



등록특허 10-2711051



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년09월27일
(11) 등록번호 10-2711051
(24) 등록일자 2024년09월24일

(30) 우선권주장
PCT/CN2018/114099 2018년 11월 06일 중국(CN)
(뒷면에 계속)

(56) 선행기술조사문현

US20120177114 A1*

Max Blaser, et. al., "Description of SDR and 360° video coding technology proposal by RWTH Aachen University", Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting: San Diego, US, 10-20 Apr. 2018. JVET-J0023-v1*

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 18 항

심사관 : 박상철

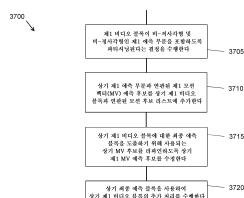
(54) 발명의 명칭 기하학적 패티션을 이용한 인터 예측을 위한 사이드 정보 시그널링

(57) 요약

비디오를 처리하기 위한 방법은, 프로세서에 의해, 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계; 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계 - 상기 제1 MV 예측 후보는 서브-블록

(뒷면에 계속)

대표도 - 도37



MV 예측 후보로부터 도출됨 -; 및 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.

(52) CPC특허분류

HO4N 19/119 (2015.01)

HO4N 19/159 (2015.01)

HO4N 19/176 (2015.01)

HO4N 19/184 (2015.01)

HO4N 19/186 (2015.01)

HO4N 19/1883 (2015.01)

HO4N 19/42 (2015.01)

HO4N 19/46 (2015.01)

HO4N 19/70 (2015.01)

(72) 발명자

류, 홍빈

중국 100080 베이징 하이뎬 디스트릭트 즈춘 로드
넘버 63 차이나 세틀라이트 커뮤니케이션즈 타워
진르터우탸오 포스트 오피스

왕, 웨

중국 100080 베이징 하이뎬 디스트릭트 즈춘 로드
넘버 63 차이나 세틀라이트 커뮤니케이션즈 타워
진르터우탸오 포스트 오피스

(30) 우선권주장

PCT/CN2018/125956 2018년12월30일 중국(CN)

PCT/CN2019/071160 2019년01월10일 중국(CN)

PCT/CN2019/071747 2019년01월15일 중국(CN)

명세서

청구범위

청구항 1

비디오 데이터 코딩 방법으로서,

비디오의 현재 블록과 상기 비디오의 비트스트림 사이의 변환을 위해, 상기 현재 블록이 크기 조건에 기초하여 기하학적 파티션 모드로 코딩되는지를 결정하는 단계 - 상기 기하학적 파티션 모드는 상기 현재 블록이 상기 크기 조건을 만족시키는 경우에 디스에이블됨 -; 및

상기 결정에 기초하여 상기 변환을 수행하는 단계

를 포함하고,

상기 현재 블록이 상기 기하학적 파티션 모드로 코딩되는 경우, 상기 비트스트림은 다수의 구문 요소를 포함하고, 그 중에서 하나의 구문 요소는 상기 현재 블록에 대한 상기 기하학적 파티션 모드의 분할 패턴을 표시하고, 다른 구문 요소들은 상기 현재 블록에 대한 다수의 병합 인덱스들을 표시하며,

상기 크기 조건은 상기 현재 블록의 폭 또는 높이가 제1 임계치보다 큰 것을 포함하고, 상기 제1 임계치는 64와 동일하고, 상기 기하학적 파티션 모드는 상기 현재 블록의 폭 또는 높이가 64보다 큰 것으로 인하여 디스에이블되는, 방법.

청구항 2

제1항에 있어서,

상기 다수의 병합 인덱스들 사이의 예측이 이용되는, 방법.

청구항 3

제1항에 있어서,

상기 기하학적 파티션 모드가 인에이블되는지를 특정하는 표시가 상기 비트스트림에 포함되는, 방법.

청구항 4

제3항에 있어서,

상기 표시는 시퀀스 파라미터 세트 레벨 또는 화상 파라미터 세트 레벨 또는 비디오 파라미터 세트 레벨 또는 화상 헤더 또는 슬라이스 헤더 또는 타일 그룹 헤더 또는 코딩 트리 단위 레벨에 있는, 방법.

청구항 5

제1항에 있어서,

상기 크기 조건은 상기 현재 블록의 폭 또는 높이가 제2 임계치보다 작은 것을 더 포함하고, 상기 제2 임계치는 8과 동일하고, 상기 기하학적 파티션 모드는 상기 현재 블록의 폭 또는 높이가 8보다 작은 것으로 인하여 디스에이블되는, 방법.

청구항 6

제1항에 있어서,

상기 크기 조건은 상기 현재 블록이 미리 정의된 폭 대 높이 비율을 갖는 것을 더 포함하는, 방법.

청구항 7

제1항에 있어서,

상기 변환은 상기 비트스트림으로부터 상기 다수의 구문 요소들을 파싱하는 것을 포함하는, 방법.

청구항 8

제1항에 있어서,

상기 기하학적 파티션 모드가 상기 변환을 위해 사용되는지는 상기 현재 블록의 높이 또는 상기 현재 블록의 폭 또는 상기 높이와 상기 폭의 비율에 의존하는, 방법.

청구항 9

제1항에 있어서,

다수의 병합 인덱스들을 표시하는 상기 구문 요소들은 상기 비트스트림 내에 개별적으로 코딩되는, 방법.

청구항 10

제1항에 있어서,

상기 기하학적 파티션 모드는 다수의 분할 패턴을 포함하고, 적어도 하나의 분할 패턴은 상기 현재 블록을 2개의 파티션으로 분할하고, 상기 2개의 파티션 중 적어도 하나는 비-정사각형 및 비-직사각형인, 방법.

청구항 11

제1항에 있어서,

다수의 병합 인덱스들의 수는 상기 기하학적 파티션 모드의 상기 분할 패턴에 따른 상기 현재 블록의 파티션들의 수와 동일한, 방법.

청구항 12

제1항에 있어서,

상기 기하학적 파티션 모드는 삼각형 파티션 모드를 포함하는, 방법.

청구항 13

제1항에 있어서,

상기 변환은 상기 비디오를 상기 비트스트림으로 인코딩하는 것을 포함하는, 방법.

청구항 14

제1항에 있어서,

상기 변환은 상기 비트스트림을 디코딩하여 상기 비디오의 픽셀 값들을 생성하는 것을 포함하는 방법.

청구항 15

제1항에 있어서,

상기 디코딩은 상기 비트스트림의 상기 구문 요소들을 파싱하는 것을 포함하는, 방법.

청구항 16

프로세서 및 명령어들을 갖는 비-일시적 메모리를 포함하는 비디오 데이터 코딩 장치로서, 상기 명령어들은 상기 프로세서에 의해 실행될 때, 상기 프로세서로 하여금:

비디오의 현재 블록과 상기 비디오의 비트스트림 사이의 변환을 위해, 상기 현재 블록이 크기 조건에 기초하여 기하학적 파티션 모드로 코딩되는지를 결정하고 - 상기 기하학적 파티션 모드는 상기 현재 블록이 상기 크기 조건을 만족시키는 경우에 디스에이블됨 -;

상기 결정에 기초하여 상기 변환을 수행하도록 하고,

상기 현재 블록이 상기 기하학적 파티션 모드로 코딩되는 경우, 상기 비트스트림은 다수의 구문 요소를 포함하

고, 그 중에서 하나의 구문 요소는 상기 현재 블록에 대한 상기 기하학적 파티션 모드의 분할 패턴을 표시하고, 다른 구문 요소들은 상기 현재 블록에 대한 다수의 병합 인덱스들을 표시하며,

상기 크기 조건은 상기 현재 블록의 폭 또는 높이가 제1 임계치보다 큰 것을 포함하고, 상기 제1 임계치는 64와 동일하고, 상기 기하학적 파티션 모드는 상기 현재 블록의 폭 또는 높이가 64보다 큰 것으로 인하여 디스에이블 되는, 비디오 데이터 코딩 장치.

청구항 17

명령어들을 저장하는 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령어들은 프로세서로 하여금:

비디오의 현재 블록과 상기 비디오의 비트스트림 사이의 변환을 위해, 상기 현재 블록이 크기 조건에 기초하여 기하학적 파티션 모드로 코딩되는지를 결정하고 - 상기 기하학적 파티션 모드는 상기 현재 블록이 상기 크기 조건을 만족시키는 경우에 디스에이블됨 -;

상기 결정에 기초하여 상기 변환을 수행하도록 하고,

상기 현재 블록이 상기 기하학적 파티션 모드로 코딩되는 경우, 상기 비트스트림은 다수의 구문 요소를 포함하고, 그 중에서 하나의 구문 요소는 상기 현재 블록에 대한 상기 기하학적 파티션 모드의 분할 패턴을 표시하고, 다른 구문 요소들은 상기 현재 블록에 대한 다수의 병합 인덱스들을 표시하며,

상기 크기 조건은 상기 현재 블록의 폭 또는 높이가 제1 임계치보다 큰 것을 포함하고, 상기 제1 임계치는 64와 동일하고, 상기 기하학적 파티션 모드는 상기 현재 블록의 폭 또는 높이가 64보다 큰 것으로 인하여 디스에이블 되는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 18

비디오 처리 장치에 의해 수행되는 방법에 의해 생성되는 비트스트림을 저장하는 비-일시적 컴퓨터 판독가능 기록 매체로서, 상기 방법은:

비디오의 현재 블록과 상기 비디오의 비트스트림 사이의 변환을 위해, 상기 현재 블록이 크기 조건에 기초하여 기하학적 파티션 모드로 코딩되는지를 결정하는 단계 - 상기 기하학적 파티션 모드는 상기 현재 블록이 상기 크기 조건을 만족시키는 경우에 디스에이블됨 -; 및

상기 결정에 기초하여 상기 현재 블록으로부터 상기 비트스트림을 생성하는 단계

를 포함하고,

상기 현재 블록이 상기 기하학적 파티션 모드로 코딩되는 경우, 상기 비트스트림은 다수의 구문 요소를 포함하고, 그 중에서 하나의 구문 요소는 상기 현재 블록에 대한 상기 기하학적 파티션 모드의 분할 패턴을 표시하고, 다른 구문 요소들은 상기 현재 블록에 대한 다수의 병합 인덱스들을 표시하며,

상기 크기 조건은 상기 현재 블록의 폭 또는 높이가 제1 임계치보다 큰 것을 포함하고, 상기 제1 임계치는 64와 동일하고, 상기 기하학적 파티션 모드는 상기 현재 블록의 폭 또는 높이가 64보다 큰 것으로 인하여 디스에이블 되는, 비-일시적 컴퓨터 판독가능 기록 매체.

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

발명의 설명**기술 분야**

[0001]

삭제

[0002]

삭제

[0003] 본 문서는 비디오 및 이미지 코딩 및 디코딩 기술들에 관한 것이다.

배경기술

디지털 비디오는 인터넷 및 다른 디지털 통신 네트워크들에서 가장 큰 대역폭 사용을 차지한다. 비디오를 수신 및 디스플레이할 수 있는 연결된 사용자 디바이스들의 수가 증가함에 따라, 디지털 비디오 사용에 대한 대역폭 수요가 계속 증가할 것으로 예상된다.

발명의 내용

[0005] 개시된 기법들은 기하학적 파티션들이 비디오 코딩 또는 디코딩을 위해 사용될 수 있는 비디오 또는 이미지 디코더 또는 인코더 실시예들에 의해 사용될 수 있다.

[0006] 하나의 예시적인 양태에서, 비디오 처리 방법이 개시된다. 이 방법은 비디오의 비디오 영역의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환이 상기 비디오 블록이 적어도 제1 예측 파티션을 포함하는 다수의 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용한다는 결정을 하는 단계, 상기 비디오 블록과 연관된 하나 이상의 서브-블록 병합 후보를 사용하여 N개의 모션 정보 후보를 상기 비디오 블록에 대한 모션 후보 리스트에 추가하는 단계 - N은 양의 정수임 -; 상기 모션 후보 리스트로부터, 상기 다수의 예측 파티션에 대한 모션 후보들을 도출하는 단계; 및 상기 다수의 예측 파티션에 대한 모션 후보에 기초하여 상기 변환을 수행하는 단계를 포함한다.

[0007] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 비디오 블록이 적어도 제1 예측 파티션을 포함하는 다수의 예측 파티션으로 파티셔닝된다고 결정하는 단계; 상기 제1 예측 파티션과 연관된 제1 후보 리스트에 제1 모션 정보 후보를 추가하고 제2 예측 파티션과 연관된 제2 후보 리스트에 연관된 제2 모션 정보 후보를 추가하는 단계 - 상기 제1 모션 정보 후보 및 상기 제2 모션 정보 후보는 단일 모션 후보로부터 상속됨 -; 상기 제1 모션 정보 후보 및/또는 상기 제2 모션 정보 후보를 사용하여 상기 변환을 수행하는 단계를 포함한다.

[0008] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환이 상기 비디오 블록이 예측 파티션을 포함하도록 파티셔닝되어 상기 예측 파티션의 서브-블록에 그의 서브-블록 모션 정보가 할당되게 되는 서브-블록 인에이블 기하학적 파티션 모드(sub-block enabled geometric partitioning mode)를 사용한다는 결정을 하는 단계; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함한다.

[0009] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은, 비디오 블록이 다수의 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 예측 파티션에 대한 모션 정보 도출을 위해 사용되는 모션 후보들로부터의 GBi(generalized bi prediction) 가중치(weighting factor)들의 상속이 상속 규칙에 따라 상기 비디오 블록의 샘플 위치들에서 인에이블되는지를 결정하는 단계; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함한다. GBi는 또한 BCW(Bi-prediction with Coding Unit based Weighting)로도 알려져 있다.

- [0010] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은, 비디오 블록이 다수의 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 적어도 비디오 블록의 샘플 위치들에서 규칙에 따라 양방향 예측의 사용이 인에이블되는지를 결정하는 단계; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함한다.
- [0011] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은, 비디오 블록이 적어도 제1 예측 파티션 및 제2 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 제1 예측 파티션의 샘플 위치들에 대해 도출된 제1 모션 정보 세트, 상기 제2 예측 파티션의 샘플 위치들에 대해 도출된 제2 모션 정보 세트와 상이한 모션 정보로부터의 제1 예측 블록 및 제2 예측 블록으로부터 최종 예측 블록을 결정하는 단계; 및 상기 최종 예측 블록에 기초하여 상기 변환을 수행하는 단계를 포함하고; 상기 비디오 블록은 다중-가설(multi-hypothesis) 기법을 사용하여 코딩된다.
- [0012] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은, 비디오의 비디오 영역의 비디오 블록과 상기 비디오의 비트스트림 표현 간의 변환을 위해, (1) 적어도 하나의 예측 파티션이 비-직사각형 및 비-정사각형 파티션이 되도록 상기 비디오 블록을 예측 파티션들로 분할하기 위해 사용되는 분할 패턴과, (2) 상기 변환을 위해 사용되는 파티션들의 후보들을 병합하기 위한 인덱스들 간의 관계를 결정하는 단계 - 상기 비트스트림 표현의 포맷은 상기 비디오 영역 레벨에서 상기 관계를 변경하는 것을 허용함 -; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함한다.
- [0013] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 코딩된 표현의 구문 요소들을 정의하는 구문 규칙에 따라 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고; 상기 구문 규칙은 상기 비디오 블록에 대한 기하학적 파티션 모드의 사용을 나타내는 제1 플래그 및 상기 비디오 블록에 대한 병합 모드의 사용을 나타내는 제2 플래그의 시그널링 순서를 특정하고; 상기 기하학적 파티션 모드는 상기 비디오 블록을 다수의 예측 파티션으로 분할하는 것을 포함하고, 적어도 하나의 예측 파티션은 비-직사각형, 비-정사각형 형상을 가지고, 상기 병합 모드는 상기 비디오 블록의 이웃 블록들로부터 모션 벡터 정보를 상속하는 것을 허용한다.
- [0014] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 코딩된 표현의 구문을 정의하는 구문 규칙에 따라 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고; 상기 구문 규칙은 상기 비디오 블록에 대한 기하학적 파티션 모드의 사용을 나타내는 플래그의 시그널링이 다른 코딩 툴이 상기 비디오 블록의 변환을 위해 사용되는지에 기초하여 선택적으로 포함된다는 것을 특정하고; 상기 기하학적 파티션 모드는 상기 비디오 블록을 다수의 예측 파티션으로 분할하는 것을 포함하고, 적어도 하나의 예측 파티션은 비-직사각형, 비-정사각형 형상이다.
- [0015] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해 기하학적 파티션 모드가 인에이블되어 있다고 결정하는 단계 - 상기 비디오 블록은 이웃 블록의 기하학적 파티션 모드에 기초한 분할 패턴에 기초하여 다수의 예측 파티션으로 분할됨 -, 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함한다.
- [0016] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오 블록이 이웃 블록의 분할 패턴에 기초한 분할 패턴에 기초하여 다수의 예측 파티션으로 분할되는 기하학적 파티션 모드에 기초하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위한 분할 패턴을 결정하는 단계; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함한다.
- [0017] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고, 상기 비디오 블록은 제1 예측 파티션을 포함하는 기하학적 파티션 모드의 분할 패턴에 따라 다수의 예측 파티션으로 파티셔닝되고; 상기 코딩된 표현은 다수의 구문 요소를 포함하고, 그 중에서 하나의 구문 요소는 상기 분할 패턴에 대응하고 다른 구문 요소는 상기 다수의 예측 파티션에 대한 병합 인덱스들에 대응한다.
- [0018] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환이 기하학적 파티션 모드에 기초한다고 결정하는 단계 - 상기 비디오 블록은 제1 예측 파티션을 포함하는 분할 패턴에 따라 다수의 예측 파티션으로 파티셔닝됨 -; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함하고; 상기 비디오 블록에 대한 상기 기하학적 파티션 모드는 컨텍스트가 상기 비디오 블록의 코딩 조건에 의존하는 컨텍스트 기반 코딩을 사용하여 상기 코딩된 표현으로 시그널링된다.

- [0019] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고, 상기 비디오 블록은 제1 예측 파티션을 갖는 분할 패턴에 따라 기하학적 예측 모드를 사용하여 다수의 예측 파티션으로 파티셔닝되고; 상기 후보 리스트의 2개의 후보 인덱스 엔트리에는 상기 코딩된 표현 내의 코드워드가 할당되고, 상기 코딩된 표현의 포맷은 상기 비디오의 세그먼트에 걸쳐 상기 코드워드의 할당을 변경하는 것을 허용한다.
- [0020] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 비디오 블록의 특성을 사용하는 규칙에 기초하여 기하학적 파티션 모드의 사용이 인에이블되는지를 결정하는 단계; 상기 결정에 따라 상기 변환을 수행하는 단계를 포함하고, 상기 기하학적 파티션 모드는 상기 비디오 블록을 다수의 예측 파티션으로 파티셔닝한다.
- [0021] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 코딩된 표현의 구문 요소들의 포맷을 정의하는 구문 규칙에 따라 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고; 상기 규칙은 제1 코딩 모드의 사용의 지시가 시그널링되는지 또는 상기 지시를 시그널링할지가 상기 비디오 블록에 대한 제2 코딩 모드의 사용에 기초한다는 것을 특정하고, 상기 제2 코딩 모드는 병합 모드이고; 상기 병합 모드는 상기 비디오 블록 전체에 대한 모션 벡터 차이 없이 병합 후보 리스트 내의 병합 후보로부터 모션 정보를 상속하는 것을 인에이블한다.
- [0022] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고, 상기 코딩된 표현의 포맷은 다수의 코딩 모드를 시그널링하는 구문 요소들이 상기 코딩된 표현에 포함되는지를 특정하는 제1 규칙 및/또는 상기 다수의 코딩 모드를 시그널링하는 구문 요소들이 상기 코딩된 표현에 포함되는 순서를 특정하는 제2 규칙에 따르고; 상기 다수의 코딩 모드는 기하학적 파티션 모드, 인터-인트라(MHIntra) 모드, 서브-블록 병합 모드 또는 MMVD(merge with motion vector differencing) 모드 중 2개 이상을 포함하고; 상기 인터-인트라 코딩 모드에서는, 상기 비디오 블록의 예측 블록이 인트라 예측 신호 및 인터 예측 신호로부터 도출되고; 상기 서브-블록 병합 모드에서는, 상기 변환이 상기 블록 내의 각각의 서브-블록에 대해 도출된 모션 정보를 사용하고; 상기 MMVD 모드에서는, 조합된 병합 및 MVD(motion vector differencing) 코딩 모드가 사용되고; 상기 병합 모드는 상기 비디오 블록 전체에 대한 MVD 없이 병합 후보 리스트 내의 병합 후보로부터 모션 정보를 상속하는 것을 인에이블한다.
- [0023] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은, 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 비디오 블록의 이웃 블록의 이웃 모션 벡터 정밀도 정보가 이웃 블록의 위치에 기초하여 현재 블록에 대한 모션 벡터 정밀도를 결정하기 위해 이용가능한지를 검사하는 단계; 및 상기 검사에 기초하여 상기 변환을 수행하는 단계를 포함한다.
- [0024] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 현재 비디오 블록의 최대 코딩 단위(largest coding unit, LCU) 행에 대한 이웃 블록의 위치에 기초하여 상기 코딩된 표현을 위해 사용되는 컨텍스트 코딩 모드를 결정하는 단계; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계를 포함한다.
- [0025] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고, 상기 코딩된 표현은 컨텍스트 모델링이 상기 현재 비디오 블록을 디코딩하기 위해 사용되는 이웃 블록으로부터의 정보로 제한되고/되거나 위치 기준을 만족시키는 것을 특정하는 규칙에 따른다.
- [0026] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계를 포함하고, 상기 코딩된 표현은 상기 현재 비디오 블록의 코딩된 표현을 나타내는 하나 이상의 구문 요소의 컨텍스트 모델링을 위해 이웃 블록들로부터의 코딩 정보의 사용을 특정하는 규칙에 따른다.
- [0027] 다른 예시적인 양태에서, 비디오를 처리하는 방법이 개시된다. 이 방법은, 프로세서에 의해, 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계; 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계 - 상기 제1 MV 예측 후보는 서브-블록 기반 모션 후보로부터 도출됨 -; 및 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.
- [0028] 다른 예시적인 양태에서, 비디오를 처리하는 방법은 제1 분할 패턴에 기초한 제1 예측 부분을 포함하도록 제1

비디오 블록을 파티셔닝하는 단계 - 상기 제1 예측 부분은 제1 병합 인덱스 및 제1 코딩된 값과 연관됨 -; 제2 분할 패턴에 기초한 제2 예측 부분을 포함하도록 제2 비디오 블록을 파티셔닝하는 단계 - 상기 제2 비디오 블록의 상기 제2 예측 부분은 제2 병합 인덱스 및 제2 코딩된 값과 연관되고, 상기 제1 코딩된 값은 상기 제2 코딩된 값과 동등하고, 상기 제2 분할 패턴, 제2 병합 인덱스, 상기 제2 비디오 블록 중 하나 이상은 상기 제2 비디오 블록이 상기 제1 비디오 블록과 상이한 시퀀스, 화상, 타일, 슬라이스, 코딩 트리 단위(CTU) 또는 블록 내에 있는 것에 기초하여 상기 제1 비디오 블록과 상이함 -; 상기 제1 비디오 블록 및 상기 제2 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.

[0029] 다른 예시적인 양태에서, 비디오를 처리하는 방법은, 프로세서에 의해, 제1 비디오 블록의 특성을 식별하는 단계; 상기 특성들에 기초하여 상기 제1 비디오 블록에 적용될 기하학적 예측 모드의 상태를 결정하는 단계 - 상기 기하학적 예측 모드의 상태는: 상기 제1 비디오 블록에 기하학적 예측 모드를 적용하는 인에이블 상태, 또는 상기 제1 비디오 블록에 기하학적 예측 모드를 적용하지 않는 디스에이블 상태 중 하나임 -; 및 상기 기하학적 예측 모드의 상태와 부합하는 상기 제1 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.

[0030] 다른 예시적인 양태에서, 비디오를 처리하는 방법은, 프로세서에 의해, 제1 비디오 블록이 제1 예측 부분 및 제2 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계 - 상기 제1 예측 부분 및 상기 제2 예측 부분은 비-직사각형 및 비-정사각형임 -; 상기 제1 예측 부분과 연관된 제1 MV 예측 후보 및 상기 제2 예측 부분과 연관된 제2 MV 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계 - 상기 제1 MV 예측 후보 및 상기 제2 MV 예측 후보는 이전에 코딩된 블록과 연관된 단일 MV 예측 후보로부터 도출됨 -; 및 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.

[0031] 다른 예시적인 양태에서, 비디오를 처리하는 방법은, 프로세서에 의해, 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계; 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계; 상기 제1 비디오 블록에 대한 최종 예측 블록을 도출하기 위해 사용되는 상기 MV 후보들을 리파인하도록 상기 제1 MV 예측 후보를 수정하는 단계; 및 상기 최종 예측 블록을 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.

[0032] 다른 예시적인 양태에서, 비디오를 처리하는 방법은, 프로세서에 의해, 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계; 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계 - 상기 제1 MV 예측 후보는 가중치들을 갖는 양방향 예측(bi-prediction) MV 예측 후보로부터 도출됨 -; 및 상기 가중치들 없이 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.

[0033] 다른 예시적인 양태에서, 비디오를 처리하는 방법은, 프로세서에 의해, 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝되고, 제1 비디오 블록이 다중-가설과 부합하도록 코딩되어 있다는 결정을 수행하는 단계를 포함한다.

[0034] 다른 예시적인 양태에서, 비디오를 처리하는 방법은, 프로세서에 의해, 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계를 포함하고, 오버랩 블록 모션 보상(overlapped block motion compensation, OBMC)이 적용되지 않는다.

[0035] 다른 예시적인 양태에서, 비디오를 처리하는 방법은, 프로세서에 의해, 제1 비디오 블록이 2개보다 많은 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계를 포함한다. 일 예에서, 상기 제1 비디오 블록은 각각이 비-직사각형 및 비-정사각형인 제1 예측 부분, 제2 예측 부분, 제3 예측 부분, 및 제4 예측 부분으로 파티셔닝되고; 상기 제1 예측 부분, 상기 제2 예측 부분, 상기 제3 예측 부분, 및 상기 제4 예측 부분을 사용하여 상기 제1 비디오 블록의 추가 처리를 수행한다.

[0036] 다른 예시적인 양태에서, 비디오 처리 방법이 개시된다. 이 방법은 코딩의 스킵/병합 모드가 제2 코딩 모드와 별개로 처리되는 것을 특정하는 코딩 규칙과 부합하도록 비디오 블록과 상기 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 제2 코딩 모드는 삼각형 파티션 모드(triangular partition mode) 또는 MHIntra(때때로 조합된 인터-인트라 모드(combined inter-intra mode), 또는 CIIP라고도 불림) 모드 또는 서브-블록 병합 리스트 모드이다.

[0037] 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 변환에서 복수의 서브-블록 코딩 틀의 사용의 지시의 특정 순서를 특정하는 규칙과 부합하도록 비디오 블록과 상기 비디오 블록의 비트스트림 표현 간

의 변환을 수행하는 단계를 포함하고, 상기 특정 순서는 (a.) MMVD(merge with motion vector difference)에 이어서 서브-블록 병합 리스트에 이어서 삼각형 파티션 모드 TPM에 이어서 MHIntra, 또는 (b.) MMVD에 이어서 서브-블록 병합 리스트에 이어서 MHIntra에 이어서 TPM 중 하나이다.

[0038] 또 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 블록의 선택적 사용을 특정하고, 상기 규칙은: (1) 상이한 최대 코딩 단위(LCU) 행에 있는 이웃 블록을 제외하는 것 또는 (2) 고정된 해상도 값을 특정하는 것 또는 (3) 비디오 블록이 아핀 모드를 사용하여 코딩되는 경우, 상이한 LCU 행에 있는 아핀-코딩된 이웃 블록을 이용불가능하게 하는 것을 포함한다.

[0039] 또 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 블록의 선택적 사용을 특정하고, 상기 규칙은: (a) 이웃 블록이 상이한 최대 코딩 단위(LCU) 행에 있는 경우 상기 비디오 블록의 컨텍스트 모델링을 위해 이웃 비디오 블록의 정보를 사용하는 것을 디스에이블하는 것 또는 (b) 상기 변환 동안에 구문 요소를 코딩하기 위해 단 하나의 컨텍스트만을 사용하는 것 또는 (c) 바이패스 코딩을 적용하는 것이 단 하나의 구문 요소에 적용되는 것으로서, 이웃 블록의 선택적 사용을 포함한다.

[0040] 또 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 비디오 블록들의 선택적 사용을 특정하고, 상기 규칙은: 이웃 비디오 블록들에 관한 정보가 현재 비디오 블록을 디코딩하기 위해 사용되고 있는 이웃 비디오 블록들에 기초하여 컨텍스트 모델링을 위해 사용되는 것을 포함한다.

[0041] 또 다른 예시적인 양태에서, 다른 비디오 처리 방법이 개시된다. 이 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 비디오 블록들의 선택적 사용을 특정하고, 상기 규칙은: 이웃 비디오 블록들에 관한 정보가 현재 비디오 블록을 디코딩하기 위해 사용되고 있는 이웃 비디오 블록들에 기초하여 컨텍스트 모델링을 위해 사용되는 것을 포함하고, 이웃 비디오 블록들은 현재 CTU 행과 상이한 CTU 행 내의, 또는 현재 CTU와 상이한 CTU 내의 비디오 블록들을 제외하고 사용된다.

[0042] 다른 예시적인 양태에서, 전술한 방법은 프로세서를 포함하는 비디오 인코더 장치에 의해 구현될 수 있다.

[0043] 또 다른 예시적인 양태에서, 이들 방법은 프로세서 실행가능 명령어들의 형식으로 구현되고 컴퓨터 판독가능 프로그램 매체 상에 저장될 수 있다.

[0044] 이를 및 다른 양태들은 본 문서에서 더 설명된다.

도면의 간단한 설명

[0045] 도 1은 병합 후보들 리스트 구성을 위한 도출 프로세스의 예를 보여준다.

도 2는 공간 병합 후보들의 위치들의 예를 보여준다.

도 3은 공간 병합 후보들의 중복 검사를 위해 고려되는 후보 쌍들의 예를 보여준다.

도 4는 $N \times 2N$ 및 $2N \times N$ 파티션들의 제2 PU에 대한 위치들의 예를 보여준다.

도 5는 시간 병합 후보에 대한 모션 벡터 스케일링의 예를 보여준다.

도 6은 시간 병합 후보 C0 및 C1에 대한 후보 위치들의 예를 보여준다.

도 7은 조합된 양방향 예측 병합 후보의 예를 보여준다.

도 8은 모션 벡터 예측 후보들에 대한 도출 프로세스의 예를 보여준다.

도 9는 공간 모션 벡터 후보에 대한 모션 벡터 스케일링의 예를 보여준다.

도 10은 CU에 대한 ATMVP 모션 예측의 예를 보여준다.

도 11은 4개의 서브-블록(A-D) 및 그의 이웃 블록들(a-d)을 갖는 하나의 CU의 예를 보여준다.

도 12는 상이한 MV 정밀도로 인코딩하는 흐름도의 예를 보여준다.

도 13은 CU를 2개의 삼각형 예측 단위(2개의 파티션 유형)로 분할하는 예들을 도시한다.

도 14는 이웃 블록들의 위치의 예를 보여준다.

도 15는 제1 가중치 그룹을 적용하는 CU의 예를 보여준다.

도 16은 모션 벡터 저장의 예를 보여준다.

도 17은 TPM 플래그 코딩에서 컨텍스트 선택을 위해 사용되는 이웃 블록들(A 및 L)의 예를 보여준다.

도 18은 OBMC가 적용되는 서브-블록들의 예를 보여준다.

도 19는 IC 파라미터들을 도출하기 위해 사용되는 이웃 샘플들의 예를 보여준다.

도 20은 단순화된 아핀 모션 모델의 예를 보여준다.

도 21은 서브-블록당 아핀 MVF의 예를 보여준다.

도 22는 4-파라미터 아핀 모델 (a) 및 6-파라미터 아핀 모델 (b)의 예를 보여준다.

도 23은 AF_INTER에 대한 MVP의 예를 보여준다.

도 24는 AF_MERGE에 대한 후보들의 예를 보여준다.

도 25는 아핀 병합 모드에 대한 후보들 위치의 예를 보여준다.

도 26은 광학 흐름 궤적(optical flow trajectory)의 예를 보여준다.

도 27은 블록 확장이 있는/없는 BIO의 예를 보여준다: a) 블록 외부의 액세스 위치들; b) 추가의 메모리 액세스 및 계산을 피하기 위해 패딩이 사용된다.

도 28은 쌍방 템플릿 매칭에 기초한 DMVR의 예를 보여준다.

도 29는 하나의 블록 내의 다수의 기하학적 파티션의 예를 보여준다.

도 30은 TPM 플래그 코딩에서 컨텍스트 선택을 위해 사용되는 이웃 블록들의 예들을 도시한다.

도 31은 비디오 처리 장치의 예의 블록도이다.

도 32는 비디오 인코더의 예시적인 구현의 블록도를 도시한다.

도 33 내지 도 41은 비디오 처리 방법들의 예들에 대한 흐름도들을 도시한다.

도 42는 컨텍스트 모델링에서 사용되는 이웃 요소의 예를 보여준다.

도 43은 pred_mode_flag의 컨텍스트를 도출하기 위해 사용되는 이웃 블록들의 예를 보여준다.

도 44 내지 도 63은 비디오 처리 방법들의 예들에 대한 흐름도들을 도시한다.

도 64는 개시된 기법들이 구현될 수 있는 예시적인 비디오 처리 시스템의 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0046] 본 문서는 압축해제된 또는 디코딩된 디지털 비디오 또는 이미지들의 품질을 개선하기 위해 이미지 또는 비디오 비트스트림들의 디코더에 의해 사용될 수 있는 다양한 기법들을 제공한다. 간략함을 위해, "비디오"라는 용어는 본 명세서에서 화상들의 시퀀스(전통적으로 비디오라고 불림) 및 개별 이미지를 둘 다를 포함하기 위해 사용된다. 더욱이, 비디오 인코더는 또한 추가 인코딩을 위해 사용되는 디코딩된 프레임들을 재구성하기 위해 인코딩 프로세스 동안 이를 기법을 구현할 수 있다.

[0047] 섹션 제목들은 이해의 편의를 위해 본 문서에서 사용되고 실시예들 및 기법들을 대응하는 섹션들로 제한하지 않는다. 그에 따라, 하나의 섹션으로부터의 실시예들은 다른 섹션들로부터의 실시예들과 조합될 수 있다.

1. 요약

[0049] 본 특허 문서는 비디오 코딩 기술들에 관한 것이다. 구체적으로는, 그것은 비디오 코딩에서 기하학적 파티션 하의 모션 벡터 코딩에 관한 것이다. 그것은 HEVC와 같은 기존의 비디오 코딩 표준, 또는 완성될 표준 (Versatile Video Coding)에 적용될 수 있다. 그것은 미래의 비디오 코딩 표준들 또는 비디오 코덱에도 적용 가능할 수 있다.

2. 논의

[0051] 비디오 코딩 표준들은 주로 잘 알려진 ITU-T 및 ISO/IEC 표준들의 개발을 통해 발전했다. ITU-T는 H.261 및 H.263을 생성하였고, ISO/IEC는 MPEG-1 및 MPEG-4 Visual을 생성하였고, 이 2개의 조직은 H.262/MPEG-2 비디오 및 H.264/MPEG-4 AVC(Advanced Video Coding) 및 H.265/HEVC [1] 표준들을 공동으로 생성하였다. H.262 이후, 비디오 코딩 표준들은 시간 예측 플러스 변환 코딩이 이용되는 하이브리드 비디오 코딩 구조에 기초한다. HEVC 이후 미래의 비디오 코딩 기술들을 탐구하기 위해, JVET(Joint Video Exploration Team)가 2015년 공동으로 VCEG 및 MPEG에 의해 설립되었다. 그때 이후, 많은 새로운 방법들이 JVET에 의해 채택되어 JEM(Joint Exploration Model)이라고 명명된 기준 소프트웨어에 넣어졌다. 2018년 4월에, VCEG(Q6/16)와 ISO/IEC JTC1 SC29/WG11(MPEG) 사이의 JVET(Joint Video Expert Team)는 HEVC에 비해 50% 비트레이트 감소를 목표로 하는 VVC 표준에 대해 연구하기 위해 생성되었다.

[0052] 도 32는 비디오 인코더의 예시적인 구현의 블록도이다. 도 32는 인코더 구현이 비디오 인코더가 또한 비디오 디코딩 기능성을 수행하는(다음 비디오 데이터를 인코딩하는 데 사용하기 위한 비디오 데이터의 압축된 표현을 재구성하는) 내장된 피드백 경로를 갖는 것을 보여준다.

2.1 HEVC/H.265에서의 인터 예측

[0054] 각각의 인터-예측된 PU는 1개 또는 2개의 참조 화상 리스트에 대한 모션 파라미터들을 갖는다. 모션 파라미터들은 모션 벡터 및 참조 화상 인덱스를 포함한다. 2개의 참조 화상 리스트 중 하나의 사용은 *inter_pred_idc*를 사용하여 시그널링될 수도 있다. 모션 벡터들은 예측자들에 대해 멜타로서 명시적으로 코딩될 수 있다.

[0055] CU가 스킵 모드로 코딩될 때, 하나의 PU는 CU와 연관되고, 중요한 잔여 계수들이 없고, 코딩된 모션 벡터 멜타 또는 참조 화상 인덱스가 없다. 병합 모드가 특정되고, 그에 의해 현재 PU에 대한 모션 파라미터들이 공간 및 시간 후보들을 포함하는 이웃 PU들로부터 획득된다. 병합 모드는 스킵 모드에 대해서뿐만 아니라 임의의 인터-예측된 PU에 적용될 수 있다. 병합 모드에 대한 대안은 모션 파라미터들의 명시적 송신이고, 여기서 모션 벡터 (더 정확하게는, 모션 벡터 예측자에 의해 모션 벡터 차이), 각각의 참조 화상 리스트 및 참조 화상 리스트 사용에 대한 대응하는 참조 화상 인덱스가 각각의 PU마다 명시적으로 시그널링된다. 그러한 모드는 본 개시내용에서 고급 모션 벡터 예측(Advanced motion vector prediction, AMVP)라고 명명된다.

[0056] 시그널링이 2개의 참조 화상 리스트 중 하나가 사용될 것임을 지시할 때, PU는 하나의 샘플 블록으로부터 생성된다. 이는 '단방향 예측(uni-prediction)'이라고 지칭된다. 단방향 예측은 P-슬라이스들 및 B-슬라이스들 둘 다에 대해 이용가능하다.

[0057] 시그널링이 둘 다의 참조 화상 리스트들이 사용될 것임을 지시할 때, PU는 2개의 샘플 블록으로부터 생성된다. 이는 '양방향 예측(bi-prediction)'이라고 지칭된다. 양방향 예측은 B-슬라이스들에 대해서만 이용가능하다.

[0058] 다음의 텍스트는 HEVC에서 특정된 인터 예측 모드들에 대한 상세 사항들을 제공한다. 설명은 병합 모드로 시작될 것이다.

2.1.1. 참조 화상 리스트

[0060] HEVC에서, 인터 예측이라는 용어는 현재 디코딩된 화상 이외의 참조 화상들의 데이터 요소들(예를 들어, 샘플 값들 또는 모션 벡터들)로부터 도출된 예측을 나타내기 위해 사용된다. H.264/AVC에서와 같이, 화상은 다수의 참조 화상으로부터 예측될 수 있다. 인터 예측을 위해 사용되는 참조 화상들은 하나 이상의 참조 화상 리스트로 조직된다. 참조 인덱스는 리스트 내의 참조 화상을 중 어느 것이 예측 신호를 생성하기 위해 사용되어야 하는지를 식별한다.

[0061] 단일 참조 화상 리스트, List 0은 P 슬라이스를 위해 사용되고 2개의 참조 화상 리스트, List 0 및 List 1은 B 슬라이스들을 위해 사용된다. List 0/1에 포함된 참조 화상들은 캡처링/디스플레이 순서의 관점에서 과거 및 미래 화상들로부터의 것일 수 있다는 점에 유의해야 한다.

2.1.2 병합 모드

2.1.2.1 병합 모드에 대한 후보들의 도출

[0063] PU가 병합 모드를 사용하여 예측될 때, 병합 후보들 리스트 내의 엔트리를 가리키는 인덱스가 비트스트림으로부터 파싱되고 모션 정보를 검색하기 위해 사용된다. 이 리스트의 구성은 HEVC 표준에 특정되고 다음의 단계들의 시퀀스에 따라 요약될 수 있다:

[0064] 단계 1: 초기 후보들 도출

[0065] 단계 1.1: 공간 후보들 도출

[0066] 단계 1.2: 공간 후보들에 대한 중복 검사

[0067] 단계 1.3: 시간 후보들 도출

[0068] 단계 2: 추가적인 후보들 삽입

[0069] 단계 2.1: 양방향 예측 후보들의 생성

[0070] 단계 2.2: 제로 모션 후보들의 삽입

[0071] 이들 단계는 또한 도 1에 개략적으로 묘사되어 있다. 공간 병합 후보 도출을 위해, 5개의 상이한 위치에 위치되는 후보들 중에서 최대 4개의 병합 후보가 선택된다. 시간 병합 후보 도출을 위해, 2개의 후보 중에서 최대 하나의 병합 후보가 선택된다. 디코더에서 각각의 PU에 대한 일정한 수의 후보들이 가정되기 때문에, 단계 1로부터 획득된 후보들의 수가 슬라이스 헤더에서 시그널링되는 병합 후보의 최대 수(MaxNumMergeCand)에 도달하지 않을 때 추가적인 후보들이 생성된다. 후보들의 수가 일정하기 때문에, 최상의 병합 후보의 인덱스는 절단된 단항 이진화(truncated unary binarization, TU)를 사용하여 인코딩된다. CU의 크기가 8과 동등하다면, 현재 CU의 모든 PU들은 $2N \times 2N$ 예측 단위의 병합 후보 리스트와 동일한 단일 병합 후보 리스트를 공유한다.

[0072] [0073] 이하에서는, 위에 언급된 단계들과 연관된 동작들이 상세히 설명된다.

[0074] [0075] 도 1은 병합 후보들 리스트 구성을 위한 도출 프로세스의 예를 보여준다.

2.1.2.2. 공간 후보들 도출

[0076] 공간 병합 후보들의 도출에서는, 도 2에 묘사된 위치들에 위치된 후보들 중에서 최대 4개의 병합 후보가 선택된다. 도출 순서는 A_1, B_1, B_0, A_0 및 B_2 이다. 위치 B_2 는 위치 A_1, B_1, B_0, A_0 의 임의의 PU가 이용가능하지 않거나 (예를 들어, 그것이 다른 슬라이스 또는 타일에 속하기 때문에) 또는 인트라 코딩될 때에만 고려된다. 위치 A_1 에서의 후보가 추가된 후에, 나머지 후보들의 추가는 코딩 효율이 개선되도록 동일 모션 정보를 갖는 후보들이 리스트로부터 제외되는 것을 보장하는 중복 검사를 거친다. 계산 복잡성을 감소시키기 위해, 모든 가능한 후보 쌍들이 언급된 중복 검사에서 고려되지는 않는다. 대신에 도 3에서 화살표와 연결된 쌍들만이 고려되고, 중복 검사를 위해 사용되는 대응하는 후보가 동일한 모션 정보를 갖지 않는 경우에만 후보가 리스트에 추가된다. 중복 모션 정보의 다른 소스는 $2N \times 2N$ 과는 상이한 파티션들과 연관된 "제2 PU"이다. 예로서, 도 4는 $N \times 2N$ 및 $2N \times N$ 의 경우 각각에 대한 제2 PU를 묘사한다. 현재 PU가 $N \times 2N$ 으로 파티셔닝될 때, 위치 A_1 에서의 후보는 리스트 구성에 고려되지 않는다. 사실, 이 후보를 추가하는 것에 의해 동일한 모션 정보를 갖는 2개의 예측 단위로 이어질 것이고, 이는 코딩 단위에서 하나의 PU만을 갖도록 중복된다. 유사하게, 현재 PU가 $2N \times N$ 으로 파티셔닝될 때 위치 B_1 은 고려되지 않는다.

[0077] [0078] 도 2는 공간 병합 후보들의 위치들의 예를 보여준다.

[0079] [0080] 도 3은 공간 병합 후보들의 중복 검사를 위해 고려되는 후보 쌍들의 예를 보여준다.

[0081] 도 4는 $N \times 2N$ 및 $2N \times N$ 파티션들의 제2 PU에 대한 위치들의 예를 보여준다.

2.1.2.3. 시간 후보들 도출

[0082] 이 단계에서는, 단 하나의 후보만이 리스트에 추가된다. 특히, 이 시간 병합 후보의 도출에서는, 주어진 참조 화상 리스트 내의 현재 화상과 가장 작은 POC 차이를 갖는 화상에 속하는 동위치(co-located) PU에 기초하여 스케일링된 모션 벡터가 도출된다. 동위치 PU의 도출을 위해 사용될 참조 화상 리스트는 슬라이스 헤더에서 명시적으로 시그널링된다. 시간 병합 후보에 대한 스케일링된 모션 벡터는 도 5에서 점선으로 예시된 바와 같이 획득되고, 이는 POC 거리를 tb 및 td를 사용하여 동위치 PU의 모션 벡터로부터 스케일링되고, 여기서 tb는 현재

화상의 참조 화상과 현재 화상 간의 POC 차이인 것으로 정의되고, td 는 동위치 화상의 참조 화상과 동위치 화상 간의 POC 차이인 것으로 정의된다. 시간 병합 후보의 참조 화상 인덱스는 제로와 동등하게 설정된다. 스케일링 프로세스의 실제적인 실현은 HEVC 사양에서 설명된다. B-슬라이스에 대해, 2개의 모션 벡터 - 하나는 참조 화상 리스트 0에 대한 것이고 다른 하나는 참조 화상 리스트 1에 대한 것임 - 가 획득되고 조합되어 양방향 예측 병합 후보를 만든다.

[0082] 도 5는 시간 병합 후보에 대한 모션 벡터 스케일링의 예를 보여준다.

[0083] 참조 프레임에 속하는 동위치 PU(Y)에서, 시간 후보에 대한 위치는, 도 6에 묘사된 바와 같이, 후보들 C_0 및 C_1 사이에서 선택된다. 위치 C_0 에서의 PU가 이용가능하지 않거나, 인트라 코딩되거나, 현재 CTU 행의 외부에 있으면, 위치 C_1 이 사용된다. 그렇지 않으면, 위치 C_0 이 시간 병합 후보의 도출에서 사용된다.

[0084] 도 6은 시간 병합 후보 C_0 및 C_1 에 대한 후보 위치들의 예를 보여준다.

2.1.2.4. 추가적인 후보들 삽입

[0086] 공간 및 시간 병합 후보들 이외에, 2개의 추가적인 유형의 병합 후보들의 있다: 조합된 양방향 예측 병합 후보 및 제로 병합 후보. 조합된 양방향 예측 병합 후보들은 공간 및 시간 병합 후보들을 이용하여 생성된다. 조합된 양방향 예측 병합 후보는 B-슬라이스에 대해서만 사용된다. 조합된 양방향 예측 후보들은 초기 후보의 제1 참조 화상 리스트 모션 파라미터들을 다른 것의 제2 참조 화상 리스트 모션 파라미터들과 조합하는 것에 의해 생성된다. 이들 2개의 투플이 상이한 모션 가설을 제공한다면, 그것들은 새로운 양방향 예측 후보를 형성할 것이다. 예로서, 도 7은 $mvL0$ 및 $refIdxL0$ 또는 $mvL1$ 및 $refIdxL1$ 을 갖는, 원래의 리스트(왼쪽) 내의 2개의 후보가 최종 리스트(오른쪽)에 추가된 조합된 양방향 예측 병합 후보를 생성하기 위해 사용되는 경우를 묘사한다. 이를 추가적인 병합 후보를 생성하기 위해 고려되는 조합들에 관한 다수의 규칙이 존재한다.

[0087] 도 7은 조합된 양방향 예측 병합 후보의 예를 보여준다.

[0088] 병합 후보들 리스트 내의 나머지 엔트리들을 채우고 따라서 MaxNumMergeCand 용량에 달하도록 제로 모션 후보들이 삽입된다. 이들 후보는 제로 공간 변위 및 제로로부터 시작되고 새로운 제로 모션 후보가 리스트에 추가될 때마다 증가하는 참조 화상 인덱스를 갖는다. 이들 후보에 의해 사용되는 참조 프레임들의 수는 단방향 및 양방향 예측에 대해 각각 1 및 2이다. 마지막으로, 이들 후보에 대해 중복 검사가 수행되지 않는다.

2.1.2.5. 병렬 처리를 위한 모션 추정 영역들

[0090] 인코딩 프로세스의 속도를 높이기 위해, 모션 추정이 병렬로 수행될 수 있고, 그에 의해 주어진 영역 내부의 모든 예측 단위들에 대한 모션 벡터들이 동시에 도출된다. 공간 이웃으로부터의 병합 후보들의 도출은 병렬 처리를 방해할 수 있는데, 그 이유는 하나의 예측 단위가 그의 연관된 모션 추정이 완료될 때까지 인접한 PU로부터 모션 파라미터들을 도출할 수 없기 때문이다. 코딩 효율과 처리 레이턴시 간의 절충을 완화하기 위해, HEVC는 "log2_parallel_merge_level_minus2" 구문 요소 [1]을 사용하여 화상 파라미터 세트에서 그 크기가 시그널링되는 모션 추정 영역(MER)을 정의한다. MER이 정의될 때, 동일한 영역에 속하는 병합 후보들은 이용불가능한 것으로 표시되고 따라서 리스트 구성에서 고려되지 않는다.

2.1.3. AMVP

[0092] AMVP는 이웃 PU들과의 모션 벡터의 공간-시간 상관을 이용하고, 이는 모션 파라미터들의 명시적 송신을 위해 사용된다. 각각의 참조 화상 리스트에 대해, 모션 벡터 후보 리스트는 먼저 왼쪽, 위의 시간적으로 이웃 PU 위치들의 이용가능성을 검사하고, 중복 후보들을 제거하고 제로 벡터를 추가하여 후보 리스트를 일정한 길이가 되도록 만드는 것에 의해 구성된다. 그 후, 인코더는 후보 리스트로부터 최상의 예측자를 선택하고 선택된 후보를 지시하는 대응하는 인덱스를 송신할 수 있다. 병합 인덱스 시그널링과 유사하게, 최상의 모션 벡터 후보의 인덱스는 절단된 단항(truncated unary)을 사용하여 인코딩된다. 이 경우에 인코딩될 최대 값은 2이다(도 8 참조). 다음의 섹션들에서는, 모션 벡터 예측 후보의 도출 프로세스에 관한 상세 사항들이 제공된다.

2.1.3.1 AMVP 후보들의 도출

[0094] 도 8은 모션 벡터 예측 후보에 대한 도출 프로세스를 요약한다.

[0095] 모션 벡터 예측에서는, 2가지 유형의 모션 벡터 후보가 고려된다: 공간 모션 벡터 후보 및 시간 모션 벡터 후보. 공간 모션 벡터 후보 도출을 위해, 도 2에 묘사된 바와 같이 5개의 상이한 위치에 위치되는 각각의 PU의

모션 벡터에 기초하여 2개의 모션 벡터 후보가 결국 도출된다.

[0096] 시간 모션 벡터 후보 도출을 위해, 2개의 상이한 동위치 위치에 기초하여 도출되는 2개의 후보로부터 하나의 모션 벡터 후보가 선택된다. 공간-시간 후보들의 제1 리스트가 만들어진 후에, 리스트 내의 중복된 모션 벡터 후보들이 제거된다. 잠재적 후보들의 수가 2보다 크면, 연관된 참조 화상 리스트 내의 참조 화상 인덱스가 1보다 큰 모션 벡터 후보들이 리스트로부터 제거된다. 공간-시간 모션 벡터 후보들의 수가 2보다 작으면, 추가적인 제로 모션 벡터 후보들이 리스트에 추가된다.

2.1.3.2. 공간 모션 벡터 후보들

[0098] 공간 모션 벡터 후보들의 도출에서는, 도 2에 묘사된 바와 같은 위치들에 위치된 PU들로부터 도출되는 5개의 잠재적 후보들 중에서 최대 2개의 후보가 고려되고, 해당 위치들은 모션 병합의 위치들과 동일하다. 현재 PU의 왼쪽에 대한 도출 순서는 A0, A1, 및 스케일링된 A0, 스케일링된 A1로서 정의된다. 현재 PU의 위쪽에 대한 도출 순서는 B0, B1, B2, 및 스케일링된 B0, 스케일링된 B1, 스케일링된 B2로서 정의된다. 따라서, 각각의 측에 대해 모션 벡터 후보로서 사용될 수 있는 4개의 경우가 있고, 여기서 2개의 경우는 공간 스케일링을 사용할 필요가 없고, 2개의 경우는 공간 스케일링이 사용되는 것이다. 이 4개의 상이한 경우는 다음과 같이 요약된다.

[0099] 공간 스케일링 없음

[0100] (1) 동일한 참조 화상 리스트, 및 동일한 참조 화상 인덱스(동일한 POC)

[0101] (2) 상이한 참조 화상 리스트, 및 동일한 참조 화상(동일한 POC)

[0102] 공간 스케일링

[0103] (3) 동일한 참조 화상 리스트이지만, 상이한 참조 화상(상이한 POC)

[0104] (4) 상이한 참조 화상 리스트, 및 상이한 참조 화상(상이한 POC)

[0105] 공간 스케일링이 없는 경우들이 먼저 검사되고 이어서 공간 스케일링이 후속된다. 참조 화상 리스트에 관계없이 이웃 PU의 참조 화상과 현재 PU의 참조 화상 간에 POC가 상이할 때 공간 스케일링이 고려된다. 왼쪽 후보들의 모든 PU들이 이용가능하지 않거나 인트라 코딩되면, 왼쪽 및 위의 MV 후보들의 별별 도출을 돋기 위해 위의 모션 벡터에 대한 스케일링이 허용된다. 그렇지 않으면, 위의 모션 벡터에 대해 공간 스케일링이 허용되지 않는다.

[0106] 도 9는 공간 모션 벡터 후보에 대한 모션 벡터 스케일링의 예를 보여준다.

[0107] 공간 스케일링 프로세스에서는, 이웃 PU의 모션 벡터는 도 9에 묘사된 바와 같이 시간 스케일링에 대한 것과 유사한 방식으로 스케일링된다. 주요 차이는 현재 PU의 인덱스 및 참조 화상 리스트가 입력으로서 주어진다는 것이다; 실제 스케일링 프로세스는 시간 스케일링의 것과 동일하다.

2.1.3.3. 시간 모션 벡터 후보들

[0109] 참조 화상 인덱스 도출 이외에, 시간 병합 후보들의 도출을 위한 모든 프로세스들은 공간 모션 벡터 후보들의 도출에 대한 것과 동일하다(도 6 참조). 참조 화상 인덱스는 디코더에 시그널링된다.

2.2. JEM에서의 새로운 인터 예측 방법들

2.2.1. 서브-CU 기반 모션 벡터 예측

[0112] QTBT를 갖는 JEM에서, 각각의 CU는 각각의 예측 방향에 대해 많아야 하나의 모션 파라미터 세트를 가질 수 있다. 큰 CU를 서브-CU들로 분할하고 큰 CU의 모든 서브-CU들에 대한 모션 정보를 도출하는 것에 의해 인코더에서 2개의 서브-CU 레벨 모션 벡터 예측 방법이 고려된다. SbTMVP(sub-block temporal motion vector prediction)라고도 지칭되는 대안적인 시간 모션 벡터 예측(Alternative temporal motion vector prediction, ATMVP) 방법은, 각각의 CU가 동위치 참조 화상에서의 현재 CU보다 작은 다수의 블록으로부터 다수의 모션 정보 세트를 폐치하는 것을 허용한다. 공간-시간 모션 벡터 예측(spatial-temporal motion vector prediction, STMVP) 방법에서는 서브-CU들의 모션 벡터들이 시간 모션 벡터 예측자 및 공간 이웃 모션 벡터를 사용하여 재귀적으로 도출된다.

[0113] 서브-CU 모션 예측을 위한 더 정확한 모션 필드를 보존하기 위해, 참조 프레임들에 대한 모션 압축은 현재 디스에이블된다.

[0114] 도 10은 CU에 대한 ATMVP 모션 예측의 예를 보여준다.

2.2.1.1. 대안적인 시간 모션 벡터 예측

[0116] 대안적인 시간 모션 벡터 예측(ATMVP) 방법에서는, 모션 벡터들 시간 모션 벡터 예측(TMVP)은 현재 CU보다 작은 블록들로부터 다수의 모션 정보 세트(모션 벡터들 및 참조 인덱스들을 포함함)를 페치하는 것에 의해 수정된다. 일부 실시예들에서, 서브-CU들은 정사각형 $N \times N$ 블록들이다(N 은 디폴트로 4로 설정된다).

[0117] ATMVP는 2개의 단계에서 CU 내의 서브-CU들의 모션 벡터들을 예측한다. 제1 단계는 소위 시간 벡터로 참조 화상 내의 대응하는 블록을 식별하는 것이다. 참조 화상은 모션 소스 화상이라고 불린다. 제2 단계는 현재 CU를 서브-CU들로 분할하고 각각의 서브-CU에 대응하는 블록으로부터 각각의 서브-CU의 참조 인덱스들뿐만 아니라 모션 벡터들을 획득하는 것이다.

[0118] 제1 단계에서, 참조 화상 및 대응하는 블록은 현재 CU의 공간 이웃 블록들의 모션 정보에 의해 결정된다. 이웃 블록들의 반복 스캐닝 프로세스를 피하기 위해, 현재 CU의 병합 후보 리스트 내의 제1 병합 후보가 사용된다. 제1 이용가능한 모션 벡터뿐만 아니라 그의 연관된 참조 인덱스는 시간 벡터 및 모션 소스 화상에 대한 인덱스로 설정된다. 이러한 방식으로, ATMVP에서, 대응하는 블록은 TMVP와 비교하여 더 정확하게 식별될 수 있고, 여기서 대응하는 블록(때때로 동위치 블록이라고 불림)은 항상 현재 CU에 대해 오른쪽 하단 또는 중심 위치에 있다.

[0119] 제2 단계에서, 현재 CU의 좌표에 시간 벡터를 추가하는 것에 의해, 모션 소스 화상에서의 시간 벡터에 의해 서브-CU의 대응하는 블록이 식별된다. 각각의 서브-CU에 대해, 그의 대응하는 블록(중심 샘플을 커버하는 가장 작은 모션 그리드)의 모션 정보는 서브-CU에 대한 모션 정보를 도출하기 위해 사용된다. 대응하는 $N \times N$ 블록의 모션 정보가 식별된 후에, 그것은 HEVC의 TMVP와 동일한 방식으로, 현재 서브-CU의 모션 벡터들 및 참조 인덱스들로 변환되고, 여기서 모션 스케일링 및 다른 절차들이 적용된다. 예를 들어, 디코더는 저지연 조건(즉, 현재 화상의 모든 참조 화상들의 POC들이 현재 화상의 POC보다 작은 것)이 충족되는지를 검사하고, 어쩌면 모션 벡터 MV_x (참조 화상 리스트 X에 대응하는 모션 벡터)를 사용하여 각각의 서브-CU에 대해 모션 벡터 MV_y (X 는 0 또는 1과 동등하고 Y 는 1-X와 동등함)를 예측한다.

2.2.1.2. 공간-시간 모션 벡터 예측(STMVP)

[0121] 이 방법에서, 서브-CU들의 모션 벡터들은, 래스터 스캔 순서에 따라, 재귀적으로 도출된다. 도 11은 이 개념을 예시한다. 4개의 4×4 서브-CU들 A, B, C, 및 D를 포함하는 8×8 CU를 고려해보자. 현재 프레임 내의 이웃 4×4 블록들은 a, b, c, 및 d로 라벨링된다.

[0122] 서브-CU A에 대한 모션 도출은 그것의 2개의 공간 이웃을 식별하는 것에 의해 시작된다. 제1 이웃은 서브-CU A 위의 $N \times N$ 블록이다(블록 c). 이 블록 c가 이용가능하지 않거나 인트라 코딩되면 서브-CU A 위의 다른 $N \times N$ 블록들이 검사된다(블록 c에서 시작하여, 왼쪽에서 오른쪽으로). 제2 이웃은 서브-CU A의 왼쪽의 블록(블록 b)이다. 블록 b가 이용가능하지 않거나 인트라 코딩되면 서브-CU A의 왼쪽의 다른 블록들이 검사된다(블록 b에서 시작하여, 위에서 아래로). 각각의 리스트에 대해 이웃 블록들로부터 획득된 모션 정보는 주어진 리스트에 대해 제1 참조 프레임으로 스케일링된다. 다음으로, HEVC에서 특정된 바와 같은 TMVP 도출의 동일한 절차를 따르는 것에 의해 서브-블록 A의 시간 모션 벡터 예측자(TMVP)가 도출된다. 위치 D에서의 동위치 블록의 모션 정보는 그에 따라 폐치되고 스케일링된다. 마지막으로, 모션 정보를 검색하고 스케일링한 후에, 모든 이용가능한 모션 벡터(최대 3개)가 각각의 참조 리스트에 대해 개별적으로 평균된다. 평균된 모션 벡터는 현재 서브-CU의 모션 벡터로서 할당된다.

2.2.1.3. 서브-CU 모션 예측 모드 시그널링

[0124] 서브-CU 모드들은 추가적인 병합 후보들로서 인에이블되고 모드들을 시그널링하기 위해 필요한 추가적인 구문 요소가 없다. 각각의 CU의 병합 후보들 리스트에 2개의 추가적인 병합 후보가 추가되어 ATMVP 모드 및 STMVP 모드를 표현한다. 시퀀스 파라미터 세트가 ATMVP 및 STMVP가 인에이블되는 것을 지시한다면, 최대 7개의 병합 후보가 사용된다. 추가적인 병합 후보들의 인코딩 로직은 HM에서의 병합 후보들에 대한 것과 동일하고, 이는 P 또는 B 슬라이스 내의 각각의 CU에 대해, 2개의 추가적인 병합 후보에 대해 2개 이상의 RD 검사가 필요함을 의미한다.

[0125] JEM에서, 병합 인덱스의 모든 빈들은 CABAC에 의해 컨텍스트 코딩된다. HEVC에서는, 제1 빈만이 컨텍스트 코딩

되고 나머지 빈들은 컨텍스트 바이-패스(context by-pass) 코딩된다.

2.2.2. 적응 모션 벡터 차이 해상도

[0126] HEVC에서, 모션 벡터 차이(motion vector difference, MVD)(모션 벡터와 PU의 예측된 모션 벡터 간)들은 use_integer_mv_flag가 슬라이스 헤더에서 0과 동등할 때 쿼터 루마 샘플들의 단위로 시그널링된다. JEM에서, 로컬 적응 모션 벡터 해상도(locally adaptive motion vector resolution, LAMVR)가 도입된다. JEM에서, MVD는 쿼터 루마 샘플들, 정수 루마 샘플들 또는 4 루마 샘플들의 단위로 코딩될 수 있다. MVD 해상도는 코딩 단위(CU) 레벨에서 제어되고, MVD 해상도 플래그들은 적어도 하나의 비-제로 MVD 성분을 갖는 각각의 CU에 대해 조건부로 시그널링된다.

[0127] 적어도 하나의 비-제로 MVD 성분을 갖는 CU에 대해, 제1 플래그는 CU에서 쿼터 루마 샘플 MV 정밀도가 사용되는지를 지시하기 위해 시그널링된다. 제1 플래그(1과 동등)가 쿼터 루마 샘플 MV 정밀도가 사용되지 않는다는 것을 지시할 때, 정수 루마 샘플 MV 정밀도 또는 4 루마 샘플 MV 정밀도가 사용되는지를 지시하기 위해 다른 플래그가 시그널링된다.

[0128] CU의 제1 MVD 해상도 플래그가 제로이거나, CU에 대해 코딩되지 않을 때(CU에서의 모든 MVD들이 제로임을 의미함), 쿼터 루마 샘플 MV 해상도가 CU에 대해 사용된다. CU가 정수-루마 샘플 MV 정밀도 또는 4-루마-샘플 MV 정밀도를 사용할 때, CU에 대한 AMVP 후보 리스트 내의 MVP들은 대응하는 정밀도로 라운딩된다.

[0129] 인코더에서, 어느 MVD 해상도가 CU에 대해 사용되어야 하는지를 결정하기 위해 CU-레벨 RD 검사들이 사용된다. 즉, CU-레벨 RD 검사는 각각의 MVD 해상도에 대해 3회 수행된다. 인코더 속도를 가속화하기 위해, 다음의 인코딩 스Kim들이 JEM에서 적용된다.

[0130] 정상 쿼터 루마 샘플 MVD 해상도를 갖는 CU의 RD 검사 동안, 현재 CU의 모션 정보(정수 루마 샘플 정확도)가 저장된다. 저장된 모션 정보(라운딩 후)는 시간 소모적인 모션 추정 프로세스가 3회 중복되지 않도록 정수 루마 샘플 및 4 루마 샘플 MVD 해상도를 갖는 동일한 CU에 대한 RD 검사 동안 추가의 작은 범위 모션 벡터 리파인먼트를 위한 시작점으로서 사용된다.

[0131] 4 루마 샘플 MVD 해상도를 갖는 CU의 RD 검사가 조건부로 호출된다. CU에 대해, RD 비용 정수 루마 샘플 MVD 해상도가 쿼터 루마 샘플 MVD 해상도보다 훨씬 클 때, CU에 대한 4 루마 샘플 MVD 해상도의 RD 검사는 스Kim된다.

[0132] 인코딩 프로세스는 도 12에 도시되어 있다. 먼저, 1/4 pel MV가 테스트되고, RD 비용이 계산되고 RDCost0으로 표시되고, 그 후 정수 MV가 테스트되고 RD 비용이 RDCost1로 표시된다. $RDCost1 < th * RDCost0$ (여기서 th는 양의 값임)이면, 4-pel MV가 테스트된다; 그렇지 않으면, 4-pel MV가 스Kim된다. 기본적으로, 모션 정보 및 RD 비용 등은 정수 또는 4-pel MV를 검사할 때 1/4 pel MV에 대해 이미 알려져 있고, 이는 정수 또는 4-pel MV의 인코딩 프로세스의 속도를 높이기 위해 재사용될 수 있다.

2.2.3. 삼각형 예측 모드

[0133] 삼각형 예측 모드(triangular prediction mode, TPM)의 개념은 모션 보상 예측을 위한 새로운 삼각형 파티션을 도입하는 것이다. 도 13에 도시된 바와 같이, 이는 CU를 대각선 또는 역 대각선 방향으로 2개의 삼각형 예측 단위로 분할한다. CU 내의 각각의 삼각형 예측 단위는 단방향 예측 후보 리스트로부터 도출되는 그 자신의 단방향 예측 모션 벡터 및 참조 프레임 인덱스를 사용하여 인터-예측된다. 삼각형 예측 단위들을 예측한 후에 대각선 예지에 대해 적응 가중 프로세스가 수행된다. 그 후, 변환 및 양자화 프로세스가 전체 CU에 적용된다. 이 모드는 스Kim 및 병합 모드들에만 적용된다는 점에 유의한다.

2.2.3.1 TPM을 위한 단방향 예측 후보 리스트

[0134] 단방향 예측 후보 리스트는 5개의 단방향 예측 모션 벡터 후보로 구성된다. 그것은, 도 14에 도시된 바와 같이, 5개의 공간 이웃 블록(1 내지 5) 및 2개의 시간 동위치 블록들(6 내지 7)을 포함하는 7개의 이웃 블록으로부터 도출된다. 도 14는 이웃 블록들의 위치의 예를 보여준다. 7개의 이웃 블록의 모션 벡터들이 수집되어 단방향 예측 모션 벡터들, 양방향 예측 모션 벡터들의 L0 모션 벡터, 양방향 예측 모션 벡터들의 L1 모션 벡터, 및 양방향 예측 모션 벡터들의 L0 및 L1 모션 벡터들의 평균된 모션 벡터의 순서에 따라 단방향 예측 후보 리스트에 넣어진다. 후보들의 수가 5보다 작으면, 제로 모션 벡터가 리스트에 추가된다.

[0135] 더 구체적으로, 다음의 단계들이 수반된다:

- [0139] 어떠한 프루닝 동작들도 없이 A1, B1, B0, A0, B2, Col 및 Col2(도 14의 블록 1-7에 대응함)로부터 모션 후보들을 획득한다.
- [0140] 변수 numCurrMergeCand = 0으로 설정한다
- [0141] numCurrMergeCand가 5보다 작고 A1, B1, B0, A0, B2, Col 및 Col2로부터 도출된 각각의 모션 후보들에 대해, 모션 후보가 단방향 예측이면(List 0 또는 List 1 중 어느 하나로부터의), 그것은 병합 리스트에 추가되고 numCurrMergeCand가 1만큼 증가된다. 그러한 추가된 모션 후보들은 '원래 단방향 예측된 후보'라고 명명된다.
- [0142] 풀 프루닝(full pruning)이 적용된다.
- [0143] numCurrMergeCand가 5보다 작고 A1, B1, B0, A0, B2, Col 및 Col2로부터 도출된 각각의 모션 후보들에 대해, 모션 후보가 양방향 예측이면, List 0으로부터의 모션 정보가 병합 리스트에 추가되고(즉, List 0으로부터의 단방향 예측으로 수정됨), numCurrMergeCand는 1만큼 증가된다. 그러한 추가된 모션 후보들은 '절단된 List0-예측된 후보'라고 명명된다.
- [0144] 풀 프루닝이 적용된다.
- [0145] numCurrMergeCand가 5보다 작고 A1, B1, B0, A0, B2, Col 및 Col2로부터 도출된 각각의 모션 후보들에 대해, 모션 후보가 양방향 예측이면, List 1로부터의 모션 정보가 병합 리스트에 추가되고(즉, List 1로부터의 단방향 예측으로 수정됨), numCurrMergeCand는 1만큼 증가된다. 그러한 추가된 모션 후보들은 '절단된 List1-예측된 후보'라고 명명된다.
- [0146] 풀 프루닝이 적용된다.
- [0147] numCurrMergeCand가 5보다 작고 A1, B1, B0, A0, B2, Col 및 Col2로부터 도출된 각각의 모션 후보들에 대해, 모션 후보가 양방향 예측이면,
- [0148] List 0 참조 화상의 슬라이스 QP가 List 1 참조 화상의 슬라이스 QP보다 작으면, List 1의 모션 정보는 List 0 참조 화상으로 먼저 스케일링되고, 2개의 MV의 평균(하나는 원래의 List 0으로부터의 것이고, 다른 하나는 List 1로부터의 스케일링된 MV임)은 병합 리스트에 추가되고, 즉 List 0 모션 후보로부터의 평균된 단방향 예측이고 numCurrMergeCand가 1만큼 증가된다.
- [0149] 그렇지 않으면, List 0의 모션 정보가 먼저 List 1 참조 화상으로 스케일링되고, 2개의 MV의 평균(하나는 원래의 List 1로부터의 것이고, 다른 하나는 List 0으로부터의 스케일링된 MV임)이 병합 리스트에 추가되고, 즉 List 1 모션 후보로부터의 평균된 단방향 예측이고 numCurrMergeCand가 1만큼 증가된다.
- [0150] 풀 프루닝이 적용된다.
- [0151] numCurrMergeCand가 5보다 작으면, 제로 모션 벡터 후보들이 추가된다.
- [0152] **2.2.3.2. 적응 가중 프로세스**
- [0153] 각각의 삼각형 예측 단위를 예측한 후에, 2개의 삼각형 예측 단위들 사이의 대각선 애지에 적응 가중 프로세스가 적용되어 전체 CU에 대한 최종 예측을 도출한다. 2개의 가중치 그룹은 다음과 같이 정의된다:
- [0154] 제1 가중치 그룹: {7/8, 6/8, 4/8, 2/8, 1/8} 및 {7/8, 4/8, 1/8}은 각각 휘도(luminance) 및 색차(chrominance) 샘플들에 대해 사용되고;
- [0155] 제2 가중치 그룹: {7/8, 6/8, 5/8, 4/8, 3/8, 2/8, 1/8} 및 {6/8, 4/8, 2/8}은 각각 휘도 및 색차 샘플들에 대해 사용되고;
- [0156] 가중치 그룹은 2개의 삼각형 예측 단위의 모션 벡터들의 비교에 기초하여 선택된다. 제2 가중치 그룹은 2개의 삼각형 예측 단위의 참조 화상들이 서로 상이하거나 그들의 모션 벡터 차이가 16 픽셀들보다 클 때 사용된다. 그렇지 않으면, 제1 가중치 그룹이 사용된다. 예가 도 15에 도시되어 있다. 도 15는 제1 가중치 그룹을 적용하는 CU의 예를 보여준다.
- [0157] **2.2.3.3. 모션 벡터 저장**
- [0158] 도 16은 모션 벡터 저장의 예를 보여준다. 삼각형 예측 단위들의 모션 벡터들(도 16에서 Mv1 및 Mv2)은 4×4 그리드들로 저장된다. 각각의 4×4 그리드에 대해, CU에서의 4×4 그리드의 위치에 따라 단방향 예측 또는 양방향 예측 모션 벡터가 저장된다. 도 16에 도시된 바와 같이, 비-가중 영역에 위치되는(즉, 대각선 애지에 위

치되지 않은) 4×4 그리드에 대해 Mv1 또는 Mv2 중 어느 하나인 단방향 예측 모션 벡터가 저장된다. 다른 한편으로, 가중 영역에 위치된 4×4 그리드에 대해 양방향 예측 모션 벡터가 저장된다. 양방향 예측 모션 벡터는 다음의 규칙에 따라 Mv1 및 Mv2로부터 도출된다:

- [0159] Mv1 및 Mv2가 상이한 방향들(L0 또는 L1)로부터의 모션 벡터를 갖는 경우, Mv1 및 Mv2는 단순히 조합되어 양방향 예측 모션 벡터를 형성한다.
- [0160] Mv1 및 Mv2 둘 다가 동일한 L0(또는 L1) 방향으로부터의 것인 경우,
- [0161] Mv2의 참조 화상이 L1(또는 L0) 참조 화상 리스트 내의 화상과 동일하면, Mv2는 화상으로 스케일링된다. Mv1 및 스케일링된 Mv2가 조합되어 양방향 예측 모션 벡터를 형성한다.
- [0162] Mv1의 참조 화상이 L1(또는 L0) 참조 화상 리스트 내의 화상과 동일하면, Mv1은 화상으로 스케일링된다. 스케일링된 Mv1 및 Mv2가 조합되어 양방향 예측 모션 벡터를 형성한다.
- [0163] 그렇지 않으면, 가중 영역에 대해 Mv1만이 저장된다.
- [0164] **2.2.3.4. 삼각형 예측 모드(TPM)의 시그널링**
- [0165] TPM이 사용되는지를 지시하는 1 비트 플래그가 먼저 시그널링될 수 있다. 그 후, 2개의 분할 패턴(도 13에 묘사된 바와 같은)의 지시, 및 2개의 파티션 각각에 대한 선택된 병합 인덱스들이 추가로 시그널링된다.
- [0166] TPM은 병합 모드의 특수 서브-모드로서 취급된다는 점에 유의한다. 따라서, 병합 플래그가 먼저 시그널링된다. 병합 플래그가 참일 때, 블록이 TPM을 선택하는지가 추가로 시그널링될 수 있다.
- [0167] 다음의 표(VVC 코덱 사양의 현재 버전을 참조함)는 현재 구문을 수정하는(볼드체 엔트리들이 변화를 나타냄) 예를 보여준다.

[0168]

7.3.4.6 코딩 단위 구문

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	서술자
if(slice_type != I) {	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
...	
} else { /* MODE_INTER */	
if(cu_skip_flag[x0][y0] == 0) {	
merge_flag [x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
merge_data (x0, y0, cbWidth, cbHeight)	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_10_active_minus1 > 0)	
ref_idx_10 [x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_10_flag [x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_11_active_minus1 > 0)	
ref_idx_11 [x0][y0]	ae(v)
if(mvd_11_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	

[0169]

MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
}	else {
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp_l1_flag [x0][y0]	ae(v)
}	else {
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	
if(sps_amvr_enabled_flag && inter_affine_flag == 0 &&	
(MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0	
MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0))	
amvr_mode [x0][y0]	ae(v)
if(sps_gbi_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI &&	
cbWidth * cbHeight >= 256)	
gbi_idx [x0][y0]	ae(v)
}	
}	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0)	
cu_cb	ae(v)
if(cu_cb)	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	
}	

[0170]

[0171]

7.3.4.8 병합 테이터 구문

코드	서술자
merge_data(x0, y0, cbWidth, cbHeight) {	
mmvd_flag[x0][y0]	ae(v)
if(mmvd_flag[x0][y0] == 1) {	
mmvd_merge_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)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if(sps_mh_intra_enabled_flag && cu_skip_flag[x0][y0] == 0 &&	
(cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
mh_intra_flag[x0][y0]	ae(v)
if(mh_intra_flag[x0][y0]) {	
if(cbWidth <= 2 * cbHeight cbHeight <= 2 * cbWidth)	
mh_intra_luma_mpm_flag[x0][y0]	ae(v)
if(mh_intra_luma_mpm_flag[x0][y0])	
mh_intra_luma_mpm_idx[x0][y0]	ae(v)
}	
}	
if(sps_triangle_enabled_flag && slice_type == B && cbWidth * cbHeight >= 16)	
merge_triangle_flag[x0][y0]	ae(v)
if(merge_triangle_flag[x0][y0])	
merge_triangle_idx[x0][y0]	ae(v)
else if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
}	
}	

[0172]

2.2.3.4.1 TPM 플래그의 시그널링

[0173]

하나의 루마 블록의 폭 및 높이를 각각 W 및 H로 표시하자. W*H < 64이면, 삼각형 예측 모드가 디스에이블된다.

[0174]

하나의 블록이 아핀 모드로 코딩될 때, 삼각형 예측 모드가 또한 디스에이블된다.

[0175]

하나의 블록이 병합 모드로 코딩될 때, 그 블록에 대해 삼각형 예측 모드가 인에이블되는지 또는 디스에이블되는지를 지시하기 위해 1 비트 플래그가 시그널링될 수 있다.

[0176]

플래그는 다음 수학식에 기초하여 3개의 컨텍스트로 코딩된다:

[0177]

Ctx 인덱스 = ((이용가능한 왼쪽 블록 L && L이 TPM으로 코딩됨?)) 1: 0)

[0178]

+ ((이용가능한 위의 블록 A && A가 TPM으로 코딩됨?)) 1: 0);

[0179]

도 17은 TPM 플래그 코딩에서 컨텍스트 선택을 위해 사용되는 이웃 블록들(A 및 L)의 예를 보여준다.

[0180]

2.2.3.4.2. 2개의 분할 패턴(도 13에 묘사된 바와 같은)의 지시, 및 2개의 파티션 각각에 대한 선택된 병합 인덱스들의 시그널링

[0181]

분할 패턴들, 2개의 파티션의 병합 인덱스들은 공동으로 코딩된다는 점에 유의한다. [5]에서, 2개의 파티션이

동일한 참조 인덱스를 사용할 수 없는 것이 제한된다. 따라서, $2(\text{분할 패턴}) * N(\text{병합 후보들의 최대 수}) * (N-1)$ 개의 가능성이 있고, 여기서 N 은 5로 설정된다. 하나의 지시가 코딩되고, 파티션 유형 간의 맵핑에서, 2개의 병합 인덱스가 아래에 정의된 어레이로부터 도출된다:

```
const uint8_t g_TriangleCombination[TRIANGLE_MAX_NUM_CANDS][3] = {
    { { 0, 1, 0 }, { 1, 0, 1 }, { 1, 0, 2 }, { 0, 0, 1 }, { 0, 2, 0 } },
    { { 1, 0, 3 }, { 1, 0, 4 }, { 1, 1, 0 }, { 0, 3, 0 }, { 0, 4, 0 } },
    { { 0, 0, 2 }, { 0, 1, 2 }, { 1, 1, 2 }, { 0, 0, 4 }, { 0, 0, 3 } },
    { { 0, 1, 3 }, { 0, 1, 4 }, { 1, 1, 4 }, { 1, 1, 3 }, { 1, 2, 1 } },
    { { 1, 2, 0 }, { 0, 2, 1 }, { 0, 4, 3 }, { 1, 3, 0 }, { 1, 3, 2 } },
    { { 1, 3, 4 }, { 1, 4, 0 }, { 1, 3, 1 }, { 1, 2, 3 }, { 1, 4, 1 } },
    { { 0, 4, 1 }, { 0, 2, 3 }, { 1, 4, 2 }, { 0, 3, 2 }, { 1, 4, 3 } },
    { { 0, 3, 1 }, { 0, 2, 4 }, { 1, 2, 4 }, { 0, 4, 2 }, { 0, 3, 4 } };
}
```

[0183] 파티션 유형(45도 또는 135도) = g_TriangleCombination[시그널링된 지시][0];

[0184] 후보 A의 병합 인덱스 = g_TriangleCombination[시그널링된 지시][1];

[0185] 후보 B의 병합 인덱스 = g_TriangleCombination[시그널링된 지시][2];

[0186] 2개의 모션 후보 A 및 B가 도출되면, 2개의 파티션의 (PU1 및 PU2) 모션 정보가 A 또는 B 중 어느 하나로부터 설정될 수 있다. PU1이 병합 후보 A 또는 B의 모션 정보를 사용하는지는 2개의 모션 후보의 예측 방향들에 의존한다. 표 1은 2개의 도출된 모션 후보 A 및 B, 그리고 2개의 파티션 간의 관계를 보여준다.

[0187] 표 1: 도출된 2개의 병합 후보(A, B)로부터의 파티션의 모션 정보의 도출

A의 예측 방향	B의 예측 방향	PU1의 모션 정보	PU2의 모션 정보
L0	L0	A (L0)	B (L0)
L1	L1	B (L1)	A (L1)
L0	L1	A (L0)	B (L1)
L1	L0	B (L0)	A (L1)

[0188] 2.2.3.4.3. 지시(`merge_triangle_idx`로 표시됨)의 엔트로피 코딩

[0189] `merge_triangle_idx`는 범위 [0, 39] 내에 있다(언급된 수 포함). K차 EG(Exponential Golomb) 코드가 `merge_triangle_idx`의 이진화를 위해 사용되고, 여기서 K는 1로 설정된다.

[0190] K차 EG

[0191] 더 적은 수의 비트들로 더 큰 수들을 인코딩하기 위해(더 작은 수들을 인코딩하기 위해 더 많은 비트들을 사용하는 대가로), 이는 음이 아닌 정수 파라미터 k를 사용하여 일반화될 수 있다. 차수-k exp-Golomb 코드로 음이

아닌 정수 x 를 인코딩하기 위해:

[0194] 위에 기술된 차수-0 exp-Golomb 코드를 사용하여 $\lfloor x/2^k \rfloor$ 를 인코딩하고, 그 후

[0195] 이진으로 $x \bmod 2^k$ 를 인코딩한다

[0196] 표 2: Exp-Golomb-k 코딩 예들

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

[0197]

2.2.4. 오버랩 블록 모션 보상

[0199] 오버랩 블록 모션 보상(OBMC)은 H.263에서 이전에 사용되었다. JEM에서는, H.263에서와 달리, OBMC는 CU 레벨에서의 구문을 사용하여 스위치 온 및 오프될 수 있다. JEM에서 OBMC가 사용될 때, OBMC는 CU의 오른쪽 및 하단 경계들을 제외한 모든 모션 보상(MC) 블록 경계들에 대해 수행된다. 더욱이, 그것은 루마 성분 및 크로마 성분 둘 다에 적용된다. JEM에서, MC 블록은 코딩 블록에 대응한다. CU가 서브-CU 모드(서브-CU 병합, 아핀 및 FRUC 모드를 포함함)로 코딩될 때, CU의 각각의 서브-블록은 MC 블록이다. 균일한 방식으로 CU 경계들을 처리하기 위해, OBMC는 모든 MC 블록 경계들에 대해 서브-블록 레벨에서 수행되고, 여기서 서브-블록 크기는 도 18에 예시된 바와 같이 4×4 와 동등하게 설정된다.

[0200] OBMC가 현재 서브-블록에 적용될 때, 현재 모션 벡터들 이외에, 4개의 연결된 이웃 서브-블록들의 모션 벡터들을, 이용가능하고 현재 모션 벡터와 동일하지 않다면, 또한 현재 서브-블록에 대한 예측 블록을 도출하기 위해 사용된다. 다수의 모션 벡터에 기초한 이들 다수의 예측 블록은 현재 서브-블록의 최종 예측 신호를 생성하기 위해 조합된다.

[0201] 이웃 서브-블록의 모션 벡터들에 기초한 예측 블록은 P_N 으로서 표시되고, N 은 이웃하는 위, 아래, 왼쪽 및 오른쪽 서브-블록에 대한 인덱스를 지시하고, 현재 서브-블록의 모션 벡터에 기초한 예측 블록은 P_C 로서 표시된다. P_N 이 현재 서브-블록에 대해 동일한 모션 정보를 포함하는 이웃 서브-블록의 모션 정보에 기초할 때, OBMC는 P_N 으로부터 수행되지 않는다. 그렇지 않으면, 모든 P_N 샘플이 P_C 내의 동일한 샘플에 추가되는데, 즉, P_N 의 4개의 행/열이 P_C 에 추가된다. 가중치 $\{1/4, 1/8, 1/16, 1/32\}$ 은 P_N 을 위해 사용되고 가중치 $\{3/4, 7/8, 15/16, 31/32\}$ 은 P_C 를 위해 사용된다. 예외는 작은 MC 블록들이고(즉, 코딩 블록의 높이 또는 폭이 4와 동등하거나 CU가 서브-CU 모드로 코딩될 때), 이에 대해 P_N 의 2개의 행/열만이 P_C 에 추가된다. 이 경우 가중치 $\{1/4, 1/8\}$ 은 P_N 을 위해 사용되고 가중치 $\{3/4, 7/8\}$ 은 P_C 를 위해 사용된다. 수직으로(수평으로) 이웃 서브-블록의 모션 벡터들에 기초하여 생성되는 P_N 에 대해, P_N 의 동일한 행(열) 내의 샘플들이 동일한 가중치를 이용하여 P_C 에 추가된다.

[0202] JEM에서, 256개 루마 샘플보다 적거나 그와 동등한 크기를 갖는 CU에 대해, 현재 CU에 대해 OBMC가 적용되는지 여부를 지시하기 위해 CU 레벨 플래그가 시그널링된다. 256개 루마 샘플보다 큰 크기를 갖거나 AMVP 모드로 코딩되지 않은 CU들에 대해, OBMC가 디폴트로 적용된다. 인코더에서, OBMC가 CU에 적용될 때, 모션 추정 스테이지 동안 그의 영향이 고려된다. 상단 이웃 블록 및 왼쪽 이웃 블록의 모션 정보를 사용하여 OBMC에 의해 형성

된 예측 신호는 현재 CU의 원래 신호의 상단 및 왼쪽 경계들을 보상하기 위해 사용되고, 그 후 정상 모션 추정 프로세스가 적용된다.

[0203] 2.2.5. 로컬 조명 보상

[0204] 로컬 조명 보상(Local Illumination Compensation, LIC)은, 스케일링 인자 a 및 오프셋 b 를 사용하여, 조명 변화에 대한 선형 모델에 기초한다. 그리고 그것은 각각의 인터-모드 코딩된 코딩 단위(CU)에 대해 적응적으로 인에이블되거나 디스에이블된다. 도 19는 IC 파라미터들을 도출하기 위해 사용되는 이웃 샘플들의 예를 보여준다.

[0205] LIC가 CU에 적용될 때, 현재 CU의 이웃 샘플들 및 그들의 대응하는 참조 샘플들을 사용하여 파라미터들 a 및 b 를 도출하기 위해 최소 제곱 오차 방법이 이용된다. 더 구체적으로, 도 19에 예시된 바와 같이, CU의 서브샘플링된(2:1 서브샘플링) 이웃 샘플들 및 참조 화상 내의 대응하는 샘플들(현재 CU 또는 서브-CU의 모션 정보에 의해 식별됨)이 사용된다. IC 파라미터들은 각각의 예측 방향에 대해 개별적으로 도출되고 적용된다.

[0206] CU가 병합 모드로 코딩될 때, 병합 모드에서의 모션 정보 사본과 유사한 방식으로, LIC 플래그는 이웃 블록들로부터 복사된다; 그렇지 않으면, LIC가 적용되는지 여부를 지시하기 위해 LIC 플래그가 CU에 대해 시그널링된다.

[0207] LIC가 화상에 대해 인에이블될 때, LIC가 CU에 대해 적용되는지 여부를 결정하기 위해 추가적인 CU 레벨 RD 검사가 필요하다. LIC가 CU에 대해 인에이블될 때, 정수 pel 모션 탐색 및 분수 pel 모션 탐색을 위해 각각, SAD 및 SATD 대신에, 평균-제거된 절대 차이의 합(mean-removed sum of absolute difference, MR-SAD) 및 평균-제거된 절대 하다마르-변환된 차이의 합(mean-removed sum of absolute Hadamard-transformed difference, MR-SATD)이 사용된다.

[0208] 인코딩 복잡성을 감소시키기 위해, 다음의 인코딩 스킴이 JEM에서 적용된다.

[0209] 현재 화상과 그것의 참조 화상을 사이에 명백한 조명 변화가 없을 때 전체 화상에 대해 LIC가 디스에이블된다. 이러한 상황을 식별하기 위해, 현재 화상의 히스토그램들 및 현재 화상의 모든 참조 화상이 인코더에서 계산된다. 현재 화상과 현재 화상의 모든 참조 화상 사이의 히스토그램 차이가 주어진 임계치보다 작으면, 현재 화상에 대해 LIC가 디스에이블된다; 그렇지 않으면, 현재 화상에 대해 LIC가 인에이블된다.

[0210] 2.2.6. 아핀 모션 보상 예측

[0211] HEVC에서, 병진 모션 모델만이 모션 보상 예측(MCP)에 적용된다. 현실 세계에 있는 동안, 많은 종류의 모션, 예를 들어, 줌 인/아웃, 회전, 원근(perspective) 모션들 및 다른 비정규 모션들이 있다. JEM에서는, 단순화된 아핀 변환 모션 보상 예측이 적용된다. 도 20에 도시된 바와 같이, 블록의 아핀 모션 필드는 2개의 제어 포인트 모션 벡터들에 의해 기술된다.

[0212] 블록의 모션 벡터 필드(MVF)는 다음의 수학식에 의해 기술된다:

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w} x - \frac{(v_{1y} - v_{0y})}{w} y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w} x + \frac{(v_{1x} - v_{0x})}{w} y + v_{0y} \end{cases} \quad (1-a)$$

[0213] [0214] 6-파라미터 아핀에 대해,

$$\begin{cases} v_x = \frac{v_{1x}-v_{0x}}{w} x + \frac{v_{2x}-v_{0x}}{h} y + v_{0x} \\ v_y = \frac{v_{1y}-v_{0y}}{w} x + \frac{v_{2y}-v_{0y}}{h} y + v_{0y} \end{cases} \quad (1-b)$$

[0215] 여기서 (v_{0x}, v_{0y}) 는 왼쪽 상단 모서리 제어 포인트의 모션 벡터이고, (v_{1x}, v_{1y}) 은 오른쪽 상단 모서리 제어 포인트의 모션 벡터이고, (v_{2x}, v_{2y}) 는 왼쪽 하단 모서리 제어 포인트의 모션 벡터이고, (x, y) 는 현재 블록 내의 왼쪽 상단 샘플에 대한 대표 포인트의 좌표를 나타낸다. VTM에서, 대표 포인트는 서브-블록의 중심 위치로 정의되는데, 예를 들어, 현재 블록 내의 왼쪽 상단 샘플에 대한 서브-블록의 왼쪽 상단 모서리의 좌표가 (xs, ys) 일 때, 대표 포인트의 좌표는 $(xs+2, ys+2)$ 로 정의된다.

[0217] 모션 보상 예측을 더 단순화하기 위해, 서브-블록 기반 아핀 변환 예측이 적용된다. 서브-블록 크기 $M \times N$ 은 수학식 2에서와 같이 도출되고, 여기서 $MvPre$ 는 모션 벡터 분수 정확도(JEM에서 1/16)이고, (v_{2x}, v_{2y}) 는 수학식 1에 따라 계산된 왼쪽 하단 제어 포인트의 모션 벡터이다.

$$\begin{cases} M = clip3(4, w, \frac{w \times MvPre}{\max(abs(v_{1x}-v_{0x}), abs(v_{1y}-v_{0y}))}) \\ N = clip3(4, h, \frac{h \times MvPre}{\max(abs(v_{2x}-v_{0x}), abs(v_{2y}-v_{0y}))}) \end{cases} \quad (2)$$

[0218] [0219] 수학식 2에 의해 도출된 후에, M 및 N 은 필요하다면 그것을 w 및 h 의 제수(divisor)로 각각 만들기 위해 하향조정되어야 한다.

[0220] [0221] 각각의 $M \times N$ 서브-블록의 모션 벡터를 도출하기 위해, 도 21에 도시된 바와 같이, 각각의 서브-블록의 중심 샘플의 모션 벡터는 수학식 1에 따라 계산되고, 1/16 분수 정확도로 라운딩된다. 그 후 본 명세서의 다른 곳에서 언급되는 모션 보상 보간 필터들을 적용하여, 도출된 모션 벡터를 갖는 각각의 서브-블록의 예측을 생성한다.

[0222] MCP 후에, 각각의 서브-블록의 높은 정확도의 모션 벡터는 라운딩되어 정상 모션 벡터와 동일한 정확도로 저장된다.

2.2.6.1. AF_INTER 모드

[0223] JEM에서는, 2개의 아핀 모드: AF_INTER 모드 및 AF_MERGE 모드가 있다. 폭 및 높이 둘 다가 8보다 큰 CU들에 대해, AF_INTER 모드가 적용될 수 있다. AF_INTER 모드가 사용되는지를 지시하기 위해 비트스트림에서 CU레벨에서의 아핀 플래그가 시그널링된다. 이 모드에서, 모션 벡터 쌍

$\{(v_0, v_1) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$ 를 갖는 후보 리스트가 이웃 블록들을 사용하여 구성된다. 도 23에 도시된 바와 같이, v_0 은 블록 A, B 또는 C의 모션 벡터들로부터 선택된다. 이웃 블록으로부터의 모션 벡터는 참조 리스트 및 이웃 블록에 대한 참조의 POC, 현재 CU에 대한 참조의 POC 및 현재 CU의 POC 간의 관계에 따라 스케일링된다. 그리고 이웃 블록 D 및 E로부터 v_1 을 선택하는 접근법은 유사하다. 후보 리스트의 수가 2보다 작으면, 리스트는 AMVP 후보들 각각을 중복시키는 것에 의해 구성된 모션 벡터 쌍에 의해 패딩된다. 후보 리스트가 2보다 큰 경우, 후보들은 먼저 이웃 모션 벡터들의 일관성(쌍 후보에서의 2개의 모션 벡터의 유사성)에 따라 분류되고 처음 2개의 후보만이 유지된다. 어느 모션 벡터 쌍 후보가 현재 CU의 제어 포인트 모션 벡터 예측(CPMVP)으로서 선택되는지를 결정하기 위해 RD 비용 검사가 사용된다. 그리고 후보 리스트에서의 CPMVP의 위치를 지시하는 인덱스가 비트스트림에서 시그널링된다. 현재 아핀 CU의 CPMVP가 결정된 후에, 아핀 모션 추정이 적용되고 제어 포인트 모션 벡터(CPMV)가 발견된다. 그 후 CPMV와 CPMVP의 차이가 비트스트림에서 시그널링된다.

[0224] AF_INTER 모드에서, 4/6 파라미터 아핀 모드가 사용될 때, 2/3 제어 포인트들이 요구되고, 따라서 도 22에 도시된 바와 같이, 이를 제어 포인트에 대해 2/3 MVD가 코딩될 필요가 있다. 일부 실시예들에서, MV를 다음과 같이 도출하는 것이 제안되는데, 즉, mvd_0 으로부터 mvd_1 및 mvd_2 가 예측된다.

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

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

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

[0225] [0226] 여기서 \overline{mv}_i , mvd_i 및 mv_1 은, 도 22b에 도시된 바와 같이, 각각 왼쪽 상단 픽셀($i = 0$), 오른쪽 상단 픽셀($i = 1$) 또는 왼쪽 하단 픽셀($i = 2$)의 예측된 모션 벡터, 모션 벡터 차이 및 모션 벡터이다. 2개의 모션 벡터(예를 들어, $mvA(xA, yA)$ 및 $mvB(xB, yB)$)의 주가는 2개의 성분의 합산과 개별적으로 동등한데, 즉, $newMV = mvA + mvB$ 이고 $newMV$ 의 2개의 성분은 $(xA + xB)$ 및 $(yA + yB)$ 로 각각 설정된다는 점에 유의한다.

2.2.6.2. AF_MERGE 모드

[0227] CU가 AF_MERGE 모드에서 적용될 때, 그것은 유효 이웃 재구성된 블록들로부터 아핀 모드로 코딩된 제1 블록을 얻는다. 그리고 후보 블록에 대한 선택 순서는 도 24 a에 도시된 바와 같이 왼쪽, 위, 오른쪽 위, 왼쪽 하단에서 왼쪽 위이다. 도 24, 캡션 (b)에 도시된 바와 같이 이웃 왼쪽 하단 블록 A가 아핀 모드로 코딩되면, 블록 A

를 포함하는 CU의 왼쪽 상단 모서리, 오른쪽 위 모서리 및 왼쪽 하단 모서리의 모션 벡터들 v_2 , v_3 및 v_4 가 도출된다. 그리고 현재 CU 상의 왼쪽 상단 모서리의 모션 벡터 v_0 이 v_2 , v_3 및 v_4 에 따라 계산된다. 둘째로, 현재 CU의 오른쪽 위의 모션 벡터 v_1 이 계산된다.

- [0229] 현재 CU v_0 및 v_1 의 CPMV가 도출된 후에, 단순화된 아핀 모션 모델 수학식 1에 따라, 현재 CU의 MVF가 생성된다. 현재 CU가 AF_MERGE 모드로 코딩되는지를 식별하기 위해, 아핀 모드에서 적어도 하나의 이웃 블록이 코딩될 때 아핀 플래그가 비트스트림에서 시그널링된다.
- [0230] 일부 실시예들에서, 아핀 병합 후보 리스트는 다음의 단계들로 구성된다:
- [0231] 상속된 아핀 후보들을 삽입한다
- [0232] 상속된 아핀 후보는 후보가 그의 유효 이웃 아핀 코딩된 블록의 아핀 모션 모델로부터 도출된다는 것을 의미한다. 공통 베이스에서, 도 25에 도시된 바와 같이, 후보 위치들에 대한 스캔 순서는: A1, B1, B0, A0 및 B2이다.
- [0233] 후보가 도출된 후에, 동일한 후보가 리스트에 삽입되었는지를 검사하기 위해 풀 프루닝 프로세스가 수행된다. 동일한 후보가 존재한다면, 도출된 후보는 폐기된다.
- [0234] 구성된 아핀 후보들을 삽입한다
- [0235] 아핀 병합 후보 리스트 내의 후보들의 수가 MaxNumAffineCand(이 기고문에서 5로 설정됨)보다 작으면, 구성된 아핀 후보들이 후보 리스트에 삽입된다. 구성된 아핀 후보는 각각의 제어 포인트의 이웃 모션 정보를 조합하는 것에 의해 후보가 구성되는 것을 의미한다.
- [0236] 제어 포인트에 대한 모션 정보는 먼저 도 25에 도시된 특정된 공간 이웃 및 시간 이웃으로부터 도출된다. CPk($k=1, 2, 3, 4$)는 k 번째 제어 포인트를 나타낸다. A0, A1, A2, B0, B1, B2 및 B3은 CPk($k=1, 2, 3$)를 예측하기 위한 공간 위치들이고; T는 CP4를 예측하기 위한 시간 위치이다.
- [0237] CP1, CP2, CP3 및 CP4의 좌표들은 각각 (0, 0), (W, 0), (H, 0) 및 (W, H)이고, 여기서 W 및 H는 현재 블록의 폭 및 높이이다.
- [0238] 각각의 제어 포인트의 모션 정보는 다음의 우선순위 순서에 따라 획득된다:
- [0239] CP1에 대해, 검사 우선순위는 B2->B3->A2이다. B2는 이용가능하다면 사용된다. 그렇지 않고, B2가 이용가능하다면, B3이 사용된다. B2 및 B3 둘 다 이용불가능하다면, A2가 사용된다. 3개의 후보 모두가 이용불가능하다면, CP1의 모션 정보가 획득될 수 없다.
- [0240] CP2에 대해, 검사 우선순위는 B1->B0이다.
- [0241] CP3에 대해, 검사 우선순위는 A1->A0이다.
- [0242] CP4에 대해, T가 사용된다.
- [0243] 둘째로, 제어 포인트들의 조합들은 아핀 병합 후보를 구성하기 위해 사용된다.
- [0244] 6-파라미터 아핀 후보를 구성하기 위해 3개의 제어 포인트의 모션 정보가 필요하다. 3개의 제어 포인트는 다음 4개의 조합({CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4}) 중 하나로부터 선택될 수 있다. 조합들 {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4}는 왼쪽 상단, 오른쪽 상단 및 왼쪽 하단 제어 포인트들에 의해 표현되는 6-파라미터 모션 모델로 변환될 것이다.
- [0245] 4-파라미터 아핀 후보를 구성하기 위해 2개의 제어 포인트의 모션 정보가 필요하다. 2개의 제어 포인트는 다음 6개의 조합({CP1, CP4}, {CP2, CP3}, {CP1, CP2}, {CP2, CP4}, {CP1, CP3}, {CP3, CP4}) 중 하나로부터 선택될 수 있다. 조합들 {CP1, CP4}, {CP2, CP3}, {CP2, CP4}, {CP1, CP3}, {CP3, CP4}는 왼쪽 상단 및 오른쪽 상단 제어 포인트들에 의해 표현되는 4-파라미터 모션 모델로 변환될 것이다.
- [0246] 구성된 아핀 후보들의 조합들은 다음의 순서로 후보 리스트에 삽입된다:
- [0247] {CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}, {CP2, CP3}, {CP1, CP4}, {CP2, CP4}, {CP3, CP4}

[0248] 조합의 참조 리스트 X(X는 0 또는 1임)에 대해, 제어 포인트들에서 가장 높은 사용 비율을 갖는 참조 인덱스는 리스트 X의 참조 인덱스로서 선택되고, 모션 벡터들은 참조 화상이 스케일링될 차이를 가리킨다.

[0249] 후보가 도출된 후에, 동일한 후보가 리스트에 삽입되었는지를 검사하기 위해 풀 프루닝 프로세스가 수행된다. 동일한 후보가 존재한다면, 도출된 후보는 폐기된다.

[0250] 제로 모션 벡터들을 이용한 패딩

[0251] 아핀 병합 후보 리스트 내의 후보들의 수가 5보다 작으면, 제로 참조 인덱스들을 갖는 제로 모션 벡터들이, 리스트가 가득 찰 때까지, 후보 리스트에 삽입된다.

2.2.7. 양방향 광학 흐름

[0253] 양방향 광학 흐름(Bi-directional Optical flow, BIO)은 양방향 예측을 위한 블록 단위(block-wise) 모션 보상에 더하여 수행되는 샘플 단위(sample-wise) 모션 리파인먼트이다. 샘플-레벨 모션 리파인먼트는 시그널링을 사용하지 않는다.

[0254] $I^{(k)}$ 를 블록 모션 보상 후에 참조 k($k=0, 1$)로부터의 루마 값이고, $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ 각각 $I^{(k)}$ 기울기의 수평 및 수직 성분들이라고 하자. 광학 흐름이 유효하다고 가정하면, 모션 벡터 필드 (v_x, v_y) 는 다음 수학식에 의해 주어진다.

$$\partial I^{(k)}/\partial t + v_x \partial I^{(k)}/\partial x + v_y \partial I^{(k)}/\partial y = 0. \quad (3)$$

[0256] 이러한 광학 흐름 수학식을 각각의 샘플의 모션 케적에 대한 에르미트 보간과 조합하면 결국에는 함수 값을 $I^{(k)}$ 및 도함수들 $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ 둘 다와 매칭되는 고유한 3차 다항식을 야기한다. $t=0$ 에서 이 다항식의 값은 BIO 예측이다:

$$pred_{BIO} = 1/2 \cdot (I^{(0)} + I^{(1)} + v_x/2 \cdot (\tau_1 \partial I^{(1)}/\partial x - \tau_0 \partial I^{(0)}/\partial x) + v_y/2 \cdot (\tau_1 \partial I^{(1)}/\partial y - \tau_0 \partial I^{(0)}/\partial y)) \quad (3)$$

[0258] 여기서, τ_0 및 τ_1 는 도 26에 도시된 바와 같은 참조 프레임들까지의 거리들을 나타낸다. 거리들 τ_0 및 τ_1 는 Ref0 및 Ref1에 대한 POC에 기초하여 계산된다: $\tau_0 = \text{POC(current)} - \text{POC(Ref0)}$, $\tau_1 = \text{POC(Ref1)} - \text{POC(current)}$. 양쪽 예측들이 동일한 시간 방향으로부터 유래한다면(둘 다 과거로부터 또는 둘 다 미래로부터), 부호들은 상이하다(즉, $\tau_0 \cdot \tau_1 < 0$). 이 경우, BIO는 예측이 동일한 시간 순간으로부터 유래하지 않는 경우(즉, $\tau_0 \neq \tau_1$)에만 적용되고, 양쪽 참조된 영역들은 비-제로 모션 ($MVx_0, MVy_0, MVx_1, MVy_1 \neq 0$)을 갖고 블록 모션 벡터들은 시간 거리 ($MVx_0/MVx_1 = MVy_0/MVy_1 = -\tau_0/\tau_1$)에 비례한다.

[0259] 모션 벡터 필드 (v_x, v_y) 는 포인트들 A 및 B에서의 값들 간의 차이 Δ 를 최소화하는 것에 의해 결정된다(도 9에서 모션 케적과 참조 프레임 평면들의 교차). 모델은 Δ 에 대한 로컬 테일러 급수의 제1 선형 항만을 사용한다:

$$\Delta = (I^{(0)} - I^{(1)})_0 + v_x (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x) + v_y (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y) \quad (5)$$

[0261] 수학식 5에서의 모든 값들은 지금까지의 표기법으로부터 생략된 샘플 위치 (i', j') 에 의존한다. 모션이 로컬 주변 영역에서 일관된다고 가정하면, 현재 예측된 포인트 (i, j) 에 중심을 둔 $(2M+1) \times (2M+1)$ 정사각형 윈도우

Ω 내부에서 Δ 를 최소화하고, 여기서 M 은 2와 동등하다:

$$(v_x, v_y) = \arg \min_{v_x, v_y} \sum_{[i', j'] \in \Omega} \Delta^2 [i', j'] \quad (6)$$

[0263] 이 최적화 문제에 대해, JEM은 먼저 수직 방향에서 그리고 그 후 수평 방향에서 최소화를 행하는 단순화된 접근법을 사용한다. 이는 다음을 야기하고

$$v_x = (s_1 + r) > m? clip3 \left(-thBIO, thBIO, -\frac{s_3}{(s_1+r)} \right) : 0 \quad (7)$$

$$v_y = (s_5 + r) > m? clip3 \left(-thBIO, thBIO, -\frac{s_6 - v_x s_2 / 2}{(s_5+r)} \right) : 0 \quad (8)$$

[0264] 여기서,

$$\begin{aligned} s_1 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_3 &= \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x), \\ s_2 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y), \\ s_5 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_6 &= \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad (9)$$

[0265] 제로 또는 매우 작은 값에 의한 나눗셈을 피하기 위해, 규칙화 파라미터들 r 및 m 이 수학식 7 및 수학식 8에 도입된다.

$$r = 500 \cdot 4^{d-8} \quad (10)$$

$$m = 700 \cdot 4^{d-8} \quad (11)$$

[0266] 여기서 d 는 비디오 샘플들의 비트 깊이이다.

[0267] BIO에 대한 메모리 액세스를 정규 양방향 예측 모션 보상에 대한 것과 동일하게 유지하기 위해, 모든 예측 및 기울기 값들 $I^{(k)}, \partial I^{(k)} / \partial x, \partial I^{(k)} / \partial y$ 가 현재 블록 내부의 위치들에 대해서만 계산된다. 수학식 9에서, 예측 블록의 경계 상의 현재 예측 포인트에 중심을 둔 $(2M+1) \times (2M+1)$ 정사각형 윈도 Ω 는 (도 27의 왼쪽 (a)에 도시된 바와 같이) 블록 외부의 위치들에 액세스할 필요가 있다. JEM에서, 블록 외부의 $I^{(k)}, \partial I^{(k)} / \partial x, \partial I^{(k)} / \partial y$ 의 값들은 블록 내부의 가장 가까운 이용 가능한 값과 동등하게 설정된다. 예를 들어, 이는 도 27의 오른쪽 (b)에 도시된 바와 같이 패딩으로서 구현될 수 있다.

[0268] BIO에 의해, 모션 필드가 각각의 샘플에 대해 리파인될 수 있는 것이 가능하다. 계산 복잡성을 감소시키기 위해, BIO의 블록 기반 설계가 JEM에서 사용된다. 모션 리파인먼트는 4×4 블록에 기초하여 계산된다. 블록 기반 BIO에서, 4×4 블록 내의 모든 샘플들의 수학식 9에서의 s_n 의 값들이 집계되고, 그 후 s_n 의 집계된 값들이 4×4 블록에 대한 BIO 모션 벡터 오프셋을 도출하기 위해 사용된다. 더 구체적으로, 다음의 공식이 블록 기반 BIO 도출을 위해 사용된다:

$$\begin{aligned} s_{1,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i', j'] \in \Omega(x,y)} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_{3,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x), \\ s_{2,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y), \\ s_{5,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_{6,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad (12)$$

[0269] 여기서 b_k 는 예측된 블록의 k 번째 4×4 블록에 속하는 샘플들의 세트를 나타낸다. 수학식 7 및 8에서의 s_n 은 연관된 모션 벡터 오프셋들을 도출하기 위해 ($(s_{n,b_k}) \gg 4$)로 대체된다.

[0274] 일부 경우들에서, BIO의 MV 연대(regiment)는 잡음 또는 비정규 모션으로 인해 신뢰할 수 없을 수 있다. 따라서, BIO에서, MV 연대의 크기는 임계치 값 thBIO로 클리핑된다. 임계치 값은 현재 화상의 참조 화상들이 모두 한 방향으로부터 유래한 것인지에 기초하여 결정된다. 현재 화상의 모든 참조 화상이 한 방향으로부터 유래한 것이면, 임계치의 값은 $12 \times 2^{14-d}$ 로 설정된다; 그렇지 않으면, 그것은 $12 \times 2^{13-d}$ 로 설정된다.

[0275] BIO에 대한 기울기들은 HEVC 모션 보상 프로세스(2D 분리가능 FIR)와 부합하는 연산들을 사용하는 모션 보상 보간과 동시에 계산된다. 이 2D 분리가능 FIR에 대한 입력은 모션 보상 프로세스 및 블록 모션 벡터의 분수 부분에 따른 분수 위치($fracX, fracY$)에 대한 것과 동일한 참조 프레임 샘플이다. 수평 기울기 $\partial I / \partial x$ 신호의 경우 먼저 디-스케일링 시프트 d-8을 갖는 분수 위치 $fracY$ 에 대응하는 $BIOfilterS$ 를 사용하여 수직으로 보간되고, 그 후 18-d만큼 디-스케일링 시프트를 갖는 분수 위치 $fracX$ 에 대응하는 수평 방향으로 기울기 필터 $BIOfilterG$ 가 적용된다. 수직 기울기 $\partial I / \partial y$ 의 경우 먼저 디-스케일링 시프트 d-8을 갖는 분수 위치 $fracY$ 에 대응하는 $BIOfilterG$ 를 사용하여 수직으로 기울기 필터가 적용되고, 그 후 18-d만큼 디-스케일링 시프트를 갖는 분수 위치 $fracX$ 에 대응하는 수평 방향으로 $BIOfilterS$ 를 사용하여 신호 변위가 수행된다. 기울기 계산 $BIOfilterG$ 및 신호 변위 $BIOfilterF$ 를 위한 보간 필터의 길이는 합리적인 복잡성을 유지하기 위해 더 짧다(6-텝). 표 3은 BIO에서 블록 모션 벡터의 상이한 분수 위치들에 대한 기울기 계산을 위해 사용되는 필터들을 보여준다. 표 4는 BIO에서의 예측 신호 생성을 위해 사용되는 보간 필터들을 보여준다.

표 3: BIO에서의 기울기 계산을 위한 필터들

분수 pel 위치	기울기를 위한 보간 필터(BIOfilterG)
0	{ 8, -39, -3, 46, -17, 5 }
1/16	{ 8, -32, -13, 50, -18, 5 }
1/8	{ 7, -27, -20, 54, -19, 5 }
3/16	{ 6, -21, -29, 57, -18, 5 }
1/4	{ 4, -17, -36, 60, -15, 4 }
5/16	{ 3, -9, -44, 61, -15, 4 }
3/8	{ 1, -4, -48, 61, -13, 3 }
7/16	{ 0, 1, -54, 60, -9, 2 }
1/2	{ -1, 4, -57, 57, -4, 1 }

표 4: BIO에서의 예측 신호 생성을 위한 보간 필터들

[0278]

표 4: BIO에서의 예측 신호 생성을 위한 보간 필터들

분수 pel 위치	예측 신호를 위한 보간 필터(BIOfilterS)
0	{ 0, 0, 64, 0, 0, 0}
1/16	{ 1, -3, 64, 4, -2, 0}
1/8	{ 1, -6, 62, 9, -3, 1}
3/16	{ 2, -8, 60, 14, -5, 1}
1/4	{ 2, -9, 57, 19, -7, 2}
5/16	{ 3, -10, 53, 24, -8, 2}
3/8	{ 3, -11, 50, 29, -9, 2}
7/16	{ 3, -11, 44, 35, -10, 3}
1/2	{ 3, -10, 35, 44, -11, 3}

[0279]

[0280]

JEM에서, BIO는 2개의 예측이 상이한 참조 화상으로부터 유래한 것일 때 모든 양방향 예측된 블록에 적용된다. LIC가 CU에 대해 인에이블될 때, BIO는 디스에이블된다.

[0281]

JEM에서, OBMC는 정상 MC 프로세스 후에 블록에 대해 적용된다. 계산 복잡성을 감소시키기 위해, OBMC 프로세스 동안 BIO가 적용되지 않는다. 이는 BIO가 그 자신의 MV를 사용할 때만 블록에 대해 MC 프로세스에서 적용되고 이웃 블록의 MV가 OBMC 프로세스 동안 사용될 때에는 MC 프로세스에서 적용되지 않는다는 것을 의미한다.

[0282]

2.2.8. 디코더 측 모션 벡터 리파인먼트

[0283]

양방향 예측 동작에서, 하나의 블록 영역의 예측에 대해, 각각 list0의 모션 벡터(MV) 및 list1의 MV를 사용하여 형성된 2개의 예측 블록이 조합되어 단일 예측 신호를 형성한다. 디코더 측 모션 벡터 리파인먼트(decoder-side motion vector refinement, DMVR) 방법에서, 양방향 예측의 2개의 모션 벡터는 쌍방 템플릿 매칭 프로세스에 의해 추가로 리파인된다. 쌍방 템플릿 매칭은 추가적인 모션 정보의 송신 없이 리파인된 MV를 획득하기 위해 참조 화상을 내의 재구성 샘플들과 쌍방 템플릿 간의 왜곡 기반 탐색을 수행하기 위해 디코더에서 적용된다.

[0284]

DMVR에서, 쌍방 템플릿은, 도 28에 도시된 바와 같이, 각각, list0의 초기 MVO 및 list1의 MV1로부터, 2개의 예측 블록의 가중된 조합(즉, 평균)으로서 생성된다. 템플릿 매칭 동작은 생성된 템플릿과 참조 화상에서의 샘플 영역(초기 예측 블록 주위의) 사이의 비용 측정치들을 계산하는 것으로 구성된다. 2개의 참조 화상 각각에 대해, 최소 템플릿 비용을 산출하는 MV가 원래의 것을 대체하기 위해 해당 리스트의 업데이트된 MV로서 간주된다. JEM에서, 9개의 MV 후보가 각각의 리스트에 대해 탐색된다. 9개의 MV 후보는 원래의 MV 및 수평 또는 수직 방향 중 어느 하나 또는 둘 다에서 원래의 MV에 대한 하나의 루마 샘플 오프셋을 갖는 8개의 주위의 MV를 포함한다. 마지막으로, 2개의 새로운 MV, 즉, 도 28에 도시된 바와 같은 MVO' 및 MV1'이 최종 양방향 예측 결과들을 생성하기 위해 사용된다. 절대 차이들의 합(SAD)이 비용 측정치로서 사용된다. 하나의 주위의 MV에 의해 생성되는 예측 블록의 비용을 계산할 때, (정수 pel로의) 라운딩된 MV는 실제 MV 대신에 예측 블록을 획득하기 위해 실제로 사용된다는 점에 유의한다.

[0285]

DMVR은, 추가적인 구문 요소들의 송신 없이, 과거의 참조 화상으로부터의 하나의 MV와 미래의 참조 화상으로부터의 다른 MV와의 양방향 예측의 병합 모드에 적용된다. JEM에서, LIC, 아핀 모션, FRUC, 또는 서브-CU 병합 후보가 CU에 대해 인에이블될 때, DMVR은 적용되지 않는다.

[0286]

2.2.9. 서브-블록 병합 후보 리스트

- [0287] 비-서브 블록 병합 후보들에 대한 정규 병합 리스트에 더하여 모든 서브-블록 관련 모션 후보들이 별개의 병합 리스트에 넣어지는 것이 제안된다.
- [0288] 서브-블록 관련 모션 후보들은 별개의 병합 리스트에 넣어지고 '서브-블록 병합 후보 리스트'로서 명명된다.
- [0289] 일 예에서, 서브-블록 병합 후보 리스트는 아핀 병합 후보들, 및 ATMVP 후보, 및/또는 서브-블록 기반 STMVP 후보를 포함한다.
- [0290] **2.2.9.1 서브-블록 병합 후보 리스트의 구성**
- [0291] 이 기고문에서, 정상 병합 리스트 내의 ATMVP 병합 후보는 아핀 병합 리스트의 제1 위치로 이동된다. 따라서 새로운 리스트(즉, 서브-블록 기반 병합 후보 리스트) 내의 모든 병합 후보들은 서브-블록 코딩 툴들에 기초한다.
- [0292] **2.2.10. 일반화된 양방향 예측 예들**
- [0293] GBi에 의해, 코딩된 코딩 단위에 대해 하나의 가중 파라미터가 시그널링되어 수학식 1을 사용하여 양방향 예측 $P_{bi-pred}$ 가 생성되고, 여기서 P0 및 P1은 각각 list-0 및 list-1 내의 참조 화상을 사용하는 모션 보상된 예측들이다. w는 1/8 정밀도로 표현된 list-1 예측에 대한 가중 파라미터이다. GBi는 또한 BCW(Bi-prediction with Coding Unit based Weighting)로도 알려져 있다.
- [0294]
$$P_{bi-pred} = ((8 - w) * P_0 + w * P_1 + 4) \gg 3 \quad (13)$$
- [0295] 현재의 GBi 설계에서, 저지연 화상에 대해 5개의 가중치 {-2/8, 3/8, 4/8, 5/8, 10/8}이 이용가능하고 비-저지연 화상에 대해 3개의 가중치 {3/8, 4/8, 5/8}가 이용가능하다.
- [0296] 코딩 단위에 대한 가중치 값을 결정하기 위해 다음의 규칙들이 적용된다:
- [0297] 시그널링 MVD(즉, 정상 인터 예측 모드)를 요구하는 각각의 양방향 예측 PU에 대해, 그것의 가중치 값은 명시적으로 시그널링된 것과 동등하게 설정된다.
- [0298] 병합 모드들로 코딩되는 각각의 양방향 예측 PU에 대해, 그것의 가중치 값 w는 연관된 병합 후보에 대해 사용되는 가중치 값으로부터 직접 추론된다.
- [0299] **2.3 이웃 구문 요소들을 사용하는 컨텍스트 모델링**
- [0300] 한 유형의 컨텍스트 모델링은 인코딩할 현재 구문 요소의 과거에 최대 2개의 이웃 구문 요소를 갖는 컨텍스트 템플릿을 수반하고, 여기서 이웃의 종류의 특정 정의는 구문 요소에 의존한다. 통상적으로, 특정 빈에 대한 이러한 종류의 컨텍스트 모델의 사양은, 도 25에서 A2 및 B3에 도시된 바와 같이, 현재 구문 요소의 왼쪽 및 상부의 이웃 요소에 대한 관련 빈 값들의 모델링 함수에 기초한다.
- [0301] **9.5.4.2.2 왼쪽 및 위의 구문 요소들을 사용하는 ctxInc의 도출 프로세스**
- [0302] 이 프로세스에 대한 입력은 현재 화상의 왼쪽 상단 샘플에 대한 현재 루마 블록의 왼쪽 상단 루마 샘플, 컬러 성분 cIdx, 현재 코딩 큐드트리 깊이 cqDepth, 및 루마 샘플들에서의 현재 코딩 블록의 폭 및 높이 cbWidth 및 cbHeight를 특정하는 루마 위치 (x0, y0)이다.
- [0303] 이 프로세스의 출력은 ctxInc이다.
- [0304] 위치 (xNbL, yNbL)은 (x0 - 1, y0)과 동등하게 설정되고, 현재 블록의 왼쪽에 직접 위치되는 블록의 이용가능성을 특정하는 변수 availableL은, (x0, y0)과 동등하게 설정된 위치 (xCurr, yCurr) 및 (xNbL, yNbL)과 동등하게 설정된 이웃 위치 (xNbY, yNbY)를 입력들로서 갖는 하위 조항 6.4에 특정된 바와 같은 z-스캔 순서로 블록에 대한 이용가능성 도출 프로세스를 호출하는 것에 의해 도출되고, 출력은 availableL에 할당된다.
- [0305] 위치 (xNbA, yNbA)는 (x0, y0 - 1)과 동등하게 설정되고, 현재 블록의 위에 직접 위치되는 코딩 블록의 이용가능성을 특정하는 변수 availableA는, (x0, y0)과 동등하게 설정된 위치 (xCurr, yCurr) 및 (xNbA, yNbA)와 동등하게 설정된 이웃 위치 (xNbY, yNbY)를 입력들로서 갖는 하위 조항 6.4에 특정된 바와 같은 z-스캔 순서로 블록에 대한 이용가능성 도출 프로세스를 호출하는 것에 의해 도출되고, 출력은 availableA에 할당된다.

[0306]

변수 sizeC, sizeTh2 및 sizeTh1은 다음과 같이 도출된다:

$$\text{sizeTh2} = (\text{MaxBtSizeY} == 128) ? 1024 : ((\text{MaxBtSizeY} == 64) ? 512 : 256) \quad (9-19)$$

$$\text{sizeTh1} = (\text{MaxBtSizeY} == 128) ? 128 : 64 \quad (9-20)$$

[0307]

$$\text{sizeC} = \text{cbWidth} * \text{cbHeight} \quad (9-21)$$

[0308]

표 9-11에 특정된 구문 요소들 alf_ctb_flag[x0][y0][cIdx], qt_split_cu_flag[x0][y0], mtt_split_cu_flag[x0][y0], cu_skip_flag[x0][y0], amvr_mode[x0][y0], inter_affine_flag[x0][y0], merge_triangle_flag[x0][y0] 및 merge_subblock_flag[x0][y0]에 대한 condL 및 condA를 갖는 ctxInc의 할당이 다음과 같이 특정된다:

[0309]

$$\text{ctxInc} = (\text{condL} \&& \text{availableL}) + (\text{condA} \&& \text{availableA}) + \text{ctxSetIdx} * 3 \quad (9-22)$$

[0310]

표 9-11 - 왼쪽 및 위의 구문 요소들을 사용하는 ctxInc의 사양

구문 요소	condL	condA	ctxSetIdx
alf_ctb_flag[x0][y0][cIdx]	alf_ctb_flag[xNbL][yNbL][cIdx]	alf_ctb_flag[xNbA][yNbA][cIdx]	cIdx
qt_split_cu_flag	cqtDepth[xNbL][yNbL] > cqtDepth	cqtDepth[xNbA][yNbA] > cqtDepth	(cqtDepth < 2) ? 0 : 1
mtt_split_cu_flag treeType == DUAL_TREE_CHROMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth	3
mtt_split_cu_flag treeType != DUAL_TREE_CHROMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth (sizeC > sizeTh2) ? 0 : ((sizeC > sizeTh1) ? 1 : 2)	
cu_skip_flag[x0][y0]	cu_skip_flag[xNbL][yNbL]	cu_skip_flag[xNbA][yNbA]	0
amvr_mode[x0][y0]	amvr_mode[xNbL][yNbL]	amvr_mode[xNbA][yNbA]	0
merge_subblock_flag[x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0
merge_triangle_flag[x0][y0]	merge_triangle_flag[xNbL][yNbL]	merge_triangle_flag[xNbA][yNbA]	0
inter_affine_flag[x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0

[0311]

2.4 예측 모드 플래그를 위한 개선된 컨텍스트

[0312]

VVC 초안 3에서, 예측 모드 플래그, 즉, pred_mode_flag는 단 하나의 컨텍스트만을 사용하여 코딩된다. 일부 실시예들에서, 현재 블록의 pred_mode_flag를 코딩하기 위한 컨텍스트는 이웃 블록들의 pred_mode_flag의 값을 사용하여 도출된다. 도 43에서 설명된 바와 같이, 컨텍스트 값은 위의 (A) 또는 왼쪽 (L) 블록이 인트라 예측 모드에 의해 코딩되면 1만큼 증가되고, 따라서 컨텍스트들의 수는 pred_mode_flag를 코딩하기 위해 2만큼 증가된다.

[0313]

VTM-3.0에서, 조합된 병합 및 인터 예측 모드, 즉 인터-인트라 모드가 또한 인에이블된다. 따라서, 컨텍스트 도출을 위해 이웃 블록을 사용할 때 인터-인트라 모드가 인트라 모드로서 간주되어야 하는지 또는 인터 모드로서 간주되어야 하는지가 문제가 된다. 이 기고문에서, 결론을 얻기 위해 상이한 솔루션들이 테스트된다.

[0314]

일부 실시예들에서, pred_mode_flag를 코딩하기 위해 사용되는 컨텍스트를 도출하기 위해, 아래에 설명되는 바

와 같이, 4개의 상이한 방법이 테스트된다.

[0316] 표 1: *pred_mode_flag*의 컨텍스트 도출을 위한 상이한 방법들의 설명

컨텍스트(ctx) 도출의 설명	
방법 #1 (3 개의 컨텍스트)	$ctx = A \text{ 가 인트라 ? } 1 : 0$ $ctx += L \text{ 이 인트라 ? } 1 : 0$
방법 #2 (2 개의 컨텍스트)	$ctx = (A \text{ 가 인트라}) (L \text{ 이 인트라}) ? 1 : 0$
방법 #3 (3 개의 컨텍스트)	$ctx = A \text{ 가 인트라 또는 인터-인트라 ? } 1 : 0$ $ctx += L \text{ 이 인트라 또는 인터-인트라 ? } 1 : 0$
방법 #4 (2 개의 컨텍스트)	$ctx = (A \text{ 가 인트라 또는 인터-인트라}) (L \text{ 이 인트라 또는 인터-인트라 ? } 1 : 0$

[0317]

3. 실시예들에 의해 해결된 문제들의 예들

[0318]

삼각형 파티션의 설계에서, 하나의 블록은 2개의 파티션으로 분할될 수 있다. 모션 보상으로 인한 메모리 대역폭을 절약하기 위해, 2개의 파티션이 단방향 예측되어야 하는 것이 요구된다. 모션 보상 프로세스 동안, 대각선 에지에 대해 양방향 예측이 사용되고 모든 다른 나머지 부분들에 대해 단방향 예측이 사용된다. 각각의 파티션에 대해 양방향 예측이 허용되면, 대각선 에지에 위치된 샘플들은 4개의 MV를 가질 것이고, 이들 중 2개는 하나의 파티션으로부터 유래한 것이고 이들 중 2개는 다른 파티션으로부터 유래한 것이다. 그러한 설계는 다음의 문제들을 갖는다:

[0319]

TPM 리스트를 구성할 때, 서브-블록 병합 후보들은 TPM 후보들을 도출하는 데 이용되지 않고, 공간 및 시간 블록들로부터 도출된 비-서브블록 병합 후보들만이 이용된다.

[0320]

서브-블록 기반의 방법은 삼각형 파티션에 대해 허용되지 않는다.

[0321]

비-대각선 에지에 대해 단방향 예측만을 허용하는 것의 제한은 더 적은 코딩 효율을 야기할 수 있다.

[0322]

삼각형 파티션들과의 DMVR, BIO 상호작용.

[0323]

2개의 파티션이 상이한 TPM 후보 인덱스들을 선택하기 위해 허용되지만, 2개의 TPM 후보가 동일한 정규 모션 후보로부터의 것인지를 확신하지 못할 수 있다. 예를 들어, 하나의 양방향 정규 모션 후보가 2개의 TPM 후보를 도출하기 위해 사용될 수 있고, 이들 각각이 하나의 파티션에 할당될 수 있다. 이 경우, 2개의 파티션은 덜 효율적일 수 있는 비-TPM 모드와 동일한 양방향 예측 정규 모션 후보를 사용할 수 있다.

[0324]

TPM은 블록 폭*높이가 64보다 작을 때 디스에이블된다. 그러나, 4x16 또는 16x4와 동등한 크기 및 4:2:0 컬러 포맷을 갖는 블록에 대해, 이는 TPM이 인에이블된 2x8 크로마 블록을 야기할 것이고, 이는 하드웨어 복잡성을 증가시킨다.

[0325]

MMVD, 서브-블록 병합 리스트, CIIP(combined inter-intra prediction) 모드라고도 불리는 MHIntra(인터-인트라), TPM을 포함한, VTM3.0에서의 수 개의 새로운 툴들이 특수 병합 모드들로서 취급된다. 따라서, 5개의 병합 모드가 있다. 5개의 병합 모드의 지시 전에 병합 플래그/스킵 플래그가 시그널링된다. 따라서, 원래의 스kip/병합 모드에 도달하기 위해, MMVD/서브-블록 병합 리스트/MHIntra(인터-인트라)/TPM의 지시는 0이 되도록 디코딩된다.

[0326]

CABAC의 컨텍스트 모델링에서, 상이한 CTU 행으로부터의 이웃 요소들이 요구될 수 있고, 이는 CTU 행 위로부터

의 요소들을 저장하기 위해 추가적인 라인 베퍼를 필요로 한다.

[0328] 수 개의 구문 요소들에 대해, 컨텍스트 모델링에서 사용되는 이웃 블록들은 재구성 블록들을 위해 사용되는 것들과 상이하다.

4. 실시예들의 예들

[0329] 제안된 방법들은 비-정사각형/비-직사각형 파티션들, 예를 들어, 기하학적 파티션들 중 임의의 것에 적용될 수 있다. 다음의 설명들에서, '삼각형 파티션 모드'를 사용하여 비-정사각형/비-직사각형 파티션 모드(TPM)의 예로서 하나를 나타내고, TPM 후보 리스트에 삽입된 모션 벡터 예측 후보는 'TPM 후보들'이라고 명명된다. 그리고 임의의 이전에 코딩된 블록들과 연관된 모션 정보는 '정규 모션 후보(regular motion candidate)'라고 명명된다. 다른 종류의 파티션들도 적용가능할 수 있다는 점에 유의해야 한다.

[0330] 아래의 상세한 기법들은 일반적인 개념들을 설명하기 위한 예들로서 간주되어야 한다. 이를 기법은 좁은 방식으로 해석되어서는 안 된다. 더욱이, 이를 기법은 임의의 방식으로 조합될 수 있다.

TPM의 확장들

[0331] 1. 서브-블록 후보들로부터 도출된 하나 또는 다수의 TPM 후보가 TPM 리스트에 추가될 수 있는 것이 제안된다.

[0332] a. 일 예에서, TPM 후보는 ATMVP 후보로부터 도출될 수 있다.

[0333] b. 일 예에서, TPM 후보는 아핀 후보로부터 도출될 수 있다.

[0334] c. 서브-블록 후보(예를 들어, ATMVP)에 대해, 그것은 다수의 모션 정보 세트를 포함할 수 있고, 각각의 세트는 하나의 서브-블록(예를 들어, ATMVP에 대해 8x8, 아핀에 대해 4x4)에 대응한다. 하나의 대표 세트가 정규 모션 후보로서 취급될 수 있고, 그것은 TPM 후보들을 도출하기 위해 사용될 수 있다.

[0335] i. 대표 세트는 모션 정보 세트가 연관되는 서브-블록 위치에 기초하여 정의될 수 있다. 예를 들어, 하나의 블록의 왼쪽 상단, 오른쪽 상단, 왼쪽 하단, 오른쪽 하단 서브-블록과 연관된 모션 정보가 선택될 수 있다.

[0336] ii. M개의 대표 세트가 선택될 수 있는데, 예를 들어, M은 파티션들의 수로 설정된다. 일 예에서, 각각의 파티션의 중심 서브-블록과 연관된 모션 정보가 선택될 수 있다.

[0337] 2. 다수의 파티션(예를 들어, 삼각형 파티션 설계에서의 2개의 파티션)이 하나의 단일 정규 모션 후보로부터 상속될 수 있는 것이 제안된다.

[0338] a. 일 예에서, 각각의 파티션에 대한 선택된 병합 인덱스들 및 분할 패턴들의 지시의 범위가 확장될 수 있다. M개의 분할 파티션, N개의 허용된 병합 후보가 있다고 가정하면, 지시의 범위는 현재의 삼각형 설계에서 $M*N*(N-1)$ 과 동등한 대신에 $M*N*N$ 으로 정의된다.

[0339] b. 일 예에서, 하나의 단일 정규 모션 후보로부터 다수의 파티션이 예측되는지를 지시하기 위해 하나의 플래그가 먼저 시그널링될 수 있다.

[0340] c. 일 예에서, 하나의 단일 정규 모션 후보로부터 다수의 파티션이 예측될 때, 그러한 정규 모션 후보는 양방향 예측된 모션 후보이어야 한다는 것이 제한된다.

[0341] i. 대안적으로, 더욱이, 2개의 파티션은 각각 정규 모션 후보의 List 0 모션 정보 및 List 1 모션 정보로 할당될 수 있다.

[0342] ii. 대안적으로, 더욱이, 가중 영역에 위치된 샘플들에 대해, 가중치 그룹은 2개의 파티션이 상이한 모션 후보들로부터 예측되는 경우들에 대해 사용되는 것들과 상이할 수 있다. 일 예에서, 평균이 항상 사용된다.

[0343] d. 그러한 방법들은 종래의 AMVP 또는 병합 또는 TPM에 더하여 추가적인 코딩 모드로서 취급될 수 있다.

[0344] 3. 비트스트림으로부터 도출된 각각의 파티션의 모션 벡터들은 하나의 TPM 코딩된 블록의 최종 예측 블록을 도출하기 위해 사용되기 전에 추가로 리파인될 수 있다.

[0345] a. 일 예에서, DMVR 기술은 전체 블록의 모션 정보로서 2개의 파티션의 모션 정보와 함께 적용될 수 있

다.

[0348] b. 일 예에서, BIO는 TPM 예측 후에 적용된다.

[0349] c. 일 예에서, 하나의 파티션이 LX로부터 단방향 예측되고 다른 파티션이 L(1-X)로부터 양방향 예측 또는 예측되면, DMVR 또는 BIO 기술은 리스트 L(1-X) 내의 다른 파티션의 모션 정보를 사용하여 적용될 수 있다.

[0350] d. 일 예에서, 리파인된 모션 정보가 저장될 수 있다.

[0351] e. 일 예에서, 파티션이 LX로부터 단방향 예측된다면, LX 내의 리파인된 모션 정보가 저장될 수 있다.

[0352] a. TPM에서의 하나 이상의 파티션(들)이 인트라 예측과 조합될 수 있는 것이 제안된다. 예를 들어, TPM 예측 및 인트라 예측은 가중 합산되어 최종 예측을 얻는다.

[0353] f. TPM에서 사용되는 하나 이상의 참조 화상이 현재 화상일 수 있는 것이 제안된다.

[0354] 4. 서브-블록 기반 기술은 TPM 코딩된 블록들에 대해 인에이블될 수 있다는 것이 제안되고 여기서 하나의 파티션 내의 각각의 서브-블록에는 그 자신의 모션 정보가 할당될 수 있다.

[0355] a. 일 예에서, 서브-블록 기반 병합 후보가 TPM 코딩된 블록들에 대한 리스트에 추가될 수 있다.

[0356] b. 일 예에서, 하나의 블록이 TPM으로 코딩될 때에도 아편 모드가 인에이블될 수 있다.

[0357] c. 대안적으로, 더욱이, 가중 영역에 위치된 것들을 제외한 모든 서브-블록들에 대해 단방향 예측이 이용된다는 것이 제한된다.

[0358] d. 대안적으로, 더욱이, 하나의 파티션 내의 상이한 서브-블록들의 모든 참조 화상들이 동일하다는 것이 제한된다.

[0359] e. 하나 또는 다수의 TPM 후보가 하나의 ATMVP 후보로부터 도출될 수 있는데, 예를 들어, 하나의 파티션 내의 각각의 서브-블록은 ATMVP 후보들의 List 0 모션 정보의 모션 정보를 상속할 수 있다.

[0360] f. 서브-블록 TPM 코딩된 블록들에 대한 병합 리스트는 서브-블록 병합 후보 리스트 구성 프로세스에서 사용되는 동일한 공간 및/또는 시간 블록들을 검사할 수 있다.

[0361] 5. 일반화된 양방향 예측이 인에이블될 때, 양방향 예측 정규 모션 후보가 2개의 가중치와 추가로 연관된다. 그러나, 그러한 양방향 예측 정규 모션 후보를 사용하여 TPM 후보들을 도출할 때, 가중치들을 상속하지 않는 것이 제안된다.

[0362] a. 일 예에서, 하나의 TPM 후보는 어떠한 가중치와도 연관되지 않는다. 따라서, 비-가중 영역에 위치된 샘플들의 예측 값들을 도출할 때, 모션 보상된 결과들이 직접 사용된다. 가중 영역에 위치된 샘플들에 대해, 선택된 가중치 그룹이 이용된다.

[0363] b. 대안적으로, 가중치들은 상속된다. 대안적으로, 더욱이, 가중 영역에 위치된 샘플들의 최종 예측 값들을 생성할 때, 상속된 가중치 중 하나가 영역에 대한 대응하는 TPM 후보로부터 도출된 예측 값에 적용될 수 있다.

[0364] 6. 하나의 TPM 코딩된 블록의 가중 영역에 위치되지 않은 샘플들에 대해 양방향 예측이 인에이블될 수 있는 것이 제안된다.

[0365] a. 일 예에서, 2개의 참조 화상이 동일한 것이 요구된다.

[0366] b. 일 예에서, 각각의 파티션에 대해, 하나의 참조 화상이 할당될 수 있다(예를 들어, 현재 설계에 따라, 2개의 파티션에 대해 동일하거나 상이할 수 있다). 한편, 제3 참조 화상 및 모션 벡터가 추가로 할당될 수 있다.

[0367] 7. 다중-가설이 TPM 코딩된 블록들에 적용될 수 있다는 것이 제안된다.

[0368] a. 일 예에서, 현재 설계를 사용하는 하나의 제1 최종 예측 블록이 먼저 생성될 수 있다. 다른 참조 화상 및 모션 벡터에 기초하여, 단방향 예측으로 제2 예측 블록이 또한 생성된다. 그리고 제1 최종 예측 블록 및 제2 예측 블록으로부터 하나의 블록의 최종 예측 블록이 도출된다.

[0369] b. 대안적으로, 현재 설계를 사용하는 하나의 제1 최종 예측 블록이 먼저 생성될 수 있다. 2개의 추가적인 참조 화상 및 모션 벡터에 기초하여, 양방향 예측으로 제2 예측 블록이 또한 생성된다. 그리고 제1 최종

예측 블록 및 제2 예측 블록으로부터 하나의 블록의 최종 예측 블록이 도출된다.

[0370] c. 대안적으로, 각각의 파티션에 대해, 양방향 예측 또는 다중-가설 예측이 먼저 적용된다. 그 후 가중 영역에 대해, 이중/다중-가설 예측으로 인한 예측 블록들의 가중 결과들이 최종 예측 값들로서 사용된다.

[0371] 8. 하나의 블록에 대해 TPM이 인에이블될 때 OBMC는 디스에이블될 수 있다.

[0372] a. 대안적으로, OBMC는 가중 영역에 위치된 샘플들에 대해 디스에이블되지만, 비-가중 영역에 위치된 샘플들에 대해 인에이블될 수 있다.

[0373] 9. 하나의 블록은 2개보다 많은 비-정사각형/비-삼각형 파티션으로 분할될 수 있다.

[0374] a. 하나의 블록은 4개의 비-정사각형/비-삼각형 파티션으로 분할될 수 있다. 일 예에서, 예가 도 29로서 묘사되어 있다.

[0375] b. 2개보다 많은 비-정사각형/비-삼각형 파티션이 존재할 때, 이들 중 2개만이 인터-코딩될 수 있는 것으로 제한된다.

[0376] c. 2개의 이웃 파티션의 예지 주위에 위치된 샘플들에 대해, 블렌딩(예를 들어, 2개의 예측 블록의 가중 평균)이 적용될 수 있다.

[0377] 도 29는 하나의 블록 내의 다수의 기하학적 파티션의 예를 보여준다.

TPM에 의해 요구되는 사이드 정보의 시그널링

[0379] 10. 분할 패턴들, 2개의 병합 인덱스들 및 코딩된 값(예를 들어, g_TriangleCombination) 사이의 맵핑은 시퀀스와 시퀀스 간에, 화상과 화상 간에, 타일과 타일 간에, 슬라이스와 슬라이스 간에, CTU와 CTU 간에, 블록과 블록 간에 적응적으로 변경될 수 있다.

[0380] a. 일 예에서, 맵핑은 구성된 병합 후보 리스트에 의존할 수 있다.

[0381] b. 일 예에서, 맵핑은 병합 후보들과 연관된 코딩된 모드에 의존할 수 있다.

[0382] 11. TPM 사용의 지시 플래그는 병합 플래그 전에 코딩될 수 있다. 즉, TPM은 또한 비-병합 모드에 대해 인에이블될 수 있다.

[0383] a. 대안적으로, 더욱이, 블록이 TPM 코딩되면 병합 플래그가 시그널링되지 않을 수 있다.

[0384] b. 대안적으로, 더욱이, 블록이 인터 코딩되면(예를 들어, 그것이 AMVP 또는 병합 모드임을 지시하기 위해) 병합 플래그가 추가로 시그널링될 수 있다.

[0385] c. TPM은 비-스킵 모드에 대해서만 허용될 수 있다.

[0386] 12. 하나의 블록이 AMVP 모드로 코딩되면 TPM 사용의 지시 플래그가 코딩될 수 있다.

[0387] a. 대안적으로, 더욱이, 하나의 블록이 병합 모드로 코딩되면 TPM 플래그가 시그널링되지 않을 수 있다.

[0388] b. 대안적으로, 더욱이, TPM 사용 플래그는 공간 또는 시간 이웃 또는 비-인접 블록들로부터 상속될 수 있다.

[0389] 13. 분할 패턴은 공간 이웃 블록들로부터 상속될 수 있다.

[0390] a. 대안적으로, 분할 패턴은 시간 블록들로부터 상속될 수 있다.

[0391] b. 일 예에서, 분할 패턴은 모션 정보와 함께 저장된다. 하나의 정규 모션 후보로부터 도출된 각각의 TPM 후보에 대해, 정규 모션 후보와 연관된 분할 패턴도 상속된다.

[0392] i. 이 경우, TPM 코딩된 블록은 모션 정보(또는 인코딩 모드) 및 분할 패턴 둘 다를 이웃 TPM 코딩된 블록으로부터 상속할 수 있다.

[0393] 14. 기하학적 파티션들을 위해 사용되는 분할 패턴들 및 병합 인덱스들이 개별적으로 시그널링될 수 있는 것이 제안된다.

[0394] a. 일 예에서, 1 비트 플래그가 개별적으로 코딩되어 TPM 분할 패턴을 지시할 수 있다.

- [0395] i. 이웃 블록들의 TPM 분할 패턴은 비트를 코딩하기 위한 컨텍스트를 도출하기 위해 사용될 수 있다.
- [0396] b. 일 예에서, 각각의 파티션에 대한 다수의 병합 인덱스가 개별적으로 코딩될 수 있다.
- [0397] i. 대안적으로, 더욱이, 상이한 파티션의 병합 인덱스들 간의 예측이 이용될 수 있다.
- [0398] c. 일 예에서, 병합 인덱스의 코딩은 분할 패턴에 대한 코딩된 1비트 플래그에 의존할 수 있다.
- [0399] d. 대안적으로, 병합 인덱스들이 먼저(공동으로 또는 개별적으로) 코딩될 수 있고, 분할 패턴에 대한 1비트 플래그가 병합 인덱스들에 따라 코딩될 수 있다.
- [0400] 15. TPM 사용의 지시 플래그에 대해 바이패스 코딩을 사용하는 것이 제안된다.
- [0401] a. 대안적으로, 플래그는 여전히 컨텍스트들로 코딩되지만, 컨텍스트는 블록 폭/높이에 의존한다.
- [0402] b. 대안적으로, 플래그는 여전히 컨텍스트들로 코딩되지만, 컨텍스트는 코딩된 깊이(예를 들어, 퀴드-트리 깊이, 바이너리 트리 및/또는 트리플 트리 깊이)에 의존한다.
- [0403] c. 대안적으로, 플래그는 여전히 컨텍스트들로 코딩되지만, 컨텍스트는 블록 형상(예를 들어, 정사각형 또는 비-정사각형) 및/또는 블록 폭과 높이들 간의 비율들(예를 들어, 1과 동등, $\frac{1}{4}$ 또는 4와 동등, $\frac{1}{2}$ 또는 2와 동등)과 같은 블록 폭/높이에 의존한다.
- [0404] d. 대안적으로, 플래그는 여전히 컨텍스트들로 코딩되지만, 컨텍스트는 하나의 블록 내의 왼쪽 상단 위치의 옆에 항상 있는 것은 아닌 이웃 블록들의 코딩된 정보에 의존한다. 블록 폭 및 높이를 각각 W 및 H로 나타내고, (0, 0)은 현재 블록 내의 왼쪽 상단 샘플의 좌표이다.
- [0405] i. 일 예에서, 위의 이웃 블록들은 $(W/2, -1)$, $(W/2-1, -1)$, $(W-1, -1)$ 로서 정의될 수 있다.
- [0406] ii. 일 예에서, 왼쪽 이웃 블록들은 $(-1, H/2)$, $(-1, H/2-1)$, $(-1, H-1)$ 로서 정의될 수 있다.
- [0407] iii. 컨텍스트 선택에서 사용될 수 있는 위의 및 왼쪽의 이웃 블록들의 일부 조합들이 도 30에 묘사되어 있다.
- [0408] iv. 일 예에서, 위의 이웃 블록들만이 선택될 수 있다. 대안적으로, 더욱이, 그러한 경우는 블록의 폭과 높이 간의 비율이 1 또는 2와 같은 임계치보다 클 때 인에이블된다.
- [0409] v. 일 예에서, 왼쪽 이웃 블록들만이 선택될 수 있다. 대안적으로, 더욱이, 그러한 경우는 블록의 높이와 폭 간의 비율이 1 또는 2와 같은 임계치보다 클 때 인에이블된다.
- [0410] 도 30은 TPM 플래그 코딩에서 컨텍스트 선택을 위해 사용되는 이웃 블록들의 예들을 도시한다. 왼쪽 상단 예 (a)는 $(W/2, -1)$ 및 $(-1, H/2)$ 이다. 오른쪽 상단 예 (b)는 $(W/2-1, -1)$ 및 $(-1, H/2-1)$ 이다. 왼쪽 하단 예 (c)는 $(0, -1)$ 및 $(-1, H/2-1)$ 이다. 오른쪽 하단 예 (d)는 $(W-1, -1)$ 및 $(-1, H/2-1)$ 이다.
- [0411] 16. TPM 리스트 내의 2개의 후보 인덱스(예를 들어, a0 및 a1, b0 및 b1)가 공동으로 코딩(또는 다른 정보와 추가 결합)될 때, 코드워드는 블록과 블록 간에, CTU와 CTU 간에, 타일과 타일 간에, 화상과 화상 간에, 시퀀스와 시퀀스 간에 적응적으로 변경될 수 있다.
- [0412] a. 일 예에서, 코드워드는 2개의 인덱스와 연관된 2개의 TPM 후보가 동일한 정규 모션 후보로부터의 것인지에 기초하여 할당될 수 있다.
- [0413] i. 일 예에서, 다음이 발생하면, 코드워드 C(a0, a1)는 C(b0, b1)보다 작을 수 있다.
- [0414] 1. 인덱스가 a0과 동등한 TPM 후보 및 인덱스가 a1과 동등한 TPM 후보는 상이한 정규 모션 후보로부터 도출되고, 인덱스가 b0과 동등한 TPM 후보 및 인덱스가 b1과 동등한 TPM 후보는 동일한 정규 모션 후보로부터 도출된다.
- [0415] b. 일 예에서, 코드워드는 2개의 인덱스와 연관된 2개의 TPM 후보가 동일한 예측 방향(또는 참조 화상 리스트)으로부터 유래한 것인지에 기초하여 할당될 수 있다.
- [0416] i. 일 예에서, 다음이 발생하면, 코드워드 C(a0, a1)는 C(b0, b1)보다 작을 수 있다.

- [0417] 1. 인덱스가 a0과 동등한 TPM 후보 및 인덱스가 a1과 동등한 TPM 후보는 2개의 예측 방향으로부터 유래한 것이고, 인덱스가 b0과 동등한 TPM 후보 및 인덱스가 b1과 동등한 TPM 후보는 동일한 예측 방향으로부터 유래한 것이다.
- [0418] 17. TPM에서 분할된 2개의 파티션은 상이한 모드들로 코딩될 수 있다.
- [0419] a. 일 예에서, 하나는 인트라-코딩되고 다른 하나는 인터-코딩된다.
- [0420] b. 다른 예에서, 하나는 병합-코딩되고, 다른 하나는 AMVP 코딩된다.
- [0421] TPM의 인에이블링
- [0422] 18. TPM은 TPM 사용 플래그가 시그널링되지 않고 컬러 성분에 대해 자동으로 디스에이블될 수 있다.
- [0423] a. 일 예에서, 크로마 블록들에 대해, 폭 또는 높이가 임계치(예를 들어, 2)보다 작거나 그와 동등하면.
- [0424] 19. TPM은 크기가 $N \times M$ 또는 $L \times N$ (예를 들어, N은 4와 동등함)과 동등한 블록에 대해 디스에이블될 수 있다.
- [0425] a. 일 예에서, M 및 N은 8, 16으로 설정된다.
- [0426] b. 대안적으로, 더욱이, 그것은 컬러 포맷이 4:2:0과 동등할 때 디스에이블된다.
- [0427] c. 일 예에서, 폭 또는 높이 또는 폭과 높이 둘 다를 갖는 블록이 둘 다 임계치 L보다 큰(또는 그와 동등한) 경우, TPM은 디스에이블될 수 있다.
- [0428] vi. 일 예에서, L은 64이다.
- [0429] d. 일 예에서, 폭 또는 높이 또는 폭과 높이 둘 다를 갖는 블록이 둘 다 임계치 L보다 작은(또는 그와 동등한) 경우, 위의 방법은 디스에이블될 수 있다.
- [0430] vii. 일 예에서, L은 4 또는 8이다.
- [0431] e. 일 예에서, 그것은 폭/높이 또는 높이/폭이 임계치 L(예를 들어, L이 16)보다 클 때 디스에이블된다.
- [0432] f. 임계치 L은 사전 정의되거나 SPS/PPS/화상/슬라이스/타일 그룹/타일 레벨에서 시그널링될 수 있다.
- [0433] g. 대안적으로, 임계치들은 블록 크기, 화상 유형, 시간 계층 인덱스 등과 같은 특정 코딩된 정보에 의존할 수 있다.
- [0434] 20. TPM은 하나의 블록에 대한 상이한 컬러 성분들에 대해 인에이블 및 디스에이블될 수 있다.
- [0435] a. 일 예에서, $N \times M$ 또는 $L \times N$ (예를 들어, N은 4와 동등함)과 동등한 루마 블록 크기에 대해, TPM은 이 루마 블록에 대해 인에이블될 수 있지만, 그것은 크로마 블록에 대해, 예를 들어, 4:2:0 컬러 포맷에 대해 디스에이블될 수 있다.
- [0436] b. 일 예에서, 그것이 하나의 컬러 성분에 대해 디스에이블되면, 연관된 예측 블록은 하나의 파티션, 예를 들어, 제1 파티션의 모션 정보 또는 예측 모드에 기초하여 도출될 수 있다.
- [0437] c. 일 예에서, 그것이 하나의 컬러 성분에 대해 디스에이블되면, 연관된 예측 블록은 2개의 파티션의 모션 정보에 기초하여 직접 도출될 수 있다.
- [0438] viii. 일 예에서, 2개의 파티션의 모션 정보가 양방향 예측 모션 벡터들에 대해 형성될 수 있는데, 예를 들어, 제1 파티션에 대한 Mv1 및 제2 파티션에 대한 Mv2는 상이한 방향들로부터의 모션 벡터들을 갖는다.
- [0439] ix. 일 예에서, 2개의 파티션의 모션 정보가 단방향 예측 모션 벡터들에 대해 형성될 수 있는데, 예를 들어, 제1 파티션에 대한 Mv1 및 제2 파티션에 대한 Mv2는 동일한 방향들로부터의 모션 벡터들을 갖고, 파티션 A의 하나의 모션 벡터는 파티션 B의 다른 참조 화상으로 스케일링될 수 있고, 파티션 B의 모션 벡터와 파티션 A의 스케일링된 MV의 평균이 사용될 수 있다.
- [0440] x. 일 예에서, 2개의 파티션의 모션 정보가 양방향 예측 모션 벡터들에 대해 형성될 수 있는데, 예를 들어, 제1 파티션에 대한 Mv1 및 제2 파티션에 대한 Mv2는 동일한 방향들로부터의 모션 벡터들을

갖고, 파티션 A의 하나의 모션 벡터는 다른 방향에서 다른 참조 화상으로 스케일링될 수 있고, 파티션 B의 모션 벡터와 파티션 A의 스케일링된 MV가 사용될 수 있다.

- [0441] xi. 위의 예에서, 모션 정보는 컬러 포맷에 기초하여 추가로 스케일링될 수 있다.
- [0442] 21. 위에 언급된 방법들에 대해, 그것들을 인에이블할지 및/또는 어떻게 인에이블할지는 SPS/PPS/VPS/화상 헤더/슬라이스 헤더/타일 그룹 헤더/CTU 행들/CTU들/영역들의 그룹에서 시그널링될 수 있다.
 - a. 일 예에서, 폭 또는 높이 중 어느 하나 또는 폭과 높이 둘 다, 또는 폭*높이가 둘 다 임계치 L보다 큰(또는 그와 동등한) 블록의 경우, 위의 방법은 디스에이블될 수 있다.
- [0443] i. 일 예에서, L은 64, 4096이다.
- [0444] b. 일 예에서, 폭 또는 높이 중 어느 하나 또는 폭과 높이 둘 다, 또는 폭*높이가 둘 다 임계치 L보다 작은(또는 그와 동등한) 블록의 경우, 위의 방법은 디스에이블될 수 있다.
- [0445] i. 일 예에서, L은 4 또는 8, 64이다.
- [0446] c. 임계치 L은 사전 정의되거나 SPS/PPS/화상/슬라이스/타일 그룹/타일 레벨에서 시그널링될 수 있다.
- [0447] d. 대안적으로, 임계치들은 블록 크기, 화상 유형, 시간 계층 인덱스 등과 같은 특정 코딩된 정보에 의존할 수 있다.
- [0448] e. 일 예에서, TPM을 적용할지는 형상(예를 들어, 그것이 정사각형 또는 비-정사각형)과 같은 블록 폭/높이 및/또는 블록 폭과 높이들 간의 비율들(예를 들어, 1, $\frac{1}{4}$ 또는 4, $\frac{1}{2}$ 또는 2)에 의존한다.
- [0449] f. TPM을 스kip/병합 모드와 상이한 모드로 취급하는 것이 제안된다. 따라서, 하나의 블록에 대해 TPM이 인에이블될 때, 연관된 병합 플래그는 거짓으로 시그널링된다.
- [0450] 22. TPM을 스kip/병합 모드와 상이한 모드로 취급하는 것이 제안된다. 따라서, 하나의 블록에 대해 TPM이 인에이블될 때, 연관된 병합 플래그는 거짓으로 시그널링된다.
- [0451] 23. MHIntra를 스kip/병합 모드와 상이한 모드로서 취급하는 것이 제안된다. 따라서, 하나의 블록에 대해 MHIntra가 인에이블될 때, 연관된 병합 플래그는 거짓으로 시그널링된다.
- [0452] 24. 서브-블록 병합 리스트를 스kip/병합 모드와 상이한 모드로서 취급하는 것이 제안된다. 따라서, 하나의 블록에 대해 서브-블록 병합 리스트가 인에이블될 때, 연관된 병합 플래그는 거짓으로 시그널링된다.
- [0453] 25. 블릿들 22-24의 방법들에 대해, 하나의 블록이 병합/스킵 모드로 디코딩될 때, TPM/MHIntra/서브-블록 병합 리스트의 지시들의 시그널링은 스kip되고 거짓인 것으로 추론된다.
- [0454] 26. TPM/MHIntra/서브-블록 병합 리스트/MMVD의 사용의 지시들의 코딩 순서는 다음과 같은 것이 제안된다:
 - a. MMVD, 서브-블록 병합 리스트, TPM, MHIntra
 - b. MMVD, 서브-블록 병합 리스트, MHIntra, TPM
 - c. 다른 종류의 순서들.
- [0455] 27. TPM/MHIntra/서브-블록 병합 리스트/MMVD의 사용의 지시들의 코딩 순서는, 블록/CU/PU/CTU/LCU 행/영역/슬라이스/화상/시퀀스와 같은, 하나의 비디오 단위와 다른 비디오 단위 간에 상이할 수 있다는 것이 제안된다.
- [0456] a. 일 예에서, 순서는 저지연 검사 플래그에 의존할 수 있다.
- [0457] b. 일 예에서, 순서는 이전에 코딩된 정보에 의존할 수 있다.
- [0458] c. 일 예에서, 순서는 하나의 블록의 치수에 의존할 수 있다.
- [0459] 28. 라인 베퍼 크기를 절약하기 위해, 현재 블록의 AMVR 정보를 코딩하기 위해 이웃 블록의 AMVR 정보(즉, MV 정밀도)를 사용할 때, 이웃 블록이 상이한 LCU 행에 위치되면, 이웃 블록은 이용불가능한 것으로 표시된다.
 - a. 일 예에서, 연관된 AMVR 정보는 디폴트 값, 즉, $\frac{1}{4}$ -pel로 설정된다.
 - b. 대안적으로, 이웃 블록은 상이한 슬라이스/타일에 위치된 것들로서 취급된다.
 - c. 일 예에서, 이웃 블록이 상이한 영역(예를 들어, 전형적으로 64x64로 설정되는 비디오 처리 데이터

단위)에 위치되면, 그러한 블록은 위에 언급된 바와 동일한 방식으로 취급된다.

[0467] 29. 라인 베퍼 크기를 절약하기 위해, 현재 아핀-코딩된 블록의 AMVR 정보를 코딩하기 위해 이웃 아핀-코딩된 블록의 AMVR 정보(즉, MV 정밀도)를 사용할 때, 이웃 블록이 상이한 LCU 행에 위치되면, 이웃 아핀-코딩된 블록은 이용불가능한 것으로 표시된다.

a. 일 예에서, 연관된 AMVR 정보는 디폴트 값, 즉, $\frac{1}{4}$ -pel로 설정된다.

b. 대안적으로, 이웃 아핀-코딩된 블록은 상이한 슬라이스/타일에 위치된 것들로서 취급된다.

[0470] c. 일 예에서, 이웃 아핀-코딩된 아핀-코딩된 블록이 상이한 영역(예를 들어, 전형적으로 64x64로 설정되는 비디오 처리 데이터 단위)에 위치되면, 그러한 블록은 위에 언급된 바와 동일한 방식으로 취급된다.

[0471] 30. 이웃 블록이 상이한 LCU 행에 있다면 컨텍스트 모델링을 위해 이웃 블록의 정보를 이용하는 것의 사용을 디스에이블하는 것이 또한 제안된다. 대안적으로, 아래에 언급된 구문 요소를 코딩하기 위해 단 하나의 컨텍스트만이 사용된다. 대안적으로, 아래에 언급된 구문 요소를 코딩하기 위해 바이패스 코딩이 적용된다.

a. 정보는 alf_ctb_flag를 포함한다.

b. 정보는 qt_split_cu_flag를 포함한다.

c. 정보는 mtt_split_cu_flag를 포함한다.

d. 정보는 mtt_split_cu_flag를 포함한다.

e. 정보는 cu_skip_flag를 포함한다.

f. 정보는 amvr_mode를 포함한다.

g. 정보는 merge_subblock_flag를 포함한다.

h. 정보는 merge_triangle_flag를 포함한다.

i. 정보는 inter_affine_flag를 포함한다.

[0481] j. 정보는: cu_skip_flag, pred_mode_flag, pcm_flag, intra_luma_ref_idx, intra_luma_mpm_flag, intra_luma_mpm_idx, intra_luma_mpm_remainder, intra_chroma_pred_mode, merge_flag, inter_pred_idc, inter_affine_flag, cu_affine_type_flag, ref_idx_10, mvp_10_flag, ref_idx_11 mvp_11_flag, amvr_flag, amvr_4pel_flag, gbi_idx, cu_cbf, mmvd_flag, mmvd_merge_flag 등을 추가로 포함할 수 있다.

[0482] CABAC 컨텍스트 모델링의 단순화

[0483] 31. 컨텍스트 모델링에서 사용된 이웃 정보는 현재 블록을 디코딩하기 위해 사용되는 블록들로부터만 유래할 수 있다는 것이 제안된다.

[0484] a. 일 예에서, cu_skip_flag에 대해, 도 25에서 A3 및 B3으로 표시된 2개의 이웃 블록의 정보를 사용하는 대신에, 도 42에서의 2개의 다른 블록(예컨대 A1 및 B1)이 이용될 수 있다.

[0485] b. 대안적으로, 더욱이, 이웃 블록들이 현재 CTU/CTU 행/영역/비디오 데이터 처리 단위 또는 슬라이스/타일/타일 그룹들/화상 경계와 같은 다른 비디오 단위들 외부에 있을 때, 그러한 이웃 블록들로부터의 이웃 정보는 파싱 프로세스에서 이용되도록 허용되지 않는다.

[0486] c. 대안적으로, 더욱이, 이웃 블록들이 현재 CTU/CTU 행/영역/비디오 데이터 처리 단위 또는 슬라이스/타일/타일 그룹들/화상 경계와 같은 다른 비디오 단위들 외부에 있을 때, 그러한 이웃 블록들로부터의 이웃 정보는 이용불가능한 것으로 취급되고 파싱 프로세스에서 이용될 수 있다.

[0487] d. 대안적으로, 더욱이, 이웃 블록들이 현재 CTU/CTU 행/영역/비디오 데이터 처리 단위 또는 슬라이스/타일/타일 그룹들/화상 경계와 같은 다른 비디오 단위들 외부에 있을 때, 그러한 이웃 블록들로부터의 이웃 정보는 디폴트 값들로 설정되고 파싱 프로세스에서 이용될 수 있다.

[0488] e. 대안적으로, 더욱이, 이웃 블록들이 현재 CTU/CTU 행/영역/비디오 데이터 처리 단위 또는 슬라이스/타일/타일 그룹들/화상 경계와 같은 다른 비디오 단위들 외부에 있을 때, 현재 CTU/CTU 행/영역/비디오 데이터 처리 단위 또는 슬라이스/타일/타일 그룹들/화상 경계와 같은 다른 비디오 단위에 있는 다른 하나 이상의 이웃

블록으로부터의 이웃 정보가 파싱 프로세스에서 대체로서 이용될 수 있다.

[0489] 32. 컨텍스트 모델링에서 사용된 이웃 정보(예컨대 이웃 블록의 구문 요소들(이웃 요소들로서 축약됨))는 이웃 블록의 코딩된 정보에 의존하는 모든 구문 요소들에 대한 일부 사전 정의된 이웃 위치들로부터 유래한 것일 수 있다는 것이 제안된다. 일 예에서, 그러한 이웃 위치들은 또한 하나의 블록의 재구성 프로세스 동안 액세스될 수 있다.

[0490] a. 일 예에서, 도 2의 A1 및 B1로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0491] b. 일 예에서, 도 2의 A0 및 B0로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0492] c. 일 예에서, 도 2의 A1 및 B2로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0493] d. 일 예에서, 도 2의 A0 및 B1로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0494] e. 일 예에서, 도 2의 A0 및 B2로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0495] f. 일 예에서, 도 2의 A0 및 A1로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0496] g. 일 예에서, 도 2의 A1 및 B0으로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0497] h. 일 예에서, 도 2의 B0 및 B1로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0498] i. 일 예에서, 도 2의 B0 및 B2로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0499] j. 일 예에서, 도 2의 B1 및 B2로부터의 이웃 요소들이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0500] k. 일 예에서, 1개의 이웃 요소만이 모든 구문 요소들에 대한 컨텍스트 모델링에서 사용된다.

[0501] 33. 현재 CTU 행 또는 CTU와 상이한 CTU 행 또는 CTU 내의 이웃 블록으로부터의 이웃 요소들이 컨텍스트 모델링에서 사용되지 않을 수 있는 것이 제안된다.

[0502] a. 대안적으로, 현재 CTU 행 또는 CTU와 상이한 CTU 행 또는 CTU 내의 이웃 블록으로부터의 이웃 요소들이 컨텍스트 모델링에서 사용되지 않을 수 있다.

[0503] b. 일 예에서, 이웃 요소가 현재 CTU 행 또는 CTU와 상이한 CTU 행 또는 CTU 내의 이웃 블록으로부터 유래한 것이면, 그것은 이용불가능한 것으로 간주된다.

[0504] c. 일 예에서, 이웃 요소가 현재 CTU 행 또는 CTU와 상이한 CTU 행 또는 CTU 내의 이웃 블록으로부터 유래한 것이면, 디폴트 값이 사용된다.

[0505] d. 일 예에서, 이웃 요소가 현재 CTU 행 또는 CTU와 상이한 CTU 행 또는 CTU 내의 이웃 블록으로부터 유래한 것이면, 그것은 대체되어야 하는데 그 이유는 다른 이웃 요소가 현재 CTU 행 또는 CTU와 상이한 CTU 행 또는 CTU 내의 다른 이웃 블록으로부터 유래한 것이기 때문이다.

[0506] e. 일 예에서, 이웃 요소가 현재 CTU 행 또는 CTU와 상이한 CTU 행 또는 CTU 내의 이웃 블록으로부터 유래한 것이면, 동일한 CTU 행 또는 CTU 내에 있는 N($N > 0$)개의 대안적인 이웃 요소들은 하나의 이용가능한 이웃 요소가 발견되거나, 모든 N개의 이웃 요소가 이용불가능할 때까지 순서대로 추가로 검사될 수 있다. 예를 들어, N은 1, 2, 3, 또는 4와 동등할 수 있다.

[0507] 34. 제안된 방법은 alf_ctb_flag, qt_split_cu_flag, mtt_split_cu_flag, cu_skip_flag, amvr_mode, inter_affine_flag, merge_triangle_flag and merge_subblock_flag, cu_skip_flag, pred_mode_flag,

pcm_flag, intra_luma_ref_idx, intra_luma_mpm_flag, intra_luma_mpm_idx, intra_luma_mpm_remainder, intra_chroma_pred_mode, merge_flag, inter_pred_idc, inter_affine_flag, cu_affine_type_flag, ref_idx_10, mvp_10_flag, ref_idx_11 mvp_11_flag, amvr_flag, amvr_4pel_flag, gbi_idx, cu_cbf, mmvd_flag, mmvd_merge_flag, pred_mode_flag 등과 같은 일부 구문 요소들에만 적용될 수 있다.

[0508]

a. 일 예에서, pred_mode_flag의 컨텍스트 모델링은 다음의 방법들로부터 도출된다:

컨텍스트(ctx) 도출의 설명	
방법 #1 (3 개의 컨텍스트)	ctx = A1 이 인트라 ? 1 : 0 ctx += B1 이 인트라 ? 1 : 0
방법 #2 (2 개의 컨텍스트)	ctx = (A1 이 인트라) (B1 이 인트라) ? 1 : 0
방법 #3 (3 개의 컨텍스트)	ctx = A1 이 인트라 또는 인터-인트라 ? 1 : 0 ctx += B1 이 인트라 또는 인터-인트라 ? 1 : 0
방법 #4 (2 개의 컨텍스트)	ctx = (A1 이 인트라 또는 인터-인트라) (B1 이 인트라 또는 인터-인트라) ? 1 : 0

[0509]

b. 대안적으로, pred_mode_flag의 컨텍스트 모델링은 A1 및/또는 B1 대신에 다른 이웃 블록들, 예컨대도 25에서 A2/B3 또는 A0/B0으로부터 도출될 수 있다.

[0510]

c. 위의 구문 pred_mode_flag는 다른 종류의 구문 요소들로 대체될 수 있다.

[0511]

d. 일 예에서, 하나의 이웃 블록이 이용불가능하면, 그것은 컨텍스트를 도출할 때 인터-코딩된 것으로서 취급된다;

[0512]

e. 일 예에서, 하나의 이웃 블록이 이용불가능하면, 그것은 컨텍스트를 도출할 때 인트라-코딩된 것으로서 취급된다;

[0513]

f. 일 예에서, 하나의 이웃 블록이 이용불가능하면, 그것은 컨텍스트를 도출할 때 인터-인트라-코딩된 것으로서 취급된다;

[0514]

g. 일 예에서, 하나의 이웃 블록이 현재 블록과 동일한 CTU 행에 있지 않다면, 그것은 컨텍스트를 도출할 때 인터-코딩된 것으로 취급된다;

[0515]

h. 일 예에서, 하나의 이웃 블록이 현재 블록과 동일한 CTU 행에 있지 않다면, 그것은 컨텍스트를 도출할 때 인트라-코딩된 것으로 취급된다;

[0516]

i. 일 예에서, 하나의 이웃 블록이 현재 블록과 동일한 CTU 행에 있지 않다면, 그것은 컨텍스트를 도출할 때 인터-인트라-코딩된 것(조합된 인트라-인터 예측 모드라고도 알려짐)으로서 취급된다;

[0517]

5. 추가적인 실시예 예들

[0518]

5.1 실시예 #1

[0519]

TPM을 병합 모드와 상이한 모드로서 시그널링하는 예들은 다음과 같이 주어진다. 새롭게 추가된 부분들은 볼드체 이탤릭체로 강조되고, 삭제된 부분은 대문자로 표시된다.

[0521]

7.3.4.6 코딩 단위 구문

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	서술자
if(slice_type != I) {	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
...	
} else /* MODE_INTER */	
if(cu_skip_flag[x0][y0] == 0) {	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else {	
if(sps_triangle_enabled_flag && slice_type == B && cbWidth * cbHeight >= 16)	
merge_triangle_flag[x0][y0]	ae(v)
if(merge_triangle_flag[x0][y0])	
merge_triangle_idx[x0][y0]	ae(v)
else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	

[0522]

}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp_l1_flag[x0][y0]	ae(v)
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	
if(sps_amvr_enabled_flag && inter_affine_flag == 0 &&	
(MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0	
MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0))	
amvr_mode[x0][y0]	ae(v)
if(sps_gbi_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI &&	
cbWidth * cbHeight >= 256)	
gbi_idx[x0][y0]	ae(v)
}	
}	
}	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0)	
cu_cbf	ae(v)

[0523]

if(cu_cbf)	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	
}	

[0524]

[0525]

7.3.4.8 병합 테이터 구문

코드	설명
merge_data(x0, y0, cbWidth, cbHeight) {	
mmvd_flag[x0][y0]	ae(v)
if(mmvd_flag[x0][y0] == 1) {	
mmvd_merge_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)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if(sps_mh_intra_enabled_flag && cu_skip_flag[x0][y0] == 0 &&	
(cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
mh_intra_flag[x0][y0]	ae(v)
if(mh_intra_flag[x0][y0]) {	
if(cbWidth <= 2 * cbHeight cbHeight <= 2 * cbWidth)	
mh_intra_luma_mpm_flag[x0][y0]	ae(v)
if(mh_intra_luma_mpm_flag[x0][y0])	
mh_intra_luma_mpm_idx[x0][y0]	ae(v)
}	
}	
IF(SPS_TRIANGLE_ENABLED_FLAG && SLICE_TYPE == B &&	
CBWIDTH * CBHEIGHT >= 16)	
MERGE_TRIANGLE_FLAG[X0][Y0]	AE(V)
IF(MERGE_TRIANGLE_FLAG[X0][Y0])	
MERGE_TRIANGLE_IDX[X0][Y0]	AE(V)
else if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
}	

[0526]

5.2 실시예 #2

[0527]

이 실시예에서, 라인 베퍼 크기는 컨텍스트 모델링을 위해 감소된다. 변화들은 다음과 같이 볼드체 이탤릭체로 강조된다.

[0528]

9.5.4.2.2 왼쪽 및 위의 구문 요소들을 사용하는 ctxInc의 도출 프로세스

[0529]

이 프로세스에 대한 입력은 현재 화상의 왼쪽 상단 샘플에 대한 현재 루마 블록의 왼쪽 상단 루마 샘플, 컬러 성분 cIdx, 현재 코딩 큐드트리 깊이 cqDepth, 및 루마 샘플들에서의 현재 코딩 블록의 폭 및 높이 cbWidth 및 cbHeight를 특정하는 루마 위치 (x0, y0)이다.

[0530]

이 프로세스의 출력은 ctxInc이다.

[0531]

위치 (xNbL, yNbL)은 (x0 - 1, y0)과 동등하게 설정되고, 현재 블록의 왼쪽에 직접 위치되는 블록의 이용가능성을 특정하는 변수 availableL은, (x0, y0)과 동등하게 설정된 위치(xCurr, yCurr) 및 (xNbL, yNbL)과 동등하게 설정된 이웃 위치 (xNbY, yNbY)를 입력들로서 갖는 하위 조항 6.4에 특정된 바와 같은 z-스캔 순서로 블록에 대한 이용가능성 도출 프로세스를 호출하는 것에 의해 도출되고, 출력은 availableL에 할당된다.

[0532]

위치 (xNbA, yNbA)는 (x0, y0 - 1)과 동등하게 설정되고, 현재 블록의 위에 직접 위치되는 코딩 블록의 이용가능성을 특정하는 변수 availableA는, (x0, y0)과 동등하게 설정된 위치(xCurr, yCurr) 및 (xNbA, yNbA)와 동등하게 설정된 이웃 위치(xNbY, yNbY)를 입력들로서 갖는 하위 조항 6.4에 특정된 바와 같은 z-

캔 순서로 블록에 대한 이용가능성 도출 프로세스를 호출하는 것에 의해 도출되고, 출력은 availableA에 할당된다.

[0534] $(x0, y0 -1) \leq (x0, y0)$ 와 비교하여 상이한 LCU 행에 있다면 availableA는 거짓으로 리셋된다.

[0535] 변수 sizeC, sizeTh2 및 sizeTh1은 다음과 같이 도출된다:

$$\text{sizeTh2} = (\text{MaxBtSizeY} == 128) ? 1024 : ((\text{MaxBtSizeY} == 64) ? 512 : 256) \quad (9-19)$$

$$\text{sizeTh1} = (\text{MaxBtSizeY} == 128) ? 128 : 64 \quad (9-20)$$

$$\text{sizeC} = \text{cbWidth} * \text{cbHeight} \quad (9-21)$$

[0536] 표 9-11에 특정된 구문 요소들 alf_ctb_flag[x0][y0][cIdx], qt_split_cu_flag[x0][y0], mtt_split_cu_flag[x0][y0], cu_skip_flag[x0][y0], amvr_mode[x0][y0], inter_affine_flag[x0][y0], merge_triangle_flag[x0][y0] and merge_subblock_flag[x0][y0]에 대한 condL 및 condA를 갖는 ctxInc의 할당이 다음과 같이 특정된다:

$$\text{ctxInc} = (\text{condL} \&\& \text{availableL}) + (\text{condA} \&\& \text{availableA}) + \text{ctxSetIdx} * 3 \quad (9-22)$$

[0539] 표 9-11 - 왼쪽 및 위의 구문 요소들을 사용하는 ctxInc의 사양

구문 요소	condL	condA	ctxSetIdx
alf_ctb_flag[x0][y0][cld x]	alf_ctb_flag[xNbL][yNbL][cIdx]	alf_ctb_flag[xNbA][yNbA][cIdx]	cIdx
qt_split_cu_flag	cqtDepth[xNbL][yNbL] > cqtDepth	cqtDepth[xNbA][yNbA] > cqtDepth	(cqtDepth < 2) ? 0 : 1
mtt_split_cu_flag treeType == DUAL_TREE_CHROMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth	3
mtt_split_cu_flag treeType != DUAL_TREE_CHROMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth	(sizeC > sizeTh2) ? 0 : ((sizeC > sizeTh1) ? 1 : 2)
cu_skip_flag[x0][y0]	cu_skip_flag[xNbL][yNbL]	cu_skip_flag[xNbA][yNbA]	0
amvr_mode[x0][y0]	amvr_mode[xNbL][yNbL]	amvr_mode[xNbA][yNbA]	0
merge_subblock_flag[x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0
merge_triangle_flag[x0][y0]	merge_triangle_flag[xNbL][yNbL]	merge_triangle_flag[xNbA][yNbA]	0
inter_affine_flag[x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0

[0540]

[0541] 도 31은 비디오 처리 장치(3100)의 블록도이다. 장치(3100)는 본 명세서에서 설명된 방법들 중 하나 이상을 구현하기 위해 사용될 수 있다. 장치(3100)는 스마트폰, 태블릿, 컴퓨터, 사물 인터넷(IoT) 수신기 등에서 구현될 수 있다. 장치(3100)는 하나 이상의 프로세서(3102), 하나 이상의 메모리(3104) 및 비디오 처리 하드웨어(3106)를 포함할 수 있다. 프로세서(들)(3102)는 본 문서에서 설명된 하나 이상의 방법을 구현하도록 구성될 수 있다. 메모리(메모리들)(3104)는 본 명세서에서 설명된 방법들 및 기법들을 구현하기 위해 사용되는 코드 및 데이터를 저장하기 위해 사용될 수 있다. 비디오 처리 하드웨어(3106)는 본 문서에서 설명된 일부 기법들을 하드웨어 회로로 구현하기 위해 사용될 수 있다.

[0542] 도 33은 비디오를 처리하는 방법(3300)에 대한 흐름도이다. 이 방법(3300)은 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계(3305), 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계(3310) - 상기 제1 MV 예측 후보는 서브-블록 기반 모션 후보로부터 도출됨 -, 및 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계(3315)를 포함한다.

[0543] 도 34는 비디오를 처리하는 방법(3400)에 대한 흐름도이다. 이 방법(3400)은 제1 분할 패턴에 기초한 예측 부분들을 포함하도록 제1 비디오 블록을 파티셔닝하는 단계(3405) - 상기 예측 부분들은 제1 병합 인덱스 및 제1 코딩된 값과 연관됨 -, 제2 분할 패턴에 기초한 예측 부분들을 포함하도록 제2 비디오 블록을 파티셔닝하는 단

계(3410) - 상기 제2 비디오 블록의 예측 부분들은 제2 병합 인덱스 및 제2 코딩된 값과 연관되고, 상기 제1 코딩된 값은 상기 제2 코딩된 값과 동등하고, 상기 제2 비디오 블록의 제2 병합 인덱스 또는 제2 분할 패턴 중 하나 이상은 상기 제2 비디오 블록이 상기 제1 비디오 블록과 상이한 시퀀스, 화상, 타일, 슬라이스, 코딩 트리 단위(CTU) 또는 블록 내에 있는 것에 기초하여 상기 제1 비디오 블록과 상이함 -, 및 상기 제1 비디오 블록 및 상기 제2 비디오 블록의 추가 처리를 수행하는 단계(3415)를 포함한다.

[0544] 도 35는 비디오를 처리하는 방법(3500)에 대한 흐름도이다. 이 방법(3500)은 제1 비디오 블록의 특성을 식별하는 단계(3505), 상기 특성들에 기초하여 상기 제1 비디오 블록에 적용될 기하학적 예측 모드의 상태를 결정하는 단계(3510) - 상기 기하학적 예측 모드의 상태는: 상기 제1 비디오 블록에 기하학적 예측 모드를 적용하는 인에 이를 상태, 또는 상기 제1 비디오 블록에 기하학적 예측 모드를 적용하지 않는 디스에이블 상태 중 하나임 -, 및 상기 기하학적 예측 모드의 상태와 부합하는 상기 제1 비디오 블록의 추가 처리를 수행하는 단계(3515)를 포함한다.

[0545] 도 36은 비디오를 처리하는 방법(3600)에 대한 흐름도이다. 이 방법(3600)은 제1 비디오 블록이 제1 예측 부분 및 제2 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계(3605) - 상기 제1 예측 부분 및 상기 제2 예측 부분은 비-직사각형 및 비-정사각형임 -, 상기 제1 예측 부분과 연관된 제1 MV 예측 후보 및 상기 제2 예측 부분과 연관된 제2 MV 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계(3610) - 상기 제1 MV 예측 후보 및 상기 제2 MV 예측 후보는 이전에 코딩된 블록과 연관된 단일 MV 예측 후보로부터 도출됨 -, 및 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계(3615)를 포함한다.

[0546] 도 37은 비디오를 처리하는 방법(3700)에 대한 흐름도이다. 이 방법(3700)은 프로세서에 의해, 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계; 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계; 상기 제1 비디오 블록에 대한 최종 예측 블록을 도출하기 위해 사용되는 상기 MV 후보들을 리파인 하도록 상기 제1 MV 예측 후보를 수정하는 단계; 및 상기 최종 예측 블록을 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계를 포함한다.

[0547] 도 38은 비디오를 처리하는 방법(3800)에 대한 흐름도이다. 이 방법(3800)은 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계(3805); 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계(3810) - 상기 제1 MV 예측 후보는 가중치들을 갖는 양방향 예측 MV 예측 후보로부터 도출됨 -; 및 상기 가중치들 없이 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계(3815)를 포함한다.

[0548] 도 39는 비디오를 처리하는 방법(3900)에 대한 흐름도이다. 이 방법(3900)은 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝되고, 상기 제1 비디오 블록이 TPM 및 다중-가설과 부합하도록 코딩되어 있다는 결정을 수행하는 단계(3905); 상기 제1 예측 부분과 연관된 제1 모션 벡터(MV) 예측 후보를 상기 제1 비디오 블록과 연관된 모션 후보 리스트에 추가하는 단계(3910); 및 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계(3915)를 포함한다.

[0549] 도 40은 비디오를 처리하는 방법(4000)에 대한 흐름도이다. 이 방법(4000)은 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제1 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계(4005) - 오버랩 블록 모션 보상(OBMC)이 적용되지 않음 -; 및 상기 제1 예측 부분을 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계(4010)를 포함한다.

[0550] 도 41은 비디오를 처리하는 방법(4100)에 대한 흐름도이다. 이 방법(4100)은 제1 비디오 블록이 2개보다 많은 예측 부분을 포함하도록 파티셔닝된다는 결정을 수행하는 단계(4105) - 상기 수행하는 단계는 상기 제1 비디오 블록이 각각 비-직사각형 및 비-정사각형인 제1 예측 부분, 제2 예측 부분, 제3 예측 부분, 및 제4 예측 부분으로 파티셔닝된다고 결정하는 것을 포함함 -; 및 상기 제1 예측 부분, 상기 제2 예측 부분, 상기 제3 예측 부분, 및 상기 제4 예측 부분을 사용하여 상기 제1 비디오 블록의 추가 처리를 수행하는 단계(4110)를 포함한다.

[0551] 방법들 3300, 3400, 3500, 3600, 3700, 3800, 3900, 4000, 및 4100을 참조하여, 인코딩에 대한 후보 및 그들의 사용을 결정하는 일부 예들이 본 문서의 섹션 4에 설명되어 있다. 예를 들어, 섹션 4에 설명된 바와 같이, 비디오 블록들은 비-정사각형 및 비-직사각형인 예측 부분들을 사용하여 처리될 수 있다.

- [0552] 방법들 3300, 3400, 3500, 3600, 3700, 3800, 3900, 4000, 및 4100을 참조하여, 비디오 블록은 모션 정보 예측과 관련된 비트스트림 생성 규칙을 사용하는 것에 의해 비트 효율이 달성될 수 있는 비디오 비트스트림에서 인코딩될 수 있다.
- [0553] 이 방법들은 상기 제1 MV 예측 후보가 대안적인 시간 모션 벡터 예측(ATMVP) 후보로부터 도출되는 것을 포함할 수 있다.
- [0554] 이 방법들은 상기 제1 MV 후보가 아핀 후보로부터 도출되는 것을 포함할 수 있다.
- [0555] 이 방법들은 상기 모션 후보 리스트에 추가된 상기 제1 MV 예측 후보가 다수의 모션 정보 세트를 포함하고, 상기 다수의 모션 정보 세트 중 하나가 이전에 코딩된 비디오 블록에 기초한 정규 모션 후보를 나타내고, 상기 정규 모션 후보가 상기 제1 MV 예측 후보를 도출하기 위해 사용되는 것을 포함할 수 있다.
- [0556] 이 방법들은 상기 다수의 모션 정보 세트가 상기 제1 비디오 블록 내의 상기 제1 예측 부분의 위치에 기초하는 것을 포함할 수 있다.
- [0557] 이 방법들은 상기 제1 비디오 블록이 상기 제1 예측 부분을 포함하는 M개의 예측 부분을 포함하도록 파티셔닝되고, 상기 다수의 모션 정보 세트의 수가 M인 것을 포함할 수 있다.
- [0558] 이 방법들은 상기 제1 비디오 블록이 또한 비-직사각형 및 비-정사각형인 제2 예측 부분을 포함하도록 파티셔닝되고, 상기 모션 후보 리스트를 사용하여 상기 제1 비디오 블록을 추가 처리하는 단계가 상기 제1 MV 예측 후보를 사용하여 상기 제1 예측 부분 및 상기 제2 예측 부분을 처리하는 단계를 포함하는 것을 포함할 수 있다.
- [0559] 이 방법들은 상기 제1 예측 부분 및 상기 제2 예측 부분에 대한 분할 패턴들의 지시의 범위가 $M*N*N$ 이고, M은 상기 제1 비디오 블록이 파티셔닝되는 예측 부분들의 수이고, N은 상기 모션 후보 리스트에 추가될 허용된 병합 후보들의 수인 것을 포함할 수 있다.
- [0560] 이 방법들은, 상기 프로세서에 의해, 상기 모션 후보 리스트 내의 단일 MV 예측 후보가 상기 제1 예측 부분 및 상기 제2 예측 부분에 대해 사용될 것임을 플래그가 지시한다고 결정하는 단계를 포함할 수 있다.
- [0561] 이 방법들은 본 명세서에서 상기 모션 후보 리스트 내의 각각의 MV 예측 후보가 별개의 MV 예측 후보들에 기초하는 상기 제1 예측 부분 및 상기 제2 예측 부분에 기초한 양방향 예측 모션 후보인 것을 포함할 수 있다.
- [0562] 이 방법들은 제1 비디오 블록이 또한 비-직사각형 및 비-정사각형인 제2 예측 부분을 포함하도록 파티셔닝되고, 상기 제1 예측 부분에는 List 0 모션 정보가 할당되고, 상기 제2 예측 부분에는 List 1 모션 정보가 할당되는 것을 포함할 수 있다.
- [0563] 이 방법들은 상이한 MV 예측 후보들로부터 예측되는 상기 제1 예측 부분 및 상기 제2 예측 부분에 기초하여 가중 영역에 위치된 샘플들에 대해 가중치 그룹이 상이한 것을 포함할 수 있다.
- [0564] 이 방법들은 방법들이 고급 모션 벡터 예측(AMVP), 병합, 또는 삼각형 파티션 모드(TPM)에 더하여 추가적인 코딩 모드로서 적용되는 것을 포함할 수 있다.
- [0565] 이 방법들은 제1 비디오 블록에 대한 최종 예측 블록을 도출하기 위해 사용되는 MV 후보를 리파인하도록 상기 모션 후보 리스트를 수정하는 단계를 포함할 수 있다.
- [0566] 이 방법들은 상기 제1 비디오 블록에 대한 모션 정보를 도출하기 위해 상기 제1 비디오 블록의 제1 예측 파티션의 모션 정보 및 제2 예측 파티션의 모션 정보와 함께 디코더 측 모션 벡터 리파인먼트(DMVR)가 적용되는 것을 포함할 수 있다.
- [0567] 이 방법들은 TPM 후에 양방향 광학 흐름(BIO)이 적용되는 것을 포함할 수 있다.
- [0568] 이 방법들은 리파인된 모션 정보가 저장되는 것을 포함할 수 있다.
- [0569] 이 방법들은 상기 제1 예측 부분이 삼각형 형상이고, 상기 제1 예측 부분이 인트라-예측과 조합되는 것을 포함할 수 있다.
- [0570] 이 방법들은 제1 예측 부분이 TPM에 따라 파티셔닝되고, TPM에서 사용되는 하나 이상의 참조 화상이 현재 화상인 것을 포함할 수 있다.
- [0571] 이 방법들은 상기 제1 비디오 블록이 TPM에 따라 상기 제1 예측 부분을 포함하도록 파티셔닝되고, 상기 제1 예

즉 부분에는 상기 제1 비디오 블록의 제2 예측 부분과 상이한 모션 정보가 할당되는 것을 포함할 수 있다.

[0572] 이 방법들은 상기 모션 후보 리스트가 TPM 코딩된 비디오 블록들에 대한 병합 후보들을 포함하는 것을 포함할 수 있다.

[0573] 이 방법들은 상기 제1 비디오 블록이 TPM에 따라 코딩되는 것에 기초하여 아핀 모드가 인에이블되는 것을 포함할 수 있다.

[0574] 이 방법들은 상기 제1 예측 부분에 대해 단방향 예측이 사용되고, 상기 제1 예측 부분이 가중 영역에 위치되지 않는 것을 포함할 수 있다.

[0575] 이 방법들은 상기 제1 예측 부분 내의 상이한 서브-블록들의 참조 화상들이 동일한 것을 포함할 수 있다.

[0576] 이 방법들은 상기 모션 후보 리스트 내의 상기 MV 예측 후보들 각각이 ATMVP 후보로부터 도출되는 것을 포함할 수 있다.

[0577] 이 방법들은 병합 후보 리스트 구성 프로세스에서 사용되는 공간 또는 시간 블록들 중 하나 또는 둘 다에 대해 병합 리스트가 검사되는 것을 포함할 수 있다.

[0578] 이 방법들은 양방향 예측이 인에이블되고, 양방향 예측 정규 모션 후보가 가중치들과 연관되고, 상기 가중치들을 사용하지 않고 TPM 후보들이 도출되는 것을 포함할 수 있다.

[0579] 이 방법들은 양방향 예측이 인에이블되고, 양방향 예측 정규 모션 후보가 가중치들과 연관되고, 상기 가중치들을 사용하여 TPM 후보들이 도출되는 것을 포함할 수 있다.

[0580] 이 방법들은 상이 제1 비디오 블록의 가중 영역에 위치되지 않은 샘플들에 대해 양방향 예측이 인에이블되고, 상기 제1 비디오 블록은 TPM에 따라 코딩된 것을 포함할 수 있다.

[0581] 이 방법들은 2개의 참조 화상이 동일한 것을 포함할 수 있다.

[0582] 이 방법들은 상기 제1 비디오 블록의 각각의 파티션에 대해 하나의 참조 화상이 할당되는 것을 포함할 수 있다.

[0583] 이 방법들은 상기 제1 비디오 블록이 TPM에 따라 코딩되고, 다중-가설이 상기 제1 비디오 블록에 적용되는 것을 포함할 수 있다.

[0584] 이 방법들은 최종 예측 블록이 현재 설계를 사용하여 생성되고, 제2 예측 블록이 단방향에 따라 그리고 상기 최종 예측 블록에 대한 것과 상이한 다른 참조 화상 및 모션 벡터에 기초하여 생성되는 것을 포함할 수 있다.

[0585] 이 방법들은 최종 예측 블록이 현재 설계를 사용하여 생성되고, 제2 예측 블록이 양방향에 따라 그리고 상기 최종 예측 블록에 대한 것과 상이한 다른 참조 화상 및 모션 벡터에 기초하여 생성되는 것을 포함할 수 있다.

[0586] 이 방법들은 상기 제1 비디오 블록의 각각의 파티션에 대해, 양방향 예측 또는 다중-가설 예측이 적용되고, 양방향 예측 또는 다중-가설로 인한 가중 영역에 대한 예측 블록들의 가중 결과들이 최종 예측 값으로서 사용되는 것을 포함할 수 있다.

[0587] 이 방법들은 상기 제1 비디오 블록이 TPM에 따라 코딩되고, 오버랩 블록 모션 보상(OBMC)이 적용되지 않는 것을 포함할 수 있다.

[0588] 이 방법들은 가중 영역에 위치된 샘플들에 대해 OBMC가 적용되지 않고, 비-가중 영역에 위치된 샘플들에 대해 OBMC가 적용되는 것을 포함할 수 있다.

[0589] 이 방법들은 상기 제1 비디오 블록이 비-직사각형 및 비-정사각형인 제2 예측 부분을 또한 포함하도록 파티셔닝 되는 것을 포함할 수 있다.

[0590] 이 방법들은 상기 제1 비디오 블록이 제3 예측 부분 및 제4 예측 부분을 또한 포함하도록 파티셔닝되고, 상기 제3 예측 부분 및 상기 제4 예측 부분은 비-직사각형 및 비-정사각형인 것을 포함할 수 있다.

[0591] 이 방법들은 상기 제1, 제2, 제3, 또는 제4 예측 부분들 중 2개가 인터-코딩되고, 다른 2개가 인터-코딩되지 않는 것을 포함할 수 있다.

[0592] 이 방법들은 상기 제1 예측 부분과 상기 제2 예측 부분 사이의 에지 주위에 위치된 샘플들에 블렌딩이 적용되는 것을 포함할 수 있다.

[0593] 이 방법들은 상기 제1 분할 패턴, 상기 제1 병합 인덱스, 및 상기 제1 코딩된 값과 연관된 맵핑이 구성된 병합

후보 리스트에 기초하는 것을 포함할 수 있다.

[0594] 이 방법들은 상기 제1 분할 패턴, 상기 제1 병합 인덱스, 및 상기 제1 코딩된 값과 연관된 맵핑이 병합 후보들과 연관된 코딩된 모드에 기초하는 것을 포함할 수 있다.

[0595] 이 방법들은 병합 플래그 전에 코딩되는 TPM 사용을 지시하는 플래그에 기초하여 비-병합 모드에 대해 TPM이 인에이블되는 것을 포함할 수 있다.

[0596] 이 방법들은, 코딩되는 TPM 사용을 지시하는 플래그에 기초하여 비-병합 모드에 대해 TPM이 인에이블되고, 상기 제1 비디오 블록이 TPM으로 코딩되지 않은 것에 기초하여 병합 플래그가 시그널링되지 않는 것을 포함할 수 있다.

[0597] 이 방법들은 상기 제1 비디오 블록이 인터-코딩되는 것에 기초하여 병합 플래그가 시그널링되는 것을 포함할 수 있다.

[0598] 이 방법들은 상기 제1 분할 패턴이 상기 제1 비디오 블록의 이웃 블록들로부터 상속되는 것을 포함할 수 있다.

[0599] 이 방법들은 상기 제1 분할 패턴이 상기 제1 비디오 블록의 시간 블록들로부터 상속되는 것을 포함할 수 있다.

[0600] 이 방법들은 상기 제1 분할 패턴 및 상기 제1 병합 인덱스가 개별적으로 시그널링되는 것을 포함할 수 있다.

[0601] 이 방법들은 상기 제1 분할 패턴에 대한 TPM의 사용을 지시하는 플래그가 바이패스 코딩을 사용하는 것을 포함할 수 있다.

[0602] 이 방법들은 상기 플래그가 상기 제1 비디오 블록의 폭 또는 높이 중 하나 또는 둘 다에 기초하는 컨텍스트로 코딩되는 것을 포함할 수 있다.

[0603] 이 방법들은 상기 플래그가 코딩된 깊이에 기초하는 컨텍스트로 코딩되는 것을 포함할 수 있다.

[0604] 이 방법들은 상기 플래그가 상기 제1 비디오 블록의 블록 형상 또는 상기 제1 비디오 블록의 폭과 높이의 비율에 기초하는 컨텍스트로 코딩되는 것을 포함할 수 있다.

[0605] 이 방법들은 상기 플래그가 상기 제1 비디오 블록의 이웃 블록들의 코딩된 정보에 기초하는 컨텍스트로 코딩되는 것을 포함할 수 있다.

[0606] 이 방법들은 상기 제1 분할 패턴이 TPM에 따르고, 상기 제1 비디오 블록의 파티션들이 제1 영역 및 제2 영역을 포함하고, 상기 제1 영역 및 상기 제2 영역이 상이한 모드들로 코딩되는 것을 포함할 수 있다.

[0607] 이 방법들은 상기 제1 영역 또는 상기 제2 영역 중 하나가 인트라-코딩되고, 다른 하나가 인터-코딩되는 것을 포함할 수 있다.

[0608] 이 방법들은 상기 제1 영역 또는 상기 제2 영역 중 하나가 병합-코딩되고, 다른 하나가 AMVP 코딩되는 것을 포함할 수 있다.

[0609] 이 방법들은 상기 제1 비디오 블록의 특성들이 컬러 성분을 포함하고, 상기 기하학적 예측 모드의 상태가 디스에이블 상태이고, 상기 기하학적 예측 모드가 시그널링되는 디스에이블 상태에 있음을 지시하는 플래그를 사용하지 않고 기하학적 예측 모드가 디스에이블 상태에 있는 것을 포함할 수 있다.

[0610] 이 방법들은 상기 제1 비디오 블록의 특성들이 임계치 범위보다 작은 폭 또는 높이를 갖는 크로마 블록들을 포함하는 것을 포함할 수 있다.

[0611] 이 방법들은 상기 특성들이 상기 제1 비디오 블록의 크기가 NxM 또는 LxN인 것을 포함하는 것을 포함할 수 있고, 여기서 N은 4이다.

[0612] 이 방법들은 상기 특성들이 4:2:0인 컬러 포맷을 포함하는 것을 포함할 수 있다.

[0613] 이 방법들은 상기 특성들이 상기 제1 비디오 블록이 그것의 폭, 높이, 또는 폭과 높이 둘 다가 임계치 L보다 크거나 그와 동등한 것을 포함하는 것을 포함할 수 있다.

[0614] 이 방법들은 임계치 L이 64인 것을 포함할 수 있다.

[0615] 이 방법들은 상기 특성들이 상기 제1 비디오 블록이 그것의 폭, 높이, 또는 폭과 높이 둘 다가 임계치 L보다 작거나 그와 동등한 것을 포함하는 것을 포함할 수 있다.

- [0616] 이 방법들은 임계치 L이 4 또는 8인 것을 포함할 수 있다.
- [0617] 이 방법들은 임계치 L이 블록 크기, 화상 유형, 또는 시간 계층 인덱스 중 하나 이상을 포함하는 코딩된 정보에 기초하는 것을 포함할 수 있다.
- [0618] 이 방법들은 상기 특성들이 상기 제1 비디오 블록의 컬러 성분인 것을 포함할 수 있다.
- [0619] 이 방법들은 비디오 파라미터 세트(VPS), 시퀀스 파라미터 세트(SPS), 화상 파라미터 세트(PPS), 화상 헤더, 타일 그룹 헤더, 슬라이스 헤더, 코딩 트리 단위(CTU)들의 행들의 그룹, 영역, 또는 CTU들의 그룹에서 시그널링되는 것에 기초하여 TPM이 인에이블되거나 적용되는 것을 포함할 수 있다.
- [0620] 이 방법들은 상기 기하학적 예측 모드가 TPM인 것을 포함할 수 있다.
- [0621] 이 방법들은 비-직사각형 및 비-정사각형 예측 부분들이 삼각형인 것을 포함할 수 있다.
- [0622] 일부 실시예들에서 비디오 처리 방법은 코딩의 스킵/병합 모드가 제2 코딩 모드와 별개로 취급되는 것을 특정하는 코딩 규칙과 부합하도록 비디오 블록과 상기 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 제2 코딩 모드는 삼각형 파티션 모드 또는 MHIntra 모드 또는 서브-블록 병합 리스트 모드이다.
- [0623] 일부 실시예들에서, 상기 비트스트림 표현은 다음과 같이 구성된다: (1) 상기 비디오 블록에 대해 상기 제2 코딩 모드가 인에이블되는 경우에는, 상기 비트스트림 표현에서의 병합 플래그가 거짓 값을 갖도록 시그널링되거나, (2) 현재 블록이 상기 스kip/병합 모드를 사용하는 경우에는, 상기 제2 코딩 모드에 대한 병합 리스트의 대응하는 지시가 상기 비트스트림 표현에서 명시적으로 시그널링되지 않는다.
- [0624] 다른 비디오 코딩 방법은 변환에서 복수의 서브-블록 코딩 툴의 사용의 지시의 특정 순서를 특정하는 규칙과 부합하도록 비디오 블록과 상기 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 특정 순서는 (a.) MMVD(merge with motion vector difference)에 이어서 서브-블록 병합 리스트에 이어서 삼각형 파티션 모드 TPM에 이어서 MHIntra, 또는 (b.) MMVD에 이어서 서브-블록 병합 리스트에 이어서 MHIntra에 이어서 TPM 중 하나이다.
- [0625] 일부 실시예들에서, 상기 규칙은 다른 비디오 부분들에 대해 또는 상이한 비디오 조건들에 기초하여 상이한 특정 순서를 사용하는 것을 추가로 특정한다.
- [0626] 일부 실시예들에서, 상기 다른 비디오 부분들은 상기 비디오 블록의 것과는 다른 비디오 슬라이스들, 또는 다른 비디오 코딩 단위들 또는 다른 비디오 예측 단위들 또는 다른 비디오 코딩 단위 행들 또는 다른 비디오 화상들 또는 다른 비디오 시퀀스들을 포함한다.
- [0627] 일부 실시예들에서, 상기 상이한 비디오 조건들은 저지연 검사 플래그들의 상이한 값들, 또는 이전에 코딩된 정보의 상이한 값들 또는 다른 비디오 블록들의 치수들을 포함한다.
- [0628] 다른 비디오 처리 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 블록의 선택적 사용을 특정하고, 상기 규칙은: (1) 상이한 최대 코딩 단위(LCU) 행에 있는 이웃 블록을 제외하는 것 또는 (2) 고정된 해상도 값을 특정하는 것 또는(3) 비디오 블록이 아핀 모드를 사용하여 코딩되는 경우, 상이한 LCU 행에 있는 아핀-코딩된 이웃 블록을 이용불가능하게 하는 것을 포함한다.
- [0629] 다른 비디오 처리 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 블록의 선택적 사용을 특정하고, 상기 규칙은: (a) 이웃 블록이 상이한 최대 코딩 단위(LCU) 행에 있는 경우 상기 비디오 블록의 컨텍스트 모델링을 위해 이웃 비디오 블록의 정보를 사용하는 것을 디스에이블하는 것 또는 (b) 상기 변환 동안에 구문 요소를 코딩하기 위해 단 하나의 컨텍스트만을 사용하는 것 또는 (c) 바이패스 코딩을 적용하는 것이 단 하나의 구문 요소에 적용되는 것으로서, 이웃 블록의 선택적 사용을 포함한다.
- [0630] 일부 실시예들에서, 상기 단 하나의 구문 요소는 a. alf_ctb_flag, b. qt_split_cu_flag, c. mtt_split_cu_flag, d. mtt_split_cu_flag, e. cu_skip_flag, f. amvr_mode, g. merge_subblock_flag, h. merge_triangle_flag, 또는 i. inter_affine_flag를 포함한다.

- [0631] 일부 실시예들에서, 상기 변환은 상기 비트스트림 표현으로부터 상기 비디오 블록의 픽셀 값들을 생성하는 것 또는 상기 비디오 블록의 픽셀 값들로부터 상기 비트스트림 표현을 생성하는 것을 포함한다.
- [0632] 다른 비디오 처리 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 비디오 블록들의 선택적 사용을 특정하고, 상기 규칙은: 이웃 비디오 블록들에 관한 정보가 현재 비디오 블록을 디코딩하기 위해 사용되고 있는 이웃 비디오 블록들에 기초하여 컨텍스트 모델링을 위해 사용되는 것을 포함한다.
- [0633] 이 방법들은 이웃 비디오 블록들이 현재 비디오 블록을 포함하는 부분의 외부에 있다고 결정하는 단계를 포함할 수 있고, 상기 부분은: 코딩 트리 단위(CTU), CTU 행, 영역, 비디오 데이터 처리 단위, 슬라이스, 타일, 타일 그룹, 또는 화상 경계이고, 상기 규칙은: (1) 이웃 비디오 블록들에 관한 정보가 파싱 기법으로부터 제외되는 것, (2) 이웃 비디오 블록들에 관한 정보가 파싱 기법에서 사용되는 것, (3) 이웃 비디오 블록들에 관한 정보가 파싱 기법에 대한 디폴트 값들로서 설정되는 것, 또는 (4) 동일한 부분에 있는 다른 이웃 비디오 블록들에 관한 정보가 파싱 기법에서 대체로서 사용되는 것을 포함한다.
- [0634] 이 방법들은 이웃 비디오 블록에 관한 정보가 이웃 비디오 블록들의 코딩된 정보에 기초하는 구문 요소들에 대한 사전 정의된 이웃 위치들로부터 유래한 것임을 포함할 수 있다.
- [0635] 다른 비디오 처리 방법은 코딩 규칙과 부합하도록 비디오 블록과 현재 비디오 블록의 비트스트림 표현 간의 변환을 수행하는 단계를 포함하고, 상기 규칙은 상기 비디오 블록의 AMVR(adaptive motion vector resolution) 코딩 모드에서 모션 벡터 예측을 특정하기 위한 이웃 비디오 블록들의 선택적 사용을 특정하고, 상기 규칙은: 이웃 비디오 블록들에 관한 정보가 현재 비디오 블록을 디코딩하기 위해 사용되고 있는 이웃 비디오 블록들에 기초하여 컨텍스트 모델링을 위해 사용되는 것을 포함하고, 이웃 비디오 블록들은 현재 CTU 행과 상이한 CTU 행 내의, 또는 현재 CTU와 상이한 CTU 내의 비디오 블록들을 제외하고 사용된다.
- [0636] 이 방법들은 제외되는 이웃 비디오 블록들로부터의 정보가 컨텍스트 모델링에 이용불가능한 것을 포함할 수 있다.
- [0637] 이 방법들은 제외되는 이웃 비디오 블록들의 정보 대신에 디폴트 값이 사용되는 것을 포함할 수 있다.
- [0638] 이 방법들은 현재 CTU 행 또는 현재 CTU 내의 대안적인 이웃 비디오 블록으로부터의 정보가 제외되는 비디오 블록들 대신에 사용되는 것을 포함할 수 있다.
- [0639] 이 방법들은 구문 요소들이: alf_ctb_flag, qt_split_cu_flag, mtt_split_cu_flag, cu_skip_flag, amvr_mode, inter_affine_flag, merge_triangle_flag and a merge_subblock_flag, cu_skip_flag, pred_mode_flag, pcm_flag, intra_luma_ref_idx, intra_luma_mpm_flag, intra_luma_mpm_idx, intra_luma_mpm_remainder, intra_chroma_pred_mode, merge_flag, inter_pred_idc, inter_affine_flag, cu_affine_type_flag, ref_idx_10,.mvp_10_flag, ref_idx_11.mvp_11_flag, amvr_flag, amvr_4pel_flag, gbi_idx, cu_cbf, mmvd_flag, 또는 mmvd_merge_flag 중 하나 이상을 포함하는 것을 포함할 수 있다.
- [0640] 이 방법들은 상기 컨텍스트 모델링이 pred_mode_flag의 사용을 포함하는 것을 포함할 수 있다.
- [0641] 이 방법들은 상기 pred_mode_flag가 다음 중 하나 이상으로부터 도출되는 것을 포함할 수 있다: ctx = A1이 인트라 ? 1 : 0 및 ctx += B1이 인트라 ? 1 : 0, ctx = (A1이 인트라) || (B1이 인트라) ? 1 : 0, ctx = A1이 인트라 또는 인터-인트라 ? 1 : 0 및 ctx += B1이 인트라 또는 인터-인트라 ? 1 : 0, 또는 ctx = (A1이 인트라 또는 인터-인트라) || (B1이 인트라 또는 인터-인트라) ? 1 : 0.
- [0642] 이 방법들은 상기 pred_mode_flag가 현재 비디오 블록의 모서리들에 인접하여 위치된 이웃 블록들로부터 도출되는 것을 포함할 수 있다.
- [0643] 이 방법들은 상기 pred-mode_flag가 다른 종류의 구문 요소들로 대체되는 것을 포함할 수 있다.
- [0644] 이 방법들은 이용불가능한 이웃 블록은 상기 컨텍스트를 도출할 때 인터-코딩된 것으로서 처리되는 것을 포함할 수 있다.
- [0645] 이 방법들은 이용불가능한 이웃 블록은 상기 컨텍스트를 도출할 때 인트라-코딩된 것으로서 처리되는 것을 포함할 수 있다.

- [0646] 이 방법들은 이용불가능한 이웃 블록은 상기 컨텍스트를 도출할 때 인터-코딩된 것으로서 처리되는 것을 포함할 수 있다.
- [0647] 이 방법들은 이웃 비디오 블록이 현재 비디오 블록과 동일한 코딩 트리 단위(CTU) 행 내에 있지 않고, 상기 이웃 비디오 블록은 컨텍스트를 도출할 때 인터-코딩된 것으로 취급되는 것을 포함할 수 있다.
- [0648] 이 방법들은 이웃 비디오 블록이 현재 비디오 블록과 동일한 코딩 트리 단위(CTU) 행 내에 있지 않고, 상기 이웃 비디오 블록은 인트라-코딩된 것으로 취급되는 것을 포함할 수 있다.
- [0649] 이 방법들은 이웃 비디오 블록이 현재 비디오 블록과 동일한 코딩 트리 단위(CTU) 행 내에 있지 않고, 상기 이웃 비디오 블록은 인트라-인트라-코딩된 것으로 취급되는 것을 포함할 수 있다.
- [0650] 기술적 솔루션들의 다음의 리스트는 바람직하게는 비디오 인코더 또는 디코더 장치의 일부 실시예들에 포함될 수 있다.
- [0651] 다음의 솔루션들의 추가적인 특징들 및 변형들이 이전 섹션에서 제공된다(예를 들어, 항목 1 내지 4, 17 및 21).
- [0652] 예시적인 솔루션 세트 A
- [0653] 1. 비디오 처리 방법(예를 들어, 도 44에 묘사된 방법(4400))으로서, 비디오의 비디오 영역의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환이 상기 비디오 블록이 적어도 제1 예측 파티션을 포함하는 다수의 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용한다는 결정을 하는 단계(4402); 상기 비디오 블록과 연관된 하나 이상의 서브-블록 병합 후보를 사용하여 N개의 모션 정보 후보를 상기 비디오 블록에 대한 모션 후보리스트에 추가하는 단계(4404) - N은 양의 정수임 -; 상기 모션 후보 리스트로부터, 상기 다수의 예측 파티션에 대한 모션 후보들을 도출하는 단계(4406); 및 상기 다수의 예측 파티션에 대한 모션 후보에 기초하여 상기 변환을 수행하는 단계(4408)를 포함하는, 방법.
- [0654] 2. 솔루션 1에 있어서, 상기 N개의 MV 후보를 도출하기 위해 서브-블록 시간 모션 벡터 예측자 후보가 사용되는, 방법.
- [0655] 3. 솔루션 1에 있어서, 상기 N개의 MV 후보를 도출하기 위해 아핀 모션 후보가 사용되는, 방법.
- [0656] 4. 솔루션 1에 있어서, 하나 이상의 서브-블록의 하나 이상의 서브-블록 모션 후보로부터의 M개의 대표 MV 후보가 상기 M개의 MV 후보를 도출하기 위해 사용되고, M은 양의 정수인, 방법.
- [0657] 5. 솔루션 4에 있어서, 상기 M개의 대표 MV 후보는 상기 하나 이상의 서브-블록의 위치들에 기초하여 선택되는, 방법.
- [0658] 6. 솔루션 4 내지 솔루션 5 중 어느 하나에 있어서, M은 상기 변환을 위해 상기 비디오 블록이 파티셔닝되는 파티션들의 수와 동등한, 방법.
- [0659] 7. 솔루션 4 내지 솔루션 5 중 어느 하나에 있어서, M = 1이고, 대표 MV 후보가 상기 제1 예측 부분의 중심 서브-블록과 연관된 모션 정보로부터 선택되는, 방법.
- [0660] 8. 비디오를 처리하기 위한 방법(예를 들어, 도 45에 묘사된 방법(4500))으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 비디오 블록이 적어도 제1 예측 파티션을 포함하는 다수의 예측 파티션으로 파티셔닝된다고 결정하는 단계(4502); 상기 제1 예측 파티션과 연관된 제1 후보 리스트에 제1 모션 정보 후보를 추가하고 제2 예측 파티션과 연관된 제2 후보 리스트에 연관된 제2 모션 정보 후보를 추가하는 단계(4504) - 상기 제1 모션 정보 후보 및 상기 제2 모션 정보 후보는 단일 모션 후보로부터 상속됨 -; 상기 제1 모션 정보 후보 및/또는 상기 제2 모션 정보 후보를 사용하여 상기 변환을 수행하는 단계(4506)를 포함하는, 방법.
- [0661] 9. 솔루션 8에 있어서, 상기 제1 예측 부분 및 상기 제2 예측 부분에 대한 분할 패턴들 및 MV 후보 인덱션들의 지시의 범위가 M*N*N이고, M은 상기 제1 비디오 블록이 파티셔닝되는 예측 부분들의 수이고, N은 상기 모션 후보 리스트에 추가될 허용된 병합 후보들의 수인, 방법.
- [0662] 10. 솔루션 8 또는 솔루션 9에 있어서, 상기 모션 후보 리스트 내의 단일 MV 예측 후보가 상기 제1 예측 부분 및 상기 제2 예측 부분 둘 다에 대해 사용될 것임을 비트스트림 표현 내의 플래그가 지시하는, 방법.

- [0663] 11. 솔루션 8에 있어서, 상기 모션 후보 리스트 내의 예측 후보는 양방향 예측 모션 후보인, 방법.
- [0664] 12. 솔루션 8에 있어서, 상기 제1 MV 후보에는 List 0 모션 정보가 할당되고 상기 제2 MV 후보에는 List 1 모션 정보가 할당되는, 방법.
- [0665] 13. 솔루션 8 내지 솔루션 12 중 어느 하나에 있어서, 상기 분할 패턴에 따라 상기 비디오 블록의 가중 영역에 위치된 샘플들에 대해 특정 가중치 그룹이 사용되고, 상기 특정 가중치 그룹은 다른 비디오 블록에 대해 사용되는 다른 가중치 그룹과 상이하고, 상기 다른 비디오 블록의 예측 부분들에 대한 후보 모션 벡터들을 결정하기 위해 상이한 모션 후보들이 사용되는, 방법.
- [0666] 14. 솔루션 13에 있어서, 상기 특정 가중치 그룹은 상기 가중 영역에 위치된 샘플들에 대한 평균에 대응하는, 방법.
- [0667] 15. 솔루션 8 중 어느 하나에 있어서, 상기 변환은 변환 동안 사용되는 AMVP(advanced motion vector prediction), 병합, 또는 TPM(triangular partition mode)에 더하여 추가적인 코딩 모드로서 상기 방법을 사용하는, 방법.
- [0668] 16. 솔루션 1 내지 솔루션 15 중 어느 하나에 있어서, 상기 변환을 수행하는 단계는 상기 비디오 블록에 대한 예측 블록에 대한 리파인된 최종 모션 벡터를 도출하는 단계를 포함하는, 방법.
- [0669] 17. 솔루션 16에 있어서, 상기 리파인된 최종 모션 벡터를 도출하는 단계는 상기 비디오 블록에 대한 모션 정보를 도출하기 위해 상기 제1 비디오 블록의 제1 예측 파티션의 모션 정보 및 제2 예측 파티션의 모션 정보에 디코더 측 모션 벡터 리파인먼트(DMVR) 툴을 사용하는 단계를 포함하는, 방법.
- [0670] 18. 솔루션 16에 있어서, 상기 변환을 수행하는 단계는 삼각형 파티션 모드 예측 후에 결과에 양방향 광학 흐름 (BIO) 툴을 결과에 적용하는 단계를 포함하는, 방법.
- [0671] 19. 솔루션 17 또는 솔루션 18에 있어서, 상기 제1 예측 부분은 제1 참조 리스트로부터 단방향 예측되고 상기 제2 예측 부분은 제2 참조 리스트를 사용하여 양방향 또는 단방향 예측되고, 그 후 상기 제2 참조 리스트 내의 제2 예측 부분의 모션 정보를 사용하여 DMVR 툴 또는 BIO 툴이 적용되는, 방법.
- [0672] 20. 솔루션 16에 있어서, 상기 제1 예측 부분이 참조 리스트 LX로부터 단방향 예측되는 것으로 인해 상기 변환 동안 참조 리스트 LX의 모션 정보를 저장하는 단계를 추가로 포함하고, X는 0 또는 1인, 방법.
- [0673] 21. 솔루션 16에 있어서, 상기 변환은 기하학적 파티션 모드에 기초한 예측 결과와 인트라 예측 모드에 기초한 예측 결과의 가중 합인 최종 예측을 사용하는, 방법.
- [0674] 22. 솔루션 16에 있어서, 기하학적 파티션 모드에서 사용되는 하나 이상의 참조 화상은 현재 화상인, 방법.
- [0675] 23. 비디오 처리 방법(예를 들어, 도 46에 묘사된 방법(4600)으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환이 상기 비디오 블록이 예측 파티션을 포함하도록 파티셔닝되어 상기 예측 파티션의 서브-블록에 그의 서브-블록 모션 정보가 할당되게 되는 서브-블록 인에이블 기하학적 파티션 모드(sub-block enabled geometric partitioning mode)를 사용한다는 결정을 하는 단계(4602); 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(4604)를 포함하는, 방법.
- [0676] 24. 솔루션 23에 있어서, 상기 서브-블록 모션 정보는 상기 비디오 블록의 변환을 위해 사용되는 후보 리스트 내의 후보로서 사용되는, 방법.
- [0677] 25. 솔루션 23에 있어서, 상기 제1 비디오 블록이 상기 기하학적 파티션 모드에 따라 코딩되는 것에 기초하여 아핀 모드가 인에이블되는, 방법.
- [0678] 26. 솔루션 23 내지 솔루션 25 중 어느 하나에 있어서, 상기 변환은, 상기 비디오 블록의 가중 영역에 있지 않은 서브-블록에 대해, 단방향 예측만을 사용하도록 제한하는, 방법.
- [0679] 27. 솔루션 23에 있어서, 상기 변환은 상기 비디오 블록의 임의의 주어진 예측 부분 내의 모든 서브-블록들에 대해 동일한 참조 화상을 사용하는 것을 특정하는 규칙에 기초하는, 방법.
- [0680] 28. 솔루션 23 내지 솔루션 27 중 어느 하나에 있어서, 상기 변환은 상기 서브-블록과 연관된 서브-블록 시간 모션 벡터 예측자(SbTMVP) 후보로부터 도출되는 모션 후보 리스트 내의 하나 이상의 모션 벡터 예측 후보를 도출하는 단계를 포함하는, 방법.

- [0681] 29. 솔루션 28에 있어서, 상기 변환은 상기 예측 파티션 내의 각각의 서브-블록이 SbTMVP 후보의 List 0 모션 정보의 모션 정보를 상속하는 규칙을 추가로 사용하는, 방법.
- [0682] 30. 솔루션 23에 있어서, 상기 비디오의 변환 동안 사용되는 서브-블록 병합 후보 리스트 프로세스에 대한 것과 동일한 순서로 상기 예측 파티션에 대한 병합 리스트가 검사되는 규칙을 추가로 사용하는, 방법.
- [0683] 31. 솔루션 1 내지 솔루션 30 중 어느 하나에 있어서, 상기 기하학적 파티션 모드는 상기 비디오 블록의 예측 파티션들을 표현하기 위해 상이한 코딩 모드들을 사용하는, 방법.
- [0684] 32. 솔루션 31에 있어서, 상기 제1 파티션 모드는 인트라 코딩을 사용하고 상기 제2 파티션 모드는 인터 코딩을 사용하는, 방법.
- [0685] 33. 솔루션 31에 있어서, 제1 예측 파티션은 병합 코딩을 사용하고 상기 제2 예측 파티션은 고급 모션 벡터 예측자 툴을 사용하는, 방법.
- [0686] 34. 솔루션 1 내지 솔루션 33 중 어느 하나에 있어서, 상기 비트스트림 표현에서의 지시는 상기 비디오 블록에 대한 상기 기하학적 파티션 모드의 사용을 시그널링하는, 방법.
- [0687] 35. 솔루션 34에 있어서, 상기 지시는 시퀀스 파라미터 세트 레벨 또는 화상 파라미터 세트 레벨 또는 비디오 파라미터 세트 레벨 또는 화상 헤더 또는 슬라이스 헤더 또는 타일 그룹 헤더 또는 코딩 트리 단위 레벨에 있는, 방법.
- [0688] 36. 솔루션 1 내지 솔루션 35 중 어느 하나에 있어서, 상기 비디오 블록이 크기 조건을 만족시키는 것으로 인해 상기 기하학적 파티션 모드가 인에이블되는, 방법.
- [0689] 37. 솔루션 36에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 큰 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.
- [0690] 38. 솔루션 36에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 작은 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.
- [0691] 39. 솔루션 37 내지 솔루션 38에 있어서, 상기 임계치는 코딩된 표현에서 시퀀스 파라미터 세트 또는 화상 파라미터 세트 또는 화상 또는 슬라이스 또는 타일 그룹 또는 타일 레벨에서 시그널링되는, 방법.
- [0692] 40. 솔루션 37 내지 솔루션 38 중 어느 하나에 있어서, 상기 크기는 상기 비디오 블록과 연관된 화상 유형 또는 시간 계층 인덱스의 함수인, 방법.
- [0693] 41. 솔루션 1 내지 솔루션 33 중 어느 하나에 있어서, 상기 비디오 블록이 사전 정의된 폭 대 높이 비율을 갖는 것으로 인해 상기 변환을 위해 상기 기하학적 파티션 모드가 사용되는, 방법.
- [0694] 다음의 솔루션들의 추가적인 특징들 및 변형들이 이전 섹션에서 제공된다(예를 들어, 항목 5, 7, 8, 9 및 21).
- [0695] 예시적인 솔루션 세트 B**
- [0696] 1. 비디오 처리 방법(예를 들어, 도 47a에 묘사된 방법(4700))으로서, 비디오 블록이 다수의 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 예측 파티션에 대한 모션 정보 도출을 위해 사용되는 모션 후보들로부터의 GBi(generalized bi prediction)가중치들의 상속이 상속 규칙에 따라 상기 비디오 블록의 샘플 위치들에서 인에이블되는지를 결정하는 단계(4702); 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(4704)를 포함하는, 방법.
- [0697] 2. 솔루션 1에 있어서, 상기 상속 규칙은 상기 비디오 블록의 모든 위치들에 대해 상기 상속이 디스에이블되는 것을 특정하는, 방법.
- [0698] 3. 솔루션 2에 있어서, 상기 모션 후보들과 연관된 GBi 가중치들에 관계없이 상기 변환이 수행되는 것을 추가로 포함하는, 방법.
- [0699] 4. 솔루션 2에 있어서, 상기 비디오 블록에 대한 GBi 가중치 인덱스는 0으로 설정되는, 방법.
- [0700] 5. 솔루션 1에 있어서, 상기 상속 규칙은 상기 비디오 블록의 모든 위치들에 대해 상기 상속이 인에이블되는 것을 특정하는, 방법.
- [0701] 6. 솔루션 5에 있어서, 상기 예측 파티션에 대해, 대응하는 GBi 가중치가 상기 비디오 블록의 모션 정보를 도출

하기 위해 사용되는 모션 후보와 연관된 것과 동등하게 설정되는 것을 추가로 포함하는, 방법.

- [0702] 7. 솔루션 1에 있어서, 상기 상속은 상기 비디오 블록의 각종 영역 외부의 샘플 위치들에 대해 디스에이블되고 상기 각종 영역 내의 샘플 위치들에서 인에이블되고, 상기 각종 영역은 2개의 예측 파티션의 경계와 교차하는, 방법.
- [0703] 8. 솔루션 7에 있어서, 상기 각종 영역을 제외한 상기 예측 파티션 내의 샘플 위치들에 대해, 대응하는 GBi 각종치가 0과 동등하게 설정되는 것을 추가로 포함하는, 방법.
- [0704] 9. 솔루션 4에 있어서, 상기 예측 파티션 내의 각종 영역에 대해, 대응하는 GBi 각종치가 상기 비디오 블록의 모션 정보를 도출하기 위해 사용되는 모션 후보와 연관된 것과 동등하게 설정되는 것을 추가로 포함하는, 방법.
- [0705] 10. 비디오 처리 방법(예를 들어, 도 47b에 묘사된 방법(4750))으로서, 비디오 블록이 다수의 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 적어도 비디오 블록의 샘플 위치들에서 규칙에 따라 양방향 예측의 사용이 인에이블되는지를 결정하는 단계(4752); 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(4754)를 포함하는, 방법.
- [0706] 11. 솔루션 10에 있어서, 상기 규칙은 상기 비디오 블록의 비-각종 영역 내의 샘플 위치들에 대해 양방향 예측이 인에이블되는 것을 특정하는, 방법.
- [0707] 12. 솔루션 11에 있어서, 상기 규칙은 서로 동일한 2개의 참조 화상을 사용하는 것에 의해서만 상기 비디오 블록의 비-각종 영역 내의 샘플 위치들에 대해 양방향 예측이 인에이블되는 것을 추가로 특정하는, 방법.
- [0708] 13. 솔루션 11 내지 솔루션 12 중 어느 하나에 있어서, 상기 양방향 예측은 제1 예측 파티션에 대한 제1 참조 프레임, 제2 예측 파티션에 대한 제2 참조 프레임 및 제1 예측 파티션 및 제2 예측 파티션 둘 다에 대한 제3 참조 프레임을 사용하고, 상기 비디오 블록은 상기 제1 예측 파티션 및 상기 제2 예측 파티션으로 구성되는, 방법.
- [0709] 14. 비디오 처리 방법(예를 들어, 도 48에 묘사된 방법(4800))으로서, 비디오 블록이 적어도 제1 예측 파티션 및 제2 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 사용하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 제1 예측 파티션의 샘플 위치들에 대해 도출된 제1 모션 정보 세트, 상기 제2 예측 파티션의 샘플 위치들에 대해 도출된 제2 모션 정보 세트와 상이한 모션 정보로부터의 제1 예측 블록 및 제2 예측 블록으로부터 최종 예측 블록을 결정하는 단계(4802); 및 상기 최종 예측 블록에 기초하여 상기 변환을 수행하는 단계(4804)를 포함하고; 상기 비디오 블록은 다중-가설(multi-hypothesis) 기법을 사용하여 코딩되는, 방법.
- [0710] 15. 솔루션 14에 있어서, 상기 제2 예측 블록은 상기 제1 모션 정보 세트 및 상기 제2 모션 정보 세트와 상이한 모션 벡터 또는 참조 화상을 사용하여 단방향 예측적으로 생성되는, 방법.
- [0711] 16. 솔루션 14에 있어서, 상기 제2 예측 블록은 2개의 추가적인 참조 화상을 사용하여 양방향 예측적으로 생성되는, 방법.
- [0712] 17. 솔루션 14 중 어느 하나에 있어서, 상기 제2 예측 블록은 상기 제1 예측 파티션에 대한 제3 모션 정보 세트 및 상기 제2 예측 파티션에 대한 제4 모션 정보 세트를 사용하여 생성되는, 방법.
- [0713] 18. 솔루션 14 내지 솔루션 17 중 어느 하나에 있어서, 각종 영역을 제외한 상기 제1 및 제2 예측 파티션 내의 상기 최종 예측 블록의 샘플들이 상기 제1 예측 블록 또는 상기 제2 예측 블록으로부터 생성되고; 상기 각종 영역 내의 샘플들에 대해, 상기 제1 예측 블록 및 상기 제2 예측 블록으로부터의 각종 예측이 사용되고; 상기 각종 영역은 2개의 예측 파티션의 경계와 교차하는, 방법.
- [0714] 19. 위의 솔루션들 중 어느 하나에 있어서, 상기 변환은 상기 비디오 블록의 샘플 위치들에서 오버랩 블록 모션 보상(OBMC) 코딩의 사용이 인에이블되는지를 특정하는 OBMC 규칙에 추가로 기초하는, 방법.
- [0715] 20. 솔루션 19에 있어서, 상기 OBMC 규칙은 상기 비디오 블록이 상기 기하학적 파티션 모드를 사용하는 것으로 인해 상기 변환에 대해 상기 OBMC 코딩이 디스에이블되는 것을 특정하는, 방법.
- [0716] 21. 솔루션 19에 있어서, 상기 OBMC 규칙은 상기 비디오 블록의 각종 영역의 샘플 위치들에서 OBMC 코딩을 디스에이블하고 나머지 샘플 위치들에서 OBMC를 인에이블하는 것을 특정하는, 방법.
- [0717] 22. 솔루션 1 내지 솔루션 21 중 어느 하나에 있어서, 상기 비디오 블록은 2개 이상의 파티션으로

파티셔닝되는, 방법.

[0718] 23. 솔루션 22에 있어서, 상기 비디오 블록은 4개의 삼각형 예측 파티션으로 파티셔닝되는, 방법.

[0719] 24. 솔루션 1 내지 솔루션 23 중 어느 하나에 있어서, 상기 비디오 블록은 비-직사각형 및 비-정사각형 파티션들인 적어도 2개의 파티션으로 파티셔닝되어 상기 적어도 2개의 파티션 중 많아야 하나가 인터 코딩되게 되는, 방법.

[0720] 25. 솔루션 22 내지 솔루션 24 중 어느 하나에 있어서, 상기 변환은 상기 4개의 파티션 중 이웃 파티션들의 에지들에서의 샘플 위치들에서 가중 예측 평균을 사용하는 것을 포함하는, 방법.

[0721] 26. 솔루션 1 내지 솔루션 25 중 어느 하나에 있어서, 상기 비트스트림 표현에서의 지시는 상기 비디오 블록에 대한 상기 기하학적 파티션 모드의 사용을 시그널링하는, 방법.

[0722] 27. 솔루션 26에 있어서, 상기 지시는 시퀀스 파라미터 세트 레벨 또는 화상 파라미터 세트 레벨 또는 비디오 파라미터 세트 레벨 또는 화상 헤더 또는 슬라이스 헤더 또는 타일 그룹 헤더 또는 코딩 트리 단위 레벨에 있는, 방법.

[0723] 28. 솔루션 1 내지 솔루션 25 중 어느 하나에 있어서, 상기 비디오 블록이 크기 조건을 만족시키는 것으로 인해 상기 기하학적 파티션 모드가 인에이블되는, 방법.

[0724] 29. 솔루션 26에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 큰 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.

[0725] 30. 솔루션 26에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 작은 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.

[0726] 31. 솔루션 29 내지 솔루션 30에 있어서, 상기 임계치는 코딩된 표현에서 시퀀스 파라미터 세트 또는 화상 파라미터 세트 또는 화상 또는 슬라이스 또는 타일 그룹 또는 타일 레벨에서 시그널링되는, 방법.

[0727] 32. 솔루션 29 내지 솔루션 30 중 어느 하나에 있어서, 상기 크기는 상기 비디오 블록과 연관된 화상 유형 또는 시간 계층 인덱스의 함수인, 방법.

[0728] 33. 솔루션 1 내지 솔루션 32 중 어느 하나에 있어서, 상기 비디오 블록이 사전 정의된 폭 대 높이 비율을 갖는 것으로 인해 상기 변환을 위해 상기 기하학적 파티션 모드가 사용되는, 방법.

[0729] 다음의 솔루션들의 추가적인 특징들 및 변형들이 이전 섹션에서 제공된다(예를 들어, 항목 10, 11, 12, 13 및 21).

예시적인 솔루션 세트 C

[0731] 1. 비디오 처리 방법(예를 들어, 도 49에 묘사된 방법(4900))으로서, 비디오의 비디오 영역의 비디오 블록과 상기 비디오의 비트스트림 표현 간의 변환을 위해, (1) 적어도 하나의 예측 파티션이 비-직사각형 및 비-정사각형 파티션이 되도록 상기 비디오 블록을 예측 파티션들로 분할하기 위해 사용되는 분할 패턴과, (2) 상기 변환을 위해 사용되는 파티션들의 후보들을 병합하기 위한 인덱스들 간의 관계를 결정하는 단계(4902) - 상기 비트스트림 표현의 포맷은 상기 비디오 영역 레벨에서 상기 관계를 변경하는 것을 허용함 -; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(4904)를 포함하는, 방법.

[0732] 2. 솔루션 1에 있어서, 상기 비디오 영역은 비디오 화상 또는 비디오 타일 또는 비디오 슬라이스 또는 비디오 코딩 트리 단위 또는 코딩 단위의 비디오 시퀀스에 대응하는, 방법.

[0733] 3. 솔루션 1 또는 솔루션 2에 있어서, 상기 관계는 상기 비디오 블록의 변환 동안 사용되는 병합 후보 리스트의 특성에 의존하는, 방법.

[0734] 4. 솔루션 1 내지 솔루션 3 중 어느 하나에 있어서, 상기 관계는 상기 병합 후보들과 연관된 코딩된 모드에 의존하는, 방법.

[0735] 5. 비디오 처리 방법(예를 들어, 도 50에 묘사된 방법(5000))으로서, 코딩된 표현의 구문 요소들을 정의하는 구문 규칙에 따라 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(5002)를 포함하고; 상기 구문 규칙은 상기 비디오 블록에 대한 기하학적 파티션 모드의 사용을 나타내는 제1 플래그 및 상기 비디오 블록에 대한 병합 모드의 사용을 나타내는 제2 플래그의 시그널링 순서를 특정하고; 상기 기하학적 파티

션 모드는 상기 비디오 블록을 다수의 예측 파티션으로 분할하는 것을 포함하고, 적어도 하나의 예측 파티션은 비-직사각형, 비-정사각형 형상을 가지고, 상기 병합 모드는 상기 비디오 블록의 이웃 블록들로부터 모션 벡터 정보를 상속하는 것을 허용하는, 방법.

- [0736] 6. 솔루션 5에 있어서, 상기 제2 플래그는 상기 제1 플래그 후에 시그널링되는, 방법.
- [0737] 7. 솔루션 5에 있어서, 상기 구문 규칙은 상기 제1 플래그의 시그널링으로 인해 상기 제2 플래그의 시그널링을 생략하는 것을 특정하는, 방법.
- [0738] 8. 솔루션 5에 있어서, 상기 제2 플래그는 상기 비디오 블록의 코딩 모드에 기초하여 조건부로 시그널링되는, 방법.
- [0739] 9. 솔루션 8에 있어서, 상기 제2 플래그는 상기 비디오 블록이 인터 코딩되는 것으로 인해 시그널링되는, 방법.
- [0740] 10. 솔루션 5에 있어서, 상기 구문 규칙은 상기 비디오 블록이 스kip 모드 코딩되는 것으로 인해 상기 제1 플래그가 생략되는 것을 특정하는, 방법.
- [0741] 11. 비디오 처리 방법(예를 들어, 도 51에 묘사된 방법(5100))으로서, 코딩된 표현의 구문 요소들을 정의하는 구문 규칙에 따라 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(5102)를 포함하고; 상기 구문 규칙은 상기 비디오 블록에 대한 기하학적 파티션 모드의 사용을 나타내는 플래그의 시그널링이 다른 코딩 틀이 상기 비디오 블록의 변환을 위해 사용되는지에 기초하여 선택적으로 포함된다는 것을 특정하고; 상기 기하학적 파티션 모드는 상기 비디오 블록을 다수의 예측 파티션으로 분할하는 것을 포함하고, 적어도 하나의 예측 파티션은 비-직사각형, 비-정사각형 형상인, 방법.
- [0742] 12. 솔루션 11에 있어서, 상기 구문 규칙은 상기 비디오 블록의 변환을 위해 고급 모션 벡터 예측 틀이 사용되면 상기 플래그를 포함하는 것을 특정하는, 방법.
- [0743] 13. 솔루션 11에 있어서, 상기 구문 규칙은 상기 비디오 블록이 병합 모드를 사용하여 코딩되면 상기 플래그를 시그널링하는 것을 디스에이블하는 것을 특정하는, 방법.
- [0744] 14. 비디오 처리 방법(예를 들어, 도 52에 묘사된 방법(5200))으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해 기하학적 파티션 모드가 인에이블되어 있다고 결정하는 단계(5202) - 상기 비디오 블록은 이웃 블록의 기하학적 파티션 모드에 기초한 분할 패턴에 기초하여 다수의 예측 파티션으로 분할됨 -, 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(5204)를 포함하는, 방법.
- [0745] 15. 비디오 처리 방법(예를 들어, 도 53에 묘사된 방법(5300))으로서, 비디오 블록이 이웃 블록의 분할 패턴에 기초한 분할 패턴에 기초하여 다수의 예측 파티션으로 분할되는 기하학적 파티션 모드에 기초하여 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위한 분할 패턴을 결정하는 단계(5302); 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(5304)를 포함하는, 방법.
- [0746] 16. 솔루션 14 또는 솔루션 15에 있어서, 상기 이웃 블록은 공간적으로 이웃 블록인, 방법.
- [0747] 17. 솔루션 14 또는 솔루션 15에 있어서, 상기 이웃 블록은 시간적으로 이웃 블록인, 방법.
- [0748] 18. 솔루션 2 내지 솔루션 4 중 어느 하나에 있어서, 상기 비디오 블록에 대한 모션 정보와 함께 상기 분할 패턴을 저장하는 단계를 추가로 포함하는, 방법.
- [0749] 19. 솔루션 2 내지 솔루션 5 중 어느 하나에 있어서, 상기 변환은 정규 모션 후보로부터 상기 다수의 파티션에 대한 모션 후보들을 도출하여 상기 정규 모션 후보와 연관된 분할 패턴이 상기 도출된 모션 후보들에 의해 상속 되게 되는 단계를 포함하는, 방법.
- [0750] 20. 솔루션 14 또는 솔루션 15에 있어서, 상기 비디오 블록은 상기 이웃 블록으로부터 모션 정보 및 상기 분할 패턴 둘 다를 상속하는, 방법.
- [0751] 21. 솔루션 1 내지 솔루션 20 중 어느 하나에 있어서, 상기 비트스트림 표현에서의 지시는 상기 비디오 블록에 대한 상기 기하학적 파티션 모드의 사용을 시그널링하는, 방법.
- [0752] 22. 솔루션 21에 있어서, 상기 지시는 시퀀스 파라미터 세트 레벨 또는 화상 파라미터 세트 레벨 또는 비디오 파라미터 세트 레벨 또는 화상 헤더 또는 슬라이스 헤더 또는 타일 그룹 헤더 또는 코딩 트리 단위 레벨에 있는, 방법.

- [0753] 23. 솔루션 1 내지 솔루션 20 중 어느 하나에 있어서, 상기 비디오 블록이 크기 조건을 만족시키는 것으로 인해 상기 기하학적 파티션 모드가 인에이블되는, 방법.
- [0754] 24. 솔루션 21에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 큰 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.
- [0755] 25. 솔루션 21에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 작은 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.
- [0756] 26. 솔루션 24 내지 솔루션 25에 있어서, 상기 임계치는 코딩된 표현에서 시퀀스 파라미터 세트 또는 화상 파라미터 세트 또는 화상 또는 슬라이스 또는 타일 그룹 또는 타일 레벨에서 시그널링되는, 방법.
- [0757] 27. 솔루션 4 내지 솔루션 5 중 어느 하나에 있어서, 상기 크기는 상기 비디오 블록과 연관된 화상 유형 또는 시간 계층 인덱스의 함수인, 방법.
- [0758] 28. 솔루션 1 내지 솔루션 20 중 어느 하나에 있어서, 상기 비디오 블록이 사전 정의된 폭 대 높이 비율을 갖는 것으로 인해 상기 변환을 위해 상기 기하학적 파티션 모드가 사용되는, 방법.
- [0759] 다음의 솔루션들의 추가적인 특징들 및 변형들이 이전 섹션에서 제공된다(예를 들어, 항목 14, 15, 16, 21).
- [0760] 예시적인 솔루션 세트 D
- [0761] 1. 비디오 처리 방법(예를 들어, 도 54에 묘사된 방법(5400))으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(5402)를 포함하고, 상기 비디오 블록은 제1 예측 파티션을 포함하는 기하학적 파티션 모드의 분할 패턴에 따라 다수의 예측 파티션으로 파티셔닝되고; 상기 코딩된 표현은 다수의 구문 요소를 포함하고, 그 중에서 하나의 구문 요소는 상기 분할 패턴에 대응하고 다른 구문 요소는 상기 다수의 예측 파티션에 대한 병합 인덱스들에 대응하는, 방법.
- [0762] 2. 솔루션 1에 있어서, 분할 패턴은 1 비트 플래그를 사용하여 시그널링되는, 방법.
- [0763] 3. 솔루션 2에 있어서, 상기 1 비트 플래그의 컨텍스트 코딩을 위해 이웃 비디오 블록의 이웃 분할 패턴이 사용되는, 방법.
- [0764] 4. 솔루션 1 내지 솔루션 3 중 어느 하나에 있어서, 상기 다수의 예측 파티션 각각에 대해 다수의 병합 인덱스가 시그널링되는, 방법.
- [0765] 5. 솔루션 1 내지 솔루션 3 중 어느 하나에 있어서, 상기 다수의 예측 파티션에 대한 상기 다수의 병합 인덱스는 예측 코딩을 사용하여 시그널링되는, 방법.
- [0766] 6. 솔루션 1에 있어서, 상기 병합 인덱스들을 시그널링하기 위해 사용되는 코딩 스킴은 상기 분할 패턴을 시그널링하기 위해 사용되는 코딩 스킴에 의존하는, 방법.
- [0767] 7. 솔루션 1에 있어서, 상기 분할 패턴을 시그널링하기 위해 사용되는 코딩 스킴은 상기 병합 인덱스들을 시그널링하기 위해 사용되는 코딩 스킴에 의존하는, 방법.
- [0768] 8. 솔루션 1에 있어서, 상기 변환은 상기 코딩된 표현으로부터 상기 다수의 구문 요소를 파싱하는 단계를 포함하는, 방법.
- [0769] 9. 비디오 처리 방법(예를 들어, 도 55에 묘사된 방법(5500))으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환이 기하학적 파티션 모드에 기초한다고 결정하는 단계(5502) - 상기 비디오 블록은 제1 예측 파티션을 포함하는 분할 패턴에 따라 다수의 예측 파티션으로 파티셔닝됨 -; 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(5504)를 포함하고; 상기 비디오 블록에 대한 상기 기하학적 파티션 모드는 컨텍스트가 상기 비디오 블록의 코딩 조건에 의존하는 컨텍스트 기반 코딩을 사용하여 상기 코딩된 표현으로 시그널링되는, 방법.
- [0770] 10. 솔루션 9에 있어서, 상기 코딩 조건은 상기 비디오 블록의 높이 또는 폭 또는 상기 높이와 상기 폭의 비율에 대응하는, 방법.
- [0771] 11. 솔루션 9에 있어서, 상기 컨텍스트는 파티셔닝 트리 내의 상기 비디오 블록의 코딩된 깊이에 의존하는, 방법.

- [0772] 12. 솔루션 9에 있어서, 상기 코딩 조건은 이웃 블록의 코딩된 정보에 대응하는, 방법.
- [0773] 13. 솔루션 12에 있어서, 상기 비디오 블록은 폭 W 및 높이 H를 갖는, 방법. 여기서, 상기 비디오 블록의 왼쪽 상단 샘플 위치는 (0, 0)으로서 표현되고, 상기 이웃 블록은 위치들 ($W/2$, -1), ($W/2-1$, -1), 또는 ($W-1$, -1)에서의 위의 이웃 블록에 대응한다.
- [0774] 14. 솔루션 13에 있어서, 상기 컨텍스트는 상기 폭과 상기 높이 간의 비율이 임계치보다 큰 것으로 인해 위의 이웃 블록에 의해 완전히 결정되는, 방법.
- [0775] 15. 솔루션 13에 있어서, 상기 비디오 블록은 폭 W 및 높이 H를 갖고, 상기 비디오 블록의 왼쪽 상단 샘플 위치는 (0, 0)으로서 표현되고, 상기 이웃 블록은 위치들 (-1, $H/2$), (-1, $H/2-1$), (-1, $H-1$)에서의 왼쪽 이웃 블록에 대응하는, 방법.
- [0776] 16. 솔루션 15에 있어서, 상기 컨텍스트는 상기 높이와 상기 폭 간의 비율이 임계치보다 큰 것으로 인해 왼쪽 이웃 블록에 의해 완전히 결정되는, 방법.
- [0777] 17. 솔루션 13 및 솔루션 15에 있어서, 상기 컨텍스트는 위의 이웃 블록과 왼쪽 이웃 블록의 조합에 의존하는, 방법.
- [0778] 18. 비디오 처리 방법(예를 들어, 도 56에 묘사된 방법(5600))으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(5602)를 포함하고, 상기 비디오 블록은 제1 예측 파티션을 갖는 분할 패턴에 따라 기하학적 예측 모드를 사용하여 다수의 예측 파티션으로 파티셔닝되고; 상기 후보 리스트의 2개의 후보 인덱스 엔트리에는 상기 코딩된 표현 내의 코드워드가 할당되고, 상기 코딩된 표현의 포맷은 상기 비디오의 세그먼트에 걸쳐 상기 코드워드의 할당을 변경하는 것을 허용하는, 방법.
- [0779] 19. 솔루션 18에 있어서, 상기 비디오의 세그먼트는 상기 비디오 블록 또는 비디오 코딩 트리 단위 또는 비디오타일 또는 비디오 화상 또는 비디오 시퀀스에 대응하는, 방법.
- [0780] 20. 솔루션 18 또는 솔루션 19 중 어느 하나에 있어서, 상기 할당은 상기 2개의 후보 인덱스 엔트리가 동일한 정규 모션 후보로부터의 것인 2개의 후보에 대응하는지에 기초하여 변경되는, 방법.
- [0781] 21. 솔루션 20에 있어서, 상기 할당은 상이한 정규 모션 후보들로부터 도출된 후보들을 나타내는 제1 후보 인덱스 쌍에 할당된 제1 코드워드가 동일한 정규 모션 후보로부터 도출된 후보들을 나타내는 제2 후보 인덱스 쌍에 할당된 제2 코드워드보다 작도록 선택되는 것인, 방법.
- [0782] 22. 솔루션 18 또는 솔루션 19 중 어느 하나에 있어서, 상기 할당은 상기 2개의 후보 인덱스 엔트리가 동일한 예측 방향 또는 동일한 참조 화상 리스트로부터의 것인 2개의 후보에 대응하는지에 기초하여 변경되는, 방법.
- [0783] 23. 솔루션 20에 있어서, 상기 할당은 상이한 예측 방향들을 갖는 후보들을 나타내는 제1 후보 인덱스 쌍에 할당된 제1 코드워드가 동일한 예측 방향을 갖는 후보들을 나타내는 제2 후보 인덱스 쌍에 할당된 제2 코드워드보다 작도록 선택되는 것인, 방법.
- [0784] 24. 솔루션 1 내지 솔루션 23 중 어느 하나에 있어서, 상기 비트스트림 표현에서의 지시는 상기 비디오 블록에 대한 상기 기하학적 파티션 모드의 사용을 시그널링하는, 방법.
- [0785] 25. 솔루션 24에 있어서, 상기 지시는 시퀀스 파라미터 세트 레벨 또는 화상 파라미터 세트 레벨 또는 비디오 파라미터 세트 레벨 또는 화상 헤더 또는 슬라이스 헤더 또는 타일 그룹 헤더 또는 코딩 트리 단위 레벨에 있는, 방법.
- [0786] 26. 솔루션 1 내지 솔루션 25 중 어느 하나에 있어서, 상기 비디오 블록이 크기 조건을 만족시키는 것으로 인해 상기 기하학적 파티션 모드가 인에이블되는, 방법.
- [0787] 27. 솔루션 26에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 큰 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.
- [0788] 28. 솔루션 26에 있어서, 상기 크기 조건은 상기 비디오 블록이 임계치보다 작은 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 사용하는 것을 특정하는, 방법.
- [0789] 29. 솔루션 27 내지 솔루션 28에 있어서, 상기 임계치는 코딩된 표현에서 시퀀스 파라미터 세트 또는 화상 파라미터 세트 또는 화상 또는 슬라이스 또는 타일 그룹 또는 타일 레벨에서 시그널링되는, 방법.

- [0790] 30. 솔루션 27 내지 솔루션 28 중 어느 하나에 있어서, 상기 크기는 상기 비디오 블록과 연관된 화상 유형 또는 시간 계층 인덱스의 함수인, 방법.
- [0791] 31. 솔루션 1 내지 솔루션 30 중 어느 하나에 있어서, 상기 비디오 블록이 사전 정의된 폭 대 높이 비율을 갖는 것으로 인해 상기 변환을 위해 상기 기하학적 파티션 모드가 사용되는, 방법.
- [0792] 다음의 솔루션들의 추가적인 특징들 및 변형들이 이전 섹션에서 제공된다(예를 들어, 항목 18 내지 21).
- [0793] 예시적인 솔루션 세트 E
- [0794] 1. 비디오 처리 방법(예를 들어, 도 57에 묘사된 방법(5700))으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 비디오 블록의 특성을 사용하는 규칙에 기초하여 기하학적 파티션 모드의 사용이 인에이블되는지를 결정하는 단계(5702); 상기 결정에 따라 상기 변환을 수행하는 단계(5704)를 포함하고, 상기 기하학적 파티션 모드는 상기 비디오 블록을 다수의 예측 파티션으로 파티셔닝하는, 방법.
- [0795] 2. 솔루션 1에 있어서, 상기 규칙은 상기 비디오 블록이 상기 비디오의 컬러 성분을 나타내는 것으로 인해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하는, 방법.
- [0796] 3. 솔루션 1에 있어서, 상기 규칙은 상기 비디오 블록이 임계치보다 작은 크기 또는 폭을 갖는 크로마 블록인 것으로 인해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하는, 방법.
- [0797] 4. 솔루션 1에 있어서, 상기 규칙은 상기 비디오 블록이 NxM 또는 KxN 크기를 갖는 것으로 인해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하고, K, M, N은 서로 상이한 정수들인, 방법.
- [0798] 5. 솔루션 4에 있어서, N은 4 또는 128과 동등한, 방법.
- [0799] 6. 솔루션 1에 있어서, 상기 규칙은 상기 비디오 블록이 임계치 L보다 큰 폭 및/또는 높이를 갖는 것으로 인해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하고, L은 정수인, 방법.
- [0800] 7. 솔루션 6에 있어서, L은 64와 동등한, 방법.
- [0801] 8. 솔루션 1에 있어서, 상기 규칙은 상기 비디오 블록이 임계치 L보다 작은 폭 및/또는 높이를 갖는 것으로 인해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하고, L은 정수인, 방법.
- [0802] 9. 솔루션 1에 있어서, 상기 규칙은 상기 비디오 블록이 폭을 높이로 나눈 값이 임계치 L보다 큰 비율을 갖는 것으로 인해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하고, L은 정수인, 방법.
- [0803] 10. 솔루션 1에 있어서, 상기 규칙은 상기 비디오 블록이 높이를 폭으로 나눈 값이 임계치 L보다 큰 비율을 갖는 것으로 인해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하고, L은 정수인, 방법.
- [0804] 11. 솔루션 8 내지 솔루션 10 중 어느 하나에 있어서, L은 8과 동등한, 방법.
- [0805] 12. 솔루션 5 내지 솔루션 10 중 어느 하나에 있어서, 임계치들 L, M, N 및 K 중 적어도 하나는 코딩된 표현에서 시퀀스 파라미터 세트 또는 화상 파라미터 세트 또는 화상 헤더 또는 슬라이스 헤더 또는 타일 그룹 헤더 또는 타일 헤더에서 시그널링되는, 방법.
- [0806] 13. 솔루션 5 내지 솔루션 12 중 어느 하나에 있어서, 임계치 L, M, N 및 K 중 적어도 하나는 상기 비디오 블록과 연관된 블록 크기 또는 화상 유형 또는 시간 계층 인덱스의 함수인, 방법.
- [0807] 14. 솔루션 1 내지 솔루션 13 중 어느 하나에 있어서, 상기 코딩된 표현은 상기 비디오 블록에 대해 상기 기하학적 파티션 모드가 디스에이블된다고 결정될 때 상기 기하학적 파티션 모드의 사용의 지시를 포함하는 것을 생략하는, 방법.
- [0808] 15. 솔루션 1에 있어서, 상기 비디오 블록의 특성은 상기 비디오 블록이 속하는 비디오의 비디오 컴포넌트를 포함하는, 방법.
- [0809] 16. 솔루션 15에 있어서, 상기 비디오 블록은 다수의 컬러 성분을 갖는 샘플들을 포함하고 상기 규칙은 하나의 컬러 성분에 대해서는 상기 기하학적 파티션 모드를 디스에이블하는 반면 다른 컬러 성분에 대해서는 상기 기하학적 파티션 모드를 인에이블하는 것을 특정하는, 방법.
- [0810] 17. 솔루션 1에 있어서, 상기 규칙은 NxM 또는 LxN 크기를 갖는 루마 블록들에 대해 상기 기하학적 파티션 모드를 인에이블하는 반면 - L, M, N은 양의 정수임 -, 상기 비디오가 4:2:0 포맷을 갖는다면 대응하는 크로마 블록

들에 대해 상기 기하학적 파티션 모드를 디스에이블하는 것을 특정하는, 방법.

[0811] 18. 솔루션 1에 있어서, 상기 비디오 블록에 대해 상기 기하학적 파티션 모드가 디스에이블된다고 결정되는 경우에는, 상기 변환은 상기 비디오 블록의 하나의 예측 파티션의 모션 정보 또는 예측 모드를 사용하여 상기 비디오 블록에 대한 예측 블록을 도출하는 것을 포함하는, 방법.

[0812] 19. 솔루션 1에 있어서, 상기 비디오 블록에 대해 상기 기하학적 파티션 모드가 디스에이블된다고 결정되는 경우에는, 상기 변환은 상기 비디오 블록의 2개의 예측 파티션과 연관된 모션 정보로부터의 도출된 모션 정보를 사용하여 상기 비디오 블록에 대한 예측 블록을 도출하는 것을 포함하는, 방법.

[0813] 20. 솔루션 19에 있어서, 2개의 상이한 방향에서의 모션 벡터들을 갖는 상기 2개의 예측 파티션의 모션 정보는 상기 비디오 블록에 대한 양방향 예측 모션 벡터들을 생성하기 위해 사용되는, 방법.

[0814] 21. 솔루션 19에 있어서, 동일한 방향에서의 모션 벡터들을 갖는 상기 2개의 예측 파티션의 모션 정보는 제1 파티션의 제1 모션 벡터가 제2 파티션의 참조 화상에 따라 스케일링되고 상기 스케일링된 모션 벡터와 상기 제2 파티션의 제2 모션 벡터의 평균이 사용되는 스케일링 동작을 사용하여 상기 비디오 블록에 대한 단방향 예측 모션 벡터를 생성하기 위해 사용되는, 방법.

[0815] 22. 솔루션 16에 있어서, 2개의 방향에서의 모션 벡터들을 갖는 상기 2개의 예측 파티션의 모션 정보는 제1 파티션의 제1 모션 벡터가 제2 파티션의 참조 화상에 따라 스케일링되고 상기 스케일링된 모션 벡터와 상기 제2 파티션의 제2 모션 벡터의 평균이 사용되는 스케일링 동작을 사용하여 상기 비디오 블록에 대한 양방향 예측 모션 벡터를 생성하기 위해 사용되는, 방법.

[0816] 23. 솔루션 21 및 솔루션 22 중 어느 하나에 있어서, 상기 스케일링 동작은 상기 비디오의 컬러 포맷에 의존하는, 방법.

[0817] 24. 솔루션 1 내지 솔루션 23 중 어느 하나에 있어서, 상기 비트스트림 표현에서의 지시는 상기 비디오 블록에 대한 상기 기하학적 파티션 모드의 사용을 시그널링하는, 방법.

[0818] 25. 솔루션 24에 있어서, 상기 지시는 시퀀스 파라미터 세트 레벨 또는 화상 파라미터 세트 레벨 또는 비디오 파라미터 세트 레벨 또는 화상 헤더 또는 슬라이스 헤더 또는 타일 그룹 헤더 또는 코딩 트리 단위 레벨에 있는, 방법.

[0819] 26. 솔루션 24 내지 솔루션 25 중 어느 하나에 있어서, 상기 크기는 상기 비디오 블록과 연관된 화상 유형 또는 시간 계층 인덱스의 함수인, 방법.

[0820] 다음의 솔루션들의 추가적인 특징들 및 변형들이 이전 섹션에서 제공된다(예를 들어, 항목 22-27).

예시적인 솔루션 세트 F

[0822] 1. 비디오 처리 방법(예를 들어, 도 58에 묘사된 방법(5800))으로서, 코딩된 표현의 구문 요소들의 포맷을 정의하는 규칙에 따라 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(5802)를 포함하고; 상기 규칙은 제1 코딩 모드의 사용의 지시가 시그널링되는지 또는 상기 지시를 시그널링할지가 상기 비디오 블록에 대한 제2 코딩 모드의 사용에 기초한다는 것을 특정하고, 상기 제2 코딩 모드는 병합 모드이고; 상기 병합 모드는 상기 비디오 블록 전체에 대한 모션 벡터 차이 없이 병합 후보 리스트 내의 병합 후보로부터 모션 정보를 상속하는 것을 인에이블하는, 방법.

[0823] 2. 솔루션 1에 있어서, 상기 제1 코딩 모드는 상기 비디오 블록이 다수의 예측 파티션으로 파티셔닝되는 기하학적 파티션 모드를 포함하는, 방법.

[0824] 3. 솔루션 2에 있어서, 상기 다수의 예측 파티션 중 적어도 하나는 비-직사각형 및 비-정사각형 예측 파티션인, 방법.

[0825] 4. 솔루션 1에 있어서, 상기 제1 코딩 모드는 상기 비디오 블록의 예측 블록이 인트라 예측 신호 및 인터 예측 신호로부터 도출되는 인터-인트라 코딩 모드를 포함하는, 방법.

[0826] 5. 솔루션 1에 있어서, 상기 제1 코딩 모드는 서브-블록 병합 코딩 모드를 포함하고, 상기 서브-블록 병합 모드에서, 상기 변환은 상기 블록 내의 각각의 서브-블록에 대해 도출된 모션 정보를 사용하는, 방법.

[0827] 6. 솔루션 1 내지 솔루션 5 중 어느 하나에 있어서, 상기 규칙은 상기 비디오 블록에 대해 상기 제2 코딩 모드가 인에이블되는 것으로 인해 상기 제1 코딩 모드의 사용을 나타내는 지시의 값이 거짓으로 설정되는 것을 특정

하는, 방법.

- [0828] 7. 솔루션 1 내지 솔루션 5 중 어느 하나에 있어서, 상기 규칙은 상기 비디오 블록에 대해 상기 제2 코딩 모드가 인에이블되는 것으로 인해 상기 제1 코딩 모드의 사용의 지시가 상기 코딩된 표현으로부터 제외되는 것을 특정하는, 방법.
- [0829] 8. 솔루션 1 내지 솔루션 7 중 어느 하나에 있어서, 상기 변환은 상기 규칙에 따라 상기 코딩된 표현을 파싱하는 것을 포함하는, 방법.
- [0830] 9. 비디오 처리 방법(예를 들어, 도 59에 묘사된 방법(5900))으로서, 비디오의 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(5902)를 포함하고, 상기 코딩된 표현의 포맷은 다수의 코딩 모드를 시그널링하는 구문 요소들이 상기 코딩된 표현에 포함되는지를 특정하는 제1 규칙 및/또는 상기 다수의 코딩 모드를 시그널링하는 구문 요소들이 상기 코딩된 표현에 포함되는 순서를 특정하는 제2 규칙에 따르고; 상기 다수의 코딩 모드는 기하학적 파티션 모드, 인터-인트라(MHIntra) 모드, 서브-블록 병합 모드 또는 MMVD(merge with motion vector differencing) 모드 중 2개 이상을 포함하고; 상기 인터-인트라 코딩 모드에서는, 상기 비디오 블록의 예측 블록이 인트라 예측 신호 및 인터 예측 신호로부터 도출되고; 상기 서브-블록 병합 모드에서는, 상기 변환이 상기 블록 내의 각각의 서브-블록에 대해 도출된 모션 정보를 사용하고; 상기 MMVD 모드에서는, 조합된 병합 및 MVD(motion vector differencing) 코딩 모드가 사용되고; 상기 병합 모드는 상기 비디오 블록 전체에 대한 MVD 없이 병합 후보 리스트 내의 병합 후보로부터 모션 정보를 상속하는 것을 인에이블하는, 방법.
- [0831] 10. 솔루션 9에 있어서, 상기 제2 규칙은 구문 요소들의 순서를: 상기 MMVD 모드, 상기 서브-블록 병합 모드, 상기 기하학적 파티션 모드, 및 MHIntra 모드로 특정하는, 방법.
- [0832] 11. 솔루션 9에 있어서, 상기 제2 규칙은 구문 요소들의 순서를: 상기 MMVD 모드, 상기 서브-블록 병합 모드, 상기 MHIntra 모드, 및 상기 기하학적 파티션 모드로 특정하는, 방법.
- [0833] 12. 솔루션 9에 있어서, 상기 제2 규칙은 구문 요소들의 순서를: 상기 서브-블록 병합 모드가 상기 MMVD 모드 이전에 시그널링되는 것으로 특정하는, 방법.
- [0834] 13. 솔루션 9에 있어서, 상기 제2 규칙은 구문 요소들의 순서를: 상기 서브-블록 병합 모드, 상기 MMVD 모드, 상기 MHIntra 모드, 및 상기 기하학적 파티션 모드로 특정하는, 방법.
- [0835] 14. 솔루션 9에 있어서, 상기 순서는 상기 비디오의 코딩 조건에 기초하여 비디오 단위 레벨에서 상기 코딩된 표현에서 변경되는, 방법.
- [0836] 15. 솔루션 12에 있어서, 상기 비디오 단위는 블록, 코딩 단위, 예측 단위, 코딩 트리 단위 또는 최대 코딩 단위 행, 비디오 슬라이스, 화상 또는 시퀀스 레벨에 대응하는, 방법.
- [0837] 16. 솔루션 12 내지 솔루션 15 중 어느 하나에 있어서, 상기 코딩 조건은 저지연 검사 플래그의 값, 이전에 코딩된 비디오 블록들의 코딩 정보, 또는 상기 비디오 블록의 치수를 포함하는, 방법.
- [0838] 17. 비디오 처리 방법(예를 들어, 도 60에 묘사된 방법(6000))으로서,
- [0839] 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 비디오 블록의 이웃 블록의 이웃 모션 벡터 정밀도 정보가 이웃 블록의 위치에 기초하여 현재 블록에 대한 모션 벡터 정밀도를 결정하기 위해 이용가능한지를 검사하는 단계(6002); 및 상기 검사에 기초하여 상기 변환을 수행하는 단계(6004)를 포함하는, 방법.
- [0840] 18. 솔루션 17에 있어서, 상기 이웃 블록 및 상기 현재 비디오 블록이 상기 비디오의 상이한 최대 코딩 단위(LCU) 행들 또는 상이한 영역들에 있는 경우에는, 상기 이웃 모션 벡터 정밀도 정보가 이용불가능한 것으로 간주되는, 방법.
- [0841] 19. 솔루션 18에 있어서, 상기 이웃 모션 벡터 정밀도 정보가 이용불가능하다고 결정하는 것으로 인해, 상기 현재 블록에 대한 모션 벡터 정밀도는 디폴트 값으로 설정되는, 방법.
- [0842] 20. 솔루션 17에 있어서, 상기 이웃 블록 및 상기 현재 비디오 블록이 상이한 최대 코딩 단위(LCU) 행들에 있는 경우에는, 상기 현재 비디오 블록에 대한 모션 벡터 정밀도는 상기 이웃 블록이 상이한 슬라이스 또는 상이한 타일에 있는 것으로 가정하는 것에 의해 결정되는, 방법.
- [0843] 21. 솔루션 17 내지 솔루션 20 중 어느 하나에 있어서, 상기 현재 비디오 블록 및 상기 이웃 블록은 아핀 코딩

모드를 사용하여 코딩되는, 방법.

[0844] 22. 비디오 처리 방법(예를 들어, 도 61에 묘사된 방법(6100))으로서,

[0845] 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 위해, 상기 현재 비디오 블록의 최대 코딩 단위(LCU) 행에 대한 이웃 블록의 위치에 기초하여 상기 코딩된 표현을 위해 사용되는 컨텍스트 코딩 모드를 결정하는 단계(6102); 및 상기 결정에 기초하여 상기 변환을 수행하는 단계(6104)를 포함하는, 방법.

[0846] 23. 솔루션 22에 있어서, 상기 이웃 블록은 상이한 LCU 행에 있는 것으로 인해 이용불가능한 것으로 간주되고, 상기 컨텍스트 코딩 모드는 구문 요소를 코딩하기 위해 단일 컨텍스트를 사용하는, 방법.

[0847] 24. 솔루션 22에 있어서, 상기 이웃 블록은 상이한 LCU 행에 있는 것으로 인해 이용불가능한 것으로 간주되고, 상기 컨텍스트 코딩 모드는 구문 요소를 코딩하기 위해 바이패스 코딩을 사용하는, 방법.

[0848] 25. 솔루션 23 내지 솔루션 24 중 어느 하나에 있어서, 상기 구문 요소는: alf_ctb_flag, qt_split_cu_flag, mtt_split_cu_flag, cu_skip_flag, amvr_mode, merge_subblock_flag, merge_triangle_flag, inter_affine_flag, cu_skip_flag, pred_mode_flag, pcm_flag, intra_luma_ref_idx, intra_luma_mpm_flag, intra_luma_mpm_idx, intra_luma_mpm_remainder, intra_chroma_pred_mode, merge_flag, inter_pred_idc, inter_affine_flag, cu_affine_type_flag, ref_idx_10, mvp_10_flag, ref_idx_11_mvp_11_flag, amvr_flag, amvr_4pel_flag, gbi_idx, cu_cbf, mmvd_flag, 또는 mmvd_merge_flag 중 하나인, 방법.

[0849] 다음의 솔루션들의 추가적인 특징들 및 변형들이 이전 섹션에서 제공된다(예를 들어, 항목 1 내지 4, 17 및 21).

예시적인 솔루션 세트 G

[0851] 1. 비디오 처리 방법(예를 들어, 도 62에 묘사된 방법(6200))으로서, 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(6202)를 포함하고, 상기 코딩된 표현은 컨텍스트 모델링이 상기 현재 비디오 블록을 디코딩하기 위해 사용되는 이웃 블록으로부터의 정보로 제한되고/되거나 위치 기준을 만족시키는 것을 특정하는 규칙에 따르는, 방법.

[0852] 2. 솔루션 1에 있어서, 상기 이웃 블록은 왼쪽 하단 블록 또는 오른쪽 상단 블록인, 방법.

[0853] 3. 솔루션 1에 있어서, 상기 규칙은 상기 현재 비디오 블록을 포함하는 현재 비디오 단위 외부에 있는 이웃 블록들의 사용을 허용하지 않는 것을 특정하는, 방법.

[0854] 4. 솔루션 1에 있어서, 상기 규칙은 이용불가능한 블록들로서 상기 현재 비디오 블록을 포함하는 현재 비디오 단위 외부에 있는 이웃 블록들을 사용하는 것을 특정하는, 방법.

[0855] 5. 솔루션 1에 있어서, 상기 규칙은 디폴트 값들을 갖는 것으로서 상기 현재 비디오 블록을 포함하는 현재 비디오 단위 외부에 있는 이웃 블록들을 사용하는 것을 특정하는, 방법.

[0856] 6. 솔루션 1에 있어서, 상기 규칙은 이용불가능한 블록들로서 대체 블록들로서 상기 현재 비디오 블록을 포함하는 현재 비디오 단위 외부에 있는 이웃 블록들을 사용하는 것을 특정하는, 방법.

[0857] 7. 솔루션 3에 있어서, 상기 현재 비디오 단위는 코딩 트리 단위 또는 코딩 트리 단위 행 또는 비디오 영역 또는 비디오 데이터 처리 단위 또는 슬라이스 또는 타일 또는 타일 그룹이거나 화상 경계에 있는, 방법.

[0858] 8. 비디오 처리 방법(예를 들어, 도 63에 묘사된 방법(6300))으로서, 비디오의 현재 비디오 블록과 상기 비디오의 코딩된 표현 간의 변환을 수행하는 단계(6302)를 포함하고, 상기 코딩된 표현은 상기 현재 비디오 블록의 코딩된 표현을 나타내는 하나 이상의 구문 요소의 컨텍스트 모델링을 위해 이웃 블록들로부터의 코딩 정보의 사용을 특정하는 규칙에 따르는, 방법.

[0859] 9. 솔루션 8에 있어서, 이웃 블록들로부터의 코딩 정보는 비트스트림 표현에서 상기 이웃 블록들을 나타내는 구문 요소들에 기초하는, 방법.

[0860] 10. 솔루션 8 또는 솔루션 9에 있어서, 상기 이웃 블록들은 상기 코딩된 표현으로부터 상기 현재 비디오 블록을 재구성하기 위해 액세스되는 블록들에 대응하는, 방법.

[0861] 11. 솔루션 8 내지 솔루션 10 중 어느 하나에 있어서, 상기 이웃 블록들은 공간 병합 후보 구성을 위해 사용되

는 5개의 위치로부터의 정확히 2개의 블록에 대응하는, 방법.

[0862] 12. 솔루션 8 내지 솔루션 10 중 어느 하나에 있어서, 상기 이웃 블록들은 공간 병합 후보 구성을 위해 사용되는 5개의 위치로부터의 정확히 1개의 블록에 대응하는, 방법.

[0863] 13. 솔루션 8에 있어서, 상기 규칙은 상기 현재 비디오 블록의 코딩 트리 단위 또는 코딩 트리 단위 행 외부에 있는 이웃 블록들을 사용하는 것을 제외하는 것을 특정하는, 방법.

[0864] 14. 솔루션 8에 있어서, 상기 규칙은 상기 현재 비디오 블록의 코딩 트리 단위 또는 코딩 트리 단위 행 외부에 있는 이웃 블록들이 컨텍스트 모델링에 이용불가능한 것으로 간주되는 구문 요소들을 사용하는 것을 특정하는, 방법.

[0865] 15. 솔루션 8에 있어서, 상기 규칙은 상기 현재 비디오 블록의 코딩 트리 단위 또는 코딩 트리 단위 행 외부에 있는 이웃 블록들이 컨텍스트 모델링을 위해 디폴트 값들을 갖는 것으로 간주되는 구문 요소들을 사용하는 것을 특정하는, 방법.

[0866] 16. 솔루션 8에 있어서, 상기 규칙은 상기 현재 비디오 블록의 코딩 트리 단위 또는 코딩 트리 단위 행 외부에 있는 이웃 블록들이 컨텍스트 모델링을 위해 외부의 이웃 블록으로 대체하는 것에 의해 사용되는 구문 요소들을 사용하는 것을 특정하는, 방법.

[0867] 17. 솔루션 8에 있어서, 상기 규칙은 상기 현재 비디오 블록의 코딩 트리 단위 또는 코딩 트리 단위 행 외부에 있는 이웃 블록들이 상기 코딩 트리 단위 내의 N개의 이용가능한 이웃 블록들 중 하나로 대체되거나 상기 코딩 트리 단위가 컨텍스트 모델링을 위해 이용가능한 코딩 정보를 포함하는 구문 요소들을 사용하는 것을 특정하고, N은 정수인, 방법.

[0868] 18. 솔루션 8 내지 솔루션 17 중 어느 하나에 있어서, 상기 하나 이상의 구문 요소는: alf_ctb_flag, qt_split_cu_flag, mtt_split_cu_flag, cu_skip_flag, amvr_mode, inter_affine_flag, merge_triangle_flag and merge_subblock_flag, cu_skip_flag, pred_mode_flag, pcm_flag, intra_luma_ref_idx, intra_luma_mpm_flag, intra_luma_mpm_idx, intra_luma_mpm_remainder, intra_chroma_pred_mode, merge_flag, inter_pred_idc, inter_affine_flag, cu_affine_type_flag, ref_idx_10, mvp_10_flag, ref_idx_11_mvp_11_flag, amvr_flag, amvr_4pel_flag, gbi_idx, cu_cbf, mmvd_flag, mmvd_merge_flag, 또는 pred_mode_flag를 포함하는, 방법.

[0869] 19. 솔루션 18에서, 구문 요소의 컨텍스트 ctx는:

[0870] `ctx = A1이 인트라 ? 1 : 0; ctx += B1이 인트라 ? 1 : 0`

[0871] 으로서 도출되고, A1은 왼쪽 하단 이웃 블록이고 B1은 오른쪽 상단 이웃 블록인, 방법.

[0872] 20. 솔루션 18에 있어서, pred_mode_flag의 컨텍스트 ctx는:

[0873] `ctx = (A1이 인트라) || (B1이 인트라) ? 1 : 0`

[0874] 으로서 도출되고, A1은 왼쪽 하단 이웃 블록이고 B1은 오른쪽 상단 이웃 블록인, 방법.

[0875] 21. 솔루션 18에서, 구문 요소의 컨텍스트 ctx는:

[0876] `ctx = A1이 인트라 또는 인터-인트라 ? 1 : 0; ctx += B1이 인트라 또는 인터-인트라 ? 1 : 0`

[0877] 으로서 도출되고; A1은 왼쪽 하단 이웃 블록이고 B1은 오른쪽 상단 이웃 블록인, 방법.

[0878] 22. 솔루션 18에서, 구문 요소의 컨텍스트 ctx는:

[0879] `ctx = (A1이 인트라 또는 인터-인트라) || (B1이 인트라 또는 인터-인트라) ? 1 : 0`

[0880] 으로서 도출되고; A1은 왼쪽 하단 이웃 블록이고 B1은 오른쪽 상단 이웃 블록인, 방법.

[0881] 23. 솔루션 18에 있어서, 구문 요소의 컨텍스트 ctx는 공간 병합 후보 구성을 위해 사용되는 5개의 위치에서의 이웃 블록들을 사용하는 것을 제외하는 것에 의해 도출되는, 방법.

[0882] 24. 솔루션 23에 있어서, 상기 구문 요소는 pred_mode_flag에 대응하는, 방법.

[0883] 25. 솔루션 18 내지 솔루션 24 중 어느 하나에 있어서, 상기 컨텍스트는 이용불가능한 이웃 블록을 인터-코딩된

블록으로서 취급하는 것에 의해 도출되는, 방법.

[0884] 26. 솔루션 18 내지 솔루션 24 중 어느 하나에 있어서, 상기 컨텍스트는 이용불가능한 이웃 블록을 인트라-코딩된 블록으로서 취급하는 것에 의해 도출되는, 방법.

[0885] 27. 솔루션 18 내지 솔루션 24 중 어느 하나에 있어서, 상기 컨텍스트는 이용불가능한 이웃 블록을 인터-인트라 코딩된 블록으로서 취급하는 것에 의해 도출되는, 방법.

[0886] 28. 솔루션 18 내지 솔루션 24 중 어느 하나에 있어서, 상기 컨텍스트는 상이한 코딩 트리 단위 행에 있는 이웃 블록을 인터-코딩된 블록으로서 상기 현재 비디오 블록으로서 취급하는 것에 의해 도출되는, 방법.

[0887] 29. 솔루션 18 내지 솔루션 24 중 어느 하나에 있어서, 상기 컨텍스트는 상이한 코딩 트리 단위 행에 있는 이웃 블록을 인트라-코딩된 블록으로서 상기 현재 비디오 블록으로서 취급하는 것에 의해 도출되는, 방법.

[0888] 30. 솔루션 18 내지 솔루션 24 중 어느 하나에 있어서, 상기 컨텍스트는 상이한 코딩 트리 단위 행에 있는 이웃 블록을 인터-인트라 코딩된 블록으로서 상기 현재 비디오 블록으로서 취급하는 것에 의해 도출되는, 방법.

[0889] 위에 열거된 솔루션 세트들에 관련하여, 일부 실시예들에서, 상기 기하학적 파티션 모드(본 문서에서 기하학적 파티션 모드라고도 불림)는 비디오 블록을 다수의 파티션으로 파티셔닝하는 것을 포함할 수 있고, 여기서 적어도 하나의 파티션은 비-정사각형 및 비-직사각형 파티션이다. 본 문서 전체에 걸쳐 더 개시되는 바와 같이, 예측 파티션들이라고 불리는 이들 파티션은 비디오 블록의 코딩 또는 디코딩 동안 사용되는 예측 블록들을 형성하기 위해 사용될 수 있다.

[0890] 위에 열거된 솔루션 세트들에 관련하여, 일부 실시예들에서, 상기 변환은 비디오 픽셀 값들 또는 다른 표현으로부터 코딩된 표현을 생성하기 위해 비디오를 인코딩하는 것(예를 들어, 트랜스코딩)을 포함할 수 있다.

[0891] 위에 열거된 솔루션 세트들에 관련하여, 일부 실시예들에서, 일부 실시예들에서, 상기 변환은 상기 코딩된 표현 또는 비트스트림이 비디오 샘플들을 생성 또는 재구성하기 위해 디코딩 로직에 의해 파생되는 비디오 디코딩 동작을 포함할 수 있다. 이와 관련하여, 상기 변환은 본 문서에서 설명된 다양한 규칙들에 따라 구문 요소들이 파생되는 파싱 프로세스를 포함한다.

[0892] 일부 실시예들에서, 개시된 기법들은 비디오 디코더 장치(예를 들어, 도 31에 도시된 바와 같은)에 의해 구현된다. 일부 실시예들에서, 비디오 인코더는 본 문서에서 설명된 다양한 기법들을 구현할 수 있다. 일부 실시예들에서, 개시된 솔루션들은 프로세서를 사용하여 위에 열거된 솔루션들을 구현하기 위한 코드를 저장하는 컴퓨터 프로그램 제품으로서 구현될 수 있다.

[0893] 개시된 기술의 일부 실시예들은 비디오 처리 툴 또는 모드를 인에이블하라는 판정 또는 결정을 하는 것을 포함한다. 일 예에서, 상기 비디오 처리 툴 또는 모드가 인에이블될 때, 상기 인코더는 비디오의 블록의 처리에서 상기 툴 또는 모드를 사용하거나 구현할 것이지만, 상기 툴 또는 모드의 사용에 기초하여 결과적인 비트스트림을 반드시 수정하는 것은 아닐 수 있다. 즉, 비디오의 블록으로부터 비디오의 비트스트림 표현으로의 변환은 상기 판정 또는 결정에 기초하여 상기 비디오 처리 툴 또는 모드가 인에이블될 때 그것을 사용할 것이다. 다른 예에서, 상기 비디오 처리 툴 또는 모드가 인에이블될 때, 상기 디코더는 상기 비트스트림이 상기 비디오 처리 툴 또는 모드에 기초하여 수정되었다는 것을 알고 상기 비트스트림을 처리할 것이다. 즉, 비디오의 비트스트림 표현으로부터 비디오의 블록으로의 변환은 상기 판정 또는 결정에 기초하여 인에이블된 상기 비디오 처리 툴 또는 모드를 사용하여 수행될 것이다.

[0894] 개시된 기술의 일부 실시예들은 비디오 처리 툴 또는 모드를 디스에이블하라는 판정 또는 결정을 하는 것을 포함한다. 일 예에서, 상기 비디오 처리 툴 또는 모드가 디스에이블될 때, 상기 인코더는 상기 비디오의 블록을 상기 비디오의 비트스트림 표현으로 변환할 때 상기 툴 또는 모드를 사용하지 않을 것이다. 다른 예에서, 상기 비디오 처리 툴 또는 모드가 디스에이블될 때, 상기 디코더는 상기 판정 또는 결정에 기초하여 인에이블된 상기 비디오 처리 툴 또는 모드를 사용하여 상기 비트스트림이 수정되지 않았다는 것을 알고 상기 비트스트림을 처리할 것이다.

[0895] 도 64는 개시된 기법들이 구현될 수 있는 예시적인 비디오 처리 시스템의 블록도이다.

[0896] 도 64는 본 명세서에 개시된 다양한 기법들이 구현될 수 있는 예시적인 비디오 처리 시스템(1900)을 보여주는 블록도이다. 다양한 구현들은 시스템(1900)의 컴포넌트들의 일부 또는 전부를 포함할 수 있다. 시스템(1900)은 비디오 콘텐츠를 수신하기 위한 입력(1902)을 포함할 수 있다. 비디오 콘텐츠는 원시 또는 압축되지 않은

포맷, 예를 들어, 8 또는 10 비트 다중-성분 픽셀 값들로 수신될 수 있거나, 압축된 또는 인코딩된 포맷일 수 있다. 입력(1902)은 네트워크 인터페이스, 주변 버스 인터페이스, 또는 저장 인터페이스를 나타낼 수 있다. 네트워크 인터페이스의 예들은 이더넷, 수동 광 네트워크(passive optical network, PON) 등과 같은 유선 인터페이스들 및 Wi-Fi 또는 셀룰러 인터페이스들과 같은 무선 인터페이스들을 포함한다.

[0897] 시스템(1900)은 본 문서에서 설명된 다양한 코딩 또는 인코딩 방법들을 구현할 수 있는 코딩 컴포넌트(1904)를 포함할 수 있다. 코딩 컴포넌트(1904)는 비디오의 평균 비트레이트를 입력(1902)으로부터 코딩 컴포넌트(1904)의 출력으로 감소시켜 비디오의 코딩된 표현을 생성할 수 있다. 따라서 코딩 기법들은 때때로 비디오 압축 또는 비디오 트랜스코딩 기법이라고 불린다. 코딩 컴포넌트(1904)의 출력은, 컴포넌트(1906)로 나타낸 바와 같이, 저장되거나 연결된 통신을 통해 송신될 수 있다. 입력(1902)에서 수신된 비디오의 저장된 또는 통신된 비트스트림(또는 코딩된) 표현은, 디스플레이 인터페이스(1910)에 송신되는 픽셀 값들 또는 디스플레이 가능한 비디오를 생성하기 위해 컴포넌트(1908)에 의해 사용될 수 있다. 비트스트림 표현으로부터 사용자-시청가능 비디오를 생성하는 프로세스는 때때로 비디오 압축해제라고 불린다. 더욱이, 특정 비디오 처리 동작들이 "코딩" 동작들 또는 툴들로 지칭되지만, 코딩 툴들 또는 동작들이 인코더에서 사용되고 코딩의 결과들을 반전시키는 대응하는 디코딩 툴들 또는 동작들이 디코더에 의해 수행될 것임이 이해될 것이다.

[0898] 주변 버스 인터페이스 또는 디스플레이 인터페이스의 예들은 USB(universal serial bus) 또는 HDMI(high definition multimedia interface) 또는 Displayport 등을 포함할 수 있다. 저장 인터페이스들의 예들은 SATA(serial advanced technology attachment), PCI, IDE 인터페이스 등을 포함한다. 본 문서에서 설명된 기법들은, 디지털 데이터 처리 및/또는 비디오 디스플레이를 수행할 수 있는 모바일 전화, 랩톱, 스마트폰 또는 다른 디바이스와 같은 다양한 전자 디바이스들에서 구현될 수 있다.

[0899] 개시된 기법들은 압축되고 있는 코딩 단위들이 전통적인 정사각형 형상의 블록들 또는 반-정사각형 형상인 직사각형 블록들과 상당히 상이한 형상을 가질 때 압축 효율을 개선하기 위해 비디오 인코더들 또는 디코더들에서 구현될 수 있다는 것이 이해될 것이다. 예를 들어, 4x32 또는 32x4 크기의 단위들과 같은 긴 또는 높은 코딩 단위들을 사용하는 새로운 코딩 툴들이 개시된 기법들로부터 이익을 얻을 수 있다.

[0900] 서브-블록 코딩 모드들이 함께 사용되거나 함께 사용으로부터 제외되는 비디오 처리에 대한 다양한 규칙들이 특정된다는 것이 또한 이해될 것이다.

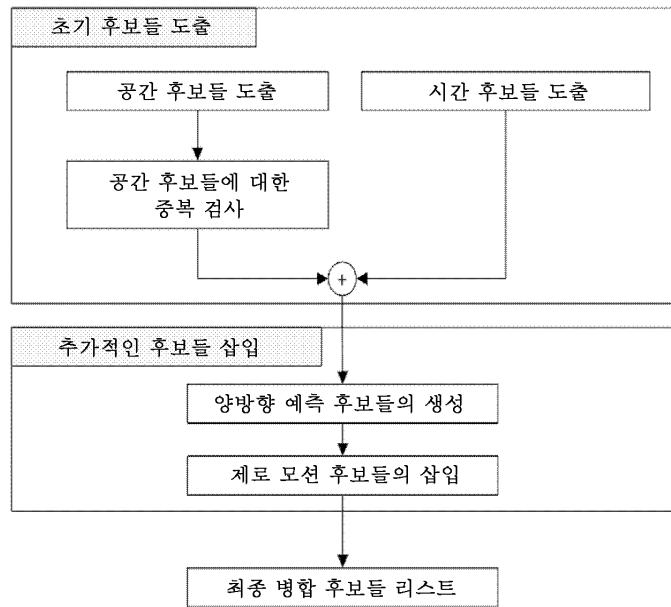
[0901] 본 문서에서 설명된 개시된 그리고 다른 솔루션들, 예들, 실시예들, 모듈들 및 기능 동작들은 디지털 전자 회로에서, 또는 컴퓨터 소프트웨어, 펌웨어, 또는 하드웨어에서 - 이 문서에 개시된 구조들 및 그들의 구조적 등가물들을 포함하여 -, 또는 이들 중 하나 이상의 조합으로 구현될 수 있다. 개시된 그리고 다른 실시예들은 하나 이상의 컴퓨터 프로그램 제품, 즉, 데이터 처리 장치에 의한 실행을 위해, 또는 데이터 처리 장치의 동작을 제어하기 위해 컴퓨터 관독가능 매체 상에 인코딩된 컴퓨터 프로그램 명령어들의 하나 이상의 모듈로서 구현될 수 있다. 컴퓨터 관독가능 매체는 머신 관독가능 저장 디바이스, 머신 관독가능 저장 기판, 메모리 디바이스, 머신 관독가능 전파 신호에 영향을 주는 물질 조성, 또는 이들 중 하나 이상의 조합일 수 있다. "데이터 처리 장치"라는 용어는 예로서 프로그램 가능 프로세서, 컴퓨터, 또는 다수의 프로세서 또는 컴퓨터를 포함하는, 데이터를 처리하기 위한 모든 장치, 디바이스, 및 머신을 포함한다. 장치는, 하드웨어에 더하여, 해당 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드, 예를 들어, 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 체제, 또는 이들 중 하나 이상의 조합을 구성하는 코드를 포함할 수 있다. 전파 신호는 적합한 수신기 장치로의 송신을 위해 정보를 인코딩하도록 생성되는 인위적으로 생성된 신호, 예를 들어, 머신 생성 전기, 광학, 또는 전자기 신호이다.

[0902] 컴퓨터 프로그램(프로그램, 소프트웨어, 소프트웨어 애플리케이션, 스크립트, 또는 코드라고도 알려져 있음)은, 컴파일된 또는 해석된 언어들을 포함하는 임의의 형식의 프로그래밍 언어로 작성될 수 있고, 그것은 독립형 프로그램(stand-alone program)으로서, 또는 모듈, 컴포넌트, 서브루틴, 또는 컴퓨팅 환경에서 사용하기에 적합한 다른 유닛으로서 배치되는 것을 포함하여, 임의의 형식으로 배치될 수 있다. 컴퓨터 프로그램은 파일 시스템 내의 파일에 반드시 대응하지는 않는다. 프로그램은 다른 프로그램들 또는 데이터(예를 들어, 마크업 언어 문서에 저장된 하나 이상의 스크립트)를 보유하는 파일의 일부 내에, 해당 프로그램에 전용되는 단일 파일 내에, 또는 다수의 코디네이셔닝된 파일들(예를 들어, 하나 이상의 모듈, 서브 프로그램들, 또는 코드의 부분들을 저장하는 파일들) 내에 저장될 수 있다. 컴퓨터 프로그램은 하나의 컴퓨터 상에서, 또는 한 장소에 위치하거나 다수의 장소에 걸쳐 분산되어 통신 네트워크에 의해 상호연결되는 다수의 컴퓨터 상에서 실행되도록 배치될 수 있다.

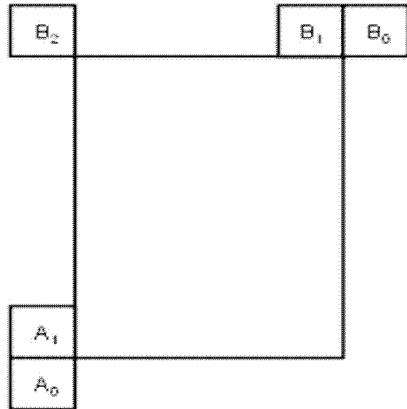
- [0903] 본 문서에서 설명된 프로세스들 및 로직 회로들은 입력 데이터에 대해 동작하고 출력을 생성하는 것에 의해 기능들을 수행하기 위해 하나 이상의 컴퓨터 프로그램을 실행하는 하나 이상의 프로그램 가능 프로세서에 의해 수행될 수 있다. 프로세스들 및 로직 회로들은 또한 특수 목적 로직 회로, 예를 들어, FPGA(field programmable gate array) 또는 ASIC(application specific integrated circuit)에 의해 수행될 수 있고, 장치가 또한 그것들로 구현될 수 있다.
- [0904] 컴퓨터 프로그램의 실행에 적합한 프로세서들은, 예로서, 범용 및 특수 목적 마이크로프로세서 둘 다, 및 임의의 종류의 디지털 컴퓨터의 임의의 하나 이상의 프로세서를 포함한다. 일반적으로, 프로세서는 판독 전용 메모리 또는 랜덤 액세스 메모리 또는 둘 다로부터 명령어들 및 데이터를 수신할 것이다. 컴퓨터의 필수 요소들은 명령어들을 수행하기 위한 프로세서 및 명령어들 및 데이터를 저장하기 위한 하나 이상의 메모리 디바이스이다. 일반적으로, 컴퓨터는 또한 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스, 예를 들어, 자기, 광자기 디스크들, 또는 광 디스크들을 포함하거나, 또는 이들로부터 데이터를 수신하거나 데이터를 이들에 전송하거나, 또는 이들 둘 다를 수행하도록 동작적으로 결합될 것이다. 그러나, 컴퓨터는 그러한 디바이스들을 가질 필요는 없다. 컴퓨터 프로그램 명령어들 및 데이터를 저장하기에 적합한 컴퓨터 판독가능 매체는, 예로서 반도체 메모리 디바이스들, 예를 들어, EPROM, EEPROM, 및 플래시 메모리 디바이스들; 자기 디스크들, 예를 들어, 내부 하드 디스크들 또는 이동식 디스크들; 광자기 디스크들; 및 CD ROM 및 DVD-ROM 디스크들을 포함하는 모든 형식의 비-휘발성 메모리, 매체 및 메모리 디바이스들을 포함한다. 프로세서 및 메모리는 특수 목적 로직 회로에 의해 보완되거나 그에 통합될 수 있다.
- [0905] 본 특허 문서가 많은 세부 사항들을 포함하고 있지만, 이들이 임의의 주제의 또는 청구될 수 있는 것의 범위에 대한 제한으로서 해석되어서는 안 되고, 오히려 특정 기법들의 특정 실시예들에 특정적일 수 있는 특징들의 설명으로서 해석되어야 한다. 개별 실시예들의 컨텍스트에서 본 특허 문서에서 설명되는 특정 특징들은 또한 단일 실시예에서 조합하여 구현될 수 있다. 반대로, 단일 실시예의 컨텍스트에서 설명되는 다양한 특징들은 또한 다수의 실시예에서 개별적으로 또는 임의의 적합한 부분조합으로 구현될 수 있다. 더욱이, 특정들이 특정 조합들로 작용하는 것으로 위에서 설명되고 심지어 처음에는 그와 같이 청구될 수 있지만, 청구된 조합으로부터의 하나 이상의 특징은 일부 경우들에서 조합으로부터 삭제될 수 있고, 청구된 조합은 부분조합 또는 부분조합의 변형에 관한 것일 수 있다.
- [0906] 유사하게, 동작들이 특정 순서로 도면들에 묘사되지만, 이는 바람직한 결과들을 달성하기 위해, 그러한 동작들이 도시된 특정 순서로 또는 순차적인 순서로 수행되거나, 모든 예시된 동작들이 수행되는 것을 요구하는 것으로 이해되어서는 안 된다. 더욱이, 본 특허 문서에서 설명된 실시예들에서의 다양한 시스템 컴포넌트들의 분리는 모든 실시예들에서 그러한 분리를 요구하는 것으로 이해되어서는 안 된다.
- [0907] 몇몇 구현들 및 예들만이 설명되고, 본 특허 문서에서 설명되고 예시된 것에 기초하여 다른 구현들, 향상들 및 변형들이 이루어질 수 있다.

도면

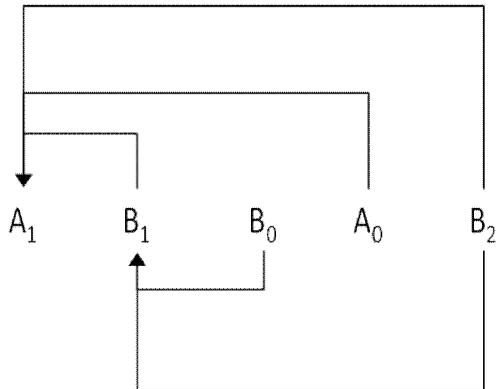
도면1



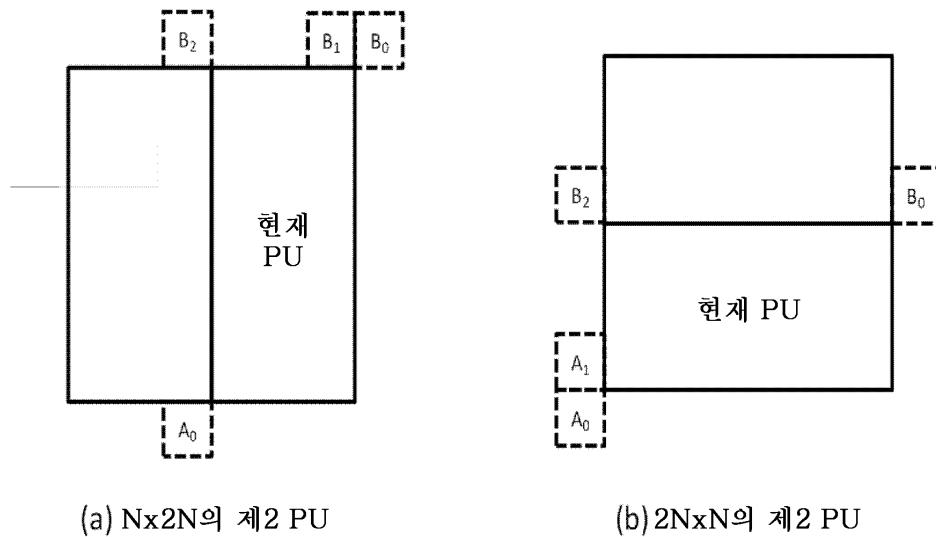
도면2



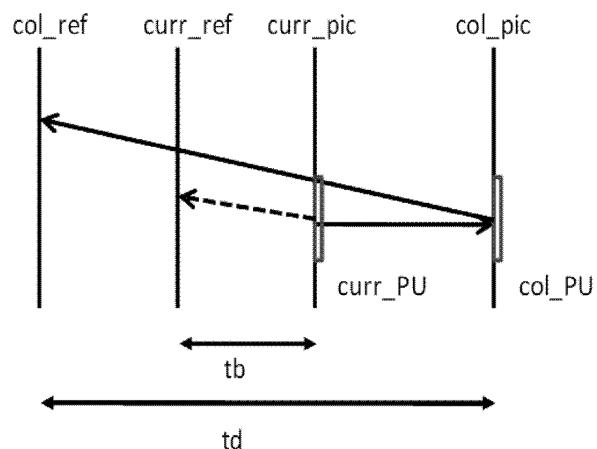
도면3



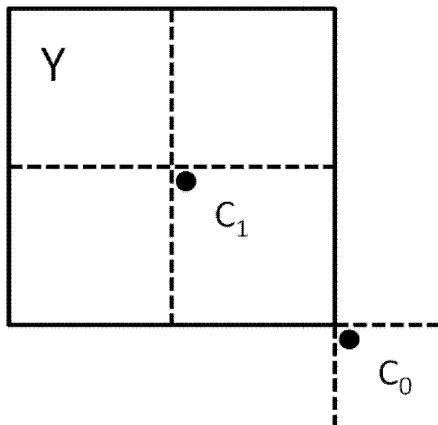
도면4



도면5



도면6



도면7

원래 병합 후보 리스트

Merge_idx	L0	L1
0	mvL0_A, ref0	-
1	-	mvL1_B, ref0
2		
3		
4		

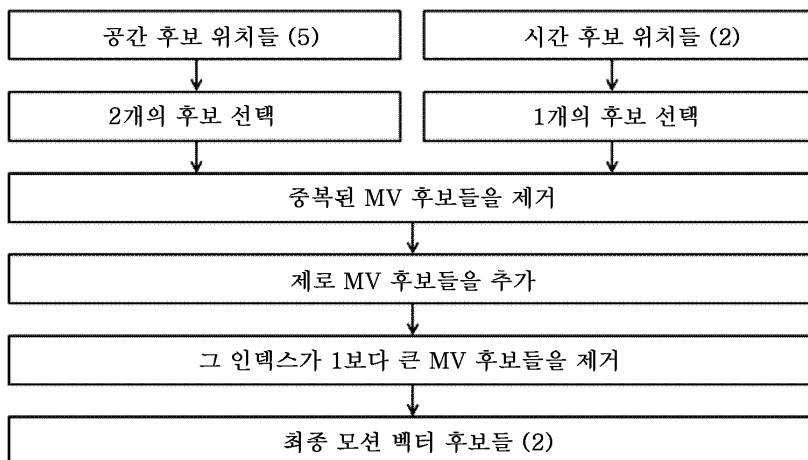
조합된 후보들을 추가한 후의 병합 후보 리스트

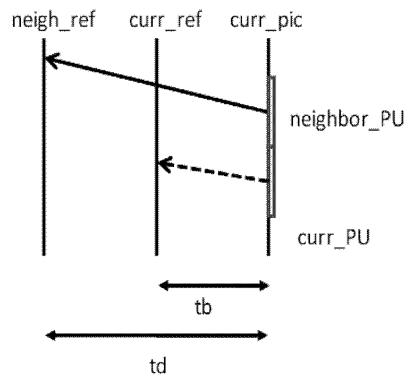
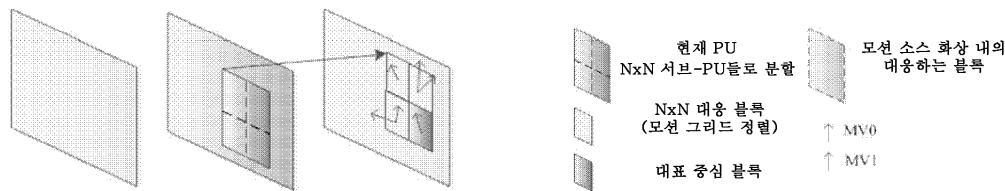
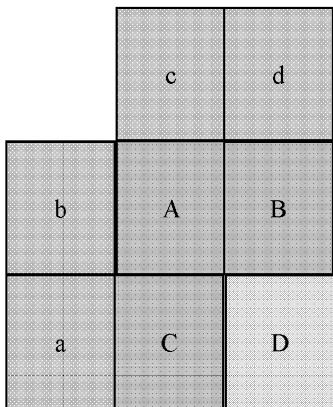
Merge_idx	L0	L1
0	mvL0_A, ref0	조합
1		mvL1_B, ref0
2	mvL0_A, ref0	mvL1_B, ref0
3		
4		

조합

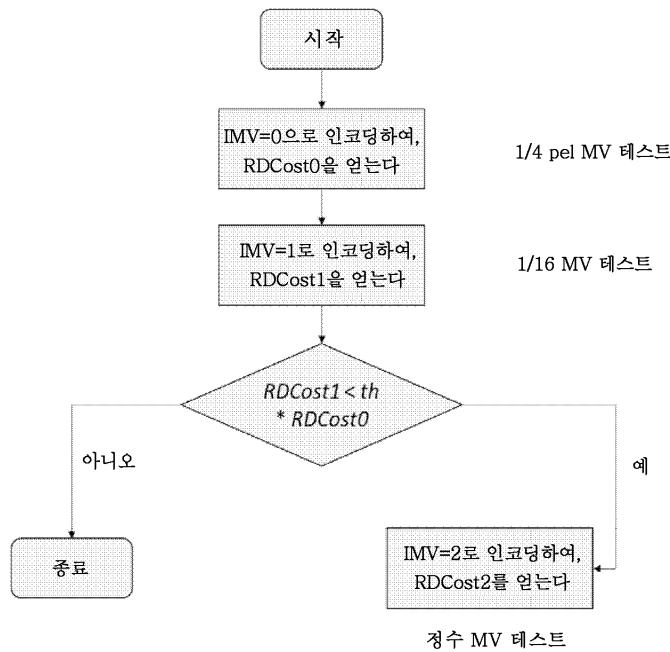
도면8

입력으로서 refidx를 갖는 각각의 참조 화상 리스트에 대해

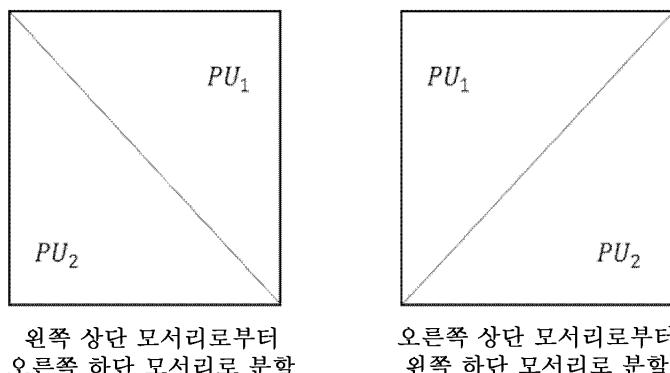


도면9**도면10****도면11**

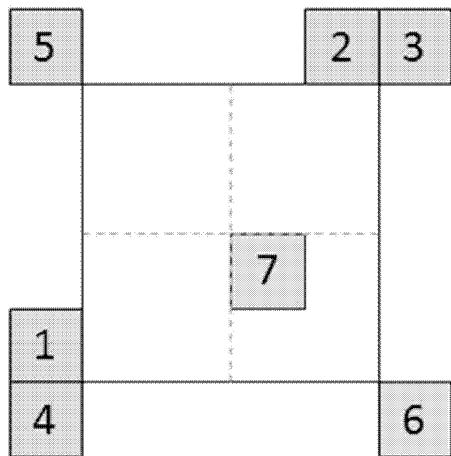
도면12



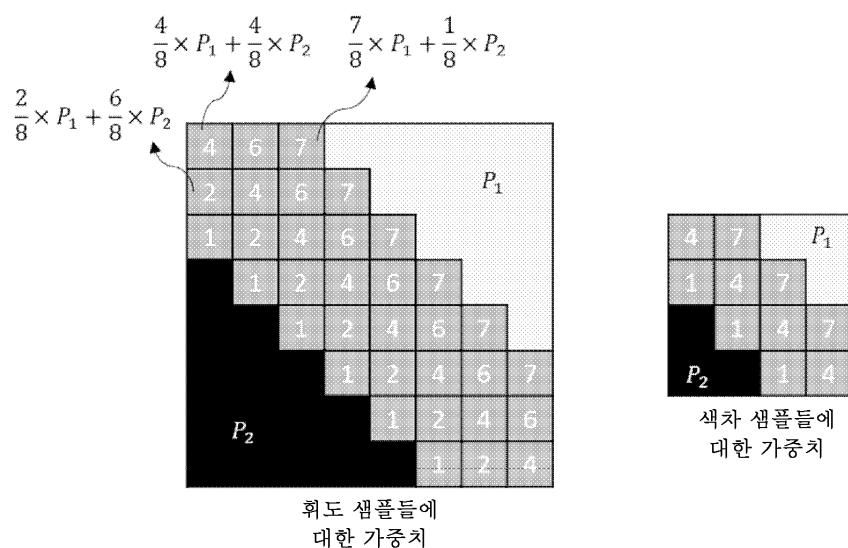
도면13



도면14



도면15

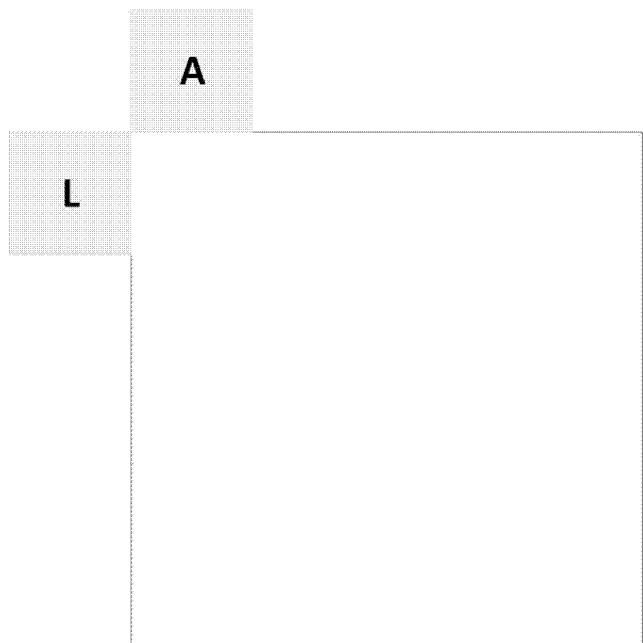


도면 16

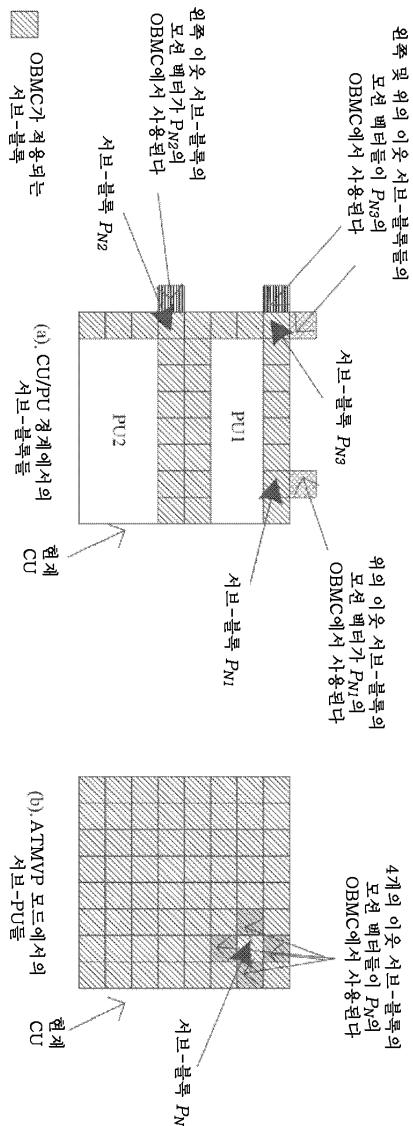
				4
	bi	Mv1	Mv1	Mv1
Mv2		bi	Mv1	Mv1
Mv2	Mv2		bi	Mv1
Mv2	Mv2	Mv2		bi

				4
	Mv1	Mv1	Mv1	bi
Mv1		Mv1		bi
Mv1		bi	Mv2	Mv2
bi		Mv2	Mv2	Mv2

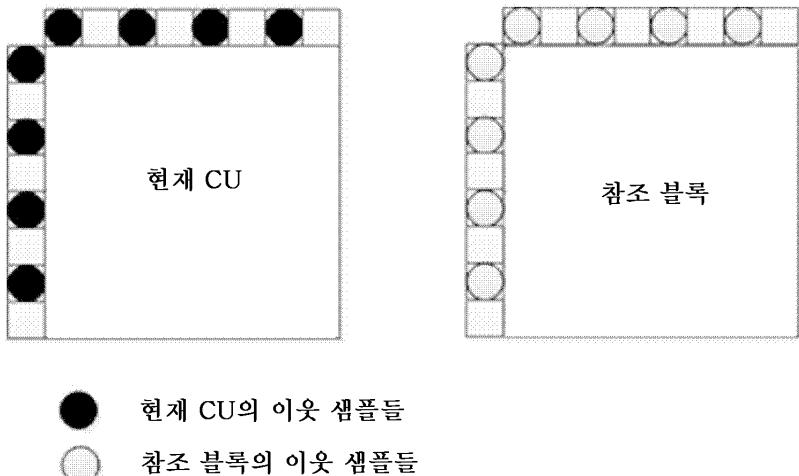
도면17



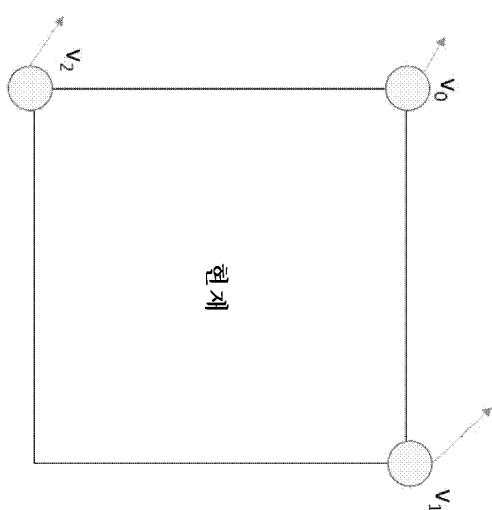
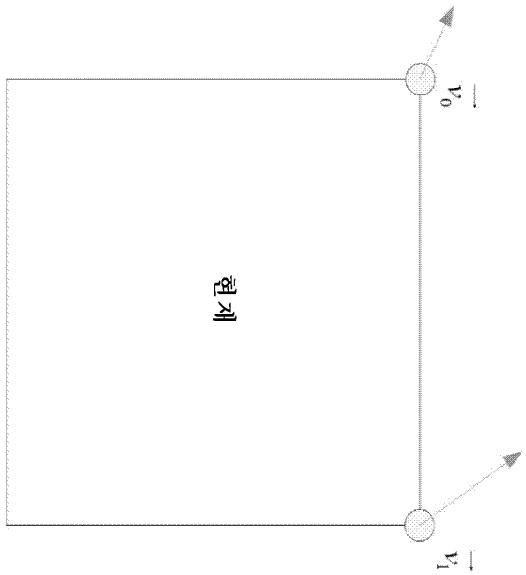
도면18



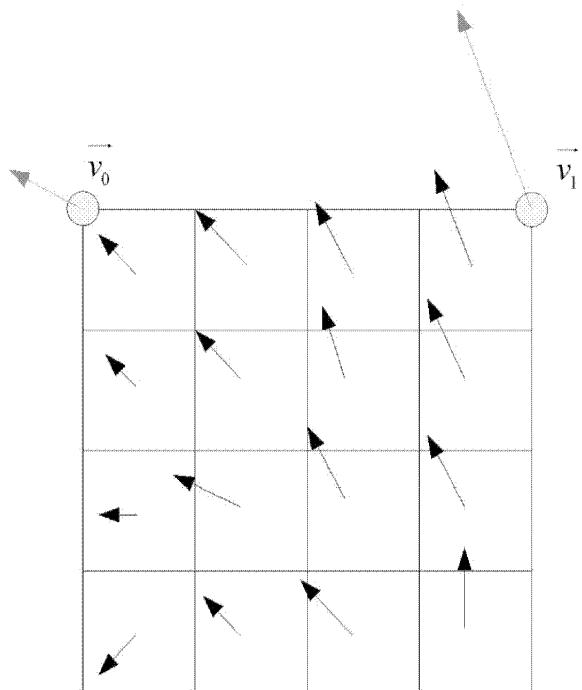
도면19



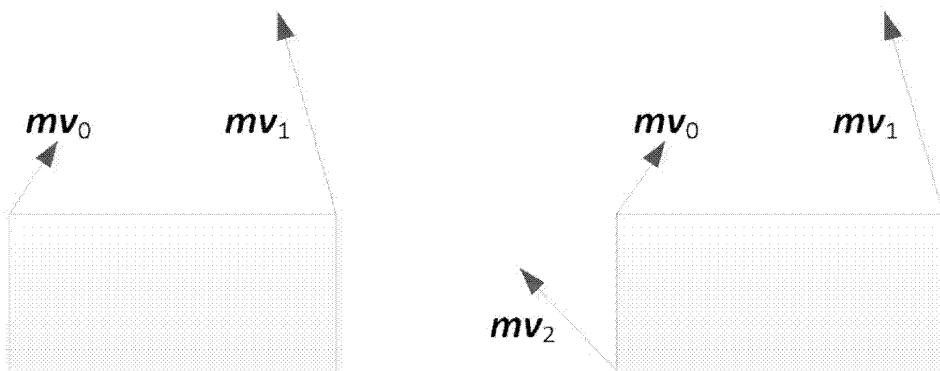
도면20



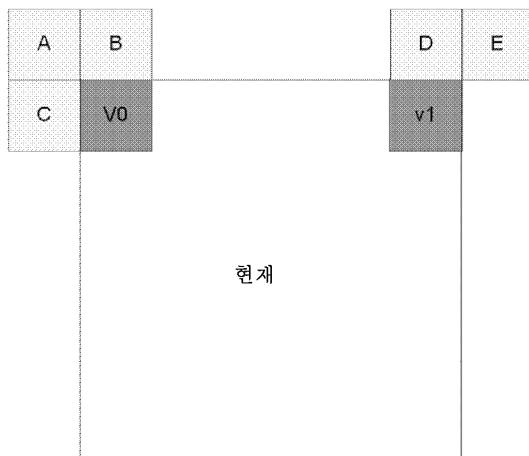
도면21



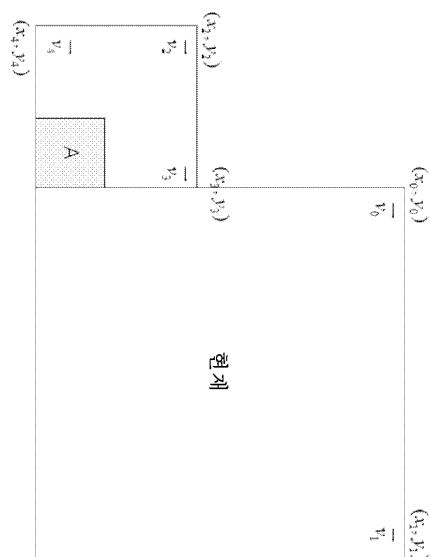
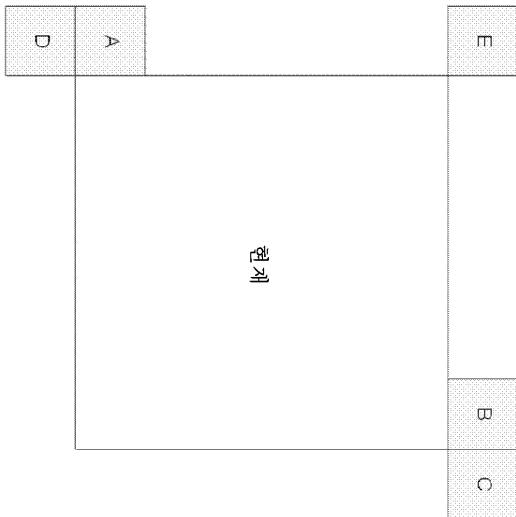
도면22



도면23



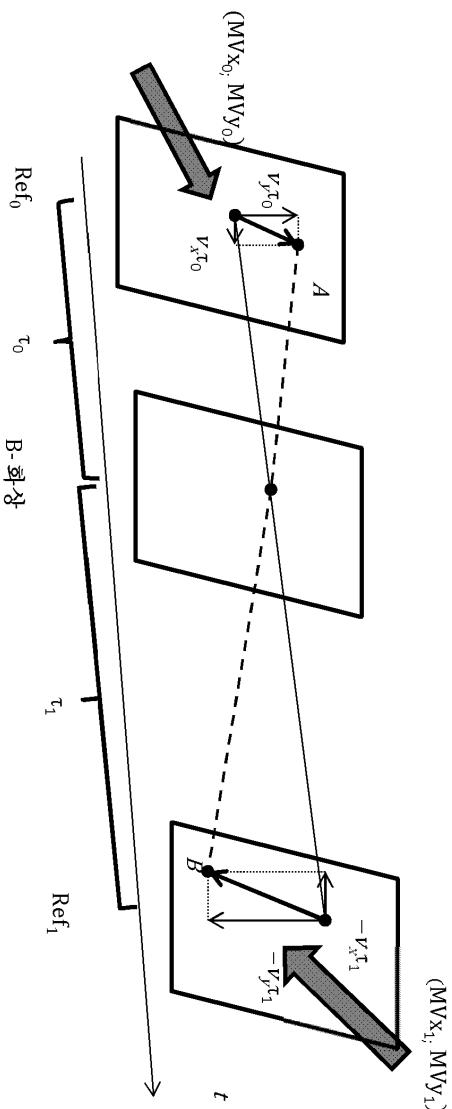
도면24



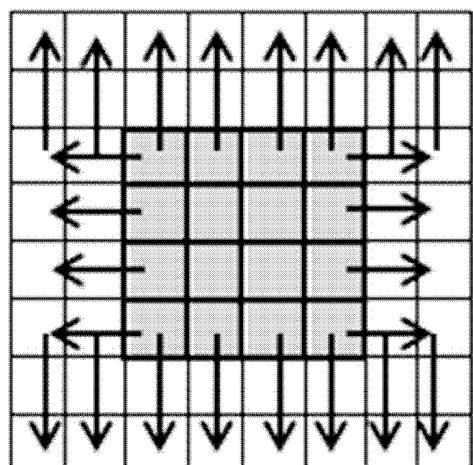
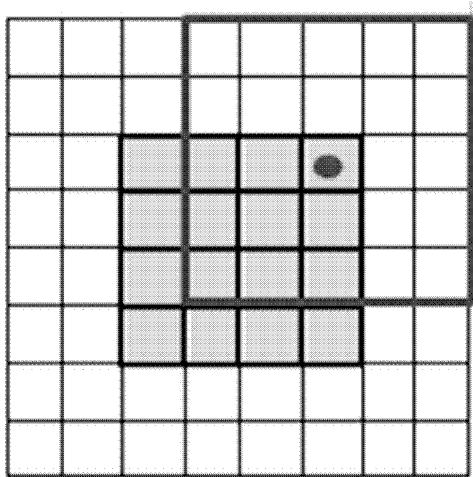
도면25



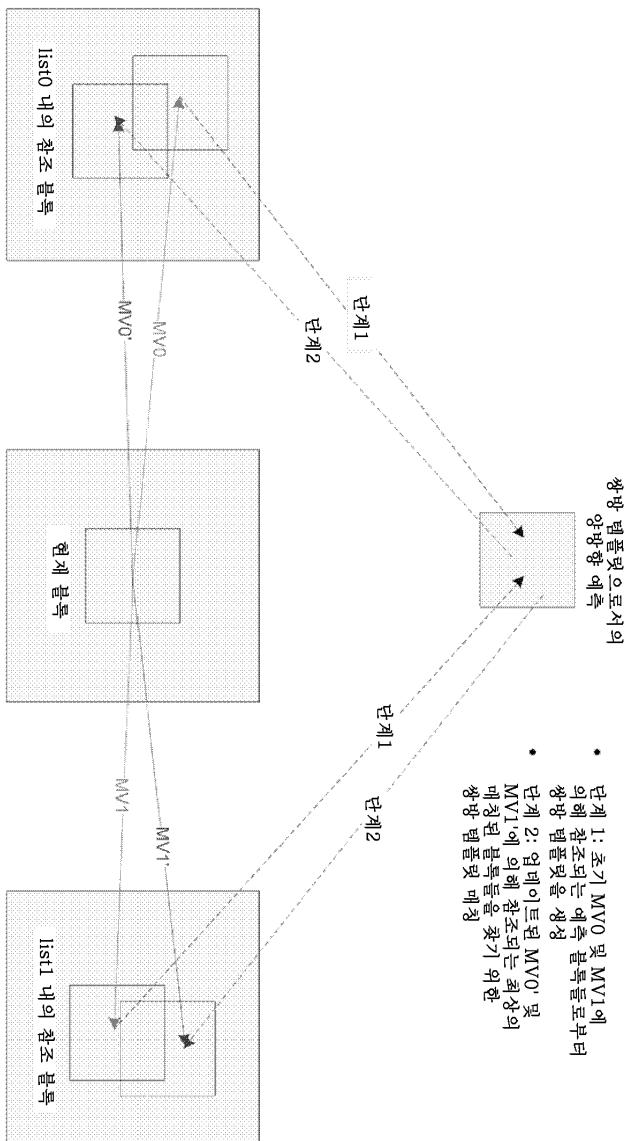
도면26



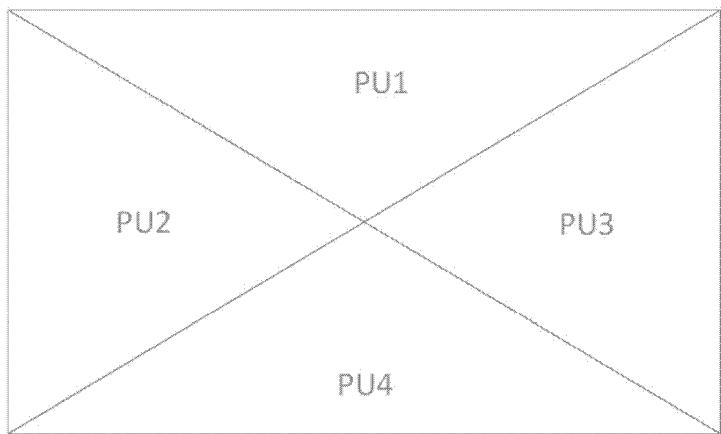
도면27



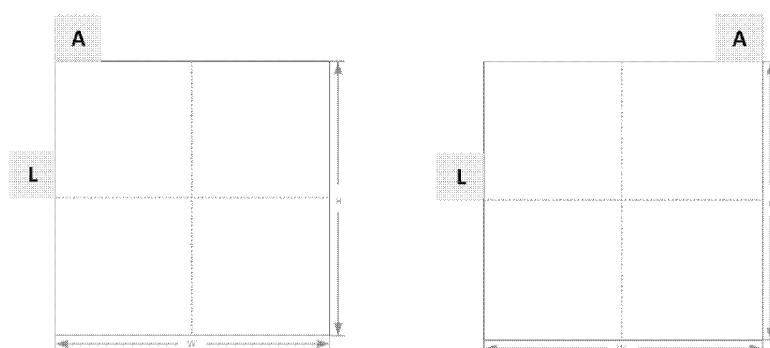
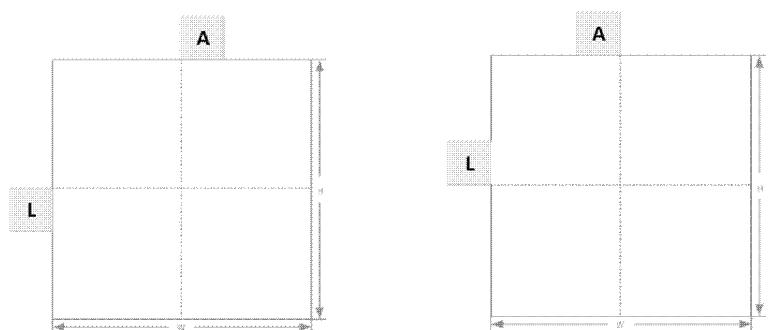
도면28



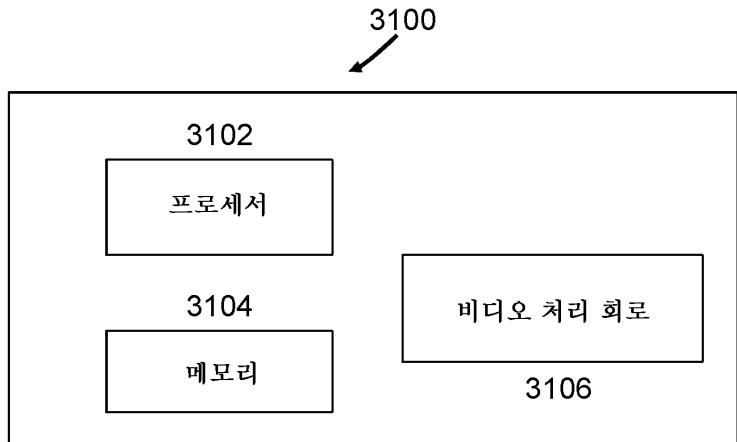
도면29



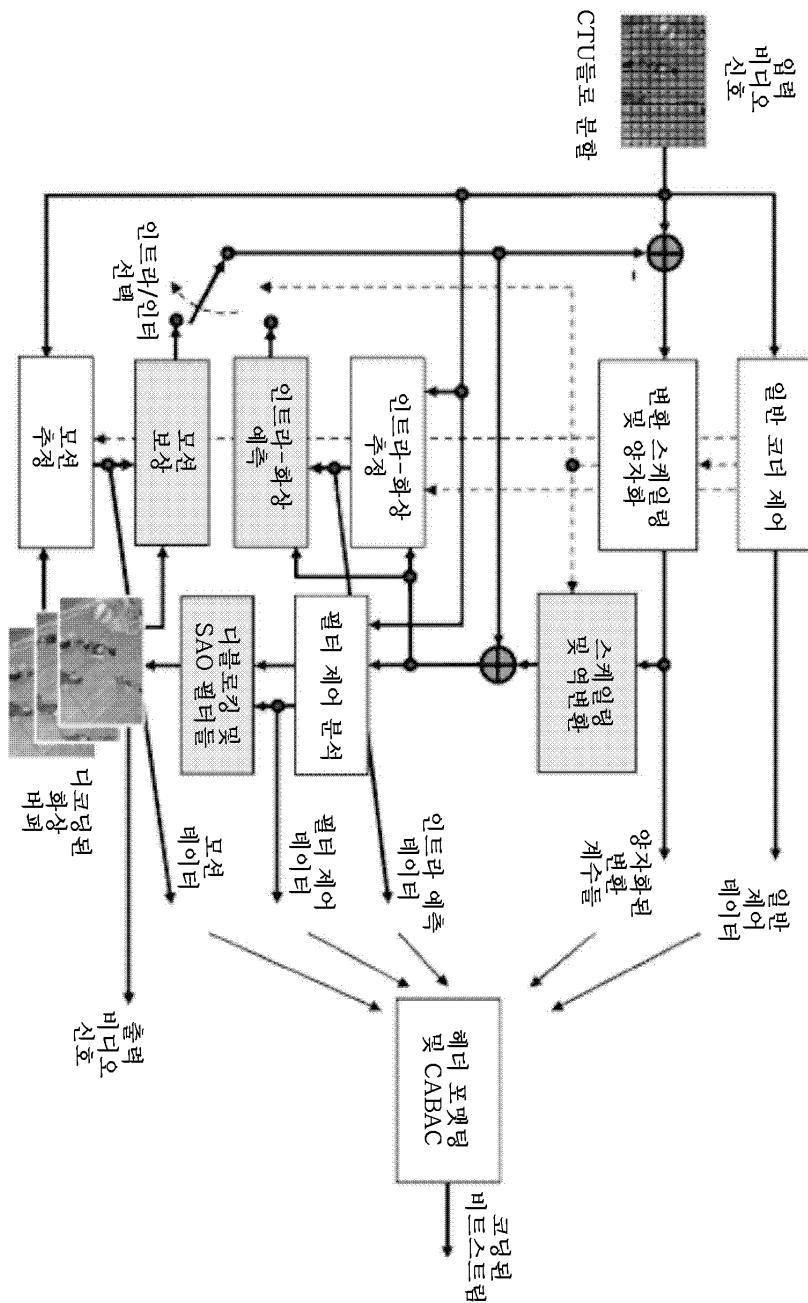
도면30



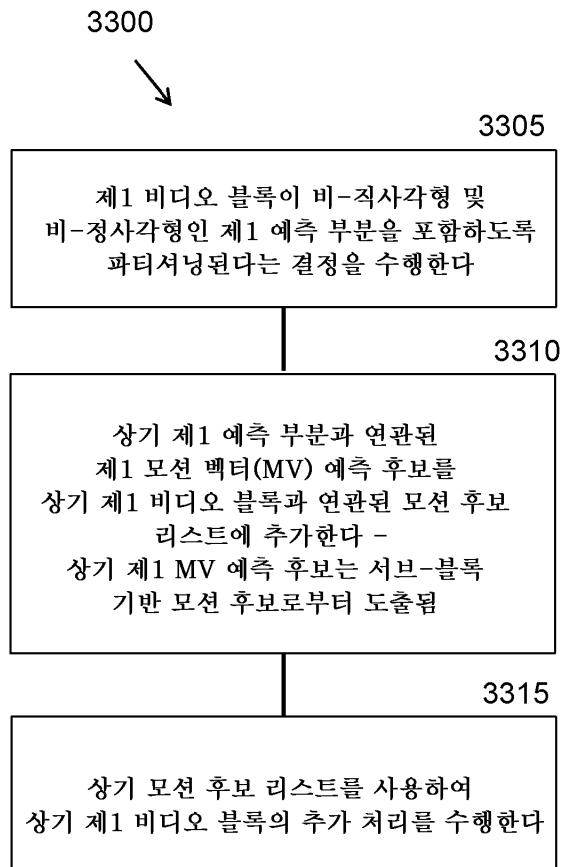
도면31



도면32



도면33



도면34

3400



3405

제1 분할 패턴에 기초한 예측 부분들을
포함하도록 제1 비디오 블록을
파티셔닝한다 - 상기 예측 부분들은
제1 병합 인덱스 및 제1 코딩된 값과 연관됨

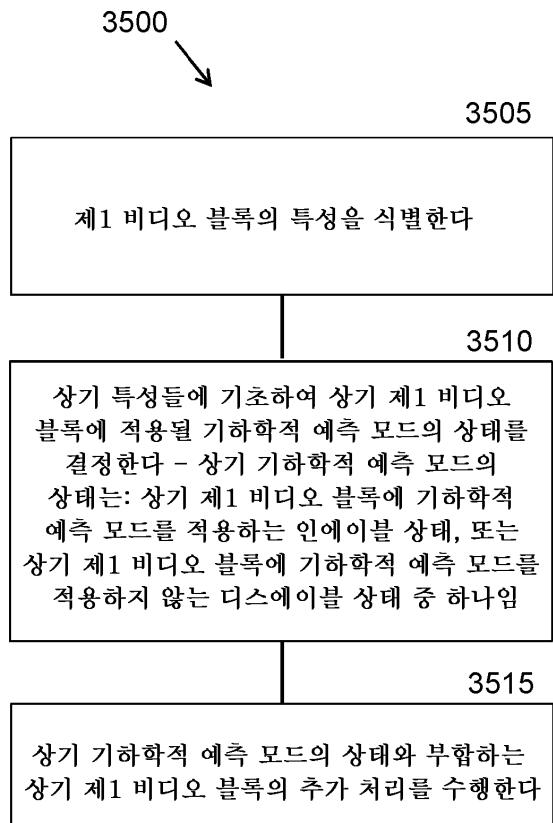
3410

제2 분할 패턴에 기초한 예측 부분을
포함하도록 제2 비디오 블록을 파티셔닝한다
- 상기 제2 비디오 블록의 예측 부분들은
제2 병합 인덱스 및 제2 코딩된 값과 연관되고,
상기 제1 코딩된 값은 상기 제2 코딩된 값과
동등하고, 상기 제2 분할 패턴 또는 상기
제2 비디오 블록의 제2 병합 인덱스 중 하나
이상은 상기 제2 비디오 블록이 상기 제1
비디오 블록과 상이한 시퀀스, 화상, 타일,
슬라이스, 코딩 트리 단위(CTU), 또는
블록 내에 있는 것에 기초하여
상기 제1 비디오 블록과 상이함

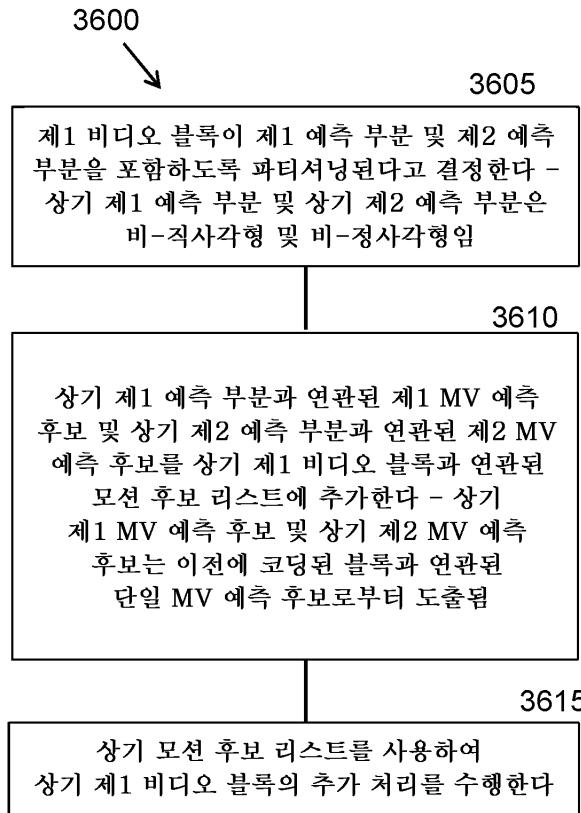
3415

상기 제1 비디오 블록 및 상기
제2 비디오 블록의 추가 처리를 수행한다

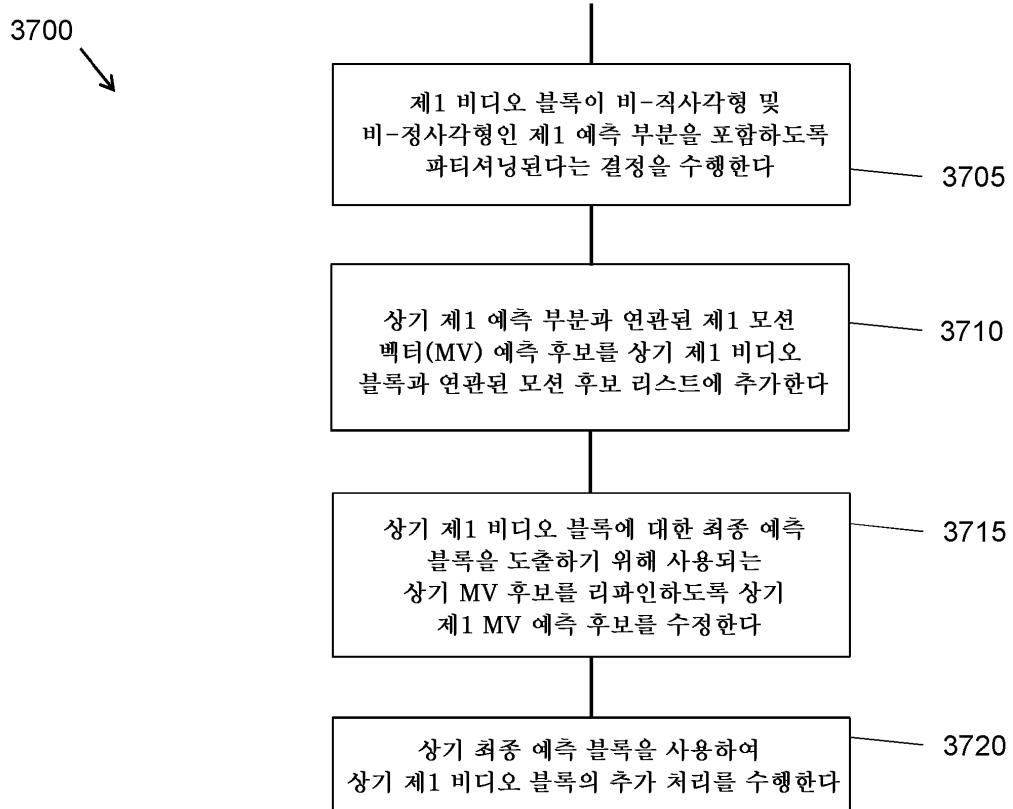
도면35



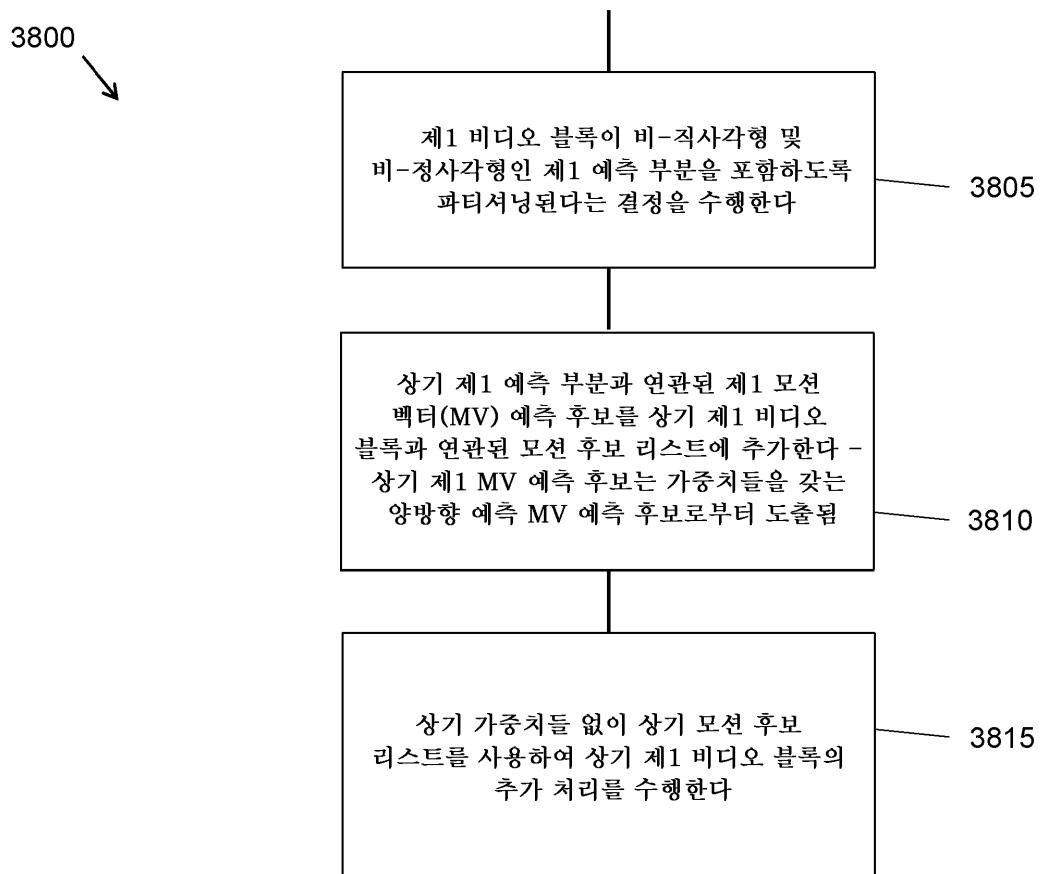
도면36



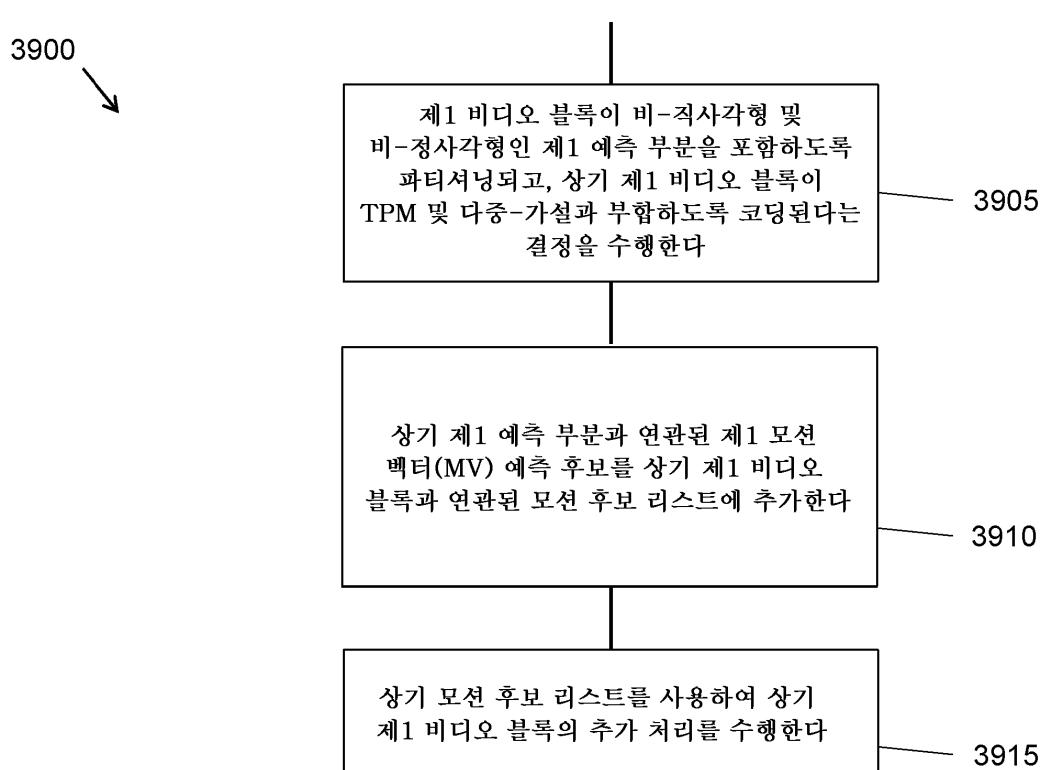
도면37



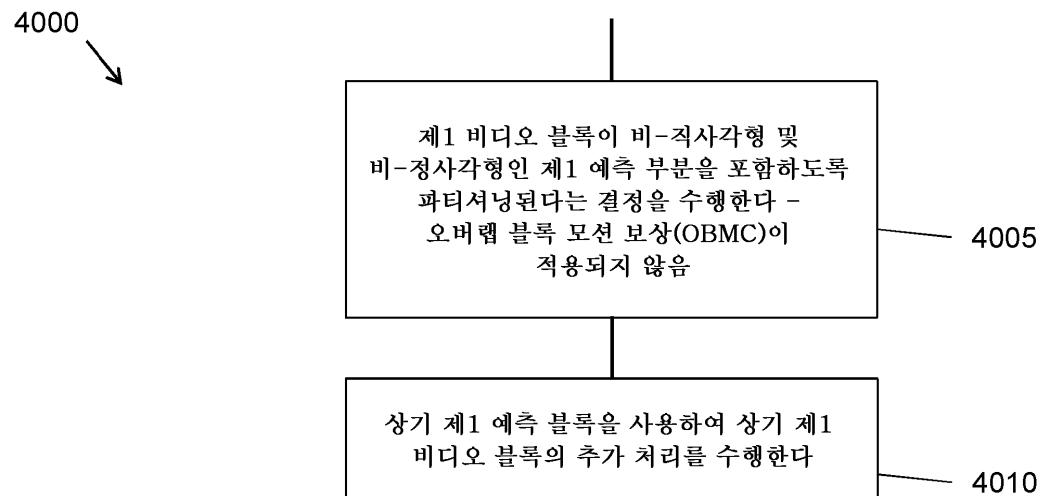
도면38



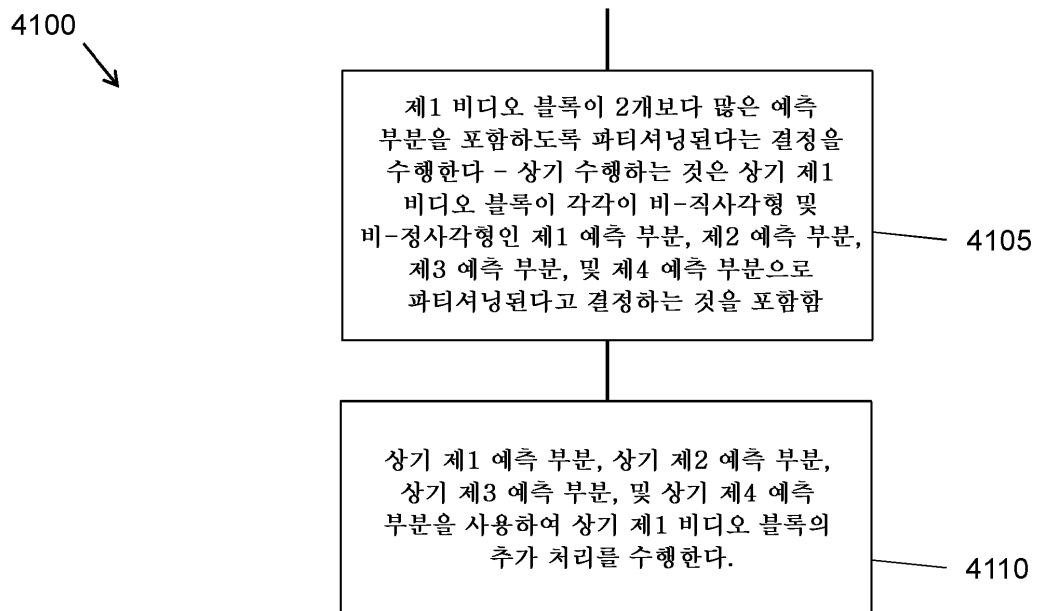
도면39



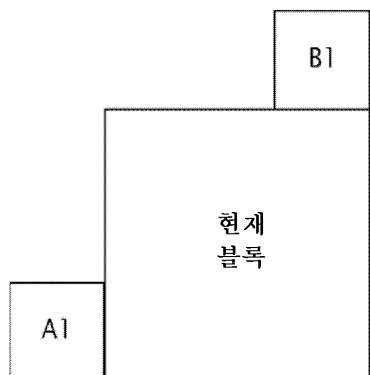
도면40



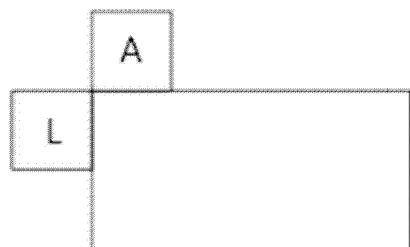
도면41



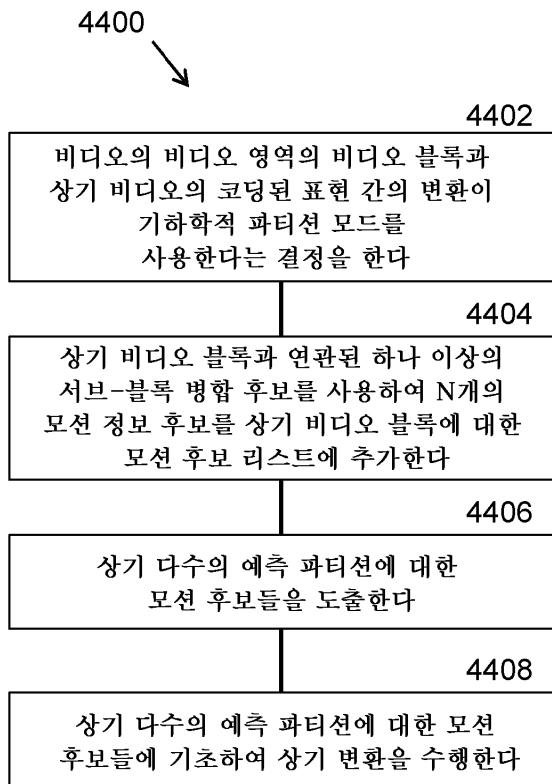
도면42



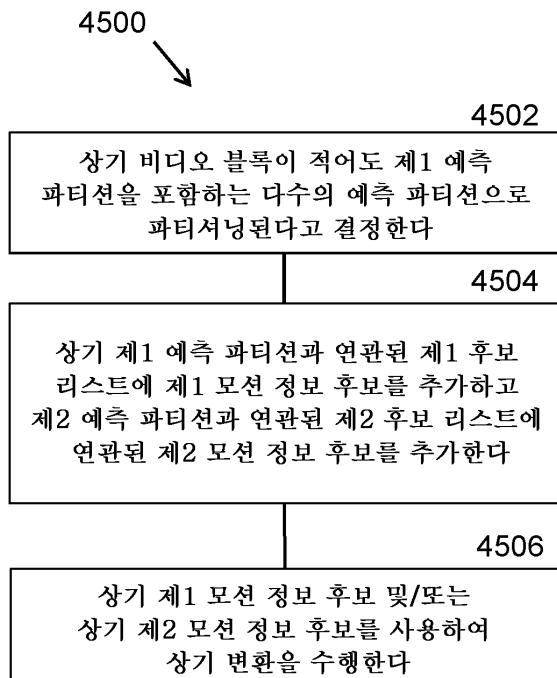
도면43



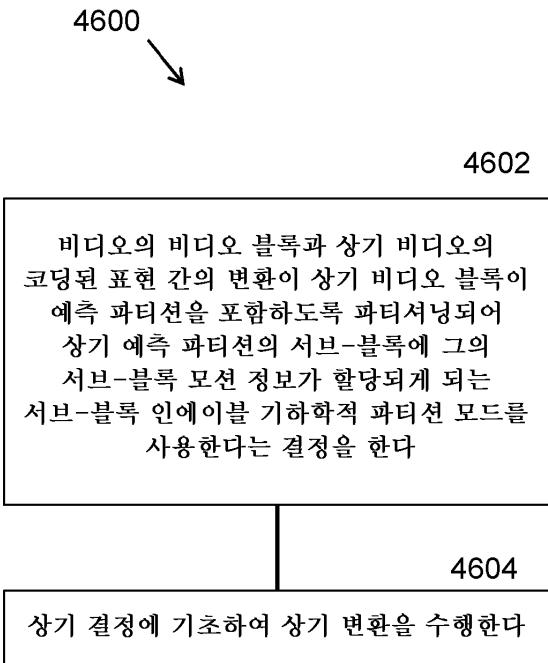
도면44



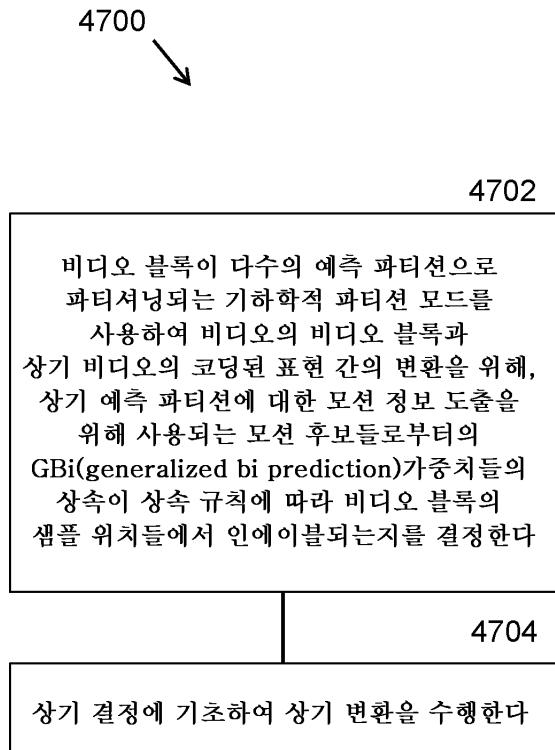
도면45



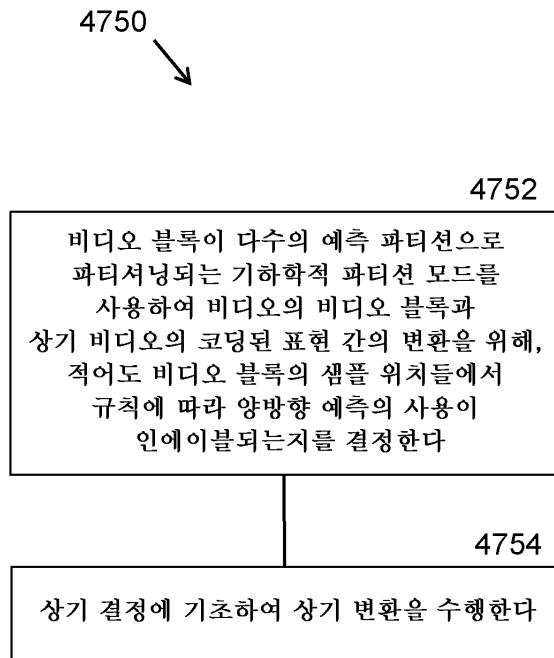
도면46



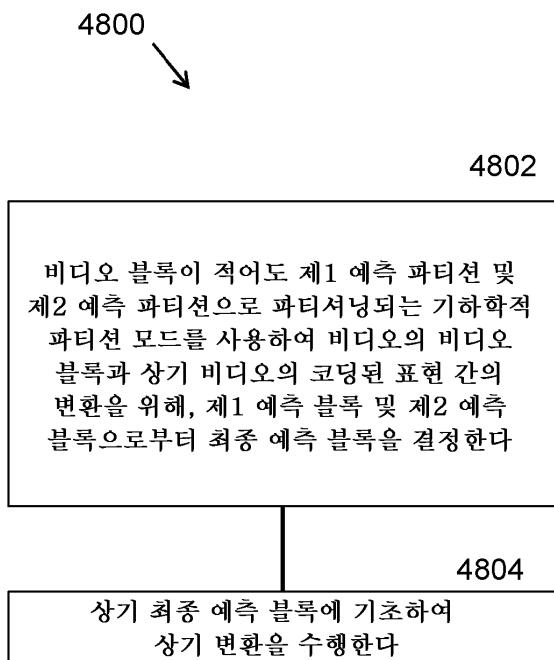
도면47a



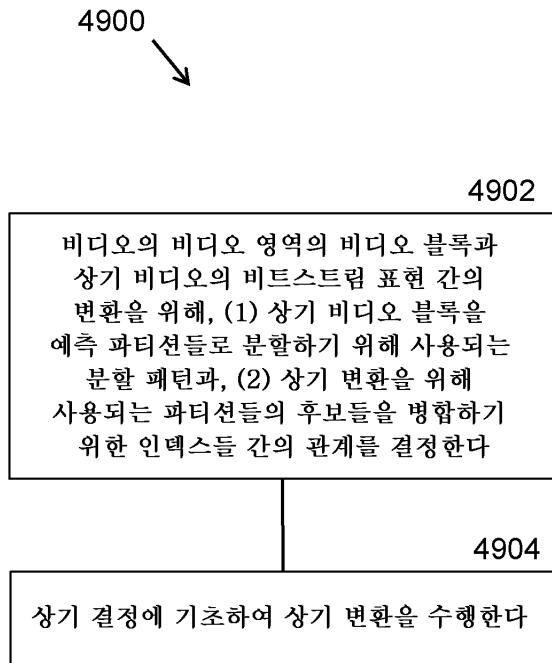
도면47b



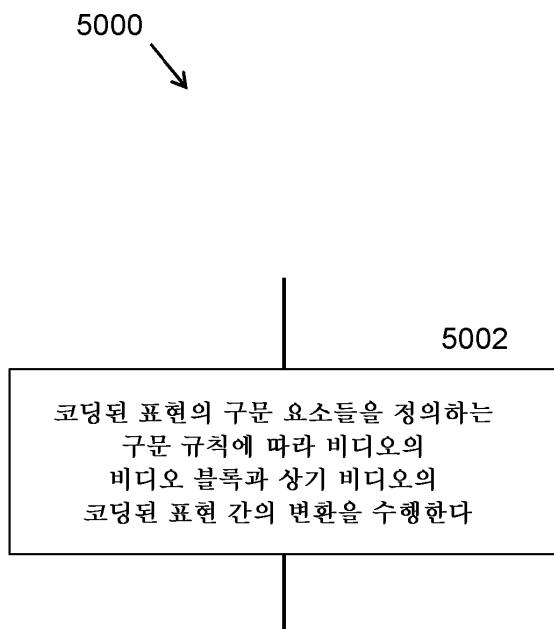
도면48



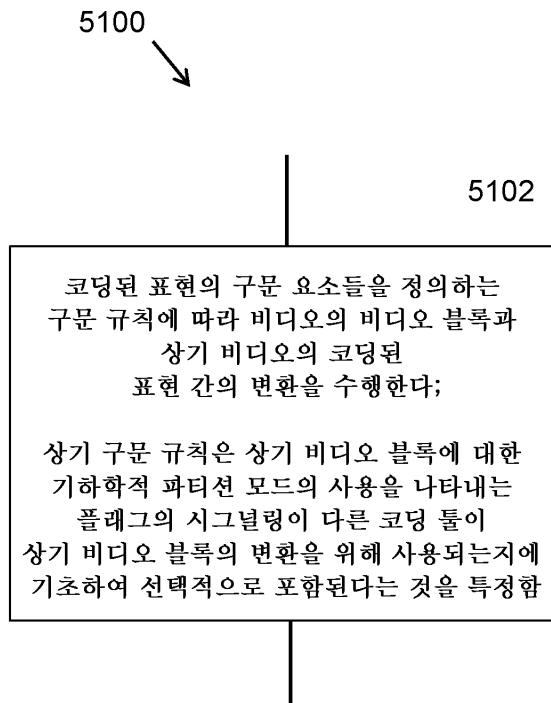
도면49



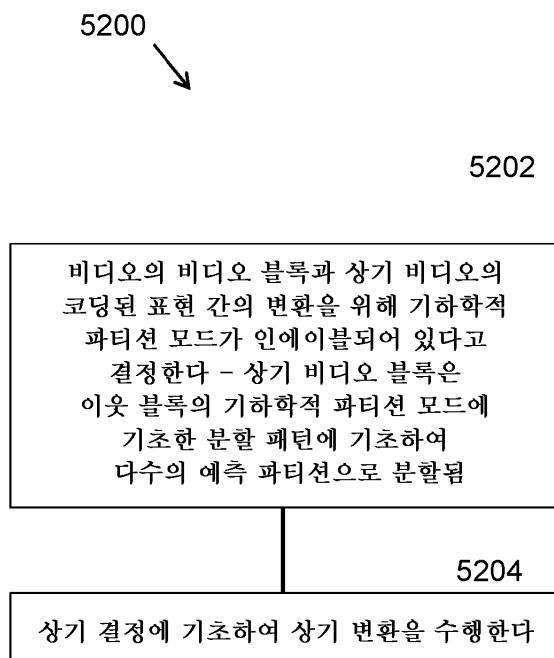
도면50



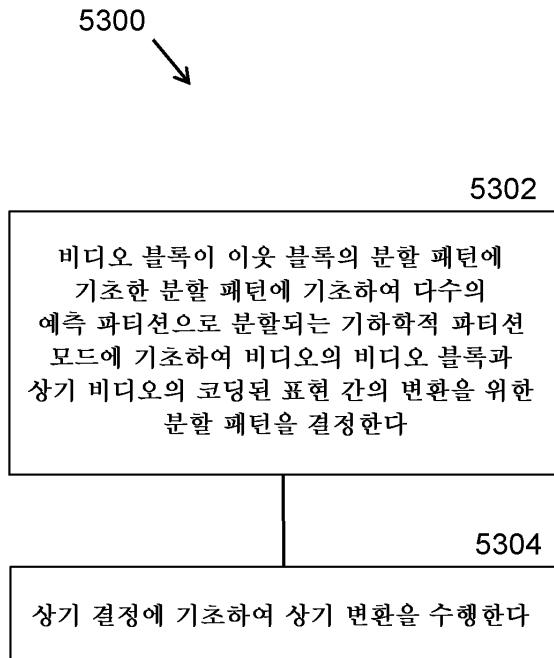
도면51



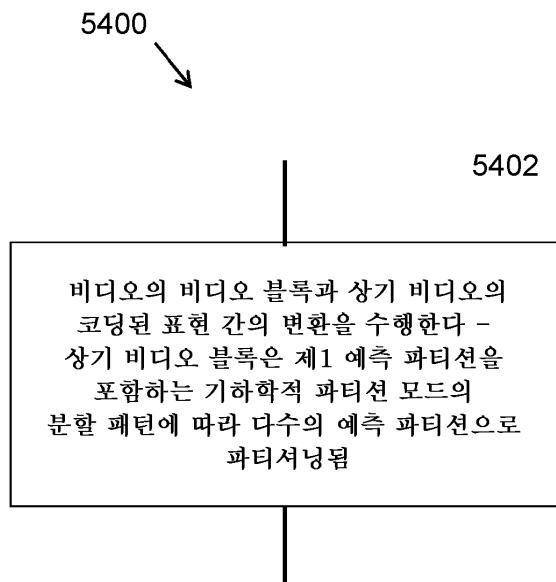
도면52



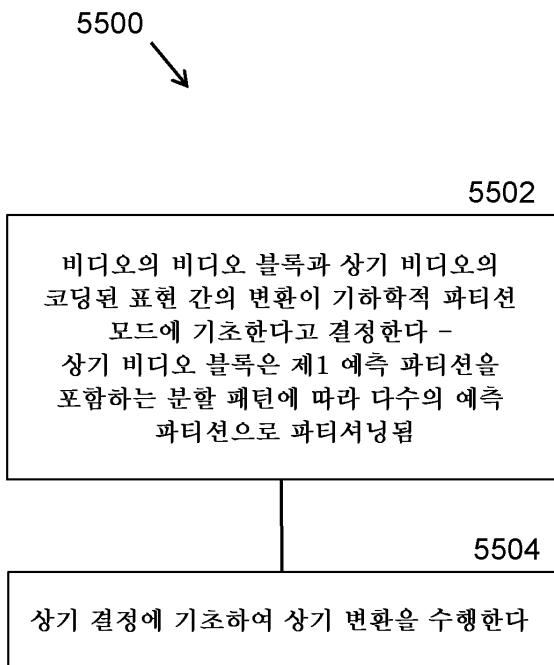
도면53



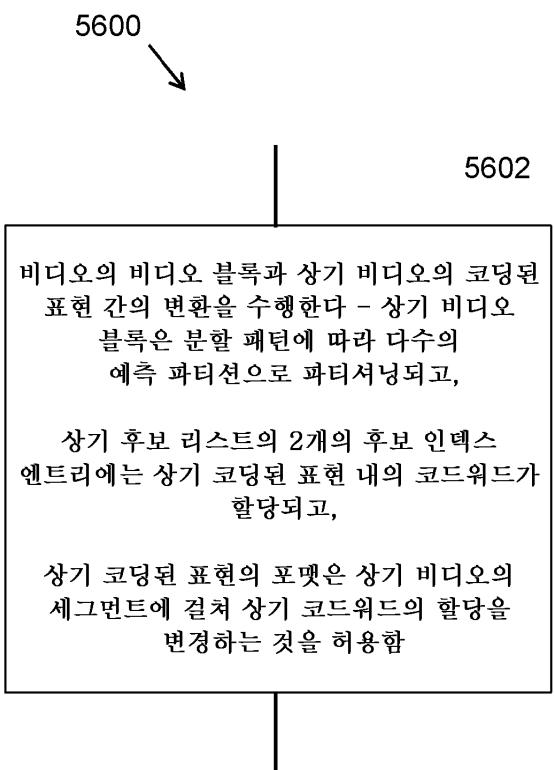
도면54

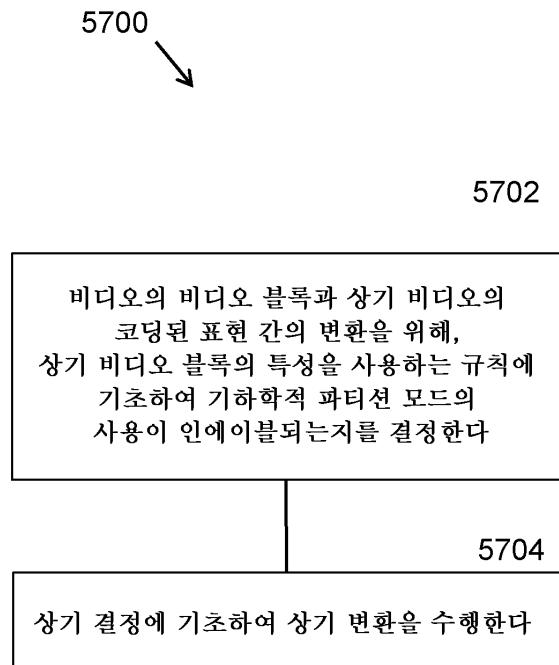
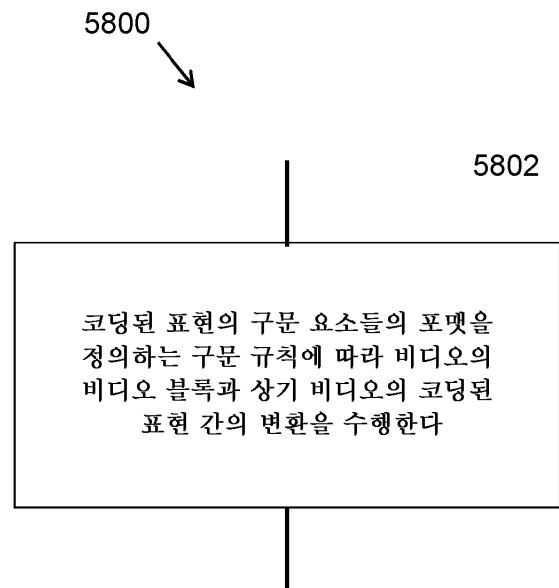


도면55

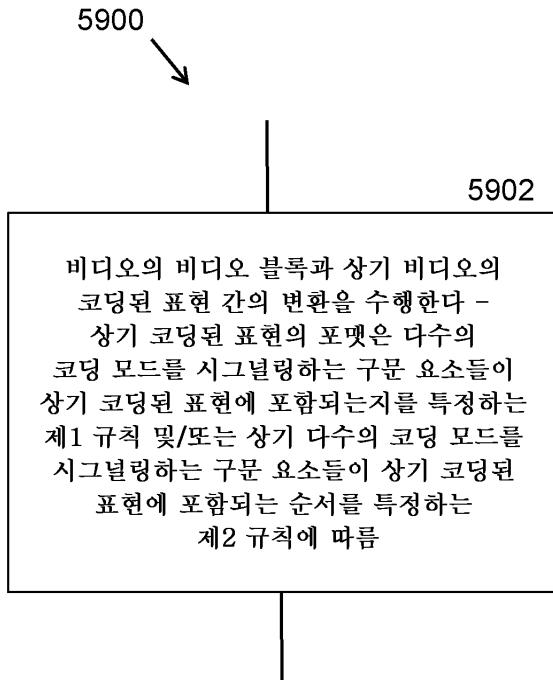


도면56

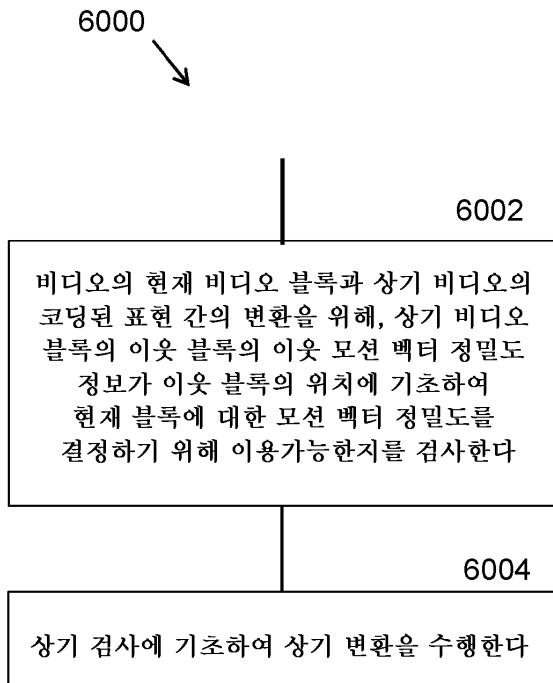


도면57**도면58**

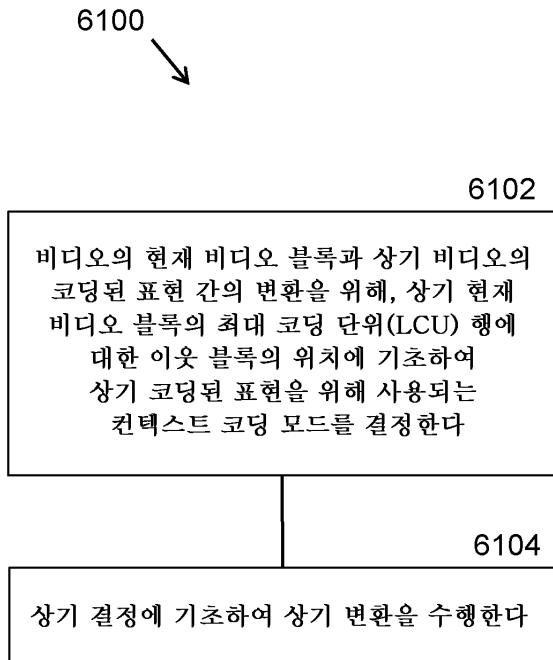
도면59



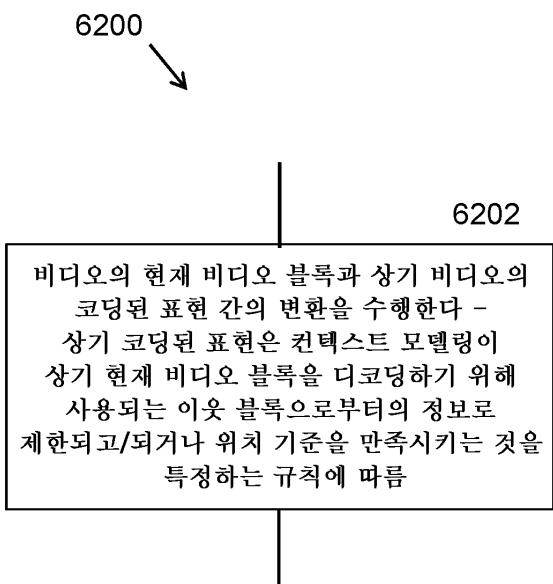
도면60



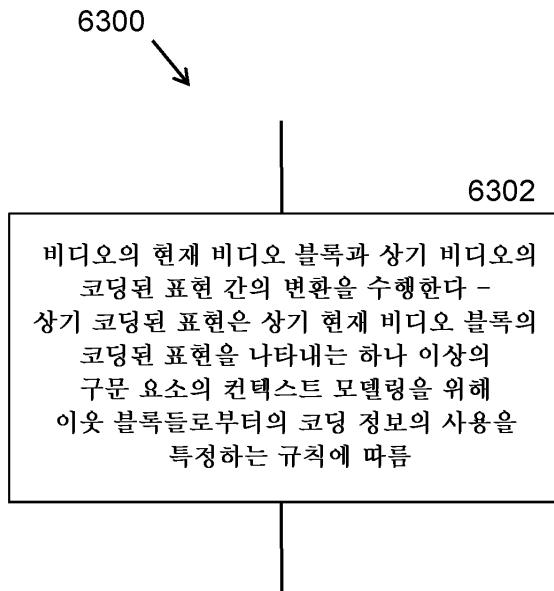
도면61



도면62



도면63



도면64

