

명세서

발명의 명칭: 영상 신호 부호화/복호화 방법 및 이를 위한 장치

기술분야

- [1] 본 발명은 영상 신호 부호화/복호화 방법 및 이를 위한 장치에 관한 것이다.

배경기술

- [2] 디스플레이 패널이 점점 더 대형화되는 추세에 따라 점점 더 높은 화질의 비디오 서비스가 요구되고 있다. 고화질 비디오 서비스의 가장 큰 문제는 데이터량이 크게 증가하는 것이며, 이러한 문제를 해결하기 위해, 비디오 압축율을 향상시키기 위한 연구가 활발하게 진행되고 있다. 대표적인 예로, 2009년에 MPEG(Motion Picture Experts Group)과 ITU-T(International Telecommunication Union-Telecommunication) 산하의 VCEG(Video Coding Experts Group)에서는 JCT-VC(Joint Collaborative Team on Video Coding)를 결성하였다. JCT-VC는 H.264/AVC에 비해 약 2배의 압축 성능을 갖는 비디오 압축 표준인 HEVC(High Efficiency Video Coding)를 제안하였으며, 2013년 1월 25일에 표준 승인되었다. 고화질 비디오 서비스의 급격한 발전에 따라 HEVC의 성능도 점차적으로 그 한계를 드러내고 있다.

발명의 상세한 설명

기술적 과제

- [3] 본 발명은 비디오 신호를 부호화/복호화함에 있어서, 오프셋 벡터를 기초로 머지 후보로부터 유도된 움직임 벡터를 리파인하는 방법 및 상기 방법을 수행하기 위한 장치를 제공하는 것을 목적으로 한다.
- [4] 본 발명은 비디오 신호를 부호화/복호화함에 있어서, 오프셋 벡터의 시그널링 방법 및 상기 방법을 수행하기 위한 장치를 제공하는 것을 목적으로 한다.
- [5] 본 발명에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제 해결 수단

- [6] 본 발명에 따른 비디오 신호 복호화/부호화 방법은, 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는지 여부를 결정하는 단계, 상기 현재 블록에 대한 머지 후보 리스트를 생성하는 단계, 상기 머지 후보 리스트를 기초로, 상기 현재 블록에 대한 머지 후보를 특정하는 단계, 및 상기 머지 후보를 기초로, 상기 현재 블록에 대한 움직임 벡터를 유도하는 단계를 포함한다. 이때, 상기 현재 블록에 상기 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 상기 현재 블록의 움직임 벡터는, 상기 머지 후보의 움직임 벡터에 오프셋 벡터를 가산하여 유도되고 상기 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수가 복수개인 경우

상기 현재 블록의 상기 머지 후보는 상기 머지 후보들 중 어느 하나를 가리키는 비트스트림으로부터 복호화되는 인덱스 정보를 기초로 선택되고, 상기 최대 개수가 1개인 경우 상기 인덱스 정보를 복호화함이 없이 상기 머지 후보가 결정될 수 있다.

- [7] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 오프셋 벡터의 크기는, 벡터 크기 후보들 중 어느 하나를 특정하는 제1 인덱스 정보를 기초로 결정될 수 있다.
- [8] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 벡터 크기 후보들의 최대값 또는 최소값 중 적어도 하나는 상기 벡터 크기 후보들의 범위를 나타내는 플래그의 값에 따라 상이하게 설정될 수 있다.
- [9] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 플래그는 픽처 파라미터 세트를 통해 시그널링될 수 있다.
- [10] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 벡터 크기 후보들의 최대값 또는 최소값 중 적어도 하나는 상기 현재 블록에 대한 움직임 벡터 정밀도에 따라 상이하게 설정될 수 있다.
- [11] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 오프셋 벡터의 크기는, 상기 제1 인덱스 정보에 의해 특정된 벡터 크기 후보가 가리키는 값에 비트 시프팅 연산을 적용함으로써 획득될 수 있다.
- [12] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 오프셋 벡터의 방향은, 벡터 방향 후보들 중 어느 하나를 특정하는 제2 인덱스 정보를 기초로 결정될 수 있다.
- [13] 본 발명에 대하여 위에서 간략하게 요약된 특징들은 후술하는 본 발명의 상세한 설명의 예시적인 양상일 뿐이며, 본 발명의 범위를 제한하는 것은 아니다.

발명의 효과

- [14] 본 발명에 의하면, 오프셋 벡터를 기초로 머지 후보의 움직임 벡터를 리파인함으로써, 인터 예측 효율을 향상시킬 수 있다.
- [15] 본 발명에 의하면, 오프셋 벡터의 크기 및 방향을 적응적으로 결정함으로써, 인터 예측 효율을 향상시킬 수 있다.
- [16] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

- [17] 도 1은 본 발명의 일 실시예에 따른 영상 부호화기(인코더기)의 블록도이다.
- [18] 도 2는 본 발명의 일 실시예에 따른 영상 복호화기(디코더기)의 블록도이다.
- [19] 도 3은 본 발명의 일 실시예에 따른 기본 코딩 트리 유닛을 도시한 도면이다.
- [20] 도 4는 코딩 블록의 다양한 분할 형태를 나타낸 도면이다.

- [21] 도 5는 코딩 트리 유닛의 분할 양상을 예시한 도면이다.
- [22] 도 6은 데이터 기본 유닛의 형태를 나타낸 도면이다.
- [23] 도 7 및 도 8은 코딩 블록이 복수의 서브 블록들로 분할되는 예를 나타낸 도면이다.
- [24] 도 9는 본 발명의 일 실시예에 따른 인터 예측 방법의 흐름도이다.
- [25] 도 10은 오브젝트의 비선형 움직임을 예시한 도면이다.
- [26] 도 11은 본 발명의 일 실시예에 따른 어파인 모션에 기초한 인터 예측 방법의 흐름도이다.
- [27] 도 12는 어파인 모션 모델 별 어파인 시드 벡터를 예시한 도면이다.
- [28] 도 13은 4 파라미터 모션 모델 하에서 서브 블록들의 어파인 벡터들을 예시한 도면이다.
- [29] 도 14는 머지 모드 하에서 현재 블록의 움직임 정보를 유도하는 과정의 흐름도이다.
- [30] 도 15는 머지 후보를 유도하기 위해 사용되는 후보 블록들을 예시한 도면이다.
- [31] 도 16은 기준 샘플들의 위치를 나타낸 도면이다.
- [32] 도 17은 머지 후보를 유도하기 위해 사용되는 후보 블록들을 예시한 도면이다.
- [33] 도 18은 기준 샘플의 위치가 변경되는 예를 도시한 도면이다.
- [34] 도 19는 기준 샘플의 위치가 변경되는 예를 도시한 도면이다.
- [35] 도 20은 인터 영역 모션 정보 테이블의 업데이트 양상을 설명하기 위한 도면이다.
- [36] 도 21은 인터 영역 머지 후보 테이블의 업데이트 양상을 나타낸 도면이다.
- [37] 도 22는 기 저장된 인터 영역 머지 후보의 인덱스가 갱신되는 예를 나타낸 도면이다.
- [38] 도 23은 대표 서브 블록의 위치를 나타낸 도면이다.
- [39] 도 24는 인터 예측 모드 별로 인터 영역 모션 정보 테이블이 생성되는 예를 나타낸 것이다.
- [40] 도 25는 룬텀 모션 정보 테이블에 포함된 인터 영역 머지 후보가 머지 후보 리스트에 추가되는 예를 나타낸 도면이다.
- [41] 도 26은 머지 후보들 중 일부에 대해서만 중복성 검사가 수행되는 예를 도시한 도면이다.
- [42] 도 27은 특정 머지 후보와의 중복성 검사가 생략되는 예를 나타낸 도면이다.
- [43] 도 28은 오프셋 벡터의 크기를 나타내는 distance_idx 및 오프셋 벡터의 방향을 나타내는 direction_idx의 값에 따른 오프셋 벡터를 나타낸 도면이다.
- [44] 도 29는 오프셋 벡터의 크기를 나타내는 distance_idx 및 오프셋 벡터의 방향을 나타내는 direction_idx의 값에 따른 오프셋 벡터를 나타낸 도면이다.
- [45] 도 30은 삼각 파티셔닝 기법이 적용되는 경우, 코딩 블록의 분할 양상을 나타낸 도면이다.
- [46] 도 31은 서브 유닛들 각각의 오프셋 벡터가 상이하게 설정된 예를 나타낸

도면이다.

- [47] 도 32는 리파인 머지 후보가 취할 수 있는 움직임 벡터 후보들을 나타낸 도면이다.
- [48] 도 33은 머지 리파인먼트 오프셋 리스트의 구성을 예시한 도면이다.
- [49] 도 34 및 도 35는 머지 오프셋 후보들에 의해 특정되는 오프셋 벡터를 나타낸 도면이다.
- [50] 도 36은 모션 벡터 예측 후보들을 유도하는데 이용되는 후보 블록들을 나타낸 도면이다.
- [51] 도 37은 리파인 모션 벡터 예측 후보로 설정될 수 있는 움직임 벡터 후보들을 나타낸 도면이다.
- [52] 도 38은 예측 벡터 리파인먼트 오프셋 리스트의 구성을 예시한 도면이다.

발명의 실시를 위한 형태

- [53] 이하에서는 도면을 참조하여 본 발명의 실시예를 상세히 설명한다.
- [54] 영상의 부호화 및 복호화는 블록 단위로 수행된다. 일 예로, 코딩 블록, 변환 블록, 또는 예측 블록에 대해, 변환, 양자화, 예측, 인루프 필터링, 또는 복원 등의 부호화/복호화 처리가 수행될 수 있다.
- [55] 이하, 부호화/복호화 대상인 블록을 '현재 블록'이라 호칭하기로 한다. 일 예로, 현재 블록은 현재 부호화/복호화 처리 단계에 따라, 코딩 블록, 변환 블록, 또는 예측 블록을 나타낼 수 있다.
- [56] 아울러, 본 명세서에서 사용되는 용어 '유닛'은 특정 부호화/복호화 프로세스를 수행하기 위한 기본 단위를 나타내고, '블록'은 소정 크기의 샘플 어레이를 나타내는 것으로 이해될 수 있다. 별도의 설명이 없는 한, '블록'과 '유닛'은 동등한 의미로 사용될 수 있다. 일 예로, 후술되는 실시예에서, 코딩 블록과 코딩 유닛은 상호 동등한 의미를 갖는 것으로 이해될 수 있다.
- [57] 도 1은 본 발명의 일 실시예에 따른 영상 부호화기(인코더)의 블록도이다.
- [58] 도 1을 참조하면, 영상 부호화 장치(100)는 픽처 분할부(110), 예측부(120, 125), 변환부(130), 양자화부(135), 재정렬부(160), 엔트로피 부호화부(165), 역양자화부(140), 역변환부(145), 필터부(150) 및 메모리(155)를 포함할 수 있다.
- [59] 도 1에 나타난 각 구성부들은 영상 부호화 장치에서 서로 다른 특징적인 기능들을 나타내기 위해 독립적으로 도시한 것으로, 각 구성부들이 분리된 하드웨어나 하나의 소프트웨어 구성단위로 이루어짐을 의미하지 않는다. 즉, 각 구성부는 설명의 편의상 각각의 구성부로 나열하여 포함한 것으로 각 구성부 중 적어도 두 개의 구성부가 합쳐져 하나의 구성부로 이루어지거나, 하나의 구성부가 복수개의 구성부로 나뉘어져 기능을 수행할 수 있고 이러한 각 구성부의 통합된 실시예 및 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.
- [60] 또한, 일부의 구성 요소는 본 발명에서 본질적인 기능을 수행하는 필수적인

구성 요소는 아니고 단지 성능을 향상시키기 위한 선택적 구성 요소일 수 있다. 본 발명은 단지 성능 향상을 위해 사용되는 구성 요소를 제외한 본 발명의 본질을 구현하는데 필수적인 구성부만을 포함하여 구현될 수 있고, 단지 성능 향상을 위해 사용되는 선택적 구성 요소를 제외한 필수 구성 요소만을 포함한 구조도 본 발명의 권리범위에 포함된다.

- [61] 픽처 분할부(110)는 입력된 픽처를 적어도 하나의 처리 단위로 분할할 수 있다. 이때, 처리 단위는 예측 단위(Prediction Unit: PU)일 수도 있고, 변환 단위(Transform Unit: TU)일 수도 있으며, 부호화 단위(Coding Unit: CU)일 수도 있다. 픽처 분할부(110)에서는 하나의 픽처에 대해 복수의 부호화 단위, 예측 단위 및 변환 단위의 조합으로 분할하고 소정의 기준(예를 들어, 비용 함수)으로 하나의 부호화 단위, 예측 단위 및 변환 단위 조합을 선택하여 픽처를 부호화 할 수 있다.
- [62] 예를 들어, 하나의 픽처는 복수개의 부호화 단위로 분할될 수 있다. 픽처에서 부호화 단위를 분할하기 위해서는 쿼드 트리 구조(Quad Tree Structure)와 같은 재귀적인 트리 구조를 사용할 수 있는데 하나의 영상 또는 최대 크기 부호화 단위(largest coding unit)를 루트로 하여 다른 부호화 단위로 분할되는 부호화 유닛은 분할된 부호화 단위의 개수만큼의 자식 노드를 가지고 분할될 수 있다. 일정한 제한에 따라 더 이상 분할되지 않는 부호화 단위는 리프 노드가 된다. 즉, 하나의 코딩 유닛에 대하여 정방형 분할만이 가능하다고 가정하는 경우, 하나의 부호화 단위는 최대 4개의 다른 부호화 단위로 분할될 수 있다.
- [63] 이하, 본 발명의 실시예에서는 부호화 단위는 부호화를 수행하는 단위의 의미로 사용할 수도 있고, 복호화를 수행하는 단위의 의미로 사용할 수도 있다.
- [64] 예측 단위는 하나의 부호화 단위 내에서 동일한 크기의 적어도 하나의 정사각형 또는 직사각형 등의 형태를 가지고 분할된 것일 수도 있고, 하나의 부호화 단위 내에서 분할된 예측 단위 중 어느 하나의 예측 단위가 다른 하나의 예측 단위와 상이한 형태 및/또는 크기를 가지도록 분할된 것일 수도 있다.
- [65] 부호화 단위를 기초로 인트라 예측을 수행하는 예측 단위를 생성시 최소 부호화 단위가 아닌 경우, 복수의 예측 단위 $N \times N$ 으로 분할하지 않고 인트라 예측을 수행할 수 있다.
- [66] 예측부(120, 125)는 인트라 예측을 수행하는 인트라 예측부(120)와 인트라 예측을 수행하는 인트라 예측부(125)를 포함할 수 있다. 예측 단위에 대해 인트라 예측을 사용할 것인지 또는 인트라 예측을 수행할 것인지를 결정하고, 각 예측 방법에 따른 구체적인 정보(예컨대, 인트라 예측 모드, 모션 벡터, 참조 픽처 등)를 결정할 수 있다. 이때, 예측이 수행되는 처리 단위와 예측 방법 및 구체적인 내용이 정해지는 처리 단위는 다를 수 있다. 예컨대, 예측의 방법과 예측 모드 등은 예측 단위로 결정되고, 예측의 수행은 변환 단위로 수행될 수도 있다. 생성된 예측 블록과 원본 블록 사이의 잔차값(잔차 블록)은 변환부(130)로 입력될 수 있다. 또한, 예측을 위해 사용한 예측 모드 정보, 모션 벡터 정보 등은

잔차값과 함께 엔트로피 부호화부(165)에서 부호화되어 복호화기에 전달될 수 있다. 특정한 부호화 모드를 사용할 경우, 예측부(120, 125)를 통해 예측 블록을 생성하지 않고, 원본 블록을 그대로 부호화하여 복호화부에 전송하는 것도 가능하다.

- [67] 인터 예측부(120)는 현재 픽처의 이전 픽처 또는 이후 픽처 중 적어도 하나의 픽처의 정보를 기초로 예측 단위를 예측할 수도 있고, 경우에 따라서는 현재 픽처 내의 부호화가 완료된 일부 영역의 정보를 기초로 예측 단위를 예측할 수도 있다. 인터 예측부(120)는 참조 픽처 보간부, 모션 예측부, 움직임 보상부를 포함할 수 있다.
- [68] 참조 픽처 보간부에서는 메모리(155)로부터 참조 픽처 정보를 제공받고 참조 픽처에서 정수 화소 이하의 화소 정보를 생성할 수 있다. 휘도 화소의 경우, 1/4 화소 단위로 정수 화소 이하의 화소 정보를 생성하기 위해 필터 계수를 달리하는 DCT 기반의 8탭 보간 필터(DCT-based Interpolation Filter)가 사용될 수 있다. 색차 신호의 경우 1/8 화소 단위로 정수 화소 이하의 화소 정보를 생성하기 위해 필터 계수를 달리하는 DCT 기반의 4탭 보간 필터(DCT-based Interpolation Filter)가 사용될 수 있다.
- [69] 모션 예측부는 참조 픽처 보간부에 의해 보간된 참조 픽처를 기초로 모션 예측을 수행할 수 있다. 모션 벡터를 산출하기 위한 방법으로 FBMA(Full search-based Block Matching Algorithm), TSS(Three Step Search), NTS(New Three-Step Search Algorithm) 등 다양한 방법이 사용될 수 있다. 모션 벡터는 보간된 화소를 기초로 1/2 또는 1/4 화소 단위의 모션 벡터값을 가질 수 있다. 모션 예측부에서는 모션 예측 방법을 다르게 하여 현재 예측 단위를 예측할 수 있다. 모션 예측 방법으로 스킵(Skip) 방법, 머지(Merge) 방법, AMVP(Advanced Motion Vector Prediction) 방법, 인트라 블록 카피(Intra Block Copy) 방법 등 다양한 방법이 사용될 수 있다.
- [70] 인트라 예측부(125)는 현재 픽처 내의 화소 정보인 현재 블록 주변의 참조 픽셀 정보를 기초로 예측 단위를 생성할 수 있다. 현재 예측 단위의 주변 블록이 인터 예측을 수행한 블록이어서, 참조 픽셀이 인터 예측을 수행한 픽셀일 경우, 인터 예측을 수행한 블록에 포함되는 참조 픽셀을 주변의 인트라 예측을 수행한 블록의 참조 픽셀 정보로 대체하여 사용할 수 있다. 즉, 참조 픽셀이 가용하지 않는 경우, 가용하지 않은 참조 픽셀 정보를 가용한 참조 픽셀 중 적어도 하나의 참조 픽셀로 대체하여 사용할 수 있다.
- [71] 인트라 예측에서 예측 모드는 참조 픽셀 정보를 예측 방향에 따라 사용하는 방향성 예측 모드와 예측을 수행시 방향성 정보를 사용하지 않는 비방향성 모드를 가질 수 있다. 휘도 정보를 예측하기 위한 모드와 색차 정보를 예측하기 위한 모드가 상이할 수 있고, 색차 정보를 예측하기 위해 휘도 정보를 예측하기 위해 사용된 인트라 예측 모드 정보 또는 예측된 휘도 신호 정보를 활용할 수 있다.

- [72] 인트라 예측을 수행할 때 예측 단위의 크기와 변환 단위의 크기가 동일할 경우, 예측 단위의 좌측에 존재하는 픽셀, 좌측 상단에 존재하는 픽셀, 상단에 존재하는 픽셀을 기초로 예측 단위에 대한 인트라 예측을 수행할 수 있다. 그러나 인트라 예측을 수행할 때 예측 단위의 크기와 변환 단위의 크기가 상이할 경우, 변환 단위를 기초로 한 참조 픽셀을 이용하여 인트라 예측을 수행할 수 있다. 또한, 최소 부호화 단위에 대해서만 $N \times N$ 분할을 사용하는 인트라 예측을 사용할 수 있다.
- [73] 인트라 예측 방법은 예측 모드에 따라 참조 화소에 AIS(Adaptive Intra Smoothing) 필터를 적용한 후 예측 블록을 생성할 수 있다. 참조 화소에 적용되는 AIS 필터의 종류는 상이할 수 있다. 인트라 예측 방법을 수행하기 위해 현재 예측 단위의 인트라 예측 모드는 현재 예측 단위의 주변에 존재하는 예측 단위의 인트라 예측 모드로부터 예측할 수 있다. 주변 예측 단위로부터 예측된 모드 정보를 이용하여 현재 예측 단위의 예측 모드를 예측하는 경우, 현재 예측 단위와 주변 예측 단위의 인트라 예측 모드가 동일하면 소정의 플래그 정보를 이용하여 현재 예측 단위와 주변 예측 단위의 예측 모드가 동일하다는 정보를 전송할 수 있고, 만약 현재 예측 단위와 주변 예측 단위의 예측 모드가 상이하면 엔트로피 부호화를 수행하여 현재 블록의 예측 모드 정보를 부호화할 수 있다.
- [74] 또한, 예측부(120, 125)에서 생성된 예측 단위를 기초로 예측을 수행한 예측 단위와 예측 단위의 원본 블록과 차이값인 잔차값(Residual) 정보를 포함하는 잔차 블록이 생성될 수 있다. 생성된 잔차 블록은 변환부(130)로 입력될 수 있다.
- [75] 변환부(130)에서는 원본 블록과 예측부(120, 125)를 통해 생성된 예측 단위의 잔차값(residual) 정보를 포함한 잔차 블록을 DCT(Discrete Cosine Transform) 또는 DST(Discrete Sine Transform)와 같은 변환 방법을 사용하여 변환시킬 수 있다. 여기서, DCT 변환 코어는 DCT2 또는 DCT8 중 적어도 하나를 포함하고, DST 변환 코어는 DST7을 포함한다. 잔차 블록을 변환하기 위해 DCT를 적용할지 또는 DST를 적용할지는 잔차 블록을 생성하기 위해 사용된 예측 단위의 인트라 예측 모드 정보를 기초로 결정할 수 있다. 잔차 블록에 대한 변환을 스킵할 수도 있다. 잔차 블록에 대한 변환을 스킵할 것인지 여부를 나타내는 플래그를 부호화할 수 있다. 변환 스킵은, 크기가 문턱값 이하인 잔차 블록, 루마 성분 또는 4:4:4 포맷 하에서의 크로마 성분에 대해 허용될 수 있다.
- [76] 양자화부(135)는 변환부(130)에서 주파수 영역으로 변환된 값들을 양자화할 수 있다. 블록에 따라 또는 영상의 중요도에 따라 양자화 계수는 변할 수 있다. 양자화부(135)에서 산출된 값은 역양자화부(140)와 재정렬부(160)에 제공될 수 있다.
- [77] 재정렬부(160)는 양자화된 잔차값에 대해 계수값의 재정렬을 수행할 수 있다.
- [78] 재정렬부(160)는 계수 스캐닝(Coefficient Scanning) 방법을 통해 2차원의 블록 형태 계수를 1차원의 벡터 형태로 변경할 수 있다. 예를 들어, 재정렬부(160)에서는 지그-재그 스캔(Zig-Zag Scan)방법을 이용하여 DC

계수부터 고주파수 영역의 계수까지 스캔하여 1차원 벡터 형태로 변경시킬 수 있다. 변환 단위의 크기 및 인트라 예측 모드에 따라 지그-재그 스캔 대신 2차원의 블록 형태 계수를 열 방향으로 스캔하는 수직 스캔, 2차원의 블록 형태 계수를 행 방향으로 스캔하는 수평 스캔이 사용될 수도 있다. 즉, 변환 단위의 크기 및 인트라 예측 모드에 따라 지그-재그 스캔, 수직 방향 스캔 및 수평 방향 스캔 중 어떠한 스캔 방법이 사용될지 여부를 결정할 수 있다.

- [79] 엔트로피 부호화부(165)는 재정렬부(160)에 의해 산출된 값들을 기초로 엔트로피 부호화를 수행할 수 있다. 엔트로피 부호화는 예를 들어, 지수 곱셈(Exponential Golomb), CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary Arithmetic Coding)과 같은 다양한 부호화 방법을 사용할 수 있다.
- [80] 엔트로피 부호화부(165)는 재정렬부(160) 및 예측부(120, 125)로부터 부호화 단위의 잔차값 계수 정보 및 블록 타입 정보, 예측 모드 정보, 분할 단위 정보, 예측 단위 정보 및 전송 단위 정보, 모션 벡터 정보, 참조 프레임 정보, 블록의 보간 정보, 필터링 정보 등 다양한 정보를 부호화할 수 있다.
- [81] 엔트로피 부호화부(165)에서는 재정렬부(160)에서 입력된 부호화 단위의 계수값을 엔트로피 부호화할 수 있다.
- [82] 역양자화부(140) 및 역변환부(145)에서는 양자화부(135)에서 양자화된 값들을 역양자화하고 변환부(130)에서 변환된 값들을 역변환한다. 역양자화부(140) 및 역변환부(145)에서 생성된 잔차값(Residual)은 예측부(120, 125)에 포함된 움직임 추정부, 움직임 보상부 및 인트라 예측부를 통해서 예측된 예측 단위와 합쳐져 복원 블록(Reconstructed Block)을 생성할 수 있다.
- [83] 필터부(150)는 디블록킹 필터, 오프셋 보정부, ALF(Adaptive Loop Filter)중 적어도 하나를 포함할 수 있다.
- [84] 디블록킹 필터는 복원된 픽처에서 블록간의 경계로 인해 생긴 블록 왜곡을 제거할 수 있다. 디블록킹을 수행할지 여부를 판단하기 위해 블록에 포함된 몇 개의 열 또는 행에 포함된 픽셀을 기초로 현재 블록에 디블록킹 필터 적용할지 여부를 판단할 수 있다. 블록에 디블록킹 필터를 적용하는 경우 필요한 디블록킹 필터링 강도에 따라 강한 필터(Strong Filter) 또는 약한 필터(Weak Filter)를 적용할 수 있다. 또한 디블록킹 필터를 적용함에 있어 수직 필터링 및 수평 필터링 수행시 수평 방향 필터링 및 수직 방향 필터링이 병행 처리되도록 할 수 있다.
- [85] 오프셋 보정부는 디블록킹을 수행한 영상에 대해 픽셀 단위로 원본 영상과의 오프셋을 보정할 수 있다. 특정 픽처에 대한 오프셋 보정을 수행하기 위해 영상에 포함된 픽셀을 일정한 수의 영역으로 구분한 후 오프셋을 수행할 영역을 결정하고 해당 영역에 오프셋을 적용하는 방법 또는 각 픽셀의 에지 정보를 고려하여 오프셋을 적용하는 방법을 사용할 수 있다.
- [86] ALF(Adaptive Loop Filtering)는 필터링한 복원 영상과 원래의 영상을 비교한

값을 기초로 수행될 수 있다. 영상에 포함된 픽셀을 소정의 그룹으로 나눈 후 해당 그룹에 적용될 하나의 필터를 결정하여 그룹마다 차별적으로 필터링을 수행할 수 있다. ALF를 적용할지 여부에 관련된 정보는 휘도 신호는 부호화 단위(Coding Unit, CU) 별로 전송될 수 있고, 각각의 블록에 따라 적용될 ALF 필터의 모양 및 필터 계수는 달라질 수 있다. 또한, 적용 대상 블록의 특성에 상관없이 동일한 형태(고정된 형태)의 ALF 필터가 적용될 수도 있다.

- [87] 메모리(155)는 필터부(150)를 통해 산출된 복원 블록 또는 픽처를 저장할 수 있고, 저장된 복원 블록 또는 픽처는 인터 예측을 수행 시 예측부(120, 125)에 제공될 수 있다.
- [88] 도 2는 본 발명의 일 실시예에 따른 영상 복호화기(디코더기)의 블록도이다.
- [89] 도 2를 참조하면, 영상 복호화기(200)는 엔트로피 복호화부(210), 재정렬부(215), 역양자화부(220), 역변환부(225), 예측부(230, 235), 필터부(240), 메모리(245)가 포함될 수 있다.
- [90] 영상 부호화기에서 영상 비트스트림이 입력된 경우, 입력된 비트스트림은 영상 부호화기와 반대의 절차로 복호화될 수 있다.
- [91] 엔트로피 복호화부(210)는 영상 부호화기의 엔트로피 부호화부에서 엔트로피 부호화를 수행한 것과 반대의 절차로 엔트로피 복호화를 수행할 수 있다. 예를 들어, 영상 부호화기에서 수행된 방법에 대응하여 지수 골롬(Exponential Golomb), CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary Arithmetic Coding)과 같은 다양한 방법이 적용될 수 있다.
- [92] 엔트로피 복호화부(210)에서는 부호화기에서 수행된 인트라 예측 및 인터 예측에 관련된 정보를 복호화할 수 있다.
- [93] 재정렬부(215)는 엔트로피 복호화부(210)에서 엔트로피 복호화된 비트스트림을 부호화부에서 재정렬한 방법을 기초로 재정렬을 수행할 수 있다. 1차원 벡터 형태로 표현된 계수들을 다시 2차원의 블록 형태의 계수로 복원하여 재정렬할 수 있다. 재정렬부(215)에서는 부호화부에서 수행된 계수 스캐닝에 관련된 정보를 제공받고 해당 부호화부에서 수행된 스캐닝 순서에 기초하여 역으로 스캐닝하는 방법을 통해 재정렬을 수행할 수 있다.
- [94] 역양자화부(220)는 부호화기에서 제공된 양자화 파라미터와 재정렬된 블록의 계수값을 기초로 역양자화를 수행할 수 있다.
- [95] 역변환부(225)는 영상 부호화기에서 수행한 양자화 결과에 대해 변환부에서 수행한 변환 즉, DCT 또는 DST에 대해 역변환 즉, 역 DCT 또는 역 DST를 수행할 수 있다. 여기서, DCT 변환 코어는 DCT2 또는 DCT8 중 적어도 하나를 포함하고, DST 변환 코어는 DST7을 포함할 수 있다. 또는, 영상 부호화기에서 변환이 스킵된 경우, 역변환부(225)에서도 역변환을 수행하지 않을 수 있다. 역변환은 영상 부호화기에서 결정된 전송 단위를 기초로 수행될 수 있다. 영상 복호화기의 역변환부(225)에서는 예측 방법, 현재 블록의 크기 및 예측 방향 등 복수의

- 정보에 따라 변환 기법(예를 들어, DCT 또는 DST)이 선택적으로 수행될 수 있다.
- [96] 예측부(230, 235)는 엔트로피 복호화부(210)에서 제공된 예측 블록 생성 관련 정보와 메모리(245)에서 제공된 이전에 복호화된 블록 또는 픽처 정보를 기초로 예측 블록을 생성할 수 있다.
- [97] 전술한 바와 같이 영상 부호화기에서의 동작과 동일하게 인트라 예측을 수행시 예측 단위의 크기와 변환 단위의 크기가 동일할 경우, 예측 단위의 좌측에 존재하는 픽셀, 좌측 상단에 존재하는 픽셀, 상단에 존재하는 픽셀을 기초로 예측 단위에 대한 인트라 예측을 수행하지만, 인트라 예측을 수행시 예측 단위의 크기와 변환 단위의 크기가 상이할 경우, 변환 단위를 기초로 한 참조 픽셀을 이용하여 인트라 예측을 수행할 수 있다. 또한, 최소 부호화 단위에 대해서만 $N \times N$ 분할을 사용하는 인트라 예측을 사용할 수도 있다.
- [98] 예측부(230, 235)는 예측 단위 판별부, 인터 예측부 및 인트라 예측부를 포함할 수 있다. 예측 단위 판별부는 엔트로피 복호화부(210)에서 입력되는 예측 단위 정보, 인트라 예측 방법의 예측 모드 정보, 인터 예측 방법의 모션 예측 관련 정보 등 다양한 정보를 입력 받고 현재 부호화 단위에서 예측 단위를 구분하고, 예측 단위가 인터 예측을 수행하는지 아니면 인트라 예측을 수행하는지 여부를 판별할 수 있다. 인터 예측부(230)는 영상 부호화기에서 제공된 현재 예측 단위의 인터 예측에 필요한 정보를 이용해 현재 예측 단위가 포함된 현재 픽처의 이전 픽처 또는 이후 픽처 중 적어도 하나의 픽처에 포함된 정보를 기초로 현재 예측 단위에 대한 인터 예측을 수행할 수 있다. 또는, 현재 예측 단위가 포함된 현재 픽처 내에서 기-복원된 일부 영역의 정보를 기초로 인터 예측을 수행할 수도 있다.
- [99] 인터 예측을 수행하기 위해 부호화 단위를 기준으로 해당 부호화 단위에 포함된 예측 단위의 모션 예측 방법이 스킵 모드(Skip Mode), 머지 모드(Merge 모드), 모션 벡터 예측 모드(AMVP Mode), 인트라 블록 커피 모드 중 어떠한 방법인지 여부를 판단할 수 있다.
- [100] 인트라 예측부(235)는 현재 픽처 내의 화소 정보를 기초로 예측 블록을 생성할 수 있다. 예측 단위가 인트라 예측을 수행한 예측 단위인 경우, 영상 부호화기에서 제공된 예측 단위의 인트라 예측 모드 정보를 기초로 인트라 예측을 수행할 수 있다. 인트라 예측부(235)에는 AIS(Adaptive Intra Smoothing) 필터, 참조 화소 보간부, DC 필터를 포함할 수 있다. AIS 필터는 현재 블록의 참조 화소에 필터링을 수행하는 부분으로써 현재 예측 단위의 예측 모드에 따라 필터의 적용 여부를 결정하여 적용할 수 있다. 영상 부호화기에서 제공된 예측 단위의 예측 모드 및 AIS 필터 정보를 이용하여 현재 블록의 참조 화소에 AIS 필터링을 수행할 수 있다. 현재 블록의 예측 모드가 AIS 필터링을 수행하지 않는 모드일 경우, AIS 필터는 적용되지 않을 수 있다.
- [101] 참조 화소 보간부는 예측 단위의 예측 모드가 참조 화소를 보간한 화소값을 기초로 인트라 예측을 수행하는 예측 단위일 경우, 참조 화소를 보간하여 정수값

이하의 화소 단위의 참조 화소를 생성할 수 있다. 현재 예측 단위의 예측 모드가 참조 화소를 보간하지 않고 예측 블록을 생성하는 예측 모드일 경우 참조 화소는 보간되지 않을 수 있다. DC 필터는 현재 블록의 예측 모드가 DC 모드일 경우 필터링을 통해서 예측 블록을 생성할 수 있다.

- [102] 복원된 블록 또는 픽처는 필터부(240)로 제공될 수 있다. 필터부(240)는 디블록킹 필터, 오프셋 보정부, ALF를 포함할 수 있다.
- [103] 영상 부호화기로부터 해당 블록 또는 픽처에 디블록킹 필터를 적용하였는지 여부에 대한 정보 및 디블록킹 필터를 적용하였을 경우, 강한 필터를 적용하였는지 또는 약한 필터를 적용하였는지에 대한 정보를 제공받을 수 있다. 영상 복호화기의 디블록킹 필터에서는 영상 부호화기에서 제공된 디블록킹 필터 관련 정보를 제공받고 영상 복호화기에서 해당 블록에 대한 디블록킹 필터링을 수행할 수 있다.
- [104] 오프셋 보정부는 부호화시 영상에 적용된 오프셋 보정의 종류 및 오프셋 값 정보 등을 기초로 복원된 영상에 오프셋 보정을 수행할 수 있다.
- [105] ALF는 부호화기로부터 제공된 ALF 적용 여부 정보, ALF 계수 정보 등을 기초로 부호화 단위에 적용될 수 있다. 이러한 ALF 정보는 특정한 파라미터 셋에 포함되어 제공될 수 있다.
- [106] 메모리(245)는 복원된 픽처 또는 블록을 저장하여 참조 픽처 또는 참조 블록으로 사용할 수 있도록 할 수 있고 또한 복원된 픽처를 출력부로 제공할 수 있다.
- [107]
- [108] 도 3은 본 발명의 일 실시예에 따른 기본 코딩 트리 유닛을 도시한 도면이다.
- [109] 최대 크기의 코딩 블록을 코딩 트리 블록이라 정의할 수 있다. 하나의 픽처는 복수개의 코딩 트리 유닛(Coding Tree Unit, CTU)으로 분할된다. 코딩 트리 유닛은 최대 크기의 코딩 유닛으로, LCU (Largest Coding Unit)라 호칭될 수도 있다. 도 3은 하나의 픽처가 복수개의 코딩 트리 유닛으로 분할된 예를 나타낸 것이다.
- [110] 코딩 트리 유닛의 크기는 픽처 레벨 또는 시퀀스 레벨에서 정의될 수 있다. 이를 위해, 코딩 트리 유닛의 크기를 나타내는 정보가 픽처 파라미터 세트 또는 시퀀스 파라미터 세트를 통해 시그널링될 수 있다.
- [111] 일 예로, 시퀀스 내 전체 픽처에 대한 코딩 트리 유닛의 크기가 128x128로 설정될 수 있다. 또는, 픽처 레벨에서 128x128 또는 256x256 중 어느 하나를 코딩 트리 유닛의 크기로 결정할 수 있다. 일 예로, 제1 픽처에서는 코딩 트리 유닛의 크기가 128x128로 설정되고, 제2 픽처에서는 코딩 트리 유닛의 크기가 256x256으로 설정될 수 있다.
- [112] 코딩 트리 유닛을 분할하여, 코딩 블록을 생성할 수 있다. 코딩 블록은 부호화/복호화 처리를 위한 기본 단위를 나타낸다. 일 예로, 코딩 블록 별로 예측 또는 변환이 수행되거나, 코딩 블록 별로 예측 부호화 모드가 결정될 수 있다.

여기서, 예측 부호화 모드는 예측 영상을 생성하는 방법을 나타낸다. 일 예로, 예측 부호화 모드는 화면 내 예측(Intra Prediction, 인트라 예측), 화면 간 예측(Inter Prediction, 인터 예측), 현재 픽처 참조(Current Picture Referencing, CPR, 또는 인트라 블록 카피(Intra Block Copy, IBC)) 또는 복합 예측(Combined Prediction)을 포함할 수 있다. 코딩 블록에 대해, 인트라 예측, 인터 예측, 현재 픽처 참조 또는 복합 예측 중 적어도 하나의 예측 부호화 모드를 이용하여, 코딩 블록에 대한 예측 블록을 생성할 수 있다.

- [113] 현재 블록의 예측 부호화 모드를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 상기 정보는 예측 부호화 모드가 인트라 모드인지 또는 인터 모드인지 여부를 나타내는 1비트 플래그일 수 있다. 현재 블록의 예측 부호화 모드가 인터 모드로 결정된 경우에 한하여, 현재 픽처 참조 또는 복합 예측이 이용 가능할 수 있다.
- [114] 현재 픽처 참조는 현재 픽처를 참조 픽처로 설정하고, 현재 픽처 내 이미 부호화/복호화가 완료된 영역으로부터 현재 블록의 예측 블록을 획득하기 위한 것이다. 여기서, 현재 픽처는 현재 블록을 포함하는 픽처를 의미한다. 현재 블록에 현재 픽처 참조가 적용되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 상기 정보는 1비트의 플래그일 수 있다. 상기 플래그가 참인 경우, 현재 블록의 예측 부호화 모드는 현재 픽처 참조로 결정되고, 상기 플래그가 거짓인 경우, 현재 블록의 예측 모드는 인터 예측으로 결정될 수 있다.
- [115] 또는, 참조 픽처 인덱스를 기초로, 현재 블록의 예측 부호화 모드가 결정될 수 있다. 일 예로, 참조 픽처 인덱스가 현재 픽처를 가리키는 경우, 현재 블록의 예측 부호화 모드는 현재 픽처 참조로 결정될 수 있다. 참조 픽처 인덱스가 현재 픽처가 아닌 다른 픽처를 가리키는 경우, 현재 블록의 예측 부호화 모드는 인터 예측으로 결정될 수 있다. 즉, 현재 픽처 참조는 현재 픽처 내 부호화/복호화가 완료된 영역의 정보를 이용한 예측 방법이고, 인터 예측은 부호화/복호화가 완료된 다른 픽처의 정보를 이용한 예측 방법이다.
- [116] 복합 예측은 인트라 예측, 인터 예측 및 현재 픽처 참조 중 둘 이상을 조합된 부호화 모드를 나타낸다. 일 예로, 복합 예측이 적용되는 경우, 인트라 예측, 인터 예측 또는 현재 픽처 참조 중 어느 하나를 기초로 제1 예측 블록이 생성되고, 다른 하나를 기초로 제2 예측 블록이 생성될 수 있다. 제1 예측 블록 및 제2 예측 블록이 생성되면, 제1 예측 블록 및 제2 예측 블록의 평균 연산 또는 가중합 연산을 통해 최종 예측 블록이 생성될 수 있다. 복합 예측이 적용되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그일 수 있다.
- [117] 도 4는 코딩 블록의 다양한 분할 형태를 나타낸 도면이다.
- [118] 코딩 블록은 쿼드 트리 분할, 바이너리 트리 분할 또는 트리플 트리 분할을 기초로 복수의 코딩 블록들로 분할될 수 있다. 분할된 코딩 블록도 다시 쿼드

트리 분할, 바이너리 트리 분할 또는 트리플 트리 분할을 기초로 다시 복수의 코딩 블록들로 분할될 수 있다.

- [119] 퀴드 트리 분할은 현재 블록을 4개의 블록들로 분할하는 분할 기법을 나타낸다. 퀴드 트리 분할의 결과, 현재 블록은 4개의 정방 형태 파티션들로 분할될 수 있다 (도 4의 (a) 'SPLIT_QT' 참조).
- [120] 바이너리 트리 분할은 현재 블록을 2개의 블록들로 분할하는 분할 기법을 나타낸다. 수직 방향을 따라(즉, 현재 블록을 가로지르는 수직선을 이용) 현재 블록을 두개의 블록들로 분할하는 것을 수직 방향 바이너리 트리 분할이라 호칭할 수 있고, 수평 방향을 따라(즉, 현재 블록을 가로지르는 수평선을 이용) 현재 블록을 두개의 블록들로 분할하는 것을 수평 방향 바이너리 트리 분할이라 호칭할 수 있다. 바이너리 트리 분할 결과, 현재 블록은 2개의 비정방 형태 파티션들로 분할될 수 있다. 도 4의 (b) 'SPLIT_BT_VER'는 수직 방향 바이너리 트리 분할 결과를 나타낸 것이고, 도 4의 (c) 'SPLIT_BT_HOR'는 수평 방향 바이너리 트리 분할 결과를 나타낸 것이다.
- [121] 트리플 트리 분할은 현재 블록을 3개의 블록들로 분할하는 분할 기법을 나타낸다. 수직 방향을 따라(즉, 현재 블록을 가로지르는 두개의 수직선을 이용) 현재 블록을 세개의 블록들로 분할하는 것을 수직 방향 트리플 트리 분할이라 호칭할 수 있고, 수평 방향을 따라(즉, 현재 블록을 가로지르는 두개의 수평선을 이용) 현재 블록을 세개의 블록들로 분할하는 것을 수평 방향 트리플 트리 분할이라 호칭할 수 있다. 트리플 트리 분할 결과, 현재 블록은 3개의 비정방 형태 파티션들로 분할될 수 있다. 이때, 현재 블록의 중앙에 위치하는 파티션의 너비/높이는 다른 파티션들의 너비/높이 대비 2배일 수 있다. 도 4의 (d) 'SPLIT_TT_VER'는 수직 방향 트리플 트리 분할 결과를 나타낸 것이고, 도 4의 (e) 'SPLIT_TT_HOR'는 수평 방향 트리플 트리 분할 결과를 나타낸 것이다.
- [122] 코딩 트리 유닛의 분할 횟수를 분할 깊이(Partitioning Depth)라 정의할 수 있다. 시퀀스 또는 픽처 레벨에서 코딩 트리 유닛의 최대 분할 깊이가 결정될 수 있다. 이에 따라, 시퀀스 또는 필처별로 코딩 트리 유닛의 최대 분할 깊이가 상이할 수 있다.
- [123] 또는, 분할 기법들 각각에 대한 최대 분할 깊이를 개별적으로 결정할 수 있다. 일 예로, 퀴드 트리 분할이 허용되는 최대 분할 깊이는 바이너리 트리 분할 및/또는 트리플 트리 분할이 허용되는 최대 분할 깊이와 상이할 수 있다.
- [124] 부호화기는 현재 블록의 분할 형태 또는 분할 깊이 중 적어도 하나를 나타내는 정보를 비트스트림을 통해 시그널링할 수 있다. 부호화기는 비트스트림으로부터 파싱되는 상기 정보에 기초하여 코딩 트리 유닛의 분할 형태 및 분할 깊이를 결정할 수 있다.
- [125] 도 5는 코딩 트리 유닛의 분할 양상을 예시한 도면이다.
- [126] 퀴드 트리 분할, 바이너리 트리 분할 및/또는 트리플 트리 분할 등의 분할 기법을 이용하여 코딩 블록을 분할하는 것을 멀티 트리 분할(Multi Tree

Partitioning)이라 호칭할 수 있다.

- [127] 코딩 블록에 멀티 트리 분할을 적용하여 생성되는 코딩 블록들을 하위 코딩 블록들이라 호칭할 수 있다. 코딩 블록의 분할 깊이가 k 인 경우, 하위 코딩 블록들의 분할 깊이는 $k+1$ 로 설정된다.
- [128] 반대로, 분할 깊이가 $k+1$ 인 코딩 블록들에 대해, 분할 깊이가 k 인 코딩 블록을 상위 코딩 블록이라 호칭할 수 있다.
- [129] 현재 코딩 블록의 분할 타입은 상위 코딩 블록의 분할 형태 또는 이웃 코딩 블록의 분할 타입 중 적어도 하나를 기초로 결정될 수 있다. 여기서, 이웃 코딩 블록은 현재 코딩 블록에 인접하는 것으로, 현재 코딩 블록의 상단 이웃 블록, 좌측 이웃 블록, 또는 좌측 상단 코너에 인접하는 이웃 블록 중 적어도 하나를 포함할 수 있다. 여기서, 분할 타입은, 쿼드 트리 분할 여부, 바이너리 트리 분할 여부, 바이너리 트리 분할 방향, 트리플 트리 분할 여부, 또는 트리플 트리 분할 방향 중 적어도 하나를 포함할 수 있다.
- [130] 코딩 블록의 분할 형태를 결정하기 위해, 코딩 블록이 분할되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 'split_cu_flag'로, 상기 플래그가 참인 것은, 머리 트리 분할 기법에 의해 코딩 블록이 분할됨을 나타낸다.
- [131] split_cu_flag가 참인 경우, 코딩 블록이 쿼드 트리 분할되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 split_qt_flag로, 상기 플래그가 참인 경우, 코딩 블록은 4개의 블록들로 분할될 수 있다.
- [132] 일 예로, 도 5에 도시된 예에서는, 코딩 트리 유닛이 쿼드 트리 분할됨에 따라, 분할 깊이가 1인 4개의 코딩 블록들이 생성되는 것으로 도시되었다. 또한, 쿼드 트리 분할 결과로 생성된 4개의 코딩 블록들 중 첫번째 코딩 블록 및 네번째 코딩 블록에 다시 쿼드 트리 분할이 적용된 것으로 도시되었다. 그 결과, 분할 깊이가 2인 4개의 코딩 블록들이 생성될 수 있다.
- [133] 또한, 분할 깊이가 2인 코딩 블록에 다시 쿼드 트리 분할을 적용함으로써, 분할 깊이가 3인 코딩 블록을 생성할 수 있다.
- [134] 코딩 블록에 쿼드 트리 분할이 적용되지 않는 경우, 코딩 블록의 크기, 코딩 블록이 픽처 경계에 위치하는지 여부, 최대 분할 깊이 또는 이웃 블록의 분할 형태 중 적어도 하나를 고려하여, 상기 코딩 블록에 바이너리 트리 분할 또는 트리플 트리 분할을 수행할 것인지 여부를 결정할 수 있다. 상기 코딩 블록에 바이너리 트리 분할 또는 트리플 트리 분할이 수행되는 것으로 결정된 경우, 분할 방향을 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 mtt_split_cu_vertical_flag일 수 있다. 상기 플래그에 기초하여, 분할 방향이 수직 방향인지 또는 수평 방향인지 여부가 결정될 수 있다. 추가로, 바이너리 트리 분할 또는 트리플 트리 분할 중 어느 것이 상기 코딩 블록에 적용되는지를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다.

상기 정보는 1비트의 플래그 `mtt_split_cu_binary_flag`일 수 있다. 상기 플래그에 기초하여, 상기 코딩 블록에 바이너리 트리 분할이 적용되는지 또는 트리플 트리 분할이 적용되는지 여부가 결정될 수 있다.

- [135] 일 예로, 도 5에 도시된 예에서는, 분할 깊이 1인 코딩 블록에 수직 방향 바이너리 트리 분할이 적용되고, 상기 분할 결과로 생성된 코딩 블록들 중 좌측 코딩 블록에는 수직 방향 트리플 트리 분할이 적용되고, 우측 코딩 블록에는 수직 방향 바이너리 트리 분할이 적용된 것으로 도시되었다.
- [136] 영상을 부호화 또는 복호화하는 장치를 구현할 때, 하드웨어 성능 상 문턱값보다 큰 영역은 처리가 어려운 문제가 발생한다. 일 예로, 하드웨어 성능 상 최대 4096개의 샘플의 동시 처리가 가능한 경우, 4096개보다 더 많은 샘플을 갖는 영역에 대해서는, 64x64 크기의 데이터 유닛을 중복 액세스하여 처리해야 문제점 및 데이터를 동시 처리할 수 없다는 문제점이 발생할 수 있다. 이처럼, 데이터 처리의 기본 유닛을 파이프 라인 기반 데이터 기본 유닛(Virtual Processing Data Unit, VPDU, 이하, 데이터 기본 유닛이라 함)이라 정의할 수 있다.
- [137] 데이터 기본 유닛은 정방형, 비정방형 또는 비-사각형 타입으로 분류될 수 있다.
- [138] 도 6은 데이터 기본 유닛의 형태를 나타낸 도면이다.
- [139] 데이터 기본 유닛들은 동시에 처리 가능한 최대 샘플의 개수와 같거나 작은 개수의 샘플을 포함할 수 있다. 일 예로, 도 6의 (a)에 도시된 예에서와 같이, 64x64 크기의 정방형 블록들을 데이터 기본 유닛으로 설정할 수 있다. 또는, 비정방형 블록이 데이터 기본 유닛으로 설정될 수 있다. 일 예로, 도 6의 (b) 또는 도 6의 (c)에 도시된 예에서와 같이, 32x128 크기의 블록 또는 64x32 크기의 블록이 데이터 기본 유닛으로 설정될 수 있다.
- [140] 도시되지는 않았지만, 삼각형, L자 형태(L-shaped) 또는 다각 형태의 데이터 기본 유닛이 정의될 수도 있다.
- [141] 데이터 기본 유닛을 결정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 데이터 기본 유닛의 크기 또는 형태 중 적어도 하나를 결정하기 위한 것일 수 있다. 상기 정보에 기초하여, 비정방형 데이터 기본 유닛의 허용 여부 또는 비-사각 형태 데이터 기본 유닛의 허용 여부가 결정될 수 있다.
- [142] 또는, 부호화기 및 복호화기에서 데이터 기본 유닛의 크기 또는 형태 중 적어도 하나가 기 정의되어 있을 수 있다.
- [143] 데이터 기본 유닛의 크기를 고려하여, 코딩 블록의 분할 형태의 허용 여부가 결정될 수 있다. 일 예로, 코딩 블록의 분할 결과 생성되는 코딩 블록이 데이터 기본 유닛보다 큰 경우, 상기 분할을 허용하지 않을 수 있다. 또는, 코딩 블록의 분할 결과 생성되는 비정방 형태의 코딩 블록이 데이터 기본 유닛보다 큰 경우, 상기 분할을 허용하지 않을 수 있다. 일 예로, 코딩 블록의 너비 또는 높이가 문턱값보다 큰 경우, 또는 코딩 블록에 포함된 샘플 수가 문턱값보다 큰 경우,

바이너리 트리 또는 트리플 트리 분할을 허용하지 않을 수 있다. 이에 따라, 바이너리 트리 또는 트리플 트리 분할과 관련된 정보에 대한 부호화가 생략될 수 있다.

- [144] 또는, 데이터 기본 유닛보다 큰 코딩 블록은 반드시 분할되도록 설정될 수 있다. 또는, 데이터 기본 유닛보다 큰 코딩 블록은 반드시 바이너리 트리 분할 또는 트리플 트리 분할이 수행되도록 설정될 수 있다. 이에 따라, 데이터 기본 유닛보다 큰 코딩 블록에 대해서는 코딩 블록의 분할 여부를 나타내는 플래그 `split_flag`가 부호화되지 않더라도, 상기 플래그의 값이 1로 유도될 수 있다.
- [145] 다른 예로, 데이터 기본 유닛보다 큰 코딩 블록을 복수의 서브 블록들로 분할할 수 있다. 여기서, 서브 블록은 예측을 위한 기본 단위인 예측 유닛 또는 변환 및/또는 양자화를 위한 기본 단위인 변환 유닛으로 설정될 수 있다. 이때, 코딩 블록을 복수의 예측 유닛들로 분할하는 것은 VPDU 예측 유닛 파티셔닝으로 정의되고, 코딩 블록을 복수의 변환 유닛들로 분할하는 것은 VPDU 변환 유닛 파티셔닝으로 정의될 수 있다.
- [146] 코딩 블록에는 VPDU 예측 유닛 파티셔닝 또는 VPDU 변환 유닛 파티셔닝 중 적어도 하나가 적용될 수 있다. VPDU 예측 유닛 파티셔닝 적용에 따른 코딩 블록의 분할 형태는 VPDU 변환 유닛 파티셔닝 적용에 따른 코딩 블록의 분할 형태와 동일하게 설정될 수 있다.
- [147] 코딩 블록에 VPDU 예측 유닛 파티셔닝만이 적용되는 경우, 서브 블록 별로 예측이 수행되나, 변환 및/또는 양자화는 코딩 블록을 대상으로 수행될 수 있다. 이때, 예측 부호화 모드, 인트라 예측 모드 또는 인터 예측 모드와 같은 예측 모드는 코딩 블록을 대상으로 결정될 수 있다.
- [148] 코딩 블록에 VPDU 변환 유닛 파티셔닝만이 적용되는 경우, 서브 블록을 대상으로 예측이 수행되나, 변환 및/또는 양자화는 서브 블록 별로 수행될 수 있다.
- [149] 도 7 및 도 8은 코딩 블록이 복수의 서브 블록들로 분할되는 예를 나타낸 도면이다.
- [150] 도 7은 정방형 데이터 기본 유닛만이 허용되는 경우의 분할 양상을 나타낸 도면이고, 도 8은 정방형 데이터 기본 유닛과 비정방형 데이터 기본 유닛이 허용되는 경우의 분할 양상을 나타낸 도면이다.
- [151] 정방형 데이터 기본 유닛만이 허용된다고 가정할 경우, 도 7의 (a) 및 (b)에서, CU0 및 CU3은 서로 다른 2개의 VPDU로 정의되고, CU1은 서로 다른 4개의 VPDU로 정의된다. 이에 따라, CU0 및 CU3을 2개의 서브 블록들로 분할하고, CU1을 4개의 서브 블록들로 분할할 수 있다.
- [152] 정방형 데이터 기본 유닛과 비정방형 데이터 기본 유닛이 허용된다고 가정할 경우, 도 8의 (a) 및 (b)에서, CU0 및 CU3은 1개의 VPDU로 정의 가능한 반면, CU1은 서로 다른 2개의 VPDU들을 이용하여 정의될 수 있다. 이에 따라, CU0 및 CU3은 서브 블록들로 분할되지 않는 반면, CU1은 2개의 서브 블록들로 분할될

수 있다.

- [153] 이때, CU1은 정방향 서브 블록들 또는 비정방향 서브 블록들로 분할될 수 있다. 일 예로, CU1을 상하로 분할하는 수평선에 기초하여, CU1을 2개의 정방향 서브 블록들로 분할할 수 있다. 또는 CU1을 좌우로 분할하는 수직선에 기초하여, CU1을 2개의 비정방향 서브 블록들로 분할할 수 있다.
- [154] 코딩 블록에 적용될 수 있는 복수의 분할 타입 후보들이 존재하는 경우, 복수의 분할 타입 후보들 중 어느 하나를 나타내는 정보를 비트스트림을 통해 시그널링될 수 있다. 일 예로, 상기 정보는 코딩 블록이 정방향 서브 블록들로 분할되는지 여부 또는 코딩 블록이 비정방향 서브 블록들로 분할되는지 여부를 나타낼 수 있다.
- [155] 또는, 코딩 블록을 정방향 서브 블록들로 분할하는 것이 코딩 블록을 비정방향 서브 블록들로 분할하는 것보다 높은 우선순위를 갖도록 설정할 수 있다. 예컨대, 코딩 블록을 비정방향 서브 블록들로 분할하는 것은, 코딩 블록을 정방향 서브 블록들로 분할할 수 없을 때 허용될 수 있다.
- [156] 또는, 부모 노드인 코딩 블록의 분할 형태에 기초하여, 코딩 블록의 분할 형태를 결정할 수 있다. 일 예로, 부모 노드인 코딩 블록이 트리플 트리에 기반하여 분할된 경우, 코딩 블록이 정방향 서브 블록들로 분할되도록 설정할 수 있다. 반면, 부모 노드인 코딩 블록이 바이너리 트리 또는 트리플 트리에 기반하여 분할된 경우, 코딩 블록이 비정방향 서브 블록들로 분할되도록 설정할 수 있다.
- [157]
- [158] 인터 예측은 이전 픽처의 정보를 이용하여, 현재 블록을 예측하는 예측 부호화 모드이다. 일 예로, 이전 픽처 내 현재 블록과 동일한 위치의 블록(이하, 콜로케이티드 블록, Collocated block)을 현재 블록의 예측 블록으로 설정할 수 있다. 이하, 현재 블록과 동일한 위치의 블록을 기초로 생성된 예측 블록을 콜로케이티드 예측 블록(Collocated Prediction Block)이라 호칭하기로 한다.
- [159] 반면, 이전 픽처에 존재한 오브젝트가 현재 픽처에서는 다른 위치로 이동하였다면, 오브젝트의 움직임을 이용하여 효과적으로 현재 블록을 예측할 수 있다. 예를 들어, 이전 픽처와 현재 픽처를 비교함으로써 오브젝트의 이동 방향 및 크기를 알 수 있다면, 오브젝트의 움직임 정보를 고려하여 현재 블록의 예측 블록(또는, 예측 영상)을 생성할 수 있다. 이하, 움직임 정보를 이용하여 생성된 예측 블록을 움직임 예측 블록이라 호칭할 수 있다.
- [160] 현재 블록에서 예측 블록을 차분하여, 잔차 블록(residual block)을 생성할 수 있다. 이때, 오브젝트의 움직임이 존재하는 경우라면, 콜로케이티드 예측 블록 대신 움직임 예측 블록을 이용함으로써, 잔차 블록의 에너지를 줄이고, 이에 따라, 잔차 블록의 압축 성능을 향상시킬 수 있다.
- [161] 위치럼, 움직임 정보를 이용하여 예측 블록을 생성하는 것을 움직임 보상 예측이라 호칭할 수 있다. 대부분의 인터 예측에서는 움직임 보상 예측에 기초하여 예측 블록을 생성할 수 있다.

- [162] 움직임 정보는 모션 벡터, 참조 픽처 인덱스, 예측 방향 또는 양방향 가중치 인덱스 중 적어도 하나를 포함할 수 있다. 모션 벡터는 오브젝트의 이동 방향 및 크기를 나타낸다. 참조 픽처 인덱스는 참조 픽처 리스트에 포함된 참조 픽처들 중 현재 블록의 참조 픽처를 특정한다. 예측 방향은 단방향 L0 예측, 단방향 L1 예측 또는 양방향 예측(L0 예측 및 L1 예측) 중 어느 하나를 가리킨다. 현재 블록의 예측 방향에 따라, L0 방향의 움직임 정보 또는 L1 방향의 움직임 정보 중 적어도 하나가 이용될 수 있다. 양방향 가중치 인덱스는 L0 예측 블록에 적용되는 가중치 및 L1 예측 블록에 적용되는 가중치를 특정한다.
- [163] 도 9는 본 발명의 일 실시예에 따른 인터 예측 방법의 흐름도이다.
- [164] 도 9를 참조하면, 인터 예측 방법은, 현재 블록의 인터 예측 모드를 결정하는 단계(S901), 결정된 인터 예측 모드에 따라 현재 블록의 움직임 정보를 획득하는 단계(S902) 및 획득된 움직임 정보에 기초하여, 현재 블록에 대한 움직임 보상 예측을 수행하는 단계(S903)를 포함한다.
- [165] 여기서, 인터 예측 모드는 현재 블록의 움직임 정보를 결정하기 위한 다양한 기법들을 나타내는 것으로, 병진(Translation) 움직임 정보를 이용하는 인터 예측 모드와, 어파인(Affine) 움직임 정보를 이용하는 인터 예측 모드를 포함할 수 있다. 일 예로, 병진 움직임 정보를 이용하는 인터 예측 모드는, 머지 모드 및 모션 벡터 예측 모드를 포함하고, 어파인 움직임 정보를 이용하는 인터 예측 모드는 어파인 머지 모드 및 어파인 모션 벡터 예측 모드를 포함할 수 있다. 현재 블록의 움직임 정보는, 인터 예측 모드에 따라, 현재 블록에 이웃하는 이웃 블록 또는 비트스트림으로부터 파싱되는 정보를 기초로 결정될 수 있다.
- [166] 이하, 어파인 움직임 정보를 이용하는 인터 예측 방법에 대해 상세히 설명하기로 한다.
- [167] 도 10은 오브젝트의 비선형 움직임을 예시한 도면이다.
- [168] 영상 내 물체의 움직임이 선형적이지 않은 움직임이 발생할 수 있다. 일 예로, 도 10에 도시된 예에서와 같이, 카메라 줌인(Zoom-in), 줌 아웃(Zoom-out), 회전(Rotation) 또는 어파인 변환 등 오브젝트의 비선형 움직임이 발생할 수 있다. 오브젝트의 비선형 움직임이 발생한 경우, 병진 움직임 벡터로는 오브젝트의 움직임을 효과적으로 표현할 수 없다. 이에 따라, 오브젝트의 비선형 움직임이 발생하는 부분에서는 병진 움직임 대신 어파인 움직임을 이용하여 부호화 효율을 향상시킬 수 있다.
- [169] 도 11은 본 발명의 일 실시예에 따른 어파인 모션에 기초한 인터 예측 방법의 흐름도이다.
- [170] 현재 블록에 어파인 모션에 기초한 인터 예측 기법이 적용되는지 여부는, 비트스트림으로부터 파싱되는 정보에 기초하여 결정될 수 있다. 구체적으로, 현재 블록에 어파인 머지 모드가 적용되는지 여부를 나타내는 플래그 또는 현재 블록에 어파인 모션 벡터 예측 모드가 적용되는지 여부를 나타내는 플래그 중 적어도 하나에 기초하여, 현재 블록에 어파인 모션에 기초한 인터 예측 기법이

적용되는지 여부를 결정할 수 있다.

[171] 현재 블록에 어파인 모션에 기초한 인터 예측 기법이 적용되는 경우, 현재 블록의 어파인 모션 모델을 결정할 수 있다(S1101). 어파인 모션 모델은 6 파라미터 어파인 모션 모델 또는 4 파라미터 어파인 모션 모델 중 적어도 하나로 결정될 수 있다. 6 파라미터 어파인 모션 모델은 6개의 파라미터를 이용하여 어파인 모션을 표현한 것이고, 4 파라미터 어파인 모션 모델은 4개의 파라미터를 이용하여 어파인 모션을 표현한 것이다.

[172] 수식식 1은 6 파라미터를 이용하여 어파인 모션을 표현한 것이다. 어파인 모션은 어파인 시드 벡터들에 의해 결정되는 소정 영역에 대한 병진 움직임을 나타낸다.

[173] [수식1]

$$v_x = ax - by + e$$

$$v_y = cx + dy + f$$

[174] 6개의 파라미터를 이용하여 어파인 모션을 표현하는 경우, 복잡한 움직임을 표현할 수 있으나, 각 파라미터들을 부호화하는데 필요한 비트 수가 많아져 부호화 효율이 저하될 수 있다. 이에 따라, 4개의 파라미터를 이용하여 어파인 모션을 표현할 수도 있다. 수식식 2는 4 파라미터를 이용하여 어파인 모션을 표현한 것이다.

[175] [수식2]

$$v_x = ax - by + e$$

$$v_y = bx + ay + f$$

[176] 현재 블록의 어파인 모션 모델을 결정하기 위한 정보가 부호화되어 비트스트림을 통해 시그널링될 수 있다. 일 예로, 상기 정보는 1비트의 플래그 'affine_type_flag'일 수 있다. 상기 플래그의 값이 0인 것은 4 파라미터 어파인 모션 모델이 적용됨을 나타내고, 상기 플래그의 값이 1인 것은 6 파라미터 어파인 모션 모델이 적용됨을 나타낼 수 있다. 상기 플래그는, 슬라이스, 타일 또는 블록(예컨대, 코딩 블록 또는 코딩 트리 유닛) 단위로 부호화될 수 있다. 슬라이스 레벨에서 플래그가 시그널링되는 경우, 상기 슬라이스 레벨에서 결정된 어파인 모션 모델이 상기 슬라이스에 속한 블록들 모두에 적용될 수 있다.

[177] 또는, 현재 블록의 어파인 인터 예측 모드에 기초하여, 현재 블록의 어파인 모션 모델을 결정할 수 있다. 일 예로, 어파인 머지 모드가 적용되는 경우, 현재 블록의 어파인 모션 모델을 4 파라미터 모션 모델로 결정할 수 있다. 반면, 어파인 모션 벡터 예측 모드가 적용되는 경우, 현재 블록의 어파인 모션 모델을 결정하기 위한 정보를 부호화하여 비트스트림을 통해 시그널링될 수 있다. 일 예로, 현재 블록에 어파인 모션 벡터 예측 모드가 적용되는 경우, 1비트의 플래그 'affine_type_flag'에 기초하여, 현재 블록의 어파인 모션 모델을 결정할 수 있다.

[178] 다음으로, 현재 블록의 어파인 시드 벡터를 유도할 수 있다(S1102). 4 파라미터

어파인 모션 모델이 선택된 경우, 현재 블록의 두 컨트롤 포인트에서의 모션 벡터들을 유도할 수 있다. 반면, 6 파라미터 어파인 모션 모델이 선택된 경우, 현재 블록의 세 컨트롤 포인트에서의 모션 벡터들이 유도될 수 있다. 컨트롤 포인트에서의 모션 벡터를 어파인 시드 벡터라 호칭할 수 있다. 컨트롤 포인트는, 현재 블록의 좌상단 코너, 우상단 코너 또는 좌하단 코너 중 적어도 하나를 포함할 수 있다.

- [179] 도 12는 어파인 모션 모델 별 어파인 시드 벡터를 예시한 도면이다.
- [180] 4 파라미터 어파인 모션 모델에서는 좌상단 코너, 우상단 코너 또는 좌하단 코너 중 두개에 대한 어파인 시드 벡터들을 유도할 수 있다. 일 예로, 도 12의 (a)에 도시된 예에서와 같이, 4 파라미터 어파인 모션 모델이 선택된 경우, 현재 블록의 좌상단 코너(예컨대, 좌상단 샘플 (x_1, y_1))에 대한 어파인 시드 벡터 sv_0 와 현재 블록의 우상단 코너(예컨대, 우상단 샘플 (x_1, y_1))에 대한 어파인 시드 벡터 sv_1 을 이용하여 어파인 벡터를 유도할 수 있다. 좌상단 코너에 대한 어파인 시드 벡터 대신 좌하단 코너에 대한 어파인 시드 벡터를 사용하거나, 우상단 코너에 대한 어파인 시드 벡터 대신 좌하단 코너에 대한 어파인 시드 벡터를 사용하는 것도 가능하다.
- [181] 6 파라미터 어파인 모션 모델에서는 좌상단 코너, 우상단 코너 및 좌하단 코너에 대한 어파인 시드 벡터들을 유도할 수 있다. 일 예로, 도 12의 (b)에 도시된 예에서와 같이, 6 파라미터 어파인 모션 모델이 선택된 경우, 현재 블록의 좌상단 코너(예컨대, 좌상단 샘플 (x_1, y_1))에 대한 어파인 시드 벡터 sv_0 , 현재 블록의 우상단 코너(예컨대, 우상단 샘플 (x_1, y_1))에 대한 어파인 시드 벡터 sv_1 및 현재 블록의 좌상단 코너(예컨대, 좌상단 샘플 (x_2, y_2))에 대한 어파인 시드 벡터 sv_2 를 이용하여 어파인 벡터를 유도할 수 있다.
- [182] 후술되는 실시예에서는, 4 파라미터 어파인 모션 모델 하에서, 좌상단 컨트롤 포인트 및 우상단 컨트롤 포인트의 어파인 시드 벡터들을 각각 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터라 호칭하기로 한다. 후술되는 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 이용하는 실시예들에서, 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터 중 적어도 하나는 좌하단 컨트롤 포인트의 어파인 시드 벡터(제3 어파인 시드 벡터) 또는 우하단 컨트롤 포인트의 어파인 시드 벡터(제4 어파인 시드 벡터)로 대체될 수 있다.
- [183] 또한, 6 파라미터 어파인 모션 모델 하에서, 좌상단 컨트롤 포인트, 우상단 컨트롤 포인트 및 좌하단 컨트롤 포인트의 어파인 시드 벡터들을 각각 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 및 제3 어파인 시드 벡터라 호칭하기로 한다. 후술되는 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 및 제3 어파인 시드 벡터를 이용하는 실시예들에서, 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 및 제3 어파인 시드 벡터 중 적어도 하나는 우하단 컨트롤 포인트의 어파인 시드 벡터(제4 어파인 시드 벡터)로 대체될 수 있다.
- [184] 어파인 시드 벡터들을 이용하여 서브 블록 별로 어파인 벡터를 유도할 수

있다(S1103). 여기서, 어파인 벡터는 어파인 시드 벡터들을 기초로 유도되는 병진 모션 벡터를 나타낸다. 서브 블록의 어파인 벡터를 어파인 서브 블록 모션 벡터 또는 서브 블록 모션 벡터라 호칭할 수 있다.

[185] 도 13은 4 파라미터 모션 모델 하에서 서브 블록들의 어파인 벡터들을 예시한 도면이다.

[186] 서브 블록의 어파인 벡터는 컨트롤 포인트의 위치, 서브 블록의 위치 및 어파인 시드 벡터를 기초로 유도될 수 있다. 일 예로, 수학식 3은 어파인 서브 블록 벡터를 유도하는 예를 나타낸다.

[187] [수식3]

$$v_x = \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)} (x - x_0) - \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)} (y - y_0) + sv_{0x}$$

$$v_y = \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)} (x - x_0) - \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)} (y - y_0) + sv_{0y}$$

[188] 상기 수학식 3에서 (x, y)는 서브 블록의 위치를 나타낸다. 여기서, 서브 블록의 위치는 서브 블록에 포함된 기준 샘플의 위치를 나타낸다. 기준 샘플은 서브 블록의 좌상단 코너에 위치하는 샘플, 또는 x축 또는 y축 좌표 중 적어도 하나가 중앙 위치인 샘플일 수 있다. (x₀, y₀)는 제1 컨트롤 포인트의 위치를 나타내고, (sv_{0x}, sv_{0y})는 제1 어파인 시드 벡터를 나타낸다. 또한, (x₁, y₁)은 제2 컨트롤 포인트의 위치를 나타내고, (sv_{1x}, sv_{1y})는 제2 어파인 시드 벡터를 나타낸다.

[189] 제1 컨트롤 포인트 및 제2 컨트롤 포인트가 각각 현재 블록의 좌상단 코너 및 우상단 코너에 대응하는 경우, x₁-x₀는 현재 블록의 너비와 동일한 값으로 설정될 수 있다.

[190] 이후, 각 서브 블록의 어파인 벡터를 이용하여, 각 서브 블록에 대한 움직임 보상 예측을 수행할 수 있다(S1104). 움직임 보상 예측의 수행 결과, 각 서브 블록에 대한 예측 블록이 생성될 수 있다. 서브 블록들의 예측 블록들이, 현재 블록의 예측 블록으로 설정될 수 있다.

[191] 다음으로, 병진 움직임 정보를 이용하는 인터 예측 방법에 대해 상세히 설명하기로 한다.

[192] 현재 블록의 움직임 정보는 현재 블록 타 블록의 움직임 정보로부터 유도될 수 있다. 여기서, 타 블록은 현재 블록보다 앞서 인터 예측으로 부호화/복호화된 블록일 수 있다. 현재 블록의 움직임 정보를 타 블록의 움직임 정보와 동일하게 설정하는 것을 머지 모드라 정의할 수 있다. 또한, 타 블록의 움직임 벡터를 현재 블록의 움직임 벡터의 예측값으로 설정하는 것을 모션 벡터 예측 모드라 정의할 수 있다.

[193] 도 14는 머지 모드 하에서 현재 블록의 움직임 정보를 유도하는 과정의 흐름도이다.

[194] 현재 블록의 머지 후보를 유도할 수 있다(S1401). 현재 블록의 머지 후보는 현재

블록보다 앞서 인터 예측으로 부호화/복호화된 블록으로부터 유도될 수 있다.

- [195] 도 15는 머지 후보를 유도하기 위해 사용되는 후보 블록들을 예시한 도면이다.
- [196] 후보 블록들은, 현재 블록에 인접하는 샘플을 포함하는 이웃 블록들 또는 현재 블록에 인접하지 않는 샘플을 포함하는 비이웃 블록들 중 적어도 하나를 포함할 수 있다. 이하, 후보 블록들을 결정하는 샘플들을 기준 샘플들이라 정의한다. 또한, 현재 블록에 인접하는 기준 샘플을 이웃 기준 샘플이라 호칭하고, 현재 블록에 인접하지 않는 기준 샘플을 비이웃 기준 샘플이라 호칭하기로 한다.
- [197] 이웃 기준 샘플은, 현재 블록의 좌측 열의 이웃 열 또는 현재 블록의 최상단 행의 이웃 행에 포함될 수 있다. 일 예로, 현재 블록의 좌측 상단 샘플의 좌표를 (0, 0)이라 할 때, (-1, H-1) 위치의 기준 샘플을 포함하는 블록, (W-1, -1) 위치의 기준 샘플을 포함하는 블록, (W, -1) 위치의 기준 샘플을 포함하는 블록, (-1, H) 위치의 기준 샘플을 포함하는 블록 또는 (-1, -1) 위치의 기준 샘플을 포함하는 블록 중 적어도 하나가 후보 블록으로 이용될 수 있다. 도면을 참조하면, 인덱스 0 내지 인덱스 4의 이웃 블록들이 후보 블록들로 이용될 수 있다.
- [198] 비이웃 기준 샘플은, 현재 블록에 인접하는 기준 샘플과의 x축 거리 또는 y축 거리 중 적어도 하나가 기 정의된 값을 갖는 샘플을 나타낸다. 일 예로, 좌측 기준 샘플과의 x축 거리가 기 정의된 값인 기준 샘플을 포함하는 블록, 상단 기준 샘플과의 y축 거리가 기 정의된 값인 비이웃 샘플을 포함하는 블록 또는 좌측 상단 기준 샘플과의 x축 거리 및 y축 거리가 기 정의된 값인 비이웃 샘플을 포함하는 블록 중 적어도 하나가 후보 블록으로 이용될 수 있다. 기 정의된 값은, 4, 8, 12, 16 등의 자연수 일 수 있다. 도면을 참조하면, 인덱스 5 내지 26의 블록들 중 적어도 하나가 후보 블록으로 이용될 수 있다.
- [199] 이웃 기준 샘플과 동일한 수직선, 수평선 또는 대각선상에 위치하지 않는 샘플을 비이웃 기준 샘플로 설정할 수도 있다.
- [200] 도 16은 기준 샘플들의 위치를 나타낸 도면이다.
- [201] 도 16에 도시된 예에서와 같이, 상단 비이웃 기준 샘플들의 x 좌표는 상단 이웃 기준 샘플들의 x 좌표와 상이하게 설정될 수 있다. 일 예로, 상단 이웃 기준 샘플의 위치가 (W-1, -1)인 경우, 상단 비이웃 기준 샘플로부터 y축으로 N만큼 떨어진 상단 비이웃 기준 샘플의 위치는 ((W/2)-1, -1-N)으로 설정되고, 상단 이웃 기준 샘플로부터 y축으로 2N만큼 떨어진 상단 비이웃 기준 샘플의 위치는 (0, -1-2N)으로 설정될 수 있다. 즉, 비인접 기준 샘플의 위치는 인접 기준 샘플의 위치와 인접 기준 샘플과의 거리를 기초로 결정될 수 있다.
- [202] 이하, 후보 블록들 중 이웃 기준 샘플을 포함하는 후보 블록을 이웃 블록이라 호칭하고, 비이웃 기준 샘플을 포함하는 블록을 비이웃 블록이라 호칭하기로 한다.
- [203] 현재 블록과 후보 블록 사이의 거리가 문턱값 이상인 경우, 상기 후보 블록은 머지 후보로서 이용 불가능한 것으로 설정될 수 있다. 상기 문턱값은 코딩 트리 유닛의 크기를 기초로 결정될 수 있다. 일 예로, 문턱값은 코딩 트리 유닛의

- 높이(ctu_height) 또는 코딩 트리 유닛의 높이에서 오프셋을 가산 또는 감산한 값(예컨대, $\text{ctu_height} \pm N$)으로 설정될 수 있다. 오프셋 N 는 부호화기 및 복호화기에서 기 정의된 값으로, 4, 8, 16, 32 또는 ctu_height 로 설정될 수 있다.
- [204] 현재 블록의 y 축 좌표와 후보 블록에 포함된 샘플의 y 축 좌표 사이의 차분이 문턱값보다 큰 경우, 후보 블록은 머지 후보로서 이용 불가능한 것으로 결정될 수 있다.
- [205] 또는, 현재 블록과 동일한 코딩 트리 유닛에 속하지 않는 후보 블록은 머지 후보로 이용 불가능한 것으로 설정될 수 있다. 일 예로, 기준 샘플이 현재 블록이 속하는 코딩 트리 유닛의 상단 경계를 벗어나는 경우, 상기 기준 샘플을 포함하는 후보 블록은 머지 후보로서 이용 불가능한 것으로 설정될 수 있다.
- [206] 만약, 현재 블록의 상단 경계가 코딩 트리 유닛의 상단 경계와 인접하는 경우, 다수의 후보 블록들이 머지 후보로서 이용 가능하지 않은 것으로 결정되어, 현재 블록의 부호화/복호화 효율이 감소할 수 있다. 이와 같은 문제점을 해소하기 위해, 현재 블록의 상단에 위치하는 후보 블록들의 개수보다 현재 블록의 좌측에 위치하는 후보 블록들의 개수가 더 많아지도록 후보 블록들을 설정할 수 있다.
- [207] 도 17은 머지 후보를 유도하기 위해 사용되는 후보 블록들을 예시한 도면이다.
- [208] 도 17에 도시된 예에서와 같이, 현재 블록의 상단 N 개의 블록 열에 속한 상단 블록들 및 현재 블록의 좌측 M 개의 블록 열에 속한 좌측 블록들을 후보 블록들로 설정할 수 있다. 이때, N 보다 M 을 더 크게 설정하여, 상단 후보 블록들의 개수보다 좌측 후보 블록들의 개수를 더 크게 설정할 수 있다.
- [209] 일 예로, 현재 블록 내 기준 샘플의 y 축 좌표와 후보 블록으로 이용될 수 있는 상단 블록의 y 축 좌표의 차분이 현재 블록 높이의 N 배를 초과하지 않도록 설정할 수 있다. 또한, 현재 블록 내 기준 샘플의 x 축 좌표와 후보 블록으로 이용될 수 있는 좌측 블록의 x 축 좌표의 차분이 현재 블록 너비의 M 배를 초과하지 않도록 설정할 수 있다.
- [210] 일 예로, 도 17에 도시된 예에서는 현재 블록의 상단 2개의 블록 열에 속한 블록들 및 현재 블록의 좌측 5개의 블록 열에 속한 블록들이 후보 블록들로 설정되는 것으로 도시되었다.
- [211] 다른 예로, 후보 블록이 현재 블록과 동일한 코딩 트리 유닛에 속하지 않는 경우, 상기 후보 블록 대신 현재 블록과 동일한 코딩 트리 유닛에 속하는 블록 또는 상기 코딩 트리 유닛의 경계에 인접하는 기준 샘플을 포함하는 블록을 이용하여 머지 후보를 유도할 수 있다.
- [212] 도 18은 기준 샘플의 위치가 변경되는 예를 도시한 도면이다.
- [213] 기준 샘플이 현재 블록과 상이한 코딩 트리 유닛에 포함되고, 상기 기준 샘플이 상기 코딩 트리 유닛의 경계에 인접하지 않는 경우, 상기 기준 샘플 대신 상기 코딩 트리 유닛의 경계에 인접하는 기준 샘플을 이용하여 후보 블록을 결정할 수 있다.
- [214] 일 예로, 도 18의 (a) 및 (b)에 도시된 예에서, 현재 블록의 상단 경계와 코딩

트리 유닛의 상단 경계가 접하는 경우, 현재 블록 상단의 기준 샘플들은 현재 블록과 상이한 코딩 트리 유닛에 속하게 된다. 현재 블록과 상이한 코딩 트리 유닛에 속하는 기준 샘플들 중 코딩 트리 유닛의 상단 경계에 인접하지 않는 기준 샘플을 코딩 트리 유닛의 상단 경계에 인접하는 샘플로 대체할 수 있다.

- [215] 일 예로, 도 18의 (a)에 도시된 예에서와 같이, 6 위치의 기준 샘플을 코딩 트리 유닛의 상단 경계에 위치하는 6' 위치의 샘플로 대체하고, 도 18의 (b)에 도시된 예에서와 같이, 15 위치의 기준 샘플을 코딩 트리 유닛의 상단 경계에 위치하는 15' 위치의 샘플로 대체할 수 있다. 이때, 대체 샘플의 y 좌표는, 코딩 트리 유닛의 인접 위치로 변경되고, 대체 샘플의 x 좌표는, 기준 샘플과 동일하게 설정될 수 있다. 일 예로, 6' 위치의 샘플은 6 위치의 샘플과 동일한 x 좌표를 갖고, 15' 위치의 샘플은 15 위치의 샘플과 동일한 x 좌표를 가질 수 있다.
- [216] 또는, 기준 샘플의 x 좌표에서 오프셋을 가산 또는 차감한 것을 대체 샘플의 x 좌표로 설정될 수 있다. 일 예로, 현재 블록의 상단에 위치하는 이웃 기준 샘플과 비이웃 기준 샘플의 x 좌표가 동일한 경우, 기준 샘플의 x 좌표에서 오프셋을 가산 또는 차감한 것을 대체 샘플의 x 좌표로 설정할 수 있다. 이는 비이웃 기준 샘플을 대체하는 대체 샘플이 다른 비이웃 기준 샘플 또는 이웃 기준 샘플과 동일한 위치가 되는 것을 방지하기 위함이다.
- [217] 도 19는 기준 샘플의 위치가 변경되는 예를 도시한 도면이다.
- [218] 현재 블록과 상이한 코딩 트리 유닛에 포함되고, 코딩 트리 유닛의 경계에 인접하지 않는 기준 샘플을 코딩 트리 유닛의 경계에 위치하는 샘플로 대체함에 있어서, 기준 샘플의 x 좌표에서 오프셋을 가산 또는 감산한 값을 대체 샘플의 x 좌표로 설정할 수 있다.
- [219] 일 예로, 도 19에 도시된 예에서, 6 위치의 기준 샘플 및 15 위치의 기준 샘플은 각각 y 좌표가 코딩 트리 유닛의 상단 경계에 인접하는 행과 동일한 6' 위치의 샘플 및 15' 위치의 샘플로 대체될 수 있다. 이때, 6' 위치의 샘플의 x 좌표는 6 위치의 기준 샘플의 x 좌표에서 $W/2$ 를 차분한 값으로 설정되고, 15' 위치의 샘플의 x 좌표는 15 위치의 기준 샘플의 x 좌표에서 $W-1$ 을 차분한 값으로 설정될 수 있다.
- [220] 도 18 및 도 19에 도시된 예와는 달리, 현재 블록의 최상단 행의 상단에 위치하는 행의 y 좌표 또는, 코딩 트리 유닛의 상단 경계의 y 좌표를 대체 샘플의 y 좌표로 설정할 수도 있다.
- [221] 도시되지는 않았지만, 코딩 트리 유닛의 좌측 경계를 기준으로 기준 샘플을 대체하는 샘플을 결정할 수도 있다. 일 예로, 기준 샘플이 현재 블록과 동일한 코딩 트리 유닛에 포함되지 않고, 코딩 트리 유닛의 좌측 경계에 인접하지도 않는 경우, 상기 기준 샘플을 코딩 트리 유닛의 좌측 경계에 인접하는 샘플로 대체할 수 있다. 이때, 대체 샘플은 기준 샘플과 동일한 y 좌표를 갖거나, 기준 샘플의 y 좌표에서 오프셋을 가산 또는 차분하여 획득된 y 좌표를 가질 수 있다.
- [222] 이후, 대체 샘플을 포함하는 블록을 후보 블록으로 설정하고, 상기 후보 블록을

- 기초로 현재 블록의 머지 후보를 유도할 수 있다.
- [223] 현재 블록과 상이한 픽처에 포함된 시간적 이웃 블록으로부터 머지 후보를 유도할 수도 있다. 일 예로, 콜로케이티드 픽처에 포함된 콜로케이티드 블록으로부터 머지 후보를 유도할 수 있다.
- [224] 머지 후보의 움직임 정보는 후보 블록의 움직임 정보와 동일하게 설정될 수 있다. 일 예로, 후보 블록의 모션 벡터, 참조 픽처 인덱스, 예측 방향 또는 양방향 가중치 인덱스 중 적어도 하나를 머지 후보의 움직임 정보로 설정할 수 있다.
- [225] 머지 후보를 포함하는 머지 후보 리스트를 생성할 수 있다(S1402). 상기 머지 후보는 현재 블록에 인접한 이웃 블록으로부터 유도된 인접 머지 후보 및 비이웃 블록으로부터 유도되는 비인접 머지 후보로 구분될 수 있다.
- [226] 머지 후보 리스트 내 머지 후보들의 인덱스는 소정 순서에 따라 할당될 수 있다. 일 예로, 인접 머지 후보에 할당되는 인덱스가 비인접 머지 후보에 할당되는 인덱스보다 작은 값을 가질 수 있다. 또는, 도 15 또는 도 17에 도시된 각 블록의 인덱스에 기초하여, 머지 후보들 각각에 인덱스를 할당할 수 있다.
- [227] 머지 후보에 복수의 머지 후보들이 포함된 경우, 복수의 머지 후보들 중 적어도 하나를 선택할 수 있다(S1403). 이때, 현재 블록의 움직임 정보가 인접 머지 후보로부터 유도되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그일 수 있다. 일 예로, 현재 블록의 움직임 정보가 인접 머지 후보로부터 유도되는지 여부를 나타내는 신택스 요소 `isAdjacentMergeFlag`가 비트스트림을 통해 시그널링될 수 있다. 신택스 요소 `isAdjacentMergeFlag`의 값이 1인 경우, 현재 블록의 움직임 정보는 인접 머지 후보를 기초로 유도될 수 있다. 반면, 신택스 요소 `isAdjacentMergeFlag`의 값이 0인 경우, 현재 블록의 움직임 정보는 비인접 머지 후보를 기초로 유도될 수 있다.
- [228] 표 1은 신택스 요소 `isAdjacentMergeFlag`를 포함하는 신택스 테이블을 나타낸 것이다.

[229] [351]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(slice_type != I) {	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA)	
{	
intra_luma_mpm_flag [x0][y0]	
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType ==	
DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
} else {	
if (cu_skip_falg[x0][y0]) {	
if (MaxNumMergeCand > 1){	
isAdjacentMergeflag	ae(v)
if (isAdjcanetMergeflag){	
merge_idx[x0][y0]	ae(v)
} else{	
NA_merge_idx[x0][y0]	ae(v)
}	
}	
} else { /* MODE_INTER*/	
merge_flag[x0][y0]	ae(v)
if (merge_flag[x0][y0]){	

if (MaxNumMergeCand > 1){	
isAdjacentMergeflag	ae(v)
if (isAdjcanetMergeflag){	
merge_idx[x0][y0]	ae(v)
} else{	
NA_merge_idx[x0][y0]	ae(v)
}	
}	
}	
if(CuPredMode[x0][y0] != MODE_INTRA)	
cu_cbf	ae(v)
if(cu_cbf) {	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	

- [230] 복수의 머지 후보들 중 어느 하나를 특정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 머지 후보 리스트에 포함된 머지 후보들 중 어느 하나의 인덱스를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다.
- [231] isAdjacentMergeflag가 1인 경우, 인접 머지 후보들 중 어느 하나를 특정하는 선택스 요소 merge_idx가 시그널링될 수 있다. 선택스 요소 merge_idx의 최대값은 인접 머지 후보들의 개수에서 1을 차분한 값으로 설정될 수 있다.
- [232] isAdjacentMergeflag가 0인 경우, 비인접 머지 후보들 중 어느 하나를 특정하는 선택스 요소 NA_merge_idx가 시그널링될 수 있다. 선택스 요소 NA_merge_idx는 비인접 머지 후보의 인덱스에서 인접 머지 후보의 개수를 차분한 값을 나타낸다. 복호화기는 NA_merge_idx에 의해 특정된 인덱스에 인접 머지 후보의 개수를 더하여, 비인접 머지 후보를 선택할 수 있다.
- [233]
- [234] 머지 후보 리스트에 포함된 머지 후보들의 개수가 문턱값보다 작은 경우, 인터 영역 모션 정보 테이블에 포함된 머지 후보를 머지 후보 리스트에 추가할 수 있다. 여기서, 문턱값은 머지 후보 리스트가 포함할 수 있는 최대 머지 후보의 개수 또는 최대 머지 후보의 개수에서 오프셋을 차감한 값일 수 있다. 오프셋은, 1 또는 2 등의 자연수일 수 있다. 인터 영역 모션 정보 테이블은 현재 블록 이전에 부호화/복호화된 블록을 기초로 유도된 머지 후보를 포함할 수 있다.
- [235] 인터 영역 모션 정보 테이블은 현재 픽처 내 인터 예측을 기초로 부호화/복호화된 블록으로부터 유도되는 머지 후보를 포함한다. 일 예로, 인터

영역 모션 정보 테이블에 포함된 머지 후보의 움직임 정보는 인터 예측을 기초로 부호화/복호화된 블록의 움직임 정보와 동일하게 설정될 수 있다. 여기서, 움직임 정보는 모션 벡터, 참조 픽처 인덱스, 예측 방향 또는 양방향 가중치 인덱스 중 적어도 하나를 포함할 수 있다.

- [236] 설명의 편의를 위해, 인터 영역 모션 정보 테이블에 포함된 머지 후보를 인터 영역 머지 후보라 호칭하기로 한다.
- [237] 인터 영역 모션 정보 테이블이 포함할 수 있는 최대 머지 후보의 개수는 부호화기 및 복호화기에서 기 정의되어 있을 수 있다. 일 예로, 인터 영역 모션 정보 테이블이 포함할 수 있는 최대 머지 후보의 개수는, 1, 2, 3, 4, 5, 6, 7, 8 또는 그 이상(예컨대, 16)일 수 있다.
- [238] 또는, 인터 영역 모션 정보 테이블의 최대 머지 후보의 개수를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는, 시퀀스, 픽처, 또는 슬라이스 레벨에서 시그널링될 수 있다.
- [239] 또는, 픽처의 크기, 슬라이스의 크기 또는 코딩 트리 유닛의 크기에 따라, 인터 영역 모션 정보 테이블의 최대 머지 후보 개수가 결정될 수 있다.
- [240] 인터 영역 모션 정보 테이블은 픽처, 슬라이스, 타일, 브릭, 코딩 트리 유닛, 또는 코딩 트리 유닛 라인(행 또는 열) 단위로 초기화될 수 있다. 일 예로, 슬라이스가 초기화되는 경우, 인터 영역 모션 정보 테이블도 초기화되어, 인터 영역 모션 정보 테이블은 어떠한 머지 후보도 포함하지 않을 수 있다.
- [241] 또는, 인터 영역 모션 정보 테이블을 초기화할 것인지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수도 있다. 상기 정보는 슬라이스, 타일, 브릭 또는 블록 레벨에서 시그널링될 수 있다. 상기 정보가 인터 영역 모션 정보 테이블을 초기화할 것을 지시하기 전까지, 기 구성된 인터 영역 모션 정보 테이블이 이용될 수 있다.
- [242] 또는, 픽처 파라미터 세트 또는 슬라이스 헤더를 통해 초기 인터 영역 머지 후보에 대한 정보가 시그널링될 수 있다. 슬라이스가 초기화되더라도, 인터 영역 모션 정보 테이블은 초기 인터 영역 머지 후보를 포함할 수 있다. 이에 따라, 슬라이스 내 첫번째 부호화/복호화 대상인 블록에 대해서도 인터 영역 머지 후보를 이용할 수 있다.
- [243] 부호화/복호화 순서에 따라, 블록들을 부호화/복호화하되, 인터 예측을 기초로 부호화/복호화된 블록들을 부호화/복호화 순서에 따라 순차적으로 인터 영역 머지 후보로 설정할 수 있다.
- [244] 도 20은 인터 영역 모션 정보 테이블의 업데이트 양상을 설명하기 위한 도면이다.
- [245] 현재 블록에 대해, 인터 예측이 수행된 경우(S2001), 현재 블록을 기초로 인터 영역 머지 후보를 유도할 수 있다(S2002). 인터 영역 머지 후보의 움직임 정보는 현재 블록의 움직임 정보와 동일하게 설정될 수 있다.
- [246] 인터 영역 모션 정보 테이블이 빈 상태인 경우(S2003), 현재 블록을 기초로

유도된 인터 영역 머지 후보를 인터 영역 모션 정보 테이블에 추가할 수 있다(S2004).

- [247] 인터 영역 모션 정보 테이블이 이미 인터 영역 머지 후보를 포함하고 있는 경우(S2003), 현재 블록의 움직임 정보(또는, 이를 기초로 유도된 인터 영역 머지 후보)에 대한 중복성 검사를 실시할 수 있다(S2005). 중복성 검사는 인터 영역 모션 정보 테이블에 기 저장된 인터 영역 머지 후보의 움직임 정보와 현재 블록의 움직임 정보가 동일한지 여부를 결정하기 위한 것이다. 중복성 검사는 인터 영역 모션 정보 테이블에 기 저장된 모든 인터 영역 머지 후보들을 대상으로 수행될 수 있다. 또는, 인터 영역 모션 정보 테이블에 기 저장된 인터 영역 머지 후보들 중 인덱스가 문턱값 이상 또는 문턱값 이하인 인터 영역 머지 후보들을 대상으로 중복성 검사를 수행할 수 있다.
- [248] 현재 블록의 움직임 정보와 동일한 움직임 정보를 갖는 인터 예측 머지 후보가 포함되어 있지 않은 경우, 현재 블록을 기초로 유도된 인터 영역 머지 후보를 인터 영역 모션 정보 테이블에 추가할 수 있다(S2008). 인터 예측 머지 후보들이 동일한지 여부는, 인터 예측 머지 후보들의 움직임 정보(예컨대, 모션 벡터 및/또는 참조 픽처 인덱스 등)가 동일한지 여부를 기초로 결정될 수 있다.
- [249] 이때, 인터 영역 모션 정보 테이블에 이미 최대 개수의 인터 영역 머지 후보들이 저장되어 있을 경우(S2006), 가장 오래된 인터 영역 머지 후보를 삭제하고(S2007), 현재 블록을 기초로 유도된 인터 영역 머지 후보를 인터 영역 모션 정보 테이블에 추가할 수 있다(S2008).
- [250] 인터 영역 머지 후보들은 각기 인덱스에 의해 식별될 수 있다. 현재 블록으로부터 유도된 인터 영역 머지 후보가 인터 영역 모션 정보 테이블에 추가되는 경우, 상기 인터 영역 머지 후보에 가장 낮은 인덱스(예컨대, 0)를 할당하고, 기 저장된 인터 영역 머지 후보들의 인덱스를 1씩 증가시킬 수 있다. 이때, 인터 영역 모션 정보 테이블에 이미 최대 개수의 인터 예측 머지 후보들이 저장되었던 경우, 인덱스가 가장 큰 인터 영역 머지 후보가 제거된다.
- [251] 또는, 현재 블록으로부터 유도된 인터 영역 머지 후보가 인터 영역 모션 정보 테이블에 추가되는 경우, 상기 인터 영역 머지 후보에 가장 큰 인덱스를 할당할 수 있다. 일 예로, 인터 영역 모션 정보 테이블에 기 저장된 인터 예측 머지 후보들의 개수가 최대값보다 작은 경우, 상기 인터 영역 머지 후보에는 기 저장된 인터 예측 머지 후보들의 개수와 동일한 값의 인덱스가 할당될 수 있다. 또는, 인터 영역 모션 정보 테이블에 기 저장된 인터 예측 머지 후보들의 개수가 최대값과 같은 경우, 상기 인터 영역 머지 후보에는 최대값에서 1을 차감한 인덱스가 할당될 수 있다. 또한, 인덱스가 가장 작은 인터 영역 머지 후보가 제거되고, 잔여 기 저장된 인터 영역 머지 후보들의 인덱스들이 1씩 감소하게 된다.
- [252] 도 21은 인터 영역 머지 후보 테이블의 업데이트 양상을 나타낸 도면이다.
- [253] 현재 블록으로부터 유도된 인터 영역 머지 후보가 인터 영역 머지 후보

테이블에 추가되면서, 상기 인터 영역 머지 후보에 가장 큰 인덱스가 할당되는 것으로 가정한다. 또한, 인터 영역 머지 후보 테이블에는 이미 최대 개수의 인터 영역 머지 후보가 저장된 것으로 가정한다.

- [254] 현재 블록으로부터 유도된 인터 영역 머지 후보 $HmvpCand[n+1]$ 를 인터 영역 머지 후보 테이블 $HmvpCandList$ 에 추가하는 경우, 기 저장된 인터 영역 머지 후보들 중 인덱스가 가장 작은 인터 영역 머지 후보 $HmvpCand[0]$ 를 삭제하고, 잔여 인터 영역 머지 후보들의 인덱스를 1씩 감소시킬 수 있다. 또한, 현재 블록으로부터 유도된 인터 영역 머지 후보 $HmvpCand[n+1]$ 의 인덱스를 최대값(도 21에 도시된 예에서는 n)으로 설정할 수 있다.
- [255] 현재 블록을 기초로 유도된 인터 영역 머지 후보와 동일한 인터 영역 머지 후보가 기 저장되어 있을 경우(S2005), 현재 블록을 기초로 유도된 인터 영역 머지 후보를 인터 영역 모션 정보 테이블에 추가하지 않을 수 있다(S2009).
- [256] 또는, 현재 블록을 기초로 유도된 인터 영역 머지 후보를 인터 영역 모션 정보 테이블에 추가하면서, 상기 인터 영역 머지 후보와 동일한 기 저장된 인터 영역 머지 후보를 제거할 수도 있다. 이 경우, 기 저장된 인터 영역 머지 후보의 인덱스가 새롭게 갱신되는 것과 동일한 효과가 야기된다.
- [257] 도 22는 기 저장된 인터 영역 머지 후보의 인덱스가 갱신되는 예를 나타낸 도면이다.
- [258] 현재 블록을 기초로 유도된 인터 영역 머지 후보 $mvCand$ 와 동일한 기 저장된 인터 예측 머지 후보의 인덱스가 $hIdx$ 인 경우, 상기 기 저장된 인터 예측 머지 후보를 삭제하고, 인덱스가 $hIdx$ 보다 큰 인터 예측 머지 후보들의 인덱스를 1만큼 감소시킬 수 있다. 일 예로, 도 22에 도시된 예에서는 $mvCand$ 와 동일한 $HmvpCand[2]$ 가 인터 영역 모션 정보 테이블 $HmvpCandList$ 에서 삭제되고, $HmvpCand[3]$ 부터 $HmvpCand[n]$ 까지의 인덱스가 1씩 감소하는 것으로 도시되었다.
- [259] 그리고, 현재 블록을 기초로 유도된 인터 영역 머지 후보 $mvCand$ 를 인터 영역 모션 정보 테이블의 마지막에 추가할 수 있다.
- [260] 또는, 현재 블록을 기초로 유도된 인터 영역 머지 후보와 동일한 기 저장된 인터 영역 머지 후보에 할당된 인덱스를 갱신할 수 있다. 예컨대, 기 저장된 인터 영역 머지 후보의 인덱스를 최소값 또는 최대값으로 변경할 수 있다.
- [261] 소정 영역에 포함된 블록들의 움직임 정보는 인터 영역 모션 정보 테이블에 추가되지 않도록 설정될 수 있다. 일 예로, 머지 처리 영역에 포함된 블록의 움직임 정보를 기초로 유도되는 인터 영역 머지 후보는 인터 영역 모션 정보 테이블에 추가하지 않을 수 있다. 머지 처리 영역에 포함된 블록들에 대해서는 부호화/복호화 순서가 정의되어 있지 않은 바, 이들 중 어느 하나의 움직임 정보를 다른 블록의 인터 예측시에 이용하는 것은 부적절하다. 이에 따라, 머지 처리 영역에 포함된 블록들을 기초로 유도된 인터 영역 머지 후보들은 인터 영역 모션 정보 테이블에 추가하지 않을 수 있다.

- [262] 서브 블록 단위로 움직임 보상 예측이 수행된 경우, 현재 블록에 포함된 복수의 서브 블록들 중 대표 서브 블록의 움직임 정보를 기초로 인터 영역 머지 후보를 유도할 수 있다. 일 예로, 현재 블록에 대해 서브 블록 머지 후보가 사용된 경우, 서브 블록들 중 대표 서브 블록의 움직임 정보를 기초로 인터 영역 머지 후보를 유도할 수 있다.
- [263] 서브 블록들의 움직임 벡터는 다음의 순서로 유도될 수 있다. 먼저, 현재 블록의 머지 후보 리스트에 포함된 머지 후보들 중 어느 하나를 선택하고, 선택된 머지 후보의 모션 벡터를 기초로, 초기 시프트 벡터(*shVector*)를 유도할 수 있다. 그리고, 코딩 블록 내 각 서브 블록의 기준 샘플(예컨대, 좌상단 샘플 또는 중간 위치 샘플)의 위치 (*xSb*, *ySb*)에 초기 시프트 벡터를 가산하여, 기준 샘플의 위치가 (*xColSb*, *yColSb*)인 시프트 서브 블록을 유도할 수 있다. 하기 수식 4는 시프트 서브 블록을 유도하기 위한 수식을 나타낸다.
- [264] [수식4]
- $$(xColSb, yColSb) = (xSb + shVector[0] \gg 4, ySb + shVector[1] \gg 4)$$
- [265] 그리고 나서, (*xColSb*, *yColSb*)를 포함하는 서브 블록의 센터 포지션에 대응하는 콜로케이션된 블록의 모션 벡터를 (*xSb*, *ySb*)를 포함하는 서브 블록의 모션 벡터로 설정할 수 있다.
- [266] 대표 서브 블록은 현재 블록의 좌측 상단 샘플 또는 중앙 샘플을 포함하는 서브 블록을 의미할 수 있다.
- [267] 도 23은 대표 서브 블록의 위치를 나타낸 도면이다.
- [268] 도 23의 (a)는 현재 블록의 좌측 상단에 위치한 서브 블록이 대표 서브 블록으로 설정된 예를 나타내고, 도 23의 (b)는 현재 블록의 중앙에 위치한 서브 블록이 대표 서브 블록으로 설정된 예를 나타낸다. 서브 블록 단위로 움직임 보상 예측이 수행된 경우, 현재 블록의 좌측 상단 샘플을 포함하는 서브 블록 또는 현재 블록의 중앙 샘플을 포함하는 서브 블록의 움직임 벡터를 기초로, 현재 블록의 인터 영역 머지 후보를 유도할 수 있다.
- [269] 현재 블록의 인터 예측 모드를 기초로, 현재 블록을 인터 영역 머지 후보로 이용할 것인지 여부를 결정할 수도 있다. 일 예로, 어파인 모션 모델을 기초로 부호화/복호화된 블록은 인터 영역 머지 후보로 이용 불가능 한 것으로 설정될 수 있다. 이에 따라, 현재 블록이 인터 예측으로 부호화/복호화되었다 하더라도, 현재 블록의 인터 예측 모드가 어파인 예측 모드인 경우에는, 현재 블록을 기초로 인터 예측 모션 정보 테이블을 업데이트 하지 않을 수 있다.
- [270] 또는, 어파인 모션 모델을 기초로 부호화/복호화된 블록에 포함된 서브 블록 중 적어도 하나의 서브 블록 벡터를 기초로 인터 영역 머지 후보를 유도할 수도 있다. 일 예로, 현재 블록의 좌측 상단에 위치하는 서브 블록, 중앙에 위치하는 서브 블록 또는 우측 상단에 위치하는 서브 블록을 이용하여 인터 영역 머지 후보를 유도할 수 있다. 또는, 복수 서브 블록들의 서브 블록 벡터들의 평균값을 인터 영역 머지 후보의 움직임 벡터로 설정할 수도 있다.

- [271] 또는, 어파인 모션 모델을 기초로 부호화/복호화된 블록의 어파인 시드 벡터들의 평균값을 기초로 인터 영역 머지 후보를 유도할 수도 있다. 일 예로, 현재 블록의 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 또는 제3 어파인 시드 벡터 중 적어도 하나의 평균을 인터 영역 머지 후보의 움직임 벡터로 설정할 수 있다.
- [272] 또는, 인터 예측 모드 별로 인터 영역 모션 정보 테이블을 구성할 수 있다. 일 예로, 인트라 블록 카피로 부호화/복호화된 블록을 위한 인터 영역 모션 정보 테이블, 병진 모션 모델을 기초로 부호화/복호화된 블록을 위한 인터 영역 모션 정보 테이블 또는 어파인 모션 모델을 기초로 부호화/복호화된 블록을 위한 인터 영역 모션 정보 테이블 중 적어도 하나가 정의될 수 있다. 현재 블록의 인터 예측 모드에 따라, 복수의 인터 영역 모션 정보 테이블 중 어느 하나가 선택될 수 있다.
- [273] 도 24는 인터 예측 모드 별로 인터 영역 모션 정보 테이블이 생성되는 예를 나타낸 것이다.
- [274] 블록이 논어파인 모션 모델을 기초로 부호화/복호화된 경우, 상기 블록을 기초로 유도된 인터 영역 머지 후보 mvCand는 인터 영역 논어파인 모션 정보 테이블 HmvpCandList에 추가될 수 있다. 반면, 블록이 어파인 모션 모델을 기초로 부호화/복호화된 경우, 상기 블록을 기초로 유도된 인터 영역 머지 후보 mvAfCand는 인터 영역 어파인 모션 정보 테이블 HmvpAfCandList에 추가될 수 있다.
- [275] 어파인 모션 모델을 기초로 부호화/복호화된 블록으로부터 유도되는 인터 영역 머지 후보에는 상기 블록의 어파인 시드 벡터들이 저장될 수 있다. 이에 따라, 상기 인터 영역 머지 후보를 현재 블록의 어파인 시드 벡터를 유도하기 위한 머지 후보로서 이용할 수 있다.
- [276]
- [277] 설명한 인터 영역 모션 정보 테이블 이외에 추가 인터 영역 모션 정보 테이블을 정의할 수도 있다. 상기 설명한 인터 영역 모션 정보 테이블(이하, 제1 인터 영역 모션 정보 테이블이라 함) 이외에 룬텀 모션 정보 테이블(이하, 제2 인터 영역 모션 정보 테이블이라 함)을 정의할 수 있다. 여기서, 룬텀 모션 정보 테이블은 룬텀 머지 후보들을 포함한다.
- [278] 제1 인터 영역 모션 정보 테이블 및 제2 인터 영역 모션 정보 테이블이 모두 빈 상태일 경우, 먼저, 제2 인터 영역 모션 정보 테이블에 인터 영역 머지 후보를 추가할 수 있다. 제2 인터 영역 모션 정보 테이블에 가용한 인터 영역 머지 후보들이 개수가 최대 개수에 다다른 이후, 비로서 제1 인터 영역 모션 정보 테이블에 인터 영역 머지 후보를 추가할 수 있다.
- [279] 또는, 하나의 인터 예측 머지 후보를 제2 인터 영역 모션 정보 테이블 및 제1 인터 영역 모션 정보 테이블 모두에 추가할 수도 있다.
- [280] 이때, 구성이 완료된 제2 인터 영역 모션 정보 테이블은 더 이상 업데이트를 수행하지 않을 수 있다. 또는, 복호화된 영역이 슬라이스의 소정의 비율 이상인

- 경우, 제2 인터 영역 모션 정보 테이블을 업데이트할 수 있다. 또는, N개의 코딩 트리 유닛 라인마다 제2 인터 영역 모션 정보 테이블을 업데이트할 수 있다.
- [281] 반면, 제1 인터 영역 모션 정보 테이블은 인터 예측으로 부호화/복호화된 블록이 발생할 때 마다 업데이트될 수 있다. 단, 제2 인터 영역 모션 정보 테이블에 추가되는 인터 영역 머지 후보는 제1 인터 영역 모션 정보 테이블을 업데이트 하는데 이용되지 않도록 설정될 수도 있다.
- [282] 제1 인터 영역 모션 정보 테이블 또는 제2 인터 영역 모션 정보 테이블 중 어느 하나를 선택하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 머지 후보 리스트에 포함된 머지 후보의 개수가 문턱값보다 작은 경우, 상기 정보가 지시하는 인터 영역 모션 정보 테이블에 포함된 머지 후보들을 머지 후보 리스트에 추가할 수 있다.
- [283] 또는, 현재 블록의 크기, 형태, 인터 예측 모드, 양방향 예측 여부, 움직임 벡터 리파인 여부 또는 삼각 파티셔닝 여부에 기초하여, 인터 영역 모션 정보 테이블을 선택할 수도 있다.
- [284] 또는, 제1 인터 영역 모션 정보 테이블에 포함된 인터 영역 머지 후보를 추가하더라도 머지 후보 리스트에 포함된 머지 후보들의 개수가 최대 머지 개수보다 작은 경우, 제2 인터 영역 모션 정보 테이블에 포함된 인터 영역 머지 후보를 머지 후보 리스트에 추가할 수 있다.
- [285] 도 25는 룬텀 모션 정보 테이블에 포함된 인터 영역 머지 후보가 머지 후보 리스트에 추가되는 예를 나타낸 도면이다.
- [286] 머지 후보 리스트에 포함된 머지 후보의 개수가 최대 개수보다 작은 경우, 제1 인터 영역 모션 정보 테이블 HmvpCandList에 포함된 인터 영역 머지 후보를 머지 후보 리스트에 추가할 수 있다. 제1 인터 영역 모션 정보 테이블에 포함된 인터 영역 머지 후보들을 머지 후보 리스트에 추가하였음에도, 머지 후보 리스트에 포함된 머지 후보의 개수가 최대 개수보다 작은 경우, 룬텀 모션 정보 테이블 HmvpLTCandList에 포함된 인터 영역 머지 후보들을 머지 후보 리스트에 추가할 수 있다.
- [287] 표 2는 룬텀 모션 정보 테이블에 포함된 인터 영역 머지 후보들을 머지 후보 리스트에 추가하는 과정을 나타낸 것이다.

[288] [표2]

For each candidate in HMVPCandList with index HMVPLTIdx = 1..numHMVPLTCand, the following ordered steps are repeated until combStop is equal to true- sameMotion is set to FALSE- If hmvpStop is equal to FALSE and numCurrMergecand is less than (MaxNumMergeCand-1), hmvpLT is set to TRUE- If HMVPLTCandList[NumLTHmvp-HMVPLTIdx] have the same motion vectors and the same reference indices with any mergeCandList[i] with I being 0.. numOrigMergeCand-1 and HasBeenPruned[i] equal to false, sameMotion is set to true- If sameMotion is equal to false, mergeCandList[numCurrMergeCand++] is set to HMVPLTCandList[NumLTHmvp-HMVPLTIdx]- If numCurrMergeCand is equal to (MaxNumMergeCand-1), hmvpLTStop is set to TRUE

- [289] 인터 영역 머지 후보가 움직임 정보 이외에 추가 정보를 포함하도록 설정할 수 있다. 일 예로, 인터 영역 머지 후보에 대해 블록의 크기, 형태 또는 블록의 파티션 정보를 추가 저장할 수 있다. 현재 블록의 머지 후보 리스트 구성 시, 인터 예측 머지 후보들 중 현재 블록과 크기, 형태 또는 파티션 정보가 동일 또는 유사한 인터 예측 머지 후보만을 사용하거나, 현재 블록과 크기, 형태 또는 파티션 정보가 동일 또는 유사한 인터 예측 머지 후보를 먼저 머지 후보 리스트에 추가할 수 있다.
- [290] 또는, 블록 크기, 형태 또는 파티션 정보 별로 인터 영역 모션 정보 테이블을 생성할 수 있다. 복수의 인터 영역 모션 정보 테이블 중 현재 블록의 형태, 크기 또는 파티션 정보에 부합하는 인터 영역 모션 정보 테이블을 이용하여, 현재 블록의 머지 후보 리스트를 생성할 수 있다.
- [291] 현재 블록의 머지 후보 리스트에 포함된 머지 후보의 개수가 문턱값보다 작은 경우, 인터 영역 모션 정보 테이블에 포함된 인터 영역 머지 후보를 머지 후보 리스트에 추가할 수 있다. 상기 추가 과정은 인덱스 기준 오름차순 또는 내림차순으로 수행된다. 일 예로, 인덱스가 가장 큰 인터 영역 머지 후보부터 머지 후보 리스트에 추가할 수 있다.
- [292] 인터 영역 모션 정보 테이블에 포함된 인터 영역 머지 후보를 머지 후보 리스트에 추가하고자 하는 경우, 인터 영역 머지 후보와 머지 후보 리스트에 기 저장된 머지 후보들간의 중복성 검사가 수행될 수 있다.
- [293] 일 예로, 표 3은 인터 영역 머지 후보가 머지 후보 리스트에 추가되는 과정을 나타낸 것이다.

[294] [표3]

For each candidate in HMVPCandList with index HMVPIdx = 1.. numCheckedHMVPCand, the following ordered steps are repeated until combStop is equal to true- sameMotion is set to false- If HMVPCandList[NumHmvp-HMVPIdx] have the same motion vectors and the same reference indices with any mergeCandList[i] with I being 0.. numOrigMergeCand-1 and HasBeenPruned[i] equal to false, sameMotion is set to true- If sameMotion is equal to false, mergeCandList[numCurrMergeCand++] is set to HMVPCandList[NumHmvp-HMVPIdx]- If numCurrMergeCand is equal to (MaxNumMergeCand-1), hmvpStop is set to TRUE
--

- [295] 중복성 검사는, 인터 영역 모션 정보 테이블에 포함된 인터 영역 머지 후보들 중 일부에 대해서만 수행될 수도 있다. 일 예로, 인덱스가 문턱값 이상 또는 문턱값 이하인 인터 영역 머지 후보에 대해서만 중복성 검사가 수행될 수 있다. 또는 인덱스가 가장 큰 N개의 머지 후보 또는 인덱스가 가장 작은 N개의 머지 후보에 대해서만 중복성 검사가 수행될 수 있다.
- [296] 또는, 머지 후보 리스트에 기 저장된 머지 후보들 중 일부에 대해서만 중복성 검사를 수행할 수 있다. 일 예로, 인덱스가 문턱값 이상 또는 문턱값 이하인 머지 후보 또는 특정 위치의 블록으로부터 유도된 머지 후보에 대해서만 중복성 검사가 수행될 수 있다. 여기서, 특정 위치는, 현재 블록의 좌측 이웃 블록, 상단 이웃 블록, 우측 상단 이웃 블록 또는 좌측 하단 이웃 블록 중 적어도 하나를 포함할 수 있다.
- [297] 도 26은 머지 후보들 중 일부에 대해서만 중복성 검사가 수행되는 예를 도시한 도면이다.
- [298] 인터 영역 머지 후보 HmvpCand[j]를 머지 후보 리스트에 추가하고자 하는 경우, 인터 영역 머지 후보에 대해 인덱스가 가장 큰 2개의 머지 후보들 mergeCandList[NumMerge-2] 및 mergeCandList[NumMerge-1]과의 중복성 검사를 수행할 수 있다. 여기서, NumMerge는 가용한 공간적 머지 후보 및 시간적 머지 후보의 개수를 나타낼 수 있다.
- [299] 도시된 예와 달리, 인터 영역 머지 후보 HmvpCand[j]를 머지 후보 리스트에 추가하고자 하는 경우, 인터 영역 머지 후보에 대해 인덱스가 가장 작은 최대 2개의 머지 후보와의 중복성 검사를 수행할 수도 있다. 예컨대, mergeCandList[0] 및 mergeCandList[1]에 대해 HmvpCand[j]와의 동일 여부를 확인할 수 있다. 또는 특정 위치에서 유도된 머지 후보에 대해서만 중복성 검사를 수행할 수 있다. 일 예로, 현재 블록의 좌측에 위치하는 주변 블록으로부터 유도된 머지 후보 또는 현재 블록의 상단에 위치하는 주변 블록으로부터 유도된 머지 후보 중 적어도 하나에 대해 중복성 검사를 수행할 수 있다. 머지 후보 리스트에 특정 위치에서 유도된 머지 후보가 존재하지 않는 경우, 중복성 검사 없이 인터 영역 머지

후보를 머지 후보 리스트에 추가할 수 있다.

[300]

[301] 제1 인터 영역 머지 후보와 동일한 머지 후보가 발견된 경우, 제2 인터 영역 머지 후보에 대한 중복성 검사시 상기 제1 인터 영역 머지 후보와 동일한 머지 후보와의 중복성 검사를 생략할 수 있다.

[302] 도 27은 특정 머지 후보와의 중복성 검사가 생략되는 예를 나타낸 도면이다.

[303] 인덱스가 i 인 인터 영역 머지 후보 $HmvpCand[i]$ 를 머지 후보 리스트에 추가하고자 하는 경우, 상기 인터 영역 머지 후보와 머지 후보 리스트에 저장된 머지 후보들 사이의 중복성 검사가 수행된다. 이때, 인터 영역 머지 후보 $HmvpCand[i]$ 와 동일한 머지 후보 $mergeCandList[j]$ 가 발견된 경우, 인터 영역 머지 후보 $HmvpCand[i]$ 를 머지 후보 리스트에 추가하지 않고, 인덱스가 $i-1$ 인 인터 영역 머지 후보 $HmvpCand[i-1]$ 와 머지 후보들 간의 중복성 검사를 수행할 수 있다. 이때, 인터 영역 머지 후보 $HmvpCand[i-1]$ 과 머지 후보 $mergeCandList[j]$ 사이의 중복성 검사는 생략할 수 있다.

[304] 일 예로, 도 27에 도시된 예에서는, $HmvpCand[i]$ 와 $mergeCandList[2]$ 가 동일한 것으로 결정되었다. 이에 따라, $HmvpCand[i]$ 는 머지 후보 리스트에 추가되지 않고, $HmvpCand[i-1]$ 에 대한 중복성 검사가 수행될 수 있다. 이때, $HmvpCand[i-1]$ 과 $mergeCandList[2]$ 사이의 중복성 검사는 생략될 수 있다.

[305] 현재 블록의 머지 후보 리스트에 포함된 머지 후보의 개수가 문턱값보다 작은 경우, 인터 영역 머지 후보 이외에도, 페어 와이즈 머지 후보 또는 제로 머지 후보 중 적어도 하나가 더 포함될 수도 있다. 페어 와이즈 머지 후보는 둘 이상의 머지 후보들의 움직임 벡터들을 평균한 값을 움직임 벡터로 갖는 머지 후보를 의미하고, 제로 머지 후보는 모션 벡터가 0인 머지 후보를 의미한다.

[306] 현재 블록의 머지 후보 리스트는 다음의 순서를 따라, 머지 후보가 추가될 수 있다.

[307] 공간적 머지 후보 - 시간적 머지 후보 - 인터 영역 머지 후보 - (인터 영역 어파인 머지 후보) - 페어 와이즈 머지 후보 - 제로 머지 후보

[308] 공간적 머지 후보는 이웃 블록 또는 비이웃 블록 중 적어도 하나로부터 유도되는 머지 후보를 의미하고, 시간적 머지 후보는 이전 참조 픽처에서 유도되는 머지 후보를 의미한다. 인터 영역 어파인 머지 후보는 어파인 모션 모델로 부호화/복호화된 블록으로부터 유도된 인터 영역 머지 후보를 나타낸다.

[309] 모션 벡터 예측 모드에서도 인터 영역 모션 정보 테이블이 이용될 수 있다. 일 예로, 현재 블록의 움직임 벡터 예측 후보 리스트에 포함된 움직임 벡터 예측 후보의 개수가 문턱값보다 작은 경우, 인터 영역 모션 정보 테이블에 포함된 인터 영역 머지 후보를 현재 블록에 대한 움직임 벡터 예측 후보로 설정할 수 있다. 구체적으로, 인터 영역 머지 후보의 모션 벡터를 움직임 벡터 예측 후보로 설정할 수 있다.

[310] 현재 블록의 움직임 벡터 예측 후보 리스트에 포함된 움직임 벡터 예측 후보들

중 어느 하나가 선택되면, 선택된 후보를 현재 블록의 움직임 벡터 예측자로 설정할 수 있다. 이후, 현재 블록의 움직임 벡터 잔차값을 복호화한 뒤, 움직임 벡터 예측자와 움직임 벡터 잔차값을 합하여 현재 블록의 움직임 벡터를 획득할 수 있다.

- [311] 현재 블록의 움직임 벡터 예측 후보 리스트는 다음의 순서를 따라, 구성될 수 있다.
- [312] 공간적 모션 벡터 예측 후보 - 시간적 모션 벡터 예측 후보 - 인터 복호화 영역 머지 후보 - (인터 복호화 영역 어파인 머지 후보) - 제로 모션 벡터 예측 후보
- [313] 공간적 모션 벡터 예측 후보는 이웃 블록 또는 비이웃 블록 중 적어도 하나로부터 유도되는 모션 벡터 예측 후보를 의미하고, 시간적 모션 벡터 예측 후보는 이전 참조 픽처에서 유도되는 모션 벡터 예측 후보를 의미한다. 인터 영역 어파인 머지 후보는 어파인 모션 모델로 부호화/복호화된 블록으로부터 유도된 인터 영역 모션 벡터 예측 후보를 나타낸다. 제로 모션 벡터 예측 후보는 움직임 벡터의 값이 0인 후보를 나타낸다.
- [314]
- [315] 현재 블록의 머지 후보가 선택되면, 선택된 머지 후보의 움직임 벡터를 초기 움직임 벡터로 설정하고, 초기 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 유도된 움직임 벡터를 이용하여, 현재 블록에 대한 움직임 보상 예측을 수행할 수 있다. 머지 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 새로운 움직임 벡터를 유도하는 것을 머지 오프셋 벡터 부호화 방법이라 정의할 수 있다.
- [316] 머지 오프셋 부호화 방법을 사용할 것인지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 `merge_offset_vector_flag`일 수 있다. 일 예로, `merge_offset_vector_flag`의 값이 1인 것은 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용됨을 나타낸다. 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 머지 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여, 현재 블록의 움직임 벡터를 유도할 수 있다. `merge_offset_vector_flag`의 값이 0인 것은 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되지 않음을 나타낸다. 머지 오프셋 부호화 방법이 적용되지 않는 경우, 머지 후보의 움직임 벡터가 현재 블록의 움직임 벡터로 설정될 수 있다.
- [317] 상기 플래그는, 스킵 모드의 적용 여부를 나타내는 스킵 플래그의 값이 참인 경우 또는 머지 모드의 적용 여부를 나타내는 머지 플래그의 값이 참인 경우에 한하여 시그널링될 수 있다. 일 예로, 현재 블록에 스킵 모드가 적용됨을 나타내는 `skip_flag`의 값이 1인 경우 또는 현재 블록에 머지 모드가 적용됨을 나타내는 `merge_flag`의 값이 1인 경우, `merge_offset_vector_flag`가 부호화되어 시그널링될 수 있다.
- [318] 현재 블록에 머지 오프셋 부호화 방법이 적용되는 것으로 결정된 경우, 머지

후보 리스트에 포함된 머지 후보들 중 어느 하나를 특정하는 정보, 오프셋 벡터의 크기를 나타내는 정보 또는 오프셋 벡터의 방향을 나타내는 정보 중 적어도 하나가 추가 시그널링될 수 있다.

- [319] 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수를 결정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수는 6 이하의 자연수로 설정될 수 있다.
- [320] 현재 블록에 머지 오프셋 부호화 방법이 적용되는 것으로 결정되는 경우, 기 설정된 최대 개수의 머지 후보만이 현재 블록의 초기 움직임 벡터로 설정될 수 있다. 즉, 머지 오프셋 부호화 방법의 적용 여부에 따라, 현재 블록이 이용할 수 있는 머지 후보들의 개수가 적응적으로 결정될 수 있다. 일 예로, `merge_offset_vector_flag`의 값이 0으로 설정된 경우, 현재 블록이 이용할 수 있는 머지 후보들의 최대 개수는 M개로 설정되는 반면, `merge_offset_vector_flag`의 값이 1로 설정된 경우, 현재 블록이 이용할 수 있는 머지 후보들의 최대 개수는 N개로 설정될 수 있다. 여기서, M은 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수를 나타내고, N은 M과 같거나 M보다 작은 자연수를 나타낸다.
- [321] 일 예로, M이 6이고, N이 2인 경우, 머지 후보 리스트에 포함된 머지 후보들 중 인덱스가 가장 작은 2개의 머지 후보들이 현재 블록에 대해 이용 가능한 것으로 설정될 수 있다. 이에 따라, 인덱스 값이 0인 머지 후보의 움직임 벡터 또는 인덱스 값이 1인 머지 후보의 움직임 벡터가 현재 블록의 초기 움직임 벡터로 설정될 수 있다. 만약, M과 N이 동일한 경우(예컨대, M과 N이 2인 경우), 머지 후보 리스트에 포함된 모든 머지 후보들이 현재 블록에 대해 이용가능한 것으로 설정될 수 있다.
- [322] 또는, 이웃 블록을 머지 후보로서 이용 가능한지 여부가 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는지 여부를 기초로 결정될 수 있다. 일 예로, `merge_offset_vector_flag`의 값이 1인 경우, 현재 블록의 우측 상단 코너에 인접하는 이웃 블록, 좌측 하단 코너에 인접하는 이웃 블록 또는 좌측 하단 코너에 인접하는 이웃 블록 중 적어도 하나는 머지 후보로서 이용 불가능한 것으로 설정될 수 있다. 이에 따라, 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 현재 블록의 우측 상단 코너에 인접하는 이웃 블록, 좌측 하단 코너에 인접하는 이웃 블록 또는 좌측 하단 코너에 인접하는 이웃 블록 중 적어도 하나의 움직임 벡터는 초기 움직임 벡터로 설정될 수 없다. 또는, `merge_offset_vector_flag`의 값이 1인 경우, 현재 블록의 시간적 이웃 블록은 머지 후보로서 이용 불가능한 것으로 설정될 수 있다.
- [323] 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 페어 와이즈 머지 후보 또는 제로 머지 후보 중 적어도 하나를 이용하지 않도록 설정할 수 있다. 이에 따라, `merge_offset_vector_flag`의 값이 1인 경우, 머지 후보 리스트에 포함된 머지 후보들의 개수가 최대 개수보다 작은 경우라 하더라도, 페어 와이즈 머지

후보 또는 제로 머지 후보 중 적어도 하나는 머지 후보 리스트에 추가되지 않을 수 있다.

- [324] 머지 후보의 움직임 벡터를 현재 블록의 초기 움직임 벡터로 설정할 수 있다. 이때, 현재 블록이 이용할 수 있는 머지 후보들의 개수가 복수인 경우, 복수의 머지 후보들 중 어느 하나를 특정하는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수가 1보다 큰 경우, 복수의 머지 후보들 중 어느 하나를 가리키는 정보 `merge_idx`가 비트스트림을 통해 시그널링될 수 있다. 즉, 머지 오프셋 부호화 방법 하에서, 복수의 머지 후보들 중 어느 하나를 특정하기 위한 정보 `merge_idx`에 의해 머지 후보가 특정될 수 있다. 현재 블록의 초기 움직임 벡터는 `merge_idx`가 가리키는 머지 후보의 움직임 벡터로 설정될 수 있다.
- [325] 반면, 현재 블록이 이용할 수 있는 머지 후보의 개수가 1개인 경우, 머지 후보를 특정하기 위한 정보의 시그널링을 생략할 수 있다. 일 예로, 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수가 1보다 크지 않은 경우, 머지 후보를 특정하기 위한 정보 `merge_idx`의 시그널링을 생략할 수 있다. 즉, 머지 오프셋 부호화 방법 하에서, 머지 후보 리스트에 1개의 머지 후보가 포함되어 있을 경우, 머지 후보를 특정하기 위한 정보 `merge_idx`의 부호화를 생략하고, 머지 후보 리스트에 포함된 머지 후보를 기초로, 초기 움직임 벡터를 결정할 수 있다. 상기 머지 후보의 움직임 벡터가 현재 블록의 초기 움직임 벡터로 설정될 수 있다.
- [326] 다른 예로, 현재 블록의 머지 후보를 결정한 뒤, 현재 블록에 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 결정할 수도 있다. 일 예로, 머지 후보가 포함할 수 있는 머지 후보의 최대 개수가 1보다 큰 경우, 머지 후보들 중 어느 하나를 특정하기 위한 정보 `merge_idx`를 시그널링할 수 있다. `merge_idx`를 기초로 머지 후보를 선택한 이후, 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는지 여부를 나타내는 `merge_offset_vector_flag`를 부호화할 수 있다. 표 4는 상술한 실시예에 따른 선택스 테이블을 나타낸 도면이다.

[327] [324]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(slice_type != I) {	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_luma_mpm_flag [x0][y0]	
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
} else { /* MODE_INTER */	
if(cu_skip_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx [x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if(merge_idx < 2 && merge_offset_vector_flag) {	
distance_idx [x0][y0]	ae(v)
direction_idx [x0][y0]	ae(v)
}	
}	

} else {	
merge_flag [x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx [x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if (merge_idx < 2 && merge_offset_vector_flag) {	
distance_idx [x0][y0]	ae(v)
direction_idx [x0][y0]	ae(v)
}	
}	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
}	

- [328] 다른 예로, 현재 블록의 머지 후보를 결정한 뒤, 결정된 머지 후보의 인덱스가 머지 오프셋 벡터 부호화 방법 적용시 이용할 수 있는 머지 후보들의 최대 개수보다 작은 경우에 한하여, 현재 블록에 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 결정할 수도 있다. 일 예로, 인덱스 정보 **merge_idx**의 값이 N보다 작은 경우에 한하여, 현재 블록에 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 나타내는 **merge_offset_vector_flag**를 부호화하여 시그널링할 수 있다. 인덱스 정보 **merge_idx**의 값이 N과 같거나 N보다 큰 경우, **merge_offset_vector_flag**의 부호화가 생략될 수 있다. **merge_offset_vector_flag**의 부호화가 생략된 경우, 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되지 않는 것으로 결정될 수 있다.
- [329] 또는, 현재 블록의 머지 후보를 결정한 뒤, 결정된 머지 후보가 양방향 움직임 정보를 갖는지 여부 또는 단방향 움직임 정보를 갖는지 여부를 고려하여, 현재 블록에 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 결정할 수 있다.

일 예로, 인덱스 정보 `merge_idx`의 값이 N보다 작고, 상기 인덱스 정보에 의해 선택된 머지 후보가 양방향 움직임 정보를 갖는 경우에 한하여, 현재 블록에 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 나타내는

`merge_offset_vector_flag`를 부호화하여 시그널링할 수 있다. 또는, 인덱스 정보 `merge_idx`의 값이 N보다 작고, 상기 인덱스 정보에 의해 선택된 머지 후보가 단방향 움직임 정보를 갖는 경우에 한하여, 현재 블록에 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 나타내는 `merge_offset_vector_flag`를 부호화하여 시그널링할 수 있다.

[330] 또는, 현재 블록의 크기, 형태 또는 현재 블록이 코딩 트리 유닛의 경계에 접하는지 여부 중 적어도 하나에 기초하여, 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 결정할 수 있다. 현재 블록의 크기, 형태 또는 현재 블록이 코딩 트리 유닛의 경계에 접하는지 여부 중 적어도 하나가 기 설정된 조건을 만족하지 않는 경우, 현재 블록에 머지 오프셋 벡터 부호화 방법을 적용할 것인지 여부를 나타내는 `merge_offset_vector_flag`의 부호화를 생략할 수 있다.

[331] 머지 후보가 선택되면, 머지 후보의 움직임 벡터를 현재 블록의 초기 움직임 벡터로 설정할 수 있다. 그리고, 오프셋 벡터의 크기를 나타내는 정보 및 오프셋 벡터의 방향을 나타내는 정보를 복호화하여, 오프셋 벡터를 결정할 수 있다. 오프셋 벡터는 수평 방향 성분 또는 수직 방향 성분을 가질 수 있다.

[332] 오프셋 벡터의 크기를 나타내는 정보는 벡터 크기 후보들 중 어느 하나를 나타내는 인덱스 정보일 수 있다. 일 예로, 벡터 크기 후보들 중 어느 하나를 나타내는 인덱스 정보 `distance_idx`가 비트스트림을 통해 시그널링될 수 있다. 표 5는 인덱스 정보 `distance_idx`의 이진화 및 `distance_idx`에 따른 오프셋 벡터의 크기를 결정하기 위한 변수 `DistFromMergeMV`의 값을 나타낸다.

[333] [표5]

<code>distance_idx[x][y]</code>	binarization	<code>DistFromMergeMV[x0][y0]</code>
0	0	1
1	10	2
2	110	4
3	1110	8
4	11110	16
5	111110	32
6	1111110	64
7	1111111	128

[334] 오프셋 벡터의 크기는 변수 `DistFromMergeMV`를 기 설정된 값으로 나누어

유도될 수 있다. 수학식 5은 오프셋 벡터의 크기를 결정하는 예를 나타낸다.

[335] [수식5]

$$abs(offsetMV) = DistFromMergeMV \ll 2$$

[336] 상기 수학식 5를 따르면, 변수 DistFromMegeMV를 4로 나눈 값 또는 변수 DistFromMergeMV를 좌측으로 2만큼 비트 시프팅한 값을 오프셋 벡터의 크기로 설정할 수 있다.

[337] 표 5에 나타난 예보다 더 많은 수의 벡터 크기 후보들 또는 더 적은 수의 벡터 크기 후보들을 사용하거나, 움직임 벡터 오프셋 크기 후보들의 범위를 표 5에 나타난 예와 상이하게 설정하는 것도 가능하다. 일 예로, 오프셋 벡터의 수평 방향 성분 또는 수직 방향 성분의 크기가 2 샘플 거리보다 크지 않도록 설정할 수 있다. 표 6은 인덱스 정보 distance_idx의 이진화 및 distance_idx에 따른 오프셋 벡터의 크기를 결정하기 위한 변수 DistFromMergeMV의 값을 나타낸다.

[338] [표6]

distance_idx[x][y]	binarization	DistFromMergeMV[x0][y0]
0	0	1
1	10	2
2	110	4
3	111	8

[339] 또는, 움직임 벡터 정밀도에 기초하여, 움직임 벡터 오프셋 크기 후보들의 범위를 상이하게 설정할 수 있다. 일 예로, 현재 블록에 대한 움직임 벡터 정밀도가 소수 펠(fractional-pel)인 경우, 인덱스 정보 distance_idx의 값들에 대응하는 변수 DistFromMergeMV의 값들은 1, 2, 4, 8, 16 등으로 설정될 수 있다. 여기서, 소수 펠은 1/16 펠(pel), 옥토 펠(Octo-pel), 쿼터 펠(Quarter-pel) 또는 하프펠(Half-pel) 중 적어도 하나를 포함한다. 반면, 현재 블록에 대한 움직임 벡터 정밀도가 정수 펠(integer-pel)인 경우, 인덱스 정보 distance_idx의 값들에 대응하는 변수 DistFromMergeMV의 값들은 4, 8, 16, 32, 64 등으로 설정될 수 있다. 즉, 현재 블록에 대한 움직임 벡터 정밀도에 따라, 변수 DistFromMergeMV를 결정하기 위해 참조하는 테이블이 상이하게 설정될 수 있다.

[340] 일 예로, 현재 블록 또는 머지 후보의 움직임 벡터 정밀도가 쿼터 펠인 경우에는 표 5를 이용하여 distance_idx가 가리키는 변수 DistFromMergeMV를 유도할 수 있다. 반면, 현재 블록 또는 머지 후보의 움직임 벡터 정밀도가 정수 펠인 경우에는 표 5에서 distance_idx가 가리키는 변수 DistFromMergeMV의 값에 N배(예컨대, 4배)를 취한 값을 변수 DistFromMergeMV의 값으로 유도할 수 있다.

[341] 움직임 벡터 정밀도를 결정하기 위한 정보가 비트스트림을 통해 시그널링될 수

있다. 일 예로, 상기 정보는, 시퀀스, 픽처, 슬라이스 또는 블록 레벨에서 시그널링될 수 있다. 이에 따라, 벡터 크기 후보들의 범위는, 비트스트림을 통해 시그널링되는 움직임 벡터 정밀도 관련 정보에 의해 상이하게 설정될 수 있다. 또는, 현재 블록의 머지 후보에 기초하여, 움직임 벡터 정밀도가 결정될 수 있다. 일 예로, 현재 블록의 움직임 벡터 정밀도는, 머지 후보의 움직임 벡터 정밀도와 동일하게 설정될 수 있다.

- [342] 또는, 오프셋 벡터의 탐색 범위를 결정하기 위한 정보를 비트스트림을 통해 시그널링할 수 있다. 탐색 범위에 기초하여 벡터 크기 후보들의 개수, 벡터 크기 후보들 중 최소값 또는 최대값 중 적어도 하나가 결정될 수 있다. 일 예로, 오프셋 벡터의 탐색 범위를 결정하기 위한 플래그 `merge_offset_vector_flag`가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 시퀀스 헤더, 픽처 헤더 또는 슬라이스 헤더를 통해 시그널링될 수 있다.
- [343] 일 예로, `merge_offset_extend_range_flag`의 값이 0이면, 오프셋 벡터의 크기가 2를 초과하지 않도록 설정될 수 있다. 이에 따라, `DistFromMergeMV`의 최대값은 8로 설정될 수 있다. 반면, `merge_offset_extend_range_flag` 값이 1 이면, 오프셋 벡터의 크기가 32 샘플 거리를 초과하지 않도록 설정될 수 있다. 이에 따라, `DistFromMergeMV`의 최대값은 128로 설정될 수 있다.
- [344] 오프셋 벡터의 크기가 문턱값보다 큰지 여부를 나타내는 플래그를 이용하여, 오프셋 벡터의 크기를 결정할 수 있다. 일 예로, 오프셋 벡터의 크기가 문턱값보다 큰지 여부를 나타내는 플래그 `distance_flag`가 비트스트림을 통해 시그널링될 수 있다. 문턱값은 1, 2, 4, 8 또는 16일 수 있다. 일 예로, `distance_flag`가 1인 것은, 오프셋 벡터의 크기가 4 보다 큼을 나타낸다. 반면, `distance_flag`가 0인 것은, 오프셋 벡터의 크기가 4 이하임을 나타낸다.
- [345] 오프셋 벡터의 크기가 문턱값보다 큰 경우, 인덱스 정보 `distance_idx`를 이용하여, 오프셋 벡터의 크기와 문턱값 사이의 차분값을 유도할 수 있다. 또는, 오프셋 벡터의 크기가 문턱값 이하인 경우, 인덱스 정보 `distance_idx`를 이용하여, 오프셋 벡터의 크기를 결정할 수 있다. 표 7은 `distance_flag` 및 `distance_idx`의 부호화 양상을 나타낸 선택스 테이블이다.

[346] [347]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(slice_type != I) {	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_luma_mpm_flag [x0][y0]	
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
} else { /* MODE_INTER */	
if(cu_skip_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1)	
{	
merge_idx [x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if(merge_idx < 2 && merge_offset_vector_flag) {	
distance_flag [x0][y0]	ae(v)
distance_idx [x0][y0]	
direction_idx [x0][y0]	ae(v)
}	

}	
} else {	
merge_flag [x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx [x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if (merge_idx < 2 && merge_offset_vector_flag) {	
distance_flag [x0][y0]	ae(v)
distance_idx [x0][y0]	ae(v)
direction_idx [x0][y0]	ae(v)
}	
}	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
}	

- [347] 수학식 6은 distance_flag과 distance_idx를 이용하여, 오프셋 벡터의 크기를 결정하기 위한 변수 DistFromMergeMV를 유도하는 예를 나타낸 것이다.
- [348] [수식6]
- $$DistFromMergeMV = N * distance_flag + (1 \ll distance_idx)$$
- [349] 수학식 6에서, distance_flag의 값은 1 또는 0으로 설정될 수 있다. distance_idx의 값은 1, 2, 4, 8, 16, 32, 64, 128 등으로 설정될 수 있다. N은 문턱값에 의해 결정되는 계수를 나타낸다. 일 예로, 문턱값이 4인 경우, N은 16으로 설정될 수 있다.
- [350] 오프셋 벡터의 방향을 나타내는 정보는 벡터 방향 후보들 중 어느 하나를 나타내는 인덱스 정보일 수 있다. 일 예로, 벡터 방향 후보들 중 어느 하나를

나타내는 인덱스 정보 *direction_idx*가 비트스트림을 통해 시그널링될 수 있다. 표 8은 인덱스 정보 *direction_idx*의 이진화 및 *direction_idx*에 따른 오프셋 벡터의 방향을 나타낸다.

[351] [표8]

<i>direction_idx</i> [x][y]	binarization	<i>sign</i> [x][y][0]	<i>sign</i> [x][y][1]
0	00	+1	0
1	01	-1	0
2	10	0	+1
3	11	0	-1

[352] 표 8에서 *sign*[0]은 수평 방향을 나타내고, *sign*[1]은 수직 방향을 나타낸다. +1은 오프셋 벡터의 x 성분 또는 y 성분의 값이 +임을 나타내고, -1은 오프셋 벡터의 x 성분 또는 y 성분의 값이 -임을 나타낸다. 수학식 7은 오프셋 벡터의 크기 및 방향에 기초하여 오프셋 벡터를 결정하는 예를 나타낸다.

[353] [수식7]

$$\text{offsetMV}[0] = \text{abs}(\text{offsetMV}) * \text{sign}[0]$$

$$\text{offsetMV}[1] = \text{abs}(\text{offsetMV}) * \text{sign}[1]$$

[354] 수학식 7에서 *offsetMV*[0]은 오프셋 벡터의 수직 방향 성분을 나타내고, *offsetMV*[1]은 오프셋 벡터의 수평 방향 성분을 나타낸다.

[355] 도 28은 오프셋 벡터의 크기를 나타내는 *distance_idx* 및 오프셋 벡터의 방향을 나타내는 *direction_idx*의 값에 따른 오프셋 벡터를 나타낸 도면이다.

[356] 도 28에 도시된 예에서와 같이, 오프셋 벡터의 크기 및 방향은 *distance_idx* 및 *direction_idx*의 값에 따라 결정될 수 있다. 오프셋 벡터의 최대 크기는 문턱값을 초과하지 않도록 설정될 수 있다. 여기서, 문턱값은 부호화기 및 복호화기에서 기 정의된 값을 가질 수 있다. 일 예로, 문턱값은 32 샘플 거리일 수 있다. 또는, 초기 움직임 벡터의 크기에 따라, 문턱값이 결정될 수 있다. 일 예로, 수평 방향에 대한 문턱값은 초기 움직임 벡터의 수평 방향 성분의 크기를 기초로 설정되고, 수직 방향에 대한 문턱값은 초기 움직임 벡터의 수직 방향 성분의 크기를 기초로 설정될 수 있다.

[357] 머지 후보가 양방향 움직임 정보를 갖는 경우, 머지 후보의 L0 움직임 벡터를 현재 블록의 L0 초기 움직임 벡터로 설정하고, 머지 후보의 L1 움직임 벡터를 현재 블록의 L1 초기 움직임 벡터로 설정할 수 있다. 이때, 머지 후보의 L0 참조 픽처와 현재 픽처와의 출력 순서 차분값(이하, L0 차분값이라 함)과 머지 후보의 L1 참조 픽처와 현재 픽처와의 출력 순서 차분값(이하, L1 차분값이라 함)을 고려하여, L0 오프셋 벡터 및 L1 오프셋 벡터를 결정할 수 있다.

[358] 먼저, L0 차분값 및 L1 차분값의 부호화 동일한 경우, L0 오프셋 벡터와 L1 오프셋 벡터는 동일하게 설정될 수 있다. 반면, L0 차분값 및 L1 차분값의 부호가

상이한 경우, L1 오프셋 벡터는 L0 오프셋 벡터와 반대 방향으로 설정될 수 있다.

[359] L0 오프셋 벡터의 크기와 L1 오프셋 벡터의 크기는 동일하게 설정될 수 있다.
또는, L0 차분값과 L1 차분값에 기초하여, L0 오프셋 벡터를 스케일링함으로써,
L1 오프셋 벡터의 크기를 결정할 수 있다.

[360] 일 예로, 수학식 8은 L0 차분값과 L1 차분값의 부호가 동일한 경우, L0 오프셋 벡터 및 L1 오프셋 벡터를 나타낸 것이다.

[361] [수식8]

$$\begin{aligned} offsetMVL0[0] &= abs(offsetMV) * sign[0] \\ offsetMVL0[1] &= abs(offsetMV) * sign[1] \\ offsetMVL1[0] &= abs(offsetMV) * sign[0] \\ offsetMVL1[1] &= abs(offsetMV) * sign[1] \end{aligned}$$

[362] 수학식 8에서, offsetMVL0[0]은 L0 오프셋 벡터의 수평 방향 성분을 나타내고, offsetMVL0[1]은 L0 오프셋 벡터의 수직 방향 성분을 나타낸다. offsetMVL1[0]은 L1 오프셋 벡터의 수평 방향 성분을 나타내고, offsetMVL1[1]은 L1 오프셋 벡터의 수직 방향 성분을 나타낸다.

[363] 수학식 9는 L0 차분값과 L1 차분값의 부호가 상이한 경우, L0 오프셋 벡터 및 L1 오프셋 벡터를 나타낸 것이다.

[364] [수식9]

$$\begin{aligned} offsetMVL0[0] &= abs(offsetMV) * sign[0] \\ offsetMVL0[1] &= abs(offsetMV) * sign[1] \\ offsetMVL1[0] &= -1 * abs(offsetMV) * sign[0] \\ offsetMVL1[1] &= -1 * abs(offsetMV) * sign[1] \end{aligned}$$

[365] 4개보다 더 많은 수의 벡터 방향 후보들을 정의할 수도 있다. 표 9 및 표 10은 8개의 벡터 방향 후보들이 정의된 예를 나타낸 것이다.

[366] [표9]

direction_idx[x][y]	binarization	sign[x][y][0]	sign[x][y][1]
0	000	+1	0
1	001	-1	0
2	010	0	+1
3	011	0	-1
4	100	+1	+1
5	101	+1	-1
6	110	-1	+1
7	111	-1	-1

[367] [표10]

direction_idx[x][y]	binarization	sign[x][y][0]	sign[x][y][1]
0	000	+1	0
1	001	-1	0
2	010	0	+1
3	011	0	-1
4	100	+1/2	+1/2
5	101	+1/2	-1/2
6	110	-1/2	+1/2
7	111	-1/2	-1/2

[368] 표 9 및 표 10에서 sign[0] 및 sign[1]의 절대값이 0보다 큰 것은 오프셋 벡터가 대각 방향임을 나타낸다. 표 8이 이용되는 경우, 대각 방향 오프셋 벡터의 x축 및 y축 성분의 크기는 abs(offsetMV)로 설정되는 반면, 표 9가 이용되는 경우, 대각 방향 오프셋 벡터의 x축 및 y축 성분의 크기는 abs(offsetMV/2)로 설정될 수 있다.

[369] 도 29는 오프셋 벡터의 크기를 나타내는 distance_idx 및 오프셋 벡터의 방향을 나타내는 direction_idx의 값에 따른 오프셋 벡터를 나타낸 도면이다.

[370] 도 29의 (a)는 표 8가 적용되는 경우의 예시이고, 도 29의 (b)는 표 9가 적용되는 경우의 예시이다.

[371] 벡터 방향 후보들의 개수 또는 크기 중 적어도 하나를 결정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 벡터 방향 후보들을 결정하기 위한 플래그 merge_offset_direction_range_flag가 비트스트림을 통해 시그널링될 수 있다. 상기 플래그는 시퀀스, 픽처 또는 슬라이스 레벨에서 시그널링될 수 있다. 일 예로, 상기 플래그의 값이 0인 경우, 표 8에 예시된 4개의 벡터 방향 후보들을 이용할 수 있다. 반면, 상기 플래그의 값이 1인 경우, 표 9 또는 표 10에 예시된 8개의 벡터 방향 후보들을 이용할 수 있다.

[372] 또는, 오프셋 벡터의 크기에 기초하여, 벡터 방향 후보들의 개수 또는 크기 중 적어도 하나를 결정할 수 있다. 일 예로, 오프셋 벡터의 크기를 결정하기 위한 변수 DistFromMergeMV의 값이 문턱값과 같거나 문턱값보다 작은 경우, 표 9 또는 표 10에 예시된 8개의 벡터 방향 후보들을 사용할 수 있다. 반면, 변수 DistFromMergeMV의 값이 문턱값보다 큰 경우, 표 8에 예시된 4개의 벡터 방향 후보들을 사용할 수 있다.

[373] 또는, 초기 움직임 벡터의 x 성분의 값 MVx 및 y 성분의 값 MVy에 기초하여, 벡터 방향 후보들의 개수 또는 크기 중 적어도 하나를 결정할 수 있다. 일 예로, MVx 및 MVy 차분 또는 차분의 절대값이 문턱값 이하인 경우, 표 9 또는 표 10에 예시된 8개의 벡터 방향 후보들을 사용할 수 있다. 반면, MVx 및 MVy 차분 또는

차분의 절대값이 문턱값보다 큰 경우, 표 8에 예시된 4개의 벡터 방향 후보들을 사용할 수 있다.

[374] 현재 블록의 움직임 벡터는 초기 움직임 벡터에 오프셋 벡터를 더하여 유도될 수 있다. 수학식 10은 현재 블록의 움직임 벡터를 결정하는 예를 나타낸다.

[375] [수식10]

$$mvL0[0] = mergeMVL0[0] + offsetMVL0[0]$$

$$mvL0[1] = mergeMVL0[1] + offsetMVL0[1]$$

$$mvL1[0] = mergeMVL1[0] + offsetMVL1[0]$$

$$mvL1[1] = mergeMVL1[1] + offsetMVL1[1]$$

[376] 수학식 10에서, mvL0는 현재 블록의 L0 움직임 벡터를 나타내고, mvL1은 현재 블록의 L1 움직임 벡터를 나타낸다. mergeMVL0는 현재 블록의 L0 초기 움직임 벡터(즉, 머지 후보의 L0 움직임 벡터)를 나타내고, mergeMVL1은 현재 블록의 L1 초기 움직임 벡터를 나타낸다. [0]은 움직임 벡터의 수평 방향 성분을 나타내고, [1]은 움직임 벡터의 수직 방향 성분을 나타낸다.

[377] 코딩 블록을 복수의 서브 유닛들로 분할한 뒤, 서브 유닛들 각각에 대해 인터 예측을 수행하는 경우에도, 머지 오프셋 벡터 부호화 방법을 적용할 수 있다. 여기서, 서브 유닛 단위로 인터 예측을 수행하는 것은, ATMVP (Advanced Temporal Motion Vector Prediction), STMVP (Spatial Temporal Motion Vector Prediction) 또는 삼각 파티셔닝 기법 중 적어도 하나를 포함할 수 있다.

[378] 일 예로, ATMVP 방법 하에서, 초기 움직임 벡터는 다음과 같이 유도될 수 있다.

[379] 먼저, 코딩 블록에 인접하는 이웃 블록으로부터 유도된 머지 후보의 움직임 벡터를 이용하여, 초기 쉬프트 벡터를 유도할 수 있다. 그리고, 초기 쉬프트 벡터를 이용하여 코딩 블록에 포함된 서브 블록의 쉬프트 블록을 유도할 수 있다. 수학식 11은 쉬프트 블록의 위치를 나타낸 도면이다.

[380] [수식11]

$$(xColSb, yColSb) = (xSb + shVector[0] >> 4, ySb + shVector[1] >> 4)$$

[381] 상기 수학식 11에서, (xColSb, yColSb)는 쉬프트 블록의 좌측 상단 샘플의 위치를 나타내고, (xSb, ySb)는 서브 블록의 좌측 상단 샘플의 위치를 나타낸다. shVector는 쉬프트 벡터를 나타낸다.

[382] 쉬프트 블록이 결정되면, 콜로케이트드 픽처 내 쉬프트 블록과 동일한 위치인 콜로케이트드 블록의 움직임 벡터는 서브 블록의 움직임 벡터로 설정할 수 있다. 즉, 콜로케이트드 블록 내 (xColSb, yColSb) 위치의 샘플을 포함하는 콜로케이트드 블록의 움직임 벡터는 (xSb, ySb) 위치의 샘플을 포함하는 서브 블록의 움직임 벡터로 설정할 수 있다.

[383] 삼각 파티셔닝 기법이 적용되는 경우, 코딩 블록은 삼각 형태인 서브 유닛들로 분할될 수 있다. 일 예로, 코딩 블록은 코딩 블록의 좌상단 및 우하단을 잇는 대각선 또는 코딩 블록의 우상단 및 좌하단을 잇는 대각선에 의해 두개의 서브 유닛들로 분할될 수 있다.

- [384] 도 30은 삼각 파티셔닝 기법이 적용되는 경우, 코딩 블록의 분할 양상을 나타낸 도면이다.
- [385] 삼각 형태의 서브 유닛들 각각의 움직임 정보는 머지 후보에 의해 특정될 수 있다. 이를 위해, 각 서브 유닛에 대해 머지 후보들 중 어느 하나를 나타내는 인덱스 정보가 시그널링될 수 있다. 일 예로, 제1 서브 유닛의 인덱스 정보 `merge_1st_idx`는 제1 서브 유닛의 머지 후보를 특정하고, 제2 서브 유닛의 인덱스 정보 `merge_2nd_idx`는 제2 서브 유닛의 머지 후보를 특정할 수 있다.
- [386] 서브 유닛들 각각의 초기 움직임 벡터를 개별적으로 결정할 수 있다. 일 예로, 코딩 블록에 어파인 모션 모델이 적용된 경우, 코딩 블록의 어파인 시드 벡터들로부터 유도된 서브 블록들의 어파인 벡터를 서브 블록들의 초기 움직임 벡터로 설정할 수 있다. 각 서브 블록의 움직임 벡터는 초기 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 유도될 수 있다.
- [387] 복수의 서브 유닛들로 분할된 코딩 블록에 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 복수의 서브 유닛들은 동일한 오프셋 벡터를 사용하도록 설정될 수 있다. 즉, 복수의 서브 유닛들 각각의 초기 움직임 벡터는 동일한 오프셋 벡터를 이용하여 수정될 수 있다.
- [388] 또는, 코딩 블록을 복수의 서브 유닛들로 분할하고, 각 서브 유닛의 오프셋 벡터를 개별적으로 결정할 수 있다. 이에 따라, 서브 유닛 중 적어도 하나의 오프셋 벡터가 다른 서브 유닛의 오프셋 벡터와 상이하게 설정될 수 있다.
- [389] 도 31은 서브 유닛들 각각의 오프셋 벡터가 상이하게 설정된 예를 나타낸 도면이다.
- [390] 도 31에 도시된 예에서와 같이, 서브 유닛 별로 오프셋 벡터의 크기를 나타내는 정보 `distance_idx` 및 오프셋 벡터의 방향을 나타내는 정보 `direction_idx`를 부호화하여 시그널링할 수 있다.
- [391] 또는, 모든 서브 유닛들에 대한 오프셋 벡터의 크기를 동일하게 설정하되, 서브 유닛들에 대한 오프셋 벡터의 방향은 개별적으로 설정할 수 있다. 일 예로, 서브 유닛들이 코딩 레벨에서 시그널링되는 `distance_idx`의 값을 공유하도록 설정하고, 각 서브 유닛별로 `direction_idx`를 부호화하여 시그널링할 수 있다.
- [392] 또는, 모든 서브 유닛들에 대한 오프셋 벡터의 방향을 동일하게 설정하되, 서브 유닛들에 대한 오프셋 벡터의 크기는 개별적으로 설정할 수 있다. 일 예로, 서브 유닛들이 코딩 레벨에서 시그널링되는 `direction_idx`의 값을 공유하도록 설정하고, 각 서브 유닛별로 `distance_idx`를 부호화하여 시그널링할 수 있다.
- [393] 코딩 블록을 분할하여 생성된 복수의 서브 유닛들 중 일부에만 머지 오프셋 벡터 부호화 방법을 적용할 수도 있다. 일 예로, 현재 블록이 제1 서브 유닛 및 제2 서브 유닛으로 분할된 경우, 제1 서브 유닛의 움직임 벡터를 머지 후보의 움직임 벡터와 동일하게 설정하고, 제2 서브 유닛의 움직임 벡터는 머지 후보의 움직임 벡터에 오프셋 벡터를 가산하여 유도할 수 있다.
- [394] 오프셋 벡터를 결정하기 위한 정보를 시그널링하는 대신, 복호화기에서 오프셋

벡터를 유도할 수도 있다. 구체적으로, 서브 블록에 포함된 예측 샘플들의 수평 방향 그라디언트(**gradient**)에 대한 평균값 및 수직 방향 그라디언트(**gradient**)에 대한 평균값을 이용하여, 오프셋 벡터를 유도할 수 있다.

- [395] 여기서, 그라디언트는, 참조 픽처 내 예측 샘플에 대응하는 복원 샘플 및 상기 복원 샘플에 이웃하는 이웃 샘플의 차분을 기초로 유도될 수 있다. 일 예로, 수평 방향 그라디언트는 복원 샘플과 좌측 및/또는 우측에 이웃하는 복원 샘플과의 차분을 나타내고, 수직 방향 그라디언트는 복원 샘플과 상단 및/또는 하단에 이웃하는 복원 샘플과의 차분을 나타낼 수 있다.
- [396] 머지 후보 리스트에 포함된 머지 후보들 중 기준 머지 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 유도된 움직임 벡터를 갖는 머지 후보를 머지 후보 리스트에 추가할 수 있다. 기준 머지 후보의 움직임 벡터에서 오프셋 벡터를 가산 또는 감산하여 유도된 움직임 벡터를 갖는 머지 후보를 리파인 머지 후보라 호칭할 수 있다.
- [397] 리파인 머지 후보의 움직임 벡터를 제외한 잔여 움직임 정보는, 기준 머지 후보와 동일하게 설정될 수 있다.
- [398] 도 32는 리파인 머지 후보가 취할 수 있는 움직임 벡터 후보들을 나타낸 도면이다.
- [399] 기준 머지 후보의 움직임 벡터가 ($MvLX[0]$, $MvLX[1]$)인 경우, 리파인 머지 후보의 움직임 벡터는 기준 머지 후보의 움직임 벡터의 x성분 또는 y성분 적어도 하나에 오프셋을 가산 또는 감산하여 유도될 수 있다. 일 예로, 리파인 머지 후보의 움직임 벡터는 ($MvLX[0]+M$, $MvLX[1]$), ($MvLX[0]-M$, $MvLX[1]$), ($MvLX[0]$, $MvLX[1]+M$) 또는 ($MvLX[0]$, $MvLX[1]-M$)로 설정될 수 있다. M 은 오프셋 벡터의 크기를 나타낸다.
- [400] 기준 머지 후보는 머지 후보 리스트 내 기 정의된 인덱스 값을 갖는 머지 후보일 수 있다. 일 예로, 머지 후보 리스트에 포함된 머지 후보들 중 인덱스 값이 가장 작은 머지 후보(즉, 인덱스 값이 0인 머지 후보) 또는 인덱스 값이 가장 큰 머지 후보가 기준 머지 후보로 설정될 수 있다. 또는, 인터 영역 모션 정보 테이블 내 인덱스 값이 가장 작은 인터 영역 머지 후보 또는 인덱스 값이 가장 큰 인터 영역 머지 후보가 기준 머지 후보로 설정될 수 있다.
- [401] 또는, 양방향 움직임 정보를 갖는 머지 후보들 중 인덱스 값이 가장 작은 머지 후보를 기준 머지 후보로 설정할 수 있다. 즉, 후보 블록들을 순차적으로 탐색하였을 때, 첫번째로 발견한 양방향 머지 후보를 기준 머지 후보로 설정할 수 있다.
- [402] 현재 블록의 크기, 형태 또는 코딩 트리 유닛의 경계에 접하는지 여부에 기초하여, 기본 머지 후보를 선택할 수도 있다. 일 예로, 현재 블록이 정방향인 경우 또는 현재 블록이 높이가 너비보다 큰 비정방향인 경우, 인덱스가 0인 머지 후보 또는 현재 블록의 상단에 위치하는 이웃 블록으로부터 유도된 머지 후보를 기준 머지 후보로 설정할 수 있다. 현재 블록이 너비가 높이보다 큰 비정방향인

경우, 인덱스가 1인 머지 후보 또는 현재 블록의 좌측에 위치하는 이웃 블록으로부터 유도된 머지 후보를 기준 머지 후보로 설정할 수 있다.

- [403] 또는, 기준 머지 후보를 특정하는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 머지 후보 리스트에 포함된 머지 후보들 중 어느 하나를 특정하는 인덱스 정보일 수 있다.
- [404] 리파인 머지 후보를 사용할 것인지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그일 수 있다. 상기 플래그의 값이 1인 경우, 머지 후보 리스트에 기준 머지 후보를 기초로 생성된 리파인 머지 후보를 추가할 수 있다. 반면, 상기 플래그의 값이 0인 경우, 머지 후보 리스트가 리파인 머지 후보를 포함하지 않을 수 있다.
- [405] 또는, 머지 후보 리스트에 기 추가된 머지 후보들의 개수가 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수보다 작은 경우, 머지 후보 리스트에 리파인 머지 후보를 추가할 수 있다. 여기서, 기 추가된 머지 후보들은 공간적 머지 후보, 시간적 머지 후보, 인터 영역 머지 후보 또는 페어 와이즈 머지 후보 중 적어도 하나를 포함할 수 있다. 일 예로, 머지 후보 리스트에 포함된 공간적 머지 후보, 시간적 머지 후보 또는 인터 영역 머지 후보 중 적어도 하나의 개수가 문턱값 이하인 경우, 리파인 머지 후보를 머지 후보 리스트에 추가할 수 있다.
- [406] 또는, 머지 후보 리스트에 기 추가된 머지 후보들의 개수가 문턱값 이상인 경우에 리파인 머지 후보를 이용할 수도 있다.
- [407] 리파인 머지 후보의 이용 여부에 따라, 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수가 상이하게 설정될 수 있다. 일 예로, 리파인 머지 후보를 이용하지 않도록 설정된 경우에는 머지 후보가 포함할 수 있는 머지 후보들의 최대 개수가 N 개로 설정되는 반면, 리파인 머지 후보를 이용하도록 설정된 경우에는 머지 후보가 포함할 수 있는 머지 후보들의 최대 개수가 $N+n$ 개로 설정될 수 있다.
- [408] 리파인 머지 후보는 머지 후보 리스트에 기 추가된 머지 후보들보다 큰 인덱스를 가질 수 있다. 일 예로, 표 11은 머지 후보 리스트를 구성 예를 나타낸 것이다.

[409] [표 11]

mergeCand[0]
mergeCand[1]
mergeCand[2]
mergeCand[3]
mergeCand[4]
mergeCand[5]
mergeCand[6] : 움직임 벡터가 (mvLx[0]+M, mvLx[1])인 리파인 머지 후보
mergeCand[7] : 움직임 벡터가 (mvLx[0]-M, mvLx[1])인 리파인 머지 후보
mergeCand[8] : 움직임 벡터가 (mvLx[0], mvLx[1]+M)인 리파인 머지 후보
mergeCand[9] : 움직임 벡터가 (mvLx[0], mvLx[1]-M)인 리파인 머지 후보

[410] 표 11에서 mergeCand[X]는 인덱스가 X인 머지 후보를 나타낸다. mvLx[0]는 기준 머지 후보의 x 성분 움직임 벡터를 나타내고, mvLx[1]은 기준 머지 후보의 y 성분 움직임 벡터를 나타낸다. 일 예로, 기준 머지 후보가 mergeCand[0]인 경우, mvLx[0] 및 mvLx[1]은 mergeCand[0]의 움직임 벡터를 나타낼 수 있다.

[411] 오프셋 벡터의 크기 M은 부호화기 및 복호화기에서 기 정의될 수 있다. 일 예로, 오프셋 벡터의 크기 M은 1 또는 4와 같이, 4 이하의 정수로 설정될 수 있다.

[412] 또는, 오프셋 벡터를 결정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는, 시퀀스, 픽처, 슬라이스 또는 블록 레벨에서 시그널링될 수 있다. 일 예로, 상술한 오프셋 벡터의 크기를 결정하기 위한 정보 distance_idx 또는 오프셋 벡터의 방향을 결정하기 위한 정보 direction_idx 중 적어도 하나를 이용하여 오프셋 벡터를 결정할 수 있다.

[413] 표 11에 나타난 예에서와 같이, 기준 머지 후보를 기초로 유도된 적어도 하나의 리파인 머지 후보를 머지 후보 리스트에 추가할 수 있다. 기 추가된 머지 후보들 중 리파인 머지 후보와 움직임 정보가 동일한 머지 후보가 존재하는 경우, 리파인 머지 후보를 머지 후보 리스트에 추가하지 않을 수 있다. 일 예로, 기준 머지 후보 mergeCand[0]를 기초로 유도된 리파인 머지 후보가 mergeCand[1] 내지 mergeCand[5] 중 어느 하나와 동일한 경우, 리파인 머지 후보를 머지 후보 리스트에 추가하지 않을 수 있다.

[414] 또는, 리파인 머지 후보와 움직임 정보가 동일한 머지 후보가 존재하는 경우, 오프셋 벡터를 변경하거나, 리파인 머지 후보와 움직임 정보가 동일한 머지 후보를 리파인 머지 후보로 재설정하여, 리파인 머지 후보를 재유도할 수 있다. 일 예로, 기준 머지 후보 mergeCand[0]를 기초로 유도된 리파인 머지 후보 mergeCand[6]의 움직임 정보가 머지 후보 mergeCand[2]와 동일한 경우, 리파인 머지 후보 mergeCand[6]의 움직임 벡터를 머지 후보 mergeCand[2]의 움직임

벡터에 오프셋 벡터를 가산 또는 감산한 값으로 변경할 수 있다. 일 예로, mergeCand[6]의 움직임 벡터를 (mergeCand[0]_mxLx[0]+M, mergeCand[0]_mvLx[1])에서 (mergeCand[2]_mxLx[0]+M, mergeCand[2]_mvLx[1])으로 변경할 수 있다. 여기서, mergeCand[X]_mvLx는 인덱스가 X인 머지 후보의 움직임 벡터를 나타낸다.

[415]

[416] 다른 예로, 적어도 하나 이상의 머지 오프셋 후보를 포함하는 머지 리파인먼트 오프셋 리스트를 이용하여, 오프셋 벡터를 결정할 수 있다. 현재 블록의 인덱스 정보에 의해 특정된 머지 후보가 기준 머지 후보인 경우, 머지 리파인먼트 오프셋 리스트를 이용하여 오프셋 벡터를 결정할 수 있다. 그리고, 머지 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여, 현재 블록의 움직임 벡터를 유도할 수 있다. 기준 머지 후보는 머지 후보 리스트 내 기 정의된 인덱스 값을 갖는 머지 후보일 수 있다. 일 예로, 머지 후보 리스트에 포함된 머지 후보들 중 인덱스 값이 가장 작은 머지 후보(즉, 인덱스 값이 0인 머지 후보) 또는 인덱스 값이 가장 큰 머지 후보가 기준 머지 후보로 설정될 수 있다. 또는, 인터 영역 모션 정보 테이블 내 인덱스 값이 가장 작은 인터 영역 머지 후보 또는 인덱스 값이 가장 큰 인터 영역 머지 후보가 기준 머지 후보로 설정될 수 있다.

[417] 도 33은 머지 리파인먼트 오프셋 리스트의 구성을 예시한 도면이다.

[418] 도 33에서 기준 머지 후보는 인덱스가 6인 머지 후보인 것으로 가정한다.

[419] 머지 후보들 중 어느 하나를 나타내는 인덱스 정보 merge_idx에 의해 특정된 머지 후보의 인덱스가 6이 아닌 경우, 머지 후보의 움직임 벡터를 현재 블록의 움직임 벡터로 설정할 수 있다.

[420] 반면, 인덱스 정보 merge_idx에 의해 특정된 머지 후보의 인덱스가 6인 경우, 머지 리파인먼트 오프셋 리스트를 이용하여 오프셋 벡터를 유도할 수 있다. 머지 리파인먼트 오프셋 리스트에 포함된 머지 오프셋 후보들 중 어느 하나를 특정하는 인덱스 정보 MrgOffset_idx가 비트스트림을 통해 시그널링될 수 있다.

[421] 오프셋 벡터가 특정되면, 기준 머지 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 현재 블록의 움직임 벡터를 유도할 수 있다.

[422] 머지 리파인먼트 오프셋 리스트는 적어도 하나 이상의 머지 오프셋 후보를 포함할 수 있다. 일 예로, 머지 리파인먼트 오프셋 리스트가 포함하는 머지 오프셋 후보들의 개수는 4개, 8개 또는 16개일 수 있다.

[423] 도 34 및 도 35는 머지 오프셋 후보들에 의해 특정되는 오프셋 벡터를 나타낸 도면이다.

[424] 도 34는 머지 오프셋 후보들의 개수가 8인 경우의 예를 나타내고, 도 35는 머지 오프셋 후보들의 개수가 16인 경우의 예를 나타낸다.

[425] 도 34의 (a)에 도시된 예에서와 같이, 머지 오프셋 후보들이 나타내는 오프셋 벡터들은 수평 방향의 움직임 벡터의 절대값 및/또는 수직 방향의 움직임 벡터의 절대값이 고정된 값을 갖도록 설정될 수 있다. 또는, 도 35에 도시된 예에서와

같이, 머지 오프셋 후보들 중 인덱스가 문턱값보다 작은 머지 오프셋 후보들이 나타내는 오프셋 벡터들은 수평 방향의 움직임 벡터의 절대값 및/또는 수직 방향의 움직임 벡터의 절대값이 제1 값을 갖도록 설정되고, 다른 머지 오프셋 후보들이 나타내는 오프셋 벡터들은 수평 방향의 움직임 벡터의 절대값 및/또는 수직 방향의 움직임 벡터의 절대값이 제2 값을 갖도록 설정될 수 있다.

[426] 또는, 도 34의 (b)에 도시된 예에서와 같이, 머지 오프셋 후보들이 나타내는 오프셋 벡터들은 수평 방향의 움직임 벡터의 절대값 및 수직 방향의 움직임 벡터의 절대값이 합이 고정된 값을 갖도록 설정될 수 있다.

[427] 복수개의 기준 머지 후보들을 설정할 수도 있다. 일 예로, 머지 후보 리스트에 포함된 머지 후보들 중 인덱스가 가장 작은 2개의 머지 후보들을 기준 머지 후보들로 설정할 수 있다. 이에 따라, 인덱스 정보 `merge_idx`에 의해 특정된 머지 후보의 인덱스가 0 또는 1인 경우, 머지 리퍼인먼트 오프셋 리스트를 이용하여 오프셋 벡터를 유도할 수 있다. 또는, 머지 후보 리스트에 포함된 머지 후보들 중 인덱스가 가장 작은 머지 후보 및 인터 영역 머지 후보 리스트에 포함된 머지 후보들 중 인덱스가 가장 큰 머지 후보를 기준 머지 후보들로 설정할 수 있다.

[428]

[429] 모션 벡터 예측 모드에서, 현재 블록의 움직임 벡터는 움직임 예측 벡터에 움직임 차분 벡터를 더하여 유도될 수 있다. 현재 블록의 움직임 예측 벡터는 적어도 하나의 움직임 예측 벡터 후보를 포함하는 모션 벡터 예측 후보 리스트를 기초로 결정될 수 있다. 일 예로, 움직임 예측 벡터 후보들 중 어느 하나를 현재 블록의 움직임 예측 벡터로 설정할 수 있다.

[430] 모션 벡터 예측 후보는 현재 블록의 공간적 이웃 블록 또는 현재 블록의 시간적 이웃 블록 중 적어도 하나를 기초로 유도될 수 있다.

[431] 도 36은 모션 벡터 예측 후보들을 유도하는데 이용되는 후보 블록들을 나타낸 도면이다.

[432] 공간적 이웃 블록은 현재 블록의 상단에 위치하는 상단 이웃 블록들 및 현재 블록의 좌측에 위치하는 좌측 이웃 블록들을 포함할 수 있다. 상단 이웃 블록들은 $(x_{Cb}+CbW, y_{Cb}-1)$ 위치의 샘플을 포함하는 블록 B0, $(x_{Cb}+CbW-1, y_{Cb}-1)$ 위치의 샘플을 포함하는 블록 B1, $(x_{Cb}-1, y_{Cb}-1)$ 위치의 샘플을 포함하는 블록 B2 또는 $(x_{Cb}, y_{Cb}-1)$ 위치의 샘플을 포함하는 블록 B3 중 적어도 하나를 포함할 수 있다. 여기서, (x_{Cb}, y_{Cb}) 는 현재 블록의 좌측 상단 샘플의 위치를 나타내고, CbW 는 현재 블록의 너비를 나타낸다. 좌측 이웃 블록들은 $(x_{Cb}-1, y_{Cb}+CbH)$ 위치의 샘플을 포함하는 블록 A0, $(x_{Cb}-1, y_{Cb}+CbH-1)$ 위치의 샘플을 포함하는 블록 A1 또는 $(x_{Cb}-1, y_{Cb})$ 위치의 샘플을 포함하는 블록 A2 중 적어도 하나를 포함할 수 있다. 여기서, CbH 는 현재 블록의 높이를 나타낸다.

[433] 시간적 이웃 블록은 콜로케이터드 블록 내 현재 블록과 동일한 위치 및 크기를 갖는 블록의 중앙 샘플을 포함하는 블록 C0 또는 상기 블록의 우측 상단 코너에 인접하는 샘플을 포함하는 블록 C1 중 적어도 하나를 포함할 수 있다.

- [434] 모션 벡터 예측 후보 리스트가 포함할 수 있는 모션 벡터 예측 후보들의 최대 개수는 2개일 수 있다. 모션 벡터 예측 후보들의 유도 순서는 다음과 같다.
- [435] 1. 좌측 이웃 블록 A0 또는 좌측 이웃 블록 A1 중 적어도 하나가 가용한 경우, 가용한 블록의 움직임 벡터를 모션 벡터 예측 후보로 설정
- [436] 2. 상단 이웃 블록 B0, 상단 이웃 블록 B1 또는 상단 이웃 블록 B2 중 적어도 하나가 가용한 경우, 가용한 블록의 움직임 벡터를 모션 벡터 예측 후보로 설정
- [437] 3. 시간적 이웃 블록이 가용한 경우, 시간적 움직임 벡터를 모션 벡터 예측 후보로 설정
- [438] 4. 제로 모션 벡터를 모션 벡터 예측 후보로 설정
- [439] 또는, 1-3의 순서에 따라 유도된 모션 벡터 예측 후보의 개수가 2개보다 작은 경우, 인터 영역 모션 정보 테이블에 포함된 움직임 벡터를 모션 벡터 예측 후보로 설정할 수도 있다. 인터 영역 모션 정보 테이블이 이용 가능한 경우, 모션 벡터 예측 후보들은 다음의 순서에 따라 유도될 수 있다.
- [440] 1. 좌측 이웃 블록 A0 또는 좌측 이웃 블록 A1 중 적어도 하나가 가용한 경우, 가용한 블록의 움직임 벡터를 모션 벡터 예측 후보로 설정
- [441] 2. 상단 이웃 블록 B0, 상단 이웃 블록 B1 또는 상단 이웃 블록 B2 중 적어도 하나가 가용한 경우, 가용한 블록의 움직임 벡터를 모션 벡터 예측 후보로 설정
- [442] 3. 시간적 이웃 블록이 가용한 경우, 시간적 움직임 벡터를 모션 벡터 예측 후보로 설정
- [443] 4. 인터 영역 모션 정보 테이블에 포함된 움직임 벡터를 모션 벡터 예측 후보로 설정
- [444] 5. 제로 모션 벡터를 모션 벡터 예측 후보로 설정
- [445] 기준 모션 벡터 예측 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 유도된 움직임 벡터를 갖는 모션 벡터 예측 후보를 모션 벡터 예측 후보 리스트에 추가할 수도 있다. 기준 모션 벡터 예측 후보의 움직임 벡터에서 오프셋 벡터를 가산 또는 감산하여 유도된 움직임 벡터를 갖는 모션 벡터 예측 후보를 리파인 모션 벡터 예측 후보라 호칭할 수 있다.
- [446] 도 37은 리파인 모션 벡터 예측 후보로 설정될 수 있는 움직임 벡터 후보들을 나타낸 도면이다.
- [447] 기준 모션 벡터 예측 후보의 움직임 벡터가 ($MvpLX[0]$, $MvpLX[1]$)인 경우, 리파인 모션 벡터 예측 후보의 움직임 벡터는 기준 모션 벡터 예측 후보의 움직임 벡터의 x성분 또는 y성분 적어도 하나에 오프셋을 가산 또는 감산하여 유도될 수 있다. 일 예로, 리파인 모션 벡터 예측 후보의 움직임 벡터는 ($MvpLX[0]+M$, $MvpLX[1]$), ($MvpLX[0]-M$, $MvpLX[1]$), ($MvpLX[0]$, $MvpLX[1]+M$) 또는 ($MvpLX[0]$, $MvpLX[1]-M$)로 설정될 수 있다. M은 오프셋 벡터의 크기를 나타낸다.
- [448] 오프셋 벡터의 크기 M은 부호화기 및 복호화기에서 기 정의될 수 있다. 일 예로, 오프셋 벡터의 크기 M는 1 또는 4와 같이, 4 이하의 정수로 설정될 수 있다.

- [449] 또는, 오프셋 벡터를 결정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는, 시퀀스, 픽처, 슬라이스 또는 블록 레벨에서 시그널링될 수 있다. 일 예로, 상술한 오프셋 벡터의 크기를 결정하기 위한 정보 `distance_idx` 또는 오프셋 벡터의 방향을 결정하기 위한 정보 `direction_idx` 중 적어도 하나를 이용하여 오프셋 벡터를 결정할 수 있다.
- [450] 기준 모션 벡터 예측 후보는 모션 벡터 예측 후보 리스트 내 기 정의된 인덱스 값을 갖는 모션 벡터 예측 후보일 수 있다. 일 예로, 모션 벡터 예측 후보 리스트에 포함된 모션 벡터 예측 후보들 중 인덱스 값이 가장 0인 모션 벡터 예측 후보 또는 인덱스 값이 1인 모션 벡터 예측 후보가 기준 모션 벡터 예측 후보로 설정될 수 있다.
- [451] 다른 예로, 적어도 하나 이상의 예측 벡터 오프셋 후보를 포함하는 머지 리파인먼트 오프셋 리스트를 이용하여, 오프셋 벡터를 결정할 수 있다. 현재 블록의 인덱스 정보에 의해 특정된 모션 벡터 예측 후보가 기준 모션 벡터 예측 후보인 경우, 예측 벡터 리파인먼트 오프셋 리스트를 이용하여 오프셋 벡터를 결정할 수 있다. 그리고, 모션 벡터 예측 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여, 현재 블록의 움직임 예측 벡터를 유도할 수 있다. 기준 모션 벡터 예측 후보는 모션 벡터 예측 후보 리스트 내 기 정의된 인덱스 값을 갖는 모션 벡터 예측 후보일 수 있다. 일 예로, 모션 벡터 예측 후보 리스트에 포함된 모션 벡터 예측 후보들 중 인덱스 값이 가장 작은 모션 벡터 예측 후보 또는 인덱스 값이 가장 큰 모션 벡터 예측 후보가 기준 모션 벡터 예측 후보로 설정될 수 있다.
- [452] 예측 벡터 오프셋 리파인먼트 리스트를 이용하여 오프셋 벡터를 산출하는 경우, 예측 벡터 후보 리스트가 포함할 수 있는 예측 벡터 후보들의 최대 개수를 2보다 큰 값으로 설정할 수도 있다.
- [453] 도 38은 예측 벡터 리파인먼트 오프셋 리스트의 구성을 예시한 도면이다.
- [454] 도 38에서 기준 예측 벡터 후보는 인덱스가 2인 예측 벡터 후보인 것으로 가정한다.
- [455] 예측 벡터 후보들 중 어느 하나를 나타내는 인덱스 정보 `AMVPcand_idx`에 의해 특정된 예측 벡터 후보의 인덱스가 2가 아닌 경우, 예측 벡터 후보의 움직임 벡터를 현재 블록의 움직임 예측 벡터로 설정할 수 있다.
- [456] 반면, 인덱스 정보 `AMVPcand_idx`에 의해 특정된 예측 벡터 후보의 인덱스가 2인 경우, 예측 벡터 리파인먼트 오프셋 리스트를 이용하여 오프셋 벡터를 유도할 수 있다. 예측 벡터 리파인먼트 오프셋 리스트에 포함된 예측 벡터 오프셋 후보들 중 어느 하나를 특정하는 인덱스 정보 `AMVPOffset_idx`가 비트스트림을 통해 시그널링될 수 있다.
- [457] 오프셋 벡터가 특정되면, 기준 예측 벡터 후보의 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 현재 블록의 움직임 예측 벡터를 유도할 수 있다.
- [458] 코딩 블록이 어파인 모션 모델을 기초로 부호화된 경우에도, 움직임 벡터의

리파인 기법이 이용될 수 있다. 일 예로, 어파인 모션 벡터 예측 모드가 적용되는 경우, 어파인 시드 예측 벡터에 어파인 시드 차분 벡터를 더하여, 코딩 블록의 어파인 시드 벡터들을 유도할 수 있다. 여기서, 어파인 시드 예측 벡터는, 코딩 블록의 공간적 이웃 블록 또는 시간적 이웃 블록의 어파인 시드 벡터를 기초로 유도될 수 있다. 어파인 시드 차분 벡터는 비트스트림으로부터 시그널링되는 정보를 기초로 결정될 수 있다. 이때, 모든 컨트롤 포인트들에 대해 동일한 어파인 시드 차분 벡터를 적용할 수 있다. 또는, 어파인 시드 벡터를 결정하기 위한 정보가, 컨트롤 포인트별로 시그널링될 수 있다.

- [459] 코딩 블록의 어파인 시드 벡터들을 기초로, 서브 블록에 대한 어파인 벡터가 유도되면, 어파인 벡터는 초기 움직임 벡터로 설정한 뒤, 오프셋 벡터를 유도할 수 있다. 각 서브 블록의 움직임 벡터는 초기 움직임 벡터에 오프셋 벡터를 가산 또는 감산하여 유도될 수 있다.
- [460] 오프셋 벡터를 결정하기 위한 정보를 시그널링하는 대신, 복호화기에서 오프셋 벡터를 유도할 수도 있다. 구체적으로, 서브 블록에 포함된 예측 샘플들의 수평 방향 그라디언트(gradient)에 대한 평균값 및 수직 방향 그라디언트(gradient)에 대한 평균값을 이용하여, 오프셋 벡터를 유도할 수 있다.
- [461]
- [462] 인트라 예측은 현재 블록 주변에 부호화/복호화가 완료된 복원 샘플을 이용하여, 현재 블록을 예측하는 것이다. 이때, 현재 블록의 인트라 예측에는, 인루프 필터가 적용되기 전의 복원 샘플이 이용될 수 있다.
- [463] 인트라 예측 기법은 매트릭스(Matrix)에 기반한 인트라 예측 및 주변 복원 샘플과의 방향성을 고려한 일반 인트라 예측을 포함한다. 현재 블록의 인트라 예측 기법을 지시하는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그일 수 있다. 또는, 현재 블록의 위치, 크기, 형태 또는 이웃 블록의 인트라 예측 기법 중 적어도 하나에 기초하여, 현재 블록의 인트라 예측 기법을 결정할 수 있다. 일 예로, 현재 블록이 픽처 바운더리를 걸쳐 존재하는 경우, 현재 블록에는 매트릭스에 기반한 인트라 예측이 적용되지 않도록 설정될 수 있다.
- [464] 매트릭스에 기반한 인트라 예측은, 부호화기 및 복호화기에서 기 저장된 매트릭스와, 현재 블록 주변의 복원 샘플 사이의 행렬 곱에 기반하여, 현재 블록의 예측 블록을 획득하는 방법이다. 기 저장된 복수개의 매트릭스들 중 어느 하나를 특정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 복호화기는 상기 정보 및 현재 블록의 크기에 기초하여, 현재 블록의 인트라 예측을 위한 매트릭스를 결정할 수 있다.
- [465] 일반 인트라 예측은, 비방향성 인트라 예측 모드 또는 방향성 인트라 예측 모드에 기초하여, 현재 블록에 대한 예측 블록을 획득하는 방법이다.
- [466]
- [467] 원본 영상에서 예측 영상을 차분하여 유도된 잔차 영상을 유도할 수 있다. 이때,

잔차 영상을 주파수 도메인으로 변경하였을 때, 주파수 성분들 중 고주파 성분들을 제거하더라도, 영상의 주관적 화질은 크게 떨어지지 않는다. 이에 따라, 고주파 성분들의 값을 작게 변환하거나, 고주파 성분들의 값을 0으로 설정한다면, 시각적 왜곡이 크게 발생하지 않으면서도 압축 효율을 증가시킬 수 있는 효과가 있다. 위 특성을 반영하여, 잔차 영상을 2차원 주파수 성분들로 분해하기 위해 현재 블록을 변환할 수 있다. 상기 변환은 DCT(Discrete Cosine Transform) 또는 DST(Discrete Sine Transform) 등의 변환 기법을 이용하여 수행될 수 있다.

- [468] DCT 또는 DST를 이용하여 현재 블록을 변환한 뒤, 변환된 현재 블록을 다시 변환할 수 있다. 이때, DCT 또는 DST에 기초한 변환을 제1 변환이라 정의하고, 제1 변환이 적용된 블록을 다시 변환하는 것을 제2 변환이라 정의할 수 있다.
- [469] 제1 변환은 복수개의 변환 코어 후보들 중 어느 하나를 이용하여 수행될 수 있다. 일 예로, DCT2, DCT8 또는 DCT7 중 어느 하나를 이용하여 제1 변환이 수행될 수 있다.
- [470] 수평 방향 및 수직 방향에 대해 상이한 변환 코어가 사용될 수도 있다. 수평 방향의 변환 코어 및 수직 방향의 변환 코어의 조합을 나타내는 정보가 비트스트림을 통해 시그널링될 수도 있다.
- [471] 제1 변환 및 제2 변환의 수행 단위가 상이할 수 있다. 일 예로, 8x8 블록에 대해 제1 변환을 수행하고, 변환된 8x8 블록 중 4x4 크기의 서브 블록에 대해 제2 변환을 수행할 수 있다. 이때, 제2 변환이 수행되지 않는 잔여 영역들의 변환 계수를 0으로 설정할 수도 있다.
- [472] 또는, 4x4 블록에 대해 제1 변환을 수행하고, 변환된 4x4 블록을 포함하는 8x8 크기의 영역에 대해 제2 변환을 수행할 수도 있다.
- [473] 제2 변환의 수행 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다.
- [474] 복호화기에서는 제2 변환의 역변환(제2 역변환)을 수행하고, 그 수행 결과에 제1 변환의 역변환(제1 역변환)을 수행할 수 있다. 상기 제2 역변환 및 제1 역변환의 수행 결과, 현재 블록에 대한 잔차 신호들이 획득될 수 있다.
- [475] 양자화는 블록의 에너지를 줄이기 위한 것으로, 양자화 과정은 변환 계수를 특정 상수값으로 나누는 과정을 포함한다. 상기 상수값은 양자화 파라미터에 의해 유도될 수 있고, 양자화 파라미터는 1부터 63사이의 값으로 정의될 수 있다.
- [476] 부호화기에서 변환 및 양자화를 수행하면, 복호화기는 역양자화 및 역변환을 통해 잔차 블록을 획득할 수 있다. 복호화기에서는 예측 블록과 잔차 블록을 더하여, 현재 블록에 대한 복원 블록을 획득할 수 있다.
- [477]
- [478] 현재 블록의 복원 블록이 획득되면, 인루프 필터링(In-loop filtering)을 통해 양자화 및 부호화 과정에서 발생하는 정보의 손실을 줄일 수 있다. 인루프 필터는 디블록킹 필터(Deblocking filter), 샘플 적응적 오프셋 필터(Sample

Adaptive Offset filter, SAO) 또는 적응적 루프 필터(Adaptive Loop Filter, ALF) 중 적어도 하나를 포함할 수 있다.

[479]

[480] 복호화 과정 또는 부호화 과정을 중심으로 설명된 실시예들을, 부호화 과정 또는 복호화 과정에 적용하는 것은, 본 발명의 범주에 포함되는 것이다. 소정의 순서로 설명된 실시예들을, 설명된 것과 상이한 순서로 변경하는 것 역시, 본 발명의 범주에 포함되는 것이다.

[481]

상술한 실시예는 일련의 단계 또는 순서도를 기초로 설명되고 있으나, 이는 발명의 시계열적 순서를 한정하는 것은 아니며, 필요에 따라 동시에 수행되거나 다른 순서로 수행될 수 있다. 또한, 상술한 실시예에서 블록도를 구성하는 구성요소(예를 들어, 유닛, 모듈 등) 각각은 하드웨어 장치 또는 소프트웨어로 구현될 수도 있고, 복수의 구성요소가 결합하여 하나의 하드웨어 장치 또는 소프트웨어로 구현될 수도 있다. 상술한 실시예는 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

산업상 이용가능성

[482] 본 발명은 영상을 부호화/복호화하는 전자 장치에 적용될 수 있다.

청구범위

- [청구항 1] 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는지 여부를 결정하는 단계;
 상기 현재 블록에 대한 머지 후보 리스트를 생성하는 단계;
 상기 머지 후보 리스트를 기초로, 상기 현재 블록에 대한 머지 후보를 특정하는 단계; 및
 상기 머지 후보를 기초로, 상기 현재 블록에 대한 움직임 벡터를 유도하는 단계를 포함하되,
 상기 현재 블록에 상기 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 상기 현재 블록의 움직임 벡터는, 상기 머지 후보의 움직임 벡터에 오프셋 벡터를 가산하여 유도되고 상기 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수가 복수개인 경우 상기 현재 블록의 상기 머지 후보는 상기 머지 후보들 중 어느 하나를 가리키는 비트스트림으로부터 복호화되는 인덱스 정보를 기초로 선택되고, 상기 최대 개수가 1개인 경우 상기 인덱스 정보를 복호화함이 없이 상기 머지 후보가 결정되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 2] 제1 항에 있어서,
 상기 오프셋 벡터의 크기는, 벡터 크기 후보들 중 어느 하나를 특정하는 제1 인덱스 정보를 기초로 결정되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 3] 제2 항에 있어서,
 상기 벡터 크기 후보들의 최대값 또는 최소값 중 적어도 하나는 상기 벡터 크기 후보들의 범위를 나타내는 플래그의 값에 따라 상이하게 설정되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 4] 제3 항에 있어서,
 상기 플래그는 픽처 레벨에서 시그널링되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 5] 제2 항에 있어서,
 상기 벡터 크기 후보들의 최대값 또는 최소값 중 적어도 하나는 상기 현재 블록에 대한 움직임 벡터 정밀도에 따라 상이하게 설정되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 6] 제2 항에 있어서,
 상기 오프셋 벡터의 크기는, 상기 제1 인덱스 정보에 의해 특정된 벡터 크기 후보가 가리키는 값에 비트 시프팅 연산을 적용함으로써 획득되는 것을 특징으로 하는, 영상 복호화 방법.
- [청구항 7] 제2 항에 있어서,
 상기 오프셋 벡터의 방향은, 벡터 방향 후보들 중 어느 하나를 특정하는

제2 인덱스 정보를 기초로 결정되는 것을 특징으로 하는, 영상 복호화 방법.

- [청구항 8] 현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는지 여부를 결정하는 단계;
 상기 현재 블록에 대한 머지 후보 리스트를 생성하는 단계;
 상기 머지 후보 리스트를 기초로, 상기 현재 블록에 대한 머지 후보를 특정하는 단계; 및
 상기 머지 후보를 기초로, 상기 현재 블록에 대한 움직임 벡터를 유도하는 단계를 포함하되,
 상기 현재 블록에 상기 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 상기 현재 블록의 움직임 벡터는 상기 머지 후보의 움직임 벡터에 오프셋 벡터를 가산하여 유도되고, 상기 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수가 복수개인 경우 상기 머지 후보들 중 상기 현재 블록의 상기 머지 후보를 가리키는 인덱스 정보를 부호화하고, 상기 최대 개수가 1개인 경우 상기 인덱스 정보의 부호화를 생략하는 것을 특징으로 하는, 영상 부호화 방법.
- [청구항 9] 제8 항에 있어서,
 복수의 벡터 크기 후보들 중 상기 오프셋 벡터의 크기를 나타내는 벡터 크기 후보를 특정하기 위한 제1 인덱스 정보를 부호화하는 단계를 더 포함하는, 영상 부호화 방법.
- [청구항 10] 제9 항에 있어서,
 상기 벡터 크기 후보들의 범위를 나타내는 플래그를 부호화하는 단계를 더 포함하고, 상기 플래그의 값에 따라 상기 벡터 크기 후보들의 최대값 또는 최소값 중 적어도 하나가 상이하게 설정되는 것을 특징으로 하는, 영상 부호화 방법.
- [청구항 11] 제10 항에 있어서,
 상기 플래그는 픽처 레벨에서 부호화되는 것을 특징으로 하는, 영상 부호화 방법.
- [청구항 12] 제9 항에 있어서,
 상기 벡터 크기 후보들의 최대값 또는 최소값 중 적어도 하나는 상기 현재 블록에 대한 움직임 벡터 정밀도에 따라 상이하게 설정되는 것을 특징으로 하는, 영상 부호화 방법.
- [청구항 13] 제9 항에 있어서,
 상기 벡터 크기 후보는 상기 오프셋 벡터의 크기에 비트 시프팅 연산을 적용함으로써 유도된 값을 갖는 것을 특징으로 하는, 영상 부호화 방법.
- [청구항 14] 제9 항에 있어서,
 복수의 벡터 방향 후보들 중 상기 오프셋 벡터의 방향을 나타내는 벡터 방향 후보를 특정하기 위한 제2 인덱스 정보를 부호화하는 단계를 더

포함하는, 영상 부호화 방법.

[청구항 15]

현재 블록에 머지 오프셋 벡터 부호화 방법이 적용되는지 여부를 결정하는 단계;

상기 현재 블록에 대한 머지 후보 리스트를 생성하고, 상기 머지 후보 리스트를 기초로, 상기 현재 블록에 대한 머지 후보를 특정하고, 상기 머지 후보를 기초로, 상기 현재 블록에 대한 움직임 벡터를 유도하는 인터 예측부를 포함하되,

상기 현재 블록에 상기 머지 오프셋 벡터 부호화 방법이 적용되는 경우, 상기 현재 블록의 움직임 벡터는, 상기 머지 후보의 움직임 벡터에 오프셋 벡터를 가산하여 유도되고 상기 머지 후보 리스트가 포함할 수 있는 머지 후보들의 최대 개수가 복수개인 경우 상기 현재 블록의 상기 머지 후보는 상기 머지 후보들 중 어느 하나를 가리키는 비트스트림으로부터 복호화되는 인덱스 정보를 기초로 선택되고, 상기 최대 개수가 1개인 경우 상기 인덱스 정보를 복호화함이 없이 상기 머지 후보가 결정되는 것을 특징으로 하는, 영상 복호화 장치.