

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2013년 9월 6일 (06.09.2013)



(10) 국제공개번호
WO 2013/129878 A1

- (51) 국제특허분류:
H04N 7/32 (2006.01)
- (21) 국제출원번호: PCT/KR2013/001659
- (22) 국제출원일: 2013년 2월 28일 (28.02.2013)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보:
61/604,538 2012년 2월 29일 (29.02.2012) US
- (71) 출원인: 엘지전자 주식회사 (LG ELECTRONICS INC.) [KR/KR]; 150-721 서울시 영등포구 여의도동 20, Seoul (KR).
- (72) 발명자: 김철근 (KIM, Chulkeun); 137-130 서울시 서초구 양재동 221 엘지전자, Convergence R&D 연구소, Seoul (KR). 박승욱 (PARK, Seungwook); 137-130 서울시 서초구 양재동 221 엘지전자, Convergence R&D 연구소, Seoul (KR). 임재현 (LIM, Jaehyun); 137-130 서울시 서초구 양재동 221 엘지전자, Convergence R&D 연구소, Seoul (KR). 전용준 (JEON, Yongjoon); 137-130 서울시 서초구 양재동 221 엘지전자, Convergence

R&D 연구소, Seoul (KR). **박준영 (PARK, Joonyoung)**; 137-130 서울시 서초구 양재동 221 엘지전자, Convergence R&D 연구소, Seoul (KR). **박내리 (PARK, Naeri)**; 137-130 서울시 서초구 양재동 221 엘지전자, Convergence R&D 연구소, Seoul (KR). **전병문 (JEON, Byeongmoon)**; 137-130 서울시 서초구 양재동 221 엘지전자, Convergence R&D 연구소, Seoul (KR).

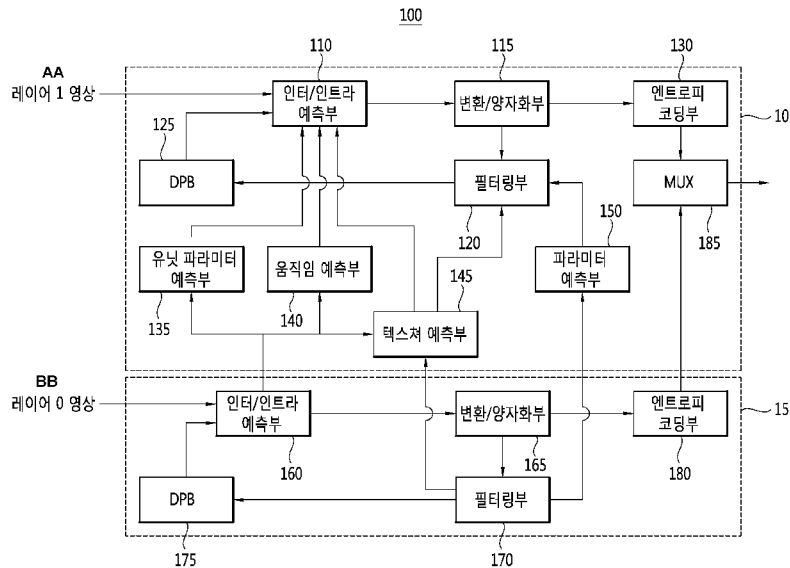
(74) 대리인: 에스앤아이피 특허법인 (S&IP PATENT & LAW FIRM); 135-080 서울시 강남구 테헤란로 14길 5 (역삼동 삼흥역삼빌딩 2층), Seoul (KR).

(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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, 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, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[다음 쪽 계속]

(54) Title: INTER-LAYER PREDICTION METHOD AND APPARATUS USING SAME

(54) 발명의 명칭: 인터 레이어 예측 방법 및 이를 이용하는 장치



(57) Abstract: The present invention relates to an inter-layer prediction method and an apparatus using same. The inter-layer prediction method comprises: a step of inducing inter-layer motion information from a reference layer; and a step of performing a prediction on the current block in the current layer using the inter-layer motion information. The inter-layer motion information may include an inter-layer motion vector induced from the reference layer.

(57) 요약서: 본 발명은 인터 레이어 예측 방법 및 이를 이용한 장치에 관한 것으로서, 참조 레이어로부터 인터 레이어 움직임 정보를 유도하는 단계 및 상기 인터 레이어 움직임 정보를 이용하여 현재 레이어 내 현재 블록에 대한 예측을 수행하는 단계를 포함하며, 상기 인터 레이어 움직임 정보는 상기 참조 레이어로부터 유도되는 인터 레이어 움직임 벡터를 포함할 수 있다.

- 110, 160 ... Inter/intra prediction unit
- 115, 165 ... Transform/quantization unit
- 130, 180 ... Entropy coding unit
- 120, 170 ... Filtering unit
- 135 ... Unit parameter prediction unit
- 140 ... Motion prediction unit
- 145 ... Texture prediction unit
- 150 ... Parameter prediction unit
- AA ... Layer 1 image
- BB ... Layer 0 image

WO 2013/129878 A1



(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, 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, ML, MR, NE, SN, TD, TG).

공개:

- 국제조사보고서와 함께 (조약 제 21 조(3))
- 청구범위 보정 기한 만료 전의 공개이며, 보정서를 접수하는 경우 그에 관하여 별도 공개함 (규칙 48.2(h))

명세서

발명의 명칭: 인터 레이어 예측 방법 및 이를 이용하는 장치 기술분야

[0001] 본 발명은 비디오 압축 기술에 관한 것으로서 더 구체적으로는 스케일러블 비디오 코딩을 수행하는 방법 및 장치에 관한 것이다.

배경기술

[0002] 최근 고해상도, 고품질의 영상에 대한 요구가 다양한 응용 분야에서 증가하고 있다. 영상이 고해상도를 가지고 고품질이 될수록 해당 영상에 관한 정보량도 함께 증가하고 있다.

[0003] 정보량의 증가로 인해 다양한 성능의 장치와 다양한 환경의 네트워크가 등장하고 있다.

[0004] 다양한 성능의 장치와 다양한 환경의 네트워크가 등장함에 따라서, 동일한 콘텐츠를 다양한 품질로 이용할 수 있게 되었다.

[0005] 구체적으로, 단말 장치가 지원할 수 있는 품질의 영상이 다양해지고, 구축된 네트워크 환경이 다양해짐으로써, 어떤 환경에서는 일반적인 품질의 영상을 이용하지만, 또 다른 환경에서는 더 높은 품질의 영상을 이용할 수 있게 된다.

[0006] 예를 들어, 휴대 단말에서 비디오 콘텐츠를 구매한 소비자가 가정 내 대화면의 디스플레이를 통해 동일한 비디오 콘텐츠를 더 큰 화면과 더 높은 해상도로 감상할 수 있게 되는 것이다.

[0007] 최근에는 HD(High Definition) 해상도를 가지는 방송이 서비스되면서 많은 사용자들은 이미 고해상도, 고품질의 영상에 익숙해지고 있고, 서비스 제공자와 사용자들은 HDTV와 더불어 HDTV의 4배 이상의 해상도를 갖는 UHD(Ultra High Definition)의 서비스에도 관심을 기울이고 있다.

[0008] 따라서, 다양한 환경에서 사용자가 요구하는 영상 서비스를 품질에 따라서 다양하게 제공하기 위해 고용량 비디오에 대한 고효율의 인코딩/디코딩 방법을 기반으로 영상의 품질, 예컨대 영상의 화질, 영상의 해상도, 영상의 크기, 비디오의 프레임 레이트 등에 스케일러빌리티를 제공하는 것이 필요하다.

발명의 요약

기술적 과제

[0009] 본 발명의 일 기술적 목적은 스케일러블 비디오 코딩의 효과를 높이고 레이어별 중복 전송되는 정보를 줄이는 방법 및 장치를 제공하는 것이다.

[0010] 본 발명의 다른 기술적 목적은 베이스 레이어의 정보를 이용하여 인헨스먼트 레이어의 인코딩/디코딩 효과를 높이는 방법 및 장치를 제공하는 것이다.

[0011] 본 발명의 또 다른 기술적 목적은 다양한 스케일러빌리티에 대응하여 베이스 레이어의 정보를 활용하는 방법 및 장치를 제공하는 것이다.

[0012] 본 발명의 또 다른 기술적 목적은 다양한 인터 레이어 예측 방법을 통해, 코딩

효유을 향상시키는 방법 및 장치를 제공하는 것이다.

- [0013] 본 발명의 또 다른 기술적 목적은 베이스 레이어의 텍스처 정보, 움직임 정보, 신텍스 정보, 유닛 정보, 파라미터 정보, 레지듀얼 정보, 차분 정보 중 적어도 하나를 이용하여 인헨스먼트 레이어의 코딩 효율을 높이는 방법 및 장치를 제공하는 것이다.

과제 해결 수단

- [0014] 본 발명의 일 실시 형태는 인터 레이어 예측 방법으로서, 참조 레이어로부터 인터 레이어 움직임 정보를 유도하는 단계 및 상기 인터 레이어 움직임 정보를 이용하여 현재 레이어 내 현재 블록에 대한 예측을 수행하는 단계를 포함하며, 상기 인터 레이어 움직임 정보는 상기 참조 레이어로부터 유도되는 인터 레이어 움직임 벡터를 포함한다. 이때, 상기 인터 레이어 움직임 벡터는 상기 참조 레이어의 움직임 벡터를 상기 참조 레이어와 상기 현재 레이어 간 해상도 비율에 따라서 스케일링하여 상기 인터 레이어 움직임 벡터를 유도할 수 있다
- [0015] 본 발명의 일 실시 형태는 인터 레이어 예측 방법을 이용하는 장치로서, 참조 레이어에 대한 예측을 수행하는 제1 예측부 및 상기 제1 예측부에 의한 예측에 기반한 인터 레이어 움직임 정보를 이용하여 상기 현재 레이어 내 현재 블록에 대한 예측을 수행하는 제2 예측부를 포함한다. 이때, 상기 인터 레이어 움직임 정보는 상기 참조 레이어로부터 유도되는 인터 레이어 움직임 벡터를 포함하고, 상기 제2 예측부는 상기 참조 레이어의 움직임 벡터를 상기 참조 레이어와 상기 현재 레이어 간 해상도 비율에 따라서 스케일링할 수 있다.

발명의 효과

- [0016] 본 발명에 의하면, 스케일러블 비디오 코딩의 효과를 높이고 레이어별 중복 전송되는 정보를 줄일 수 있다.
- [0017] 본 발명에 의하면, 베이스 레이어의 정보를 이용하여 인헨스먼트 레이어의 인코딩/디코딩 효과를 높일 수 있다.
- [0018] 본 발명에 의하면 다양한 스케일러빌리티에 대응하여 베이스 레이어의 정보를 활용할 수 있다.
- [0019] 본 발명에 의하면, 다양한 인터 레이어 예측 방법을 통해, 코딩 효율을 향상시킬 수 있다.

도면의 간단한 설명

- [0020] 도 1은 본 발명의 일 실시예에 따라 스케일러빌리티를 지원하는 비디오 인코딩 장치를 개략적으로 도시한 블록도이다.
- [0021] 도 2는 본 발명에 따라서 스케일러블 코딩을 수행하는 인코딩 장치에서의 레이어 간 예측에 관한 일 예를 설명하는 블록도이다.
- [0022] 도 3은 본 발명의 일 실시예에 따라 스케일러빌리티를 지원하는 비디오 디코딩 장치를 개략적으로 도시한 블록도이다.
- [0023] 도 4는 본 발명에 따라서 스케일러블 코딩을 수행하는 디코딩 장치에서의

레이어 간 예측에 관한 일 예를 설명하는 블록도이다.

- [0024] 도 5는 본 발명에 따라서 인터 레이어 인트라 예측이 수행되는 예를 개략적으로 설명하는 도면이다.
- [0025] 도 6은 본 발명에 따라서 인터 레이어 인트라 예측 과정에서 적용되는 리스케일링(다운샘플링/업샘플링)의 일 예를 개략적으로 나타낸 것이다.
- [0026] 도 7은 본 발명에 따라서 페이즈 쉬프트 업샘플링(phase shifted up-samplimng)을 수행하는 일 예를 개략적으로 설명하는 도면이다.
- [0027] 도 8은 본 발명에 따라서 Interpolation_filter_indicator를 이용하는 방법의 일 예로서, Interpolation_filter_indicator의 값이 10인 경우의 샘플링 방법을 간단히 설명하는 도면이다.
- [0028] 도 9는 다른 레이어를 참조하지 않고 레이어 내에서 인터 예측을 수행하는 경우에 이용되는 움직임 정보의 후보들에 대한 일 예를 간단히 나타낸 것이다.
- [0029] 도 10은 본 발명에 따라서 인터 레이어 움직임 예측을 수행하는 방법의 일 예를 개략적으로 설명하는 순서도이다.
- [0030] 도 11은 본 발명에 따라서 참조 레이어의 움직임 정보를 유도하는 방법을 개략적으로 설명하는 도면이다.
- [0031] 도 12는 본 발명에 따라서 mvIL을 스케일링하는 방법을 개략적으로 설명하는 도면이다.
- [0032] 도 13은 본 발명에 따라서 인터 레이어 선택스 예측을 수행하는 방법의 일 예를 간단히 설명하는 도면이다.
- [0033] 도 14는 본 발명에 따라서 인터 레이어 레지듀얼 예측을 적용하는 방법을 개략적으로 설명한 도면이다.
- [0034] 도 15는 본 발명에 따른 인터 레이어 유닛 정보 예측의 일 예를 개략적으로 설명하는 도면이다.
- [0035] 도 16은 본 발명에 따라서 인터 레이어 유닛 예측을 적용하는 일 예를 개략적으로 설명하는 도면이다.
- [0036] 도 17은 본 발명에 따라서 인터 레이어 유닛 예측을 적용하는 다른 예를 개략적으로 설명하는 도면이다.
- [0037] 도 18은 본 발명에 따라서 인터 레이어 유닛 예측을 적용하는 또 다른 예를 개략적으로 설명하는 도면이다.
- [0038] 도 19는 본 발명에 따라서 적응적 인터 레이어 텍스처 예측을 수행하는 방법의 일 예를 개략적으로 설명하는 도면이다.
- [0039] 도 20은 본 발명에 따른 인터 레이어 필터 파라미터 예측에 관하여 개략적으로 설명하는 도면이다.
- [0040] 도 21은 본 발명에 따라서 인터 레이어 차분 모드를 적용하는 경우에, 인트라 예측을 수행하는 방법을 개략적으로 설명하는 도면이다.

발명의 실시를 위한 형태

- [0041] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니다. 본 명세서에서 사용하는 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명의 기술적 사상을 한정하려는 의도로 사용되는 것은 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0042] 한편, 본 발명에서 설명되는 도면상의 각 구성들은 비디오 인코딩 장치/디코딩 장치에서 서로 다른 특징적인 기능들에 관한 설명의 편의를 위해 독립적으로 도시된 것으로서, 각 구성들이 서로 별개의 하드웨어나 별개의 소프트웨어로 구현된다는 것을 의미하지는 않는다. 예컨대, 각 구성 중 두 개 이상의 구성이 합쳐져 하나의 구성을 이룰 수도 있고, 하나의 구성이 복수의 구성으로 나뉘어질 수도 있다. 각 구성이 통합 및/또는 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.
- [0043] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 이하, 도면상의 동일한 구성 요소에 대해서는 동일한 참조 부호를 사용하고 동일한 구성 요소에 대해서 중복된 설명은 생략한다.
- [0044] 스케일러빌리티를 지원하는 비디오 코딩 방법(이하, '스케일러블 코딩'이라 함)에서는 입력 신호들을 레이어별로 처리할 수 있다. 레이어에 따라서 입력 신호(입력 영상)들은 해상도(resolution), 프레임 레이트(frame rate), 비트 텡스(bit-depth), 컬러 포맷(color format), 에스펙트 율(aspect ratio) 중 적어도 하나가 상이할 수 있다.
- [0045] 본 명세서에서, 스케일러블 코딩이라 함은 스케일러블 인코딩과 스케일러블 디코딩을 포함한다.
- [0046] 스케일러블 인코딩/디코딩에서는 레이어 간의 차이를 이용하여, 즉 스케일러빌리티에 기반하여, 레이어 간의 예측을 수행함으로써 정보의 중복 전송/처리를 줄이고 압축 효율을 높일 수 있다.
- [0047] 도 1은 본 발명의 일 실시예에 따라 스케일러빌리티를 지원하는 비디오 인코딩 장치를 개략적으로 도시한 블록도이다.
- [0048] 도 1을 참조하면, 인코딩 장치(100)는 레이어 1에 대한 인코딩부(105)와 레이어 0에 대한 인코딩부(155)를 포함한다.
- [0049] 레이어 0은 베이스 레이어, 참조 레이어 혹은 하위 레이어일 수 있으며, 레이어 1은 인핸스먼트 레이어, 현재 레이어 혹은 상위 레이어일 수 있다.
- [0050] 레이어 1의 인코딩부(105)는 인터/인트라 예측부(110), 변환/양자화부(115),

필터링부(120), DPB(Decoded Picture Buffer, 125), 엔트로피 코딩부(130), 유닛 파라미터 예측부(135), 움직임 예측/리스케일링(rescaling)부(140), 텍스처 예측/리스케일링부(145), 파라미터 예측부(150) 및 MUX(Multiplexer, 185)를 포함한다.

- [0051] 레이어 0의 인코딩부(155)는 인터/인트라 예측부(160), 변환/양자화부(165), 필터링부(170), DPB(175) 및 엔트로피 코딩부(180)를 포함한다.
- [0052] 인터/인트라 예측부(110, 160)는 입력된 영상에 대하여 인터 예측과 인트라 예측을 수행할 수 있다. 인터/인트라 예측부(110, 160)는 소정의 처리 단위로 예측을 수행할 수 있다. 예측의 수행 단위는 코딩 유닛(Coding Unit: CU)일 수도 있고, 예측 유닛(Prediction Unit: PU)일 수도 있으며, 변환 유닛(Transform Unit: TU)일 수도 있다.
- [0053] 예컨대, 인터/인트라 예측부(110, 160)는 CU 단위로 인터 예측을 적용할 것인지 인트라 예측을 적용할 것인지를 결정하고, PU 단위로 예측의 모드를 결정하며, PU 단위 혹은 TU 단위로 예측을 수행할 수도 있다. 수행되는 예측은 예측 블록의 생성과 레지듀얼 블록(레지듀얼 신호)의 생성을 포함한다.
- [0054] 인터 예측을 통해서는 현재 픽처의 이전 픽처 및/또는 이후 픽처 중 적어도 하나의 픽처의 정보를 기초로 예측을 수행하여 예측 블록을 생성할 수 있다. 인트라 예측을 통해서는 현재 픽처 내의 픽셀 정보를 기초로 예측을 수행하여 예측 블록을 생성할 수 있다.
- [0055] 인터 예측의 모드 또는 방법으로서, 스킵(skip) 모드, 머지(merge) 모드, MVP(Motion Vector Prediction) 방법 등이 있다. 인터 예측에서는 예측 대상인 현재 PU에 대하여 참조 픽처를 선택하고, 참조 픽처 내에서 현재 PU에 대응하는 참조 블록을 선택할 수 있다. 인터/인트라 예측부(160)는 참조 블록을 기반으로 예측 블록을 생성할 수 있다.
- [0056] 예측 블록은 정수 샘플 단위로 생성될 수도 있고, 정수 이하 픽셀 단위로 생성될 수도 있다. 이때, 움직임 벡터 역시 정수 픽셀 단위 혹은 정수 픽셀 이하의 단위로 표현될 수 있다.
- [0057] 인터 예측에 있어서 움직임 정보 즉, 참조 픽처의 인덱스, 움직임 벡터, 레지듀얼 신호 등의 정보는 엔트로피 인코딩되어 디코딩 장치에 전달된다. 스킵 모드가 적용되는 경우에는 레지듀얼을 생성, 변환, 양자화, 전송하지 않을 수 있다.
- [0058] 인트라 예측에서 예측 모드는 33개의 방향성 예측 모드와 적어도 2개 이상의 비방향성 모드를 가질 수 있다. 비방향성 모드는 DC 예측 모드 및 플래이너 모드(Planar 모드)을 포함할 수 있다. 인트라 예측에서는 참조 샘플에 필터를 적용한 후 예측 블록을 생성할 수도 있다.
- [0059] PU는 다양한 사이즈/형태의 블록일 수 있으며, 예컨대 인터 예측의 경우에 PU는 $2N \times 2N$ 블록, $2N \times N$ 블록, $N \times 2N$ 블록, 또는 $N \times N$ 블록 (N 은 정수) 동일 수 있다. 인트라 예측의 경우에 PU는 $2N \times 2N$ 블록 또는 $N \times N$ 블록 (N 은 정수) 동일

수 있다. 이때, $N \times N$ 블록 크기의 PU는 특정한 경우에만 적용하도록 설정할 수 있다. 예컨대 최소 크기 CU에 대해서만 $N \times N$ 블록 크기의 PU를 이용하도록 정하거나 인트라 예측에 대해서만 이용하도록 정할 수도 있다. 또한, 상술한 크기의 PU 외에, $N \times mN$ 블록, $mN \times N$ 블록, $2N \times mN$ 블록 또는 $mN \times 2N$ 블록 ($m < 1$) 등의 PU를 더 정의하여 사용할 수도 있다.

- [0060] 변환/양자화부(115, 165)는 변환 블록 단위로 레지듀얼 블록에 대한 변환을 수행하여 변환 계수를 생성하고, 변환 계수를 양자화한다.
- [0061] 변환 블록은 샘플들의 사각형 블록으로서 동일한 변환이 적용되는 블록이다. 변환 블록은 변환 유닛(TU)일 수 있으며, 쿼드 트리(quad tree) 구조를 가질 수 있다.
- [0062] 변환/양자화부(115, 165)는 레지듀얼 블록에 적용된 예측 모드와 변환 블록의 크기에 따라서 변환을 수행해서 변환 계수들의 2차원 어레이를 생성할 수 있다. 예컨대, 레지듀얼 블록에 인트라 예측이 적용되었고 블록이 4×4 의 레지듀얼 배열이라면, 레지듀얼 블록을 DST(Discrete Sine Transform)를 이용하여 변환하고, 그 외의 경우라면 레지듀얼 블록을 DCT(Discrete Cosine Transform)를 이용하여 변환할 수 있다.
- [0063] 또한, 변환/양자화부(115, 165)는 예측 모드와 변환 블록의 크기에 상관없이 특정 변환을 고정적으로 사용할 수도 있다. 예컨대, 변환/양자화부(115, 165)는 모든 변환 블록에 DST만을 적용할 수 있다. 또한, 변환/양자화부(115, 165)는 모든 변환 블록에 DCT만을 적용할 수도 있다.
- [0064] 변환/양자화부(115, 165)는 변환 계수들을 양자화하여 양자화된 변환 계수를 생성할 수 있다.
- [0065] 변환/양자화부(115, 165)는 양자화된 변환 계수를 엔트로피 코딩부(130, 180)로 전달할 수 있다. 이때, 변환/양자화부(165)는 양자화된 변환 계수의 2차원 어레이를 소정의 스캔 순서에 따라 1차원 어레이로 재정렬하여 엔트로피 코딩부(130, 180)로 전달할 수도 있다. 또한, 변환/양자화부(115, 165)는 인트라 예측을 위해, 레지듀얼과 예측 블록을 기반으로 생성된 복원 블록을 변환/양자화하지 않고, 필터링부(120, 170)에 전달할 수 있다.
- [0066] 한편, 변환/양자화부(115, 165)는 필요에 따라서, 변환을 생략(skip)하고 양자화만 수행하거나 변환과 양자화를 모두 생략할 수도 있다. 예컨대, 변환/양자화부(115, 165)는 특정한 예측 방법이 적용되거나 특정 크기를 갖는 블록, 혹은 특정 예측 블록이 적용된 특정한 크기의 블록에 대하여 변환을 생략할 수도 있다.
- [0067] 엔트로피 코딩부(130, 180)는 양자화된 변환 계수들에 대한 엔트로피 인코딩을 수행할 수 있다. 엔트로피 인코딩에는 예를 들어, 지수 골롬(Exponential Golomb), CABAC(Context-Adaptive Binary Arithmetic Coding) 등과 같은 인코딩 방법을 사용할 수 있다.
- [0068] 필터링부(120, 170)는 더블록킹 필터, ALF(Adaptive Loop Filter), SAO(Sample

Adaptive Offset)를 복원된 픽처에 적용할 수 있다.

- [0069] 디블록킹 필터는 복원된 픽처에서 블록 간의 경계에 생긴 왜곡을 제거할 수 있다. ALF(Adaptive Loop Filter)는 디블록킹 필터를 통해 블록이 필터링된 후 복원된 영상과 원래의 영상을 비교한 값을 기초로 필터링을 수행할 수 있다. SAO는 디블록킹 필터가 적용된 레지듀얼 블록에 대하여, 픽셀 단위로 원본 영상과의 오프셋 차이를 복원하며, 밴드 오프셋(Band Offset), 에지 오프셋(Edge Offset) 등의 형태로 적용된다.
- [0070] 필터링부(120, 170)는 디블록킹 필터, ALF, SAO를 모두 적용하지 않고 디블록킹 필터만 적용하거나 디블록킹 필터와 ALF만 적용하거나 디블록킹 필터와 SAO만을 적용할 수도 있다.
- [0071] DPB(125, 175)는 필터링부(125, 170)로부터 복원 블록 또는 복원 픽처를 전달받아 저장할 수 있다. DPB(125, 175)는 복원 블록 또는 픽처를 인터 예측을 수행하는 인터/인트라 예측부(110, 160)에 제공할 수 있다.
- [0072] 레이어 0의 엔트로피 코딩부(180)에서 출력되는 정보와 레이어 1의 엔트로피 코딩부(130)에서 출력되는 정보는 MUX(185)에서 멀티플렉싱되어 비트스트림으로 출력될 수 있다.
- [0073] 한편, 레이어 1의 인코딩부(105)는 레이어 0의 정보를 이용하여 레이어 1의 영상에 대한 예측을 수행하는 인터 레이어 예측을 위해 유닛 파라미터 예측부(135), 움직임 예측/리스케일링부(140), 텍스처 예측/리스케일링부(145), 파라미터 예측부(150) 등을 포함할 수 있다.
- [0074] 유닛 파라미터 예측부(135)는 베이스 레이어의 유닛(CU, PU 및/또는 TU) 정보를 유도하여 인헨스먼트 레이어의 유닛 정보로 사용하도록 하거나, 베이스 레이어의 유닛 정보를 기반으로 인헨스먼트 레이어의 유닛 정보가 결정될 수 있도록 한다.
- [0075] 움직임 예측부(140)는 인터 레이어 움직임 예측을 수행한다. 인터 레이어 움직임 예측을 인터 레이어 인터 예측이라고도 한다. 움직임 예측부(140)는 참조 레이어(베이스 레이어)의 움직임 정보를 이용하여 현재 레이어(인헨스먼트 레이어)의 현재 블록에 대한 예측을 수행할 수 있다.
- [0076] 움직임 예측부(140)는 필요한 경우에, 참조 레이어의 움직임 정보를 스케일링할 수 있다.
- [0077] 텍스처 예측부(145)는 레이어 0의 정보를 기반으로 인터 레이어 텍스처 예측을 수행할 수 있다. 인터 레이어 텍스처 예측은 인터 레이어 인트라 예측 혹은 인트라 BL(Base Layer) 예측이라고도 불린다. 텍스처 예측은 참조 레이어의 참조 블록이 복원되어 있는 경우에 이용될 수 있다. 인터 레이어 텍스처 예측에서는 참조 레이어 내 참조 블록의 텍스처를 인헨스먼트 레이어의 현재 블록에 대한 예측 값으로 사용할 수 있다. 이때, 참조 블록의 텍스처는 업샘플링에 의해 스케일링될 수 있다.
- [0078] 파라미터 예측부(150)는 베이스 레이어에서 사용한 파라미터를 유도하여

- 인헨스먼트 레이어에서 재사용하도록 하거나 베이스 레이어에서 사용한 파라미터를 기반으로 인헨스먼트 레이어에 대한 파라미터를 예측할 수 있다.
- [0079] 한편, 여기서는 설명의 편의를 위해, 레이어 1의 인코딩부(105)가 MUX(185)를 포함하는 것으로 설명하였으나, MUX는 레이어 1의 인코딩부(105) 및 레이어 0의 인코딩부(155)와는 별도의 장치 혹은 모듈일 수 있다.
- [0080] 도 2는 본 발명에 따라서 스케일러블 코딩을 수행하는 인코딩 장치에서의 레이어 간 예측에 관한 일 예를 설명하는 블록도이다.
- [0081] 도 2를 참조하면, 레이어 1의 예측부(210)는 인터/인트라 예측부(220) 및 인터 레이어 예측부(230)를 포함한다.
- [0082] 레이어 1의 예측부(210)는 레이어 0의 정보로부터 레이어 1의 예측에 필요한 인터 레이어 예측을 수행할 수 있다.
- [0083] 예컨대, 인터 레이어 예측부(230)는 레이어 0의 인터/인트라 예측부(250) 및/또는 필터링부(260)로부터 레이어 0의 정보를 전달받아 레이어 1의 예측에 필요한 인터 레이어 예측을 수행할 수 있다.
- [0084] 레이어 1의 인터/인트라 예측부(220)는 레이어 1의 정보를 이용하여 인터 예측 혹은 인트라 예측을 수행할 수 있다.
- [0085] 또한, 레이어 1의 인터/인트라 예측부(220)는 인터 레이어 예측부(230)로부터 전달된 정보를 이용하여 레이어 0의 정보에 기반한 예측을 수행할 수도 있다.
- [0086] 아울러, 레이어 1의 필터링부(240)는 레이어 0의 정보에 기반하여 필터링을 수행할 수도 있고, 레이어 0의 정보에 기반하여 필터링을 수행할 수도 있다. 레이어 0의 정보는 레이어 0의 필터링부(260)으로부터 레이어 1의 필터링부(240)에 전달될 수도 있고, 레이어 1의 인터 레이어 예측부(230)로부터 레이어 1의 필터링부(240)에 전달될 수도 있다.
- [0087] 한편, 레이어 0으로부터 인터 레이어 예측부(230)로 전달되는 정보로는 레이어 0의 유닛 파라미터에 관한 정보, 레이어 0의 움직임 정보, 레이어 0의 텍스처 정보, 레이어 0의 필터 파라미터 정보 중 적어도 하나일 수 있다.
- [0088] 따라서, 인터 레이어 예측부(230)는 도 1에서 인터 레이어 예측을 수행하는 유닛 파라미터 예측부(135), 움직임 예측부(140), 텍스처 예측부(145), 파라미터 예측부(150) 중 일부 또는 전부를 포함할 수 있다.
- [0089] 또한, 레이어 1에서, 인터/인트라 예측부(220)는 도 1의 인터/인트라 예측부(110)에 대응할 수 있고, 필터링부(240)는 도 1의 필터링부(120)에 대응할 수 있다. 레이어 0에서, 인터/인트라 예측부(250)는 도 1의 인터/인트라 예측부(160)에 대응할 수 있고, 필터링부(260)는 도 1의 필터링부(170)에 대응할 수 있다.
- [0090] 도 3은 본 발명의 일 실시예에 따라 스케일러블리티를 지원하는 비디오 디코딩 장치를 개략적으로 도시한 블록도이다.
- [0091] 도 3을 참조하면, 디코딩 장치(300)는 레이어 1의 디코딩부(310)와 레이어 0의 디코딩부(350)를 포함한다.

- [0092] 레이어 0은 베이스 레이어, 참조 레이어 혹은 하위 레이어일 수 있으며, 레이어 1은 인헨스먼트 레이어, 현재 레이어 혹은 상위 레이어일 수 있다.
- [0093] 레이어 1의 디코딩부(310)는 엔트로피 디코딩부(315), 재정렬부(320), 역양자화부(325), 역변환부(330), 예측부(335), 필터링부(340), 메모리를 포함할 수 있다.
- [0094] 레이어 0의 디코딩부(350)는 엔트로피 디코딩부(355), 재정렬부(360), 역양자화부(365), 역변환부(370), 필터링부(380), 메모리(385)를 포함할 수 있다.
- [0095] 인코딩 장치로부터 영상 정보를 포함하는 비트스트림이 전송되면, DEMUX(305)는 레이어별로 정보를 디멀티플렉싱하여 각 레이어별 디코딩 장치로 전달할 수 있다.
- [0096] 엔트로피 디코딩부(315, 355)는 인코딩 장치에서 사용한 엔트로피 코딩 방식에 대응하여 엔트로피 디코딩을 수행할 수 있다. 예컨대, 인코딩 장치에서 CABAC이 사용된 경우에, 엔트로피 디코딩부(315, 355)도 CABAC을 이용하여 엔트로피 디코딩을 수행할 수 있다.
- [0097] 엔트로피 디코딩부(315, 355)에서 디코딩된 정보 중 예측 블록을 생성하기 위한 정보는 예측부(335, 375)로 제공되고, 엔트로피 디코딩부(315, 355)에서 엔트로피 디코딩이 수행된 레지듀얼 값, 즉 양자화된 변환 계수는 재정렬부(320, 360)로 입력될 수 있다.
- [0098] 재정렬부(320, 360)는 엔트로피 디코딩부(315, 355)에서 엔트로피 디코딩된 비트스트림의 정보, 즉 양자화된 변환 계수를 인코딩 장치에서 재정렬한 방법을 기초로 재정렬할 수 있다.
- [0099] 예컨대, 재정렬부(320, 360)는 1차원 어레이의 양자화된 변환 계수들을 다시 2차원 어레이의 계수들로 재정렬할 수 있다. 재정렬부(320, 360)는 현재 블록(변환 블록)에 적용된 예측 모드 및/또는 변환 블록의 크기를 기반으로 스케닝을 수행하여 계수(양자화된 변환 계수)들의 2차원 어레이를 생성할 수 있다.
- [0100] 역양자화부(325, 365)는 인코딩 장치에서 제공된 양자화 파라미터와 재정렬된 블록의 계수값을 기초로 역양자화를 수행하여 변환 계수들을 생성할 수 있다.
- [0101] 역양자화부(325, 365)는 소정의 조건에 따라서 혹은 인코딩 장치에서의 양자화 방식에 따라서, 엔트로피 디코딩된 레지듀얼을 역양자화하지 않고 역변환부(330, 370)에 전달할 수도 있다.
- [0102] 역변환부(330, 370)는 변환 계수들에 대하여 인코딩 장치의 변환부가 수행한 변환에 대한 역변환을 수행할 수 있다. 역변환부(330, 370)는 인코딩 장치에서 수행된 DCT(Discrete Cosine Transform) 및 DST(Discrete Sine Transform)에 대해 역DCT 및/또는 역DST를 수행할 수 있다.
- [0103] 인코딩 장치에서 DCT 및/또는 DST는 예측 방법, 현재 블록의 크기 및 예측 방향 등 복수의 정보에 따라 선택적으로 수행될 수 있고, 디코딩 장치의 역변환부(330, 370)는 인코딩 장치에서 수행된 변환 정보를 기초로 역변환을

수행할 수 있다.

- [0104] 예컨대, 역변환부(330, 370)은 예측 모드/블록 크기에 따라서 역DCT와 역DST를 적용할 수 있다. 가령, 역변환부(330, 370)은 인트라 예측이 적용된 4x4 루마 블록에 대해서 역DST를 적용할 수도 있다.
- [0105] 또한, 역변환부(330, 370)는 예측 모드/블록 크기에 상관 없이, 특정 역변환 방법을 고정적으로 사용할 수도 있다. 예컨대, 역변환부(330, 370)는 모든 변환 블록에 역DST만을 적용할 수 있다. 또한, 역변환부(330, 370)는 모든 변환 블록에 역DCT만을 적용할 수도 있다.
- [0106] 역변환부(330, 370)는 변환 계수들 혹은 변환 계수의 블록을 역변환하여 레지듀얼 혹은 레지듀얼 블록을 생성할 수 있다.
- [0107] 역변환부(330, 370)는 또한, 필요에 따라서 혹은 인코딩 장치에서 인코딩된 방식에 따라서, 변환을 생략(skip) 할 수도 있다. 예컨대, 역변환(330, 370)는 특정한 예측 방법이 적용되거나 특정 크기를 갖는 블록, 혹은 특정 예측 블록이 적용된 특정한 크기의 블록에 대하여 변환을 생략할 수도 있다.
- [0108] 예측부(335, 375)는 엔트로피 디코딩부(315, 355)로부터 전달된 예측 블록 생성 관련 정보와 메모리(345, 385)에서 제공된 이전에 디코딩된 블록 및/또는 픽처 정보를 기초로 현재 블록에 대한 예측 블록을 생성할 수 있다.
- [0109] 현재 블록에 대한 예측 모드가 인트라 예측(intra prediction) 모드인 경우에, 예측부(335, 375)는 현재 픽처 내의 픽셀 정보를 기초로 현재 블록에 대한 인트라 예측을 수행할 수 있다.
- [0110] 현재 블록에 대한 예측 모드가 인터 예측(inter prediction) 모드인 경우에, 예측부(335, 375)는 현재 픽처의 이전 픽처 또는 이후 픽처 중 적어도 하나의 픽처에 포함된 정보를 기초로 현재 블록에 대한 인터 예측을 수행할 수 있다. 인터 예측에 필요한 움직임 정보의 일부 또는 전부는 인코딩 장치로부터 수신한 정보를 확인하고, 이에 대응하여 유도될 수 있다.
- [0111] 인터 예측의 모드로서 스킵 모드가 적용되는 경우에는 인코딩 장치로부터 레지듀얼이 전송되지 않으며 예측 블록을 복원 블록으로 할 수 있다.
- [0112] 한편, 레이어 1의 예측부(335)는 레이어 1 내의 정보만을 이용하여 인터 예측 또는 인트라 예측을 수행할 수도 있고, 다른 레이어(레이어 0)의 정보를 이용하여 인터 레이어 예측을 수행할 수도 있다.
- [0113] 예컨대, 레이어 1의 예측부(335)는 레이어 1의 움직임 정보, 레이어 1의 텍스처 정보, 레이어 1의 유닛 정보, 레이어 1의 파라미터 정보 중 하나를 이용하여 현재 블록에 대한 예측을 수행할 수 있다. 또한, 레이어 1의 예측부(335)는 레이어 1의 움직임 정보, 레이어 1의 텍스처 정보, 레이어 1의 유닛 정보, 레이어 1의 파라미터 정보 중 복수의 정보를 이용하여 현재 블록에 대한 예측을 수행할 수도 있다.
- [0114] 레이어 1의 예측부(335)는 레이어 0의 예측부(375)로부터 레이어 1의 움직임 정보를 전달받아서 움직임 예측을 수행할 수 있다. 인터 레이어 움직임 예측을

인터 레이어 인터 예측이라고도 한다. 인터 레이어 움직임 예측에 의해, 참조 레이어(베이스 레이어)의 움직임 정보를 이용하여 현재 레이어(인헨스먼트 레이어)의 현재 블록에 대한 예측이 수행될 수 있다. 예측부(335)는 필요한 경우에, 참조 레이어의 움직임 정보를 스케일링 하여 이용할 수도 있다.

- [0115] 레이어 1의 예측부(335)는 레이어 0의 예측부(375)로부터 레이어 1의 텍스처 정보를 전달받아서 인터 레이어 텍스처 예측을 수행할 수 있다. 인터 레이어 텍스처 예측은 인터 레이어 인트라 예측 혹은 인트라 BL(Base Layer) 예측이라고도 불린다. 인터 레이어 텍스처 예측은 참조 레이어의 참조 블록이 복원되어 있는 경우에 이용할 수 있다. 인터 레이어 텍스처 예측에서는 참조 레이어 내 참조 블록의 텍스처를 인헨스먼트 레이어의 현재 블록에 대한 예측 값으로 사용할 수 있다. 이때, 참조 블록의 텍스처는 업샘플링에 의해 스케일링될 수 있다.
- [0116] 레이어 1의 예측부(335)는 레이어 0의 예측부(375)로부터 레이어 1의 유닛 파라미터 정보를 전달받아서 유닛 파라미터 예측을 수행할 수 있다. 유닛 파라미터 예측에 의해, 베이스 레이어의 유닛(CU, PU 및/또는 TU) 정보가 인헨스먼트 레이어의 유닛 정보로 사용되거나, 베이스 레이어의 유닛 정보를 기반으로 인헨스먼트 레이어의 유닛 정보가 결정될 수 있다.
- [0117] 레이어 1의 예측부(335)는 레이어 0의 예측부(375)로부터 레이어 1의 필터링에 관한 파라미터 정보를 전달받아서 파라미터 예측을 수행할 수도 있다. 파라미터 예측에 의해, 베이스 레이어에서 사용한 파라미터를 유도하여 인헨스먼트 레이어에서 재사용하거나, 베이스 레이어에서 사용한 파라미터를 기반으로 인헨스먼트 레이어에 대한 파라미터를 예측할 수 있다.
- [0118] 가산기(390, 395)는 예측부(335, 375)에서 생성된 예측 블록과 역변환부(330, 370)에서 생성된 레지듀얼 블록을 이용해 복원 블록을 생성할 수 있다. 이 경우, 가산기(390, 395)를 복원 블록을 생성하는 별도의 유닛(복원 블록 생성부)로 볼 수 있다.
- [0119] 가산기(390, 395)에서 복원된 블록 및/또는 픽처는 필터링부(340, 380)로 제공될 수 있다.
- [0120] 필터링부(340, 380)는 복원된 블록 및/또는 픽처에 디블록킹 필터링, SAO(Sample Adaptive Offset) 및/또는 ALF 등을 적용할 수 있다.
- [0121] 필터링부(340, 380)는 디블록킹 필터, ALF, SAO를 모두 적용하지 않고, 디블록킹 필터만 적용하거나, 디블록킹 필터와 ALF만 적용하거나, 디블록킹 필터와 SAO만을 적용할 수도 있다.
- [0122] 도 3의 예를 참조하면, 레이어 1의 필터링부(340)는 레이어 1의 예측부(335) 및/또는 레이어 1의 필터링부(380)으로부터 전달되는 파라미터 정보를 이용하여 복원된 픽처에 대한 필터링을 수행할 수도 있다. 예컨대, 레이어 1에서 필터링부(340)는 레이어 0에서 적용된 필터링의 파라미터로부터 예측된 파라미터를 이용하여 레이어 1에 대한 혹은 레이어 간의 필터링을 적용할 수

있다.

- [0123] 메모리(345, 385)는 복원된 픽처 또는 블록을 저장하여 참조 픽처 또는 참조 블록으로 사용할 수 있도록 할 수 있다. 메모리(345, 385)는 저장된 복원 픽처를 소정의 출력부(미도시) 혹은 디스플레이(미도시)를 통해 출력할 수도 있다.
- [0124] 도 3의 예에서는 재정렬부, 역양자화부, 역변환부 등으로 나누어 설명하였으나, 도 1의 인코딩 장치에서와 같이, 역양자화/역변환부의 한 모듈에서 재정렬, 역양자화, 역변환을 순서대로 수행하도록 디코딩 장치를 구성할 수도 있다.
- [0125] 반대로, 도 3의 예에서는 예측부로 설명하였으나, 도 1의 예와 같이 레이어 1의 예측부는 다른 레이어(레이어 0)의 정보를 이용하여 예측을 수행하는 인터 레이어 예측부와 다른 레이어(레이어 0)의 정보를 이용하지 않고 예측을 수행하는 인터/인트라 예측부를 포함할 수도 있다.
- [0126] 도 4는 본 발명에 따라서 스케일러블 코딩을 수행하는 디코딩 장치에서의 레이어 간 예측에 관한 일 예를 설명하는 블록도이다.
- [0127] 도 4를 참조하면, 레이어 1의 예측부(410)는 인터/인트라 예측부(420) 및 인터 레이어 예측부(430)를 포함한다.
- [0128] 레이어 1의 예측부(410)는 레이어 0의 정보로부터 레이어 1의 예측에 필요한 인터 레이어 예측을 수행할 수 있다.
- [0129] 예컨대, 인터 레이어 예측부(430)는 레이어 0의 인터/인트라 예측부(450) 및/또는 필터링부(460)로부터 레이어 0의 정보를 전달받아 레이어 1의 예측에 필요한 인터 레이어 예측을 수행할 수 있다.
- [0130] 레이어 1의 인터/인트라 예측부(420)는 레이어 1의 정보를 이용하여 인터 예측 혹은 인트라 예측을 수행할 수 있다.
- [0131] 또한, 레이어 1의 인터/인트라 예측부(420)는 인터 레이어 예측부(430)로부터 전달된 정보를 이용하여 레이어 0의 정보에 기반한 예측을 수행할 수도 있다.
- [0132] 레이어 1의 필터링부(440)는 레이어 0의 정보에 기반하여 필터링을 수행할 수도 있고, 레이어 0의 정보에 기반하여 필터링을 수행할 수도 있다. 레이어 0의 정보는 레이어 0의 필터링부(460)으로부터 레이어 1의 필터링부(440)에 전달될 수도 있고, 레이어 1의 인터 레이어 예측부(430)로부터 레이어 1의 필터링부(440)에 전달될 수도 있다.
- [0133] 한편, 레이어 0으로부터 인터 레이어 예측부(430)로 전달되는 정보로는 레이어 0의 유닛 파라미터에 관한 정보, 레이어 0의 움직임 정보, 레이어 0의 텍스처 정보, 레이어 0의 필터 파라미터 정보 중 적어도 하나일 수 있다.
- [0134] 레이어 1에서 예측부(410)는 도 3의 예측부(335)에 대응할 수 있고, 필터링부(440)는 도 3의 필터링부(340)에 대응할 수 있다. 레이어 0에서, 예측부(450)는 도 3의 예측부(375)에 대응할 수 있고, 필터링부(460)는 도 3의 필터링부(380)에 대응할 수 있다.
- [0135] 또한, 도시하지는 않았지만, 인터 레이어 예측부(430)는 수행하는 인터 레이어 예측의 종류(예컨대, 움직임 예측, 텍스처 예측, 유닛 파라미터 예측, 파라미터

예측)에 따라서, 움직임 예측부, 텍스처 예측부, 유닛 파라미터 예측부, 파라미터 예측부를 포함할 수도 있다.

[0136] 스케일러블 비디오 코딩에서는 다른 레이어의 정보를 이용하여 현재 레이어의 정보를 예측하는 인터 레이어 예측이 수행될 수 있다. 도 1 내지 도 4의 예에서 설명한 바와 같이, 인터 레이어 예측의 예로서 움직임 예측, 텍스처 예측, 유닛 예측, 파라미터 예측 등을 고려할 수 있다.

[0137] 이하, 도면을 참조하여, 각 인터 레이어 예측에 대하여 구체적으로 설명한다.

[0138]

[0139] 인터 레이어 인트라 예측

[0140] 인터 레이어 인트라 예측은 인터 레이어 텍스처 예측 혹은 인트라 BL(Base Layer) 예측이라고도 불린다. 본 명세서에서는 설명의 편의를 위해, 인터 레이어 인트라 예측과 텍스처 예측 그리고 인트라 BL의 명칭들을 혼용하여 사용할 수도 있다.

[0141] 이때, 베이스 레이어의 복원된 영상과 인헨스먼트 레이어의 영상 크기 또는 해상도를 동일하게 맞추기 위해, 베이스 레이어의 복원된 영상에 대한 업샘플링이 수행될 수 있다.

[0142] 예컨대, DCTIF(DCT based Interpolation Filter)를 적용하여 업샘플링이 수행될 수 있다. 가령, 루마 샘플에 대해서는 8 탭의 DCTIF를 적용하여 업샘플링을 수행하고, 크로마 샘플에 대해서는 4 탭의 DCTIF를 적용하여 업샘플링을 수행할 수 있다.

[0143] 업샘플링은 인터폴레이션(interpolation)을 적용함으로써 수행될 수 있다.

[0144] 한편, 본 발명에 따른 인터 레이어 인트라 예측은 코딩/디코딩의 처리 단위별로 종래와는 다른 특징을 나타낸다. 예컨대, 코딩 블록(예컨대, CU) 레벨에서 보면, 베이스 레이어의 블록 크기에 종속되지 않고, 독자적인 CU 파티션에 따라서 적용될 수 있다.

[0145] 또한, 예측 블록(예컨대, PU) 레벨에서 보면, 인터 레이어 인트라 예측의 코스트는 현재 레이어(인헨스먼트 레이어) 내에서의 인트라 예측에 대한 코스트와 비교될 수 있다. 이때, 인터 레이어 인트라 예측은 인트라 예측 모드를 사용하는 것이 아니라, 베이스 레이어(참조 레이어)의 복원 영상을 참조 영상으로 사용하여 인트라 예측과 RDO(Rate-Distortion Optimization)를 비교하게 된다.

[0146] 인터 레이어 인트라 예측을 적용하기 위해, PU에 대한 새로운 모드를 생성하지 않고, 상위 레벨에서 플래그 형태도 인터 레이어 인트라 예측이 적용되는지가 시그널링될 수 있다. 예컨대, 블록 모드 혹은 파티셔닝에 대해 파싱이 수행되기 전에 분할 플래그(split_flag) 다음 위치에서 플래그 형태로 인터 레이어 인트라 예측의 사용 여부에 관한 지시가 전송될 수 있다.

[0147] 변환 블록(예컨대, TU) 레벨에서 보면, HEVC의 쿼드 트리 구조를 그대로 유지한 채로, 인터 레이어 인트라 예측을 적용하여 변환을 수행할 수 있다.

- [0148] 도 5는 본 발명에 따라서 인터 레이어 인트라 예측이 수행되는 예를 개략적으로 설명하는 도면이다.
- [0149] 도 5의 예에서는, 베이스 레이어의 영상(510)과 인헨스먼트 레이어의 영상(530)은 동일하게 64x64의 크기를 가지지만, 인헨스먼트 레이어의 영상(530)이 베이스 레이어의 영상(510)보다 4 배의 해상도를 가지고 있다.
- [0150] 따라서, 인헨스먼트 레이어의 영상(530)의 예측시에 베이스 레이어의 영상(510)을 참조하기 위해, 인트라 예측 모드로 복원된 베이스 레이어의 영상(510)을 업샘플링하여 참조 영상(520)을 구성한 후, 인헨스먼트 레이어의 예측에 이용한다.
- [0151] 인헨스먼트 레이어의 영상(530)에서 그림자 처리된 부분은, 인터 레이어 인트라 예측이 적용되지 않는 블록을 나타낸다.
- [0152] 이처럼 인터 레이어 인트라 예측을 적용하는 경우에는 베이스 레이어의 영상에 대한 리스케일링이 수행될 수 있다.
- [0153] 구체적으로, 인코딩 장치에서는 입력 영상이 서로 다른 해상도를 가지는 복수의 레이어로 나누어 인코딩/디코딩될 수 있도록 입력 영상을 다운샘플링하는 과정을 수행할 수 있다. 또한, 인코딩 장치/디코딩 장치는 인코딩/디코딩 과정에서 하위 레이어의 영상을 참조 영상으로 사용하기 위해 하위 레이어의 복원 영상을 업샘플링할 수도 있다.
- [0154] 다운샘플링과 업샘플링을 수행하는 경우에, 페이즈(phase) 특성이 맞지 않으면 인코딩/디코딩 과정에서 손실이 발생하게 되며 코딩 성능에 직접적인 영향을 미친다.
- [0155] 도 6은 본 발명에 따라서 인터 레이어 인트라 예측 과정에서 적용되는 리스케일링(다운샘플링/업샘플링)의 일 예를 개략적으로 나타낸 것이다.
- [0156] 도 6(a)에서는 동일 위치(col-located) 정수 샘플을 다운샘플링된 샘플로 사용하는 경우를 설명하고 있다. 도 6(b)에서는 동일 위치 정수 샘플을 사용하지 않고 1/2 페이즈 어긋난 샘플을 생성하여 사용하는 예를 설명하고 있다.
- [0157] 도 6(a)와 도 6(b)에 도시된 바와 같이, 오리지널 샘플을 다운샘플링한 샘플에 하프펠(half-pel) 업샘플링 필터를 적용하면 다운샘플링된 샘플 위치로부터 1/2 페이즈 어긋난 위치에 샘플이 생성될 수 있다.
- [0158] 다운샘플링을 할 때 페이즈 이동이 없었던 도 6(a)의 경우에는 오리지널 샘플과 동일한 위치에 업샘플링에 의해 샘플들이 생성된다. 반면에, 다운샘플링을 할 때 페이즈 이동이 있었던 도 6(b)의 경우에는 오리지널 샘플과는 다른 위치(1/2 페이즈 어긋난 위치)에 샘플들이 생성된다. 따라서, 도 6(b)의 경우에는 페이즈의 불일치에 의해 오리지널 샘플과 업샘플링된 샘플 사이에 손실이 발생할 수 있다.
- [0159] 이 문제를 해결하기 위해, 업샘플링 과정에서 오리지널 샘플(오리지널 영상)들과의 페이즈를 일치시키는 페이즈 쉬프트 업샘플링을 수행하는 것을 고려할 수 있다.
- [0160] 도 7은 본 발명에 따라서 페이즈 쉬프트 업샘플링(phase shifted up-sampling)을

수행하는 일 예를 개략적으로 설명하는 도면이다.

- [0161] 도 7에서 아래쪽 샘플들은 오리지널 샘플로부터 1/2 페이즈 이동하여 다운샘플링된 샘플들과 다운샘플링된 샘플들로부터 업샘플링된 샘플들을 함께 나타낸 것이다.
- [0162] 도 7의 예에서는 다운샘플링 과정에서의 페이즈 이동을 보상하기 위해, 업샘플링 과정에서 다운샘플링된 샘플들에 대하여 1/4 페이즈, 3/4 페이즈 위치에서 업샘플링을 수행한다.
- [0163] 다운샘플링된 샘플들에 대하여 1/4 페이즈, 3/4 페이즈의 샘플을 보간함으로써 페이즈의 미스매치를 제거할 수 있다. 도 7을 참조하면, 다운샘플링된 샘플들에 대한 1/4 페이즈, 3/4 페이즈 위치에서 업샘플링된 샘플들은 오리지널 샘플들과 페이즈 차이가 없는 것을 확인할 수 있다.
- [0164] 이처럼 업샘플링 필터링을 적용할 때, 페이즈 이동을 얼마나 하여 오리지널 샘플과 페이즈를 일치시킬 것인지는 다운샘플링시에 적용된 페이즈 이동에 따라서 결정될 수 있다.
- [0165] 예컨대, 디코딩 장치에서 오리지널 샘플과 업샘플링된 샘플 간에 페이즈의 불일치가 발생하지 않도록 업샘플링 필터링을 적용하기 위해서는, 인코딩 장치에서 적용한 다운샘플링 필터에 대한 정보 또는 디코딩 장치에서 사용할 업샘플링 필터에 대한 정보가 인코딩 장치로부터 디코딩 장치로 전송될 필요가 있다.
- [0166] 아래 표 1은 본 발명에 따라서 페이즈를 일치시키기 위해 전송되는 다운샘플링/업샘플링 필터 정보의 일 예를 나타낸 것이다.

[0167] <표 1>

[0168]

필터 길이: 사용한 필터의 탭 크기

필터 계수: 각 탭에 적용되는 계수 값

쉬프트 페이즈(shifted phase): 다운샘플링/업샘플링된 영상의 페이즈 변경 값

- [0169] 필터의 정보가 인코딩 장치로부터 디코딩 장치로 직접(explicitly) 전송되는 대신 룩업 테이블을 이용할 수도 있다.
- [0170] 이때, 인코딩 장치는 룩업 테이블상에서 필터 정보를 지시하는 인덱스를 디코딩 장치로 전송할 수 있다. 전송되는 인덱스는 인코딩 장치에서 적용된 다운샘플링 필터의 정보를 지시하는 인덱스일 수도 있고, 디코딩 장치에서 적용할 업샘플링 필터의 정보를 지시하는 인덱스일 수도 있다.
- [0171] 표 2는 본 발명에 따라서 필터 정보를 전달하는데 이용되는 룩업 테이블의 일 예를 간단히 나타낸 것이다.

[0172] <표 2>

[0173]

필터 인덱스	내용(description)
00	8-탭 / 하프-펠(half-pel) 인터폴레이션 필터
01	4-탭 / 하프-펠 인터폴레이션(interpolation) 필터
10	8-탭 / 1/4, 3/4 페이즈 인터폴레이션 필터
11	4-탭 / 1/4, 3/4 페이즈 인터폴레이션 필터

[0174] 업샘플링/다운샘플링 필터 정보는 부호화/복호화 과정의 소정 레벨에서 저장/전송될 수 있다. 예컨대, 필터 정보는 시퀀스 파라미터 셋에서 전송될 수 있다. 이 경우, 동일 시퀀스 내에서는 별도의 시그널링이 없는 한 동일한 필터들이 적용될 수 있다. 또한, 필터 정보를 픽처 파라미터 셋에서 전송하여 별도의 시그널링이 없는 한 동일 픽처 내에서는 동일한 필터들이 적용되도록 할 수도 있다. 혹은 필터 정보를 슬라이스 헤더에 저장하여 전송함으로써 별도의 시그널링이 없는 한 동일한 슬라이스 내에서는 동일한 필터들이 적용되도록 할 수도 있다.

[0175] 표 3은 필터 정보를 시퀀스 파라미터 셋에 저장하여 전송하는 경우의 예를 간단히 나타낸 것이다.

[0176] <표 3>

seq_parameter_set_rbsp() {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
...	
adaptive_loop_filter_enabled_flag	u(1)
Interpolation_filter_indicator	u(2)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
temporal_id_nesting_flag	u(1)
rbsp_trailing_bits()	
}	

[0178] 표 3에서 Interpolation_filter_indicator는 사용할 인터폴레이션(보간) 필터의 종류를 지시한다.

[0179] 표 3에서는 시퀀스 파라미터 셋에서 필터 정보를 저장하는 경우의 신택스 구조를 나타냈으나, 이는 본 발명에 따른 일 예일뿐이며, Interpolation_filter_indicator는 상술한 바와 같이, 픽처 파라미터 셋 또는 슬라이스 헤더에서 전송될 수도 있다.

[0180] Interpolation_filter_indicator가 지시하는 필터의 종류는 필터의 특징을 나타내는 정보로서, 표 1에서와 같이 페이즈, 탭 수, 탭의 계수 등을 포함한다.

[0181] 즉, Interpolation_filter_indicator는 표 4와 같이, 룩업 테이블의 인덱스 값을 지시할 수 있다. 표 4는 리스케일링에 적용되는 필터 정보를 Interpolation_filter_indicator를 통해 지시하는데 이용되는 테이블의 일 예를 나타낸 것이다.

[0182] <표 4>

Interpolation_filter_indicator	내용(description)
00	8-탭 / 하프-펠(half-pel) 인터플레이션 필터
01	4-탭 / 하프-펠 인터플레이션(interpolation) 필터
10	8-탭 / 1/4, 3/4 페이즈 인터플레이션 필터
11	4-탭 / 1/4, 3/4 페이즈 인터플레이션 필터

[0184] 도 8은 본 발명에 따라서 Interpolation_filter_indicator를 이용하는 방법의 일 예로서, Interpolation_filter_indicator의 값이 10인 경우의 샘플링 방법을 간단히 설명하는 도면이다.

[0185] 도 8(a)는 1/2 페이즈 이동이 적용된 다운샘플링의 경우를 나타내며, 도 8(b)는 도 8(a)에서 다운샘플링된 샘플로부터 업샘플링을 수행하는 경우를 나타낸다.

[0186] 도 8(b)에서는 오리지널 샘플들과 업샘플링된 샘플들 사이의 페이즈를 일치시키기 위해, Interpolation_filter_indicator가 지시하는 대로 8-탭 / 1/4, 3/4 페이즈 인터플레이션 필터를 적용하여 업샘플링을 수행한다.

[0187] 다시 말하면, Interpolation_filter_indicator는 디코딩 장치가 적용할 업샘플링 필터를 지시하며, 예컨대, Interpolation_filter_indicator의 값이 10인 경우에는 페이즈를 일치시키기 위해 '8-탭 / 1/4, 3/4 페이즈 인터플레이션 필터'가 적용되어야 함을 지시한다. 동시에, 8-탭 / 1/4, 3/4 페이즈 인터플레이션 필터가 적용되어 페이즈를 일치시킨다는 것은 인코딩 장치에서 1/2 페이즈 이동이 적용된 다운샘플링이 수행되었음을 의미한다.

[0188] 지금까지 설명한 인터 레이어 인트라 예측은 도 1내지 도 4에서 설명한 인터 레이어 예측부(예컨대, 텍스처 예측부)에서 수행될 수 있다. 인터 레이어 인트라 예측에서는 참조 레이어 내 참조 블록의 텍스처를 인헨스먼트 레이어의 현재 블록에 대한 예측 값으로 사용할 수 있다. 이때, 참조 블록의 텍스처는 업샘플링에 의해 스케일링될 수 있다.

[0189] 인터 레이어 인트라 예측을 적용할 것인지는 CU의 분할 여부를 지시하는 플래그에 이어서 플래그 형태로 지시될 수 있다. 인터 레이어 인트라 예측에서

스케일링이 적용되는 경우에, 필터의 정보는 부호화되어 전달될 수 있다. 이때, 전달되는 정보는 상술한 바와 같다.

[0190]

[0191] 인터 레이어 움직임 예측

[0192] 인터 레이어 움직임 예측은 인터 레이어 인터 예측이라고도 불리며, 본 명세서에서는 발명의 이해를 돕기 위해, 필요에 따라서 인터 레이어 움직임 예측과 인터 레이어 인터 예측의 표현을 혼용할 수 있다.

[0193] 인터 레이어 움직임 예측에서는 참조 레이어(베이스 레이어)의 움직임 정보를 이용하여 현재 레이어(인헨스먼트 레이어)의 현재 블록에 대한 예측을 수행할 수 있다.

[0194] 인터 레이어 움직임 예측은 도 1 내지 도 4의 예측부 혹은 인터 레이어 예측부에서 수행될 수 있다. 이하, 설명의 편의를 위해, 예측부에서 인터 레이어 움직임 예측을 수행하는 것으로 설명한다.

[0195] 도 9는 다른 레이어를 참조하지 않고 레이어 내에서 인터 예측을 수행하는 경우(이하, '인터 예측'이라 함)에 이용되는 움직임 정보의 후보들에 대한 일 예를 간단히 나타낸 것이다.

[0196] 도 9에서 $A_0, A_1, B_0, B_1, B_2, COL$ 은 해당 블록을 지시할 수도 있고, 해당 블록의 움직임 정보를 지시할 수도 있다. 이때, 해당 블록의 움직임 정보는 움직임 벡터일 수도 있고, 움직임 벡터와 참조 픽처 인덱스일 수도 있다.

[0197] 여기서는 인터 예측의 방법에 관해 베이스 레이어의 경우를 일 예로서 설명하도록 한다.

[0198] 베이스 레이어에서의 인터 예측은 도 1 내지 도 4의 예측부 혹은 인터/인트라 예측부에서 수행될 수 있다. 이하, 설명의 편의를 위해, 예측부에서 인터 예측을 수행하는 것으로 설명한다.

[0199] 인터 예측의 모드로는 머지(merge) 모드, 스킵 모드 그리고 MVP(Motion Vector Predicot)를 이용하는 모드가 있다. MVP를 이용하는 모드를 설명의 편의를 위해 AMVP(advanced MVP) 모드라고 할 수 있다.

[0200] 머지 모드에서는 도 9에 도시된 주변 블록(neighboring block)들의 움직임 정보(이하, 움직임 정보 후보라 함)들 중에서 선택된 움직임 정보를 현재 블록의 움직임 정보로 이용할 수 있다. 선택되는 움직임 정보 후보를 지시하는 인코딩 장치로부터 디코딩 장치로 전송될 수 있다.

[0201] 스킵 모드에서는 머지 모드와 동일하게 선택된 움직임 정보 후보의 움직임 정보를 현재 블록의 움직임 정보로 사용하지만, 레지듀얼이 생성/전송되지 않는다.

[0202] 머지 모드 또는 스킵 모드가 적용되는 경우에, 예측부는 현재 블록 주변의 공간적 후보들인 A_0, A_1, B_0, B_1, B_2 에 대한 가용성을 판단할 수 있다. 가용성 판단은 소정의 순서로 진행될 수 있다. 예컨대, $A_1 \rightarrow B_1 \rightarrow B_0 \rightarrow A_1 \rightarrow B_2$ 의 순서로 진행될 수 있다.

- [0203] 이때, 각 후보들의 가용성 판단에는 이전 후보와의 동일성 판단이 포함될 수 있다. 예컨대, B_1 에 대하여, A_1 과 움직임 정보가 동일한지를 고려하여 가용성 판단을 할 수 있다. 구체적으로, A_1 이 가용하고 A_1 과 B_1 의 움직임 정보가 동일하면 B_1 은 가용하지 않은 것으로 판단할 수 있다.
- [0204] 같은 방법으로, B_0 는 B_1 과 움직임 정보가 동일한지를 고려하여 가용성 판단을 할 수 있으며, A_0 은 A_1 과 움직임 정보가 동일한지를 고려하여 가용성 판단을 할 수 있다.
- [0205] B_2 는 A_1 과 움직임 정보가 동일한지, B_1 과 움직임 정보가 동일한지를 모두 고려하여 가용성 판단을 할 수 있으며, 이때, 앞선 네 후보(A_0 , A_1 , B_0 , B_1)가 모두 가용한 경우에는 가용하지 않은 것으로 판단할 수도 있다.
- [0206] COL 후보를 사용하는 경우에는, 참조 픽처 리스트를 이용하여 COL 후보가 포함된 COL 픽처를 특정할 수 있다. 현재 블록과 동일 LCU 내에 있는 COL 블록에 대하여 소정 위치를 포함하는 예측 블록의 움직임 정보를 COL 후보로 할 수 있다. 이때, COL 후보의 움직임 벡터는 COL 픽처와 현재 픽처의 참조 픽처들을 고려하여 스케일링될 수 있으며, COL 후보의 참조 인덱스는 소정의 값(예컨대, 0)으로 설정될 수 있다.
- [0207] COL 후보를 포함하여 가용한 것(available)으로 판단된 후보들로, 가용성 판단 순서에 따라 머지 후보 리스트가 구성될 수 있다. 이때, 현재 블록의 슬라이스 타입이 B(즉, 양방향 예측이 적용되는 슬라이스)이고 현재 머지 후보 리스트에 포함된 후보의 개수가 최대 개수보다 적은 경우에는 머지 후보 리스트에 후보들(조합 양 예측 후보, combined bi-prediction candidates)이 추가될 수 있다.
- [0208] 상술한 바와 같이, 머지 후보 리스트가 구성된 후에도 머지 후보 리스트에 포함된 후보의 개수가 최대 개수보다 적은 경우에는 소정의 후보(예컨대, 제로 머지 후보)가 머지 후보 리스트에 추가될 수 있다.
- [0209] 예측부는 인코딩 장치로부터 전송된 정보(예컨대, 머지 인덱스 merge_idx)가 머지 후보 리스트 상에서 지시하는 후보의 움직임 정보를 현재 블록에 대한 움직임 정보로 이용하여 인터 예측을 수행할 수 있다. 예컨대, 예측부는 머지 인덱스에 의해 선택된 후보의 움직임 정보가 지시하는 샘플들을 현재 블록의 예측 블록으로 할 수 있다.
- [0210] 한편, AMVP 모드가 적용되는 경우에도, 예측부는 MVP 후보로 구성되는 AMVP 리스트를 구성할 수 있다.
- [0211] AMVP 모드에서, 예측부는 $A_0 \rightarrow A_1$ 으로 후보의 가용성을 판단하고, $B_0 \rightarrow B_1 \rightarrow B_2$ 으로 후보의 가용성을 판단한다.
- [0212] $A_0 \rightarrow A_1$ 의 순서로 후보의 가용성을 판단할 때, 예측부는 ① 가용한 후보로서 현재 블록과 동일한 참조 픽처를 가지는 후보가 있는 경우 해당 후보를 AMVP 리스트에 포함시킬 수 있다. ①을 만족하는 후보가 없는 경우에, 예측부는 ② 현재 픽처와 현재 픽처의 참조 픽처 사이의 POC(Picture Order Count) 차이 및 현재 픽처와 후보의 참조 픽처 사이 POC 차이에 기반하여, 가용한 것으로 먼저 검색된

- 후보의 움직임 벡터를 스케일링할 수 있다. 예측부는 스케일링된 움직임 벡터를 AMVP 리스트에 포함시킬 수 있다.
- [0213] $B_0 \rightarrow B_1 \rightarrow B_2$ 의 순서로 후보의 가용성을 판단할 때, 예측부는 ① 가용한 후보로서 현재 블록과 동일한 참조 픽처를 가지는 후보가 있는 경우 해당 후보를 AMVP 리스트에 포함시킨다. ①을 만족하는 후보가 없고 A_0 과 A_1 중에 가용한 후보가 없는 경우에, 예측부는 ② 현재 픽처와 현재 픽처의 참조 픽처 사이의 POC(Picture Order Count) 차이 및 현재 픽처와 후보의 참조 픽처 사이 POC 차이에 기반하여, 가용한 것으로 먼저 검색된 후보의 움직임 벡터를 스케일링할 수 있다. 예측부는 스케일링된 움직임 벡터를 AMVP 리스트에 포함시킬 수 있다.
- [0214] COL 후보(시간적 후보)를 사용하는 경우에는, 참조 픽처 리스트를 이용하여 COL 후보가 포함된 COL 픽처를 특정할 수 있다. 현재 블록과 동일 LCU 내에 있는 COL 블록에 대하여 소정 위치를 포함하는 예측 블록의 움직임 정보를 COL 후보로 할 수 있다. 이때, COL 후보의 움직임 벡터는 COL 픽처와 현재 픽처의 참조 픽처들을 고려하여 스케일링될 수 있다.
- [0215] $A_0 \rightarrow A_1$ 의 순서로 후보의 가용성을 판단하여 결정된 MVP 후보를 A라고 하고, $B_0 \rightarrow B_1 \rightarrow B_2$ 의 순서로 후보의 가용성을 판단하여 결정된 MVP 후보를 B라고 하며, 시간적 후보에 대한 가용성 판단에 의해 결정된 MVP 후보를 COL이라고 할 때, AMVP 리스트는 [A B COL]의 순서로 구성될 수 있다.
- [0216] 이때, 예측부는 A와 B가 동일한 경우에는 둘 중 하나를 AMVP 리스트로부터 삭제할 수 있다.
- [0217] 또한, 예측부는 A, B 및 COL이 모두 유효한 경우에는 AMVP 리스트 내 MVP 후보들의 개수를 2로 조정할 수 있다. 예컨대, 예측부는 A와 B로 AMVP 리스트를 구성하고, COL을 AMVP 리스트에서 제거할 수 있다.
- [0218] 예측부는 AMVP 리스트 내 후보들의 개수가 2보다 적은 경우에는 제로(0) 움직임 벡터를 후보로 추가할 수도 있다.
- [0219] 인코딩 장치는 AMVP 리스트 상에서 현재 블록에 대한 인터 예측에 사용할 MVP를 지시하는 MVP 인덱스, 움직임 벡터의 차분 mvd(motion vector difference)와 참조 픽처 리스트 상에서 현재 블록에 대한 참조 픽처를 지시하는 참조 인덱스를 디코딩 장치로 전송할 수 있다. 참조 픽처 리스트는 인터 예측에 사용될 수 있는 참조 픽처들의 리스트로서, 순방향 예측에 대한 L0과 역방향 예측에 대한 L1이 있다.
- [0220] 예측부는 MVP 인덱스가 지시하는 MVP와 mvd로부터 유도한 움직임 벡터와 참조 인덱스가 지시하는 참조 픽처를 기반으로 현재 블록에 대한 예측 블록을 생성할 수 있다.
- [0221] 머지 모드/스킵 모드 또는 AMVP 모드를 적용하여 예측 블록이 생성되면, 예측부는 예측 블록과 레지듀얼을 기반으로 현재 블록에 대한 복원 블록을 생성할 수 있다. 스킵 모드가 적용되는 경우에는 레지듀얼이 전송되지 않으므로, 예측부는 예측 블록을 복원 블록으로 이용할 수 있다.

- [0222] 지금까지는 인터 예측의 방법으로서 베이스 레이어의 경우를 예로 설명하였으나, 인헨스먼트 레이어에서도 다른 레이어의 정보를 이용하지 않고 인터 예측을 수행하는 경우에는 위에서 설명한 것과 동일한 방법으로 인터 예측을 수행할 수 있다.
- [0223] 베이스 레이어에 대하여 상술한 바와 같이 인터 예측이 수행되면, 인헨스먼트 레이어에서는 베이스 레이어의 움직임 정보를 이용하여 인터 레이어 움직임 예측(inter-layer motion prediction)을 수행할 수 있다.
- [0224] 인터 레이어 움직임 예측은 인코딩 장치와 디코딩 장치의 예측부에서 수행할 수 있다.
- [0225] 도 10은 본 발명에 따라서 인터 레이어 움직임 예측을 수행하는 방법의 일 예를 개략적으로 설명하는 순서도이다.
- [0226] 도 10을 참조하면, 인헨스먼트 레이어의 예측부는 참조 레이어의 움직임 정보를 유도할 수 있다(S1010). 예컨대, 인헨스먼트 레이어의 인터 레이어 예측부는 참조 레이어의 예측부로부터 전달된 정보를 기반으로 참조 레이어의 움직임 정보를 유도할 수 있다. 혹은 인헨스먼트 레이어의 예측부는 인코딩 장치로부터 전달된 정보를 기반으로 참조 레이어의 움직임 정보를 유도할 수도 있다.
- [0227] 이때, 인헨스먼트 레이어의 예측부는 유도된 참조 레이어의 움직임 정보, 예컨대 유도된 참조 레이어의 움직임 벡터를 스케일링할 수도 있다.
- [0228] 인헨스먼트 레이어의 예측부는 참조 레이어의 움직임 정보를 이용하여 현재 블록에 대한 인터 레이어 인터 예측을 수행할 수 있다(S1020). 예컨대, 인헨스먼트 레이어의 인터/인트라 예측부는 인터 레이어 예측부가 유도한 참조 레이어의 움직임 정보를 머지 모드/스킵 모드 혹은 AMVP 모드의 한 후보로서 이용하여 현재 블록에 대한 예측을 수행할 수 있다.
- [0229] 이하, 도면을 참조하여, 본 발명에 따른 인터 레이어 인트라 예측을 구체적으로 설명한다.
- [0230] **1. 참조 레이어의 움직임 정보 유도**
- [0231] 도 11은 본 발명에 따라서 참조 레이어의 움직임 정보를 유도하는 방법을 개략적으로 설명하는 도면이다. 도 11에서는 현재 레이어는 참조 레이어의 상위 레이어로서, 현재 레이어의 해상도가 참조 레이어의 해상도보다 높은 경우를 예로서 설명한다.
- [0232] 도 11을 참조하면, 현재 레이어의 PU(1110, 현재 PU)에 대응하는 참조 레이어의 PU(1100, 참조 PU)를 기반으로 현재 PU가 특정될 수 있다.
- [0233] 현재 PU를 특정하는 위치를 (xCurr, yCurr)이라고 하고, 현재 PU에 대응하는 참조 레이어의 위치, 예컨대 참조 PU를 특정하는 위치를 (xRef, yRef)라고 하자.
- [0234] 참조 레이어의 움직임 벡터로부터 유도할 인터 레이어 움직임 벡터를 mvIL이라고 하고, (xRef, yRef)에 의해 특정되는 참조 레이어의 움직임 벡터(예컨대, 참조 PU의 움직임 벡터)를 mvRL이라고 하자.

- [0235] 도 11에서 nPSW는 현재 PU(1110)의 너비(width)이고, nPSH는 현재 PU(1110)의 높이(height)이다.
- [0236] 예측부는 현재 PU를 특정하고, 현재 PU의 위치를 기반으로 참조 PU를 특정하여, 참조 PU의 움직임 정보(예컨대 움직임 벡터)를 유도할 수 있다.
- [0237] (1) 현재 PU를 특정하는 위치 (xCurr, yCurr) 결정
- [0238] 현재 PU를 특정하는 위치 (xCurr, yCurr)는 아래 ① ~ ⑫와 같은 후보들 중에 어느 하나로 결정될 수 있다.
- [0239] ① $LT = (xP, yP)$
- [0240] ② $RT = (xP + nPSW - 1, yP)$
- [0241] ③ $LB = (xP, yP + nPSH - 1)$
- [0242] ④ $RB = (xP + nPSW - 1, yP + nPSH - 1)$
- [0243] ⑤ $LT' = (xP - 1, yP - 1)$
- [0244] ⑥ $RT' = (xP + nPSW, yP - 1)$
- [0245] ⑦ $LB' = (xP - 1, yP + nPSH)$
- [0246] ⑧ $RB' = (xP + nPSW, yP + nPSH)$
- [0247] ⑨ $C0 = (xP + (nPSW \gg 1) - 1, yP + (nPSH \gg 1) - 1)$
- [0248] ⑩ $C1 = (xP + (nPSW \gg 1), yP + (nPSH \gg 1) - 1)$
- [0249] ⑪ $C2 = (xP + (nPSW \gg 1) - 1, yP + (nPSH \gg 1))$
- [0250] ⑫ $C3 = (xP + (nPSW \gg 1), yP + (nPSH \gg 1))$
- [0251] 현재 PU를 특정하는 위치 (xCurr, yCurr)는 ① ~ ⑫ 중 어느 하나로 결정되어 고정적으로 사용될 수도 있고, 인코딩 장치에서 RDO를 통해 결정한 뒤 어느 위치를 (xCurr, yCurr)로 사용할 것인지를 시그널링할 수도 있다.
- [0252] 혹은 참조 레이어(베이스 레이어)에서 PU를 특정하는 위치에 대응하여, 동일한 위치를 현재 레이어(인헨스먼트 레이어)의 PU를 특정하는 위치로 결정할 수도 있다. 예컨대, 참조 레이어에서 PU 내 좌상단을 PU를 특정하는 위치로 이용하는 경우에는, 현재 레이어에서도 PU 내 좌상단 $LT = (xP, yP)$ 를 (xCurr, yCurr)로 결정하여 이용할 수도 있다.
- [0253] (2) 참조 레이어의 대상 위치 (xRef, yRef)
- [0254] 참조 레이어에서 움직임 벡터를 가져올 위치(참조 PU의 위치)는 현재 PU의 위치로부터 현재 레이어와 참조 레이어 간의 비에 따라서 결정될 수 있다.
- [0255] 수식 1은 본 발명에 따라서 참조 레이어에서 움직임 벡터를 가져올 위치를 결정하는 방법을 나타낸 것이다.
- [0256] <수식 1>
- [0257] $xRef = xCurr / scale$
- [0258] $yRef = yCurr / scale$
- [0259] 이때, 참조 레이어에 대한 현재 레이어의 비를 나타내는 scale은 두 레이어 해상도에 따라서 결정될 수 있다. 예컨대, 현재 레이어의 해상도가 참조 레이어의 해상도의 2 배라면, 적용되는 scale의 값은 2가 된다. 현재 레이어의

- 해상도와 참조 레이어의 해상도가 동일하다면, 적용되는 *scale*의 값은 1이 된다.
- [0260] 여기서서는 현재 레이어와 참조 레이어 사이의 해상도 비로서 *scale*의 값을 결정하였으나, 본 발명은 이에 한정되지 않는다. *scale*은 현재 레이어와 참조 레이어 사이에 적용되는 스케일러빌리티의 종류에 따라서 결정될 수도 있다. 예컨대, *scale*은 현재 레이어와 참조 레이어 사이의 픽처 크기 비, 프레임 레이트 비일 수도 있다.
- [0261] 예측부는 (*xRef*, *yRef*) 위치의 움직임 벡터, 즉 (*xRef*, *yRef*)를 커버하는 PU(참조 PU)의 움직임 벡터를 *mvRL*로서 유도할 수 있다.
- [0262] 또한, 예측부는 (*xRef*, *yRef*)를 커버하는 PU(참조 PU)의 참조 인덱스를 인터 레이어 움직임 예측에 사용할 참조 인덱스 *refIdxIL*로서 유도할 수 있다.
- [0263] 예측부는 *mvRL*을 스케일링하여 인터 레이어 움직임 예측(인터 레이어 인터 예측)에 사용할 움직임 벡터 *mvIL*를 유도할 수 있다.
- [0264] 수식 2는 본 발명에 따라서 *mvRL*을 스케일링하여 *mvIL*을 유도하는 방법을 나타낸 것이다.
- [0265] <수식 2>
- [0266] $mvIL = scale * mvRL$
- [0267] 수식 2에서 계수 *scale*은 수식 1에서와 마찬가지로 참조 레이어에 대한 현재 레이어의 비를 나타낸다. 예컨대, 현재 레이어의 해상도가 참조 레이어의 해상도의 2 배라면, 적용되는 *scale*의 값은 2가 된다.
- [0268] 현재 레이어의 해상도와 참조 레이어의 해상도가 동일하다면, 적용되는 *scale*의 값은 1이 되며, 예측부는 *mvRL*을 *mvIL*로 이용할 수 있다.
- [0269] 2. 참조 레이어로부터 유도한 움직임 정보를 이용한 인터 레이어 인터 예측
- [0270] 예측부는 참조 레이어로부터 유도한 움직임 정보를 이용하여 현재 레이어(인헨스먼트 레이어)의 현재 블록에 대한 인터 레이어 인터 예측을 수행할 수 있다. 참조 레이어로부터 유도한 움직임 정보는 움직임 벡터 *mvIL*과 참조 인덱스 *refIdxIL*을 포함한다.
- [0271] 예컨대, 머지 모드/스킵 모드가 적용되는 경우에, 예측부는 *mvIL*과 *refIdxIL*을 머지 후보로서 현재 블록에 대한 머지 후보 리스트에 추가할 수 있다.
- [0272] 또한, AMVP 모드가 적용되는 경우에, 예측부는 *mvIL*을 MVP 후보로서 현재 블록에 대한 AMVP 리스트에 추가할 수 있다.
- [0273] (1) 머지 모드가 적용되는 경우
- [0274] 표 5는, 앞서 설명한 바와 같이, 인터 예측으로서 다른 레이어를 참조하지 않고 레이어 내에서 머지 모드가 적용되는 경우에, 구성되는 머지 후보 리스트의 일 예를 나타낸 것이다.
- [0275] <표 5>
- [0276]

인터 예측의 머지 후보 리스트

- A_1 이 유효한 경우, A_1
- B_1 이 유효한 경우, B_1
- B_0 이 유효한 경우, B_0
- A_0 이 유효한 경우, A_0
- 상기 A_1, B_1, B_0, A_0 의 전부가 유효하지는 않고 B_2 가 유효한 경우, B_2
- COL
- 최대 후보 개수를 넘지 않는 경우, 조합 양 예측(combined bi prediction) 후보
- 최대 후보 개수를 넘지 않는 경우, 제로 움직임 벡터 후보

- [0277] 표 5에서 $A_1, B_1, B_0, A_0, B_2, COL$ 은 도 9의 $A_1, B_1, B_0, A_0, B_2, COL$ 와 동일하다. 또한, 표 5에서 인터 예측의 머지 후보 리스트를 구성하는 방법은 앞서 도 9를 참조하여 설명한 인터 예측의 머지 모드에서와 동일하다.
- [0278] 표 5에서 제일 위에 위치하는 후보에는 가장 작은 값의 인덱스가 할당될 수 있고, 가장 아래에 위치하는 후보에는 가장 큰 값의 인덱스가 할당될 수 있다.
- [0279] 반면, 인터 레이어 움직임 예측이 적용되는 경우에, 예측부는 표 5와 달리 참조 레이어로부터 유도한 움직임 정보를 포함하는 머지 후보 리스트를 구성할 수 있다. 여기서 설명의 편의를 위해, 참조 레이어로부터 유도한 움직임 정보를 참조 레이어 후보(reference layer candidate) REF라고 한다.
- [0280] REF는 mvIL과 refIdxIL을 포함한다.
- [0281] 표 6은 본 발명에 따라서 인터 레이어 움직임 예측의 머지 모드가 적용되는 경우에 예측부에 의해 구성되는 머지 후보 리스트의 일 예를 나타낸 것이다. 표 6은 본 발명에 따라서 참조 레이어 후보가 머지 후보 리스트에 추가되는 순서를 설명한다.
- [0282] <표 6>
- [0283]

인터 레이어 움직임 예측의 머지 후보 리스트

- ㉑
- A_1 이 유효한 경우, A_1
- B_1 이 유효한 경우, B_1
- ㉒
- B_0 이 유효한 경우, B_0
- A_0 이 유효한 경우, A_0
- 상기 A_1, B_1, B_0, A_0 의 전부가 유효하지는 않고 B_2 가 유효한 경우, B_2
- ㉓
- COL
- ㉔
- 최대 후보 개수를 넘지 않는 경우, 조합 양 예측(combined bi prediction) 후보
- ㉕
- 최대 후보 개수를 넘지 않는 경우, 제로 움직임 벡터 후보
- ㉖

[0284] 표 6에서 $A_1, B_1, B_0, A_0, B_2, COL$ 은 도 9의 $A_1, B_1, B_0, A_0, B_2, COL$ 와 동일하다. 또한, 표 6에서 $A_1, B_1, B_0, A_0, B_2, COL$ 로 머지 후보 리스트를 구성하는 방법은 앞서 도 9를 참조하여 설명한 바와 동일하다.

[0285] 표 6에서 가장 위에 위치하는 후보에는 가장 작은 값의 인덱스가 할당될 수 있고, 가장 아래에 위치하는 후보에는 가장 큰 값의 인덱스가 할당될 수 있다.

[0286] 다만, 예측부는 도 9에서 설명한 바와 같이, A_1, B_1, B_0, A_0, B_2 의 유효성을 판단할 때마다, 앞선 후보와의 동일성을 고려할 수 있다.

[0287] 또한, 예측부는 A_1, B_1, B_0, A_0, B_2 간의 동일성을 배제하고 리스트를 구성한 뒤, $A_1, B_1, B_0, A_0, B_2, COL$ 간의 동일성을 한번에 판단할 수 있다. 이 경우, 동일한 후보 중 하나를 남기는 작업은 COL의 유효성을 판단한 후에 수행될 수 있다.

[0288] 인터 레이어 움직임 예측의 머지 모드가 적용되는 경우에, 예측부는 표 6의 머지 후보 리스트에서 ㉑ ~ ㉖ 중의 어느 한 순서에 REF를 포함시킬 수 있다.

[0289] 예컨대, 예측부는 REF에 가장 작은 인덱스(예컨대, 0)을 할당하여 머지 후보 리스트의 ㉑ 순서에 포함시킬 수 있다. 즉, 예측부는 머지 후보 리스트에 REF를 가장 먼저 추가할 수 있다.

[0290] 또한, 예측부는 REF에 가장 큰 인덱스(예컨대, 머지 후보 리스트에 포함될 수

- 있는 최대 후보 개수 - 1)를 할당하여 머지 후보 리스트의 ㉑ 순서에 포함시킬 수도 있다. 즉, 예측부는 머지 후보 리스트에 REF를 가장 늦게 추가할 수도 있다.
- [0291] 예측부는 공간적 후보들 다음의 순서인 ㉒에 REF를 추가할 수도 있다. 즉, 예측부는 공간적 후보들의 유효성을 판단한 후에 REF를 머지 후보 리스트에 추가할 수도 있다.
- [0292] 또한, 예측부는 현재 레이어 내의 후보들 중 단방향 예측에 대한 후보들 다음 그리고 조합 양방향 후보 전의 순서인 ㉓에 REF를 추가할 수도 있다. 즉, 예측부는 현재 레이어 내의 후보들 중 단방향 예측의 후보들에 대한 유효성을 판단하고, 조합 양방향 후보를 추가하기 전에 REF를 머지 후보 리스트에 추가할 수도 있다.
- [0293] 예측부는 현재 레이어 내의 후보들을 모두 고려한 뒤의 순서인 ㉔에 REF를 추가할 수도 있다. 즉, 예측부는 현재 레이어 내의 후보들에 대한 유효성을 모두 검토한 후에 REF를 머지 후보 리스트에 추가할 수도 있다.
- [0294] 아울러, 예측부는 현재 블록 좌측의 후보와 상측의 후보를 하나씩 고려한 후의 순서인 ㉕에 REF를 추가할 수도 있다. 즉, 예측부는 현재 블록 좌측의 후보와 상측의 후보의 유효성을 하나씩 검토한 후에 REF를 머지 후보 리스트에 포함시킬 수도 있다.
- [0295] 표 6의 인터 레이어 움직임 예측의 머지 후보 리스트에 포함되는 후보의 개수, 즉 최대 후보 개수는 표 5와 동일할 수 있다. 이 경우, 최대 후보 개수를 맞추기 위해 REF가 포함되는 위치에 따라서, REF보다 후순위에 있는 현재 레이어의 후보들이 머지 후보 리스트에서 제외될 수도 있다. 또한, REF보다 선순위에 있는 현재 레이어의 후보들로 최대 후보 개수가 충족되는 경우에는 REF가 머지 후보 리스트에서 제외될 수도 있다.
- [0296] 표 6의 인터 레이어 움직임 예측의 머지 후보 리스트에 포함되는 후보의 개수, 즉 최대 후보 개수는 표 5와 상이할 수도 있다. 예컨대, 표 6에서 최대 후보 개수는 REF를 고려하여 표 5의 경우보다 하나 더 많을 수 있다. 이 경우에는 현재 레이어의 후보들로 머지 후보 리스트가 구성된 뒤, REF를 소정의 위치 혹은 소정의 순서에 포함시킴으로써 머지 후보 리스트를 완성한다고 볼 수도 있다. 이때, 머지 후보 리스트 내에서 REF의 위치 혹은 추가되는 순서는 미리 정해져 있을 수도 있고, 인코딩 장치로부터 지시되거나 디코딩 장치에 의해 유도될 수 있다.
- [0297] 머지 후보 리스트 내의 후보들 중에서 어떤 후보를 이용하여 머지 모드를 수행할 것인지는 인코딩 장치로부터 지시될 수 있다. 예컨대, 예측부는 인코딩 장치로부터 수신한 정보(예컨대, 머지 인덱스 `merge_idx`)가 표 6과 같은 머지 후보 리스트 상에서 지시하는 후보를 선택하고, 선택된 움직임 정보가 지시하는 블록을 기반으로 현재 블록에 대한 예측 블록을 생성할 수 있다.
- [0298] (2) MVP (AMVP) 모드가 적용되는 경우
- [0299] 표 7은, 앞서 설명한 바와 같이, 인터 예측으로서 다른 레이어를 참조하지 않고

현재 레이어 내에서 MVP 모드가 적용되는 경우에, 구성되는 후보 리스트의 일 예를 나타낸 것이다. 본 명세서에서는 MVP를 이용하는 인터 예측의 모드를 AMVP 모드라고 하며, 이때 이용되는 후보 MVP들로 구성되는 리스트를 AMVP 리스트라고 한다.

[0300] <표 7>

[0301]

인터 예측의 AMVP 리스트
<ul style="list-style-type: none"> - A_0 및 A_1으로부터 선택된 A - B_0, B_1 및 B_2로부터 선택된 B - A 또는 B가 유효하지 않은 경우, COL - 최대 후보 개수를 넘지 않는 경우, 제로 움직임 벡터 후보

[0302] 표 7에서 A_1 , B_1 , B_0 , A_0 , B_2 , COL 그리고 A와 B는 도 9의 A_1 , B_1 , B_0 , A_0 , B_2 , COL 및 도 9를 이용하여 설명한 MVP 모드에서의 A 및 B와 동일하다. 또한, 표 7에서 인터 예측의 MVP 후보 리스트를 구성하는 방법은 앞서 도 9를 참조하여 설명한 인터 예측의 MVP 모드에서와 동일하다.

[0303] 표 7에서 제일 위에 위치하는 후보에는 가장 작은 값의 인덱스가 할당될 수 있고, 가장 아래에 위치하는 후보에는 가장 큰 값의 인덱스가 할당될 수 있다.

[0304] 반면, 인터 레이어 움직임 예측이 적용되는 경우에, 예측부는 표 7과 달리 참조 레이어로부터 유도한 움직임 정보를 포함하는 MVP 후보 리스트를 구성할 수 있다. 여기서는 참조 레이어로부터 유도한 움직임 벡터를 참조 레이어 후보(reference layer candidate) REF라고 한다.

[0305] REF는 mvIL을 포함한다.

[0306] 표 8은 본 발명에 따라서 인터 레이어 움직임 예측의 MVP 모드가 적용되는 경우에 예측부에 의해 구성되는 AMVP 리스트의 일 예를 나타낸 것이다. 표 8은 본 발명에 따라서 참조 레이어 후보가 AMVP 리스트에 추가되는 순서를 설명한다.

[0307] <표 8>

[0308]

인터 레이어 움직임 예측의 AMVP 리스트

- ㉑
- A_0 및 A_1 으로부터 선택된 A
- ㉒
- B_0 , B_1 및 B_2 로부터 선택된 B
- ㉓
- COL
- ㉔
- 최대 후보 개수를 넘지 않는 경우, 제로 움직임 벡터 후보
- ㉕

- [0309] 표 8에서 A_1 , B_1 , B_0 , A_0 , B_2 , COL 그리고 A와 B는 도 9의 A_1 , B_1 , B_0 , A_0 , B_2 , COL 및 도 9를 이용하여 설명한 MVP 모드에서의 A 및 B와 동일하다. 또한, 표 8에서 A, B, COL로 AMVP 리스트를 구성하는 방법은 앞서 도 9를 참조하여 설명한 바와 동일하다.
- [0310] 표 8에서 가장 위에 위치하는 후보에는 가장 작은 값의 인덱스가 할당될 수 있고, 가장 아래에 위치하는 후보에는 가장 큰 값의 인덱스가 할당될 수 있다.
- [0311] 예측부는 도 9에서 설명한 바와 같이, A, B 사이의 동일성을 판단하여 A와 B가 동일한 경우에는 A와 B 중 하나를 AMVP 리스트에서 제외할 수 있다. 이 과정은 B의 유효성을 판단할 때 수행될 수도 있고, A와 B를 판단한 후에 수행될 수도 있으며, COL의 유효성을 판단하는 경우 혹은 COL의 유효성을 판단한 후에 수행될 수도 있다.
- [0312] 인터 레이어 움직임 예측의 MVP 모드가 적용되는 경우에, 예측부는 표 8의 AMVP 리스트에서 ㉑ ~ ㉕ 중의 어느 한 순서에 REF를 포함시킬 수 있다.
- [0313] 예컨대, 예측부는 REF에 가장 작은 인덱스(예컨대, 0)을 할당하여 ㉑의 순서에 포함시킬 수 있다. 즉, 예측부는 AMVP 리스트에 REF를 가장 먼저 추가할 수 있다.
- [0314] 또한, 예측부는 REF에 가장 큰 인덱스(예컨대, AMVP 리스트에 포함될 수 있는 최대 후보 개수 - 1)를 할당하여 ㉕의 순서에 추가시킬 수도 있다. 즉, 예측부는 AMVP 리스트의 마지막에 REF를 추가할 수도 있다.
- [0315] 또한, 예측부는 현재 레이어 내의 후보들을 모두 고려한 뒤의 순서인 ㉑에 REF를 포함시킬 수도 있다.

- [0316] 예측부는 공간적 후보들 다음의 순서인 ㉔에 REF를 추가시킬 수도 있다. 또한, 예측부는 현재 블록 좌측의 후보를 고려한 후, 상측의 후보를 고려하기 전의 위치인 ㉕에 REF를 포함시킬 수도 있다.
- [0317] 표 8의 인터 레이어 움직임 예측의 AMVP 리스트에 포함되는 후보의 개수, 즉 최대 후보 개수는 표 7과 동일할 수 있다. 이 경우, 최대 후보 개수를 맞추기 위해 REF가 포함되는 위치에 따라서, REF보다 후순위에 있는 현재 레이어의 후보들이 AMVP 리스트에서 제외될 수도 있다. 또한, REF보다 선순위에 있는 현재 레이어의 후보들로 최대 후보 개수가 충족되는 경우에는 REF가 AMVP 리스트에서 제외될 수도 있다.
- [0318] 예컨대, 최대 후보 개수가 2인 경우, ㉔ 위치의 REF는 AMVP 리스트에서 제외될 수도 있다.
- [0319] 표 8의 인터 레이어 움직임 예측의 AMVP 리스트에 포함되는 후보의 개수, 즉 최대 후보 개수는 표 7과 상이할 수도 있다. 예컨대, 표 8에서 최대 후보 개수는 REF를 고려하여 표 7의 경우(예컨대, 2)보다 하나 더 많을 수 있다. 이 경우에는 현재 레이어의 후보들로 AMVP 리스트가 구성된 뒤, REF를 소정의 위치에 포함시킴으로써 AMVP 리스트를 완성한다고 볼 수도 있다. 이때, AMVP 리스트 내에서 REF의 위치는 미리 정해져 있을 수도 있고, 인코딩 장치로부터 지시되거나 디코딩 장치에 의해 유도될 수 있다.
- [0320] AMVP 리스트 내의 후보들 중에서 어떤 후보를 이용하여 MVP 모드를 수행할 것인지는 인코딩 장치로부터 지시될 수 있다. 예컨대, 예측부는 인코딩 장치로부터 수신한 정보가 표 6과 같은 AMVP리스트 상에서 지시하는 후보를 선택하고, 선택된 움직임 벡터와 인코딩 장치로부터 수신한 mvd를 이용하여 현재 블록에 대한 움직임 벡터를 유도할 수 있다. 예측부는 유도한 움직임 벡터와 인코딩 장치로부터 수신한 참조 인덱스가 지시하는 참조 픽처를 기반으로 현재 블록에 대한 예측 블록을 생성할 수 있다.
- [0321] 한편, 소정의 조건을 만족하면, 예측부는 AMVP 리스트에 REF를 추가하기 전에 mvIL을 스케일링할 수 있다.
- [0322] 예컨대, 현재 PU의 참조 인덱스와 참조 PU의 참조 인덱스(refIdxIL)가 상이한 경우에는 mvIL을 스케일링할 수 있다. 다시 말하면, 현재 PU의 참조 인덱스가 지시하는 픽처(현재 PU의 참조 픽처)의 POC와 참조 PU의 참조 인덱스가 지시하는 픽처(참조 PU의 참조 픽처)의 POC가 상이한 경우에는, mvIL을 스케일링할 수 있다.
- [0323] 도 12는 본 발명에 따라서 mvIL을 스케일링하는 방법을 개략적으로 설명하는 도면이다.
- [0324] 도 12를 참조하면, 현재 레이어에서 현재 픽처의 POC와 현재 PU(1200)의 참조 픽처(1220)의 POC 간 차이를 tb라고 하고, 참조 레이어에서 현재 픽처의 POC와 참조 PU(1210)의 참조 픽처(1230)의 POC 간 차이를 td라고 한다.
- [0325] 현재 PU(1200)의 참조 픽처(1220)에 대한 POC인 pocRef와 참조 PU(1210)의

참조 픽처(1230)에 대한 POC인 pocRefLayer가 상이한 경우에, 예측부는 REF 즉, mvIL을 스케일링하여 AMVP에 포함시킬 수 있다.

[0326] mvIL은 수식 3과 같은 방법으로 스케일링될 수 있다. 스케일링된 mvIL을 mvIL'이라고 한다.

[0327] <수식 3>

[0328] $tx = (16384 + (\text{Abs}(td) \gg 1)) / td$

[0329] $\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6)$

[0330] $mvIL' = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvIL) * ((\text{Abs}(\text{DistScaleFactor} * mvIL) + 127) \gg 8))$

[0331] $td = \text{Clip3}(-128, 127, \text{pocCurr} - \text{pocRefLayer})$

[0332] $tb = \text{Clip3}(-128, 127, \text{pocCurr} - \text{pocRef})$

[0333] 수식 3에서, pocCurr은 현재 픽처의 POC를 나타내며, pocRef는 현재 PU의 참조 인덱스에 의해 지시되는 픽처의 POC이고, pocRefLayer는 참조 PU의 참조 인덱스 즉, (xRef, yRef)의 참조 인덱스에 의해 지시되는 픽처의 POC이다.

[0334] 또한, Abs(x)는 x가 0보다 작은 경우에는 -x가 되고, x가 0보다 크거나 같은 경우에는 x가 된다. Clip3(x,y,c)는 z가 x보다 작은 경우엔 x가 되고, z가 y보다 큰 경우엔 y가 되며, 그 외의 경우에는 z가 된다.

[0335] 예측부는 pocRef와 pocRefLayer가 상이한 경우에, 수식 3과 같이 각 레이어에서 참조 픽처까지의 거리를 기반으로 인터 레이어 움직임 벡터를 스케일링하고, 스케일링된 움직임 벡터 후보(즉, 스케일링된 mvIL 혹은 스케일링된 REF)를 AMVP 리스트에 추가할 수 있다.

[0336] 이때, 예측부는 REF(즉 mvIL) 대신 스케일링된 REF(즉, 스케일링된 mvIL)을 AMVP 리스트에 포함시켜서, 앞서 설명한 바와 동일한 방식으로 AMVP 모드를 적용한 인터 레이어 움직임 예측을 수행할 수 있다.

[0337] 인터 레이어 움직임 예측을 적용하는 경우에, CU 레벨의 관점에서 보면, 베이스 레이어의 블록 크기와 무관하게 모든 CU 파티션을 그대로 사용할 수 있다. 또한, PU 레벨에서 보면, 인코딩 장치가 인터 예측과 인터 레이어 움직임 예측에 대한 RDO를 수행함으로써 최적의 예측 모드를 적용할 수 있다.

[0338]

[0339] 인터 레이어 신택스 예측

[0340] 인터 레이어 신택스 예측에서는 참조 레이어의 신택스 정보를 이용하여 현재 블록의 텍스처를 예측하거나 생성한다. 이때, 현재 블록의 예측에 이용하는 참조 레이어의 신택스 정보는 인트라 예측 모드에 관한 정보, 움직임 정보 등일 수 있다.

[0341] 예컨대, 참조 레이어가 P 슬라이스나 B 슬라이스이지만, 슬라이스 내에 참조 블록은 인트라 예측 모드가 적용된 블록일 수 있다. 이 경우에는 참조 레이어의 신택스 정보 중 인트라 모드를 이용하여 현재 레이어의 텍스처를 생성/예측하는 인터 레이어 예측을 수행할 수 있다. 구체적으로, 참조 레이어가 P 슬라이스나 B

슬라이스이지만, 슬라이스 내에 참조 블록은 인트라 예측 모드가 적용된 블록인 경우에, 인터 레이어 신택스 예측을 적용하면, 현재 블록에 대하여 (1) 참조 블록의 인트라 예측 모드로 (2) 현재 레이어에서 현재 블록 주변의 참조 픽셀을 이용하여, 인트라 예측을 수행할 수 있다.

- [0342] 참조 레이어가 P 슬라이스나 B 슬라이스이고 슬라이스 내의 참조 블록이 인터 예측 모드가 적용된 블록인 경우에는, 앞서 설명한 바와 같이 인터 레이어 움직임 예측과 동일한 방식으로, 참조 레이어의 신택스 정보 중 움직임 정보를 스케일링하여 현재 레이어에 대한 텍스처를 생성/예측하는 인터 레이어 예측을 수행할 수 있다.
- [0343] 이처럼, 인터 레이어 신택스 예측은 인터 레이어 움직임 예측과 인터 레이어 텍스처 예측을 혼용하는 방식으로 볼 수 있다.
- [0344] 인터 레이어 움직임 예측은 앞서 설명한 바와 같이, 참조 레이어의 움직임 정보를 이용하여 인헨스먼트 레이어에서 예측 신호(예측 블록)를 생성하는 예측 방법이다.
- [0345] 이때, 참조 레이어의 움직임 정보는 레이어 간 해상도의 변화에 따라서 스케일링될 수 있다.
- [0346] 참조 레이어에서 인트라 예측 모드가 적용되어서 움직임 정보가 없는 블록이 참조 블록인 경우에는, 상술한 바와 같이 참조 레이어의 인트라 예측 모드를 가져와 인헨스먼트 레이어의 주변 화소로부터 현재 블록을 예측하여 예측 신호를 생성할 수 있다.
- [0347] 인터 레이어 신택스 예측 혹은 인터 레이어 움직임 예측의 경우에, 루마 성분에 대한 움직임 정보는 참조 레이어로부터 4x4 블록 단위로 가져올 수 있다. 이 경우, 크로마 성분에 대한 움직임 정보로서 루마 성분에 대한 움직임 정보를 사용할 수도 있다.
- [0348] 인터 레이어 신택스 예측을 적용하면, PU 레벨에서 봤을 때, 새로운 PU 모드를 생성하지 않고, 상위 레벨에서 어떤 예측 모드를 적용할 것인지를 시그널링함으로써, 참조 레이어 내 참조 블록에 적용된 예측 모드에 따라서 인터 레이어 인트라 예측과 인터 레이어 움직임 예측을 적응적으로 적용할 수 있다.
- [0349] 인터 레이어 신택스 예측을 적용할 것인지를 지시하는 플래그는 CU의 분할을 지시하는 플래그(예컨대 CU_split_flag) 다음에 플래그 형태로 전송될 수도 있다.
- [0350] 도 13은 본 발명에 따라서 인터 레이어 신택스 예측을 수행하는 방법의 일 예를 간단히 설명하는 도면이다.
- [0351] 도 13을 참조하면, 참조 레이어(1300)의 참조 블록(1310)에 인트라 블록과 인터 블록이 존재한다.
- [0352] 도 13의 경우에 인터 레이어 신택스 예측을 적용하면, 참조 블록(1310)을 현재 레이어에 맞춰 업샘플링한 픽처(1320)를 구성하여 인터 레이어 신택스 예측을 적용해 복원 블록을 생성할 수 있다(1330).
- [0353] 이때, 참조 레이어에서 인트라 예측 모드가 적용된 블록(intra)으로부터는

인트라 예측 모드를 참조하고, 인터 예측 모드가 적용된 블록(MV)으로부터는 움직임 정보를 참조하여 인터 레이어 신택스 예측을 수행할 수 있다.

[0354] 현재 레이어의 대상 픽처에 대하여 인터 레이어 신택스 예측은 다른 예측 모드들과 혼용될 수도 있으며, 도 13의 현재 픽처(1340)에서 빗금 영역은 다른 예측 모드가 적용될 수 있는 영역을 나타낸다.

[0355]

[0356] 인터 레이어 레지듀얼 예측

[0357] 인터 레이어 레지듀얼 예측은 참조 레이어의 레지듀얼 신호를 이용하여 인헨스먼트 레이어의 레지듀얼 예측 영상을 생성하고 인헨스먼트 레이어에서는 레지듀얼 예측 영상을 참조하여 레지듀얼 영상을 부호화/복호화하는 방법이다.

[0358] 인터 레이어 레지듀얼 예측은 도 1 내지 도 4의 예측부 혹은 인터 레이어 예측부에서 수행될 수 있다. 이하, 설명의 편의를 위해, 예측부에서 인터 레이어 레지듀얼 예측을 수행하는 것으로 설명한다.

[0359] 예측부는 참조 레이어의 레지듀얼 신호를 레이어 간 해상도의 차이 혹은 비(ratio)에 따라서 스케일링하여 인헨스먼트 레이어에 대한 레지듀얼 예측 영상을 생성할 수 있다.

[0360] 인코딩 장치에서는 인터 레이어 신택스 예측 및 인터 레이어 인트라 예측과는 별도로 인터 레이어 레지듀얼 예측에 대한 RDO를 수행할 수도 있다. 만약 RDO 경쟁을 통해 인터 레이어 레지듀얼 예측을 적용하는 것으로 결정되면, 레지듀얼 신호를 예측 하기 위한 플래그, 즉 인터 레이어 레지듀얼 예측을 적용하는 것을 지시하는 플래그는 레지듀얼(변환 계수) 앞쪽에 CU 단위로 부호화되어 전송될 수 있다.

[0361] 도 14는 본 발명에 따라서 인터 레이어 레지듀얼 예측을 적용하는 방법을 개략적으로 설명한 도면이다.

[0362] 도 14를 참조하면, 인터 레이어 예측을 위해 참조 레이어의 레지듀얼 신호(1400)에서 참조되는 참조 블록 부분(1410)을 레이어 간 해상도 비에 따라서 스케일링하여 레지듀얼 예측 영상(1420)이 생성될 수 있다.

[0363] 레지듀얼 예측 영상(1420)을 기반으로 현재 블록에 대한 레지듀얼 영상(1430)이 복원될 수 있다. 이때, 빗금 처리된 영역은 인터 레이어 레지듀얼 예측이 적용되지 않는 영역을 나타낸다.

[0364] 인코딩 장치는 모든 예측 모드의 레지듀얼 신호에 대하여 인터 레이어 레지듀얼 예측을 적용할 것인지를 RDO를 기반으로 판단할 수 있다. 인터 레이어 레지듀얼 예측을 적용할 것인지를 지시하는 정보는 플래그 형태로 계수(예컨대, 레지듀얼의 양자화된 변환 계수)에 대한 정보보다 앞서 전송될 수 있다.

[0365]

[0366] 인터 레이어 예측을 위한 신택스 구조

[0367] 인코딩 장치는 지금까지 설명한 인터 레이어 예측을 위해 필요한 정보를 시그널링할 수 있다. 또한, 디코딩 장치는 시그널링된 정보를 수신하여 인터

레이어 예측을 수행할 수 있다.

[0368] 여기서, 지금까지 설명한 혹은 이후에 추가적으로 설명할 인터 레이어 예측의 수행 방법과 필요한 정보들의 일 예를 나타낸 신택스들에 대하여 설명한다.

[0369] 표 9는 본 발명에 따른 NAL 유닛 신택스의 일 예를 나타낸 것이다.

[0370] <표 9>

nal_unit(NumBytesInNALunit) {	Descriptor
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
if(nal_unit_type == 1 nal_unit_type == 4 nal_unit_type == 5 nal_unit_type == SVC_NAL)	
temporal_id	u(3)
output_flag	u(1)
reserved_one_4bits	u(4)
if(nal_unit_type == SVC_NAL)	
nal_unit_header_svc_extension()	
nalUnitHeaderBytes += 2	
}	
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte [NumBytesInRBSP++]	b(8)
rbsp_byte [NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte [NumBytesInRBSP++]	b(8)
}	
}	

[0372] 표 10은 본 발명에 따른 NAL 유닛 헤더 SVC 확장 신택스(NAL unit header SVC extension syntax)의 일 예를 나타낸 것이다.

[0373] <표 10>

nal_unit_header_svc_extension() {	Descriptor
idr_flag	u(1)
dependency_id	u(3)
quality_id	u(4)
}	

[0375] 표 11은 본 발명에 따른 시퀀스 파라미터 셋 RBSP 신택스의 일 예를 나타낸

것이다.

[0376] <표 11>

[0377]

	Descriptor
seq_parameter_set_rbsp() {	
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
[Ed. (BB): Not in HM, further discuss separate_colour_plane_flag]	
max_temporal_layers_minus1	u(3)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
[Ed. (BB): chroma bit depth present in HM software but not used further]	
pcm_enabled_flag	u(1)
if (pcm_enabled_flag) {	
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
}	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
max_num_ref_frames	ue(v)
num_reorder_frames	ue(v)
max_dec_frame_buffering	ue(v)
max_latency_increase	ue(v)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
if (pcm_enabled_flag) {	
log2_min_pcm_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_coding_block_size	ue(v)
}	
max_transform_hierarchy_depth_inter	ue(v)

[0378]

max_transform_hierarchy_depth_intra	ue(v)
scaling_list_enable_flag	
chroma_pred_from_luma_enabled_flag	u(1)
deblocking_filter_in_APS_enabled_flag	u(1)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
if (pcm_enabled_flag)	
pcm_loop_filter_disable_flag	u(1)
temporal_id_nesting_flag	u(1)
[Ed. (BB): x y padding syntax missing here, present in HM software]	
if (log2_min_coding_block_size_minus3 == 0)	
inter_4x4_enabled_flag	u(1)
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
if (num_tile_columns_minus1 != 0 num_tile_rows_minus1 != 0) {	
uniform_spacing_flag	u(1)
if (!uniform_spacing_flag) {	
for (i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for (i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
tile_boundary_independence_flag	u(1)
if (tile_boundary_independence_flag == 1)	
loop_filter_across_tile_flag	u(1)
}	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbbsp_trailing_bits()	
}	

[0379] 표 12는 본 발명에 따른 서브셋 시퀀스 파라미터 RBSP 선택스의 일 예를 나타낸 것이다.

[0380] <표 12>

[0381] subset_seq_parameter_set_rbsp() {	Descriptor
seq_parameter_set_rbsp()	
if(profile_idc == SVC_PROFILE) {	
seq_parameter_set_svc_extension()	
}	
}	

[0382] 표 13은 본 발명에 따른 시퀀스 파라미터 셋 SVC 확장 신택스(Sequence parameter set SVC extension syntax)의 일 예를 나타낸 것이다.

[0383] <표 13>

[0384]	seq_parameter_set_svc_extension() {	Descriptor
	singleloop_decoding_flag	u(1)
	}	

[0385] 표 14는 본 발명에 따른 픽처 파라미터 셋 RBSP 신택스의 일 예를 나타낸 것이다.

[0386] <표 14>

[0387]	pic_parameter_set_rbsp() {	Descriptor
	pic_parameter_set_id	ue(v)
	seq_parameter_set_id	ue(v)
	num_short_term_ref_pic_sets	ue(v)
	for(idx = 0; idx < num_short_term_ref_pic_sets; idx++)	
	short_term_ref_pic_set(idx)	
	long_term_ref_pics_present_flag	u(1)
	entropy_coding_synchro	u(v)
	cabac_istate_reset_flag	u(1)
	if(entropy_coding_synchro)	
	num_substreams_minus1	ue(v)
	num_temporal_layer_switching_point_flags	ue(v)
	for(i = 0; i < num_temporal_layer_switching_point_flags; i++)	
	temporal_layer_switching_point_flag[i]	u(1)
	num_ref_idx_l0_default_active_minus1	ue(v)
	num_ref_idx_l1_default_active_minus1	ue(v)
	[Ed. (BB): not present in HM software]	
	pic_init_qp_minus26	se(v)
	constrained_intra_pred_flag	u(1)
	enable_temporal_mvp_flag	u(1)
	slice_granularity	u(2)
	max_cu_qp_delta_depth	ue(v)
	chroma_cb_qp_offset	se(v)
	chroma_cr_qp_offset	se(v)
	weighted_pred_flag	u(1)
	weighted_bipred_idc	u(2)
	tile_info_present_flag	u(1)
	tile_control_present_flag	u(1)

[0388]

if(tile_info_present_flag == 1) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
if(num_tile_columns_minus1 != 0 num_tile_rows_minus1 != 0) {	
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
}	
}	
if(tile_control_present_flag) {	
if(num_tile_columns_minus1 != 0 num_tile_rows_minus1 != 0)	
{	
tile_boundary_independence_flag	u(1)
if(tile_boundary_independence_flag == 1)	
loop_filter_across_tile_flag	u(1)
}	
}	
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0389] 표 15는 본 발명에 따른 스케일링 리스트 데이터 신택스의 일 예를 나타낸 것이다.

[0390] <표 15>

[0391]

scaling_list_param() {	Descriptor
scaling_list_present_flag	u(1)
if(scaling_list_present_flag)	
for(SizeID = 0; SizeID < 4; SizeID++)	
for(MatrixID = 0; MatrixID < (SizeID == 3) ? 2 : 6; MatrixID++) {	
scaling_list_pred_mode_flag	u(1)
if(!scaling_list_pred_mode_flag)	
scaling_list_pred_matrix_id_delta	ue(v)
else	
scaling_list(ScalingList[SizeID][MatrixID][0], (1 << (4 + (sizeID << 1))))	
}	
}	
}	

[0392] 표 16은 본 발명에 따른 스케일링 리스트 선택스의 일 예를 나타낸 것이다.

[0393] <표 16>

scaling_list(ScalingList, coefNum) {	Descriptor
nextcoef = 8	u(1)
for(i=0; i < coefNum, i++) {	
scaling_list_delta_coef	se(v)
nextcoef = (nextcoef + scaling_list_delta_coef + 256) % 256	
ScalingList[i] = nextcoef	
}	
}	

[0395] 표 17은 본 발명에 따른 적응적 파라미터 셋 RBSP 선택스(Adaptation parameter set RBSP syntax)의 일 예를 나타낸 것이다.

[0396] <표 17>

[0397]

	Descriptor
aps_rbsp() {	
aps_id	ue(v)
aps_scaling_list_data_present_flag	u(1)
if(aps_scaling_list_data_present_flag)	
scaling_list_param()	
aps_deblocking_filter_flag	u(1)
if(aps_deblocking_filter_flag) {	
disable_deblocking_filter_flag	u(1)
if(!disable_deblocking_filter_flag) {	
beta_offset_div2	se(v)
tc_offset_div2	se(v)
}	
aps_sample_adaptive_offset_flag	u(1)
if(aps_sample_adaptive_offset_flag)	
sao_param()	
aps_adaptive_loop_filter_flag	u(1)
if(aps_adaptive_loop_filter_flag)	
alf_param()	
aps_extension_flag	u(1)
if(aps_extension_flag)	
while(more_rbsp_data())	
aps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0398] 표 18은 본 발명에 따른 추가적 인헨스먼트 정보 RBSP 선택스(Supplemental enhancement information RBSP syntax)의 일 예를 나타낸 것이다.

[0399] <표 18>

[0400]

	Descriptor
sei_rbsp() {	
do	
sei_message()	
while(more_rbsp_data())	
rbsp_trailing_bits()	
}	

[0401] 표 19는 본 발명에 따른 추가적 인헨스먼트 정보 메시지 선택스(Supplemental enhancement information message syntax)를 나타낸 것이다.

[0402] <표 19>

[0403]

	Descriptor
sei_message() {	
payloadType = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadType += 255	
}	
last_payload_type_byte	u(8)
payloadType += last_payload_type_byte	
payloadSize = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadSize += 255	
}	
last_payload_size_byte	u(8)
payloadSize += last_payload_size_byte	
sei_payload(payloadType, payloadSize)	
}	

[0404] 표 20은 본 발명에 따른 액세스 유닛 디리미터 RBSP 신택스(Access unit delimiter RBSP syntax)의 일 예를 나타낸 것이다.

[0405] <표 20>

	Descriptor
access_unit_delimiter_rbsp() {	
primary_pic_type	u(3)
rbsp_trailing_bits()	
}	

[0407] 표 21은 본 발명에 따른 필러(filler) 데이터 RBSP 신택스의 일 예를 나타낸 것이다.

[0408] <표 21>

	Descriptor
filler_data_rbsp() {	
while(next_bits(8) == 0xFF)	
ff_byte /* equal to 0xFF */	f(8)
rbsp_trailing_bits()	
}	

[0410] 표 22는 본 발명에 따른 슬라이스 레이어 RBSP 신택스의 일 예를 나타낸 것이다.

[0411] <표 22>

[0412]

slice_layer_rbsp() {	Descriptor
slice_header()	
slice_data()	
rbsp_slice_trailing_bits()	
}	

[0413] 표 23은 본 발명에 따른 슬라이스 레이어 확장 RBSP 신택스의 일 예를 나타낸 것이다.

[0414] <표 23>

slice_layer_extension_rbsp() {	Descriptor
slice_header_in_scalable_extension()	
slice_data_in_scalable_extension()	
}	

[0416] 표 24는 본 발명에 따른 RBSP 슬라이스 트레일링 비트 신택스(RBSP slice trailing bits syntax)의 일 예를 나타낸 것이다.

[0417] <표 24>

rbsp_slice_trailing_bits() {	Descriptor
rbsp_trailing_bits()	
while(more_rbsp_trailing_data())	
cabac_zero_word /* equal to 0x0000 */	f(16)
}	

[0419] 표 25는 본 발명에 따른 RBSP 트레일링 비트 신택스의 일 예를 나타낸 것이다.

[0420] <표 25>

rbsp_trailing_bits() {	Descriptor
rbsp_stop_one_bit /* equal to 1 */	f(1)
while(!byte_aligned())	
rbsp_alignment_zero_bit /* equal to 0 */	f(1)
}	

[0422] 표 26은 본 발명에 따른 RBSP 바이트 정렬 신택스(RBSP byte alignment syntax)의 일 예를 나타낸 것이다.

[0423] <표 26>

byte_align() {	Descriptor
while(!byte_aligned())	
bit_equal_to_one	f(1)
}	

[0425] 표 27은 본 발명에 따른 슬라이스 헤더의 일 예를 나타낸 것이다.

[0426] <표 27>

[0427]

	Descriptor
slice_header() {	
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag == 0)	
slice_address	u(v)
slice_type	ue(v)
entropy_slice_flag	u(1)
if(!entropy_slice_flag) {	
pic_parameter_set_id	ue(v)
if(IdrPicFlag) {	
idr_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
} else {	
pic_order_cnt_lsb	u(v)
short_term_ref_pic_set_pps_flag	u(1)
if(!short_term_ref_pic_set_pps_flag)	
short_term_ref_pic_set(num_short_term_ref_pic_sets)	
} else	
short_term_ref_pic_set_idx	u(v)
if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i = 0; i < num_long_term_pics; i++) {	
delta_poc_lsb_lt_minus1[i]	ue(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
}	
}	
}	
if(scaling_list_enable_flag deblocking_filter_in_APS_enabled_flag sample_adaptive_offset_enabled_flag adaptive_loop_filter_enabled_flag) {	
if(sample_adaptive_offset_enabled_flag)	
slice_sample_adaptive_offset_flag	u(1)
if(adaptive_loop_filter_enabled_flag)	
slice_adaptive_loop_filter_flag	u(1)
aps_id	ue(v)
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	

[0428]

num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
ref_pic_list_modification()	
ref_pic_list_combination()	
}	
if(slice_type != I)	
cabac_init_idc	ue(v)
if(!entropy_slice_flag) {	
slice_qp_delta	se(v)
inherit_dbl_params_from_APS_flag	u(1)
if (!inherit_dbl_params_from_APS_flag) {	
disable_deblocking_filter_flag	u(1)
if (!disable_deblocking_filter_flag) {	
beta_offset_div2	se(v)
tc_offset_div2	se(v)
}	
}	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
if((weighted_pred_flag && slice_type == P) (weighted_bipred_idc == 1 && slice_type == B))	
pred_weight_table()	
}	
if(slice_type == P slice_type == B)	
S_minus_max_num_merge_cand	ue(v)
if(adaptive_loop_filter_enabled_flag && aps_adaptive_loop_filter_flag)	
alf_cu_control_param()	
for(i = 0; i < num_substreams_minus1 + 1; i++) {	
substream_length_mode	u(2)
substream_length[i]	u(v)
}	
}	

[0429] 표 28은 본 발명에 따른, 스케일러블 확장에서의 슬라이스 헤더 선택스(Slice header in scalable extension syntax)의 일 예를 나타낸 것이다.

[0430] <표 28>

[0431]

	Descriptor
slice_header_in_scalable_extension() {	
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag == 0)	
slice_address	u(v)
slice_type	ue(v)
entropy_slice_flag	u(1)
if(!entropy_slice_flag) {	
pic_parameter_set_id	ue(v)
if(IdrPicFlag) {	
idr_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
} else {	
pic_order_cnt_lsb	u(v)
short_term_ref_pic_set_pps_flag	u(1)
if(!short_term_ref_pic_set_pps_flag)	
short_term_ref_pic_set(num_short_term_ref_pic_sets)	
} else	
short_term_ref_pic_set_idx	u(v)
if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i = 0; i < num_long_term_pics; i++) {	
delta_poc_lsb_lt_minus1[i]	ue(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
}	
}	
}	
if(scaling_list_enable_flag deblocking_filter_in_APS_enabled_flag sample_adaptive_offset_enabled_flag adaptive_loop_filter_enabled_flag) {	
if(sample_adaptive_offset_enabled_flag)	
slice_sample_adaptive_offset_flag	u(1)
if(adaptive_loop_filter_enabled_flag)	

[0432]

slice_adaptive_loop_filter_flag	u(1)
aps_id	ue(v)
}	
if(slice_type == EP slice_type == EB) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == EB)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
ref_pic_list_modification()	
ref_pic_list_combination()	
}	
if(singleloop_decoding_flag) {	
if(slice_type == EI) //Syntax related inter-layer prediction	
inter_layer_intra_prediction_flag	u(1)
inter_layer_differential_coding_flag	u(1)
else {	
inter_layer_syntax_prediction_flag	u(1)
inter_layer_residual_prediction_flag	u(1)
}	
}	
else {	
inter_layer_intra_prediction_flag	u(1)
if(slice_type != EI) {	
inter_layer_residual_prediction_flag	u(1)
}	
inter_layer_differential_coding_flag	u(1)
}	
if(slice_type != EI)	
cabac_init_idc	ue(v)
if(!entropy_slice_flag) {	
slice_qp_delta	se(v)
inherit_dbl_params_from_APS_flag	u(1)
if(!inherit_dbl_params_from_APS_flag) {	
disable_deblocking_filter_flag	u(1)
if(!disable_deblocking_filter_flag) {	

[0433]

beta_offset_div2	se(v)
tc_offset_div2	se(v)
}	
}	
if(slice_type == EB)	
collocated_from_l0_flag	u(1)
if((weighted_pred_flag && slice_type == EP) (weighted_bipred_idc == 1 && slice_type == EB))	
pred_weight_table()	
}	
if(slice_type == EP slice_type == EB)	
5_minus_max_num_merge_cand	ue(v)
if(adaptive_loop_filter_enabled_flag && aps_adaptive_loop_filter_flag)	
alf_cu_control_param()	
for(i = 0; i < num_substreams_minus1 + 1; i++){	
substream_length_mode	u(2)
substream_length[i]	u(v)
}	
}	

[0434] 표 29는 본 발명에 따른 단기 참조 픽처 세트 선택스(Short-term reference picture set syntax)의 일 예를 나타낸 것이다.

[0435] <표 29>

[0436]

	Descriptor
short_term_ref_pic_set(idx) {	
inter_ref_pic_set_prediction_flag	u(1)
if(inter_ref_pic_set_prediction_flag) {	
delta_idx_minus1	ue(v)
delta_rps_sign	u(1)
abs_delta_rps_minus1	ue(v)
for(j = 0; j <= NumDeltaPocs[RIdx]; j++) {	
ref_idc0[j]	u(1)
if(!ref_idc0[j])	
ref_idc1[j]	u(1)
}	
}	
else {	
num_negative_pics	ue(v)
num_positive_pics	ue(v)
for(i = 0; i < num_negative_pics; i++) {	
delta_poc_s0_minus1[i]	ue(v)
used_by_curr_pic_s0_flag[i]	u(1)
}	
for(i = 0; i < num_positive_pics; i++) {	
delta_poc_s1_minus1[i]	ue(v)
used_by_curr_pic_s1_flag[i]	u(1)
}	
}	
}	

[0437] 표 30은 본 발명에 따른 참조 픽처 리스트 수정 선택스(Reference picture list modification syntax)의 일 예를 나타낸 것이다.

[0438] <표 30>

[0439]

	Descriptor
ref_pic_list_modification() {	
if(slice_type == P slice_type == B slice_type == EP slice_type == EB) {	
ref_pic_list_modification_flag_l0	u(1)
if(ref_pic_list_modification_flag_l0)	
do {	
ref_pic_list_modification_idc	ue(v)
if(ref_pic_list_modification_idc != 3)	
ref_pic_set_idx	ue(v)
} while(ref_pic_list_modification_idc != 3)	
}	
if(slice_type == B slice_type == EB) {	
ref_pic_list_modification_flag_l1	u(1)
if(ref_pic_list_modification_flag_l1)	
do {	
ref_pic_list_modification_idc	ue(v)
if(ref_pic_list_modification_idc != 3)	
ref_pic_set_idx	ue(v)
} while(ref_pic_list_modification_idc != 3)	
}	
}	
}	

[0440] 표 31은 본 발명에 따른 참조 픽처 리스트 조합 선택스(Reference picture lists combination syntax)의 일 예를 나타낸 것이다.

[0441] <표 31>

	Descriptor
ref_pic_list_combination() {	
if(slice_type == B slice_type == EB) {	
ref_pic_list_combination_flag	u(1)
if(ref_pic_list_combination_flag) {	
num_ref_idx_lc_active_minus1	ue(v)
ref_pic_list_modification_flag_lc	u(1)
if(ref_pic_list_modification_flag_lc)	
for (i=0; i <= num_ref_idx_lc_active_minus1; i++) {	
pic_from_list_0_flag	u(1)
ref_idx_list_curr	ue(v)
}	
}	
}	
}	
}	

[0443] 표 32는 본 발명에 따른 샘플 적응적 오프셋 파라미터 선택스(Sample adaptive offset parameter syntax)의 일 예를 나타낸 것이다.

[0444] <표 32>

[0445]

sao_param() {	Descriptor
sao_split_param(0, 0, 0, 0)	
sao_offset_param(0, 0, 0, 0)	
sao_flag_cb	u(1)
if(sao_flag_cb) {	
sao_split_param(0, 0, 0, 1)	
sao_split_param(0, 0, 0, 1)	
}	
sao_flag_cr	u(1)
if(sao_flag_cr) {	
sao_split_param(0, 0, 0, 2)	
sao_split_param(0, 0, 0, 2)	
}	
}	
sao_split_param(rx, ry, saoDepth, cIdx) {	Descriptor
if(saoDepth < SaoMaxDepth)	
sao_split_flag [cIdx][saoDepth][rx][ry]	u(1)
else	
sao_split_flag[cIdx][saoDepth][rx][ry] = 0	
if(sao_split_flag[cIdx][saoDepth][rx][ry]) {	
sao_split_param(2*rx + 0, 2*ry + 0, saoDepth + 1, cIdx)	
sao_split_param(2*rx + 1, 2*ry + 0, saoDepth + 1, cIdx)	
sao_split_param(2*rx + 0, 2*ry + 1, saoDepth + 1, cIdx)	
sao_split_param(2*rx + 1, 2*ry + 1, saoDepth + 1, cIdx)	
}	
}	
sao_offset_param(rx, ry, saoDepth, cIdx) {	Descriptor
if(sao_split_flag[cIdx][saoDepth][rx][ry]) {	
sao_offset_param(2*rx + 0, 2*ry + 0, saoDepth + 1, cIdx)	
sao_offset_param(2*rx + 1, 2*ry + 0, saoDepth + 1, cIdx)	
sao_offset_param(2*rx + 0, 2*ry + 1, saoDepth + 1, cIdx)	
sao_offset_param(2*rx + 1, 2*ry + 1, saoDepth + 1, cIdx)	
} else {	
sao_type_idx [cIdx][saoDepth][rx][ry]	ue(v)
if(sao_type_idx[cIdx][saoDepth][rx][ry] != 0)	
for(i = 0; i < NumSaoClass[sao_type_idx]; i++)	
sao_offset [cIdx][saoDepth][x0][y0][i]	se(v)
}	
}	

[0446] 표 33은 본 발명에 따른 적응적 루프 필터 파라미터 선택스의 일 예를 나타낸 것이다.

[0447] <표 33>

[0448]	alf_param() {	Descriptor
	alf_region_adaptation_flag	u(1)
	alf_length_luma_minus_5_div2	ue(v)
	alf_no_filters_minus1	ue(v)
	if (alf_no_filters_minus1 == 1)	
	alf_start_second_filter	ue(v)
	else if (alf_no_filters_minus1 > 1) {	
	for (i=1; i < (alf_region_adaptation_flag ? 16 : 15); i++)	
	alf_filter_pattern[i]	u(1)
	}	
	if (AlfNumFilters > 1)	
	alf_pred_method	u(1)
	for (i=0; i < AlfNumFilters; i++)	
	alf_nb_pred_luma[i]	u(1)
	for (i=0; i < AlfNumFilters; i++)	
	for (j=0; j < AlfCodedLengthLuma; j++)	
	alf_coeff_luma[i][j]	ge(v)
	alf_chroma_idc	ue(v)
	if (alf_chroma_idc) {	
	alf_length_chroma_minus_5_div2	ue(v)
	for (i = 0; i < AlfCodedLengthChroma; i++)	
	alf_coeff_chroma[i]	se(v)
	}	
	}	
	}	

[0449] 표 34는 본 발명에 따른 적응적 루프 필터 코딩 유닛 제어 파라미터 선택스(Adaptive loop filter coding unit control parameter syntax)의 일 예를 나타낸 것이다,

[0450] <표 34>

[0451]	alf_cu_control_param() {	Descriptor
	alf_cu_control_flag	u(1)
	if(alf_cu_control_flag) {	
	alf_cu_control_max_depth	ue(v)
	alf_length_cu_control_info	se(v)
	for(i = 0; i < NumAlfCuFlag; i++)	
	alf_cu_flag[i]	u(1)
	}	
	}	

[0452] 표 35는 본 발명에 따른 예측 가중치 테이블 선택스(Prediction weight table syntax)의 일 예를 나타낸 것이다.

[0453] <표 35>

[0454]

	Descriptor
pred_weight_table() {	
luma_log2_weight_denom	ue(v)
if(chroma_format_idc != 0)	
delta_chroma_log2_weight_denom	se(v)
if (slice_type == P (slice_type == B && ref_pic_list_combination_flag == 0)) {	
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++) {	
luma_weight_l0_flag	u(1)
if(luma_weight_l0_flag) {	
delta_luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if (chroma_format_idc != 0) {	
chroma_weight_l0_flag	u(1)
if(chroma_weight_l0_flag)	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l0[i][j]	se(v)
delta_chroma_offset_l0[i][j]	se(v)
}	
}	
}	
}	
if(slice_type == B) {	
if (ref_pic_list_combination_flag == 0) {	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {	
luma_weight_l1_flag	u(1)
if(luma_weight_l1_flag) {	
delta_luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_format_idc != 0) {	
chroma_weight_l1_flag	u(1)
if(chroma_weight_l1_flag)	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)

[0455]

	Descriptor
slice_data_in_scalable_extension () {	
CurrTbAddr = LCUAddress	
moreDataFlag = 1	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlagIdx = -1	
do {	
XLCU = HorLumaLocation(CurrTbAddr)	
YLCU = VerLumaLocation(CurrTbAddr)	
moreDataFlag = coding_tree_in_scalable_extension (XLCU, YLCU, Log2TbSize, 0)	
CurrTbAddr = NextTbAddress(CurrTbAddr)	
if(CurrTbAddr == firstTbInTileAddr)	
rbsp_trailingbits()	
} while(moreDataFlag)	
}	

[0462] 표 38은 본 발명에 따른 코딩 트리 선택스의 일 예를 나타낸 것이다.

[0463] <표 38>

[0464]

	Descriptor
coding_tree(x0, y0, log2CUSize, cuDepth) {	
if(x0 + (1 << log2CUSize) <= PicWidthInSamples _L && y0 + (1 << log2CUSize) <= PicHeightInSamples _L && cuAddress(x0, y0) >= SliceAddress && log2CUSize > Log2MinCUSize) {	
split_coding_unit_flag[x0][y0]	ae(v)
}	
if(adaptive_loop_filter_flag && alf_cu_control_flag) {	
if(cuDepth <= alf_cu_control_max_depth)	
if(cuDepth == alf_cu_control_max_depth split_coding_unit_flag[x0][y0] == 0)	
AlfCuFlagIdx++	
}	
if(cu_qp_delta_enabled_flag && log2CUSize >= log2MinCUDQPSize)	
IsCuQpDeltaCoded = 0	
if(split_coding_unit_flag[x0][y0]) {	
x1 = x0 + ((1 << log2CUSize) >> 1)	
y1 = y0 + ((1 << log2CUSize) >> 1)	
if(cuAddress(x1, y0) > SliceAddress)	
moreDataFlag = coding_tree(x0, y0, log2CUSize - 1, cuDepth + 1)	
if(cuAddress(x0, y1) > SliceAddress && moreDataFlag && x1 < PicWidthInSamples _L)	
moreDataFlag = coding_tree(x1, y0, log2CUSize - 1, cuDepth + 1)	
if(cuAddress(x1, y1) > SliceAddress && moreDataFlag && y1 < PicHeightInSamples _L)	
moreDataFlag = coding_tree(x0, y1, log2CUSize - 1, cuDepth + 1)	
if(moreDataFlag && x1 < PicWidthInSamples _L && y1 < PicHeightInSamples _L)	
moreDataFlag = coding_tree(x1, y1, log2CUSize - 1, cuDepth + 1)	
} else {	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlag[x0][y0] = alf_cu_flag[AlfCuFlagIdx]	
coding_unit(x0, y0, log2CUSize)	
if(granularity_block_boundary(x0, y0, log2CUSize)) {	
end_of_slice_flag	ae(v)

[0465]	moreDataFlag = !end_of_slice_flag	
	} else	
	moreDataFlag = 1	
	}	
	return moreDataFlag	
	}	

[0466] 표 39는 본 발명에 따른, 스케일러블 확장에서 코딩 트리 선택스(Coding tree in scalable extension syntax)의 일 예를 나타낸 것이다.

[0467] <표 39>

[0468]

	Descriptor
coding_tree_in_scalable_extension (x0, y0, log2CUSize, cuDepth) {	
if(x0 + (1 << log2CUSize) <= PicWidthInSamples _L && y0 + (1 << log2CUSize) <= PicHeightInSamples _L && cuAddress(x0, y0) >= SliceAddress && log2CUSize > Log2MinCUSize) {	
split_coding_unit_flag[x0][y0]	ae(v)
}	
if(adaptive_loop_filter_flag && alf_cu_control_flag) {	
if(cuDepth <= alf_cu_control_max_depth)	
if(cuDepth == alf_cu_control_max_depth split_coding_unit_flag[x0][y0] == 0)	
AlfCuFlagIdx++	
}	
if(cu_qp_delta_enabled_flag && log2CUSize >= log2MinCUDQPSize)	
IsCuQpDeltaCoded = 0	
if(split_coding_unit_flag[x0][y0]) {	
x1 = x0 + ((1 << log2CUSize) >> 1)	
y1 = y0 + ((1 << log2CUSize) >> 1)	
if(cuAddress(x1, y0) > SliceAddress)	
moreDataFlag = coding_tree_in_scalable_extension (x0, y0, log2CUSize - 1, cuDepth + 1)	
if(cuAddress(x0, y1) > SliceAddress && moreDataFlag && x1 < PicWidthInSamples _L)	
moreDataFlag = coding_tree_in_scalable_extension (x1, y0, log2CUSize - 1, cuDepth + 1)	
if(cuAddress(x1, y1) > SliceAddress && moreDataFlag && y1 < PicHeightInSamples _L)	
moreDataFlag = coding_tree_in_scalable_extension (x0, y1, log2CUSize - 1, cuDepth + 1)	
if(moreDataFlag && x1 < PicWidthInSamples _L && y1 < PicHeightInSamples _L)	
moreDataFlag = coding_tree_in_scalable_extension (x1, y1, log2CUSize - 1, cuDepth + 1)	

[0469]

} else {	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlag[x0][y0] = alf_cu_flag[AlfCuFlagIdx]	
coding_unit_in_scalable_extension (x0, y0, log2CUSize)	
if(granularity_block_boundary(x0, y0, log2CUSize)) {	
end_of_slice_flag	ae(v)
moreDataFlag = !end_of_slice_flag	
} else	
moreDataFlag = 1	
}	
return moreDataFlag	
}	

[0470] 표 40은 본 발명에 따른 코딩 유닛 선택스의 일 예를 나타낸 것이다.

[0471] <표 40>

[0472]	coding_unit(x0, y0, log2CUSize) {	Descriptor
	if(slice_type != I)	
	skip_flag [x0][y0]	ae(v)
	if(skip_flag[x0][y0])	
	prediction_unit(x0, y0, log2CUSize)	
	else if(slice_type != I log2CUSize == Log2MinCUSize) {	
	if(slice_type != I)	
	pred_mode_flag	ae(v)
	if(PredMode != MODE_INTRA log2CUSize == Log2MinCUSize)	
	part_mode	ae(v)
	x1 = x0 + ((1 << log2CUSize) >> 1)	
	y1 = y0 + ((1 << log2CUSize) >> 1)	
	x2 = x1 - ((1 << log2CUSize) >> 2)	
	y2 = y1 - ((1 << log2CUSize) >> 2)	
	x3 = x1 + ((1 << log2CUSize) >> 2)	
	y3 = y1 + ((1 << log2CUSize) >> 2)	
	if(PartMode == PART_2Nx2N) {	
	prediction_unit(x0, y0, log2CUSize)	
	} else if(PartMode == PART_2NxN) {	
	prediction_unit(x0, y0, log2CUSize)	
	prediction_unit(x0, y1, log2CUSize)	
	} else if(PartMode == PART_Nx2N) {	
	prediction_unit(x0, y0, log2CUSize)	
	prediction_unit(x1, y0, log2CUSize)	
	} else if(PartMode == PART_2NxN) {	
	prediction_unit(x0, y0, log2CUSize)	
	prediction_unit(x0, y2, log2CUSize)	
	} else if(PartMode == PART_2NxN) {	
	prediction_unit(x0, y0, log2CUSize)	
	prediction_unit(x0, y3, log2CUSize)	
	} else if(PartMode == PART_nLx2N) {	
	prediction_unit(x0, y0, log2CUSize)	
	prediction_unit(x2, y0, log2CUSize)	
	} else if(PartMode == PART_nRx2N) {	
	prediction_unit(x0, y0, log2CUSize)	
	prediction_unit(x3, y0, log2CUSize)	
	} else { /* PART_NxN */	
	prediction_unit(x0, y0, log2CUSize)	
	prediction_unit(x1, y0, log2CUSize)	

[0473]

prediction_unit(x0, y1 , log2CUSize)	
prediction_unit(x1, y1 , log2CUSize)	
}	
if(!pcm_flag) {	
transform_tree(x0, y0, log2CUSize, log2CUSize, log2CUSize, 0, 0)	
transform_coeff(x0, y0, x0, y0, log2CUSize, log2CUSize, 0, 0)	
}	
}	
}	

[0474] 표 41은 본 발명에 따른, 스케일러블 확장에서 코딩 유닛 선택스(Coding unit in_scalable extension syntax)의 일 예를 나타낸 것이다.

[0475] <표 41>

[0476]

	Descriptor
coding_unit_in_scalable_extension (x0, y0, log2CUSize) {	
if(ILDiffCodingFlag)	
il_diff_mode[x0][y0]	
if(slice_type != EI)	
skip_flag[x0][y0]	ae(v)
if ((ILIntraPredFlag ILSyntaxPredFlag)	
&& !skip_flag[x0][y0] && !il_diff_mode[x0][y0])	
il_mode[x0][y0]	ae(v)
if (!il_mode[x0][y0]){	
if(skip_flag[x0][y0])	
prediction_unit(x0, y0, log2CUSize)	
else if(slice_type !=EI log2CUSize == Log2MinCUSize) {	
if(slice_type !=EI)	
pred_mode_flag	ae(v)
if(PredMode != MODE_INTRA log2CUSize ==	
Log2MinCUSize)	
part_mode	ae(v)
x1 = x0 + ((1 << log2CUSize) >> 1)	
y1 = y0 + ((1 << log2CUSize) >> 1)	
x2 = x1 - ((1 << log2CUSize) >> 2)	
y2 = y1 - ((1 << log2CUSize) >> 2)	
x3 = x1 + ((1 << log2CUSize) >> 2)	
y3 = y1 + ((1 << log2CUSize) >> 2)	
if(PartMode == PART_2Nx2N) {	
prediction_unit(x0, y0, log2CUSize)	
} else if(PartMode == PART_2NxN) {	
prediction_unit(x0, y0, log2CUSize)	
prediction_unit(x0, y1, log2CUSize)	
} else if(PartMode == PART_Nx2N) {	
prediction_unit(x0, y0, log2CUSize)	
prediction_unit(x1, y0, log2CUSize)	
} else if(PartMode == PART_2NxN_U) {	
prediction_unit(x0, y0, log2CUSize)	
prediction_unit(x0, y2, log2CUSize)	
} else if(PartMode == PART_2NxN_D) {	
prediction_unit(x0, y0, log2CUSize)	
prediction_unit(x0, y3, log2CUSize)	
} else if(PartMode == PART_nLx2N) {	
prediction_unit(x0, y0, log2CUSize)	

[0477]

prediction_unit(x2, y0 , log2CUSize)	
} else if(PartMode == PART_nRx2N) {	
prediction_unit(x0, y0 , log2CUSize)	
prediction_unit(x3, y0 , log2CUSize)	
} else { /* PART_NxN */	
prediction_unit(x0, y0 , log2CUSize)	
prediction_unit(x1, y0 , log2CUSize)	
prediction_unit(x0, y1 , log2CUSize)	
prediction_unit(x1, y1 , log2CUSize)	
}	
}	
}	
if(ILResPredFlag && !ILIntraPredFlag && !il_diff_mode[x0][y0])	
il_res_mode[x0][y0]	ae(v)
if(!pcm_flag && !skip_flag[x0][y0]) {	
transform_tree(x0, y0, log2CUSize, log2CUSize, log2CUSize, 0, 0)	
transform_coeff(x0, y0, x0, y0, log2CUSize, log2CUSize, 0, 0)	
}	

[0478] 표 42는 본 발명에 따른 예측 유닛 선택스의 일 예를 나타낸 것이다.

[0479] <표 42>

[0480]

	Descriptor
prediction_unit(x0, y0, log2CUSize) {	
if(skip_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else if(PredMode == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag && log2CUSize >= Log2MinIPCMCUSize && log2CUSize <= Log2MaxIPCMCUSize)	
pcm_flag	ae(v)
if(pcm_flag) {	
while (!byte_aligned())	
pcm_alignment_zero_bit	u(v)
for(i = 0; i < 1 << (log2CUSize << 1); i++)	
pcm_sample_luma [i]	u(v)
for(i = 0; i < (1 << (log2CUSize << 1)) >> 1; i++)	
pcm_sample_chroma [i]	u(v)
} else {	
prev_intra_luma_pred_flag [x0][y0]	ae(v)
if(prev_intra_luma_pred_flag[x0][y0])	
mpm_flag [x0][y0]	ae(v)
else	
rem_intra_luma_pred_mode [x0][y0]	ae(v)
intra_chroma_pred_mode [x0][y0]	ae(v)
SignaledAsChromaDC = (chroma_pred_from_luma_enabled_flag ? intra_chroma_pred_mode[x0][y0] == 3 : intra_chroma_pred_mode[x0][y0] == 2)	
}	
} else { /* MODE_INTER */	
merge_flag [x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(slice_type == B slice_type == EB)	
inter_pred_flag [x0][y0]	ae(v)
if(inter_pred_flag[x0][y0] == Pred_LC) {	
if(num_ref_idx_lc_active_minus1 > 0)	
ref_idx_lc [x0][y0]	ae(v)

[0481]

mvd_coding(mvd_lc[x0][y0][0], mvd_lc[x0][y0][1])	
mvp_lc_flag[x0][y0]	ae(v)
}	
else { /* Pred L0 or Pred BI */	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(mvd_l0[x0][y0][0], mvd_l0[x0][y0][1])	
mvp_l0_flag[x0][y0]	ae(v)
}	
if(inter_pred_flag[x0][y0] == Pred BI) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
mvd_coding(mvd_l1[x0][y0][0], mvd_l1[x0][y0][1])	
mvp_l1_flag[x0][y0]	ae(v)
}	
}	
}	
}	

[0482] 표 43은 본 발명에 따른 움직임 벡터 차분 코딩 신택스의 일 예를 나타낸 것이다.

[0483] <표 43>

mvd_coding(mvd x, mvd y) {	Descriptor
abs_mvd_greater0_flag[0]	ae(v)
abs_mvd_greater0_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0])	
abs_mvd_greater1_flag[0]	ae(v)
if(abs_mvd_greater0_flag[1])	
abs_mvd_greater1_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0]) {	
if(abs_mvd_greater1_flag[0])	
abs_mvd_minus2[0]	ae(v)
mvd_sign_flag[0]	ae(v)
}	
if(abs_mvd_greater0_flag[1]) {	
if(abs_mvd_greater1_flag[1])	
abs_mvd_minus2[1]	ae(v)
mvd_sign_flag[1]	ae(v)
}	
mvd_x = abs_mvd_greater0_flag[0] * (abs_mvd_minus2[0] + 2) * (1 - 2 * mvd_sign_flag[0])	
mvd_y = abs_mvd_greater0_flag[1] * (abs_mvd_minus2[1] + 2) * (1 - 2 * mvd_sign_flag[1])	
}	

[0485] 표 44는 본 발명에 따른 변환 트리 신택스의 일 예를 나타낸 것이다.

[0486] <표 44>

[0487]

	Descriptor
transform_tree(x0, y0, log2CUSize, log2TrafoWidth, log2TrafoHeight, trafoDepth, blkIdx) {	
if(trafoDepth == 0 && IntraSplitFlag == 0 && PredMode != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
no_residual_data_flag	ae(v)
if(!no_residual_data_flag) {	
log2TrafoSize = (log2TrafoWidth + log2TrafoHeight) >> 1	
intraSplitFlag = (IntraSplitFlag && trafoDepth == 0 ? 1 : 0)	
interSplitFlag = (max_transform_hierarchy_depth_inter == 0 && PredMode == MODE_INTER && PartMode != PART_2Nx2N && trafoDepth == 0)	
maxDepth = (PredMode == MODE_INTRA ? max_transform_hierarchy_depth_intra + IntraSplitFlag : max_transform_hierarchy_depth_inter + InterSplitFlag)	
xBase = x0 - (x0 & (1 << log2TrafoWidth))	
yBase = y0 - (y0 & (1 << log2TrafoHeight))	
if(log2TrafoSize <= Log2MaxTrafoSize && log2TrafoSize > Log2MinTrafoSize && trafoDepth < maxDepth && !intraSplitFlag && !interSplitFlag)	
split_transform_flag [x0][y0][trafoDepth]	ae(v)
if(PredMode != MODE_INTRA && log2TrafoSize <= Log2MaxTrafoSize) {	
firstChromaCbf = (log2TrafoSize == Log2MaxTrafoSize trafoDepth == 0) ? 1 : 0	
if(firstChromaCbf log2TrafoSize > Log2MinTrafoSize) { [Ed. (WJ): Log2MinTrafoSize or 2?]	
if(firstChromaCbf cbf_cb[xBase][yBase][trafoDepth - 1]) {	
readCbf = true	
if(blkIdx == 3 && log2TrafoSize < Log2MaxTrafoSize)	
readCbf = cbf_cb[xBase][yBase][trafoDepth] cbf_cb[xBase + (1 << log2TrafoWidth)][yBase][trafoDepth] cbf_cb[xBase][yBase + (1 << log2TrafoHeight)][trafoDepth]	
if(!readCbf)	
cbf_cb[x0][y0][trafoDepth] = 1	
else	
cbf_cb [x0][y0][trafoDepth]	ae(v)
}	
if(firstChromaCbf cbf_cr[xBase][yBase][trafoDepth - 1]) {	
readCbf = true	
if(blkIdx == 3 && log2TrafoSize < Log2MaxTrafoSize)	

[0488]

<pre> readCbf = cbf_cr[xBase][yBase][trafoDepth] cbf_cr[xBase + (1 << log2TrafoWidth)][yBase][trafoDepth] cbf_cr[xBase][yBase + (1 << log2TrafoHeight)][trafoDepth] </pre>	
<pre> if (!readCbf) </pre>	
<pre> cbf_cr[x0][y0][trafoDepth] = 1 </pre>	
<pre> else </pre>	
<pre> cbf_cr[x0][y0][trafoDepth] </pre>	ae(v)
<pre> } </pre>	
<pre> } </pre>	
<pre> } </pre>	
<pre> if(split_transform_flag[x0][y0][trafoDepth]) { </pre>	
<pre> if(InterTUSplitDirection == 2) { </pre>	
<pre> x1 = x0 + ((1 << log2TrafoWidth) >> 1) </pre>	
<pre> y1 = y0 </pre>	
<pre> x2 = x0 </pre>	
<pre> y2 = y0 + ((1 << log2TrafoHeight) >> 1) </pre>	
<pre> x3 = x1 </pre>	
<pre> y3 = y2 </pre>	
<pre> } else { </pre>	
<pre> x1 = x0 + ((1 << log2TrafoWidth) >> 2) * InterTUSplitDirection </pre>	
<pre> y1 = y0 + ((1 << log2TrafoHeight) >> 2) * (1 - InterTUSplitDirection) </pre>	
<pre> x2 = x1 + ((1 << log2TrafoWidth) >> 2) * InterTUSplitDirection </pre>	
<pre> y2 = y1 + ((1 << log2TrafoHeight) >> 2) * (1 - InterTUSplitDirection) </pre>	
<pre> x3 = x2 + ((1 << log2TrafoWidth) >> 2) * InterTUSplitDirection </pre>	
<pre> y3 = y2 + ((1 << log2TrafoHeight) >> 2) * (1 - InterTUSplitDirection) </pre>	
<pre> log2TrafoHeight = log2TrafoHeight + 2 * InterTUSplitDirection - 1 </pre>	
<pre> log2TrafoWidth = log2TrafoWidth - 2 * InterTUSplitDirection + 1 </pre>	
<pre> } </pre>	
<pre> transform_tree(x0, y0, log2CUSize, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1, 0) </pre>	
<pre> transform_tree(x1, y1, log2CUSize, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1, 1) </pre>	
<pre> transform_tree(x2, y2, log2CUSize, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1, 2) </pre>	
<pre> transform_tree(x3, y3, log2CUSize, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1, 3) </pre>	
<pre> } else { </pre>	
<pre> if(PredMode == MODE_INTRA trafoDepth != 0 cbf_cb[x0][y0][trafoDepth] cbf_cr[x0][y0][trafoDepth]) { </pre>	

[0489]

readCbf = true	
if(blkIdx == 3 && PredMode != MODE_INTRA && ((log2CUSize <= Log2MaxTrafoSize+1) (log2TrafoSize < Log2MaxTrafoSize))	
readCbf = cbf_luma[xBase][yBase][trafoDepth] cbf_luma[xBase + (1 << log2TrafoWidth)][yBase][trafoDepth] cbf_luma[xBase][yBase + (1 << log2TrafoHeight)][trafoDepth] cbf_cb[xBase][yBase][trafoDepth - 1] cbf_cr[xBase][yBase][trafoDepth - 1]	
if (!readCbf)	
cbf_luma[x0][y0][trafoDepth] = 1	
else	
cbf_luma [x0][y0][trafoDepth]	ae(v)
}	
if(PredMode == MODE_INTRA)	
if(log2TrafoSize > Log2MinTrafoSize) { [Ed. (WJ): Log2MinTrafoSize or 2?]	
cbf_cb [x0][y0][trafoDepth]	ae(v)
cbf_cr [x0][y0][trafoDepth]	ae(v)
} else if(blkIdx == 0) {	
cbf_cb [x0][y0][trafoDepth - 1]	ae(v)
cbf_cr [x0][y0][trafoDepth - 1]	ae(v)
}	
}	
}	
}	

[0490] 표 45는 본 발명에 따른 변환 계수 선택스의 일 예를 나타낸 것이다.

[0491] <표 45>

[0492]

transform_coeff(x0, y0, xC, yC, log2TrafoWidth, log2TrafoHeight, trafoDepth, blkIdx) {	Descriptor
if(cbf_luma[x0][y0][trafoDepth] cbf_cb[x0][y0][trafoDepth] cbf_cr[x0][y0][trafoDepth]) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta	ae(v)
IsCuQpDeltaCoded = 1	
}	
if(split_transform_flag[x0][y0][trafoDepth]) {	
if(InterTUSplitDirection == 2) {	
x1 = x0 + ((1 << log2TrafoWidth) >> 1)	
y1 = y0	
x2 = x0	
y2 = y0 + ((1 << log2TrafoHeight) >> 1)	
x3 = x1	
y3 = y2	
} else {	
x1 = x0 + ((1 << log2TrafoWidth) >> 2) * InterTUSplitDirection	
y1 = y0 + ((1 << log2TrafoHeight) >> 2) * (1 - InterTUSplitDirection)	
x2 = x1 + ((1 << log2TrafoWidth) >> 2) * InterTUSplitDirection	
y2 = y1 + ((1 << log2TrafoHeight) >> 2) * (1 - InterTUSplitDirection)	
x3 = x2 + ((1 << log2TrafoWidth) >> 2) * InterTUSplitDirection	
y3 = y2 + ((1 << log2TrafoHeight) >> 2) * (1 - InterTUSplitDirection)	
log2TrafoHeight = log2TrafoHeight + 2 * InterTUSplitDirection - 1	
log2TrafoWidth = log2TrafoWidth - 2 * InterTUSplitDirection + 1	
}	
transform_coeff(x0, y0, x0, y0, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1,	
0)	
transform_coeff(x1, y1, x0, y0, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1,	
1)	
transform_coeff(x2, y2, x0, y0, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1,	
2)	
transform_coeff(x3, y3, x0, y0, log2TrafoWidth - 1, log2TrafoHeight - 1, trafoDepth + 1,	
3)	
} else {	
log2TrafoSize = ((log2TrafoWidth + log2TrafoHeight) >> 1)	
log2TrafoSizeC = ((log2TrafoSize == Log2MinTrafoSizeC) ?	
log2TrafoSize : log2TrafoSize - 1)	

[0493]

if (PredMode == MODE_INTRA) {	
scanIdx = ScanType[log2TrafoSize - 2][IntraPredMode]	
scanIdxC = ScanType[log2TrafoSize - 2][IntraPredModeC]	
} else {	
scanIdx = 0	
scanIdxC = 0	
}	
if (cbf_luma[x0][y0][trafoDepth])	
residual_coding(x0, y0, log2TrafoWidth, log2TrafoHeight, scanIdx, 0)	
if (log2TrafoSize > Log2MinTrafoSize) { [Ed. (WJ): Log2MinTrafoSize or 2?]	
if (cbf_cb[x0][y0][trafoDepth])	
residual_coding(x0, y0, log2TrafoSizeC, trafoDepth, scanIdxC, 1)	
if (cbf_cr[x0][y0][trafoDepth])	
residual_coding(x0, y0, log2TrafoSizeC, trafoDepth, scanIdxC, 2)	
} else if (blkIdx == 3) {	
if (cbf_cb[x0][y0][trafoDepth])	
residual_coding(xC, yC, log2TrafoSizeC, trafoDepth, scanIdxC, 1)	
if (cbf_cr[x0][y0][trafoDepth])	
residual_coding(xC, yC, log2TrafoSizeC, trafoDepth, scanIdxC, 2)	
}	
}	
}	
}	

[0494] 표 46은 본 발명에 따른 레지듀얼 코딩 선택스의 일 예를 나타낸 것이다.

[0495] <표 46>

[0496]

	Descriptor
residual_coding (x0, y0, log2TrafoWidth, log2TrafoHeight, scanIdx, cldx) {	
last_significant_coeff_x_prefix	ae(v)
last_significant_coeff_y_prefix	ae(v)
if (last_significant_coeff_x_prefix > 3)	
last_significant_coeff_x_suffix	ae(v)
if (last_significant_coeff_y_prefix > 3)	
last_significant_coeff_y_suffix	ae(v)
numCoeff = 0	
do {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][numCoeff][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][numCoeff][1]	
numCoeff++	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
numLastSubset = (numCoeff - 1) >> 4	
for(i = numLastSubset; i >= 0; i--) {	
offset = i << 4	
if(max(log2TrafoWidth, log2TrafoHeight) > 3) {	
xCG = ScanOrder[log2TrafoWidth - 2][log2TrafoHeight - 2][scanIdx][i][0]	
yCG = ScanOrder[log2TrafoWidth - 2][log2TrafoHeight - 2][scanIdx][i][1]	
rightCGFlag = (xCG == (1 << (log2TrafoWidth - 2)) - 1) ? 0 :	
significant_coeff_group_flag[xCG + 1][yCG]	
bottomCGFlag = (yCG == (1 << (log2TrafoHeight - 2)) - 1) ? 0 :	
significant_coeff_group_flag[xCG][yCG + 1]	
if((i < numLastSubset) && (rightCGFlag + bottomCGFlag < 2) && (i > 0))	
significant_coeff_group_flag [xCG][yCG]	ae(v)
for(n = 15; n >= 0; n--) {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
if ((n + offset) < (numCoeff - 1) &&	
significant_coeff_group_flag [xCG][yCG]) {	
numNZInCG = (i == numLastSubset) ? 1 : 0	
if ((n > 0) (rightCGFlag + bottomCGFlag == 2) (i == 0)	
(numNZInCG > 0)) {	
significant_coeff_flag [xC][yC]	ae(v)
numNZInCG += significant_coeff_flag [xC][yC]	
else	
significant_coeff_flag [xC][yC] = 1	
}	

[0497]

}	
} else {	
for(n = 15; n >= 0; n--) {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
if((n + offset) < (numCoeff - 1))	
significant_coeff_flag[xC][yC]	ae(v)
}	
}	
for(n = 15; n >= 0; n--) {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
if(significant_coeff_flag[xC][yC])	
coeff_abs_level_greater1_flag[n]	ae(v)
}	
for(n = 15; n >= 0; n--) {	
if(coeff_abs_level_greater1_flag[n])	
coeff_abs_level_greater2_flag[n]	ae(v)
}	
for(n = 15; n >= 0; n--) {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
if(significant_coeff_flag[xC][yC]) {	
coeff_sign_flag[n]	ae(v)
}	
for(n = 15; n >= 0; n--) {	
if(coeff_abs_level_greater2_flag[n])	
coeff_abs_level_minus3[n]	ae(v)
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
if(significant_coeff_flag[xC][yC]) {	
transCoeffLevel[x0][y0][cIdx][xC][yC] =	
(coeff_abs_level_minus3[n] + 3) * (1 - 2 * coeff_sign_flag[n])	
} else	
transCoeffLevel[x0][y0][cIdx][xC][yC] = 0	
}	
}	
}	
}	

[0498] 상술한 신택스들 중에서 **nal_unit_type**은 표 47과 같이 정의될 수 있다.

[0499] <표 47>

[0500]

nal_unit_type	NAL 유닛의 콘텐츠와 RBSP 선택스 구조	NAL 유닛 타입 클래스
0	미특정(Unspecified)	non-VCL
1	non-IDR and non-CRA 픽처의 부호화된 슬라이스(Coded slice of a non-IDR and non-CRA picture) slice_layer_rbsp()	VCL
2-3	리저브드(Reserved)	n/a
4	CRA 픽처의 부호화된 슬라이스(Coded slice of a CRA picture) slice_layer_rbsp()	VCL
5	IDR 픽처의 부호화된 슬라이스(Coded slice of an IDR picture) slice_layer_rbsp()	VCL
6	추가적 향상 정보(Supplemental enhancement information (SEI)) sei_rbsp()	non-VCL
7	시퀀스 파라미터 세트(Sequence parameter set) seq_parameter_set_rbsp()	non-VCL
8	픽처 파라미터 세트(Picture parameter set) pic_parameter_set_rbsp()	non-VCL
9	액세스 유닛 디리미터(Access unit delimiter) access_unit_delimiter_rbsp()	non-VCL
10-11	리저브드(Reserved)	n/a
12	필러 데이터(Filler data) filler_data_rbsp()	non-VCL
13	시퀀스 파라미터 세트 확장(Sequence parameter set extension) seq_parameter_set_extension_rbsp()	non-VCL
14	적응 파라미터 세트(Adaptation parameter set) aps_rbsp()	non-VCL
15	서브셋 시퀀스 파라미터 세트(Subset sequence parameter set) subset_seq_parameter_set_rbsp()	non-VCL
16-19	리저브드(Reserved)	n/a
20 or SVC_NAL	부호화된 슬라이스 확장(Coded slice extension) slice_layer_extension_rbsp()	VCL
21-23	리저브드(Reserved)	n/a
24..63	미특정(Unspecified)	non-VCL

- [0501] **dependency_id** 는 각 NAL 유닛의 종속성(dependency) 관계를 나타내기 위해 전송되는 ID(identification) 번호이다.
- [0502] **quality_id** 는 각 NAL 유닛의 퀄리티 레벨(quality level)을 나타내기 위해 전송되는 ID(identification) 번호이다.
- [0503] **singleloop_decoding_flag**의 값이 1이면 스케일러블 비트스트림의 최상위 레이어에서만 움직임 보상(motion compensation)이 수행된다.
singleloop_decoding_flag의 값이 0이면 모든 레이어에서 움직임 보상이

허용된다.

[0504] **slice_type**은 표 48에 따른 슬라이스의 코딩 타입을 특정한다.

[0505] <표 48>

[0506]

slice_type	slice_type의 명칭
0, 5	EP (스케일러블 확장에서 P 슬라이스, P slice in scalable extension)
1, 6	EB (스케일러블 확장에서 B 슬라이스, B slice in scalable extension)
2, 7	EI (I slice in scalable extension)

[0507] **inter_layer_intra_prediction_flag**의 값이 1이면, 코딩 유닛에서 인터 레이어 인트라 예측의 동작(operation)이 적응적으로 사용된다는 것을 특정한다. 그렇지 않으면, 인터 레이어 인트라 예측이 사용되지 않는다.

inter_layer_intra_prediction_flag가 없는 경우에는, **inter_layer_intra_prediction_flag**의 값을 0으로 추정할 수 있다. 또한, **ILIntraPredFlag**는 다음과 같이 유도될 수 있다.

[0508] - **singleloop_decoding_flag**의 값이 1일 때, **inter_layer_intra_prediction_flag**의 값이 1이고 **slice_type**이 EI이면, **ILIntraPredFlag**의 값을 1로 설정한다. 그렇지 않은 경우에는 **ILIntraPredFlag**의 값을 0으로 설정한다.

[0509] - **singleloop_decoding_flag**의 값이 0일 때, **ILIntraPredFlag**의 값은 **inter_layer_intra_prediction_flag**와 동일하게 설정된다.

[0510] **inter_layer_syntax_prediction_flag**의 값이 1이면, 코딩 유닛에서 인터 레이어 선택스 예측의 동작이 적응적으로 이용된다는 것을 특정한다. 그렇지 않으면, 인터 레이어 선택스 예측이 이용되지 않는다. **inter_layer_syntax_prediction_flag**이 없는 경우에는, **inter_layer_syntax_prediction_flag**의 값을 0으로 추정할 수 있다. 또한, **ILSyntaxPredFlag**은 다음과 같이 유도될 수 있다.

[0511] - **inter_layer_syntax_prediction_flag**의 값이 1이고 **slice_type**이 EI가 아니면, **ILSyntaxPredFlag**는 1로 설정된다. 그렇지 않은 경우에, **ILSyntaxPredFlag**는 0으로 설정된다.

[0512] **inter_layer_residual_prediction_flag**의 값이 1이면, 코딩 유닛에서 인터 레이어 레지듀얼 예측의 동작이 적응적으로 이용된다는 것을 특정한다. 그렇지 않으면, 인터 레이어 레지듀얼 예측이 이용되지 않는다.

inter_layer_residual_prediction_flag가 없으면, is not present, **inter_layer_residual_prediction_flag**의 값은 0으로 추정될 수 있다. 또한, **ILResPredFlag**는 다음과 같이 유도될 수 있다.

[0513] - **inter_layer_residual_prediction_flag**의 값이 1이고 **slice_type**이 EI가 아니면, **ILResPredFlag**는 1로 설정된다. 그렇지 않으면, **ILResPredFlag**는 0으로 설정된다.

- [0514] **cabac_init_idc**는 콘텍스트 변수(context variable)에 대한 초기화 프로세스에 사용되는 초기화 표(initialisation table)를 결정하기 위한 인덱스를 특정한다. cabac_init_idc는 0 이상 2 이하의 값을 가질 수 있다.
- [0515] **il_mode**는 베이스 레이어의 정보를 사용할지 여부를 지시는 선택스이다. il_mode는 singleloop_decoding_flag 값에 따라 다음과 같은 역할을 한다
- [0516] singleloop_decoding_flag=1이고 slice_type=EI인 경우, il_mode의 값이 1이면 복원된 베이스 레이어를 현재 레이어와의 해상도 차이를 기반으로 스케일링한 후, 현재 CU의 예측 정보로 사용할 수 있다. il_mode의 값이 0이면 베이스 레이어의 복원 정보를 사용하지 않는다.
- [0517] singleloop_decoding_flag=1이고 slice_type !=EI인 경우, il_mode의 값이 1이면 베이스 레이어의 움직임 정보, 인트라 예측 정보를 대응하는 현재 레이어와의 해상도 차이를 고려하여 스케일링한 후 복사하여 예측 신호를 만들고 그 값을 예측자(predictor)로 사용할 수 있다. il_mode의 값이 0이면 베이스 레이어의 복원 정보를 사용하지 않는다.
- [0518] singleloop_decoding_flag=0이고 il_mode의 값이 1이면 베이스 레이어의 복원된 픽셀 정보를 해상도를 고려한 스케일링 후 예측 정보로 사용할 수 있다.
- [0519] **il_res_mode**가 1이면 대응하는 베이스 레이어의 복원된 레지듀얼 데이터를 해상도를 고려하여 스케일링 한 후 현재 CU의 레지듀얼에 대한 예측자로 사용할 수 있다. il_res_mode의 값이 0이면 대응하는 베이스 레이어의 복원된 레지듀얼 데이터를 사용하지 않는다.
- [0520] **inter_layer_differential_coding_flag**의 값이 1이면, 코딩 유닛에서 인터 레이어 차분(differential) 코딩의 동작이 적응적으로 이용되는 것을 특정한다. 그렇지 않으면, 인터 레이어 차분 코딩이 이용되지 않는다. inter_layer_differential_coding_flag이 없으면, inter_layer_differential_coding_flag의 값을 0으로 추정한다. 또한, ILDiffCodingFlag는 다음과 같이 유도될 수 있다.
- [0521] - singleloop_decoding_flag의 값이 1이고 inter_layer_differential_coding_flag의 값이 1이며, slice_type이 EI이면, ILDiffCodingFlag의 값은 1로 설정된다. 그렇지 않으면, ILDiffCodingFlag의 값은 0으로 설정된다. is set to equal to 0.
- [0522] - singleloop_decoding_flag의 값이 1이 아니면, ILDiffCodingFlag의 값은 inter_layer_differential_coding_flag과 동일하게 설정된다.
- [0523]
- [0524] **인터 레이어 유닛 예측(inter-layer unit prediction)**
- [0525] 스케일러블 코딩에 사용되는 입력 신호(signal)는 레이어 간에 해상도, 프레임 레이트(frame rate), 비트 맵스, 컬러 포맷, 에스펙트 율(aspect ratio) 등이 상이할 수 있다.
- [0526] 이 점을 고려하여 레이어 간 예측(인터 레이어 예측)을 수행함으로써 중복(redundancy)을 줄이고 시뮬캐스트(simulcast) 대비 코딩 효율을 높일 수 있다.

- [0527] 예컨대, 정보의 중복량을 줄이는 한 방법으로 인헨스먼트 레이어에서 전송되는 처리 단위, 즉 CU, PU 및 또는 TU에 관한 정보를 베이스 레이어의 정보를 이용하여 줄이는 방법을 사용할 수 있다.
- [0528] 베이스 레이어의 유닛(CU, PU 및/또는 TU) 정보를 이용하여 인헨스먼트 레이어에서 전송되는 유닛 정보를 줄이는 방법을 인터 레이어 유닛 예측이라고 한다.
- [0529] 인터 레이어 유닛 예측은 도 1 내지 도 4에서 설명한 예측부 또는 인터 레이어 예측부에서 수행될 수 있다. 여기서는 설명의 편의를 위해, 예측부가 인터 레이어 유닛 예측을 수행하는 것으로 설명한다.
- [0530] 예컨대, 인터 레이어 유닛 예측이 적용되는 경우에, 예측부가 베이스 레이어의 유닛 정보를 획득하면 이를 기반으로 인헨스먼트 레이어의 파티셔닝이 수행될 수 있다.
- [0531] 도 15는 본 발명에 따른 인터 레이어 유닛 정보 예측의 일 예를 개략적으로 설명하는 도면이다.
- [0532] 도 15의 예에서는 인헨스먼트 레이어의 해상도가 베이스 레이어의 해상도보다 2 배 높다고 가정한다.
- [0533] 도 15에서 레이어 0은 베이스 레이어 혹은 참조 레이어일 수 있으며, 레이어 1은 인헨스먼트 레이어 혹은 현재 레이어일 수 있다.
- [0534] 도 15를 참조하면, 레이어 0의 LCU 블록(1510)은 다양한 CU, PU 또는 TU로 분할되어 있다. 여기서는, 이런 CU, PU 및/또는 TU의 분할 정보를 설명의 편의를 위해 CU 정보 혹은 유닛 정보라고 한다.
- [0535] 예측부는 베이스 레이어의 CU 정보를 업스케일링한 정보인 참조 CU 정보(1520)를 생성할 수 있다. 예측부는 참조 CU 정보(1520)를 이용하여 레이어 0의 CU 정보를 유도할 수 있다.
- [0536] 예컨대, 참조 CU 정보(1520)로부터 베이스 레이어의 CU 정보를 유도하여 레이어 1에 적용하는 경우에는, 레이어 1의 LCU 구조로서 LCU0(1530)과 같은 유닛 구조가 이용될 수 있다. 참조 CU 정보(1520)를 이용하지 않는 경우 즉, 베이스 레이어의 CU 정보를 이용하지 않는 경우에는, 레이어 1의 LCU 구조로서 LCU1(1540)과 같은 유닛 구조가 이용될 수 있다.
- [0537] 레이어 0의 CU 정보(유닛 정보)를 이용하는 것이 코딩 효율을 높이는데 도움이 될 수도 있고 도움이 되지 않을 수도 있으므로, 인코딩 장치는 레이어 0의 CU 정보를 사용할 것인지를 적응적으로 시그널링할 수 있다.
- [0538] 여기서는 설명의 편의를 위해, 유닛 정보가 CU, PU, TU의 분할 정보인 것으로 설명하였으나, CU, PU, TU의 분할 구조 이외에 CU, PU, TU에 관한 정보를 포함할 수 있다. 다시 말하면, 베이스 레이어의 CU 정보는 트리 구조 또는 파티셔닝에 관한 정보일 수도 있고, PU 정보(스킵 모드인지 논-스킵 모드인지 여부, 예측 방향, 움직임 벡터, 참조 인덱스 등)일 수도 있으며, 둘 모두일 수도 있다.

[0539] 예컨대, 인터 레이어 유닛 예측이 트리/파티셔닝 정보의 예측에 관한 것이라면, 다음과 같은 선택스들을 이용할 수 있다.

[0540] 표 49는 본 발명에 따른, 스케일러블 확장 레이어에 대한 슬라이스 데이터 선택스의 일 예를 나타낸 것이다.

[0541] <표 49>

[0542]	slice_data() {	Descriptor
	CurrTbAddr = first_tb_in_slice	
	moreDataFlag = 1	
	if(adaptive_loop_filter_flag && alf_cu_control_flag)	
	AlfCuFlagIdx = -1	
	do {	
	xCU = HorLumaLocation(CurrTbAddr)	
	yCU = VerLumaLocation(CurrTbAddr)	
	IsCuQpDeltaCoded = 0	
	bl_tree_info_skip_flag	ae(v)
	coding_tree(xCU, yCU, Log2TbSize)	
	if(!entropy_coding_mode_flag)	
	moreDataFlag = more_rbsp_data()	
	else {	
	end_of_slice_flag	ae(v)
	moreDataFlag = !end_of_slice_flag	
	}	
	CurrTbAddr = NextTbAddress(CurrTbAddr)	
	} while(moreDataFlag)	
	}	

[0543] 표 50은 본 발명에 따른, 스케일러블 확장 레이어에 대한 코딩 트리 선택스의 일 예를 나타낸 것이다.

[0544] <표 50>

[0545]

	Descriptor
coding_tree(x0, y0, log2CUSize) {	
if(x0 + (1 << log2CUSize) <= PicWidthInSamples _L && y0 + (1 << log2CUSize) <= PicHeightInSamples _L && log2CUSize > Log2MinCUSize)	
if(!bl_tree_info_skip_flag && BLSplitInfo[x0][y0])	
split_coding_unit_flag[x0][y0]	u(1) ae(v)
if(adaptive_loop_filter_flag && alf_cu_control_flag) {	
cuDepth = Log2MaxCUSize - log2CUSize	
if(cuDepth <= alf_cu_control_max_depth)	
if(cuDepth == alf_cu_control_max_depth split_coding_unit_flag[x0][y0] == 0)	
AlfCuFlagIdx++	
}	
if(split_coding_unit_flag[x0][y0]) {	
x1 = x0 + ((1 << log2CUSize) >> 1)	
....	

- [0546] **bl_tree_info_skip_flag** 의 값이 1이면 베이스 레이어(참조 레이어 혹은 레이어 0) 의 트리 정보를 그대로 사용한다는 것을 지시한다. **bl_tree_info_skip_flag** 의 값이 0이면 베이스 레이어의 트리 정보를 사용하지 않는다는 것을 지시한다.
- [0547] **bl_tree_info_skip_flag**에 관한 이해를 돕기 위해 도 5를 참조할 수 있다. 만일 **bl_tree_info_skip_flag** 의 값이 1이면 베이스 레이어의 트리 정보가 CU 정보 업스케일링에 의해 인헨스먼트 레이어(현재 레이어 혹은 레이어 1)의 해상도에 맞게 업스케일링된다. 따라서, 현재LCU(Largest Coding Unit)의 **split_coding_unit_flag** 값들은 업스케일링된 베이스 레이어의 분할 정보와 동일한 값으로 유도될 수 있다.
- [0548] **BLSplitInfo[x0][y0]** 는 업스케일링된 CU 정보에 분할 정보가 존재하는 경우에는 1의 값을 가지고, 업스케일링된 CU 정보에 분할 정보가 존재하지 않을 경우에는 0의 값을 가진다.
- [0549] 예를 들어, 인헨스먼트 레이어의 해상도가 베이스 레이어의 해상도의 2 배이고 인헨스먼트 레이어와 베이스 레이어의 CU 깊이(depth)가 동일할 경우, 업샘플링된 베이스 레이어의 분할 정보는 인헨스먼트 레이어의 분할 정보보다 깊이가 한 단계 더 낮게 된다.
- [0550] 만약 베이스 레이어에 분할 정보가 존재하지 않는다면, **BLSplitInfo**를 0으로 설정하고 인헨스먼트 레이어만을 위한 추가 정보(예컨대, **split_coding_unit_flag**)를 전송하여 업스케일링된 베이스 레이어로부터 예측되는 경우의 분할 정보보다 더 자세한 분할 정보를 알려줄 수 있다.
- [0551] 유사한 방법을 TU에도 적용할 수 있다. 예컨대, 소정의 플래그 정도를 사용하여 현재 레이어에 대한 TU 분할 정보를 처리하도록 할 수 있다.
- [0552] 표 51은 본 발명에 따른, 스케일러블 확장에 있어서 코딩 유닛 선택스의 일 예를

나타낸 것이다. 표 51의 예에서는 `bl_tu_info_skip_flag`라는 플래그를 사용하여 현재 레이어를 위한 TU 분할 정보 전송을 생략할 수 있다.

[0553] <표 51>

[0554]		Descriptor
	<code>coding_unit(x0, y0, log2CUSize) {</code>	
	<code> if(entropy_coding_mode_flag && slice_type != I)</code>	
	<code> skip_flag[x0][y0]</code>	<code>u(1) ae(v)</code>
	<code> ...</code>	
	<code> prediction_unit(x1, y1, log2CUSize - 1, log2CUSize - 1, 3)</code>	
	<code> }</code>	
	<code>}</code>	
	<code>if(!pcm_flag) {</code>	
	<code> bl_tu_info_skip_flag</code>	<code>u(1) ae(v)</code>
	<code> transform_tree(x0, y0, log2CUSize, 0, 0)</code>	
	<code> transform_coeff(x0, y0, log2CUSize, 0, 0)</code>	
	<code> transform_coeff(x0, y0, log2CUSize, 0, 1)</code>	
	<code> transform_coeff(x0, y0, log2CUSize, 0, 2)</code>	
	<code> }</code>	
	<code>}</code>	

[0555] 표 52는 본 발명에 따른, 스케일러블 확장에 있어서, 변환 트리 선택스의 일 예를 나타낸 것이다.

[0556] <표 52>

[0557]

	Descriptor
transform_tree(x0, y0, log2TrafoSize, trafoDepth, blkIdx) {	
if(entropy_coding_mode_flag && trafoDepth == 0 && IntraSplitFlag == 0)	
{	
if(PredMode != MODE_INTRA)	
no_residual_data_flag	u(1) ae(v)
residualDataPresentFlag = !no_residual_data_flag	
} else {	
....	
if(log2TrafoSize <= Log2MaxTrafoSize && !intraSplitFlag && log2TrafoSize > Log2MinTrafoSize && trafoDepth < maxDepth && bl tu info skip flag)	
split_transform_flag [x0][y0][trafoDepth]	u(1)
} else	
cbp_and_split_transform	vlc(n,v)
}	
if(log2TrafoSize <= Log2MaxTrafoSize && log2TrafoSize > Log2MinTrafoSize && trafoDepth < maxDepth && !intraSplitFlag && entropy_coding_mode_flag && bl tu info skip flag)	
split_transform_flag [x0][y0][trafoDepth]	u(1) ae(v)
if(PredMode != MODE_INTRA && log2TrafoSize <= Log2MaxTrafoSize && entropy_coding_mode_flag) {	
firstChromaCbf = (log2TrafoSize == Log2MaxTrafoSize trafoDepth == 0 ? 1 : 0)	
if(firstChromaCbf log2TrafoSize > Log2MinTrafoSize) {	
xBasis = x0 - (x0 & ((1 << log2TrafoSize) - 1))	
....	
}	
}	
}	

- [0558] 앞서 설명한 바와 같이, 표 51 및 표 52에서, bl_tu_info_skip_flag의 값이 1이면 베이스 레이어의 업스케일된 TU 분할 정보를 현재 레이어에서 그대로 사용할 수 있다. bl_tu_info_skip_flag의 값이 0이면 현재 레이어를 위한 TU 분할 정보가 인코딩 장치로부터 독립적으로 전송될 수 있다.
- [0559] 한편, 인터 레이어 유닛 예측을 적용하는데 있어서, CU/PU/TU의 정보들을 조합하여 사용할 수도 있다. 발명의 이해를 돕기 위해, 여기서는 베이스 레이어와 인헨스먼트 레이어 사이의 공간 비(spatial ratio)가 2인 경우, 즉 두 레이어의 해상도가 2 배 차이 나는 경우를 예로서 설명한다.
- [0560] 도 16은 본 발명에 따라서 인터 레이어 유닛 예측을 적용하는 일 예를 개략적으로 설명하는 도면이다.
- [0561] 도 16에서는 인터 레이어 유닛 예측을 적용할 때 베이스 레이어 구조를 리샘플링 하는 방법에 관해서 설명한다.
- [0562] 도 16(a)는 베이스 레이어의 CU 및 PU 구조를 나타낸다.
- [0563] 도 16(a)에서 베이스 레이어 블록(1600)은 인트라 코딩된 블록이고, 분할 영역(1620)은 NxN 파티션이며, 나머지 영역은 2Nx2N 파티션이라고 가정한다.
- [0564] 도 16(a)에서 참조 블록(1610)의 CU/PU 정보를 인헨스먼트 레이어에서

사용하기 위해, 두 배의 업샘플링을 수행하면, 베이스 레이어의 분할 구조를 어느 레벨(예를 들어, CU 레벨 혹은 PU 레벨)까지 이용할지에 따라서, 도 16(b)의 블록(1630)과 같은 분할(partition) 구조 혹은 도 16(c) 블록(1640)과 같은 분할 구조를 얻을 수 있다.

- [0565] 이 때, 베이스 레이어의 CU 정보만을 인헨스먼트 레이어에서 사용하고자 한다면 인헨스먼트 레이어의 CU 분할 구조를 블록(1630)의 분할 구조와 같이 구성할 수 있다. 이 경우에는 인헨스먼트 레이어의 영상이 블록 (1630)보다 더 세분화된 분할 구조를 가지는 경우는 커버할 수 없다.
- [0566] 반면, 베이스 레이어의 CU 정보뿐만 아니라, PU 정보까지 함께 인헨스먼트 레이어 의 CU 분할 구조에 반영하는 경우도 16(c)의 블록(1640)과 같이 베이스 레이어에서 $2N \times 2N$ 파티션이 아닌 더 세분화된 파티션(예를 들면 $N \times N$ 파티션)을 가지는 부분에 대응되는 인헨스먼트 레이어의 영역에서도 추가적인 CU분할이 가능하다. 따라서, 이 경우에 인헨스먼트 레이어의 영상은 보다 세분화된 분할 구조를 가질 수 있게 된다.
- [0567] 한편, 도 16에서는 베이스 레이어가 인트라 코딩된 블록(intra coded block)인 경우를 예로서 설명하였으나, 동일한 방법을 베이스 레이어가 인터 코딩된 블록(inter coded block)인 경우에도 적용할 수 있다.
- [0568] 도 17은 본 발명에 따라서 인터 레이어 유닛 예측을 적용하는 다른 예를 개략적으로 설명하는 도면이다.
- [0569] 도 17에서는 베이스 레이어의 블록(1700)이 인터 코딩된 블록이고, 영역(1720)는 $N \times 2N$ 파티션이고, 영역(1730)은 $2N \times N$ 파티션이라고 가정하자.
- [0570] 인터 코딩된 블록의 경우에는 $2N \times 2N$ 파티션, $N \times N$ 파티션 이외에도, $2N \times N$ 파티션, $N \times 2N$ 파티션, $2N \times nU$ 파티션, $2N \times nD$ 파티션, $nL \times 2N$ 파티션, $nR \times 2N$ 파티션과 같이 다양한 형태 파티션 타입을 사용할 수 있다.
- [0571] 베이스 레이어의 참조 블록(1710)의 CU/PU 정보를 인헨스먼트 레이어에서 사용하기 위해, 두 배의 업샘플링을 수행하면, 베이스 레이어의 분할 구조를 어느 레벨(예를 들어 CU 레벨 혹은 PU 레벨)까지 이용할지에 따라서, 도 17(b)의 블록(1740)과 같은 분할 구조 혹은 도 17(c)의 블록(1750)과 같은 분할 구조를 얻을 수 있다.
- [0572] 베이스 레이어의 CU 정보만을 인헨스먼트 레이어에서 사용하고자 하면, 참조 블록(1710)에서 영역(1720) 및 영역(1730)의 구조를 제외하고, CU의 분할 구조만을 업샘플링하여 인헨스먼트 레이어의 CU 분할 구조를 도 17(b)의 블록(1740)과 같이 구성할 수 있다.
- [0573] 또한, 베이스 레이어의 PU 정보까지 인헨스먼트 레이어의 CU 분할 구조에 반영하고자 하는 경우에는, 베이스 레이어에서 $2N \times 2N$ 파티션이 아닌 파티션 (예컨대, 영역(1720), 영역(1730)의 파티션)을 가지는 부분에 대응하는 인헨스먼트 레이어의 영역에 대해서, 도 17(c)와 같이 추가적인 CU 분할이 가능하다.

- [0574] 다시 말하면, 도 16 및 도 17에서 베이스 레이어의 PU 정보까지 인헨스먼트 레이어의 CU 분할 구조에 반영하고자 하는 경우에는, 베이스 레이어에서 PU 파티셔닝이 있는 영역에 대응하는 인헨스먼트 레이어의 영역에서 CU 분할이 이루어질 수 있다.
- [0575] 한편, 인헨스먼트 레이어에서 베이스 레이어의 분할 구조를 선택적으로 사용하는 방법을 적용할 수도 있다.
- [0576] 도 18은 본 발명에 따라서 인터 레이어 유닛 예측을 적용하는 또 다른 예를 개략적으로 설명하는 도면이다.
- [0577] 도 18에서는 인헨스먼트 레이어에서 베이스 레이어의 분할 구조를 재사용하는 방법에 관하여 설명한다.
- [0578] 도 18(a)와 같이 베이스 레이어의 블록(1800)이 $2N \times 2N$ 혹은 $N \times N$ 과 같이 정사각형 모양의 파티션을 가지는 경우와, $2N \times nU$, $2N \times nD$, $nL \times 2N$ 혹은 $nR \times 2N$ 과 같이 직사각형 형태의 파티션을 가지는 있다고 가정하자.
- [0579] 이 경우, CU 분할(split)은 정사각형의 형태($2N \times 2N$ 혹은 $N \times N$)로만 이루어지므로, 인헨스먼트 레이어에서 베이스 레이어의 분할 구조를 선택적으로 이용하는 방법을 고려할 수 있다.
- [0580] 예를 들어, 도 18(b)와 같이 파티션 형태에 상관없이, 참조 블록(1810)의 분할 구조를 인헨스먼트 레이어의 블록(1840)에 대한 분할 구조로서 모두 반영할 수 있다.
- [0581] 또한, 도 18(c)와 같이, 참조 블록의 PU 분할 구조 중 정사각형의 파티션(1820)은 인헨스먼트 레이어의 CU 분할 구조에 반영하고, 참조 블록의 PU 분할 구조 중 직사각형의 파티션(1830)은 인헨스먼트 레이어의 CU 분할 구조에 반영하지 않을 수 있다.
- [0582] 이와 같이, CU/PU/TU 정보들을 조합하여 사용하는 방법은, 임의의 단위별로 선택적인 사용이 가능하다. 즉, 시퀀스, 픽처들의 그룹, 단일 픽처, 복수의 슬라이스, 단일 슬라이스, 복수의 LCU, 단일 LCU 등과 같이 임의의 단위를 기준으로 베이스 레이어의 CU/PU/TU 정보들을 조합하여 인헨스먼트 레이어에 사용하는 방법을 선택할 것인지를 결정할 수 있다.
- [0583]
- [0584] 적응적 인터 레이어 텍스처 예측
- [0585] 인터 레이어 예측에 있어서 필터를 적용하여 참조 영상의 품질을 향상시킬 수 있다. 예컨대, 인터 레이어 인트라 예측을 수행하는 경우에, 참조 영상의 품질을 향상시키기 위해, 참조 영상에 필터를 적용할 수 있다.
- [0586] 공간적 스케일러빌리티(spatial scalability)의 경우를 예로 들면, 하위 레이어의 텍스처 정보에 업샘플링 필터를 적용하여 레이어 간 해상도를 동일하게 조정된 후에, 조정된 하위 레이어의 텍스처 정보를 참조 영상으로 이용하였다.
- [0587] 이때, 업샘플링 필터가 적용된 하위 레이어의 텍스처 정보에 추가로 필터링을 적용한 후 상위 레이어의 예측을 위한 참조 영상으로 사용할 수 있다. 본

- 명세서에서는 이 예측 방법을 적응적 인터 레이어 텍스처 예측이라고 표시한다.
- [0588] 도 19는 본 발명에 따라서 적응적 인터 레이어 텍스처 예측을 수행하는 방법의 일 예를 개략적으로 설명하는 도면이다.
- [0589] 도 19를 참조하면, 텍스처 예측부(1910)는 도 1의 텍스처 예측부에 대응한다. 따라서, 텍스처 예측부는 텍스처 예측과 함께 필요한 경우 리스케일링을 수행할 수 있다.
- [0590] 베이스 레이어에서 필터링부(1950)에 의해 적용되는 필터링은 다운 컨버터(1940)에서 다운 컨버팅된 입력 시퀀스와의 차이를 줄이기 위한 것이다.
- [0591] 인헨스먼트 레이어에서, 업샘플링부(1930)에 의해 업샘플링된 텍스처가 필터링부(1920)에서 필터링되는 것은 업샘플링된 베이스 레이어의 텍스처와 입력 시퀀스 사이의 오류를 줄이기 위한 것이다.
- [0592] 이때, 적용되는 필터는 상술한 바와 같은 목적을 달성하기 위한 필터면 된다. 예컨대, 필터링부(1920)에서 적용되는 필터는 인코딩 장치/디코딩 장치에서 소정의 탭 수와 계수로 미리 특정되어 있는 필터일 수도 있으며, 필터 파라미터(탭 수, 계수 등)가 적응적으로 시그널링 되는 필터일 수도 있다.
- [0593] 적응적 인터 레이어 텍스처 예측을 사용할 경우, 참조 영상의 품질 향상으로 인해서 코딩 효율이 향상될 수 있다
- [0594] 적응적 인터 레이어 텍스처 예측을 적용하는 경우에, 참조 영상의 품질은 향상되지만 복잡도가 증가하고 추가적인 필터 파라미터의 인코딩/디코딩이 필요하다. 따라서, 상황(예컨대, 인코딩 장치와 디코딩 장치의 성능, 입력 시퀀스와 업샘플링된 베이스 레이어 텍스처 간의 오차 등)에 따라서 적응적 인터 레이어 텍스처 예측을 사용할 것인지를 결정할 수 있다.
- [0595] 따라서, 필터링부(1920)에서 적용되는 필터를 적응적으로 인코딩/디코딩 루프에서 적용되는 필터라고 할 수 있다. 설명의 편의를 위해, 본 명세서에서는 필터링부(1920)에서 적용하는 필터를 인터 레이어 ALF라고 한다. 인코딩 장치/디코딩 장치가 인루프 필터로서의 ALF를 이용하는 경우라면, 인루프 필터로서의 ALF(Adaptive Loop Filter)를 인터 레이어 ALF로 사용할 수도 있다.
- [0596] 적응적 인터 레이어 텍스처 예측을 사용하기 위해서는 인터 레이어 ALF의 적용 여부를 지시하는 플래그가 필요하다. 인터 레이어 ALF의 적용 여부를 지시하는 플래그를 `inter_layer_adaptive_loop_filter_enabled_flag`라고 하자.
- [0597] `inter_layer_adaptive_loop_filter_enabled_flag`의 값이 0 이면, 인터 레이어 ALF를 적용하지 않는 것을 지시하고, `inter_layer_adaptive_loop_filter_enabled_flag`의 값이 1이면, 인터 레이어 ALF를 적용하는 것을 지시한다.
- [0598] `inter_layer_adaptive_loop_filter_enabled_flag`는 시퀀스 파라미터 세트(Sequence parameter set)에 정의될 수 있다.
- [0599] 표 53은 본 발명에 따라서 적응적 인터 레이어 텍스처 예측을 적용하기 위한 시퀀스 파라미터 세트의 일 예를 나타낸 것이다.
- [0600] <표 53>

[0601]	seq_parameter_set_rbsp() {	Descriptor
	profile_idc	u(8)
	reserved_zero_8bits /* equal to 0 */	u(8)
	level_idc	u(8)
	...	
	adaptive_loop_filter_enabled_flag	u(1)
	Inter_layer_adaptive_loop_filter_enabled_flag	u(1)
	pcm_loop_filter_disable_flag	u(1)
	cu_qp_delta_enabled_flag	u(1)
	temporal_id_nesting_flag	u(1)
	rbsp_trailing_bits()	
	}	

[0602] inter_layer_adaptive_loop_filter_enabled_flag는 시퀀스 파라미터 세트(Sequence parameter set)가 아닌 슬라이스 헤더에 정의될 수도 있다.

[0603] 표 54는 본 발명에 따라서 적응적 인터 레이어 텍스처 예측을 적용하기 위한 슬라이스 헤더의 일 예를 나타낸 것이다.

[0604] <표 54>

[0605]	slice_header() {	Descriptor
	...	
	Inter_layer_adaptive_loop_filter_enabled_flag	u(1)
	if(adaptive_loop_filter_enabled_flag) {	
	if(!shared_pps_info_enabled_flag)	
	alf_param()	
	alf_cu_control_param()	
	}	
	if(inter_layer_adaptive_loop_filter_enabled_flag){	
	alf_param()	
	alf_cu_control_param()	
	}	
	...	

[0606] 인코딩 장치와 디코딩 장치가 인루프 필터로서의 ALF를 사용하는 경우에, 픽처 파라미터 세트에서 ALF의 사용을 위해 사용하는 신택스를 인터 레이어 ALF에도 동일하게 적용해 볼 수 있다.

[0607] 표 55는 본 발명에 따라서 적응적 인터 레이어 텍스처 예측을 적용하기 위한 픽처 파라미터 세트 신택스의 일 예를 나타낸 것이다.

[0608] <표 55>

[0609]

	Descriptor
pic_parameter_set_rbsp() {	
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
entropy_coding_mode_flag	u(1)
shared_pps_info_enabled_flag	u(1)
if(shared_pps_info_enabled_flag){	
if(adaptive_loop_filter_enabled_flag)	
alf_param()	
if(Inter_layer_adaptive_loop_filter_enabled_flag)	
alf_param()	
}	
if(cu_qp_delta_enabled_flag)	
max_cu_qp_delta_depth	u(4)
rbsp_trailing_bits()	
}	

[0610] 표 55에서 shared_pps_info_enabled_flag의 값이 1이면 인터 레이어 ALF의 파라미터 세트를 사용할 수 있다. 표 55와 달리, 인루프 필터로서의 ALF와 별도로 인터 레이어 ALF의 적용을 위한 플래그를 사용하는 방법도 적용할 수 있음은 물론이다.

[0611] 표 55는 본 발명에 따라서 shared_pps_info_enabled_flag를 인터 레이어 ALF에 적용하는 경우, 슬라이스 데이터 선택스의 예를 나타낸 것이다.

[0612] <표 56>

	Descriptor
slice_header() {	
...	
if(adaptive_loop_filter_enabled_flag) {	
if(!shared_pps_info_enabled_flag)	
alf_param()	
alf_cu_control_param()	
}	
if(inter_layer_adaptive_loop_filter_enabled_flag){	
if(!shared_pps_info_enabled_flag)	
alf_param()	
alf_cu_control_param()	
}	
...	

[0614]

[0615] 인터 레이어 필터 파라미터 예측

[0616] 루프 필터링 과정에서 적용 가능한 필터로서, 더블록킹 필터, SAO(Sample Adaptive Offset) 그리고 Adaptive Loop Filter(ALF)의 세 가지가 있다.

[0617] 루프 필터로서, 세 가지 필터가 모두 적용될 수도 있고, 일부만 적용될 수도

있다. 예컨대, 더블록킹 필터만 적용될 수도 있고, 더블록킹 필터와 SAO만 적용될 수도 있다.

[0618] 인코딩 장치는 복원 영상이 입력 영상에 가장 가까울 수 있도록 필터 파라미터를 결정하고, 결정된 필터 파라미터를 디코딩 장치로 전송할 수 있다. 따라서, 스케일러블 코딩의 특성상 베이스 레이어와 인헨스먼트 레이어에 사용되는 영상이 상당히 유사하므로, 필터 파라미터가 두 레이어 사이에서 유사할 가능성이 높다.

[0619] 따라서, 베이스 레이어에서 사용한 파라미터를 인헨스먼트 레이어에서 재사용하는 방법을 고려할 수 있다. 이를 인터 레이어 필터 파라미터 예측이라고 한다.

[0620] 인터 레이어 필터 파라미터 예측은 도 1 내지 도 4의 예측부 혹은 인터 레이어 예측부에서 수행할 수도 있고, 인헨스먼트 레이어의 필터부에서 수행할 수도 있다.

[0621] 도 20은 본 발명에 따른 인터 레이어 필터 파라미터 예측에 관하여 개략적으로 설명하는 도면이다.

[0622] 도 20을 참조하면, 베이스 레이어의 인루프 필터링부(2020)에서 사용한 필터 파라미터를 현재 레이어의 인루프 필터링부(2010)에서 재사용할 수 있다.

[0623] SAO 파라미터는 적응 파라미터 세트(adaptive parameter set)에서 전송될 수 있다. ALF의 파라미터의 경우도 적응 파라미터 세트에서 전송될 수 있다.

[0624] 표 57은 본 발명에 따른, 스케일러블 확장에 있어서 적응 파라미터 세트의 예를 나타낸 것이다.

[0625] <표 57>

[0626]

	Descriptor
aps_rbsp_in_scalable_extension () {	
aps_id	ue(v)
aps_sample_adaptive_offset_flag	u(1)
aps_adaptive_loop_filter_flag	u(1)
if(aps_sample_adaptive_offset_flag aps_adaptive_loop_filter_flag) {	
aps_cabac_use_flag	u(1)
if(aps_cabac_use_flag) {	
aps_cabac_init_idc	ue(v)
aps_cabac_init_qp_minus26	se(v)
}	
}	
/* Insert non-CABAC stuff above this line */	
base_pred_alf_param_flag	u(1)
if(aps_adaptive_loop_filter_flag && ! base_pred_alf_param_flag) {	
alf_data_byte_count /* to enable skipping past data without parsing it */	u(8)
/* byte_align() this byte align to happen between the non-CABAC and CABAC parts of the alf_param() Once there is an all CABAC alf_param(), enable this byte_align() */	
alf_param()	
byte_align()	
}	
/* insert CABAC stuff below this line; make sure its byte-aligned */	
base_pred_sao_param_flag	u(1)
if(aps_sample_adaptive_offset_flag && ! base_pred_sao_param_flag) {	
sao_data_byte_count /* to enable skipping past data without parsing it */	u(8)
byte_align ()	
sao_param()	
/* byte_align() this final byte align unnecessary as being taken care of by rbsp_trailing_bits() */	
rbsp_trailing_bits()	
}	
}	

- [0627] **aps_id** 는 슬라이스 헤더에서 참조되는 적응 파라미터 세트(adaptation parameter set)를 식별한다. aps_id의 값은 0이상 TBD 이하의 값을 가지며, 레벨/프로파일링에 의존한다.
- [0628] **aps_sample_adaptive_offset_flag**의 값이 1이면, 현재 적응 파라미터 세트에서 참조하는 슬라이스에 대하여 SAO가 온(ON)인 것을 특징한다. aps_sample_adaptive_offset_flag 의 값이 0이면, 현재 적응 파라미터 세트에서 참조하는 슬라이스에 대하여 SAO가 오프(OFF)인 것을 특징한다. 활성화된 적응 파라미터 세트가 없다면, aps_sample_adaptive_offset_flag 값은 0으로 추정된다.
- [0629] **aps_adaptive_loop_filter_flag**의 값이 1이면, 현재 적응 파라미터 세트에서 참조하는 슬라이스에 대하여 ALF가 온(ON)인 것을 특징한다. aps_adaptive_loop_filter_flag 의 값이 0이면 현재 적응 파라미터 세트가 참조하는

- 슬라이스에 대하여 ALF가 오프(OFF)인 것을 특징한다. 활성화된 적응 파라미터 세트가 없다면, `aps_adaptive_loop_filter_flag`의 값은 0으로 추정된다.
- [0630] `aps_cabac_use_flag`의 값이 1이면 `sao_param()`과 `alf_param()`에 대하여 CABAC 디코딩 과정이 이용되며, `aps_cabac_use_flag`의 값이 0이면 `sao_param()`와 `alf_param()`에 대하여 CAVLC 디코딩 과정이 이용된다.
- [0631] `aps_cabac_init_idc`는 SAO와 ALF의 콘텍스트 변수들에 대한 초기화 프로세스에서 이용되는 초기화 테이블을 결정하는 인덱스를 특징한다. `aps_cabac_init_idc`의 값은 0 이상 2 이하의 범위에 있다.
- [0632] `aps_cabac_init_qp_minus26`는 양자화 파라미터-26값을 특징한다. 이때, 양자화 파라미터는 SAO와 ALF의 콘텍스트 변수들에 대한 초기화 프로세스에 이용된다.
- [0633] `alf_data_byte_count`와 `sao_data_byte_point`는 바이트 수(number of bytes)를 특징한다.
- [0634] `base_pred_alf_param_flag`의 값이 1이면, 베이스 레이어에서 사용된 ALF 파라미터가 현재 레이어에 사용된다는 것을 특징한다. `base_pred_alf_param_flag`의 값이 0이면 현재 레이어에 대한 ALF 파라미터가 현재 레이어에 사용된다는 것을 특징한다.
- [0635] `base_pred_sao_param_flag`의 값이 1이면, 베이스 레이어에서 사용된 SAO 파라미터가 현재 레이어에 사용된다는 것을 특징한다. `base_pred_sao_param_flag`의 값이 0이면, 현재 레이어에 대한 SAO 파라미터가 현재 레이어에 사용되지 않는다는 것을 특징한다.
- [0636]
- [0637] 인터 레이어 차분 픽처 코딩
- [0638] 베이스 레이어의 복원 영상(reconstructed picture)을 R_{BL} 이라 하고, R_{BL} 을 인헨스먼트 레이어의 해상도(resolution)에 맞추어 업샘플링 시킨 영상을 UR_{BL} 이라 하자. 그리고, 인헨스먼트 레이어의 복원 영상을 R_{EL} 이라 하자.
- [0639] 복원 영상은 인루프 필터링을 적용하기 전의 영상일 수 있다. 또한, 복원 영상은 인루프 필터(deblocking filter, sample adaptive offset filter 및/또는 adaptive loop filter) 중 일부를 적용한 후의 영상일 수도 있다. 더 나아가, 복원 영상은 인루프 필터 전부를 적용한 후의 영상일 수도 있다.
- [0640] 여기서, R_{EL} 로부터 UR_{BL} 의 값을 뺀 차분 영상을 D 라고 하면, D 영상들의 도메인(domain)에서 독립적인 부호화/복호화를 수행할 수 있다. 이 방법을 본 명세서에서는 인터 레이어 차분 픽처 코딩(Inter-layer differential picture coding) 혹은 인터 레이어 차분 모드(inter-layer differential mode: IL-Diff mode)라고 한다.
- [0641] 인터 레이어 차분 모드는 시퀀스 단위, 픽처 단위, 슬라이스 단위, LCU(Largest CU) 단위, 코딩 유닛(CU) 단위 혹은 예측 유닛(PU) 단위에서 적용될 수 있다. 인터 레이어 차분 모드를 적용하고자 하는 처리 단위에서, 인터 레이어 차분 모드를 사용할지를 알려주는 플래그가 인코딩 장치로부터 디코딩 장치로

전송될 수 있다.

- [0642] 인터 레이어 차분 모드가 적용된 처리 단위에 대해서는, 다른 스케일러빌리티(*scalability*)를 이용한 코딩(인코딩/디코딩)은 수행하지 않고, 단일 레이어 코딩(인코딩/디코딩) 방법만을 이용할 수도 있다. 이 경우, 다른 스케일러빌리티를 이용한 코딩을 수행할지 여부를 지시하기 위한 비트를 절약할 수 있다.
- [0643] 인터 레이어 차분 모드는 도 1 내지 도 4의 예측부 혹은 인터 레이어 예측부에서 수행할 수 있다. 여기서는 설명의 편의를 위해, 예측부에서 수행하는 것으로 설명한다.
- [0644] (1) 인터 레이어 차분 모드에 대한 인트라 예측 (*intra prediction for IL-Diff mode*)
- [0645] 도 21은 본 발명에 따라서 인터 레이어 차분 모드를 적용하는 경우에, 인트라 예측을 수행하는 방법을 개략적으로 설명하는 도면이다.
- [0646] 도 21을 참조하면, 인헨스먼트 레이어의 픽처(2100)는 현재 블록(2115) 전후의 복원 영역(2105)과 미복원 영역(2110)을 포함한다. 복원 영역(2105)으로부터 복원 영상 \mathbf{R}_{EL} 을 얻을 수 있다. 인헨스먼트 레이어의 픽처(2100)에 대한 복원이 완료되면, 픽처(2100)가 복원 영상 \mathbf{R}_{EL} 이 될 수 있다.
- [0647] 한편, 베이스 레이어의 복원 영상 \mathbf{R}_{BL} (2120)을 업샘플링한 영상 \mathbf{UR}_{BL} (2125)는 현재 블록(2115)에 대응하는 블록 \mathbf{P}_{BL} (2130)을 포함한다.
- [0648] 인코딩 과정에서 예측부(인코딩 장치의 예측부)는 수식 4와 같이 베이스 레이어의 복원 영상과 인헨스먼트 레이어의 복원 영상 사이의 차분 \mathbf{D} 를 유도할 수 있다.
- [0649] <수식 4>
- [0650] $\mathbf{D} = \mathbf{R}_{EL} - \mathbf{UR}_{BL}$
- [0651] 수식 4에선 \mathbf{R}_{EL} 은 아직 복원되지 않은 영역(2110)의 존재로 인해서, 디블록킹 필터, SAO 혹은 ALF 등과 같은 인루프 필터가 적용되지 않은 상태의 영상을 사용할 수도 있다.
- [0652] 베이스 레이어의 픽처(1220)는 모든 영역이 복원된 상태이기 때문에, \mathbf{R}_{BL} 은 인루프 필터가 적용된 상태의 복원 영상일 수도 있고, 인루프 필터의 일부가 적용된 상태의 복원 영상일 수도 있으며, 인루프 필터가 적용되지 않은 상태의 복원 영상일 수도 있다.
- [0653] 예측부는 차분 영상 \mathbf{D} (2140)에서, 미복원 영역(2150)을 제외하고, 복원 영역(2145)의 픽셀값들을 참조하여, 현재 블록(2155)에 대하여 인트라 예측을 수행할 수 있다.
- [0654] 디코딩 과정에서 예측부(디코딩 장치의 예측부)는 현재 블록과 동일 위치에 존재하는 \mathbf{UR}_{BL} 내의 블록값 \mathbf{P}_{BL} 을 이용하여 수식 5와 같이 현재 블록을 복원할 수 있다.
- [0655] <수식 5>
- [0656] $\mathbf{R}_{EL} = \mathbf{P}_D + \mathbf{P}_{BL} + \mathbf{RES}$

- [0657] 수식 5에서 P_D 는 차분 영상 D 의 복원된 영역으로부터 인트라 예측을 수행함에 따라서 생성된 예측 블록이며, RES 는 레지듀얼 블록이다.
- [0658] (2) 인터 에이어 차분 모드에 대한 인터 예측 (inter prediction for IL-Diff mode)
- [0659] 인터 레이어 차분 모드를 적용할 때, 현재 블록에 대한 인터 예측을 수행하기 위해 예측부는 현재 픽처의 참조 픽처에 대한 차분 영상 D_R 을 생성한다. 예컨대, 예측부는 참조 픽처의 인헨스먼트 레이어 픽처에 대한 복원 영상과 참조 픽처의 베이스 레이어 픽처에 대한 복원 영상을 이용하여, 현재 픽처의 참조 픽처에 대한 차분 영상 D_R 을 생성한다.
- [0660] 예측부는 참조 픽처의 차분 영상 D_R 을 기반으로 현재 블록의 차분 영상 도메인에서 예측 블록 P_D 를 생성할 수 있다.
- [0661] 예측부는 예측 블록을 이용하여 수식 6과 같이 현재 블록을 복원할 수 있다.
- [0662] <수식 6>
- [0663] $R_{EL} = P_D + P_{BL} + RES$
- [0664] 수식 6에서 R_{EL} 은 인헨스먼트 레이어에서 복원된 현재 블록이다. P_{BL} 은 UR_{BL} 내에서 현재 블록과 동일 위치에 존재하는 블록이며, RES 는 레지듀얼 블록이다.
- [0665] 참조 영상의 차분 영상 D_R 은 미리 생성되어 DPB(Decoded Picture Buffer)에 저장될 수 있다. DPB는 도 1 및 도 3에서 설명한 메모리에 대응할 수 있다.
- [0666] 또한, 참조 영상의 차분 영상 D_R 은 R_{EL} 을 생성할 때마다, 현재 블록의 움직임 정보에 의해 현재 블록의 복원을 위해 필요한 위치로 특정된 블록에 대해서 산출될 수도 있다.
- [0667] 인터 레이어 차분 모드에 대한 인트라 예측의 경우와 달리 인터 레이어 차분 모드에 대한 인터 예측에서는, 참조 픽처에 대한 차분 영상을 생성할 때 인헨스먼트 레이어에서 복원된 참조 픽처로서 인루프 필터가 적용되지 않은 상태의 복원 픽처뿐만 아니라, 인루프 필터의 일부 또는 전부가 적용된 상태의 복원 픽처를 이용할 수도 있다.
- [0668] 본 명세서에서는 설명의 편의를 위해, 스케일러블 비디오 코딩이 지원되는 멀티 레이어 구조에서 각 레이어별로 특정 시점(예컨대, POC(Picture Order Count) 혹은 AU(Access Unit))에 복원되는 샘플들의 어레이를 '픽처'라고 표현하였다.
- [0669] 이와 관련하여, 디코딩 되어 출력되는 레이어(현재 레이어)에서 특정 시점에 복원된 혹은 복원되는 전체 샘플 어레이를 픽처라고 하여, 참조되는 레이어의 복원된 혹은 복원되는 샘플 어레이와 구분할 수도 있다. 참조되는 레이어에서 특정 시점에 복원된 혹은 복원되는 샘플 어레이는 리프리젠테이션(representation), 참조 레이어 픽처, 참조 레이어 샘플 어레이, 참조 레이어 텍스처 등으로 칭할 수도 있다. 이 경우, 하나의 AU에 대해서는 현재 레이어에서 복원된 하나의 디코딩(인코딩)된 픽처가 출력될 수 있다.
- [0670]
- [0671] 상술한 예시적인 시스템에서, 방법들은 일련의 단계 또는 블록으로써 순서도를

기초로 설명되고 있지만, 본 발명은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 상술한 실시예들은 다양한 양태의 예시들을 포함한다. 따라서, 본 발명은 이하의 특허청구범위 내에 속하는 모든 다른 교체, 수정 및 변경을 포함한다고 할 것이다.

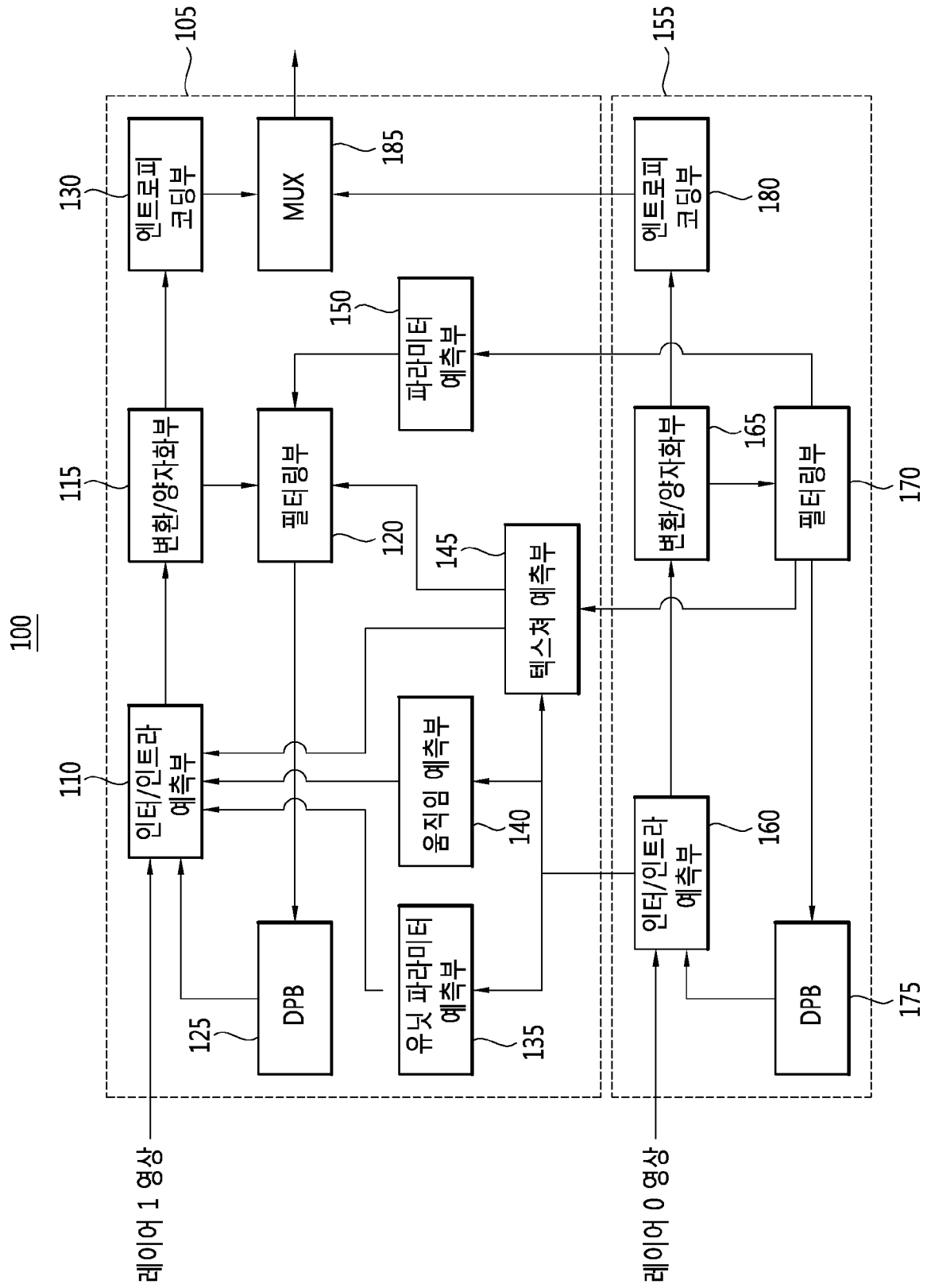
청구범위

- [청구항 1] 참조 레이어로부터 인터 레이어 움직임 정보를 유도하는 단계; 및 상기 인터 레이어 움직임 정보를 이용하여 현재 레이어 내 현재 블록에 대한 예측을 수행하는 단계를 포함하며, 상기 인터 레이어 움직임 정보는 상기 참조 레이어로부터 유도되는 인터 레이어 움직임 벡터를 포함하고, 상기 인터 레이어 움직임 벡터는 상기 참조 레이어의 움직임 벡터를 상기 참조 레이어와 상기 현재 레이어 간 해상도 비율에 따라서 스케일링하여 상기 인터 레이어 움직임 벡터를 유도하는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 2] 제1항에 있어서, 상기 참조 레이어의 움직임 벡터는, 상기 참조 레이어 내에서 상기 현재 블록을 특정하는 위치에 대응하는 참조 위치의 움직임 벡터인 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 3] 제2항에 있어서, 상기 현재 블록을 특정하는 위치는 상기 현재 블록의 좌상단 위치인 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 4] 제2항에 있어서, 상기 참조 위치는, 상기 현재 블록을 특정하는 위치를 상기 참조 레이어와 상기 현재 레이어 간 해상도 비율에 따라서 스케일링하여 유도되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 5] 제2항에 있어서, 상기 참조 위치의 움직임 벡터는, 상기 참조 위치를 포함하는 예측 블록의 움직임 벡터인 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 6] 제1항에 있어서, 상기 예측을 수행하는 단계에서는, 상기 현재 블록에 대한 예측 모드에 따라서, 상기 인터 레이어 움직임 정보를 포함하는 후보 리스트를 구성하고, 상기 후보 리스트 내 후보들 중에서 선택된 움직임 정보를 이용하여 상기 현재 블록에 대한 예측을 수행하는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 7] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 머지 모드이며, 상기 인터 레이어 움직임 정보는 머지 모드에 대한 후보로서 상기 후보 리스트의 제일 앞에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 8] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 머지 모드이며, 상기 인터 레이어 움직임 정보는 머지 모드에 대한 후보로서 상기

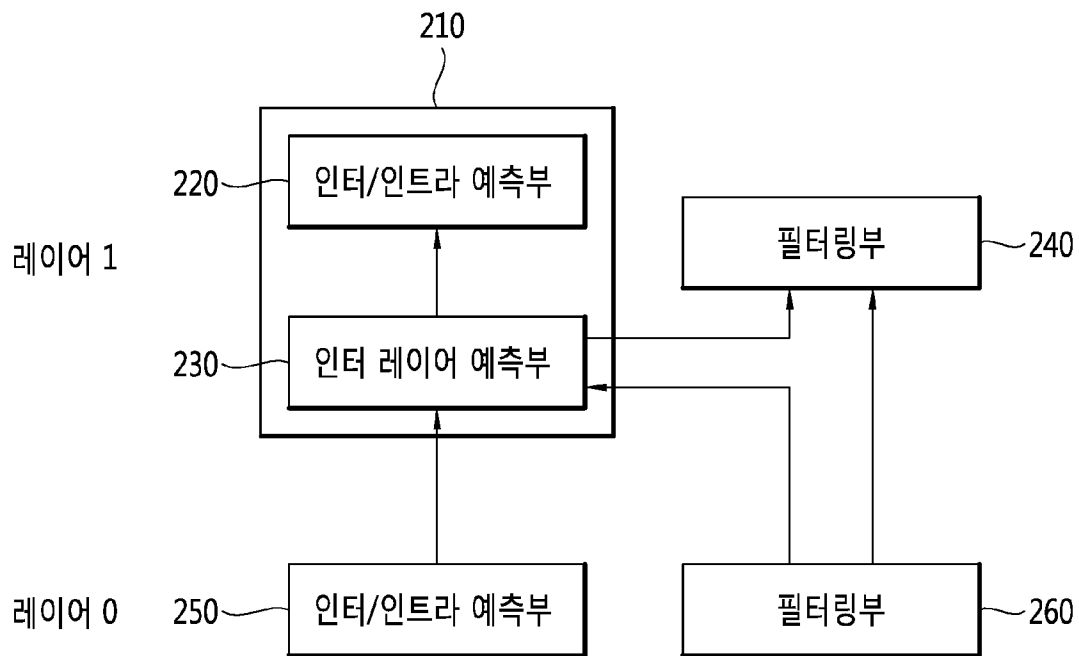
- 후보 리스트의 마지막에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 9] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 머지 모드이며,
상기 인터 레이어 움직임 정보는 머지 모드에 대한 후보로서 상기 현재 레이어의 시간적 후보보다 나중에 상기 후보 리스트에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 10] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 머지 모드이며,
상기 인터 레이어 움직임 정보는 머지 모드에 대한 후보로서 상기 현재 레이어의 공간적 후보들보다 나중에 상기 후보 리스트에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 11] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 MVP(Motion Vector Predictor)를 이용하는 모드이며,
상기 인터 레이어 움직임 벡터는 상기 현재 블록에 대한 후보 MVP로서 상기 후보 리스트의 제일 앞에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 12] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 MVP(Motion Vector Predictor)를 이용하는 모드이며,
상기 인터 레이어 움직임 벡터는 상기 현재 블록에 대한 후보 MVP로서 상기 후보 리스트의 마지막에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 13] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 MVP(Motion Vector Predictor)를 이용하는 모드이며,
상기 인터 레이어 움직임 벡터는 상기 현재 블록에 대한 후보 MVP로서 상기 현재 블록의 좌측에 위치하는 공간적 후보보다 나중에 상기 후보 리스트에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 14] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 MVP(Motion Vector Predictor)를 이용하는 모드이며,
상기 인터 레이어 움직임 벡터는 상기 현재 블록에 대한 후보 MVP로서 상기 현재 블록의 상측에 위치하는 공간적 후보보다 나중에 상기 후보 리스트에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 15] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 MVP(Motion Vector Predictor)를 이용하는 모드이며,
상기 인터 레이어 움직임 벡터는 상기 현재 블록에 대한 후보 MVP로서 상기 현재 레이어의 공간적 후보들보다 나중에 상기

- 후보 리스트에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 16] 제6항에 있어서, 상기 현재 블록에 대한 예측 모드는 MVP(Motion Vector Predictor)를 이용하는 모드이며, 상기 인터 레이어 움직임 벡터는 상기 현재 블록에 대한 후보 MVP로서 스케일링된 후에 상기 후보 리스트에 추가되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 17] 제16항에 있어서, 상기 참조 레이어의 움직임 벡터는, 상기 참조 레이어 내에서 상기 현재 블록을 특징하는 위치에 대응하는 참조 위치를 포함하는 참조 블록의 움직임 벡터이며, 상기 예측을 수행하는 단계에서는, 상기 현재 블록의 참조 픽처와 상기 참조 블록의 참조 픽처가 상이한 경우에, 상기 인터 레이어 움직임 벡터를 스케일링하여 상기 후보 리스트에 추가하는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 18] 제16항에 있어서, 상기 인터 레이어 움직임 벡터는, 현재 픽처와 상기 현재 픽처의 참조 픽처 간 POC(Picture Order Count) 차이; 및 상기 픽처와 상기 참조 레이어 내에서 상기 인터 레이어 움직임 정보가 유도되는 픽처의 참조 픽처 간 POC 차이에 기반하여 스케일링 되는 것을 특징으로 하는 인터 레이어 예측 방법.
- [청구항 19] 참조 레이어에 대한 예측을 수행하는 제1 예측부; 및 상기 제1 예측부에 의한 예측에 기반한 인터 레이어 움직임 정보를 이용하여 상기 현재 레이어 내 현재 블록에 대한 예측을 수행하는 제2 예측부를 포함하며, 상기 인터 레이어 움직임 정보는 상기 참조 레이어로부터 유도되는 인터 레이어 움직임 벡터를 포함하고, 상기 제2 예측부는 상기 참조 레이어의 움직임 벡터를 상기 참조 레이어와 상기 현재 레이어 간 해상도 비율에 따라서 스케일링하는 것을 특징으로 하는 스케일러블 비디오 디코딩 장치.
- [청구항 20] 제19항에 있어서, 상기 제2 예측부는, 상기 현재 블록에 대한 예측 모드에 따라서, 상기 인터 레이어 움직임 정보를 포함하는 후보 리스트를 구성하고, 상기 후보 리스트 내 후보들 중에서 선택된 움직임 정보를 이용하여 상기 현재 블록에 대한 예측을 수행하는 것을 특징으로 하는 스케일러블 비디오 디코딩 장치.

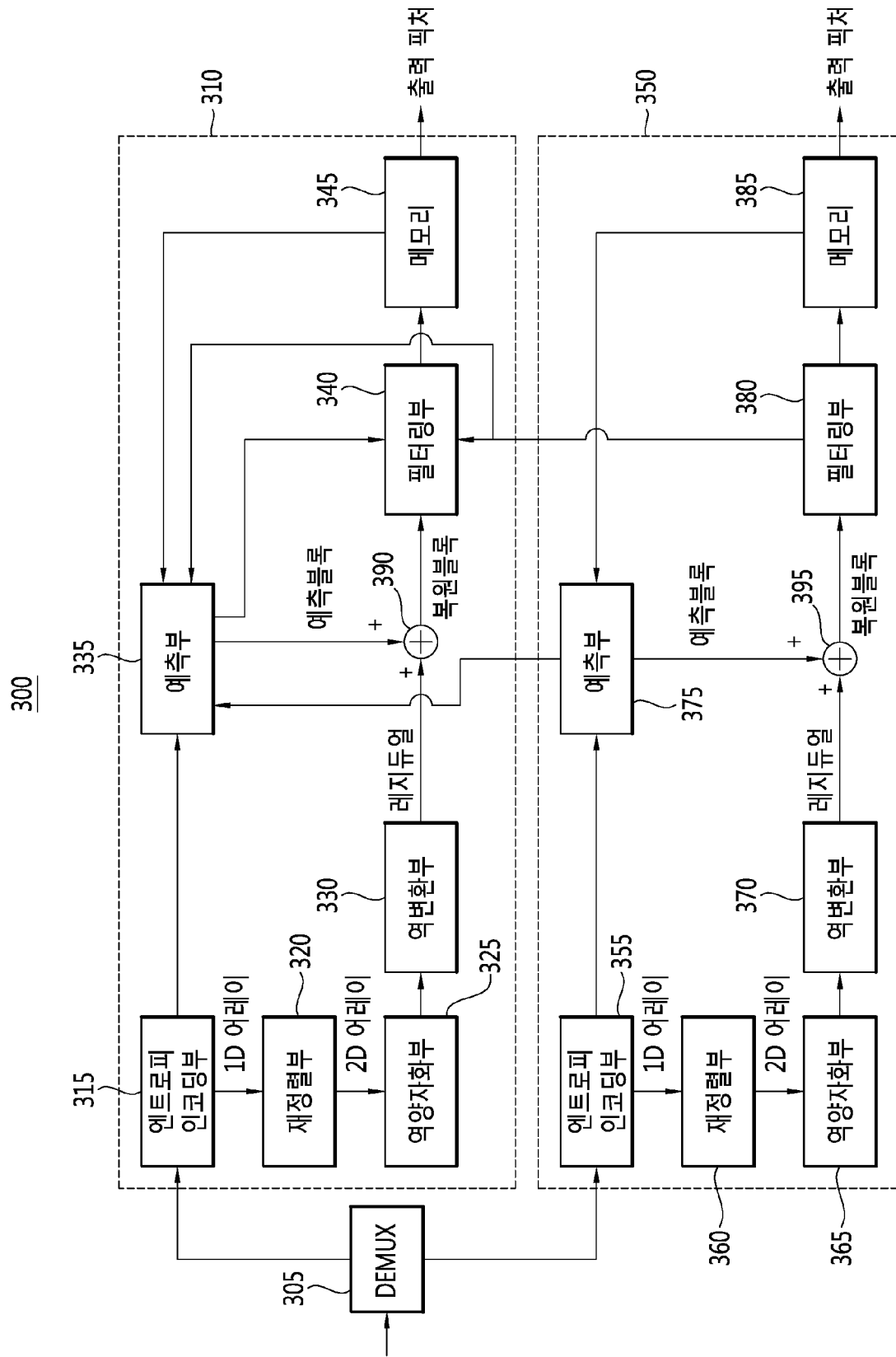
[Fig. 1]



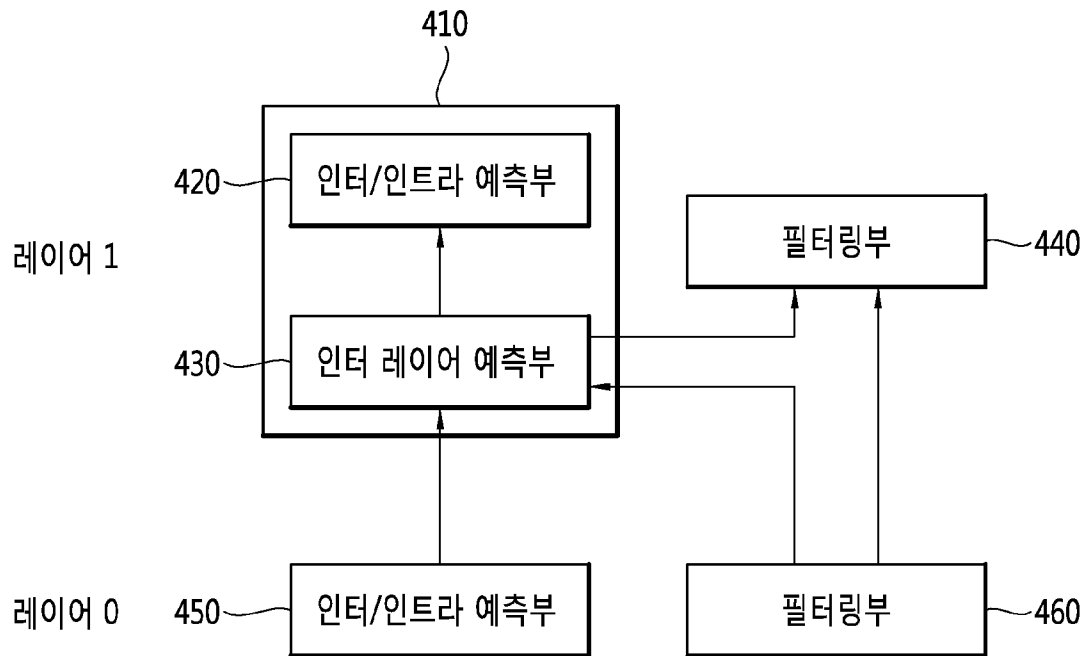
[Fig. 2]



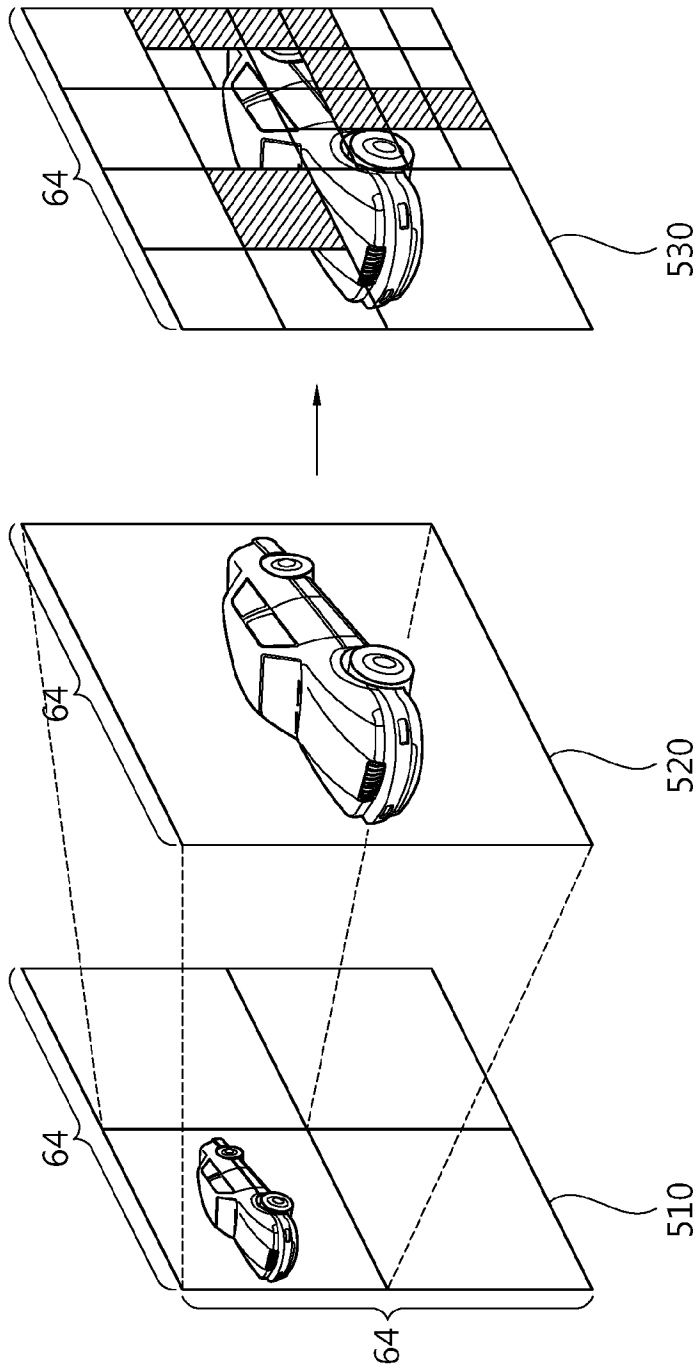
[Fig. 3]



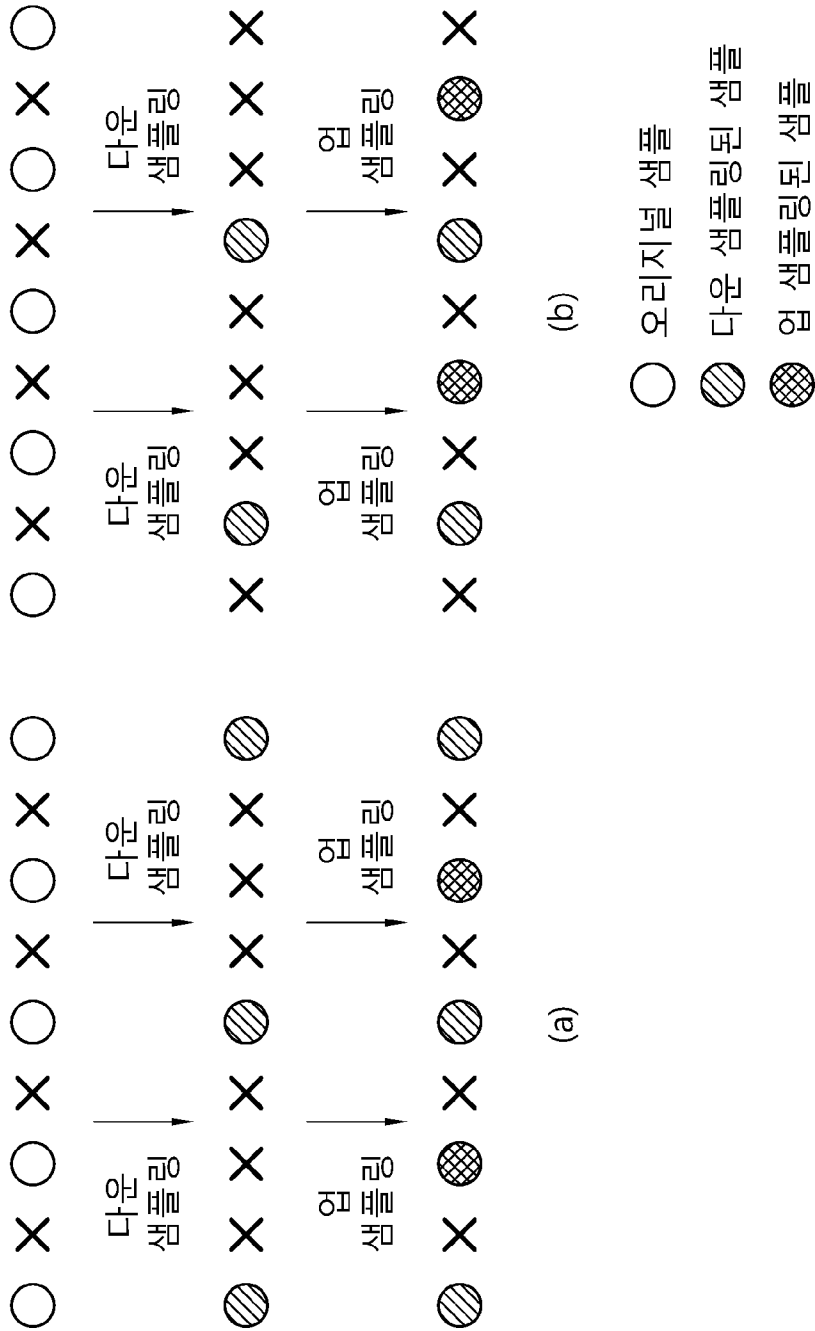
[Fig. 4]



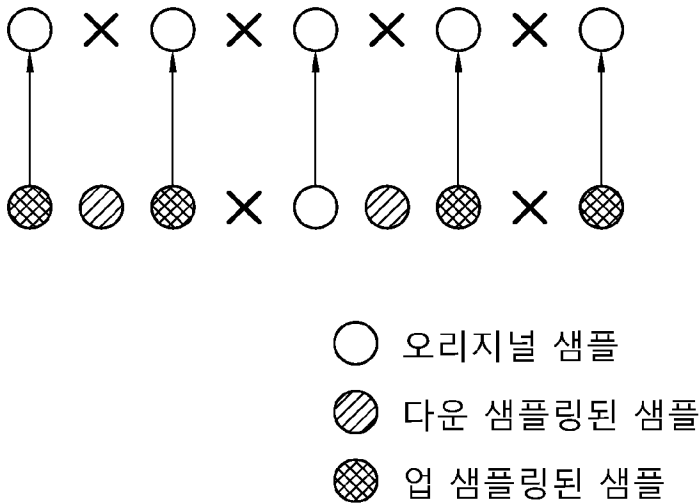
[Fig. 5]



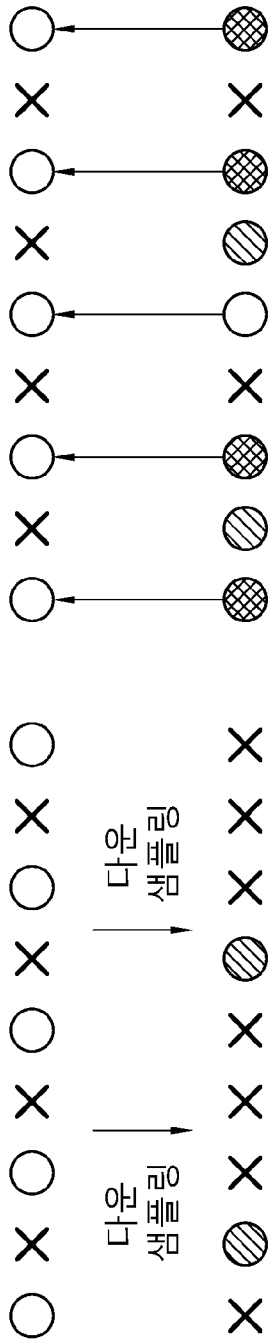
[Fig. 6]



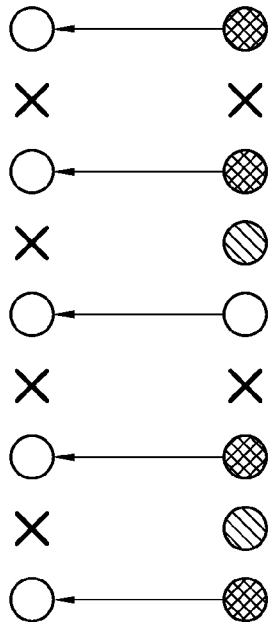
[Fig. 7]



[Fig. 8]



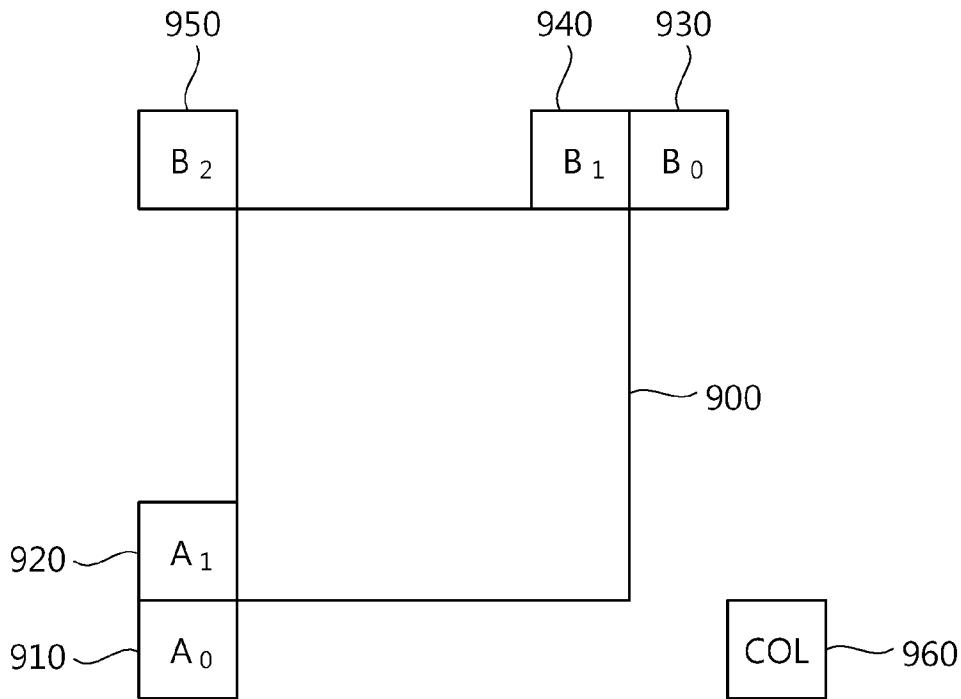
(a)



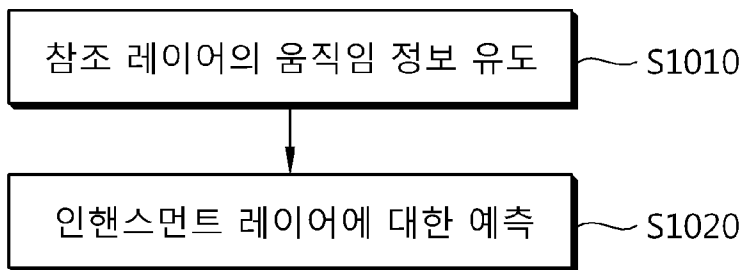
(b)

- 오리지널 샘플
- ⊗ 다운 샘플링된 샘플
- ⊘ 다운 샘플링된 샘플

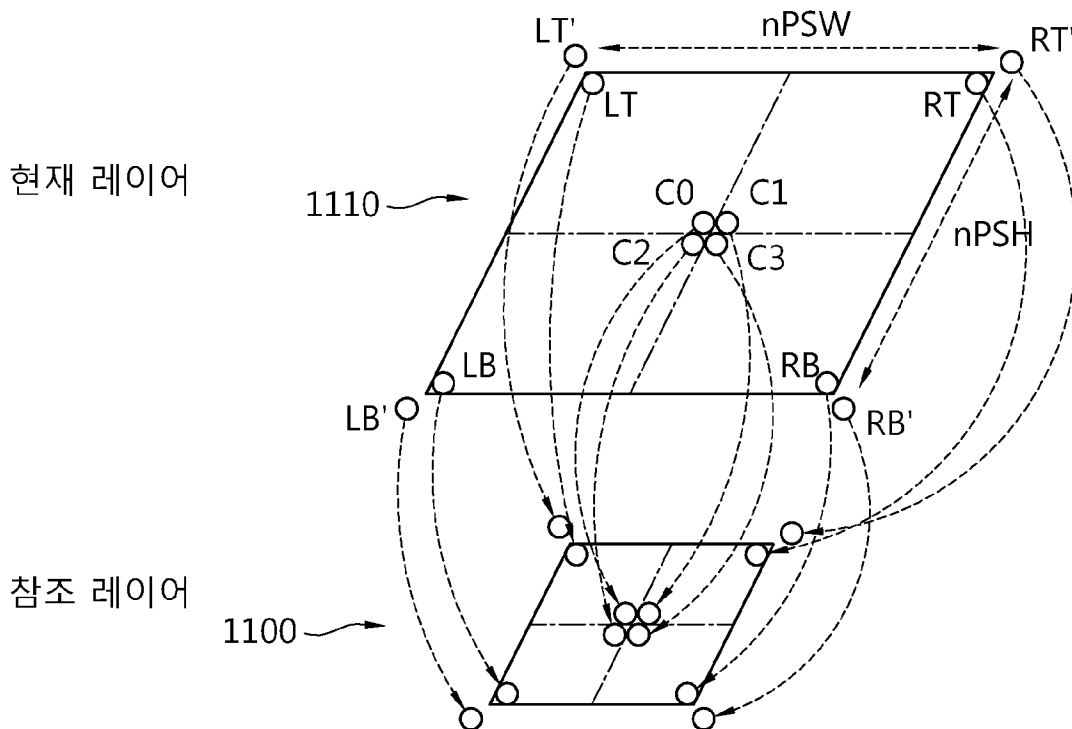
[Fig. 9]



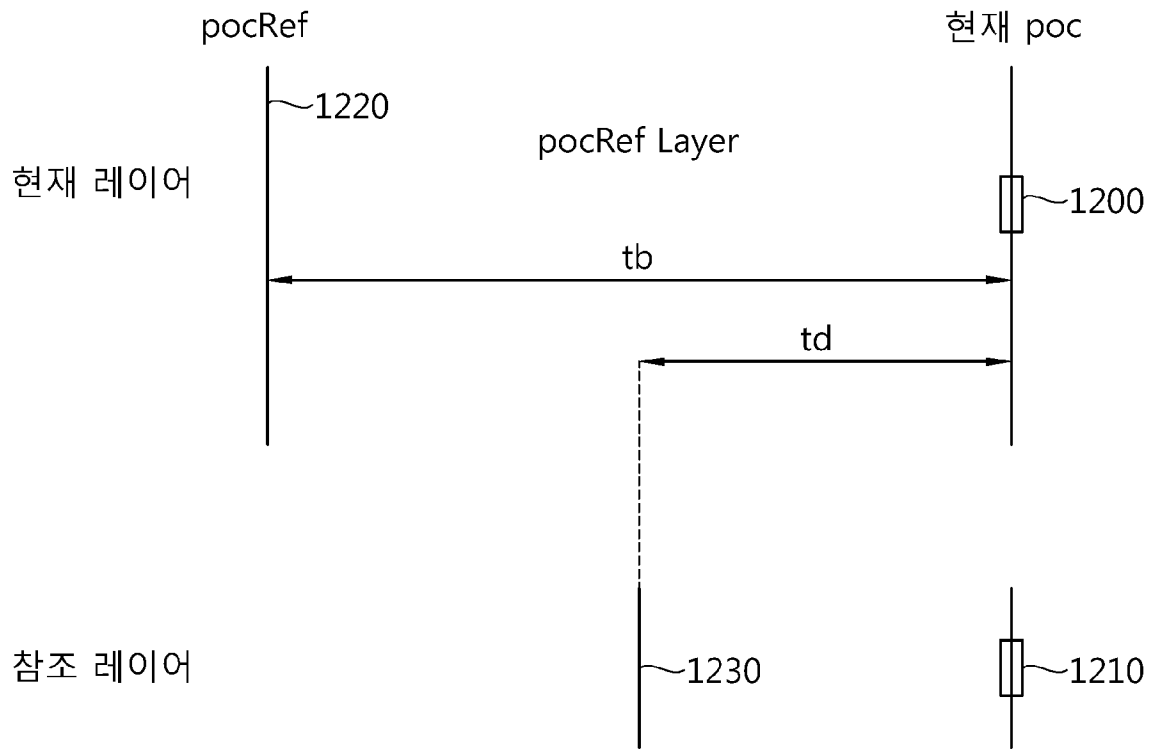
[Fig. 10]



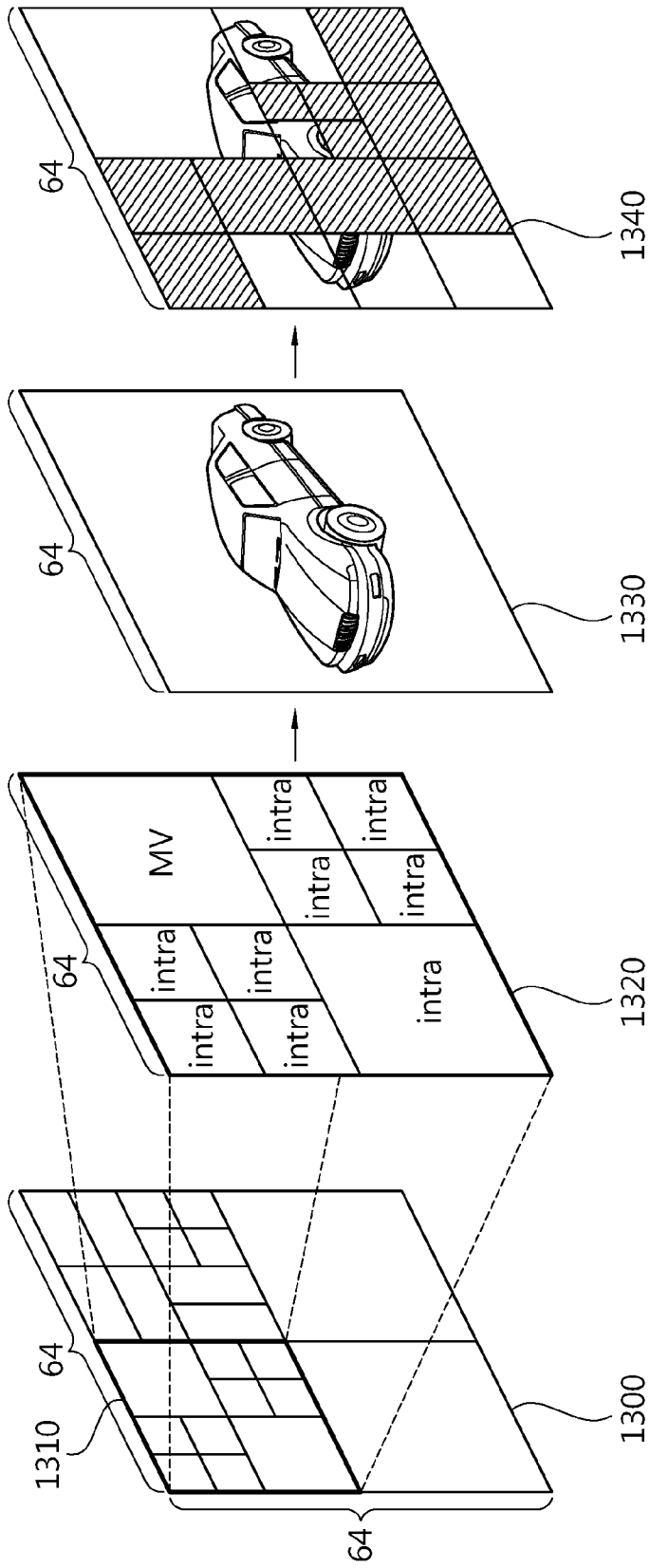
[Fig. 11]



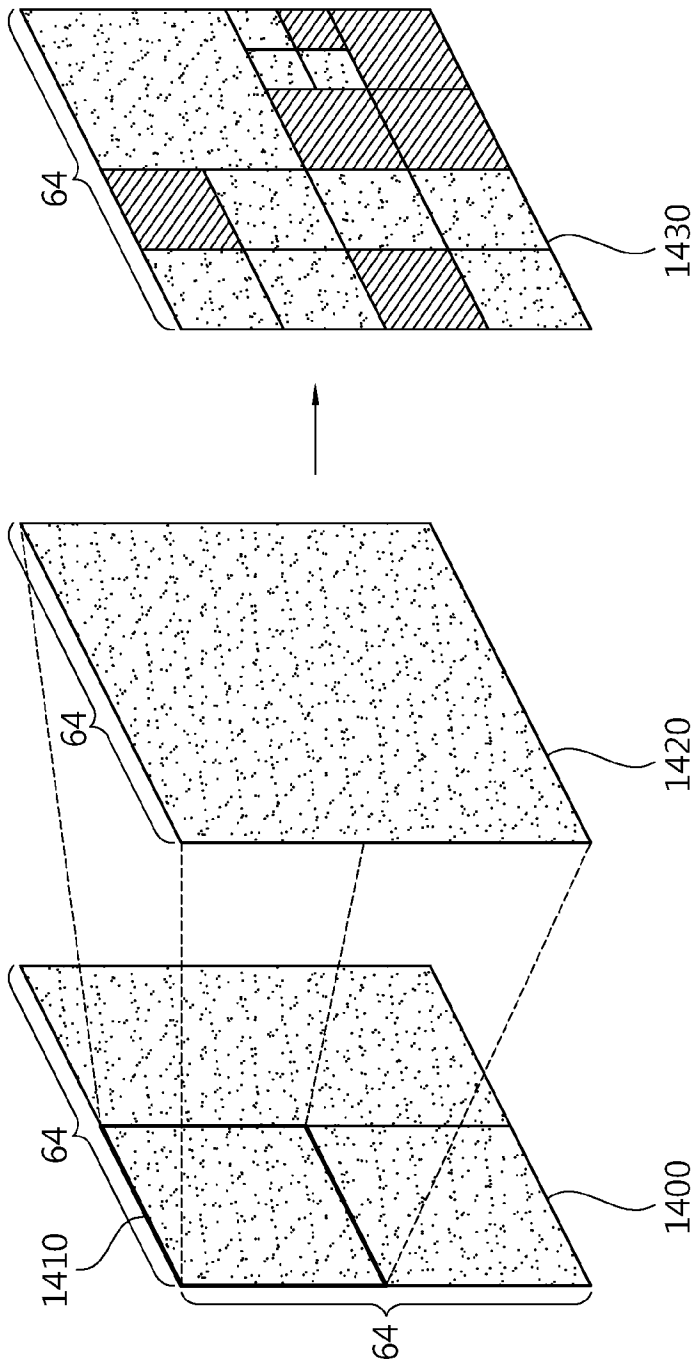
[Fig. 12]



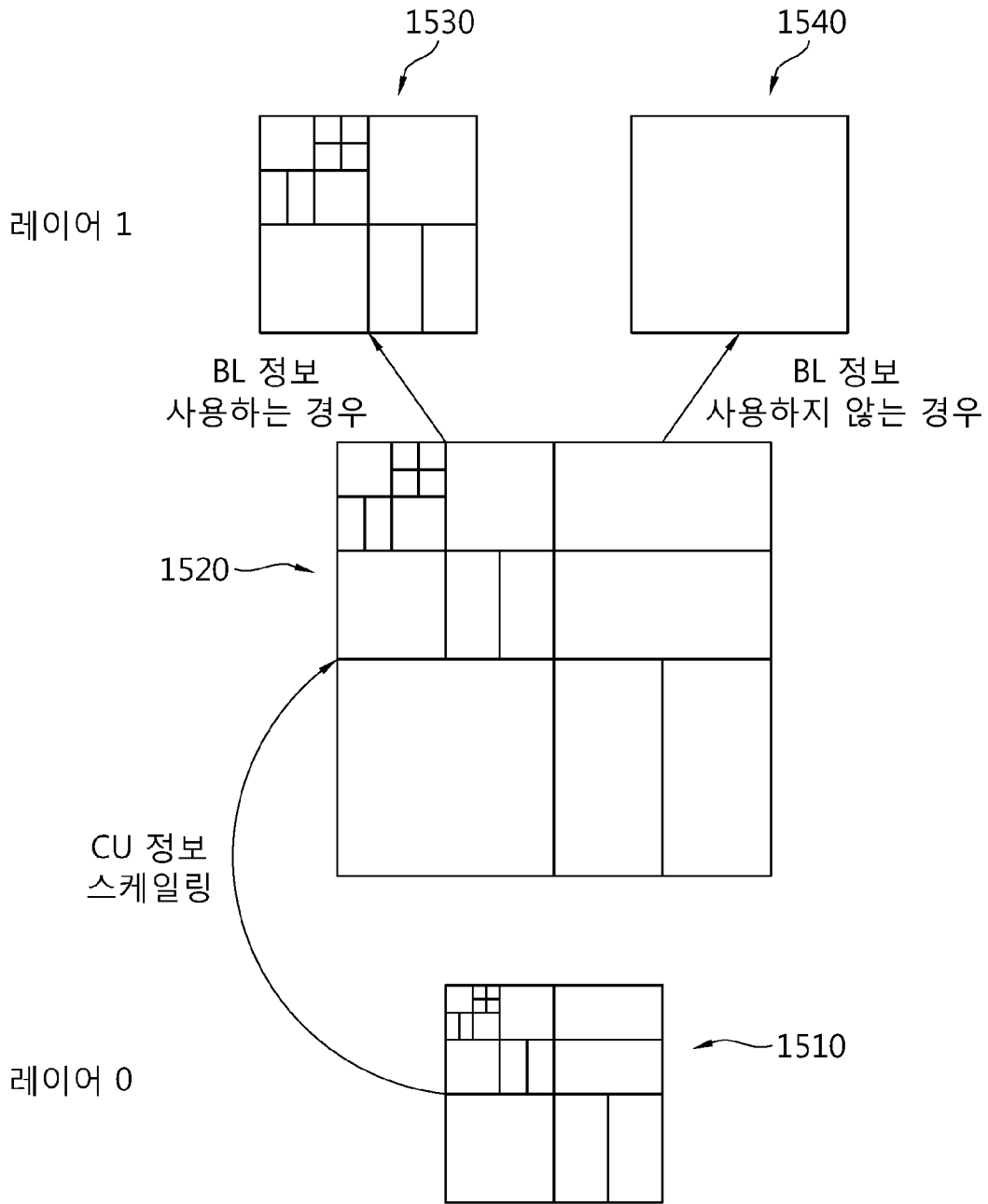
[Fig. 13]



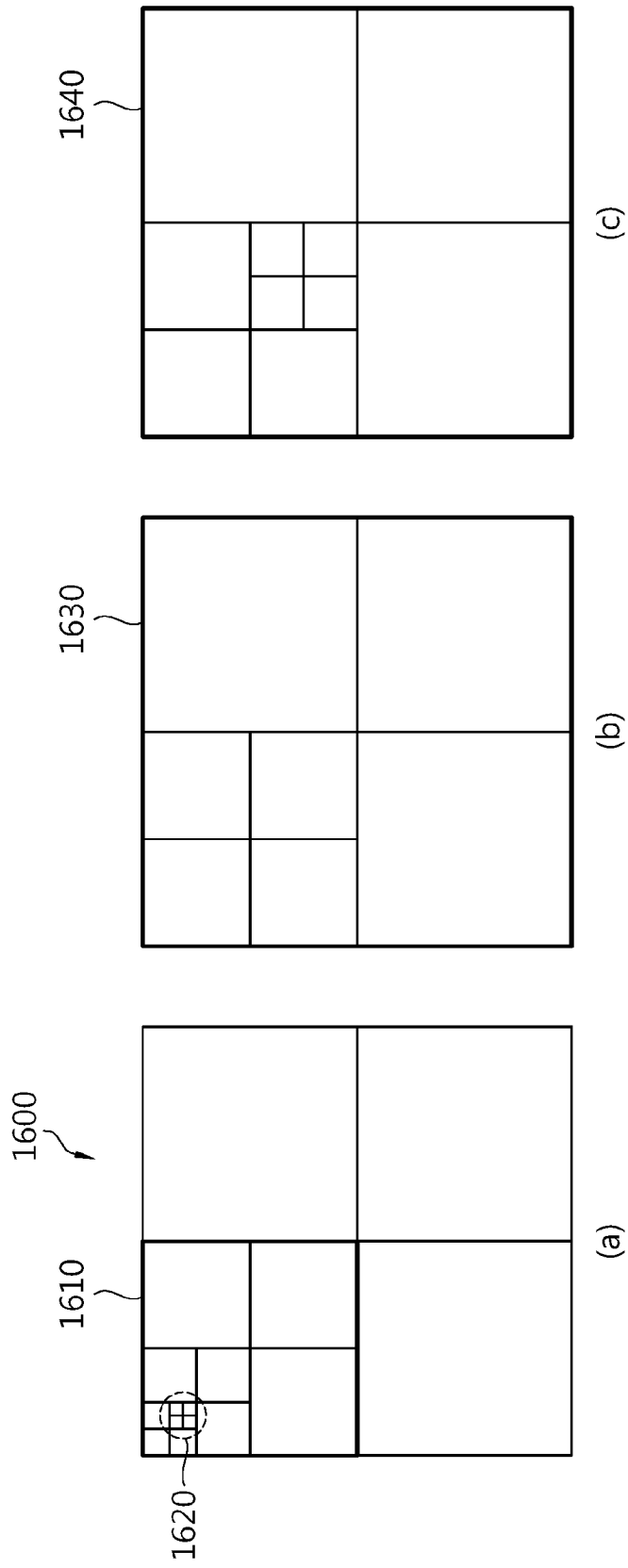
[Fig. 14]



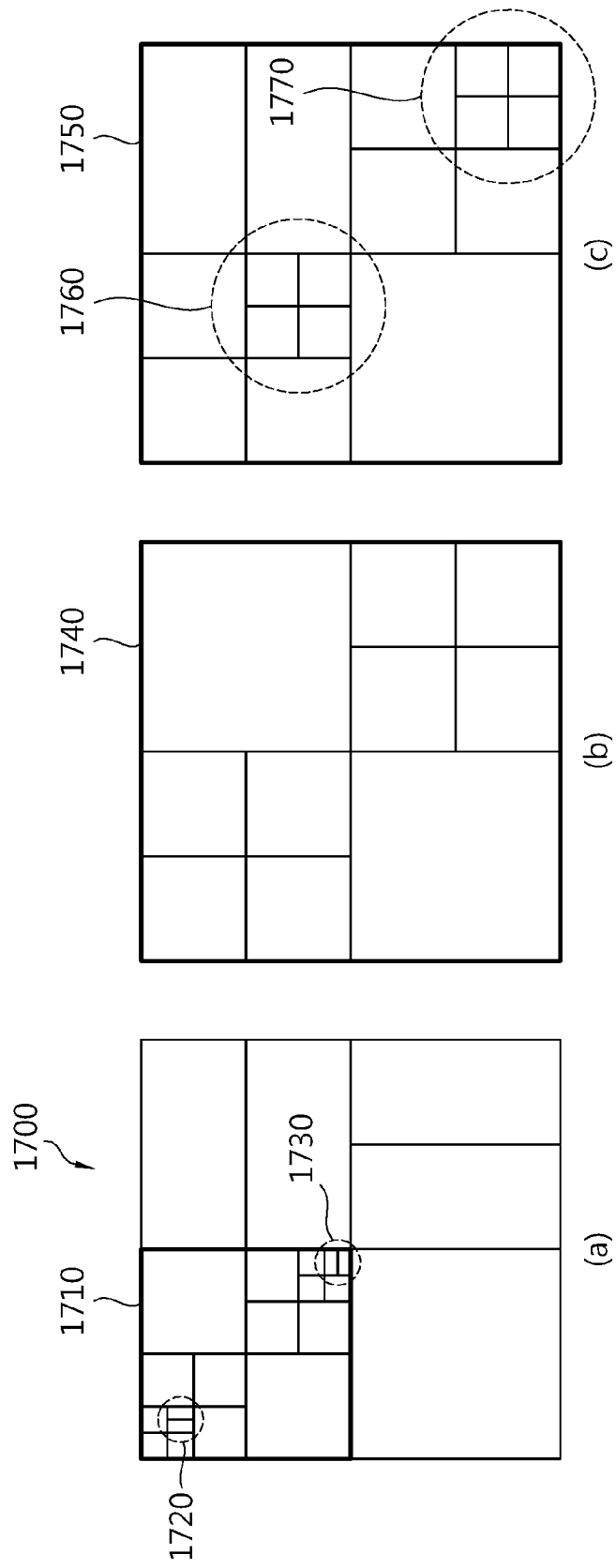
[Fig. 15]



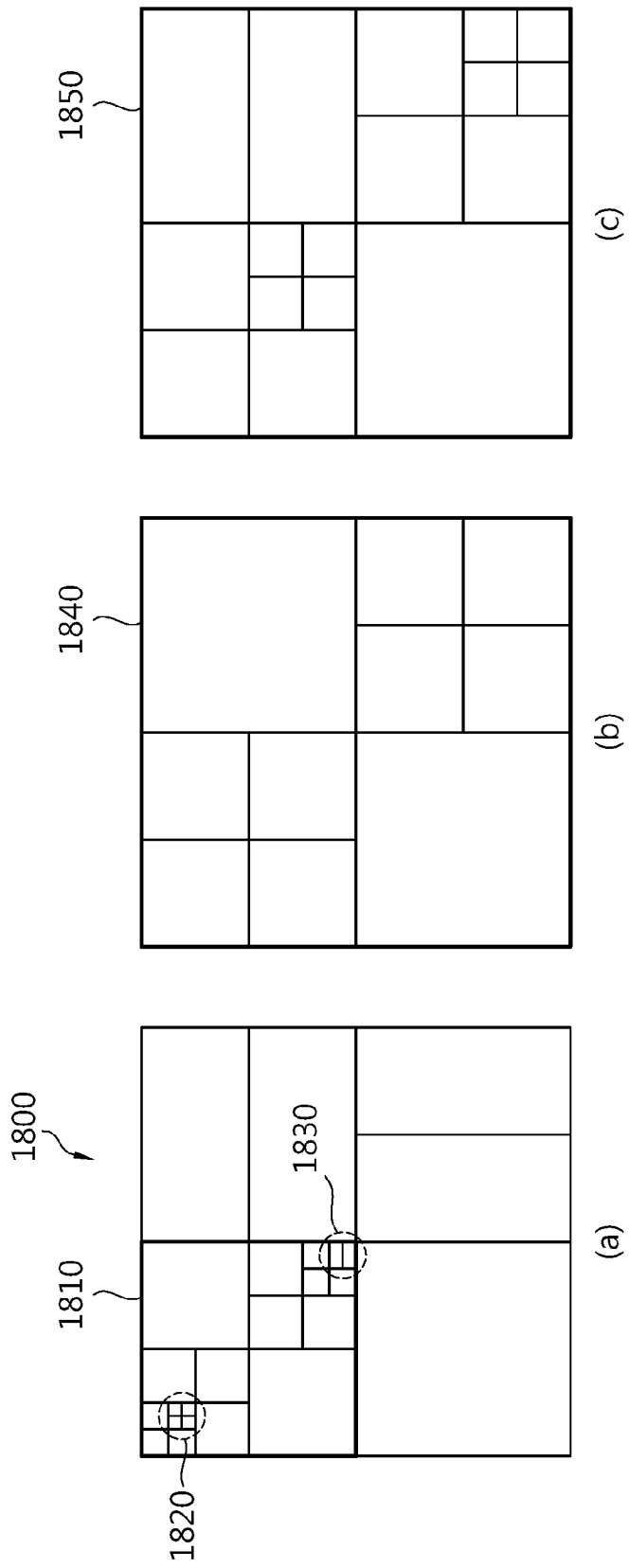
[Fig. 16]



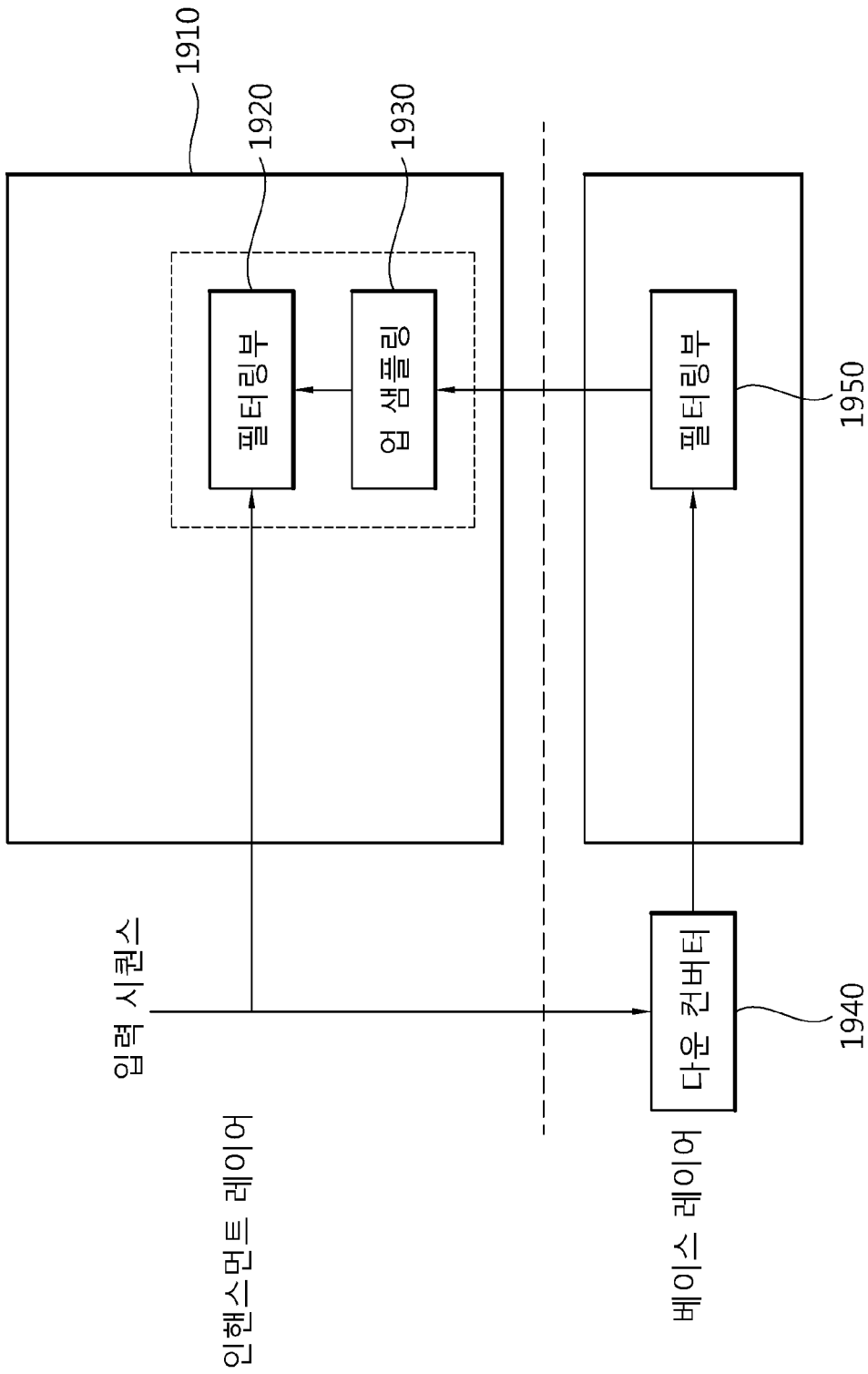
[Fig. 17]



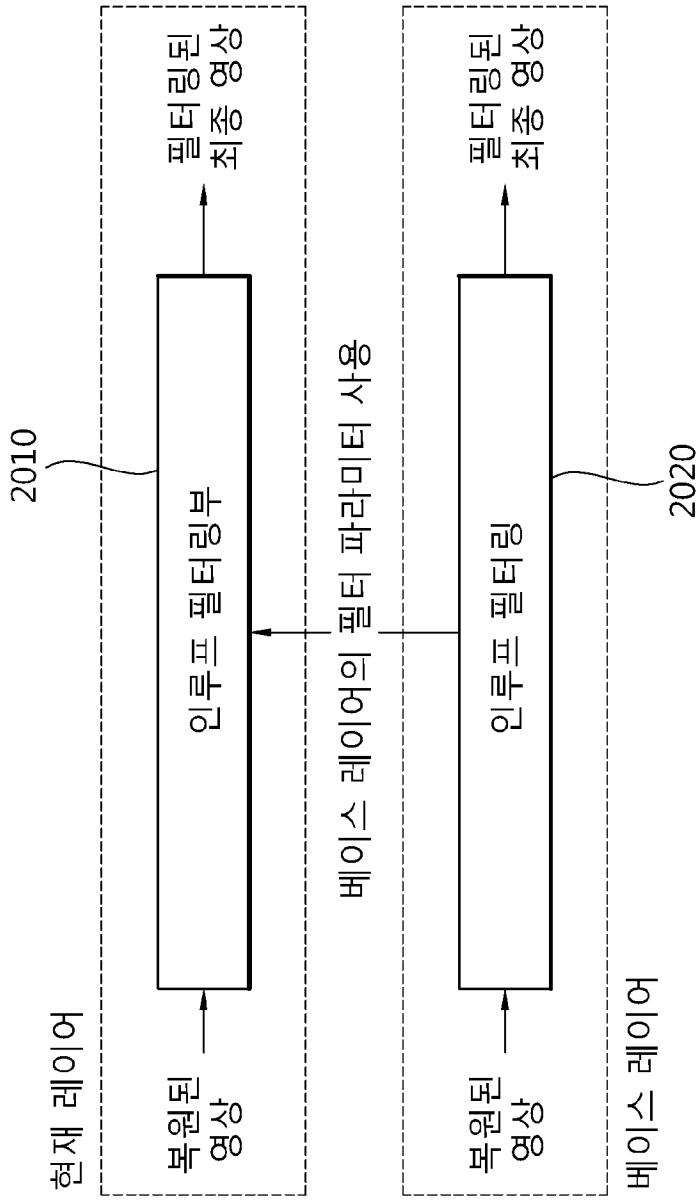
[Fig. 18]



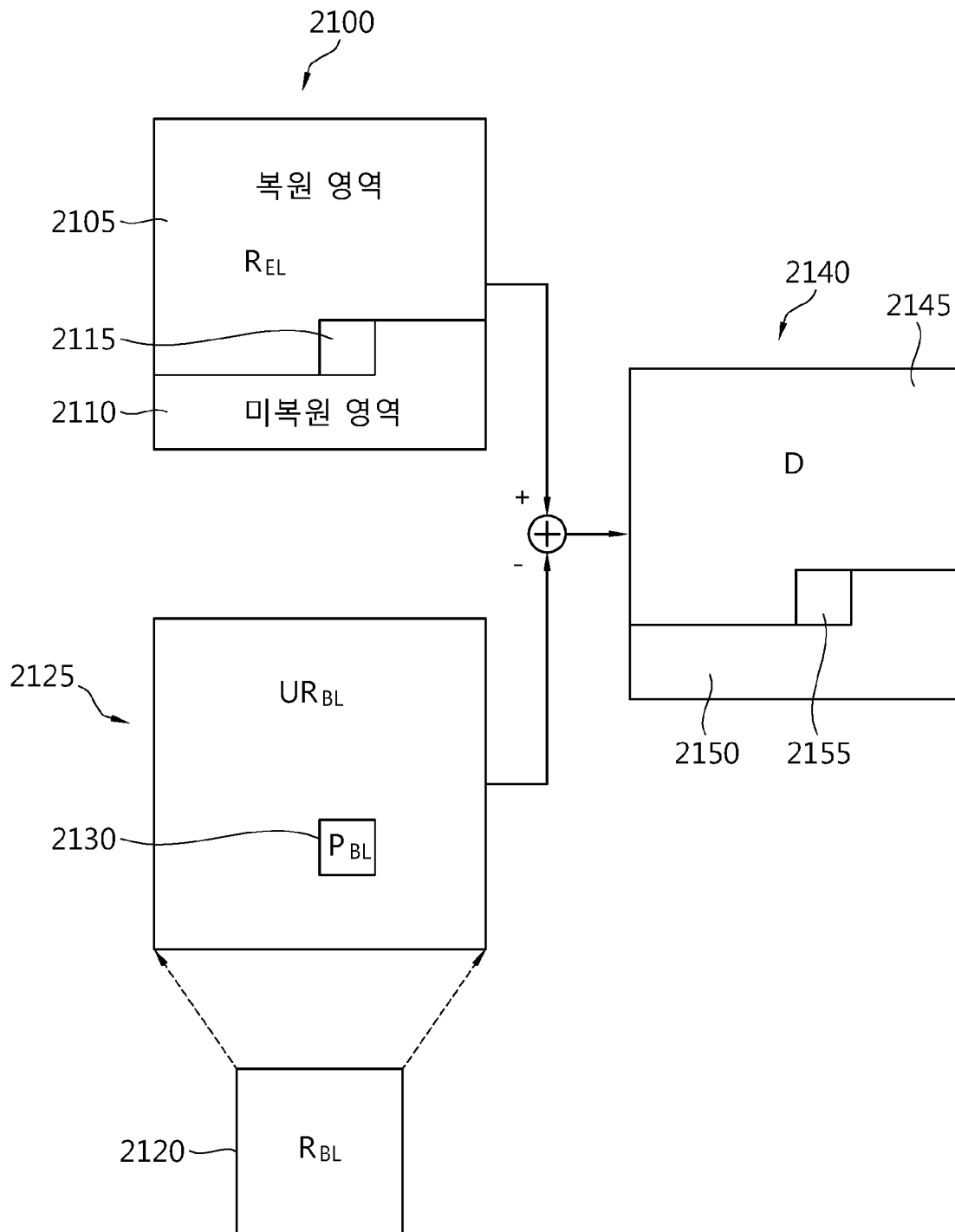
[Fig. 19]



[Fig. 20]



[Fig. 21]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2013/001659

A. CLASSIFICATION OF SUBJECT MATTER

H04N 7/32(2006.01)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 7/32; G06K 9/36; H04N 7/26; H04N 7/12

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: resolution, motion vector, motion information, scaling, reference layer, current layer, interlayer prediction

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 1694074 A1 (THOMSON LICENSING) 23 August 2006 See paragraphs [0005], [0009] and [0025]; claims 1 and 9; and figure 1.	1-5,19
A		6-18,20
Y	WOLF, Koen De et al. "Adaptive Residual Interpolation: a Tool for Efficient Spatial Scalability in Digital Video Coding." Proceedings of the 2006 International Conference on Image Processing, Computer Vision, & Pattern Recognition. Las Vegas, Nevada, USA. 26-29 June 2006, vol. 1 < http://www.researchgate.net/publication/220808864_Adaptive_Residual_Interpolation_a_Tool_for_Efficient_Spatial_Scalability_in_Digital_Video_Coding?ev=sim_pub > See page 3, left column, line 46 - page 3, right column, line 15.	1-5,19
A	KR 10-2009-0012986 A (Information and Communications University Industry-Academy Cooperation et al.) 04 February 2009 See paragraphs [0023]-[0041]; and figures 1-3 and 8.	1-20
A	US 2011-0038421 A1 (SCHWARZ, Heiko et al.) 17 February 2011 See paragraphs [0071]-[0083]; and figures 1a and 2.	1-20
A	US 2005-0190979 A1 (LI, Shipeng et al.) 01 September 2005 See paragraphs [0003]-[0008]; claims 1-2; and figures 2-10.	1-20

 Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

24 JUNE 2013 (24.06.2013)

Date of mailing of the international search report

24 JUNE 2013 (24.06.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Seonsa-ro, Daejeon 302-701,
Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2013/001659

Patent document cited in search report	Publication date	Patent family member	Publication date
EP 1694074 A1	23.08.2006	CN 101073265 A	14.11.2007
		CN 101073265 B	22.08.2012
		CN 101204092 A	18.06.2008
		CN 101204092 B	03.11.2010
		CN 101213840 A	02.07.2008
		CN 101213840 B	02.02.2011
		EP 1694075 A1	23.08.2006
		EP 1817914 A1	15.08.2007
		EP 1849303 A2	31.10.2007
		EP 1894412 A1	05.03.2008
		FR 2879066 A1	09.06.2006
		FR 2879066 B1	06.04.2007
		JP 2008-530926 A	07.08.2008
		JP 2008-530927 A	07.08.2008
		JP 5065051 B2	31.10.2012
		KR 10-1233854 B1	15.02.2013
		KR 10-2007-0103459 A	23.10.2007
		US 2008-0267291 A1	30.10.2008
		US 2009-0028245 A1	29.01.2009
		US 8345762 B2	01.01.2013
WO 2006-058921 A1	08.06.2006		
WO 2006-087314 A1	24.08.2006		
KR 10-2009-0012986 A	04.02.2009	KR 10-0944333 B1	02.03.2010
US 2011-0038421 A1	17.02.2011	CN 101044762 A	26.09.2007
		CN 101095357 A	26.12.2007
		CN 101095357 B	15.06.2011
		CN 101854550 A	06.10.2010
		CN 101854550 B	13.02.2013
		CN 101931810 A	29.12.2010
		CN 101931810 B	19.12.2012
		CN 101931811 A	29.12.2010
		CN 101931811 B	10.10.2012
		CN 101931812 A	29.12.2010
		CN 101931812 B	21.12.2011
		CN 101931813 A	29.12.2010
		CN 101931813 B	31.10.2012
		CN 101938651 A	05.01.2011
		CN 101938651 B	31.10.2012
		CN 101959066 A	26.01.2011
		CN 101959066 B	10.10.2012
		CN 101980535 A	23.02.2011
		CN 101980535 B	12.12.2012
		CN 102055985 A	11.05.2011
CN 102055985 B	05.09.2012		
CN 102055986 A	11.05.2011		
CN 102055986 B	21.11.2012		
CN 102082952 A	01.06.2011		

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2013/001659

Patent document cited in search report	Publication date	Patent family member	Publication date
		CN 102082952 B	08.08.2012
		CN 102098514 A	15.06.2011
		DE 102004059978 A1	27.04.2006
		DE 102004059978 B4	07.09.2006
		DE 102004059993 A1	27.04.2006
		DE 102004059993 B4	31.08.2006
		EP 1800488 A1	27.06.2007
		EP 1800490 A1	27.06.2007
		EP 2538677 A1	26.12.2012
		EP 2538678 A1	26.12.2012
		JP 2008-517498 A	22.05.2008
		JP 2008-517499 A	22.05.2008
		JP 2011-101410 A	19.05.2011
		JP 2011-109691 A	02.06.2011
		JP 5122288 B2	16.01.2013
		MX 2007004408 A	15.06.2007
		MX 2007004409 A	17.07.2007
		US 2006-0083308 A1	20.04.2006
		US 2006-0083309 A1	20.04.2006
		US 2012-0250759 A1	04.10.2012
		US 2012-0250760 A1	04.10.2012
		US 8218628 B2	10.07.2012
		WO 2006-042611 A1	27.04.2006
		WO 2006-042612 A1	27.04.2006
US 2005-0190979 A1	01.09.2005	US 2005-0190978 A1	01.09.2005
		US 2010-0020882 A1	28.01.2010
		US 7580461 B2	25.08.2009
		US 7627037 B2	01.12.2009
		US 8243812 B2	14.08.2012

A. 발명이 속하는 기술분류(국제특허분류(IPC))
H04N 7/32(2006.01)i

B. 조사된 분야
조사된 최소문헌(국제특허분류를 기재)
H04N 7/32; G06K 9/36; H04N 7/26; H04N 7/12

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: 해상도, 모션 벡터, 움직임 정보, 스케일링, 참조 레이어, 현재 레이어, 인터레이어 예측


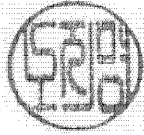
C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
Y	EP 1694074 A1 (THOMSON LICENSING) 2006.08.23 단락 [0005], [0009]와 [0025]; 청구항 1과 9; 및 도면 1 참조.	1-5,19
A		6-18,20
Y	WOLF, KOEN DE 외 3명. `Adaptive Residual Interpolation: a Tool for Efficient Spatial Scalability in Digital Video Coding.` Proceedings of the 2006 International Conference on Image Processing, Computer Vision, & Pattern Recognitron. Las Vegas, Nevada, USA. 2006.06.26-29, Vol. 1 < http://www.researchgate.net/publication/220808864_Adaptive_Residual_Interpolation_a_Tool_for_Efficient_Spatial_Scalability_in_Digital_Video_Coding?ev=sim_pub > 페이지 3, 왼쪽 칼럼, 라인 46 - 페이지 3, 오른쪽 칼럼, 라인 15 참조.	1-5,19
A	KR 10-2009-0012986 A (한국정보통신대학교 산학협력단 외 1명) 2009.02.04 단락 [0023]-[0041]; 및 도면 1-3과 8 참조.	1-20
A	US 2011-0038421 A1 (SCHWARZ, HEIKO 외 2명) 2011.02.17 단락 [0071]-[0083]; 및 도면 1a와 2 참조.	1-20
A	US 2005-0190979 A1 (LI, SHIPENG 외 4명) 2005.09.01 단락 [0003]-[0008]; 청구항 1-2; 및 도면 2-10 참조.	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
 “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 2013년 06월 24일 (24.06.2013)	국제조사보고서 발송일 2013년 06월 24일 (24.06.2013)
--	---

ISA/KR의 명칭 및 우편주소  대한민국 특허청 (302-701) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사) 팩스 번호 82-42-472-7140	심사관 노지명 전화번호 82-42-481-8528 
--	--

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
EP 1694074 A1	2006.08.23	CN 101073265 A	2007.11.14
		CN 101073265 B	2012.08.22
		CN 101204092 A	2008.06.18
		CN 101204092 B	2010.11.03
		CN 101213840 A	2008.07.02
		CN 101213840 B	2011.02.02
		EP 1694075 A1	2006.08.23
		EP 1817914 A1	2007.08.15
		EP 1849303 A2	2007.10.31
		EP 1894412 A1	2008.03.05
		FR 2879066 A1	2006.06.09
		FR 2879066 B1	2007.04.06
		JP 2008-530926 A	2008.08.07
		JP 2008-530927 A	2008.08.07
		JP 5065051 B2	2012.10.31
		KR 10-1233854 B1	2013.02.15
		KR 10-2007-0103459 A	2007.10.23
		US 2008-0267291 A1	2008.10.30
		US 2009-0028245 A1	2009.01.29
		US 8345762 B2	2013.01.01
		WO 2006-058921 A1	2006.06.08
		WO 2006-087314 A1	2006.08.24
		KR 10-2009-0012986 A	2009.02.04
US 2011-0038421 A1	2011.02.17	CN 101044762 A	2007.09.26
		CN 101095357 A	2007.12.26
		CN 101095357 B	2011.06.15
		CN 101854550 A	2010.10.06
		CN 101854550 B	2013.02.13
		CN 101931810 A	2010.12.29
		CN 101931810 B	2012.12.19
		CN 101931811 A	2010.12.29
		CN 101931811 B	2012.10.10
		CN 101931812 A	2010.12.29
		CN 101931812 B	2011.12.21
		CN 101931813 A	2010.12.29
		CN 101931813 B	2012.10.31
		CN 101938651 A	2011.01.05
		CN 101938651 B	2012.10.31
		CN 101959066 A	2011.01.26
		CN 101959066 B	2012.10.10
		CN 101980535 A	2011.02.23
		CN 101980535 B	2012.12.12
		CN 102055985 A	2011.05.11
		CN 102055985 B	2012.09.05
		CN 102055986 A	2011.05.11
		CN 102055986 B	2012.11.21
CN 102082952 A	2011.06.01		

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
		CN 102082952 B	2012.08.08
		CN 102098514 A	2011.06.15
		DE 102004059978 A1	2006.04.27
		DE 102004059978 B4	2006.09.07
		DE 102004059993 A1	2006.04.27
		DE 102004059993 B4	2006.08.31
		EP 1800488 A1	2007.06.27
		EP 1800490 A1	2007.06.27
		EP 2538677 A1	2012.12.26
		EP 2538678 A1	2012.12.26
		JP 2008-517498 A	2008.05.22
		JP 2008-517499 A	2008.05.22
		JP 2011-101410 A	2011.05.19
		JP 2011-109691 A	2011.06.02
		JP 5122288 B2	2013.01.16
		MX 2007004408 A	2007.06.15
		MX 2007004409 A	2007.07.17
		US 2006-0083308 A1	2006.04.20
		US 2006-0083309 A1	2006.04.20
		US 2012-0250759 A1	2012.10.04
		US 2012-0250760 A1	2012.10.04
		US 8218628 B2	2012.07.10
		WO 2006-042611 A1	2006.04.27
		WO 2006-042612 A1	2006.04.27
US 2005-0190979 A1	2005.09.01	US 2005-0190978 A1	2005.09.01
		US 2010-0020882 A1	2010.01.28
		US 7580461 B2	2009.08.25
		US 7627037 B2	2009.12.01
		US 8243812 B2	2012.08.14