma coeff abs[sfIdx][j]	uek(v)
if(alf luma coeff abs[sfIdx][j])	
alf luma coeff sign[sfIdx][j]	u(1)
}	
}	
if(alf luma clip flag) {	
for(sfIdx = 0; sfIdx <= alf luma num filters signalled minus1; sfIdx++) {	
for (j = 0; j < 12; j++)	
alf luma clip idx[sfIdx][j]	u(2)
}	
}	
}	
}	

[도47b]

if(alf chroma filter signal flag) {	
alf chroma num alt filters minus1	ue(v)
for(altIdx = 0; altIdx <= alf chroma num alt filters minus1; altIdx++) {	
alf chroma clip flag[altIdx]	u(1)
for(j = 0; j < 6; j++) {	
alf chroma coeff abs[altIdx][j]	uek(v)
if(alf chroma coeff abs[altIdx][j] > 0)	
alf chroma coeff sign[altIdx][j]	u(1)
}	
if(alf chroma clip flag[altIdx]) {	
for(j = 0; j < 6; j++)	
alf chroma clip idx[altIdx][j]	u(2)
}	
}	
}	
if (alf cross component cb filter signal flag) {	
alf cross component cb filters signalled minus1	ue(v) 혹은 u(2)
for(k = 0; k < (alf cross component cb filters signalled minus1+1); k++) {	
for (j = 0; j < 8; j++)	
alf_cross_component_cb_coeff_plus32[k][j]	u(6) 혹은 uek(v)
}	
}	
if (alf cross component cr filter signal flag) {	
alf cross component cr filters signalled minus1	ue(v) 혹은 u(2)
for(k = 0; k < (alf cross component cr filters signalled minus1+1); k++) {	
for (j = 0; j < 8; j++)	
alf_cross_component_cr_coeff_plus32[k][j]	u(6) 혹은 uek(v)
}	
}	
}	

[도48]

	Descriptor
coding tree unit() {	
xCtb = CtbAddrX << CtbLog2SizeY	
yCtb = CtbAddrY << CtbLog2SizeY	
if(slice sao luma flag slice sao chroma flag)	
sao(CtbAddrX, CtbAddrY)	
if(slice alf enabled flag) {	
alf_ctb_flag [0][CtbAddrX][CtbAddrY]	ae(v)
if(alf_ctb_flag[0][CtbAddrX][CtbAddrY]) {	
if(slice num alf aps ids luma > 0)	
alf_usc_aps_flag	ae(v)
if(alf use aps flag) {	
if(slice num alf aps ids luma > 1)	
alf_luma_prev_filter_idx	ae(v)
} else	
alf_luma_fixed_filter_idx	ae(v)
}	
}	
if(slice cross component alf cb enabled flag)	
alf_ctb_cross_component_cb_idc [CtbAddrX][CtbAddrY]	ae(v)
if(slice cross component alf cb enabled flag == 0 alf_ctb_cross_component_cb_idc[CtbAddrX][CtbAddrY] == 0)	
if(slice alf chroma idc == 1 slice alf chroma idc == 3) {	
alf_ctb_flag [1][CtbAddrX][CtbAddrY]	ae(v)
if(alf_ctb_flag[1][CtbAddrX][CtbAddrY] && alf_chroma_num_alt_filters_minus1 > 0)	
alf_ctb_filter_alt_idx [0][CtbAddrX][CtbAddrY]	ae(v)
}	
if(slice cross component alf cr enabled flag)	
alf_ctb_cross_component_cr_idc [CtbAddrX][CtbAddrY]	ae(v)
if(slice cross component alf cr enabled flag == 0 alf_ctb_cross_component_cr_idc[CtbAddrX][CtbAddrY] == 0)	
if(slice alf chroma idc == 2 slice alf chroma idc == 3) {	
alf_ctb_flag [2][CtbAddrX][CtbAddrY]	ae(v)
if(alf_ctb_flag[2][CtbAddrX][CtbAddrX] && alf_chroma_num_alt_filters_minus1 > 0)	
alf_ctb_filter_alt_idx [1][CtbAddrX][CtbAddrX]	ae(v)
}	
if(slice type == 1 && qtbt dual tree intra flag)	
dual tree implicit qt split (xCtb, yCtb, CtbSizeY, 0)	
else	
coding_tree(xCtb, yCtb, CtbSizeY, CtbSizeY, 1, 1, 0, 0, 0, 0, SINGLE_TREE, MODE_TYPE_ALL)	
}	

[도49]

	Descriptor
alf_data(adaptation parameter set id) {	
...	
if(alf_luma_clip_flag) {	
for(sfl_idx = 0; sfl_idx <= alf_luma_num_filters_signalled_minus1; sfl_idx++) {	
for(j = 0; j < 12; j++) {	
if(filtCoeff[sfl_idx][j])	
alf_luma_clip_idx [sfl_idx][j]	u(2)
}	
}	
}	
...	
}	
if(alf_chroma_clip_flag) {	
for(j = 0; j < 6; j++) {	
if(alf_chroma_coeff_abs [j])	
alf_chroma_clip_idx [j]	u(2)
}	
}	
...	

[도50]

7.4.6.3 Adaptive loop filter data semantics

...

alf_luma_clip_idx[sIdx][j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the signalled luma filter indicated by sIdx. When **alf_luma_clip_idx**[sIdx][j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_luma_clip_idx**[sIdx][j] with sIdx = 0..alf_luma_num_filters_signalled_minus1 and j = 0..11 shall be in the range of 0 to 3, inclusive.

...

alf_chroma_clip_idx[j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the chroma filter. When **alf_chroma_clip_idx**[j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_chroma_clip_idx**[j] with j = 0..5 shall be in the range of 0 to 3, inclusive.

[도51]

alf data(adaptation parameter set id) {	Descriptor
...	
if(alf_luma_clip_flag) {	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++) {	
for (j = 0; j < 12; j++) {	
if(filtCoeff[sIdx][j])	
alf_luma_clip_idx [sIdx][j]	tu(v)
}	
}	
}	
...	
}	
if(alf_chroma_clip_flag) {	
for(j = 0; j < 6; j++) {	
if(alf_chroma_coeff_abs [j])	
alf_chroma_clip_idx [j]	tu(v)
}	
}	
...	

[도52]

7.4.6.3 Adaptive loop filter data semantics

...

alf_luma_clip_idx[sIdx][j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the signalled luma filter indicated by sIdx. When **alf_luma_clip_idx**[sIdx][j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_luma_clip_idx**[sIdx][j] with sIdx = 0..alf_luma_num_filters_signalled_minus1 and j = 0..11 shall be in the range of 0 to 3, inclusive. The maximum value maxVal of the truncated unary binarization tu(v) is set equal to 3.

...

alf_chroma_clip_idx[j] specifies the clipping index of the clipping value to use before multiplying by the j-th coefficient of the chroma filter. When **alf_chroma_clip_idx**[j] is not present, it is inferred to be equal 0 (no clipping). It is a requirement of bitstream conformance that the values of **alf_chroma_clip_idx**[j] with j = 0..5 shall be in the range of 0 to 3, inclusive. The maximum value maxVal of the truncated unary binarization tu(v) is set equal to 3.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2020/016558

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/117(2014.01)i; H04N 19/82(2014.01)i; H04N 19/119(2014.01)i; H04N 19/176(2014.01)i; H04N 19/70(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 19/117(2014.01); H04N 19/105(2014.01); H04N 19/11(2014.01); H04N 19/124(2014.01); H04N 19/82(2014.01)

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: 적응적(adaptive), 루프(loop), 필터(filter), 루마(luma), 크로마(chroma), 블록(block), 크로스 컴포넌트(cross component), sps_ccalf_enabled_flag

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-2019-0062302 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 05 June 2019 (2019-06-05) See paragraphs [0114] and [0511].	1-20
A	TAQUET, Jonathan et al. NON-CE5 : CCALF design simplifications. JVET-P0331. Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH. pp. 1-5, 07 October 2019. See page 5.	1-20
A	NAM, Junghak et al. CE5-related: Cleanup on cross-component adaptive loop filter. JVET-P0543-v2. Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH. pp. 1-9, 01 October 2019. See pages 1-2.	1-20
A	KR 10-2019-0042579 A (QUALCOMM INCORPORATED) 24 April 2019 (2019-04-24) See paragraph [0108].	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	“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
“A” document defining the general state of the art which is not considered to be of particular relevance	“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
“D” document cited by the applicant in the international application	“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
“E” earlier application or patent but published on or after the international filing date	“&” document member of the same patent family
“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	

Date of the actual completion of the international search

22 February 2021

Date of mailing of the international search report

23 February 2021

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon Building 4, 189 Cheongsaro, Seo-gu, Daejeon 35208

Facsimile No. +82-42-481-8578

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2020/016558

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-2018-0019549 A (LG ELECTRONICS INC.) 26 February 2018 (2018-02-26) See paragraphs [0126]-[0127].	1-20

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/KR2020/016558

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
KR	10-2019-0062302	A	05 June 2019	CN	111434109	A	17 July 2020
				EP	3737093	A1	11 November 2020
				WO	2019-107911	A1	06 June 2019
KR	10-2019-0042579	A	24 April 2019	CN	109691102	A	26 April 2019
				EP	3507984	A1	10 July 2019
				JP	2019-525679	A	05 September 2019
				US	2018-0063527	A1	01 March 2018
				WO	2018-045207	A1	08 March 2018
KR	10-2018-0019549	A	26 February 2018	EP	3313079	A1	25 April 2018
				US	10750170	B2	18 August 2020
				US	2018-0160116	A1	07 June 2018
				WO	2016-204374	A1	22 December 2016

A. 발명이 속하는 기술분류(국제특허분류(IPC)) H04N 19/117(2014.01)i; H04N 19/82(2014.01)i; H04N 19/119(2014.01)i; H04N 19/176(2014.01)i; H04N 19/70(2014.01)i		
B. 조사된 분야 조사된 최소문헌(국제특허분류를 기재) H04N 19/117(2014.01); H04N 19/105(2014.01); H04N 19/11(2014.01); H04N 19/124(2014.01); H04N 19/82(2014.01) 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: 적응적(adaptive), 루프(loop), 필터(filter), 루마(luma), 크로마(chroma), 블록(block), 크로스 컴포넌트(cross component), sps_ccalf_enabled_flag		
C. 관련 문헌		
카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	KR 10-2019-0062302 A (한국전자통신연구원) 2019.06.05 단락 [0114], [0511]	1-20
A	JONATHAN TAQUET 등, 'NON-CE5 : CCALF design simplifications', JVET-P0331, oint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, 페이지 1-5, 2019.10.07 페이지 5	1-20
A	JUNGHAK NAM 등, 'CE5-related: Cleanup on cross-component adaptive loop filter', JVET-P0543-v2, oint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, 페이지 1-9, 2019.10.01 페이지 1-2	1-20
A	KR 10-2019-0042579 A (헬컴 인코포레이티드) 2019.04.24 단락 [0108]	1-20
A	KR 10-2018-0019549 A (엔지전자 주식회사) 2018.02.26 단락 [0126]-[0127]	1-20
☐ 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 이후에 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 2021년02월22일(22.02.2021)		국제조사보고서 발송일 2021년02월23일(23.02.2021)
ISA/KR의 명칭 및 우편주소 대한민국 특허청 (35208) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사) 팩스 번호 +82-42-481-8578		심사관 양정록 전화번호 +82-42-481-5709

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
KR 10-2019-0062302 A	2019/06/05	CN 111434109 A	2020/07/17
		EP 3737093 A1	2020/11/11
		WO 2019-107911 A1	2019/06/06
KR 10-2019-0042579 A	2019/04/24	CN 109691102 A	2019/04/26
		EP 3507984 A1	2019/07/10
		JP 2019-525679 A	2019/09/05
		US 2018-0063527 A1	2018/03/01
		WO 2018-045207 A1	2018/03/08
KR 10-2018-0019549 A	2018/02/26	EP 3313079 A1	2018/04/25
		US 10750170 B2	2020/08/18
		US 2018-0160116 A1	2018/06/07
		WO 2016-204374 A1	2016/12/22