



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년08월30일
(11) 등록번호 10-2701594
(24) 등록일자 2024년08월28일

- (51) 국제특허분류(Int. Cl.)
H04N 19/52 (2014.01) H04N 19/105 (2014.01)
H04N 19/139 (2014.01) H04N 19/176 (2014.01)
H04N 19/70 (2014.01)
- (52) CPC특허분류
H04N 19/52 (2015.01)
H04N 19/105 (2015.01)
- (21) 출원번호 10-2021-7037074
- (22) 출원일자(국제) 2020년05월21일
심사청구일자 2023년05월02일
- (85) 번역문제출일자 2021년11월12일
- (65) 공개번호 10-2022-0009952
- (43) 공개일자 2022년01월25일
- (86) 국제출원번호 PCT/CN2020/091539
- (87) 국제공개번호 WO 2020/233661
국제공개일자 2020년11월26일
- (30) 우선권주장
PCT/CN2019/087805 2019년05월21일 중국(CN)
- (56) 선행기술조사문헌
B. Bross, et. al, "Versatile Video Coding (Draft 5)", 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, 19 - 27 Mar. 2019, JVET-N1001-v5*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
베이징 바이트댄스 네트워크 테크놀로지 컴퍼니, 리미티드
중국, 베이징 100041, 스징산 디스트릭트, 스징 로드, 넘버 30, 넘버 3빌딩, 2층, 룸 비-0035
바이트댄스 아이엔씨
미국, 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
- (72) 발명자
장 카이
미국 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
장 리
미국 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
(뒷면에 계속)
- (74) 대리인
성병기

전체 청구항 수 : 총 18 항

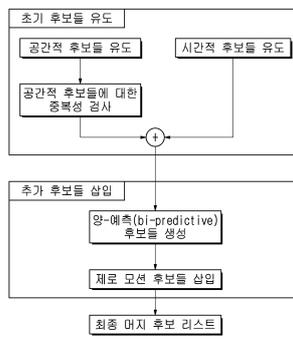
심사관 : 박상철

(54) 발명의 명칭 서브블록 머지 모드에서의 선택스 시그널링

(57) 요약

시각 미디어 처리 방법은 시각 미디어 데이터에 어파인 적응적 모션 벡터 해상도(AMVR) 기술을 사용할 것인지 여부를 상기 시각 미디어 데이터의 비트스트림 표현 내의 상기 AMVR 기술 관련한 제어 정보에 의존하여 결정하는 단계를 포함하고, 상기 시각적 미디어 데이터와 상기 시각적 미디어 데이터의 상기 비트스트림 표현 간의 변환을 수행하는 단계를 포함하며, 이때 상기 제어 정보는 규칙에 기반하여 상기 비트스트림 표현에 포함되거나 배제된다.

대표도 - 도1



(52) CPC특허분류

HO4N 19/139 (2015.01)

HO4N 19/176 (2015.01)

HO4N 19/70 (2015.01)

(72) 발명자

리우 흥빈

진르터우타오 포스트 오피스 차이나 세틀라이트 커뮤니케이션 타워 63 지춘 로드 하이디안 디스트릭트 베이징 100080 중국

왕 유에

진르터우타오 포스트 오피스 차이나 세틀라이트 커뮤니케이션 타워 63 지춘 로드 하이디안 디스트릭트 베이징 100080 중국

명세서

청구범위

청구항 1

비디오 데이터 처리 방법으로서,

비디오의 시각 미디어 데이터(visual media data)와 상기 비디오 비트스트림 간의 변환을 위해, 제1 레벨에서 상기 시각 미디어 데이터를 위해 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 인에이블(enable)되는지 여부를 지시하는 제1 정보를 결정하는 단계;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제2 정보가 포함되는 지를 결정하는 단계, 상기 제2 정보는 서브 블록 기반 시간적 모션 벡터 예측(subblock-based temporal motion vector prediction)이 상기 제1 레벨에서 인에이블되는지 여부를 지시하고;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제3 정보가 포함되는지를 결정하는 단계, 상기 제3 정보는 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 제2 레벨에서 상기 시각적 미디어 데이터를 위해 인에이블되는지 여부를 지시하고;

상기 제3 정보에 기반하여, 상기 비트스트림내에 제4 정보가 포함되는지를 결정하는 단계, 상기 제4 정보는 상기 제2 레벨에서 상기 시각적 미디어 데이터와 연관된 동일 위치 참조 픽처(collocated reference picture)에 관련된 정보를 지시하고; 및

적어도 상기 제4 정보에 기반하여 상기 변환을 수행하는 단계를 포함하되,

상기 변환 동안, 후보 리스트(candidate list)가 구성되어 지고,

상기 후보 리스트의 최대 개수가 상기 비트스트림에 포함되는 제5 정보에 기반하는지 여부는 상기 제1 레벨에서의 코딩 모드를 지시하는 제6 정보에 기반하되, 상기 제5 정보는 5 에서 정수를 빼고 코딩되어지는, 비디오 데이터 처리 방법.

청구항 2

제1항에 있어서,

상기 제4 정보는 어느 참조 픽처 리스트로부터 상기 동일 위치 참조 픽처가 유도되는 지를 지시하는, 비디오 데이터 처리 방법.

청구항 3

제1항에 있어서,

상기 제4 정보는 상기 동일 위치 참조 픽처의 참조 인덱스(reference index)를 지시하는, 비디오 데이터 처리 방법.

청구항 4

제1항에 있어서,

상기 시간적 모션 벡터 예측 코딩 모드가 인에이블한 경우에 대응하여 상기 제4 정보가 상기 비트스트림내에 포함되는, 비디오 데이터 처리 방법.

청구항 5

제1항에 있어서,

상기 시간적 모션 벡터 예측 코딩 모드가 디스에이블(disable)한 경우에 기반하여 상기 제4 정보는 상기 비트스트림내에 포함되지 않는, 비디오 데이터 처리 방법.

청구항 6

제1항에 있어서,

상기 후보 리스트의 최대 후보 개수는 상기 시간적 모션 벡터 예측 코딩 모드가 인에이블되는지 여부에 종속되는, 비디오 데이터 처리 방법.

청구항 7

제1항에 있어서,

상기 후보 리스트의 최대 후보 개수는 서브 블록 기반 시간적 모션 벡터 예측 코딩 모드가 인에이블되는지 여부에 종속되는, 비디오 데이터 처리 방법.

청구항 8

제1항에 있어서,

상기 변환은 상기 시각 미디어 데이터를 상기 비트스트림으로 인코딩하는 것을 포함하는, 비디오 데이터 처리 방법.

청구항 9

제1항에 있어서,

상기 변환은 상기 시각 미디어 데이터를 상기 비트스트림으로부터 디코딩하는 것을 포함하는, 비디오 데이터 처리 방법.

청구항 10

프로세서와 상기 프로세서에 의해 수행되는 명령을 포함하는 비일시적 메모리를 포함하는 비디오 데이터 처리 장치로서, 상기 프로세서에 의해 수행되는 명령은 상기 프로세서가,

비디오의 시각 미디어 데이터(visual media data)와 상기 비디오 비트스트림 간의 변환을 위해, 제1 레벨에서 상기 시각 미디어 데이터를 위해 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 인에이블(enable)되는지 여부를 지시하는 제1 정보를 결정하게 하고;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제2 정보가 포함되는 지를 결정하게 하되, 상기 제2 정보는 서브 블록 기반 시간적 모션 벡터 예측(subblock-based temporal motion vector prediction)이 상기 제1 레벨에서 인에이블되는지 여부를 지시하고;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제3 정보가 포함되는지를 결정하게 하되, 상기 제3 정보는 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 제2 레벨에서 상기 시각적 미디어 데이터를 위해 인에이블되는지 여부를 지시하고;

상기 제3 정보에 기반하여, 상기 비트스트림내에 제4 정보가 포함되는지를 결정하게 하되, 상기 제4 정보는 상

기 제2 레벨에서 상기 시각적 미디어 데이터와 연관된 동일 위치 참조 픽처(collocated reference picture)에 관련된 정보를 지시하고; 및

적어도 상기 제4 정보에 기반하여 상기 변환을 수행하게 하되,

상기 변환 동안, 후보 리스트(candidate list)가 구성되어 지고,

상기 후보 리스트의 최대 개수가 상기 비트스트림에 포함되는 제5 정보에 기반하는지 여부는 상기 제1 레벨에서의 코딩 모드를 지시하는 제6 정보에 기반하되, 상기 제5 정보는 5 에서 정수를 빼고 코딩되어지는, 비디오 데이터 처리 장치.

청구항 11

제10항에 있어서,

상기 제4 정보는 어느 참조 픽처 리스트로부터 상기 동일 위치 참조 픽처가 유도되는 지를 지시하는, 비디오 데이터 처리 장치.

청구항 12

제10항에 있어서,

상기 제4 정보는 상기 동일 위치 참조 픽처의 참조 인덱스(reference index)를 지시하는, 비디오 데이터 처리 장치.

청구항 13

제10항에 있어서,

상기 시간적 모션 벡터 예측 코딩 모드가 인에이블한 경우에 대응하여 상기 제4 정보가 상기 비트스트림내에 포함되는, 비디오 데이터 처리 장치.

청구항 14

제10항에 있어서,

상기 시간적 모션 벡터 예측 코딩 모드가 디스에이블(disable)한 경우에 기반하여 상기 제4 정보는 상기 비트스트림내에 포함되지 않는, 비디오 데이터 처리 장치.

청구항 15

제10항에 있어서,

상기 후보 리스트의 최대 후보 개수는 상기 시간적 모션 벡터 예측 코딩 모드가 인에이블되는지 여부에 종속되는, 비디오 데이터 처리 장치.

청구항 16

제10항에 있어서,

상기 후보 리스트의 최대 후보 개수는 서브 블록 기반 시간적 모션 벡터 예측 코딩 모드가 인에이블되는지 여부에 종속되는, 비디오 데이터 처리 장치.

청구항 17

프로세서에 의해 수행되는 명령을 포함하는 비밀시적 컴퓨터 판독 가능한 저장 매체로서, 상기 명령은 상기 프로세서가,

비디오의 시각 미디어 데이터(visual media data)와 상기 비디오 비트스트림 간의 변환을 위해, 제1 레벨에서 상기 시각 미디어 데이터를 위해 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 인에이블(enable)되는지 여부를 지시하는 제1 정보를 결정하게 하고;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제2 정보가 포함되는 지를 결정하게 하되, 상기 제2 정보는 서브 블록 기반 시간적 모션 벡터 예측(subblock-based temporal motion vector prediction)이 상기 제1 레벨에서 인에이블되는지 여부를 지시하고;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제3 정보가 포함되는지를 결정하게 하되, 상기 제3 정보는 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 제2 레벨에서 상기 시각적 미디어 데이터를 위해 인에이블되는지 여부를 지시하고;

상기 제3 정보에 기반하여, 상기 비트스트림내에 제4 정보가 포함되는지를 결정하게 하되, 상기 제4 정보는 상기 제2 레벨에서 상기 시각적 미디어 데이터와 연관된 동일 위치 참조 픽처(collocated reference picture)에 관련된 정보를 지시하고; 및

적어도 상기 제4 정보에 기반하여 상기 변환을 수행하게 하되,

상기 변환 동안, 후보 리스트(candidate list)가 구성되어 지고,

상기 후보 리스트의 최대 개수가 상기 비트스트림에 포함되는 제5 정보에 기반하는지 여부는 상기 제1 레벨에서의 코딩 모드를 지시하는 제6 정보에 기반하되, 상기 제5 정보는 5 에서 정수를 빼고 코딩되어지는, 비밀시적 컴퓨터 판독 가능한 저장 매체.

청구항 18

비디오 처리 장치에 의해 수행되는 방법에 의해 생성된 비트스트림을 저장하는 비밀시적 컴퓨터 판독 가능한 기록 매체로서, 상기 방법은,

비디오의 시각 미디어 데이터(visual media data)와 상기 비디오 비트스트림 간의 변환을 위해, 제1 레벨에서 상기 시각 미디어 데이터를 위해 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 인에이블(enable)되는지 여부를 지시하는 제1 정보를 결정하는 단계;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제2 정보가 포함되는 지를 결정하는 단계, 상기 제2 정보는 서브 블록 기반 시간적 모션 벡터 예측(subblock-based temporal motion vector prediction)이 상기 제1 레벨에서 인에이블되는지 여부를 지시하고;

상기 제1 정보에 기반하여, 상기 비트스트림내에 제3 정보가 포함되는지를 결정하는 단계, 상기 제3 정보는 시간적 모션 벡터 예측 코딩 모드(temporal motion vector prediction coding mode)가 제2 레벨에서 상기 시각적 미디어 데이터를 위해 인에이블되는지 여부를 지시하고;

상기 제3 정보에 기반하여, 상기 비트스트림내에 제4 정보가 포함되는지를 결정하는 단계, 상기 제4 정보는 상기 제2 레벨에서 상기 시각적 미디어 데이터와 연관된 동일 위치 참조 픽처(collocated reference picture)에 관련된 정보를 지시하고; 및

적어도 상기 제4 정보에 기반하여 상기 변환을 수행하는 단계를 포함하되,

상기 변환 동안, 후보 리스트(candidate list)가 구성되어 지고,

상기 후보 리스트의 최대 개수가 상기 비트스트림에 포함되는 제5 정보에 기반하는지 여부는 상기 제1 레벨에서의 코딩 모드를 지시하는 제6 정보에 기반하되, 상기 제5 정보는 5 에서 정수를 빼고 코딩되어지는, 비밀시적 컴퓨터 판독 가능한 기록 매체.

청구항 19

삭제

발명의 설명

기술 분야

[0001] 관련 출원들에 대한 상호 참조 : 본 출원은, 2019년 5월 21일에 제출된 국제출원 No. PCT/CN2019/087805의 우선권과 혜택을 주장하는, 2020년 5월 21일에 제출된 국제출원 No. PCT/CN2020/091539의 국내 단계 출원이다. 상기 출원들의 전체 개시물은 본 출원의 개시의 일부로서 참고로 통합된다. 본 특허 문서는 비디오 코딩/디코딩 기술들, 디바이스들 및 시스템들에 관한 것이다.

배경 기술

[0002] 비디오 압축 기술의 발전에도 불구하고 디지털 비디오는 여전히 인터넷 및 기타 디지털 통신 네트워크에서 가장 큰 대역폭을 사용한다. 비디오를 수신하고 표시할 수 있는 접속된 사용자 장치의 수가 증가함에 따라, 디지털 비디오 사용을 위한 대역폭 수요가 계속 증가할 것으로 예상된다.

발명의 내용

해결하려는 과제

[0003] 본 문서는 서브-블록 기반 모션 벡터 개선을 사용하여 비디오 코딩 또는 디코딩이 실시되는 다양한 실시예들 및 기술들을 설명한다.

과제의 해결 수단

[0004] 일 예시적 양상에서, 시각 미디어 처리 방법이 개시된다. 상기 방법은 시각 미디어 데이터의 비트스트림 표현에서의 어파인 적응적 모션 벡터 해상도(AMVR: affine adaptive motion vector resolution) 기술과 관련된 제어 정보에 기반하여 상기 시각 미디어 데이터에 상기 어파인 AMVR 기술을 사용할지 여부를 결정하는 단계 및 상기 시각 미디어 데이터와 상기 시각 미디어 데이터의 상기 비트스트림 표현 간의 변환을 수행하는 단계를 포함하며, 여기서 상기 제어 정보는 규칙에 기반하여 상기 비트스트림 표현에 포함되거나 포함되지 않는다.

[0005] 다른 예시적 양상에서, 시각 미디어 처리의 다른 방법이 개시된다. 상기 방법은 서브블록 기반 시간적 모션 벡터 예측(sbTMVP) 기술이 시각 미디어 데이터에 적용되는지 여부를 결정하는 단계, 상기 결정에 대응하여, 상기 시각 미디어 데이터에 포함된 현재 비디오 블록에 대한 서브 블록 머지 후보 리스트를 생성하는 단계, 그리고 상기 서브 블록 머지 후보 리스트를 사용하여 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0006] 또 다른 예시적 양상에서, 시각 미디어 처리의 다른 방법이 개시된다. 상기 방법은 현재 비디오 블록과 비트스트림 표현 간의 변환 동안 하나 또는 그 이상의 디폴트 머지 후보를 상기 변환을 위한 서브블록 머지 후보 리스트에 추가하는 단계 및 상기 추가된 하나 또는 그 이상의 디폴트 머지 후보를 가지는 상기 서브블록 머지 후보 리스트를 사용하여 상기 변환을 수행하는 단계를 포함한다.

[0007] 또 다른 예시적 양상에서, 시각 미디어 처리의 다른 방법이 개시된다. 상기 방법은 비디오의 현재 비디오 블록과 비트스트림 표현 간의 변환 동안 상기 변환에 대한 서브블록 기반 시간적 모션 벡터 예측(sbTMVP)의 적용가능성을 판단하는 단계-여기서, 상기 비트스트림의 하나 또는 그 이상의 비트들이 상기 판단에 해당한다-와 상기 판단에 기반하여 상기 변환을 수행하는 단계를 포함한다.

[0008] 또 다른 예시적 양상에서, 시각 미디어 처리의 다른 방법이 개시된다. 상기 방법은 시간적 모션 벡터 예측(TMVP: temporal motion vector prediction) 단계 또는 서브블록 기반 시간적 모션 벡터 예측(sbTMVP) 단계와 연관된 조건에 기반하여 선택적으로 서브블록 머지 후보 리스트를 작성하는(building) 단계 및 상기 서브블록 머지 후보 리스트에 기반하여 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 사이에 변환을 수행하는 단계를 포함한다.

[0009] 또 다른 예시적 양상에서, 시각 미디어 처리의 다른 방법이 개시된다. 상기 방법은, 시각 미디어 데이터와 상

기 시각 미디어 데이터의 비트스트림 표현 간의 변환 동안, 상기 시각 미디어 데이터와 연관된 동일 위치 참조 픽처(collocated reference picture)와 관련한 정보를 판단하는 단계-여기서, 상기 정보는 시간적 모션 정보에 접근하는 코딩 모드가 가능한지(enabled) 여부에 기반하여 상기 비트스트림에 포함되거나 포함되지 않는다(excluded)-그리고 상기 동일 위치 참조 픽처에 기반하여 상기 시각 미디어 데이터와 상기 시각 미디어 데이터의 상기 비트스트림 표현 간의 변환을 수행하는 단계를 포함한다.

[0010] 또 다른 예시적 양상에서, 시각 미디어 처리의 다른 방법이 개시된다. 상기 방법은 시각 미디어 데이터의 비디오 블록과 상기 시각 미디어 데이터의 비트스트림 표현 간의 변환을 수행하는 단계 그리고, 상기 변환 동안, 규칙에 기반하여 상기 비트스트림 표현에 포함되거나 포함되지 않는 제어 정보에 의존하는 광학 흐름을 사용하는 예측 개선(PROF: predictive refinement using optical flow)의 적용여부를 결정하는 단계를 포함한다.

[0011] 또 다른 예시적 양상에서, 상기 기술된 방법을 구현하도록 구성된 프로세서를 포함하는 비디오 인코딩 및/또는 디코딩 장치가 개시된다.

[0012] 또 다른 예시적 양상에서, 컴퓨터 판독 가능한 매체가 개시된다. 상기 컴퓨터 판독 가능한 매체는 상기 기술된 방법들 중 하나를 구현하는, 프로세서에 의해 실행가능한 코드를 저장한다.

[0013] 이러한 그리고 다른 양상들은 본 문서에서 더 설명된다.

발명의 효과

[0014] 본 문서는 서브-블록 기반 모션 벡터 개선을 사용하여 비디오 코딩 또는 디코딩이 실시되는 다양한 실시예들 및 기술들을 제안함에 의해, 비디오 코딩 및 기코딩 효율이 증가되는 효과가 있다.

도면의 간단한 설명

- [0015] 도 1은 머지 후보 리스트 구성을 위한 예시적 유도 과정을 도시한다.
- 도2는 공간적 머지 후보들의 위치들의 예들을 보여준다.
- 도 3은 공간적 머지 후보들의 중복성 검사를 위해 고려되는 후보 쌍들의 예들을 보여준다.
- 도4A 내지 도4B는 NX2N 및2NXN으로 각각 분할되는 제2 PU에 대한 예시적 위치들을 보여준다.
- 도 5는 시간적 머지 후보에 대한 모션 벡터 스케일링의 예이다.
- 도 6은 시간적 머지 후보 C0 및 C1에 대한 후보 위치들의 예들을 보여준다.
- 도 7은 결합 양-예측(bi-predictive) 머지 후보의 예를 보여준다.
- 도 8은 모션 벡터 예측 후보에 대한 유도 과정을 요약한다.
- 도 9는 공간적 모션 벡터 후보에 대한 모션 벡터 스케일링의 예를 보여준다.
- 도 10은 CU에 대한 ATMVP 모션 예측의 예를 보여준다.
- 도 11은 4개의 서브블록들(A-D) 및 그것들의 주변 블록들(a-d)을 가지는 하나의 CU의 예를 보여준다.
- 도 12는 상이한 MV 정밀도로 인코딩하는 예시적 순서도이다.
- 도 13은 (a) 135도 파티션 타입(좌측 상단 코너에서 우측 하단 코너로 분할) 및 (b) 45도 분할 패턴들을 보여준다.
- 도 14는 주변 블록들의 예시적 위치들을 보여준다.
- 도 15는 TPM 플래그 코딩에서 문맥 선택을 위해 사용되는 주변 블록들(A 및 L)을 보여준다.
- 도 16은 4 파라미터 어파인 및 (b) 6 파라미터 어파인 모델들을 보여준다.
- 도 17은 서브블록 당 어파인 MVF의 예를 보여준다.
- 도 18은 (a) 4 파라미터 어파인 모델 및 (b) 6 파라미터 어파인 모델을 보여준다.
- 도 19는 승계된 어파인 후보들에 대한 AF_INTER를 위한 MVP를 보여준다.

- 도 20은 구성된 어파인 후보들에 대한 AF_INTER를 위한 MVP를 보여준다.
- 도 21은 (a) 5개의 주변 블록들 및 (b) CPMV 예측자 유도의 예를 보여준다.
- 도 22는 어파인 머지 모드를 위한 예시적 후보 위치들을 보여준다.
- 도 23은 ATMVP가 사용하는 공간적 주변 블록들의 예를 보여준다.
- 도 24는 공간적 주변으로부터 모션 시프트를 적용하고 해당 동일 위치 서브 CU들로부터 모션 정보를 스케일링하여 서브 CU 모션 필드를 유도하는 예를 보여준다.
- 도 25는 어파인 머지 모드에 대한 후보들의 위치를 보여준다.
- 도 26은 변형된 머지 리스트 구성 과정을 보여준다.
- 도 27은 서브블록 MV VSB 및 픽셀 $\Delta v(i, j)$ (붉은색 화살표)를 보여준다.
- 도 28은 본 문서에 설명된 방법들을 구현하기 위해 사용되는 하드웨어 플랫폼의 예를 보여주는 블록도이다.
- 도 29는 비디오 처리의 예시적 방법을 위한 순서도이다.
- 도 30은 개시된 기술들이 구현될 수 있는 예시적 비디오 처리 시스템의 블록도이다.
- 도 31은 시각 미디어 처리의 예시적 방법을 위한 순서도이다.
- 도 32는 시각 미디어 처리의 예시적 방법을 위한 순서도이다.
- 도 33는 시각 미디어 처리의 예시적 방법을 위한 순서도이다.
- 도 34는 시각 미디어 처리의 예시적 방법을 위한 순서도이다.
- 도 35는 시각 미디어 처리의 예시적 방법을 위한 순서도이다.
- 도 36는 시각 미디어 처리의 예시적 방법을 위한 순서도이다.
- 도 37는 시각 미디어 처리의 예시적 방법을 위한 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 문서에서 사용된 섹션 제목들은 이해의 편의를 위한 것으로 실시예들 및 기술들을 해당 섹션들로 한정하지 않는다. 게다가, 특정 실시예들이 다기능 비디오 코딩(Versatile Video Coding) 및 기타 특정한 비디오 코덱들을 참조하여 설명되지만, 개시된 기술들은 다른 비디오 코딩 기술들에도 적용될 수 있다. 또한, 일부 실시예들이 비디오 코딩 단계들을 상세하게 설명하지만, 코딩을 되돌리는 디코딩 단계들이 디코더에 의해 구현됨이 이해될 것이다. 용어 "비디오 처리"는 비디오 코딩 또는 압축, 비디오 디코딩 또는 압축 해제, 그리고 비디오 픽셀들이 하나의 압축된 포맷에서 다른 압축된 포맷으로 또는 상이한 압축된 비트 레이트로 표현되는 비디오 트랜스 코딩을 포괄한다.

[0017] **1. 개략적 소개**

[0018] 본 문서는 비디오 코딩 기술들에 관한 것이다. 구체적으로, 본 문서는 비디오 코딩에서 모션 벡터 코딩에 관한 것이다. 본 문서는 HEVC와 같은 기존 비디오 코딩 표준 또는 최종 확정될 표준(VVC, Versatile Video Coding)에 적용될 수 있다. 또한 본 문서는 미래의 비디오 코딩 표준 들 또는 비디오 코덱에도 적용 가능할 수 있다.

[0019] **2. 초기 논의**

[0020] 비디오 코딩 표준들은 잘 알려진 ITU-T 및 ISO/IEC 표준들의 개발을 통해 주로 발전해 왔다. ITU-T는 H.261 및 H.263을 제작했으며, ISO/IEC는 MPEG-1 및 MPEG-4 Visual을 제작하였다. 두 기관은 H.262/MPEG-2 Video, H.264/MPEG-4 Advanced Video Coding (AVC) 그리고 H.265/HEVC 표준들을 공동으로 제작하였다. H.262 이후, 비디오 코딩 표준들은 시간 예측과 변환 코딩이 함께 활용되는 하이브리드 비디오 코딩 구조를 기반으로 한다. HEVC를 넘어서는 미래형 비디오 코딩 기술들을 탐구하기 위해, 2015년에 VCEG와 MPEG는 공동 비디오 탐사 팀 (Joint Video Exploration Team (JVET))를 공동 설립하였다. 그 이후로, 많은 새로운 기법들이 JVET에 의해 채택되어 공동 탐사 모델 (Joint Exploration Model (JEM)) [3, 4]이라는 참조 소프트웨어에 포함되었다.

2018년 4월, HEVC 대비 비트 레이트 50% 감소를 목표로하는 VVC 표준 작업을 위해, VCEG(Q6/16)와 ISO/IEC JTC1 SC29/WG11(MPEG) 간의 공동 비디오 전문가 팀 (Joint Video Expert Team (JVET))이 결성되었다.

[0021] VVC 초안(draft)의 최신 버전 즉, Versatile Video Coding (Draft 5)는 phenix.it-sudparis.eu/jvet/doc_end_user/documents/14_Geneva/wg11/JVET-N1001-v5.zip에서 확인할 수 있다.

[0022] VVC의 최신 참조 소프트웨어인 VTM은 vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-5.0에서 확인할 수 있다.

[0023] **2. HEVC/H.265에서 화면간 예측**

[0024] 화면간 예측된 각각의 예측 유닛(Prediction unit (PU))은 하나 또는 두 개의 참조 픽처 리스트에 대한 모션 파라미터들을 가진다. 모션 파라미터들은 모션 벡터 및 참조 픽처 인덱스를 포함한다. 상기 두 참조 픽처 리스트들 중 하나의 사용은 *inter_pred_idc*를 사용하여 시그널링될 수도 있다. 모션 벡터들은 예측자(predictor)들과 관련하여 델타들(deltas)로 명시적으로 인코딩될 수 있다.

[0025] CU가 스킵 모드로 코딩될 경우, 하나의 PU가 상기 CU와 연관되며, 의미있는 잔여 계수들, 코딩된 모션 벡터 델타 또는 참조 픽처 인덱스가 존재하지 않는다. 머지 모드가 지정되면, 현재 PU에 대한 모션 파라미터들이 공간적 및 시간적 후보들을 포함하여 주변 PU들로부터 획득된다. 상기 머지 모드는 스킵 모드에 대해서 뿐만 아니라, 어떠한 화면간 예측된 PU에도 적용될 수 있다. 머지 모드의 대안은 모션 파라미터들의 명시적인 전송으로, 이 경우 모션 벡터(더 자세히 말하면, 모션 벡터 차분들(MVD: motion vector differences)), 각 참조 픽처 리스트에 대한 해당 참조 픽처 인덱스, 그리고 참조 픽처 리스트 사용법이 각각의 PU별로 명시적으로 시그널링된다. 본 문서에서 이러한 모드는 향상된 모션 벡터 예측(AMVP: Advanced Motion Vector Prediction)이라고 명명된다.

[0026] 시그널링이 두 참조 픽처 리스트 중 하나를 사용함을 지시할 경우, 상기 PU는 샘플들 중 하나의 블록으로부터 생성된다. 이것을 '단-예측'(uni-prediction)이라고 한다. 단-예측은 P 슬라이스들과 B 슬라이스들 모두에 대해 사용할 수 있다.

[0027] 시그널링이 두 참조 픽처 리스트 모두를 사용함을 지시할 경우, 상기 PU는 샘플들 중 두 개의 블록으로부터 생성된다. 이를 '양-예측'(bi-prediction)이라고 한다. 양-예측은 B 슬라이스들에 대해서만 사용할 수 있다.

[0028] 아래 텍스트는 HEVC에 명시된 화면간 예측 모드들에 대한 상세한 사항들을 제시한다. 설명은 머지 모드에서부터 시작할 것이다.

[0029] **2.1.1. 참조픽처 리스트**

[0030] HEVC에서 용어 "화면내 예측"은 현재 디코딩된 픽처가 아닌 참조 픽처들의 데이터 요소들(예를 들어, 샘플 값들 또는 모션 벡터들)로부터 유도된 예측을 의미하는 데에 사용된다. H.264/AVC에서와 같이, 픽처는 다수의 참조 픽처들로부터 예측될 수 있다. 화면간 예측을 위해 사용되는 참조 픽처들은 하나 또는 그 이상의 참조 픽처 리스트들로 구성된다. 참조 인덱스는 리스트 상의 참조 픽처들 중 어느 것이 예측 신호를 생성하는 데에 사용되어야 하는지를 알려준다.

[0031] 단일 참조 픽처 리스트 List 0는 P 슬라이스 및 2개의 참조 픽처 리스트들에 대해 사용된다. List 0 및 List 1은 B 슬라이스들에 대해 사용된다. List 0/1에 포함된 참조 픽처들은 캡처/디스플레이 순서상 과거 및 미래 픽처들로부터 획득될 수 있다.

[0032] **2.1.2. 머지 모드**

[0033] 머지 모드를 사용하여 PU가 예측되는 경우, *머지 후보 리스트*의 항목을 가리키는 인덱스는 비트스트림으로부터 파싱되어 모션 정보를 검색하는 데 사용된다. 이 리스트의 구성은 HEVC 표준에 명시되며, 아래와 같은 순서의 단계들에 따라 요약될 수 있다.

- [0034] · 단계 1: 초기 후보들 유도
- [0035] ○ 단계 1.1: 공간적 후보들 유도
- [0036] ○ 단계 1.2: 공간적 후보들에 대한 중복성 검사
- [0037] ○ 단계 1.3: 시간적 후보들 유도

- [0038] · 단계 2: 추가 후보들 삽입
- [0039] o 단계 2.1: 양-예측(bi-predictive) 후보들 생성
- [0040] o 단계 2.2: 제로 모션 후보들 삽입
- [0041] 이 단계들은 도 1에서도 개략적으로 설명되고 있다. 공간적 머지 후보 유도를 위해, 5개의 상이한 위치에 있는 후보들 중에서 최대 4개의 머지 후보들이 선택된다. 시간적 머지 후보 유도를 위해, 두 후보들 중 최대 하나의 머지 후보가 선택된다. 각 PU에 대해 일정한 수의 후보들이 디코더에서 가정되기 때문에, 단계 1에서 획득되는 후보들의 수가 슬라이스 헤더에서 시그널링되는 머지 후보의 최대 수(MaxNumMergeCand)에 미치지 않으면, 추가 후보들이 생성된다. 후보들의 수가 일정하기 때문에 최적 머지 후보의 인덱스는 절단된 단항 이진화(TU)를 사용하여 인코딩 된다. 만약 CU의 크기가 8이면, 현재 CU의 모든 PU들은 2N x 2N 예측 유닛의 머지 후보 리스트와 동일한 단일 머지 후보 리스트를 공유한다.
- [0042] 아래에서, 위에 언급된 단계들과 연관된 작업들이 상세히 설명된다.
- [0043] 도 1은 머지 후보 리스트 구성을 위한 예시적 유도 과정을 보여준다.
- [0044] **2.1.2.2. 공간적 후보들 유도**
- [0045] 공간적 머지 후보들 유도 시, 최대 4개의 머지 후보들이 도 2에 도시된 위치들에 있는 후보들 가운데에서 선택된다. 유도 순서는 A_1 , B_1 , B_0 , A_0 및 B_2 순이다. 위치 A_1 , B_1 , B_0 의 어느 한 PU가 사용될 수 없는 경우(예를 들어, 그것이 다른 슬라이스 또는 타일에 속하기 때문) 또는 화면내 코딩된 경우에만, 위치 B_2 가 고려된다. 위치 A_1 의 후보가 추가된 후, 나머지 후보들의 추가는 중복 검사를 거친다. 이를 통해, 동일한 모션 정보를 가진 후보들이 리스트에서 제외되어 코딩 효율이 향상된다. 계산 복잡성을 줄이기 위해, 상기 언급된 중복 검사에서 가능한 모든 후보 쌍들이 고려되는 것은 아니다. 그 대신, 도 3에서 화살표로 연결된 쌍들만이 고려되고, 중복 검사에 사용되는 해당 후보가 동일한 모션 정보를 가지고 있지 않은 경우에만 후보가 리스트에 추가된다. 중복되는 모션 정보의 다른 소스는 2Nx2N과는 상이한 분할들과 관련된 "제2 PU"이다. 일 예로, 도4는 각각 Nx2N 및 2NxN의 경우에 대한 제2 PU를 도시한다. 현재 PU가 Nx2N으로 분할되면, A_1 위치에서 후보는 리스트 구성에 고려되지 않는다. 실제로, 이 후보를 추가함으로써 동일한 모션 정보를 갖는 2개의 예측 유닛들이 귀결될 것이며, 이는 코딩 유닛에서 단지 하나의 PU를 갖는 것에 불필요하다(redundant). 마찬가지로, 현재 PU가 2NxN으로 분할되는 경우, 위치 B_1 은 고려되지 않는다.
- [0046] **2.1.2.3. 시간적 후보들 유도**
- [0047] 이 단계에서는 오직 하나의 후보만이 상기 리스트에 추가된다. 특히, 이 시간적 머지 후보의 유도 시, 주어진 참조 픽처 리스트 내에서 현재 픽처와 가장 작은 POC(Picture Order Count) 차이를 갖는 픽처에 속하는 동일 위치의(co-located) PU를 기반으로, 스케일링된 모션 벡터가 유도된다. 상기 동일 위치 PU의 유도에 사용될 상기 참조 픽처 리스트는 슬라이스 헤더에서 명시적으로 시그널링된다. 시간적 머지 후보를 위한 상기 스케일링된 모션 벡터는 도의 점선으로 도시된 것과 같이 획득된다. 상기 모션 벡터는 POC 거리들인 t_b 및 t_d 를 사용하여 상기 동일 위치 PU의 모션 벡터로부터 스케일링 된다. 여기서, t_b 는 현재 픽처의 참조 픽처와 현재 픽처 사이의 POC 차이로 정의되고, t_d 는 동일 위치 픽처의 참조 픽처와 동일 위치 픽처 사이의 POC 차이로 정의된다. 시간적 머지 후보의 참조 픽처 인덱스는 0으로 설정된다. B 슬라이스의 경우, 2개의 모션 벡터 중 하나는 참조 픽처 리스트 0을 위해 다른 하나는 참조 픽처 리스트 1을 위해 획득되고, 이것들이 결합되어 양-예측(bi-predictive) 머지 후보가 만들어진다.
- [0048] 도 5는 시간적 머지 후보에 대한 모션 벡터 스케일링의 예이다.
- [0049] 참조 프레임에 속하는 동일 위치 PU(Y)에서, 시간적 후보에 대한 위치는 도 6에 도시된 바와 같이 후보 C_0 및 C_1 중에서 선택된다. 위치 C_0 의 PU가 사용될 수 없거나 화면내 코딩되었거나 현재 코딩 트리 유닛(CTU, 최대 코딩 유닛(LCU: largest coding unit)으로도 알려짐) 행의 외부에 있는 경우, 위치 C_1 이 사용된다. 그 외의 경우, 위치 C_0 이 시간적 머지 후보의 유도에 사용된다.
- [0050] 도 6은 시간적 머지 후보 C_0 및 C_1 에 대한 후보 위치들의 예들을 보여준다.

[0051] **2.1.2.4. 추가 후보들의 삽입**

[0052] 시공간적 머지 후보들 외에도 결합 양-예측(bi-predictive) 머지 후보 및 제로 머지 후보라는 두 가지 추가적인 유형의 머지 후보들이 있다. 결합 양-예측 머지 후보는 시공간적 머지 후보들을 활용하여 생성된다. 결합 양-예측 머지 후보는 오직 B 슬라이스에 대해서만 사용된다. 결합 양-예측 후보들은 최초 후보의 제1 참조 픽처 리스트 모션 파라미터들과 다른 후보의 제2 참조 픽처 리스트 파라미터들을 결합하여 생성된다. 이 두 튜플들이 상이한 모션 가설들을 제공하는 경우, 그것들은 새로운 양-예측 후보를 형성하게 된다. 일 예로, 도 7은 mvL0 및 refIdxL0 또는 mvL1 및 refIdxL1을 가지는 원본 리스트(좌측)의 두 후보들이 최종 리스트(우측)에 추가된 결합 양-예측 후보를 생성하는 데 사용되는 경우를 도시한다. 이러한 추가 머지 후보들을 생성하는 것으로 간주되는 결합들에 관한 많은 규칙들이 있다.

[0053] 제로 모션 후보들이 삽입되어 머지 후보 리스트의 나머지 항목들을 채우고 이로써 MaxNumMergeCand 용량에 도달한다. 이 후보들은 0에서 시작하여 새로운 제로 모션 후보가 리스트에 추가될 때마다 증가하는 참조 픽처 인덱스와 제로 공간 변위를 가진다.

[0054] 더 구체적으로, 아래 단계들이 머지 리스트가 채워질 때까지 순차적으로 실시된다.

[0055] 1. 변수 numRef를 P 슬라이스에 대한 리스트 0와 연관된 참조 픽처의 수 또는 B 슬라이스에 대한 2개의 리스트들에서의 참조 픽처들의 최소 수로 설정.

[0056] 2. 반복되지 않는(non-repeated) 제로 모션 후보들을 추가:

[0057] 0에서 시작하여 최대값이 numRef-1인 변수 i에 대해, MV가 (0, 0)으로 설정되고 참조 픽처 인덱스가 i로 설정된 디폴트 모션 후보를 리스트 0(P 슬라이스의 경우)에 대해 또는 양 리스트 모두에 대해(B 슬라이스의 경우) 추가.

[0058] 3. MV는 (0, 0)로 설정되고, 리스트 0의 참조 픽처 인덱스는 0으로(P 슬라이스의 경우) 그리고 양 리스트들의 참조 픽처 인덱스들은 0으로(B 슬라이스의 경우) 설정된 상태에서 반복되는 제로 모션 후보들을 추가.

[0059] 최종적으로, 이러한 후보들에 대해서는 중복성 검사가 실시되지 않는다.

[0060] **2.1.3. AMVP**

[0061] AMVP는 모션 파라미터들의 명시적 전송에 사용되는, 모션 벡터와 인접 PU들의 시공간적 상관 관계를 활용한다. 그것은, 먼저 좌측 및 상부의 시간적으로 인접한 PU 위치들의 가용성을 확인하고, 중복 후보들을 제거하고, 후보 리스트가 일정한 길이가 되도록 제로 벡터를 추가함으로써, 모션 벡터 후보 리스트를 구성한다. 다음으로, 인코더는 상기 후보 리스트에서 최상의 예측자를 선택하고 선택한 후보를 지시하는 해당 인덱스를 전송할 수 있다. 머지 인덱스 시그널링과 마찬가지로, 상기 최상의 모션 벡터 후보의 인덱스는 절단된 단항(truncated unary)을 사용하여 인코딩된다. 이 경우 인코딩 될 최대 값은 2이다(도 8 참조). 아래 섹션들에서는 모션 벡터 예측 후보의 유도 과정에 대해 상세히 설명한다.

[0062] **2.1.3.1. AMVP 후보들의 유도**

[0063] 도 8은 모션 벡터 예측 후보에 대한 유도 과정을 요약한다.

[0064] 모션 벡터 예측 시, 공간적 모션 벡터 후보와 시간적 모션 벡터 후보라는 두 가지 유형의 모션 벡터 후보들이 고려된다. 공간적 모션 벡터 후보 유도를 위해, 도 2에 도시된 바와 같이 5개의 상이한 위치에 있는 각 PU의 모션 벡터들에 기반하여 2개의 모션 벡터 후보가 최종적으로 유도된다.

[0065] 시간적 모션 벡터 후보 유도를 위해, 하나의 모션 벡터 후보가 2개의 상이한 동등 위치들을(co-located positions) 기반으로 도출된 2개의 후보들 중에서 선택된다. 시공간 후보들의 제1 리스트가 생성된 후, 중복된 모션 벡터 후보들이 상기 리스트에서 제거된다. 만약 잠재적 후보들의 수가 2보다 크면, 연관된 참조 픽처 리스트 내의 참조 픽처 인덱스가 1보다 큰 모션 벡터 후보는 상기 리스트에서 제거된다. 시공간적 모션 벡터 후보의 수가 2 미만인 경우, 추가 제로 모션 벡터 후보들이 상기 리스트에 추가된다.

[0066] **2.1.3.2. 공간적 모션 벡터 후보들**

[0067] 공간적 모션 벡터 후보들의 유도 시, 최대 2개의 후보들이 5개의 잠재적 후보들 가운데에서 고려되며, 이 잠재적 후보들은 도 2에 도시된 위치들에 있는 PU들로부터 유도된다. 이 위치들은 모션 머지의 위치들과 동일하다. 현재 PU의 좌측에 대한 유도 순서는 A₀, A₁, 스케일링된 A₀, 스케일링된 A₁ 순으로 정의된다. 현재 PU의 위쪽에

대한 유도 순서는 B_0, B_1, B_2 , 스케일링된 B_0 , 스케일링된 B_1 , 스케일링된 B_2 순으로 정의된다. 따라서, 각 측면에 대해 모션 벡터 후보로 사용할 수 있는 4개의 경우들이 있는데, 2개의 경우는 공간적 스케일링을 사용할 필요가 없는 경우이고, 2개의 경우는 공간적 스케일링이 사용되는 경우이다. 상기 4개의 서로 다른 경우들은 아래와 같이 요약된다.

- [0068] * 공간적 스케일링 없음
- [0069] (1) 동일한 참조 픽처 리스트 및 동일한 참조 픽처 인덱스 (동일한 POC)
- [0070] (2) 상이한 참조 픽처 리스트 및 동일한 참조 픽처 (동일한 POC)

- [0071] * 공간적 스케일링
- [0072] (3) 동일한 참조 픽처 리스트 및 상이한 참조 픽처 (상이한 POC)
- [0073] (4) 상이한 참조 픽처 리스트 및 상이한 참조 픽처 (상이한 POC)

[0074] 공간적 스케일링이 없는 경우들을 먼저 확인한 다음 공간적 스케일링을 허용하는 경우들을 확인한다. 공간적 스케일링은 참조 픽처 리스트와 상관없이 주변 PU의 참조 픽처와 현재 PU의 참조 픽처 사이의 POC가 다를 때 고려된다. 좌측 후보들의 모든 PU들이 사용될 수 없거나 화면내 코딩 된 경우, 좌측 및 상부 MV 후보들의 병렬 유도를 돕기 위해 상부 모션 벡터에 대한 스케일링이 허용된다. 그렇지 않은 경우, 공간적 스케일링은 상부 모션 벡터에 대해 허용되지 않는다.

[0075] 도 9는 공간적 모션 벡터 후보에 대한 모션 벡터 스케일링의 예를 보여준다.

[0076] 도 9에 도시되는 것과 같이, 공간적 스케일링 과정에서, 주변 PU의 모션 벡터는 시간적 스케일링의 경우와 유사한 방식으로 스케일링된다. 주요한 차이점은 현재 PU의 참조 픽처 리스트와 인덱스가 입력으로 제공된다는 점이다. 실제 스케일링 과정은 시간적 스케일링 프로세스와 동일하다.

[0077] **2.1.3.3. 시간적 모션 벡터 후보들**

[0078] 참조 픽처 인덱스 유도와는 별개로, 시간적 머지 후보들의 유도를 위한 모든 과정들은 공간적 모션 벡터 후보들의 유도를 위한 것과 같다(도 6 참조). 참조 픽처 인덱스는 디코더로 시그널링된다.

[0079] **2.2. JEM 에서 서브 CU 기반 모션 벡터 예측 방법들**

[0080] QTBT를 가지는 JEM에서, 각 CU는 각각의 예측 방향에 대해 모션 파라미터들의 한 세트까지 가질 수 있다. 큰 CU를 서브 CU들로 분할하고 상기 큰 CU의 상기 모든 서브 CU들에 대해 모션 정보를 유도함으로써, 2개의 서브 CU 레벨 모션 벡터 예측 방법들이 인코더에서 고려된다. 대안적인 시간적 모션 벡터 예측(ATMVP) 방법은 각 CU가 동일 위치의 참조 픽처에 있는 현재 CU 보다 작은 복수의 블록들로부터 모션 정보의 복수의 세트들을 불러오도록 허용한다. 시공간적 모션 벡터 예측(STMVP) 방법에서, 서브 CU들의 모션 벡터들은 시간적 모션 벡터 예측자 및 공간적 주변 모션 벡터를 사용하여 반복적으로 유도된다.

[0081] 서브 CU 모션 예측에 대해 좀 더 정확한 모션 필드를 유지하기 위해, 참조 프레임들에 대한 모션 압축이 현재 불가능하게 된다.

[0082] 도 10은 CU에 대한 ATMVP 모션 예측의 예를 보여준다.

[0083] **2.2.1. 대안적인 시간적 모션 벡터 예측**

[0084] 대안적인 시간적 모션 벡터 예측(ATMVP: alternative temporal motion vector prediction) 방법에서, 모션 벡터 시간적 모션 벡터 예측(TMVP: temporal motion vector prediction)은 현재 CU 보다 작은 블록들로부터 (모션 벡터들 및 참조 인덱스들을 포함하는) 모션 정보의 복수의 세트들을 불러오으로써 수정된다. 서브 CU들은 정사각형 $N \times N$ 블록들이다(N 은 디폴트로 4로 설정된다).

[0085] ATMVP는 CU 내에서 서브 CU들의 모션 벡터들을 두 단계로 예측한다. 첫 번째 단계는 소위 시간적 벡터를 가지는 참조 픽처에서 해당 블록을 식별하는 것이다. 상기 참조 픽처는 또한 모션 소스 픽처로도 지칭된다. 두 번째 단계는 현재 CU를 서브 CU들로 분할하고 각각의 서브 CU에 상응하는 블록으로부터 각 서브 CU의 참조 인덱스들뿐 아니라 모션 벡터들을 획득하는 것이다.

[0086] 첫 번째 단계에서, 참조 픽처 및 해당 블록은 현재 CU의 공간적 주변 블록들의 모션 정보로 판단된다. 주변 블록들에 대한 반복적인 스캐닝 과정을 방지하기 위해, 현재 CU의 머지 후보 리스트 내의 제1 머지 후보가 사용된

다. 제1 가용 모션 벡터뿐 아니라 그것과 연관된 참조 인덱스가 상기 모션 소스 픽처에 대한 시간적 벡터 및 인덱스로 설정된다. 이러한 방식으로, ATMVP에서, 해당 블록은 TMVP에 비해 더 정확하게 식별될 수 있으며, 이때, (간혹 동일 위치 블록으로 불리는) 해당 블록은 현재 CU에 대해 항상 우측 아래 또는 가운데 위치에 있다.

[0087] 두 번째 단계에서, 서브 CU의 해당 블록은 모션 소스 픽처 내의 시간적 벡터에 의해 또는 시간적 벡터에 현재 CU의 좌표를 추가함으로써 식별된다. 각각의 서브 CU에 대해, 그것에 상응하는 블록(가운데 샘플을 덮은 최소 모션 그리드)의 모션 정보는 상기 서브 CU에 대한 모션 정보를 유도하는 데에 사용된다. 해당 $N \times N$ 블록의 모션 정보가 식별된 후에, HEVC의 TMVP와 동일한 방식으로 현재 서브 CU의 모션 벡터들 및 참조 인덱스들로 변환된다. 이때, 모션 스케일링 및 기타 과정들이 적용된다. 예를 들어, 디코더는 저지연 조건(즉, 현재 픽처의 모든 참조 픽처들의 POC들이 현재 픽처의 POC 보다 작다)이 충족되는지 여부를 확인하고, 가능한 경우, 모션 벡터 MV_x (참조 픽처 리스트 X에 해당하는 모션 벡터)를 사용하여 각 서브 CU에 대해 모션 벡터 MV_y (X는 0 또는 1과 같고, Y는 1-X와 같다)를 예측한다.

[0088] **2.2.2. 시공간적 모션 벡터 예측 (STMVP)**

[0089] 이 방법에서, 서브 CU들의 모션 벡터들은 래스터 스캔 순서를 따라 재귀적으로 유도된다. 도 11은 이 개념을 도시한다. 네 개의 4×4 서브 CU들인 A, B, C 및 D를 포함하는 8×8 CU를 살펴보자. 현재 프레임에서 주변 4×4 블록들은 a, b, c 및 d로 표시된다.

[0090] 서브 CU A에 대한 모션 유도는 그것의 두 공간적 이웃들을 식별하면서 시작된다. 첫 번째 이웃은 서브 CU A 위에 있는 $N \times N$ 블록(블록 c)이다. 만약 이 블록 c가 가용하지 않거나 화면내 코딩된 것이라면, 서브 CU A 위의 다른 $N \times N$ 블록들을 (블록 c에서 출발하여 좌측에서 우측으로) 확인한다. 두 번째 이웃은 서브 CU A 좌측에 있는 블록(블록 b)이다. 만약 이 블록 b가 가용하지 않거나 화면내 코딩된 것이라면, 서브 CU A 좌측의 다른 $N \times N$ 블록들을 (블록 b에서 출발하여 위에서 아래로) 확인한다. 각각의 리스트에 대해 주변 블록들로부터 획득된 모션 정보는 주어진 리스트에 대한 제1 참조 프레임으로 스케일링된다. 다음으로, 서브 블록 A의 시간적 모션 벡터 예측자(TMVP)가 HEVC에 명시된 TMVP 유도와 같은 과정을 준수하여 유도된다. 블록 D에서 동일 위치 블록의 모션 정보가 획득되고 그에 맞게 스케일링된다. 마지막으로, 모션 정보를 조회하여 스케일링한 후, 각 참조 리스트에 대해 개별적으로 모든 가용한 모션 벡터들(3 이하)의 평균을 낸다. 평균화된 모션 벡터는 현재 서브 CU의 모션 벡터로 할당된다.

[0091] **2.2.3. 서브 CU 모션 예측 모드 시그널링**

[0092] 서브 CU 모드들은 추가적인 머지 후보들로서 가능하며, 상기 모드들을 시그널링하는데 요구되는 추가적인 구문 요소는 존재하지 않는다. ATMVP 모드 및 STMVP 모드를 나타내기 위해 각 CU의 머지 후보 리스트에 2개의 추가적인 머지 후보들이 추가된다. 만약 시퀀스 파라미터 세트가 ATMVP 및 STMVP가 가능함을 지시한다면, 최대 7개의 머지 후보들까지 사용된다. 추가적인 머지 후보들의 인코딩 로직은 HM에서의 머지 후보들에 대한 것과 같다. 즉, P 또는 B 슬라이스에서의 각 CU에 대해, 2개의 추가적인 머지 후보들에 대해 2개의 추가적인 RD 값이 필요하다.

[0093] JEM에서, 머지 인덱스의 모든 빈(bins)은 CABAC에 의해 문맥 코딩된다. 반면에, HEVC에서는, 오직 첫 번째 빈만이 문맥 코딩되며, 나머지 빈들은 문맥 바이패스 코딩된다.

[0094] **2.3. VVC 에서 화면간 예측 방법들**

[0095] 화면간 예측 개선을 위한 몇 가지 새로운 코딩 툴들로 MVD 시그널링을 위한 적응적 모션 벡터 차이 해상도 (AMVR: adaptive motion vector difference resolution), 어파인 예측 모드, 삼각 예측 모드(TPM: triangular prediction mode), ATMVP, 일반화된 양-예측(GBI: generalized bi-prediction) 그리고 양방향 광학 흐름(BIO: bi-directional optical flow)과 같은 것들이 있다.

[0096] **2.3.1. 적응적 모션 벡터 차이 해상도**

[0097] HEVC에서, (PU의 모션 벡터와 예측된 모션 벡터 간의) 모션 벡터 차이들(MVDs)은, use_inter_my_flag가 슬라이스 헤더에서 0과 같을 경우, 쿼터(quarter) 휘도 샘플들의 단위로 시그널링된다. VVC에서는 지역적으로 적응적인 모션 벡터 해상도(LAMVR: locally adaptive motion vector resolution)가 도입된다. VVC에서, MVD는 쿼터 휘도 샘플들, 정수 휘도 샘플들 또는 4 휘도 샘플들(즉, 1/4-픽셀, 1-픽셀, 4-픽셀)의 단위로 코딩된다. MVD 해상도는 코딩 유닛(CU) 레벨에서 제어되며, MVD 해상도 플래그들은 적어도 하나의 논-제로 MVD 구성요소를 가지는 각 CU에 대해 조건적으로 시그널링된다.

- [0098] 적어도 하나의 논-제로 MVD 구성요소를 가지는 CU에 대해, 제1 플래그는 쿼터 휘도 샘플 MV 정밀도가 상기 CU에서 사용되는지 여부를 지시하기 위해 시그널링된다. (1과 같은) 상기 제1 플래그가 쿼터 휘도 샘플 MV 정밀도가 사용되지 않음을 지시하는 경우, 정수 휘도 샘플 MV 정밀도 또는 4 휘도 샘플 MV 정밀도가 사용되는지 여부를 지시하기 위해 다른 플래그가 시그널링된다.
- [0099] CU의 제1 MVD 해상도 플래그가 0이거나 CU에 대해 코딩되지 않은 경우(상기 CU의 모든 MVD들이 0임을 의미), 쿼터휘도 샘플 MV 해상도가 상기 CU에 대해 사용된다. CU가 정수 휘도 샘플 MV 정밀도 또는 4 휘도 샘플 MV 정밀도를 사용하는 경우, 상기 CU에 대한 AMVP 후보 리스트내의 MVP들은 해당 정밀도로 라운딩된다.
- [0100] 인코더에서, CU 레벨 RD 체크를 사용하여 어느 MVD 해상도가 CU를 위해 사용될 것인지를 결정한다. 즉, CU 레벨 RD 체크는 각 MVD 해상도에 대해 3회 실시된다. 인코더의 속도를 높이기 위해, 아래 인코딩 방식들이 JEM에서 적용된다.
- [0101] · 정상 쿼터 휘도 샘플 MVD 해상도를 가지는 CU의 RD 체크 동안, 현재 CU의 모션 정보(정수 휘도 샘플 정확도)가 저장된다. (라운딩 후) 상기 저장된 모션 정보는, 정수 휘도 샘플 및 4 휘도 샘플 MVD 해상도를 가지는 동일한 CU에 대한 RD 체크 동안 추가 소범위 모션 벡터 개선을 위한 출발점으로 사용되어 시간이 많이 소요되는 모션 추정 과정이 3회 중복되지 않는다.
- [0102] · 4 휘도 샘플 MVD 해상도를 가지는 CU의 RD 체크는 조건적으로 적용된다. CU에 대해, 정수 휘도 샘플 MVD 해상도의 RD 코스트가 쿼터 샘플 MVD 해상도의 RD 코스트보다 훨씬 큰 경우, 상기 CU에 대한 4휘도 샘플 MVD 해상도의 RD 체크는 생략된다.
- [0103] 인코딩 과정은 도 12에 도시된다. 우선, 1/4 픽셀(pe1) MV가 테스트되며, RD 코스트가 계산되고 $RDCost0$ 로 표시된다. 다음으로, 정수 MV가 테스트되고 RC 코스트가 $RDCost1$ 으로 표시된다. 만약 $RDCost1 < th * RDCost0$ 라면(이때, th 는 양의 임계값), 4 픽셀 MV가 테스트되며, 그렇지 않을 경우, 4 픽셀 MV는 생략된다. 기본적으로, 정수 또는 4 픽셀 MV를 테스트하는 경우, 모션 정보 및 RD 코스트 등은 1/4 픽셀 MV에 대해 이미 알려져 있으며, 정수 또는 4 픽셀 MV의 인코딩 과정의 속도를 높이기 위해 재사용 될 수 있다.
- [0104] VVC에서, AMVR은 어파인 예측 모드에도 적용될 수 있으며, 이때 해상도들은 1/16 픽셀, 1/4 픽셀 및 1 픽셀에서 선택될 수 있다.
- [0105] 도 12는 상이한 MV 정밀도로 인코딩하는 순서도이다.
- [0106] **2.3.2. 삼각예측 모드**
- [0107] 삼각 예측 모드(TPM: triangular prediction mode)의 개념은 모션 보상된 예측을 위해 새로운 삼각 분할(partition)을 도입하기 위한 것이다. 도 13에 보이는 것과 같이, TPM은 CU를2개의 삼각 예측 유닛들로 대각선 또는 반 대각선 방향을 따라 분할한다. 상기 CU에서 각각의 삼각 예측 유닛은 단일한 단-예측 후보 리스트에서 유도된 자체 단-예측 모션 벡터 및 참조 프레임 인덱스를 사용하여 화면간 예측된다. 상기 삼각 예측 유닛들을 예측한 후 적응적 가중치부여 (adaptive weighting) 과정이 대각 모서리(edge)에 대해 수행된다. 그 다음으로, 변환 및 양자화 과정들이 전체 CU에 적용된다. 이 모드는 오직 머지 모드에만 적용됨을 유의해야 한다(주의: 스킵 모드는 특수한 머지 모드로 간주된다).
- [0108] **2.3.2.1. TPM을 위한 단-예측 후보 리스트**
- [0109] TPM 모션 후보 리스트로 불리는 단-예측 후보 리스트는 5개의 단-예측 모션 벡터 후보들로 구성된다. 그것은 도 14에 보이는 것과 같이 5개의 공간적 주변 블록들(1 내지 5) 및 2개의 시간적 동일 위치 블록들(6 및 7)을 포함하는 7개의 주변 블록들로부터 유도된다. 상기 7개의 주변 블록들의 모션 벡터들은 수집되고 단-예측 후보 리스트에 다음 순서로 포함된다: 양-예측 모션 벡터들의 L0 모션 벡터, 양-예측 모션 벡터들의 L1 모션 벡터, 그리고 양-예측 모션 벡터들의 L0 및 L1 모션 벡터들의 평균 모션 벡터. 만약 후보들의 수가 5 미만이라면, 제로 모션 벡터가 상기 리스트에 추가된다. TPM을 위해 이 리스트에 추가된 모션 후보들은 TPM 후보들이라 불리며, 시/공간적 블록들로부터 유도된 모션 정보는 정규 모션 후보들이라 불린다.
- [0110] 구체적으로 아래 단계들이 포함된다.
- [0111] 1) 공간적 주변 블록들로부터 유도된 정규 모션 후보를 추가할 때 완전 프루닝 작업들로 (도 14의 블록 1 내도 7에 해당하는) $A_1, B_1, B_0, A_0, B_2, Co1$ 및 $Co12$ 로부터 정규 모션 후보들을 획득한다.

- [0112] 2) 변수 numCurrMergeCand를 0으로 설정한다.
- [0113] 3) $A_1, B_1, B_0, A_0, B_2, Col$ 및 $Col2$ 로부터 유도된 각각의 정규 모션 후보들에 대해, 만약 프루닝되지 않고, numCurrMergeCand이 5 미만이면, 만약 상기 정규 모션 후보가 (리스트 0 또는 리스트 1으로부터) 단-예측이면, numCurrMergeCand가 1씩 증가하는 TPM 후보로 머지 리스트에 직접적으로 추가된다. 이러한 TPM 후보는 "본래적으로(originally) 단-예측된 후보"로 불린다. 완전 프루닝이 적용된다.
- [0114] 4) $A_1, B_1, B_0, A_0, B_2, Col$ 및 $Col2$ 로부터 유도된 각각의 정규 모션 후보들에 대해, 만약 프루닝되지 않고, numCurrMergeCand이 5 미만이면, 만약 상기 정규 모션 후보가 양-예측이면, 리스트 0에서 유도되는 모션 정보가 새로운 TPM 후보로서 TPM 머지 리스트에 추가되며(즉, 리스트 0으로부터의 단-예측으로 변형되며), numCurrMergeCand는 1씩 증가한다. 이러한 TPM 후보는 "절삭된 리스트0에서 예측된 후보"로 불린다. 완전 프루닝이 적용된다.
- [0115] 5) $A_1, B_1, B_0, A_0, B_2, Col$ 및 $Col2$ 로부터 유도된 각각의 정규 모션 후보들에 대해, 만약 프루닝되지 않고, numCurrMergeCand이 5 미만이면, 만약 상기 정규 모션 후보가 양-예측이면, 리스트 1에서 유도되는 모션 정보가 TPM 머지 리스트에 추가되며(즉, 리스트 1로부터의 단-예측으로 변형되며), numCurrMergeCand는 1씩 증가한다. 이러한 TPM 후보는 "절삭된 리스트1에서 예측된 후보"로 불린다. 완전 프루닝이 적용된다.
- [0116] 6) $A_1, B_1, B_0, A_0, B_2, Col$ 및 $Col2$ 로부터 유도된 각각의 정규 모션 후보들에 대해, 프루닝되지 않고, numCurrMergeCand이 5 미만이며, 상기 정규 모션 후보가 양-예측이면,
 - [0117] - 만약 리스트0 참조 픽처의 슬라이스 QP가 리스트1 참조 픽처의 슬라이스 QP 보다 작다면, 리스트1의 모션 정보는 우선 리스트0 참조 픽처로 스케일링되고, 두 MV들(하나는 원본 리스트0로부터, 다른 하나는 리스트1로부터 스케일링된 MV)의 평균이 TPM 머지 리스트에 추가되며, 이러한 후보는 numCurrMergeCand가 1씩 증가하는 가운데 리스트0 모션 후보로부터 획득된 평균 단-예측이라 불린다.
 - [0118] - 그렇지 않은 경우, 리스트0의 모션 정보는 먼저 리스트1 참조 픽처로 스케일링되고, 두 MV들(하나는 원본 리스트1로부터, 다른 하나는 리스트0로부터 스케일링된 MV)의 평균이 TPM 머지 리스트에 추가되며, 이러한 후보는 numCurrMergeCand가 1씩 증가하는 가운데 리스트1 모션 후보로부터 획득된 평균 단-예측이라 불린다. 완전 프루닝이 적용된다.
- [0119] 7) numCurrMergeCand가 5 미만이며, 제0 모션 벡터 후보들이 추가된다.
- [0120] 도 14는 주변 블록들의 예시적 위치들을 보여준다.
- [0121] 리스트에 후보를 삽입할 때, 만약 그것이 이전에 추가된 모든 후보들과 비교되어 그것들 중 하나와 동일한지를 확인해야 한다면, 이러한 과정을 완전 프루닝이라 부를 수 있다.
- [0122] **2.3.2.2. 적응적 가중치 부여 과정**
- [0123] 각각의 삼각 예측 유닛을 예측한 후에, 적응적 가중치부여 과정이 2개의 삼각 예측 유닛들 사이의 대각 모서리에 적용되어 전체 CU에 대한 최종 예측을 유도한다. 2개의 가중치 요인 그룹들이 아래와 같이 정의된다.
 - [0124] · 제1 가중치 요인 그룹: {7/8, 6/8, 4/8, 2/8, 1/8} 및 {7/8, 4/8, 1/8}은 휘도 및 색차 샘플들에 대해 각각 사용된다.
 - [0125] · 제2 가중치 요인 그룹: {7/8, 6/8, 5/8, 4/8, 3/8, 2/8, 1/8} 및 {6/8, 4/8, 2/8}는 휘도 및 색차 샘플들에 대해 각각 사용된다.
- [0126] 가중치 요인 그룹은 2개의 삼각 예측 유닛들의 모션 벡터들의 비교에 기반하여 선택된다. 제2 가중치 요인 그룹은 상기 2개의 삼각 예측 유닛들의 참조 픽처들이 서로 다르거나 그것들의 모션 벡터 차이가 16 픽셀 보다 큰 경우 사용된다. 그렇지 않은 경우, 제1 가중치 요인 그룹이 사용된다. 도 15는 일 예를 보여준다.
- [0127] **2.3.2.3. 삼각 예측 모드 (TPM)의 시그널링**
- [0128] TPM이 사용되는지 여부를 지시하는 일 비트 플래그가 가장 먼저 시그널링될 수 있다. 그 후에, (도 13에 도시되는 것과 같은) 2개의 분할 패턴들의 지시 및 2 파티션들 각각에 대해 선택된 머지 인덱스들이 더 시그널링된다.

[0129] **2.3.2.3.1. TPM 플래그의 시그널링**

[0130] 하나의 휘도 블록의 너비와 높이를 각각 W와 H로 표시하자. 만약 $W \cdot H < 64$ 이면, 삼각 예측 모드가 비활성화된다.

[0131] 한 블록이 어파인 모드로 코딩되면, 삼각 예측 모드가 역시 비활성화된다.

[0132] 한 블록이 머지 모드로 코딩되면, 일 비트 플래그가 시그널링되어 삼각 예측 모드가 상기 블록에 대해 활성화되는지 비활성화되는지 여부를 지시할 수 있다.

[0133] 상기 플래그는 아래 수학적식에 기반하여 3개의 문맥들로 코딩된다.

[0134]
$$\text{Ctx index} = ((\text{left block L available} \ \&\& \ \text{L is coded with TPM?}) \ 1: \ 0) + ((\text{Above block A available} \ \&\& \ \text{A is coded with TPM?}) \ 1: \ 0);$$

[0135] 도 15는 TPM 플래그 코딩에서 문맥 선택을 위해 사용되는 주변 블록들(A 및 L)을 보여준다.

[0136] **2.3.2.3.2. 2 분할 패턴들(도13에 도시)의 지시의 시그널링 및 2 파티션들 각각에 대해 선택된 머지 인덱스들**

[0137] 분할 패턴들, 2 파티션들의 머지 인덱스들은 함께 코딩된다는 사실에 유의해야 한다. 일부 구현들에서, 상기 2 파티션들은 동일한 참조 인덱스를 사용할 수 있도록 제한된다. 따라서, 2 (분할 패턴들) * N (머지 후보들의 최대수) * (N-1)개의 가능성들이 있으며, 이때 N은 5로 설정된다. 하나의 지시가 코딩되고, 상기 분할 패턴들 간의 매핑, 2개의 머지 인덱스들 그리고 코딩된 지시가 아래에서 정의된 배열(array)로부터 유도된다.

[0138] `const uint8_t g_TriangleCombination[TRIANGLE_MAX_NUM_CANDS][3] = {`

[0139] `{ 0, 1, 0 }, { 1, 0, 1 }, { 1, 0, 2 }, { 0, 0, 1 }, { 0, 2, 0 },`

[0140] `{ 1, 0, 3 }, { 1, 0, 4 }, { 1, 1, 0 }, { 0, 3, 0 }, { 0, 4, 0 },`

[0141] `{ 0, 0, 2 }, { 0, 1, 2 }, { 1, 1, 2 }, { 0, 0, 4 }, { 0, 0, 3 },`

[0142] `{ 0, 1, 3 }, { 0, 1, 4 }, { 1, 1, 4 }, { 1, 1, 3 }, { 1, 2, 1 },`

[0143] `{ 1, 2, 0 }, { 0, 2, 1 }, { 0, 4, 3 }, { 1, 3, 0 }, { 1, 3, 2 },`

[0144] `{ 1, 3, 4 }, { 1, 4, 0 }, { 1, 3, 1 }, { 1, 2, 3 }, { 1, 4, 1 },`

[0145] `{ 0, 4, 1 }, { 0, 2, 3 }, { 1, 4, 2 }, { 0, 3, 2 }, { 1, 4, 3 },`

[0146] `{ 0, 3, 1 }, { 0, 2, 4 }, { 1, 2, 4 }, { 0, 4, 2 }, { 0, 3, 4 } };`

[0147] 분할 패턴들 (45도 또는 135도) = `g_TriangleCombination[시그널링된 지시][0];`

[0148] 후보 A의 머지 인덱스 = `g_TriangleCombination[시그널링된 지시][1];`

[0149] 후보 B의 머지 인덱스 = `g_TriangleCombination[시그널링된 지시][2]`

[0150] 일단 상기 두 모션 후보들 A 및 B가 유도되면, 상기 두 파티션들(PU1 및 PU2)의 모션 정보가 A 또는 B로부터 설정될 수 있다. PU1가 머지 후보 A 또는 B의 모션 정보를 사용하는지 여부는 상기 두 모션 후보들의 예측 방향들에 의존한다. 표1은 상기 두 파티션들과 함께, 2개의 유도된 모션 후보들 A와 B 사이의 관계를 보여준다.

표 1

[0151]

A의 예측 방향	B의 예측 방향	PU1의 모션 정보	PU2의 모션 정보
LO	LO	A (LO)	B (LO)
L1	L1	B (L1)	A (L1)
LO	L1	A (LO)	B (L1)
L1	LO	B (LO)	A (L1)

[0152] <표1: 유도된 머지 후보들(A, B)로부터 파티션들의 모션 정보의 유도>

[0153] **2.3.2.3.3. (merge_triangle_idx 에 의해 표시되는) 지시의 엔트로피 코딩**

[0154] `merge_triangle_idx`는 포괄적으로 범위 [0, 39]내 있다. K차 지수 골롬(EG: Exponential Golomb) 코드가

merge_triangle_idx의 이진화를 위해 사용되며, 여기서 K는 1로 설정된다.

[0155] (더 작은 수들을 인코딩하는 데에 더 많은 비트들을 사용하는 것을 감수하여) 더 작은 비트들에서 더 큰 수들을 인코딩하기 위해, 이것은 음이 아닌(nonnegative) 정수 파라미터 k를 사용하여 일반화될 수 있다. 음이 아닌 정수 x를 k차 지수 곱셈 코드로 인코딩하기 위해,

[0156] 1. 위에서 설명된 0차 지수 곱셈 코드를 사용하여 $x/2^k$ 를 인코딩한다. 그리고 나서,

[0157] 2. $x \bmod 2^k$ 를 이진으로 인코딩한다.

표 2

x	k=0	k=1	k=2	x	k=0	k=1	k=2
0	1	10	100	10	0001011	001100	01110
1	010	11	101	11	0001100	001101	01111
2	011	0100	110	12	0001101	001110	0010000
3	00100	0101	111	13	0001110	001111	0010001
4	00101	0110	01000	14	0001111	00010000	0010010
5	00110	0111	01001	15	000010000	00010001	0010011
6	00111	001000	01010	16	000010001	00010010	0010100
7	0001000	001001	01011	17	000010010	00010011	0010101
8	0001001	001010	01100	18	000010011	00010100	0010110
9	0001010	001011	01101	19	000010100	00010101	0010111

[0159] <표 2: 지수-곱셈-k 코딩 예들>

[0160] 2.3.3. 어파인 모션 보상 예측

[0161] HEVC에서, 오직 병진 모션 모델만이 모션 보상 예측(MCP: motion compensation prediction)에 대해 적용된다. 반면 실제 세계에는 줌 인/아웃, 회전, 원근적 움직임 및 기타 불규칙 움직임 같은 여러 종류의 모션들이 있다. VVC에서, 단순화된 어파인 변환 모션 보상 예측은 4-파라미터 어파인 모델 및 6-파라미터 어파인 모델과 함께 적용된다. 도 16에 보이는 것과 같이, 블록의 어파인 모션 필드는 상기 4-파라미터 어파인 모델에 대한 2개의 제어 포인트 모션 벡터들(CPMVs) 그리고 상기 6-파라미터 어파인 모델에 대한 3개의 CPMV들에 의해 기술된다.

[0162] 도 16은 단순화된 어파인 모션 모델들의 예들을 보여준다.

[0163] 블록의 모션 벡터 필드(MVF)는 아래 수학식들에 의해 기술되는데, 수학식 (1)에서는 4-파라미터 어파인 모델로 (이때, 4 파라미터들은 변수들 a, b, e 및 f로 정의된다), 수학식 (2)에서는 6-파라미터 어파인 모델로(이때, 6 파라미터들은 변수들 a, b, c, d, e 및 f로 정의된다)로 각각 기술된다.

수학식 1

$$\begin{cases} mv^h(x, y) = ax - by + e = \frac{(mv_1^h - mv_0^h)}{w}x - \frac{(mv_1^v - mv_0^v)}{w}y + mv_0^h \\ mv^v(x, y) = bx + ay + f = \frac{(mv_1^v - mv_0^v)}{w}x + \frac{(mv_1^h - mv_0^h)}{w}y + mv_0^v \end{cases}$$

[0164]

수학식 2

$$\begin{cases} mv^h(x, y) = ax + cy + e = \frac{(mv_1^h - mv_0^h)}{w}x + \frac{(mv_2^h - mv_0^h)}{h}y + mv_0^h \\ mv^v(x, y) = bx + dy + f = \frac{(mv_1^v - mv_0^v)}{w}x + \frac{(mv_2^v - mv_0^v)}{h}y + mv_0^v \end{cases}$$

[0165]

[0166]

여기서, (mv_0^h, mv_0^v) 는 좌측 상단 코너 제어 포인트의 모션 벡터이며, (mv_1^h, mv_1^v) 는 우측 상단 코너 제어 포인트의 모션 벡터, 그리고 (mv_2^h, mv_2^v) 는 좌측 하단 코너 제어 포인트의 모션 벡터로, 이 세 모션 벡터들은 모두 제어 포인트 모션 벡터들(CPMV: control point motion vectors)로 불린다. (x, y) 는 현재 블록 내부의 좌측 상단 샘플에 대한 대표 포인트의 좌표이며, $(mv^h(x, y), mv^v(x, y))$ 는 (x, y) 에 위치하는 샘플에 대해 유도된 모션 벡터이다. 상기 CP 모션 벡터들은 (어파인 AMVP 모드에서처럼) 시그널링되거나 (어파인 머지 모드에서처럼) 그때 그때 유도될 수 있다. w 와 h 는 현재 블록의 너비와 높이이다. 실제로, 라운딩 연산을 동반한 우측 시프트(right-shift)으로 나눗셈을 실시한다. VTM에서, 상기 대표 포인트는 서브 블록의 중심 위치로 정의된다. 즉, 현재 블록 내의 좌측 상단 샘플에 관련된 서브 블록의 좌측 상단 코너의 좌표가 (x_s, y_s) 일 때, 상기 대표 포인트의 좌표는 (x_s+2, y_s+2) 로 정의된다. 각각의 서브 블록(즉, VTM에서 4x4)에 대해, 상기 대표 포인트는 전체 서브 블록의 모션 벡터를 유도하는 데에 활용된다.

[0167]

모션 보상 예측을 더 단순화시키기 위해, 서브 블록 기반 어파인 변환 예측이 적용된다. 각 $M \times N$ 서브 블록 (현재 VVC에서 M 과 N 모두 4로 설정)의 모션 벡터를 유도하기 위해, 각 서브 블록의 가운데 샘플의 모션 벡터가 도 17에 보이는 것과 같이 수학식 (1) 및 수학식 (2)에 따라 계산되고, 1/16 분수 정확도로 라운딩된다. 그 다음으로, 유도된 모션 벡터를 가지는 각 서브 블록의 예측을 생성하기 위해, 1/16 픽셀에 대한 모션 보상 보간 필터들이 적용된다. 1/16 픽셀에 대한 상기 보간 필터들은 어파인 모드에 의해 도입된다.

[0168]

MCP 후에, 각 서브 블록의 고정확도(high accuracy) 모션 벡터는 정규 모션 벡터(normal motion vector)와 같은 정확도로 라운딩되고 보존된다.

[0169]

2.3.3.1. 어파인 예측의 시그널링

[0170]

병진 모션 모델과 유사하게, 어파인 예측에 기인하는 부가 정보를 시그널링하기 위한 2개의 모드들이 있다. 이 것들은 AFFINE_INTER 및 AFFINE_MERGE 모드들이다.

[0171]

2.3.3.2. AF_INTER 모드

[0172]

너비와 높이가 모두 8보다 큰 CU들에 대해서는 AF_INTER 모드가 적용될 수 있다. CU 레벨에서 어파인 플래그는 AF_INTER 모드가 사용되는지 여부를 지시하기 위해 비트스트림으로 시그널링된다.

[0173]

이 모드에서, 각각의 참조 픽처 리스트 (List 0 또는 List 1)에 대해, 어파인 AMVP 후보 리스트가 3가지 타입의 어파인 모션 예측자들을 아래 순서로 사용하여 구성된다. 여기서, 각 후보는 현재 블록의 추정된 CPMV들을 포함한다. 인코더 측에서 발견되는 (도 20의 mv_0, mv_1, mv_2 와 같은) 최적의 CPMV들과 추정된 CPMV의 차이들이 시그널링된다. 또한, 추정된 CPMV들이 유도되는 어파인 AMVP 후보의 인덱스가 더 시그널링된다.

[0174]

1) 승계된(inherited) 어파인 모션 예측자들

[0175]

점점 순서는 HEVC AMVP 리스트 구성에서 공간적 MVP들의 순서와 유사하다. 첫째, 좌측의 승계된 어파인 모션 예측자가 어파인 코딩되고 현재 블록에서와 동일한 참조 픽처를 가지는 $\{A1, A0\}$ 에서의 첫 번째 블록으로부터 유도된다. 둘째, 상부의 승계된 어파인 모션 예측자가 어파인 코딩되고 현재 블록에서와 동일한 참조 픽처를 가지는 $\{B1, B0, B2\}$ 에서의 첫 번째 블록으로부터 유도된다. 5개의 블록들 A1, A0, B1, B0 그리고 B2는 도 19에 도시된다.

[0176]

주변 블록이 어파인 모드로 코딩된 것으로 확인되면, 상기 주변 블록을 덮는 코딩 유닛의 CPMV들을 사용하여 현

재 블록의 CPMV들의 예측자들을 유도한다. 예를 들어, 만약 A1이 비-어파인 모드로 코딩되고 A0가 4-파라미터 어파인 모드로 코딩된다면, 상기 좌측의 승계된 어파인 MV 예측자는 A0으로부터 유도될 것이다. 이 경우, 도 21B에서 좌측 상단 CPMV에 대해 MV_0^N 로 표시되고 우측 상단 CPMV에 대해 MV_1^N 로 표시되는, A0을 덮는 CU의 CPMV들을 활용하여 현재 블록의 좌측 상단 (좌표 (x0, y0)), 우측 상단 (좌표 (x1, y1)) 및 우측 하단(좌표 (x2, y2)) 위치들에 대해 MV_0^C , MV_1^C 및 MV_2^C 로 표시되는 현재 블록의 추정된 CPMV들을 유도한다.

[0177] 2) 구성된 어파인 모션 예측자들

[0178] 구성된 어파인 모션 예측자는, 도 20에서 보이는 것과 같이, 동일한 참조 픽처를 가지는 주변 화면간 예측된 블록들로부터 유도된 제어 포인트 모션 벡터들(CPMVs)로 구성된다. 만약 현재 어파인 모션 모델이 4-파라미터 어파인이라면, CPMV들의 수는 2이고, 그렇지 않고서 만약 현재 어파인 모션 모델이 6-파라미터 어파인이라면, CPMV들의 수는 3이다. 화면간 코딩되고 현재 블록에서와 같은 참조 픽처를 가지는 그룹 {A, B, C}의 첫 번째 블록에서의 MV에 의해 좌측 상단 CPMV \overline{mv}_0 가 유도된다. 화면간 코딩되고 현재 블록에서와 같은 참조 픽처를 가지는 그룹 {D, E}의 첫 번째 블록에서의 MV에 의해 우측 상단 CPMV \overline{mv}_1 가 유도된다. 화면간 코딩되고 현재 블록에서와 같은 참조 픽처를 가지는 그룹 {F, G}의 첫 번째 블록에서의 MV에 의해 좌측 하단 CPMV \overline{mv}_2 가 유도된다.

[0179] - 만약 현재 어파인 모션 모델이 4-파라미터 어파인이라면, 오직 \overline{mv}_0 및 \overline{mv}_1 이 모두 발견되는 경우에만, 즉, \overline{mv}_0 및 \overline{mv}_1 이 현재 블록의 좌측 상단 (좌표 (x0, y0)) 및 우측 상단 (좌표 (x1, y1)) 위치들에 대한 추정된 CPMV들로 사용되는 경우에만, 구성된 어파인 모션 예측자는 후보 리스트로 삽입된다.

[0180] - 만약 현재 어파인 모션 모델이 6-파라미터 어파인이라면, 오직 \overline{mv}_0 , \overline{mv}_1 및 \overline{mv}_2 이 모두 발견되는 경우에만, 즉, \overline{mv}_0 , \overline{mv}_1 및 \overline{mv}_2 이 현재 블록의 좌측 상단 (좌표 (x0, y0)), 우측 상단 (좌표 (x1, y1)) 그리고 우측 하단(좌표 (x2, y2)) 위치들에 대한 추정된 CPMV들로 사용되는 경우에만, 구성된 어파인 모션 예측자는 후보 리스트로 삽입된다.

[0181] 구성된 어파인 모션 예측자를 후보 리스트에 삽입할 때에는 어떤 프루닝 과정도 적용되지 않는다.

[0182] 3) 일반(normal) AMVP 모션 예측자들

[0183] 어파인 모션 예측자들의 수가 최대치에 도달할 때까지 아래가 적용된다.

[0184] 1) 이용 가능하면, 모든 CPMV들을 \overline{mv}_2 와 같이 설정하여 어파인 모션 예측자를 유도한다.

[0185] 2) 이용 가능하면, 모든 CPMV들을 \overline{mv}_1 와 같이 설정하여 어파인 모션 예측자를 유도한다.

[0186] 3) 이용 가능하면, 모든 CPMV들을 \overline{mv}_0 와 같이 설정하여 어파인 모션 예측자를 유도한다.

[0187] 4) 이용 가능하면, 모든 CPMV들을 HEVC TMVP와 같이 설정하여 어파인 모션 예측자를 유도한다.

[0188] 5) 모든 CPMV들을 제로 MV로 설정하여 어파인 모션 예측자를 유도한다.

[0189] \overline{mv}_i 는 구성된 어파인 모션 예측자에서 이미 유도되었음에 유의하라.

[0190] AF_INTER 모드에서 4/6-파라미터 어파인 모드가 사용되는 경우, 2/3 제어 포인트들이 요구되며, 따라서, 도 18에 보이는 것처럼, 2/3 MVD가 이 제어 포인트들에 대해 코딩될 필요가 있다. JVET-K0337에서, MV를 다음과 같이 유도하라고 제안된다. 즉, mvd_0 으로부터 mvd_1 및 mvd_2 가 예측된다.

$$mv_0 = \overline{mv}_0 + mvd_0$$

$$mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

$$mv_2 = \overline{mv}_2 + mvd_2 + mvd_0$$

[0191]

[0192] 여기서, 도 18(b)에 도시된 것처럼, \overline{mv}_i , mvd_i 및 mv_i 은 각각 예측된 모션 벡터, 모션 벡터 차이, 그리고 좌측 상단 픽셀($i = 0$), 우측 상단 픽셀($i = 1$) 또는 좌측 하단 픽셀($i = 2$) 각각의 모션 벡터이다. 두 모션 벡터들의 합은(예를 들어, $mvA(xA, yA)$ 및 $mvB(xB, yB)$) 두 구성요소들 각각의 총합과 같다는 것을 유의하라. 예를 들어, $newMV = mvA + mvB$ 이며, $newMV$ 의 두 구성요소들이 각각 $(xA + xB)$ 및 $(yA + yB)$ 로 설정됨에 유의하라.

[0193]

2.3.3.3. AF_MERGE 모드

[0194] CU가 AF_MERGE 모드에서 적용되는 경우, 그것은 복원된 유효한 주변 블록들로부터 어파인 모드로 코딩된 제1 블록을 획득한다. 도 21(a)에 도시된 바와 같이, 후보 블록의 선택 순서는 좌측에서 출발하여 상단, 우측 상단, 좌측 하단 그리고 좌측 상단 순이다(A, B, C, D, E 순으로 표시). 예를 들어, 좌측 하단 주변 블록이 도 21(b)에서 A0으로 표시된 것과 같이 어파인 모드로 코딩된 경우, 상기 블록 A를 포함하는 주변 CU/PU의 좌측 상단 코너, 우측 상단 코너 및 좌측 하단 코너의 제어 포인트(CP) 모션 벡터들 mv_0^N , mv_1^N 및 mv_2^N 이 획득된다. 그리고 상기 현재 CU/PU의 좌측 상단 코너/우측 상단/좌측 하단의 (오직 6-파라미터 어파인 모델에 대해서만 사용되는) 모션 벡터 mv_0^C , mv_1^C 및 mv_2^C 는 mv_0^N , mv_1^N 및 mv_2^N 를 기반으로 산출된다. VTM-2.0에서 좌측 상단 코너에 위치하는 서브 블록(예를 들어, VTM의 4X4 블록)은 mv_0 을 보관하고, 우측 상단 코너에 위치하는 서브 블록은, 현재 블록이 어파인 코딩된 경우, mv_1 을 보관한다는 사실에 유의해야 한다. 만약 현재 블록이 6-파라미터 어파인 모델로 코딩되는 경우, 좌측 하단 코너에 위치한 서브 블록은 mv_2 를 저장하며, 그렇지 않은 경우(4-파라미터 어파인 모델로 코딩되는 경우) LB는 mv_2' 를 저장한다. 다른 서브 블록들은 MC를 위해 사용되는 MV들을 저장한다.

[0195] 현재 CU의 CPMV들 mv_0^C , mv_1^C 및 mv_2^C 이 유도된 후, 단순화된 어파인 모션 모델 수학적식 (1) 및 (2)에 따라 상기 현재 CU의 MVF가 생성된다. 상기 현재 CU가 AF_MERGE 모드로 코딩되는지 여부를 확인하기 위해, 어파인 모드로 코딩되는 적어도 하나의 주변 블록이 있을 때, 어파인 플래그가 비트스트림에서 시그널링된다.

[0196] JVET-L0142 및 JVET-L0632에서, 어파인 머지 후보 리스트는 아래 단계들로 구성된다.

[0197] 1) 승계된 어파인 후보들을 삽입한다.

[0198] 승계된 어파인 후보가 의미하는 것은, 상기 후보가 그것의 주변 어파인 코딩된 블록의 어파인 모션 모델로부터 유도되었다는 것이다. 최대 2개의 승계된 어파인 후보들이 주변 블록들의 어파인 모션 모델로부터 유도되고 후보 리스트에 삽입된다. 좌측 예측자에 대해, 스캔 순서는 {A0, A1}이고, 상부 예측자에 대해, 스캔 순서는 {B0, B1, B2}이다.

[0199] 2) 구성된 어파인 후보들을 삽입한다.

[0200] 어파인 머지 후보 리스트 상의 후보들의 수가 MaxNumAffineCand (예를 들어, 5) 미만이면, 구성된 어파인 후보들이 후보 리스트에 삽입된다. 구성된 어파인 후보가 의미하는 것은, 상기 후보가 각 제어 포인트의 주변 모션 정보를 결합하여 구성된다는 것이다.

[0201] a) 제어 포인트에 대한 모션 정보는 먼저 도 22에 보이는 명시된 공간적 주변 블록들 및 시간적 주변 블록으로부터 유도된다. CP_k ($k=1, 2, 3, 4$)는 k 번째 제어 포인트를 나타낸다. A0, A1, A2, B0, B1, B2 및 B3은 CP_k ($k=1, 2, 3$)를 예측하기 위한 공간적 위치들이다. T는 CP_4 를 예측하기 위한 시간적 위치이다.

[0202] CP1, CP2, CP3 및 CP4의 좌표들은 각각 (0, 0), (W, 0), (H, 0) 및 (W, H)이며, 여기서 W 및 H는 현재 블록의 너비와 높이이다.

[0203] 각 제어 포인트의 모션 정보는 아래 우선 순위에 따라 획득된다.

- [0204] - CP1에 대해, 점검 우선순위는 B2->B3->A2. B2가 사용가능한 경우 사용된다. 그렇지 않은 경우, B2가 사용 가능하면, B3가 사용된다. 만약 B2와 B3가 모두 사용 가능하지 않다면, A2가 사용된다. 만약 상기 세 후보들이 모두 사용 가능하지 않다면, CP1의 모션 정보가 획득될 수 없다.
- [0205] - CP2에 대해, 점검 우선순위는 B1->B0이다.
- [0206] - CP3에 대해, 점검 우선순위는 A1->A0이다.
- [0207] - CP4에 대해, T가 사용된다.
- [0208] b) 두 번째로, 제어 포인트들의 조합들은 어파인 머지 후보를 구성하는 데에 사용된다.
- [0209] I. 3개의 제어 포인트들의 모션 정보가 6-파라미터 어파인 후보를 구성하기 위해 필요하다. 상기 3개의 제어 포인트들은 다음 4가지 조합들 가운데 하나에서 선택될 수 있다: {CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4} 및 {CP1, CP3, CP4}. 상기 조합들{CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4} 및 {CP1, CP3, CP4}은 좌측 상단, 우측 상단 및 좌측 하단 제어 포인트들로 표현되는 6-파라미터 모션 모델로 변환될 것이다.
- [0210] II. 2개의 제어 포인트들의 모션 정보가 4-파라미터 어파인 후보를 구성하기 위해 필요하다. 상기 2개의 제어 포인트들은 다음 2가지 조합들 가운데 하나에서 선택될 수 있다: {CP1, CP2} 및 {CP1, CP3}. 상기 2가지 조합들은 좌측 상단 및 우측 상단 제어 포인트들로 표현되는 4-파라미터 모션 모델로 변환될 것이다.
- [0211] III. 구성된 어파인 후보들의 조합들은 다음 순서로 후보 리스트에 삽입된다.
- [0212] {CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}
- [0213] i. 각 조합에 대해, 각 CP에 대한 리스트 X의 참조 인덱스들이 점검되며, 만약 그것들이 모두 같다면, 이 조합은 리스트 X에 대해 유효한 CPMV들을 가진다. 만약 상기 조합이 리스트 0 및 리스트 1 모두에 대해 유효한 CPMV들을 가지고 있지 않다면, 이 조합은 무효(invalid)로 표시된다. 그렇지 않은 경우, 그것은 유효하며(valid), CPMV들은 서브 블록 머지 리스트로 포함된다.
- [0214] 3) 제로 어파인 모션 벡터 후보들로 패딩
- [0215] 어파인 머지 후보 리스트 상의 후보들의 수가 5 미만이면, 서브 블록 머지 후보 리스트에 대해, 4-파라미터 머지 후보의 MV들은 (0, 0)으로 설정되고, 예측 방향은 리스트 0으로부터 단-예측(P 슬라이스) 및 양-예측(B 슬라이스)로 설정된다.
- [0216] **2.3.4. VVC 에서 머지리스트 설계**
- [0217] VVC에서 지원되는 3가지 상이한 머지 리스트 구성 과정들이 있다.
- [0218] 1) **서브 블록 머지 후보 리스트:** 이것은 ATMVP 및 어파인 머지 후보들을 포함한다. 하나의 머지 리스트 구성 과정이 어파인 모드를 및 ATMVP 모드 모두에 대해 공유된다. 여기서, 상기 ATMVP 및 어파인 머지 후보들은 순서대로 추가될 수 있다. 서브 블록 머지 리스트 크기는 슬라이스 헤더에서 시그널링되며, 최대 값은 5이다.
- [0219] 2) **단-예측 TPM 머지 리스트:** 삼각 예측 모드에 대해, 2개의 파티션들이 각각의 머지 후보 인덱스를 선택하더라도, 2개의 파티션들을 위한 하나의 머지 리스트 구성 과정은 공유된다. 이 머지 리스트를 구성할 때, 블록의 공간적 주변 블록들 및 2개의 시간적 블록들이 확인된다. 공간적 주변 블록들 및 시간적 블록들로부터 유도된 모션 정보는 우리의 IDF에서 정규 모션 후보들이라 불린다. 이러한 정규 모션 후보들은 다수의 TPM 후보들을 유도하는 데에 더 활용된다. 변환은 전체 블록 레벨에서 실시되며, 2개의 파티션들이라도 그것들 자신의 예측 블록들을 생성하기 위해 상이한 모션 벡터들을 사용할 수 있다는 것에 유의해야 한다. 단-예측 TPM 머지 리스트 크기는 5로 고정된다.
- [0220] 3) **정규(regular) 머지 리스트:** 나머지 코딩 블록들에 대해, 하나의 머지 리스트 구성 과정이 공유된다. 여기서, 공간적/시간적/HMVP, 쌍(pairwise) 결합 양-예측 머지 후보들 및 제로 모션 후보들이 순서대로 삽입될 수 있다. 정규 머지 리스트 크기는 슬라이스 헤더에서 시그널링되며, 최대 값은 6이다.
- [0221] **2.3.4.1. 서브 블록 머지 후보 리스트**
- [0222] 비(non) 서브 블록 머지 후보들을 위한 정규 머지 리스트에 덧붙여 모든 서브 블록 관련 모션 후보들을 별도의 머지 리스트에 포함시킬 것이 제안된다.

- [0223] 상기 서브 블록 관련 모션 후보들이 포함되는 별도의 머지 리스트는 "서브 블록 머지 후보 리스트"로 불린다.
- [0224] 일 예에서, 서브 블록 머지 후보 리스트는 어파인 머지 후보들, ATMVP 후보, 그리고/또는 서브 블록 기반 STMVP 후보를 포함한다.
- [0225] **2.3.4.1.1. JVET-L0278**
- [0226] 이 기고문에서, 일반(normal) 머지 리스트 상의 ATMVP 머지 후보는 어파인 머지 리스트의 제1 위치로 이동된다. 그래서 새로운 리스트 (즉, 서브 블록 기반 머지 후보 리스트) 상의 모든 머지 후보들이 서브 블록 코딩 툴들(tools)에 기반하게 된다.
- [0227] **2.3.4.1.2. JVET-N1001 에서 ATMVP**
- [0228] ATMVP는 서브 블록 기반 시간적 모션 벡터 예측(SbTMVP: subblock-based temporal motion vector prediction)으로도 알려졌다.
- [0229] JVET-N1001에서, (어파인 머지 후보 리스트로도 알려진) 서브 블록 머지 후보 리스트로 알려진 특수 머지 후보 리스트가 정규 머지 후보 리스트 외에 추가된다. 상기 서브 블록 머지 후보 리스트는 다음 순서로 후보들로 채워진다.
 - [0230] a. ATMVP 후보 (이용 가능하거나 이용 가능하지 않을 수 있다)
 - [0231] b. 승계된 어파인 후보들
 - [0232] c. 동일 위치의(collocated) 참조 픽처에서 MV들을 이용하는 TMVP 기반의 구성된 어파인 후보들을 포함하는, 구성된 어파인 후보들
 - [0233] d. 제로 MV 4-파라미터 어파인 모델로서 패딩
- [0234] VTM은 서브 블록 기반 시간적 모션 벡터 예측(SbTMVP) 방법을 지원한다. HEVC에서의 시간적 모션 벡터 예측(TMVP)와 마찬가지로, SbTMVP는 동일 위치의 픽처에서 모션 필드를 사용하여 현재 픽처 내의 CU들에 대한 모션 벡터 예측 및 머지 모드를 향상시킨다. TMVP에 의해 이용되는 같은 동일 위치 픽처는 SbTMVP를 위해 이용된다. SbTMVP는 아래 두 가지 주요한 측면에서 TMVP와 다르다.
 - [0235] 1. TMVP는 CU 레벨에서 모션을 예측하지만, SbTMVP는 서브 CU 레벨에서 모션을 예측한다.
 - [0236] 2. TMVP가 동일 위치의 픽처 내의 동일 위치의 블록으로부터 시간적 모션 벡터들을 가지고 오는 반면에 (동일 위치의 블록은 현재 CU에 대해 우측 하단 또는 가운데 블록이다), SbTMVP는 시간적 모션 정보를 동일 위치의 픽처로부터 가져오기 전에 모션 시프트를 적용한다. 이때, 모션 시프트는 현재 CU의 공간적 주변 블록들 중 하나에서의 모션 벡터로부터 획득된다.
 - [0237] SbTMVP 절차는 도 23 및 도 24에 도시된다. SbTMVP는 현재 CU내의 서브 CU들의 모션 벡터들을 두 단계로 예측한다. 첫 번째 단계에서, 도 23의 공간적 주변 블록 A1이 검토된다. 만약 A1이 그것의 참조 픽처와 같은 동일 위치 픽처를 사용하는 모션 벡터를 가진다면, 이 모션 벡터는 적용될 모션 시프트로 선택된다. 만약 그러한 모션이 식별되지 않으며, 모션 시프트는 (0, 0)으로 설정된다.
 - [0238] 두 번째 단계에서, 단계 1에서 식별된 모션 시프트가 적용되어(즉, 현재 블록의 좌표들에 추가되어) 도 24에서 보이는 것과 같은 동일 위치 픽처로부터 서브 CU 레벨 모션 정보 (모션 벡터들 및 참조 인덱스들)을 획득한다. 도 24의 예는 모션 시프트가 블록 A1의 모션으로 설정되었음을 가정한다. 그 다음으로, 각각의 서브 CU에 대해, 상기 동일 위치 픽처 내에 있는 그것의 해당 블록(가운데 샘플을 덮는 가장 작은 모션 그리드)의 모션 정보를 사용하여 상기 서브 CU에 대한 모션 정보를 유도한다. 상기 동일 위치 서브 CU의 모션 정보가 식별된 후, 그것은 현재 서브 CU의 모션 벡터들 및 참조 인덱스들로 변환되는데, 이 방식은 시간적 모션 스케일링이 현재 CU의 참조 픽처들에 맞춰 시간적 모션 벡터들의 참조 픽처들을 정렬하도록 적용되는 HEVC의 TMVP 절차와 유사하다.
 - [0239] VTM에서 SbTMVP 후보 및 어파인 머지 후보들을 모두 포함하는 결합 서브블록 기반 머지 리스트는 서브 블록 기반 머지 모드의 시그널링을 위해 사용된다. SbTMVP 모드는 시퀀스 파라미터 세트(SPS) 플래그에 의해 활성화/비활성화된다. 만약 SbTMVP 모드가 활성화되면, SbTMVP 예측자는 서브 블록 기반 머지 후보들의 리스트의 첫 번째 항목으로 추가되며, 그 다음으로 어파인 머지 후보들이 따른다. 서브 블록 기반 머지 리스트의 크기는 SPS에서 시그널링되고, 서브 블록 기반 머지 리스트의 허용된 최대 크기는 VTM4에서 5이다.

[0240] SbtMVP에서 사용되는 서브 CU 크기는 8x8로 고정되며, 어파인 머지 모드에 대해서와 같이, SbtMVP 모드는 너비와 높이가 모두 8 이상인 CU에만 적용 가능하다.

[0241] 추가적인 SbtMVP 머지 후보의 인코딩 로직은 다른 머지 후보들에 대한 것과 같다. 즉, P 또는 B 슬라이스에서 각 CU에 대해, 추가적인 RD 체크가 수행되어 SbtMVP 후보를 사용할지 여부를 결정한다.

[0242] 도 23 내지 24는 VVC에서 SbtMVP 절차의 예를 보여준다.

[0243] 서브 블록 머지 후보 리스트에서 후보들의 최대 수는 MaxNumSubblockMergeCand로 표시된다.

[0244] **2.3.4.1.3. 서브블록 머지 리스트와 관련된 선택스/시멘틱스**

[0245] **7.3.2.3. Sequence parameter set RBSP syntax**

표 3

seq_parameter_set_rbsp() {	설명자(Descriptor)
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)
profile_tier_level(sps_max_sub_layers_minus1)	
gra_enabled_flag	u(1)
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1);	
i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
long_term_ref_pics_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
qtbtt_dual_tree_intra_flag	u(1)

log2_ctu_size_minus2	ue(v)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slices != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if (sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
sps_pcm_enabled_flag	u(1)
if(sps_pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)
log2_min_pcm_luma_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_luma_coding_block_size	ue(v)
pcm_loop_filter_disabled_flag	u(1)
}	
if((CtbSizeY / MinCbSizeY + 1) <= (pic_width_in_luma_samples / MinCbSizeY - 1)) {	
sps_ref_wraparound_enabled_flag	u(1)
if(sps_ref_wraparound_enabled_flag)	
sps_ref_wraparound_offset_minus1	ue(v)
}	
<i>sps_temporal_mvp_enabled_flag</i>	<i>u(1)</i>
<i>if(sps_temporal_mvp_enabled_flag)</i>	
<i>sps_sbtmvp_enabled_flag</i>	<i>u(1)</i>
<i>sps_amvr_enabled_flag</i>	<i>u(1)</i>
sps_bdof_enabled_flag	u(1)
sps_smvd_enabled_flag	u(1)
<i>sps_affine_amvr_enabled_flag</i>	<i>u(1)</i>
sps_dmvr_enabled_flag	u(1)
sps_mmvd_enabled_flag	u(1)
sps_cclm_enabled_flag	u(1)
if(sps_cclm_enabled_flag && chroma_format_idc == 1)	
sps_cclm_colocated_chroma_flag	u(1)
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	

<code>sps_explicit_mts_intra_enabled_flag</code>	u(1)
<code>sps_explicit_mts_inter_enabled_flag</code>	u(1)
<code>}</code>	
<code>sps_sbt_enabled_flag</code>	u(1)
<code>if(sps_sbt_enabled_flag)</code>	
<code>sps_sbt_max_size_64_flag</code>	u(1)
<code>sps_affine_enabled_flag</code>	u(1)
<code>if(sps_affine_enabled_flag)</code>	
<code>sps_affine_type_flag</code>	u(1)
<code>sps_bcw_enabled_flag</code>	u(1)
<code>sps_ibc_enabled_flag</code>	u(1)
<code>sps_ciip_enabled_flag</code>	u(1)
<code>if(sps_mmvd_enabled_flag)</code>	
<code>sps_fpel_mmvd_enabled_flag</code>	u(1)
<code>sps_triangle_enabled_flag</code>	u(1)
<code>sps_lmcs_enabled_flag</code>	u(1)
<code>sps_ladf_enabled_flag</code>	u(1)
<code>if (sps_ladf_enabled_flag) {</code>	
<code>sps_num_ladf_intervals_minus2</code>	u(2)
<code>sps_ladf_lowest_interval_qp_offset</code>	se(v)
<code>for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {</code>	
<code>sps_ladf_qp_offset[i]</code>	se(v)
<code>sps_ladf_delta_threshold_minus1[i]</code>	ue(v)
<code>}</code>	
<code>}</code>	
<code>timing_info_present_flag</code>	u(1)
<code>if(timing_info_present_flag) {</code>	
<code>num_units_in_tick</code>	u(32)
<code>time_scale</code>	u(32)
<code>hrd_parameters_present_flag</code>	u(1)
<code>if(hrd_parameters_present_flag)</code>	
<code>hrd_parameters(sps_max_sub_layers_minus1)</code>	
<code>}</code>	
<code>vui_parameters_present_flag</code>	u(1)
<code>if(vui_parameters_present_flag)</code>	
<code>vui_parameters()</code>	
<code>sps_extension_flag</code>	u(1)
<code>if(sps_extension_flag)</code>	
<code>while(more_rbsp_data())</code>	
<code>sps_extension_data_flag</code>	u(1)
<code>rbsp_trailing_bits()</code>	
<code>}</code>	

[0247] 7.3.5.1. General slice header syntax

㉞ 4

<code>slice_header() {</code>	설명자(Descriptor)
<code>slice_pic_parameter_set_id</code>	ue(v)
<code>if(rect_slice_flag NumBricksInPic > 1)</code>	
<code>slice_address</code>	u(v)
<code>if(!rect_slice_flag && !single_brick_per_slice_flag)</code>	
<code>num_bricks_in_slice_minus1</code>	ue(v)

slice_type	ue(v)
if(NalUnitType == GRA_NUT)	
recovery_poc_cnt	se(v)
slice_pic_order_cnt_lsb	u(v)
if(NalUnitType == IDR_W_RADL NalUnitType == IDR_N_LP NalUnitType == CRA_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((NalUnitType != IDR_W_RADL && NalUnitType != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (
i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i	
])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++	
) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]	
])	
slice_poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplsIdx[0	
]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[
1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplsIdx[i]]	
> 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)

slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
if (slice_type != I) {	
if(sps_temporal_mvp_enabled_flag)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
}	
if((weighted_pred_flag && slice_type == P)	
(weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2)	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
} else if (sps_ibc_enabled_flag)	
six_minus_max_num_merge_cand	ue(v)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
num_alf_aps_ids	tb(v)
for(i = 0; i < num_alf_aps_ids; i++)	
slice_alf_aps_id_luma[i]	u(5)
slice_alf_chroma_idc	tu(v)
if(slice_alf_chroma_idc && (slice_type != I	
num_alf_aps_ids != 1))	
slice_alf_aps_id_chroma	u(5)
}	
}	
}	
}	
dep_quant_enabled_flag	u(1)

if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	
slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(5)
if(!(qtbt_dual_tree_intra_flag && slice_type == I))	
slice_chroma_residual_scale_flag	u(1)
}	
if (entropy_coding_sync_enabled_flag)	
num_entry_point_offsets	ue(v)
if(NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
byte_alignment()	
}	

[0249] 7.3.7.7. Merge data syntax

표 5

[0250]

	설명자(Descriptor)
merge_data(x0, y0, cbWidth, cbHeight) {	
if (CuPredMode[x0][y0] == MODE_IBC) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag cbWidth * cbHeight != 32)	
regular_merge_flag[x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] == 1){	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag && cbWidth * cbHeight != 32)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	

- [0267] - availableFlagA₁, refIdxLXA₁, predFlagLXA₁ 및 mvLXA₁의 유도를 위해, 아래와 같이 적용된다.
- [0268] - 주변 휘도 코딩 블록 내의 휘도 위치(xNbA₁, yNbA₁)는 (xCb - 1, yCb + cbHeight *? *1)와 같게 설정된다.
- [0269] - 조항 6.4.X [Ed. (BB): 주변 블록 가용성 검토 절차 tbd]에 명시된 바와 같은, 블록에 대한 가용성 유도 과정은 (xCb, yCb)과 같게 설정된 현재 휘도 위치 (xCurr, yCurr) 및 주변 휘도 위치 (xNbA₁, yNbA₁)를 입력으로 하여 적용되며, 출력은 블록 가용성 플래그 availableA₁에 할당된다.
- [0270] - 변수들 availableFlagA₁, refIdxLXA₁, predFlagLXA₁ 및 mvLXA₁은 다음과 같이 유도된다.
- [0271] - 만약 availableA₁이 FALSE와 같으면, availableFlagA₁이 0과 같게 설정되고, mvLXA₁의 양 성분들이 모두 0과 같게 설정되고, refIdxLXA₁은 -1과 같게 설정되고 그리고 predFlagLXA₁은 0과 같게 설정되며, 여기서 X는 0 또는 1이고 bcwIdxA₁은 0과 같게 설정된다.
- [0272] - 그 외의 경우, availableFlagA₁은 1과 같게 설정되고, 다음과 같은 할당이 이루어진다.
- [0273]
$$mvLXA_1 = MvLX[xNbA_1][yNbA_1] \quad (8-485)$$
- [0274]
$$refIdxLXA_1 = RefIdxLX[xNbA_1][yNbA_1] \quad (8-486)$$
- [0275]
$$predFlagLXA_1 = PredFlagLX[xNbA_1][yNbA_1] \quad (8-487)$$
- [0276] - 조항 8.5.5.3에 명시된 바와 같은, 서브 블록기반 시간적 머지 후보들에 대한 유도 과정은 휘도 위치 (xCb, yCb), 휘도 코딩 블록 너비 cbWidth, 휘도 코딩 블록 높이 cbHeight, 가용성 플래그 availableFlagA₁, 참조 인덱스 refIdxLXA₁, 예측 리스트 활용 플래그 predFlagLXA₁, 그리고 모션 벡터 mvLXA₁를 입력들로 하고 가용성 플래그 availableFlagSbCol, 휘도 코딩 서브 블록들의 수(수평 방향: numSbX, 수직 방향: numSbY), 참조 인덱스들 refIdxLXSbCol, 휘도 모션 벡터들 mvLXSbCol[xSbIdx][ySbIdx] 그리고 예측 리스트 활용 플래그들 predFlagLXSbCol[xSbIdx][ySbIdx]을 출력으로 하여 적용되며, 이때 xSbIdx = 0..numSbX - 1 및 ySbIdx=0..numSbY-1이고 X는 0 또는 1이다.
- [0277] 2. sps_affine_enabled_flag가 1과 같은 경우, 샘플 위치들 (xNbA₀, yNbA₀), (xNbA₁, yNbA₁), (xNbA₂, yNbA₂), (xNbB₀, yNbB₀), (xNbB₁, yNbB₁), (xNbB₂, yNbB₂), (xNbB₃, yNbB₃), 및 변수들 numSbX 및 numSbY는 아래와 같이 유도된다.
- [0278]
$$(xA_0, yA_0) = (xCb - 1, yCb + cbHeight) \quad (8-488)$$
- [0279]
$$(xA_1, yA_1) = (xCb - 1, yCb + cbHeight - 1) \quad (8-489)$$
- [0280]
$$(xA_2, yA_2) = (xCb - 1, yCb) \quad (8-490)$$
- [0281]
$$(xB_0, yB_0) = (xCb + cbWidth , yCb - 1) \quad (8-491)$$
- [0282]
$$(xB_1, yB_1) = (xCb + cbWidth - 1, yCb - 1) \quad (8-492)$$
- [0283]
$$(xB_2, yB_2) = (xCb - 1, yCb - 1) \quad (8-493)$$
- [0284]
$$(xB_3, yB_3) = (xCb, yCb - 1) \quad (8-494)$$
- [0285]
$$numSbX = cbWidth \gg 2 \quad (8-495)$$
- [0286]
$$numSbY = cbHeight \gg 2 \quad (8-496)$$
- [0287] 3. sps_affine_enabled_flag가 1과 같은 경우, 변수 availableFlagA는 FALSE와 같게 설정되고, 아래 단계가 (xNbA₀, yNbA₀)부터 (xNbA₁, yNbA₁)까지 (xNbA_k, yNbA_k)에 대해 적용된다.

- [0288] - 조항 6.4.X [Ed. (BB): 주변 블록 가용성 검토 절차 tbd]에 명시된 바와 같은, 블록에 대한 가용성 유도 과정은 (xCb, yCb)과 함께 설정된 현재 휘도 위치 (xCurr, yCurr) 및 주변 휘도 위치 (xNbA_k, yNbA_k)를 입력으로 하여 적용되며, 출력은 블록 가용성 플래그 availableA_k에 할당된다.
- [0289] - availableA_k가 TRUE와 함께 설정되고 MotionModelIdx[xNbA_k][yNbA_k]가 0보다 크며, availableFlagA가 FALSE와 같이 설정되면, 아래 단계가 적용된다.
- [0290] - 변수 availableFlagA가 TRUE와 함께 설정되면, motionModelIdxA는 MotionModelIdx[xNbA_k][yNbA_k]와 함께 설정되고, (xNb, yNb)는 (CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k])와 함께 설정되고, nbW는 CbWidth[xNbA_k][yNbA_k]와 함께 설정되고, nbH는 CbHeight[xNbA_k][yNbA_k]와 함께 설정되고, numCpMv는 MotionModelIdx[xNbA_k][yNbA_k] + 1과 함께 설정되고, 그리고 bcwIdxA는 BcwIdx[xNbA_k][yNbA_k]와 함께 설정된다.
- [0291] - 0 또는 1로 대체되는 X에 대해, 아래 단계가 적용된다.
- [0292] - PredFlagLX[xNbA_k][yNbA_k]가 1과 같은 경우, 조항 8.5.5.5에 명시된 바와 같은, 주변 블록으로부터 휘도 어파인 제어 포인트 모션 벡터들의 유도 과정은 휘도 코딩 블록 위치 (xCb, yCb), 휘도 코딩 블록 너비 및 높이 (cbWidth, cbHeight), 주변 휘도 코딩 블록 위치 (xNb, yNb), 주변 휘도 코딩 블록 너비 및 높이 (nbW, nbH) 그리고 제어 포인트 모션 벡터들의 수 numCpMv를 입력으로 하고 제어 포인트 모션 벡터 예측자 후보들 cpMvLXA[cpIdx] (cpIdx=0..numCpMv-1)을 출력으로 하여 적용된다.
- [0293] - 다음 할당들이 이루어진다.
- [0294] $predFlagLXA = PredFlagLX[xNbA_k][yNbA_k]$ (8-497)
- [0295] $refIdxLXA = RefIdxLX[xNbA_k][yNbA_k]$ (8-498)
- [0296] 4. sps_affine_enabled_flag가 1과 같으면, 변수 availableFlagB가 FALSE와 함께 설정되고, 아래와 같은 단계가 (xNbB₀, yNbB₀)부터 (xNbB₂, yNbB₂)까지 (xNbB_k, yNbB_k)에 대해 적용된다.
- [0297] - 조항 6.4.X [Ed. (BB): 주변 블록 가용성 검토 절차 tbd]에 명시된 바와 같은, 블록에 대한 가용성 유도 과정은 (xCb, yCb)과 함께 설정된 현재 휘도 위치 (xCurr, yCurr) 및 주변 휘도 위치 (xNbB_k, yNbB_k)를 입력으로 하여 적용되며, 출력은 블록 가용성 플래그 availableB_k에 할당된다.
- [0298] - availableB_k가 TRUE와 함께 설정되고 MotionModelIdx[xNbB_k][yNbB_k]가 0보다 크며, availableFlagB가 FALSE와 함께 설정되면, 아래 단계가 적용된다.
- [0299] - 변수 availableFlagB가 TRUE와 함께 설정되고, motionModelIdxB는 MotionModelIdx[xNbB_k][yNbB_k]와 함께 설정되고, (xNb, yNb)는 (CbPosX[xNbB_k][yNbB_k], CbPosY[xNbB_k][yNbB_k])와 함께 설정되고, nbW는 CbWidth[xNbB_k][yNbB_k]와 함께 설정되고, nbH는 CbHeight[xNbB_k][yNbB_k]와 함께 설정되고, numCpMv는 MotionModelIdx[xNbB_k][yNbB_k] + 1과 함께 설정되고, 그리고 bcwIdxB는 BcwIdx[xNbB_k][yNbB_k]와 함께 설정된다.
- [0300] - 0 또는 1로 대체되는 X에 대해, 아래 단계가 적용된다.
- [0301] - PredFlagLX[xNbB_k][yNbB_k]가 TRUE와 함께 설정되면, 조항 8.5.5.5에 명시된 바와 같은, 주변 블록으로부터 휘도 어파인 제어 포인트 모션 벡터들의 유도 과정은 휘도 코딩 블록 위치 (xCb, yCb), 휘도 코딩 블록 너비 및 높이 (cbWidth, cbHeight), 주변 휘도 코딩 블록 위치 (xNb, yNb), 주변 휘도 코딩 블록 너비 및 높이 (nbW, nbH) 그리고 제어 포인트 모션 벡터들의 수 numCpMv를 입력으로 제어 포인트 모션 벡터 후보들 cpMvLXB[cpIdx] (cpIdx = 0 .. numCpMv - 1)을 출력으로 하여 적용된다.
- [0302] - 다음 할당들이 이루어진다.

- [0303] $predFlagLXB = PredFlagLX[xNbB_k][yNbB_k]$ (8-499)
- [0304] $refIdxLXB = RefIdxLX[xNbB_k][yNbB_k]$ (8-500)
- [0305] 5. $sps_affine_enabled_flag$ 가 1과 같으면, 조항 8.5.5.6에 명시된 바와 같은, 구성된 어파인 제어 포인트 모션 벡터 머지 후보들에 대한 유도 과정은 휘도 코딩 블록 위치 (xCb , yCb), 휘도 코딩 블록 너비 및 높이 ($cbWidth$, $cbHeight$), 가용성 플래그들 $availableA_0$, $availableA_1$, $availableA_2$, $availableB_0$, $availableB_1$, $availableB_2$, $availableB_3$ 을 입력으로하고 가용성 플래그들 $availableFlagConstK$, 참조 인덱스들 $refIdxLXConstK$, 예측 리스트 활용 플래그들 $predFlagLXConstK$, 모션 모델 인덱스들 $motionModelIdcConstK$, 양-예측 가중치 인덱스들 $bcwIdxConstK$ 및 $cpMvpLXConstK[cpIdx]$ (X 는 0 또는 1, $K = 1..6$, $cpIdx = 0..2$)를 출력으로 하여 적용된다.
- [0306] 6. 초기 서브 블록 머지 후보 리스트 $subblockMergeCandList$ 는 아래와 같이 구성된다.
- [0307] $i = 0$
- [0308] $if(availableFlagSbCol)$
- [0309] $subblockMergeCandList[i++] = SbCol$
- [0310] $if(availableFlagA \&\& i < MaxNumSubblockMergeCand)$
- [0311] $subblockMergeCandList[i++] = A$
- [0312] $if(availableFlagB \&\& i < MaxNumSubblockMergeCand)$
- [0313] $subblockMergeCandList[i++] = B$
- [0314] $if(availableFlagConst1 \&\& i < MaxNumSubblockMergeCand)$
- [0315] $subblockMergeCandList[i++] = Const1$ (8-501)
- [0316] $if(availableFlagConst2 \&\& i < MaxNumSubblockMergeCand)$
- [0317] $subblockMergeCandList[i++] = Const2$
- [0318] $if(availableFlagConst3 \&\& i < MaxNumSubblockMergeCand)$
- [0319] $subblockMergeCandList[i++] = Const3$
- [0320] $if(availableFlagConst4 \&\& i < MaxNumSubblockMergeCand)$
- [0321] $subblockMergeCandList[i++] = Const4$
- [0322] $if(availableFlagConst5 \&\& i < MaxNumSubblockMergeCand)$
- [0323] $subblockMergeCandList[i++] = Const5$
- [0324] $if(availableFlagConst6 \&\& i < MaxNumSubblockMergeCand)$
- [0325] $subblockMergeCandList[i++] = Const6$
- [0326] 7. 변수 $numCurrMergeCand$ 및 $numOrigMergeCand$ 는 $subblockMergeCandList$ 에서의 머지 후보들의 수와 같게 설정된다.
- [0327] 8. $numCurrMergeCand$ 이 $MaxNumSubblockMergeCand$ 미만이면, $numCurrMrgeCand$ 가 $MaxNumSubblockMergeCand$ 와 같게 될 때까지 아래 단계가 반복되며, 이때 $mvZero[0]$ 와 $mvZero[1]$ 는 모두 0과 같다.
- [0328] - 참조 인덱스들, 예측 리스트 활용 플래그들 그리고 $zeroCand_m$ 의 모션 벡터들은 아래와 같이 유도되며, 이때 m 은 $(numCurrMergeCand - numOrigMergeCand)$ 와 같다.
- [0329] $refIdxL0ZeroCand_m = 0$ (8-502)
- [0330] $predFlagL0ZeroCand_m = 1$ (8-503)

- [0331] $cpMvL0ZeroCand_m[0] = mvZero$ (8-504)
- [0332] $cpMvL0ZeroCand_m[1] = mvZero$ (8-505)
- [0333] $cpMvL0ZeroCand_m[2] = mvZero$ (8-506)
- [0334] $refIdxL1ZeroCand_m = (slice_type = = B) ? 0 : -1$ (8-507)
- [0335] $predFlagL1ZeroCand_m = (slice_type = = B) ? 1 : 0$ (8-508)
- [0336] $cpMvL1ZeroCand_m[0] = mvZero$ (8-509)
- [0337] $cpMvL1ZeroCand_m[1] = mvZero$ (8-510)
- [0338] $cpMvL1ZeroCand_m[2] = mvZero$ (8-511)
- [0339] $motionModelIdcZeroCand_m = 1$ (8-512)
- [0340] $bcwIdxZeroCand_m = 0$ (8-513)
- [0341] - m이 (numCurrMergeCand-numOrigMergeCand)와 같은 후보 zeroCand_m은 subblockMergeCandList의 끝에서 추가되며, numCurrMergeCand는 아래와 같이 1씩 증가한다.
- [0342] $subblockMergeCandList[numCurrMergeCand++] = zeroCand_m$ (8-514)
- [0343] 변수들 refIdxL0, refIdxL1, predFlagL0[xSbIdx][ySbIdx], predFlagL1[xSbIdx][ySbIdx], mvL0[xSbIdx][ySbIdx], mvL1[xSbIdx][ySbIdx], mvCL0[xSbIdx][ySbIdx] 및 mvCL1[xSbIdx][ySbIdx](이때, xSbIdx=0..numSbX-1, ySbIdx=0..numSbY-1)는 아래와 같이 유도된다.
- [0344] - 만약 subblockMergeCandList[merge_subblock_idx[xCb][yCb]]가 SbCo1와 같다면, 양-예측 가중치 인덱스 bcwIdx는 0과 같게 설정되고, 아래가 적용된다. 이때 X는 0 또는 1이다.
- [0345] $refIdxLX = refIdxLXSbCo1$ (8-515)
- [0346] - xSbIdx=0..numSbX-1, ySbIdx=0..numSbY-1에 대해, 아래가 적용된다.
- [0347] $predFlagLX[xSbIdx][ySbIdx] = predFlagLXSbCo1[xSbIdx][ySbIdx]$ (8-516)
- [0348] $mvLX[xSbIdx][ySbIdx][0] = mvLXSbCo1[xSbIdx][ySbIdx][0]$ (8-517)
- [0349] $mvLX[xSbIdx][ySbIdx][1] = mvLXSbCo1[xSbIdx][ySbIdx][1]$ (8-518)
- [0350] - predFlagLX[xSbIdx][ySbIdx]가 1과 같으면, 조항 8.5.2.13의 색차 모션 벡터들에 대한 유도 절차가 mvLX[xSbIdx][ySbIdx] 및 refIdxLX를 입력들로 mvCLX[xSbIdx][ySbIdx]를 출력으로 하여 적용된다.
- [0351] - $x = xCb..xCb + cbWidth - 1$ 및 $y = yCb..yCb + cbHeight - 1$ 에 대해 아래의 할당이 이루어진다.
- [0352] $MotionModelIdc[x][y] = 0$ (8-519)
- [0353] - 그 외의 경우 (subblockMergeCandList[merge_subblock_idx[xCb][yCb]]가 SbCo1와 같지 않으면), 아래가 적용되며, 이때 X는 0 또는 1이다.
- [0354] - 다음 할당들이 이루어지며, 이때 N은 서브 블록 머지 후보 리스트 subblockMergeCandList 상의 merge_subblock_idx[xCb][yCb] 위치에 있는 후보이다(N = subblockMergeCandList[merge_subblock_idx[xCb][yCb]]).
- [0355] $refIdxLX = refIdxLXN$ (8-520)
- [0356] $predFlagLX[0][0] = predFlagLXN$ (8-521)
- [0357] $cpMvLX[0] = cpMvLXN[0]$ (8-522)

[0358] $cpMvLX[1] = cpMvLXN[1]$ (8-523)

[0359] $cpMvLX[2] = cpMvLXN[2]$ (8-524)

[0360] $numCpMv = motionModelIdxN + 1$ (8-525)

[0361] $bcwIdx = bcwIdxN$ (8-526)

[0362] For $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$, the following applies:

[0363] $predFlagLX[xSbIdx][ySbIdx] = predFlagLX[0][0]$ (8-527)

[0364] - $predFlagLX[0][0]$ 가 1과 같으면, 조항 8.5.5.9에 명시된 것과 같은 어파인 제어 포인트 모션 벡터들로부터 모션 벡터 어레이들에 대한 유도 절차가 적용되며, 이때 입력은 휘도 코딩 블록 위치 (xCb , yCb), 휘도 코딩 블록 너비 $cbWidth$, 휘도 예측 블록 높이 $cbHeight$, 제어 포인트 모션 벡터들의 수 $numCpMv$, 제어 포인트 모션 벡터들 $cpMvLX[cpIdx]$ ($cpIdx$ 는 0..2) 그리고 휘도 코딩 서브 블록들의 수 $numSbX$ (수평 방향) 및 $numSbY$ (수직 방향)이고, 출력은 색차 서브 블록 모션 벡터 어레이 $mvCLX[xSbIdx][ySbIdx]$ ($xSbIdx = 0..numSbX - 1$, $ySbIdx = 0 .. numSbY - 1$)이다.

[0365] - $x = xCb..xCb + cbWidth - 1$ and $y = yCb..yCb + cbHeight - 1$ 에 대해 아래의 할당이 이루어진다.

[0366] $MotionModelIdx[x][y] = numCpMv - 1$ (8-528)

[0367] **8.5.5.6 구성된 어파인 제어 포인트 모션 벡터 머지 후보들에 대한 유도 과정**

[0368] 이 과정의 입력들은 다음과 같다.

[0369] - 현재 픽처의 좌측 상단 휘도 샘플에 대한 현재 휘도 코딩 블록의 좌측 상단 샘플을 명시하는 휘도 위치 (xCb , yCb),

[0370] - 현재 휘도 코딩 블록의 너비와 높이를 명시하는 두 변수들 $cbWidth$ 및 $cbHeight$

[0371] - 가용성 플래그들 $availableA_0$, $availableA_1$, $availableA_2$, $availableB_0$, $availableB_1$, $availableB_2$ 및 $availableB_3$,

[0372] - 동일 위치들 ($xNbA_0$, $yNbA_0$), ($xNbA_1$, $yNbA_1$), ($xNbA_2$, $yNbA_2$), ($xNbB_0$, $yNbB_0$), ($xNbB_1$, $yNbB_1$), ($xNbB_2$, $yNbB_2$) 및 ($xNbB_3$, $yNbB_3$).

[0373] 이 과정의 출력은 다음과 같다.

[0374] - 구성된 어파인 제어 포인트 모션 벡터 머지 후보들의 가용성 플래그 $availableFlagConstK$ ($K = 1..6$),

[0375] - 참조 인덱스들 $refIdxLXConstK$ ($K = 1..6$, X 는 0 또는 1),

[0376] - 예측 리스트 활용 플래그들 $predFlagLXConstK$ ($K = 1..6$, X 는 0 또는 1),

[0377] - 어파인 모션 모델 인덱스들 $motionModelIdxConstK$ ($K = 1..6$),

[0378] - 양-예측 가중치 인덱스들 $bcwIdxConstK$ ($K = 1..6$),

[0379] - 구성된 어파인 제어 포인트 모션 벡터들 $cpMvLXConstK[cpIdx]$ ($cpIdx = 0..2$, $K = 1..6$ 그리고 X 는 0 또는 1).

[0380] 제1 (좌측 상부) 제어 포인트 모션 벡터 $cpMvLXCorner[0]$, 참조 인덱스 $refIdxLXCorner[0]$, 예측 리스트 활용 플래그 $predFlagLXCorner[0]$, 양-예측 가중치 인덱스 $bcwIdxCorner[0]$ 및 가용성 플래그 $availableFlagCorner[0]$ 는 아래와 같이 유도된다. 이때, X 는 0과 1이다.

[0381] - 가용성 플래그 $availableFlagCorner[0]$ 는 FALSE와 같게 설정된다.

[0382] - TL을 B_2 , B_3 및 A_2 로 대체하여, ($xNbTL$, $yNbTL$)에 대해 아래가 적용된다.

[0383] - $availableTL$ 가 TRUE와 같고 $availableFlagCorner[0]$ 가 FALSE와 같으면, 아래가 적용된다. 이때, X 는 0과 1이다.

- [0384] $\text{refIdxLXCorner}[0] = \text{RefIdxLX}[xNbTL][yNbTL]$ (8-572)
- [0385] $\text{predFlagLXCorner}[0] = \text{PredFlagLX}[xNbTL][yNbTL]$ (8-573)
- [0386] $\text{cpMvLXCorner}[0] = \text{MvLX}[xNbTL][yNbTL]$ (8-574)
- [0387] $\text{bcwIdxCorner}[0] = \text{BcwIdx}[xNbTL][yNbTL]$ (8-575)
- [0388] $\text{availableFlagCorner}[0] = \text{TRUE}$ (8-576)
- [0389] 제2 (우측 상부) 제어 포인트 모션 벡터 $\text{cpMvLXCorner}[1]$, 참조 인덱스 $\text{refIdxLXCorner}[1]$, 예측 리스트 활용 플래그 $\text{predFlagLXCorner}[1]$, 양-예측 가중치 인덱스 $\text{bcwIdxCorner}[1]$ 및 가용성 플래그 $\text{availableFlagCorner}[1]$ 는 아래와 같이 유도된다. 이때, X는 0과 1이다.
- [0390] - 가용성 플래그 $\text{availableFlagCorner}[1]$ 는 FALSE와 같게 설정된다.
- [0391] - TL을 B_1 , 및 B_0 로 대체하여, $(xNbTR, yNbTR)$ 에 대해 아래가 적용된다.
- [0392] - availableTR 이 TRUE와 같고 $\text{availableFlagCorner}[1]$ 이 FALSE와 같으면, 아래가 적용된다. 이때, X는 0과 1이다.
- [0393] $\text{refIdxLXCorner}[1] = \text{RefIdxLX}[xNbTR][yNbTR]$ (8-577)
- [0394] $\text{predFlagLXCorner}[1] = \text{PredFlagLX}[xNbTR][yNbTR]$ (8-578)
- [0395] $\text{cpMvLXCorner}[1] = \text{MvLX}[xNbTR][yNbTR]$ (8-579)
- [0396] $\text{bcwIdxCorner}[1] = \text{BcwIdx}[xNbTR][yNbTR]$ (8-580)
- [0397] $\text{availableFlagCorner}[1] = \text{TRUE}$ (8-581)
- [0398] 제3 (좌측 하단) 제어 포인트 모션 벡터 $\text{cpMvLXCorner}[2]$, 참조 인덱스 $\text{refIdxLXCorner}[2]$, 예측 리스트 활용 플래그 $\text{predFlagLXCorner}[2]$, 양-예측 가중치 인덱스 $\text{bcwIdxCorner}[2]$ 및 가용성 플래그 $\text{availableFlagCorner}[2]$ 는 아래와 같이 유도된다. 이때, X는 0과 1이다.
- [0399] - 가용성 플래그 $\text{availableFlagCorner}[2]$ 는 FALSE와 같게 설정된다.
- [0400] - BL을 A_1 , 및 A_0 로 대체하여, $(xNbBL, yNbBL)$ 에 대해 아래가 적용된다.
- [0401] - availableBL 이 TRUE와 같고 $\text{availableFlagCorner}[2]$ 가 FALSE와 같으면, 아래가 적용된다. 이때, X는 0과 1이다.
- [0402] $\text{refIdxLXCorner}[2] = \text{RefIdxLX}[xNbBL][yNbBL]$ (8-582)
- [0403] $\text{predFlagLXCorner}[2] = \text{PredFlagLX}[xNbBL][yNbBL]$ (8-583)
- [0404] $\text{cpMvLXCorner}[2] = \text{MvLX}[xNbBL][yNbBL]$ (8-584)
- [0405] $\text{bcwIdxCorner}[2] = \text{BcwIdx}[xNbBL][yNbBL]$ (8-585)
- [0406] $\text{availableFlagCorner}[2] = \text{TRUE}$ (8-586)
- [0407] 제4 (동일 위치 우측 하단) 제어 포인트 모션 벡터 $\text{cpMvLXCorner}[3]$, 참조 인덱스 $\text{refIdxLXCorner}[3]$, 예측 리스트 활용 플래그 $\text{predFlagLXCorner}[3]$, 양-예측 가중치 인덱스 $\text{bcwIdxCorner}[3]$ 및 가용성 플래그 $\text{availableFlagCorner}[3]$ 은 아래와 같이 유도된다. 이때, X는 0과 1이다.
- [0408] - 시간적 머지 후보에 대한 참조 인덱스들 $\text{refIdxLXCorner}[3]$ 은 0과 같게 설정되며, 이때 X는 0 또는 1이다.
- [0409] - 변수들 mvLXCol 및 $\text{availableFlagLXCol}$ 은 아래와 같이 유도된다. X는 0 또는 1이다.
- [0410] - 만약 $\text{slice_temporal_mvp_enabled_flag}$ 가 0과 같으면, both components of mvLXCol 의 양 구성요소들이 모두 0과 같게 설정되고 $\text{availableFlagLXCol}$ 가 0과 같게 설정된다.
- [0411] - 그렇지 않은 경우($\text{slice_temporal_mvp_enabled_flag}$ 가 1과 같은 경우), 아래가 적용된다.

- [0438] - refIdxLXCorner[0]가 refIdxLXCorner[2]과 같다.
- [0439] - 그렇지 않은 경우, availableFlagLX는 FALSE와 같게 설정된다.
- [0440] - availableFlagLX가 TRUE와 같으면, 아래 할당들이 이루어진다.
- [0441] predFlagLXConst1 = 1 (8-597)
- [0442] refIdxLXConst1 = refIdxLXCorner[0] (8-598)
- [0443] cpMvLXConst1[0] = cpMvLXCorner[0] (8-599)
- [0444] cpMvLXConst1[1] = cpMvLXCorner[1] (8-600)
- [0445] cpMvLXConst1[2] = cpMvLXCorner[2] (8-601)
- [0446] - 양-예측 가중치 인덱스 bcwIdxConst1은 아래와 같이 유도된다.
- [0447] - 만약 availableFlagL0가 1과 같고 availableFlagL1이 1과 같으면, 조항 8.5.5.10에 명시된 것
과 같은 구성된 어파인 제어 포인트 모션 벡터 머지 후보들에 대한 양-예측 가중치 인덱스를 위한 유도 과정은
양-예측 가중치 인덱스들 bcwIdxCorner[0], bcwIdxCorner[1] 및 bcwIdxCorner[2]를 입력들로하여 적용되며, 출
력은 양-예측 가중치 인덱스 bcwIdxConst1으로 할당된다.
- [0448] - 그렇지 않은 경우, 상기 양-예측 가중치 인덱스 bcwIdxConst1은 0과 같게 설정된다.
- [0449] - 변수들 availableFlagConst1 및 motionModelIdcConst1은 아래와 같이 유도된다.
- [0450] - 만약 availableFlagL0 또는 availableFlagL1이 1과 같으면, availableFlagConst1은 TRUE와 같
게 설정되고, motionModelIdcConst1은 2로 설정된다.
- [0451] - 그렇지 않은 경우, availableFlagConst1은 FALSE와 같게 설정되고, motionModelIdcConst1은 0
으로 설정된다.
- [0452] 2. availableFlagCorner[0]이 TRUE와 같고 availableFlagCorner[1]이 TRUE와 같고, 그리고
availableFlagCorner[3]가 TRUE와 같으면, 아래 단계가 적용된다.
- [0453] - 0 또는 1로 대체되는 X에 대해, 아래와 같이 적용된다.
- [0454] - 변수 availableFlagLX는 아래와 같이 유도된다.
- [0455] - 만약 아래 조건들이 모두 TRUE이면, availableFlagLX는 TRUE와 같게 설정된다.
- [0456] - predFlagLXCorner[0]가 1과 같다.
- [0457] - predFlagLXCorner[1]이 1과 같다.
- [0458] - predFlagLXCorner[3]가 1과 같다.
- [0459] - refIdxLXCorner[0]가 refIdxLXCorner[1]과 같다.
- [0460] - refIdxLXCorner[0]가 refIdxLXCorner[3]와 같다.
- [0461] - 그렇지 않은 경우, availableFlagLX는 FALSE와 같게 설정된다.
- [0462] - availableFlagLX가 TRUE와 같으면, 다음 할당들이 이루어진다.
- [0463] predFlagLXConst2 = 1 (8-602)
- [0464] refIdxLXConst2 = refIdxLXCorner[0] (8-603)
- [0465] cpMvLXConst2[0] = cpMvLXCorner[0] (8-604)
- [0466] cpMvLXConst2[1] = cpMvLXCorner[1] (8-605)
- [0467] cpMvLXConst2[2] = cpMvLXCorner[3] + cpMvLXCorner[0] - cpMvLXCorner[1]
(8-606)

- [0468] $cpMvLXConst2[2][0] = Clip3(-2^{17}, 2^{17} - 1, cpMvLXConst2[2][0])$
- [0469] (8-607)
- [0470] $cpMvLXConst2[2][1] = Clip3(-2^{17}, 2^{17} - 1, cpMvLXConst2[2][1])$ (8-608)
- [0471] - 양-예측 가중치 인덱스 bcwIdxConst2는 아래와 같이 유도된다.
- [0472] - 만약 availableFlagL0가 1과 같고 availableFlagL1이 1과 같으면, 조항 8.5.5.10에 명시된 것과 같은 구성된 어파인 제어 포인트 모션 벡터 머지 후보들에 대한 양-예측 가중치 인덱스를 위한 유도 과정은 양-예측 가중치 인덱스들 bcwIdxCorner[0], bcwIdxCorner[1] 및 bcwIdxCorner[3]를 입력들로 하여 적용되며, 출력은 양-예측 가중치 인덱스 bcwIdxConst2로 할당된다.
- [0473] - 그렇지 않은 경우, 상기 양-예측 가중치 인덱스 bcwIdxConst2는 0과 같게 설정된다.
- [0474] - 변수들 availableFlagConst2 및 motionModelIdcConst2는 아래와 같이 유도된다.
- [0475] - 만약 availableFlagL0 또는 availableFlagL1이 1과 같으면, availableFlagConst2는 TRUE와 같게 설정되고, motionModelIdcConst2는 2와 같게 설정된다.
- [0476] - 그렇지 않은 경우, availableFlagConst2는 FALSE와 같게 설정되고, motionModelIdcConst2는 0와 같게 설정된다.
- [0477] 3. availableFlagCorner[0]가 TRUE와 같고, availableFlagCorner[2]가 TRUE와 같고 그리고 availableFlagCorner[3]가 TRUE와 같으면, 아래 단계가 적용된다.
- [0478] - 0 또는 1로 대체되는 X에 대해, 아래와 같이 적용된다.
- [0479] - 변수 availableFlagLX는 아래와 같이 유도된다.
- [0480] - 만약 아래 조건들이 모두 TRUE이면, availableFlagLX는 TRUE와 같게 설정된다.
- [0481] - predFlagLXCorner[0]가 1과 같다.
- [0482] - predFlagLXCorner[2]가 1과 같다.
- [0483] - predFlagLXCorner[3]가 1과 같다.
- [0484] - refIdxLXCorner[0]가 refIdxLXCorner[2]와 같다.
- [0485] - refIdxLXCorner[0]가 refIdxLXCorner[3]와 같다.
- [0486] - 그렇지 않은 경우, availableFlagLX는 FALSE와 같게 설정된다.
- [0487] - availableFlagLX가 TRUE와 같으면, 다음 할당들이 이루어진다.
- [0488] $predFlagLXConst3 = 1$ (8-609)
- [0489] $refIdxLXConst3 = refIdxLXCorner[0]$ (8-610)
- [0490] $cpMvLXConst3[0] = cpMvLXCorner[0]$ (8-611)
- [0491] $cpMvLXConst3[1] = cpMvLXCorner[3] + cpMvLXCorner[0] - cpMvLXCorner[2]$
(8-612)
- [0492] $cpMvLXConst3[1][0] = Clip3(-2^{17}, 2^{17} - 1, cpMvLXConst3[1][0])$ (8-613)
- [0493] $cpMvLXConst3[1][1] = Clip3(-2^{17}, 2^{17} - 1, cpMvLXConst3[1][1])$ (8-614)
- [0494] $cpMvLXConst3[2] = cpMvLXCorner[2]$ (8-615)
- [0495] - 양-예측 가중치 인덱스 bcwIdxConst3는 아래와 같이 유도된다.
- [0496] - 만약 availableFlagL0가 1과 같고 availableFlagL1이 1과 같으면, 조항 8.5.5.10에 명시된 것과 같은 구성된 어파인 제어 포인트 모션 벡터 머지 후보들에 대한 양-예측 가중치 인덱스를 위한 유도 과정은

양-예측 가중치 인덱스들 bcwIdxCorner[0], bcwIdxCorner[2] 및 bcwIdxCorner[3]를 입력들로 하여 적용되며, 출력은 양-예측 가중치 인덱스 bcwIdxConst3로 할당된다.

- [0497] - 그렇지 않은 경우, 상기 양-예측 가중치 인덱스 bcwIdxConst3는 0과 같게 설정된다.
- [0498] - 변수들 availableFlagConst3 및 motionModelIdcConst3는 다음과 같이 유도된다.
- [0499] - availableFlagL0 또는 availableFlagL1이 1과 같으면, availableFlagConst3는 TRUE와 같게 설정되고, motionModelIdcConst3는 2와 같게 설정된다.
- [0500] - 그렇지 않은 경우, availableFlagConst3는 FALSE와 같게 설정되고 motionModelIdcConst3는 0과 같게 설정된다.
- [0501] 4. availableFlagCorner[1]이 TRUE와 같고, availableFlagCorner[2]가 TRUE와 같고, 그리고 availableFlagCorner[3]가 TRUE와 같으면, 아래 단계가 적용된다.
- [0502] - 0 또는 1로 대체되는 X에 대해, 아래와 같이 적용된다.
- [0503] - 변수 availableFlagLX는 아래와 같이 유도된다.
- [0504] - 만약 아래 조건들이 모두 TRUE이면, availableFlagLX는 TRUE와 같게 설정된다.
- [0505] - predFlagLXCorner[1]이 1과 같다.
- [0506] - predFlagLXCorner[2]가 1과 같다.
- [0507] - predFlagLXCorner[3]가 1과 같다.
- [0508] - refIdxLXCorner[1]이 refIdxLXCorner[2]와 같다.
- [0509] - refIdxLXCorner[1]이 refIdxLXCorner[3]와 같다.
- [0510] - 그렇지 않은 경우, availableFlagLX는 FALSE와 같게 설정된다.
- [0511] - availableFlagLX이 TRUE와 같으면, 다음 할당들이 이루어진다.
- [0512] predFlagLXConst4 = 1 (8-616)
- [0513] refIdxLXConst4 = refIdxLXCorner[1] (8-617)
- [0514] cpMvLXConst4[0] = cpMvLXCorner[1] + cpMvLXCorner[2] - cpMvLXCorner[3]
(8-618)
- [0515] cpMvLXConst4[0][0] = Clip3(-2¹⁷, 2¹⁷ - 1, cpMvLXConst4[0][0]) (8-619)
- [0516] cpMvLXConst4[0][1] = Clip3(-2¹⁷, 2¹⁷ - 1, cpMvLXConst4[0][1]) (8-620)
- [0517] cpMvLXConst4[1] = cpMvLXCorner[1] (8-621)
- [0518] cpMvLXConst4[2] = cpMvLXCorner[2] (8-622)
- [0519] - 양-예측 가중치 인덱스 bcwIdxConst4는 다음과 같이 유도된다.
- [0520] - 만약 availableFlagL0가 1과 같고 availableFlagL1이 1과 같으면, 조항 8.5.5.10에 명시된 것과 같은 구성된 어과인 제어 포인트 모션 벡터 머지 후보들에 대한 양-예측 가중치 인덱스를 위한 유도 과정은 양-예측 가중치 인덱스들 bcwIdxCorner[1], bcwIdxCorner[2] 및 bcwIdxCorner[3]를 입력들로 하여 적용되며, 출력은 양-예측 가중치 인덱스 bcwIdxConst4로 할당된다.
- [0521] - 그렇지 않은 경우, 상기 양-예측 가중치 인덱스 bcwIdxConst4는 0과 같게 설정된다.
- [0522] - 변수들 availableFlagConst4 및 motionModelIdcConst4는 다음과 같이 유도된다.
- [0523] - 만약 availableFlagL0 또는 availableFlagL1이 1과 같으면, availableFlagConst4는 TRUE와 같게 설정되고, motionModelIdcConst4는 2와 같게 설정된다.
- [0524] - 그렇지 않은 경우, availableFlagConst4는 FALSE와 같게 설정되고, motionModelIdcConst4는 0

과 함께 설정된다.

- [0525] 가변성 플래그들 availableFlagConstK, 참조 인덱스들 refIdxLXConstK, 예측 리스트 활용 플래그들 predFlagLXConstK, 어파인 모션 모델 인덱스들 motionModelIdcConstK 및 구성된 어파인 제어 포인트 모션 벡터들 cpMvLXConstK[cpIdx] (cpIdx = 0..2 및 X는 0 또는 1)을 포함하는 마지막 2개의 구성된 어파인 제어 포인트 모션 벡터 머지 후보들ConstK (K = 5..6)은 다음과 같이 유도된다.
 - [0526] 5. availableFlagCorner[0]가 TRUE와 같고 availableFlagCorner[1]이 TRUE와 같으면, 아래와 같이 적용된다.
 - [0527] - 0 또는 1로 대체되는 X에 대해, 아래와 같이 적용된다.
 - [0528] - 변수 availableFlagLX는 아래와 같이 유도된다.
 - [0529] - 만약 아래 조건들이 모두 TRUE이면, availableFlagLX는 TRUE와 같게 설정된다.
 - [0530] - predFlagLXCorner[0]가 1과 같다.
 - [0531] - predFlagLXCorner[1]이 1과 같다.
 - [0532] - refIdxLXCorner[0]가 refIdxLXCorner[1]과 같다.
 - [0533] - 그렇지 않은 경우, availableFlagLX는 FALSE와 같게 설정된다.
 - [0534] - availableFlagLX가 TRUE와 같으면, 다음 할당들이 이루어진다.
 - [0535] predFlagLXConst5 = 1 (8-623)
 - [0536] refIdxLXConst5 = refIdxLXCorner[0] (8-624)
 - [0537] cpMvLXConst5[0] = cpMvLXCorner[0] (8-625)
 - [0538] cpMvLXConst5[1] = cpMvLXCorner[1] (8-626)
 - [0539] - 양-예측 가중치 인덱스 bcwIdxConst5는 다음과 같이 유도된다.
 - [0540] - 만약 availableFlagL0가 1과 같고 availableFlagL1이 1과 같고 그리고 bcwIdxCorner[0]가 bcwIdxCorner[1]과 같으면, bcwIdxConst5는 bcwIdxCorner[0]와 같게 설정된다.
 - [0541] - 그렇지 않은 경우, 상기 양-예측 가중치 인덱스 bcwIdxConst5는 0과 같게 설정된다.
 - [0542] - 변수들 availableFlagConst5 및 motionModelIdcConst5는 다음과 같이 유도된다.
 - [0543] - 만약 availableFlagL0 또는 availableFlagL1이 1과 같으면, availableFlagConst5는 TRUE와 같게 설정되고, motionModelIdcConst5는 1과 같게 설정된다.
 - [0544] - 그렇지 않은 경우, availableFlagConst5는 FALSE와 같게 설정되고, motionModelIdcConst5는 0과 같게 설정된다.
- [0545] 6. availableFlagCorner[0]가 TRUE와 같고 availableFlagCorner[2]가 TRUE와 같으면, 아래와 같이 적용된다.
 - [0546] - 0 또는 1로 대체되는 X에 대해, 아래와 같이 적용된다.
 - [0547] - 변수 availableFlagLX는 아래와 같이 유도된다.
 - [0548] - 만약 아래 조건들이 모두 TRUE이면, availableFlagLX는 TRUE와 같게 설정된다.
 - [0549] - predFlagLXCorner[0]가 1과 같다.
 - [0550] - predFlagLXCorner[2]가 1과 같다.
 - [0551] - refIdxLXCorner[0]가 refIdxLXCorner[2]와 같다.
 - [0552] - 그렇지 않은 경우, availableFlagLX는 FALSE와 같게 설정된다.
 - [0553] - availableFlagLX가 TRUE와 같으면, 다음이 적용된다.

- [0554] - 제2 제어 포인트 모션 벡터 cpMvLXCorner[1]은 다음과 같이 유도된다.
- [0555]
$$\text{cpMvLXCorner}[1][0] = (\text{cpMvLXCorner}[0][0] \ll 7) +$$
- [0556]
$$((\text{cpMvLXCorner}[2][1] - \text{cpMvLXCorner}[0][1]) \ll (7 + \text{Log2}(\text{cbHeight} / \text{cbWidth}))) \quad (8-627)$$
- [0557]
$$\ll (7 + \text{Log2}(\text{cbHeight} / \text{cbWidth}))$$
- [0558]
$$\text{cpMvLXCorner}[1][1] = (\text{cpMvLXCorner}[0][1] \ll 7) +$$
- [0559]
$$((\text{cpMvLXCorner}[2][0] - \text{cpMvLXCorner}[0][0]) \ll (7 + \text{Log2}(\text{cbHeight} / \text{cbWidth}))) \quad (8-628)$$
- [0560]
$$\ll (7 + \text{Log2}(\text{cbHeight} / \text{cbWidth}))$$
- [0561] - 조항 8.5.2.14에서 명시된 것과 같은 모션 벡터들에 대한 라운딩 절차는 cpMvLXCorner[1]과 함께 설정된 mvX, 7과 함께 설정된 rightShift 그리고 0과 함께 설정된 leftShift를 입력으로 하고 라운딩된 cpMvLXCorner[1]을 출력으로 하여 적용된다.
- [0562] - 다음 할당들이 이루어진다.
- [0563]
$$\text{predFlagLXConst6} = 1 \quad (8-629)$$
- [0564]
$$\text{refIdxLXConst6} = \text{refIdxLXCorner}[0] \quad (8-630)$$
- [0565]
$$\text{cpMvLXConst6}[0] = \text{cpMvLXCorner}[0] \quad (8-631)$$
- [0566]
$$\text{cpMvLXConst6}[1] = \text{cpMvLXCorner}[1] \quad (8-632)$$
- [0567]
$$\text{cpMvLXConst6}[0][0] = \text{Clip3}(-2^{17}, 2^{17} - 1, \text{cpMvLXConst6}[0][0]) \quad (8-633)$$
- [0568]
$$\text{cpMvLXConst6}[0][1] = \text{Clip3}(-2^{17}, 2^{17} - 1, \text{cpMvLXConst6}[0][1]) \quad (8-634)$$
- [0569]
$$\text{cpMvLXConst6}[1][0] = \text{Clip3}(-2^{17}, 2^{17} - 1, \text{cpMvLXConst6}[1][0]) \quad (8-635)$$
- [0570]
$$\text{cpMvLXConst6}[1][1] = \text{Clip3}(-2^{17}, 2^{17} - 1, \text{cpMvLXConst6}[1][1]) \quad (8-636)$$
- [0571] - 양-예측 가중치 인덱스 bcwIdxConst6는 다음과 같이 유도된다.
- [0572] - 만약 availableFlagL0가 1과 같고 availableFlagL1이 1과 같고 그리고 bcwIdxCorner[0]가 bcwIdxCorner[2]와 같으면, bcwIdxConst6는 bcwIdxCorner[0]와 함께 설정된다.
- [0573] - 그렇지 않은 경우, 상기 양-예측 가중치 인덱스 bcwIdxConst6는 0과 함께 설정된다.
- [0574] - 변수들 availableFlagConst6 및 motionModelIdcConst6는 다음과 같이 유도된다.
- [0575] - 만약 availableFlagL0 또는 availableFlagL1이 1과 같으면, availableFlagConst6는 TRUE와 함께 설정되고, motionModelIdcConst6는 1과 함께 설정된다.
- [0576] - 그렇지 않은 경우, availableFlagConst6는 FALSE와 함께 설정되고, motionModelIdcConst6는 0과 함께 설정된다.
- [0577] **2.3.4.2. 정규 머지 리스트**
- [0578] 상기 VVC에서의 머지 리스트 설계와 달리, 히스토리 기반 모션 벡터 예측(HMVP: history-based motion vector prediction)이 이용된다.
- [0579] HMVP에서, 이전에 코딩된 모션 정보는 저장된다. 이전에 코딩된 블록의 모션 정보는 HMVP 후보로 정의된다. 복수의 HMVP 후보들의 HMVP 테이블이라고 불리는 테이블에 저장되며, 이 테이블은 인코딩/디코딩 과정 동안 그 때 그 때 유지된다. 새로운 슬라이스를 코딩/디코딩하기 시작할 때 HMVP 테이블은 비어 있다. 화면간 코딩된 블록이 있을 때 마다, 관련된 모션 정보는 테이블의 마지막 항목에 새로운 HMVP 후보로 추가된다. 전체적인 코딩 흐름은 도 25에 도시된다.
- [0580] HVMP 후보들은 AMVP 및 머지 후보 리스트 구성 과정들 모두에서 사용될 수 있다. 도 26은 변형된 머지 후보 리스트 구성 과정을 도시한다. 머지 후보 리스트가 TMVP 후보 삽입 후에 가득 차 있지 않으면, HMVP 테이블에 저

장된 HMVP 후보들을 활용하여 상기 머지 후보 리스트를 채울 수 있다. 한 블록이 모션 정보의 측면에서 가장 가까운 주변 블록과 더 높은 상관성을 대개의 경우 가지고 있음을 고려하여, 상기 테이블의 HMVP 후보들은 인덱스의 내림 차순으로 삽입된다. 상기 테이블의 마지막 항목은 상기 리스트에 가장 먼저 추가되며, 반면에 첫 번째 항목은 마지막에 추가된다. 마찬가지로, 중복성 제거가 HMVP 후보들에 적용된다. 가용한 머지 후보들의 총 수가 시그널링이 허용된 머지 후보들의 최대 수에 도달하면, 머지 후보 리스트 구성 과정이 종료된다.

[0581] 도 25는 어파인 머지 모드에 대한 후보의 위치를 보여준다.

[0582] 도 26은 변형된 머지 리스트 구성 과정을 보여준다.

[0583] **2.3.5 JVET-N0236**

[0584] 이 기고문은 광학 흐름으로 서브 블록 기반 어파인 모션 보상 예측을 개선하는 방법을 제안한다. 서브 블록 기반 어파인 모션 보상이 수행된 후에, 예측 샘플은 광학 흐름 방정식으로 도출한 차이를 추가하여 개선된다. 이것을 광학 흐름에 의한 예측 개선(Prof: prediction refinement with optical flow)라 부른다. 제안된 방법은 메모리 액세스 대역폭을 증가시키지 않고서 픽셀 레벨 입도에서 화면내 예측을 달성할 수 있다.

[0585] 모션 보상의 더 섬세한 입도를 달성하기 위해, 이 기고문은 광학 흐름으로 서브 블록 기반 어파인 모션 보상 예측을 개선하는 방법을 제안한다. 서브 블록 기반 어파인 모션 보상이 수행된 후에, 휘도 예측 샘플은 광학 흐름 방정식에 의해 도출된 차이를 추가하여 개선된다. 제안된 Prof는 다음과 같은 4 단계들로 기술된다.

[0586] 단계 1) 서브 블록 어파인 모션 보상이 서브 블록 예측 $I(i,j)$ 을 생성하기 위해 수행된다.

[0587] 단계 2) 상기 서브 블록 예측의 공간적 그래디언트 $g_x(i,j)$ 및 $g_y(i,j)$ 는 3-탭 필터 [-1, 0, 1]를 사용하여 각 샘플 위치에서 계산된다.

$$g_x(i,j) = I(i+1,j) - I(i-1,j)$$

$$g_y(i,j) = I(i,j+1) - I(i,j-1)$$

[0588]

[0589] 서브 블록 예측은 그래디언트 계산을 위해 각 측면에서 한 픽셀 씩 연장된다. 메모리 대역폭 및 복잡성을 감소시키기 위해, 연장된 경계들의 픽셀들은 참조 픽처 내의 가장 가까운 정수 픽셀 위치에서부터 복사된다. 따라서 패딩 영역에 대한 추가 보간이 방지된다.

[0590] 단계 3) 휘도 예측 개선이 광학 흐름 방정식에 의해 계산된다.

$$\Delta I(i,j) = g_x(i,j) * \Delta v_x(i,j) + g_y(i,j) * \Delta v_y(i,j)$$

[0591]

[0592] 여기서, 도 27에서 보이는 것과 같이, $\Delta v(i,j)$ 는 $v(i,j)$ 로 표시되는 샘플 위치 (i,j) 에 대해 계산된 픽셀 MV와 픽셀 (i,j) 가 속하는 서브 블록의 서브 블록 MV 사이의 차이이다.

[0593] 어파인 모델 파라미터들 및 서브 블록 중심에 대한 픽셀 위치가 서브 블록 단위에서는 변하지 않기 때문에, $\Delta v(i,j)$ 는 제1 서브 블록에 대해 계산되고 동일 CU 내의 다른 서브 블록들에 대해 재사용될 수 있다. x와 y를 픽셀 위치에서부터 서브 블록의 중심으로의 수평 및 수직 오프셋들이라 하면, $\Delta v(x,y)$ 는 아래 방정식에 의해 도출될 수 있다.

$$\begin{cases} \Delta v_x(x, y) = c * x + d * y \\ \Delta v_y(x, y) = e * x + f * y \end{cases}$$

[0594]

[0595]

4-파라미터 어파인 모델의 경우,

$$\begin{cases} c = f = \frac{v_{1x} - v_{0x}}{w} \\ e = -d = \frac{v_{1y} - v_{0y}}{w} \end{cases}$$

[0596]

[0597]

6-파라미터 어파인 모델의 경우,

$$\begin{cases} c = \frac{v_{1x} - v_{0x}}{w} \\ d = \frac{v_{2x} - v_{0x}}{h} \\ e = \frac{v_{1y} - v_{0y}}{w} \\ f = \frac{v_{2y} - v_{0y}}{h} \end{cases}$$

[0598]

[0599]

여기서, (v_{0x}, v_{0y}) , (v_{1x}, v_{1y}) 및 (v_{2x}, v_{2y}) 는 좌측 상단, 후측 상단 그리고 좌측 하단의 제어 포인트 모션 벡터들이고, w 및 h 는 CU의 너비와 높이이다.

[0600]

단계 4) 마지막으로, 휘도 예측 개선이 서브 블록 예측 $I(i, j)$ 에 추가된다. 최종 예측 I' 는 아래 방정식에 의해 생성된다.

$$I'(i, j) = I(i, j) + \Delta I(i, j)$$

[0601]

[0602]

2.3.6. ATMVP의 개선점들과 관련한 PCT/CN2018/125420 및 PCT/CN2018/116889

[0603]

이 문서들에서, 우리는 ATMVP의 설계를 더 합리적이고 효율적으로 만들기 위한 몇 가지 접근법들을 개시하였으며, 둘 다 전부 참조로서 포함된다.

[0604]

3. 실시예들에 의해 해결되는 문제들의 예

[0605]

VVC의 현재 설계에서, 서브 블록 기반 예측 모드는 다음과 같은 문제들을 가지고 있다.

[0606]

1) SPS에서 어파인 AMVR 플래그는 정규(regular) AMVR이 턴 오프(turned off)될 때 턴 온(turned on)될 수 있다.

[0607]

2) SPS에서 어파인 AMVR 플래그는 어파인 모드가 턴 오프(turned off)될 때 턴 온(turned on)될 수 있다.

[0608]

3) ATMVP가 적용되지 않을 때, MaxNumSubblockMergeCand가 적절하게 설정되지 않는다.

[0609]

4) TMVP가 슬라이스에 대해 비활성화되고 ATMVP가 시퀀스에 대해 활성화될 때, B 슬라이스들에 대한 동일 위치(collocated) 픽처들이 식별되지 않지만, 동일 위치 픽처가 ATMVP 과정에 필요하다.

[0610]

5) TMVP와 ATMVP 모두 참조 픽처들로부터 모션 정보를 가져오기를 요구하며, 이는 현재 설계에서는 차선으

로 동일한 것으로 가정된다.

[0611] 6) PROF는 그것의 on/off를 제어할 플래그를 가져야 한다.

[0612] **4. 예시적 실시예들**

[0613] 아래의 상세한 발명들은 일반 개념들을 설명하는 예들로 간주되어야 한다. 이 발명들은 협소한 방식으로 해석되어서는 안 된다. 더 나아가, 이 발명들은 어떤 방식으로든 결합될 수 있다. 비록 일부 실시예들이 대안적인 시간적 모션 벡터 예측(ATMVP) 기술들의 예들을 사용하여 설명되지만, 대안적인 실시예들에서는 서브 블록 기반 시간적 모션 벡터 예측(sbTMVP) 기술들이 적용될 수 있다.

[0614]아래에서 설명되는 방법들은 모션 후보 리스트들의 다른 유형들(예를 들어, AMVP 후보 리스트)에도 적용될 수 있다.

[0615] 1. 어파인 AMVR에 대한 제어 정보가 시그널링되는지 여부는 어파인 예측이 적용되는지 여부에 의존할 수 있다.

[0616] a) 일 예에서, 어파인 예측이 적용되지 않으면, 어파인 AMVR에 대한 제어 정보는 시그널링되지 않는다.

[0617] b) 일 예에서, 어파인 예측이 적합성(conformance) 비트스트림에서 적용되지 않으면, 어파인 AMVR이 무효화되어야 한다(예를 들어, 어파인 AMVR의 사용법이 거짓(false)으로 시그널링되어야 한다).

[0618] c) 일 예에서, 어파인 예측이 적용되지 않으면, 어파인 AMVR에 대해 시그널링된 제어 정보는 무시될 수 있으며 적용되지 않는 것으로 추론될 수 있다.

[0619] 2. 어파인 AMVR에 대한 제어 정보가 시그널링되는지 여부는 정규(regular) AMVR이 적용되는지 여부에 의존할 수 있다.

[0620] a) 일 예에서, 정규 AMVR이 적용되지 않으면, 어파인 AMVR에 대한 제어 정보는 시그널링되지 않는다.

[0621] b) 일 예에서, 정규 AMVR이 적합성(conformance) 비트스트림에서 적용되지 않으면, 어파인 AMVR이 무효화되어야 한다(예를 들어, 어파인 AMVR의 사용법이 거짓(false)으로 시그널링되어야 한다).

[0622] c) 일 예에서, 정규 AMVR이 적용되지 않으면, 어파인 AMVR에 대해 시그널링된 제어 정보는 무시될 수 있으며 적용되지 않는 것으로 추론될 수 있다.

[0623] d) 일 예에서, 적응적 모션 벡터 해상도의 지시(예를 들어, 플래그)가 시퀀스/픽처/슬라이스/타일 그룹/타일/블록/기타 비디오 유닛들에서 시그널링되어 정규 AMVR(예를 들어, 병진 모션에 적용된 AMVR) 및 어파인 AMVR(예를 들어, 어파인 모션에 적용된 AMVR)과 같은 복수의 코딩 방법들을 위한 AMVR의 사용법을 제어할 수 있다.

[0624] i. 일 예에서, 이러한 지시는 SPS/DPS/VPS/PPS/픽처 헤더/슬라이스 헤더/타일 그룹 헤더에서 시그널링될 수 있다.

[0625] ii. 또는, 더 나아가, 정규 AMVR 및/또는 어파인 AMVR의 사용법의 지시를 시그널링 할 것인지 여부는 상기 지시에 의존할 수 있다.

[0626] 1) 일 예에서, 이러한 지시가 적응적 모션 벡터 해상도가 비활성화되었음을 지시하는 경우, 정규 AMVR의 사용법의 지시를 시그널링하는 단계가 생략될 수 있다.

[0627] 2) 일 예에서, 이러한 지시가 적응적 모션 벡터 해상도가 비활성화되었음을 지시하는 경우, 어파인 AMVR의 사용법의 지시를 시그널링하는 단계가 생략될 수 있다.

[0628] iii. 또는, 더 나아가, 어파인 AMVR의 사용법의 지시를 시그널링 할 것인지 여부는 상기 지시 및 어파인 예측 모드의 사용법에 의존할 수 있다.

[0629] 1) 예를 들어, 어파인 예측 모드가 비활성화되면, 이러한 지시는 생략될 수 있다.

[0630] iv. 일 예에서, 현재 슬라이스/타일 그룹/픽처가 오직 선행 픽처들로부터만 예측

될 수 있으면, 이러한 지시는 시그널링되지 않을 수 있으며, 거짓으로 도출될 수 있다.

- [0631] v. 일 예에서, 현재 슬라이스/타일 그룹/픽처가 오직 후행 픽처들로부터만 예측될 수 있으면, 이러한 지시는 시그널링되지 않을 수 있으며, 거짓으로 도출될 수 있다.
- [0632] vi. 일 예에서, 현재 슬라이스/타일 그룹/픽처가 선행 픽처들 및 후행 픽처들 모두로부터 예측될 수 있으면, 이러한 지시는 시그널링될 수 있다.
- [0633] 3. 어파인 AMVR에 대한 제어 정보가 시그널링되는지 여부는 정규 AMVR이 적용되는지 여부 및 어파인 예측이 적용되는지 여부에 의존할 수 있다.
- [0634] a) 일 예에서, 어파인 예측이 적용되지 않거나 정규 AMVR이 적용되지 않으면, 어파인 AMVR에 대한 제어 정보는 시그널링되지 않는다.
- [0635] i. 일 예에서, 어파인 예측이 적용되지 않거나 정규 AMVR이 적합성 비트스트림에서 적용되지 않으면, 어파인 AMVR이 비활성화되어야 한다(예를 들어, 어파인 AMVR의 사용법이 거짓으로 시그널링되어야 한다).
- [0636] ii. 일 예에서, 어파인 예측이 적용되지 않거나 정규 AMVR이 적용되지 않으면, 어파인 AMVR에 대해 시그널링된 제어 정보는 무시될 수 있으며 적용되지 않는 것으로 추론될 수 있다.
- [0637] b) 일 예에서, 어파인 예측이 적용되지 않으며 정규 AMVR이 적용되지 않으면, 어파인 AMVR에 대한 제어 정보는 시그널링되지 않는다.
- [0638] i. 일 예에서, 어파인 예측이 적용되지 않으며 정규 AMVR이 적합성 비트스트림에서 적용되지 않으면, 어파인 AMVR이 비활성화되어야 한다(예를 들어, 어파인 AMVR의 사용법이 거짓으로 시그널링되어야 한다).
- [0639] ii. 일 예에서, 어파인 예측이 적용되지 않으며 정규 AMVR이 적용되지 않으면, 어파인 AMVR에 대해 시그널링된 제어 정보는 무시될 수 있으며 적용되지 않는 것으로 추론될 수 있다.
- [0640] 4. 서브 블록 머지 후보 리스트에서의 후보들의 최대 수(MaxNumSubblockMergeCand로 표시)는 ATMVP가 활성화되는지 여부에 의존할 수 있다. ATMVP가 활성화되는지 여부는 SPS의 sps_sbtmvp_enabled_flag에 의해서만 지시되지 않을 수 있다.
- [0641] a) 예를 들어, ATMVP가 활성화되는지 여부는 시퀀스 레벨에서 시그널링되는 플래그(예를 들어, SPS에서의 sps_sbtmvp_enabled_flag)에 의존할 수 있다. 그것은 또한 VPS, DPS, APS, PPS, 슬라이스 헤더, 타일 그룹 헤더, 픽처 헤더 등과 같은 시퀀스/픽처/슬라이스/타일 그룹/타일 레벨에서의 기타 다른 비디오 유닛들에서 시그널링되는 하나 또는 복수의 선택스 요소들에도 의존할 수 있다.
- [0642] i. 또는, ATMVP가 활성화되는지 여부는 시그널링 없이 암묵적으로 도출될 수 있다.
- [0643] ii. 예를 들어, TMVP가 픽처 또는 슬라이스 또는 타일 그룹에 대해 활성화되지 않는다면, ATMVP는 픽처 또는 슬라이스 또는 타일 그룹에 대해 활성화되지 않는다.
- [0644] b) 예를 들어, (five_minus_max_num_subblock_merge_cand 와 같이) MaxNumSubblockMergeCand와 관련된 선택스 요소(들)을 시그널링 할지 여부 및/또는 어떻게 시그널링할지는 ATMVP가 활성화되는지 여부에 의존할 수 있다.
- [0645] i. 예를 들어, ATMVP가 활성화되지 않으면, five_minus_max_num_subblock_merge_cand는 적합성 비트스트림에서 제한될 수 있다.
- [0646] 1) 예를 들어, ATMVP가 활성화되지 않으면, five_minus_max_num_subblock_merge_cand가 고정된 수와 같도록 허용되지 않는다. 두 예들에서, 상기 고정된 수는 0 또는 5일 수 있다.
- [0647] 2) 예를 들어, ATMVP가 활성화되지 않으면, five_minus_max_num_subblock_merge_cand가 고정된 수 보다 크도록 허용되지 않는다. 일 예에서, 상기 고정된 수는 4일 수 있다.
- [0648] 3) 예를 들어, ATMVP가 활성화되지 않으면,

five_minus_max_num_subblock_merge_cand가 고정된 수 보다 작도록 허용되지 않는다. 일 예에서, 상기 고정된 수는 1일 수 있다.

- [0649] ii. 예를 들어, five_minus_max_num_subblock_merge_cand는 그것이 존재하지 않을 때 five_minus_max_num_subblock_merge_cand-(ATMVP is enabled ? 0 : 1)로 설정될 수 있으며, 이때 ATMVP가 활성화되는지 여부는 SPS에서의 플래그(예를 들어, sps_sbtmvp_enabled_flag)에만 의존하지는 않는다.
- [0650] c) 예를 들어, MaxNumSubblockMergeCand는 하나 또는 복수의 선택스 요소들(예를 들어, five_minus_max_num_subblock_merge_cand) 및 ATMVP가 활성화되는지 여부에 따라 도출될 수 있다.
- [0651] i. 예를 들어, MaxNumSubblockMergeCand는 아래와 같이 도출될 수 있다.
MaxNumSubblockMergeCand = 5- five_minus_max_num_subblock_merge_cand - (ATMVP is enabled ? 0 : 1).
- [0652] d) ATMVP가 활성화되고, 어파인 모션 예측이 비활성화되면, **MaxNumSubblockMergeCand**가 1로 설정될 수 있다.
- [0653] 5. (병진 및/또는 어파인 모션을 가지는) 디폴트 후보들이 서브 블록 머지 후보 리스트에 추가될 수 있다. 디폴트 후보는 서브 블록 예측과 같은 예측 타입을 가지거나 또는 전체(whole) 블록 예측을 가질 수 있다.
- [0654] a) 일 예에서, 디폴트 후보에 대한 전체 블록 예측은 병진 모션 모델을 따를 수 있다(예를 들어, 정규 머지 후보에 대한 전체 블록 예측).
- [0655] b) 일 예에서, 디폴트 후보에 대한 서브 블록 예측은 병진 모션 모델을 따를 수 있다(예를 들어, ATMVP 후보에 대한 서브 블록 예측).
- [0656] c) 일 예에서, 디폴트 후보에 대한 서브 블록 예측은 어파인 모션 모델을 따를 수 있다(예를 들어, 어파인 머지 후보에 대한 서브 블록 예측).
- [0657] d) 일 예에서, 디폴트 후보는 0과 같은 어파인 플래그를 가질 수 있다.
- [0658] i. 또는, 디폴트 후보는 1과 같은 어파인 플래그를 가질 수 있다.
- [0659] e) 일 예에서, 블록에 대한 후속 절차들은 상기 블록이 디폴트 후보로 코딩되는지 여부에 의존할 수 있다.
- [0660] i. 일 예에서, 상기 블록은 전체 블록 예측으로 코딩된 것으로 취급된다(예를 들어, 선택된 디폴트 후보는 전체 블록 예측을 가진다). 그리고
 - [0661] 1) 예를 들어, PROF는 상기 블록에 적용되지 않을 수 있다.
 - [0662] 2) 예를 들어, DMVR(디코더측 모션 벡터 개선)이 상기 블록에 적용될 수 있다.
 - [0663] 3) 예를 들어, BDOF(양방향 광학 흐름)이 상기 블록에 적용될 수 있다.
 - [0664] 4) 예를 들어, 디블로킹 필터가 상기 블록의 서브 블록들 사이의 경계선들에 적용되지 않을 수 있다.
- [0665] ii. 일 예에서, 상기 블록은 서브 블록 예측으로 코딩된 것으로 취급된다(예를 들어, 선택된 디폴트 후보는 서브 블록 예측을 가진다). 그리고
 - [0666] 1) 예를 들어, PROF가 상기 블록에 적용될 수 있다.
 - [0667] 2) 예를 들어, DMVR(디코더측 모션 벡터 개선)은 상기 블록에 적용되지 않을 수 있다.
 - [0668] 3) 예를 들어, BDOF(양방향 광학 흐름)은 상기 블록에 적용되지 않을 수 있다.
 - [0669] 4) 예를 들어, 디블로킹 필터가 상기 블록의 서브 블록들 사이의 경계선들에 적용될 수 있다.
- [0670] iii. 일 예에서, 상기 블록은 병진 예측으로 코딩된 것으로 취급된다. 그리고
 - [0671] 1) 예를 들어, PROF는 상기 블록에 적용되지 않을 수 있다.

- [0694] b) 일 예에서, ATMVP에 대한 정보는 I 슬라이스 또는 I 타일 그룹 또는 I 픽처에 대해 시그널링되지 않을 수 있다.
- [0695] c) 일 예에서, ATMVP에 대한 정보는 ATMVP가 시퀀스 레벨에서 활성화되도록 시그널링되는 경우(예를 들어, sps_sbtmvp_enabled_flag가 1과 같은 경우)에만 시그널링될 수 있다.
- [0696] d) 일 예에서, TMVP가 슬라이스 또는 타일 그룹 또는 픽처에 대해 비활성화되는 경우, ATMVP에 대한 정보는 상기 슬라이스 또는 상기 타일 그룹 또는 상기 픽처에 대해 시그널링되지 않을 수 있다.
- [0697] i. 예를 들어, ATMVP는 이 경우에 비활성화된 것으로 추론될 수 있다.
- [0698] e) ATMVP의 사용법에 대한 시그널링된 정보와 무관하게 TMV가 슬라이스 (또는 타일 그룹 또는 픽처)에 대해 비활성화되면, ATMVP는 상기 슬라이스 (또는 타일 그룹 또는 픽처)에 대해 비활성화되는 것으로 추론될 수 있다.
- [0699] 7. 서브 블록 기반 시간적 머지 후보들(예를 들어, 시간적 어파인 모션 후보)를 추가할지 여부는 TMVP의 사용법에 의존할 수 있다.
 - [0700] a) 또는, sps_temporal_mvp_enabled_flag의 값에 의존할 수 있다.
 - [0701] b) 또는, slice_temporal_mvp_enabled_flag의 값에 의존할 수 있다.
 - [0702] c) sps_temporal_mvp_enabled_flag 또는 slice_temporal_mvp_enabled_flag가 참인 경우, 서브 블록 기반 시간적 머지 후보들은 서브 블록 머지 후보에 추가될 수 있다.
 - [0703] i. 또는, sps_temporal_mvp_enabled_flag 와 slice_temporal_mvp_enabled_flag가 모두 참인 경우, 서브 블록 기반 시간적 머지 후보들은 서브 블록 머지 후보에 추가될 수 있다.
 - [0704] ii. 또는, sps_temporal_mvp_enabled_flag 또는 slice_temporal_mvp_enabled_flag가 거짓인 경우, 서브 블록 시간적 머지 후보들은 서브 블록 머지 후보에 추가되어서는 안 된다.
 - [0705] d) 또는, 서브 블록 기반 시간적 머지 후보들을 추가하는 것에 대한 지시는 시퀀스/픽처/슬라이스/타일 그룹/타일/브릭/기타 비디오 유닛들에서 시그널링될 수 있다.
 - [0706] i. 또는, 더 나아가, 그것은 시간적 모션 벡터 예측의 사용법에 따라 조건적으로 시그널링될 수 있다(예를 들어, sps_temporal_mvp_enabled_flag 및/또는 slice_temporal_mvp_enabled_flag).
- [0707] 8. 동일 위치 참조 픽처의 지시, 예를 들어, 동일 위치 참조 픽처가 유도된 참조 픽처 리스트 (예를 들어, collocated_from_10_flag) 및/또는 동일 위치 참조 픽처의 참조 인덱스는 시간적 모션 정보에 접근하기를 요청하는 복수의 코딩 방법들의 사용법에 따라 조건적으로 시그널링될 수 있다.
 - [0708] a) 일 예에서, 조건은 ATMVP 및 TMVP 중 하나의 활성화이다.
 - [0709] b) 일 예에서, 조건은 ATMVP, TMVP, 어파인 모션 정보 예측 중 하나의 활성화이다.
- [0710] 9. 일 예에서, ATMVP와 TMVP가 모두 현재 픽처/슬라이스/타일 그룹에 대해 활성화되는 경우에만, 서브 블록 기반 시간적 머지 후보들은 서브 블록 머지 후보 리스트에 포함될 수 있다.
 - [0711] a) 또는, ATMVP가 현재 픽처/슬라이스/타일 그룹에 대해 활성화되는 경우에만, 서브 블록 기반 시간적 머지 후보들이 서브 블록 머지 후보 리스트에 포함될 수 있다.
- [0712] 10. MaxNumSubblockMergeCand는 서브 블록 기반 시간적 머지 후보들이 사용될 수 있는지 여부에 의존할 수 있다.
 - [0713] a) 또는, MaxNumSubblockMergeCand는 TMVP가 사용될 수 있는지 여부에 의존할 수 있다.
 - [0714] b) 예를 들어, 서브 블록 기반 시간적 머지 후보들 (또는 TMVP)가 사용될 수 없는 경우, MaxNumSubblockMergeCand는 4보다 커서는 안 된다.
 - [0715] i. 예를 들어, 서브 블록 기반 시간적 머지 후보들 (또는 TMVP)가 사용될 수 없고 ATMVP가 사용될 수 있는 경우, MaxNumSubblockMergeCand는 4 보다 커서는 안 된다.
 - [0716] ii. 예를 들어, 서브 블록 기반 시간적 머지 후보들 (또는 TMVP)가 사용될 수 없고 ATMVP

도 사용될 수 없는 경우, MaxNumSubblockMergeCand는 4 보다 커서는 안 된다.

[0717] iii. 예를 들어, 서브 블록 기반 시간적 머지 후보들 (또는 TMVP)가 사용될 수 없고 ATMVP가 사용될 수 있는 경우, MaxNumSubblockMergeCand는 3 보다 커서는 안 된다.

[0718] 11. PROF를 수행할지 여부 및/또는 어떻게 수행할지를 지시하기 위한 하나 또는 그 이상의 선택스 요소들은 VPS, DPS, APS, PPS, 슬라이스 헤더, 타일 그룹 헤더, 픽처 헤더, CTU, CU, PU 등과 같은 시퀀스/픽처/슬라이스/타일 그룹/타일/CTU 행/CTU/CU/PU/TU 레벨의 임의의 비디오 유닛들에서 시그널링될 수 있다.

[0719] a) 일 예에서, (PROF가 활성화되는지 여부를 지시하는 플래그와 같은) 상기 하나 또는 그 이상의 선택스 요소들은 어파인 예측이 활성화되는지 여부를 지시하기 위한 선택스 요소(들) 같은 다른 선택스 요소들에 의존하여 조건적으로 시그널링될 수 있다.

[0720] i. 예를 들어, 어파인 예측이 비활성화되면, PROF가 활성화되는지 여부를 지시하기 위한 선택스 요소(들)이 시그널링되지 않을 수 있으며, PROF는 비활성화된 것으로 추론된다.

[0721] b) 일 예에서, 어파인 예측이 적합성 비트스트림에서 비활성화되는 경우, PROF가 활성화되는지 여부를 지시하기 위한 선택스 요소(들)은 PROF가 비활성화되는 것으로 설정되어야 한다.

[0722] c) 일 예에서, 어파인 예측이 적합성 비트스트림에서 비활성화되는 경우, PROF가 활성화되는지 여부를 지시하기 위해 시그널링된 선택스 요소(들)은 무시되고 PROF가 비활성화되는 것으로 추론된다.

[0723] d) 일 예에서, 하나의 선택스 요소는 PROF가 오직 단-예측에만 적용되는지 여부를 지시하기 위해 시그널링될 수 있다.

[0724] **5. 실시예들**

[0725] 이하의 모든 실시예들에 대해, 선택스 요소들은 SPS/PPS/슬라이스 헤더/픽처 헤더/타일 그룹 헤더/타일 또는 기타 비디오 유닛들과 같은 상이한 레벨에서 시그널링될 수 있다.

[0726] **5.1. 실시예#1 : SPS/PPS/Slice 헤더/타일 그룹 헤더에서 sps_affine_amvr_enabled_flag에 관한 선택스 설계의 예들**

표 6

[0727]	sps_amvr_enabled_flag	u(1)
	꺾	
	if(sps_amvr_enabled_flag)	
	sps_affine_amvr_enabled_flag	u(1)

[0728] 또는

표 7

[0729]	sps_affine_enabled_flag	u(1)
	꺾	
	if(sps_affine_enabled_flag)	
	sps_affine_amvr_enabled_flag	u(1)

[0730] 또는

표 8

[0731]	sps_affine_enabled_flag	u(1)
	꺾	
	sps_amvr_enabled_flag	u(1)
	꺾	
	if(sps_affine_enabled_flag && sps_amvr_enabled_flag)	

<i>sps_affine_amvr_enabled_flag</i>	<i>u(1)</i>
-------------------------------------	-------------

[0732] 또는

표 9

<i>sps_amvr_enabled_flag</i>	<i>u(1)</i>
절	
<i>sps_affine_enabled_flag</i>	<i>u(1)</i>
절	
<i>if(sps_affine_enabled_flag && sps_amvr_enabled_flag)</i>	
<i>sps_affine_amvr_enabled_flag</i>	<i>u(1)</i>

[0734] 5.2. 실시예#2 : *five_minus_max_num_subblock_merge_cand* 에 관한 시멘틱의 예들

[0735] *five_minus_max_num_subblock_merge_cand*는 슬라이스에서 지원되는 서브 블록 기반 머지 모션 벡터 예측(MVP) 후보들의 최대수를 5에서 차감한 것을 명시한다. *five_minus_max_num_subblock_merge_cand*가 존재하지 않는 경우, $5 - (\text{sps_sbtmvp_enabled_flag} \ \&\& \ \text{slice_temporal_mvp_enabled_flag})$ 와 같은 것으로 추론된다. 서브 블록 기반 머지 MVP 후보들의 최대 수 *MaxNumSubblockMergeCand*는 다음과 같이 유도된다.

[0736] $\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand}$

[0737] *MaxNumSubblockMergeCand*의 값은 0부터 5까지의 범위(0과 5포함) 내에 있어야 한다.

[0738] 또는, 아래와 같이 적용된다.

[0739] ATMVP가 활성화되는 경우, *MaxNumSubblockMergeCand*의 값은 0에서 1까지의 범위(0과 1 포함) 내에 있어야 한다. 어파인이 활성화되는 경우, *MaxNumSubblockMergeCand*의 값은 0부터 5까지의 범위(0과 5포함) 내에 있어야 한다.

[0740] 5.3. 실시예#3 : 슬라이스 헤더 (또는 타일그룹 헤더) 에서의 ATMVP에 대한 신택스 요소의 예들

표 10

<i>if (slice_type != I) {</i>	
<i>if(sps_temporal_mvp_enabled_flag)</i>	
<i>slice_temporal_mvp_enabled_flag</i>	<i>u(1)</i>
<i>if(slice_type == B)</i>	
<i>mvd_l1_zero_flag</i>	<i>u(1)</i>
<i>if(cabac_init_present_flag)</i>	
<i>cabac_init_flag</i>	<i>u(1)</i>
<i>if(slice_temporal_mvp_enabled_flag) {</i>	
<i>if(slice_type == B)</i>	
<i>collocated_from_10_flag</i>	<i>u(1)</i>
<i>}</i>	
<i>if(sps_sbtmvp_enabled_flag && slice_temporal_mvp_enabled_flag)</i>	
<i>slice_sbtmvp_enabled_flag</i>	<i>u(1)</i>

[0742] 또는

표 11

<i>if (slice_type != I) {</i>	
<i>if(sps_temporal_mvp_enabled_flag)</i>	
<i>slice_temporal_mvp_enabled_flag</i>	<i>u(1)</i>
<i>if(sps_sbtmvp_enabled_flag)</i>	

<i>slice_sbtmvp_enabled_flag</i>	<i>u(1)</i>
<i>if(slice_type == B)</i>	
<i>mvd_l1_zero_flag</i>	<i>u(1)</i>
<i>if(cabac_init_present_flag)</i>	
<i>cabac_init_flag</i>	<i>u(1)</i>
<i>if(slice_temporal_mvp_enabled_flag // slice_sbtmvp_enabled_flag) {</i>	
<i>if(slice_type == B)</i>	
<i>collocated_from_10_flag</i>	<i>u(1)</i>
<i>}</i>	

[0744] 5.4. 실시예#4 : 슬라이스 헤더 (또는 타일그룹 헤더) 에서 서브블록 기반 시간적 머지 후보들에 대한 선택스 요소의 예들

표 12

<i>if (slice_temporal_mvp_enabled_flag && MaxNumSubblockMergeCand > 0)</i>	
<i>sub_block_tmvp_merge_candidate_enabled_flag</i>	<i>u(1)</i>

[0746] *sub_block_tmvp_merge_candidate_enabled_flag*는 서브 블록 기반 시간적 머지 후보들이 사용될 수 있는지 여부를 명시한다.

[0747] 존재하지 않는 경우, *sub_block_tmvp_merge_candidate_enabled_flag*는 0으로 추론된다.

[0748] 8.5.5.6 구성된 어파인 제어 포인트 모션 벡터 머지 후보들에 대한 유도 과정

[0749] ...

[0750] 제4 (동일 위치 우측 하단) 제어 포인트 모션 벡터 *cpMvLXCorner*[3], 참조 인덱스 *refIdxLXCorner*[3], 예측 리스트 활용 플래그 *predFlagLXCorner*[3], 양-예측 가중치 인덱스 *bcwIdxCorner*[3] 및 가용성 플래그 *availableFlagCorner*[3] (X는 0과 1)는 다음과 같이 유도된다.

[0751] - 시간적 머지 후보에 대한 참조 인덱스들 *refIdxLXCorner*[3] (X는 0 또는 1)은 0과 같게 설정된다.

[0752] - 변수들 *mvLXCol* 및 *availableFlagLXCol* (X는 0 또는 1)은 다음과 같이 유도된다.

[0753] - *SLICE_TEMPORAL_MVP_ENABLED_FLAG sub_block_tmvp_merge_candidate_enabled_flag*가 0과 같으면, *mvLXCol*의 양 구성성분들이 모두 0과 같게 설정되고, *availableFlagLXCol*이 0과 같게 설정된다.

[0754] - 그렇지 않은 경우 (*SLICE_TEMPORAL_MVP_ENABLED_FLAG sub_block_tmvp_merge_candidate_enabled_flag*가 1과 같으면), 다음과 같이 적용된다.

[0755] ...

[0756] 5.5. 실시예#5 : SPS 에서 PROF 를 제어하기 위한 선택스/시멘틱스의 예들

표 13

<i>sps_affine_enabled_flag</i>	<i>u(1)</i>
<i>if(sps_affine_enabled_flag)</i>	
<i>sps_prof_flag</i>	<i>u(1)</i>

[0758] *sps_prof_flag*는 PROF가 화면간 예측을 위해 사용될 수 있는지 여부를 명시한다. 만약 *sps_prof_flag*가 0과 같으면, PROF는 적용되지 않는다. 그렇지 않은 경우(*sps_prof_flag*가 1과 같은 경우), PROF는 적용된다. 존재하지 않는 경우, *sps_prof_flag*의 값은 0과 같은 것으로 추론된다.

[0759] 도 28은 비디오 처리 장치(1400)의 블록도이다. 상기 장치(1400)는 여기에 설명된 방법들 중 하나 또는 그 이상을 실시하는 데에 사용될 수 있다. 장치(1400)는 스마트폰, 태블릿, 컴퓨터, 사물인터넷(IoT) 수신기 등에서 구현될 수 있다. 장치(1400)는 하나 또는 그 이상의 프로세서들(1402), 하나 또는 그 이상의 메모리들(1404),

그리고 비디오 처리 하드웨어(1406)을 포함할 수 있다. 프로세서(들)(1402)는 본 문서에서 개시되는 하나 또는 그 이상의 방법들을 실시하도록 구성될 수 있다. 메모리(메모리들)(1404)은 본 명세서에 개시된 방법들 및 기술들을 실시하는 데에 사용되는 데이터 및 코드를 저장하는 데에 사용될 수 있다. 비디오 처리 하드웨어(1406)는 본 문서에서 개시되는 일부 기술들을 하드웨어 회로에서 실시할 수 있다.

- [0760] 도 29는 비디오 처리의 예시적 방법(2900)을 위한 순서도이다. 방법(2900)은, 비디오의 비트스트림 표현이 규칙에 기반하여 어파인 적응적 모션 벡터 해상도 기술과 관련된 제어 정보를 선택적으로 포함하도록, 상기 어파인 적응적 모션 벡터 해상도 기술을 사용하여 상기 비디오의 현재 비디오 블록과 상기 비디오의 상기 비트스트림 표현 사이의 변환을 수행하는 단계(2902)를 포함한다.
- [0761] 아래 예시들의 리스트는 다른 문제들 가운데 본 문서에 기술된 기술적 문제들을 다룰 수 있는 실시예들을 제공한다.
- [0762] 1. 비디오 처리 방법으로서, 비디오의 비트스트림 표현이 규칙에 기반하여 어파인 적응적 모션 벡터 해상도 기술과 관련된 제어 정보를 선택적으로 포함하도록, 상기 어파인 적응적 모션 벡터 해상도 기술을 사용하여 상기 비디오의 현재 비디오 블록과 상기 비디오의 상기 비트스트림 표현 사이의 변환을 수행하는 단계를 포함하는, 비디오 처리 방법.
- [0763] 2. 예시 1에 있어서, 상기 규칙은 상기 변환 동안 어파인 예측이 사용되는 경우에 상기 제어 정보를 포함할 것을 그리고 상기 변환 동안 어파인 예측이 사용되지 않는 경우에 상기 제어 정보를 생략할 것을 명시하는, 비디오 처리 방법.
- [0764] 3. 예시 1에 있어서, 상기 규칙은 어파인 예측이 상기 변환 동안 적용되지 않는 경우 상기 변환 동안 적응적 모션 벡터 해상도 (adaptive motion vector resolution) 단계를 사용하는 것을 배제함을 명시하는, 비디오 처리 방법.
- [0765] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 1항(item)에서 제시된다.
- [0766] 4. 예시 1에 있어서, 상기 규칙은 정규(regular) 적응적 모션 벡터 해상도 단계가 상기 변환 동안 사용되는지 여부에 기반하여 상기 제어 정보를 포함하거나 생략함을 명시하는, 비디오 처리 방법.
- [0767] 5. 예시 4에 있어서, 상기 규칙은 상기 정규(regular) 적응적 모션 벡터 해상도 단계가 상기 변환 동안 적용되지 않는 경우 상기 제어 정보가 생략됨을 명시하는, 비디오 처리 방법.
- [0768] 6. 예시 1에 있어서, 상기 제어 정보는 다수의 적응적 모션 벡터 해상도 기술들의 상기 변환 동안의 사용을 지시하는 동일한 필드를 포함하는, 비디오 처리 방법.
- [0769] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 2항에서 제시된다.
- [0770] 7. 예시 1에 있어서, 상기 규칙은 정규 적응적 모션 벡터 해상도 및 어파인 예측이 상기 변환동안 사용되는지 여부에 기반하여 상기 제어 정보를 포함하거나 생략함을 명시하는, 비디오 처리 방법.
- [0771] 8. 예시 7에 있어서, 상기 규칙은 상기 정규 적응적 모션 벡터 해상도 및 어파인 예측이 모두 상기 변환동안 적용되지 않는 경우 상기 제어 정보를 생략함을 명시하는, 비디오 처리 방법.
- [0772] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 3항에서 제시된다.
- [0773] 9. 비디오 처리 방법으로서, 현재 비디오 블록과 비트 스트림 표현 간의 변환 동안, 상기 변환을 위한 서브 블록 머지 후보 리스트를 결정하는 단계를 포함하고, 상기 서브 블록 머지 후보 리스트를 사용하여 상기 변환을 수행하는 단계를 포함하며, 상기 서브 블록 머지 후보 리스트에서 후보들의 최대 수는 대안적인 시간적 모션 벡터 예측(ATMVP)이 상기 변환에 적용되는지 여부에 의존하는, 비디오 처리 방법.
- [0774] 10. 예시 9에 있어서, 상기 비트스트림에서의 필드는 대안적인 시간적 모션 벡터 예측이 상기 변환에 적용되는지 여부를 지시하는, 비디오 처리 방법.
- [0775] 11. 예시 10에 있어서, 상기 필드는 시퀀스 레벨, 비디오 파라미터 레벨, 픽처 파라미터 세트 레벨, 슬라이스 레벨, 타일 그룹 레벨, 또는 픽처 헤더 레벨에 있는, 비디오 처리 방법.
- [0776] 12. 예시 9에 있어서, ATMVP가 상기 변환에 적용되고 어파인 예측이 상기 변환에 대해 비활성화되는 경우, 상기 후보들의 최대 수가 1로 설정되는, 비디오 처리 방법.

- [0777] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 4항에서 제시된다.
- [0778] 13. 비디오 처리 방법으로서, 현재 비디오 블록과 비트스트림 표현 간의 변환 동안 하나 또는 그 이상의 디폴트 머지 후보들을 상기 변환을 위한 서브 블록 머지 후보 리스트에 추가하는 단계를 포함하고, 추가된 하나 또는 그 이상의 디폴트 머지 후보들을 가지는 상기 서브 블록 머지 후보 리스트를 사용하여 상기 변환을 수행하는 단계를 포함하는, 비디오 처리 방법.
- [0779] 14. 예시 13에 있어서, 디폴트 후보는 서브 블록 예측 타입과 관련되는, 비디오 처리 방법.
- [0780] 15. 예시 14에 있어서, 상기 서브 블록 예측 타입이 병진 모션 모델 또는 어과인 모션 모델에 기반한 예측을 포함하는, 비디오 처리 방법.
- [0781] 16. 예시 13에 있어서, 디폴트 후보는 전체 블록 예측 타입과 관련되는, 비디오 처리 방법.
- [0782] 17. 예시 14에 있어서, 상기 전체 블록 예측 타입이 병진 모션 모델 또는 어과인 모션 모델에 기반한 예측을 포함하는, 비디오 처리 방법.
- [0783] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 5항에서 제시된다.
- [0784] 18. 비디오 처리 방법으로서, 비디오의 현재 비디오 블록과 비트 스트림 표현 간의 변환 동안, 대안적인 시간적 모션 벡터 예측(ATMVP)의 상기 변환에 대한 적용가능성을 결정하는 단계를 포함하고, 상기 결정에 기반하여 상기 변환을 수행하는 단계를 포함하고, 상기 비트스트림 표현의 하나 또는 그 이상의 비트들이 상기 결정에 해당하는, 비디오 처리 방법.
- [0785] 19. 예시 18에 있어서, 상기 하나 또는 그 이상의 비트들은 픽처 헤더, 슬라이스 헤더 또는 타일 그룹 헤더에 있는, 비디오 처리 방법.
- [0786] 20. 예시 18 내지 19에 있어서, 상기 변환은 시간적 모션 벡터 예측(TMVP)을 사용하는 상기 비디오의 상기 변환을 위해 사용되는 다른 동일위치(collocated) 픽처와는 상이한 동일위치 픽처를 ATMVP를 위해 사용하는, 비디오 처리 방법.
- [0787] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 6항에서 제시된다.
- [0788] 21. 비디오 처리 방법으로서, 시간적 모션 벡터 예측(TMVP) 단계 또는 대안적인 시간적 모션 벡터 예측(ATMVP) 단계와 관련된 조건에 기반하여 선택적으로 서브 블록 머지 후보 리스트를 구성하는 단계를 포함하고, 상기 서브 블록 머지 후보 리스트에 기반하여 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하는, 비디오 처리 방법.
- [0789] 22. 예시 21에 있어서, 상기 조건은 시퀀스 파라미터 세트 레벨, 슬라이스 레벨, 타일 레벨 또는 브릭 레벨에서 상기 비트스트림 표현 내의 플래그의 존재에 해당하는, 비디오 처리 방법.
- [0790] 23. 예시 21에 있어서, 상기 현재 비디오 블록이 속하는 픽처, 타일, 또는 타일 그룹에 대해 대안적인 모션 벡터 예측과 TMVP 단계가 모두 활성화되는 경우에만, 상기 서브 블록 머지 후보 리스트가 서브 블록 기반 시간적 머지 후보들을 사용하여 구성되는, 비디오 처리 방법.
- [0791] 24. 예시 21에 있어서, 상기 현재 비디오 블록이 속하는 픽처, 타일, 또는 타일 그룹에 대해 ATMVP와 TMVP 단계가 모두 활성화되는 경우에만, 상기 서브 블록 머지 후보 리스트가 서브 블록 기반 시간적 머지 후보들을 사용하여 구성되는, 비디오 처리 방법.
- [0792] 25. 예시 21에 있어서, 상기 현재 비디오 블록이 속하는 픽처, 타일, 또는 타일 그룹에 대해 ATMVP가 활성화되고 TMVP 단계가 비활성화되는 경우에만, 상기 서브 블록 머지 후보 리스트가 서브 블록 기반 시간적 머지 후보들을 사용하여 구성되는, 비디오 처리 방법.
- [0793] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 7항 및 9항에서 제시된다.
- [0794] 26. 예시 21 내지 25에 있어서, 서브 블록 기반 시간적 머지 후보들이 상기 변환 동안 사용되는지 여부에 기초하여 상기 비트스트림 표현 내의 플래그가 포함되거나 생략되는, 비디오 처리 방법.
- [0795] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 10항에서 제시된다.
- [0796] 27. 비디오 처리 방법으로서, 광학 흐름을 사용하는 예측 개선(PROF)을 규칙에 기반하여 선택적으로 사용하여 비디오의 현재 비디오 블록과 상기 비디오의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기

규칙은 (1) 상기 비트스트림 내에 필드의 포함 또는 생략 또는 (2) 어파인 예측이 상기 변환에 적용되는지 여부를 포함하는, 비디오 처리 방법.

- [0797] 28. 예시 27에 있어서, 상기 규칙은 상기 변환에 대한 어파인 예측의 비활성화로 인한 PROF의 비활성화를 명시하는, 비디오 처리 방법.
- [0798] 29. 예시 27에 있어서, 어파인 예측이 비활성화되는 경우, PROF가 상기 변환에 대해 비활성화된다고 추론되는, 비디오 처리 방법.
- [0799] 30. 예시 27에 있어서, 상기 규칙은 상기 비트스트림 표현내의 해당 플래그에 기반하는 단-예측을 위해서만 PROF를 사용할 것을 명시하는, 비디오 처리 방법.
- [0800] 위의 예시들과 관련되는 추가적인 예시들 및 실시예들은 섹션 4의 11항에서 제시된다.
- [0801] 31. 비디오 처리 장치로서, 예시 1 내지 30 가운데 하나 또는 그 이상을 실시하도록 구성된 프로세서를 포함하는, 비디오 처리 장치.
- [0802] 32. 코드를 저장하는 컴퓨터 판독 가능한 매체로서, 상기 코드는 프로세서에 의해 실행되면 상기 프로세서가 예시 1 내지 30 가운데 하나 또는 그 이상에서의 방법을 실시하도록 하는, 컴퓨터 판독 가능한 매체.
- [0803] 본 문서에서 예시들을 열거함에 있어, 용어 "변환(conversion)"은 현재 비디오 블록에 대한 비트스트림 표현의 생성 또는 비트스트림 표현으로부터 현재 비디오 블록을 생성하는 것을 가리킬 수 있다. 비트스트림 표현은 비트들의 인접한 그룹을 표현할 필요가 없으며 헤더 필드들 또는 코딩된 픽셀 값 정보를 나타내는 코드워드들(codewords)에 포함되는 비트들로 분할될 수 있다.
- [0804] 상기 예시들에서, 규칙은 미리 정의되어 인코더들 및 디코더들에 열려질 수 있다.
- [0805] 도 30은 본 명세서에서 개시되는 다양한 기술들이 실시되는 예시적 비디오 처리 시스템(3000)의 블록도이다. 다양한 실시예들은 시스템(3000)의 구성 요소들 일부 또는 전부를 포함할 수 있다. 시스템(3000)은 비디오 콘텐츠를 수신하기 위한 입력(3002)을 포함할 수 있다. 상기 비디오 콘텐츠는 원상태(raw) 또는 압축되지 않은 포맷, 예를 들어, 8 또는 10 비트 다중 성분 픽셀 값들로 또는 압축되거나 인코딩된 포맷으로 수신될 수 있다. 입력(3002)은 네트워크 인터페이스, 주변 버스 인터페이스, 또는 스토리지 인터페이스를 나타낼 수 있다. 네트워크 인터페이스의 예들은 이더넷, 수동 광 네트워크(passive optical network (PON) 등과 같은 유선 인터페이스들 및 와이파어나 셀룰러 인터페이스 같은 무선 인터페이스를 포함한다.
- [0806] 시스템(3000)은 본 문서에서 개시되는 다양한 코딩 또는 인코딩 방법들을 실시할 수 있는 코딩 구성요소(3004)를 포함할 수 있다. 코딩 구성요소(3004)는 입력(3002)으로부터 코딩 구성요소(3004)의 출력까지 비디오의 코딩된 표현을 생성하기 위해 비디오의 평균 비트레이트 감소시킬 수 있다. 따라서 코딩 기술들은 종종 비디오 압축 또는 비디오 트랜스코딩 기술들이라 불린다. 코딩 구성요소(3004)의 출력은, 구성요소(3006)에 의해 표현되는 것처럼, 저장되거나 또는 연결된 통신을 통해 전송된다. 입력(3002)에서 수신된 비디오의 저장되거나 통신된 비트스트림(또는 코딩된) 표현은 구성요소(3008)에 의해 픽셀 값들을 생성하는 데에 사용되거나 또는 디스플레이 인터페이스(3010)로 보내지는 디스플레이 가능한 비디오를 생성하는 데에 사용될 수 있다. 비트스트림 표현으로부터 사용자가 볼 수 있는 비디오를 생성하는 과정을 종종 비디오 압축 해제(decompression)이라 부른다. 더 나아가, 특정한 비디오 처리 작업들은 "코딩" 작업들 또는 도구들이라 불리므로, 상기 코딩 도구들 또는 작업들이 인코더에서 사용되고 코딩의 결과를 되돌리는 상응하는 디코딩 도구들 또는 작업들은 디코더에 의해 수행될 것임을 이해할 수 있을 것이다.
- [0807] 주변 버스 인터페이스 또는 디스플레이 인터페이스의 예들은 범용 직렬 버스(universal serial bus(USB)) 또는 고해상도 멀티미디어 인터페이스 (high definition multimedia interface (HDMI)) 또는 디스플레이포트 등을 포함할 수 있다. 스토리지 인터페이스의 예들은 SATA(직렬 고급 기술 연결), PCI, IDE 인터페이스 등을 포함한다. 본 문서에서 개시되는 기술들은 이동 전화, 노트북, 스마트폰, 또는 디지털 데이터 처리 및/또는 비디오 디스플레이를 실행할 수 있는 기타 디바이스들과 같은 다양한 전자 디바이스들에서 구현될 수 있다.
- [0808] 도 31은 시각 미디어 처리의 예시적 방법을 위한 순서도이다. 이 순서도의 단계들은 본 문서의 섹션 4의 예시적 실시예 1과 연계하여 논의된다. 단계 3102에서, 과정은 어파인 적응적 모션 벡터 해상도(AMVR) 기술을 시각 미디어 데이터에 사용할 것인지 여부를 상기 시각 미디어 데이터의 비트스트림 표현 상의 상기 어파인 AMVR 기술 관련한 제어 정보에 의존하여 결정하며, 이때 상기 제어 정보는 규칙에 기반하여 상기 비트스트림 표현에 포함되거나 배제된다. 단계 3104에서, 과정은 상기 시각 미디어 데이터와 상기 시각 미디어 데이터의 비트스트림

표현 간의 변환을 수행한다.

- [0809] 도 32는 시각 미디어 처리의 예시적 방법을 위한 순서도이다. 이 순서도의 단계들은 본 문서의 섹션 4의 예시적 실시예 4와 연계하여 논의된다. 단계 3202에서, 과정은 서브 블록 기반 시간적 모션 벡터 예측(sbTMVP) 기술이 시각적 미디어 데이터에 적용되는지 여부를 결정한다. 단계 3204에서, 상기 결정에 상응하여, 과정은 상기 시각 미디어 데이터에 포함된 현재 비디오 블록에 대한 서브 블록 머지 후보 리스트를 생성한다. 단계 3206에서, 과정은 상기 서브 블록 머지 후보 리스트를 사용하여 상기 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행한다.
- [0810] 도 33은 시각 미디어 처리의 예시적 방법을 위한 순서도이다. 이 순서도의 단계들은 본 문서의 섹션 4의 예시적 실시예 5와 연계하여 논의된다. 단계 3302에서, 과정은 현재 비디오 블록과 비트스트림 표현 간의 변환 동안 하나 또는 그 이상의 디폴트 머지 후보들을 상기 변환을 위한 서브 블록 머지 후보 리스트에 추가한다. 단계 3304에서, 과정은 추가된 하나 또는 그 이상의 디폴트 머지 후보들을 가지는 상기 서브 블록 머지 후보 리스트를 사용하여 상기 변환을 수행한다.
- [0811] 도 34는 시각 미디어 처리의 예시적 방법을 위한 순서도이다. 이 순서도의 단계들은 본 문서의 섹션 4의 예시적 실시예 6와 연계하여 논의된다. 단계 3402에서, 과정은, 비디오의 현재 비디오 블록과 비트스트림 표현 간의 변환 동안, 서브 블록 기반 시간적 모션 벡터 예측(sbTMVP)의 상기 변환에 대한 적용가능성을 결정하며, 이때 상기 비트스트림 내의 하나 또는 그 이상의 비트들은 상기 결정에 해당한다. 단계 3404에서, 과정은 상기 결정에 기반하여 상기 변환을 수행한다.
- [0812] 도 35는 시각 미디어 처리의 예시적 방법을 위한 순서도이다. 이 순서도의 단계들은 본 문서의 섹션 4의 예시적 실시예 7와 연계하여 논의된다. 단계 3502에서, 과정은 시간적 모션 벡터 예측(TMVP) 단계 또는 서브 블록 기반 시간적 모션 벡터 예측(sbTMVP) 단계와 관련된 조건에 기반하여 선택적으로 서브 블록 머지 후보 리스트를 구성한다. 단계 3504에서, 과정은 상기 서브 블록 머지 후보 리스트에 기반하여 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행한다.
- [0813] 도 36은 시각 미디어 처리의 예시적 방법을 위한 순서도이다. 이 순서도의 단계들은 본 문서의 섹션 4의 예시적 실시예 8와 연계하여 논의된다. 단계 3602에서, 시각 미디어 데이터와 상기 시각 미디어 데이터의 비트스트림 표현 간의 변환 동안, 과정은 상기 시각 미디어 데이터와 관련된 동일 위치 참조 픽처와 관계된 정보를 결정하며, 이때 상기 정보는 코딩 모드 접근 시간적 모션 정보가 활성화되는지 여부에 기반하여 상기 비트스트림에 포함되거나 배제된다. 단계 3604에서, 과정은 지시에 따라 상기 동일 위치 참조 픽처를 결정한다. 단계 3606에서, 단계는 상기 동일 위치 참조 픽처에 기반하여, 상기 시각 미디어 데이터와 상기 시각 미디어 데이터의 비트스트림 표현 간의 변환을 수행한다.
- [0814] 도 37은 시각 미디어 처리의 예시적 방법을 위한 순서도이다. 이 순서도의 단계들은 본 문서의 섹션 4의 예시적 실시예 11와 연계하여 논의된다. 단계 3702에서, 과정은 시각 미디어 데이터의 비디오 블록과 상기 시각 미디어 데이터의 비트스트림 표현 간의 변환을 수행한다. 단계 3704에서, 상기 변환 동안, 과정은 규칙에 기반하여 상기 비트스트림에 포함되거나 배제되는 제어 정보에 의존하여, 광학 흐름을 사용하는 예측 개선(PROF)를 적용할 것인지 여부를 결정한다.
- [0815] 이제 본 문서의 일부 실시예들이 조항 기반 포맷으로 제시된다.
- [0816] A1. 시각 미디어 처리 방법으로서,
- [0817] 어파인 적응적 모션 벡터 해상도(AMVR) 기술을 시각 미디어 데이터에 사용할 것인지 여부를 상기 시각 미디어 데이터의 비트스트림 표현 상의 상기 어파인 AMVR 기술 관련한 제어 정보에 의존하여-이때, 상기 제어 정보는 규칙에 기반하여 상기 비트스트림 표현에 포함되거나 배제된다-결정하는 단계를 포함하고,
- [0818] 상기 시각 미디어 데이터와 상기 시각 미디어 데이터의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0819] A2. 조항 A1에 있어서, 상기 규칙은 아래 (1), (2), (3) 중 적어도 하나에 기반하여 상기 제어 정보가 상기 비트스트림 표현에 포함되거나 배제됨을 명시하는, 시각 미디어 처리 방법.
- [0820] (1) 어파인 예측이 상기 변환 동안 사용되는지 여부
- [0821] (2) 정규 AMVR 기술이 상기 변환 동안 사용되는지 여부

- [0822] (3) 적응적 모션 벡터 해상도와 관련된 정보가 상기 AMVR 기술이 활성화되거나 비활성화됨을 지시하는지 여부
- [0823] A3. 조항 A1에 있어서, 상기 규칙은 어파인 예측이 상기 변환 동안 사용되지 않는 경우 상기 제어 정보를 배제함을 명시하는, 시각 미디어 처리 방법.
- [0824] A4. 조항 A1에 있어서, 상기 규칙은 어파인 예측이 상기 변환 동안 사용되는 경우 상기 제어 정보를 포함함을 명시하는, 시각 미디어 처리 방법.
- [0825] A5. 조항 A1에 있어서, 어파인 예측이 사용되지 않는 경우 상기 어파인 AMVR 기술의 사용이 상기 변환 동안 비활성화되는, 시각 미디어 처리 방법.
- [0826] A6. 조항 A2에 있어서, 적응적 모션 벡터 해상도와 관련된 상기 정보가 다수의 코딩 기술들에 대한 상기 AMVR 기술의 사용을 제어하기 위해 사용되며, 상기 다수의 코딩 기술들은 정규 AMVR 기술 및 어파인 AMVR 기술을 포함하는, 시각 미디어 처리 방법.
- [0827] A7. 조항 A2 또는 A6의 어느 하나 이상에 있어서, 적응적 모션 벡터 해상도와 관련된 상기 정보가 상기 AMVR 기술이 비활성화됨을 지시하는 경우, 상기 규칙은 상기 어파인 AMVR 기술과 관련된 상기 제어 정보를 배제함을 명시하는, 시각 미디어 처리 방법.
- [0828] A8. 조항 A2 또는 A6의 어느 하나 이상에 있어서, 적응적 모션 벡터 해상도와 관련된 상기 정보가 상기 AMVR 기술이 활성화됨을 지시하는 경우, 상기 규칙은 상기 어파인 AMVR 기술과 관련된 상기 제어 정보를 포함함을 명시하는, 시각 미디어 처리 방법.
- [0829] A9. 조항 A2 및 조항 A6 내지 조항 A8의 어느 하나 이상에 있어서, 적응적 모션 벡터 해상도와 관련된 상기 정보가 시퀀스 레벨에서 상기 비트스트림 내에 시그널링되는, 시각 미디어 처리 방법.
- [0830] A10. 조항 A1 내지 조항 A9의 어느 하나 이상에 있어서, 상기 규칙은 정규 AMVR 기술이 상기 변환 동안 사용되는 경우 상기 어파인 AMVR 기술과 관련된 상기 제어 정보를 포함함을 더 명시하는, 시각 미디어 처리 방법.
- [0831] A11. 조항 A1 내지 조항 A9의 어느 하나 이상에 있어서, 상기 규칙은 정규 AMVR 기술이 상기 변환 동안 적용되지 않는 경우 상기 어파인 AMVR 기술과 관련된 상기 제어 정보를 배제함을 더 명시하는, 시각 미디어 처리 방법.
- [0832] A12. 조항 A5에 있어서, 상기 정규 AMVR 기술이 상기 변환 동안 적용되지 않는 경우 상기 어파인 AMVR 기술이 비활성화되는, 시각 미디어 처리 방법.
- [0833] A13. 조항 A1 내지 조항 A4의 어느 하나 이상에 있어서, 상기 제어 정보는 다수의 적응적 모션 벡터 해상도 기술들의 상기 변환 동안의 사용을 지시하는 동일한 필드를 포함하는, 시각 미디어 처리 방법.
- [0834] A14. 조항 A1 내지 조항 A13의 어느 하나 이상에 있어서, 어파인 예측이 상기 변환 동안 사용되지 않는 경우,
상기 어파인 AMVR 기술과 관련된 상기 제어 정보를 무시하는 단계를 더 포함하는, 시각 미디어 처리 방법.
- [0836] A15. 조항 A1 내지 조항 A14의 어느 하나 이상에 있어서, 상기 규칙이 정규 적응적 모션 벡터 해상도 및 어파인 예측이 상기 변환동안 사용되는지 여부에 기반하여 상기 제어 정보를 포함하거나 배제함을 명시하는, 시각 미디어 처리 방법.
- [0837] A16. 조항 A15에 있어서, 상기 규칙이 상기 정규 적응적 모션 벡터 해상도 및 어파인 예측 중 적어도 하나가 상기 변환 동안 적용되지 않는 경우 상기 제어 정보를 배제함을 명시하는, 시각 미디어 처리 방법.
- [0838] A17. 조항 A1 내지 조항 A16의 어느 하나 이상에 있어서, 상기 규칙이 어파인 예측의 사용과 관련된 하나 또는 그 이상의 기준에 기반하여 상기 비트스트림 내에 상기 정보를 포함하거나 배제함을 더 명시하는, 시각 미디어 처리 방법.
- [0839] A18. 조항 A17에 있어서, 상기 규칙이 어파인 예측의 사용이 비활성화되는 경우 상기 비트 스트림 내에서 상기 제어 정보를 배제함을 더 명시하고, 상기 비트스트림 내에 상기 정보의 부재(lack)가 어파인 AVMR의 사용이 비활성화되었음을 추론하는 데에 사용되는, 시각 미디어 처리 방법.
- [0840] A19. 조항 A1 내지 조항 A18의 어느 하나 이상에 있어서, 현재 슬라이스, 타일 그룹, 또는 픽처가 오직 상기 시각 미디어 데이터와 연관된 하나 또는 그 이상의 선행 픽처들로부터만 예측되는 경우, 상기 제어 정보가 상기

비트스트림에서 배제되는, 시각 미디어 처리 방법.

- [0841] A20. 조항 A1 내지 조항 A18의 어느 하나 이상에 있어서, 현재 슬라이스, 타일 그룹, 또는 픽처가 오직 상기 시각 미디어 데이터와 연관된 하나 또는 그 이상의 후행 픽처들로부터만 예측되는 경우, 상기 제어 정보가 상기 비트스트림에서 배제되는, 시각 미디어 처리 방법.
- [0842] A21. 조항 A1 내지 조항 A18의 어느 하나 이상에 있어서, 현재 슬라이스, 타일 그룹, 또는 픽처가 상기 시각 미디어 데이터와 연관된 하나 또는 그 이상의 후행 또는 선행 픽처들로부터 예측되는 경우, 상기 제어 정보가 상기 비트스트림에서 배제되는, 시각 미디어 처리 방법.
- [0843] A22. 조항 A1에 있어서, 어파인 예측 또는 정규 AMVR 기술이 상기 변환 동안 사용되지 않는 경우 상기 어파인 AMVR 기술이 비활성화되는, 시각 미디어 처리 방법.
- [0844] A23. 조항 A22에 있어서, 만약 어파인 AMVR이 비활성화되면, 상기 어파인 AMVR 기술과 관련된 상기 제어 정보가 상기 비트스트림 표현에서 배제되는, 시각 미디어 처리 방법.
- [0845] A24. 조항 A1에 있어서, 상기 규칙은 어파인 예측 또는 상기 정규 AMVR 기술 중 적어도 하나가 상기 변환 동안 적용되지 않는 경우 상기 어파인 AMVR 기술과 관련된 상기 정보를 배제함을 더 명시하는, 시각 미디어 처리 방법.
- [0846] A25. 조항 A1에 있어서, 상기 규칙은 어파인 예측도 상기 정규 AMVR 기술도 상기 변환 동안 적용되지 않는 경우 상기 어파인 AMVR 기술과 관련된 상기 정보를 배제함을 더 명시하는, 시각 미디어 처리 방법.
- [0847] A26. 조항 A1 내지 조항 A25에 있어서, 상기 제어 정보가 현재 비디오 블록 또는 상기 미디어 데이터의 다른 비디오 블록과 연관된 시퀀스, 픽처, 슬라이스, 타일 그룹, 타일 및 브릭 중 하나에 포함되는, 시각 미디어 처리 방법.
- [0848] A27. 조항 26에 있어서, 상기 제어 정보가 시퀀스 파라미터 세트 (SPS: sequence parameter set), 디코딩 파라미터 세트 (DPS: decoding parameter set), 비디오 파라미터 세트 (VPS: video parameter set), 픽처 파라미터 세트 (PPS: picture parameter set), 픽처 헤더, 슬라이스 헤더, 또는 타일 그룹 헤더 중 하나에 포함되는, 시각 미디어 처리 방법.
- [0849] B1. 시각 미디어 처리 방법에 있어서,
- [0850] 서브 블록 기반 시간적 모션 벡터 예측(sbTMVP) 기술이 시각 미디어 데이터에 적용되는지 여부를 결정하는 단계를 포함하고,
- [0851] 상기 결정에 대응하여, 상기 미디어 데이터에 포함된 현재 비디오 블록에 대해 서브 블록 머지 후보 리스트를 생성하는 단계를 포함하고, 그리고
- [0852] 상기 서브 블록 머지 후보 리스트를 사용하여, 상기 현재 비디오 블록과 상기 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0853] B2. 조항 B1에 있어서, 상기 서브 블록 머지 후보 리스트에서 후보들의 최대 수는 상기 sbTMVP 기술이 상기 변환에 적용되는지 여부 또는 시간적 모션 벡터 예측(TMVP) 기술이 상기 변환에 적용되는지 여부 중 적어도 하나에 의존하는, 시각 미디어 처리 방법.
- [0854] B3. 조항 B1에 있어서, 상기 서브 블록 머지 후보 리스트에서 후보들의 최대 수는 서브 블록 기반 시간적 머지 후보들이 사용되는지 여부에 의존하는, 시각 미디어 처리 방법.
- [0855] B4. 조항 B3에 있어서, 시간적 모션 벡터 예측(TMVP) 기술 및 상기 sbTMVP 기술이 상기 변환에 적용되면, 서브 블록 기반 시간적 머지 후보들이 사용되는, 시각 미디어 처리 방법.
- [0856] B5. 조항 B1 및 조항 B2 중 어느 하나 이상에 있어서, 상기 비트스트림 내의 필드가 상기 sbTMVP 기술이 상기 변환에 적용되는지 여부를 분명하게 지시하는, 시각 미디어 처리 방법.
- [0857] B6. 조항 B5에 있어서, 상기 필드가 시퀀스 레벨, 비디오 파라미터 세트 레벨, 픽처 파라미터 세트 레벨, 슬라이스 레벨, 타일 그룹 레벨, 또는 픽처 헤더 레벨에 있는, 시각 미디어 처리 방법.
- [0858] B7. 조항 B1에 있어서,
- [0859] 시간적 모션 벡터 예측(TMVP) 기술이 픽처, 슬라이스, 또는 타일 그룹에 적용되지 않는다는 것을

감지하자마자, 상기 sbTMVP 기술이 상기 픽처, 상기 슬라이스 또는 상기 타일 그룹에 적용되지 않는다는 것을 판단하는 단계를 더 포함하는, 시각 미디어 처리 방법.

- [0860] B8. 조항 B2에 있어서, 상기 서브 블록 머지 후보 리스트의 후보들의 최대 값과 관련된 상기 비트스트림 내의 하나 또는 그 이상의 신택스 요소들이 상기 sbTMVP 기술이 상기 변환에 적용되는지 여부에 의존하는, 시각 미디어 처리 방법.
- [0861] B9. 조항 B8에 있어서,
- [0862] 상기 sbTMVP 기술이 상기 변환에 적용되지 않는다는 것을 판단하자마자, 상기 하나 또는 그 이상의 신택스 요소들의 값을 제한하는 단계를 더 포함하는, 시각 미디어 처리 방법.
- [0863] B10. 조항 B6에 있어서, 상기 서브 블록 머지 후보 리스트 내의 후보들의 최대 수와 관련된 상기 하나 또는 그 이상의 신택스 요소들이 상기 비트스트림 표현 내에 선택적으로 포함되거나 배제되는, 시각 미디어 처리 방법.
- [0864] B11. 조항 B2 내지 조항 B10 중 어느 하나 이상에 있어서, 상기 서브 블록 머지 후보 리스트 내의 후보들의 최대 수가 0, 1 또는 5인, 시각 미디어 처리 방법.
- [0865] C1. 시각 미디어 처리 방법에 있어서,
- [0866] 현재 비디오 블록과 비트스트림 표현 간의 변환 동안 하나 또는 그 이상의 디폴트 머지 후보들을 상기 변환을 위한 서브블록 머지 후보 리스트에 추가하는 단계를 포함하고, 그리고
- [0867] 상기 추가된 하나 또는 그 이상의 디폴트 머지 후보들을 가지는 상기 서브블록 머지 후보 리스트를 사용하여 상기 변환을 수행하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0868] C2. 조항 C1에 있어서, 디폴트 후보는 서브 블록 예측 타입과 연관되는, 시각 미디어 처리 방법.
- [0869] C3. 조항 C2에 있어서, 상기 서브 블록 타입은 병진 모션 모델 또는 어파인 모션 모델에 기반하는 예측을 포함하는, 시각 미디어 처리 방법.
- [0870] C4. 조항 C1에 있어서, 디폴트 후보는 전체 블록 예측 타입과 연관되는, 시각 미디어 처리 방법.
- [0871] C5. 조항 C2에 있어서, 상기 전체 블록 예측 타입이 병진 모션 모델 또는 어파인 모션 모델에 기반하는 예측을 포함하는, 시각 미디어 처리 방법.
- [0872] D1. 시각 미디어 처리 방법에 있어서,
- [0873] 비디오의 현재 비디오 블록과 비트스트림 표현 간의 변환 동안, 서브 블록 기반 시간적 모션 벡터 예측(sbTMVP)의 상기 변환에 대한 적용가능성을 결정하는 단계-이때, 상기 비트스트림 내의 하나 또는 그 이상의 비트들은 상기 결정에 해당한다-를 포함하고, 그리고
- [0874] 상기 결정에 기반하여 상기 변환을 수행하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0875] D2. 조항 D1에 있어서, 상기 하나 또는 그 이상의 비트들은 픽처 헤더, 슬라이스 헤더, 또는 타일 그룹 헤더에 포함되는, 시각 미디어 처리 방법.
- [0876] D3. 조항 D1 내지 조항 D2 중 어느 하나 이상에 있어서, 상기 변환은 시간적 모션 벡터 예측(TMVP)을 사용하는 상기 비디오의 상기 변환을 위해 사용되는 다른 동일위치(collocated) 픽처와는 상이한 동일위치 픽처를 ATMVP를 위해 사용하는, 시각 미디어 처리 방법.
- [0877] D4. 조항 D1 내지 조항 D3 중 어느 하나 이상에 있어서, 상기 하나 또는 그 이상의 비트들은 I 슬라이스, I 타일, 또는 I 픽처에 포함되는, 시각 미디어 처리 방법.
- [0878] D5. 조항 D1 내지 조항 D3 중 어느 하나 이상에 있어서, 상기 하나 또는 그 이상의 비트들은, sbTMVP가 시퀀스 레벨에서 적용되는 경우, 상기 시퀀스 레벨에서 상기 비트스트림 표현에 포함되는, 시각 미디어 처리 방법.
- [0879] D6. 조항 D1 내지 조항 D3 중 어느 하나 이상에 있어서, 상기 하나 또는 그 이상의 비트들은, TMVP가 슬라이스 레벨, 타일 그룹 레벨, 또는 픽처 레벨에 대해 비활성화되면, 상기 슬라이스 레벨, 상기 타일 그룹 레벨, 또는 상기 픽처 레벨에서 상기 비트스트림 표현에서 배제되는, 시각 미디어 처리 방법.

- [0880] D7. 조항 D6에 있어서, sbTMVP가 적용 불가능한 것으로 판단되는, 시각 미디어 처리 방법.
- [0881] D8. 조항 D1에 있어서, TMVP가 슬라이스, 타일 그룹, 또는 픽처에 대해 비활성화되며,
- [0882] 상기 비트스트림에서의 상기 하나 또는 그 이상의 비트들에 상관없이, sbTMVP가 상기 슬라이스, 상기 타일 그룹, 또는 상기 픽처에 대해 비활성화된다고 추론하는 단계를 더 포함하는, 시각 미디어 처리 방법.
- [0883] E1. 시각 미디어 처리 방법에 있어서,
- [0884] 시간적 모션 벡터 예측(TMVP) 단계 또는 서브 블록 기반 시간적 모션 벡터 예측(sbTMVP) 단계와 연관된 조건에 기반하여 선택적으로 서브 블록 머지 후보 리스트를 구성하는 단계, 그리고
- [0885] 상기 서브 블록 머지 후보 리스트에 기반하여 현재 비디오 블록과 상기 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0886] E2. 조항 E1에 있어서, 상기 조건은 시퀀스 파라미터 세트 레벨, 슬라이스 레벨, 타일 레벨 또는 브릭 레벨에서 상기 비트스트림 표현 내의 플래그 및/또는 상기 플래그 값의 존재에 해당하는, 시각 미디어 처리 방법.
- [0887] E3. 조항 E1에 있어서, 상기 현재 비디오 블록이 속하는 픽처, 타일, 타일 그룹, 또는 슬라이스에 대해 sbTMVP 단계와 TMVP 단계가 모두 활성화되는 경우에만, 상기 서브 블록 머지 후보 리스트가 서브 블록 기반 시간적 머지 후보들을 사용하여 구성되는, 시각 미디어 처리 방법.
- [0888] E4. 조항 E1에 있어서, 상기 현재 비디오 블록이 속하는 픽처, 타일, 타일 그룹, 또는 슬라이스에 대해 sbTMVP 단계가 활성화되는 경우에만, 상기 서브 블록 머지 후보 리스트가 서브 블록 기반 시간적 머지 후보들을 사용하여 구성되는, 시각 미디어 처리 방법.
- [0889] E5. 조항 E1에 있어서, 상기 현재 비디오 블록이 속하는 픽처, 타일, 타일 그룹, 또는 슬라이스에 대해 sbTMVP가 활성화되고 TMVP 단계가 비활성화되는 경우에만, 상기 서브 블록 머지 후보 리스트가 서브 블록 기반 시간적 머지 후보들을 사용하여 구성되는, 시각 미디어 처리 방법.
- [0890] E6. 조항 E1 내지 E5 중 어느 하나 이상에 있어서, 서브 블록 기반 시간적 머지 후보들이 상기 변환 동안 사용되는지 여부에 기초하여 상기 비트스트림 표현 내의 플래그가 포함되거나 생략되는, 시각 미디어 처리 방법.
- [0891] E7. 조항 E2에 있어서, 상기 플래그가 sps_temporal_mv_enabled_flag 및 sps_sbtmvp_enabled_flag 중 적어도 하나를 포함하는, 시각 미디어 처리 방법.
- [0892] F1. 시각 미디어 처리 방법에 있어서,
- [0893] 시각 미디어 데이터와 상기 시각 미디어 데이터의 비트스트림 표현 간의 변환 동안, 상기 시각 미디어 데이터와 관련된 동일 위치 참조 픽처와 관계된 정보-이때 상기 정보는 코딩 모드 접근 시간적 모션 정보가 활성화되는지 여부에 기반하여 상기 비트스트림에 포함되거나 배제된다-를 결정하는 단계를 포함하고, 그리고
- [0894] 지시에 따라 상기 동일 위치 참조 픽처를 결정하는 단계를 포함하고, 그리고
- [0895] 상기 동일 위치 참조 픽처에 기반하여, 상기 시각 미디어 데이터와 상기 시각 미디어 데이터의 비트 스트림 간의 변환을 수행하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0896] F2. 조항 F1에 있어서, 상기 정보는 상기 동일위치 참조 픽처 및/또는 상기 동일위치 참조 픽처의 참조 인덱스를 유도하는 데에 사용되는 참조 픽처 리스트를 지시하는, 시각 미디어 처리 방법.
- [0897] F3. 조항 F1에 있어서, 상기 코딩 모드는 시간적 모션 벡터 예측(TMVP) 코딩 기술을 포함하는, 시각 미디어 처리 방법.
- [0898] G1. 시각 미디어 처리 방법에 있어서,
- [0899] 시각 미디어 데이터의 비디오 블록과 상기 시각 미디어 데이터의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 그리고
- [0900] 상기 변환 동안, 규칙에 기반하여 상기 비트스트림에 포함되거나 배제되는 제어 정보에 의존하여, 광학 흐름을 사용하는 예측 개선(PROF)를 적용할 것인지 여부를 결정하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0901] G2. 조항 G1에 있어서, 상기 규칙은 어파인 예측이 상기 변환에 적용되는지 여부를 기반으로 상기

비트스트림 표현 안에 상기 정보를 포함하거나 배제함을 명시하는, 시각 미디어 처리 방법.

- [0902] G3. 조항 G1 내지 조항 G2 중 어느 하나 이상에 있어서, PROF의 적용이 상기 변환에 대한 어파인 예측의 비활성화에 기반하여 비활성화되는, 시각 미디어 처리 방법.
- [0903] G4. 조항 G2에 있어서, 상기 규칙은 어파인 예측이 비활성화되는 경우 상기 제어 정보를 배제함을 명시하는, 시각 미디어 처리 방법.
- [0904] G5. 조항 G4에 있어서, 어파인 예측이 비활성화되며,
- [0905] 상기 변환에 대해 PROF가 비활성화되는 것을 추론하는 단계를 더 포함하는, 시각 미디어 처리 방법.
- [0906] G6. 조항 G2에 있어서, 상기 비트스트림 표현 내의 해당 플래그에 기반하는 단-예측에 대해서만 PROF의 적용이 연관되는, 시각 미디어 처리 방법.
- [0907] G7. 조항 G1 내지 조항 G6에 있어서, 상기 제어 정보가 현재 비디오 블록 또는 상기 시각 미디어 데이터의 다른 비디오 블록과 연관되는 시퀀스, 픽처, 슬라이스, 타일 그룹, 타일, 브릭 중 하나에 포함되는, 시각 미디어 처리 방법.
- [0908] G8. 조항 G7에 있어서, 상기 제어 정보가 시퀀스 파라미터 세트 (SPS), 디코딩 파라미터 세트 (DPS), 비디오 파라미터 세트 (VPS), 픽처 파라미터 세트 (PPS), 픽처 헤더, 슬라이스 헤더, 또는 타일 그룹 헤더 중 하나에 포함되는, 시각 미디어 처리 방법.
- [0909] H1. 조항 A1 내지 조항 G8 중 어느 하나 이상에 있어서, 상기 변환은 상기 현재 비디오 블록으로부터 상기 비트스트림 표현을 생성하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0910] H2. 조항 A1 내지 조항 G8 중 어느 하나 이상에 있어서, 상기 변환은 상기 비트스트림 표현으로부터 상기 현재 비디오 블록의 픽셀 값들을 생성하는 단계를 포함하는, 시각 미디어 처리 방법.
- [0911] H3. 비디오 인코더 장치로서, 조항 A1 내지 조항 G8 중 어느 하나 이상에서의 방법을 실시하기 위해 구성된 프로세서를 포함하는, 비디오 인코더 장치.
- [0912] H4. 비디오 디코더 장치로서, 조항 A1 내지 조항 G8 중 어느 하나 이상에서의 방법을 실시하기 위해 구성된 프로세서를 포함하는, 비디오 디코더 장치.
- [0913] H5. 코드를 저장하는 컴퓨터 판독 가능한 매체로서, 상기 코드는 조항 A1 내지 G8 가운데 하나 이상에서의 방법을 실시하기 위한, 프로세서에 의해 실행 가능한 명령들을 구현하는, 컴퓨터 판독 가능한 매체.
- [0914] 본 문서에서, 용어 "비디오 처리" 또는 "시각 미디어 처리"는 비디오 인코딩, 비디오 디코딩, 비디오 압축 또는 비디오 압축 해제를 가리킬 수 있다. 예를 들어, 비디오 압축 알고리즘들은 비디오의 픽셀 표현으로부터 상응하는 비트스트림 표현으로 변환하거나 그 반대의 경우에 적용될 수 있다. 현재 비디오 블록의 비트스트림 표현은, 예를 들어, 선택의 의해 정의되는 대로, 비트스트림 내에서 같은 위치에 있거나(co-located) 상이한 곳에 분산되어 있는 비트들에 해당할 수 있다. 예를 들어, 매크로블록은 변환되고 코딩된 오류 잔차 값들로 그리고 헤더들 및 비트스트림의 다른 필드들의 비트들을 사용하여 인코딩될 수 있다. 게다가, 변환 동안, 디코더는, 상기 해결수단들에서 설명된 것과 같이, 일부 필드들이 존재하거나 없을 수 있다는 지식으로 상기 판단에 기반하여 비트스트림을 파싱할 수 있다. 유사하게, 인코더는 특정한 선택스 필드들이 포함되거나 포함되지 말아야 한다는 것을 판단할 수 있으며, 그에 따라 상기 선택스 필드들을 코딩된 표현에 포함시키거나 배제시킴으로써 상기 코딩된 표현을 생성할 수 있다. 서브 블록 기반 모션 벡터 개선의 사용을 포함하는 기술들을 사용하여 압출 효율을 향상시키기 위해 개시된 기술들이 비디오 인코더들 또는 디코더들에서 구현될 수 있다는 것을 알 수 있을 것이다.
- [0915] 개시된 그리고 기타 해결수단들, 예들, 실시예들, 모듈들 그리고 본 문서에 설명된 기능적 작업들은 디지털 전자 회로에서, 본 문서에 개시된 구조들 및 그것들의 구조적 등가물들을 포함하는 컴퓨터 소프트웨어, 펌웨어 또는 하드웨어에서 또는 그것들 가운데 하나 또는 그 이상의 조합들에서 구현될 수 있다. 개시된 그리고 기타 실시예들은 하나 또는 그 이상의 컴퓨터 프로그램 제품들로, 즉, 데이터 처리 장치에 의해 실행되거나 상기 데이터 처리 장치의 동작을 제어하기 위한 컴퓨터 판독 가능한 매체에 코딩된 컴퓨터 프로그램 명령들의 하나 또는 그 이상의 모듈들로 구현될 수 있다. 상기 컴퓨터 판독 가능한 매체는 기계 판독 가능한 저장 디바이스, 기계 판독 가능한 저장 기관, 메모리 디바이스, 기계 판독 가능한 전파 신호를 가져오는 물질의 조성, 또는 이들 중 하나 이상의 조합일 수 있다. 용어 "데이터 처리 장치"는 데이터를 처리하기 위한 모든 장치, 디바이스 및 기

계를 포괄하며, 예를 들어, 프로그래머블 프로세서, 컴퓨터, 또는 다중 프로세서 또는 컴퓨터들을 포함한다. 상기 장치는, 하드웨어 외에도, 당해 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드를 포함할 수 있고, 코드는 예를 들어, 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 시스템, 또는 이들 중 하나 이상의 조합을 구성한다. 전파 신호는 인공적으로 생성된 신호, 예를 들어, 기계에 의해 생성된 전기적, 광학적 또는 전자기적 신호일 수 있으며, 이 신호는 적절한 수신 장치로 전송하기 위한 정보를 인코딩 하기 위해 생성된다.

[0916] 컴퓨터 프로그램(프로그램, 소프트웨어, 소프트웨어 애플리케이션, 스크립트 또는 코드로도 알려짐)은 컴파일 또는 인터프리터 언어를 포함하는 모든 형태의 프로그래밍 언어로 작성될 수 있으며, 독립형 프로그램이나 모듈, 컴포넌트, 서브루틴 또는 컴퓨터 환경에서 사용하기에 적합한 그 밖의 유닛을 포함하는 임의의 형태로도 배치될 수 있다. 컴퓨터 프로그램은 파일 시스템의 파일에 반드시 상응해야 하는 것은 아니다. 프로그램은 다른 프로그램 또는 데이터를 보유하는 파일의 일부에 저장되거나(예를 들어, 마크업 언어 문서 내에 저장되는 하나 이상의 스크립트), 당해 프로그램 전용의 단일 파일에 저장되거나, 또는 다수의 조직화된 파일들(예를 들어, 하나 이상의 모듈, 서브프로그램, 코드의 부분을 저장하는 파일)에 저장될 수 있다. 컴퓨터 프로그램은 하나의 컴퓨터에서, 또는 한 위치에 배치되거나 또는 다수의 위치에 걸쳐서 분산되고 통신 네트워크에 의해 접속된 다수의 컴퓨터에서 실행되도록 배치될 수 있다.

[0917] 본 문서에서 설명된 프로세스들 및 논리 흐름들은 하나 이상의 프로그래머블 프로세서에 의해 수행될 수 있고, 이 프로그래머블 프로세서는 입력 데이터에 작용하여 출력을 생성함으로써 기능을 수행하는 하나 이상의 컴퓨터 프로그램들을 실행한다. 상기 프로세서들 및 논리 흐름들은, 예를 들어, FPGA(field programmable gate array) 또는 ASIC(application specific integrated circuit)과 같은 전용 논리 회로에 의해 수행될 수 있고, 장치 또한 전용 논리 회로로서 구현될 수 있다.

[0918] 컴퓨터 프로그램의 실행에 적합한 프로세서들은, 예를 들어, 범용 및 전용 마이크로프로세서들 및 임의 종류의 디지털 컴퓨터의 하나 이상의 프로세서들을 포함한다. 일반적으로, 프로세서는 판독 전용 메모리(ROM), 또는 랜덤 액세스 메모리(RAM), 또는 둘 다로부터 명령어들과 데이터를 수신할 것이다. 컴퓨터의 필수 구성요소는 명령들을 실행하는 프로세서 및 명령들 및 데이터를 저장하는 하나 이상의 메모리 디바이스들이다. 일반적으로, 컴퓨터는 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스, 예를 들어, 자기 디스크, 광자기 디스크, 또는 광디스크를 포함하거나, 또는 이 디바이스와 데이터를 송수신하기 위하여 동작적으로(operatively) 결합될 것이다. 하지만 컴퓨터는 이러한 디바이스를 구비할 필요는 없다. 컴퓨터 프로그램 명령어들과 데이터를 저장하기 적합한 컴퓨터 판독가능 매체에는, 예를 들어, 반도체 메모리 디바이스(예를 들어, EPROM, EEPROM, 플래시 메모리 디바이스), 자기 디스크(예를 들어, 내부 하드디스크, 착탈식 디스크), 광자기 디스크 및 CD-ROM과 DVD-ROM 디스크를 포함하는 모든 형태의 비휘발성 메모리, 매체 및 메모리 디바이스가 포함된다. 프로세서와 메모리는 전용 논리 회로에 의해 보완되거나 또는 전용 논리 회로에 통합될 수 있다.

[0919] 본 특허문서가 다수의 특정한 구현 세부사항을 포함하고 있지만, 이는 임의 발명의 범위나 청구할 사항의 범위에 대한 어떠한 제한으로도 이해되어서는 안 되며, 특정한 기술들의 특정한 실시예들에 고유할 수 있는 특징들의 설명으로서 이해되어야 한다. 별개의 실시예들의 맥락에서 본 특허 문서에서 설명된 소정의 특징들은 조합되어 단일 실시예로 구현될 수도 있다. 반대로, 단일 실시예의 맥락에서 설명한 다양한 특징들은 복수의 실시예에서 별개로 구현되거나 어떤 적당한 하위 조합으로도 구현 가능하다. 또한, 앞에서 특징들이 특정 조합에서 동작하는 것으로서 설명되고 그와 같이 청구되기까지 하였지만, 청구된 조합으로부터의 하나 또는 그 이상의 특징들은 일부 경우에 해당 조합으로부터 삭제될 수 있으며, 청구된 조합은 하위 조합이나 하위 조합의 변형으로 될 수 있다.

[0920] 마찬가지로, 동작들이 도면들에서 특정한 순서로 묘사되고 있지만, 이는 그러한 동작들이 바람직한 결과를 얻기 위해, 도시한 특정 순서나 순차적인 순서로 수행되어야 한다거나, 도시된 모든 동작들이 수행되어야 한다는 것을 의미하지 않는다. 더욱이, 본 특허 문서에 기재된 실시예에 있어서 다양한 시스템 구성요소들의 분리는 모든 실시예에서 그러한 분리를 요구하는 것으로 이해되어서는 안 된다.

[0921] 단지 몇 가지 구현들 및 예시들이 설명되었으며, 기타 구현, 개선 및 변형은 본 특허 문서에 설명되고 도시된 것에 기초하여 이루어질 수 있다.

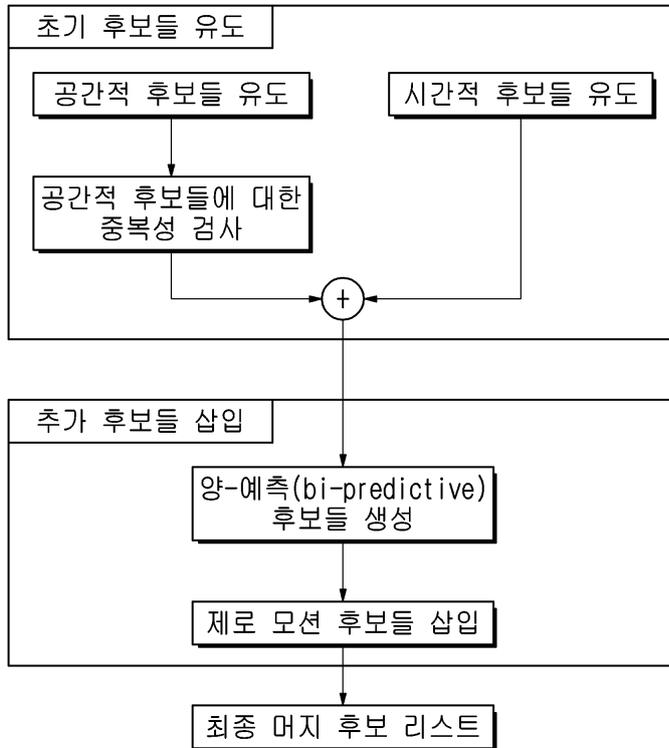
부호의 설명

[0922] 3004 : 코딩 구성요소

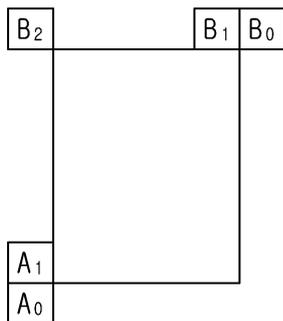
3010 : 인터페이스

도면

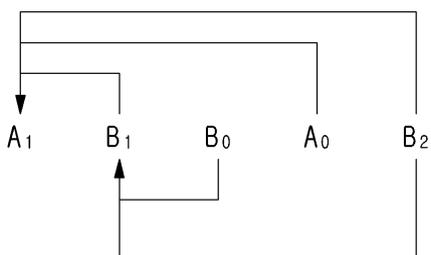
도면1



도면2



도면3

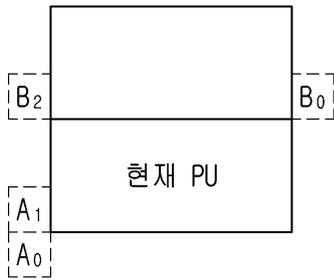


도면4a



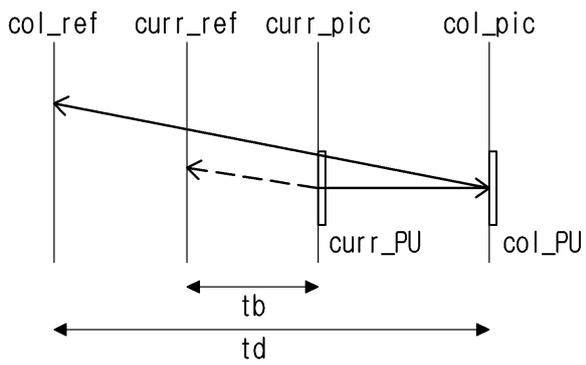
(a) $N \times 2N$ 의 제2 PU

도면4b

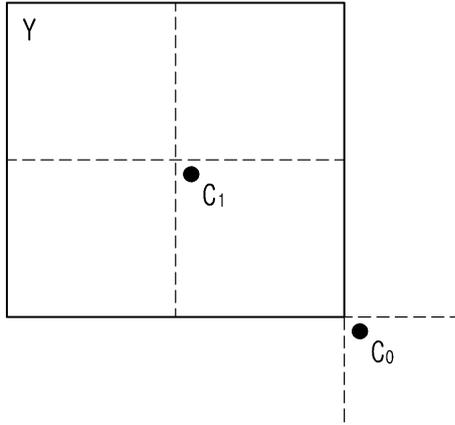


(b) $2N \times N$ 의 제2 PU

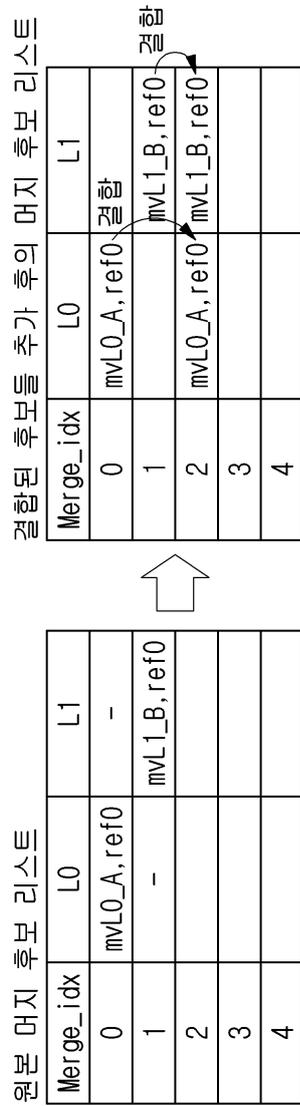
도면5



도면6

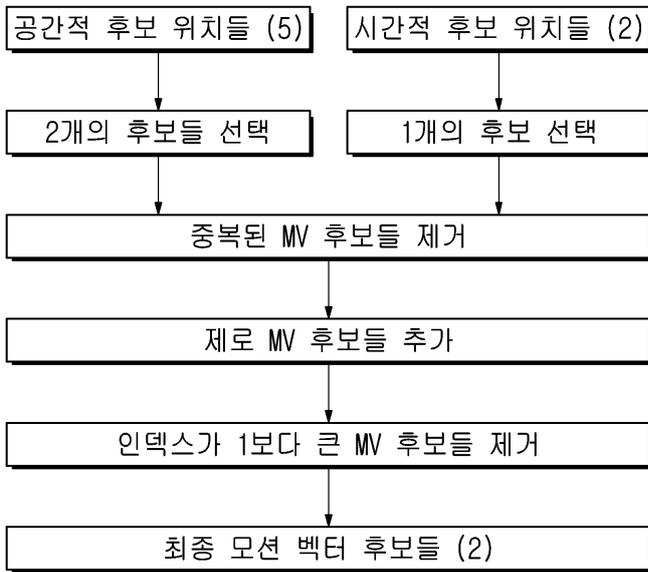


도면7

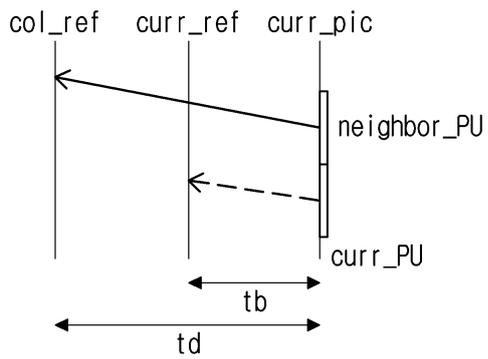


도면8

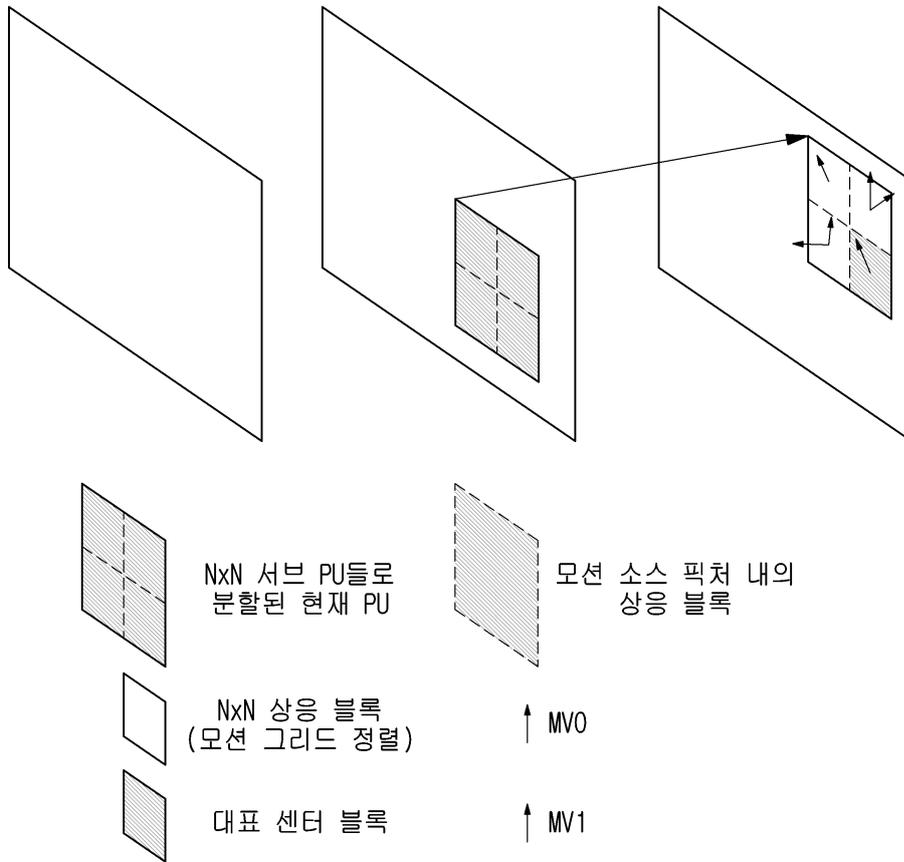
refidx를 입력으로 하는 각 참조 픽처 리스트에 대해



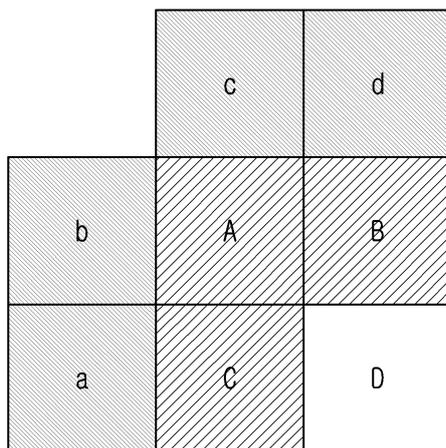
도면9



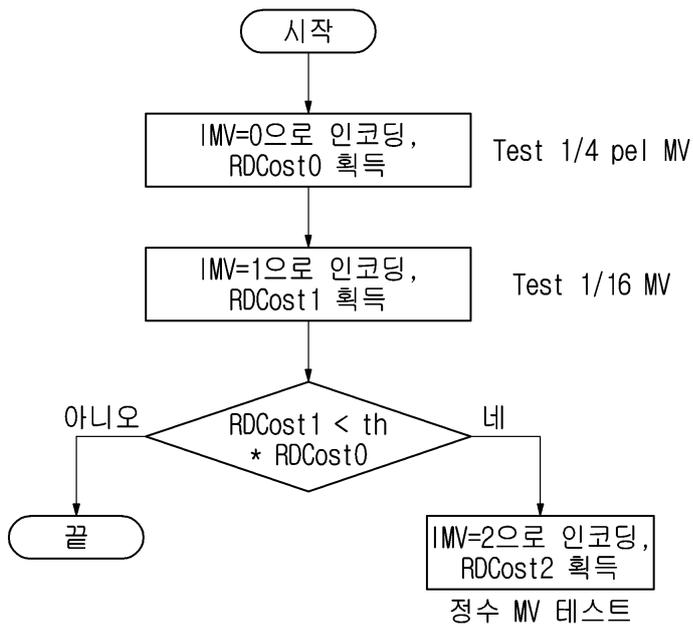
도면10



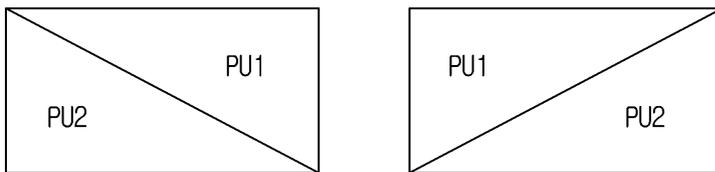
도면11



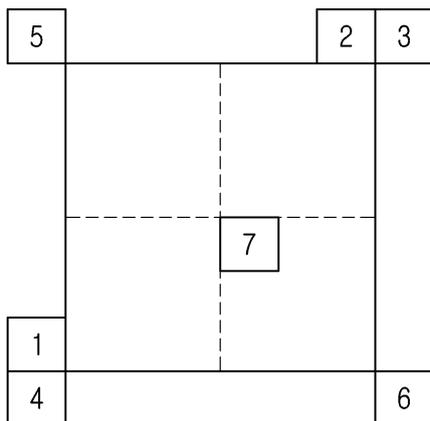
도면12



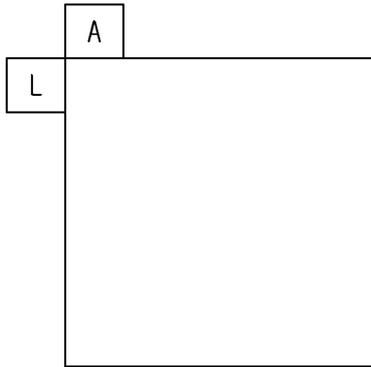
도면13



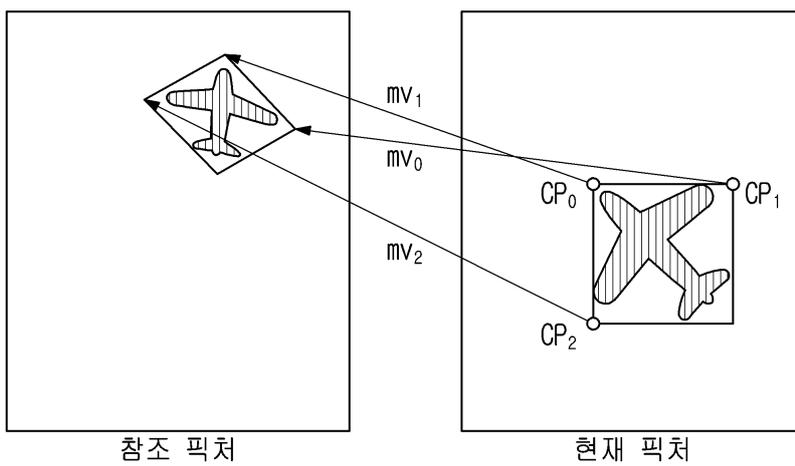
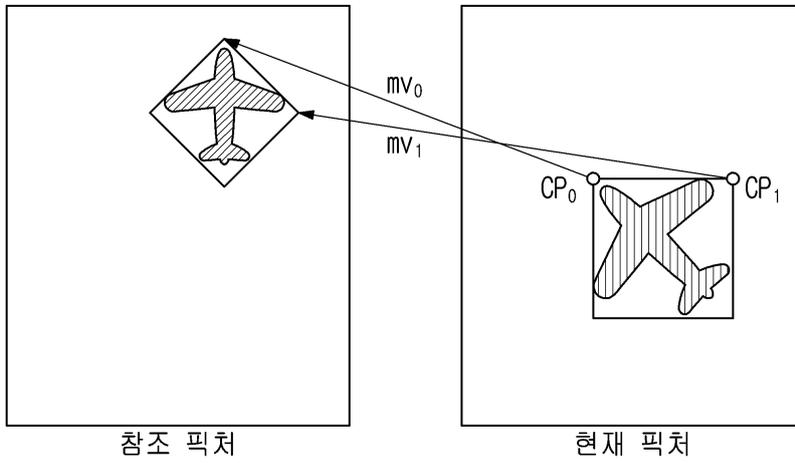
도면14



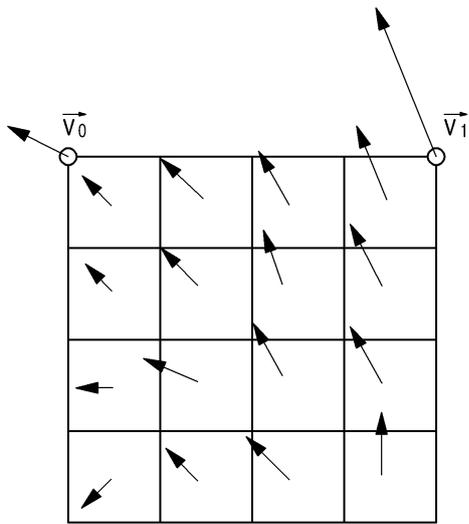
도면15



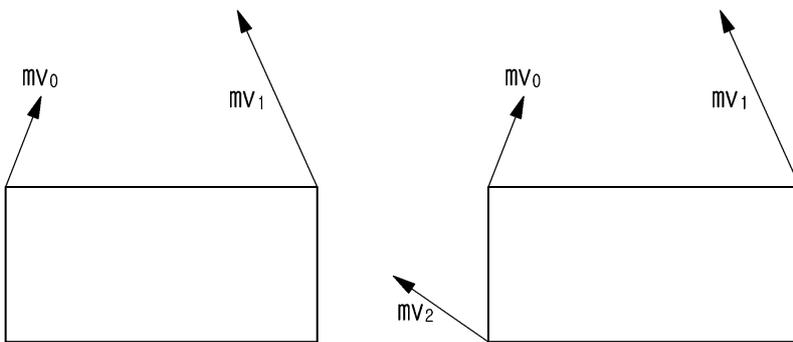
도면16



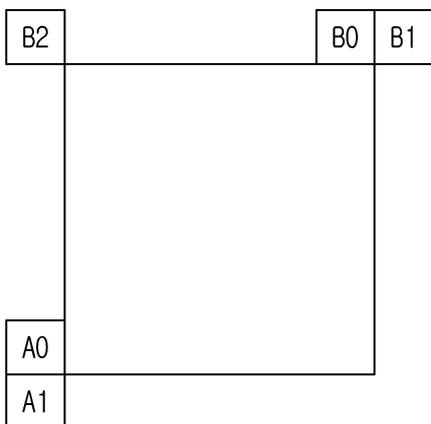
도면17



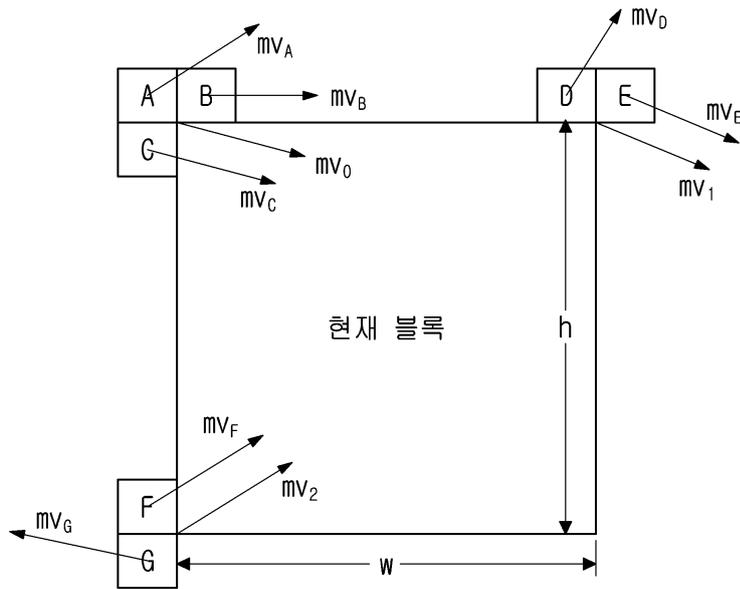
도면18



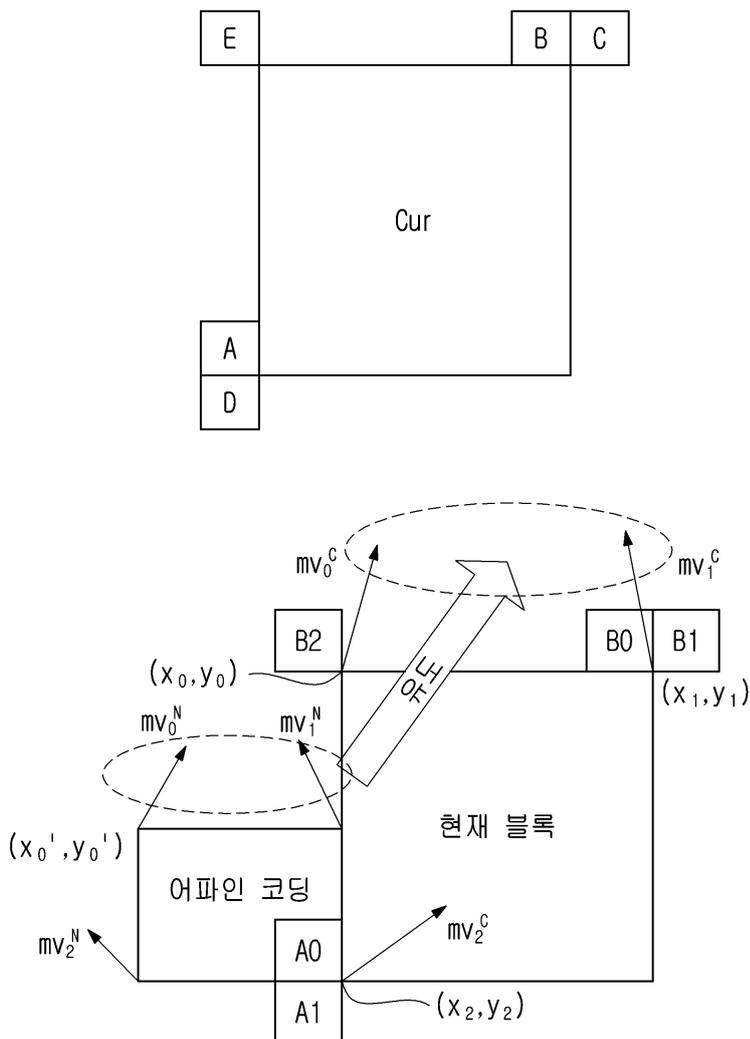
도면19



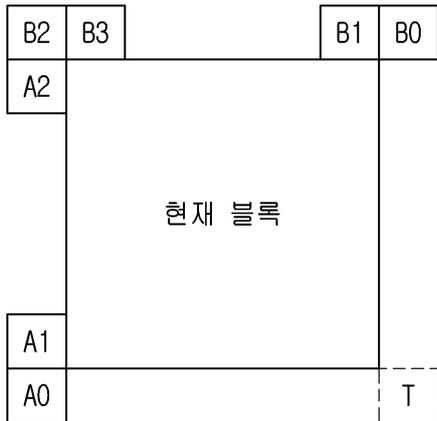
도면20



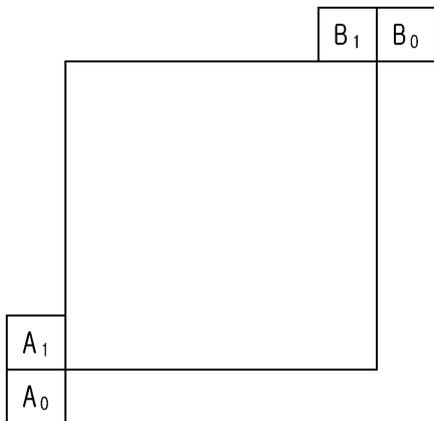
도면21



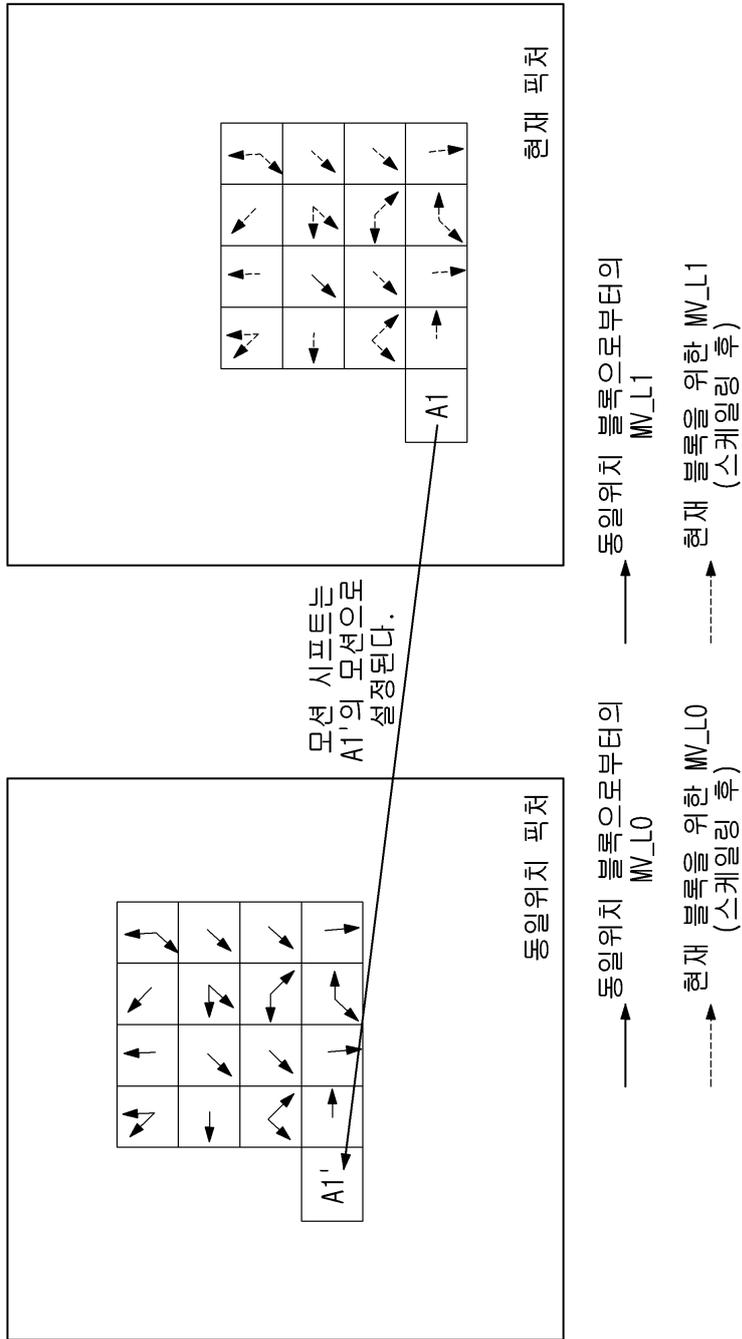
도면22



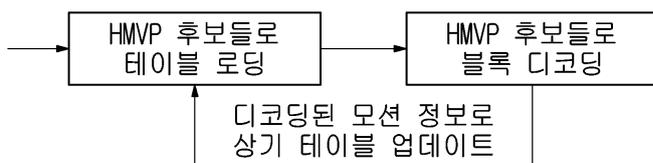
도면23



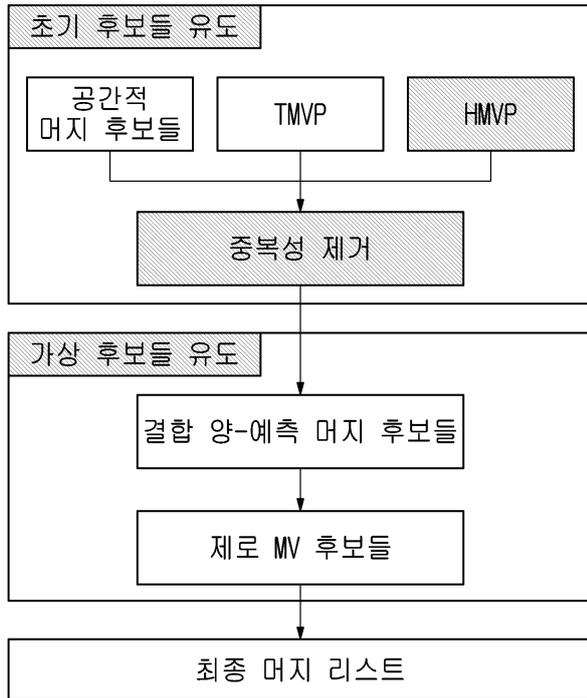
도면24



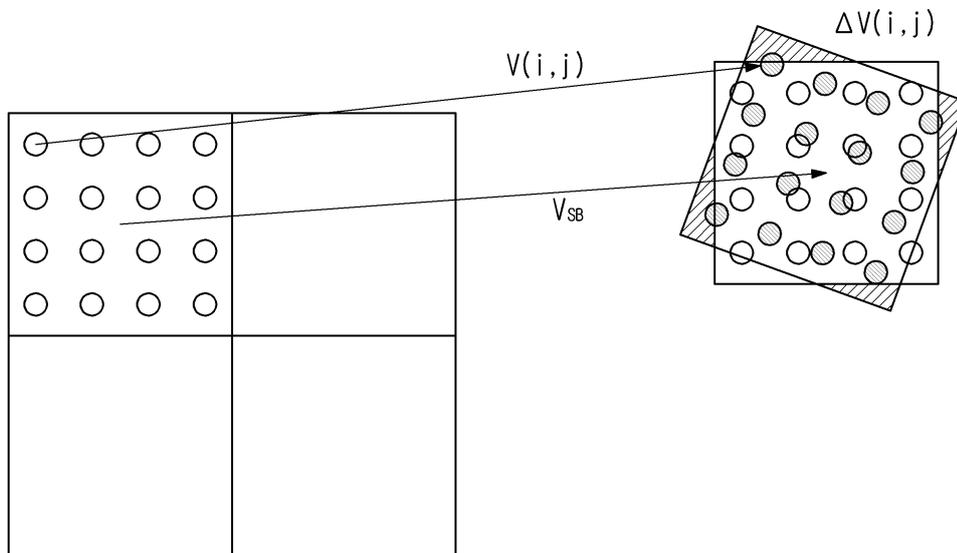
도면25



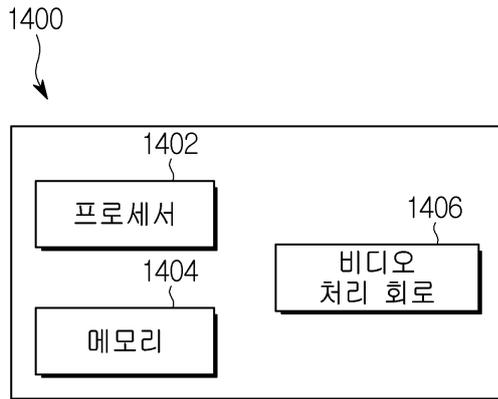
도면26



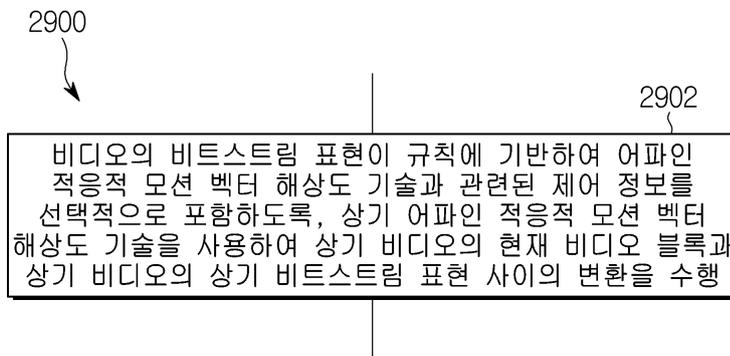
도면27



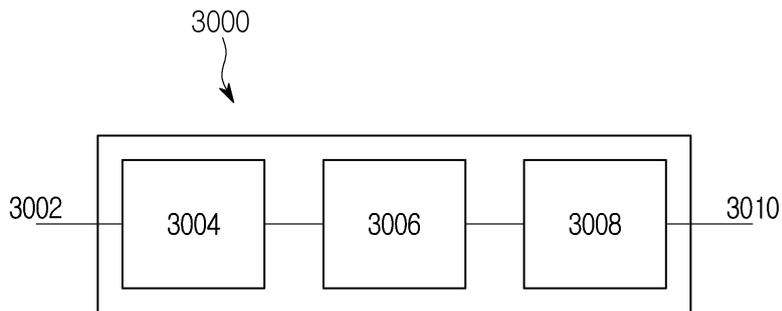
도면28



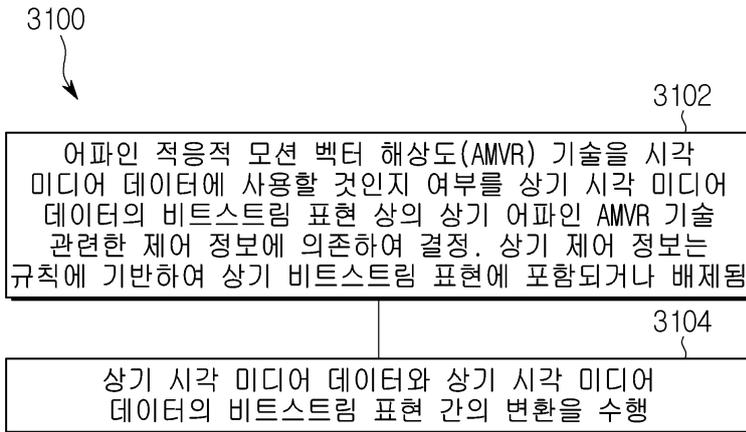
도면29



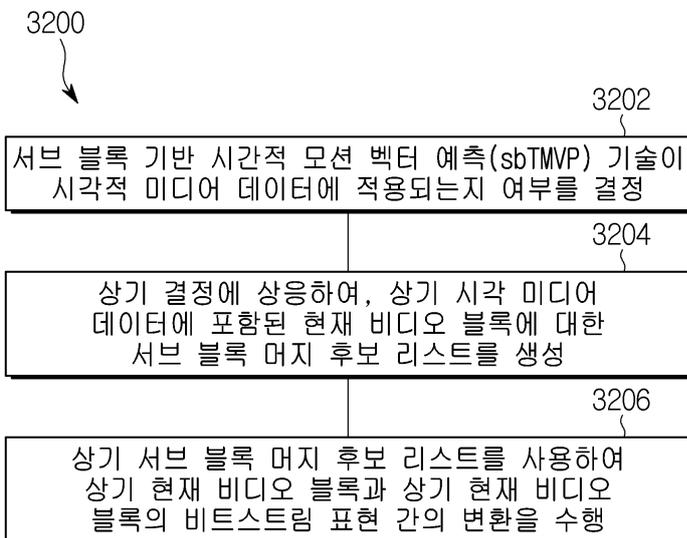
도면30



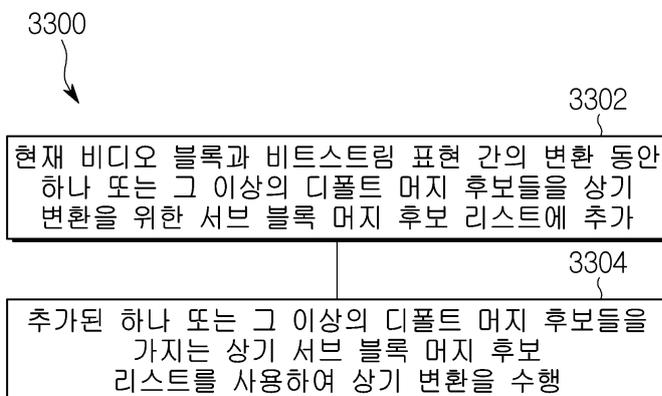
도면31



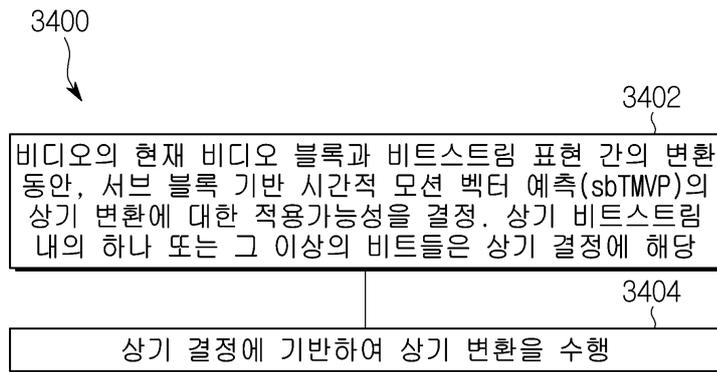
도면32



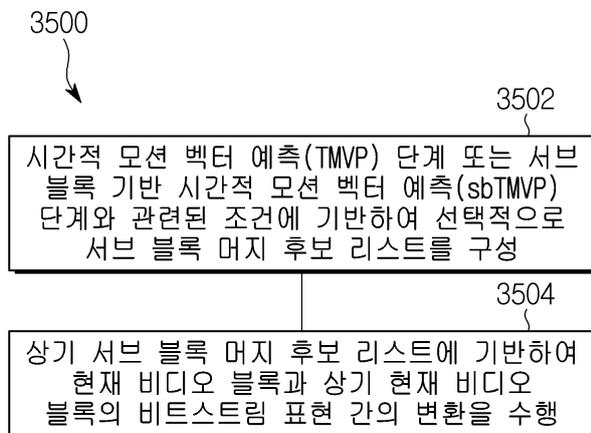
도면33



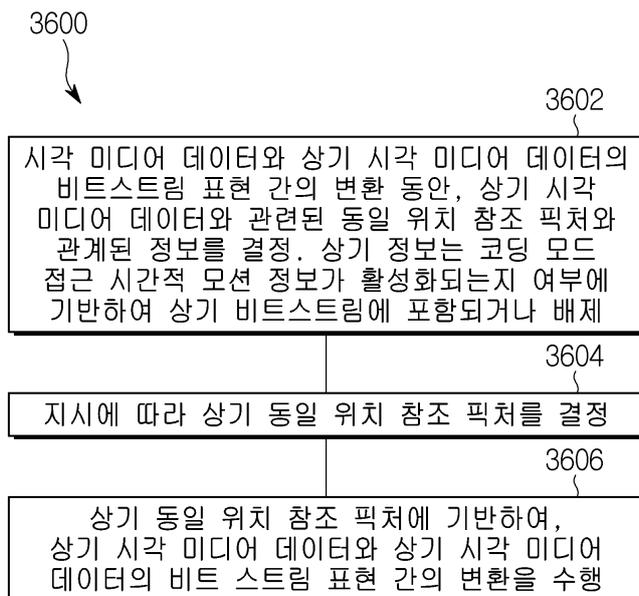
도면34



도면35



도면36



도면37

