

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일

2020 년 12 월 17 일 (17.12.2020) WIPO | PCT



(10) 국제공개번호

WO 2020/251257 A1

(51) 국제특허분류:

H04N 19/527 (2014.01) H04N 19/105 (2014.01)
H04N 19/109 (2014.01) H04N 19/176 (2014.01)
H04N 19/132 (2014.01) H04N 19/70 (2014.01)
H04N 19/137 (2014.01)

(21) 국제출원번호: PCT/KR2020/007521

(22) 국제출원일: 2020 년 6 월 10 일 (10.06.2020)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
62/861,986 2019 년 6 월 14 일 (14.06.2019) US

(71) 출원인: 엘지전자 주식회사 (LG ELECTRONICS INC.) [KR/KR]; 07336 서울시 영등포구 여의대로 128, Seoul (KR).

(72) 발명자: 박내리 (PARK, Naeri); 06772 서울시 서초구 양재대로11길 19 LG전자 특허센터, Seoul (KR). 남정학 (NAM, Junghak); 06772 서울시 서초구 양재대로11길 19 LG전자 특허센터, Seoul (KR). 장형문 (JANG, Hyeonmoon); 06772 서울시 서초구 양재대로11길 19 LG전자 특허센터, Seoul (KR).

(74) 대리인: 인비전 특허법인 (ENVISION PATENT & LAW FIRM); 06193 서울시 강남구 테헤란로 70길 16, 8층, Seoul (KR).

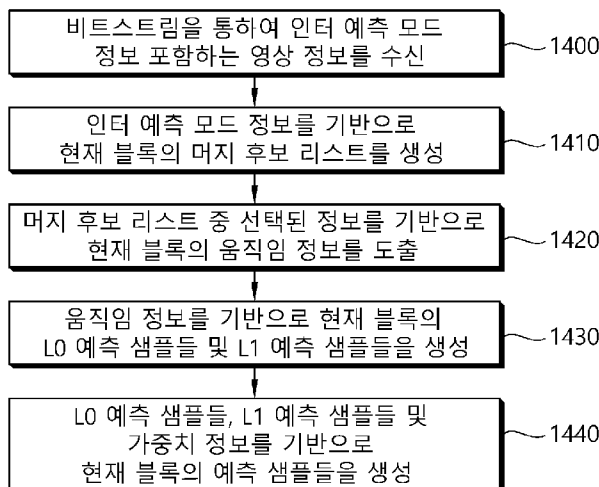
(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK,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

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

(54) Title: IMAGE DECODING METHOD AND DEVICE FOR DERIVING WEIGHT INDEX INFORMATION FOR GENERATION OF PREDICTION SAMPLE

(54) 발명의 명칭: 예측 샘플을 생성하기 위한 가중치 인덱스 정보를 도출하는 영상 디코딩 방법 및 그 장치

[도14]



1400 ... Receive image information including inter prediction mode information through bitstream
1410 ... Generate merge candidate list of current block on basis of inter prediction mode information
1420 ... Derive motion information of current block on basis of information selected from merge candidate list
1430 ... Generate L0 prediction samples and L1 prediction samples of current block on basis of motion information
1440 ... Generate prediction samples of current block on basis of L0 prediction samples, L1 prediction samples, and weight information

(57) Abstract: According to the disclosure of the present document, when the inter prediction type of a current block indicates bi-prediction, weight index information for candidates in a merge candidate list or a sub-block merge candidate list can be derived, and thus coding efficiency can be increased.

(57) 요약서: 본 문서의 개시에 따르면, 현재 블록의 인터 예측 타입이 쌍 예측(bi-prediction)을 나타내는 경우, 머지 후보 리스트 또는 서브블록 머지 후보 리스트 내의 후보를 위한 가중치 인덱스 정보를 도출할 수 있고, 코딩 효율을 높일 수 있다.

[다음 쪽 계속]

WO 2020/251257 A1

공개:

— 국제조사보고서와 함께 (조약 제21조(3))

명세서

발명의 명칭: 예측 샘플을 생성하기 위한 가중치 인덱스 정보를 도출하는 영상 디코딩 방법 및 그 장치

기술분야

- [1] 본 기술은 예측 샘플을 생성하기 위한 가중치 인덱스 정보를 도출하는 영상 디코딩 방법 및 그 장치에 관한 것이다.

배경기술

- [2] 최근 4K 또는 8K 이상의 UHD(Ultra High Definition) 영상/비디오와 같은 고해상도, 고품질의 영상/비디오에 대한 수요가 다양한 분야에서 증가하고 있다. 영상/비디오 데이터가 고해상도, 고품질이 될수록 기존의 영상/비디오 데이터에 비해 상대적으로 전송되는 정보량 또는 비트량이 증가하기 때문에 기존의 유무선 광대역 회선과 같은 매체를 이용하여 영상 데이터를 전송하거나 기존의 저장 매체를 이용해 영상/비디오 데이터를 저장하는 경우, 전송 비용과 저장 비용이 증가된다.
- [3] 또한, 최근 VR(Virtual Reality), AR(Artificial Reality) 콘텐츠나 홀로그램 등의 실감 미디어(Immersive Media)에 대한 관심 및 수요가 증가하고 있으며, 게임 영상과 같이 현실 영상과 다른 영상 특성을 갖는 영상/비디오에 대한 방송이 증가하고 있다.
- [4] 이에 따라, 상기와 같은 다양한 특성을 갖는 고해상도 고품질의 영상/비디오의 정보를 효과적으로 압축하여 전송하거나 저장하고, 재생하기 위해 고효율의 영상/비디오 압축 기술이 요구된다.

발명의 상세한 설명

기술적 과제

- [5] 본 문서의 기술적 과제는 영상 코딩 효율을 높이는 방법 및 장치를 제공함에 있다.
- [6] 본 문서의 다른 기술적 과제는 인터 예측에서 예측 샘플을 생성하기 위한 가중치 인덱스 정보를 유도하는 방법 및 장치를 제공함에 있다.
- [7] 본 문서의 또 다른 기술적 과제는 쌍 예측 시 어파인 머지 후보 리스트 내의 후보를 위한 가중치 인덱스 정보를 유도하는 방법 및 장치를 제공함에 있다.

과제 해결 수단

- [8] 본 문서의 일 실시예에 따르면, 디코딩 장치에 의하여 수행되는 영상 디코딩 방법이 제공된다. 상기 방법은 비트스트림을 통하여 인터 예측 모드 정보를 포함하는 영상 정보를 수신하는 단계; 상기 인터 예측 모드 정보를 기반으로 현재 블록의 머지 후보 리스트를 생성하는 단계; 상기 머지 후보 리스트 중 선택된 후보를 기반으로 상기 현재 블록의 움직임 정보를 도출하는 단계; 상기 움직임 정보를 기반으로 상기 현재 블록의 L0 예측 샘플들 및 L1 예측 샘플들을

생성하는 단계; 및 상기 L0 예측 샘플들, 상기 L1 예측 샘플들 및 가중치 정보를 기반으로 상기 현재 블록의 예측 샘플들을 생성하고, 상기 가중치 정보는 상기 선택된 후보에 대한 가중치 인덱스 정보를 기반으로 도출되는 단계를 포함하고, 상기 후보들은 어파인 머지 후보를 포함하고, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함하고, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 도출되고, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에 대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 도출된다.

- [9] 본 문서의 다른 일 실시예에 따르면, 인코딩 장치에 의하여 수행되는 비디오 인코딩 방법을 제공한다. 상기 방법은 현재 블록의 인터 예측 모드를 결정하고, 상기 인터 예측 모드를 나타내는 인터 예측 모드 정보를 생성하는 단계; 상기 인터 예측 모드를 기반으로 상기 현재 블록의 머지 후보 리스트를 생성하는 단계; 상기 머지 후보 리스트에 포함된 후보들 중 하나의 후보를 나타내는 선택 정보를 생성하는 단계; 및 상기 인터 예측 모드 정보 및 상기 선택 정보를 포함하는 영상 정보를 인코딩하는 단계를 포함하고, 상기 후보들은 어파인 머지 후보를 포함하고, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함하고, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 나타내어지고, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에 대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 나타내어진다.

- [10] 본 문서의 또 다른 일 실시예에 따르면, 디코딩 장치로 하여금 영상 디코딩 방법을 수행하도록 야기하는 영상 정보를 포함하는 비트스트림이 저장된 컴퓨터 판독가능 디지털 저장 매체를 제공한다. 상기 영상 디코딩 방법은, 비트스트림을 통하여 인터 예측 모드 정보를 포함하는 영상 정보를 획득하는 단계; 상기 인터 예측 모드 정보를 기반으로 현재 블록의 머지 후보 리스트를 생성하는 단계; 상기 머지 후보 리스트에 포함된 후보들 중 하나의 후보를 선택하는 단계; 상기 선택된 후보를 기반으로 상기 현재 블록의 움직임 정보를 도출하는 단계; 상기 움직임 정보를 기반으로 상기 현재 블록의 L0 예측 샘플들 및 L1 예측 샘플들을 생성하는 단계; 및 상기 L0 예측 샘플들, 상기 L1 예측 샘플들 및 가중치 정보를 기반으로 상기 현재 블록의 예측 샘플들을 생성하고, 상기 가중치 정보는 상기 선택된 후보에 대한 가중치 인덱스 정보를 기반으로

도출되는 단계를 포함하고, 상기 후보들은 어파인 머지 후보를 포함하고, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함하고, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 도출되고, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에 대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 도출된다.

발명의 효과

- [11] 본 문서에 따르면 전반적인 영상/비디오 압축 효율을 높일 수 있다..
- [12] 본 문서에 따르면 인터 예측 시 움직임 벡터 후보를 효율적으로 구성할 수 있다.
- [13] 본 문서에 따르면 효율적으로 가중치 기반 쌍 예측을 수행할 수 있다.

도면의 간단한 설명

- [14] 도 1은 본 문서의 실시예들이 적용될 수 있는 비디오/영상 코딩 시스템의 예를 개략적으로 나타낸다.
- [15] 도 2는 본 문서의 실시예들이 적용될 수 있는 비디오/영상 인코딩 장치의 구성을 개략적으로 설명하는 도면이다.
- [16] 도 3은 본 문서의 실시예들이 적용될 수 있는 비디오/영상 디코딩 장치의 구성을 개략적으로 설명하는 도면이다.
- [17] 도 4는 인터 예측 절차를 예시적으로 나타낸다.
- [18] 도 5는 인터 예측에서 머지 모드를 설명하기 위한 도면이다.
- [19] 도 6은 어파인 움직임 모델을 통하여 표현되는 움직임을 예시적으로 나타낸다.
- [20] 도 7a 및 도 7b는 어파인 움직임 예측을 위한 CPMV를 예시적으로 나타낸다.
- [21] 도 8은 어파인 MVF가 서브블록 단위에서 결정되는 경우를 예시적으로 나타낸다.
- [22] 도 9는 인터 예측에서 어파인 머지 모드를 설명하기 위한 도면이다.
- [23] 도 10은 어파인 머지 모드에서 후보들의 위치를 설명하기 위한 도면이다.
- [24] 도 11은 인터 예측에서 SbTMVP를 설명하기 위한 도면이다.
- [25] 도 12 및 13은 본 문서의 실시예(들)에 따른 비디오/영상 인코딩 방법 및 관련 컴포넌트의 일 예를 개략적으로 나타낸다.
- [26] 도 14 및 15는 본 문서의 실시예(들)에 따른 영상/비디오 디코딩 방법 및 관련 컴포넌트의 일 예를 개략적으로 나타낸다.
- [27] 도 16은 본 문서에서 개시된 실시예들이 적용될 수 있는 콘텐츠 스트리밍 시스템의 예를 나타낸다.

발명의 실시를 위한 형태

- [28] 본 개시는 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바,

특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 개시를 특정 실시예에 한정하려고 하는 것이 아니다. 본 명세서에서 상용하는 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 개시의 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.

- [29] 한편, 본 개시에서 설명되는 도면상의 각 구성들은 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 개시의 본질에서 벗어나지 않는 한 본 개시의 권리범위에 포함된다.
- [30] 본 명세서에서 “A 또는 B(A or B)”는 “오직 A”, “오직 B” 또는 “A와 B 모두”를 의미할 수 있다. 달리 표현하면, 본 명세서에서 “A 또는 B(A or B)”는 “A 및/또는 B(A and/or B)”으로 해석될 수 있다. 예를 들어, 본 명세서에서 “A, B 또는 C(A, B or C)”는 “오직 A”, “오직 B”, “오직 C”, 또는 “A, B 및 C의 임의의 모든 조합(any combination of A, B and C)”를 의미할 수 있다.
- [31] 본 명세서에서 사용되는 슬래쉬(/)나 쉼표(comma)는 “및/또는(and/or)”을 의미할 수 있다. 예를 들어, “A/B”는 “A 및/또는 B”를 의미할 수 있다. 이에 따라 “A/B”는 “오직 A”, “오직 B”, 또는 “A와 B 모두”를 의미할 수 있다. 예를 들어, “A, B, C”는 “A, B 또는 C”를 의미할 수 있다.
- [32] 본 명세서에서 “적어도 하나의 A 및 B(at least one of A and B)”는, “오직 A”, “오직 B” 또는 “A와 B 모두”를 의미할 수 있다. 또한, 본 명세서에서 “적어도 하나의 A 또는 B(at least one of A or B)”나 “적어도 하나의 A 및/또는 B(at least one of A and/or B)”라는 표현은 “적어도 하나의 A 및 B(at least one of A and B)”와 동일하게 해석될 수 있다.
- [33] 또한, 본 명세서에서 “적어도 하나의 A, B 및 C(at least one of A, B and C)”는, “오직 A”, “오직 B”, “오직 C”, 또는 “A, B 및 C의 임의의 모든 조합(any combination of A, B and C)”를 의미할 수 있다. 또한, “적어도 하나의 A, B 또는 C(at least one of A, B or C)”나 “적어도 하나의 A, B 및/또는 C(at least one of A, B and/or C)”는 “적어도 하나의 A, B 및 C(at least one of A, B and C)”를 의미할 수 있다.
- [34] 또한, 본 명세서에서 사용되는 괄호는 “예를 들어(for example)”를 의미할 수

있다. 구체적으로, “예측(인트라 예측)”로 표시된 경우, “예측”의 일례로 “인트라 예측”이 제안된 것일 수 있다. 달리 표현하면 본 명세서의 “예측”은 “인트라 예측”으로 제한(limit)되지 않고, “인트라 예측”이 “예측”의 일례로 제안될 것일 수 있다. 또한, “예측(즉, 인트라 예측)”으로 표시된 경우에도, “예측”의 일례로 “인트라 예측”이 제안된 것일 수 있다.

- [35] 본 명세서에서 하나의 도면 내에서 개별적으로 설명되는 기술적 특징은, 개별적으로 구현될 수도 있고, 동시에 구현될 수도 있다.
- [36] 이하, 첨부한 도면들을 참조하여, 본 개시의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 이하, 도면상의 동일한 구성 요소에 대해서는 동일한 참조 부호를 사용하고 동일한 구성 요소에 대해서 중복된 설명은 생략될 수 있다.
- [37] 도 1은 본 개시가 적용될 수 있는 비디오/영상 코딩 시스템의 예를 개략적으로 나타낸다.
- [38] 도 1을 참조하면, 비디오/영상 코딩 시스템은 제1 장치(소스 디바이스) 및 제2 장치(수신 디바이스)를 포함할 수 있다. 소스 디바이스는 인코딩된 비디오(video)/영상(image) 정보 또는 데이터를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여 수신 디바이스로 전달할 수 있다.
- [39] 상기 소스 디바이스는 비디오 소스, 인코딩 장치, 전송부를 포함할 수 있다. 상기 수신 디바이스는 수신부, 디코딩 장치 및 렌더러를 포함할 수 있다. 상기 인코딩 장치는 비디오/영상 인코딩 장치라고 불릴 수 있고, 상기 디코딩 장치는 비디오/영상 디코딩 장치라고 불릴 수 있다. 송신기는 인코딩 장치에 포함될 수 있다. 수신기는 디코딩 장치에 포함될 수 있다. 렌더러는 디스플레이부를 포함할 수도 있고, 디스플레이부는 별개의 디바이스 또는 외부 컴포넌트로 구성될 수도 있다.
- [40] 비디오 소스는 비디오/영상의 캡처, 합성 또는 생성 과정 등을 통하여 비디오/영상을 획득할 수 있다. 비디오 소스는 비디오/영상 캡처 디바이스 및/또는 비디오/영상 생성 디바이스를 포함할 수 있다. 비디오/영상 캡처 디바이스는 예를 들어, 하나 이상의 카메라, 이전에 캡처된 비디오/영상을 포함하는 비디오/영상 아카이브 등을 포함할 수 있다. 비디오/영상 생성 디바이스는 예를 들어 컴퓨터, 태블릿 및 스마트폰 등을 포함할 수 있으며 (전자적으로) 비디오/영상을 생성할 수 있다. 예를 들어, 컴퓨터 등을 통하여 가상의 비디오/영상이 생성될 수 있으며, 이 경우 관련 데이터가 생성되는 과정으로 비디오/영상 캡처 과정이 갈음될 수 있다.
- [41] 인코딩 장치는 입력 비디오/영상을 인코딩할 수 있다. 인코딩 장치는 압축 및 코딩 효율을 위하여 예측, 변환, 양자화 등 일련의 절차를 수행할 수 있다. 인코딩된 데이터(인코딩된 비디오/영상 정보)는 비트스트림(bitstream) 형태로 출력될 수 있다.
- [42] 전송부는 비트스트림 형태로 출력된 인코딩된 비디오/영상 정보 또는 데이터를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여

수신 디바이스의 수신부로 전달할 수 있다. 디지털 저장 매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장 매체를 포함할 수 있다. 전송부는 미리 정해진 파일 포맷을 통하여 미디어 파일을 생성하기 위한 엘리먼트를 포함할 수 있고, 방송/통신 네트워크를 통한 전송을 위한 엘리먼트를 포함할 수 있다. 수신부는 상기 비트스트림을 수신/추출하여 디코딩 장치로 전달할 수 있다.

- [43] 디코딩 장치는 인코딩 장치의 동작에 대응하는 역양자화, 역변환, 예측 등 일련의 절차를 수행하여 비디오/영상을 디코딩할 수 있다.
- [44] 렌더러는 디코딩된 비디오/영상을 렌더링할 수 있다. 렌더링된 비디오/영상은 디스플레이부를 통하여 디스플레이될 수 있다.
- [45] 이 문서는 비디오/영상 코딩에 관한 것이다. 예를 들어 이 문서에서 개시된 방법/실시예는 VVC (versatile video coding) 표준, EVC (essential video coding) 표준, AV1 (AOMedia Video 1) 표준, AVS2 (2nd generation of audio video coding standard) 또는 차세대 비디오/영상 코딩 표준(ex. H.267 or H.268 등)에 개시되는 방법에 적용될 수 있다.
- [46] 이 문서에서는 비디오/영상 코딩에 관한 다양한 실시예들을 제시하며, 다른 언급이 없는 한 상기 실시예들은 서로 조합되어 수행될 수도 있다.
- [47] 이 문서에서 비디오(video)는 시간의 흐름에 따른 일련의 영상(image)들의 집합을 의미할 수 있다. 픽처(picture)는 일반적으로 특정 시간대의 하나의 영상을 나타내는 단위를 의미하며, 슬라이스(slice)/타일(tile)는 코딩에 있어서 픽처의 일부를 구성하는 단위이다. 슬라이스/타일은 하나 이상의 CTU(coding tree unit)를 포함할 수 있다. 하나의 픽처는 하나 이상의 슬라이스/타일로 구성될 수 있다.
- [48] 타일은 특정 타일 열 및 특정 타일 열 이내의 CTU들의 사각 영역이다(A tile is a rectangular region of CTUs within a particular tile column and a particular tile row in a picture). 상기 타일 열은 CTU들의 사각 영역이고, 상기 사각 영역은 상기 픽처의 높이와 동일한 높이를 갖고, 너비는 픽처 파라미터 세트 내의 선택스 요소들에 의하여 명시될 수 있다(The tile column is a rectangular region of CTUs having a height equal to the height of the picture and a width specified by syntax elements in the picture parameter set). 상기 타일 행은 CTU들의 사각 영역이고, 상기 사각 영역은 픽처 파라미터 세트 내의 선택스 요소들에 의하여 명시되는 너비를 갖고, 높이는 상기 픽처의 높이와 동일할 수 있다(The tile row is a rectangular region of CTUs having a height specified by syntax elements in the picture parameter set and a width equal to the width of the picture). 타일 스캔은 픽처를 파티셔닝하는 CTU들의 특정 순차적 오더링을 나타낼 수 있고, 상기 CTU들은 타일 내 CTU 래스터 스캔으로 연속적으로 정렬될 수 있고, 픽처 내 타일들은 상기 픽처의 상기 타일들의 래스터 스캔으로 연속적으로 정렬될 수 있다(A tile scan is a specific sequential ordering of CTUs partitioning a picture in which the CTUs are ordered consecutively in CTU raster scan in a tile whereas tiles in a picture are ordered consecutively in a

raster scan of the tiles of the picture). 슬라이스는 다수의 완전한 타일들 또는 하나의 NAL 유닛에 포함될 수 있는 픽처의 하나의 타일 내 다수의 연속적인 CTU 행들을 포함할 수 있다. 이 문서에서 타일 그룹과 슬라이스는 혼용될 수 있다. 예를 들어 본 문서에서 tile group/tile group header는 slice/slice header로 불릴 수 있다.

- [49] 한편, 하나의 픽처는 둘 이상의 서브픽처로 구분될 수 있다. 서브픽처는 픽처 내 하나 이상의 슬라이스들의 사각 리전일 수 있다(an rectangular region of one or more slices within a picture).
- [50] 픽셀(pixel) 또는 펠(pel)은 하나의 픽처(또는 영상)을 구성하는 최소의 단위를 의미할 수 있다. 또한, 픽셀에 대응하는 용어로서 '샘플(sample)'이 사용될 수 있다. 샘플은 일반적으로 픽셀 또는 픽셀의 값을 나타낼 수 있으며, 루마(luma) 성분의 픽셀/픽셀값만을 나타낼 수도 있고, 크로마(chroma) 성분의 픽셀/픽셀값만을 나타낼 수도 있다.
- [51] 유닛(unit)은 영상 처리의 기본 단위를 나타낼 수 있다. 유닛은 픽처의 특정 영역 및 해당 영역에 관련된 정보 중 적어도 하나를 포함할 수 있다. 하나의 유닛은 하나의 루마 블록 및 두개의 크로마(ex. cb, cr) 블록을 포함할 수 있다. 유닛은 경우에 따라서 블록(block) 또는 영역(area) 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, MxN 블록은 M개의 열과 N개의 행으로 이루어진 샘플들(또는 샘플 어레이) 또는 변환 계수(transform coefficient)들의 집합(또는 어레이)을 포함할 수 있다.
- [52] 도 2는 본 개시가 적용될 수 있는 비디오/영상 인코딩 장치의 구성을 개략적으로 설명하는 도면이다. 이하 비디오 인코딩 장치라 함은 영상 인코딩 장치를 포함할 수 있다.
- [53] 도 2를 참조하면, 인코딩 장치(200)는 영상 분할부(image partitioner, 210), 예측부(predictor, 220), 레지듀얼 처리부(residual processor, 230), 엔트로피 인코딩부(entropy encoder, 240), 가산부(adder, 250), 필터링부(filter, 260) 및 메모리(memory, 270)를 포함하여 구성될 수 있다. 예측부(220)는 인터 예측부(221) 및 인트라 예측부(222)를 포함할 수 있다. 레지듀얼 처리부(230)는 변환부(transformer, 232), 양자화부(quantizer 233), 역양자화부(dequantizer 234), 역변환부(inverse transformer, 235)를 포함할 수 있다. 레지듀얼 처리부(230)는 감산부(subtractor, 231)를 더 포함할 수 있다. 가산부(250)는 복원부(reconstructor) 또는 복원 블록 생성부(reconstructged block generator)로 불릴 수 있다. 상술한 영상 분할부(210), 예측부(220), 레지듀얼 처리부(230), 엔트로피 인코딩부(240), 가산부(250) 및 필터링부(260)는 실시예에 따라 하나 이상의 하드웨어 컴포넌트(예를 들어 인코더 칩셋 또는 프로세서)에 의하여 구성될 수 있다. 또한 메모리(270)는 DPB(decoded picture buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구성될 수도 있다. 상기 하드웨어 컴포넌트는 메모리(270)을 내/외부 컴포넌트로 더 포함할 수도 있다.

- [54] 영상 분할부(210)는 인코딩 장치(200)에 입력된 입력 영상(또는, 픽처, 프레임)를 하나 이상의 처리 유닛(processing unit)으로 분할할 수 있다. 일 예로, 상기 처리 유닛은 코딩 유닛(coding unit, CU)이라고 불릴 수 있다. 이 경우 코딩 유닛은 코딩 트리 유닛(coding tree unit, CTU) 또는 최대 코딩 유닛(largest coding unit, LCU)으로부터 QTBT (Quad-tree binary-tree ternary-tree) 구조에 따라 재귀적으로(recursively) 분할될 수 있다. 예를 들어, 하나의 코딩 유닛은 쿼드 트리 구조, 바이너리 트리 구조, 및/또는 터너리 구조를 기반으로 하위(deeper) 템스의 복수의 코딩 유닛들로 분할될 수 있다. 이 경우 예를 들어 쿼드 트리 구조가 먼저 적용되고 바이너리 트리 구조 및/또는 터너리 구조가 나중에 적용될 수 있다. 또는 바이너리 트리 구조가 먼저 적용될 수도 있다. 더 이상 분할되지 않는 최종 코딩 유닛을 기반으로 본 개시에 따른 코딩 절차가 수행될 수 있다. 이 경우 영상 특성에 따른 코딩 효율 등을 기반으로, 최대 코딩 유닛이 바로 최종 코딩 유닛으로 사용될 수 있고, 또는 필요에 따라 코딩 유닛은 재귀적으로(recursively) 보다 하위 템스의 코딩 유닛들로 분할되어 최적의 사이즈의 코딩 유닛이 최종 코딩 유닛으로 사용될 수 있다. 여기서 코딩 절차라 함은 후술하는 예측, 변환, 및 복원 등의 절차를 포함할 수 있다. 다른 예로, 상기 처리 유닛은 예측 유닛(PU: Prediction Unit) 또는 변환 유닛(TU: Transform Unit)을 더 포함할 수 있다. 이 경우 상기 예측 유닛 및 상기 변환 유닛은 각각 상술한 최종 코딩 유닛으로부터 분할 또는 파티셔닝될 수 있다. 상기 예측 유닛은 샘플 예측의 단위일 수 있고, 상기 변환 유닛은 변환 계수를 유도하는 단위 및/또는 변환 계수로부터 레지듀얼 신호(residual signal)를 유도하는 단위일 수 있다.
- [55] 유닛은 경우에 따라서 블록(block) 또는 영역(area) 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, $M \times N$ 블록은 M개의 열과 N개의 행으로 이루어진 샘플들 또는 변환 계수(transform coefficient)들의 집합을 나타낼 수 있다. 샘플은 일반적으로 픽셀 또는 픽셀의 값을 나타낼 수 있으며, 휘도(luma) 성분의 픽셀/픽셀값만을 나타낼 수도 있고, 채도(chroma) 성분의 픽셀/픽셀값만을 나타낼 수도 있다. 샘플은 하나의 픽처(또는 영상)을 픽셀(pixel) 또는 펄(pel)에 대응하는 용어로서 사용될 수 있다.
- [56] 인코딩 장치(200)는 입력 영상 신호(원본 블록, 원본 샘플 어레이)에서 인터 예측부(221) 또는 인트라 예측부(222)로부터 출력된 예측 신호(예측된 블록, 예측 샘플 어레이)를 감산하여 레지듀얼 신호(residual signal, 잔여 블록, 잔여 샘플 어레이)를 생성할 수 있고, 생성된 레지듀얼 신호는 변환부(232)로 전송된다. 이 경우 도시된 바와 같이 인코딩 장치(200) 내에서 입력 영상 신호(원본 블록, 원본 샘플 어레이)에서 예측 신호(예측 블록, 예측 샘플 어레이)를 감산하는 유닛은 감산부(231)라고 불릴 수 있다. 예측부는 처리 대상 블록(이하, 현재 블록이라 함)에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부는 현재 블록 또는 CU 단위로 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있다.

예측부는 각 예측모드에 대한 설명에서 후술하는 바와 같이 예측 모드 정보 등 예측에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(240)로 전달할 수 있다. 예측에 관한 정보는 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.

- [57] 인트라 예측부(222)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 상기 참조되는 샘플들은 예측 모드에 따라 상기 현재 블록의 주변(neighbor)에 위치할 수 있고, 또는 떨어져서 위치할 수도 있다. 인트라 예측에서 예측 모드들은 복수의 비방향성 모드와 복수의 방향성 모드를 포함할 수 있다. 비방향성 모드는 예를 들어 DC 모드 및 플래너 모드(Planar 모드)를 포함할 수 있다. 방향성 모드는 예측 방향의 세밀한 정도에 따라 예를 들어 33개의 방향성 예측 모드 또는 65개의 방향성 예측 모드를 포함할 수 있다. 다만, 이는 예시로서 설정에 따라 그 이상 또는 그 이하의 개수의 방향성 예측 모드들이 사용될 수 있다. 인트라 예측부(222)는 주변 블록에 적용된 예측 모드를 이용하여, 현재 블록에 적용되는 예측 모드를 결정할 수도 있다.
- [58] 인터 예측부(221)는 참조 픽처 상에서 움직임 벡터에 의해 특정되는 참조 블록(참조 샘플 어레이)을 기반으로, 현재 블록에 대한 예측된 블록을 유도할 수 있다. 이때, 인터 예측 모드에서 전송되는 움직임 정보의 양을 줄이기 위해 주변 블록과 현재 블록 간의 움직임 정보의 상관성에 기초하여 움직임 정보를 블록, 서브블록 또는 샘플 단위로 예측할 수 있다. 상기 움직임 정보는 움직임 벡터 및 참조 픽처 인덱스를 포함할 수 있다. 상기 움직임 정보는 인터 예측 방향(L0 예측, L1 예측, Bi 예측 등) 정보를 더 포함할 수 있다. 인터 예측의 경우에, 주변 블록은 현재 픽처 내에 존재하는 공간적 주변 블록(spatial neighboring block)과 참조 픽처에 존재하는 시간적 주변 블록(temporal neighboring C)을 포함할 수 있다. 상기 참조 블록을 포함하는 참조 픽처와 상기 시간적 주변 블록을 포함하는 참조 픽처는 동일할 수도 있고, 다를 수도 있다. 상기 시간적 주변 블록은 동일 위치 참조 블록(collocated reference block), 동일 위치 CU(colCU) 등의 이름으로 불릴 수 있으며, 상기 시간적 주변 블록을 포함하는 참조 픽처는 동일 위치 픽처(collocated picture, colPic)라고 불릴 수도 있다. 예를 들어, 인터 예측부(221)는 주변 블록들을 기반으로 움직임 정보 후보 리스트를 구성하고, 상기 현재 블록의 움직임 벡터 및/또는 참조 픽처 인덱스를 도출하기 위하여 어떤 후보가 사용되는지를 지시하는 정보를 생성할 수 있다. 다양한 예측 모드를 기반으로 인터 예측이 수행될 수 있으며, 예를 들어 스킵 모드와 머지 모드의 경우에, 인터 예측부(221)는 주변 블록의 움직임 정보를 현재 블록의 움직임 정보로 이용할 수 있다. 스킵 모드의 경우, 머지 모드와 달리 레지듀얼 신호가 전송되지 않을 수 있다. 움직임 정보 예측(motion vector prediction, MVP) 모드의 경우, 주변 블록의 움직임 벡터를 움직임 벡터 예측자(motion vector predictor)로 이용하고, 움직임 벡터 차분(motion vector difference)을 시그널링함으로써 현재 블록의 움직임 벡터를 지시할 수 있다.

- [59] 예측부(220)는 후술하는 다양한 예측 방법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부는 하나의 블록에 대한 예측을 위하여 인트라 예측 또는 인터 예측을 적용할 수 있을 뿐 아니라, 인트라 예측과 인터 예측을 동시에 적용할 수 있다. 이는 combined inter and intra prediction (CIIP)라고 불릴 수 있다. 또한, 예측부는 블록에 대한 예측을 위하여 인트라 블록 카피(intra block copy, IBC) 예측 모드에 기반할 수도 있고 또는 팔레트 모드(palette mode)에 기반할 수도 있다. 상기 IBC 예측 모드 또는 팔레트 모드는 예를 들어 SCC(screen content coding) 등과 같이 게임 등의 콘텐츠 영상/동영상 코딩을 위하여 사용될 수 있다. IBC는 기본적으로 현재 픽처 내에서 예측을 수행하나 현재 픽처 내에서 참조 블록을 도출하는 점에서 인터 예측과 유사하게 수행될 수 있다. 즉, IBC는 본 문서에서 설명되는 인터 예측 기법들 중 적어도 하나를 이용할 수 있다. 팔레트 모드는 인트라 코딩 또는 인트라 예측의 일 예로 볼 수 있다. 팔레트 모드가 적용되는 경우 팔레트 테이블 및 팔레트 인덱스에 관한 정보를 기반으로 픽처 내 샘플 값을 시그널링할 수 있다.
- [60] 상기 예측부 (인터 예측부(221) 및/또는 상기 인트라 예측부(222) 포함)를 통해 생성된 예측 신호는 복원 신호를 생성하기 위해 이용되거나 레지듀얼 신호를 생성하기 위해 이용될 수 있다. 변환부(232)는 레지듀얼 신호에 변환 기법을 적용하여 변환 계수들(transform coefficients)을 생성할 수 있다. 예를 들어, 변환 기법은 DCT(Discrete Cosine Transform), DST(Discrete Sine Transform), KLT(Karhunen-Loeve Transform), GBT(Graph-Based Transform), 또는 CNT(Conditionally Non-linear Transform) 중 적어도 하나를 포함할 수 있다. 여기서, GBT는 픽셀 간의 관계 정보를 그래프로 표현한다고 할 때 이 그래프로부터 얻어진 변환을 의미한다. CNT는 이전에 복원된 모든 픽셀(all previously reconstructed pixel)를 이용하여 예측 신호를 생성하고 그에 기초하여 획득되는 변환을 의미한다. 또한, 변환 과정은 정사각형의 동일한 크기를 갖는 픽셀 블록에 적용될 수도 있고, 정사각형이 아닌 가변 크기의 블록에도 적용될 수 있다.
- [61] 양자화부(233)는 변환 계수들을 양자화하여 엔트로피 인코딩부(240)로 전송되고, 엔트로피 인코딩부(240)는 양자화된 신호(양자화된 변환 계수들에 관한 정보)를 인코딩하여 비트스트림으로 출력할 수 있다. 상기 양자화된 변환 계수들에 관한 정보는 레지듀얼 정보라고 불릴 수 있다. 양자화부(233)는 계수 스캔 순서(scan order)를 기반으로 블록 형태의 양자화된 변환 계수들을 1차원 벡터 형태로 재정렬할 수 있고, 상기 1차원 벡터 형태의 양자화된 변환 계수들을 기반으로 상기 양자화된 변환 계수들에 관한 정보를 생성할 수도 있다. 엔트로피 인코딩부(240)는 예를 들어 지수 골롬(exponential Golomb), CAVLC(context-adaptive variable length coding), CABAC(context-adaptive binary arithmetic coding) 등과 같은 다양한 인코딩 방법을 수행할 수 있다. 엔트로피 인코딩부(240)는 양자화된 변환 계수들 외 비디오/이미지 복원에 필요한

정보들(예컨대 선택 요소들(syntax elements)의 값 등)을 함께 또는 별도로 인코딩할 수도 있다. 인코딩된 정보(ex. 인코딩된 비디오/영상 정보)는 비트스트림 형태로 NAL(network abstraction layer) 유닛 단위로 전송 또는 저장될 수 있다. 상기 비디오/영상 정보는 어댑테이션 파라미터 세트(APS), 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS) 또는 비디오 파라미터 세트(VPS) 등 다양한 파라미터 세트에 관한 정보를 더 포함할 수 있다. 또한 상기 비디오/영상 정보는 일반 제한 정보(general constraint information)을 더 포함할 수 있다. 본 문서에서 인코딩 장치에서 디코딩 장치로 전달/시그널링되는 정보 및/또는 선택 요소들은 비디오/영상 정보에 포함될 수 있다. 상기 비디오/영상 정보는 상술한 인코딩 절차를 통하여 인코딩되어 상기 비트스트림에 포함될 수 있다. 상기 비트스트림은 네트워크를 통하여 전송될 수 있고, 또는 디지털 저장매체에 저장될 수 있다. 여기서 네트워크는 방송망 및/또는 통신망 등을 포함할 수 있고, 디지털 저장매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장매체를 포함할 수 있다. 엔트로피 인코딩부(240)로부터 출력된 신호는 전송하는 전송부(미도시) 및/또는 저장하는 저장부(미도시)가 인코딩 장치(200)의 내/외부 엘리먼트로서 구성될 수 있고, 또는 전송부는 엔트로피 인코딩부(240)에 포함될 수도 있다.

[62] 양자화부(233)로부터 출력된 양자화된 변환 계수들은 예측 신호를 생성하기 위해 이용될 수 있다. 예를 들어, 양자화된 변환 계수들에 역양자화부(234) 및 역변환부(235)를 통해 역양자화 및 역변환을 적용함으로써 레지듀얼 신호(레지듀얼 블록 or 레지듀얼 샘플들)를 복원할 수 있다. 가산부(155)는 복원된 레지듀얼 신호를 인터 예측부(221) 또는 인트라 예측부(222)로부터 출력된 예측 신호에 더함으로써 복원(reconstructed) 신호(복원 픽처, 복원 블록, 복원 샘플 어레이)가 생성될 수 있다. 스킵 모드가 적용된 경우와 같이 처리 대상 블록에 대한 레지듀얼이 없는 경우, 예측된 블록이 복원 블록으로 사용될 수 있다. 가산부(250)는 복원부 또는 복원 블록 생성부라고 불릴 수 있다. 생성된 복원 신호는 현재 픽처 내 다음 처리 대상 블록의 인트라 예측을 위하여 사용될 수 있고, 후술하는 바와 같이 필터링을 거쳐서 다음 픽처의 인터 예측을 위하여 사용될 수도 있다.

[63] 한편 픽처 인코딩 및/또는 복원 과정에서 LMCS (luma mapping with chroma scaling)가 적용될 수도 있다.

[64] 필터링부(260)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(260)은 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정된 복원 픽처를 메모리(270), 구체적으로 메모리(270)의 DPB에 저장할 수 있다. 상기 다양한 필터링 방법은 예를 들어, 디블록킹 필터링, 샘플 적응적 오프셋(sample adaptive offset), 적응적 루프 필터(adaptive loop filter), 양방향 필터(bilateral filter) 등을 포함할 수 있다. 필터링부(260)은 각 필터링 방법에 대한 설명에서 후술하는

바와 같이 필터링에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(240)로 전달할 수 있다. 필터링 관한 정보는 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.

- [65] 메모리(270)에 전송된 수정된 복원 픽처는 인터 예측부(221)에서 참조 픽처로 사용될 수 있다. 인코딩 장치는 이를 통하여 인터 예측이 적용되는 경우, 인코딩 장치(100)와 디코딩 장치에서의 예측 미스매치를 피할 수 있고, 부호화 효율도 향상시킬 수 있다.
- [66] 메모리(270) DPB는 수정된 복원 픽처를 인터 예측부(221)에서의 참조 픽처로 사용하기 위해 저장할 수 있다. 메모리(270)는 현재 픽처 내 움직임 정보가 도출된(또는 인코딩된) 블록의 움직임 정보 및/또는 이미 복원된 픽처 내 블록들의 움직임 정보를 저장할 수 있다. 상기 저장된 움직임 정보는 공간적 주변 블록의 움직임 정보 또는 시간적 주변 블록의 움직임 정보로 활용하기 위하여 인터 예측부(221)에 전달할 수 있다. 메모리(270)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(222)에 전달할 수 있다.
- [67] 한편, 본 문서에서 양자화/역양자화 및/또는 변환/역변환 중 적어도 하나는 생략될 수 있다. 상기 양자화/역양자화가 생략되는 경우, 상기 양자화된 변환 계수는 변환 계수라고 불릴 수 있다. 상기 변환/역변환이 생략되는 경우, 상기 변환 계수는 계수 또는 레지듀얼 계수라고 불릴 수도 있고, 또는 표현의 통일성을 위하여 변환 계수라고 여전히 불릴 수도 있다.
- [68] 또한, 본 문서에서 양자화된 변환 계수 및 변환 계수는 각각 변환 계수 및 스케일링된(scaled) 변환 계수라고 지칭될 수 있다. 이 경우 레지듀얼 정보는 변환 계수(들)에 관한 정보를 포함할 수 있고, 상기 변환 계수(들)에 관한 정보는 레지듀얼 코딩 신택스를 통하여 시그널링될 수 있다. 상기 레지듀얼 정보(또는 상기 변환 계수(들)에 관한 정보)를 기반으로 변환 계수들이 도출될 수 있고, 상기 변환 계수들에 대한 역변환(스케일링)을 통하여 스케일링된 변환 계수들이 도출될 수 있다. 상기 스케일링된 변환 계수들에 대한 역변환(변환)을 기반으로 레지듀얼 샘플들이 도출될 수 있다. 이는 본 문서의 다른 부분에서도 마찬가지로 적용/표현될 수 있다.
- [69] 도 3은 본 개시가 적용될 수 있는 비디오/영상 디코딩 장치의 구성을 개략적으로 설명하는 도면이다.
- [70] 도 3을 참조하면, 디코딩 장치(300)는 엔트로피 디코딩부(entropy decoder, 310), 레지듀얼 처리부(residual processor, 320), 예측부(predictor, 330), 가산부(adder, 340), 필터링부(filter, 350) 및 메모리(memory, 360)를 포함하여 구성될 수 있다. 예측부(330)는 인트라 예측부(331) 및 인터 예측부(332)를 포함할 수 있다. 레지듀얼 처리부(320)는 역양자화부(dequantizer, 321) 및 역변환부(inverse transformer, 321)를 포함할 수 있다. 상술한 엔트로피 디코딩부(310), 레지듀얼 처리부(320), 예측부(330), 가산부(340) 및 필터링부(350)는 실시예에 따라 하나의 하드웨어 컴포넌트(예를 들어 디코더 칩셋 또는 프로세서)에 의하여 구성될 수

있다. 또한 메모리(360)는 DPB(decoded picture buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구성될 수도 있다. 상기 하드웨어 컴포넌트는 메모리(360)을 내/외부 컴포넌트로 더 포함할 수도 있다.

[71] 비디오/영상 정보를 포함하는 비트스트림이 입력되면, 디코딩 장치(300)는 도 3의 인코딩 장치에서 비디오/영상 정보가 처리된 프로세스에 대응하여 영상을 복원할 수 있다. 예를 들어, 디코딩 장치(300)는 상기 비트스트림으로부터 획득한 블록 분할 관련 정보를 기반으로 유닛들/블록들을 도출할 수 있다. 디코딩 장치(300)는 인코딩 장치에서 적용된 처리 유닛을 이용하여 디코딩을 수행할 수 있다. 따라서 디코딩의 처리 유닛은 예를 들어 코딩 유닛일 수 있고, 코딩 유닛은 코딩 트리 유닛 또는 최대 코딩 유닛으로부터 쿼드 트리 구조, 바이너리 트리 구조 및/또는 터너리 트리 구조를 따라서 분할될 수 있다. 코딩 유닛으로부터 하나 이상의 변환 유닛이 도출될 수 있다. 그리고, 디코딩 장치(300)를 통해 디코딩 및 출력된 복원 영상 신호는 재생 장치를 통해 재생될 수 있다.

[72] 디코딩 장치(300)는 도 3의 인코딩 장치로부터 출력된 신호를 비트스트림 형태로 수신할 수 있고, 수신된 신호는 엔트로피 디코딩부(310)를 통해 디코딩될 수 있다. 예를 들어, 엔트로피 디코딩부(310)는 상기 비트스트림을 파싱하여 영상 복원(또는 픽처 복원)에 필요한 정보(ex. 비디오/영상 정보)를 도출할 수 있다. 상기 비디오/영상 정보는 어댑테이션 파라미터 세트(APS), 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS) 또는 비디오 파라미터 세트(VPS) 등 다양한 파라미터 세트에 관한 정보를 더 포함할 수 있다. 또한 상기 비디오/영상 정보는 일반 제한 정보(general constraint information)를 더 포함할 수 있다. 디코딩 장치는 상기 파라미터 세트에 관한 정보 및/또는 상기 일반 제한 정보를 더 기반으로 픽처를 디코딩할 수 있다. 본 문서에서 후술되는 시그널링/수신되는 정보 및/또는 신택스 요소들은 상기 디코딩 절차를 통하여 디코딩되어 상기 비트스트림으로부터 획득될 수 있다. 예컨대, 엔트로피 디코딩부(310)는 지수 곱셈 부호화, CAVLC 또는 CABAC 등의 코딩 방법을 기초로 비트스트림 내 정보를 디코딩하고, 영상 복원에 필요한 신택스 엘리먼트의 값, 레지듀얼에 관한 변환 계수의 양자화된 값 등을 출력할 수 있다. 보다 상세하게, CABAC 엔트로피 디코딩 방법은, 비트스트림에서 각 신택스 요소에 해당하는 빈을 수신하고, 디코딩 대상 신택스 요소 정보와 주변 및 디코딩 대상 블록의 디코딩 정보 혹은 이전 단계에서 디코딩된 심볼/빈의 정보를 이용하여 문맥(context) 모델을 결정하고, 결정된 문맥 모델에 따라 빈(bin)의 발생 확률을 예측하여 빈의 산술 디코딩(arithmetic decoding)을 수행하여 각 신택스 요소의 값에 해당하는 심볼을 생성할 수 있다. 이때, CABAC 엔트로피 디코딩 방법은 문맥 모델 결정 후 다음 심볼/빈의 문맥 모델을 위해 디코딩된 심볼/빈의 정보를 이용하여 문맥 모델을 업데이트할 수 있다. 엔트로피 디코딩부(310)에서 디코딩된 정보 중 예측에 관한 정보는 예측부(인터 예측부(332) 및 인트라 예측부(331))로 제공되고, 엔트로피 디코딩부(310)에서 엔트로피 디코딩이 수행된 레지듀얼 값, 즉 양자화된 변환

계수들 및 관련 파라미터 정보는 레지듀얼 처리부(320)로 입력될 수 있다. 레지듀얼 처리부(320)는 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플들, 레지듀얼 샘플 어레이)를 도출할 수 있다. 또한, 엔트로피 디코딩부(310)에서 디코딩된 정보 중 필터링에 관한 정보는 필터링부(350)으로 제공될 수 있다. 한편, 인코딩 장치로부터 출력된 신호를 수신하는 수신부(미도시)가 디코딩 장치(300)의 내/외부 엘리먼트로서 더 구성될 수 있고, 또는 수신부는 엔트로피 디코딩부(310)의 구성요소일 수도 있다. 한편, 본 문서에 따른 디코딩 장치는 비디오/영상/픽처 디코딩 장치라고 불릴 수 있고, 상기 디코딩 장치는 정보 디코더(비디오/영상/픽처 정보 디코더) 및 샘플 디코더(비디오/영상/픽처 샘플 디코더)로 구분할 수도 있다. 상기 정보 디코더는 상기 엔트로피 디코딩부(310)를 포함할 수 있고, 상기 샘플 디코더는 상기 역양자화부(321), 역변환부(322), 가산부(340), 필터링부(350), 메모리(360), 인터 예측부(332) 및 인트라 예측부(331) 중 적어도 하나를 포함할 수 있다.

- [73] 역양자화부(321)에서는 양자화된 변환 계수들을 역양자화하여 변환 계수들을 출력할 수 있다. 역양자화부(321)는 양자화된 변환 계수들을 2차원의 블록 형태로 재정렬할 수 있다. 이 경우 상기 재정렬은 인코딩 장치에서 수행된 계수 스캔 순서를 기반으로 재정렬을 수행할 수 있다. 역양자화부(321)는 양자화 파라미터(예를 들어 양자화 스텝 사이즈 정보)를 이용하여 양자화된 변환 계수들에 대한 역양자화를 수행하고, 변환 계수들(transform coefficient)을 획득할 수 있다.
- [74] 역변환부(322)에서는 변환 계수들을 역변환하여 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플 어레이)를 획득하게 된다.
- [75] 예측부는 현재 블록에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부는 엔트로피 디코딩부(310)로부터 출력된 상기 예측에 관한 정보를 기반으로 상기 현재 블록에 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있고, 구체적인 인트라/인터 예측 모드를 결정할 수 있다.
- [76] 예측부(330)는 후술하는 다양한 예측 방법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부는 하나의 블록에 대한 예측을 위하여 인트라 예측 또는 인터 예측을 적용할 수 있을 뿐 아니라, 인트라 예측과 인터 예측을 동시에 적용할 수 있다. 이는 combined inter and intra prediction (CIIP)라고 불릴 수 있다. 또한, 예측부는 블록에 대한 예측을 위하여 인트라 블록 카피(intra block copy, IBC) 예측 모드에 기반할 수도 있고 또는 팔레트 모드(palette mode)에 기반할 수도 있다. 상기 IBC 예측 모드 또는 팔레트 모드는 예를 들어 SCC(screen content coding) 등과 같이 게임 등의 콘텐츠 영상/동영상 코딩을 위하여 사용될 수 있다. IBC는 기본적으로 현재 픽처 내에서 예측을 수행하나 현재 픽처 내에서 참조 블록을 도출하는 점에서 인터 예측과 유사하게 수행될 수 있다. 즉, IBC는 본 문서에서 설명되는 인터 예측 기법들 중 적어도 하나를 이용할 수 있다. 팔레트

모드는 인트라 코딩 또는 인트라 예측의 일 예로 볼 수 있다. 팔레트 모드가 적용되는 경우 팔레트 테이블 및 팔레트 인덱스에 관한 정보가 상기 비디오/영상 정보에 포함되어 시그널링될 수 있다.

- [77] 인트라 예측부(331)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 상기 참조되는 샘플들은 예측 모드에 따라 상기 현재 블록의 주변(neighbor)에 위치할 수 있고, 또는 떨어져서 위치할 수도 있다. 인트라 예측에서 예측 모드들은 복수의 비방향성 모드와 복수의 방향성 모드를 포함할 수 있다. 인트라 예측부(331)는 주변 블록에 적용된 예측 모드를 이용하여, 현재 블록에 적용되는 예측 모드를 결정할 수도 있다.
- [78] 인터 예측부(332)는 참조 픽처 상에서 움직임 벡터에 의해 특정되는 참조 블록(참조 샘플 어레이)을 기반으로, 현재 블록에 대한 예측된 블록을 유도할 수 있다. 이때, 인터 예측 모드에서 전송되는 움직임 정보의 양을 줄이기 위해 주변 블록과 현재 블록 간의 움직임 정보의 상관성에 기초하여 움직임 정보를 블록, 서브블록 또는 샘플 단위로 예측할 수 있다. 상기 움직임 정보는 움직임 벡터 및 참조 픽처 인덱스를 포함할 수 있다. 상기 움직임 정보는 인터 예측 방향(L0 예측, L1 예측, Bi 예측 등) 정보를 더 포함할 수 있다. 인터 예측의 경우에, 주변 블록은 현재 픽처 내에 존재하는 공간적 주변 블록(spatial neighboring block)과 참조 픽처에 존재하는 시간적 주변 블록(temporal neighboring block)을 포함할 수 있다. 예를 들어, 인터 예측부(332)는 주변 블록들을 기반으로 움직임 정보 후보 리스트를 구성하고, 수신한 후보 선택 정보를 기반으로 상기 현재 블록의 움직임 벡터 및/또는 참조 픽처 인덱스를 도출할 수 있다. 다양한 예측 모드를 기반으로 인터 예측이 수행될 수 있으며, 상기 예측에 관한 정보는 상기 현재 블록에 대한 인터 예측의 모드를 지시하는 정보를 포함할 수 있다.
- [79] 가산부(340)는 획득된 레지듀얼 신호를 예측부(인트라 예측부(332) 및/또는 인트라 예측부(331) 포함)로부터 출력된 예측 신호(예측된 블록, 예측 샘플 어레이)에 더함으로써 복원 신호(복원 픽처, 복원 블록, 복원 샘플 어레이)를 생성할 수 있다. 스킵 모드가 적용된 경우와 같이 처리 대상 블록에 대한 레지듀얼이 없는 경우, 예측된 블록이 복원 블록으로 사용될 수 있다.
- [80] 가산부(340)는 복원부 또는 복원 블록 생성부라고 불릴 수 있다. 생성된 복원 신호는 현재 픽처 내 다음 처리 대상 블록의 인트라 예측을 위하여 사용될 수 있고, 후술하는 바와 같이 필터링을 거쳐서 출력될 수도 있고 또는 다음 픽처의 인터 예측을 위하여 사용될 수도 있다.
- [81] 한편, 픽처 디코딩 과정에서 LMCS (luma mapping with chroma scaling)가 적용될 수도 있다.
- [82] 필터링부(350)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(350)는 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정된 복원 픽처를 메모리(360), 구체적으로 메모리(360)의 DPB에 전송할 수 있다. 상기

다양한 필터링 방법은 예를 들어, 디블록킹 필터링, 샘플 적응적 오프셋(sample adaptive offset), 적응적 루프 필터(adaptive loop filter), 양방향 필터(bilateral filter) 등을 포함할 수 있다.

- [83] 메모리(360)의 DPB에 저장된 (수정된) 복원 픽처는 인터 예측부(332)에서 참조 픽처로 사용될 수 있다. 메모리(360)는 현재 픽처 내 움직임 정보가 도출된(또는 디코딩된) 블록의 움직임 정보 및/또는 이미 복원된 픽처 내 블록들의 움직임 정보를 저장할 수 있다. 상기 저장된 움직임 정보는 공간적 주변 블록의 움직임 정보 또는 시간적 주변 블록의 움직임 정보로 활용하기 위하여 인터 예측부(332)에 전달할 수 있다. 메모리(360)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(331)에 전달할 수 있다.
- [84] 본 명세서에서, 인코딩 장치(100)의 필터링부(260), 인터 예측부(221) 및 인트라 예측부(222)에서 설명된 실시예들은 각각 디코딩 장치(300)의 필터링부(350), 인터 예측부(332) 및 인트라 예측부(331)에도 동일 또는 대응되도록 적용될 수 있다.
- [85] 상술한 바와 같이 비디오 코딩을 수행함에 있어 압축 효율을 높이기 위하여 예측을 수행한다. 이를 통하여 코딩 대상 블록인 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록을 생성할 수 있다. 여기서 상기 예측된 블록은 공간 도메인(또는 픽셀 도메인)에서의 예측 샘플들을 포함한다. 상기 예측된 블록은 인코딩 장치 및 디코딩 장치에서 동일하게 도출되며, 상기 인코딩 장치는 원본 블록의 원본 샘플 값 자체가 아닌 상기 원본 블록과 상기 예측된 블록 간의 레지듀얼에 대한 정보(레지듀얼 정보)를 디코딩 장치로 시그널링함으로써 영상 코딩 효율을 높일 수 있다. 디코딩 장치는 상기 레지듀얼 정보를 기반으로 레지듀얼 샘플들을 포함하는 레지듀얼 블록을 도출하고, 상기 레지듀얼 블록과 상기 예측된 블록을 합하여 복원 샘플들을 포함하는 복원 블록을 생성할 수 있고, 복원 블록들을 포함하는 복원 픽처를 생성할 수 있다.
- [86] 상기 레지듀얼 정보는 변환 및 양자화 절차를 통하여 생성될 수 있다. 예를 들어, 인코딩 장치는 상기 원본 블록과 상기 예측된 블록 간의 레지듀얼 블록을 도출하고, 상기 레지듀얼 블록에 포함된 레지듀얼 샘플들(레지듀얼 샘플 어레이)에 변환 절차를 수행하여 변환 계수들을 도출하고, 상기 변환 계수들에 양자화 절차를 수행하여 양자화된 변환 계수들을 도출하여 관련된 레지듀얼 정보를 (비트스트림을 통하여) 디코딩 장치로 시그널링할 수 있다. 여기서 상기 레지듀얼 정보는 상기 양자화된 변환 계수들의 값 정보, 위치 정보, 변환 기법, 변환 커널, 양자화 파라미터 등의 정보를 포함할 수 있다. 디코딩 장치는 상기 레지듀얼 정보를 기반으로 역양자화/역변환 절차를 수행하고 레지듀얼 샘플들(또는 레지듀얼 블록)을 도출할 수 있다. 디코딩 장치는 예측된 블록과 상기 레지듀얼 블록을 기반으로 복원 픽처를 생성할 수 있다. 인코딩 장치는 또한 이후 픽처의 인터 예측을 위한 참조를 위하여 양자화된 변환 계수들을 역양자화/역변환하여 레지듀얼 블록을 도출하고, 이를 기반으로 복원 픽처를

생성할 수 있다.

[87] 도 4는 인터 예측 절차를 예시적으로 나타낸다.

[88] 도 4를 참조하면, 인터 예측 절차는 인터 예측 모드 결정 단계, 결정된 예측 모드에 따른 움직임 정보 도출 단계, 도출된 움직임 정보에 기반한 예측 수행(예측 샘플 생성) 단계를 포함할 수 있다. 상기 인터 예측 절차는 상술한 바와 같이 인코딩 장치 및 디코딩 장치에서 수행될 수 있다. 본 문서에서 코딩 장치라 함은 인코딩 장치 및/또는 디코딩 장치를 포함할 수 있다.

[89] 도 4를 참조하면, 코딩 장치는 현재 블록에 대한 인터 예측 모드를 결정한다(S400). 픽처 내 현재 블록의 예측을 위하여 다양한 인터 예측 모드가 사용될 수 있다. 예를 들어, 머지 모드, 스킵 모드, MVP(motion vector prediction) 모드, 어파인(Affine) 모드, 서브블록 머지 모드, MMVD (merge with MVD) 모드 등 다양한 모드가 사용될 수 있다. DMVR (Decoder side motion vector refinement) 모드, AMVR(adaptive motion vector resolution) 모드, Bi-prediction with CU-level weight (BCW), Bi-directional optical flow (BDOF) 등이 부수적인 모드로 더 혹은 대신 사용될 수 있다. 어파인 모드는 어파인 움직임 예측(affine motion prediction) 모드라고 불릴 수도 있다. MVP 모드는 AMVP(advanced motion vector prediction) 모드라고 불릴 수도 있다. 본 문서에서 일부 모드 및/또는 일부 모드에 의하여 도출된 움직임 정보 후보는 다른 모드의 움직임 정보 관련 후보들 중 하나로 포함될 수도 있다. 예를 들어, HMVP 후보는 상기 머지/스킵 모드의 머지 후보로 추가될 수 있고, 또는 상기 MVP 모드의 mvp 후보로 추가될 수도 있다. 상기 HMVP 후보가 상기 머지 모드 또는 스킵 모드의 움직임 정보 후보로 사용되는 경우, 상기 HMVP 후보는 HMVP 머지 후보라고 불릴 수 있다.

[90] 현재 블록의 인터 예측 모드를 가리키는 예측 모드 정보가 인코딩 장치로부터 디코딩 장치로 시그널링될 수 있다. 상기 예측 모드 정보는 비트스트림에 포함되어 디코딩 장치에 수신될 수 있다. 상기 예측 모드 정보는 다수의 후보 모드들 중 하나를 지시하는 인덱스 정보를 포함할 수 있다. 또는, 플래그 정보의 계층적 시그널링을 통하여 인터 예측 모드를 지시할 수도 있다. 이 경우 상기 예측 모드 정보는 하나 이상의 플래그들을 포함할 수 있다. 예를 들어, 스킵 플래그를 시그널링하여 스킵 모드 적용 여부를 지시하고, 스킵 모드가 적용 안되는 경우에 머지 플래그를 시그널링하여 머지 모드 적용 여부를 지시하고, 머지 모드가 적용 안되는 경우에 MVP 모드 적용되는 것으로 지시하거나 추가적인 구분을 위한 플래그를 더 시그널링할 수도 있다. 어파인 모드는 독립적인 모드로 시그널링될 수도 있고, 또는 머지 모드 또는 MVP 모드 등에 종속적인 모드로 시그널링될 수도 있다. 예를 들어, 어파인 모드는 어파인 머지 모드 및 어파인 MVP 모드를 포함할 수 있다.

[91] 코딩 장치는 상기 현재 블록에 대한 움직임 정보를 도출한다(S410). 상기 움직임 정보 도출을 상기 인터 예측 모드를 기반으로 도출될 수 있다.

[92] 코딩 장치는 현재 블록의 움직임 정보를 이용하여 인터 예측을 수행할 수 있다.

인코딩 장치는 움직임 추정(motion estimation) 절차를 통하여 현재 블록에 대한 최적의 움직임 정보를 도출할 수 있다. 예를 들어, 인코딩 장치는 현재 블록에 대한 원본 픽처 내 원본 블록을 이용하여 상관성이 높은 유사한 참조 블록을 참조 픽처 내의 정해진 탐색 범위 내에서 분수 픽셀 단위로 탐색할 수 있고, 이를 통하여 움직임 정보를 도출할 수 있다. 블록의 유사성은 위상(phase) 기반 샘플 값들의 차를 기반으로 도출할 수 있다. 예를 들어, 블록의 유사성은 현재 블록(or 현재 블록의 템플릿)과 참조 블록(or 참조 블록의 템플릿) 간 SAD를 기반으로 계산될 수 있다. 이 경우 탐색 영역 내 SAD가 가장 작은 참조 블록을 기반으로 움직임 정보를 도출할 수 있다. 도출된 움직임 정보는 인터 예측 모드 기반으로 여러 방법에 따라 디코딩 장치로 시그널링될 수 있다.

- [93] 코딩 장치는 상기 현재 블록에 대한 움직임 정보를 기반으로 인터 예측을 수행한다(S420). 코딩 장치는 상기 움직임 정보를 기반으로 상기 현재 블록에 대한 예측 샘플(들)을 도출할 수 있다. 상기 예측 샘플들을 포함하는 현재 블록은 예측된 블록이라고 불릴 수 있다.
- [94] 도 5는 인터 예측에서 머지 모드를 설명하기 위한 도면이다.
- [95] 머지 모드(merge mode)가 적용되는 경우, 현재 예측 블록의 움직임 정보가 직접적으로 전송되지 않고, 주변 예측 블록의 움직임 정보를 이용하여 상기 현재 예측 블록의 움직임 정보를 유도하게 된다. 따라서, 머지 모드를 이용하였음을 알려주는 플래그 정보 및 주변의 어떤 예측 블록을 이용하였는지를 알려주는 머지 인덱스를 전송함으로써 현재 예측 블록의 움직임 정보를 지시할 수 있다. 상기 머지 모드는 레귤러 머지 모드(regular merge mode)라고 불릴 수 있다. 예를 들어, 상기 머지 모드는 `regular_merge_flag` 신텍스 요소의 값이 1인 경우에 적용될 수 있다.
- [96] 인코딩 장치는 머지 모드를 수행하기 위해서 현재 예측 블록의 움직임 정보를 유도하기 위해 이용되는 머지 후보 블록(merge candidate block)을 쉼치(search)하여야 한다. 예를 들어, 상기 머지 후보 블록은 최대 5개까지 이용될 수 있으나, 본 문서의 실시예(들)은 이에 한정되지 않는다. 그리고, 상기 머지 후보 블록의 최대 개수는 슬라이스 헤더 또는 타일 그룹 헤더에서 전송될 수 있으나, 본 문서의 실시예(들)은 이에 한정되지 않는다. 상기 머지 후보 블록들을 찾은 후, 인코딩 장치는 머지 후보 리스트를 생성할 수 있고, 이들 중 가장 작은 비용을 갖는 머지 후보 블록을 최종 머지 후보 블록으로 선택할 수 있다.
- [97] 본 문서는 상기 머지 후보 리스트를 구성하는 머지 후보 블록에 대한 다양한 실시예를 제공할 수 있다.
- [98] 예를 들어, 상기 머지 후보 리스트는 5개의 머지 후보 블록을 이용할 수 있다. 예를 들어, 4개의 공간적 머지 후보(spatial merge candidate)와 1개의 시간적 머지 후보(temporal merge candidate)를 이용할 수 있다. 구체적인 예로, 공간적 머지 후보의 경우, 도 4에 도시된 블록들을 공간적 머지 후보로 이용할 수 있다. 이하, 상기 공간적 머지 후보 또는 후술하는 공간적 MVP 후보는 SMVP로 불릴 수

있고, 상기 시간적 머지 후보 또는 후술하는 시간적 MVP 후보는 TMVP로 불릴 수 있다.

[99] 상기 현재 블록에 대한 머지 후보 리스트는 예를 들어 다음과 같은 절차를 기반으로 구성될 수 있다.

[100] 코딩 장치(인코딩 장치/디코딩 장치)는 현재 블록의 공간적 주변 블록들을 탐색하여 도출된 공간적 머지 후보들을 머지 후보 리스트에 삽입할 수 있다. 예를 들어, 상기 공간적 주변 블록들은 상기 현재 블록의 좌하측 코너 주변 블록, 좌측 주변 블록, 우상측 코너 주변 블록, 상측 주변 블록, 좌상측 코너 주변 블록들을 포함할 수 있다. 다만, 이는 예시로서 상술한 공간적 주변 블록들 이외에도 우측 주변 블록, 하측 주변 블록, 우하측 주변 블록 등 추가적인 주변 블록들이 더 상기 공간적 주변 블록들로 사용될 수 있다. 코딩 장치는 상기 공간적 주변 블록들을 우선 순위를 기반으로 탐색하여 가용한 블록들을 검출하고, 검출된 블록들의 움직임 정보를 상기 공간적 머지 후보들로 도출할 수 있다. 예를 들어, 인코딩 장치 또는 디코딩 장치는 도 5에 도시된 5개의 블록들을 $A1 \rightarrow B1 \rightarrow B0 \rightarrow A0 \rightarrow B2$ 와 같이 순서대로 탐색하고, 가용한 후보들을 순차적으로 인덱싱하여 머지 후보 리스트로 구성할 수 있다.

[101] 코딩 장치는 상기 현재 블록의 시간적 주변 블록을 탐색하여 도출된 시간적 머지 후보를 상기 머지 후보 리스트에 삽입할 수 있다. 상기 시간적 주변 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 픽처인 참조 픽처 상에 위치할 수 있다. 상기 시간적 주변 블록이 위치하는 참조 픽처는 콜로케이티드(collocated) 픽처 또는 콜(col) 픽처라고 불릴 수 있다. 상기 시간적 주변 블록은 상기 콜(col) 픽처 상에서의 상기 현재 블록에 대한 동일 위치 블록(co-located block)의 우하측 코너 주변 블록 및 우하측 센터 블록의 순서로 탐색될 수 있다. 한편, 움직임 정보 압축(motion data compression)이 적용되는 경우, 상기 콜(col) 픽처에 일정 저장 단위마다 특정 움직임 정보를 대표 움직임 정보로 저장할 수 있다. 이 경우, 상기 일정 저장 단위 내의 모든 블록에 대한 움직임 정보를 저장할 필요가 없으며 이를 통하여 움직임 정보 압축(motion data compression) 효과를 얻을 수 있다. 이 경우, 일정 저장 단위는 예를 들어, 16×16 샘플 단위, 또는 8×8 샘플 단위 등으로 미리 정해질 수도 있고, 또는 인코딩 장치에서 디코딩 장치로 상기 일정 저장 단위에 대한 사이즈 정보가 시그널링될 수도 있다. 상기 움직임 정보 압축(motion data compression)이 적용되는 경우 상기 시간적 주변 블록의 움직임 정보는 상기 시간적 주변 블록이 위치하는 상기 일정 저장 단위의 대표 움직임 정보로 대체될 수 있다. 즉, 이 경우 구현 측면에서 보면, 상기 시간적 주변 블록의 좌표에 위치하는 예측 블록이 아닌, 상기 시간적 주변 블록의 좌표(좌상단 샘플 포지션)를 기반으로 일정 값만큼 산술적 오른쪽 쉬프트 후 산술적 왼쪽 쉬프트한 위치를 커버하는 예측 블록의 움직임 정보를 기반으로 상기 시간적 머지 후보가 도출될 수 있다. 예를 들어, 상기 일정 저장 단위가 $2n \times 2n$ 샘플 단위인 경우, 상기 시간적 주변 블록의 좌표가 $(xTnb, yTnb)$ 라 하면, 수정된 위치인

$((x_{Tnb} > n) < n)$, $(y_{Tnb} > n) < n)$ 에 위치하는 예측 블록의 움직임 정보가 상기 시간적 머지 후보를 위하여 사용될 수 있다. 구체적으로 예를 들어, 상기 일정 저장 단위가 16×16 샘플 단위인 경우, 상기 시간적 주변 블록의 좌표가 (x_{Tnb}, y_{Tnb}) 라 하면, 수정된 위치인 $((x_{Tnb} > 4) < 4)$, $(y_{Tnb} > 4) < 4)$ 에 위치하는 예측 블록의 움직임 정보가 상기 시간적 머지 후보를 위하여 사용될 수 있다. 또는 예를 들어, 상기 일정 저장 단위가 8×8 샘플 단위인 경우, 상기 시간적 주변 블록의 좌표가 (x_{Tnb}, y_{Tnb}) 라 하면, 수정된 위치인 $((x_{Tnb} > 3) < 3)$, $(y_{Tnb} > 3) < 3)$ 에 위치하는 예측 블록의 움직임 정보가 상기 시간적 머지 후보를 위하여 사용될 수 있다.

- [102] 코딩 장치는 현재 머지 후보들의 개수가 최대 머지 후보들의 개수보다 작은지 여부를 확인할 수 있다. 상기 최대 머지 후보들의 개수는 미리 정의되거나 인코딩 장치에서 디코딩 장치로 시그널링될 수 있다. 예를 들어, 인코딩 장치는 상기 최대 머지 후보들의 개수에 관한 정보를 생성하고, 인코딩하여 비트스트림 형태로 상기 디코더로 전달할 수 있다. 상기 최대 머지 후보들의 개수가 다 채워지면 이후의 후보 추가 과정은 진행하지 않을 수 있다.
- [103] 상기 확인 결과 상기 현재 머지 후보들의 개수가 상기 최대 머지 후보들의 개수보다 작은 경우, 코딩 장치는 추가 머지 후보를 상기 머지 후보 리스트에 삽입할 수 있다. 예를 들어, 상기 추가 머지 후보는 후술하는 히스토리 기반 머지 후보(들)(history based merge candidate(s)), 페어와이즈 평균 머지 후보(들)(pair-wise average merge candidate(s)), ATMVP, 조합된 쌍 예측(combined bi-predictive) 머지 후보 (현재 슬라이스/타일 그룹의 슬라이스/타일 그룹 타입이 B 타입인 경우) 및/또는 영 벡터 머지 후보 중 적어도 하나를 포함할 수 있다.
- [104] 상기 확인 결과 상기 현재 머지 후보들의 개수가 상기 최대 머지 후보들의 개수보다 작지 않은 경우, 코딩 장치는 상기 머지 후보 리스트의 구성을 종료할 수 있다. 이 경우 인코딩 장치는 RD(rate-distortion) 코스트(cost) 기반으로 상기 머지 후보 리스트를 구성하는 머지 후보들 중 최적의 머지 후보를 선택할 수 있으며, 상기 선택된 머지 후보를 가리키는 선택 정보(ex. merge index)를 디코딩 장치로 시그널링할 수 있다. 디코딩 장치는 상기 머지 후보 리스트 및 상기 선택 정보를 기반으로 상기 최적의 머지 후보를 선택할 수 있다.
- [105] 상기 선택된 머지 후보의 움직임 정보가 상기 현재 블록의 움직임 정보로 사용될 수 있으며, 상기 현재 블록의 움직임 정보를 기반으로 상기 현재 블록의 예측 샘플들을 도출할 수 있음을 상술한 바와 같다. 인코딩 장치는 상기 예측 샘플들을 기반으로 상기 현재 블록의 레지듀얼 샘플들을 도출할 수 있으며, 상기 레지듀얼 샘플들에 관한 레지듀얼 정보를 디코딩 장치로 시그널링할 수 있다. 디코딩 장치는 상기 레지듀얼 정보를 기반으로 도출된 레지듀얼 샘플들 및 상기 예측 샘플들을 기반으로 복원 샘플들을 생성할 수 있고, 이를 기반으로 복원 픽처를 생성할 수 있음은 상술한 바와 같다.
- [106] 스킵 모드(skip mode)가 적용되는 경우, 앞에서 머지 모드가 적용되는 경우와

동일한 방법으로 상기 현재 블록의 움직임 정보를 도출할 수 있다. 다만, 스킵 모드가 적용되는 경우 해당 블록에 대한 레지듀얼 신호가 생략되며 따라서 예측 샘플들이 바로 복원 샘플들로 이용될 수 있다. 상기 스킵 모드는 예를 들어 `cu_skip_flag` 신택스 요소의 값이 1인 경우에 적용될 수 있다.

[107] 한편, 상기 페어와이즈 평균 머지 후보(pair-wise average merge candidate)는 페어와이즈 평균 후보 또는 페어와이즈 후보라고 불릴 수 있다. 페어와이즈 평균 후보(들)는 기존의(existing) 머지 후보 리스트에서 미리 정의된 후보들의 페어들을 평균하여 생성될 수 있다. 또한, 미리 정의된 페어들은 $\{(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3)\}$ 와 같이 정의될 수 있다. 여기서, 숫자들은 머지 후보 리스트에 대한 머지 인덱스들을 나타낼 수 있다. 평균화된 움직임 벡터(averaged motion vector)는 각 참조 리스트에 대해 별도로 계산될 수 있다. 예를 들어, 2개의 움직임 벡터들이 하나의 리스트 내에서 이용 가능한 경우, 상기 2개의 움직임 벡터들은 서로 다른 참조 픽처들을 가리켜도 평균화될 수 있다. 예를 들어, 하나의 움직임 벡터만이 이용 가능한 경우, 곧바로(directly) 하나가 사용될 수 있다. 예를 들어, 이용 가능한 움직임 벡터가 없는 경우, 리스트를 유효하지 않은 상태로 유지할 수 있다.

[108] 예를 들어, 페어와이즈 평균 머지 후보들이 추가된 이후에도 머지 후보 리스트가 가득차지 않은 경우 즉, 머지 후보 리스트 내에 현재 머지 후보들의 개수가 최대 머지 후보들의 개수보다 작은 경우, 최대 머지 후보 번호가 나타날 때까지 영 벡터(zero MVP)가 마지막으로 삽입될 수 있다. 즉, 머지 후보 리스트 내의 현재 머지 후보들의 개수가 최대 머지 후보 개수가 될 때까지 영 벡터를 삽입할 수 있다.

[109] 한편, 기존에는 코딩 블록의 움직임을 표현하기 위해 오직 하나의 움직임 벡터를 사용할 수 있었다. 즉, 옮김(translation) 움직임 모델이 사용될 수 있었다. 다만, 이러한 방법이 블록 단위에서 최적의 움직임을 표현했을 수 있으나, 실제 각 샘플의 최적의 움직임은 아니며, 샘플 단위에서 최적의 움직임 벡터를 결정할 수 있다면 코딩 효율을 높일 수 있다. 이를 위해, 어파인 움직임 모델(affine motion model)이 사용될 수 있다. 어파인 움직임 모델을 사용하여 코딩하는 어파인 움직임 예측 방법은 후술하는 내용과 같이 4가지 움직임을 효율적으로 표현할 수 있다.

[110] 도 6은 어파인 움직임 모델을 통하여 표현되는 움직임을 예시적으로 나타낸다.

[111] 도 6을 참조하면 상기 어파인 움직임 모델을 통하여 표현될 수 있는 움직임은 병진(translate) 움직임, 스케일(scale) 움직임, 회전(rotate) 움직임 및 전단(shear) 움직임을 포함할 수 있다. 즉, 도 6에 도시된 시간의 흐름에 따라 영상(의 일부)이 평면 이동하는 병진 움직임뿐만 아니라, 시간의 흐름에 따라 영상(의 일부)이 스케일(scale)되는 스케일 움직임, 시간의 흐름에 따라 영상(의 일부)이 회전하는 회전 움직임, 시간의 흐름에 따라 영상(의 일부)이 평형 사변형 변형되는 전단 움직임을 상기 어파인 움직임 예측을 통하여 효율적으로 표현할 수 있다.

- [112] 인코딩 장치/디코딩 장치는 상기 어파인 움직임 예측을 통하여 현재 블록의 컨트롤 포인트(control point, CP)들에서의 움직임 벡터들을 기반으로 영상의 왜곡 형태를 예측할 수 있고, 이를 통하여 예측의 정확도를 높임으로서 영상의 압축 성능을 향상시킬 수 있다. 또한, 상기 현재 블록의 주변 블록의 움직임 벡터를 이용하여 상기 현재 블록의 적어도 하나의 컨트롤 포인트에 대한 움직임 벡터가 유도될 수 있는바, 추가되는 부가 정보에 대한 데이터량 부담을 줄이고, 인터 예측 효율을 상당히 향상시킬 수 있다.
- [113] 어파인 움직임 모델이 표현할 수 있는 움직임 중 3가지 움직임(움김(translation), 스케일(scale), 회전(rotate))을 표현하는 어파인 움직임 모델은 유사(similarity)(또는 간소화된(simplified)) 어파인 움직임 모델이라고 불릴 수 있다. 그러나, 어파인 움직임 모델이 상술한 움직임 모델에 한정되는 것은 아니다.
- [114] 어파인 움직임 예측 방법은 2개, 3개 또는 4개의 움직임 벡터를 이용하여 블록의 각 샘플 단위에서 움직임 벡터를 표현할 수 있다.
- [115] 도 7a 및 도 7b는 어파인 움직임 예측을 위한 CPMV를 예시적으로 나타낸다.
- [116] 어파인 움직임 예측은 2개 이상의 컨트롤 포인트 움직임 벡터(Control Point Motion Vector, CPMV)들을 이용하여 블록이 포함하는 샘플 위치의 움직임 벡터를 결정할 수 있다. 이 때, 움직임 벡터들의 집합을 어파인 움직임 벡터 필드(MVF: Motion Vector Field)라고 나타낼 수 있다.
- [117] 예를 들어, 도 7a는 2개의 CPMV가 이용되는 경우를 나타낼 수 있으며, 이는 4-파라미터 어파인 모델이라고 불릴 수 있다. 이 경우, (x, y) 샘플 위치에서의 움직임 벡터는 예를 들어 수학식 1과 같이 결정될 수 있다.
- [118] [수식1]

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{1y}-mv_{0y}}{W}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{1x}-mv_{0x}}{W}y + mv_{0y} \end{cases}$$

- [119] 예를 들어, 도 7b는 3개의 CPMV가 이용되는 경우를 나타낼 수 있으며, 이는 6-파라미터 어파인 모델이라고 불릴 수 있다. 이 경우, (x, y) 샘플 위치에서의 움직임 벡터는 예를 들어 수학식 2와 같이 결정될 수 있다.
- [120] [수식2]

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{2x}-mv_{0x}}{H}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{2y}-mv_{0y}}{H}y + mv_{0y} \end{cases}$$

- [121] 수학식 1 및 수학식 2에서, $\{v_x, v_y\}$ 는 (x, y) 위치에서의 움직임 벡터를 나타낼 수

있다. 또한, $\{v_{0x}, v_{0y}\}$ 는 코딩 블록의 좌상단 코너 위치의 컨트롤 포인트(CP: Control Point)의 CPMV를 나타낼 수 있고, $\{v_{1x}, v_{1y}\}$ 는 우상단 코너 위치의 CP의 CPMV를 나타낼 수 있고, $\{v_{2x}, v_{2y}\}$ 는 좌하단 코너 위치의 CP의 CPMV를 나타낼 수 있다. 또한, W는 현재 블록의 너비(width)를 나타낼 수 있고, H는 현재 블록의 높이(height)를 나타낼 수 있다.

- [122] 도 8은 어파인 MVF가 서브블록 단위에서 결정되는 경우를 예시적으로 나타낸다.
- [123] 인코딩/디코딩 과정에서 어파인 MVF는 샘플 단위 혹은 이미 정의된 서브블록 단위에서 결정될 수 있다. 예를 들어, 샘플 단위로 결정하는 경우, 각 샘플 값을 기준으로 움직임 벡터가 얻어질 수 있다. 또는 예를 들어, 서브블록 단위로 결정하는 경우, 서브블록의 중앙(센터 우하측, 즉 중앙 4개의 샘플들 중 우하측 샘플) 샘플 값 기준으로 해당 블록의 움직임 벡터를 얻을 수 있다. 즉, 어파인 움직임 예측에서 현재 블록의 움직임 벡터는 샘플 단위 또는 서브블록 단위로 도출될 수 있다.
- [124] 도 8의 경우 어파인 MVF가 4x4 서브블록 단위에서 결정되나, 서브블록의 사이즈는 다양하게 변형될 수 있다.
- [125] 즉, 어파인 예측이 가용한 경우, 현재 블록에 적용 가능한 움직임 모델은 3가지(움김 움직임 모델(translational motion model), 4-파라미터 어파인 움직임 모델(4-parameter affine motion model), 6-파라미터 어파인 움직임 모델(6-parameter affine motion model))을 포함할 수 있다. 여기서 움김 움직임 모델은 기존 블록 단위 움직임 벡터가 사용되는 모델을 나타낼 수 있고, 4-파라미터 어파인 움직임 모델은 2개의 CPMV가 사용되는 모델을 나타낼 수 있고, 6-파라미터 어파인 움직임 모델은 3개의 CPMV가 사용되는 모델을 나타낼 수 있다.
- [126] 한편, 어파인 움직임 예측은 어파인 MVP(또는 어파인 인터) 모드 또는 어파인 머지 모드를 포함할 수 있다.
- [127] 도 9는 인터 예측에서 어파인 머지 모드를 설명하기 위한 도면이다.
- [128] 예를 들어, 어파인 머지 모드에서 CPMV는 어파인 움직임 예측으로 코딩된 주변 블록의 어파인 움직임 모델에 따라 결정될 수 있다. 예를 들어, 씨치(search) 순서 상의 어파인 움직임 예측으로 코딩된 주변 블록이 어파인 머지 모드를 위해 사용될 수 있다. 즉, 주변 블록들 중 적어도 하나가 어파인 움직임 예측으로 코딩된 경우, 현재 블록은 어파인 머지 모드로 코딩될 수 있다. 여기서, 어파인 머지 모드는 AF_MERGE라고 불릴 수 있다.
- [129] 어파인 머지 모드가 적용되는 경우, 주변 블록의 CPMV들을 이용하여 현재 블록의 CPMV들이 도출될 수 있다. 이 경우 주변 블록의 CPMV들이 그대로 현재 블록의 CPMV들로 사용될 수도 있고, 주변 블록의 CPMV들이 상기 주변 블록의 사이즈 및 상기 현재 블록의 사이즈 등을 기반으로 수정되어 현재 블록의 CPMV들로 사용될 수 있다.

- [130] 한편, 서브블록 단위로 움직임 벡터(MV: Motion Vector)가 도출되는 어파인 머지 모드의 경우에는, 서브블록 머지 모드라고 불릴 수 있으며, 이는 서브블록 머지 플래그(또는 merge_subblock_flag 신텍스 요소)를 기반으로 지시될 수 있다. 또는 merge_subblock_flag 신텍스 요소의 값이 1인 경우, 서브블록 머지 모드가 적용되는 것이 지시될 수 있다. 이 경우 후술하는 어파인 머지 후보 리스트는 서브블록 머지 후보 리스트라고 불릴 수도 있다. 이 경우, 상기 서브블록 머지 후보 리스트에는 후술하는 SbTMVP로 도출된 후보가 더 포함될 수 있다. 이 경우, 상기 SbTMVP로 도출된 후보는 상기 서브블록 머지 후보 리스트의 0번째 인덱스의 후보로 이용될 수 있다. 다시 말해, 상기 SbTMVP로 도출된 후보는 상기 서브블록 머지 후보 리스트 내에서 후술하는 승계된 어파인 후보(inherited affine candidate) 또는 구성된 어파인 후보(constructed affine candidate)보다 앞에 위치할 수 있다.
- [131] 어파인 머지 모드가 적용되는 경우, 현재 블록에 대한 CPMV들 도출을 위하여 어파인 머지 후보 리스트가 구성될 수 있다. 예를 들어, 어파인 머지 후보 리스트는 다음 후보들 중 적어도 하나를 포함할 수 있다. 1) 승계된(inherited) 어파인 머지 후보. 2) 구성된(constructed) 어파인 머지 후보. 3) 제로 움직임 벡터 후보(또는 영 벡터). 여기서, 상기 승계된 어파인 머지 후보는 주변 블록이 어파인 모드로 코딩된 경우, 주변 블록의 CPMVs를 기반으로 도출되는 후보이고, 상기 구성된 어파인 머지 후보는 각 CPMV 단위로 해당 CP의 주변 블록의 MV를 기반으로 CPMVs를 구성하여 도출된 후보이고, 제로 움직임 벡터 후보는 그 값이 0인 CPMV들로 구성된 후보를 나타낼 수 있다.
- [132] 상기 어파인 머지 후보 리스트는 예를 들어 다음과 같이 구성될 수 있다.
- [133] 최대 2개의 승계된 어파인 후보가 있을 수 있고, 승계된 어파인 후보는 주변 블록들의 어파인 움직임 모델로부터 도출될 수 있다. 주변 블록들은 하나의 좌측 주변 블록과 상측의 주변 블록을 포함할 수 있다. 후보 블록들은 도 4와 같이 위치할 수 있다. 좌측 예측자(left predictor)를 위한 스캔 순서는 A1->A0일 수 있고, 상측 예측자(above predictor)를 위한 스캔 순서는 B1->B0->B2일 수 있다. 좌측 및 상측 각각으로부터 하나의 승계된 후보만이 선택될 수 있다. 두 개의 승계된 후보들 간에는 프루닝 체크(pruning check)가 수행되지 않을 수 있다.
- [134] 주변 어파인 블록이 확인되는 경우, 확인한 블록의 컨트롤 포인트 움직임 벡터들이 현재 블록의 어파인 머지 리스트 내의 CPMVP 후보를 도출하기 위해 사용될 수 있다. 여기서, 주변 어파인 블록은 현재 블록의 주변 블록 중 어파인 예측 모드로 코딩된 블록을 나타낼 수 있다. 예를 들어, 도 8을 참조하면, 좌하측(bottom-left) 주변 블록 A가 어파인 예측 모드로 코딩된 경우, 주변 블록 A의 좌상측(top-left) 코너, 우상측(top-right) 코너 및 좌하측(bottom-left) 코너의 움직임 벡터 v2, v3 및 v4가 획득될 수 있다. 주변 블록 A가 4-파라미터 어파인 움직임 모델로 코딩된 경우, 현재 블록의 2개의 CPMV들이 v2 및 v3에 따라 산출될 수 있다. 주변 블록 A가 6-파라미터 어파인 움직임 모델로 코딩된 경우,

현재 블록의 3개의 CPMV들 v2, v3 및 v4에 따라 산출될 수 있다.

[135] 도 10은 어파인 머지 모드에서 후보들의 위치를 설명하기 위한 도면이다.

[136] 구성된(constructed) 어파인 후보는 각 컨트롤 포인트의 주변의 translational 움직임 정보를 조합하여 구성되는 후보를 의미할 수 있다. 컨트롤 포인트들의 움직임 정보는 특정된 공간적 주변 및 시간적 주변으로부터 도출될 수 있다. CPMV_k($k=0, 1, 2, 3$)는 k 번째 컨트롤 포인트를 나타낼 수 있다.

[137] 도 10을 참조하면, CPMV0을 위해 B2->B3->A2 순서에 따라 블록들이 체크될 수 있고, 첫 번째로 이용 가능한 블록의 움직임 벡터가 사용될 수 있다. CPMV1를 위해 B1->B0 순서에 따라 블록들이 체크될 수 있고, CPMV2을 위해 A1->A0 순서에 따라 블록들이 체크될 수 있다. TMVP(temporal motion vector predictor)는 이용 가능한 경우 CPMV3으로 사용될 수 있다.

[138] 4개의 컨트롤 포인트들의 움직임 벡터들이 획득된 후, 어파인 머지 후보들은 획득한 움직임 정보들을 기반으로 생성될 수 있다. 컨트롤 포인트 움직임 벡터들의 조합은 {CPMV0, CPMV1, CPMV2}, {CPMV0, CPMV1, CPMV3}, {CPMV0, CPMV2, CPMV3}, {CPMV1, CPMV2, CPMV3}, {CPMV0, CPMV1} 및 {CPMV0, CPMV2} 중 어느 하나에 해당할 수 있다.

[139] 3개의 CPMV들의 조합은 6-파라미터 어파인 머지 후보를 구성할 수 있고, 2개의 CPMV들의 조합은 4-파라미터 어파인 머지 후보를 구성할 수 있다. 움직임 스케일링 과정을 회피하기 위해, 컨트롤 포인트들의 참조 인덱스들이 서로 다른 경우, 컨트롤 포인트 움직임 벡터들의 관련된 조합들은 버려질 수 있다.

[140] 도 11은 인터 예측에서 SbTMVP를 설명하기 위한 도면이다.

[141] 한편, SbTMVP(subblock-based temporal motion vector prediction) 방법이 이용될 수도 있다. 예를 들어, SbTMVP는 ATMVP(advanced temporal motion vector prediction)라고 불릴 수도 있다. SbTMVP는 움직임 벡터 예측 및 현재 픽처 내의 CU들에 대한 머지 모드를 향상하기 위해 콜로케이티드 픽처(collocated picture) 내의 움직임 필드를 이용할 수 있다. 여기서, 콜로케이티드 픽처는 콜(col) 픽처라고 불릴 수도 있다.

[142] 예를 들어, SbTMVP는 서브블록(또는 서브 CU) 레벨에서 움직임을 예측할 수 있다. 또한, SbTMVP는 콜 픽처로부터 시간적 움직임을 패칭(fetching)하기 전에 움직임 쉬프트(shift)를 적용할 수 있다. 여기서, 움직임 쉬프트는 현재 블록의 공간적 주변 블록들 중 하나의 움직임 벡터로부터 획득될 수 있다.

[143] SbTMVP는 두 단계에 따라 현재 블록(또는 CU) 내의 서브블록(또는 서브 CU)의 움직임 벡터를 예측할 수 있다.

[144] 제1 단계에서 공간적 주변 블록들은 도 4의 A1, B1, B0 및 A0의 순서에 따라 시험될 수 있다. 콜(col) 픽처를 자신의 참조 픽처로서 사용하는 움직임 벡터를 가지는 첫 번째 공간적 주변 블록이 확인될 수 있고, 움직임 벡터는 적용될 움직임 쉬프트로 선택될 수 있다. 공간적 주변 블록으로부터 이러한 움직임이 확인되지 않는 경우, 움직임 쉬프트는 (0, 0)으로 설정될 수 있다.

- [145] 제2 단계에서는 제1 단계에서 확인된 움직임 쉬프트가 col 픽처로부터 서브블록 레벨 움직임 정보(움직임 벡터 및 참조 인덱스들)를 획득하기 위해 적용될 수 있다. 예를 들어, 움직임 쉬프트가 현재 블록의 좌표에 추가될 수 있다. 예를 들어, 움직임 쉬프트가 도 4의 A1의 움직임으로 설정될 수 있다. 이 경우 각 서브블록들에 대하여 col 픽처 내의 대응 블록의 움직임 정보는 서브블록의 움직임 정보를 도출하기 위해 사용될 수 있다. 시간적 움직임 스케일링은 시간적 움직임 벡터들의 참조 픽처들과 현재 블록의 참조 픽처들을 정렬하기 위해 적용될 수 있다.
- [146] SbTVMP 후보 및 어파인 머지 후보들을 모두 포함하는 조합된 서브블록 기반 머지 리스트는 어파인 머지 모드의 시그널링에 사용될 수 있다. 여기서, 어파인 머지 모드는 서브블록 기반 머지 모드라고 불릴 수 있다. SbTVMP 모드는 SPS(sequence parameter set)에 포함된 플래그에 의해 이용 가능 또는 이용 불가능할 수 있다. SbTMVP 모드가 이용 가능한 경우, SbTMVP 예측자는 서브블록 기반 머지 후보들의 리스트의 제1 엔트리로 추가될 수 있고, 어파인 머지 후보들이 다음으로 따라올 수 있다. 어파인 머지 후보 리스트의 최대 허용되는 사이즈는 5개일 수 있다.
- [147] SbTMVP에서 사용되는 서브 CU(또는 서브블록)의 사이즈는 8x8로 고정될 수 있고, 어파인 머지 모드에서와 마찬가지로 SbTMVP 모드는 너비 및 높이가 모두 8 이상인 블록에만 적용될 수 있다. 추가적인 SbTMVP 머지 후보의 인코딩 로직은 다른 머지 후보들과 동일할 수 있다. 즉, P 또는 B 슬라이스 내의 각 CU에 대하여 추가적인 RD(rate-distortion) 코스트를 이용하는 RD 체크가 SbTMVP 후보를 이용할지 결정하기 위해 수행될 수 있다.
- [148] 한편, 예측 모드에 따라 도출된 움직임 정보를 기반으로 현재 블록에 대한 예측된 블록을 도출할 수 있다. 상기 예측된 블록은 상기 현재 블록의 예측 샘플들(예측 샘플 어레이)를 포함할 수 있다. 현재 블록의 움직임 벡터가 분수 샘플 단위를 가리키는 경우, 보간(interpolation) 절차가 수행될 수 있으며, 이를 통하여 참조 픽처 내에서 분수 샘플 단위의 참조 샘플들을 기반으로 상기 현재 블록의 예측 샘플들이 도출될 수 있다. 현재 블록에 어파인 인터 예측(어파인 예측 모드)이 적용되는 경우, 샘플/서브블록 단위 MV를 기반으로 예측 샘플들을 생성할 수 있다. 쌍(Bi) 예측이 적용되는 경우, L0 예측(즉, 참조 픽처 리스트 L0 내 참조 픽처와 MVL0를 이용한 예측)을 기반으로 도출된 예측 샘플들과 L1 예측(즉, 참조 픽처 리스트 L1 내 참조 픽처와 MVL1을 이용한 예측)을 기반으로 도출된 예측 샘플들의 (위상에 따른) 가중합(weighted sum) 또는 가중평균(weighted average)을 통하여 도출된 예측 샘플들이 현재 블록의 예측 샘플들로 이용될 수 있다. 여기서, L0 방향의 움직임 벡터는 L0 움직임 벡터 또는 MVL0라고 불릴 수 있고, L1 방향의 움직임 벡터는 L1 움직임 벡터 또는 MVL1이라고 불릴 수 있다. 쌍(Bi) 예측이 적용되는 경우, L0 예측에 이용된 참조 픽처와 L1 예측에 이용된 참조 픽처가 현재 픽처를 기준으로 서로 다른 시간적

방향에 위치하는 경우(즉, 쌍 예측이면서 양방향 예측에 해당하는 경우), 이를 트루(true) 쌍 예측이라고 부를 수 있다.

[149] 또한, 도출된 예측 샘플들을 기반으로 복원 샘플들 및 복원 픽처가 생성될 수 있고, 이후 인루프 필터링 등의 절차가 수행될 수 있음은 전술한 바와 같다.

[150] 한편, 현재 블록에 쌍예측이 적용되는 경우, 가중평균을 기반으로 예측 샘플들을 도출할 수 있다. 예를 들어, 가중 평균을 이용한 쌍 예측은 BCW(Bi-prediction with CU-level Weight), BWA(Bi-prediction with Weighted Average) 또는 가중평균 쌍 예측(wighted averaging bi-prediction)이라고 불릴 수 있다.

[151] 기존에 쌍 예측 신호(즉, 쌍 예측 샘플들)은 L0 예측 신호(L0 예측 샘플들)과 L1 예측 신호(L1 예측 샘플들)의 단순 평균을 통하여 도출될 수 있었다. 즉, 쌍 예측 샘플들은 L0 참조 픽처 및 MVL0에 기반한 L0 예측 샘플들과 L1 참조 픽처 및 MVL1에 기반한 L1 예측 샘플들의 평균으로 도출되었다. 다만, 쌍 예측이 적용되는 경우, 다음과 같이 L0 예측 신호와 L1 예측 신호의 가중평균을 통하여 쌍 예측 신호(쌍 예측 샘플들)을 도출할 수도 있다. 예를 들어, 쌍 예측 신호(쌍 예측 샘플들)은 수학식 3과 같이 도출될 수 있다.

[152] [수식3]

$$P_{\text{bi-pred}} = ((8 - w) * P_0 + w * P_1 + 4) \gg 3$$

[153] 수학식 3에서, Pbi-pred는 쌍 예측 신호의 값 즉, 쌍 예측이 적용되어 도출된 예측 샘플 값을 나타낼 수 있고, w는 가중치를 나타낼 수 있다. 또한, P0는 L0 예측 신호의 값 즉, L0 예측이 적용되어 도출된 예측 샘플 값을 나타낼 수 있고, P1는 L1 예측 신호의 값 즉, L1 예측이 적용되어 도출된 예측 샘플 값을 나타낼 수 있다.

[154] 예를 들어, 가중평균 쌍 예측에서는 5개의 가중치가 허용될 수 있다. 예를 들어, 상기 5개의 가중치(w)는 -2, 3, 4, 5 또는 10를 포함할 수 있다. 즉, 가중치(w)는 -2, 3, 4, 5 또는 10를 포함하는 가중치 후보들 중 하나로 결정될 수 있다. 쌍 예측이 적용되는 각 CU에 대하여 가중치 w는 2가지 방법 중 하나에 의해 결정될 수 있다. 첫 번째 방법은 병합되지 않은 CU에 대하여 가중치 인덱스가 움직임 벡터 차분(motion vector difference) 이후에 시그널링될 수 있다. 두 번째 방법은 병합된 CU에 대하여 가중치 인덱스가 머지 후보 인덱스를 기반으로 주변 블록들로부터 추론될 수 있다.

[155] 예를 들어, 가중평균 쌍 예측은 256개 이상의 루마 샘플들을 가지는 CU에 적용될 수 있다. 즉, CU의 너비 및 높이의 곱이 256보다 크거나 같은 경우에 가중평균 쌍 예측이 적용될 수 있다. 저지연(low-delay) 픽처의 경우, 5개의 가중치가 사용될 수 있고, 저지연이 아닌(non-low-delay) 픽처의 경우 3개의

가중치가 사용될 수 있다. 예를 들어, 상기 3개의 가중치는 3, 4 또는 5를 포함할 수 있다.

- [156] 예를 들어, 인코딩 장치에서, 빠른 씨치 알고리즘은 인코딩 장치의 복잡성(complexity)을 크게 증가시키지 않으면서 가중치 인덱스를 찾기 위해 적용될 수 있다. 이러한 알고리즘은 다음과 같이 요약될 수 있다. 예를 들어, AMVR(adaptive motion vector resolution)과 조합될 때(인터 예측 모드로 AMVR이 이용될 때) 현재 픽처가 저지연 픽처인 경우, 동일하지 않은 가중치는 1-pel 및 4-pel 움직임 벡터 정밀도에 대해 조건부로 체크될 수 있다. 예를 들어, 어파인(affine)과 조합될 때(인터 예측 모드로 어파인 예측 모드가 이용될 때), 어파인 예측 모드가 현재 베스트 모드(best mode)로 선택된 경우, 어파인 ME(Motion Estimation)는 동일하지 않은 가중치들에 대해 수행될 수 있다. 예를 들어, 쌍 예측의 2개의 참조 픽처가 동일한 경우, 동일하지 않은 가중치들이 조건부로 체크될 수 있다. 예를 들어, 현재 픽처와 참조 픽처 간의 POC 거리, 코딩 QP(quantization parameter) 및 시간적 레벨(temporal level)에 의존하여 특정 조건이 만족되는 경우, 동일하지 않은 가중치들이 씨치되지 않을 수 있다.
- [157] 예를 들어, BCW 가중치 인덱스(또는 가중치 인덱스)는 하나의 컨텍스트 코딩된 빈(context coded bin) 및 뒤따르는 바이패스 코딩된 빈(bypass coded bin)을 이용하여 코딩될 수 있다. 첫 번째 컨텍스트 코딩된 빈은 동일한 가중치가 사용되는지 여부를 나타낼 수 있다. 첫 번째 컨텍스트 코딩된 빈을 기반으로 동일하지 않은 가중치가 사용되는 경우, 사용될 동일하지 않은 가중치를 지시하기 위해 바이 패스 코딩을 사용하여 추가적인 빈이 시그널링될 수 있다.
- [158] 한편, 쌍 예측이 적용되는 경우 예측 샘플들을 생성하는 데 이용되는 가중치 정보는 머지 후보 리스트에 포함된 후보들 중 선택된 후보에 대한 가중치 인덱스 정보를 기반으로 도출될 수 있다.
- [159] 본 문서의 일 실시예에 따르면, 머지 모드를 위한 움직임 벡터 후보 구성 시, 시간적 움직임 벡터 후보에 대한 가중치 인덱스 정보는 하기와 같이 도출할 수 있다. 예를 들어, 시간적 움직임 벡터 후보가 쌍 예측을 사용할 때 가중 평균을 위한 가중치 인덱스 정보를 도출할 수 있다. 즉, 인터 예측 타입이 쌍(bi) 예측인 경우 머지 후보 리스트 내의 시간적 머지 후보(또는 시간적 움직임 벡터 후보)를 위한 가중치 인덱스 정보를 도출할 수 있다.
- [160] 예를 들어, 시간적 움직임 벡터 후보에 대하여 가중 평균을 위한 가중치 인덱스 정보는 항상 0으로 도출될 수 있다. 여기서, 가중치 인덱스 정보가 0인 것은 각 참조 방향(즉, 쌍 예측에서 L0 예측 방향 및 L1 예측 방향)의 가중치가 동일함을 의미할 수 있다. 예를 들어, 머지 모드를 위한 루마 성분의 움직임 벡터를 도출하는 절차는 다음의 표 1과 같을 수 있다.

[161] [31]

8.4.2.2 Derivation process for luma motion vectors for merge mode

This process is only invoked when `merge_flag[xCb][yCb]` is equal to 1, where (xCb, yCb) specify the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture.

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable `cbWidth` specifying the width of the current coding block in luma samples,
- a variable `cbHeight` specifying the height of the current coding block in luma samples.

Outputs of this process are:

- the luma motion vectors in 1/16 fractional-sample accuracy `mvL0[0][0]` and `mvL1[0][0]`,
- the reference indices `refIdxL0` and `refIdxL1`,
- the prediction list utilization flags `predFlagL0[0][0]` and `predFlagL1[0][0]`,
- the bi-prediction weight index `gbIdx`.

The bi-prediction weight index `gbIdx` is set equal to 0.

The motion vectors `mvL0[0][0]` and `mvL1[0][0]`, the reference indices `refIdxL0` and `refIdxL1` and the prediction utilization flags `predFlagL0[0][0]` and `predFlagL1[0][0]` are derived by the following ordered steps:

1. The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width `cbWidth`, and the luma coding block height `cbHeight` as inputs, and the output being the availability flags `availableFlagA0`, `availableFlagA1`, `availableFlagB0`, `availableFlagB1` and `availableFlagB2`, the reference indices `refIdxLXA0`, `refIdxLXA1`, `refIdxLXB0`, `refIdxLXB1` and `refIdxLXB2`, the prediction list utilization flags `predFlagLXA0`, `predFlagLXA1`, `predFlagLXB0`, `predFlagLXB1` and `predFlagLXB2`, and the motion vectors `mvLXA0`, `mvLXA1`, `mvLXB0`, `mvLXB1` and `mvLXB2`, with X being 0 or 1, and the bi-prediction weight indices `gbIdxA0`, `gbIdxA1`, `gbIdxB0`, `gbIdxB1`, `gbIdxB2`.
2. The reference indices, `refIdxLXCol`, with X being 0 or 1, and the bi-prediction weight index `gbIdxCol` for the temporal merging candidate `Col` are set equal to 0.

[162]

3. The derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL0Col` as inputs, and the output being the availability flag `availableFlagL0Col` and the temporal motion vector `mvL0Col`. The variables `availableFlagCol`, `predFlagL0Col` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \quad (8-283)$$

$$\text{predFlagL0Col} = \text{availableFlagL0Col} \quad (8-284)$$

$$\text{predFlagL1Col} = 0 \quad (8-285)$$

$$\text{gbIdxCol} = 0 \quad (8-xxx)$$

4. When `tile_group_type` is equal to B, the derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL1Col` as inputs, and the output being the availability flag `availableFlagL1Col` and the temporal motion vector `mvL1Col`. The variables `availableFlagCol` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (8-286)$$

[163]

predFlagL1Col = availableFlagL1Col (8-287)

5. The merging candidate list, mergeCandList, is constructed as follows:

```

i = 0
if( availableFlagA1 )
    mergeCandList[ i++ ] = A1
if( availableFlagB1 )
    mergeCandList[ i++ ] = B1
if( availableFlagB0 )
    mergeCandList[ i++ ] = B0
if( availableFlagA0 )
    mergeCandList[ i++ ] = A0
if( availableFlagB2 )
    mergeCandList[ i++ ] = B2
if( availableFlagCol )
    mergeCandList[ i++ ] = Col
  
```

(8-288)

6. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList.
7. When numCurrMergeCand is less than (MaxNumMergeCand - 1) and NumHmvpCand is greater than 0, the following applies:
- The derivation process of history-based merging candidates as specified in 8.4.2.6 is invoked with mergeCandList, and numCurrMergeCand as inputs, and modified mergeCandList and numCurrMergeCand as outputs.
 - numOrigMergeCand is set equal to numCurrMergeCand.
8. The derivation process for pairwise average merging candidates specified in clause 8.4.2.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList, numCurrMergeCand and numOrigMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0avgCand_k and refIdxL1avgCand_k, the prediction list utilization flags predFlagL0avgCand_k and predFlagL1avgCand_k and the motion vectors mvL0avgCand_k and mvL1avgCand_k of every new candidate avgCand_k being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate avgCand_k being added into mergeCandList is set equal to 0. The number of candidates being added, numAvgMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand). When numAvgMergeCand is greater than 0, k ranges from 0 to numAvgMergeCand - 1, inclusive.

[164]

9. The derivation process for zero motion vector merging candidates specified in clause 8.4.2.5 is invoked with the mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList and numCurrMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0zeroCand_m and refIdxL1zeroCand_m, the prediction list utilization flags predFlagL0zeroCand_m and predFlagL1zeroCand_m and the motion vectors mvL0zeroCand_m and mvL1zeroCand_m of every new candidate zeroCand_m being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate zeroCand_m being added into mergeCandList is set equal to 0. The number of candidates being added, numZeroMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand - numAvgMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand - 1, inclusive.
10. The variable mergeIdxOffset is set equal to 0.
11. When mmvd_flag[xCb][yCb] is equal to 1, the variable mmvdCnt is set equal to 0 and the following applies until mmvdCnt is greater than (merge_idx[xCb][yCb] + mergeIdxOffset) or mmvdCnt is equal to MaxNumMergeCand:
- When candidate mergeCandList[mmvdCnt] uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1.
 - The variable mmvdCnt is incremented by 1.

[165]

12. The following assignments are made with N being the candidate at position $\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}$ in the merging candidate list mergeCandList ($N = \text{mergeCandList}[\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}]$) and X being replaced by 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-289)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-290)$$

$$\text{mvLX}[0][0][0] = \text{mvLXN}[0] \quad (8-291)$$

$$\text{mvLX}[0][0][1] = \text{mvLXN}[1] \quad (8-292)$$

$$\text{gbiIdx} = \text{gbiIdxN} \quad (8-293)$$

13. When $\text{mmvd_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, the following applies:

- The derivation process for merge motion vector difference as specified in 8.4.2.7 is invoked with the luma location (xCb, yCb), the luma motion vectors $\text{mvL0}[0][0]$, $\text{mvL1}[0][0]$, the reference indices refIdxL0 , refIdxL1 and the prediction list utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$ as inputs, and the motion vector differences mMvdL0 and mMvdL1 as outputs.
- The motion vector difference mMvdLX is added to the merge motion vectors mvLX for X being 0 and 1 as follows:

$$\text{mvLX}[0][0][0] += \text{mMvdLX}[0] \quad (8-294)$$

$$\text{mvLX}[0][0][1] += \text{mMvdLX}[1] \quad (8-295)$$

[166] 상기 표 1을 참조하면, gbiIdx 는 쌍 예측 가중치 인덱스를 나타낼 수 있으며, gbiIdxCol 은 시간적 머지 후보(예를 들어, 머지 후보 리스트 내의 시간적 움직임 벡터 후보)에 대한 쌍 예측 가중치 인덱스를 나타낼 수 있다. 상기 머지 모드를 위한 루마 성분의 움직임 벡터를 도출하는 절차(8.4.2.2의 목차 3)에서 상기 gbiIdxCol 은 0으로 도출될 수 있다. 즉, 시간적 움직임 벡터 후보의 가중치 인덱스는 0으로 도출될 수 있다.

[167] 또는, 시간적 움직임 벡터 후보에 대한 가중평균을 위한 가중치 인덱스는 콜로케이티드(collocated) 블록의 가중치 인덱스 정보에 기반하여 도출될 수 있다. 여기서, 콜로케이티드 블록은 콜(col) 블록, 동일 위치 블록 또는 동일 위치 참조 블록이라고 불릴 수 있으며, 콜 블록은 참조 픽처 상에서 현재 블록과 동일한 위치의 블록을 나타낼 수 있다. 예를 들어, 머지 모드를 위한 루마 성분의 움직임 벡터를 도출하는 절차는 다음의 표 2와 같을 수 있다.

[168] [32]

8.4.2.2 Derivation process for luma motion vectors for merge mode

This process is only invoked when $\text{merge_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, where (xCb, yCb) specify the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture.

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable cbWidth specifying the width of the current coding block in luma samples,
- a variable cbHeight specifying the height of the current coding block in luma samples.

Outputs of this process are:

- the luma motion vectors in 1/16 fractional-sample accuracy $\text{mvL0}[0][0]$ and $\text{mvL1}[0][0]$,
- the reference indices refIdxL0 and refIdxL1 ,
- the prediction list utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$,
- the bi-prediction weight index gbIdx .

The bi-prediction weight index gbIdx is set equal to 0.

The motion vectors $\text{mvL0}[0][0]$ and $\text{mvL1}[0][0]$, the reference indices refIdxL0 and refIdxL1 and the prediction utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$ are derived by the following ordered steps:

1. The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width cbWidth , and the luma coding block height cbHeight as inputs, and the output being the availability flags availableFlagA_0 , availableFlagA_1 , availableFlagB_0 , availableFlagB_1 and availableFlagB_2 , the reference indices refIdxLXA_0 , refIdxLXA_1 , refIdxLXB_0 , refIdxLXB_1 and refIdxLXB_2 , the prediction list utilization flags predFlagLXA_0 , predFlagLXA_1 , predFlagLXB_0 , predFlagLXB_1 and predFlagLXB_2 , and the motion vectors mvLXA_0 , mvLXA_1 , mvLXB_0 , mvLXB_1 and mvLXB_2 , with X being 0 or 1, and the bi-prediction weight indices gbIdxA_0 , gbIdxA_1 , gbIdxB_0 , gbIdxB_1 , gbIdxB_2 .
2. The reference indices, refIdxLXCol , with X being 0 or 1, and the bi-prediction weight index gbIdxCol for the temporal merging candidate Col are set equal to 0.

[169]

3. The derivation process for temporal luma motion vector prediction as specified in in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width cbWidth , the luma coding block height cbHeight and the variable refIdxL0Col as inputs, and the output being the availability flag $\text{availableFlagL0Col}$ and the temporal motion vector mvL0Col . The variables availableFlagCol , predFlagL0Col and predFlagL1Col are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \quad (8-283)$$

$$\text{predFlagL0Col} = \text{availableFlagL0Col} \quad (8-284)$$

$$\text{predFlagL1Col} = 0 \quad (8-285)$$

$$\text{gbIdxCol} = 0 \quad (8-xxx)$$

4. When tile_group_type is equal to B, the derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (xCb, yCb) , the luma coding block width cbWidth , the luma coding block height cbHeight and the variable refIdxL1Col as inputs, and the output being the availability flag $\text{availableFlagL1Col}$ and the temporal motion vector mvL1Col . The variables availableFlagCol and predFlagL1Col are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (8-286)$$

[170]

predFlagL1Col = availableFlagL1Col (8-287)

gbiIdxCol = gbiIdxCol (x-xxx)

5. The merging candidate list, mergeCandList, is constructed as follows:

```

i = 0
if( availableFlagA1 )
    mergeCandList[ i++ ] = A1
if( availableFlagB1 )
    mergeCandList[ i++ ] = B1
if( availableFlagB0 )
    mergeCandList[ i++ ] = B0
if( availableFlagA0 )
    mergeCandList[ i++ ] = A0
if( availableFlagB2 )
    mergeCandList[ i++ ] = B2
if( availableFlagCol )
    mergeCandList[ i++ ] = Col

```

(8-288)

6. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList.
7. When numCurrMergeCand is less than (MaxNumMergeCand - 1) and NumHmvpCand is greater than 0, the following applies:
- The derivation process of history-based merging candidates as specified in 8.4.2.6 is invoked with mergeCandList, and numCurrMergeCand as inputs, and modified mergeCandList and numCurrMergeCand as outputs.
 - numOrigMergeCand is set equal to numCurrMergeCand.

[171]

8. The derivation process for pairwise average merging candidates specified in clause 8.4.2.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList, numCurrMergeCand and numOrigMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0avgCand_k and refIdxL1avgCand_k, the prediction list utilization flags predFlagL0avgCand_k and predFlagL1avgCand_k and the motion vectors mvL0avgCand_k and mvL1avgCand_k of every new candidate avgCand_k being added into mergeCandList. The bi-prediction weight index gbiIdx of every new candidate avgCand_k being added into mergeCandList is set equal to 0. The number of candidates being added, numAvgMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand). When numAvgMergeCand is greater than 0, k ranges from 0 to numAvgMergeCand - 1, inclusive.
9. The derivation process for zero motion vector merging candidates specified in clause 8.4.2.5 is invoked with the mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList and numCurrMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0zeroCand_m and refIdxL1zeroCand_m, the prediction list utilization flags predFlagL0zeroCand_m and predFlagL1zeroCand_m and the motion vectors mvL0zeroCand_m and mvL1zeroCand_m of every new candidate zeroCand_m being added into mergeCandList. The bi-prediction weight index gbiIdx of every new candidate zeroCand_m being added into mergeCandList is set equal to 0. The number of candidates being added, numZeroMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand - numAvgMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand - 1, inclusive.
10. The variable mergeIdxOffset is set equal to 0.
11. When mmvd_flag[xCb][yCb] is equal to 1, the variable mmvdCnt is set equal to 0 and the following applies until mmvdCnt is greater than (merge_idx[xCb][yCb] + mergeIdxOffset) or mmvdCnt is equal to MaxNumMergeCand:

[172]

- When candidate mergeCandList[mmvdCnt] uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1.
 - The variable mmvdCnt is incremented by 1.
12. The following assignments are made with N being the candidate at position merge_idx[xCb][yCb] + mergeIdxOffset in the merging candidate list mergeCandList (N = mergeCandList[merge_idx[xCb][yCb] + mergeIdxOffset]) and X being replaced by 0 or 1:
- $$\text{refIdxLX} = \text{refIdxLXN} \quad (8-289)$$
- $$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-290)$$
- $$\text{mvLX}[0][0][0] = \text{mvLXN}[0] \quad (8-291)$$
- $$\text{mvLX}[0][0][1] = \text{mvLXN}[1] \quad (8-292)$$
- $$\text{gbIdx} = \text{gbIdxN} \quad (8-293)$$
13. When mmvd_flag[xCb][yCb] is equal to 1, the following applies:
- The derivation process for merge motion vector difference as specified in 8.4.2.7 is invoked with the luma location (xCb, yCb), the luma motion vectors mvL0[0][0], mvL1[0][0], the reference indices refIdxL0, refIdxL1 and the prediction list utilization flags predFlagL0[0][0] and predFlagL1[0][0] as inputs, and the motion vector differences mMvdL0 and mMvdL1 as outputs.
 - The motion vector difference mMvdLX is added to the merge motion vectors mvLX for X being 0 and 1 as follows:
- $$\text{mvLX}[0][0][0] += \text{mMvdLX}[0] \quad (8-294)$$
- $$\text{mvLX}[0][0][1] += \text{mMvdLX}[1] \quad (8-295)$$

[173]

상기 표 2를 참조하면, gbIdx는 쌍 예측 가중치 인덱스를 나타낼 수 있으며, gbIdxCol은 시간적 머지 후보(예를 들어, 머지 후보 리스트 내의 시간적 움직임 벡터 후보)에 대한 쌍 예측 가중치 인덱스를 나타낼 수 있다. 상기 머지 모드를 위한 루마 성분의 움직임 벡터를 도출하는 절차에서 슬라이스의 타입 또는 타일 그룹의 타입이 B인 경우(8.4.2.2의 목차 4)에는 상기 gbIdxCol이 gbIdxCol로 도출될 수 있다. 즉, 시간적 움직임 벡터 후보의 가중치 인덱스는 콜(col) 블록의 가중치 인덱스로 도출될 수 있다.

[174]

한편, 본 문서의 다른 실시예에 따르면, 서브블록 단위의 머지 모드를 위한 움직임 벡터 후보 구성 시, 시간적 움직임 벡터 후보에 대한 가중평균을 위한 가중치 인덱스를 도출할 수 있다. 여기서, 서브블록 단위의 머지 모드는 (서브블록 단위의) 어파인 머지 모드라고 불릴 수 있다. 시간적 움직임 벡터 후보는 서브블록 기반 시간적 움직임 벡터 후보를 나타낼 수 있으며, SbTMVP(또는 ATMVP) 후보라고 불릴 수도 있다. 즉, 인터 예측 타입이 쌍(bi) 예측인 경우 어파인 머지 후보 리스트 또는 서브블록 머지 후보 리스트 내의 SbTMVP 후보(또는 서브블록 기반 시간적 움직임 벡터 후보)에 대한 가중치 인덱스 정보를 도출할 수 있다.

[175]

예를 들어, 서브블록 기반 시간적 움직임 벡터 후보에 대한 가중평균을 위한 가중치 인덱스 정보는 항상 0으로 도출될 수 있다. 여기서, 가중치 인덱스 정보가 0인 것은 각 참조 방향(즉, 쌍 예측에서 L0 예측 방향 및 L1 예측 방향)의

가중치가 동일함을 의미할 수 있다. 예를 들어, 서브블록 머지 모드 내의 움직임 벡터 및 참조 인덱스를 도출하는 절차 및 서브블록 기반 시간적 머지 후보를 도출하는 절차는 각각 다음의 표 3 및 표 4와 같을 수 있다.

[176] [표3]

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables $cbWidth$ and $cbHeight$ specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flag arrays $predFlagL0[xSbIdx][ySbIdx]$ and $predFlagL1[xSbIdx][ySbIdx]$,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy $mvL0[xSbIdx][ySbIdx]$ and $mvL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy $mvCL0[xSbIdx][ySbIdx]$ and $mvCL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdx$.

The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:

1. When $sps_sbtmvp_enabled_flag$ is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.

[177]

- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (xCb, yCb), the luma coding block width cbWidth, the luma coding block height cbHeight, the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀, availableFlagB₁, the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀, refIdxLXB₁, the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀, predFlagLXB₁ and the motion vectors mvLXA₀, mvLXA₁, mvLXB₀, mvLXB₁ as inputs and the output being the availability flag availableFlagSbCol, the bi-prediction weight index gbiIdxSbCol, the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY, the reference indices refIdxLXSbCol, the luma motion vectors mvLXSbCol[xSbIdx][ySbIdx] and the prediction list utilization flags predFlagLXSbCol[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX – 1, ySbIdx = 0..numSbY – 1 and X being 0 or 1.
- 2. When sps_affine_enabled_flag is equal to 1, the sample locations (xNbA₀, yNbA₀), (xNbA₁, yNbA₁), (xNbA₂, yNbA₂), (xNbB₀, yNbB₀), (xNbB₁, yNbB₁), (xNbB₂, yNbB₂), (xNbB₃, yNbB₃), and the variables numSbX and numSbY are derived as follows:

$$(xA_0, yA_0) = (xCb - 1, yCb + cbHeight) \quad (8-536)$$

$$(xA_1, yA_1) = (xCb - 1, yCb + cbHeight - 1) \quad (8-537)$$

$$(xA_2, yA_2) = (xCb - 1, yCb) \quad (8-538)$$

$$(xB_0, yB_0) = (xCb + cbWidth, yCb - 1) \quad (8-539)$$

$$(xB_1, yB_1) = (xCb + cbWidth - 1, yCb - 1) \quad (8-540)$$

$$(xB_2, yB_2) = (xCb - 1, yCb - 1) \quad (8-541)$$

$$(xB_3, yB_3) = (xCb, yCb - 1) \quad (8-542)$$

[178]

- $$numSbX = cbWidth >> 2 \quad (8-543)$$
- $$numSbY = cbHeight >> 2 \quad (8-544)$$
- 3. When sps_affine_enabled_flag is equal to 1, the variable availableFlagA is set equal to FALSE and the following applies for (xNbA_k, yNbA_k) from (xNbA₀, yNbA₀) to (xNbA₁, yNbA₁):
 - The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbA_k, yNbA_k) as inputs, and the output is assigned to the block availability flag availableA_k.
 - When availableA_k is equal to TRUE and MotionModelIdx[xNbA_k][yNbA_k] is greater than 0 and availableFlagA is equal to FALSE, the following applies:
 - The variable availableFlagA is set equal to TRUE, motionModelIdxA is set equal to MotionModelIdx[xNbA_k][yNbA_k], (xNb, yNb) is set equal to (CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k]), nbW is set equal to CbWidth[xNbA_k][yNbA_k], nbH is set equal to CbHeight[xNbA_k][yNbA_k], numCpMv is set equal to MotionModelIdx[xNbA_k][yNbA_k] + 1, and gbiIdxA is set equal to GbiIdx[xNbA_k][yNbA_k].
 - For X being replaced by either 0 or 1, the following applies:
 - When PredFlagLX[xNbA_k][yNbA_k] is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height (cbWidth, cbHeight), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates cpMvLXA[cpIdx] with cpIdx = 0..numCpMv – 1 as output.
 - The following assignments are made:

$$predFlagLXA = PredFlagLX[xNbA_k][yNbA_k] \quad (8-545)$$

$$refIdxLXA = RefIdxLX[xNbA_k][yNbA_k] \quad (8-546)$$

[179]

4. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagB` is set equal to `FALSE` and the following applies for $(xNbB_k, yNbB_k)$ from $(xNbB_0, yNbB_0)$ to $(xNbB_2, yNbB_2)$:
- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (xCb, yCb) set equal to $(xNbB_k, yNbB_k)$ and the neighbouring luma location $(xNbB_k, yNbB_k)$ as inputs, and the output is assigned to the block availability flag `availableBk`.
 - When `availableBk` is equal to `TRUE` and `MotionModelIdc[ xNbBk ][ yNbBk ]` is greater than 0 and `availableFlagB` is equal to `FALSE`, the following applies:
 - The variable `availableFlagB` is set equal to `TRUE`, `motionModelIdcB` is set equal to `MotionModelIdc[ xNbBk ][ yNbBk ]`, (xNb, yNb) is set equal to $(CbPosX[xNbAB][yNbB_k], CbPosY[xNbB_k][yNbB_k])$, `nbW` is set equal to `CbWidth[ xNbBk ][ yNbBk ]`, `nbH` is set equal to `CbHeight[ xNbBk ][ yNbBk ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbBk ][ yNbBk ] + 1`, and `gbIdxB` is set equal to `GbIdx[ xNbBk ][ yNbBk ]`.
 - For X being replaced by either 0 or 1, the following applies:
 - When `PredFlagLX[ xNbBk ][ yNbBk ]` is equal to `TRUE`, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the neighbouring luma coding block location (xNb, yNb) , the neighbouring luma coding block width and height (nbW, nbH) , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXB[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.
 - The following assignments are made:

[180]

$$\text{predFlagLXB} = \text{PredFlagLX}[xNbB_k][yNbB_k] \quad (8-547)$$

$$\text{refIdxLXB} = \text{RefIdxLX}[xNbB_k][yNbB_k] \quad (8-548)$$

5. When `sps_affine_enabled_flag` is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the availability flags `availableA0`, `availableA1`, `availableA2`, `availableB0`, `availableB1`, `availableB2`, `availableB3` as inputs, and the availability flags `availableFlagConstK`, the reference indices `refIdxLXConstK`, prediction list utilization flags `predFlagLXConstK`, motion model indices `motionModelIdcConstK` and `cpMvLXConstK[ cpIdx ]` with X being 0 or 1, $K = 1..6$, `cpIdx = 0..2` as outputs and `gbIdxConstK` is set equal to 0 with $K = 1..6$.

6. The initial subblock merging candidate list, `subblockMergeCandList`, is constructed as follows:

```

i = 0
if( availableFlagSbCol )
    subblockMergeCandList[ i++ ] = SbCol
if( availableFlagA && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = A
if( availableFlagB && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = B
if( availableFlagConst1 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const1
if( availableFlagConst2 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const2
if( availableFlagConst3 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const3
if( availableFlagConst4 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const4
if( availableFlagConst5 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const5
if( availableFlagConst6 && i < MaxNumSubblockMergeCand )
    subblockMergeCandList[ i++ ] = Const6

```

(8-549)

[181]

7. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the subblockMergeCandList.
8. When numCurrMergeCand is less than MaxNumSubblockMergeCand, the following is repeated until numCurrMergeCand is equal to MaxNumSubblockMergeCand, with mvZero[0] and mvZero[1] both being equal to 0:
 - The reference indices, the prediction list utilization flags and the motion vectors of zeroCand_m with m equal to (numCurrMergeCand – numOrigMergeCand) are derived as follows:

$$\text{refIdxL0ZeroCand}_m = 0 \quad (8-550)$$

$$\text{predFlagL0ZeroCand}_m = 1 \quad (8-551)$$

$$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero} \quad (8-552)$$

$$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero} \quad (8-553)$$

$$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero} \quad (8-554)$$

$$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1 \quad (8-555)$$

$$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0 \quad (8-556)$$

$$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero} \quad (8-557)$$

$$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero} \quad (8-558)$$

$$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero} \quad (8-559)$$

$$\text{motionModelIdxZeroCand}_m = 1 \quad (8-560)$$

$$\text{gbIdxZeroCand}_m = 0 \quad (8-561)$$

[182]

- The candidate zeroCand_m with m equal to (numCurrMergeCand – numOrigMergeCand) is added at the end of subblockMergeCandList and numCurrMergeCand is incremented by 1 as follows:

$$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$$

The variables refIdxL0, refIdxL1, predFlagL0[xSbIdx][ySbIdx], predFlagL1[xSbIdx][ySbIdx], mvL0[xSbIdx][ySbIdx], mvL1[xSbIdx][ySbIdx], mvCL0[xSbIdx][ySbIdx], and mvCL1[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX – 1, ySbIdx = 0..numSbY – 1 are derived as follows:
- If subblockMergeCandList[merge_subblock_idx[xCb][yCb]] is equal to SbCol, the bi-prediction weight index gbIdx is set equal to 0 and the following applies with X being 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$$
- For xSbIdx = 0..numSbX – 1, ySbIdx = 0..numSbY – 1, the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] \quad (8-564)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][0] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][0] \quad (8-565)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][1] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][1] \quad (8-566)$$
- When predFlagLX[xSbIdx][ySbIdx], is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with mvLX[xSbIdx][ySbIdx] and refIdxLX as inputs, and the output being mvCLX[xSbIdx][ySbIdx].
- The following assignment is made for $x = \text{xCb} \dots \text{xCb} + \text{cbWidth} - 1$ and $y = \text{yCb} \dots \text{yCb} + \text{cbHeight} - 1$:

$$\text{MotionModelIdx}[x][y] = 0 \quad (8-567)$$
- Otherwise (subblockMergeCandList[merge_subblock_idx[xCb][yCb]] is not equal to SbCol), the following applies with X being 0 or 1:

[183]

- The following assignments are made with N being the candidate at position $\text{merge_subblock_idx}[x_{Cb}][y_{Cb}]$ in the subblock merging candidate list $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[x_{Cb}][y_{Cb}]]$):

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$$

$$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$$

$$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$$

$$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$$

$$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-574)$$

- For $x_{SbIdx} = 0 \dots \text{numSbX} - 1$, $y_{SbIdx} = 0 \dots \text{numSbY} - 1$, the following applies:

$$\text{predFlagLX}[x_{SbIdx}][y_{SbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$$

- When $\text{predFlagLX}[0][0]$ is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (x_{Cb}, y_{Cb}) , the luma coding block width $cbWidth$, the luma prediction block height $cbHeight$, the number of control point motion vectors numCpMv , the control point motion vectors $\text{cpMvLX}[cpIdx]$ with $cpIdx$ being 0..2, and the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY as inputs, the luma subblock motion vector array $\text{mvLX}[x_{SbIdx}][y_{SbIdx}]$ and the chroma subblock motion vector array $\text{mvCLX}[x_{SbIdx}][y_{SbIdx}]$ with $x_{SbIdx} = 0 \dots \text{numSbX} - 1$, $y_{SbIdx} = 0 \dots \text{numSbY} - 1$ as outputs.

- The following assignment is made for $x = x_{Cb} \dots x_{Cb} + cbWidth - 1$ and $y = y_{Cb} \dots y_{Cb} + cbHeight - 1$:

$$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$$

[184]

[54]

8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location (x_{Cb}, y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable $cbWidth$ specifying the width of the current coding block in luma samples,
- a variable $cbHeight$ specifying the height of the current coding block in luma samples,
- the availability flags availableFlagA_0 , availableFlagA_1 , availableFlagB_0 , and availableFlagB_1 of the neighbouring coding units,
- the reference indices refIdxLXA_0 , refIdxLXA_1 , refIdxLXB_0 , and refIdxLXB_1 of the neighbouring coding units,
- the prediction list utilization flags predFlagLXA_0 , predFlagLXA_1 , predFlagLXB_0 , and predFlagLXB_1 of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy mvLXA_0 , mvLXA_1 , mvLXB_0 , and mvLXB_1 of the neighbouring coding units.

Outputs of this process are:

- the availability flag $\text{availableFlagSbCol}$,
- the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY ,
- the reference indices refIdxLOSbCol and refIdxLISbCol ,
- the luma motion vectors in 1/16 fractional-sample accuracy $\text{mvLOSbCol}[x_{SbIdx}][y_{SbIdx}]$ and $\text{mvLISbCol}[x_{SbIdx}][y_{SbIdx}]$ with $x_{SbIdx} = 0 \dots \text{numSbX} - 1$, $y_{SbIdx} = 0 \dots \text{numSbY} - 1$,
- the prediction list utilization flags $\text{predFlagLOSbCol}[x_{SbIdx}][y_{SbIdx}]$ and $\text{predFlagLISbCol}[x_{SbIdx}][y_{SbIdx}]$ with $x_{SbIdx} = 0 \dots \text{numSbX} - 1$, $y_{SbIdx} = 0 \dots \text{numSbY} - 1$,
- the bi-prediction weight index gbIdxSbCol .

The gbIdxSbCol is set equal to 0.

...

- [185] 상기 표 3 및 표 4를 참조하면, $gbIdx$ 는 쌍 예측 가중치 인덱스를 나타낼 수 있으며, $gbIdxSbCol$ 은 서브블록 기반 시간적 머지 후보(예를 들어, 서브블록 기반 머지 후보 리스트 내의 시간적 움직임 벡터 후보)에 대한 쌍 예측 가중치 인덱스를 나타낼 수 있고, 상기 서브블록 기반 시간적 머지 후보를 도출하는 절차(8.4.4.3)에서 상기 $gbIdxSbCol$ 은 0으로 도출될 수 있다. 즉, 서브블록 기반 시간적 움직임 벡터 후보의 가중치 인덱스는 0으로 도출될 수 있다.
- [186] 또는, 서브블록 기반 시간적 움직임 벡터 후보에 대한 가중평균을 위한 가중치 인덱스 정보는 시간적 센터(center) 블록의 가중치 인덱스 정보에 기반하여 도출될 수 있다. 예를 들어, 상기 시간적 센터 블록은 col 블록 또는 col 블록의 센터에 위치한 서브블록 또는 샘플을 나타낼 수 있으며, 구체적으로는 col 블록의 중앙 4개의 서브블록 또는 샘플 중 우하측에 위치한 서브블록 또는 샘플을 나타낼 수 있다. 예를 들어, 이 경우 서브블록 머지 모드 내의 움직임 벡터 및 참조 인덱스를 도출하는 절차, 서브블록 기반 시간적 머지 후보를 도출하는 절차 및 서브블록 기반 시간적 머지를 위한 베이스 움직임 정보를 도출하는 절차는 각각 다음의 표 5, 표 6 및 표 7과 같을 수 있다.
- [187] [표5]

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables $cbWidth$ and $cbHeight$ specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flag arrays $predFlagL0[xSbIdx][ySbIdx]$ and $predFlagL1[xSbIdx][ySbIdx]$,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy $mvL0[xSbIdx][ySbIdx]$ and $mvL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy $mvCL0[xSbIdx][ySbIdx]$ and $mvCL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdx$.

The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:

1. When $sps_sbtmvp_enabled_flag$ is equal to 1, the following applies:
 - The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.

[188]

- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (xCb, yCb), the luma coding block width cbWidth, the luma coding block height cbHeight, the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀, availableFlagB₁, the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀, refIdxLXB₁, the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀, predFlagLXB₁ and the motion vectors mvLXA₀, mvLXA₁, mvLXB₀, mvLXB₁ as inputs and the output being the availability flag availableFlagSbCol, the bi-prediction weight index gbIdxSbCol, the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY, the reference indices refIdxLXSbCol, the luma motion vectors mvLXSbCol[xSbIdx][ySbIdx] and the prediction list utilization flags predFlagLXSbCol[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1 and X being 0 or 1.
- 2. When sps_affine_enabled_flag is equal to 1, the sample locations (xNbA₀, yNbA₀), (xNbA₁, yNbA₁), (xNbA₂, yNbA₂), (xNbB₀, yNbB₀), (xNbB₁, yNbB₁), (xNbB₂, yNbB₂), (xNbB₃, yNbB₃), and the variables numSbX and numSbY are derived as follows:

$$(xA_0, yA_0) = (xCb - 1, yCb + cbHeight) \quad (8-536)$$

$$(xA_1, yA_1) = (xCb - 1, yCb + cbHeight - 1) \quad (8-537)$$

$$(xA_2, yA_2) = (xCb - 1, yCb) \quad (8-538)$$

$$(xB_0, yB_0) = (xCb + cbWidth, yCb - 1) \quad (8-539)$$

$$(xB_1, yB_1) = (xCb + cbWidth - 1, yCb - 1) \quad (8-540)$$

$$(xB_2, yB_2) = (xCb - 1, yCb - 1) \quad (8-541)$$

$$(xB_3, yB_3) = (xCb, yCb - 1) \quad (8-542)$$

[189]

- $$numSbX = cbWidth >> 2 \quad (8-543)$$
- $$numSbY = cbHeight >> 2 \quad (8-544)$$
- 3. When sps_affine_enabled_flag is equal to 1, the variable availableFlagA is set equal to FALSE and the following applies for (xNbA_k, yNbA_k) from (xNbA₀, yNbA₀) to (xNbA₁, yNbA₁):
 - The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbl] is invoked with the current luma location (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbA_k, yNbA_k) as inputs, and the output is assigned to the block availability flag availableA_k.
 - When availableA_k is equal to TRUE and MotionModelIdx[xNbA_k][yNbA_k] is greater than 0 and availableFlagA is equal to FALSE, the following applies:
 - The variable availableFlagA is set equal to TRUE, motionModelIdxA is set equal to MotionModelIdx[xNbA_k][yNbA_k], (xNb, yNb) is set equal to (CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k]), nbW is set equal to CbWidth[xNbA_k][yNbA_k], nbH is set equal to CbHeight[xNbA_k][yNbA_k], numCpMv is set equal to MotionModelIdx[xNbA_k][yNbA_k] + 1, and gbIdxA is set equal to GbIdx[xNbA_k][yNbA_k].
 - For X being replaced by either 0 or 1, the following applies:
 - When PredFlagLX[xNbA_k][yNbA_k] is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height (cbWidth, cbHeight), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates cpMvLXA[cpIdx] with cpIdx = 0..numCpMv - 1 as output.
 - The following assignments are made:

$$predFlagLXA = PredFlagLX[xNbA_k][yNbA_k] \quad (8-545)$$

$$refIdxLXA = RefIdxLX[xNbA_k][yNbA_k] \quad (8-546)$$

[190]

4. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagB` is set equal to FALSE and the following applies for $(xNbB_k, yNbB_k)$ from $(xNbB_0, yNbB_0)$ to $(xNbB_2, yNbB_2)$:
- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(xCurr, yCurr)$ set equal to (xCb, yCb) and the neighbouring luma location $(xNbB_k, yNbB_k)$ as inputs, and the output is assigned to the block availability flag `availableBk`.
 - When `availableBk` is equal to TRUE and `MotionModelIdc[xNbBk][yNbBk]` is greater than 0 and `availableFlagB` is equal to FALSE, the following applies:
 - The variable `availableFlagB` is set equal to TRUE, `motionModelIdcB` is set equal to `MotionModelIdc[xNbBk][yNbBk]`, (xNb, yNb) is set equal to $(CbPosX[xNbAB][yNbB_k], CbPosY[xNbB_k][yNbB_k])$, `nbW` is set equal to `CbWidth[xNbBk][yNbBk]`, `nbH` is set equal to `CbHeight[xNbBk][yNbBk]`, `numCpMv` is set equal to `MotionModelIdc[xNbBk][yNbBk]+1`, and `gbIdxB` is set equal to `gbIdx[xNbBk][yNbBk]`.
 - For X being replaced by either 0 or 1, the following applies:
 - When `PredFlagLX[xNbBk][yNbBk]` is equal to TRUE, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the neighbouring luma coding block location (xNb, yNb) , the neighbouring luma coding block width and height (nbW, nbH) , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXB[cpIdx]` with `cpIdx = 0 .. numCpMv - 1` as output.
 - The following assignments are made:

[191]

$$\text{predFlagLXB} = \text{PredFlagLX}[xNbB_k][yNbB_k] \quad (8-547)$$

$$\text{refIdxLXB} = \text{RefIdxLX}[xNbB_k][yNbB_k] \quad (8-548)$$

5. When `sps_affine_enabled_flag` is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (xCb, yCb) , the luma coding block width and height $(cbWidth, cbHeight)$, the availability flags `availableA0`, `availableA1`, `availableA2`, `availableB0`, `availableB1`, `availableB2`, `availableB3` as inputs, and the availability flags `availableFlagConstK`, the reference indices `refIdxLXConstK`, prediction list utilization flags `predFlagLXConstK`, motion model indices `motionModelIdcConstK` and `cpMvpLXConstK[cpIdx]` with X being 0 or 1, K = 1..6, `cpIdx = 0..2` as outputs and `gbIdxConstK` is set equal to 0 with K = 1..6..
6. The initial subblock merging candidate list, `subblockMergeCandList`, is constructed as follows:
- ```

i = 0
if(availableFlagSbCol)
 subblockMergeCandList[i++] = SbCol
if(availableFlagA && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = A
if(availableFlagB && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = B
if(availableFlagConst1 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const1
if(availableFlagConst2 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const2
if(availableFlagConst3 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const3
if(availableFlagConst4 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const4
if(availableFlagConst5 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const5
if(availableFlagConst6 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const6

```
- (8-549)
7. The variable `numCurrMergeCand` and `numOrigMergeCand` are set equal to the number of merging candidates in the `subblockMergeCandList`.

[192]

8. When numCurrMergeCand is less than MaxNumSubblockMergeCand, the following is repeated until numCurrMergeCand is equal to MaxNumSubblockMergeCand, with mvZero[0] and mvZero[1] both being equal to 0:

- The reference indices, the prediction list utilization flags and the motion vectors of zeroCand<sub>m</sub> with m equal to ( numCurrMergeCand – numOrigMergeCand ) are derived as follows:

$$\text{refIdxL0ZeroCand}_m = 0 \quad (8-550)$$

$$\text{predFlagL0ZeroCand}_m = 1 \quad (8-551)$$

$$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero} \quad (8-552)$$

$$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero} \quad (8-553)$$

$$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero} \quad (8-554)$$

$$\text{refIdxL1ZeroCand}_m = (\text{tile\_group\_type} == B) ? 0 : -1 \quad (8-555)$$

$$\text{predFlagL1ZeroCand}_m = (\text{tile\_group\_type} == B) ? 1 : 0 \quad (8-556)$$

$$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero} \quad (8-557)$$

$$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero} \quad (8-558)$$

$$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero} \quad (8-559)$$

$$\text{motionModelIdxZeroCand}_m = 1 \quad (8-560)$$

$$\text{gbIdxZeroCand}_m = 0 \quad (8-561)$$

[193]

- The candidate zeroCand<sub>m</sub> with m equal to ( numCurrMergeCand – numOrigMergeCand ) is added at the end of subblockMergeCandList and numCurrMergeCand is incremented by 1 as follows:

$$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$$

The variables refIdxL0, refIdxL1, predFlagL0[ xSbIdx ][ ySbIdx ], predFlagL1[ xSbIdx ][ ySbIdx ], mvL0[ xSbIdx ][ ySbIdx ], mvL1[ xSbIdx ][ ySbIdx ], mvCL0[ xSbIdx ][ ySbIdx ], and mvCL1[ xSbIdx ][ ySbIdx ] with xSbIdx = 0..numSbX – 1, ySbIdx = 0..numSbY – 1 are derived as follows:

- If subblockMergeCandList[ merge\_subblock\_idx[ xCb ][ yCb ] ] is equal to SbCol, the bi-prediction weight index gbIdx is set equal to 0 and the following applies with X being 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$$

- For xSbIdx = 0..numSbX – 1, ySbIdx = 0..numSbY – 1, the following applies:

$$\text{predFlagLX}[ \text{xSbIdx} ][ \text{ySbIdx} ] = \text{predFlagLXSbCol}[ \text{xSbIdx} ][ \text{ySbIdx} ] \quad (8-564)$$

$$\text{mvLX}[ \text{xSbIdx} ][ \text{ySbIdx} ][ 0 ] = \text{mvLXSbCol}[ \text{xSbIdx} ][ \text{ySbIdx} ][ 0 ] \quad (8-565)$$

$$\text{mvLX}[ \text{xSbIdx} ][ \text{ySbIdx} ][ 1 ] = \text{mvLXSbCol}[ \text{xSbIdx} ][ \text{ySbIdx} ][ 1 ] \quad (8-566)$$

- When predFlagLX[ xSbIdx ][ ySbIdx ], is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with mvLX[ xSbIdx ][ ySbIdx ] and refIdxLX as inputs, and the output being mvCLX[ xSbIdx ][ ySbIdx ].

- The following assignment is made for x = xCb..xCb + cbWidth – 1 and y = yCb..yCb + cbHeight – 1:

$$\text{MotionModelIdx}[ x ][ y ] = 0 \quad (8-567)$$

- Otherwise (subblockMergeCandList[ merge\_subblock\_idx[ xCb ][ yCb ] ] is not equal to SbCol), the following applies with X being 0 or 1:

[194]

- The following assignments are made with N being the candidate at position  $\text{merge\_subblock\_idx}[x_{Cb}][y_{Cb}]$  in the subblock merging candidate list  $\text{subblockMergeCandList}$  ( $N = \text{subblockMergeCandList}[\text{merge\_subblock\_idx}[x_{Cb}][y_{Cb}]]$ ):

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$$

$$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$$

$$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$$

$$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$$

$$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-574)$$

- For  $x_{SbIdx} = 0..numSbX - 1$ ,  $y_{SbIdx} = 0..numSbY - 1$ , the following applies:

$$\text{predFlagLX}[x_{SbIdx}][y_{SbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$$

- When  $\text{predFlagLX}[0][0]$  is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location  $(x_{Cb}, y_{Cb})$ , the luma coding block width  $cbWidth$ , the luma prediction block height  $cbHeight$ , the number of control point motion vectors  $numCpMv$ , the control point motion vectors  $\text{cpMvLX}[cpIdx]$  with  $cpIdx$  being  $0..2$ , and the number of luma coding subblocks in horizontal direction  $numSbX$  and in vertical direction  $numSbY$  as inputs, the luma subblock motion vector array  $\text{mvLX}[x_{SbIdx}][y_{SbIdx}]$  and the chroma subblock motion vector array  $\text{mvCLX}[x_{SbIdx}][y_{SbIdx}]$  with  $x_{SbIdx} = 0..numSbX - 1$ ,  $y_{SbIdx} = 0..numSbY - 1$  as outputs.

- The following assignment is made for  $x = x_{Cb}..x_{Cb} + cbWidth - 1$  and  $y = y_{Cb}..y_{Cb} + cbHeight - 1$ :

$$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$$

[195]

[ 36 ]

#### 8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location  $(x_{Cb}, y_{Cb})$  of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable  $cbWidth$  specifying the width of the current coding block in luma samples,
- a variable  $cbHeight$  specifying the height of the current coding block in luma samples,
- the availability flags  $\text{availableFlagA}_0$ ,  $\text{availableFlagA}_1$ ,  $\text{availableFlagB}_0$ , and  $\text{availableFlagB}_1$  of the neighbouring coding units,
- the reference indices  $\text{refIdxLXA}_0$ ,  $\text{refIdxLXA}_1$ ,  $\text{refIdxLXB}_0$ , and  $\text{refIdxLXB}_1$  of the neighbouring coding units,
- the prediction list utilization flags  $\text{predFlagLXA}_0$ ,  $\text{predFlagLXA}_1$ ,  $\text{predFlagLXB}_0$ , and  $\text{predFlagLXB}_1$  of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy  $\text{mvLXA}_0$ ,  $\text{mvLXA}_1$ ,  $\text{mvLXB}_0$ , and  $\text{mvLXB}_1$  of the neighbouring coding units.

Outputs of this process are:

- the availability flag  $\text{availableFlagSbCol}$ ,
- the number of luma coding subblocks in horizontal direction  $numSbX$  and in vertical direction  $numSbY$ ,
- the reference indices  $\text{refIdxL0SbCol}$  and  $\text{refIdxL1SbCol}$ ,
- the luma motion vectors in 1/16 fractional-sample accuracy  $\text{mvL0SbCol}[x_{SbIdx}][y_{SbIdx}]$  and  $\text{mvL1SbCol}[x_{SbIdx}][y_{SbIdx}]$  with  $x_{SbIdx} = 0..numSbX - 1$ ,  $y_{SbIdx} = 0..numSbY - 1$ ,
- the prediction list utilization flags  $\text{predFlagL0SbCol}[x_{SbIdx}][y_{SbIdx}]$  and  $\text{predFlagL1SbCol}[x_{SbIdx}][y_{SbIdx}]$  with  $x_{SbIdx} = 0..numSbX - 1$ ,  $y_{SbIdx} = 0..numSbY - 1$ ,
- the bi-prediction weight index  $\text{gbIdxSbCol}$ .

The availability flag  $\text{availableFlagSbCol}$  is derived as follows.

[196]

- If one or more of the following conditions is true, availableFlagSbCol is set equal to 0.
  - tile\_group\_temporal\_mvp\_enable\_flag is equal to 0.
  - sps\_sbtmvp\_flag is equal to 0.
  - cbWidth is less than 8.
  - cbHeight is less than 8.
- Otherwise, the following ordered steps apply:
  1. The location ( xCtb, yCtb ) of the top-left sample of the luma coding tree block that contains the current coding block and the location ( xCtr, yCtr ) of the below-right center sample of the current luma coding block are derived as follows:
 
$$xCtb = (xCb \gg CtuLog2Size) \ll CtuLog2Size \quad (8-577)$$

$$yCtb = (yCb \gg CtuLog2Size) \ll CtuLog2Size \quad (8-578)$$

$$xCtr = xCb + (cbWidth / 2) \quad (8-579)$$

$$yCtr = yCb + (cbHeight / 2) \quad (8-580)$$
  2. The luma location ( xColCtrCb, yColCtrCb ) is set equal to the top-left sample of the collocated luma coding block covering the location given by ( xCtr, yCtr ) inside ColPic relative to the top-left luma sample of the collocated picture specified by ColPic.

[197]

3. The derivation process for subblock-based temporal merging base motion data as specified in clause 8.4.4.4 is invoked with the location ( xCtb, yCtb ), the location ( xColCtrCb, yColCtrCb ), the availability flags availableFlagA<sub>0</sub>, availableFlagA<sub>1</sub>, availableFlagB<sub>0</sub> and availableFlagB<sub>1</sub>, and the prediction list utilization flags predFlagLXA<sub>0</sub>, predFlagLXA<sub>1</sub>, predFlagLXB<sub>0</sub> and predFlagLXB<sub>1</sub>, and the reference indices refIdxLXA<sub>0</sub>, refIdxLXA<sub>1</sub>, refIdxLXB<sub>0</sub> and refIdxLXB<sub>1</sub>, and the motion vectors mvLXA<sub>0</sub>, mvLXA<sub>1</sub>, mvLXB<sub>0</sub> and mvLXB<sub>1</sub>, with X being 0 and 1 as inputs and the motion vectors ctrMvLX, the prediction list utilization flags ctrPredFlagLX and the reference indices ctrRefIdxLX of the collocated block, with X being 0 and 1, the bi-prediction weight index gbIdxSbCol, and the temporal motion vector tempMV as outputs.
4. The variable availableFlagSbCol is derived as follows:
  - If both ctrPredFlagL0 and ctrPredFlagL1 are equal to 0, availableFlagSbCol is set equal to 0.
  - Otherwise, availableFlagSbCol is set equal to 1.

When availableFlagSbCol is equal to 1, the following applies:

- The variables numSbX, numSbY, sbWidth, sbHeight and refIdxLXSbCol are derived as follows:
 
$$\text{numSbX} = \text{cbWidth} \gg 3 \quad (8-581)$$

$$\text{numSbY} = \text{cbHeight} \gg 3 \quad (8-582)$$

$$\text{sbWidth} = \text{cbWidth} / \text{numSbX} \quad (8-583)$$

$$\text{sbHeight} = \text{cbHeight} / \text{numSbY} \quad (8-584)$$

$$\text{refIdxLXSbCol} = 0 \quad (8-585)$$
- For xSbIdx = 0..numSbX - 1 and ySbIdx = 0..numSbY - 1, the motion vectors mvLXSbCol[ xSbIdx ][ ySbIdx ] and prediction list utilization flags predFlagLXSbCol[ xSbIdx ][ ySbIdx ] are derived as follows:

[198]

- The luma location (  $x_{Sb}$ ,  $y_{Sb}$  ) specifying the top-left sample of the current coding subblock relative to the top-left luma sample of the current picture is derived as follows:

$$x_{Sb} = x_{Cb} + x_{SbIdx} * sbWidth \quad (8-586)$$

$$y_{Sb} = y_{Cb} + y_{SbIdx} * sbHeight \quad (8-587)$$

- The location (  $x_{ColSb}$ ,  $y_{ColSb}$  ) of the collocated subblock inside ColPic is derived as follows.

$$x_{ColSb} = Clip3( x_{Ctb}, \\ Min( CurPicWidthInSamplesY - 1, x_{Ctb} + ( 1 \ll CtbLog2SizeY ) + 3 ), \\ x_{Sb} + ( tempMv[0] \gg 4 ) ) \quad (8-588)$$

$$y_{ColSb} = Clip3( y_{Ctb}, \\ Min( CurPicHeightInSamplesY - 1, y_{Ctb} + ( 1 \ll CtbLog2SizeY ) - 1 ), \\ y_{Sb} + ( tempMv[1] \gg 4 ) ) \quad (8-589)$$

- The variable currCb specifies the luma coding block covering the current coding subblock inside the current picture.
- The variable colCb specifies the luma coding block covering the modified location given by ( (  $x_{ColSb} \gg 3$  )  $\ll$  3, (  $y_{ColSb} \gg 3$  )  $\ll$  3 ) inside the ColPic.
- The luma location (  $x_{ColCb}$ ,  $y_{ColCb}$  ) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (  $x_{ColCb}$ ,  $y_{ColCb}$  ), refIdxL0 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL0SbCol[  $x_{SbIdx}$  ][  $y_{SbIdx}$  ] and availableFlagL0SbCol.
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (  $x_{ColCb}$ ,  $y_{ColCb}$  ), refIdxL1 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL1SbCol[  $x_{SbIdx}$  ][  $y_{SbIdx}$  ] and availableFlagL1SbCol.
- When availableFlagL0SbCol and availableFlagL1SbCol are both equal to 0, the following applies for X being 0 and 1:

$$mvLXSbCol[ x_{SbIdx} ][ y_{SbIdx} ] = ctrMvLX \quad (8-590)$$

$$predFlagLXSbCol[ x_{SbIdx} ][ y_{SbIdx} ] = ctrPredFlagLX \quad (8-591)$$

[199] [37]

**8.4.4.4 Derivation process for subblock-based temporal merging base motion data**

Inputs to this process are:

- the location ( xCtb, yCtb ) of the top-left sample of the luma coding tree block that contains the current coding block,
- the location ( xColCtrCb, yColCtrCb ) of the top-left sample of the collocated luma coding block that covers the below-right center sample.
- the availability flags availableFlagA<sub>0</sub>, availableFlagA<sub>1</sub>, availableFlagB<sub>0</sub>, and availableFlagB<sub>1</sub> of the neighbouring coding units,
- the reference indices refIdxLXA<sub>0</sub>, refIdxLXA<sub>1</sub>, refIdxLXB<sub>0</sub>, and refIdxLXB<sub>1</sub> of the neighbouring coding units,
- the prediction list utilization flags predFlagLXA<sub>0</sub>, predFlagLXA<sub>1</sub>, predFlagLXB<sub>0</sub>, and predFlagLXB<sub>1</sub> of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy mvLXA<sub>0</sub>, mvLXA<sub>1</sub>, mvLXB<sub>0</sub>, and mvLXB<sub>1</sub> of the neighbouring coding units.

Outputs of this process are:

- the motion vectors ctrMvL0 and ctrMvL1,
- the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1,
- the reference indices ctrRefIdxL0 and ctrRefIdxL1,
- the temporal motion vector tempMV,
- the bi-prediction weight index gbiIdxSbCol.

The variable tempMv is set as follows:

$$\text{tempMv}[0] = 0 \quad (8-592)$$

$$\text{tempMv}[1] = 0 \quad (8-593)$$

The variable currPic specifies the current picture.

The variable availableFlagN is set equal to FALSE, and the following applies:

- When availableFlagA<sub>1</sub> is equal to 1, the following applies:
  - availableFlagN is set equal to TRUE,
  - refIdxLXN is set equal to refIdxLXA<sub>0</sub> and mvLXN is set equal to mvLXA<sub>0</sub>, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagLB<sub>1</sub> is equal to 1, the following applies:
  - availableFlagN is set equal to TRUE,

[200]

- $\text{refIdxLXN}$  is set equal to  $\text{refIdxLXB}_0$  and  $\text{mvLXN}$  is set equal to  $\text{mvLXB}_0$ , for  $X$  being replaced by 0 and 1.
- When  $\text{availableFlagN}$  is equal to FALSE and  $\text{availableFlagB}_0$  is equal to 1, the following applies:
  - $\text{availableFlagN}$  is set equal to TRUE.
  - $\text{refIdxLXN}$  is set equal to  $\text{refIdxLXB}_1$  and  $\text{mvLXN}$  is set equal to  $\text{mvLXB}_1$ , for  $X$  being replaced by 0 and 1.
- When  $\text{availableFlagN}$  is equal to FALSE and  $\text{availableFlagA}_0$  is equal to 1, the following applies:
  - $\text{availableFlagN}$  is set equal to TRUE.
  - $\text{refIdxLXN}$  is set equal to  $\text{refIdxLXA}_1$  and  $\text{mvLXN}$  is set equal to  $\text{mvLXA}_1$ , for  $X$  being replaced by 0 and 1.

When  $\text{availableFlagN}$  is equal to TRUE, the following applies:

- If all of the following conditions are true,  $\text{tempMV}$  is set equal to  $\text{mvL1N}$ :
  - $\text{predFlagL1N}$  is equal to 1,
  - $\text{DiffPicOrderCnt}(\text{ColPic}, \text{RefPicList1}[\text{refIdxL1N}])$  is equal to 0,
  - $\text{DiffPicOrderCnt}(\text{aPic}, \text{currPic})$  is less than or equal to 0 for every picture  $\text{aPic}$  in every reference picture list of the current tile group,
  - $\text{tile\_group\_type}$  is equal to B,
  - $\text{collocated\_from\_l0\_flag}$  is equal to 0.
- Otherwise if all of the following conditions are true,  $\text{tempMV}$  is set equal to  $\text{mvL0N}$ :
  - $\text{predFlagL0N}$  is equal to 1,
  - $\text{DiffPicOrderCnt}(\text{ColPic}, \text{RefPicList0}[\text{refIdxL0N}])$  is equal to 0.

[201]

The location (  $x\text{ColCb}$ ,  $y\text{ColCb}$  ) of the collocated block inside  $\text{ColPic}$  is derived as follows.

```

xColCb = Clip3(xCtb,
 Min(CurPicWidthInSamplesY - 1, xCtb + (1 << CtbLog2SizeY) + 3), (8-59
4)
 xColCtCb + (tempMv[0] >> 4))

yColCb = Clip3(yCtb,
 Min(CurPicHeightInSamplesY - 1, yCtb + (1 << CtbLog2SizeY) - 1), (8-5
95)
 yColCtCb + (tempMv[1] >> 4))

```

The array  $\text{colPredMode}$  is set equal to the prediction mode array  $\text{CuPredMode}$  of the collocated picture specified by  $\text{ColPic}$ .

The motion vectors  $\text{ctrMvL0}$  and  $\text{ctrMvL1}$ , the prediction list utilization flags  $\text{ctrPredFlagL0}$  and  $\text{ctrPredFlagL1}$ , and the reference indices  $\text{ctrRefIdxL0}$  and  $\text{ctrRefIdxL1}$  are derived as follows:

- If  $\text{colPredMode}[\text{xColCb}][\text{yColCb}]$  is equal to MODE\_INTER, the following applies:
  - The variable  $\text{currCb}$  specifies the luma coding block covering (  $\text{xCtCb}$ ,  $\text{yCtCb}$  ) inside the current picture.
  - The variable  $\text{colCb}$  specifies the luma coding block covering the modified location given by ( (  $\text{xColCb} \gg 3$  ) << 3, (  $\text{yColCb} \gg 3$  ) << 3 ) inside the  $\text{ColPic}$ .
  - The luma location (  $\text{xColCb}$ ,  $\text{yColCb}$  ) is set equal to the top-left sample of the collocated luma coding block specified by  $\text{colCb}$  relative to the top-left luma sample of the collocated picture specified by  $\text{ColPic}$ .
  - The  $\text{gbIdxSbCol}$  is set equal to  $\text{gbIdxcolCb}$ .

[202]

- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with `currCb`, `colCb`, `(xColCb, yColCb)`, `centerRefIdxL0`, and `sbFlag` set equal to 1 as inputs and the output being assigned to `ctrMvL0` and `ctrPredFlagL0`.
- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with `currCb`, `colCb`, `(xColCb, yColCb)`, `centerRefIdxL1`, and `sbFlag` set equal to 1 as inputs and the output being assigned to `ctrMvL1` and `ctrPredFlagL1`.
- Otherwise, the following applies:
 

$$\text{ctrPredFlagL0} = 0 \quad (8-596)$$

$$\text{ctrPredFlagL1} = 0 \quad (8-597)$$

[203]

상기 표 5, 표 6 및 표 7을 참조하면, `gbIdx`는 쌍 예측 가중치 인덱스를 나타낼 수 있으며, `gbIdxSbCol`은 서브블록 기반 시간적 머지 후보(예를 들어, 서브블록 기반 머지 후보 리스트 내의 시간적 움직임 벡터 후보)를 위한 쌍 예측 가중치 인덱스를 나타낼 수 있다. 서브블록 기반 시간적 머지를 위한 베이스 움직임 정보를 도출하는 절차(8.4.4.4)에서 상기 `gbIdxSbCol`은 `gbIdxcolCb`로 도출될 수 있다. 즉, 서브블록 기반 시간적 움직임 벡터 후보의 가중치 인덱스는 시간적 센터 블록의 가중치 인덱스로 도출될 수 있다. 예를 들어, 상기 시간적 센터 블록은 `col`(col) 블록 또는 `col`(col) 블록의 센터에 위치한 서브블록 또는 샘플을 나타낼 수 있으며, 구체적으로는 `col` 블록의 중앙 4개의 서브블록 또는 샘플 중 우하측에 위치한 서브블록 또는 샘플을 나타낼 수 있다.

[204]

또는, 서브블록 기반 시간적 움직임 벡터 후보에 대한 가중평균을 위한 가중치 인덱스 정보는 각각의 서브블록 단위의 가중치 인덱스 정보를 기반으로 도출될 수 있고, 서브블록이 이용 가능하지 않은 경우, 시간적 센터(center) 블록의 가중치 인덱스 정보를 기반으로 도출될 수 있다. 예를 들어, 상기 시간적 센터 블록은 `col`(col) 블록 또는 `col`(col) 블록의 센터에 위치한 서브블록 또는 샘플을 나타낼 수 있으며, 구체적으로는 `col` 블록의 중앙 4개의 서브블록 또는 샘플 중 우하측에 위치한 서브블록 또는 샘플을 나타낼 수 있다. 예를 들어, 이 경우 서브블록 머지 모드 내의 움직임 벡터 및 참조 인덱스를 도출하는 절차, 서브블록 기반 시간적 머지 후보를 도출하는 절차 및 서브블록 기반 시간적 머지를 위한 베이스 움직임 정보를 도출하는 절차는 다음의 표 8, 표 9 및 표 10과 같을 수 있다.

[205] [ 35.8 ]

**8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode**

Inputs to this process are:

- a luma location (  $x_{Cb}$ ,  $y_{Cb}$  ) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables  $cbWidth$  and  $cbHeight$  specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction  $numSbX$  and in vertical direction  $numSbY$ ,
- the reference indices  $refIdxL0$  and  $refIdxL1$ ,
- the prediction list utilization flag arrays  $predFlagL0[xSbIdx][ySbIdx]$  and  $predFlagL1[xSbIdx][ySbIdx]$ ,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy  $mvL0[xSbIdx][ySbIdx]$  and  $mvL1[xSbIdx][ySbIdx]$  with  $xSbIdx = 0..numSbX - 1$ ,  $ySbIdx = 0..numSbY - 1$ ,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy  $mvCL0[xSbIdx][ySbIdx]$  and  $mvCL1[xSbIdx][ySbIdx]$  with  $xSbIdx = 0..numSbX - 1$ ,  $ySbIdx = 0..numSbY - 1$ ,
- the bi-prediction weight index  $gbIdx$ .

The variables  $numSbX$ ,  $numSbY$  and the subblock merging candidate list,  $subblockMergeCandList$  are derived by the following ordered steps:

1. When  $sps\_sbtmvp\_enabled\_flag$  is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (  $x_{Cb}$ ,  $y_{Cb}$  ), the luma coding block width  $cbWidth$ , the luma coding block height  $cbHeight$  and the luma coding block width as inputs, and the output being the availability flags  $availableFlagA_0$ ,  $availableFlagA_1$ ,  $availableFlagB_0$ ,  $availableFlagB_1$  and  $availableFlagB_2$ , the reference indices  $refIdxLXA_0$ ,  $refIdxLXA_1$ ,  $refIdxLXB_0$ ,  $refIdxLXB_1$  and  $refIdxLXB_2$ , the prediction list utilization flags  $predFlagLXA_0$ ,  $predFlagLXA_1$ ,  $predFlagLXB_0$ ,  $predFlagLXB_1$  and  $predFlagLXB_2$ , and the motion vectors  $mvLXA_0$ ,  $mvLXA_1$ ,  $mvLXB_0$ ,  $mvLXB_1$  and  $mvLXB_2$ , with  $X$  being 0 or 1.

[206]

- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (  $x_{Cb}$ ,  $y_{Cb}$  ), the luma coding block width  $cbWidth$ , the luma coding block height  $cbHeight$ , the availability flags  $availableFlagA_0$ ,  $availableFlagA_1$ ,  $availableFlagB_0$ ,  $availableFlagB_1$ , the reference indices  $refIdxLXA_0$ ,  $refIdxLXA_1$ ,  $refIdxLXB_0$ ,  $refIdxLXB_1$ , the prediction list utilization flags  $predFlagLXA_0$ ,  $predFlagLXA_1$ ,  $predFlagLXB_0$ ,  $predFlagLXB_1$  and the motion vectors  $mvLXA_0$ ,  $mvLXA_1$ ,  $mvLXB_0$ ,  $mvLXB_1$  as inputs and the output being the availability flag  $availableFlagSbCol$ , the number of luma coding subblocks in horizontal direction  $numSbX$  and in vertical direction  $numSbY$ , the reference indices  $refIdxLXSbCol$ , the bi-prediction weight index  $gbIdxSbCol[xSbIdx][ySbIdx]$ , the luma motion vectors  $mvLXSbCol[xSbIdx][ySbIdx]$  and the prediction list utilization flags  $predFlagLXSbCol[xSbIdx][ySbIdx]$  with  $xSbIdx = 0..numSbX - 1$ ,  $ySbIdx = 0..numSbY - 1$  and  $X$  being 0 or 1.
2. When  $sps\_affine\_enabled\_flag$  is equal to 1, the sample locations (  $x_{NbA_0}$ ,  $y_{NbA_0}$  ), (  $x_{NbA_1}$ ,  $y_{NbA_1}$  ), (  $x_{NbA_2}$ ,  $y_{NbA_2}$  ), (  $x_{NbB_0}$ ,  $y_{NbB_0}$  ), (  $x_{NbB_1}$ ,  $y_{NbB_1}$  ), (  $x_{NbB_2}$ ,  $y_{NbB_2}$  ), (  $x_{NbB_3}$ ,  $y_{NbB_3}$  ), and the variables  $numSbX$  and  $numSbY$  are derived as follows:

$$(x_{A_0}, y_{A_0}) = (x_{Cb} - 1, y_{Cb} + cbHeight) \quad (8-536)$$

$$(x_{A_1}, y_{A_1}) = (x_{Cb} - 1, y_{Cb} + cbHeight - 1) \quad (8-537)$$

$$(x_{A_2}, y_{A_2}) = (x_{Cb} - 1, y_{Cb}) \quad (8-538)$$

$$(x_{B_0}, y_{B_0}) = (x_{Cb} + cbWidth, y_{Cb} - 1) \quad (8-539)$$

$$(x_{B_1}, y_{B_1}) = (x_{Cb} + cbWidth - 1, y_{Cb} - 1) \quad (8-540)$$

$$(x_{B_2}, y_{B_2}) = (x_{Cb} - 1, y_{Cb} - 1) \quad (8-541)$$

$$(x_{B_3}, y_{B_3}) = (x_{Cb}, y_{Cb} - 1) \quad (8-542)$$

[207]

$$\text{numSbX} = \text{cbWidth} \gg 2 \quad (8-543)$$

$$\text{numSbY} = \text{cbHeight} \gg 2 \quad (8-544)$$

3. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagA` is set equal to FALSE and the following applies for  $(xNbA_k, yNbA_k)$  from  $(xNbA_0, yNbA_0)$  to  $(xNbA_1, yNbA_1)$ :

- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location  $(xCurr, yCurr)$  set equal to  $(xCb, yCb)$  and the neighbouring luma location  $(xNbA_k, yNbA_k)$  as inputs, and the output is assigned to the block availability flag `availableAk`.
- When `availableAk` is equal to TRUE and `MotionModelIdc[xNbAk][yNbAk]` is greater than 0 and `availableFlagA` is equal to FALSE, the following applies:
  - The variable `availableFlagA` is set equal to TRUE, `motionModelIdcA` is set equal to `MotionModelIdc[xNbAk][yNbAk]`,  $(xNb, yNb)$  is set equal to  $(CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k])$ , `nbW` is set equal to `CbWidth[xNbAk][yNbAk]`, `nbH` is set equal to `CbHeight[xNbAk][yNbAk]`, `numCpMv` is set equal to `MotionModelIdc[xNbAk][yNbAk] + 1`, and `gbIdxA` is set equal to `GbiIdx[xNbAk][yNbAk]`.
  - For X being replaced by either 0 or 1, the following applies:
    - When `PredFlagLX[xNbAk][yNbAk]` is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location  $(xCb, yCb)$ , the luma coding block width and height  $(cbWidth, cbHeight)$ , the neighbouring luma coding block location  $(xNb, yNb)$ , the neighbouring luma coding block width and height  $(nbW, nbH)$ , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXA[cpIdx]` with `cpIdx = 0 .. numCpMv - 1` as output.
    - The following assignments are made:

$$\text{predFlagLXA} = \text{PredFlagLX}[xNbA_k][yNbA_k] \quad (8-545)$$

$$\text{refIdxLXA} = \text{RefIdxLX}[xNbA_k][yNbA_k] \quad (8-546)$$

[208]

4. When `sps_affine_enabled_flag` is equal to 1, the variable `availableFlagB` is set equal to FALSE and the following applies for  $(xNbB_k, yNbB_k)$  from  $(xNbB_0, yNbB_0)$  to  $(xNbB_2, yNbB_2)$ :

- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location  $(xCurr, yCurr)$  set equal to  $(xCb, yCb)$  and the neighbouring luma location  $(xNbB_k, yNbB_k)$  as inputs, and the output is assigned to the block availability flag `availableBk`.
- When `availableBk` is equal to TRUE and `MotionModelIdc[xNbBk][yNbBk]` is greater than 0 and `availableFlagB` is equal to FALSE, the following applies:
  - The variable `availableFlagB` is set equal to TRUE, `motionModelIdcB` is set equal to `MotionModelIdc[xNbBk][yNbBk]`,  $(xNb, yNb)$  is set equal to  $(CbPosX[xNbB_k][yNbB_k], CbPosY[xNbB_k][yNbB_k])$ , `nbW` is set equal to `CbWidth[xNbBk][yNbBk]`, `nbH` is set equal to `CbHeight[xNbBk][yNbBk]`, `numCpMv` is set equal to `MotionModelIdc[xNbBk][yNbBk] + 1`, and `gbIdxB` is set equal to `GbiIdx[xNbBk][yNbBk]`.
  - For X being replaced by either 0 or 1, the following applies:
    - When `PredFlagLX[xNbBk][yNbBk]` is equal to TRUE, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location  $(xCb, yCb)$ , the luma coding block width and height  $(cbWidth, cbHeight)$ , the neighbouring luma coding block location  $(xNb, yNb)$ , the neighbouring luma coding block width and height  $(nbW, nbH)$ , and the number of control point motion vectors `numCpMv` as input, the control point motion vector predictor candidates `cpMvLXB[cpIdx]` with `cpIdx = 0 .. numCpMv - 1` as output.
    - The following assignments are made:

[209]

$$\text{predFlagLXB} = \text{PredFlagLX}[xNbB_k][yNbB_k] \quad (8-547)$$

$$\text{refIdxLXB} = \text{RefIdxLX}[xNbB_k][yNbB_k] \quad (8-548)$$

5. When `sps_affine_enabled_flag` is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location ( `xCb`, `yCb` ), the luma coding block width and height ( `cbWidth`, `cbHeight` ), the availability flags `availableA0`, `availableA1`, `availableA2`, `availableB0`, `availableB1`, `availableB2`, `availableB3` as inputs, and the availability flags `availableFlagConstK`, the reference indices `refIdxLXConstK`, prediction list utilization flags `predFlagLXConstK`, motion model indices `motionModelIdxConstK` and `cpMvpLXConstK[cpIdx]` with `X` being 0 or 1, `K` = 1..6, `cpIdx` = 0..2 as outputs and `gbIdxConstK` is set equal to 0 with `K` = 1..6..
6. The initial subblock merging candidate list, `subblockMergeCandList`, is constructed as follows:

```

i = 0
if(availableFlagSbCol)
 subblockMergeCandList[i++] = SbCol
if(availableFlagA && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = A
if(availableFlagB && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = B
if(availableFlagConst1 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const1
if(availableFlagConst2 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const2
if(availableFlagConst3 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const3
if(availableFlagConst4 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const4
if(availableFlagConst5 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const5
if(availableFlagConst6 && i < MaxNumSubblockMergeCand)
 subblockMergeCandList[i++] = Const6

```

(8-549)

[210]

7. The variable `numCurrMergeCand` and `numOrigMergeCand` are set equal to the number of merging candidates in the `subblockMergeCandList`.
8. When `numCurrMergeCand` is less than `MaxNumSubblockMergeCand`, the following is repeated until `numCurrMergeCand` is equal to `MaxNumSubblockMergeCand`, with `mvZero[0]` and `mvZero[1]` both being equal to 0:
- The reference indices, the prediction list utilization flags and the motion vectors of `zeroCandm` with `m` equal to ( `numCurrMergeCand` - `numOrigMergeCand` ) are derived as follows:

$$\text{refIdxL0ZeroCand}_m = 0 \quad (8-550)$$

$$\text{predFlagL0ZeroCand}_m = 1 \quad (8-551)$$

$$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero} \quad (8-552)$$

$$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero} \quad (8-553)$$

$$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero} \quad (8-554)$$

$$\text{refIdxL1ZeroCand}_m = (\text{tile\_group\_type} == B) ? 0 : -1 \quad (8-555)$$

$$\text{predFlagL1ZeroCand}_m = (\text{tile\_group\_type} == B) ? 1 : 0 \quad (8-556)$$

$$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero} \quad (8-557)$$

$$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero} \quad (8-558)$$

$$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero} \quad (8-559)$$

$$\text{motionModelIdxZeroCand}_m = 1 \quad (8-560)$$

$$\text{gbIdxZeroCand}_m = 0 \quad (8-561)$$

[211]

- The candidate  $\text{zeroCand}_m$  with  $m$  equal to  $(\text{numCurrMergeCand} - \text{numOrigMergeCand})$  is added at the end of  $\text{subblockMergeCandList}$  and  $\text{numCurrMergeCand}$  is incremented by 1 as follows:

$$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$$

The variables  $\text{refIdxL0}$ ,  $\text{refIdxL1}$ ,  $\text{predFlagL0}[\text{xSbIdx}][\text{ySbIdx}]$ ,  $\text{predFlagL1}[\text{xSbIdx}][\text{ySbIdx}]$ ,  $\text{mvL0}[\text{xSbIdx}][\text{ySbIdx}]$ ,  $\text{mvL1}[\text{xSbIdx}][\text{ySbIdx}]$ ,  $\text{mvCL0}[\text{xSbIdx}][\text{ySbIdx}]$ , and  $\text{mvCL1}[\text{xSbIdx}][\text{ySbIdx}]$  with  $\text{xSbIdx} = 0.. \text{numSbX} - 1$ ,  $\text{ySbIdx} = 0.. \text{numSbY} - 1$  are derived as follows:

- If  $\text{subblockMergeCandList}[\text{merge\_subblock\_idx}[\text{xCb}][\text{yCb}]]$  is equal to  $\text{SbCol}$ , the bi-prediction weight index  $\text{gbIdx}$  is set equal to 0 and the following applies with  $X$  being 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$$

- For  $\text{xSbIdx} = 0.. \text{numSbX} - 1$ ,  $\text{ySbIdx} = 0.. \text{numSbY} - 1$ , the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] \quad (8-564)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][0] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][0] \quad (8-565)$$

$$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][1] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][1] \quad (8-566)$$

- When  $\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}]$  is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with  $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$  and  $\text{refIdxLX}$  as inputs, and the output being  $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$ .

- The following assignment is made for  $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$  and  $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$ :

$$\text{MotionModelIdx}[x][y] = 0 \quad (8-567)$$

- Otherwise ( $\text{subblockMergeCandList}[\text{merge\_subblock\_idx}[\text{xCb}][\text{yCb}]]$  is not equal to  $\text{SbCol}$ ), the following applies with  $X$  being 0 or 1:

[212]

- The following assignments are made with  $N$  being the candidate at position  $\text{merge\_subblock\_idx}[\text{xCb}][\text{yCb}]$  in the subblock merging candidate list  $\text{subblockMergeCandList}$  ( $N = \text{subblockMergeCandList}[\text{merge\_subblock\_idx}[\text{xCb}][\text{yCb}]]$ ):

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$$

$$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$$

$$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$$

$$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$$

$$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-574)$$

- For  $\text{xSbIdx} = 0.. \text{numSbX} - 1$ ,  $\text{ySbIdx} = 0.. \text{numSbY} - 1$ , the following applies:

$$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$$

- When  $\text{predFlagLX}[0][0]$  is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location ( $\text{xCb}$ ,  $\text{yCb}$ ), the luma coding block width  $\text{cbWidth}$ , the luma prediction block height  $\text{cbHeight}$ , the number of control point motion vectors  $\text{numCpMv}$ , the control point motion vectors  $\text{cpMvLX}[\text{cpIdx}]$  with  $\text{cpIdx}$  being 0..2, and the number of luma coding subblocks in horizontal direction  $\text{numSbX}$  and in vertical direction  $\text{numSbY}$  as inputs, the luma subblock motion vector array  $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$  and the chroma subblock motion vector array  $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$  with  $\text{xSbIdx} = 0.. \text{numSbX} - 1$ ,  $\text{ySbIdx} = 0.. \text{numSbY} - 1$  as outputs.

- The following assignment is made for  $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$  and  $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$ :

$$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$$

[213] [ 359 ]

**8.4.4.3 Derivation process for subblock-based temporal merging candidates**

Inputs to this process are:

- a luma location (  $x_{Cb}$ ,  $y_{Cb}$  ) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable  $cbWidth$  specifying the width of the current coding block in luma samples,
- a variable  $cbHeight$  specifying the height of the current coding block in luma samples.
- the availability flags  $availableFlagA_0$ ,  $availableFlagA_1$ ,  $availableFlagB_0$ , and  $availableFlagB_1$  of the neighbouring coding units,
- the reference indices  $refIdxLXA_0$ ,  $refIdxLXA_1$ ,  $refIdxLXB_0$ , and  $refIdxLXB_1$  of the neighbouring coding units,
- the prediction list utilization flags  $predFlagLXA_0$ ,  $predFlagLXA_1$ ,  $predFlagLXB_0$ , and  $predFlagLXB_1$  of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy  $mvLXA_0$ ,  $mvLXA_1$ ,  $mvLXB_0$ , and  $mvLXB_1$  of the neighbouring coding units.

Outputs of this process are:

- the availability flag  $availableFlagSbCol$ ,
- the number of luma coding subblocks in horizontal direction  $numSbX$  and in vertical direction  $numSbY$ ,
- the reference indices  $refIdxL0SbCol$  and  $refIdxL1SbCol$ ,
- the luma motion vectors in 1/16 fractional-sample accuracy  $mvL0SbCol[xSbIdx][ySbIdx]$  and  $mvL1SbCol[xSbIdx][ySbIdx]$  with  $xSbIdx = 0..numSbX - 1$ ,  $ySbIdx = 0..numSbY - 1$ ,
- the bi-prediction weight index  $gbiIdxSbCol[xSbIdx][ySbIdx]$ , the prediction list utilization flags  $predFlagL0SbCol[xSbIdx][ySbIdx]$  and  $predFlagL1SbCol[xSbIdx][ySbIdx]$  with  $xSbIdx = 0..numSbX - 1$ ,  $ySbIdx = 0..numSbY - 1$ .

[214]

The availability flag  $availableFlagSbCol$  is derived as follows.

- If one or more of the following conditions is true,  $availableFlagSbCol$  is set equal to 0.
  - $tile\_group\_temporal\_mvp\_enable\_flag$  is equal to 0.
  - $sps\_sbtmp\_flag$  is equal to 0.
  - $cbWidth$  is less than 8.
  - $cbHeight$  is less than 8.
- Otherwise, the following ordered steps apply:
  1. The location (  $x_{Ctb}$ ,  $y_{Ctb}$  ) of the top-left sample of the luma coding tree block that contains the current coding block and the location (  $x_{Ctr}$ ,  $y_{Ctr}$  ) of the below-right center sample of the current luma coding block are derived as follows:
 
$$x_{Ctb} = (x_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-577)$$

$$y_{Ctb} = (y_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-578)$$

$$x_{Ctr} = x_{Cb} + (cbWidth / 2) \quad (8-579)$$

$$y_{Ctr} = y_{Cb} + (cbHeight / 2) \quad (8-580)$$
  2. The luma location (  $x_{ColCtrCb}$ ,  $y_{ColCtrCb}$  ) is set equal to the top-left sample of the collocated luma coding block covering the location given by (  $x_{Ctr}$ ,  $y_{Ctr}$  ) inside  $ColPic$  relative to the top-left luma sample of the collocated picture specified by  $ColPic$ .

[215]

3. The derivation process for subblock-based temporal merging base motion data as specified in clause 8.4.4.4 is invoked with the location (xCtb, yCtb), the location (xCbCol, yCbCol), the availability flags availableFlagA<sub>0</sub>, availableFlagA<sub>1</sub>, availableFlagB<sub>0</sub> and availableFlagB<sub>1</sub>, and the prediction list utilization flags predFlagLXA<sub>0</sub>, predFlagLXA<sub>1</sub>, predFlagLXB<sub>0</sub> and predFlagLXB<sub>1</sub>, and the reference indices refIdxLXA<sub>0</sub>, refIdxLXA<sub>1</sub>, refIdxLXB<sub>0</sub> and refIdxLXB<sub>1</sub>, and the motion vectors mvLXA<sub>0</sub>, mvLXA<sub>1</sub>, mvLXB<sub>0</sub> and mvLXB<sub>1</sub>, with X being 0 and 1 as inputs and the motion vectors ctrMvLX, the prediction list utilization flags ctrPredFlagLX and the reference indices ctrRefIdxLX of the collocated block, with X being 0 and 1, the bi-prediction weight index ctrgbiIdx, and the temporal motion vector tempMV as outputs.
4. The variable availableFlagSbCol is derived as follows:
  - If both ctrPredFlagL0 and ctrPredFlagL1 are equal to 0, availableFlagSbCol is set equal to 0.
  - Otherwise, availableFlagSbCol is set equal to 1.

When availableFlagSbCol is equal to 1, the following applies:

- The variables numSbX, numSbY, sbWidth, sbHeight and refIdxLXSbCol are derived as follows:
 
$$\text{numSbX} = \text{cbWidth} \gg 3 \quad (8-581)$$

$$\text{numSbY} = \text{cbHeight} \gg 3 \quad (8-582)$$

$$\text{sbWidth} = \text{cbWidth} / \text{numSbX} \quad (8-583)$$

$$\text{sbHeight} = \text{cbHeight} / \text{numSbY} \quad (8-584)$$

$$\text{refIdxLXSbCol} = 0 \quad (8-585)$$
- For xSbIdx = 0 .. numSbX – 1 and ySbIdx = 0 .. numSbY – 1, the motion vectors mvLXSbCol[ xSbIdx ][ ySbIdx ] and prediction list utilization flags predFlagLXSbCol[ xSbIdx ][ ySbIdx ] are derived as follows:

[216]

- The luma location (xSb, ySb) specifying the top-left sample of the current coding subblock relative to the top-left luma sample of the current picture is derived as follows:
 
$$\text{xSb} = \text{xCb} + \text{xSbIdx} * \text{sbWidth} \quad (8-586)$$

$$\text{ySb} = \text{yCb} + \text{ySbIdx} * \text{sbHeight} \quad (8-587)$$
- The location (xCbSb, yCbSb) of the collocated subblock inside ColPic is derived as follows.
 
$$\begin{aligned} \text{xCbSb} = & \text{Clip3}( \text{xCtb}, \\ & \text{Min}( \text{CurPicWidthInSamplesY} - 1, \text{xCtb} + ( 1 \ll \text{CtbLog2SizeY} ) + 3 ), \\ & \text{xSb} + ( \text{tempMv}[0] \gg 4 ) ) \end{aligned} \quad (8-588)$$

$$\begin{aligned} \text{yCbSb} = & \text{Clip3}( \text{yCtb}, \\ & \text{Min}( \text{CurPicHeightInSamplesY} - 1, \text{yCtb} + ( 1 \ll \text{CtbLog2SizeY} ) - 1 ), \\ & \text{ySb} + ( \text{tempMv}[1] \gg 4 ) ) \end{aligned} \quad (8-589)$$
- The variable currCb specifies the luma coding block covering the current coding subblock inside the current picture.
- The variable colCb specifies the luma coding block covering the modified location given by ( (xCbSb >> 3) << 3, (yCbSb >> 3) << 3 ) inside the ColPic.
- The luma location (xCbCb, yCbCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.
- The gbiIdxSbCol[ xSbIdx ][ ySbIdx ] is set equal to gbiIdxcolCb.

[217]

- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, ( xColCb, yColCb ), refIdxL0 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL0SbCol[ xSbIdx ][ ySbIdx ] and availableFlagL0SbCol.
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, ( xColCb, yColCb ), refIdxL1 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL1SbCol[ xSbIdx ][ ySbIdx ] and availableFlagL1SbCol.
- When availableFlagL0SbCol and availableFlagL1SbCol are both equal to 0, the following applies for X being 0 and 1:
 
$$\text{mvLXSbCol}[ \text{xSbIdx} ][ \text{ySbIdx} ] = \text{ctrMvLX} \quad (8-590)$$

$$\text{predFlagLXSbCol}[ \text{xSbIdx} ][ \text{ySbIdx} ] = \text{ctrPredFlagLX} \quad (8-591)$$

$$\text{gbIdxSbCol}[ \text{xSbIdx} ][ \text{ySbIdx} ] = \text{ctrgbIdx} \quad (\text{x-xxx})$$

[218] [ 10 ]

#### 8.4.4.4 Derivation process for subblock-based temporal merging base motion data

Inputs to this process are:

- the location ( xCtb, yCtb ) of the top-left sample of the luma coding tree block that contains the current coding block,
- the location ( xColCtrCb, yColCtrCb ) of the top-left sample of the collocated luma coding block that covers the below-right center sample.
- the availability flags availableFlagA<sub>0</sub>, availableFlagA<sub>1</sub>, availableFlagB<sub>0</sub>, and availableFlagB<sub>1</sub> of the neighbouring coding units,
- the reference indices refIdxLXA<sub>0</sub>, refIdxLXA<sub>1</sub>, refIdxLXB<sub>0</sub>, and refIdxLXB<sub>1</sub> of the neighbouring coding units,
- the prediction list utilization flags predFlagLXA<sub>0</sub>, predFlagLXA<sub>1</sub>, predFlagLXB<sub>0</sub>, and predFlagLXB<sub>1</sub> of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy mvLXA<sub>0</sub>, mvLXA<sub>1</sub>, mvLXB<sub>0</sub>, and mvLXB<sub>1</sub> of the neighbouring coding units.

Outputs of this process are:

- the motion vectors ctrMvL0 and ctrMvL1,
- the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1,
- the reference indices ctrRefIdxL0 and ctrRefIdxL1,
- the temporal motion vector tempMV,
- the bi-prediction weight index ctrgbIdx.

The variable tempMv is set as follows:

$$\text{tempMv}[ 0 ] = 0 \quad (8-592)$$

$$\text{tempMv}[ 1 ] = 0 \quad (8-593)$$

The variable currPic specifies the current picture.

The variable availableFlagN is set equal to FALSE, and the following applies:

- When availableFlagA<sub>1</sub> is equal to 1, the following applies:
  - availableFlagN is set equal to TRUE,

[219]

- refIdxLXN is set equal to refIdxLXA<sub>0</sub> and mvLXN is set equal to mvLXA<sub>0</sub>, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagLB<sub>1</sub> is equal to 1, the following applies:
  - availableFlagN is set equal to TRUE,
  - refIdxLXN is set equal to refIdxLXB<sub>0</sub> and mvLXN is set equal to mvLXB<sub>0</sub>, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagB<sub>0</sub> is equal to 1, the following applies:
  - availableFlagN is set equal to TRUE.
  - refIdxLXN is set equal to refIdxLXB<sub>1</sub> and mvLXN is set equal to mvLXB<sub>1</sub>, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagA<sub>0</sub> is equal to 1, the following applies:
  - availableFlagN is set equal to TRUE.
  - refIdxLXN is set equal to refIdxLXA<sub>1</sub> and mvLXN is set equal to mvLXA<sub>1</sub>, for X being replaced by 0 and 1.

When availableFlagN is equal to TRUE, the following applies:

- If all of the following conditions are true, tempMV is set equal to mvL1N:
  - predFlagL1N is equal to 1,
  - DiffPicOrderCnt(ColPic, RefPicList1[refIdxL1N]) is equal to 0,
  - DiffPicOrderCnt(aPic, currPic) is less than or equal to 0 for every picture aPic in every reference picture list of the current tile group,
  - tile\_group\_type is equal to B,
  - collocated\_from\_l0\_flag is equal to 0.

[220]

- Otherwise if all of the following conditions are true, tempMV is set equal to mvL0N:
  - predFlagL0N is equal to 1,
  - DiffPicOrderCnt(ColPic, RefPicList0[refIdxL0N]) is equal to 0.

The location ( xColCb, yColCb ) of the collocated block inside ColPic is derived as follows.

```

xColCb = Clip3(xCtb,
 Min(CurPicWidthInSamplesY - 1, xCtb + (1 << CtbLog2SizeY) + 3), (8-59
4)
 xColCtCb + (tempMv[0] >> 4))

yColCb = Clip3(yCtb,
 Min(CurPicHeightInSamplesY - 1, yCtb + (1 << CtbLog2SizeY) - 1), (8-5
95)
 yColCtCb + (tempMv[1] >> 4))

```

The array colPredMode is set equal to the prediction mode array CuPredMode of the collocated picture specified by ColPic.

The motion vectors ctrMvL0 and ctrMvL1, the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1, and the reference indices ctrRefIdxL0 and ctrRefIdxL1 are derived as follows:

- If colPredMode[xColCb][yColCb] is equal to MODE\_INTER, the following applies:
  - The variable currCb specifies the luma coding block covering ( xCtCb, yCtCb ) inside the current picture.
  - The variable colCb specifies the luma coding block covering the modified location given by ( ( xColCb >> 3 ) << 3, ( yColCb >> 3 ) << 3 ) inside the ColPic.

[221]

- The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.
- The gbiIdxSbCol is set equal to ctrgbiIdx.
- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), centerRefIdxL0, and sbFlag set equal to 1 as inputs and the output being assigned to ctrMvL0 and ctrPredFlagL0.
- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), centerRefIdxL1, and sbFlag set equal to 1 as inputs and the output being assigned to ctrMvL1 and ctrPredFlagL1.
- Otherwise, the following applies:
 

$$\text{ctrPredFlagL0} = 0 \quad (8-596)$$

$$\text{ctrPredFlagL1} = 0 \quad (8-597)$$

[222]

상기 표 8, 표 9 및 표 10을 참조하면, gbiIdx는 쌍 예측 가중치 인덱스를 나타낼 수 있으며, gbiIdxSbCol은 서브블록 기반 시간적 머지 후보(예를 들어, 서브블록 기반 머지 후보 리스트 내의 시간적 움직임 벡터 후보)를 위한 쌍 예측 가중치 인덱스를 나타낼 수 있다. 서브블록 기반 시간적 머지를 위한 베이스 움직임 정보를 도출하는 절차(8.4.4.3)에서 상기 gbiIdxSbCol은 gbiIdxcolCb로 도출될 수 있다. 또는 조건(예를 들어, availableFlagL0SbCol 및 availableFlagL1SbCol가 모두 0인 경우)에 따라 서브블록 기반 시간적 머지를 위한 베이스 움직임 정보를 도출하는 절차(8.4.4.3)에서 상기 gbiIdxSbCol은 ctrgbiIdx로 도출될 수 있으며, 서브블록 기반 시간적 머지를 위한 베이스 움직임 정보를 도출하는 절차(8.4.4.4)에서 상기 ctrgbiIdx는 gbiIdxSbCol로 도출될 수 있다. 즉, 서브블록 기반 시간적 움직임 벡터 후보의 가중치 인덱스는 각각의 서브블록 단위의 가중치 인덱스로 도출될 수 있고, 서브블록이 이용 가능하지 않은 경우, 시간적 센터 블록의 가중치 인덱스로 도출될 수 있다. 예를 들어, 상기 시간적 센터 블록은 콜(col) 블록 또는 콜(col) 블록의 센터에 위치한 서브블록 또는 샘플을 나타낼 수 있으며, 구체적으로는 콜 블록의 중앙 4개의 서브블록 또는 샘플 중 우하측에 위치한 서브블록 또는 샘플을 나타낼 수 있다.

[223]

한편, 본 문서의 또 다른 실시예에 따르면, 머지 모드를 위한 움직임 벡터 후보 구성 시, 페어와이즈(pair-wise) 후보에 대한 가중치 인덱스 정보를 도출할 수 있다. 예를 들어, 머지 후보 리스트에 페어와이즈 후보가 포함될 수 있으며, 상기 페어와이즈 후보의 가중평균을 위한 가중치 인덱스 정보가 도출될 수 있다. 상기 페어와이즈 후보는 상기 머지 후보 리스트 내의 다른 머지 후보들을 기반으로 도출될 수 있으며, 상기 페어와이즈 후보가 쌍 예측을 사용하는 경우, 가중평균을 위한 가중치 인덱스를 도출할 수 있다. 즉, 인터 예측 타입이 쌍(bi) 예측인 경우 머지 후보 리스트 내의 페어와이즈 후보를 위한 가중치 인덱스 정보를 도출할 수 있다.

[224]

상기 페어와이즈 후보는 상기 머지 후보 리스트에 포함된 후보들 중 다른 2개의

- 머지 후보들(예를 들어, cand0 및 cand1)을 기반으로 도출될 수 있다.
- [225] 예를 들어, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 2개의 머지 후보들 중 어느 하나의 머지 후보(예를 들어, 머지 후보 cand0 또는 머지 후보 cand1)의 가중치 인덱스 정보에 기반하여 도출될 수 있다. 예를 들어, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 2개의 머지 후보들 중 쌍 예측을 사용하는 후보의 가중치 인덱스 정보를 기반으로 도출될 수 있다.
- [226] 또는, 상기 다른 2개의 머지 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 동일한 경우, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 도출될 수 있다. 한편, 상기 다른 2개의 머지 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않은 경우, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 디폴트(DEFAULT) 가중치 인덱스 정보를 기반으로 도출될 수 있다. 디폴트 가중치 인덱스 정보는 L0 예측 샘플들 및 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당할 수 있다.
- [227] 또는, 상기 다른 2개의 머지 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 동일한 경우, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 도출될 수 있다. 한편, 상기 다른 2개의 머지 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않은 경우, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 다른 2개의 후보 각각의 가중치 인덱스 정보 중 디폴트(DEFAULT) 가중치 인덱스 정보가 아닌 가중치 인덱스 정보를 기반으로 도출될 수 있다. 디폴트 가중치 인덱스 정보는 L0 예측 샘플들 및 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당할 수 있다.
- [228] 한편, 본 문서의 또 다른 실시예에 따르면, 서브블록 단위의 머지 모드를 위한 움직임 벡터 후보 구성 시, 시간적 움직임 벡터 후보에 대한 가중평균을 위한 가중치 인덱스 정보를 도출할 수 있다. 여기서, 서브블록 단위의 머지 모드는 (서브블록 단위의) 어파인 머지 모드라고 불릴 수 있다. 시간적 움직임 벡터 후보는 서브블록 기반 시간적 움직임 벡터 후보를 나타낼 수 있으며, SbTMVP(또는 ATMVP) 후보라고 불릴 수도 있다. 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 현재 블록의 좌측 주변 블록의 가중치 인덱스 정보에 기반하여 도출될 수 있다. 즉, SbTMVP로 유도된 후보가 쌍 예측을 사용하는 경우, 현재 블록의 좌측 주변 블록의 가중치 인덱스를 서브블록 기반 머지 모드를 위한 가중치 인덱스로 도출할 수 있다.
- [229] 예를 들어, SbTMVP 후보는 현재 블록의 공간적으로 인접한 좌측 블록(또는 좌측 주변 블록)을 기준으로 콜(col) 블록을 도출할 수 있으므로, 상기 좌측 주변 블록의 가중치 인덱스는 신뢰받을 수 있다고 볼 수 있다. 이에 따라, 상기 SbTMVP 후보에 대한 가중치 인덱스는 좌측 주변 블록의 가중치 인덱스로 도출될 수 있다.

- [230] 한편, 본 문서의 또 다른 실시예에 따르면, 어파인 머지 모드를 위한 움직임 벡터 후보 구성 시, 어파인 머지 후보가 쌍 예측을 사용할 때 가중평균을 위한 가중치 인덱스 정보를 도출할 수 있다. 즉, 인터 예측 타입이 쌍(bi) 예측인 경우 어파인 머지 후보 리스트 또는 서브블록 머지 후보 리스트 내의 후보에 대한 가중치 인덱스 정보를 도출할 수 있다.
- [231] 예를 들어, 어파인 머지 후보 중 구성된(constructed) 어파인 머지 후보는 현재 블록의 공간적(spatial)으로 인접한 블록(또는 공간적 주변 블록) 또는 시간적(temporal)으로 인접한 블록(또는 시간적 주변 블록)의 움직임 정보를 기반으로 CP0, CP1, CP2 또는 CP3 후보를 도출하여 어파인 모델로 MVF를 도출하는 후보를 나타낼 수 있다. 예를 들어, CP0는 현재 블록의 좌상단 샘플 포지션에서 위치하는 컨트롤 포인트를 나타낼 수 있고, CP1은 현재 블록의 우상단 샘플 포지션에 위치하는 컨트롤 포인트를 나타낼 수 있고, CP2는 현재 블록의 좌하단 샘플 포지션에 위치하는 컨트롤 포인트를 나타낼 수 있다. 또한, CP3는 현재 블록의 우하단 샘플 포지션에 위치하는 컨트롤 포인트를 나타낼 수 있다.
- [232] 예를 들어, 어파인 머지 후보 중 구성된(constructed) 어파인 머지 후보는 {CP0, CP1, CP2}, {CP0, CP1, CP3}, {CP0, CP2, CP3}, {CP1, CP2, CP3}, {CP0, CP1} 및 {CP0, CP2}와 같이 현재 블록의 각 컨트롤 포인트의 조합에 기초하여 생성될 수 있다. 예를 들어, 어파인 머지 후보는 {CPMV0, CPMV1, CPMV2}, {CPMV0, CPMV1, CPMV3}, {CPMV0, CPMV2, CPMV3}, {CPMV1, CPMV2, CPMV3}, {CPMV0, CPMV1} 및 {CPMV0, CPMV2} 중 적어도 하나를 포함할 수 있다. CPMV0, CPMV1, CPMV2 및 CPMV3은 CP0, CP1, CP2 및 CP3 각각에 대한 움직임 벡터에 해당할 수 있다.
- [233] 일 실시예에서, 상기 어파인 머지 후보가 현재 블록의 좌상측에 위치하는 CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 제 0 가중치 인덱스 정보는 상기 CPO의 주변 블록들 중 상기 CPMV0의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에 해당할 수 있다. 이 때, 상기 CP0의 주변 블록들은 현재 블록의 좌상측 코너 주변 블록, 좌상측 코너 주변 블록의 하측에 인접한 좌측 주변 블록 및 좌상측 코너 주변 블록의 우측에 인접한 상측 주변 블록을 포함할 수 있다.
- [234] 한편, 상기 어파인 머지 후보가 현재 블록의 좌상측에 위치하는 CP0에 대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 현재 블록의 우상측에 위치하는 CP1(Control Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 제 1 가중치 인덱스 정보는 상기 CP1의 주변 블록들 중 상기 CPMV1의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에 해당할 수 있다. 이 때, 상기 CP1의 주변 블록들은 현재 블록의 우상측 코너 주변 블록 및 우상측 코너 주변 블록의 좌측에 인접한 상측 주변

블록을 포함할 수 있다.

- [235] 상술한 방법에 따르면, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 {CPMV0, CPMV1, CPMV2}, {CPMV0, CPMV1, CPMV3}, {CPMV0, CPMV2, CPMV3}, {CPMV1, CPMV2, CPMV3}, {CPMV0, CPMV1} 및 {CPMV0, CPMV2} 각각에 대하여 첫번째 CPMV의 도출을 위해 사용되는 블록의 가중치 인덱스 정보에 기반하여 도출될 수 있다.
- [236] 상기 어파인 머지 후보에 대한 가중치 인덱스 정보를 도출하는 다른 실시예에 따르면, 현재 블록의 좌상측에 위치하는 CP0에 대한 가중치 인덱스 정보와 현재 블록의 우상측에 위치하는 CP1에 대한 가중치 인덱스 정보가 동일한 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 제 0 가중치 인덱스 정보는 상기 CPO의 주변 블록들 중 상기 CPMV0의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에 해당할 수 있다. 한편, 현재 블록의 좌상측에 위치하는 CP0에 대한 가중치 인덱스 정보와 현재 블록의 우상측에 위치하는 CP1에 대한 가중치 인덱스 정보가 동일하지 않는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 디폴트(DEFAULT) 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 디폴트 가중치 인덱스 정보는 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당할 수 있다.
- [237] 상기 어파인 머지 후보에 대한 가중치 인덱스 정보를 도출하는 또 다른 실시예에 따르면, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 각 후보의 가중치 인덱스 중 발생 빈도가 높은 후보의 가중치 인덱스로 도출될 수 있다. 예를 들어, CP0 후보 블록들 중 CP0에서의 움직임 벡터로 결정된 후보 블록의 가중치 인덱스, CP1 후보 블록들 중 CP1에서의 움직임 벡터로 결정된 후보 블록의 가중치 인덱스, CP2 후보 블록들 중 CP2에서의 움직임 벡터로 결정된 후보 블록의 가중치 인덱스 및/또는 CP3 후보 블록들 중 CP3에서의 움직임 벡터로 결정된 후보 블록의 가중치 인덱스 중 가장 많이 중복되는 가중치 인덱스를 상기 어파인 머지 후보의 가중치 인덱스로 도출할 수 있다.
- [238] 예를 들어, 상기 컨트롤 포인트로 CP0 및 CP1가 이용될 수 있고, CP0, CP1 및 CP2가 이용될 수도 있으며, CP3는 이용되지 않을 수도 있다. 다만 예를 들어, 어파인 블록(어파인 예측 모드로 코딩되는 블록)의 CP3 후보를 활용하고자 할 때에는 상술한 실시예들에서 설명한 시간적 후보 블록에서의 가중치 인덱스를 도출하는 방법이 활용될 수 있다.
- [239] 도 12 및 13은 본 문서의 실시예(들)에 따른 비디오/영상 인코딩 방법 및 관련 컴포넌트의 일 예를 개략적으로 나타낸다.
- [240] 도 12에서 개시된 방법은 도 2 또는 도 13에서 개시된 인코딩 장치에 의하여 수행될 수 있다. 구체적으로 예를 들어, 도 12의 S1200 내지 S1220은 도 13의 상기 인코딩 장치(200)의 예측부(220)에 의하여 수행될 수 있고, 도 12의 S1230은 도 13의 상기 인코딩 장치(200)의 엔트로피 인코딩부(240)에 의하여 수행될 수 있다.

또한, 도 12에서 도시하지 않았으나, 도 12에서 상기 인코딩 장치(200)의 예측부(220)에 의하여 예측 샘플들 또는 예측 관련 정보를 도출할 수 있고, 상기 인코딩 장치(200)의 레지듀얼 처리부(230)에 의하여 원본 샘플들 또는 예측 샘플들로부터 레지듀얼 정보가 도출될 수 있고, 상기 인코딩 장치(200)의 엔트로피 인코딩부(240)에 의하여 레지듀얼 정보 또는 예측 관련 정보로부터 비트스트림이 생성될 수 있다. 도 12에서 개시된 방법은 본 문서에서 상술한 실시예들을 포함할 수 있다.

[241] 도 12을 참조하면, 인코딩 장치는 현재 블록의 인터 예측 모드를 결정하고, 인터 예측 모드를 나타내는 인터 예측 모드 정보를 생성할 수 있다(S1200). 예를 들어, 인코딩 장치는 현재 블록에 적용할 인터 예측 모드로 머지 모드, 어파인(머지) 모드 또는 서브블록 머지 모드를 결정할 수 있고, 이를 나타내는 인터 예측 모드 정보를 생성할 수 있다.

[242] 인코딩 장치는 인터 예측 모드를 기반으로 현재 블록의 머지 후보 리스트를 생성할 수 있다(S1210). 예를 들어, 인코딩 장치는 결정한 인터 예측 모드에 따라 머지 후보 리스트를 생성할 수 있다. 여기서, 결정한 인터 예측 모드가 어파인 머지 모드 또는 서브블록 머지 모드인 경우, 상기 머지 후보 리스트는 어파인 머지 후보 리스트 또는 서브블록 머지 후보 리스트 등과 같이 불릴 수 있으나, 간략히 머지 후보 리스트라고 불릴 수도 있다.

[243] 예를 들어, 상기 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수가 될 때까지 머지 후보 리스트에 후보가 삽입될 수 있다. 여기서, 후보는 현재 블록의 움직임 정보(또는 움직임 벡터)를 도출하기 위한 후보 또는 후보 블록을 나타낼 수 있다. 예를 들어, 후보 블록은 현재 블록의 주변 블록에 대한 탐색을 통해 도출될 수 있다. 예를 들어, 주변 블록은 현재 블록의 공간적 주변 블록 및/또는 시간적 주변 블록을 포함할 수 있으며, 공간적 주변 블록이 우선적으로 탐색되어(공간적 머지) 후보가 도출될 수 있고, 이후 시간적 주변 블록이 탐색되어(시간적 머지) 후보로 도출될 수 있으며, 도출된 후보들은 상기 머지 후보 리스트에 삽입될 수 있다. 예를 들어, 상기 머지 후보 리스트는 상기 후보들을 삽입한 이후에도 상기 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수보다 적은 경우, 추가 후보를 삽입할 수 있다. 예를 들어, 추가 후보는 history based merge candidate(s), pair-wise average merge candidate(s), ATMVP, combined bi-predictive 머지 후보(현재 슬라이스/타일 그룹의 슬라이스/타일 그룹 타입이 B 타입인 경우) 및/또는 영벡터 머지 후보 중 적어도 하나를 포함할 수 있다.

[244] 또는 예를 들어, 어파인 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수가 될 때까지 어파인 머지 후보 리스트에 후보가 삽입될 수 있다. 여기서, 후보는 현재 블록의 CPMV(Control Point Motion Vector)를 포함할 수 있다. 또는 상기 후보는 상기 CPMV를 도출하기 위한 후보 또는 후보 블록을 나타낼 수도 있다. 상기 CPMV는 현재 블록의 CP(Control Point)에서의 움직임 벡터를 나타낼 수 있다. 예를 들어, CP는 2개, 3개 또는 4개일 수 있으며, 현재 블록의 좌상측(또는

좌상측 코너), 우상측(또는 우상측 코너), 좌하측(또는 좌하측 코너) 또는 우하측(또는 우하측 코너) 중 적어도 일부에 위치할 수 있으며, 각 위치마다 하나의 CP만이 존재할 수 있다.

- [245] 예를 들어, 후보는 현재 블록의 주변 블록(또는 현재 블록의 CP의 주변 블록)에 대한 탐색을 통해 도출될 수 있다. 예를 들어, 어파인 머지 후보 리스트는 승계된(inherited) 어파인 머지 후보, 구성된(constructed) 어파인 머지 후보 또는 제로(zero) 움직임 벡터 후보 중 적어도 하나를 포함할 수 있다. 예를 들어, 어파인 머지 후보 리스트는 우선 상기 승계된 어파인 머지 후보를 삽입할 수 있고, 이후 구성된 어파인 머지 후보를 삽입할 수 있다. 또한, 어파인 머지 후보 리스트에 구성된 어파인 머지 후보까지 삽입하였으나, 상기 어파인 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수보다 작은 경우, 나머지는 제로 움직임 벡터 후보로 채울 수 있다. 여기서, 제로 움직임 벡터 후보는 영벡터라고 불릴 수도 있다. 예를 들어, 어파인 머지 후보 리스트는 샘플 단위로 움직임 벡터가 도출되는 어파인 머지 모드에 따른 리스트일 수 있으나, 서브블록 단위로 움직임 벡터가 도출되는 어파인 머지 모드에 따른 리스트일 수도 있다. 이 경우, 어파인 머지 후보 리스트는 서브블록 머지 후보 리스트라고 불릴 수도 있으며, 서브블록 머지 후보 리스트는 SbTMVP로 도출된 후보(또는 SbTMVP 후보)도 포함될 수 있다. 예를 들어, SbTMVP 후보가 서브블록 머지 후보 리스트에 포함되는 경우, 서브블록 머지 후보 리스트 내에서 승계된 어파인 머지 후보 및 구성된 어파인 머지 후보보다 앞에 위치할 수 있다.
- [246] 인코딩 장치는 머지 후보 리스트에 포함된 후보들 중 하나의 후보를 나타내는 선택 정보를 생성할 수 있다(S1220). 예를 들어, 머지 후보 리스트는 공간적 머지 후보, 시간적 머지 후보, 페어와이즈 후보 또는 영 벡터 후보 중 적어도 일부를 포함할 수 있고, 현재 블록의 인터 예측을 위해 이러한 후보들 중 하나의 후보를 선택할 수 있다. 또는 예를 들어, 서브블록 머지 후보 리스트는 승계된 어파인 머지 후보, 구성된 어파인 머지 후보, SbTMVP 후보 또는 영 벡터 후보 중 적어도 일부를 포함할 수 있고, 현재 블록의 인터 예측을 위해 이러한 후보들 중 하나의 후보를 선택할 수 있다.
- [247] 예를 들어, 상기 선택 정보는 상기 머지 후보 리스트 내의 선택된 후보를 나타내는 인덱스 정보를 포함할 수 있다. 예를 들어, 상기 선택 정보는 머지 인덱스 정보 또는 서브블록 머지 인덱스 정보라고 불릴 수도 있다.
- [248] 또한, 인코딩 장치는 현재 블록의 인터 예측 타입을 쌍 예측(bi-prediction)으로 나타내는 인터 예측 타입 정보를 생성할 수 있다. 예를 들어, 상기 현재 블록의 인터 예측 타입은 L0 예측, L1 예측 또는 쌍(bi) 예측 중 쌍 예측으로 결정될 수 있으며, 이를 나타내는 인터 예측 타입 정보를 생성할 수 있다. 여기서, L0 예측은 참조 픽처 리스트 0 기반의 예측을 나타낼 수 있고, L1 예측은 참조 픽처 리스트 1 기반의 예측을 나타낼 수 있으며, 쌍 예측은 참조 픽처 리스트 0 및 참조 픽처 리스트 1 기반의 예측을 나타낼 수 있다. 예를 들어, 인코딩 장치는 상기 인터

예측 타입을 기반으로 인터 예측 타입 정보를 생성할 수 있다. 예를 들어, 상기 인터 예측 타입 정보는 `inter_pred_idc` 신택스 요소를 포함할 수 있다.

[249] 인코딩 장치는 인터 예측 모드 정보 및 선택 정보를 포함하는 영상 정보를 인코딩할 수 있다(S1230). 예를 들어, 상기 영상 정보는 비디오 정보라고 불릴 수도 있다. 상기 영상 정보는 본 문서의 상술한 실시예(들)에 따른 다양한 정보를 포함할 수 있다. 예를 들어, 영상 정보는 예측 관련 정보 또는 레지듀얼 관련 정보 중 적어도 일부를 포함할 수 있다. 예를 들어, 상기 예측 관련 정보는 상기 인터 예측 모드 정보, 선택 정보 및 인터 예측 타입 정보 중 적어도 일부를 포함할 수 있다. 예를 들어, 인코딩 장치는 상술한 정보들(또는 신택스 요소들) 모두 또는 일부를 포함하는 영상 정보를 인코딩하여 비트스트림 또는 인코딩된 정보를 생성할 수 있다. 또는 비트스트림 형태로 출력할 수 있다. 또한, 상기 비트스트림 또는 인코딩된 정보는 네트워크 또는 저장매체를 통하여 디코딩 장치로 전송될 수 있다.

[250] 도 12에 도시하지 않았으나, 예를 들어, 인코딩 장치는 현재 블록의 예측 샘플들을 생성할 수 있다. 또는 예를 들어, 인코딩 장치는 선택된 후보를 기반으로 현재 블록의 예측 샘플들을 생성할 수 있다. 또는 예를 들어, 인코딩 장치는 선택된 후보를 기반으로 움직임 정보를 도출할 수 있고, 움직임 정보를 기반으로 현재 블록의 예측 샘플들을 생성할 수 있다. 예를 들어, 인코딩 장치는 쌍 예측에 따라 L0 예측 샘플들 및 L1 예측 샘플들을 생성할 수 있고, 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들을 기반으로 현재 블록의 예측 샘플들을 생성할 수 있다. 이 경우, 쌍 예측을 위한 가중치 인덱스 정보(또는 가중치 정보)를 이용하여 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들로부터 현재 블록의 예측 샘플들을 생성할 수 있다. 여기서, 가중치 정보는 상기 가중치 인덱스 정보를 기반으로 나타내어질 수 있다.

[251] 다시 말해, 예를 들어 인코딩 장치는 선택된 후보를 기반으로 현재 블록의 L0 예측 샘플들 및 L1 예측 샘플들을 생성할 수 있다. 예를 들어, 현재 블록의 인터 예측 타입이 쌍 예측으로 결정된 경우, 현재 블록의 예측을 위해 참조 픽처 리스트 0 및 참조 픽처 리스트 1이 이용될 수 있다. 예를 들어, 상기 L0 예측 샘플들은 상기 참조 픽처 리스트 0을 기반으로 도출된 현재 블록의 예측 샘플들을 나타낼 수 있고, 상기 L1 예측 샘플들은 상기 참조 픽처 리스트 1을 기반으로 도출된 현재 블록의 예측 샘플들을 나타낼 수 있다.

[252] 예를 들어, 상기 후보들은 공간적 머지 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 공간적 머지 후보인 경우, 상기 공간적 머지 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다.

[253] 예를 들어, 상기 후보들은 시간적 머지 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 시간적 머지 후보인 경우, 상기 시간적 머지 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로

상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다.

- [254] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 페어와이즈 후보인 경우, 상기 페어와이즈 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다. 예를 들어, 상기 페어와이즈 후보는 상기 머지 후보 리스트에 포함된 후보들 중 다른 2개의 후보를 기반으로 도출될 수 있다.
- [255] 또는 예를 들어, 상기 머지 후보 리스트는 서브블록 머지 후보 리스트일 수 있으며, 어파인 머지 후보, 서브블록 머지 후보 또는 SbTMVP 후보가 선택될 수도 있다. 여기서, 서브블록 단위의 어파인 머지 후보는 서브블록 머지 후보라고 불릴 수도 있다.
- [256] 예를 들어, 상기 후보들은 서브블록 머지 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 서브블록 머지 후보인 경우, 상기 서브블록 머지 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다. 예를 들어, 상기 서브블록 머지 후보는 CPMV(Control Point Motion Vector)들을 포함할 수 있고, 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들은 상기 CPMV들을 기반으로 서브블록 단위로 예측이 수행되어 생성될 수 있다.
- [257] 여기서, CPMV는 상기 현재 블록의 CP(Control Point)의 주변 블록들 중 하나의 블록을 기반으로 나타내어질 수 있다. 예를 들어, CP는 2개, 3개 또는 4개일 수 있으며, 현재 블록의 좌상측(또는 좌상측 코너), 우상측(또는 우상측 코너), 좌하측(또는 좌하측 코너) 또는 우하측(또는 우하측 코너) 중 적어도 일부에 위치할 수 있으며, 각 위치마다 하나의 CP만이 존재할 수 있다.
- [258] 예를 들어, 상기 CP가 상기 현재 블록의 좌상측에 위치하는 CP0일 수 있다. 이 경우, 상기 주변 블록들은 상기 현재 블록의 좌상측 코너 주변 블록, 상기 좌상측 코너 주변 블록의 하측에 인접한 좌측 주변 블록 및 상기 좌상측 코너 주변 블록의 우측에 인접한 상측 주변 블록을 포함할 수 있다. 또는, 상기 주변 블록들은 도 10에서 A<sub>2</sub> 블록, B<sub>2</sub> 블록 또는 B<sub>3</sub> 블록을 포함할 수 있다.
- [259] 또는 예를 들어, 상기 CP가 상기 현재 블록의 우상측에 위치하는 CP1일 수 있다. 이 경우, 상기 주변 블록들은 상기 현재 블록의 우상측 코너 주변 블록 및 상기 우상측 코너 주변 블록의 좌측에 인접한 상측 주변 블록을 포함할 수 있다. 또는, 상기 주변 블록들은 도 10에서 B<sub>0</sub> 블록 또는 B<sub>1</sub> 블록을 포함할 수 있다.
- [260] 또는 예를 들어, 상기 CP가 상기 현재 블록의 좌하측에 위치하는 CP2일 수 있다. 이 경우, 상기 주변 블록들은 상기 현재 블록의 좌하측 코너 주변 블록 및 상기 좌하측 코너 주변 블록의 상측에 인접한 좌측 주변 블록을 포함할 수 있다. 또는, 상기 주변 블록들은 도 10에서 A<sub>0</sub> 블록 또는 A<sub>1</sub> 블록을 포함할 수 있다.
- [261] 또는 예를 들어, 상기 CP가 상기 현재 블록의 우하측에 위치하는 CP3일 수 있다. 여기서, CP3은 RB라고 불릴 수도 있다. 이 경우, 상기 주변 블록들은 상기

현재 블록의 콜(col) 블록 또는 상기 콜 블록의 우하측 코너 주변 블록을 포함할 수 있다. 여기서, 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함할 수 있다. 또는 상기 주변 블록은 도 10에서 T 블록을 포함할 수 있다.

- [262] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 SbTMVP 후보인 경우, 상기 현재 블록의 좌측 주변 블록을 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다. 예를 들어, 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들은 서브블록 단위로 예측이 수행되어 생성될 수 있다.
- [263] 예를 들어, L0 움직임 정보는 L0 참조 픽처 인덱스 및 L0 움직임 벡터 등을 포함할 수 있고, L1 움직임 정보는 L1 참조 픽처 인덱스 및 L1 움직임 벡터 등을 포함할 수 있다. L0 참조 픽처 인덱스는 참조 픽처 리스트 0에서 참조 픽처를 나타내는 정보를 포함할 수 있고, L1 참조 픽처 인덱스는 참조 픽처 리스트 1에서 참조 픽처를 나타내는 정보를 포함할 수 있다.
- [264] 예를 들어, 인코딩 장치는 L0 예측 샘플들, L1 예측 샘플들 및 가중치 정보를 기반으로 현재 블록의 예측 샘플들을 생성할 수 있다. 예를 들어, 상기 가중치 정보는 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 상기 가중치 인덱스 정보는 쌍 예측을 위한 가중치 인덱스 정보를 나타낼 수 있다. 예를 들어, 상기 가중치 정보는 L0 예측 샘플들 또는 L1 예측 샘플들의 가중평균을 위한 정보를 포함할 수 있다. 즉, 상기 가중치 인덱스 정보는 상기 가중평균에 이용된 가중치에 대한 인덱스 정보를 나타낼 수 있고, 상기 가중평균을 기반으로 예측 샘플들을 생성하는 절차에서 가중치 인덱스 정보를 생성할 수도 있다. 예를 들어, 가중치 인덱스 정보는 3개 또는 5개의 가중치들 중 어느 하나의 가중치를 나타내는 정보를 포함할 수 있다. 예를 들어, 상기 가중평균은 BCW(Bi-prediction with CU-level Weight) 또는 BWA(Bi-prediction with Weighted Average)에서의 가중평균을 나타낼 수 있다.
- [265] 예를 들어, 상기 후보들은 시간적 머지 후보를 포함할 수 있으며, 상기 시간적 머지 후보에 대한 가중치 인덱스 정보는 0으로 나타내어질 수 있다. 즉, 시간적 머지 후보에 대한 가중치 인덱스 정보는 0으로 나타내어질 수 있다. 여기서, 0인 가중치 인덱스 정보는 각 참조 방향(즉, 쌍 예측에서 L0 예측 방향 및 L1 예측 방향)의 가중치가 동일함을 나타낼 수 있다. 또는 예를 들어, 상기 후보들은 시간적 머지 후보를 포함할 수 있으며, 상기 가중치 인덱스 정보는 콜(col) 블록의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 즉, 시간적 머지 후보를 위한 가중치 인덱스 정보는 콜(col) 블록의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 여기서, 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함할 수 있다.

- [266] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함할 수 있으며, 상기 가중치 인덱스 정보는 상기 페어와이즈 후보를 도출하는데 이용된 머지 후보 리스트 내의 다른 2개의 후보 중 하나의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 즉, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 페어와이즈 후보를 도출하는데 이용된 머지 후보 리스트 내의 다른 2개의 후보 중 하나의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다.
- [267] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함하고, 상기 페어와이즈 후보는 상기 후보들 중 다른 2개의 후보를 기반으로 나타내어질 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 서로 동일한 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않는 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 디폴트(DEFAULT) 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 이 때, 상기 디폴트 가중치 인덱스 정보는 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당할 수 있다.
- [268] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함하고, 상기 페어와이즈 후보는 상기 후보들 중 다른 2개의 후보를 기반으로 나타내어질 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 서로 동일한 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않는 경우, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 다른 2개의 후보 각각의 가중치 인덱스 정보 중 디폴트(DEFAULT) 가중치 인덱스 정보가 아닌 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 상기 디폴트 가중치 인덱스 정보는 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당할 수 있다.
- [269] 예를 들어, 상기 머지 후보 리스트는 서브블록 머지 후보 리스트일 수 있으며, 어파인 머지 후보, 서브블록 머지 후보 또는 SbTMVP 후보가 선택될 수도 있다. 여기서, 서브블록 단위의 어파인 머지 후보는 서브블록 머지 후보라고 불릴 수도 있다.
- [270] 예를 들어, 상기 후보들은 어파인 머지 후보를 포함할 수 있으며, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함할 수 있다.
- [271] 예를 들어, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에 대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지

후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 나타내어질 수 있다.

[272] 상기 제 0 가중치 인덱스 정보는 상기 CPO의 주변 블록들 중 상기 CPMV0의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에 해당하고, 상기 CP0의 주변 블록들은 상기 현재 블록의 좌상측 코너 주변 블록, 상기 좌상측 코너 주변 블록의 하측에 인접한 좌측 주변 블록 및 상기 좌상측 코너 주변 블록의 우측에 인접한 상측 주변 블록을 포함할 수 있다.

[273] 상기 제 1 가중치 인덱스 정보는 상기 CP1의 주변 블록들 중 상기 CPMV1의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에 해당하고, 상기 CP1의 주변 블록들은 상기 현재 블록의 우상측 코너 주변 블록 및 상기 우상측 코너 주변 블록의 좌측에 인접한 상측 주변 블록을 포함할 수 있다.

[274] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 좌측 주변 블록의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는 상기 좌측 주변 블록의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다.

[275] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 0으로 나타내어질 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는 0으로 나타내어질 수 있다. 여기서, 0인 가중치 인덱스 정보는 각 참조 방향(즉, 쌍 예측에서 L0 예측 방향 및 L1 예측 방향)의 가중치가 동일함을 나타낼 수 있다.

[276] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 가중치 인덱스 정보는 콜(col) 블록 내의 센터 블록의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는 콜(col) 블록 내의 센터 블록의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 여기서, 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함할 수 있고, 상기 센터 블록은 상기 콜 블록의 중앙에 위치하는 4개의 서브블록 중 우하측 서브블록을 포함할 수 있다.

[277] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 가중치 인덱스 정보는 콜(col) 블록의 서브블록들 각각의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는 콜(col) 블록의 서브블록들 각각의 가중치 인덱스 정보를 기반으로 나타내어질 수 있다.

[278] 또는 도 12에 도시하지 않았으나, 예를 들어, 인코딩 장치는 상기 예측 샘플들과 원본 샘플들을 기반으로 레지듀얼 샘플들을 도출할 수 있다. 이 경우, 상기 레지듀얼 샘플들을 기반으로 레지듀얼 관련 정보가 도출될 수 있다. 상기

레지듀얼 관련 정보를 기반으로 레지듀얼 샘플들을 도출될 수 있다. 상기 레지듀얼 샘플들 및 상기 예측 샘플들을 기반으로 복원 샘플들이 생성될 수 있다. 상기 복원 샘플들을 기반으로 복원 블록 및 복원 픽처가 도출될 수 있다. 또는 예를 들어, 인코딩 장치는 레지듀얼 관련 정보 또는 예측 관련 정보를 포함하는 영상 정보를 인코딩할 수 있다.

- [279] 예를 들어, 인코딩 장치는 상술한 정보들(또는 선택스 요소들) 모두 또는 일부를 포함하는 영상 정보를 인코딩하여 비트스트림 또는 인코딩된 정보를 생성할 수 있다. 또는 비트스트림 형태로 출력할 수 있다. 또한, 상기 비트스트림 또는 인코딩된 정보는 네트워크 또는 저장매체를 통하여 디코딩 장치로 전송될 수 있다. 또는, 상기 비트스트림 또는 인코딩된 정보는 컴퓨터 판독 가능한 저장 매체에 저장될 수 있으며, 상기 비트스트림 또는 상기 인코딩된 정보는 상술한 영상 인코딩 방법에 의해 생성될 수 있다.
- [280] 도 14 및 도 15는 본 문서의 실시예(들)에 따른 비디오/영상 디코딩 방법 및 관련 컴포넌트의 일 예를 개략적으로 나타낸다.
- [281] 도 14에서 개시된 방법은 도 3 또는 도 15에서 개시된 디코딩 장치에 의하여 수행될 수 있다. 구체적으로 예를 들어, 도 14의 S1400은 도 15에서 상기 디코딩 장치(300)의 엔트로피 디코딩부(310)에 의하여 수행될 수 있고, 도 14의 S1410 내지 S1440은 도 15에서 상기 디코딩 장치(300)의 예측부(330)에 의하여 수행될 수 있다. 또한, 도 14에서 도시하지 않았으나, 도 15에서 상기 디코딩 장치(300)의 엔트로피 디코딩부(310)에 의하여 비트스트림으로부터 예측 관련 정보 또는 레지듀얼 정보가 도출할 수 있고, 상기 디코딩 장치(300)의 레지듀얼 처리부(320)에 의하여 레지듀얼 정보로부터 레지듀얼 샘플들이 도출될 수 있고, 상기 디코딩 장치(300)의 예측부(330)에 의하여 예측 관련 정보로부터 예측 샘플들이 도출될 수 있고, 상기 디코딩 장치(300)의 가산부(340)에 의하여 레지듀얼 샘플들 또는 예측 샘플들로부터 복원 블록 또는 복원 픽처가 도출될 수 있다. 도 14에서 개시된 방법은 본 문서에서 상술한 실시예(들)을 포함할 수 있다.
- [282] 도 14를 참조하면, 디코딩 장치는 비트스트림을 통하여 인터 예측 모드 정보를 포함하는 영상 정보를 수신할 수 있다(S1400). 예를 들어, 상기 영상 정보는 비디오 정보라고 불릴 수도 있다. 상기 영상 정보는 본 문서의 상술한 실시예(들)에 따른 다양한 정보를 포함할 수 있다. 예를 들어, 영상 정보는 예측 관련 정보 또는 레지듀얼 관련 정보 중 적어도 일부를 포함할 수 있다.
- [283] 예를 들어, 상기 예측 관련 정보는 인터 예측 모드 정보 또는 인터 예측 타입 정보를 포함할 수 있다. 예를 들어, 상기 인터 예측 모드 정보는 다양한 인터 예측 모드 중 적어도 일부를 나타내는 정보를 포함할 수 있다. 예를 들어, 머지 모드, 스킵 모드, MVP(motion vector prediction) 모드, 어파인(Affine) 모드, 서브블록 머지 모드 또는 MMVD(merge with MVD) 모드 등 다양한 모드가 사용될 수 있다. 또한, DMVR(Decoder side motion vector refinement) 모드, AMVR(adaptive motion vector resolution) 모드, BCW(Bi-prediction with CU-level weight) 또는

BDOF(Bi-directional optical flow) 등이 부수적인 모드로 더 또는 대신 사용될 수 있다. 예를 들어, 상기 인터 예측 타입 정보는 `inter_pred_idc` 선택스 요소를 포함할 수 있다. 또는 상기 인터 예측 타입 정보는 L0 예측, L1 예측 또는 쌍(bi) 예측 중 어느 하나를 나타내는 정보를 포함할 수 있다.

- [284] 디코딩 장치는 인터 예측 모드 정보를 기반으로 현재 블록의 머지 후보 리스트를 생성할 수 있다(S1410). 예를 들어, 디코딩 장치는 상기 인터 예측 모드 정보를 기반으로 현재 블록의 인터 예측 모드를 머지 모드, 어파인 (머지) 모드 또는 서브블록 머지 모드로 결정할 수 있고, 결정한 인터 예측 모드에 따라 머지 후보 리스트를 생성할 수 있다. 여기서, 인터 예측 모드가 어파인 머지 모드 또는 서브블록 머지 모드로 결정된 경우, 상기 머지 후보 리스트는 어파인 머지 후보 리스트 또는 서브블록 머지 후보 리스트 등과 같이 불릴 수 있으나, 간략히 머지 후보 리스트라고 불릴 수도 있다.
- [285] 예를 들어, 상기 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수가 될 때까지 머지 후보 리스트에 후보가 삽입될 수 있다. 여기서, 후보는 현재 블록의 움직임 정보(또는 움직임 벡터)를 도출하기 위한 후보 또는 후보 블록을 나타낼 수 있다. 예를 들어, 후보 블록은 현재 블록의 주변 블록에 대한 탐색을 통해 도출될 수 있다. 예를 들어, 주변 블록은 현재 블록의 공간적 주변 블록 및/또는 시간적 주변 블록을 포함할 수 있으며, 공간적 주변 블록이 우선적으로 탐색되어 (공간적 머지) 후보가 도출될 수 있고, 이후 시간적 주변 블록이 탐색되어 (시간적 머지) 후보로 도출될 수 있으며, 도출된 후보들은 상기 머지 후보 리스트에 삽입될 수 있다. 예를 들어, 상기 머지 후보 리스트는 상기 후보들을 삽입한 이후에도 상기 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수보다 적은 경우, 추가 후보를 삽입할 수 있다. 예를 들어, 추가 후보는 history based merge candidate(s), pair-wise average merge candidate(s), ATMVP, combined bi-predictive 머지 후보 (현재 슬라이스/타일 그룹의 슬라이스/타일 그룹 타입이 B 타입인 경우) 및/또는 영벡터 머지 후보 중 적어도 하나를 포함할 수 있다.
- [286] 또는 예를 들어, 어파인 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수가 될 때까지 어파인 머지 후보 리스트에 후보가 삽입될 수 있다. 여기서, 후보는 현재 블록의 CPMV(Control Point Motion Vector)를 포함할 수 있다. 또는 상기 후보는 상기 CPMV를 도출하기 위한 후보 또는 후보 블록을 나타낼 수도 있다. 상기 CPMV는 현재 블록의 CP(Control Point)에서의 움직임 벡터를 나타낼 수 있다. 예를 들어, CP는 2개, 3개 또는 4개일 수 있으며, 현재 블록의 좌상측(또는 좌상측 코너), 우상측(또는 우상측 코너), 좌하측(또는 좌하측 코너) 또는 우하측(또는 우하측 코너) 중 적어도 일부에 위치할 수 있으며, 각 위치마다 하나의 CP만이 존재할 수 있다.
- [287] 예를 들어, 후보 블록은 현재 블록의 주변 블록(또는 현재 블록의 CP의 주변 블록)에 대한 탐색을 통해 도출될 수 있다. 예를 들어, 어파인 머지 후보 리스트는 승계된(inherited) 어파인 머지 후보, 구성된(constructed) 어파인 머지 후보 또는

제로(zero) 움직임 벡터 후보 중 적어도 하나를 포함할 수 있다. 예를 들어, 어파인 머지 후보 리스트는 우선 상기 승계된 어파인 머지 후보를 삽입할 수 있고, 이후 구성된 어파인 머지 후보를 삽입할 수 있다. 또한, 어파인 머지 후보 리스트에 구성된 어파인 머지 후보까지 삽입하였으나, 상기 어파인 머지 후보 리스트 내의 후보의 개수가 최대 후보 개수보다 작은 경우, 나머지는 제로 움직임 벡터 후보로 채울 수 있다. 여기서, 제로 움직임 벡터 후보는 영벡터라고 불릴 수도 있다. 예를 들어, 어파인 머지 후보 리스트는 샘플 단위로 움직임 벡터가 도출되는 어파인 머지 모드에 따른 리스트일 수 있으나, 서브블록 단위로 움직임 벡터가 도출되는 어파인 머지 모드에 따른 리스트일 수도 있다. 이 경우, 어파인 머지 후보 리스트는 서브블록 머지 후보 리스트라고 불릴 수도 있으며, 서브블록 머지 후보 리스트는 SbTMVP로 도출된 후보(또는 SbTMVP 후보)도 포함될 수 있다. 예를 들어, SbTMVP 후보가 서브블록 머지 후보 리스트에 포함되는 경우, 서브블록 머지 후보 리스트 내에서 승계된 어파인 머지 후보 및 구성된 어파인 머지 후보보다 앞에 위치할 수 있다.

[288] 디코딩 장치는 머지 후보 리스트 중 선택된 후보를 기반으로 현재 블록의 움직임 정보를 도출할 수 있다(S1420). 예를 들어, 머지 후보 리스트는 공간적 머지 후보, 시간적 머지 후보, 페어와이즈 후보 또는 영 벡터 후보 중 적어도 일부를 포함할 수 있고, 현재 블록의 인터 예측을 위해 이러한 후보들 중 하나의 후보를 선택할 수 있다. 또는 예를 들어, 서브블록 머지 후보 리스트는 승계된 어파인 머지 후보, 구성된 어파인 머지 후보, SbTMVP 후보 또는 영 벡터 후보 중 적어도 일부를 포함할 수 있고, 현재 블록의 인터 예측을 위해 이러한 후보들 중 하나의 후보를 선택할 수 있다. 예를 들어, 상기 선택된 후보는 선택 정보를 기반으로 상기 머지 후보 리스트로부터 선택될 수 있다. 예를 들어, 상기 선택 정보는 상기 머지 후보 리스트 내의 선택된 후보를 나타내는 인덱스 정보를 포함할 수 있다. 예를 들어, 상기 선택 정보는 머지 인덱스 정보 또는 서브블록 머지 인덱스 정보라고 불릴 수도 있다. 예를 들어, 상기 선택 정보는 상기 영상 정보에 포함될 수 있다. 또는 상기 선택 정보는 상기 인터 예측 모드 정보에 포함될 수도 있다.

[289] 디코딩 장치는 움직임 정보를 기반으로 현재 블록의 L0 예측 샘플들 및 L1 예측 샘플들을 생성할 수 있다(S1430). 예를 들어, 디코딩 장치는 인터 예측 타입이 쌍 예측으로 도출되는 경우 선택된 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보를 도출할 수 있다. 디코딩 장치는 인터 예측 타입 정보를 기반으로 현재 블록의 인터 예측 타입을 쌍 예측(bi-prediction)으로 도출할 수 있다. 예를 들어, 상기 현재 블록의 인터 예측 타입은 상기 인터 예측 타입 정보를 기반으로 L0 예측, L1 예측 또는 쌍(bi) 예측 중 쌍 예측으로 도출될 수 있다. 여기서, L0 예측은 참조 픽처 리스트 0 기반의 예측을 나타낼 수 있고, L1 예측은 참조 픽처 리스트 1 기반의 예측을 나타낼 수 있으며, 쌍 예측은 참조 픽처 리스트 0 및 참조 픽처 리스트 1 기반의 예측을 나타낼 수 있다. 예를 들어, 상기 인터 예측 타입 정보는

inter\_pred\_idc 인덱스 요소를 포함할 수 있다.

- [290] 예를 들어, L0 움직임 정보는 L0 참조 픽처 인덱스 및 L0 움직임 벡터 등을 포함할 수 있고, L1 움직임 정보는 L1 참조 픽처 인덱스 및 L1 움직임 벡터 등을 포함할 수 있다. L0 참조 픽처 인덱스는 참조 픽처 리스트 0에서 참조 픽처를 나타내는 정보를 포함할 수 있고, L1 참조 픽처 인덱스는 참조 픽처 리스트 1에서 참조 픽처를 나타내는 정보를 포함할 수 있다.
- [291] 예를 들어, 상기 후보들은 공간적 머지 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 공간적 머지 후보인 경우, 상기 공간적 머지 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다.
- [292] 예를 들어, 상기 후보들은 시간적 머지 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 시간적 머지 후보인 경우, 상기 시간적 머지 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다.
- [293] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 페어와이즈 후보인 경우, 상기 페어와이즈 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다. 예를 들어, 상기 페어와이즈 후보는 상기 머지 후보 리스트에 포함된 후보들 중 다른 2개의 후보를 기반으로 도출될 수 있다.
- [294] 또는 예를 들어, 상기 머지 후보 리스트는 서브블록 머지 후보 리스트일 수 있으며, 어파인 머지 후보, 서브블록 머지 후보 또는 SbTMVP 후보가 선택될 수도 있다. 여기서, 서브블록 단위의 어파인 머지 후보는 서브블록 머지 후보라고 불릴 수도 있다.
- [295] 예를 들어, 상기 후보들은 어파인 머지 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 어파인 머지 후보인 경우, 상기 어파인 머지 후보를 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다. 예를 들어, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함할 수 있고, 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들은 상기 CPMV들을 기반으로 서브블록 단위로 예측이 수행되어 생성될 수 있다.
- [296] 여기서, CPMV는 상기 현재 블록의 CP(Control Point)의 주변 블록들 중 하나의 블록을 기반으로 도출될 수 있다. 예를 들어, CP는 2개, 3개 또는 4개일 수 있으며, 현재 블록의 좌상측(또는 좌상측 코너), 우상측(또는 우상측 코너), 좌하측(또는 좌하측 코너) 또는 우하측(또는 우하측 코너) 중 적어도 일부에 위치할 수 있으며, 각 위치마다 하나의 CP만이 존재할 수 있다.
- [297] 예를 들어, 상기 CP가 상기 현재 블록의 좌상측에 위치하는 CP0일 수 있다. 이 경우, 상기 주변 블록들은 상기 현재 블록의 좌상측 코너 주변 블록, 상기 좌상측

코너 주변 블록의 하측에 인접한 좌측 주변 블록 및 상기 좌상측 코너 주변 블록의 우측에 인접한 상측 주변 블록을 포함할 수 있다. 또는, 상기 주변 블록들은 도 10에서  $A_2$  블록,  $B_2$  블록 또는  $B_3$  블록을 포함할 수 있다.

[298] 또는 예를 들어, 상기 CP가 상기 현재 블록의 우상측에 위치하는 CP1일 수 있다. 이 경우, 상기 주변 블록들은 상기 현재 블록의 우상측 코너 주변 블록 및 상기 우상측 코너 주변 블록의 좌측에 인접한 상측 주변 블록을 포함할 수 있다. 또는, 상기 주변 블록들은 도 10에서  $B_0$  블록 또는  $B_1$  블록을 포함할 수 있다.

[299] 또는 예를 들어, 상기 CP가 상기 현재 블록의 좌하측에 위치하는 CP2일 수 있다. 이 경우, 상기 주변 블록들은 상기 현재 블록의 좌하측 코너 주변 블록 및 상기 좌하측 코너 주변 블록의 상측에 인접한 좌측 주변 블록을 포함할 수 있다. 또는, 상기 주변 블록들은 도 10에서  $A_0$  블록 또는  $A_1$  블록을 포함할 수 있다.

[300] 또는 예를 들어, 상기 CP가 상기 현재 블록의 우하측에 위치하는 CP3일 수 있다. 여기서, CP3은 RB라고 불릴 수도 있다. 이 경우, 상기 주변 블록들은 상기 현재 블록의 콜(col) 블록 또는 상기 콜 블록의 우하측 코너 주변 블록을 포함할 수 있다. 여기서, 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함할 수 있다. 또는 상기 주변 블록들은 도 10에서 T 블록을 포함할 수 있다.

[301] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있다. 예를 들어, 상기 선택된 후보가 상기 SbTMVP 후보인 경우, 상기 현재 블록의 좌측 주변 블록을 기반으로 L0 움직임 정보 및 L1 움직임 정보가 도출될 수 있고, 이를 기반으로 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들이 생성될 수 있다. 예를 들어, 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들은 서브블록 단위로 예측이 수행되어 생성될 수 있다.

[302] 디코딩 장치는 L0 예측 샘플들, L1 예측 샘플들 및 가중치 정보를 기반으로 현재 블록의 예측 샘플들을 생성할 수 있다(S1440). 예를 들어, 상기 가중치 정보는 머지 후보 리스트에 포함된 후보들 중 선택된 후보에 대한 가중치 인덱스 정보를 기반으로 도출될 수 있다. 예를 들어, 상기 가중치 정보는 L0 예측 샘플들 또는 L1 예측 샘플들의 가중평균을 위한 정보를 포함할 수 있다. 즉, 상기 가중치 인덱스 정보는 상기 가중평균에 이용된 가중치에 대한 인덱스 정보를 나타낼 수 있고, 상기 가중평균은 가중치 인덱스 정보를 기반으로 수행될 수 있다. 예를 들어, 가중치 인덱스 정보는 3개 또는 5개의 가중치들 중 어느 하나의 가중치를 나타내는 정보를 포함할 수 있다. 예를 들어, 상기 가중평균은 BCW(Bi-prediction with CU-level Weight) 또는 BWA(Bi-prediction with Weighted Average)에서의 가중평균을 나타낼 수 있다.

[303] 예를 들어, 상기 후보들은 시간적 머지 후보를 포함할 수 있으며, 상기 시간적 머지 후보에 대한 가중치 인덱스 정보는 0으로 도출될 수 있다. 즉, 시간적 머지 후보를 위한 가중치 인덱스 정보는 0으로 도출될 수 있다. 여기서, 0인 가중치 인덱스 정보는 각 참조 방향(즉, 쌍 예측에서 L0 예측 방향 및 L1 예측 방향)의

가중치가 동일함을 나타낼 수 있다.

- [304] 예를 들어, 상기 후보들은 시간적 머지 후보를 포함할 수 있으며, 상기 시간적 머지 후보에 대한 가중치 인덱스 정보는 콜(col) 블록의 가중치 인덱스 정보를 기반으로 도출될 수 있다. 즉, 시간적 머지 후보에 대한 가중치 인덱스 정보는 콜(col) 블록의 가중치 인덱스 정보를 기반으로 도출될 수 있다. 여기서, 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함할 수 있다.
- [305] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함할 수 있으며, 상기 가중치 인덱스 정보는 상기 페어와이즈 후보를 도출하는데 이용된 머지 후보 리스트 내의 다른 2개의 후보 중 하나의 가중치 인덱스 정보로 도출될 수 있다. 즉, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 페어와이즈 후보를 도출하는데 이용된 머지 후보 리스트 내의 다른 2개의 후보 중 하나의 가중치 인덱스 정보로 도출될 수 있다.
- [306] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함하고, 상기 페어와이즈 후보는 상기 후보들 중 다른 2개의 후보를 기반으로 도출될 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 서로 동일한 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않는 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 디폴트(DEFAULT) 가중치 인덱스 정보를 기반으로 도출될 수 있다. 이 때, 상기 디폴트 가중치 인덱스 정보는 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당할 수 있다.
- [307] 예를 들어, 상기 후보들은 페어와이즈(pair-wise) 후보를 포함하고, 상기 페어와이즈 후보는 상기 후보들 중 다른 2개의 후보를 기반으로 도출될 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 서로 동일한 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않는 경우, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 다른 2개의 후보 각각의 가중치 인덱스 정보 중 디폴트(DEFAULT) 가중치 인덱스 정보가 아닌 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 디폴트 가중치 인덱스 정보는 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당할 수 있다.
- [308] 예를 들어, 상기 머지 후보 리스트는 서브블록 머지 후보 리스트일 수 있으며, 어파인 머지 후보, 서브블록 머지 후보 또는 SbTMVP 후보가 선택될 수도 있다. 여기서, 서브블록 단위의 어파인 머지 후보는 서브블록 머지 후보라고 불릴 수도 있다.

- [309] 예를 들어, 상기 후보들은 어파인 머지 후보를 포함할 수 있으며, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함할 수 있다.
- [310] 예를 들어, 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 도출될 수 있다. 상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에 대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 도출될 수 있다.
- [311] 상기 제 0 가중치 인덱스 정보는 상기 CPO의 주변 블록들 중 상기 CPMV0의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에 해당하고, 상기 CP0의 주변 블록들은 상기 현재 블록의 좌상측 코너 주변 블록, 상기 좌상측 코너 주변 블록의 하측에 인접한 좌측 주변 블록 및 상기 좌상측 코너 주변 블록의 우측에 인접한 상측 주변 블록을 포함할 수 있다.
- [312] 상기 제 1 가중치 인덱스 정보는 상기 CP1의 주변 블록들 중 상기 CPMV1의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에 해당하고, 상기 CP1의 주변 블록들은 상기 현재 블록의 우상측 코너 주변 블록 및 상기 우상측 코너 주변 블록의 좌측에 인접한 상측 주변 블록을 포함할 수 있다.
- [313] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 좌측 주변 블록의 가중치 인덱스 정보를 기반으로 도출될 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는 상기 좌측 주변 블록의 가중치 인덱스 정보를 기반으로 도출될 수 있다.
- [314] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 0으로 도출될 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는 0으로 도출될 수 있다. 여기서, 0인 가중치 인덱스 정보는 각 참조 방향(즉, 쌍 예측에서 L0 예측 방향 및 L1 예측 방향)의 가중치가 동일함을 나타낼 수 있다.
- [315] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 가중치 인덱스 정보는 쿨(col) 블록 내의 센터 블록의 가중치 인덱스 정보를 기반으로 도출될 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는 쿨(col) 블록 내의 센터 블록의 가중치 인덱스 정보를 기반으로 도출될 수 있다. 여기서, 상기 쿨 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함할 수 있고, 상기 센터 블록은 상기 쿨 블록의 중앙에 위치하는 4개의 서브블록 중 우하측 서브블록을 포함할 수 있다.
- [316] 또는 예를 들어, 상기 후보들은 SbTMVP 후보를 포함할 수 있으며, 상기 가중치 인덱스 정보는 쿨(col) 블록의 서브블록들 각각의 가중치 인덱스 정보를 기반으로 도출될 수 있다. 즉, SbTMVP 후보를 위한 가중치 인덱스 정보는

콜(col) 블록의 서브블록들 각각의 가중치 인덱스 정보를 기반으로 도출될 수 있다.

- [317]    도 14에 도시하지 않았으나, 예를 들어 디코딩 장치는 상기 영상 정보에 포함된 레지듀얼 관련 정보를 기반으로 레지듀얼 샘플들을 도출할 수 있다. 또한, 디코딩 장치는 상기 예측 샘플들 및 상기 레지듀얼 샘플들을 기반으로 복원 샘플들을 생성할 수 있다. 상기 복원 샘플들을 기반으로 복원 블록 및 복원 픽처가 도출될 수 있다.
- [318]    예를 들어, 디코딩 장치는 비트스트림 또는 인코딩된 정보를 디코딩하여 상술한 정보들(또는 선택스 요소들) 모두 또는 일부를 포함하는 영상 정보를 획득할 수 있다. 또한, 상기 비트스트림 또는 인코딩된 정보는 컴퓨터 판독 가능한 저장 매체에 저장될 수 있으며, 상술한 디코딩 방법이 수행되도록 야기할 수 있다.
- [319]    상술한 실시예에서, 방법들은 일련의 단계 또는 블록으로써 순서도를 기초로 설명되고 있지만, 해당 실시예는 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 당업자라면 순서도에 나타내어진 단계들이 배타적이지 않고, 다른 단계가 포함되거나 순서도의 하나 또는 그 이상의 단계가 본 문서의 실시예들의 범위에 영향을 미치지 않고 삭제될 수 있음을 이해할 수 있을 것이다.
- [320]    상술한 본 문서의 실시예들에 따른 방법은 소프트웨어 형태로 구현될 수 있으며, 본 문서에 따른 인코딩 장치 및/또는 디코딩 장치는 예를 들어 TV, 컴퓨터, 스마트폰, 셋톱박스, 디스플레이 장치 등의 영상 처리를 수행하는 장치에 포함될 수 있다.
- [321]    본 문서에서 실시예들이 소프트웨어로 구현될 때, 상술한 방법은 상술한 기능을 수행하는 모듈(과정, 기능 등)로 구현될 수 있다. 모듈은 메모리에 저장되고, 프로세서에 의해 실행될 수 있다. 메모리는 프로세서 내부 또는 외부에 있을 수 있고, 잘 알려진 다양한 수단으로 프로세서와 연결될 수 있다. 프로세서는 ASIC(application-specific integrated circuit), 다른 칩셋, 논리 회로 및/또는 데이터 처리 장치를 포함할 수 있다. 메모리는 ROM(read-only memory), RAM(random access memory), 플래쉬 메모리, 메모리 카드, 저장 매체 및/또는 다른 저장 장치를 포함할 수 있다. 즉, 본 문서에서 설명한 실시예들은 프로세서, 마이크로 프로세서, 컨트롤러 또는 칩 상에서 구현되어 수행될 수 있다. 예를 들어, 각 도면에서 도시한 기능 유닛들은 컴퓨터, 프로세서, 마이크로 프로세서, 컨트롤러 또는 칩 상에서 구현되어 수행될 수 있다. 이 경우 구현을 위한 정보(ex. information on instructions) 또는 알고리즘이 디지털 저장 매체에 저장될 수 있다.
- [322]    또한, 본 문서의 실시예(들)이 적용되는 디코딩 장치 및 인코딩 장치는 멀티미디어 방송 송수신 장치, 모바일 통신 단말, 홈 시네마 비디오 장치, 디지털 시네마 비디오 장치, 감시용 카메라, 비디오 대화 장치, 비디오 통신과 같은 실시간 통신 장치, 모바일 스트리밍 장치, 저장 매체, 캠코더, 주문형

비디오(VoD) 서비스 제공 장치, OTT 비디오(Over the top video) 장치, 인터넷 스트리밍 서비스 제공 장치, 3차원(3D) 비디오 장치, VR(virtual reality) 장치, AR(argumente reality) 장치, 화상 전화 비디오 장치, 운송 수단 단말 (ex. 차량(자율주행차량 포함) 단말, 비행기 단말, 선박 단말 등) 및 의료용 비디오 장치 등에 포함될 수 있으며, 비디오 신호 또는 데이터 신호를 처리하기 위해 사용될 수 있다. 예를 들어, OTT 비디오(Over the top video) 장치로는 게임 콘솔, 블루레이 플레이어, 인터넷 접속 TV, 홈시어터 시스템, 스마트폰, 태블릿 PC, DVR(Digital Video Recorder) 등을 포함할 수 있다.

- [323] 또한, 본 문서의 실시예(들)이 적용되는 처리 방법은 컴퓨터로 실행되는 프로그램의 형태로 생산될 수 있으며, 컴퓨터가 판독할 수 있는 기록 매체에 저장될 수 있다. 본 문서의 실시예(들)에 따른 데이터 구조를 가지는 멀티미디어 데이터도 또한 컴퓨터가 판독할 수 있는 기록 매체에 저장될 수 있다. 상기 컴퓨터가 판독할 수 있는 기록 매체는 컴퓨터로 읽을 수 있는 데이터가 저장되는 모든 종류의 저장 장치 및 분산 저장 장치를 포함한다. 상기 컴퓨터가 판독할 수 있는 기록 매체는, 예를 들어, 블루레이 디스크(BD), 범용 직렬 버스(USB), ROM, PROM, EPROM, EEPROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크 및 광학적 데이터 저장 장치를 포함할 수 있다. 또한, 상기 컴퓨터가 판독할 수 있는 기록 매체는 반송파(예를 들어, 인터넷을 통한 전송)의 형태로 구현된 미디어를 포함한다. 또한, 인코딩 방법으로 생성된 비트스트림이 컴퓨터가 판독할 수 있는 기록 매체에 저장되거나 유무선 통신 네트워크를 통해 전송될 수 있다.
- [324] 또한, 본 문서의 실시예(들)는 프로그램 코드에 의한 컴퓨터 프로그램 제품으로 구현될 수 있고, 상기 프로그램 코드는 본 문서의 실시예(들)에 의해 컴퓨터에서 수행될 수 있다. 상기 프로그램 코드는 컴퓨터에 의해 판독가능한 캐리어 상에 저장될 수 있다.
- [325] 도 16은 본 문서에서 개시된 실시예들이 적용될 수 있는 콘텐츠 스트리밍 시스템의 예를 나타낸다.
- [326] 도 16을 참조하면, 본 문서의 실시예들이 적용되는 콘텐츠 스트리밍 시스템은 크게 인코딩 서버, 스트리밍 서버, 웹 서버, 미디어 저장소, 사용자 장치 및 멀티미디어 입력 장치를 포함할 수 있다.
- [327] 상기 인코딩 서버는 스마트폰, 카메라, 캠코더 등과 같은 멀티미디어 입력 장치들로부터 입력된 콘텐츠를 디지털 데이터로 압축하여 비트스트림을 생성하고 이를 상기 스트리밍 서버로 전송하는 역할을 한다. 다른 예로, 스마트폰, 카메라, 캠코더 등과 같은 멀티미디어 입력 장치들이 비트스트림을 직접 생성하는 경우, 상기 인코딩 서버는 생략될 수 있다.
- [328] 상기 비트스트림은 본 문서의 실시예들이 적용되는 인코딩 방법 또는 비트스트림 생성 방법에 의해 생성될 수 있고, 상기 스트리밍 서버는 상기 비트스트림을 전송 또는 수신하는 과정에서 일시적으로 상기 비트스트림을 저장할 수 있다.

- [329] 상기 스트리밍 서버는 웹 서버를 통한 사용자 요청에 기초하여 멀티미디어 데이터를 사용자 장치에 전송하고, 상기 웹 서버는 사용자에게 어떠한 서비스가 있는지를 알려주는 매개체 역할을 한다. 사용자가 상기 웹 서버에 원하는 서비스를 요청하면, 상기 웹 서버는 이를 스트리밍 서버에 전달하고, 상기 스트리밍 서버는 사용자에게 멀티미디어 데이터를 전송한다. 이때, 상기 콘텐츠 스트리밍 시스템은 별도의 제어 서버를 포함할 수 있고, 이 경우 상기 제어 서버는 상기 콘텐츠 스트리밍 시스템 내 각 장치 간 명령/응답을 제어하는 역할을 한다.
- [330] 상기 스트리밍 서버는 미디어 저장소 및/또는 인코딩 서버로부터 콘텐츠를 수신할 수 있다. 예를 들어, 상기 인코딩 서버로부터 콘텐츠를 수신하게 되는 경우, 상기 콘텐츠를 실시간으로 수신할 수 있다. 이 경우, 원활한 스트리밍 서비스를 제공하기 위하여 상기 스트리밍 서버는 상기 비트스트림을 일정 시간동안 저장할 수 있다.
- [331] 상기 사용자 장치의 예로는, 휴대폰, 스마트 폰(smart phone), 노트북 컴퓨터(laptop computer), 디지털방송용 단말기, PDA(personal digital assistants), PMP(portable multimedia player), 네비게이션, 슬레이트 PC(slate PC), 태블릿 PC(tablet PC), 울트라북(ultrabook), 웨어러블 디바이스(wearable device, 예를 들어, 위치형 단말기 (smartwatch), 글래스형 단말기 (smart glass), HMD(head mounted display)), 디지털 TV, 데스크탑 컴퓨터, 디지털 사이니지 등이 있을 수 있다.
- [332] 상기 콘텐츠 스트리밍 시스템 내 각 서버들은 분산 서버로 운영될 수 있으며, 이 경우 각 서버에서 수신하는 데이터는 분산 처리될 수 있다.
- [333] 본 명세서에 기재된 청구항들은 다양한 방식으로 조합될 수 있다. 예를 들어, 본 명세서의 방법 청구항의 기술적 특징이 조합되어 장치로 구현될 수 있고, 본 명세서의 장치 청구항의 기술적 특징이 조합되어 방법으로 구현될 수 있다. 또한, 본 명세서의 방법 청구항의 기술적 특징과 장치 청구항의 기술적 특징이 조합되어 장치로 구현될 수 있고, 본 명세서의 방법 청구항의 기술적 특징과 장치 청구항의 기술적 특징이 조합되어 방법으로 구현될 수 있다.

## 청구범위

- [청구항 1] 디코딩 장치에 의하여 수행되는 영상 디코딩 방법에 있어서,  
비트스트림을 통하여 인터 예측 모드 정보를 포함하는 영상 정보를  
수신하는 단계;  
상기 인터 예측 모드 정보를 기반으로 현재 블록의 머지 후보 리스트를  
생성하는 단계;  
상기 머지 후보 리스트 중 선택된 후보를 기반으로 상기 현재 블록의  
움직임 정보를 도출하는 단계;  
상기 움직임 정보를 기반으로 상기 현재 블록의 L0 예측 샘플들 및 L1  
예측 샘플들을 생성하는 단계; 및  
상기 L0 예측 샘플들, 상기 L1 예측 샘플들 및 가중치 정보를 기반으로  
상기 현재 블록의 예측 샘플들을 생성하고, 상기 가중치 정보는 상기  
선택된 후보에 대한 가중치 인덱스 정보를 기반으로 도출되는 단계를  
포함하고,  
상기 후보들은 어파인 머지 후보를 포함하고, 상기 어파인 머지 후보는  
CPMV(Control Point Motion Vector)들을 포함하고,  
상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는  
CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지  
후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스  
정보를 기반으로 도출되고,  
상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에  
대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한  
가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control  
Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 도출되는, 영상  
디코딩 방법.
- [청구항 2] 제 1 항에 있어서,  
상기 제 0 가중치 인덱스 정보는 상기 CP0의 주변 블록들 중 상기  
CPMV0의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에  
해당하고,  
상기 CP0의 주변 블록들은 상기 현재 블록의 좌상측 코너 주변 블록, 상기  
좌상측 코너 주변 블록의 하측에 인접한 좌측 주변 블록 및 상기 좌상측  
코너 주변 블록의 우측에 인접한 상측 주변 블록을 포함하는, 영상 디코딩  
방법.
- [청구항 3] 제 1 항에 있어서,  
상기 제 1 가중치 인덱스 정보는 상기 CP1의 주변 블록들 중 상기  
CPMV1의 도출을 위하여 사용되는 블록의 가중치 인덱스 정보에  
해당하고,

상기 CP1의 주변 블록들은 상기 현재 블록의 우상측 코너 주변 블록 및 상기 우상측 코너 주변 블록의 좌측에 인접한 상측 주변 블록을 포함하는, 영상 디코딩 방법.

[청구항 4] 제1항에 있어,  
상기 후보들은 페어와이즈(pair-wise) 후보를 포함하고, 상기 페어와이즈 후보는 상기 후보들 중 다른 2개의 후보를 기반으로 도출되고,  
상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 2개의 후보 중 하나의 가중치 인덱스 정보를 기반으로 도출되는, 영상 디코딩 방법.

[청구항 5] 제 1 항에 있어서,  
상기 후보들은 페어와이즈(pair-wise) 후보를 포함하고, 상기 페어와이즈 후보는 상기 후보들 중 다른 2개의 후보를 기반으로 도출되고,  
상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 서로 동일한 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 도출되고,  
상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않는 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 디폴트(DEFAULT) 가중치 인덱스 정보를 기반으로 도출되고,  
상기 디폴트 가중치 인덱스 정보는 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당하는, 영상 디코딩 방법.

[청구항 6] 제 1 항에 있어서,  
상기 후보들은 페어와이즈(pair-wise) 후보를 포함하고, 상기 페어와이즈 후보는 상기 후보들 중 다른 2개의 후보를 기반으로 도출되고,  
상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 제 1 가중치 인덱스 정보로 서로 동일한 경우 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 제 1 가중치 인덱스 정보를 기반으로 도출되고,  
상기 다른 2개의 후보 각각의 가중치 인덱스 정보가 서로 동일하지 않는 경우, 상기 페어와이즈 후보에 대한 가중치 인덱스 정보는 상기 다른 2개의 후보 각각의 가중치 인덱스 정보 중 디폴트(DEFAULT) 가중치 인덱스 정보가 아닌 가중치 인덱스 정보를 기반으로 도출되고,  
상기 디폴트 가중치 인덱스 정보는 상기 L0 예측 샘플들 및 상기 L1 예측 샘플들 각각에 동일한 가중치를 부여하는 가중치 인덱스 정보에 해당하는, 영상 디코딩 방법.

[청구항 7] 제1항에 있어서,  
상기 후보들은 SbTMVP(subblock-based temporal motion vector prediction) 후보를 포함하고,  
상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 좌측 주변 블록의 가중치 인덱스 정보를 기반으로 도출되는, 영상 디코딩 방법.

- [청구항 8] 제1항에 있어서,  
 상기 후보들은 SbTMVP(subblock-based temporal motion vector prediction) 후보를 포함하고,  
 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 0으로 도출되는, 영상 디코딩 방법.
- [청구항 9] 제1항에 있어서,  
 상기 후보들은 SbTMVP(subblock-based temporal motion vector prediction) 후보를 포함하고,  
 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 콜(col) 블록 내의 센터 블록의 가중치 인덱스 정보를 기반으로 도출되고,  
 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함하고,  
 상기 센터 블록은 상기 콜 블록의 중앙에 위치하는 4개의 서브블록 중 우하측 서브블록을 포함하는, 영상 디코딩 방법.
- [청구항 10] 제1항에 있어서,  
 상기 후보들은 SbTMVP(subblock-based temporal motion vector prediction) 후보를 포함하고,  
 상기 SbTMVP 후보에 대한 가중치 인덱스 정보는 콜(col) 블록의 서브블록들 각각의 가중치 인덱스 정보를 기반으로 도출되고,  
 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함하는, 영상 디코딩 방법.
- [청구항 11] 제1항에 있어서,  
 상기 후보들은 시간적 머지 후보를 포함하고,  
 상기 시간적 머지 후보에 대한 가중치 인덱스 정보는 0으로 도출되는, 영상 디코딩 방법.
- [청구항 12] 제1항에 있어서,  
 상기 후보들은 시간적 머지 후보를 포함하고,  
 상기 시간적 머지 후보에 대한 가중치 인덱스 정보는 콜(col) 블록의 가중치 인덱스 정보를 기반으로 도출되고,  
 상기 콜 블록은 상기 현재 블록이 위치하는 현재 픽처와 다른 참조 픽처 내에 상기 현재 블록과 동일한 위치의 블록을 포함하는, 영상 디코딩 방법.
- [청구항 13] 인코딩 장치에 의하여 수행되는 영상 인코딩 방법에 있어서,  
 현재 블록의 인터 예측 모드를 결정하고, 상기 인터 예측 모드를 나타내는 인터 예측 모드 정보를 생성하는 단계;  
 상기 인터 예측 모드를 기반으로 상기 현재 블록의 머지 후보 리스트를 생성하는 단계;

상기 머지 후보 리스트에 포함된 후보들 중 하나의 후보를 나타내는 선택 정보를 생성하는 단계; 및

상기 인터 예측 모드 정보 및 상기 선택 정보를 포함하는 영상 정보를 인코딩하는 단계를 포함하고,

상기 후보들은 어파인 머지 후보를 포함하고, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함하고,

상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는

CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 나타내어지고,

상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에 대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 나타내어지는, 영상 인코딩 방법.

[청구항 14] 영상 디코딩 장치가 영상 디코딩 방법을 수행하도록 야기하는 인코딩된 정보를 저장하는 컴퓨터 판독 가능한 저장 매체에 있어서, 상기 영상 디코딩 방법은:

비트스트림을 통하여 인터 예측 모드 정보를 포함하는 영상 정보를 획득하는 단계;

상기 인터 예측 모드 정보를 기반으로 현재 블록의 머지 후보 리스트를 생성하는 단계;

상기 머지 후보 리스트에 포함된 후보들 중 하나의 후보를 선택하는 단계;

상기 선택된 후보를 기반으로 상기 현재 블록의 움직임 정보를 도출하는 단계;

상기 움직임 정보를 기반으로 상기 현재 블록의 L0 예측 샘플들 및 L1 예측 샘플들을 생성하는 단계; 및

상기 L0 예측 샘플들, 상기 L1 예측 샘플들 및 가중치 정보를 기반으로 상기 현재 블록의 예측 샘플들을 생성하고, 상기 가중치 정보는 상기 선택된 후보에 대한 가중치 인덱스 정보를 기반으로 도출되는 단계를 포함하고,

상기 후보들은 어파인 머지 후보를 포함하고, 상기 어파인 머지 후보는 CPMV(Control Point Motion Vector)들을 포함하고,

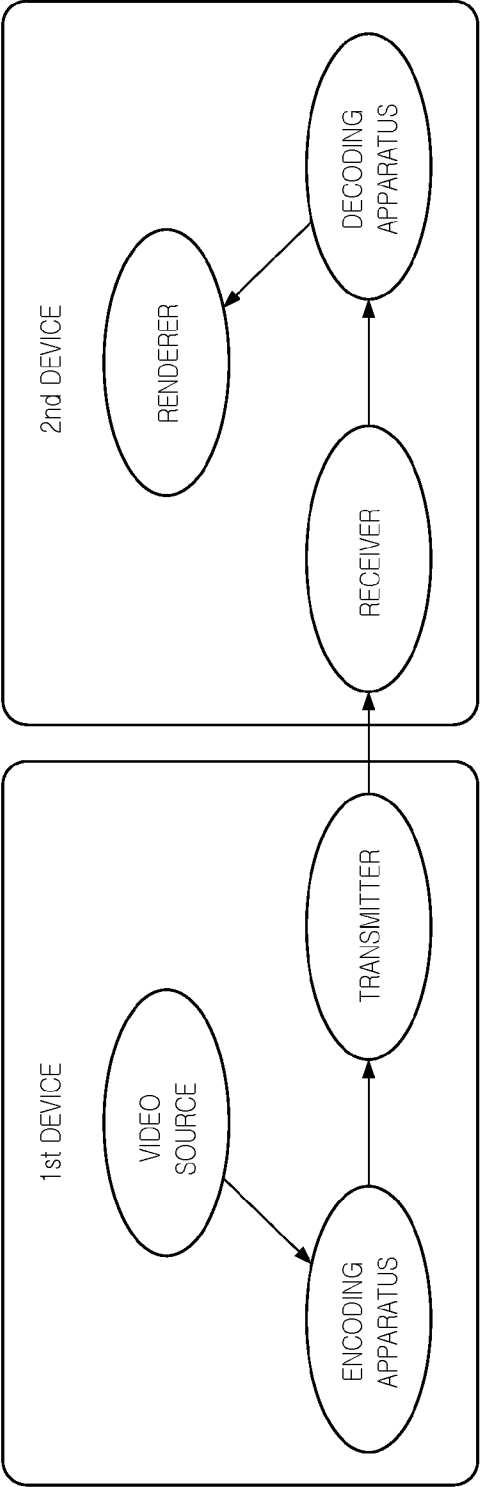
상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는

CP0(Control Point 0)에 대한 CPMV0를 포함하는 경우, 상기 어파인 머지 후보에 대한 가중치 인덱스 정보는 상기 CP0에 대한 제 0 가중치 인덱스 정보를 기반으로 도출되고,

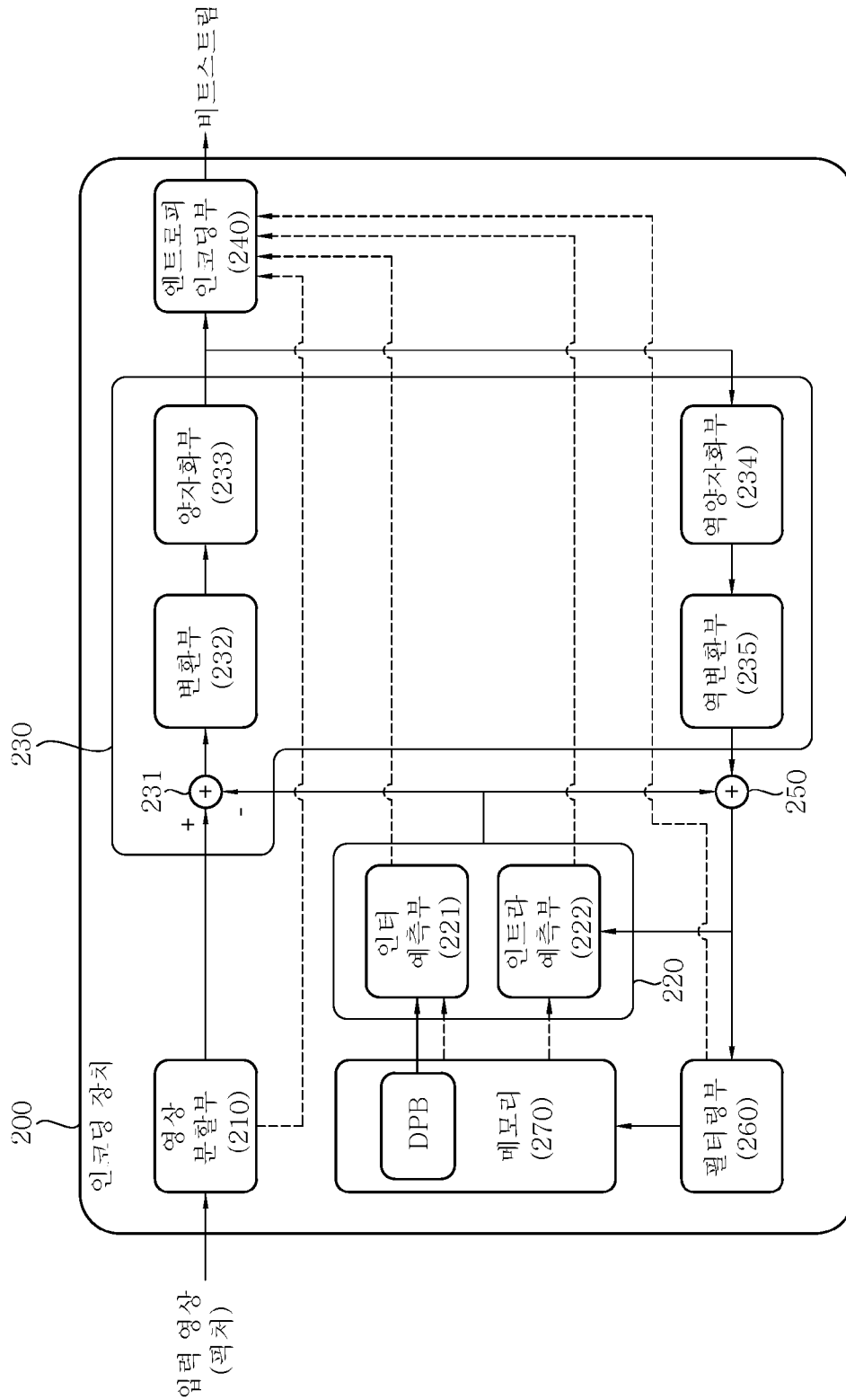
상기 어파인 머지 후보가 상기 현재 블록의 좌상측에 위치하는 CP0에

대한 CPMV0를 포함하지 않는 경우, 상기 어파인 머지 후보에 대한  
가중치 인덱스 정보는 상기 현재 블록의 우상측에 위치하는 CP1(Control  
Point 1)에 대한 제 1 가중치 인덱스 정보를 기반으로 도출되는, 컴퓨터  
판독 가능한 저장 매체.

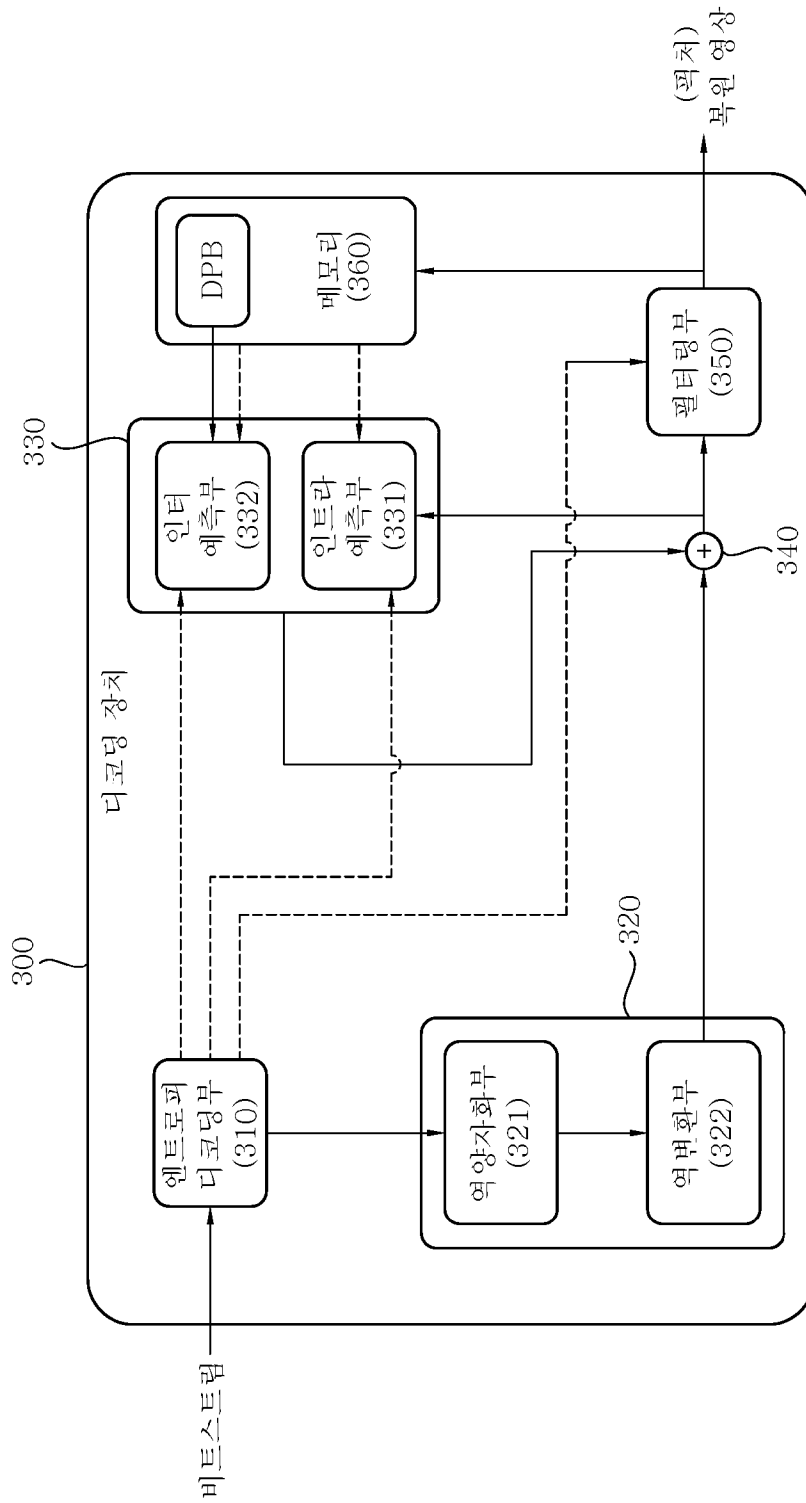
[도1]



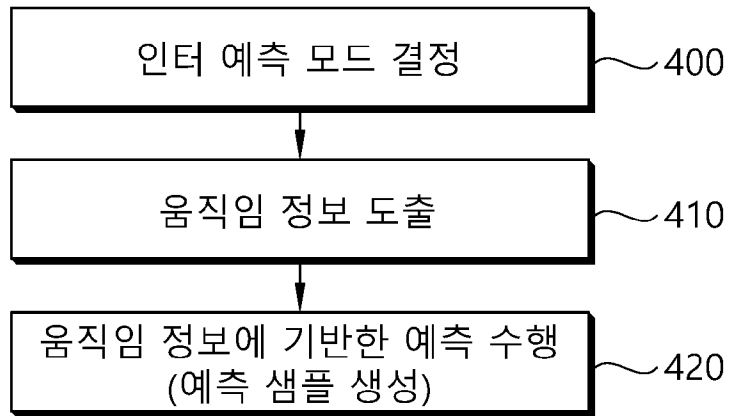
[도2]



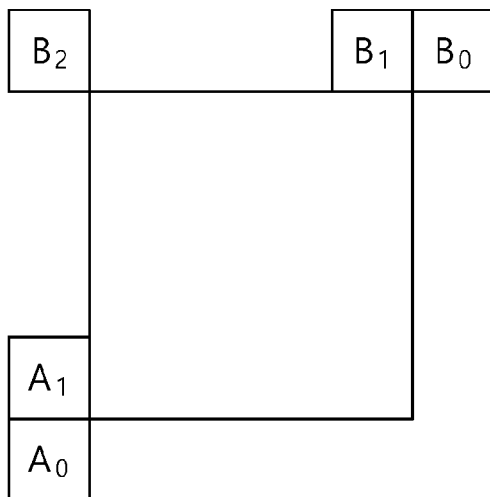
[도3]



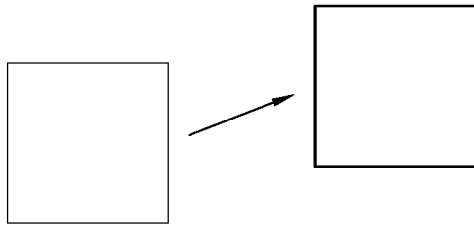
[도4]



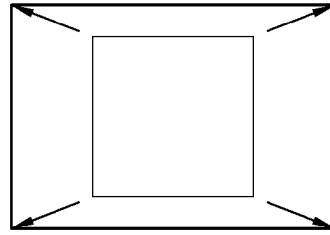
[도5]



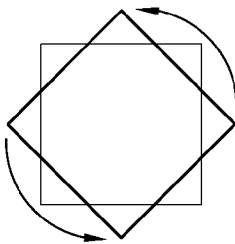
[도6]



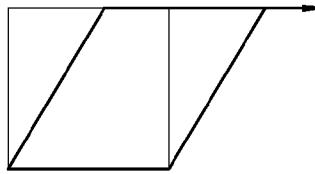
Translate



Scale

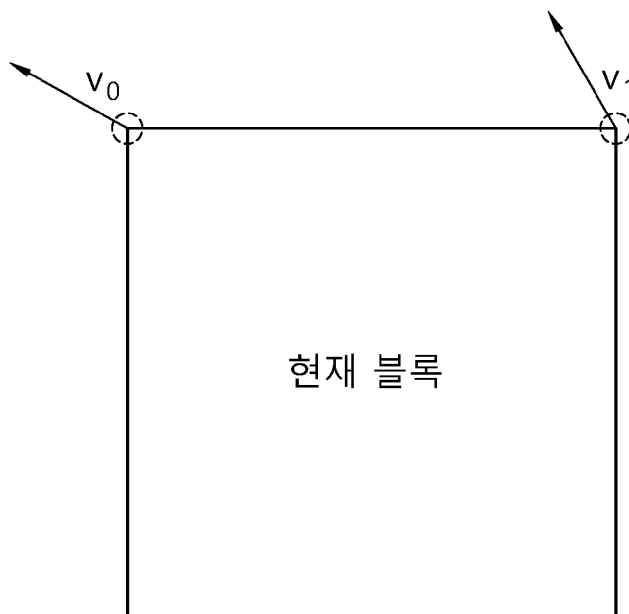


Rotate

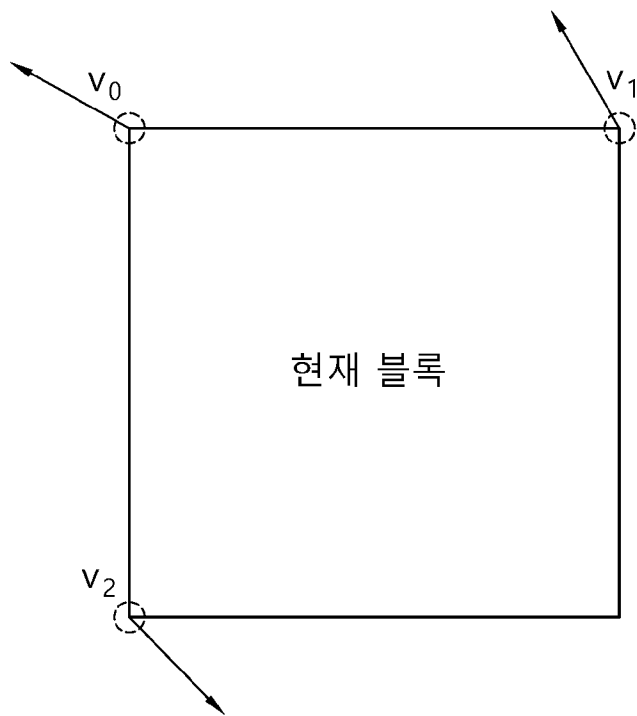


Shear

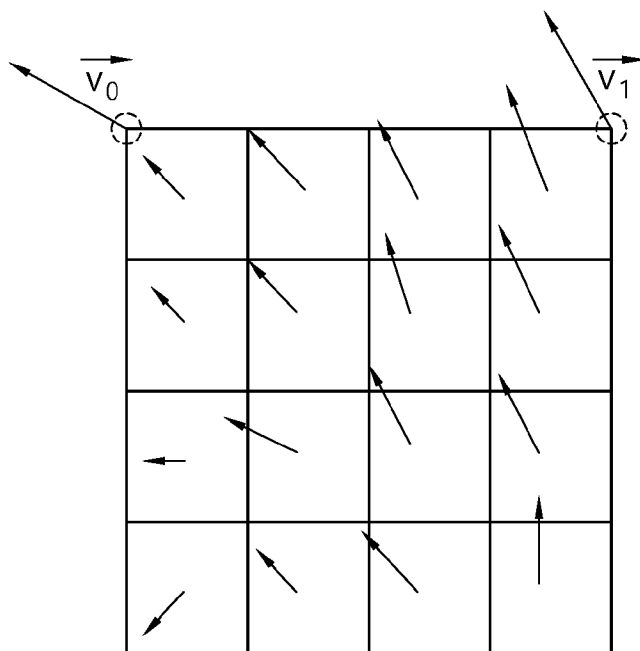
[도7a]



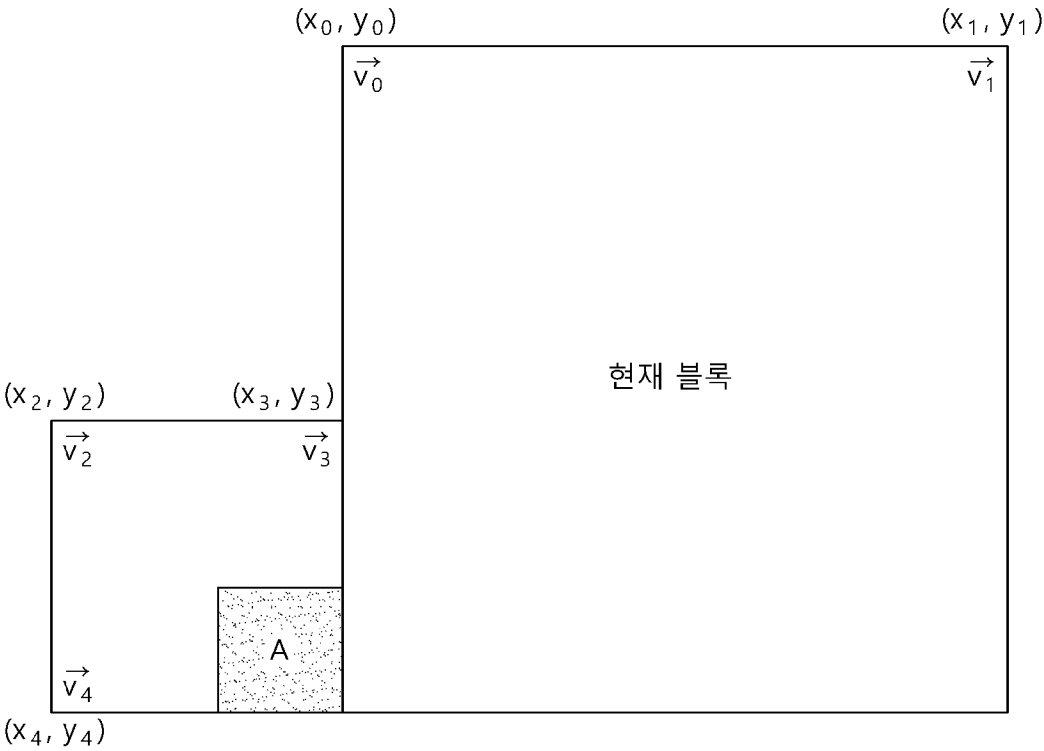
[도7b]



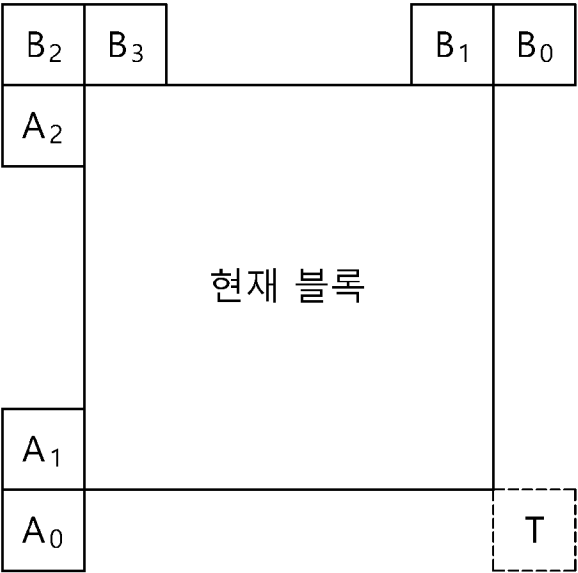
[도8]



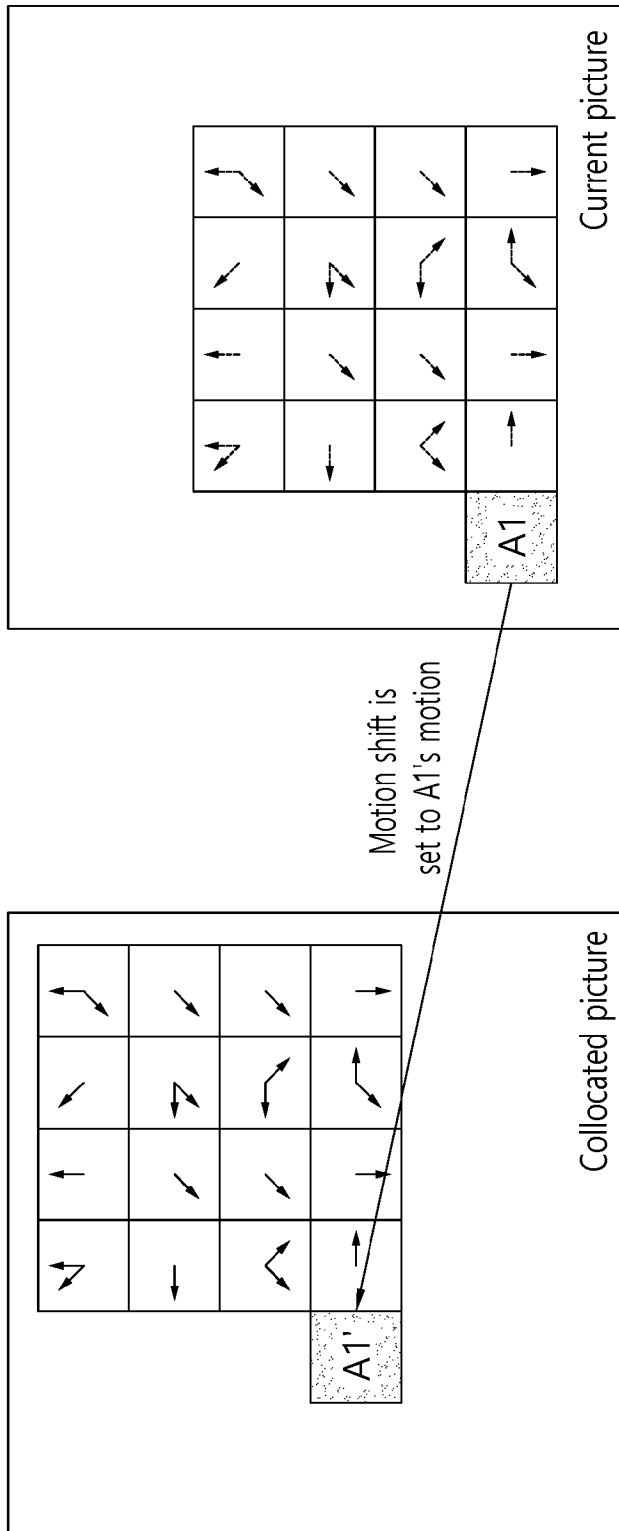
[도9]



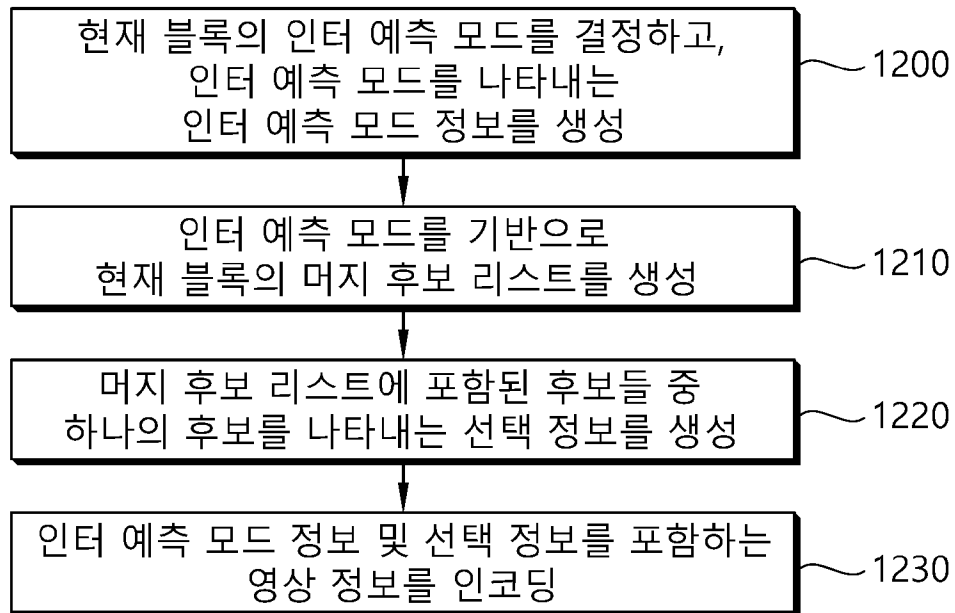
[도10]



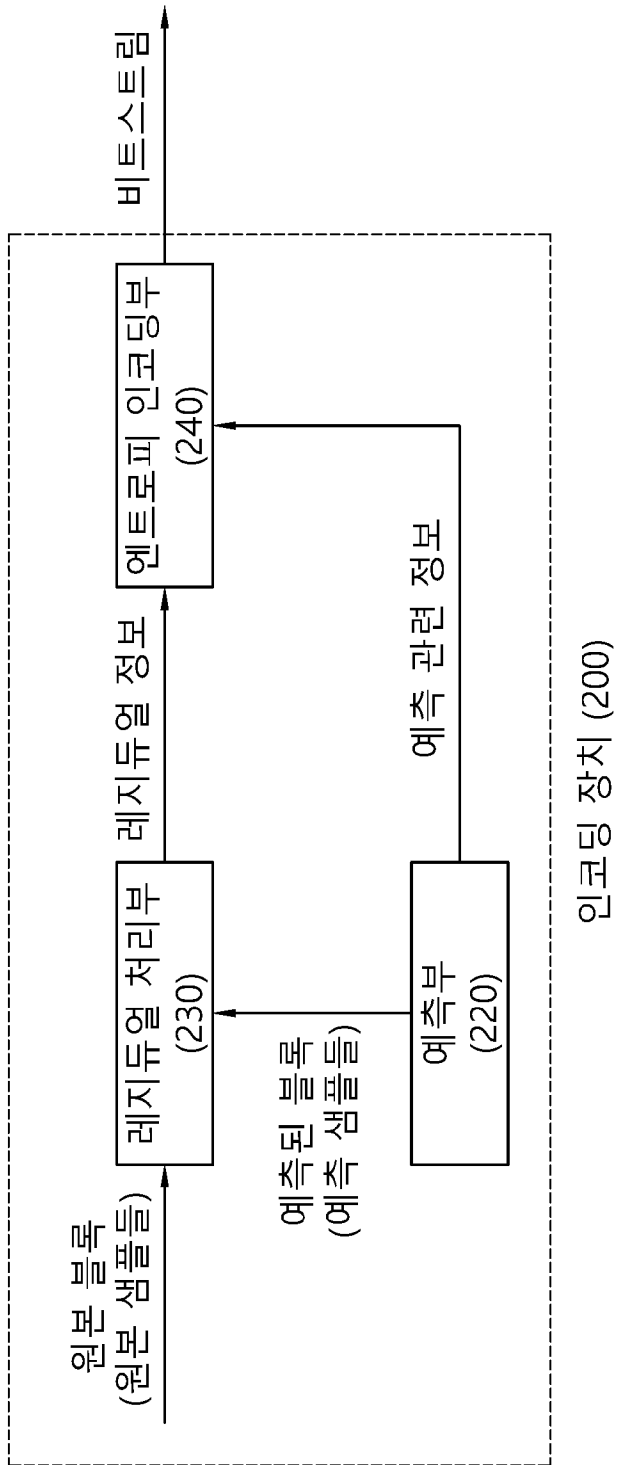
[도11]



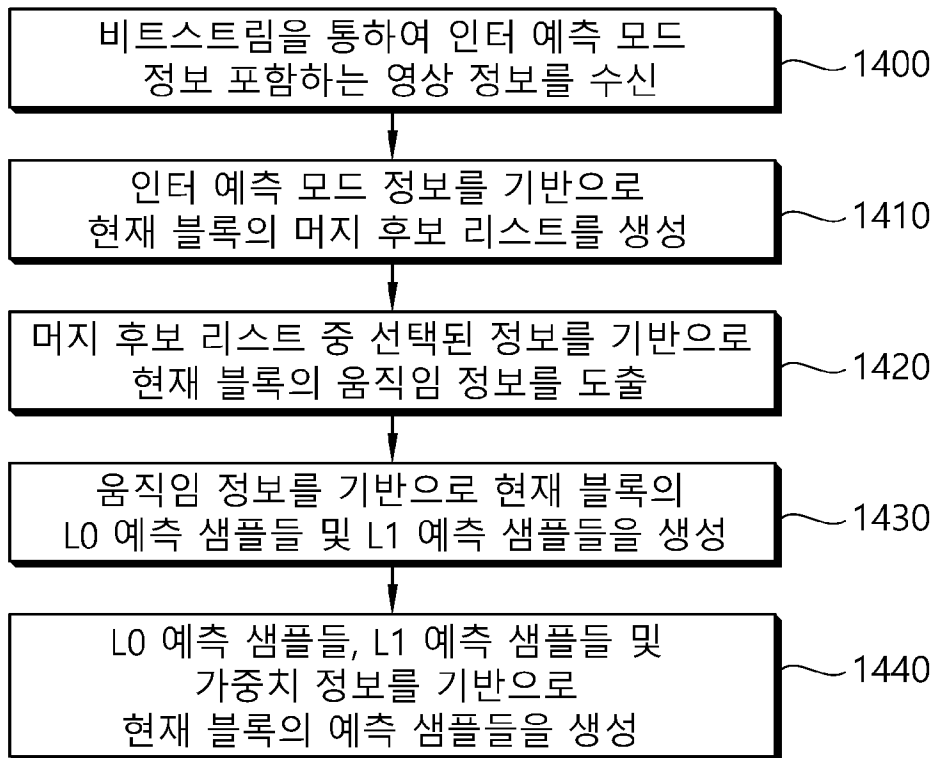
[도12]



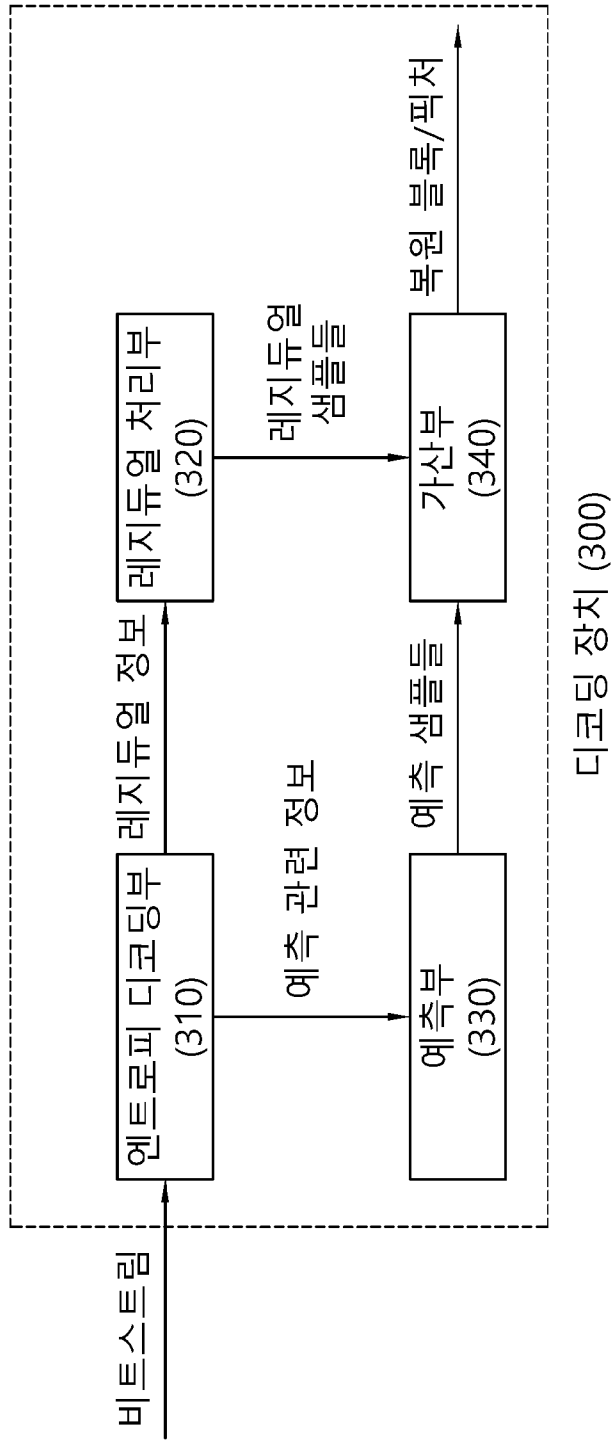
[도13]



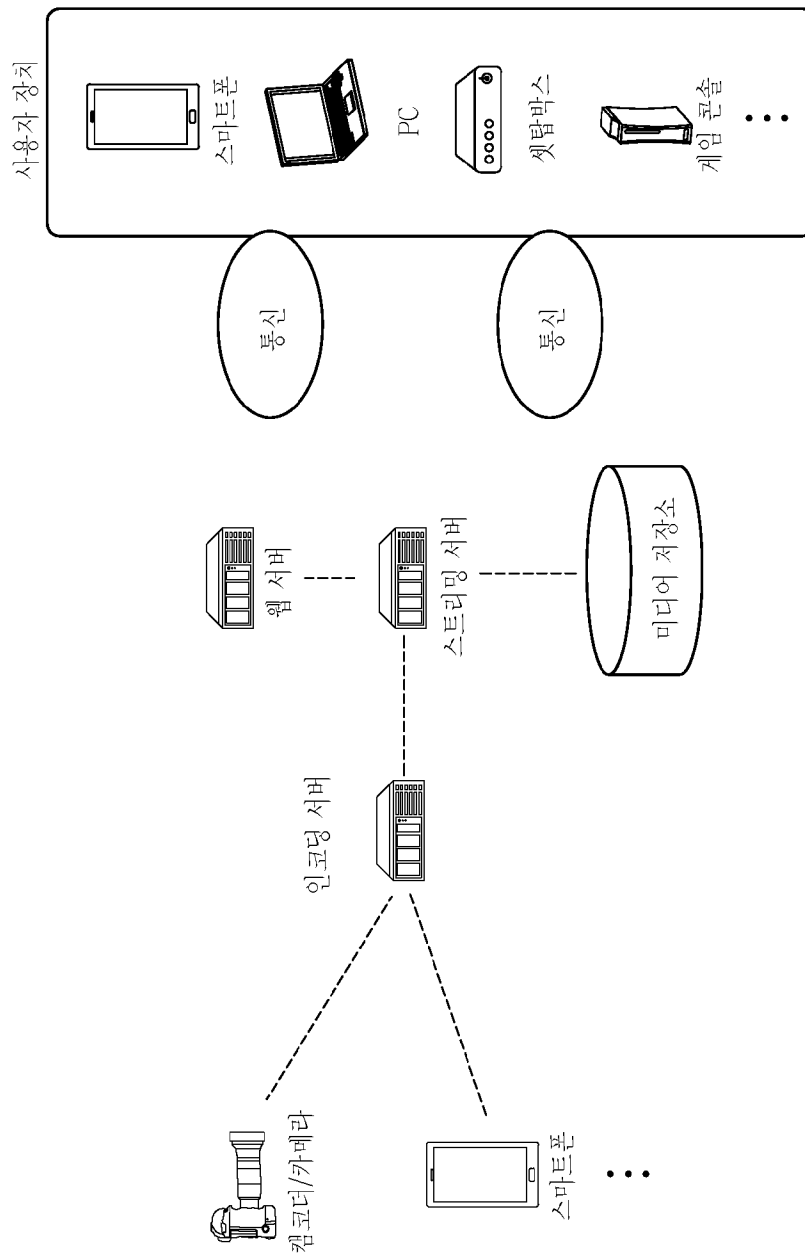
[도14]



[도15]



[도16]



## INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2020/007521

## A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/527(2014.01)i, H04N 19/109(2014.01)i, H04N 19/132(2014.01)i, H04N 19/137(2014.01)i, H04N 19/105(2014.01)i, H04N 19/176(2014.01)i, H04N 19/70(2014.01)i

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

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N 19/527; H04N 19/105; H04N 19/122; H04N 19/126; H04N 19/463; H04N 7/12; H04N 19/109; H04N 19/132; H04N 19/137; H04N 19/176; H04N 19/70

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: decoding, affine, merge, candidate, weight, index

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                         | Relevant to claim No. |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Y         | BROSS, Benjamin et al. Versatile Video Coding (Draft 5). JVET-N1001-v8. Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. 14th Meeting: Geneva, CH. 11 June 2019, pages 1-385<br>See pages 55-56, 191-205, 213, 215, 226. | 1-14                  |
| Y         | KR 10-2017-0124079 A (INDUSTRY ACADEMY COOPERATION FOUNDATION OF SEJONG UNIVERSITY) 09 November 2017<br>See paragraph [0113].                                                                                                                              | 1-14                  |
| A         | KR 10-2019-0018624 A (VID SCALE, INC) 25 February 2019<br>See paragraph [0084]; and figure 6.                                                                                                                                                              | 1-14                  |
| A         | KR 10-2018-0131570 A (INTELLECTUAL DISCOVERY CO., LTD.) 10 December 2018<br>See claims 1-4; and figure 58.                                                                                                                                                 | 1-14                  |
| A         | US 8457200 B2 (ANDERSSON, Kenneth et al.) 04 June 2013<br>See claims 1-10; and figure 1.                                                                                                                                                                   | 1-14                  |



Further documents are listed in the continuation of Box C.



See patent family annex.

\* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

31 AUGUST 2020 (31.08.2020)

Date of mailing of the international search report

31 AUGUST 2020 (31.08.2020)

Name and mailing address of the ISA/KR



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

Facsimile No. +82-42-481-8578

Authorized officer

Telephone No.

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.

**PCT/KR2020/007521**

| Patent document<br>cited in search report | Publication<br>date | Patent family<br>member | Publication<br>date |
|-------------------------------------------|---------------------|-------------------------|---------------------|
| KR 10-2017-0124079 A                      | 09/11/2017          | CN 109479142 A          | 15/03/2019          |
|                                           |                     | KR 10-2017-0124080 A    | 09/11/2017          |
|                                           |                     | WO 2017-188782 A2       | 02/11/2017          |
|                                           |                     | WO 2017-188782 A3       | 02/08/2018          |
| KR 10-2019-0018624 A                      | 25/02/2019          | CN 109496430 A          | 19/03/2019          |
|                                           |                     | EP 3456049 A1           | 20/03/2019          |
|                                           |                     | JP 2019-519148 A        | 04/07/2019          |
|                                           |                     | US 2019-0230350 A1      | 25/07/2019          |
|                                           |                     | WO 2017-197146 A1       | 16/11/2017          |
| KR 10-2018-0131570 A                      | 10/12/2018          | CN 109417619 A          | 01/03/2019          |
|                                           |                     | EP 3451662 A2           | 06/03/2019          |
|                                           |                     | EP 3451662 A4           | 22/04/2020          |
|                                           |                     | US 2019-0149836 A1      | 16/05/2019          |
|                                           |                     | WO 2017-188779 A2       | 02/11/2017          |
| US 8457200 B2                             | 04/06/2013          | WO 2017-188779 A3       | 02/08/2018          |
|                                           |                     | CN 101491107 A          | 22/07/2009          |
|                                           |                     | CN 101491107 B          | 18/07/2012          |
|                                           |                     | EP 2039171 A1           | 25/03/2009          |
|                                           |                     | EP 2039171 A4           | 28/09/2011          |
|                                           |                     | EP 2039171 B1           | 05/10/2016          |
|                                           |                     | US 2009-0257492 A1      | 15/10/2009          |
|                                           |                     | WO 2008-004940 A1       | 10/01/2008          |

**A. 발명이 속하는 기술분류(국제특허분류(IPC))**

H04N 19/527(2014.01)i, H04N 19/109(2014.01)i, H04N 19/132(2014.01)i, H04N 19/137(2014.01)i, H04N 19/105(2014.01)i, H04N 19/176(2014.01)i, H04N 19/70(2014.01)i

**B. 조사된 분야**

조사된 최소문헌(국제특허분류를 기재)

H04N 19/527; H04N 19/105; H04N 19/122; H04N 19/126; H04N 19/463; H04N 7/12; H04N 19/109; H04N 19/132; H04N 19/137; H04N 19/176; H04N 19/70

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: 복호화(decoding), 어파인(affine), 병합(merge), 후보(candidate), 가중치(weight), 인덱스(index)

**C. 관련 문헌**

| 카테고리* | 인용문헌명 및 관련 구절(해당하는 경우)의 기재                                                                                                                                                                                                                   | 관련 청구항 |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| Y     | BENJAMIN BROSS 등, 'Versatile Video Coding (Draft 5)', JVET-N1001-v8, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting: Geneva, CH, 2019.06.11, 페이지 1-385<br>페이지 55-56, 191-205, 213, 215, 226 | 1-14   |
| Y     | KR 10-2017-0124079 A (세종대학교산학협력단) 2017.11.09<br>단락 [0113]                                                                                                                                                                                    | 1-14   |
| A     | KR 10-2019-0018624 A (브이아이디 스케일, 인크.) 2019.02.25<br>단락 [0084]; 및 도면 6                                                                                                                                                                        | 1-14   |
| A     | KR 10-2018-0131570 A (인텔렉추얼디스커버리 주식회사) 2018.12.10<br>청구항 1-4; 및 도면 58                                                                                                                                                                        | 1-14   |
| A     | US 8457200 B2 (KENNETH ANDERSSON 등) 2013.06.04<br>청구항 1-10; 및 도면 1                                                                                                                                                                           | 1-14   |

☐ 추가 문헌이 C(계속)에 기재되어 있습니다.

☒ 대응특허에 관한 별지를 참조하십시오.

\* 인용된 문헌의 특별 카테고리:

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

“D” 본 국제출원에서 출원인이 인용한 문헌

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

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

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

“P” 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

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

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

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

“&” 동일한 대응특허문헌에 속하는 문헌

국제조사의 실제 완료일

2020년 08월 31일 (31.08.2020)

국제조사보고서 발송일

2020년 08월 31일 (31.08.2020)

ISA/KR의 명칭 및 우편주소



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

팩스 번호 +82-42-481-8578

심사관

양정록

전화번호 +82-42-481-5709



| 국제조사보고서에서<br>인용된 특허문헌 | 공개일        | 대응특허문헌                                                                                                                         | 공개일                                                                                            |
|-----------------------|------------|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| KR 10-2017-0124079 A  | 2017/11/09 | CN 109479142 A<br>KR 10-2017-0124080 A<br>WO 2017-188782 A2<br>WO 2017-188782 A3                                               | 2019/03/15<br>2017/11/09<br>2017/11/02<br>2018/08/02                                           |
| KR 10-2019-0018624 A  | 2019/02/25 | CN 109496430 A<br>EP 3456049 A1<br>JP 2019-519148 A<br>US 2019-0230350 A1<br>WO 2017-197146 A1                                 | 2019/03/19<br>2019/03/20<br>2019/07/04<br>2019/07/25<br>2017/11/16                             |
| KR 10-2018-0131570 A  | 2018/12/10 | CN 109417619 A<br>EP 3451662 A2<br>EP 3451662 A4<br>US 2019-0149836 A1<br>WO 2017-188779 A2<br>WO 2017-188779 A3               | 2019/03/01<br>2019/03/06<br>2020/04/22<br>2019/05/16<br>2017/11/02<br>2018/08/02               |
| US 8457200 B2         | 2013/06/04 | CN 101491107 A<br>CN 101491107 B<br>EP 2039171 A1<br>EP 2039171 A4<br>EP 2039171 B1<br>US 2009-0257492 A1<br>WO 2008-004940 A1 | 2009/07/22<br>2012/07/18<br>2009/03/25<br>2011/09/28<br>2016/10/05<br>2009/10/15<br>2008/01/10 |