



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0037137
(43) 공개일자 2024년03월21일

(51) 국제특허분류(Int. Cl.)
H04N 19/105 (2014.01) H04N 19/176 (2014.01)
H04N 19/577 (2014.01) H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/105 (2015.01)
H04N 19/176 (2015.01)
(21) 출원번호 10-2023-0019325
(22) 출원일자 2023년02월14일
심사청구일자 없음
(30) 우선권주장
JP-P-2022-146436 2022년09월14일 일본(JP)

(71) 출원인
샤프 가부시키키가이샤
일본국 오사카후 사카이시 사카이쿠 타쿠미초 1반
치
(72) 발명자
야스기, 유키노부
일본 590-8522 오사카 사카이시 사카이쿠 타쿠미
초 1 샤프 가부시키키가이샤 내
이카이, 토모히로
일본 590-8522 오사카 사카이시 사카이쿠 타쿠미
초 1 샤프 가부시키키가이샤 내
아오노, 토모코
일본 590-8522 오사카 사카이시 사카이쿠 타쿠미
초 1 샤프 가부시키키가이샤 내
(74) 대리인
장수길, 박충범

전체 청구항 수 : 총 8 항

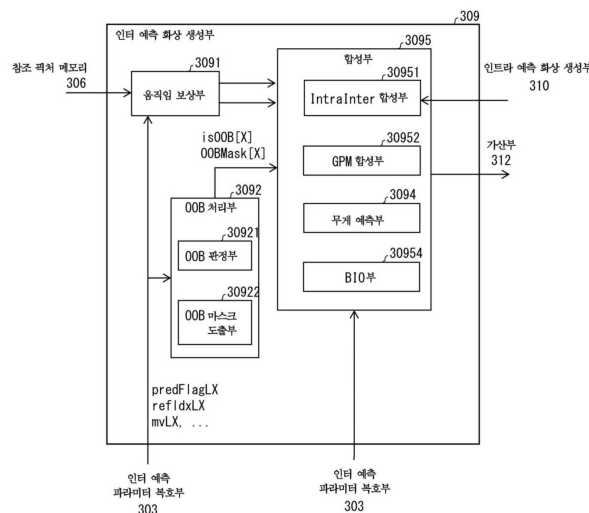
(54) 발명의 명칭 동화상 복호 장치 및 동화상 부호화 장치

(57) 요약

[과제] 부호화 효율을 향상시킬 수 있는 동화상 부호화·복호 장치를 제공한다.

[해결 수단] 동화상 복호 장치는 참조 블록의 좌표를 픽처의 좌표와 비교함으로써, 참조 블록이 OOB 처리의 대상인지 여부를 판정하는 OOB 판정부와, 참조 블록에 포함되는 화소의 좌표를 픽처의 좌표와 비교함으로써, 각 화소의 이용 가부를 나타내는 마스크 데이터를 도출하는 OOB 마스크 도출부를 구비하며, 상기 OOB 판정부는 상기 참조 블록이 포함되는 참조 픽처에 적용되는 처리의 유무를 이용하여 상기 참조 블록이 OOB 처리 대상인지 여부를 판정하는 것을 특징으로 한다.

대표도 - 도4



(52) CPC특허분류

H04N 19/577 (2015.01)

H04N 19/70 (2015.01)

명세서

청구범위

청구항 1

참조 블록의 좌표를 픽처의 좌표와 비교함으로써, 참조 블록이 OOB 처리의 대상인지 여부를 판정하는 OOB 판정부와, 참조 블록에 포함되는 화소의 좌표와 픽처 경계 혹은 영역 경계의 좌표와 비교함으로써, 각 화소의 이용 가부를 나타내는 마스크 데이터를 도출하는 OOB 마스크 도출부를 구비하는 동화상 복호 장치로서,

상기 OOB 판정부는 상기 참조 블록이 포함되는 참조 픽처에 적용되는 처리의 유무를 이용하여 상기 참조 블록이 OOB 처리 대상인지 여부를 판정하는 것을 특징으로 하는 동화상 복호 장치.

청구항 2

제1항에 있어서, 상기 OOB 판정부는 L0 참조 영역의 유효 영역과 L1 참조 영역의 유효 영역을 비교하여, 참조 영역의 유효 영역이 큰 참조 픽처에 대해 OOB 판정을 무효로 하는 것을 특징으로 하는 동화상 복호 장치.

청구항 3

제1항에 있어서, 상기 OOB 판정부는 상기 L0 참조 영역과 상기 L1 참조 영역의 유효 영역의 면적의 비교에 의해, 상기 유효 영역의 크기를 판정하는 것을 특징으로 하는 동화상 복호 장치.

청구항 4

제1항에 있어서, 상기 OOB 판정부는 상기 L0 참조 영역과 상기 L1 참조 영역의 유효 영역의 폭 혹은 높이의 비교에 의해, 상기 유효 영역의 크기를 판정하는 것을 특징으로 하는 동화상 복호 장치.

청구항 5

제1항에 있어서, 상기 OOB 판정부는 블록 단위의 가중 양예측(BCW: Bi-prediction with CU-level Weight)이 적용되는 블록에서는, 상기 OOB 판정을 무효로 하는 것을 특징으로 하는 동화상 복호 장치.

청구항 6

제1항에 있어서, 상기 OOB 마스크 도출부는 NxM 화소(N, M는 양의 정수)를 단위로 하여 마스크로서 도출하는 것을 특징으로 하는 동화상 복호 장치.

청구항 7

제1항에 있어서, 상기 OOB 판정부는 대상 블록이 소정 사이즈 이하인 경우에는 OOB 판정을 무효로 하는 것을 특징으로 하는 동화상 복호 장치.

청구항 8

참조 블록의 좌표를 픽처의 좌표와 비교함으로써, 참조 블록이 OOB 처리의 대상인지 여부를 판정하는 OOB 판정부와, 참조 블록에 포함되는 화소의 좌표를 픽처의 좌표와 비교함으로써, 각 화소의 이용 가부를 나타내는 마스크 데이터를 도출하는 OOB 마스크 도출부를 구비하는 동화상 부호화 장치로서,

상기 OOB 판정부는 상기 참조 블록이 포함되는 참조 픽처에 적용되는 처리의 유무를 이용하여 상기 참조 블록이 OOB 처리 대상인지 여부를 판정하는 것을 특징으로 하는 동화상 부호화 장치.

발명의 설명

기술 분야

본 발명의 실시형태는 동화상 복호 장치 및 동화상 부호화 장치에 관한 것이다.

[0001]

배경 기술

- [0002] 동화상을 효율적으로 전송 또는 기록하기 위해, 동화상을 부호화함으로써 부호화 데이터를 생성하는 동화상 부호화 장치, 및 당해 부호화 데이터를 복호함으로써 복호 화상을 생성하는 동화상 복호 장치가 이용되고 있다.
- [0003] 구체적인 동화상 부호화 방식으로서, 예를 들어 H.264/AVC나 HEVC(High-Efficiency Video Coding), Versatile Video Coding(VVC) 방식 등을 들 수 있다.
- [0004] 이러한 동화상 부호화 방식에서는, 동화상을 구성하는 화상(픽처)은 화상을 분할함으로써 얻어지는 슬라이스, 슬라이스를 분할함으로써 얻어지는 부호화 트리 유닛(CTU: Coding Tree Unit), 부호화 트리 유닛을 분할함으로써 얻어지는 부호화 단위(부호화 유닛(Coding Unit: CU)으로 불리는 경우도 있다), 및 부호화 단위를 분할함으로써 얻어지는 변환 유닛(TU: Transform Unit)으로 이루어진 계층 구조에 의해 관리되며, CU별로 부호화/복호된다.
- [0005] 또한, 이러한 동화상 부호화 방식에서는, 통상, 입력 화상을 부호화/복호함으로써 얻어지는 국소 복호 화상을 기초로 예측 화상이 생성되고, 당해 예측 화상을 입력 화상(원화상)으로부터 감산하여 얻어지는 예측 오차(「차분 화상」 또는 「잔차 화상」이라고 부르는 경우도 있다)가 부호화된다. 예측 화상의 생성 방법으로서, 화면간 예측(인터 예측) 및 화면내 예측(인트라 예측)을 들 수 있다. VVC에서는, 도 8(a)와 같이, 움직임 벡터(motion vector)의 좌표를 랩 어라운드(wrap-around) 처리하여, 수평 방향의 좌표계를 픽처의 좌단과 우단을 연속시켜 움직임 보상을 할 수 있다. 따라서, 360도 파노라마 화상이나 360도 화상과 같이 픽처의 좌우단이 공간적으로 연속하는 영상에 랩 어라운드 처리를 적용하면, 부호화 효율을 향상시킬 수 있다. 또한, VVC에서는, 도 8(b)와 같이 픽처 단위로 해상도를 변화시켜 움직임 보상을 하는 참조 픽처 리샘플링(RPR: Reference Picture Resampling) 처리가 가능하다. 인터넷상에서의 동화상 전달과 같은 전송 레이트가 변동하는 서비스에 RPR 처리를 적용하면, 화질을 향상시킬 수 있다.
- [0006] 또한, 비특허문헌 1에는, 참조 화상의 픽처 외의 영역을 포함하는 양예측의 한쪽 참조 블록의 일부 혹은 모든 영역을, 다른쪽 참조 블록의 일부 혹은 모든 영역으로 대체하는 OOB(Out-Of-Boundary) 처리 기술이 개시되어 있다.

선행기술문헌

비특허문헌

- [0007] (비특허문헌 0001) 비특허문헌 1: "AHG12: Enhanced bi-directional motion compensation", Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 25th Meeting, by teleconference, JVET-Y0125

발명의 내용

해결하려는 과제

- [0008] 비특허문헌 1에 기재된 방법에서는, 참조 블록이 픽처 외의 영역을 포함하는 경우, 픽처 외의 영역을 나타내는 마스크를 도출하면서, 양예측 화상을 도출한다. 따라서 계산량이 많다고 하는 과제가 있다.

과제의 해결 수단

- [0009] 상기 과제를 해결하기 위해, 본 발명의 일 태양에 관한 동화상 복호 장치는,
- [0010] 참조 블록의 좌표를 픽처의 좌표와 비교함으로써, 참조 블록이 OOB 처리의 대상인지 여부를 판정하는 OOB 판정부와, 참조 블록에 포함되는 화소의 좌표와 픽처 경계 혹은 영역 경계의 좌표와 비교함으로써, 각 화소의 이용가부를 나타내는 마스크 데이터를 도출하는 OOB 마스크 도출부를 구비하며, 상기 OOB 판정부는 상기 참조 블록이 포함되는 참조 픽처에 적용되는 처리의 유무를 이용하여 상기 참조 블록이 OOB 처리 대상인지 여부를 판정하는 것을 특징으로 한다.
- [0011] 본 발명의 일 태양에 관한 동화상 복호 장치에 있어서, 상기 OOB 판정부는 L0 참조 영역의 유효 영역과 L1 참조 영역의 유효 영역을 비교하여, 참조 영역의 유효 영역이 큰 참조 픽처에 대해 OOB 판정을 무효로 하는 것을 특

정으로 한다.

발명의 효과

[0012] 본 발명의 태양에 따르면, 동화상 부호화·복호 처리에서 OOB 처리에 드는 계산량을 삭감할 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 실시형태에 관한 화상 전송 시스템의 구성을 나타내는 개략도이다.

도 2는 부호화 스트림의 데이터의 계층 구조를 나타내는 도면이다.

도 3은 동화상 복호 장치의 구성을 나타내는 개략도이다.

도 4는 인터 예측 화상 생성부의 구성을 나타내는 개략도이다.

도 5는 인터 예측 파라미터 도출부의 구성을 나타내는 개략도이다.

도 6은 동화상 복호 장치의 개략적 동작을 설명하는 흐름도이다.

도 7은 동화상 부호화 장치의 구성을 나타내는 블록도이다.

도 8은 종래 기술의 랩 어라운드 처리와 참조 픽처 리샘플링 처리의 예를 나타내는 도면이다.

도 9는 OOB 처리의 개략적 동작을 설명하는 흐름도이다.

도 10은 OOB 판정의 예를 나타내는 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0014] (제1 실시형태)

[0015] 이하, 도면을 참조하면서 본 발명의 실시형태에 대하여 설명한다.

[0016] 도 1은 본 실시형태에 관한 화상 전송 시스템(1)의 구성을 나타내는 개략도이다.

[0017] 화상 전송 시스템(1)은 대상 화상을 부호화한 부호화 스트림을 전송하고, 전송된 부호화 스트림을 복호하여 화상을 표시하는 시스템이다. 화상 전송 시스템(1)은 동화상 부호화 장치(화상 부호화 장치)(11), 네트워크(21), 동화상 복호 장치(화상 복호 장치)(31) 및 동화상 표시 장치(화상 표시 장치)(41)를 포함하여 구성된다.

[0018] 동화상 부호화 장치(11)에는 화상(T)이 입력된다.

[0019] 네트워크(21)는 동화상 부호화 장치(11)가 생성한 부호화 스트림(Te)을 동화상 복호 장치(31)에 전송한다. 네트워크(21)는 인터넷(Internet), 광역 네트워크(WAN: Wide Area Network), 소규모 네트워크(LAN: Local Area Network) 또는 이들의 조합이다. 네트워크(21)는 반드시 양방향 통신망으로 한정되지 않으며, 지상 디지털 방송, 위성 방송 등의 방송파를 전송하는 일방향의 통신망일 수도 있다. 또한, 네트워크(21)는 DVD(Digital Versatile Disc: 등록 상표), BD(Blu-ray Disc: 등록 상표) 등의 부호화 스트림(Te)을 기록한 저장 매체로 대체될 수도 있다.

[0020] 동화상 복호 장치(31)는 네트워크(21)가 전송한 부호화 스트림(Te)의 각각을 복호하여, 복호한 1 또는 복수의 복호 화상(Td)을 생성한다.

[0021] 동화상 표시 장치(41)는 동화상 복호 장치(31)가 생성한 1 또는 복수의 복호 화상(Td)의 전부 또는 일부를 표시한다. 동화상 표시 장치(41)는 예를 들어 액정 디스플레이, 유기 EL(Electro-luminescence) 디스플레이 등의 표시 디바이스를 구비한다. 디스플레이의 형태로서는, 거치, 모바일, HMD 등을 들 수 있다. 또한, 동화상 복호 장치(31)가 높은 처리 능력을 갖는 경우에는, 화질이 높은 화상을 표시하고, 보다 낮은 처리 능력밖에 갖지 않는 경우에는, 높은 처리 능력, 표시 능력을 필요로 하지 않는 화상을 표시한다.

[0022] <연산자>

[0023] 본 명세서에서 이용하는 연산자를 이하에 기재한다.

[0024] >>는 우측 비트 시프트, <<는 좌측 비트 시프트, &는 비트와이즈 AND, |은 비트와이즈 OR, |=은 OR 대입 연산자이고, ||은 논리합을 나타낸다.

- [0025] $x?y:z$ 는 x 가 참(0 이외)인 경우에 y , x 가 거짓(0)인 경우에 z 를 취하는 3항 연산자이다.
- [0026] $\text{Clip3}(a, b, c)$ 는 c 를 a 이상 b 이하의 값으로 클립하는 함수이며, $c < a$ 의 경우에는 a 를 반환(return)하고, $c > b$ 의 경우에는 b 를 반환하고, 그 외의 경우에는 c 를 반환하는 함수이다(단, $a \leq b$).
- [0027] $\text{ClipH}(o, W, x)$ 는 $x < 0$ 의 경우 x , $x > W-1$ 의 경우 $x-o$, 그 이외의 경우 x 를 반환하는 함수이다.
- [0028] $\text{sign}(a)$ 는 $a > 0$ 의 경우 1, $a == 0$ 의 경우 1, $a < 0$ 의 경우 -1을 반환하는 함수이다.
- [0029] $\text{abs}(a)$ 는 a 의 절대값을 반환하는 함수이다.
- [0030] $\text{Int}(a)$ 는 a 의 정수값을 반환하는 함수이다.
- [0031] $\text{floor}(a)$ 는 a 이하의 최대의 정수를 반환하는 함수이다.
- [0032] $\text{ceil}(a)$ 는 a 이상의 최소 정수를 반환하는 함수이다.
- [0033] a/d 는 d 에 의한 a 의 나눗셈(소수점 이하 버림)을 나타낸다.
- [0034] <부호화 스트림(Te)의 구조>
- [0035] 본 실시형태에 관한 동화상 부호화 장치(11) 및 동화상 복호 장치(31)의 상세한 설명에 앞서, 동화상 부호화 장치(11)에 의해 생성되고, 동화상 복호 장치(31)에 의해 복호되는 부호화 스트림(Te)의 데이터 구조에 대하여 설명한다.
- [0036] 도 2는 부호화 스트림(Te)에서의 데이터의 계층 구조를 나타내는 도면이다. 부호화 스트림(Te)은 예시적으로 시퀀스, 및 시퀀스를 구성하는 복수의 픽처를 포함한다. 도 2의 (a)~(f)는 각각 시퀀스 SEQ를 규정하는 부호화 비디오 시퀀스, 픽처 PICT를 규정하는 부호화 픽처, 슬라이스 S를 규정하는 부호화 슬라이스, 슬라이스 데이터를 규정하는 부호화 슬라이스 데이터, 부호화 슬라이스 데이터에 포함되는 부호화 트리 유닛, 부호화 트리 유닛에 포함되는 부호화 유닛을 나타내는 도면이다.
- [0037] (부호화 비디오 시퀀스)
- [0038] 부호화 비디오 시퀀스에서는, 처리 대상의 시퀀스 SEQ를 복호하기 위해 동화상 복호 장치(31)가 참조하는 데이터의 집합이 규정되어 있다. 시퀀스 SEQ는 도 2에 나타내는 바와 같이, 비디오 파라미터 세트(Video Parameter Set), 시퀀스 파라미터 세트 SPS(Sequence Parameter Set), 픽처 파라미터 세트 PPS(Picture Parameter Set), Adaptation Parameter Set(APS), 픽처 PICT, 및 부가 확장 정보 SEI(Supplemental Enhancement Information)를 포함하고 있다.
- [0039] 비디오 파라미터 세트 VPS는 복수의 레이어로 구성되어 있는 동화상에서, 복수의 동화상에 공통되는 부호화 파라미터의 집합 및 동화상에 포함되는 복수의 레이어 및 개개의 레이어와 관련하는 부호화 파라미터의 집합이 규정되어 있다.
- [0040] 시퀀스 파라미터 세트 SPS에서는, 대상 시퀀스를 복호하기 위해 동화상 복호 장치(31)가 참조하는 부호화 파라미터의 집합이 규정되어 있다. 예를 들어, 픽처의 폭이나 높이가 규정된다. 아울러, SPS는 복수 존재할 수도 있다. 그 경우, PPS로부터 복수의 SPS 중 어느 것인가를 선택한다.
- [0041] 픽처 파라미터 세트 PPS에서는, 대상 시퀀스 내의 각 픽처를 복호하기 위해 동화상 복호 장치(31)가 참조하는 부호화 파라미터의 집합이 규정되어 있다. 예를 들어, 픽처의 복호에 이용되는 양자화 폭의 기준값(pic_init_qp_minus26)이나 가중 예측의 적용을 나타내는 플래그(weighted_pred_flag)가 포함된다. 아울러, PPS는 복수 존재할 수도 있다. 그 경우, 대상 시퀀스 내의 각 픽처로부터 복수의 PPS 중 어느 것인가를 선택한다.
- [0042] (부호화 픽처)
- [0043] 부호화 픽처에서는, 처리 대상의 픽처 PICT를 복호하기 위해 동화상 복호 장치(31)가 참조하는 데이터의 집합이 규정되어 있다. 픽처 PICT는 도 2에 나타내는 바와 같이, 슬라이스 0~슬라이스 NS-1을 포함한다(NS는 픽처 PICT에 포함되는 슬라이스의 총수).
- [0044] 아울러, 이하, 슬라이스 0~슬라이스 NS-1의 각각을 구별할 필요가 없는 경우, 부호의 첨자를 생략하고 기술하는 경우가 있다. 또한, 이하에 설명하는 부호화 스트림(Te)에 포함되는 데이터이며, 첨자를 붙이고 있는 다른 데

이터에 대해서도 마찬가지이다.

[0045] (부호화 슬라이스)

[0046] 부호화 슬라이스에서는, 처리 대상의 슬라이스 S를 복호하기 위해 동화상 복호 장치(31)가 참조하는 데이터의 집합이 규정되어 있다. 슬라이스는 도 2에 나타내는 바와 같이, 슬라이스 헤더 및 슬라이스 데이터를 포함하고 있다.

[0047] 슬라이스 헤더에는, 대상 슬라이스의 복호 방법을 결정하기 위해 동화상 복호 장치(31)가 참조하는 부호화 파라미터군이 포함된다. 슬라이스 타입을 지정하는 슬라이스 타입 지정 정보(slice_type)는 슬라이스 헤더에 포함되는 부호화 파라미터의 일 예이다.

[0048] 슬라이스 타입 지정 정보에 의해 지정 가능한 슬라이스 타입으로서, (1) 부호화 시에 인트라 예측만을 이용하는 I 슬라이스, (2) 부호화 시에 단방향 예측, 또는 인트라 예측을 이용하는 P 슬라이스, (3) 부호화 시에 단방향 예측, 양방향 예측, 또는 인트라 예측을 이용하는 B 슬라이스 등을 들 수 있다. 아울러, 인트라 예측은 단방향 예측, 양방향 예측으로 한정되지 않으며, 보다 많은 참조 픽처를 이용하여 예측 화상을 생성할 수도 있다. 이하, P, B 슬라이스라고 부르는 경우에는, 인트라 예측을 이용할 수 있는 블록을 포함하는 슬라이스를 가리킨다.

[0049] 아울러, 슬라이스 헤더는 픽처 파라미터 세트 PPS에의 참조(pic_parameter_set_id)를 포함하고 있을 수도 있다.

[0050] (부호화 슬라이스 데이터)

[0051] 부호화 슬라이스 데이터에서는, 처리 대상의 슬라이스 데이터를 복호하기 위해 동화상 복호 장치(31)가 참조하는 데이터의 집합이 규정되어 있다. 슬라이스 데이터는 도 2(d)에 나타내는 바와 같이, CTU를 포함하고 있다. CTU는 슬라이스를 구성하는 고정 사이즈(예를 들어 64x64)의 블록이며, 최대 부호화 단위(LCU: Largest Coding Unit)라고 부르는 경우도 있다.

[0052] (부호화 트리 유닛)

[0053] 도 2에는, 처리 대상의 CTU를 복호하기 위해 동화상 복호 장치(31)가 참조하는 데이터의 집합이 규정되어 있다. CTU는 재귀적인 쿼드 트리 분할(QT(Quad Tree) 분할), 이진 트리 분할(BT(Binary Tree) 분할) 혹은 삼항 트리 분할(TT(Ternary Tree) 분할)에 의해 부호화 처리의 기본적인 단위인 부호화 유닛 CU로 분할된다. BT 분할과 TT 분할을 합쳐 멀티 트리 분할(MT(Multi Tree) 분할)이라고 부른다. 재귀적인 쿼드 트리 분할에 의해 얻어지는 나무 구조의 노드를 부호화 노드(Coding Node)라고 칭한다. 쿼드 트리, 이진 트리 및 삼항 트리의 중간 노드는 부호화 노드이며, CTU 자신도 최상위의 부호화 노드로서 규정된다. 또한, 최하위의 부호화 노드는 부호화 유닛으로서 규정된다.

[0054] (부호화 유닛)

[0055] 도 2는 처리 대상의 부호화 유닛을 복호하기 위해 동화상 복호 장치(31)가 참조하는 데이터의 집합이 규정되어 있다. 구체적으로는, CU는 CU 헤더 CUH, 예측 파라미터, 변환 파라미터, 양자화 변환 계수 등으로 구성된다. CU 헤더에서는 예측 모드 등이 규정된다.

[0056] 예측 처리는 CU 단위로 수행되는 경우와, CU를 더 분할한 서브CU 단위로 수행되는 경우가 있다. CU와 서브CU의 사이즈가 동일한 경우에는, CU 중의 서브CU는 하나이다. CU가 서브CU의 사이즈보다 큰 경우, CU는 서브CU로 분할된다. 예를 들어 CU가 8x8, 서브CU가 4x4인 경우, CU는 수평 2분할, 수직 2분할로 이루어진 4개의 서브CU로 분할된다.

[0057] 예측의 종류(예측 모드)는 인트라 예측(MODE_INTRA)과 인트라 예측(MODE_INTER), 인트라 블록 카피(MODE_IBC)가 있다. 인트라 예측은 동일 픽처 내의 예측이고, 인트라 예측은 서로 상이한 픽처 사이(예를 들어, 표시 시각 사이, 레이어 화상 사이)에서 수행되는 예측 처리를 가리킨다.

[0058] 변환·양자화 처리는 CU 단위로 수행되는데, 양자화 변환 계수는 4x4 등의 서브블록 단위로 엔트로피 부호화할 수도 있다.

[0059] (예측 파라미터)

[0060] 예측 화상은 블록에 부수하는 예측 파라미터에 의해 도출된다. 예측 파라미터에는, 인트라 예측과 인트라 예측의 예측 파라미터가 있다.

- [0061] (인터 예측의 예측 파라미터)
- [0062] 인터 예측의 예측 파라미터에 대하여 설명한다. 인터 예측 파라미터는 예측 리스트 이용 플래그 predFlagL0과 predFlagL1, 참조 픽처 인덱스 refIdxL0과 refIdxL1, 움직임 벡터 mvL0과 mvL1로 구성된다. predFlagL0, predFlagL1은 참조 픽처 리스트(L0 리스트, L1 리스트)가 이용되는지 여부를 나타내는 플래그이며, 값이 1인 경우에 대응하는 참조 픽처 리스트가 이용된다. 아울러, 본 명세서 중 「XX인지 여부를 나타내는 플래그」라고 기재하는 경우, 플래그가 0 이외(예를 들어 1)을 XX인 경우, 0을 XX가 아닌 경우로 하여, 논리 부정, 논리곱 등에서는 1을 참, 0을 거짓으로 취급한다(이하 동일). 다만, 실제 장치나 방법에서는 참값, 거짓값으로서 다른 값을 이용할 수도 있다.
- [0063] 인터 예측 파라미터를 도출하기 위한 신택스 요소에는, 예를 들어 머지 플래그 merge_flag(general_merge_flag), 머지 인덱스 merge_idx, merge_subblock_flag, regular_merge_flag, ciip_flag, merge_gpm_partition_idx, merge_gpm_idx0, merge_gpm_idx1, inter_pred_idc, 참조 픽처 인덱스 refIdxLX, mvp_LX_idx, 차분 벡터 mvdLX, 움직임 벡터 정밀도 모드 amvr_mode가 있다. merge_subblock_flag는 서브블록 단위의 인터 예측을 이용하느냐를 나타내는 플래그이다. regular_merge_flag는 통상 머지 모드 혹은 MMVD가 사용되는지를 나타내는 플래그이다. ciip_flag는 CIIP(combined inter-picture merge and intra-picture prediction) 모드를 이용하느냐 여부를 나타내는 플래그이다. merge_gpm_partition_idx는 GPM 모드의 분할 형상을 나타내는 인덱스이다. merge_gpm_idx0, merge_gpm_idx1은 GPM 모드의 머지 인덱스를 나타내는 인덱스이다. inter_pred_idc는 AMVP 모드에서 이용하는 참조 픽처를 선택하기 위한 인터 예측 식별자이다. mvp_LX_idx는 움직임 벡터를 도출하기 위한 예측 벡터 인덱스이다.
- [0064] (참조 픽처 리스트)
- [0065] 참조 픽처 리스트는 참조 픽처 메모리(306)에 저장된 참조 픽처로 이루어진 리스트이다. 개개의 CU에서는, 참조 픽처 리스트 RefPicListX(X=0 또는 1) 중의 어느 픽처를 실제로 참조할지를 refIdxLX에서 지정한다. 아울러, LX는 L0 예측과 L1 예측을 구별하지 않는 경우에 이용되는 기술 방법이며, 이후에서는, LX를 L0, L1로 치환함으로써 L0 리스트에 대한 파라미터와 L1 리스트에 대한 파라미터를 구별한다.
- [0066] (머지 예측과 AMVP 예측)
- [0067] 예측 파라미터의 복호(부호화) 방법에는, 머지 예측(merge) 모드(머지 모드)와 AMVP(Advanced Motion Vector Prediction, 적응 움직임 벡터 예측) 모드가 있으며, general_merge_flag는 이들을 식별하기 위한 플래그이다. 머지 모드는 움직임 벡터 차분의 일부 혹은 전부를 생략하는 예측 모드이며, 예측 리스트 이용 플래그 predFlagLX, 참조 픽처 인덱스 refIdxLX, 움직임 벡터 mvLX를 부호화 데이터에 포함시키지 않고, 이미 처리한 근방 블록의 예측 파라미터 등으로부터 도출하는 모드이다. AMVP 모드는 inter_pred_idc, refIdxLX, mvLX를 부호화 데이터에 포함시키는 모드이다. 아울러, mvLX는 예측 벡터 mvpLX를 식별하는 mvp_LX_idx와 차분 벡터 mvdLX로서 부호화된다. 또한 움직임 벡터 차분을 생략·간소화하는 예측 모드의 총칭을 일반 머지 모드(general merge mode)라고 부르며, general_merge_flag에 의해 일반 머지 모드와 AMVP 예측을 선택할 수도 있다.
- [0068] general_merge_flag가 1인 경우, 별도 regular_merge_flag를 전송할 수도 있다. regular_merge_flag가 1인 경우에, 통상 머지 모드 혹은 MMVD를 선택하고, 그 이외의 경우에 CIIP 모드 혹은 GPM 모드를 선택할 수도 있다. CIIP 모드는 인터 예측 화상과 인트라 예측 화상의 가중합에 의해 예측 화상을 생성한다. GPM 모드는 대상 CU를 선분으로 분할되는 2개의 비직사각형의 예측 단위로서 예측 화상을 생성한다.
- [0069] inter_pred_idc는 참조 픽처의 종류 및 수를 나타내는 값이며, PRED_L0, PRED_L1, PRED_BI의 어느 것인가의 값을 취한다. PRED_L0, PRED_L1은 각각 L0 리스트, L1 리스트에서 관리된 1매의 참조 픽처를 이용하는 단예측을 나타낸다. PRED_BI는 L0 리스트와 L1 리스트에서 관리된 2매의 참조 픽처를 이용하는 양예측을 나타낸다.
- [0070] merge_idx는 처리가 완료된 블록으로부터 도출되는 예측 파라미터 후보(머지 후보) 중, 어느 예측 파라미터를 대상 블록의 예측 파라미터로서 이용할지를 나타내는 인덱스이다.
- [0071] (움직임 벡터)
- [0072] mvLX는 상이한 2개의 픽처상의 블록 사이의 시프트량을 나타낸다. mvLX에 관한 예측 벡터, 차분 벡터를 각각 mvpLX, mvdLX라고 부른다.

- [0073] (인터 예측 식별자 `inter_pred_idc`와 예측 리스트 이용 플래그 `predFlagLX`)
- [0074] `inter_pred_idc`와 `predFlagL0`, `predFlagL1`의 관계는 이하와 같으며, 상호 변환 가능하다.
- [0075]
$$\text{inter_pred_idc} = (\text{predFlagL1} < 1) + \text{predFlagL0}$$
- [0076]
$$\text{predFlagL0} = \text{inter_pred_idc} \& 1$$
- [0077]
$$\text{predFlagL1} = \text{inter_pred_idc} >> 1$$
- [0078] 아울러, 인터 예측 파라미터는 예측 리스트 이용 플래그를 이용할 수도 있고, 인터 예측 식별자를 이용할 수도 있다. 또한, 예측 리스트 이용 플래그를 이용한 판정은 인터 예측 식별자를 이용한 판정으로 치환할 수도 있다. 반대로, 인터 예측 식별자를 이용한 판정은 예측 리스트 이용 플래그를 이용한 판정으로 치환할 수도 있다.
- [0079] (양예측 biPred의 판정)
- [0080] 양예측인지의 플래그 `biPred`는 2개의 예측 리스트 이용 플래그가 모두 1인지에 의해 도출할 수 있다. 예를 들어 이하의 식으로 도출할 수 있다.
- [0081]
$$\text{biPred} = (\text{predFlagL0} == 1 \&\& \text{predFlagL1} == 1)$$
- [0082] 혹은, `biPred`는 인터 예측 식별자가 2개의 예측 리스트(참조 픽처)를 사용하는 것을 나타내는 값인지 여부에 의해서도 도출할 수 있다. 예를 들어 이하의 식으로 도출할 수 있다.
- [0083]
$$\text{biPred} = (\text{inter_pred_idc} == \text{PRED_BI}) ? 1 : 0$$
- [0084] (동화상 복호 장치의 구성)
- [0085] 본 실시형태에 관한 동화상 복호 장치(31)(도 3)의 구성에 대하여 설명한다.
- [0086] 동화상 복호 장치(31)는 엔트로피 복호부(301), 파라미터 복호부(예측 화상 복호 장치)(302), 루프 필터(305), 참조 픽처 메모리(306), 예측 파라미터 메모리(307), 예측 화상 생성부(예측 화상 생성 장치)(308), 역양자화·역변환부(311), 및 가산부(312), 예측 파라미터 도출부(320)를 포함하여 구성된다. 아울러, 후술하는 동화상 부호화 장치(11)에 맞추어, 동화상 복호 장치(31)에 루프 필터(305)가 포함되지 않는 구성도 있다.
- [0087] 파라미터 복호부(302)는 또한, 헤더 복호부(3020), CT 정보 복호부(3021) 및 CU 복호부(3022)(예측 모드 복호부)를 구비하고 있으며, CU 복호부(3022)는 또한 TU 복호부(3024)를 구비하고 있다. 이들을 총칭하여 복호 모듈이라고 부를 수도 있다. 헤더 복호부(3020)는 부호화 데이터로부터 VPS, SPS, PPS, APS 등의 파라미터 세트 정보, 슬라이스 헤더(슬라이스 정보)를 복호한다. CT 정보 복호부(3021)는 부호화 데이터로부터 CT를 복호한다. CU 복호부(3022)는 부호화 데이터로부터 CU를 복호한다. TU 복호부(3024)는 부호화 데이터로부터 CU를 복호한다.
- [0088] TU 복호부(3024)는 TU에 예측 오차가 포함되어 있는 경우, 부호화 데이터로부터 QP 갱신 정보와 양자화 변환 계수를 복호한다. QP 갱신 정보는 양자화 파라미터 QP의 예측값인 양자화 파라미터 예측값 `qPred`로부터의 차분값이다.
- [0089] 예측 화상 생성부(308)는 인터 예측 화상 생성부(309)(도 4) 및 인트라 예측 화상 생성부(310)를 포함하여 구성된다.
- [0090] 예측 파라미터 도출부(320)는 인터 예측 파라미터 도출부(303)(도 5) 및 인트라 예측 파라미터 도출부를 포함하여 구성된다.
- [0091] 또한, 이후에서는 처리의 단위로서 CTU, CU를 사용한 예를 기재하지만, 이 예로 한정되지 않으며, 서브CU 단위로 처리를 할 수도 있다. 혹은 CTU, CU를 블록, 서브CU를 서브블록으로 바꾸어 읽고, 블록 혹은 서브블록 단위의 처리로 할 수도 있다.
- [0092] 엔트로피 복호부(301)는 외부로부터 입력된 부호화 스트림(Te)에 대해 엔트로피 복호를 수행하여, 개개의 부호(신텍스 요소)를 복호한다. 엔트로피 부호화에는, 신텍스 요소의 종류나 주위 상황에 따라 적응적으로 선택한 콘텍스트(확률 모델)를 이용하여 신텍스 요소를 가변 길이 부호화하는 방식과, 미리 정해진 표 혹은 계산식을 이용하여 신텍스 요소를 가변 길이 부호화하는 방식이 있다.

- [0093] 엔트로피 복호부(301)는 복호한 부호를 파라미터 복호부(302)에 출력한다. 복호한 부호란, 예를 들어 예측 모드 `predMode`, `general_merge_flag`, `merge_idx`, `inter_pred_idc`, `refIdxLX`, `mvp_LX_idx`, `mvdLX`, `amvr_mode` 등이다. 어느 부호를 복호할지의 제어는 파라미터 복호부(302)의 지시를 기초로 수행된다.
- [0094] (기본 플로우)
- [0095] 도 6은 동화상 복호 장치(31)의 개략적 동작을 설명하는 흐름도이다.
- [0096] (S1100: 파라미터 세트 정보 복호) 헤더 복호부(3020)는 부호화 데이터로부터 VPS, SPS, PPS 등의 파라미터 세트 정보를 복호한다.
- [0097] (S1200: 슬라이스 정보 복호) 헤더 복호부(3020)는 부호화 데이터로부터 슬라이스 헤더(슬라이스 정보)를 복호한다.
- [0098] 이하, 동화상 복호 장치(31)는 대상 픽처에 포함되는 각 CTU에 대해, S1300부터 S5000의 처리를 반복함으로써 각 CTU의 복호 화상을 도출한다.
- [0099] (S1300: CTU 정보 복호) CT 정보 복호부(3021)는 부호화 데이터로부터 CTU를 복호한다.
- [0100] (S1400: CT 정보 복호) CT 정보 복호부(3021)는 부호화 데이터로부터 CT를 복호한다.
- [0101] (S1500: CU 복호) CU 복호부(3022)는 S1510, S1520을 실시하여, 부호화 데이터로부터 CU를 복호한다.
- [0102] (S1510: CU 정보 복호) CU 복호부(3022)는 부호화 데이터로부터 CU 정보, 예측 정보, TU 분할 플래그, CU 잔차 플래그 등을 복호한다.
- [0103] (S1520: TU 정보 복호) TU 복호부(3024)는 TU에 예측 오차가 포함되어 있는 경우에, 부호화 데이터로부터 양자화 예측 오차 등을 복호한다.
- [0104] (S2000: 예측 화상 생성) 예측 화상 생성부(308)는 대상 CU에 포함되는 각 블록에 대해, 예측 정보를 기초로 예측 화상을 생성한다.
- [0105] (S3000: 역양자화·역변환) 역양자화·역변환부(311)는 대상 CU에 포함되는 각 TU에 대해, 역양자화·역변환 처리를 실행한다.
- [0106] (S4000: 복호 화상 생성) 가산부(312)는 예측 화상 생성부(308)로부터 공급되는 예측 화상과, 역양자화·역변환부(311)로부터 공급되는 예측 오차를 가산함으로써, 대상 CU의 복호 화상을 생성한다.
- [0107] (S5000: 루프 필터) 루프 필터(305)는 복호 화상에 디블록킹 필터, SAO, ALF 등의 루프 필터를 걸어 복호 화상을 생성한다.
- [0108] 루프 필터(305)는 부호화 루프 내에 마련한 필터이며, 블록 왜곡이나 링잉 왜곡을 제거하여, 화질을 개선하는 필터이다. 루프 필터(305)는 가산부(312)가 생성한 CU의 복호 화상에 대해, 디블록킹 필터, 샘플 적응 오프셋(SAO), 적응 루프 필터(ALF) 등의 필터를 가한다.
- [0109] 참조 픽처 메모리(306)는 가산부(312)가 생성한 CU의 복호 화상을 대상 픽처 및 대상 CU별로 미리 정한 위치에 저장한다.
- [0110] 예측 파라미터 메모리(307)는 복호 대상의 CTU 혹은 CU별로 미리 정한 위치에 예측 파라미터를 저장한다. 구체적으로는, 예측 파라미터 메모리(307)는 파라미터 복호부(302)가 복호한 파라미터 및 엔트로피 복호부(301)가 분리한 예측 모드 `predMode` 등을 저장한다.
- [0111] 예측 화상 생성부(308)에는, `predMode`, 예측 파라미터 등이 입력된다. 또한, 예측 화상 생성부(308)는 참조 픽처 메모리(306)로부터 참조 픽처를 읽어낸다. 예측 화상 생성부(308)는 `predMode`가 나타내는 예측 모드로, 예측 파라미터와 읽어낸 참조 픽처(참조 블록)를 이용하여 블록 혹은 서브블록의 예측 화상을 생성한다. 여기서, 참조 블록이란, 참조 픽처상의 화소의 집합(통상 직사각형이므로 블록이라고 부른다)이며, 예측 화상을 생성하기 위해 참조하는 영역이다.
- [0112] (인터 예측 파라미터 도출부의 구성)
- [0113] 도 5에 나타내는 바와 같이, 인터 예측 파라미터 도출부(303)는 파라미터 복호부(302)로부터 입력된 선택스 요소를 기초로, 예측 파라미터 메모리(307)에 저장된 예측 파라미터를 참조하여 인터 예측 파라미터를 도출한다.

또한, 인터 예측 파라미터를 인터 예측 화상 생성부(309), 예측 파라미터 메모리(307)에 출력한다. 인터 예측 파라미터 도출부(303) 및 그 내부의 요소인 AMVP 예측 파라미터 도출부(3032), 머지 예측 파라미터 도출부(3036), MMVD 예측부(30376), MV 가산부(3038)는 동화상 부호화 장치, 동화상 복호 장치에서 공통되는 수단이므로, 이들을 총칭하여 움직임 벡터 도출부(움직임 벡터 도출 장치)라고 칭할 수도 있다.

- [0114] general_merge_flag가 1, 즉 머지 예측 모드를 나타내는 경우, merge_idx를 도출하여, 머지 예측 파라미터 도출부(3036)에 출력한다.
- [0115] general_merge_flag가 0, 즉 AMVP 예측 모드를 나타내는 경우, AMVP 예측 파라미터 도출부(3032)는 inter_pred_idc, refIdxLX나 mvp_LX_idx로부터 mvpLX를 도출한다.
- [0116] (MV 가산부)
- [0117] MV 가산부(3038)에서는 도출된 mvpLX와 mvdLX를 가산하여, mvLX를 도출한다.
- [0118] (머지 예측)
- [0119] 머지 예측 파라미터 도출부(3036)는 예측 파라미터(predFlagLX, mvLX, refIdxLX)를 포함하여 구성되는 머지 후보를 도출하여, 머지 후보 리스트를 구성한다. 또한 머지 예측 파라미터 도출부(3036)는 머지 후보 리스트에 포함되는 머지 후보 중, merge_idx가 나타내는 머지 후보 N의 움직임 정보(mvLXN[0], mvLXN[1]), predFlagLXN, refIdxLXN을 대상 블록의 인터 예측 파라미터로서 선택한다. 머지 예측 파라미터 도출부(3036)는 선택한 머지 후보의 인터 예측 파라미터를 예측 파라미터 메모리(307)에 저장하는 동시에, 인터 예측 화상 생성부(309)에 출력한다.
- [0120] (인터 예측 화상 생성부(309))
- [0121] predMode가 인터 예측을 나타내는 경우, 인터 예측 화상 생성부(309)는 인터 예측 파라미터 도출부(303)로부터 입력된 인터 예측 파라미터와 참조 픽처를 이용하여 인터 예측에 의해 블록 혹은 서브블록의 예측 화상을 생성한다.
- [0122] 도 4는 본 실시형태에 관한 예측 화상 생성부(308)에 포함되는 인터 예측 화상 생성부(309)의 구성을 나타내는 개략도이다. 인터 예측 화상 생성부(309)는 움직임 보상부(예측 화상 생성 장치)(3091), OOB 처리부(3092), 합성부(3095)를 포함하여 구성된다. 합성부(3095)는 무게 예측부(3094)를 포함하여 구성된다.
- [0123] (움직임 보상)
- [0124] 움직임 보상부(3091)(보간 화상 생성부(3091))는 인터 예측 파라미터 도출부(303)로부터 입력된 인터 예측 파라미터(predFlagLX, refIdxLX, mvLX)를 기초로, 참조 픽처 메모리(306)로부터 참조 블록을 읽어냄으로써 보간 화상(움직임 보상 화상)을 생성한다. 참조 블록은 refIdxLX에서 지정된 참조 픽처 RefPicLX상에서, 대상 블록의 위치로부터 mvLX 시프트한 위치의 블록이다. 여기서, mvLX가 정수 정밀도가 아닌 경우에는, 움직임 보상 필터로 불리는 소수 위치의 화소를 생성하기 위한 필터를 가하여, 보간 화상을 생성한다.
- [0125] 움직임 보상부(3091)는 $bW \times bH$ 사이즈의 블록의 좌상 좌표(xPb , yPb), 예측 블록 내 좌표(xL , yL), 움직임 벡터($mvLX[0]$, $mvLX[1]$)에 대응하는 정수 위치($xInt$, $yInt$) 및 위상($xFrac$, $yFrac$)을 이하의 식으로 도출한다(MC-P1).
- [0126]
$$xInt = xPb + (mvLX[0] \gg (\log_2 MVPREC)) + xL$$
- [0127]
$$xFrac = mvLX[0] \& (MVPREC - 1)$$
- [0128]
$$yInt = yPb + (mvLX[1] \gg (\log_2 MVPREC)) + yL$$
- [0129]
$$yFrac = mvLX[1] \& (MVPREC - 1)$$
- [0130] 여기서, MVPREC는 mvLX의 정밀도($1/MVPREC$ 화소 정밀도), $\log_2 MVPREC = \log_2(MVPREC)$, $x = 0 \dots bW - 1$, $y = 0 \dots bH - 1$ 을 나타낸다. 예를 들어 MVPREC=16일 수도 있다.
- [0131] 아울러, RPR을 수행하기 위해, 후술하는 (MC-P2)와 같이 도출할 수 있다. 또한, (MC-P1), (MC-P2)로 도출된 ($xInt$, $yInt$)에 대해, 움직임 보상부(3091)는 랩 어라운드를 위해 위치를 수정할 수도 있다.
- [0132] 서브픽처 경계를 픽처 경계로 하는 플래그가 유효인 경우($sps_subpic_treated_as_pic_flag == 1$)인 동시에, 참조 화상의 서브픽처의 수가 1보다 큰 경우(참조 화상 refPicLX를 위한 $sps_num_subpics_minus1 > 0$), 하기 식

이 성립된다(MC-P2).

- [0133] $xInt = Clip3(SubpicLeftBoundaryPos, SubpicRightBoundaryPos, refWraparoundEnabledFlag ? ClipH((PpsRefWraparoundOffset) * MinCbSizeY, picW, xInt) : xInt)$
- [0134] $yInt = Clip3(SubpicTopBoundaryPos, SubpicBotBoundaryPos, yInt)$
- [0135] 여기서, SubpicLeftBoundaryPos, SubpicRightBoundaryPos, SubpicTopBoundaryPos, SubpicBottomBoundaryPos는 각각 서브픽처의 좌, 우, 상, 하의 경계 위치.
- [0136] 상기 이외의 경우(서브픽처 경계를 픽처 경계로 하는 플래그가 무효인 경우(sps_subpic_treated_as_pic_flag == 0) 혹은, 참조 화상의 서브픽처의 수가 1인 경우(참조 화상 refPicLX를 위한 sps_num_subpics_minus1 == 0), 하기 식이 성립한다(MC-P2).
- [0137] $xInt = Clip3(0, picW - 1, refWraparoundEnabledFlag ? ClipH((PpsRefWraparoundOffset) * MinCbSizeY, picW, xInt) : xInt)$
- [0138] $yInt = Clip3(0, picH - 1, yInt)$
- [0139] 여기서, 각 파라미터를 이하와 같이 설정한다.
- [0140] $refWraparoundEnabledFlag = pps_ref_wraparound_enabled_flag \&\& ! refPicIsScaled$
- [0141] $PpsRefWraparoundOffset = \frac{pps_pic_width_in_luma_samples}{MinCbSizeY} - pps_pic_width_minus_wraparound_offset$
- [0142] MinCbSizeY는 소정의 상수 혹은 변수(예를 들어 4), pps_pic_width_minus_wraparound_offset은 랩 어라운드 위치를 나타내는 부호화 데이터로부터 복호되는 오프셋.
- [0143] 움직임 보상부(3091)는 참조 픽처 refImg에 보간 필터를 이용하여 수평 보간 처리를 수행함으로써, 일시적 화상 temp[[]]를 도출한다. 이하의 Σ 는 $k=0..NTAP-1$ 의 k 에 관한 합, mcFilter[Frac][k]는 위상 Frac에서의 k 번째 보간 필터 계수이며, shift1은 값의 레인지를 조정하는 정규화 파라미터, offset1= $1 \ll (\text{shift1}-1)$ 이다.
- [0144] $temp[x][y] = (\Sigma mcFilter[xFrac][k] * refImg[xInt+k-NTAP/2+1][yInt]+offset1) \gg shift1$
- [0145] 계속해서, 움직임 보상부(3091)는 일시적 화상 temp[[]]를 수직 보간 처리에 의해, 보간 화상 Pred[[]]를 도출한다. 이하의 Σ 는 $k=0..NTAP-1$ 의 k 에 관한 합, shift2는 값의 레인지를 조정하는 정규화 파라미터, offset2= $1 \ll (\text{shift2}-1)$ 이다.
- [0146] $Pred[x][y] = (\Sigma mcFilter[yFrac][k] * temp[x][y+k-NTAP/2+1]+offset2) \gg shift2$ (식 MC-1)
- [0147] (무게 예측)
- [0148] 무게 예측부(3094)는 보간 화상 PredLX에 무게 계수를 곱셈함으로써 블록의 예측 화상을 생성한다.
- [0149] (픽처 범위 외(OOB) 처리)
- [0150] OOB 처리부(3092)는 OOB 판정부(30921) 및 OOB 마스크 도출부(30922)를 포함한다. OOB 판정부(30921)는 인터 예측 파라미터 도출부(303)로부터 입력된 인터 예측 파라미터(predFlagLX, refIdxLX, mvLX 등)를 기초로, 양예측 모드의 각 참조 블록이 참조 픽처의 유효 범위 외(OOB: Out-Of-Boundary)의 영역을 포함하는지 여부를 판정한다.
- [0151] 아울러, 상기 판정은 픽처 혹은 서브픽처의 유효 범위 외의 영역을 포함하는지 여부를 판정할 수도 있다. 또한, 슬라이스 외의 영역을 포함하는지 여부를 이용하여 판정할 수도 있다.
- [0152] 도 9를 참조하여 설명한다.
- [0153] OOB 판정부(30921)가 참조 블록의 일부 화소가 픽처 외(유효 범위 외), 즉 참조 블록이 OOB라고 판정한 경우(isOOB==true, S1002), OOB 마스크 도출부(30922)는 참조 블록의 어느 영역이 유효 범위 외인지를 나타내는 마스크 데이터(마스크값, OOB 마스크)를 도출한다(S1004). 인터 예측 화상 생성부(309)는 OOB인 참조 블록에 대해, OOB 마스크를 적용하여 예측 화상을 생성한다(S1006). 인터 예측 화상 생성부(309)는 OOB가 아닌 참조 블록에 대해, 통상의 예측 화상을 생성한다(S1008). 이하에서 상세를 설명한다.

- [0154] (OOB 판정)
- [0155] OOB 판정부(30921)는 양예측에 있어서, 2개(제1 및 제2)의 참조 블록에 대해, 각 참조 블록의 일부가 픽처 외(OOB)인지 여부를 판정한다. 참조 화상 refPicLX에 대응하는 LX 참조 블록(X=0 또는 1)의 영역은 블록 좌상 좌표(xRefLX, yRefLX)와 블록 사이즈(폭 및 높이: bW, bH)로 표시된다. 픽처의 폭과 높이를 각각 picW, picH로 한다. 아울러, 이하에서는 mvLX의 정밀도 MVPREC에 따라 mvLX를 정수화한 값을 좌표계에 이용한다. 이들 값을 이용하여, OOB 판정부(30921)는 다음 식에 의해 LX 참조 블록이 픽처 외(유효 범위 외)를 포함하는지 여부를 나타내는 진리값 isOOB[X]를 도출한다. 또한 유효 범위 내의 영역을 유효 영역이라고 기재한다.
- [0156] 도 10에 일 예를 나타낸다. 도 10은 정수 단위의 도면이며, 움직임 벡터 mvLX', 참조 화소 위치(xRefLX', yRefLX')는 움직임 벡터 mvLX, 참조 화소 위치(xRefLX, yRefLX)를 정수 표현한 파라미터이다. 도면의 예는 X=0이지만, X=1인 경우도 마찬가지이다.
- [0157] $xRefLX = (xPb \ll \log 2 MVPREC) + mvLX[0]$ (DecideOOB1)
- [0158] $yRefLX = (yPb \ll \log 2 MVPREC) + mvLX[1]$
- [0159] $picMinX = 0 - MVPREC/2$
- [0160] $picMinY = 0 - MVPREC/2$
- [0161] $picMaxX = ((picW-1) \ll \log 2 MVPREC) + MVPREC/2$
- [0162] $picMaxY = ((picH-1) \ll \log 2 MVPREC) + MVPREC/2$
- [0163] $isOOB[X] = (xRefLX < picMinX) \mid \mid (yRefLX < picMinY) \mid \mid$
- [0164] $(xRefLX + (bW \ll \log 2 MVPREC) - 1 \geq picMaxX) \mid \mid$
- [0165] $(yRefLX + (bH \ll \log 2 MVPREC) - 1 \geq picMaxY)$
- [0166] $\log 2 MVPREC$ 비트 좌측 시프트(" $\ll \log 2 MVPREC$ ")를 MVPREC와의 곱셈(" $* MVPREC$ ")으로 처리할 수도 있고, 후술하는 " $* MVPREC$ "를 " $\ll \log 2 MVPREC$ "로 처리할 수도 있다.
- [0167] 혹은, (DecideOOB1)에서의 isOOB[X]의 도출식은 하기 식일 수도 있다.
- [0168] $isOOB[X] = (xRefLX \leq picMinX) \mid \mid (yRefLX \leq picMinY) \mid \mid$
- [0169] $(xRefLX + (bW \ll \log 2 MVPREC) - 1 \geq picMaxX) \mid \mid$
- [0170] $(yRefLX + (bH \ll \log 2 MVPREC) - 1 \geq picMaxY)$
- [0171] 혹은, OOB 판정부(30921)는 반올림을 위한 항을 생략하고, 다음과 같이 간략하게 판정할 수도 있다.
- [0172] $picMaxX = (picW-1) \ll \log 2 MVPREC$ (DecideOOB2)
- [0173] $picMaxY = (picH-1) \ll \log 2 MVPREC$
- [0174] $isOOB[X] = (xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0175] $(xRefLX + (bW \ll \log 2 MVPREC) - 1 > picMaxX) \mid \mid$
- [0176] $(yRefLX + (bH \ll \log 2 MVPREC) - 1 > picMaxY)$
- [0177] 또한, OOB 판정부(30921)는 소수 단위의 경계 좌표가 아니라 정수 단위의 좌표로 판정을 수행할 수도 있다.
- [0178] $xRefLX = xPb + (mvLX[0] \gg \log 2 MVPREC)$ (DecideOOB3)
- [0179] $yRefLX = yPb + (mvLX[1] \gg \log 2 MVPREC)$
- [0180] $isOOB[X] = (xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0181] $(xRefLX + bW - 1 \geq picW-1) \mid \mid$
- [0182] $(yRefLX + bH - 1 \geq picH-1)$
- [0183] 혹은, (DecideOOB3)에서의 isOOB[X]의 도출식은 하기 식일 수도 있다.

[0184] $isOOB[X] = (xRefLX \leq 0) \mid \mid (yRefLX \leq 0) \mid \mid$

[0185] $(xRefLX + bW - 1 \geq picW-1) \mid \mid$

[0186] $(yRefLX + bH - 1 \geq picH-1)$

[0187] 상기에 따르면, 움직임 벡터 $mvLX[]$ 의 MV 정밀도에 의한 우측 시프트에 의해 얻어지는 정수 좌표를 이용하기 때문에, 소수 정밀도로 판정하는 것보다 처리량을 간략화할 수 있다는 효과를 나타낸다.

[0188] OOB 판정부(30921)는 블록 사이즈에 따라 OOB 처리를 적용하는지 여부를 전환할 수도 있다. 예를 들어, OOB 판정부(30921)가 $isOOB[X]$ 의 도출에 다음 식을 이용하여, 폭 및 높이가 모두 $BLKTHW1$, $BLKTHH1$ 화소 이상인 블록에만 $isOOB=true$ 로 설정하고, OOB 처리를 적용할 수도 있다.

[0189] $isOOB[X] = ((bW \geq BLKTHW1) \&\& (bH \geq BLKTHH1)) \&\&$

[0190] $((xRefLX < picMinX) \mid \mid (yRefLX < picMinY) \mid \mid$

[0191] $(xRefLX + bW * MVPREC - 1 \geq picMaxX) \mid \mid$

[0192] $(yRefLX + bH * MVPREC - 1 \geq picMaxY))$

[0193] 여기서 OOB 처리를 적용한다란, $isOOB[X]=true$ 로 하는 것이다. 무게 예측부(3094)에서 어느 것인가의 참조 화상을 가중하지 않고 우선하여 이용해 예측 화상 $Pred[][]$ 를 도출하는 것을 의미한다.

[0194] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.

[0195] $isOOB[X] = (bW \geq BLKTHW1) \&\& (bH \geq BLKTHH1) \&\&$

[0196] $((xRefLX \leq picMinX) \mid \mid (yRefLX \leq picMinY) \mid \mid$

[0197] $(xRefLX + (bW << \log 2 MVPREC) - 1 \geq picMaxX) \mid \mid$

[0198] $(yRefLX + (bH << \log 2 MVPREC) - 1 \geq picMaxY))$

[0199] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.

[0200] $picMaxX = (picW-1) << \log 2 MVPREC$

[0201] $picMaxY = (picH-1) << \log 2 MVPREC$

[0202] $isOOB[X] = (bW \geq BLKTHW1) \&\& (bH \geq BLKTHH1) \&\&$

[0203] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$

[0204] $(xRefLX + (bW << \log 2 MVPREC) - 1 > picMaxX) \mid \mid$

[0205] $(yRefLX + (bH << \log 2 MVPREC) - 1 > picMaxY))$

[0206] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.

[0207] $xRefLX = xPb + (mvLX[0] >> \log 2 MVPREC)$

[0208] $yRefLX = yPb + (mvLX[1] >> \log 2 MVPREC)$

[0209] $isOOB[X] = (bW \geq BLKTHW1) \&\& (bH \geq BLKTHH1) \&\&$

[0210] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$

[0211] $(xRefLX + bW - 1 \geq picW-1) \mid \mid$

[0212] $(yRefLX + bH - 1 \geq picH-1))$

[0213] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.

[0214] $isOOB[X] = (bW \geq BLKTHW1) \&\& (bH \geq BLKTHH1) \&\&$

[0215] $((xRefLX \leq 0) \mid \mid (yRefLX \leq 0) \mid \mid$

[0216] $(xRefLX + bW - 1 \geq picW-1) \mid \mid$

- [0217] $(yRefLX + bH - 1 \geq picH - 1)$
- [0218] 예를 들어, BLKTHW1=4, 8, 16, 32, BLKTHH1=4, 8, 16, 32 등일 수도 있다. OOB를 적용하는 블록 사이즈의 임계값은 8x8, 4x4, 4x8, 8x4, 16x16 등일 수도 있다.
- [0219] 혹은 큰 블록에서 OOB 처리를 금지할 수도 있다.
- [0220] $isOOB[X] = (bW \leq BLKTHW2) \&\& (bH \leq BLKTHW2) \&\&$
- [0221] $((xRefLX < picMinX) \mid \mid (yRefLX < picMinY) \mid \mid$
- [0222] $(xRefLX + bW * MVPREC - 1 \geq picMaxX) \mid \mid$
- [0223] $(yRefLX + bH * MVPREC - 1 \geq picMaxY))$
- [0224] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.
- [0225] $isOOB[X] = (bW \leq BLKTHW2) \&\& (bH \leq BLKTHW2) \&\&$
- [0226] $((xRefLX \leq picMinX) \mid \mid (yRefLX \leq picMinY) \mid \mid$
- [0227] $(xRefLX + (bW << \log2MVPREC) - 1 \geq picMaxX) \mid \mid$
- [0228] $(yRefLX + (bH << \log2MVPREC) - 1 \geq picMaxY))$
- [0229] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.
- [0230] $picMaxX = (picW - 1) << \log2MVPREC$
- [0231] $picMaxY = (picH - 1) << \log2MVPREC$
- [0232] $isOOB[X] = (bW \leq BLKTHW2) \&\& (bH \leq BLKTHW2) \&\&$
- [0233] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0234] $(xRefLX + (bW << \log2MVPREC) - 1 > picMaxX) \mid \mid$
- [0235] $(yRefLX + (bH << \log2MVPREC) - 1 > picMaxY))$
- [0236] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.
- [0237] $xRefLX = xPb + (mvLX[0] >> \log2MVPREC)$
- [0238] $yRefLX = yPb + (mvLX[1] >> \log2MVPREC)$
- [0239] $isOOB[X] = (bW \leq BLKTHW2) \&\& (bH \leq BLKTHW2) \&\&$
- [0240] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0241] $(xRefLX + bW - 1 \geq picW - 1) \mid \mid$
- [0242] $(yRefLX + bH - 1 \geq picH - 1))$
- [0243] 혹은 하기 식으로 $isOOB[X]$ 를 도출할 수도 있다.
- [0244] $xRefLX = xPb + (mvLX[0] >> \log2MVPREC)$
- [0245] $yRefLX = yPb + (mvLX[1] >> \log2MVPREC)$
- [0246] $isOOB[X] = (bW \leq BLKTHW2) \&\& (bH \leq BLKTHW2) \&\&$
- [0247] $((xRefLX \leq 0) \mid \mid (yRefLX \leq 0) \mid \mid$
- [0248] $(xRefLX + bW - 1 \geq picW - 1) \mid \mid$
- [0249] $(yRefLX + bH - 1 \geq picH - 1))$
- [0250] 예를 들어, BLKTHW2=16, 32, 64, BLKTHH2=16, 32, 64 등일 수도 있다.
- [0251] 또는 특정 범위의 크기의 블록에만 OOB 처리가 적용되도록 할 수도 있다.

- [0252] $isOoB[X] = (bW > BLKTHW3) \&\& (bW \leq BLKTHW4) \&\& (bH > BLKTHH3) \&\& (bH \leq BLKTHH4) \&\&$
- [0253] $((xRefLX < picMinX) \mid \mid (yRefLX < picMinY) \mid \mid$
- [0254] $(xRefLX + bW * MVPREC - 1 \geq picMaxX) \mid \mid$
- [0255] $(yRefLX + bH * MVPREC - 1 \geq picMaxY))$
- [0256] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.
- [0257] $isOoB[X] = (bW > BLKTHW3) \&\& (bW \leq BLKTHW4) \&\& (bH > BLKTHH3) \&\& (bH \leq BLKTHH4) \&\&$
- [0258] $((xRefLX \leq picMinX) \mid \mid (yRefLX \leq picMinY) \mid \mid$
- [0259] $(xRefLX + (bW < \log_2 MVPREC) - 1 \geq picMaxX) \mid \mid$
- [0260] $(yRefLX + (bH < \log_2 MVPREC) - 1 \geq picMaxY))$
- [0261] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.
- [0262] $picMaxX = (picW - 1) < \log_2 MVPREC$
- [0263] $picMaxY = (picH - 1) < \log_2 MVPREC$
- [0264] $isOoB[X] = (bW > BLKTHW3) \&\& (bW \leq BLKTHW4) \&\& (bH > BLKTHH3) \&\& (bH \leq BLKTHH4) \&\&$
- [0265] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0266] $(xRefLX + (bW < \log_2 MVPREC) - 1 > picMaxX) \mid \mid$
- [0267] $(yRefLX + (bH < \log_2 MVPREC) - 1 > picMaxY))$
- [0268] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.
- [0269] $xRefLX = xPb + (mvLX[0] > \log_2 MVPREC)$
- [0270] $yRefLX = yPb + (mvLX[1] > \log_2 MVPREC)$
- [0271] $isOoB[X] = (bW > BLKTHW3) \&\& (bW \leq BLKTHW4) \&\& (bH > BLKTHH3) \&\& (bH \leq BLKTHH4) \&\&$
- [0272] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0273] $(xRefLX + bW - 1 \geq picW - 1) \mid \mid$
- [0274] $(yRefLX + bH - 1 \geq picH - 1))$
- [0275] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.
- [0276] $isOoB[X] = (bW > BLKTHW3) \&\& (bW \leq BLKTHW4) \&\& (bH > BLKTHH3) \&\& (bH \leq BLKTHH4) \&$
- [0277] $((xRefLX \leq 0) \mid \mid (yRefLX \leq 0) \mid \mid$
- [0278] $(xRefLX + bW - 1 \geq picW - 1) \mid \mid$
- [0279] $(yRefLX + bH - 1 \geq picH - 1))$
- [0280] 예를 들어, $(BLKTHW3, BLKTHW4) = \{(4, 16), (8, 32), (16, 32), (32, 64)\}$, $(BLKTHH3, BLKTHH4) = \{(4, 16), (8, 32), (16, 32), (32, 64)\}$ 동일 수도 있다.
- [0281] 또는 면적의 대소에 의해 판정하여 OOB 처리가 적용되도록 할 수도 있다.
- [0282] $isOoB[X] = (bW * bH \geq BLKTHW5 * BLKTHH5) \&\&$
- [0283] $((xRefLX < picMinX) \mid \mid (yRefLX < picMinY) \mid \mid$
- [0284] $(xRefLX + bW * MVPREC - 1 \geq picMaxX) \mid \mid$
- [0285] $(yRefLX + bH * MVPREC - 1 \geq picMaxY))$
- [0286] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.

- [0287] $isOoB[X] = (bW * bH \geq BLKTHW5 * BLKTHH5) \&\&$
- [0288] $((xRefLX \leq picMinX) \mid \mid (yRefLX \leq picMinY) \mid \mid$
- [0289] $(xRefLX + (bW < \log 2 MVPREC) - 1 \geq picMaxX) \mid \mid$
- [0290] $(yRefLX + (bH < \log 2 MVPREC) - 1 \geq picMaxY))$
- [0291] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.
- [0292] $picMaxX = (picW - 1) < \log 2 MVPREC$
- [0293] $picMaxY = (picH - 1) < \log 2 MVPREC$
- [0294] $isOoB[X] = (bW * bH \geq BLKTHW5 * BLKTHH5) \&\&$
- [0295] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0296] $(xRefLX + (bW < \log 2 MVPREC) - 1 > picMaxX) \mid \mid$
- [0297] $(yRefLX + (bH < \log 2 MVPREC) - 1 > picMaxY))$
- [0298] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.
- [0299] $xRefLX = xPb + (mvLX[0] > \log 2 MVPREC)$
- [0300] $yRefLX = yPb + (mvLX[1] > \log 2 MVPREC)$
- [0301] $isOoB[X] = (bW * bH \geq BLKTHW5 * BLKTHH5) \&\&$
- [0302] $((xRefLX < 0) \mid \mid (yRefLX < 0) \mid \mid$
- [0303] $(xRefLX + bW - 1 \geq picW - 1) \mid \mid$
- [0304] $(yRefLX + bH - 1 \geq picH - 1))$
- [0305] 혹은 하기 식으로 $isOoB[X]$ 를 도출할 수도 있다.
- [0306] $isOoB[X] = (bW * bH \geq BLKTHW5 * BLKTHH5) \&\&$
- [0307] $((xRefLX \leq 0) \mid \mid (yRefLX \leq 0) \mid \mid$
- [0308] $(xRefLX + bW - 1 \geq picW - 1) \mid \mid$
- [0309] $(yRefLX + bH - 1 \geq picH - 1))$
- [0310] 예를 들어, $BLKTHW5=4, 8, 16, 32$, $BLKTHH5=4, 8, 16, 32$ 등일 수도 있다. 혹은 $BLKTHW5 * BLKTHH5$ 대신 $BLKTHS5$ 를 이용할 수도 있다. 예를 들어, $BLKTHS5=32, 64, 128$ 등일 수도 있다.
- [0311] 상기 예에서는 (DecideOoB1)의 OoB 판정식을 기초로 블록 사이즈를 제한하는 조건을 추가했지만, (DecideOoB2) 혹은 (DecideOoB3)의 OoB 판정식을 기초로 블록 사이즈를 제한하는 조건을 추가할 수도 있다.
- [0312] 상기 예에서는, $isOoB[X]$ 의 도출식의 조건의 일부로서 $(xRefLX < picMinX) \mid \mid (yRefLX < picMinY)$ 를 이용했지만, $(xRefLX \leq picMinX) \mid \mid (yRefLX \leq picMinY)$ 를 이용할 수도 있다.
- [0313] 이상과 같이, OoB 처리가 적용되는 블록 사이즈를 제한하여, 계산량을 저감할 수 있다.
- [0314] OoB 판정부(30921)는 $isOoB[X]$ 와 $isOoB[1-X]$ ($X=0, 1$)의 양쪽이 1이 되는 것을 피하는 구성일 수도 있다. 즉 참조 블록의 쌍방에서, 인터 예측 화상 생성부(309)가 OoB 마스크를 이용한 예측 화상 생성을 하는 것을 금지한다. 이로써, 참조 블록의 한쪽은 반드시 $isOoB[X]=false$ 가 즉 OoB 처리를 이용하지 않고 적용되어, 후단에서 OoB 마스크의 작성 및 마스크의 적용을 생략할 수 있다. 따라서, 계산량의 삭감이 가능해진다.
- [0315] <OoB 판정 생략 구성의 예 1>
- [0316] OoB 판정부(30921)는 L0 참조 영역과 L1 참조 영역의 유효 범위의 크기에 따른 판정에 의해 $isOoB[X]$ ($X=0, 1$)를 도출할 수도 있다. LX 참조 블록 중 픽처에 포함되는 유효 면적 InsideAreaX가 더 큰 쪽을 $X=K$ ($K=0, 1$)로 한다. 유효 면적이 큰 참조 픽처의 OoB 판정을 무효(OoB 판정 유무 플래그 $isOoB[K]=false$)로 하고, 유효 면적

이 작은 다른쪽(isOOB[!K] 혹은 isOOB[1-K])의 OOB 판정을 유효(isOOB[1-K]=true)로 한다. OOB 판정부(30921)는 예를 들어 다음 식에 의해 InsideAreaX를 구한다. 아울러, (false, true)의 값으로서 (0, 1)을 이용할 수도 있고, 별도의 값(1, 0)을 이용할 수도 있다. LX 참조 영역의 유효 범위는 LX 참조 블록과 픽처의 공통 부분에서 도출할 수도 있다.

[0317] $xRefLX = xPb * MVPREC + mvLX[0]$

[0318] $yRefLX = yPb * MVPREC + mvLX[1]$

[0319] $picMinX = 0 - MVPREC/2$

[0320] $picMinY = 0 - MVPREC/2$

[0321] $picMaxX = (picW-1)*MVPREC + MVPREC/2$

[0322] $picMaxY = (picH-1)*MVPREC + MVPREC/2$

[0323] $InsideWidthX = (\min(picMaxX, xRefLX + bW*MVPREC-1) - \max(picMinX, xRefLX))$

[0324] $InsideHeightX = (\min(picMaxY, yRefLX + bH*MVPREC-1) - \max(picMinY, yRefLX))$

[0325] $InsideAreaX = (InsideWidthX > 0 \ \&\& \ InsideHeightX > 0) ? InsideWidthX * InsideHeightX : 0$

[0326] InsideWidthX 또는 InsideHeightX의 값이 음이 되는 경우는, 참조 블록은 픽처에 공통 부분은 없기(참조 블록은 픽처에 포함되지 않기) 때문에 InsideAreaX=0으로 할 수도 있다.

[0327] 사이즈(picW, picH, bW, bH)와 정밀도 MVPREC의 곱셈은 이하와 같이, 사이즈를 MVPREC의 대수(對數)로 좌 비트 시프트 "<<" 처리할 수도 있다.

[0328] $xRefLX = (xPb << \log_2 MVPREC) + mvLX[0]$

[0329] $yRefLX = (yPb << \log_2 MVPREC) + mvLX[1]$

[0330] 혹은, picMaxX, picMaxY는 이하에서 도출할 수도 있다.

[0331] $picMaxX = (picW-1)*MVPREC$

[0332] $picMaxY = (picH-1)*MVPREC$

[0333] 혹은, xRefLX, yRefLX는 소수 단위의 경계 좌표가 아니라 정수 단위의 좌표로 도출할 수도 있다.

[0334] $xRefLX = xPb + mvLX[0]*MVPREC$

[0335] $yRefLX = yPb + mvLX[1]*MVPREC$

[0336] OOB 판정부(30921)는 L0 참조 블록과 L1 참조 블록에 대해 각각 InsideArea0과 InsideArea1을 도출하고, 하기 isOOB[X]를 도출한다.

[0337] $if (InsideArea0 \geq InsideArea1) \{$

[0338] $isOOB[0] = false, isOOB[1] = true$

[0339] $\} else \{$

[0340] $isOOB[0] = true, isOOB[1] = false$

[0341] $\}$

[0342] 또한 하기와 같이 한쪽의 isOOB[X]의 판정값을 먼저 도출하고, 다른쪽 isOOB[Y](Y=1-X)의 판정값을 isOOB[X]의 판정값으로부터 도출할 수도 있다.

[0343] $isOOB[0] = (InsideArea0 \geq InsideArea1) ? 0 : 1$

[0344] $isOOB[1] = 1 - isOOB[0]$

[0345] 상기 조건식은 "InsideArea0 > InsideArea1"일 수도 있다.

[0346] 상기 구성에 의하면, 유효 영역이 큰 쪽의 참조 픽처에서의 OOB 판정을 무효(플래그 isOOB[K]=false)로 하고,

OOB 판정을 생략(OOB 처리를 적용하지 않는다)할 수 있기 때문에, 예측 화상 생성에서의 계산량의 삭감이 가능해진다. 특히, 유효 영역이 큰 경우에는, OOB 판정을 무효로 해도 화질의 저하가 거의 없다.

[0347] <OOB 판정 생략 구성의 예 2>

[0348] OOB 판정부(30921)는 2개의 참조 화상에 관한 유효 영역 InsideArea의 사이즈가 동일한 경우에는 양쪽 모두 OOB 판정을 무효로 할 수도 있다. 구체적으로는, OOB 판정을 생략하는 것을 나타내는 값(false)이 되도록 isOOB[X]를 도출할 수도 있다.

[0349] if (InsideArea0 > InsideArea1) {

[0350] isOOB[0] = false, isOOB[1] = true

[0351] } else if (InsideArea0 < InsideArea1) {

[0352] isOOB[0] = true, isOOB[1] = false

[0353] } else {

[0354] isOOB[0] = false, isOOB[1] = false

[0355] }

[0356] 상기 구성에 의하면, 예측 화상 생성에서의 계산량의 삭감이 가능해진다.

[0357] <OOB 판정 생략 구성의 예 3>

[0358] OOB 판정부(30921)는 LX 참조 영역의 유효 범위(예를 들어, LX 참조 블록과 픽처의 공통 부분)의 종횡의 크기를 나타내는 InsideWidthX, InsideHeightX의 대소 관계를 이용하여, OOB 판정 유무 플래그 isOOB[X]를 도출할 수도 있다. 예를 들어, 특정 LX 참조 블록의 유효 영역의 폭과 높이가 둘다 다른쪽의 LY 참조 블록(Y=1-X)의 폭과 높이보다 큰 경우에, LX 참조 블록의 OOB 판정을 무효(isOOB=false)로 할 수도 있다.

[0359] if (InsideWidth0 > InsideWidth1 && InsideHeight0 > InsideHeight1) {

[0360] isOOB[0] = false, isOOB[1] = true

[0361] } else {

[0362] isOOB[0] = true, isOOB[1] = false

[0363] }

[0364] 또는 이하일 수도 있다.

[0365] if (InsideWidth0 >= InsideWidth1 && InsideHeight0 > InsideHeight1) {

[0366] isOOB[0] = false, isOOB[1] = true

[0367] } else {

[0368] isOOB[0] = true, isOOB[1] = false

[0369] }

[0370] 혹은 이하일 수도 있다.

[0371] if (InsideWidth0 >= InsideWidth1 && InsideHeight0 >= InsideHeight1) {

[0372] isOOB[0] = false, isOOB[1] = true

[0373] } else {

[0374] isOOB[0] = true, isOOB[1] = false

[0375] }

[0376] 또는, 특정 LX 참조 블록의 유효 영역의 폭과 높이가 둘다 다른쪽의 LY 참조 블록(Y=1-X)의 폭과 높이보다 큰 경우, LX 참조 블록의 OOB 판정을 무효(isOOB=false)로 하고, 2개의 유효 영역의 폭 혹은 높이가 동일한

경우에, OOB 판정을 양쪽 모두 무효로 할 수도 있다.

```
[0377] if (InsideWidth0 > InsideWidth1 && InsideHeight0 > InsideHeight1) {
[0378]     isOOB[0] = false, isOOB[1] = true
[0379] } else if (InsideWidth0 < InsideWidth1 && InsideHeight0 < InsideHeight1) {
[0380]     isOOB[0] = true, isOOB[1] = false
[0381] } else {
[0382]     isOOB[0] = false, isOOB[1] = false
[0383] }
```

또는, 상기 판정을 폭만 혹은 높이만을 이용하여 OOB 판정을 수행할 수도 있다.

```
[0385] if (InsideWidth0 > InsideWidth1) {
[0386]     isOOB[0] = false, isOOB[1] = true
[0387] } else {
[0388]     isOOB[0] = true, isOOB[1] = false
[0389] }
```

또는, 이하일 수도 있다

```
[0391] if (InsideHeight0 > InsideHeight1) {
[0392]     isOOB[0] = false, isOOB[1] = true
[0393] } else {
[0394]     isOOB[0] = true, isOOB[1] = false
[0395] }
```

상기 구성에 따르면, 유효 영역이 큰 참조 블록(혹은 양쪽의 참조 블록)의 OOB 판정을 생략할 수 있기 때문에, 예측 화상 생성에서의 계산량의 삭감이 가능해진다. 또한, 면적을 도출할 필요가 없어서 곱셈의 계산량을 삭감할 수 있다.

<정리>

이상 정리하면, L0 참조 영역의 유효 영역과 L1 참조 영역의 유효 영역을 비교하여, 참조 영역이 큰 쪽의 참조 픽처의 OOB 판정을 무효로 할 수도 있다. 또한, 참조 영역이 큰지 여부의 판정을 L0과 L1의 유효 영역의 면적의 비교에 의해 수행할 수도 있다. 또한, 참조 영역이 큰지 여부의 판정을 L0과 L1의 유효 영역의 폭 혹은 높이의 비교에 의해 수행할 수도 있다.

(OOB 마스크 도출 처리)

OOB 마스크 도출부(30922)는 참조 블록에 대해 OOB 마스크(OOBMask[X])를 도출한다. OOBMask[X]는 bWxbH 화소의 2차원 화상(마스크)이며, OOBMask[X][px][py](px=0..bW-1, py=0..bH-1)의 값은 0(false)이면 화소(px, py)가 픽처 내, 1(true)이면 픽처 외인 것을 나타낸다. 아울러, false와 true는 0, 1로 한정되지 않으며, 별도의 값, 예를 들어 2진수로 BitDepth 이상의 길이 0이나 1의 열로 이루어진 비트 마스크값 "0000000000b", "1111111111b" 등일 수도 있다. 또한 16진수의 "000", "3FF"나, "000", "FFF", "0000", "FFFF"를 이용할 수도 있다. OOB 마스크 도출부(30922)는 OOB 판정부(30921)가 OOB 처리 대상이 아니라고(isOOB[X]==false) 판정한 블록에 대해서는, OOBMask[X]의 모든 값을 false로 한다. 또한, OOB 판정부(30921)가 OOB 처리 대상이라고(isOOB[X]==true) 판정한 블록에 대해서는, 다음 식을 이용하여 마스크를 도출한다. 아울러, 부등호(">=" 또는 "<=")는 등호를 포함하지 않아도(">" 또는 "<") 상관없다.

```
[0401] xRefLX = (xPb<<log2MVPREC) + mvLX[0]
```

```
[0402] yRefLX = (yPb<<log2MVPREC) + mvLX[1]
```

- [0403] $\text{OOBMask}[X][px][py] = ((x\text{RefLX} + px < \text{picMinX}) \mid \mid (y\text{RefLX} + py < \text{picMinY}) \mid \mid$
- [0404] $(x\text{RefLX} + px \geq \text{picMaxX}) \mid \mid (y\text{RefLX} + py \geq \text{picMaxY})) ? 1 : 0$
- [0405] 단 $px=0..bW-1$, $py=0..bH-1$ 이다. 상기는 (Decide00B1)에서의 $\text{isOOB}[X]$ 의 판정식을 기초로 OOBMask를 도출했다. 이외에도, (Decide00B2), (Decide00B3)에서의 $\text{isOOB}[X]$ 의 판정식을 기초로 OOBMask를 도출할 수도 있다. 또한, (Decide00B1), (Decide00B2), (Decide00B3)에 유효 범위의 제약을 가한 $\text{isOOB}[X]$ 의 판정식을 기초로 OOBMask를 도출할 수도 있다.
- [0406] 혹은, OOBMask의 도출식은 하기 식일 수도 있다.
- [0407] $\text{OOBMask}[X][px][py] = ((x\text{RefLX} + px < \text{picMinX}) \mid \mid (y\text{RefLX} + py < \text{picMinY}) \mid \mid$
- [0408] $(x\text{RefLX} + px + (bW \ll \log 2\text{MVPREC}) - 1 \geq \text{picMaxX}) \mid \mid$
- [0409] $(y\text{RefLX} + py + (bH \ll \log 2\text{MVPREC}) - 1 \geq \text{picMaxY})) ? 1 : 0$
- [0410] 혹은, OOBMask의 도출식은 하기 식일 수도 있다.
- [0411] $\text{OOBMask}[X][px][py] = ((x\text{RefLX} + px \leq \text{picMinX}) \mid \mid (y\text{RefLX} + py \leq \text{picMinY}) \mid \mid$
- [0412] $(x\text{RefLX} + px + (bW \ll \log 2\text{MVPREC}) - 1 \geq \text{picMaxX}) \mid \mid$
- [0413] $(y\text{RefLX} + py + (bH \ll \log 2\text{MVPREC}) - 1 \geq \text{picMaxY})) ? 1 : 0$
- [0414] 혹은, 반올림을 위한 항을 생략하고, 다음과 같이 OOBMask를 도출할 수도 있다.
- [0415] $\text{picMaxX} = (\text{picW}-1) \ll \log 2\text{MVPREC}$
- [0416] $\text{picMaxY} = (\text{picH}-1) \ll \log 2\text{MVPREC}$
- [0417] $\text{OOBMask}[X][px][py] = ((x\text{RefLX} + px < 0) \mid \mid (y\text{RefLX} + py < 0) \mid \mid$
- [0418] $(x\text{RefLX} + (bW \ll \log 2\text{MVPREC}) - 1 > \text{picMaxX}) \mid \mid$
- [0419] $(y\text{RefLX} + (bH \ll \log 2\text{MVPREC}) - 1 > \text{picMaxY})) ? 1 : 0$
- [0420] 혹은, 정수 단위의 움직임 벡터를 이용하여 OOBMask를 도출할 수도 있다.
- [0421] $x\text{RefLX} = x\text{Pb} + (\text{mvLX}[0] \gg \log 2\text{MVPREC})$
- [0422] $y\text{RefLX} = y\text{Pb} + (\text{mvLX}[1] \gg \log 2\text{MVPREC})$
- [0423] $\text{OOBMask}[X][px][py] = ((x\text{RefLX} + px < 0) \mid \mid (y\text{RefLX} + py < 0) \mid \mid$
- [0424] $(x\text{RefLX} + px + bW - 1 \geq \text{picW}-1) \mid \mid$
- [0425] $(y\text{RefLX} + py + bH - 1 \geq \text{picH}-1)) ? 1 : 0$
- [0426] 혹은, OOBMask의 도출식은 하기 식일 수도 있다.
- [0427] $\text{OOBMask}[X] = ((x\text{RefLX} + px \leq 0) \mid \mid (y\text{RefLX} + py \leq 0) \mid \mid$
- [0428] $(x\text{RefLX} + px + bW - 1 \geq \text{picW}-1) \mid \mid$
- [0429] $(y\text{RefLX} + py + bH - 1 \geq \text{picH}-1)) ? 1 : 0$
- [0430] OOB 마스크 도출부(30922)는 OOBMask[X]를 NxM 화소 단위의 마스크로서 도출할 수도 있다. 즉, 하나의 마스크 값을 NxM 화소의 OOB 판정에 대응시킨다. 이때 OOBMask[X]의 크기는 $(bW/N) * (bH/M)$ 이다. 각 마스크값 $\text{OOBMask}[X][x][y](x=0..bW/N-1, y=0..bH/M-1)$ 은 예를 들어 이하와 같이 구해진다.
- [0431] $x\text{RefLX} = (x\text{Pb} \ll \log 2\text{MVPREC}) + \text{mvLX}[0]$
- [0432] $y\text{RefLX} = (y\text{Pb} \ll \log 2\text{MVPREC}) + \text{mvLX}[1]$
- [0433] $\text{OOBMask}[X][px][py] = ((x\text{RefLX} + px * N < \text{picMinX}) \mid \mid$
- [0434] $(y\text{RefLX} + py * M < \text{picMinY}) \mid \mid$

[0435] $(xRefLX + px*N + N \geq picMaxX) \mid \mid$

[0436] $(yRefLX + py*M + M \geq picMaxY)) ? 1 : 0$

[0437] 여기서 $px=0..bW/N-1$, $py=0..bH/M-1$ 이다. N 및 M 의 항은 생략할 수도 있다. 상기는 (DecideOOB1)에서의 $isOOB[X]$ 의 판정식을 기초로 OOBMask를 도출했다. 이외에도, (DecideOOB2), (DecideOOB3)에서의 $isOOB[X]$ 의 판정식을 기초로 OOBMask를 도출할 수도 있다. 또한, (DecideOOB1), (DecideOOB2), (DecideOOB3)에 유효 범위의 제약을 가한 $isOOB[X]$ 의 판정식을 기초로 OOBMask를 도출할 수도 있다.

[0438] 혹은, OOBMask의 도출식은 하기 식일 수도 있다.

[0439] $OOBMask[X][px][py] = ((xRefLX + px*N < picMinX) \mid \mid (yRefLX + py*N < picMinY) \mid \mid$

[0440] $(xRefLX + px*N + N + bW*MVPREC - 1 \geq picMaxX) \mid \mid$

[0441] $(yRefLX + py*N + M + bH*MVPREC - 1 \geq picMaxY)) ? 1 : 0$

[0442] 혹은, 반올림을 위한 항을 생략하고, 다음과 같이 OOBMask를 도출할 수도 있다.

[0443] $picMaxX = (picW-1) \ll \log 2MVPREC$

[0444] $picMaxY = (picH-1) \ll \log 2MVPREC$

[0445] $OOBMask[X][px][py] = ((xRefLX + px*N < 0) \mid \mid (yRefLX + py*N < 0) \mid \mid$

[0446] $(xRefLX + px*N + N + bW*MVPREC - 1 > picMaxX) \mid \mid$

[0447] $(yRefLX + py*N + M + bH*MVPREC - 1 > picMaxY)) ? 1 : 0$

[0448] 혹은, 정수 단위의 움직임 벡터를 이용하여 OOBMask를 도출할 수도 있다.

[0449] $xRefLX = xPb + (mvLX[0] \gg \log 2MVPREC)$

[0450] $yRefLX = yPb + (mvLX[1] \gg \log 2MVPREC)$

[0451] $OOBMask[X][px][py] = ((xRefLX + px*N < 0) \mid \mid (yRefLX + py*N < 0) \mid \mid$

[0452] $(xRefLX + px*N + N + bW - 1 \geq picW-1) \mid \mid$

[0453] $(yRefLX + py*N + M + bH - 1 \geq picH-1)) ? 1 : 0$

[0454] 이로써 NxM 화소 단위로 마스크가 동일한 값이 되기 때문에, 마스크에 필요한 메모리를 삭감하는 동시에 계산량을 삭감할 수 있다.

[0455] (OOB 예측 화상 생성)

[0456] 움직임 보상부(3091)(보간 화상 생성부(3091))는 양예측의 경우에는, 상기 (식 MC-1)로 도출한 $Pred[][]$ 를, L0 리스트, L1 리스트별로 보간 화상 $PredL0[][]$ 과 $PredL1[][]$ 로써 설정한다. 그리고, 무게 예측부(3094)는 $PredL0[][]$ 과 $PredL1[][]$ 로부터 보간 화상 $Pred[][]$ 를 생성한다. 양예측의 참조 블록이 OOB 처리의 대상인 경우, 예측 화상의 생성에 당해 참조 블록에 대응하는 OOBMask[X]가 이용된다. 이때 $Pred[][]$ 는 다음과 같이 도출된다.

[0457] $if (OOBMask[0][px][py] \&\& !OOBMask[1][px][py]) \{ (식 WP-1)$

[0458] $Pred[px][py] = (PredL1[px][py] + offset1) \gg shift1$

[0459] $\} else if (!OOBMask[0][px][py] \&\& OOBMask[1][px][py]) \{$

[0460] $Pred[px][py] = (PredL0[px][py] + offset1) \gg shift1$

[0461] $\} else \{$

[0462] $Pred[px][py] = (PredL0[px][py] + PredL1[px][py] + offset2) \gg shift2$

[0463] $\}$

[0464] 여기서 $px=0..bW-1$, $py=0..bH-1$ 이다. 또한, $shift1=Max(2, 14-bitDepth)$, $shift2=Max(3, 15-bitDepth)$,

$offset1=1<<(shift1-1)$, $offset2=1<<(shift2-1)$ 이다. 아울러, $shift1$ 은 보간 화상 비트 심도(예를 들어 14 bit)를 원래 비트 심도로 되돌리기 위한 시프트(예를 들어 14-bitDepth)이다. $shift2$ 는 원래 비트로 되돌리는 동시에 평균을 곱한 우 시프트양, $offset1$, $offset2$ 는 $shift1$, $shift2$ 만큼 우 시프트할 때의 반올림 값이다.

[0465] 아울러, CU 단위의 명시적 가중 양예측(BCW) 혹은 인트라 인터 결합 예측 CIIP(combined inter-picture merge and intra-picture prediction)를 이용하는지 여부에 따라 상기 처리를 전환할 수도 있다. $bcwIdx$ 는 BCW의 무게를 나타내는 인덱스이다. $ciip_flag$ 는 CIIP의 온 오프를 나타내는 플래그이다. $bcwIdx==0$ 혹은 CIIP 플래그($ciip_flag$)가 1인 경우에는 상기 (식 WP-1)의 처리를 수행한다. 그 이외의 경우에는, 이하의 처리를 수행한다.

[0466] $w1 = bcwWLut[bcwIdx]$

[0467] $w0 = (8 - w1)$

[0468] 여기서 $bcwWLut[k] = \{4, 5, 3, 10, -2\}$

[0469] if ($OOBMask[0][px][py] \&\& !OOBMask[1][px][py]$) {

[0470] $Pred[px][py] = (PredL1[px][py] + offset1) \gg shift1$

[0471] } else if ($!OOBMask[0][px][py] \&\& OOBMask[1][px][py]$) {

[0472] $Pred[px][py] = (PredL0[px][py] + offset1) \gg shift1$

[0473] } else {

[0474] $Pred[px][py] = (w0 * PredL0[px][py] + w1 * PredL1[px][py] + offset3) \gg (shift1+3)$

[0475] }

[0476] 여기서 $offset3=1<<(shift1+2)$ 이다.

[0477] 블록 단위로 LX 참조 픽처를 사용하는지 여부를 나타내는 플래그 $predFlagLX$ 와, $OOBMask$ 에 의한 처리를 하나의 분기로 정리하여 이하와 같이 처리하면 처리량을 경감할 수 있는 효과를 나타낸다.

[0478] if ($((predFlagL0 == 1 \&\& predFlagL1 == 0) \mid \mid (!OOBMask[0][px][py] \&\& OOBMask[1][px][py]))$) { (식 WP-2)

[0479] $Pred[px][py] = (PredL0[px][py] + offset1) \gg shift1$

[0480] } else if ($((predFlagL0 == 0 \&\& predFlagL1 == 1) \mid \mid (OOBMask[0][px][py] \&\& !OOBMask[1][px][py]))$) {

[0481] $Pred[px][py] = (PredL1[px][py] + offset1) \gg shift1$

[0482] } else if ($bcwIdx == 0 \mid \mid ciip_flag == 1$) {

[0483] $Pred[px][py] = (PredL0[px][py] + PredL1[px][py] + offset2) \gg shift2$

[0484] } else { // $bcwIdx \neq 0$ and $ciip_flag == 0$

[0485] $Pred[px][py] = (w0 * PredL0[px][py] + w1 * PredL1[px][py] + offset3) \gg (shift1+3)$

[0486] }

[0487] <생략 구성>

[0488] OOB 판정부(30921)는 2매의 참조 픽처상의 참조 블록의 가중합에 의해 예측 화상을 생성하는 가중 예측에서는, 블록에 OOB 처리를 적용하지 않을 수도 있다. 즉, 가중 예측이 적용되어 있는 블록은 OOB 판정 유무 플래그 $isOOB[X]$ 를 false로 한다. 이로써, 양방향의 가중 예측이 적용되는 참조 블록의 한쪽이 이용되지 않는 것에 의한 왜곡의 발생 혹은 부호화 효율의 저하를 방지할 수 있다. 이러한 가중 예측에는, BCW(Bi-prediction with CU-level Weight) 예측과 WP(Weighted prediction) 예측이 있다.

[0489] BCW 예측은 블록 단위로 무게 계수($w0$, $w1$)를 선택하고, 양예측에서의 L0 보간 화상 및 L1 보간 화상에 무게를 곱셈하여 예측 화상을 생성한다. 파라미터 복호부(302)는 부호화 유닛 CU에서 무게 계수를 나타내는 파라미터 bcw_idx 를 부호화 데이터로부터 복호한다. 파라미터 부호화부(111)는 bcw_idx 를 부호화한다. 머지 예측의 경우에는, 참조 블록(인접 블록)의 $bcwIdx$ 를 대상 블록의 $bcwIdx$ 로 설정한다. 그 이외의 경우에는, 대상 블록의

bcwIdx에 bcw_idx를 설정한다. bcwIdx!=0이면, 당해 CU에 BCW 예측을 적용한다. OOB 관정부(30921)는 BCW 예측을 적용하는 경우(bcwIdx!=0), isOOB[X]를 false로 한다.

[0490] WP 예측은 슬라이스 단위로 참조 픽처별 무게 계수(w_{e0} , w_{e1})와 오프셋($offset_0$, $offset_1$)을 전환하며, CU 단위로 참조 픽처 인덱스를 기초로 무게 계수와 오프셋을 선택하고, L0 보간 화상 및 L1 보간 화상에 무게를 곱셈하여 예측 화상을 생성한다. 파라미터 복호부(302)는 PPS의 부호화 데이터로부터, B 픽처에 WP 예측을 적용하는지 여부를 나타내는 pps_weighted_bipred_flag를 복호한다. pps_weighted_bipred_flag==1의 경우, 당해 PPS의 B 픽처에 WP 예측을 적용한다.

[0491] (w_{e0} , w_{e1}), ($offset_0$, $offset_1$)은 슬라이스 단위로 부호화 데이터로부터 복호되는 무게 계수, 오프셋이며, 참조 픽처 인덱스에 따라 도출된다.

[0492] OOB 관정부(30921)는 명시적인 WP 예측을 적용하는 경우(pps_weighted_bipred_flag==1), isOOB[X]를 false로 한다. 아울러, 참조 픽처를 2매 사용하지 않는 경우(predFlagLX(X=0, 1) 중 어느 것인가가 0)는, OOB 처리를 적용할 수도 있다.

[0493] 상술한 생략 구성을 이용하는 경우에는, OOBMask[0], OOBMask[1] 모두 1이 되지 않는기 때문에, 무게 예측부(3094)는 이하의 분기 처리에 의해 보간 화상을 도출할 수 있다.

```
[0494] if (OOBMask[0][px][py]) {
[0495]     Pred[px][py] = (we1 * PredL1[px][py] + offset1) >> shift1
[0496] } else if (OOBMask[1][px][py]) {
[0497]     Pred[px][py] = (we0 * PredL0[px][py] + offset0) >> shift1
[0498] } else {
[0499]     if (bcwIdx == 0 && pps_weighted_bipred_flag == 1) {
[0500]         Pred[px][py] = (we0 * PredL0[px][py] + we1 * PredL1[px][py] + offset2) >> shift2
[0501]     } else { // bcwIdx !=0 and pps_weighted_bipred_flag == 0)
[0502]         Pred[px][py] = (w0 * PredL0[px][py] + w1 * PredL1[px][py] + offset3)>>(shift1+3))
[0503]     }
[0504] }
```

[0505] 또한, 미리 PredL0, PredL1, PredBI를 산출해 두고, 이하의 마스크 처리로 최종적인 Pred를 도출하는 것이 적당하다.

[0506] 우선, PredL0, PredL1, PredBI를 산출한다.

[0507] $PredL0[px][py] = (we0 * PredL0[px][py] + offset0) >> shift1$

[0508] $PredL1[px][py] = (we1 * PredL1[px][py] + offset1) >> shift1$

[0509] $PredBI[px][py] = (we0 * PredL0[px][py] + we1 * PredL1[px][py] + offset2) >> shift2$ (WP 예측의 경우)

[0510] $PredBI[px][py] = (w0 * PredL0[px][py] + w1 * PredL1[px][py] + offset3) >> (shift1+3)$ (BCW 예측의 경우)

[0511] 다음으로, OOBMask[0][px][py], OOBMask[1][px][py]의 값으로서, 2진수로 BitDepth 길이의 "111111111b", "000000000b"의 비트 마스크를 이용하는 경우, 이하의 처리로 최종적인 Pred를 도출한다.

[0512] $Pred[px][py] =$

[0513] $(PredBI[px][py] \& (!OOBMask[0][px][py] \& !OOBMask[1][px][py])) +$

[0514] $(PredL0[px][py] \& OOBMask[0][px][py]) +$

[0515] $(PredL1[px][py] \& OOBMask[1][px][py])$

[0516] 아울러, true=1, false=0의 값인 경우에는, 0, 1으로부터 비트 마스크값 "000000000b", "111111111b"를 생성

하며 이하의 처리일 수도 있다. 여기서 "+"는 "|"일 수도 있다(이하의 마스크 연산 처리에서도 동일).

```
[0517] Pred[px][py]= (식 excl-1)
[0518] (PredBI[px][py] & (0-((!OOBMask[0][px][py] & !OOBMask[1][px][py]) ? 1 : 0)))+
[0519] (PredL0[px][py] & (0-(OOBMask[0][px][py] ? 1 : 0))) +
[0520] (PredL1[px][py] & (0-(OOBMask[1][px][py] ? 1 : 0)))
[0521] 상기 식은 화소값의 비트 심도 이상의 비트폭을 갖는 마스크를 작성하기 위해, 2차(0 또는 1)의 마스크를 0(모든 비트가 0, 화소값이 무효) 또는 -1(모든 비트가 1, 화소값이 유효)로 변환하는 계산을 포함하고 있다. 예측 화상의 도출은 이로 한정되지 않으며, PredBI, PredL0, PredL1에 대응하는 3항 중 OOB 마스크에 의해 무효가 되는 항이 0이 되는 계산을 이용하면 무방하다. 예를 들어 화소값의 비트 심도를 최대 16 비트로 할 때, 다음 식을 이용할 수도 있다.
```

```
[0522] Pred[px][py]=
[0523] (PredBI[px][py] & ((!OOBMask[0][px][py] & !OOBMask[1][px][py])? 0xFFFF : 0))+
[0524] (PredL0[px][py] & (OOBMask[0][px][py] ? 0xFFFF : 0)) +
[0525] (PredL1[px][py] & (OOBMask[1][px][py] ? 0xFFFF : 0))
```

3항의 결합에 가산을 이용하지 않고, 하기와 같이 비트 OR을 이용할 수도 있다.

```
[0527] Pred[px][py]=
[0528] (PredBI[px][py] & ((!OOBMask[0][px][py] & !OOBMask[1][px][py])? 0xFFFF : 0)) |
[0529] (PredL0[px][py] & (OOBMask[0][px][py] ? 0xFFFF : 0)) |
[0530] (PredL1[px][py] & (OOBMask[1][px][py] ? 0xFFFF : 0))
```

혹은 곱셈을 이용할 수도 있다.

```
[0532] Pred[px][py]=
[0533] (PredBI[px][py] * ((!OOBMask[0][px][py] & !OOBMask[1][px][py])? 1 : 0)) +
[0534] (PredL0[px][py] * (OOBMask[0][px][py] ? 1 : 0)) +
[0535] (PredL1[px][py] * (OOBMask[1][px][py] ? 1 : 0))
```

상기에 의하면 배타 처리를 간략화할 수 있다. 배타 처리에 의해 (식 excl-1)의 연산보다 처리를 간략화할 수 있다.

[0537] 무게 예측부(3094)는 isOOB의 값을 참조하여 미리 처리를 나눌 수도 있다. 예를 들어 이하와 같이 할 수도 있다. 이와 같이 하면, isOOB[X]가 false인 경우에 화소별로 OOBMask[X][px][py]를 참조할 필요가 없어, 계산량을 삭감할 수 있다는 효과를 나타낸다. 예를 들어 무게를 사용하지 않는 경우, 이하와 같이 예측 화상을 생성할 수도 있다. if (isOOB[0] && !isOOB[1]) {

```
[0538] if (OOBMask[0][px][py]) {
[0539] Pred[px][py] = (PredL1[px][py] + offset1) >> shift1
[0540] } else {
[0541] Pred[px][py] = (PredL0[px][py] + PredL1[px][py] + offset2) >> shift2
[0542] }
[0543] } else if (!isOOB[0] && isOOB[1]) {
[0544] if (OOBMask[1][px][py]) {
[0545] Pred[px][py] = (PredL0[px][py] + offset1) >> shift1
```

```

[0546]     } else {
[0547]     Pred[px][py] = (PredL0[px][py] + PredL1[px][py] + offset2) >> shift2
[0548]     }
[0549]     } else if (isOOB[0] && isOOB[1]) {
[0550]     if (OOBMask[0][px][py] && !OOBMask[1][px][py]) {
[0551]     Pred[px][py] = (PredL1[px][py] + offset1) >> shift1
[0552]     } else if (!OOBMask[0][px][py] && OOBMask[1][px][py]) {
[0553]     Pred[px][py] = (PredL0[px][py] + offset1) >> shift1 } else {
[0554]     Pred[px][py] = (PredL0[px][py] + PredL1[px][py] + offset2) >> shift2
[0555]     }
[0556]     } else {
[0557]     Pred[px][py] = (PredL0[px][py] + PredL1[px][py] + offset2) >> shift2
[0558]     }
[0559]     예를 들어 CU 단위의 명시적 가중 양예측(BCW)의 경우(bcwIdx!=0인 동시에, pps_weighted_bipred_flag==0), 이
하와 같이 예측 화상을 생성할 수도 있다.
[0560]     if (isOOB[0] && !isOOB[1]) {
[0561]     if (OOBMask[0][px][py]) {
[0562]     Pred[px][py] = (PredL1[px][py] + offset1) >> shift1
[0563]     } else {
[0564]     Pred[px][py] = (w0 * PredL0[px][py] + w1 * PredL1[px][py] + offset3) >> (shift1+3)
[0565]     }
[0566]     } else if (!isOOB[0] && isOOB[1]) {
[0567]     if (OOBMask[1][px][py]) {
[0568]     Pred[px][py] = (PredL0[px][py] + offset1) >> shift1
[0569]     } else {
[0570]     Pred[px][py] = (w0 * PredL0[px][py] + w1 * PredL1[px][py] + offset3) >> (shift1+3)
[0571]     }
[0572]     } else if (isOOB[0] && isOOB[1]) {
[0573]     if (OOBMask[0][px][py] && !OOBMask[1][px][py]) {
[0574]     Pred[px][py] = (PredL1[px][py] + offset1) >> shift1
[0575]     } else if (!OOBMask[0][px][py] && OOBMask[1][px][py]) {
[0576]     Pred[px][py] = (PredL0[px][py] + offset1) >> shift1
[0577]     } else {
[0578]     Pred[px][py] = (w0 * PredL0[px][py] + w1 * PredL1[px][py] + offset3) >> (shift1+3)
[0579]     }
[0580]     } else {

```

[0581] $\text{Pred}[px][py] = (w0 * \text{PredL0}[px][py] + w1 * \text{PredL1}[px][py] + \text{offset3}) \gg (\text{shift1}+3)$

[0582] }

[0583] 예를 들어 WP 예측의 경우($\text{bcwIdx}=0$ 인 동시에, $\text{pps_weighted_bipred_flag}=1$), 이하와 같이 예측 화상을 생성할 수도 있다.

[0584] if ($\text{isOOB}[0] \ \&\& \ !\text{isOOB}[1]$) {

[0585] if ($\text{OOBMask}[0][px][py]$) {

[0586] $\text{Pred}[px][py] = (w1 * \text{PredL1}[px][py] + \text{offset1}) \gg \text{shift1}$

[0587] } else {

[0588] $\text{Pred}[px][py] = (w0 * \text{PredL0}[px][py] + w1 * \text{PredL1}[px][py] + \text{offset2}) \gg \text{shift2}$

[0589] }

[0590] } else if ($\!isOOB[0] \ \&\& \ \text{isOOB}[1]$) {

[0591] if ($\text{OOBMask}[1][px][py]$) {

[0592] $\text{Pred}[px][py] = (w0 * \text{PredL0}[px][py] + \text{offset1}) \gg \text{shift1}$

[0593] } else {

[0594] $\text{Pred}[px][py] = (w0 * \text{PredL0}[px][py] + w1 * \text{PredL1}[px][py] + \text{offset2}) \gg \text{shift2}$

[0595] }

[0596] } else if ($\text{isOOB}[0] \ \&\& \ \text{isOOB}[1]$) {

[0597] if ($\text{OOBMask}[0][px][py] \ \&\& \ !\text{OOBMask}[1][px][py]$) {

[0598] $\text{Pred}[px][py] = (w1 * \text{PredL1}[px][py] + \text{offset1}) \gg \text{shift1}$

[0599] } else if ($\!OOBMask[0][px][py] \ \&\& \ \text{OOBMask}[1][px][py]$) {

[0600] $\text{Pred}[px][py] = (w0 * \text{PredL0}[px][py] + \text{offset1}) \gg \text{shift1}$

[0601] } else {

[0602] $\text{Pred}[px][py] = (w0 * \text{PredL0}[px][py] + w1 * \text{PredL1}[px][py] + \text{offset2}) \gg \text{shift2}$

[0603] }

[0604] } else {

[0605] $\text{Pred}[px][py] = (w0 * \text{PredL0}[px][py] + w1 * \text{PredL1}[px][py] + \text{offset2}) \gg \text{shift2}$

[0606] }

[0607] 인터 예측 화상 생성부(309)는 생성한 블록의 예측 화상을 가산부(312)에 출력한다.

[0608] (동화상 부호화 장치의 구성)

[0609] 다음으로, 본 실시형태에 관한 동화상 부호화 장치(11)의 구성에 대하여 설명한다. 도 7은 본 실시형태에 관한 동화상 부호화 장치(11)의 구성을 나타내는 블록도이다. 동화상 부호화 장치(11)는 예측 화상 생성부(101), 감산부(102), 변환·양자화부(103), 역양자화·역변환부(105), 가산부(106), 루프 필터(107), 예측 파라미터 메모리(예측 파라미터 저장부, 프레임 메모리)(108), 참조 픽처 메모리(참조 화상 저장부, 프레임 메모리)(109), 부호화 파라미터 결정부(110), 파라미터 부호화부(111), 예측 파라미터 도출부(120), 엔트로피 부호화부(104)를 포함하여 구성된다.

[0610] 예측 화상 생성부(101)는 CU별로 예측 화상을 생성한다. 예측 화상 생성부(101)는 이미 설명한 인터 예측 화상 생성부(309)와 인트라 예측 화상 생성부를 포함하고 있으며, 설명을 생략한다.

[0611] 감산부(102)는 예측 화상 생성부(101)로부터 입력된 블록의 예측 화상의 화소값을 화상(T)의 화소값으로부터 감

산하여 예측 오차를 생성한다. 감산부(102)는 예측 오차를 변환·양자화부(103)에 출력한다.

- [0612] 변환·양자화부(103)는 감산부(102)로부터 입력된 예측 오차에 대해, 주파수 변환에 의해 변환 계수를 산출하고, 양자화에 의해 양자화 변환 계수를 도출한다. 변환·양자화부(103)는 양자화 변환 계수를 파라미터 부호화부(111) 및 역양자화·역변환부(105)에 출력한다.
- [0613] 변환·양자화부(103)는 분리 변환부(제1 변환부), 비분리 변환부(제2 변환부), 스케일링부를 구비하고 있다.
- [0614] 분리 변환부는 예측 오차에 분리 변환을 적용한다. 스케일링부는 변환 계수에 대해, 양자화 매트릭스에 의한 스케일링을 수행한다.
- [0615] 역양자화·역변환부(105)는 동화상 복호 장치(31)에서의 역양자화·역변환부(311)와 동일하며, 설명을 생략한다. 산출한 예측 오차는 가산부(106)에 출력된다.
- [0616] 파라미터 부호화부(111)는 헤더 부호화부(1110), CT 정보 부호화부(1111), CU 부호화부(1112)(예측 모드 부호화부)를 구비하고 있다. CU 부호화부(1112)는 또한 TU 부호화부(1114)를 구비하고 있다. 이하, 각 모듈의 개략 동작을 설명한다.
- [0617] 헤더 부호화부(1110)는 헤더 정보, 분할 정보, 예측 정보, 양자화 변환 계수 등의 파라미터의 부호화 처리를 수행한다.
- [0618] CT 정보 부호화부(1111)는 QT, MT(BT, TT) 분할 정보 등을 부호화한다.
- [0619] CU 부호화부(1112)는 CU 정보, 예측 정보, 분할 정보 등을 부호화한다.
- [0620] TU 부호화부(1114)는 TU에 예측 오차가 포함되어 있는 경우에, QP 갱신 정보와 양자화 예측 오차를 부호화한다.
- [0621] CT 정보 부호화부(1111), CU 부호화부(1112)는 인터 예측 파라미터(predMode, general_merge_flag, merge_idx, inter_pred_idc, refIdxLX, mvp_LX_idx, mvdLX), 인트라 예측 파라미터, 양자화 변환 계수 등의 선택스 요소를 파라미터 부호화부(111)에 공급한다.
- [0622] 엔트로피 부호화부(104)에는, 파라미터 부호화부(111)로부터 양자화 변환 계수와 부호화 파라미터(분할 정보, 예측 파라미터)가 입력된다. 엔트로피 부호화부(104)는 이들을 엔트로피 부호화하여 부호화 스트림(Te)을 생성하고, 출력한다.
- [0623] 예측 파라미터 도출부(120)는 인터 예측 파라미터 부호화부(112), 인트라 예측 파라미터 부호화부를 포함하는 수단이며, 부호화 파라미터 결정부(110)로부터 입력된 파라미터로부터 인트라 예측 파라미터 및 인터 예측 파라미터를 도출한다. 도출된 인트라 예측 파라미터 및 인터 예측 파라미터는 파라미터 부호화부(111)에 출력된다.
- [0624] (인트라 예측 파라미터 부호화부의 구성)
- [0625] 인터 예측 파라미터 부호화부(112)는 파라미터 부호화 제어부(1121), 인터 예측 파라미터 도출부(303)를 포함하여 구성된다. 인터 예측 파라미터 도출부(303)는 동화상 복호 장치와 공통 구성이다. 파라미터 부호화 제어부(1121)는 머지 인덱스 도출부(11211)와 벡터 후보 인덱스 도출부(11212)를 포함한다.
- [0626] 머지 인덱스 도출부(11211)는 머지 후보 등을 도출하여, 인터 예측 파라미터 도출부(303)에 출력한다. 벡터 후보 인덱스 도출부(11212)는 예측 벡터 후보 등을 도출하여, 인터 예측 파라미터 도출부(303)와 파라미터 부호화부(111)에 출력한다.
- [0627] (인트라 예측 파라미터 부호화부의 구성)
- [0628] 인트라 예측 파라미터 부호화부는 파라미터 부호화 제어부와 인트라 예측 파라미터 도출부를 구비한다. 인트라 예측 파라미터 도출부는 동화상 복호 장치와 공통 구성이다.
- [0629] 다만, 동화상 복호 장치와 달리, 인터 예측 파라미터 도출부(303), 인트라 예측 파라미터 도출부의 입력은 부호화 파라미터 결정부(110), 예측 파라미터 메모리(108)이며, 파라미터 부호화부(111)에 출력한다.
- [0630] 가산부(106)는 예측 화상 생성부(101)로부터 입력된 예측 블록의 화소값과 역양자화·역변환부(105)로부터 입력된 예측 오차를 화소별로 가산하여 복호 화상을 생성한다. 가산부(106)는 생성한 복호 화상을 참조 픽처 메모리(109)에 저장한다.
- [0631] 루프 필터(107)는 가산부(106)가 생성한 복호 화상에 대해, 디블록킹 필터, SAO, ALF를 가한다. 아울러, 루프

필터(107)는 반드시 상기 3종류의 필터를 포함하지 않을 수도 있으며, 예를 들어 더블록킹 필터만의 구성일 수도 있다.

[0632] 예측 파라미터 메모리(108)는 부호화 파라미터 결정부(110)가 생성한 예측 파라미터를, 대상 픽처 및 CU별로 미리 정한 위치에 저장한다.

[0633] 참조 픽처 메모리(109)는 루프 필터(107)가 생성한 복호 화상을 대상 픽처 및 CU별로 미리 정한 위치에 저장한다.

[0634] 부호화 파라미터 결정부(110)는 부호화 파라미터의 복수의 세트 중, 하나의 세트를 선택한다. 부호화 파라미터란, 상술한 QT, BT 혹은 TT 분할 정보, 예측 파라미터, 혹은 이들과 관련하여 생성되는 부호화의 대상이 되는 파라미터이다. 예측 화상 생성부(101)는 이들 부호화 파라미터를 이용하여 예측 화상을 생성한다.

[0635] 부호화 파라미터 결정부(110)는 복수의 세트의 각각에 대해 정보량의 크기와 부호화 오차를 나타내는 RD 코스트값을 산출한다. RD 코스트값은 예를 들어 부호량과 제공 오차에 계수 λ 를 곱한 값과의 합이다. 부호량은 양자화 오차와 부호화 파라미터를 엔트로피 부호화하여 얻어지는 부호화 스트림(Te)의 정보량이다. 제공 오차는 감산부(102)에서 산출된 예측 오차의 제곱합이다. 계수 λ 는 미리 설정된 제로보다 큰 실수이다. 부호화 파라미터 결정부(110)는 산출한 코스트값이 최소가 되는 부호화 파라미터의 세트를 선택한다. 부호화 파라미터 결정부(110)는 결정한 부호화 파라미터를 예측 파라미터 도출부(120)에 출력한다.

[0636] 아울러, 상술한 실시형태에서의 동화상 부호화 장치(11), 동화상 복호 장치(31)의 일부, 예를 들어 엔트로피 부호부(301), 파라미터 복호부(302), 루프 필터(305), 예측 화상 생성부(308), 역양자화·역변환부(311), 가산부(312), 예측 파라미터 도출부(320), 예측 화상 생성부(101), 감산부(102), 변환·양자화부(103), 엔트로피 부호화부(104), 역양자화·역변환부(105), 루프 필터(107), 부호화 파라미터 결정부(110), 파라미터 부호화부(111), 예측 파라미터 도출부(120)를 컴퓨터에서 실현하도록 할 수도 있다. 그 경우, 이 제어 기능을 실현하기 위한 프로그램을 컴퓨터 판독 가능한 기록 매체에 기록하고, 이 기록 매체에 기록된 프로그램을 컴퓨터 시스템에 읽어들이어 실행함으로써 실현할 수도 있다. 아울러, 여기서 말하는 「컴퓨터 시스템」이란, 동화상 부호화 장치(11), 동화상 복호 장치(31)의 어느 것인가에 내장된 컴퓨터 시스템이며, OS나 주변 기기 등의 하드웨어를 포함하는 것으로 한다. 또한, 「컴퓨터 판독 가능한 기록 매체」란, 플렉시블 디스크, 광자기 디스크, ROM, CD-ROM 등의 가반(portable) 매체, 컴퓨터 시스템에 내장되는 하드 디스크 등의 저장 장치를 말한다. 또한 「컴퓨터 판독 가능한 기록 매체」란, 인터넷 등의 네트워크나 전화 회선 등의 통신 회선을 통해 프로그램을 송신하는 경우의 통신선과 같이, 단시간, 동적으로 프로그램을 보유하는 것, 그 경우의 서버나 클라이언트가 되는 컴퓨터 시스템 내부의 휘발성 메모리와 같이, 일정 시간 프로그램을 보유하고 있는 것도 포함할 수도 있다. 또한 상기 프로그램은 앞에서 설명한 기능의 일부를 실현하기 위한 것일 수도 있고, 또한 앞에서 설명한 기능을 컴퓨터 시스템에 이미 기록되어 있는 프로그램과의 조합으로 실현할 수 있는 것일 수도 있다.

[0637] 또한, 상술한 실시형태에서의 동화상 부호화 장치(11), 동화상 복호 장치(31)의 일부 또는 전부를, LSI(Large Scale Integration) 등의 집적 회로로서 실현할 수도 있다. 동화상 부호화 장치(11), 동화상 복호 장치(31)의 각 기능 블록은 개별적으로 프로세서화할 수도 있고, 일부 또는 전부를 집적하여 프로세서화할 수도 있다. 또한, 집적 회로화의 방법은 LSI로 한정되지 않으며, 전용 회로 또는 범용 프로세서로 실현할 수도 있다. 또한, 반도체 기술의 진보에 의해 LSI를 대체하는 집적 회로화의 기술이 출현한 경우, 당해 기술에 의한 집적 회로를 이용할 수도 있다.

[0638] 이상, 도면을 참조하여 이 발명의 일 실시형태에 대해 상세히 설명했으나, 구체적인 구성이 상술한 것으로 한정되는 것은 아니며, 이 발명의 요지를 벗어나지 않는 범위 내에서 다양한 설계 변경 등을 하는 것이 가능하다.

[0639] 본 발명의 실시형태가 상술한 실시형태로 한정되는 것은 아니며, 청구항에 나타난 범위에서 다양한 변경이 가능하다. 즉, 청구항에 나타난 범위에서 적절히 변경한 기술적 수단을 조합하여 얻어지는 실시형태에 대해서도 본 발명의 기술적 범위에 포함된다.

산업상 이용가능성

[0640] 본 발명의 실시형태는 화상 데이터가 부호화된 부호화 데이터를 복호하는 동화상 복호 장치, 및 화상 데이터가 부호화된 부호화 데이터를 생성하는 동화상 부호화 장치에 적합하게 적용할 수 있다. 또한, 동화상 부호화 장치에 의해 생성되고, 동화상 복호 장치에 의해 참조되는 부호화 데이터의 데이터 구조에 적합하게 적용할 수 있다.

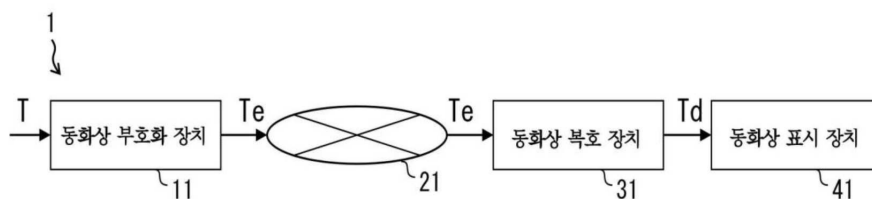
부호의 설명

[0641]

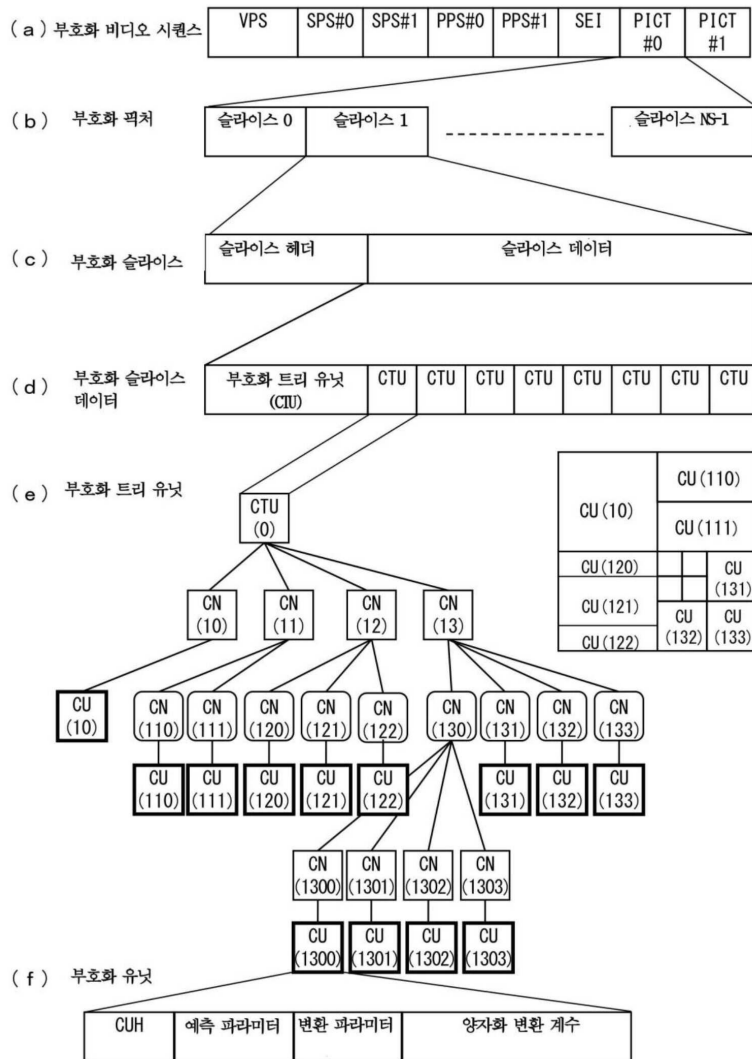
31: 동화상 복호 장치
 301: 엔트로피 복호부
 302: 파라미터 복호부
 3022: CU 복호부
 3024: TU 복호부
 303: 인터 예측 파라미터 도출부
 305, 107: 루프 필터
 306, 109: 참조 픽처 메모리
 307, 108: 예측 파라미터 메모리
 308, 101: 예측 화상 생성부
 309: 인터 예측 화상 생성부
 3092: OOB 처리부
 30921: OOB 판정부
 30922: OOB 마스크 도출부
 311, 105: 역양자화 · 역변환부
 312, 106: 가산부
 320: 예측 파라미터 도출부
 11: 동화상 부호화 장치
 102: 감산부
 103: 변환 · 양자화부
 104: 엔트로피 부호화부
 110: 부호화 파라미터 결정부
 111: 파라미터 부호화부
 112: 인터 예측 파라미터 부호화부
 120: 예측 파라미터 도출부

도면

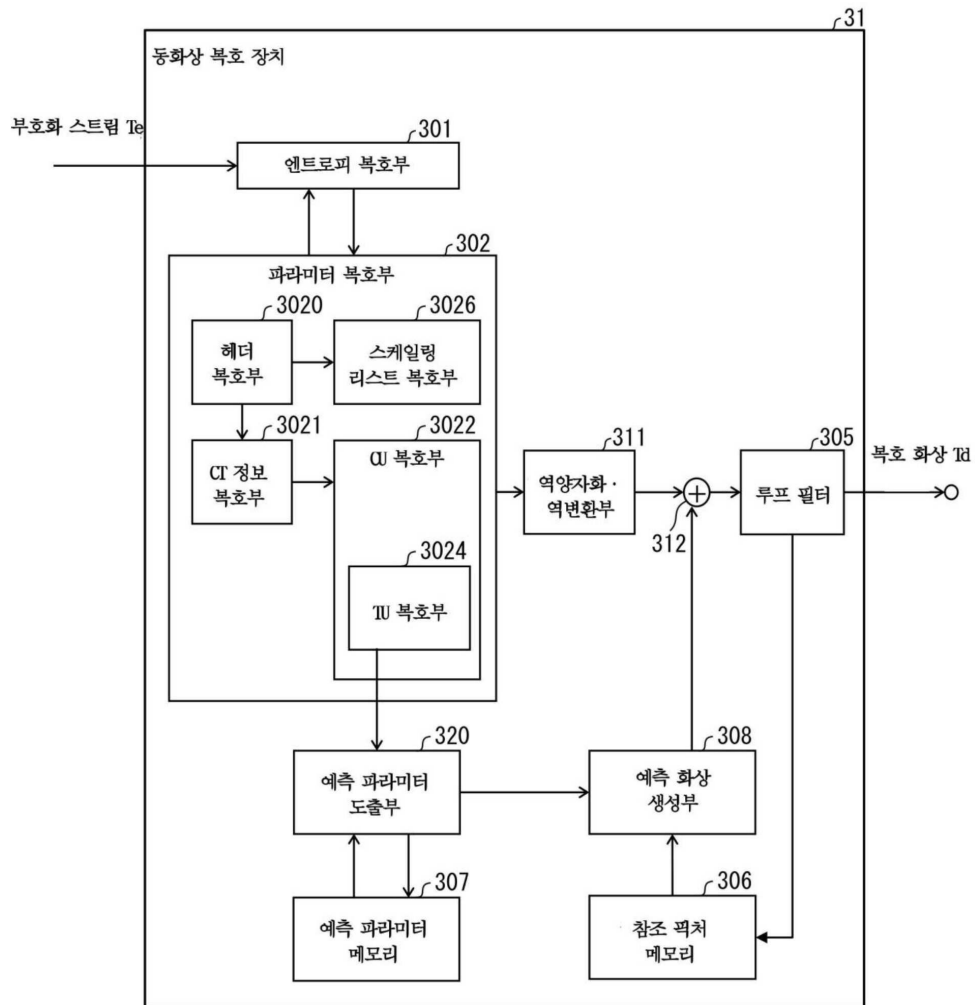
도면1



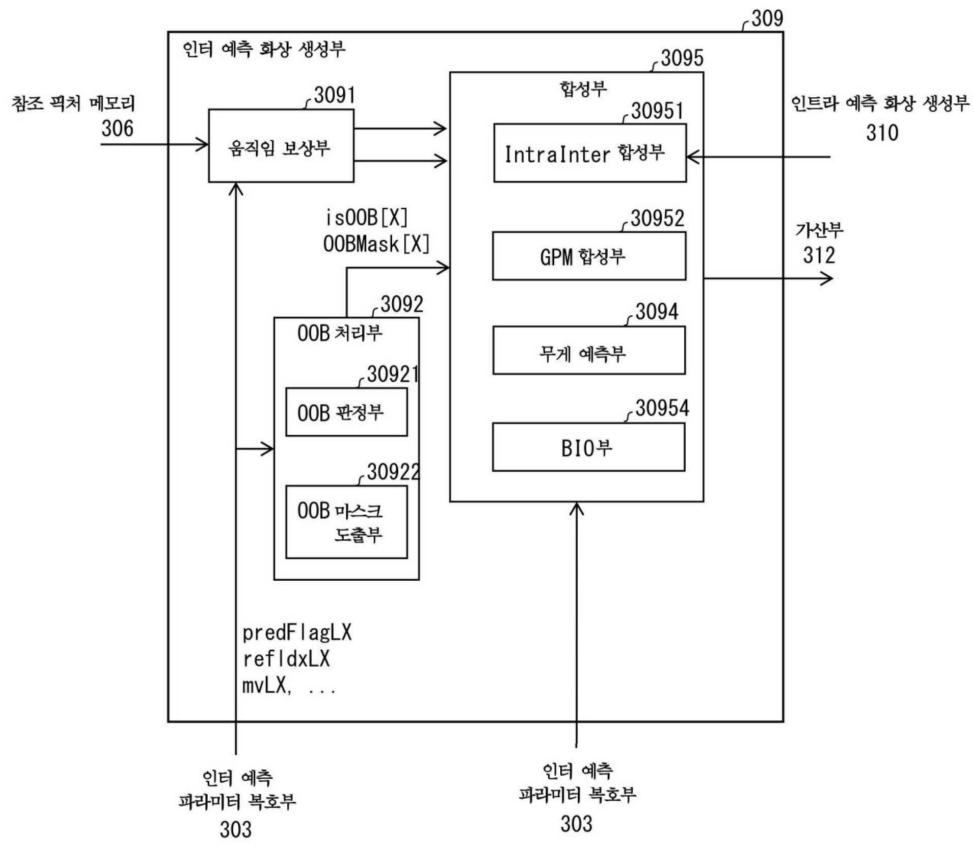
도면2



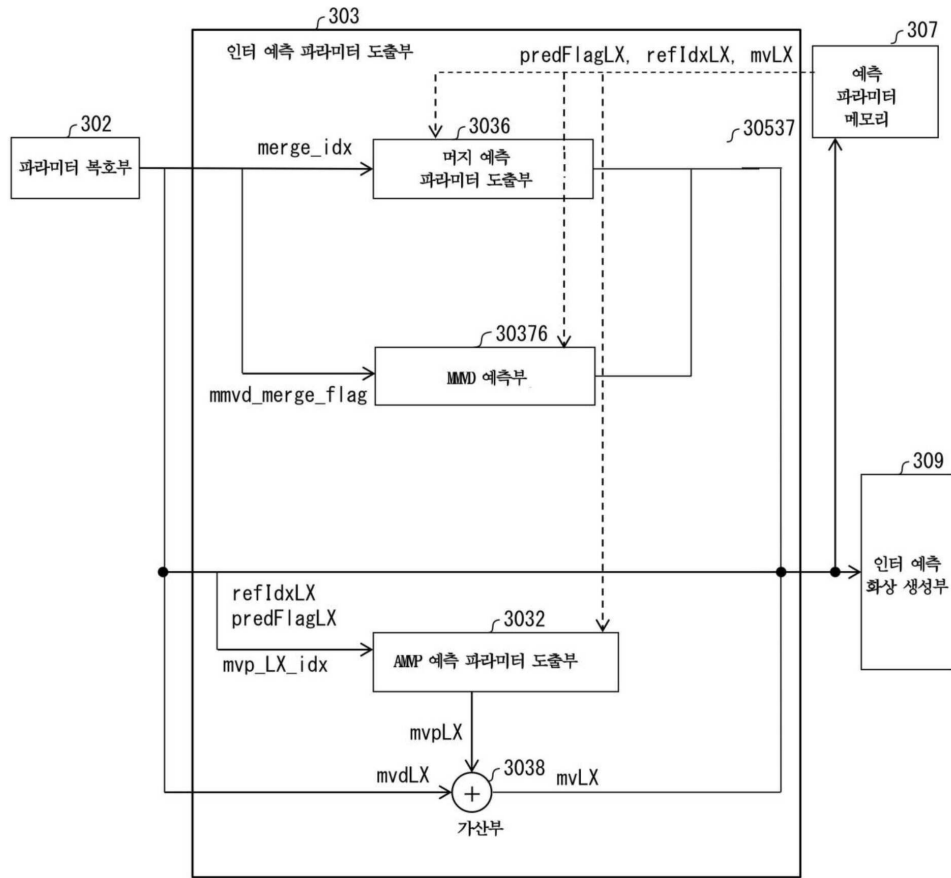
도면3



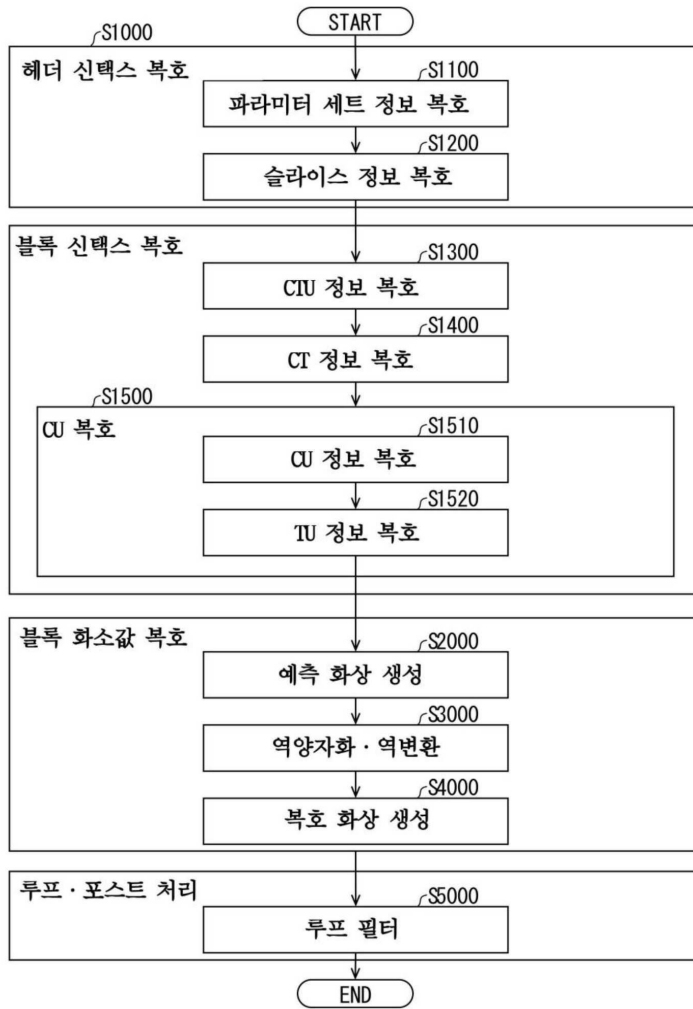
도면4



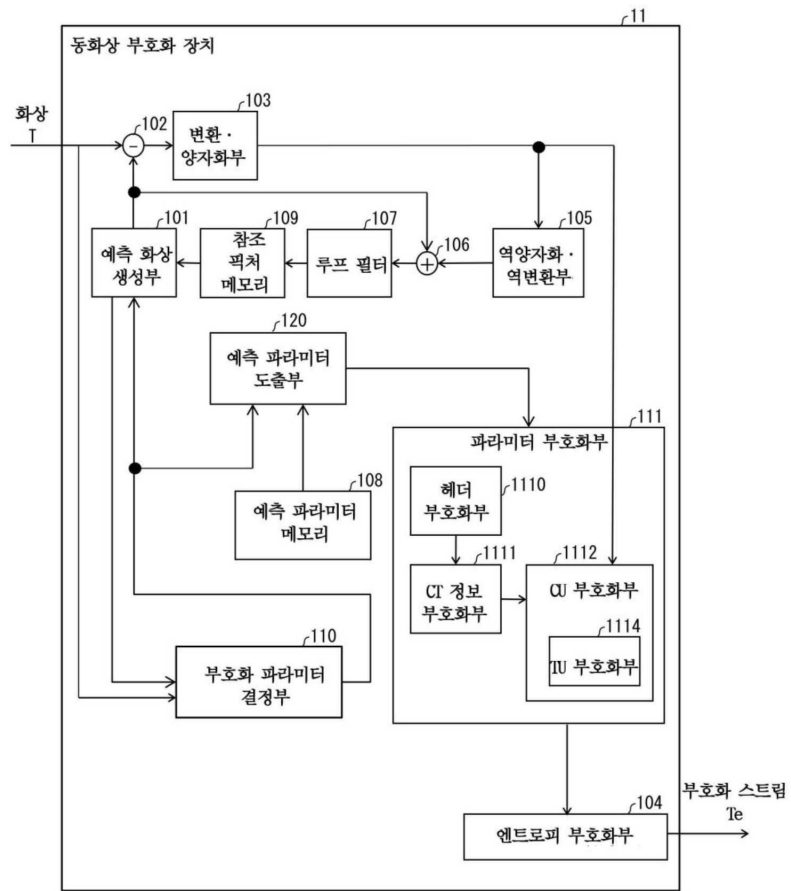
도면5



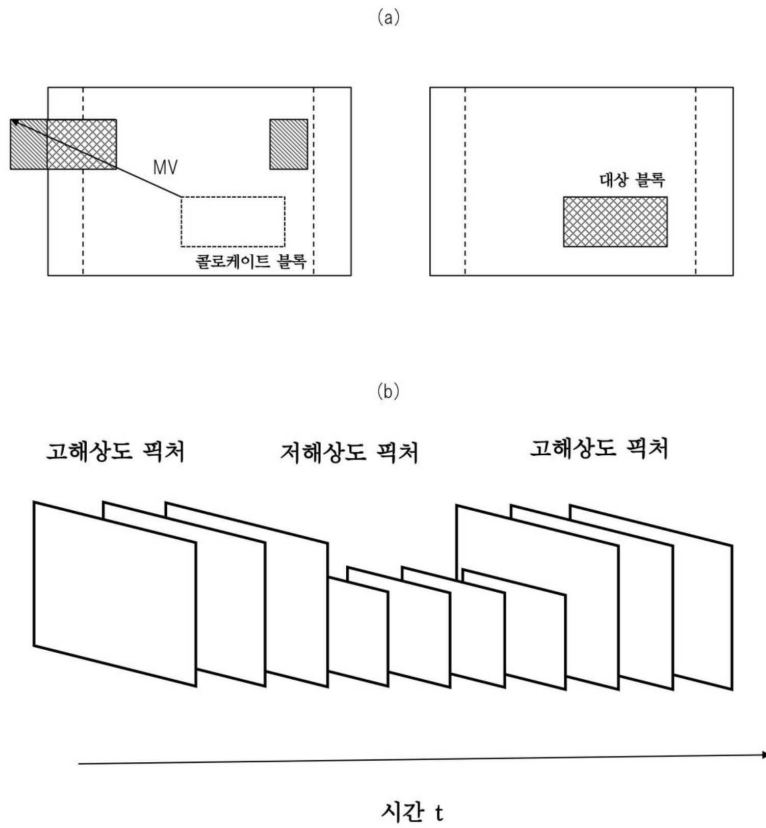
도면6



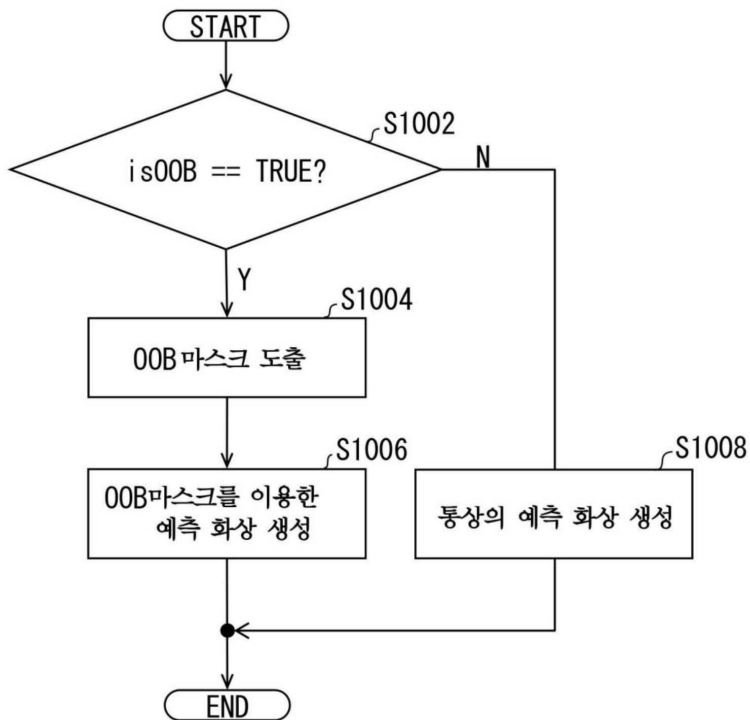
도면7



도면8



도면9



도면10

