(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0011199
(43) 공개일자 2024년01월25일

(51) 국제특허분류(Int. Cl.)
H04N 19/52 (2014.01) *H04N 19/105* (2014.01)
H04N 19/119 (2014.01) *H04N 19/139* (2014.01)
H04N 19/176 (2014.01) *H04N 19/573* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/52 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2023-7045033
(22) 출원일자(국제) 2022년06월28일
심사청구일자 2023년12월27일
(85) 번역문제출일자 2023년12월27일
(86) 국제출원번호 PCT/US2022/035375
(87) 국제공개번호 WO 2023/278489
국제공개일자 2023년01월05일
(30) 우선권주장
63/215,957 2021년06월28일 미국(US)

(71) 출원인
베이징 다지아 인터넷 인포메이션 테크놀로지 컴퍼니 리미티드
중국 100085 베이징 하이뎬 디스트릭트 시얼치 웨스트 로드 넘버 16 빌딩 12 플로어 8 룸 101
(72) 발명자
천 웨이
미국 92130 캘리포니아주 샌디에이고 카멜 캐년 로드 12569
궈 처-웨이
중국 100092 베이징 베이징 하이뎬 디스트릭트 상디 웨스트 로드 넘버 6 빌딩 1 플로어 1 룸 101 디1-7
(뒷면에 계속)
(74) 대리인
유미특허법인

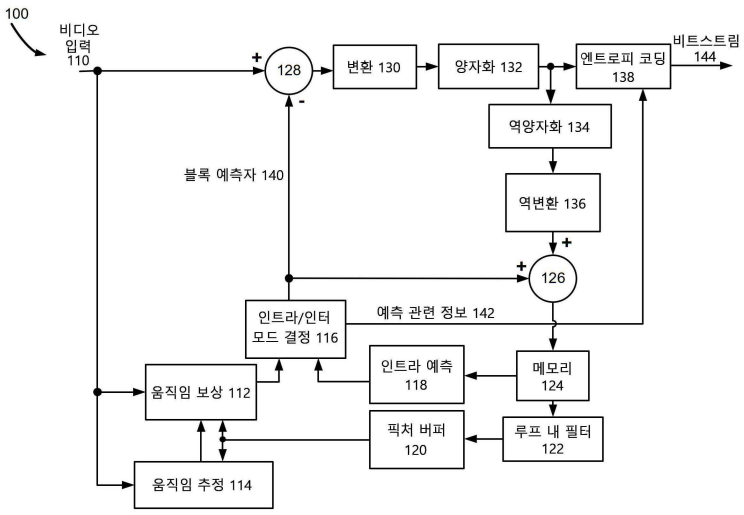
전체 청구항 수 : 총 14 항

(54) 발명의 명칭 **움직임 벡터 미세조정을 이용한 기하학적 파티션 모드를 위한 방법 및 디바이스**

(57) 요 약

GPM에서 비디오 블록을 디코딩하기 위한 방법은: 비디오 블록을 2개의 기하학적 파티션으로 파티셔닝하는 단계; 정규 병합 후보를 추가함으로써 단방향 움직임 벡터(MV) 후보 리스트를 구성하는 단계; 후보 리스트가 가득 차 있지 않다고 결정하는 것에 대한 응답으로, 정규 병합 후보 리스트의 양방향 예측 MV들로부터 도출된 추가 단방향 MV를 후보 리스트에 추가함으로써 제1 업데이트된 후보 리스트를 구성하는 단계; 제1 업데이트된 후보 리스트가 가득 차 있지 않다고 결정하는 것에 대한 응답으로, 쌍별 평균 후보를 제1 업데이트된 후보 리스트에 추가함으로써 제2 업데이트된 후보 리스트를 구성하는 단계; 제2 업데이트된 후보 리스트가 가득 차 있지 않다고 결정하는 것에 대한 응답으로, 최대 길이에 도달할 때까지 0의 단방향 MV를 제2 업데이트된 후보 리스트에 주기적으로 추가하는 단계; 및 각각의 기하학적 파티션에 대한 단방향 MV를 각각 생성하는 단계를 포함한다.

대 표 도



(52) CPC특허분류
H04N 19/119 (2015.01)
H04N 19/139 (2015.01)
H04N 19/176 (2015.01)
H04N 19/573 (2015.01)
H04N 19/70 (2015.01)

(72) 발명자
주 홍-정
중국 100092 베이징 베이징 하이디안 디스트릭트 상디 웨스트 로드 넘버 6 빌딩 1 1층 룸101디1-7
옌 닝
중국 100092 베이징 베이징 하이뎬 디스트릭트 상디 웨스트 로드 넘버 6 빌딩 1 플로어 1 룸 101디1-7
천 이-원
미국 92122 캘리포니아주 샌디에이고 스위트 400 유니버시티 센터 레인 8910
왕 샹린
미국 92122 캘리포니아주 샌디에이고 스위트 400 유니버시티 센터 레인 8910
위 빙
중국 100092 베이징 베이징 상디 웨스트 로드 넘버 6 빌딩 1 플로어 1 룸 101디1-7
슈 샤오위
미국 92130 캘리포니아주 샌디에이고 앨리소 밸리 웨이 6691

명 세 서

청구범위

청구항 1

기하학적 파티션 모드(GPM: geometry partition mode)에서 비디오 블록을 디코딩하기 위한 방법으로서,

상기 비디오 블록을 제1 기하학적 파티션 및 제2 기하학적 파티션으로 파티셔닝하는 단계;

복수의 정규 병합 후보(regular merge candidates)를 추가함으로써 상기 GPM의 단방향 움직임 벡터(MV: motion vector) 후보 리스트를 구성하는 단계;

상기 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로, 정규 병합 후보 리스트의 하나 이상의 양방향 예측 MV로부터 도출된 하나 이상의 추가 단방향 MV를 상기 단방향 MV 후보 리스트에 추가함으로써 제1 업데이트된 단방향 MV 후보 리스트를 구성하는 단계;

상기 제1 업데이트된 단방향 MV 후보 리스트가 가득 차 있지 않다고 결정하는 것에 대한 응답으로, 하나 이상의 쌍별 평균 후보(pairwise average candidate)를 상기 제1 업데이트된 단방향 MV 후보 리스트에 추가함으로써 제2 업데이트된 단방향 MV 후보 리스트를 구성하는 단계;

상기 제2 업데이트된 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로, 최대 길이에 도달할 때까지 0의 단방향 MV를 상기 제2 업데이트된 단방향 MV 후보 리스트에 주기적으로 추가하는 단계; 및

상기 제1 기하학적 파티션에 대한 단방향 MV 및 상기 제2 기하학적 파티션에 대한 단방향 MV를 생성하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

제1 참조 픽처 리스트에서 홀수 병합 인덱스를 갖는 하나 이상의 MV 후보 및 제2 참조 픽처 리스트에서 짝수 병합 인덱스를 갖는 하나 이상의 MV를 획득함으로써 상기 하나 이상의 추가 단방향 MV를 도출하는 단계를 더 포함하는, 방법.

청구항 3

제2항에 있어서,

상기 제1 참조 픽처 리스트는 참조 픽처 L0이고, 상기 제2 참조 픽처 리스트는 L1인, 방법.

청구항 4

제2항에 있어서,

상기 제1 참조 픽처 리스트는 참조 픽처 L1이고, 상기 제2 참조 픽처 리스트는 참조 픽처 L0인, 방법.

청구항 5

제1항에 있어서,

상기 하나 이상의 쌍별 평균 후보를 상기 제1 업데이트된 단방향 MV 후보 리스트에 추가하는 단계는:

제1 참조 픽처 리스트 또는 제2 참조 픽처 리스트에서 처음 2개의 단방향 MV 후보를 획득하는 단계; 및

상기 처음 2개의 단방향 MV 후보가 동일한 참조 픽처를 지시한다는 결정에 대한 응답으로, 상기 처음 2개의 단방향 MV 후보를 평균함으로써 쌍별 평균 후보를 획득하는 단계

를 더 포함하는 방법.

청구항 6

제5항에 있어서,

상기 처음 2개의 단방향 MV 후보가 상이한 참조 픽처를 지시한다는 결정에 대한 응답으로, 상기 처음 2개의 단방향 MV 후보를 평균함으로써 쌍별 평균 후보의 크기를 결정하고 첫 번째 단방향 MV 후보의 참조 픽처를 상기 쌍별 평균 후보의 참조 픽처로서 결정함으로써 상기 쌍별 평균 후보를 획득하는 단계를 더 포함하는 방법.

청구항 7

제5항에 있어서,

상기 처음 2개의 단방향 MV 후보가 상이한 참조 픽처를 지시한다고 결정하는 것에 대한 응답으로, 상기 처음 2개의 단방향 MV 후보를 평균함으로써 상기 쌍별 평균 후보의 크기를 결정하고 두 번째 단방향 MV 후보의 참조 픽처를 상기 쌍별 평균 후보의 참조 픽처로서 결정함으로써 상기 쌍별 평균 후보를 획득하는 단계를 더 포함하는 방법.

청구항 8

제1항에 있어서,

상기 단방향 MV 후보 리스트로부터 중복 후보를 제거하는 단계를 더 포함하는 방법.

청구항 9

제8항에 있어서,

추가 단방향 MV가 상기 단방향 MV 후보 리스트 내의 후보와 동일하다는 결정에 대한 응답으로, 상기 추가 단방향 MV를 상기 단방향 MV 후보 리스트에 추가하는 것을 생략하는 단계를 더 포함하는 방법.

청구항 10

제9항에 있어서,

상기 추가 단방향 MV와 상기 단방향 MV 후보 리스트 내의 후보 간의 차이가 MV 임계치보다 더 작다는 결정에 대한 응답으로, 상기 추가 단방향 MV가 상기 단방향 MV 후보 리스트 내의 후보와 동일한 것으로 결정하는 단계를 더 포함하며,

상기 MV 임계치는 고정 임계치 또는 상기 비디오 블록의 블록 크기에 기초한 변수인, 방법.

청구항 11

제5항에 있어서,

쌍별 평균 후보가 상기 제1 업데이트된 단방향 MV 후보 리스트 내의 후보와 동일하다는 결정에 대한 응답으로, 상기 쌍별 평균 후보를 상기 제1 업데이트된 단방향 MV 후보 리스트에 추가하는 것을 생략하는 단계를 더 포함하는 방법.

청구항 12

제11항에 있어서,

상기 쌍별 평균 후보와 상기 제1 업데이트된 단방향 MV 후보 리스트 내의 후보 간의 차이가 MV 임계치보다 더 작다고 결정하는 것에 대한 응답으로, 상기 쌍별 평균 후보가 상기 후보와 동일하다고 결정하는 단계를 더 포함하며,

상기 MV 임계치는 고정 임계치 또는 상기 비디오 블록의 블록 크기에 기초한 변수를 포함하는, 방법.

청구항 13

비디오 코딩을 위한 장치로서,

하나 이상의 프로세서; 및

상기 하나 이상의 프로세서에 의해 실행 가능한 명령을 저장하도록 구성된 비-일시적 컴퓨터 판독 가능 저장 매

체를 포함하며,

상기 하나 이상의 프로세서는 상기 명령의 실행 시, 제1항 내지 제12항 중 어느 한 항의 방법을 수행하도록 구성되는,

장치.

청구항 14

하나 이상의 컴퓨터 프로세서에 의해 실행될 때, 상기 하나 이상의 컴퓨터 프로세서로 하여금 제1항 내지 제12항 중 어느 한 항의 방법을 수행하게 하는 컴퓨터 실행 가능 명령을 저장하는,

비-일시적 컴퓨터 판독 가능 저장 매체.

발명의 설명

기 술 분 야

[0001] 본 출원은 2021년 6월 28일자로 출원된 가출원 제63/215,957호에 대한 우선권에 기반하며 그러한 우선권을 주장하고, 이 가출원의 개시내용은 그 전체가 모든 목적을 위해 인용에 의해 본 명세서에 포함된다.

[0002] 본 개시내용은 비디오 코딩 및 압축에 관한 것이다. 보다 구체적으로, 본 개시내용은 각도 가중 예측(AWP: angular weighted predict) 모드로도 또한 알려진 기하학적 파티션 모드(GPM: geometric partition mode)의 코딩 효율을 개선하는 것에 관한 방법 및 장치에 관한 것이다.

배 경 기 술

[0003] 비디오 데이터를 압축하기 위해 다양한 비디오 코딩 기법이 사용될 수 있다. 비디오 코딩은 하나 이상의 비디오 코딩 표준에 따라 수행된다. 예를 들어, 오늘날, 일부 잘 알려진 비디오 코딩 표준은 VVC(Versatile Video Coding), (H.265 또는 MPEG-H Part2로도 또한 알려진) HEVC(High Efficiency Video Coding) 및 (H.264 또는 MPEG-4 파트 10으로도 또한 알려진) AVC(Advanced Video Coding)를 포함하며, 이는 ISO/IEC MPEG 및 ITU-T VECG에 의해 공동으로 개발된다. AV1(AOMedia Video 1)은 그의 선행 표준 VP9에 대한 계승자로서 AOM(Alliance for Open Media)에 의해 개발되었다. 디지털 오디오 및 디지털 비디오 압축 표준을 지칭하는 AVS(Audio Video Coding)는 중국의 오디오 및 비디오 코딩 표준 작업그룹에 의해 개발된 다른 비디오 압축 표준 시리즈이다. 기존의 비디오 코딩 표준의 대부분은 유명한 하이브리드 비디오 코딩 프레임워크 상에, 즉 비디오 이미지 또는 시퀀스에 존재하는 리던던시를 감소시키기 위해 블록 기반 예측 방법(예컨대, 인터 예측(inter-prediction), 인트라 예측(intra-prediction))을 사용하고 예측 에러의 에너지를 압축하기 위해 변환 코딩을 사용하여 구축된다. 비디오 코딩 기법의 중요한 목표는 비디오 품질에 대한 열화를 피하거나 최소화하면서 더 낮은 비트 레이트를 사용하는 형태로 비디오 데이터를 압축하는 것이다.

발명의 내용

[0004] 본 개시내용은 비디오 코딩을 위한 방법 및 장치와 및 비-일시적 컴퓨터 판독 가능 저장 매체를 제공한다.

[0005] 본 개시내용의 제1 양상에 따르면, GPM에서 비디오 블록을 디코딩하기 위한 방법이 제공된다. 이 방법은 비디오 블록을 제1 기하학적 파티션 및 제2 기하학적 파티션으로 파티셔닝하는 단계를 포함할 수 있다. 이 방법은 복수의 정규 병합 후보를 추가함으로써 GPM의 단방향 움직임 벡터(MV: motion vector) 후보 리스트를 구성하는 단계를 포함할 수 있다. 이 방법은 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로 정규 병합 후보 리스트의 하나 이상의 양방향 예측 MV로부터 도출된 하나 이상의 추가 단방향 MV를 단방향 MV 후보 리스트에 추가함으로써 제1 업데이트된 단방향 MV 후보 리스트를 구성하는 단계를 포함할 수 있다.

[0006] 이 방법은 제1 업데이트된 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로 제1 업데이트된 단방향 MV 후보 리스트에 하나 이상의 쌍별 평균 후보를 추가함으로써 제2 업데이트된 단방향 MV 후보 리스트를 구성하는 단계를 포함할 수 있다. 이 방법은 또한, 제2 업데이트된 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로, 최대 길이에 도달할 때까지 0의 단방향 MV를 제2 업데이트된 단방향 MV 후보 리스트에 주기적으로 추가하는 단계를 포함할 수 있다. 이 방법은 제1 기하학적 파티션에 대한 단방향 MV 및 제2 기하학적 파티션에 대한 단방향 MV를 생성하는 단계를 더 포함할 수 있다.

[0007] 본 개시내용의 제2 양상에 따르면, 비디오 디코딩을 위한 장치가 제공된다. 이 장치는 하나 이상의 프로세서들 및 비-일시적 컴퓨터 판독 가능 저장 매체를 포함할 수 있다 비-일시적 컴퓨터 판독 가능 저장 매체는 하나 이상의 프로세서에 의해 실행 가능한 명령을 저장하도록 구성된다. 하나 이상의 프로세서는 명령의 실행 시에, 제1 양상의 방법을 수행하도록 구성된다.

[0008] 본 개시내용의 제3 양상에 따르면, 비-일시적 컴퓨터 판독 가능 저장 매체가 제공된다. 비-일시적 컴퓨터 판독 가능 저장 매체는 하나 이상의 컴퓨터 프로세서에 의해 실행될 때, 하나 이상의 컴퓨터 프로세서로 하여금 제1 양상의 방법을 수행하게 하는 컴퓨터 실행 가능 명령을 저장할 수 있다.

도면의 간단한 설명

[0009] 본 명세서의 일부에 포함되며 이를 구성하는 첨부 도면이 본 개시내용에 부합하는 예를 예시하며, 설명과 함께 본 개시내용의 원리를 설명하는 역할을 한다.

도 1은 본 개시내용의 일례에 따른 인코더의 블록도이다.

도 2는 본 개시내용의 일례에 따른 디코더의 블록도이다.

도 3a는 본 개시내용의 일례에 따른 다중 타입 트리 구조의 블록 파티션을 예시하는 도면이다.

도 3b는 본 개시내용의 일례에 따른 다중 타입 트리 구조의 블록 파티션을 예시하는 도면이다.

도 3c는 본 개시내용의 일례에 따른 다중 타입 트리 구조의 블록 파티션을 예시하는 도면이다.

도 3d는 본 개시내용의 일례에 따른 다중 타입 트리 구조의 블록 파티션을 예시하는 도면이다.

도 3e는 본 개시내용의 일례에 따른 다중 타입 트리 구조의 블록 파티션을 예시하는 도면이다.

도 4는 본 개시내용의 일례에 따른, 허용되는 기하학적 파티션 모드(GPM) 파티션의 예시이다.

도 5는 본 개시내용의 일례에 따른, 단방향 예측(uni-prediction) 움직임 벡터 선택을 예시하는 표이다.

도 6a는 본 개시내용의 일례에 따른, 움직임 벡터 차(MMVD: motion vector differences) 모드의 예시이다.

도 6b는 본 개시내용의 일례에 따른 MMVD 모드의 예시이다.

도 7은 본 개시내용의 일례에 따른 템플릿 매칭(TM) 알고리즘의 예시이다.

도 8은 본 개시내용의 일례에 따른, GPM에서 비디오 블록을 디코딩하는 방법이다.

도 9는 본 개시내용의 일례에 따라, 사용자 인터페이스와 결합된 컴퓨팅 환경을 예시하는 도면이다.

도 10은 본 개시내용의 일부 예에 따라 비디오 블록을 인코딩 및 디코딩하기 위한 시스템을 예시하는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0010] 이제 실시예에 대해 상세히 언급될 것이며, 그 예는 첨부 도면에 예시된다. 다음의 설명은, 달리 표현되지 않는 한, 상이한 도면의 동일한 번호가 동일한 또는 유사한 엘리먼트를 나타내는 첨부 도면을 참조한다. 실시예의 다음의 설명에서 제시된 구현은 본 개시내용과 일치하는 모든 구현을 표현하는 것은 아니다. 대신에, 이들은 단지, 첨부된 청구항에서 언급되는 바와 같은 본 개시내용에 관련된 양상과 일치하는 장치 및 방법의 예일 뿐이다.

[0011] 본 개시내용에서 사용된 용어는 단지 특정한 실시예를 설명할 목적을 위한 것이며 본 개시내용을 제한하는 것으로 의도되지 않는다. 본 개시내용 및 첨부된 청구항의 설명에서 사용된 바와 같이, 단수 형태는 맥락이 명확하게 달리 지시하지 않는 한, 복수 형태도 포함하는 것으로 의도된다. 또한, 본 명세서에서 사용되는 "및/또는"이라는 용어는 연관된 열거된 항목 중 하나 이상의 항목의 임의의 또는 모든 가능한 조합을 의미하고 포함하는 것으로 의도된다고 이해될 것이다.

[0012] "제1", "제2" 등의 용어는 본 명세서에서 다양한 정보를 설명하는 데 사용될 수 있지만, 이러한 정보는 이러한 용어로 제한되지는 않아야 한다고 이해될 것이다. 이러한 용어는 정보의 하나의 카테고리를 다른 카테고리와 구별하기 위해서만 사용된다. 예를 들어, 본 개시내용의 범위를 벗어나지 않으면서, 제1 정보는 제2 정보로 지칭될 수 있고; 유사하게, 제2 정보는 또한 제1 정보로 지칭될 수 있다. 본 명세서에서 사용되는 바와 같이, "~

라면"이라는 용어는 맥락에 따라 "~일 때" 또는 "~ 시에" 또는 "판단에 대한 응답으로"를 의미하는 것으로 이해될 수 있다.

[0013] 제1 세대 AVS 표준은 중국 국가 표준 "Information Technology, Advanced Audio Video Coding, Part 2: Video"(AVS1로 알려짐) 및 "Information Technology, Advanced Audio Video Coding Part 16: Radio Television Video"(AVS+로 알려짐)를 포함한다. 이는 MPEG-2 표준과 비교하여 동일한 지각 품질로 약 50% 비트 레이트 절감을 제공할 수 있다. AVS1 표준 비디오 파트는 2006년 2월에 중국 국가 표준으로 반포되었다. 2세대 AVS 표준은, 추가 HD TV 프로그램의 송신을 주로 타깃으로 하는 일련의 중국 국가 표준 "Information Technology, Efficient Multimedia Coding"(AVS2로 알려짐)을 포함한다. AVS2의 코딩 효율은 AVS+의 코딩 효율의 2배이다. 2016년 5월, AVS2가 중국 국가 표준으로 발표되었다. 한편, AVS2 표준 비디오 파트는 애플리케이션을 위한 하나의 국제 표준으로서 전기 전자 기술자 협회(IEEE: Institute of Electrical and Electronics Engineers)에 의해 제출되었다. AVS3 표준은 최신 국제 표준 HEVC의 코딩 효율을 능가하는 것을 목표로 하는 UHD 비디오 애플리케이션을 위한 하나의 신세대 비디오 코딩 표준이다. 2019년 3월, 제68차 AVS 회의에서, AVS3-P2 베이스라인이 완성되었으며, 이는 HEVC 표준에 비해 대략 30% 비트 레이트 절감을 제공한다. 현재, AVS3 표준의 기준 구현을 설명하기 위해 AVS 그룹에 의해 유지되는 고성능 모델(HPM: high performance model)로 지칭되는 하나의 기준 소프트웨어가 있다.

[0014] HEVC와 같이, AVS3 표준은 블록 기반 하이브리드 비디오 코딩 프레임워크 상에 구축된다.

[0015] 도 10은 본 개시내용의 일부 구현에 따라 병렬로 비디오 블록을 인코딩 및 디코딩하기 위한 예시적인 시스템(10)을 예시하는 블록도이다. 도 1에 도시된 바와 같이, 시스템(10)은 목적지 디바이스(14)에 의해 나중에 디코딩될 비디오 데이터를 생성하여 인코딩하는 소스 디바이스(12)를 포함한다. 소스 디바이스(12) 및 목적지 디바이스(14)는 데스크톱 컴퓨터(desktop) 또는 랩톱(laptop) 컴퓨터, 태블릿(tablet) 컴퓨터, 스마트폰(smart phones), 셋톱 박스(set-top boxes), 디지털 미디어 플레이어(digital media players), 비디오 게임 콘솔(gaming consoles), 비디오 스트리밍(streaming) 디바이스 등을 포함하는 광범위한 전자 디바이스 중 임의의 디바이스를 포함할 수 있다. 일부 구현에서, 소스 디바이스(12) 및 목적지 디바이스(14)는 무선 통신 능력을 갖추고 있다.

[0016] 일부 구현에서, 목적지 디바이스(14)는 링크(16)를 통해 디코딩될 인코딩된 비디오 데이터를 수신할 수 있다. 링크(16)는 인코딩된 비디오 데이터를 소스 디바이스(12)에서 목적지 디바이스(14)로 이동시킬 수 있는 임의의 타입의 통신 매체 또는 디바이스를 포함할 수 있다. 일례로, 링크(16)는 소스 디바이스(12)가 인코딩된 비디오 데이터를 목적지 디바이스(14)에 실시간으로 직접 전송할 수 있게 하는 통신 매체를 포함할 수 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜(protocol)과 같은 통신 표준에 따라 변조되어 목적지 디바이스(14)로 전송될 수 있다. 통신 매체는 무선 주파수(RF: Radio Frequency) 스펙트럼 또는 하나 이상의 물리적 전송 라인과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수 있다. 통신 매체는 패킷 기반 네트워크, 이를테면 근거리 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 일부를 형성할 수 있다. 통신 매체는 라우터, 스위치, 기지국, 또는 소스 디바이스(12)로부터 목적지 디바이스(14)로의 통신을 가능하게 하는 데 유용할 수 있는 임의의 다른 장비를 포함할 수 있다.

[0017] 일부 다른 구현에서, 인코딩된 비디오 데이터는 출력 인터페이스(22)로부터 저장 디바이스(32)로 전송될 수 있다. 이어서, 저장 디바이스(32) 내의 인코딩된 비디오 데이터는 입력 인터페이스(28)를 통해 목적지 디바이스(14)에 의해 액세스될 수 있다. 저장 디바이스(32)는 하드 드라이브, 블루레이 디스크, 디지털 다기능 디스크(DVD: Digital Versatile Disk), 콤팩트 디스크 판독 전용 메모리(CD-ROM: Compact Disc Read-Only Memory), 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적절한 디지털 저장 매체와 같은 다양한 분산 또는 로컬 액세스 데이터 저장 매체 중 임의의 매체를 포함할 수 있다. 추가 예에서, 저장 디바이스(32)는 소스 디바이스(12)에 의해 생성된 인코딩된 비디오 데이터를 유지할 수 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수 있다. 목적지 디바이스(14)는 스트리밍 또는 다운로드를 통해 저장 디바이스(32)로부터 저장된 비디오 데이터에 액세스할 수 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 인코딩된 비디오 데이터를 목적지 디바이스(14)로 전송할 수 있는 임의의 타입의 컴퓨터일 수 있다. 예시적인 파일 서버는 (예컨대, 웹사이트용) 웹 서버, 파일 전송 프로토콜(FTP: File Transfer Protocol) 서버, 네트워크 부착 저장(NAS: Network Attached Storage) 디바이스 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스(14)는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는 데 적합한 무선 채널(예컨대, 무선 충실도(Wi-Fi: Wireless Fidelity) 접속), 유선 접속(예컨대, 디지털 가입자 회선(DSL: Digital Subscriber Line), 케이블 모뎀 등), 또는 이 둘의 조합을 포함하는 임의의 표준 데이터 접속을 통해,

인코딩된 비디오 데이터에 액세스할 수 있다. 저장 디바이스(32)로부터 인코딩된 비디오 데이터의 전송은 스트리밍 전송, 다운로드 전송 또는 이 둘의 조합일 수 있다.

[0018] 도 10에 도시된 바와 같이, 소스 디바이스(12)는 비디오 소스(18), 비디오 인코더(20) 및 출력 인터페이스(22)를 포함한다. 비디오 소스(18)는 비디오 캡처 디바이스, 예컨대 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브(archive), 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스, 및/또는 컴퓨터 그래픽 데이터를 소스 비디오로서 생성하기 위한 컴퓨터 그래픽 시스템, 또는 이러한 소스의 조합을 포함할 수 있다. 일례로, 비디오 소스(18)가 보안 감시 시스템의 비디오 카메라라면, 소스 디바이스(12) 및 목적지 디바이스(14)는 카메라폰 또는 비디오폰을 형성할 수 있다. 그러나 본 출원에서 설명되는 구현은 일반적으로 비디오 코딩에 적용 가능할 수 있고, 무선 및/또는 유선 애플리케이션에 적용될 수 있다.

[0019] 캡처되거나, 사전 캡처되거나 또는 컴퓨터 생성된 비디오는 비디오 인코더(20)에 의해 인코딩될 수 있다. 인코딩된 비디오 데이터는 소스 디바이스(12)의 출력 인터페이스(22)를 통해 목적지 디바이스(14)로 직접 전송될 수 있다. 인코딩된 비디오 데이터는 또한(또는 대안으로) 디코딩 및/또는 재생을 위해 목적지 디바이스(14) 또는 다른 디바이스에 의한 추후 액세스를 위해 저장 디바이스(32)에 저장될 수 있다. 출력 인터페이스(22)는 모뎀 및/또는 송신기를 더 포함할 수 있다.

[0020] 목적지 디바이스(14)는 입력 인터페이스(28), 비디오 디코더(30) 및 디스플레이 디바이스(34)를 포함한다. 입력 인터페이스(28)는 수신기 및/또는 모뎀을 포함할 수 있고 인코딩된 비디오 데이터를 링크(16)를 통해 수신할 수 있다. 링크(16)를 통해 전달되는 또는 저장 디바이스(32) 상에 제공되는 인코딩된 비디오 데이터는 비디오 데이터를 디코딩할 때 비디오 디코더(30)에 의한 사용을 위해 비디오 인코더(20)에 의해 생성된 다양한 신택스 엘리먼트를 포함할 수 있다. 이러한 신택스 엘리먼트는 통신 매체 상에서 전송되거나, 저장 매체 상에 저장되거나, 파일 서버 상에 저장된 인코딩된 비디오 데이터 내에 포함될 수 있다.

[0021] 일부 구현에서, 목적지 디바이스(14)는 통합 디스플레이 디바이스일 수 있는 디스플레이 디바이스(34) 및 목적지 디바이스(14)와 통신하도록 구성되는 외부 디스플레이 디바이스일 수 있다. 디스플레이 디바이스(34)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이(LCD: Liquid Crystal Display), 플라즈마 디스플레이, 유기 발광 다이오드(OLED: Organic Light Emitting Diode) 디스플레이 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스 중 임의의 디스플레이 디바이스를 포함할 수 있다.

[0022] 비디오 인코더(20) 및 비디오 디코더(30)는 독점적 또는 산업 표준, 이를테면 VVC, HEVC, MPEG-4, Part 10, AVC 또는 이러한 표준의 확장에 따라 동작할 수 있다. 본 출원은 특정 비디오 인코딩/디코딩 표준에 제한되지 않고 다른 비디오 인코딩/디코딩 표준에 적용 가능할 수 있다고 이해되어야 한다. 소스 디바이스(12)의 비디오 인코더(20)는 이러한 현재 또는 미래의 표준 중 임의의 표준에 따라 비디오 데이터를 인코딩하도록 구성될 수 있다는 것이 일반적으로 고려된다. 유사하게, 목적지 디바이스(14)의 비디오 디코더(30)는 이러한 현재 또는 미래의 표준 중 임의의 표준에 따라 비디오 데이터를 디코딩하도록 구성될 수 있다는 것이 일반적으로 또한 고려된다.

[0023] 비디오 인코더(20) 및 비디오 디코더(30)는 각각, 하나 이상의 마이크로프로세서, 디지털 신호 프로세서(DSP), 주문형 집적 회로(ASIC), 필드 프로그래밍 가능 게이트 어레이(FPGA), 이산 로직, 소프트웨어, 하드웨어, 펌웨어, 또는 이들의 임의의 조합과 같은 다양한 적합한 인코더 및/또는 디코더 회로 중 임의의 회로로서 구현될 수 있다. 전자 디바이스는 부분적으로는 소프트웨어에서 구현될 때, 소프트웨어에 대한 명령을 적절한 비-일시적 컴퓨터 판독 가능 매체에 저장하고, 본 개시내용에 개시된 비디오 인코딩/디코딩 동작을 수행하도록 하나 이상의 프로세서를 사용하여 하드웨어에서 명령을 실행할 수 있다. 비디오 인코더(20) 및 비디오 디코더(30)는 각각 하나 이상의 인코더 또는 디코더에 포함될 수 있으며, 이들 중 어느 하나는 개개의 디바이스에서 조합된 인코더/디코더(CODEC)의 일부로서 통합될 수 있다.

[0024] 도 1은 VVC에 대한 블록 기반 비디오 인코더의 일반적인 도면을 도시한다. 구체적으로, 도 1은 통상적인 인코더(100)를 도시한다. 인코더(100)는 도 10에 도시된 바와 같은 비디오 인코더(20)일 수 있다. 인코더(100)는 비디오 입력(110), 움직임 보상(112), 움직임 추정(114), 인트라/인터 모드 결정(116), 블록 예측자(140), 가산기(128), 변환(130), 양자화(132), 예측 관련 정보(142), 인트라 예측(118), 픽처 버퍼(120), 역양자화(134), 역변환(136), 가산기(126), 메모리(124), 루프 내 필터(122), 엔트로피 코딩(138) 및 비트스트림(144)을 갖는다.

[0025] 인코더(100)에서는, 비디오 프레임이 처리를 위해 복수의 비디오 블록으로 파티셔닝된다. 각각의 주어진 비디

오 블록에 대해, 인터 예측 접근 방식 또는 인트라 예측 접근 방식에 기반하여 예측이 형성된다.

[0026] 비디오 입력(110)의 일부인 현재 비디오 블록과 블록 예측자(140)의 일부인 현재 비디오 블록의 예측자 간의 차이를 표현하는 예측 잔차가 가산기(128)로부터 변환(130)으로 전송된다. 그런 다음, 엔트로피 감소를 위해 변환(130)으로부터 양자화(132)로 변환 계수가 송신된다. 이어서, 양자화된 계수가 엔트로피 코딩(138)에 공급되어 압축된 비디오 비트스트림을 생성한다. 도 1에 도시된 바와 같이, 인트라/인터 모드 결정(116)로부터의 예측 관련 정보(142), 이를테면 비디오 블록 파티션 정보, 움직임 벡터(MV: motion vector)들, 참조 픽처 인덱스 및 인트라 예측 모드가 또한 엔트로피 코딩(138)을 통해 공급되어 압축된 비트스트림(144)에 저장된다. 압축된 비트스트림(144)은 비디오 비트스트림을 포함한다.

[0027] 인코더(100)에서는, 예측 목적으로 픽셀을 재구성하기 위해 디코더 관련 회로가 또한 필요하다. 먼저, 역양자화(134) 및 역변환(136)을 통해 예측 잔차가 재구성된다. 이러한 재구성된 예측 잔차는 블록 예측자(140)와 조합되어, 현재 비디오 블록에 대한 필터링되지 않은 재구성된 픽셀을 생성한다.

[0028] 공간 예측(또는 "인트라 예측")은 현재 비디오 블록과 동일한 비디오 프레임에서 (기준 샘플로 지칭되는) 이미 코딩된 이웃 블록의 샘플로부터의 픽셀들을 사용하여 현재 비디오 블록을 예측한다.

[0029] ("인터 예측"으로도 또한 지칭되는) 시간 예측은 이미 코딩된 비디오 픽처로부터의 재구성된 픽셀을 사용하여 현재 비디오 블록을 예측한다. 시간 예측은 비디오 신호에 내재된 시간 중복성을 감소시킨다. 주어진 코딩 유닛(CU: coding unit) 또는 코딩 블록에 대한 시간 예측 신호는 대개, 현재 CU와 이것의 시간 기준 사이의 움직임의 양 및 방향을 지시하는 하나 이상의 MV에 의해 시그널링된다. 추가로, 다수의 참조 픽처가 지원된다면, 하나의 참조 픽처 인덱스가 추가로 송신되며, 이는 참조 픽처 저장소 내의 어느 참조 픽처로부터 시간 예측 신호가 오는지를 식별하는 데 사용된다.

[0030] 움직임 추정(114)은 비디오 입력(110) 및 픽처 버퍼(120)로부터의 신호를 취하여 움직임 추정 신호를 움직임 보상(112)에 출력한다. 움직임 보상(112)은 비디오 입력(110), 픽처 버퍼(120)로부터의 신호 및 움직임 추정(114)으로부터의 움직임 추정 신호를 취하여 인트라/인터 모드 결정(116)에 움직임 보상 신호를 출력한다.

[0031] 공간 및/또는 시간 예측이 수행된 후에, 인코더(100)의 인트라/인터 모드 결정(116)은 예를 들어, 레이트 왜곡 최적화 방법에 기반하여 최상의 예측 모드를 선택한다. 그 다음, 현재 비디오 블록으로부터 블록 예측자(140)가 감산되고, 결과적인 예측 잔차가 변환(130) 및 양자화(132)를 사용하여 역상관(de-correlate)된다. 결과적인 양자화된 잔류 계수는 역양자화(134)에 의해 역양자화되고, 역변환(136)에 의해 역변환되어, 재구성된 잔차를 형성하고, 재구성된 잔차는 이어서, 예측 블록에 다시 더해져 CU의 재구성된 신호를 형성한다. 또한, 재구성된 CU가 픽처 버퍼(120)의 참조 픽처 저장소에 들어가 차후 비디오 블록을 코딩하는 데 사용되기 전에, 추가 루프 내 필터링(122), 이를테면 블록 분리(deblocking) 필터, 샘플 적응 오프셋(SAO: sample adaptive offset) 및/또는 적응형 루프 내 필터(ALF: adaptive in-loop filter)가 재구성된 CU에 적용될 수 있다. 출력 비디오 비트스트림(144)을 형성하기 위해, 코딩 모드(인터 또는 인트라), 예측 모드 정보, 움직임 정보 및 양자화된 잔차 계수가 모두 엔트로피 코딩 유닛(138)에 송신되어 추가로 압축되고 패킹되어 비트스트림을 형성한다.

[0032] 도 1은 일반적인 블록 기반 하이브리드 비디오 인코딩 시스템의 블록도를 제공한다. 입력 비디오 신호는 (코딩 유닛(CU)들로 지칭되는) 블록 단위로 처리된다. 쿼드트리(quad-trees)에만 기초하여 블록을 파티셔닝하는 HEVC와 상이하게, AVS3에서, 하나의 코딩 트리 유닛(CTU: coding tree unit)은 쿼드/2진/확장 쿼드트리에 기초한 다양한 국소 특징에 적응하도록 CU로 분할된다. 추가로, HEVC에서 다수의 파티션 유닛 타입의 개념이 제거되는데, 즉 CU, 예측 유닛(PU: prediction unit) 및 변환 유닛(TU: transform unit)의 분리는 AVS3에 존재하지 않으며; 대신에, 각각의 CU가 추가 파티션 없이 예측과 변환 모두를 위한 기본 유닛으로서 항상 사용된다. AVS3의 트리 파티션 구조에서, 하나의 CTU는 먼저 쿼드트리 구조에 기반하여 파티셔닝된다. 그 다음, 각각의 쿼드 트리 리프(leaf) 노드가 2진 및 확장 쿼드트리 구조에 기반하여 추가로 파티셔닝될 수 있다.

[0033] 도 3a, 도 3b, 도 3c, 도 3d 및 도 3e에 도시된 바와 같이, 5가지 분할 타입, 즉 4진 파티셔닝, 수평 2진 파티셔닝, 수직 2진 파티셔닝, 수평 확장 쿼드트리 파티셔닝 및 수직 확장 쿼드트리 파티셔닝이 존재한다.

[0034] 도 3a는 본 개시내용에 따른, 다중 타입 트리 구조의 블록 4진 파티션을 예시하는 도면을 도시한다.

[0035] 도 3b는 본 개시내용에 따른, 다중 타입 트리 구조의 블록 수직 2진 파티션을 예시하는 도면을 도시한다.

[0036] 도 3c는 본 개시내용에 따른, 다중 타입 트리 구조의 블록 수평 2진 파티션을 예시하는 도면을 도시한다.

[0037] 도 3d는 본 개시내용에 따른, 다중 타입 트리 구조의 블록 수직 3진 파티션을 예시하는 도면을 도시한다.

[0038] 도 3e는 본 개시내용에 따른, 다중 타입 트리 구조의 블록 수평 3진 파티션을 예시하는 도면을 도시한다.

[0039] 도 1에서, 공간 예측 및/또는 시간 예측이 수행될 수 있다. 공간 예측(또는 "인트라 예측")은 동일한 비디오 픽처/슬라이스에서 (참조 샘플로 지칭되는) 이미 코딩된 이웃 블록의 샘플로부터의 픽셀을 사용하여 현재 비디오 블록을 예측한다. 공간 예측은 비디오 신호에 내재된 공간 중복성을 감소시킨다. ("인터 예측" 또는 "움직임 보상 예측"으로도 또한 지칭되는) 시간 예측은 이미 코딩된 비디오 픽처로부터의 재구성된 픽셀을 사용하여 현재 비디오 블록을 예측한다. 시간 예측은 비디오 신호에 내재된 시간 중복성을 감소시킨다. 주어진 CU에 대한 시간 예측 신호는 대개, 현재 CU와 이것의 시간 기준 사이의 움직임의 양 및 방향을 지시하는 하나 이상의 움직임 벡터(MV)에 의해 시그널링된다. 또한, 다수의 참조 픽처가 지원된다면, 하나의 참조 픽처 인덱스가 추가로 송신되며, 이는 참조 픽처 저장소 내의 어느 참조 픽처로부터 시간 예측 신호가 오는지를 식별하는 데 사용된다. 공간 및/또는 시간 예측 이후에, 인코더의 모드 결정 블록은 예를 들어, 레이트 왜곡 최적화 방법에 기반하여 최상의 예측 모드를 선택한다. 그 다음, 현재 비디오 블록으로부터 예측 블록이 감산되고; 예측 잔차가 변환을 사용하여 역상관된 다음에 양자화된다. 양자화된 잔차 계수는 역양자화되고 역변환되어, 재구성된 잔차를 형성하고, 재구성된 잔차는 이어서, 예측 블록에 다시 더해져 CU의 재구성된 신호를 형성한다. 추가로, 재구성된 CU가 참조 픽처 저장소에 들어가 차후 비디오 블록을 코딩하기 위한 참조로서 사용되기 전에, 루프 내 필터링, 이를테면 블록 분리 필터, 샘플 적응 오프셋(SAO) 및 적응형 루프 내 필터(ALF)가 재구성된 CU에 적용될 수 있다. 출력 비디오 비트스트림을 형성하기 위해, 코딩 모드(인터 또는 인트라), 예측 모드 정보, 움직임 정보 및 양자화된 잔차 계수가 모두 엔트로피 코딩 유닛에 송신되어 추가로 압축되고 패킹성한다.

[0040] 도 2는 VVC에 대한 비디오 디코더의 일반적인 블록도를 도시한다. 구체적으로, 도 2는 통상적인 디코더(200) 블록도를 도시한다. 블록 기반 비디오 디코더(200)는 도 10에 도시된 바와 같은 비디오 디코더(30)일 수 있다. 디코더(200)는 비트스트림(210), 엔트로피 디코딩(212), 역양자화(214), 역변환(216), 가산기(218), 인트라/인터 모드 선택(220), 인트라 예측(222), 메모리(230), 루프 내 필터(228), 움직임 보상(224), 픽처 버퍼(226), 예측 관련 정보(234) 및 비디오 출력(232)을 갖는다.

[0041] 이 디코더(200)는 도 1의 인코더(100)에 상주하는 재구성 관련 섹션과 유사하다. 디코더(200)에서는, 착신 비디오 비트스트림(210)이 먼저 엔트로피 디코딩(212)을 통해 디코딩되어 양자화된 계수 레벨 및 예측 관련 정보를 도출한다. 이어서, 양자화된 계수 레벨이 역양자화(214) 및 역변환(216)을 통해 처리되어, 재구성된 예측 잔차를 획득한다. 인트라/인터 모드 선택기(220)에서 구현되는 블록 예측자 메커니즘은 디코딩된 예측 정보에 기초하여 인트라 예측(222) 또는 움직임 보상(224)을 수행하도록 구성된다. 합산기(218)를 사용하여, 역변환(216)으로부터의 재구성된 예측 잔차와 블록 예측자 메커니즘에 의해 생성된 예측 출력을 합산함으로써 한 세트의 필터링되지 않은 재구성된 픽셀이 획득된다.

[0042] 재구성된 블록은 추가로, 참조 픽처 저장소로서 기능하는 픽처 버퍼(226)에 저장되기 전에 루프 내 필터(228)를 거칠 수 있다. 픽처 버퍼(226) 내의 재구성된 비디오가 송신되어 디스플레이 디바이스를 구동할 수 있을 뿐만 아니라, 차후의 비디오 블록을 예측하는 데 사용될 수 있다. 루프 내 필터(228)가 온 전환되는 상황에서, 이러한 재구성된 픽셀에 대해 필터링 동작이 수행되어 최종 재구성된 비디오 출력(232)을 도출한다.

[0043] 도 2는 블록 기반 비디오 디코더의 일반적인 블록도를 제공한다. 비디오 비트스트림은 엔트로피 디코딩 유닛에서 먼저 엔트로피 디코딩된다. 코딩 모드 및 예측 정보가 (인트라 코딩된다면) 공간 예측 유닛 또는 (인터 코딩된다면) 시간 예측 유닛 중 어느 하나에 전송되어 예측 블록을 형성한다. 잔차 변환 계수는 역양자화 유닛 및 역변환 유닛에 전송되어 잔차 블록을 재구성한다. 그 다음, 예측 블록과 잔차 블록이 함께 추가된다. 재구성된 블록은 추가로, 참조 픽처 저장소에 저장되기 전에 루프 내 필터링을 거칠 수 있다. 그런 다음, 참조 픽처 저장소 내의 재구성된 비디오가 디스플레이를 위해 송신될 뿐만 아니라, 차후의 비디오 블록을 예측하는 데 사용된다.

[0044] 본 개시내용의 초점은 VVC 표준과 AVS3 표준 모두에서 사용되는 기하학적 파티션 모드(GPM: geometric partition mode)의 코딩 성능을 개선하는 것이다. AVS3에서, 툴은 또한, 각도 가중 예측(AWP)으로 알려져 있으며, 이는 GPM의 동일한 설계 사상을 따르지만, 특정 설계 세부사항들에서 약간의 미묘한 차이들을 갖는다. 본 개시내용의 설명을 가능하게 하기 위해, 이하에서 VVC 표준의 기존 GPM 설계는 GPM/AWP 툴의 주요 양상들을 설명하기 위한 예로서 사용된다. 한편, 본 개시내용에서 제안된 기술과 밀접하게 관련되면, VVC 표준과 AVS3 표준 모두에 적용되는 움직임 벡터 차를 이용한 병합 모드(MMVD: merge mode with motion vector differences)로 지칭되는 다른 기존의 인터 예측 기술이 또한 간략히 검토된다. 그 후, 현재 GPM/AWP 설계의 일부 단점이 식별

된다. 마지막으로, 제안된 방법이 상세히 제공된다. VVC 표준의 기존 GPM 설계가 본 개시내용 전반에 걸쳐 예로서 사용되지만, 현대 비디오 코딩 기술의 당해 기술분야에서 통상의 지식을 가진 자에게, 제안된 기술은 또한 동일한 또는 유사한 설계 사상을 갖는 다른 GPM/AWP 설계 또는 다른 코딩 툴에 적용될 수 있음을 주목한다.

[0045] **기하학적 파티션 모드(GPM)**

[0046] VVC에서, 인터 예측을 위해 기하학적 파티셔닝 모드가 지원된다. 기하학적 파티셔닝 모드는 하나의 특수 병합 모드로서 하나의 CU 레벨 플래그에 의해 시그널링된다. 현재의 GPM 설계에서, 각각의 가능한 CU 크기에 대해 GPM 모드에 의해 총 64개의 파티션이 지원되며, 폭과 높이 둘 다 8×64 및 64×8을 제외하고 8 이상 64 이하이다.

[0047] 이 모드가 사용될 때, CU는 도 4에 도시된 바와 같이 기하학적으로 위치된 직선에 의해 2개의 파트들로 분할된다(아래에 설명이 제공됨). 분할선의 위치는 특정 파티션의 각도 및 오프셋 파라미터들로부터 수학적으로 도출된다. CU에서의 기하학적 파티션의 각각의 파트는 각자의 움직임을 사용하여 인터 예측되고; 각각의 파티션에 대해 단방향 예측만이 허용되는데, 즉 각각의 파트는 하나의 움직임 벡터 및 하나의 기준 인덱스를 갖는다. 단방향 예측 움직임 제약은, 종래의 양방향 예측(bi-prediction)과 동일하게, 각각의 CU에 대해 2개의 움직임 보상된 예측만이 필요하다는 것을 보장하기 위해 적용된다. 현재 CU에 대해 기하학적 파티셔닝 모드가 사용된다면, 기하학적 파티션의 파티션 모드를 표시하는 기하학적 파티션 인덱스(각도 및 오프셋), 및 (각각의 파티션에 대해 하나씩) 2개의 병합 인덱스들이 추가로 시그널링된다. 최대 GPM 후보 크기의 수는 시퀀스 레벨에서 명시적으로 시그널링된다.

[0048] 도 4는 허용되는 GPM 파티션을 도시하며, 여기서 각각의 픽처에서의 분할은 하나의 동일한 분할 방향을 갖는다.

[0049] **단방향 예측 후보 리스트 구성**

[0050] 하나의 기하학적 파티션에 대한 단방향 예측 움직임 벡터를 도출하기 위해, 하나의 단방향 예측 후보 리스트가 먼저 정규 병합 후보 리스트 생성 프로세스로부터 직접 도출된다. n을 기하학적 단방향 예측 후보 리스트에서의 단방향 예측 움직임의 인덱스로서 표기한다. X가 n의 패리티와 동일한, n번째 병합 후보의 LX 움직임 벡터가 기하학적 파티셔닝 모드에 대한 n번째 단방향 예측 움직임 벡터로서 사용된다.

[0051] 이러한 움직임 벡터들은 (아래에서 설명되는) 도 5에서 "x"로 마킹된다. n개의 확장된 병합 후보의 대응하는 LX 움직임 벡터가 존재하지 않는 경우, 동일한 후보의 L(1 - X) 움직임 벡터는 기하학적 파티셔닝 모드에 대한 단방향 예측 움직임 벡터로서 대신 사용된다.

[0052] 도 5는 GPM에 대한 병합 후보 리스트의 움직임 벡터들로부터의 단방향 예측 움직임 벡터 선택을 도시한다.

[0053] **기하학적 파티션 에지를 따르는 혼합**

[0054] 각각의 기하학적 파티션이 각자의 움직임을 사용하여 획득된 후, 기하학적 파티션 에지 주위에서 샘플을 도출하기 위해 2개의 단방향 예측 신호들에 혼합이 적용된다. CU의 각각의 포지션에 대한 혼합 가중치는 각각의 개별 샘플 포지션으로부터 대응하는 파티션 에지까지의 거리에 기반하여 도출된다.

[0055] **GPM 시그널링 설계**

[0056] 현재 GPM 설계에 따르면, GPM의 사용은 CU 레벨에서 하나의 플래그를 시그널링함으로써 표시된다. 플래그는, 현재 CU가 병합 모드 또는 스킵 모드에 의해 코딩될 때에만 시그널링된다. 구체적으로, 플래그가 1과 같을 때, 이는 현재 CU가 GPM에 의해 예측됨을 표시한다. 그렇지 않으면(플래그가 0와 같으면), CU는 정규 병합 모드, 움직임 벡터 차를 이용한 병합 모드, 조합된 인터 및 인트라 예측 등과 같은 다른 병합 모드에 의해 코딩된다. GPM이 현재 CU에 대해 인에이블될 때, 하나의 신택스 엘리먼트, 즉 merge_gpm_partition_idx는 (도 4에 도시된 바와 같이 CU를 2개의 파티션으로 분할하는, CU 중심으로부터의 직선의 방향 및 오프셋을 특정하는) 적용된 기하학적 파티션 모드를 표시하도록 추가로 시그널링된다. 그 후, 2개의 신택스 엘리먼트 merge_gpm_idx0 및 merge_gpm_idx1은 제1 GPM 파티션 및 제2 GPM 파티션에 사용되는 단방향 예측 병합 후보의 인덱스들을 표시하도록 시그널링된다. 보다 구체적으로, 그러한 2개의 신택스 엘리먼트는 "단방향 예측 병합 리스트 구성" 섹션에서 설명된 바와 같이 단방향 예측 병합 리스트로부터 2개의 GPM 파티션의 단방향 MV을 결정하는 데 사용된다. 현재의 GPM 설계에 따르면, 2개의 단방향 MV를 보다 상이하게 하기 위해, 2개의 인덱스는 동일할 수 없다. 이러한 사전 지식에 기초하여, 제1 GPM 파티션의 단방향 병합 인덱스가 먼저 시그널링되고, 제2 GPM 파티션의 단방향 예측 병합 인덱스의 시그널링 오버헤드를 감소시키기 위한 예측자로서 사용된다. 상세하게는, 제2 단방향 예측 병합 인덱스가 제1 단방향 예측 병합 인덱스보다 작다면, 제2 단방향 예측 병합 인덱스의 원래 값이 직접

적으로 시그널링된다. 그렇지 않으면(제2 단방향 예측 병합 인덱스가 제1 단방향 예측 병합 인덱스보다 더 크다면), 그 값은 비트스트림에 시그널링되기 전에 1만큼 감산된다. 디코더 측에서, 제1 단방향 예측 병합 인덱스가 먼저 디코딩된다. 그런 다음, 제2 단방향 예측 병합 인덱스의 디코딩을 위해, 파싱된 값이 제1 단방향 예측 병합 인덱스보다 작다면, 제2 단방향 예측 병합 인덱스는 파싱된 값과 동일하게 설정되고; 그렇지 않으면(파싱된 값이 제1 단방향 예측 병합 인덱스 이상이면), 제2 단방향 예측 병합 인덱스는 파싱된 값에 1을 더한 것과 동일하게 설정된다. 표 1은 현재 VVC 규격에서 GPM 모드에 대해 사용되는 기존의 신택스 엘리먼트를 예시한다.

[0057] **표 1** VVC 규격의 병합 데이터 신택스 표 내의 기존의 GPM 신택스 엘리먼트

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
……	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
merge_gpm_idx0[x0][y0]	ae(v)
if(MaxNumGpmMergeCand > 2)	
merge_gpm_idx1[x0][y0]	ae(v)
}	
……	
}	

[0058]

[0059] 다른 한편으로는, 현재 GPM 설계에서, 절단된 단항 코드가 2개의 단방향 병합 인덱스, 즉 merge_gpm_idx0 및 merge_gpm_idx1의 이진화를 위해 사용된다. 추가로, 2개의 단방향 예측 병합 인덱스가 동일할 수 없기 때문에, merge_gpm_idx0 및 merge_gpm_idx1에 대해 각각 MaxGPMMergeCand - 1 및 MaxGPMMergeCand - 2와 동일하게 설정되는 2개의 단방향 예측 병합 인덱스의 코드워드를 절단하기 위해 상이한 최대값이 사용된다. MaxGPMMergeCand는 단방향 예측 병합 리스트 내의 후보의 수이다. 디코더 측에서, 수신된 merge_gpm_idx1의 값이 merge_gpm_idx0의 값 이상일 때, merge_gpm_idx0과 merge_gpm_idx1의 값이 동일할 수 없다면, merge_gpm_idx1의 값은 1만큼 증가될 것이다.

[0060] GPM/AWP 모드가 적용될 때, 신택스 merge_gpm_partition_idx를 2진 비트의 스트링으로 변환하기 위해 2개의 상이한 이진화 방법이 적용된다. 구체적으로, 신택스 엘리먼트는 VVC 표준 및 AVS3 표준에서 각각 고정 길이 코드 및 절단된 2진 코드에 의해 이진화된다. 한편, AVS3의 AWP 모드에 대해, 상이한 최대값이 신택스 엘리먼트의 값의 이진화에 대해 사용된다. 구체적으로, AVS3에서, 허용된 GPM/AWP 파티션 모드의 수는 56인 반면(즉, merge_gpm_partition_idx의 최대값은 55임), 그 수는 VVC에서 64로 증가된다(즉, merge_gpm_partition_idx의 최대값은 63임).

[0061] **움직임 벡터 차를 이용한 병합 모드(MMVD)**

[0062] 하나의 현재 블록의 공간/시간 이웃들로부터 하나의 현재 블록의 움직임 정보를 도출하는 종래의 병합 모드에 부가하여, MMVD/UMVE 모드가 VVC 표준 및 AVS 표준 모두에서 하나의 특수 병합 모드로서 도입된다. 구체적으로, VVC 및 AVS3 모두에서, 모드는 코딩 블록 레벨에서 하나의 MMVD 플래그에 의해 시그널링된다. MMVD 모드에서, 정규 병합 모드에 대한 병합 리스트에서 처음 2개의 후보가 MMVD에 대한 2개의 기본 병합 후보로서 선택된다. 하나의 기본 병합 후보가 선택되어 시그널링된 후, 선택된 병합 후보의 움직임에 부가되는 움직임 벡터 차(MVD: motion vector difference)를 표시하기 위해 추가 신택스 엘리먼트가 시그널링된다. MMVD 신택스 엘리먼트는 기본 병합 후보를 선택하기 위한 병합 후보 플래그, MVD 크기를 특정하기 위한 거리 인덱스, 및 MVD 방향을 표시하기 위한 방향 인덱스를 포함한다.

[0063] 기존의 MMVD 설계에서, 거리 인덱스는 시작점으로부터의 한 세트의 미리 정의된 오프셋에 기반하여 정의되는 MVD 크기를 특정한다. 도 6a 및 도 6b에 도시된 바와 같이, 오프셋이 시작 MV(즉, 선택된 기본 병합 후보의 MV)의 수평 또는 수직 컴포넌트에 추가된다.

[0064] 도 6a는 L0 기준에 대한 MMVD 모드를 도시한다. 도 6b는 L1 기준에 대한 MMVD 모드를 도시한다.

[0065] 표 2는 AVS3에 각각 적용되는 MVD 오프셋을 예시한다.

[0066] **표 2** AVS3에서 사용되는 MVD 오프셋

거리 IDX	0	1	2	3	4
오프셋 (루마 샘플 단위)	1/4	1/2	1	2	4

[0067]

[0068] 표 3에 도시된 바와 같이, 방향 인덱스는 시그널링된 MVD의 부호를 특정하는 데 사용된다. MVD 부호의 의미는 시작 MV에 따라 변형될 수 있다는 것이 주목된다. 시작 MV가 단방향 예측 MV 또는 양방향 예측 MV이며, MV는 POC가 현재 픽처의 POC보다 둘 다 더 크거나 현재 픽처의 POC보다 둘 다 더 작은 2개의 참조 픽처를 가리키는 경우, 시그널링된 부호는 시작 MV에 추가된 MVD의 부호이다. 시작 MV가, 하나의 픽처의 POC가 현재 픽처보다 더 크고 다른 픽처의 POC가 현재 픽처보다 더 작은 2개의 참조 픽처를 가리키는 양방향 예측 MV일 때, 시그널링된 부호는 L0 MVD에 적용되고, 시그널링된 부호의 반대 값은 L1 MVD에 적용된다.

[0069] **표 3** 방향 인덱스에 의해 특정된 MVD 부호

방향 IDX	00	01	10	11
x 축	+	–	N/A	N/A
y 축	N/A	N/A	+	–

[0070]

[0071] **정규 인터 모드에 대한 움직임 시그널링**

[0072] HEVC 표준과 유사하게, 병합/스킵 모드 외에도, VVC와 AVS3 모두는 하나의 인터 CU가 비트스트림에서 자신의 움직임 정보를 명시적으로 특정할 수 있게 한다. 전체적으로, VVC와 AVS3 모두에서의 움직임 정보 시그널링은 HEVC 표준에서의 움직임 정보 시그널링과 동일하게 유지된다. 구체적으로, 하나의 인터 예측 신택스, 즉 inter_pred_idc는 먼저, 리스트 L0으로부터의 예측 신호인지, L1로부터의 예측 신호인지 또는 이 둘 모두로부터의 예측 신호인지를 표시하도록 시그널링된다. 각각의 사용된 기준 리스트에 대해, 대응하는 참조 픽처는 대응하는 기준 리스트에 대한 하나의 참조 픽처 인덱스 ref_idx_lx (x = 0, 1)을 시그널링함으로써 식별되고, 대응하는 MV는 MV 예측자(MVP)를 선택하는 데 사용되는 하나의 MVP 인덱스 mvp_lx_flag (x = 0, 1)에 의해 표현되며, 타깃 MV와 선택된 MVP 사이의 움직임 벡터 차(MVD)가 뒤따른다. 추가로, VVC 표준에서, 하나의 제어 플래그 mvd_l1_zero_flag가 슬라이스 레벨에서 시그널링된다. mvd_l1_zero_flag가 0과 같을 때, L1 MVD는 비트스트림에서 시그널링되고; 그렇지 않으면(mvd_l1_zero_flag 플래그가 1과 같은 경우), L1 MVD는 시그널링되지 않고, L1 MVD의 값은 인코더 및 디코더에서 항상 0으로 추론된다.

[0073] **CU 레벨 가중치를 이용한 양방향 예측**

[0074] VVC 및 AVS3 전의 이전 표준에서, 가중 예측(WP: weighted prediction)이 적용되지 않을 때, 2개의 참조 픽처로부터 획득된 단방향 예측 신호를 평균함으로써 양방향 예측 신호가 생성된다. VVC에서, 하나의 툴 코딩, 즉 CU 레벨 가중치를 이용한 양방향 예측(BCW: bi-prediction with CU-level weight)이 양방향 예측의 효율을 개선하기 위해 도입되었다. 구체적으로, 단순한 평균 대신에, BCW의 양방향 예측은 다음과 같이 도시된 바와 같이 2개의 예측 신호의 가중된 평균을 가능하게 함으로써 확장된다:

[0075] $$P'(i,j) = \left((8-w)\cdot P_0(i,j) + w\cdot P_1(i,j) + 4\right) \gg 3$$

[0076] VVC에서, 현재 픽처가 하나의 저지연 픽처일 때, 하나의 BCW 코딩 블록의 가중치는 한 세트의 미리 정의된 가중값 $w \in \{-2, 3, 4, 5, 10\}$으로부터 선택되는 것이 허용되며, 4의 가중치는 2개의 단방향 예측 신호가 동일하게 가중되는 종래의 양방향 예측 경우를 나타낸다. 저지연의 경우, 단지 3개의 가중치 $w \in \{3, 4, 5\}$만이 허용된다. 일반적으로 말해서, WP와 BCW 사이에 일부 설계 유사성이 존재하지만, 2개의 코딩 툴은 상이한 입도에서 조명 변화 문제를 해결하는 것을 목표로 한다. 그러나 WP와 BCW 간의 상호 작용이 VVC 설계를 잠재적으로 복잡하게 할 수 있기 때문에, 2개의 툴이 동시에 인에이블되는 것이 허용되지 않는다. 구체적으로, WP가 하나의 슬라이스에 대해 인에이블될 때, 슬라이스 내의 모든 양방향 예측 CU에 대한 BCW 가중치는 시그널링되지 않고 4인 것으로 추론된다(즉, 동일한 가중치가 적용됨).

[0077] **템플릿 매칭**

[0078] 템플릿 매칭(TM)은, 현재 CU의 최상부 및 좌측 이웃의 재구성된 샘플로 구성된 하나의 템플릿과 참조 픽처 내의 참조 블록(즉, 템플릿과 동일한 크기) 간의 최상의 매치를 찾음으로써 현재 CU의 움직임 정보를 미세조정하기 위한 디코더 측 MV 도출 방법이다. 도 7에 예시된 바와 같이, 하나의 MV는 [- 8, +8] 픽셀(pel) 탐색 범위 내에서 현재 CU의 초기 움직임 벡터 주위에서 탐색될 것이다. 최상의 매치는 현재 템플릿과 참조 템플릿 사이의 최저 매칭 비용, 예를 들어 절대 차의 합(SAD: sum of absolute difference), 절대 변환 차의 합(SATD: sum of absolute transformd difference) 등을 달성하는 MV로서 정의될 수 있다. 인터 코딩을 위해 TM 모드를 적용하기 위한 2개의 상이한 방식이 존재한다:

[0079] AMVP 모드에서, 현재 블록 템플릿과 참조 블록 템플릿 사이의 최소 차이에 도달하는 것을 택하기 위해 템플릿 매칭 차이에 기초하여 MVP 후보가 결정되고, 그 다음, TM은 MV 미세조정을 위해 이러한 특정 MVP 후보에 대해서만 수행한다. TM은 반복적인 다이아몬드 탐색을 사용함으로써 [-8, +8] 픽셀 탐색 범위 내의 전체 픽셀(full-pel) MVD 정밀도(또는 4 픽셀 AMVR 모드의 경우 4 픽셀)에서 시작하여 이 MVP 후보를 미세조정한다. AMVP 후보는 아래 표 13에 특정된 바와 같이 AMVR 모드에 따라 1/2 픽셀 및 1/4 픽셀이 순차적으로 이어지는 전체 픽셀 MVD 정밀도(또는 4 픽셀 AMVR 모드에 대해 4 픽셀)를 갖는 교차 탐색을 사용함으로써 추가로 미세조정될 수 있다. 이 탐색 프로세스는 MVP 후보가 TM 프로세스 후에 AMVR 모드에 의해 표시된 것과 동일한 MV 정밀도를 여전히 유지하는 것을 보장한다.

[0080] 표 14

탐색 패턴	AMVR 모드				병합 모드	
	4 픽셀	전체 픽셀	1/2 픽셀	1/4 픽셀	AltIF=0	AltIF=1
4 픽셀 다이아몬드	v					
4 픽셀 크로스	v					
전체 픽셀 다이아몬드		v	v	v	v	v
전체 픽셀 크로스		v	v	v	v	v
1/2 픽셀 크로스			v	v	v	v
1/4 픽셀 크로스				v	v	
1/8 픽셀 크로스					v	

[0081]

[0082] 병합 모드에서, 유사한 탐색 방법이 병합 인덱스에 의해 표시된 병합 후보에 적용된다. 위의 표에 도시된 바와 같이, TM은 (AMVR이 1/2 픽셀 모드일 때 사용되는) 대안적인 보간 필터가 병합된 움직임 정보에 따라 사용되는지 여부에 의존하여, 1/8 픽셀 MVD 정밀도까지 완전히 수행하거나 또는 1/2 픽셀 MVD 정밀도를 초과하는 것들을 생략할 수 있다.

[0083] 위에서 언급된 바와 같이, 2개의 GPM 파티션의 예측 샘플을 생성하기 위해 사용되는 단방향 움직임은 정규 병합 후보로부터 직접적으로 획득된다. 공간/시간 이웃 블록의 MV 사이에 강한 상관이 존재하지 않는 경우, 병합 후보로부터 도출된 단방향 MV는 각각의 GPM 파티션의 실제 움직임을 캡처하기에 충분히 정확하지 않을 수 있다. 움직임 추정은 더 정확한 움직임을 제공할 수 있지만, 이는 기존의 단방향 MV에 추가로 적용될 수 있는 임의의 움직임 미세조정으로 인해 무시할 수 없는 시그널링 오버헤드의 대가를 치른다. 다른 한편으로, MVMD 모드는 VVC 표준과 AVS3 표준 모두에서 이용되며, 이는 MVD 시그널링 오버헤드를 감소시키기 위한 하나의 효율적인 시그널링 메커니즘인 것으로 입증되었다. 따라서 GPM을 MMVD 모드와 조합하는 것이 또한 유리할 수 있다. 이러한 조합은 각각의 GPM 파티션의 개별 움직임을 캡처하기 위해 더 정확한 MV를 제공함으로써 GPM 툴의 전체 코딩 효율을 잠재적으로 개선할 수 있다.

[0084] 앞서 논의된 바와 같이, VVC 표준과 AVS3 표준 모두에서, GPM 모드는 병합/스킵 모드에만 적용된다. 모든 비-병합 인터 CU가 GPM의 유연한 비-직사각형 파티션으로부터 이익을 얻을 수 없다면, 이러한 설계는 코딩 효율의 관점에서 최적이 아닐 수 있다. 다른 한편으로는, 위에서 언급된 것과 동일한 이유 때문에, 정규 병합/스킵 모드로부터 도출된 단방향 예측 움직임 후보는 2개의 기하학적 파티션의 실제 움직임을 캡처하기에 항상 정밀하지

는 않다. 이러한 분석에 기초하여, 비-병합 인터 모드(즉, 비트스트림에서 자신의 움직임 정보를 명시적으로 시그널링하는 CU)로의 GPM 모드의 합리적인 확장에 의해 여분의 코딩 이득이 예상될 수 있다. 그러나 MV 정확도에 대한 개선은 증가된 시그널링 오버헤드를 희생시킨다. 따라서 GPM 모드를 명시적 인터 모드에 효율적으로 적용하기 위해, 2개의 기하학적 파티션에 대한 더 정확한 MV를 제공하면서 시그널링 비용을 최소화할 수 있는 하나의 효과적인 시그널링 방식을 식별하는 것이 중요할 것이다.

[0085] **제안된 방법**

[0086] 본 개시내용에서, 각각의 GPM 파티션에 적용되는 기존의 단방향 MV에 추가로, 추가 움직임 미세조정들을 적용함으로써 GPM의 코딩 효율을 더 개선하기 위한 방법이 제안된다. 제안된 방법은 움직임 벡터 미세조정을 이용한 기하학적 파티션 모드(GPM-MVR)로 명명된다. 추가로, 제안된 방식에서, 움직임 미세조정은 기존의 MMVD 설계의 하나의 유사한 방식으로, 즉 움직임 미세조정의 한 세트의 미리 정의된 MVD 크기 및 방향에 기초하여 시그널링된다.

[0087] 본 개시내용의 다른 양상에서, GPM 모드를 명시적 인터 모드로 확장하기 위한 솔루션이 제공된다. 설명을 가능하게 하기 위해, 이러한 방식은 명시적 움직임 시그널링을 이용한 기하학적 파티션 모드(GPM-EMS: geometric partition mode with explicit motion signaling)로서 명명된다. 구체적으로, 정규 인터 모드와의 더 양호한 조화를 달성하기 위해, 기존의 움직임 시그널링 메커니즘, 즉 MVP + MVD가 제안된 GPM-EMS 방식에서 2개의 기하학적 파티션의 대응하는 단방향 MV를 특정하는 데 이용된다.

[0088] **별개의 움직임 벡터 미세조정을 이용한 기하학적 파티션 모드**

[0089] GPM의 코딩 효율을 개선하기 위해, 이러한 섹션에서, 별개의 움직임 벡터 미세조정을 이용한 하나의 개선된 기하학적 파티션 모드가 제안된다. 구체적으로, GPM 파티션이 주어지면, 제안된 방법은 먼저, 기존의 단방향 예측 병합 후보 리스트로부터 2개의 GPM 파티션에 대한 단방향 MV를 식별하고 이를 기본 MV로서 사용하기 위해 기존의 신택스 merge_gpm_idx0 및 merge_gpm_idx1을 사용한다. 2개의 기본 MV가 결정된 후에, 2개의 GPM 파티션의 기본 MV에 부가하여 개별적으로 적용되는 움직임 미세조정의 값을 특정하기 위해, 2개의 세트의 새로운 신택스 엘리먼트가 도입된다. 구체적으로, 2개의 플래그, 즉 gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag가 먼저, GPM-MVR이 제1 GPM 파티션 및 제2 GPM 파티션에 각각 적용되는지 여부를 표시하도록 시그널링한다. 하나의 GPM 파티션의 플래그가 1과 같은 경우, 파티션의 기본 MV에 적용되는 MVR의 대응하는 값, 즉 MVR의 크기를 특정하기 위한 (신택스 엘리먼트 gpm_mvr_partIdx0_distance_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) 하나의 거리 인덱스 및 MVR의 방향을 특정하기 위한 (신택스 엘리먼트 gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) 하나의 방향 인덱스가 MMVD 스타일로 시그널링된다. 표 4는 제안된 GPM-MVR 방법에 의해 도입되는 신택스 엘리먼트를 예시한다.

[0090] **표 4** 2개의 GPM 파티션에 대한 별개의 MVR을 이용한 제안된 GPM-MVR 방법(방법 1)의 신택스 엘리먼트

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
.....	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
gpm_mvr_parTIdx0_enable_flag[x0][y0]	ae(v)
if(gpm_mvr_parTIdx0_enable_flag[x0][y0]) {	
gpm_mvr_parTIdx0_direc to in_idx[x0][y0]	ae(v)
gpm_mvr_parTIdx0_distance_idx[x0][y0]	ae(v)
}	
if(merge_gpm_idx0 != merge_gpm_idx1 || gpm_mvr_parTIdx0_enable_flag)	
gpm_mvr_parTIdx1_enable_flag[x0][y0]	ae(v)
if(gpm_mvr_parTIdx1_enable_flag[x0][y0]) {	
gpm_mvr_parTIdx1_direction_idx[x0][y0]	ae(v)
gpm_mvr_parTIdx1_distance_idx[x0][y0]	ae(v)
}	
}	
.....	
}	

[0091]

[0092] 표 4에 도시된 바와 같은 제안된 신택스 엘리먼트에 기초하여, 디코더에서, 각각의 GPM 파티션의 단방향 예측 샘플을 생성하기 위해 사용되는 최종 MV는 시그널링된 움직임 벡터 미세조정 및 대응하는 기본 MV의 합과 같다. 실제로, 상이한 세트의 MVR 크기 및 방향이 미리 정의되어 제안된 GPM-MVR 방식에 적용될 수 있으며, 이는 움직임 벡터 정확도와 시그널링 오버헤드 사이의 다양한 절충을 제공할 수 있다. 하나의 특정 예에서, 제안된 GPM-MVR 방식에 대한 VVC 표준에서 사용된 8개의 MVD 오프셋(즉, 1/4, 1/2, 1, 2, 4, 8, 16 및 32 픽셀) 및 4개의 MVD 방향(즉, (+/- x 축 및 y 축)을 재사용하는 것이 제안된다. 다른 예에서, AVS3 표준에서 사용되는 기존의 5개의 MVD 오프셋 {1/4, 1/2, 1, 2 및 4 픽셀} 및 4개의 MVD 방향(즉, +/- x 축 및 y 축)이 제안된 GPM-MVR 방식에 적용된다.

[0093] "GPM 시그널링 설계" 섹션에서 논의된 바와 같이, 2개의 GPM 파티션에 사용되는 단방향 MV가 동일할 수 없기 때문에, 2개의 단방향 병합 인덱스를 상이하게 강제하는 하나의 제약이 기존의 GPM 설계에 적용된다. 그러나 제안된 GPM-MVR 방식에서, 기존의 GPM 단방향 MV에 추가로, 추가 움직임 미세조정이 적용된다. 따라서 2개의 GPM 파티션의 기본 MV가 동일한 경우에도, 2개의 움직임 벡터 미세조정의 값이 동일하지 않은 한, 2개의 파티션을 예측하는 데 사용되는 최종 단방향 MV는 여전히 상이할 수 있다. 위의 고려사항에 기반하여, 제안된 GPM-MVR 방식이 적용될 때, (2개의 단방향 예측 병합 인덱스를 서로 다르도록 제약하는) 제약이 제거된다. 추가로, 2개의 단방향 예측 병합 인덱스가 동일한 것이 허용되기 때문에, merg_gpm_idx0과 merge_gpm_idx1 모두의 이진화에 대해 동일한 최대값 MaxGPMMergeCand - 1이 사용되며, MaxGPMMergeCand는 단방향 예측 병합 리스트 내의 후보의 수이다.

[0094] 앞서 분석된 바와 같이, 2개의 GPM 파티션의 단방향 예측 병합 인덱스(즉, merge_gpm_idx0 및 merge_gpm_idx

1)가 동일할 때, 2개의 파티션에 대해 사용되는 최종 MV가 상이하다는 것을 보장하기 위해 2개의 움직임 벡터 미세조정의 값은 동일할 수 없다. 이러한 조건에 기초하여, 본 개시내용의 일 실시예에서, 2개의 GPM 파티션이 동일(즉, merge_gpm_idx0이 merge_gpm_idx1과 동일)할 때, 제1 GPM 파티션의 MVR을 사용하여 제2 GPM 파티션의 MVR의 시그널링 오버헤드를 감소시키기 위한 하나의 시그널링 리던던시 제거 방법이 제안된다. 일례로, 다음의 시그널링 조건들이 적용된다:

[0095] 첫째, 플래그 gpm_mvr_partIdx0_enable_flag가 0과 같을 때(즉, GPM-MVR이 제1 GPM 파티션에 적용되지 않음), gpm_mvr_partIdx1_enable_flag의 플래그가 시그널링되는 것이 아니라, 1인 것으로 추론된다(즉, GPM-MVR이 제2 GPM 파티션에 적용된다).

[0096] 둘째, 두 플래그 gpm_mvr_partIdx0_enable_flag와 플래그 gpm_mvr_partIdx1_enable_flag 모두가 1과 같고(즉, GPM-MVR이 2개의 GPM 파티션에 적용되고) gpm_mvr_partIdx0_direction_idx가 gpm_mvr_partIdx1_direction_idx와 같을(즉, 2개의 GPM 파티션의 MVR이 동일한 방향을 가질) 때, 제1 GPM 파티션의 MVR의 크기(즉, gpm_mvr_partIdx0_distance_idx)가 제2 GPM 파티션의 MVR의 크기(즉, gpm_mvr_partIdx1_distance_idx)를 예측하는 데 사용된다. 구체적으로, gpm_mvr_partIdx1_distance_idx가 gpm_mvr_partIdx0_distance_idx보다 더 작다면, 그 원래의 값이 직접적으로 시그널링된다. 그렇지 않으면(gpm_mvr_partIdx1_distance_idx가 gpm_mvr_partIdx0_distance_idx보다 더 크다면), 그 값은 비트스트림에 시그널링되기 전에 1만큼 감산된다. 디코더 측에서, gpm_mvr_partIdx1_distance_idx의 값을 디코딩하기 위해, 파싱된 값이 gpm_mvr_partIdx0_distance_idx보다 더 작다면, gpm_mvr_partIdx1_distance_idx는 파싱된 값과 동일하게 설정되고; 그렇지 않으면(파싱된 값이 gpm_mvr_partIdx0_distance_idx 이상이라면), gpm_mvr_partIdx1_distance_idx는 파싱된 값에 1을 더한 것과 동일하게 설정된다. 이러한 경우, 오버헤드를 더 감소시키기 위해, 상이한 최대값 MaxGPMMVRDistance - 1 및 MaxGPMMVRDistance - 2가 gpm_mvr_partIdx0_distance_idx 및 gpm_mvr_partIdx1_distance_idx의 이진화를 위해 사용될 수 있으며, MaxGPMMVRDistance는 움직임 벡터 미세조정에 대한 허용된 크기의 수이다.

[0097] 다른 실시예에서, MVR 크기가 MVR 크기에 앞서 시그널링되도록 시그널링 순서를 gpm_mvr_partIdx0_direction_idx/ gpm_mvr_partIdx1_direction_idx 및 gpm_mvr_partIdx0_distance_idx/ gpm_mvr_partIdx1_distance_idx로 스위칭하는 것이 제안된다. 이런 식으로, 앞서 설명된 바와 동일한 로직에 후속하여, 인코더/디코더는 제1 GPM 파티션의 MVR 방향을 사용하여 제2 GPM 파티션의 MVR 방향의 시그널링을 컨디셔닝할 수 있다. 다른 실시예에서, 제2 GPM 파티션의 MVR 크기 및 방향을 먼저 시그널링하고 이를 사용하여 제2 GPM 파티션의 MVR 크기 및 방향의 시그널링을 컨디셔닝하는 것이 제안된다.

[0098] 다른 실시예에서, 기존의 GPM 신택스 엘리먼트의 시그널링 전에 GPM-MVR 관련 신택스 엘리먼트를 시그널링하는 것이 제안된다. 구체적으로, 이러한 설계에서, 2개의 플래그 gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag가 먼저, GPM-MVR이 제1 GPM 파티션 및 제2 GPM 파티션에 각각 적용되는지 여부를 표시하도록 시그널링한다. 하나의 GPM 파티션의 플래그가 1과 같은 경우, (신택스 엘리먼트 gpm_mvr_partIdx0_distance_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) 거리 인덱스 및 (신택스 엘리먼트 gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) 방향 인덱스가 MVR의 방향을 특정하도록 시그널링된다. 그 후, 기존의 신택스 syntax merge_gpm_idx0 및 merge_gpm_idx1이 2개의 GPM 파티션에 대한 단방향 MV, 즉 기본 MV를 식별하도록 시그널링된다. 표 5는 제안된 GPM-MVR 시그널링 방식을 예시한다.

[0099] **표 5** 2개의 GPM 파티션에 대한 별개의 MVR을 이용한 제안된 GPM-MVR 방법(방법 2)의 신택스 엘리먼트

[0100]

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
……	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
gpm_mvr_partIdx0_enable_flag[x0][y0]	ae(v)
if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_directoin_idx[x0][y0]	ae(v)
gpm_mvr_partIdx0_distance_idx[x0][y0]	ae(v)
}	
gpm_mvr_partIdx1_enable_flag[x0][y0]	ae(v)
if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_direction_idx[x0][y0]	ae(v)
gpm_mvr_partIdx1_distance_idx[x0][y0]	ae(v)
}	
}	
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
……	
}	

[0101]

[0102] 표 4의 시그널링 방법과 유사하게, 2개의 GPM 파티션의 예측에 사용되는 결과적인 MV가 동일하지 않음을 보장하기 위해 표 5의 GPM-MVR 시그널링 방법이 적용될 때, 특정 조건이 적용될 수 있다. 구체적으로, 제1 GPM 파티션 및 제2 GPM 파티션에 적용되는 MVR의 값에 의존하여 단방향 예측 병합 인덱스 merge_gpm_idx0 및 merge_gpm_idx1의 시그널링을 제한하기 위해 다음의 조건이 제안된다:

[0103] 먼저, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 0과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 둘 모두에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값은 동일할 수 없다;

[0104] 둘째, gpm_mvr_partIdx0_enable_flag가 1과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 인에이블되고) gpm_mvr_partIdx1_enable_flag가 0과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0105] 셋째, gpm_mvr_partIdx0_enable_flag가 0과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 디세이블되고) gpm_mvr_partIdx1_enable_flag가 1과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0106] 넷째, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 1과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 모두에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되는지 여부에 대한 결정은 2개의 GPM 파티션에 적용되는 (gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx0_distance_idx, 그리고 gpm_mvr_partIdx1_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) MVR의 값에 의존한다. 2개의 MVR의 값이 같다면, merge_gpm_idx0 및 merge_gpm_idx1은 동일한 것이 허용되지 않는다. 그렇지 않으면(2개의 MVR의 값이 동일하지 않다면), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0107] 위의 4개의 경우들에서, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되지 않을 때, 하나의 파티션의 인덱스 값은 다른 파티션의 인덱스 값에 대한 예측자로서 사용될 수 있다. 하나의 방법에서, merge_gpm_idx0을 먼저 시그널링하고 그의 값을 사용하여 merge_gpm_idx1을 예측하는 것이 제안된다. 구체적으로, 인코더에서, merge_gpm_idx1이 merge_gpm_idx0보다 클 때, 디코더에 송신되는 merge_gpm_idx1의 값은 1만큼 감소된다. 디코더에서, merge_gpm_idx1의 수신된 값이 merge_gpm_idx0의 수신된 값 이상일 때, merge_gpm_idx1의 값은 1만큼 증가된다. 다른 방법에서, merge_gpm_idx1을 먼저 시그널링하고 그의 값을 사용하여 merge_gpm_idx0을 예측하는 것이 제안된다. 따라서 이러한 경우, 인코더에서, merge_gpm_idx0이 merge_gpm_idx1보다 클 때, 디코더에 송신되는 merge_gpm_idx0의 값은 1만큼 감소된다. 디코더에서, merge_gpm_idx0의 수신된 값이 merge_gpm_idx1의 수신된 값 이상일 때, merge_gpm_idx0의 값은 1만큼 증가된다. 추가로, 기존의 GPM 시그널링 설계와 유사하게, 상이한 최대값 MaxGPMMergeCand - 1 및 MaxGPMMergeCand - 2가 시그널링 순서에 따라 제1 인덱스 값 및 제2 인덱스 값의 이진화를 위해 각각 사용될 수 있다. 다른 한편으로, 2개의 인덱스 값 사이에 어떠한 상관도 없기 때문에 merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용될 때, 동일한 최대값 MaxGPMMergeCand - 1이 2개의 인덱스 값 모두의 이진화에 사용된다.

[0108] 위의 방법에서, 시그널링 비용을 감소시키기 위해, merge_gpm_idx0 및 merge_gpm_idx1의 이진화를 위해 상이한 최대값이 적용될 수 있다. 대응하는 최대값의 선택은 (gpm_mvr_partIdx0_enable, gpm_mvr_partIdx1_enable, gpm_mvr_partIdx0_direction_idx, gpm_mvr_partIdx1_direction_idx, gpm_mvr_partIdx0_distance_idx 및 gpm_mvr_partIdx1_distance_idx_에 의해 표시된 바와 같은) MVR의 디코딩된 값에 의존한다. 그러한 설계는 상이한 GPM 신택스 엘리먼트 사이에 바람직하지 않은 파싱 의존성을 도입하며, 이는 전반에 걸쳐 전체 파싱에 영향을 미칠 수 있다. 이러한 문제를 해결하기 위해, 일 실시예에서, merge_gpm_idx0 및 merge_gpm_idx1의 값을 파싱하기 위해 항상 하나의 동일한 최대 값(예컨대, MaxGPMMergeCand - 1)을 사용하는 것이 제안된다. 이러한 방법이 사용될 때, 2개의 GPM 파티션의 2개의 디코딩된 MV가 동일한 것을 방지하기 위해 하나의 비트스트림 적합성 제약이 사용될 수 있다. 다른 방법에서, 이는 또한, 2개의 GPM 파티션의 디코딩된 MV가 동일한 것이 허용되도록 이러한 비-아이덴티티 제약을 제거할 수 있다. 다른 한편으로, 그러한 방법이 적용될 때(즉, merge_gpm_idx0 및 merge_gpm_idx1에 대해 동일한 최대값을 사용할 때), merge_gpm_idx0/merge_gpm_idx1과 다른 GPM-MVR 신택스 엘리먼트 사이에 어떠한 파싱 의존성도 존재하지 않는다. 따라서 그러한 신택스 엘리먼트를 시그널링하는 순서는 더 이상 중요하지 않다. 일례로, merge_gpm_idx0/merge_gpm_idx1의 시그널링을 gpm_mvr_partIdx0_enable, gpm_mvr_partIdx1_enable, gpm_mvr_partIdx0_direction_idx, gpm_mvr_partIdx1_direction_idx, gpm_mvr_partIdx0_distance_idx 및 gpm_mvr_partIdx1_distance_idx의 시그널링의 앞으로 이동시키는 것이 제안된다.

[0109] **대칭 움직임 벡터 미세조정을 이용한 기하학적 파티션 모드**

[0110] 위에서 논의된 GPM-MVR 방법의 경우, 2개의 별개의 MVR 값이 시그널링되는데, 하나의 MVR 값은 단 하나의 GPM 파티션의 기본 MV를 개선하기 위해 적용된다. 이러한 방법은 각각의 GPM 파티션에 대한 독립적인 움직임 미세조정을 가능하게 함으로써 예측 정확도의 개선과 관련하여 효율적일 수 있다. 그러나 이러한 유연한 움직임 미세조정은, 2개의 상이한 세트의 GMP-MVR 신택스 엘리먼트가 인코더로부터 디코더로 송신될 필요가 있다면, 시그널링 오버헤드를 증가시키는 대가를 치른다. 시그널링 오버헤드를 감소시키기 위해, 이 섹션에서, 대칭 움직임 벡터 미세조정을 이용한 하나의 기하학적 파티션 모드가 제안된다. 구체적으로, 이 방법에서, 하나의 단일 MVR 값이 하나의 GPM CU에 대해 시그널링되고, 현재 픽처의 픽처 순서 카운트(POC: picture order count) 값과 2개의 GPM 파티션과 연관된 참조 픽처 간의 대칭 관계에 따라 2개의 GPM 파티션 모두에 대해 사용된다. 표 6은 제안된 방법이 적용될 때의 신택스 엘리먼트를 예시한다.

[0111] **표 6** 2개의 GPM 파티션에 대한 대칭 MVR을 이용한 제안된 GPM-MVR 방법(방법 1)의 신택스 엘리먼트

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
.....	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
gpm_mvr_enable_flag[x0][y0]	ae(v)
if(gpm_mvr_enable_flag[x0][y0]) {	
gpm_mvr_direction_idx[x0][y0]	ae(v)
gpm_mvr_distance_idx[x0][y0]	ae(v)
}	
.....	
}	

[0112]

[0113] 표 6에 도시된 바와 같이, (merge_gpm_idx0 및 merge_gpm_idx1에 기초하여) 2개의 GPM 파티션의 기본 MV가 선택된 후에, GPM-MVR 모드가 현재 GPM CU에 적용되는지 여부를 표시하도록 하나의 플래그 gpm_mvr_enable_flag가 시그널링된다. 플래그가 1과 같을 때, 이는 2개의 GPM 파티션의 기본 MV를 향상시키기 위해 움직임 미세조정이 적용됨을 표시한다. 그렇지 않으면(플래그가 0과 같을 때), 이는 2개의 파티션 중 어느 것에도 움직임 미세조정이 적용되지 않음을 표시한다. GPM-MVR 모드가 인에이블된다면, 적용된 MVR의 값을 방향 인덱스 gpm_mvr_direction_idx 및 크기 인덱스 gpm_mvr_distance_idx에 의해 특정하도록 추가 신택스 엘리먼트가 추가로 시그널링된다. 추가로, MMVD 모드와 유사하게, MVR 부호의 의미는 현재 픽처의 POC와 GPM 파티션의 2개의 참조 픽처 간의 관계에 따라 변형될 수 있다. 구체적으로, 2개의 참조 픽처의 두 POC 모두가 현재 픽처의 POC보다 크거나 작을 때, 시그널링된 부호는 2개의 기본 MV 모두에 부가되는 MVR의 부호이다. 그렇지 않으면(하나의 참조 픽처의 POC는 현재 픽처보다 큰 한편, 다른 참조 픽처의 POC는 현재 픽처보다 작은 경우), 시그널링된 부호가 제1 GPM 파티션의 MVR에 적용되고 반대 부호가 제2 GPM 파티션에 적용된다. 표 6에서, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0114] 다른 실시예에서, 별개로 2개의 GPM 파티션에 대한 GPM-MVR 모드의 인에이블/디세이블을 별개로 제어하기 위해 2개의 상이한 플래그를 시그널링하는 것이 제안된다. 그러나 GPM-MVR 모드가 인에이블될 때, 신택스 엘리먼트 gpm_mvr_direction_idx 및 gpm_mvr_distance_idx에 기반하여 단 하나의 MVR만이 시그널링된다. 이러한 시그널링 방법의 대응하는 신택스 표는 표 7에 예시된다.

[0115] **표 7** 2개의 GPM 파티션에 대한 대칭 MVR을 이용한 제안된 GPM-MVR 방법(방법 2)의 신택스 엘리먼트

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
.....	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
gpm_mvr_partIdx0_enable_flag[x0][y0]	ae(v)
if(merge_gpm_idx0 != merge_gpm_idx1 || gpm_mvr_partIdx0_enable_flag)	
gpm_mvr_partIdx1_enable_flag[x0][y0]	ae(v)
if(gpm_mvr_partIdx0_enable_flag[x0][y0] || gpm_mvr_partIdx1 _enable_flag[x0][y0]) {	
gpm_mvr_direction_idx[x0][y0]	ae(v)
gpm_mvr_distance_idx[x0][y0]	ae(v)
}	
.....	
}	

[0116]

[0117] 표 7의 시그널링 방법이 적용될 때, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다. 그러나 2개의 GPM 파티션에 적용된 결과적인 MV가 중복이 아님을 보장하기 위해, 플래그 gpm_mvr_partIdx0_enable_flag가 0과 같을 때(즉, GPM-MVR이 제1 GPM 파티션에 적용되지 않음), 플래그 gpm_mvr_partIdx1_enable_flag가 시그널링되는 것이 아니라, 1인 것으로 추론된다(즉, GPM-MVR이 제2 GPM 파티션에 적용된다).

[0118] **GPM-MVR에 대한 허용된 MVR의 적응**

[0119] 위에서 논의된 GPM-MVR 방법에서, MVR 값의 하나의 고정된 그룹이 하나의 비디오 시퀀스에서 인코더와 디코더 모두에서의 GPM CU에 대해 사용된다. 그러한 설계는 고해상도를 갖는 또는 격렬한 움직임을 갖는 비디오 콘텐츠에 대해 차선이다. 그러한 경우에, MV는 훨씬 큰 경향이 있어, 고정된 MVR 값은 그러한 블록의 실제 움직임을 캡처하기에 최적이 아닐 수 있다. GPM-MVR 모드의 코딩 성능을 더 개선하기 위해, 본 개시내용에서는, 시퀀스 레벨, 픽처/슬라이스 픽처, 코딩 블록 그룹 레벨 등과 같은 다양한 코딩 레벨에서 GPM-MVR 모드에 의해 선택되는 것이 허용되는 MVR 값의 적응을 지원하는 것이 제안된다. 예를 들어, 다수의 MVR 세트뿐만 아니라 대응하는 코드워드가 상이한 비디오 시퀀스의 특정 움직임 특징에 따라 오프라인으로 도출될 수 있다. 인코더는 최상의 MVR 세트를 선택하고, 선택된 세트의 대응하는 인덱스를 디코더에 시그널링할 수 있다.

[0120] 본 개시내용의 특정 실시예에서, 8개의 오프셋 크기(즉, 1/4, 1/2, 1, 2, 4, 8, 16 및 32 픽셀) 및 4개의 MVR 방향(즉, +/- x 축 및 y 축)을 포함하는 디폴트 MVR 오프셋 외에도, 아래의 표에 정의된 바와 같은 다른 MVR 오프셋이 GPM-MVR 모드에 대해 제안된다.

[0121] 표 15

거리 IDX	0	1	2	3	4	5	6	7	8
오프셋(루마 샘플 단위)	1/4	1/2	1	2	3	4	6	8	16

[0122]

[0123] 표 16

방향 IDX	000	001	010	011	100	101	110	111
x 축	+1	–1	0	0	+1/2	-1/2	-1/2	+1/2
y 축	0	0	+1	–1	+1/2	-1/2	+1/2	-1/2

[0124]

[0125] 위의 표 15 및 표 16에서, x 축 및 y 축의 +1/2 값 및 -1/2 값은 수평 방향 및 수직 방향의 대각선 방향(+45° 및 -45°)을 표시한다. 표 15 및 표 16에 도시된 바와 같이, 기존의 MVR 오프셋 세트와 비교하여, 제2 MVR 오프셋 세트는 2개의 새로운 오프셋 크기(즉, 3 픽셀 및 6 픽셀) 및 4개의 오프셋 방향(45° , 135° , 225° 및 315°)을 도입한다. 새롭게 추가된 MVR 오프셋은 제2 MVR 오프셋 세트를 정교한 움직임을 갖는 비디오 블록을 코딩하기에 더 적합하게 만든다. 추가로, 2개의 MVR 오프셋 세트 사이의 적응적 스위칭을 가능하게 하기 위해, MVR 오프셋의 어떤 세트가 코딩 레벨 하에서 적용된 GPM-MVR 모드에 대해 선택되는지를 표시하도록 하나의 특정 코딩 레벨(예컨대, 시퀀스, 픽처, 슬라이스, CTU 및 코딩 블록 등)에서 시그널링하기 위한 하나의 제어 플래그가 제안된다. 제안된 적응이 픽처 레벨에서 실행된다고 가정하면, 하기 표 17은 픽처 헤더에서 시그널링된 대응하는 신택스 엘리먼트를 예시한다.

[0126] 표 17

pic ture_head er_struc ture( ) {	De scriptor
.....	
if(sps_mmvd_full pel_on ly_en a bled_flag)	
ph_mmvd_full pel_on ly_flag	u(1)
pres enceFlag = 0	
if(!pps_rpl_in fo_in_ph_flag) /* This con di tion is in ten tion al ly not merged in to the next, to avoid pos si ble in ter pre ta tion of RplsIdx[i] not hav ing a spec i fied val ue. */	
pres enceFlag = 1	
else if(num_ref_en tries[1][RplsIdx[1]] > 0)	
pres enceFlag = 1	
if(pres enceFlag) {	
ph_mvd_ll_zero_flag	u(1)
if(sps_bdof_con trol_pre sent_in_ph_flag)	
ph_bdof_dis a bled_flag	u(1)
if(sps_dmvr_con trol_pre sent_in_ph_flag)	
ph_dmvr_dis a bled_flag	u(1)
}	
if(!pps_rpl_in fo_in_ph_flag || num_ref_en tries[1][RplsIdx[1]] > 0)	
ph_gpm_mvr_off set_set_flag	u(1)
if(sps_prof_con trol_pre sent_in_ph_flag)	
ph_prof_dis a bled_flag	u(1)
.....	
}	

[0127]

[0128] 위의 표 17에서, 픽처에 대해 사용되는 대응하는 GPM MVR 오프셋의 선택을 표시하기 위해 새로운 플래그 ph_gpm_mvr_offset_set_flag가 사용된다. 플래그가 0과 같을 때, 이는 디폴트 MVR 오프셋(즉, 1/4, 1/2, 1, 2, 4, 8, 16 및 32 픽셀의 크기 및 4개의 MVR 방향 +/- x 축 및 y 축)이 픽처에서 GPM-MVR 모드에 적용됨을 의미한다. 그렇지 않으면, 플래그가 1과 같을 때, 이는 제2 MVR 오프셋(즉, 1/4, 1/2, 1, 2, 3, 4, 6, 8, 16 픽셀의 크기 및 8개의 MVR 방향 +/- x 축, y 축 및 45° , 135° , 225° 및 315°)이 픽처에서 GPM-MVR 모드에 적용됨을 의미한다.

[0129]]MVR 오프셋을 시그널링하기 위해, 상이한 방법이 적용될 수 있다. 첫째로, MVR 방향이 일반적으로 통계적으로 균등하게 분포된다는 것을 고려하면, MVR 방향을 이진화하기 위해 고정 길이 코드워드를 사용하는 것이 제안된다. 디폴트 MVR 오프셋을 예로서 취하면, 총 4개의 방향이 있으며, 00, 01, 10 및 11의 코드워드가 4개의 방향을 표현하는 데 사용될 수 있다. 다른 한편으로, MVR 오프셋 크기는 비디오 콘텐츠의 특정 움직임 특징에 적응되는 가변 분포들을 가질 수 있기 때문에, MVR 크기를 이진화하기 위해 가변 길이 코드워드를 사용하는 것이 제안된다. 아래의 표 18은 디폴트 MVR 오프셋 세트 및 제2 MVR 오프셋 세트의 MVR 크기의 이진화를 위해 사용될 수 있는 하나의 특정 코드워드 표를 보여준다.

[0130] 표 18

디폴트 MVR 오프셋 세트		제 2 MVR 오프셋 세트	
MVR 오프셋	이진화	MVR 오프셋	이진화
1/4 픽셀	001	1/4 픽셀	001
1/2 픽셀	1	1/2 픽셀	1
1 픽셀	01	1 픽셀	01
2 픽셀	0001	2 픽셀	0001
4 픽셀	00001	3 픽셀	00001
8 픽셀	000001	4 픽셀	000001
16 픽셀	0000001	6 픽셀	0000001
32 픽셀	0000000	8 픽셀	00000001
		16 픽셀	00000000

[0131]

[0132] 다른 실시예에서, 디폴트 MVR 오프셋 세트와 제2 MVR 오프셋 세트의 MVR 오프셋 크기를 이진화하기 위해 상이한 고정 길이 가변 코드워드가 또한 적용될 수 있는데, 예컨대 컨텍스트 적응 2진 산술 코딩(CABAC: context-adaptive binary arithmetic coding) 엔진의 다양한 0/1 통계에 적응하기 위해 빈들 "0"과 "1"이 교환될 수 있다.

[0133] 하나의 특정 예에서, MVR 크기의 값을 이진화하기 위해 2개의 상이한 코드워드 표가 제공된다. 아래의 표는 제1 코드워드 표 및 제2 코드워드 표에 적용되는 디폴트 MVR 오프셋 세트 및 제2 MVR 오프셋 세트의 대응하는 코드워드를 예시한다. 표 19는 제1 코드워드 표 내의 MVR 오프셋 크기의 코드워드를 도시한다. 표 20은 제2 코드워드 표 내의 MVR 오프셋 크기의 코드워드를 도시한다.

[0134] 표 19

디폴트 MVR 오프셋 세트		제 2 MVR 오프셋 세트	
MVR 오프셋	이진화	MVR 오프셋	이진화
1/4 픽셀	1	1/4 픽셀	1
1/2 픽셀	10	1/2 픽셀	10
1 픽셀	110	1 픽셀	110
2 픽셀	1110	2 픽셀	1110
4 픽셀	11110	3 픽셀	11110
8 픽셀	111110	4 픽셀	111110
16 픽셀	1111110	6 픽셀	1111110
32 픽셀	1111111	8 픽셀	11111110
		16 픽셀	11111111

[0135]

[0136] 표 20

디폴트 MVR 오프셋 세트		제 2 MVR 오프셋 세트	
MVR 오프셋	이진화	MVR 오프셋	이진화
1/4 픽셀	111110	1/4 픽셀	111110
1/2 픽셀	1	1/2 픽셀	1
1 픽셀	10	1 픽셀	10
2 픽셀	110	2 픽셀	110
4 픽셀	1110	3 픽셀	1110
8 픽셀	11110	4 픽셀	11110
16 픽셀	1111110	6 픽셀	1111110
32 픽셀	1111111	8 픽셀	11111110
		16 픽셀	11111111

[0137]

[0138] 2개의 코드워드 표 사이의 적응형 스위칭을 가능하게 하기 위해, 코딩 레벨 하에서 MVR 크기를 이진화하기 위해 어느 코드워드 표가 사용되는지를 특정하도록 하나의 특정 코딩 레벨(예컨대, 시퀀스, 픽처, 슬라이스, CTU 및 코딩 블록 등)에서 시그널링하기 위한 하나의 표시 플래그가 제안된다. 제안된 적응이 픽처 레벨에서 수행된다고 가정하면, 아래의 표 21은 픽처 헤더에서 시그널링된 대응하는 신택스 엘리먼트를 예시하며, 여기서 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다.

[0139] 표 21

pic ture_head er_struc ture( ) {	De scrip tor
.....	
if(sps_mmvd_full pel_on ly_en a bled_flag)	
ph_mmvd_full pel_on ly_flag	u(1)
pres enceFlag = 0	
if(!pps_rpl_in fo_in_ph_flag) /* This con di tion is in ten tion al ly not merged in to the next, to avoid pos si ble in ter pre ta tion of RplsIdx[i] not hav ing a spec i fied val ue. */	
pres enceFlag = 1	
else if(num_ref_en tries[1][RplsIdx[1]] > 0)	
pres enceFlag = 1	
if(pres enceFlag) {	
ph_mvd_ll_zero_flag	u(1)
if(sps_bdof_con trol_pre sent_in_ph_flag)	
ph_bdof_dis a bled_flag	u(1)
if(sps_dmvr_con trol_pre sent_in_ph_flag)	
ph_dmvr_dis a bled_flag	u(1)
}	
if(!pps_rpl_in fo_in_ph_flag || num_ref_en tries[1][RplsIdx[1]] > 0) {	
ph_gpm_mvr_off set_set_flag	***u(1)***
ph_gpm_mvr_step_codeword_flag	***u(1)***
}	
if(sps_prof_con trol_pre sent_in_ph_flag)	
ph_prof_dis a bled_flag	u(1)
.....	
}	

[0140]

[0141] 위의 신택스 표에서, 새로운 플래그 ph_gpm_mvr_step_codeword_flag는 픽처의 MVR 크기의 이진화에 사용되는 대응하는 코드워드 표의 선택을 표시하기 위해 사용된다. 플래그가 0과 같을 때, 이는 제1 코드워드 표가 픽처에 대해 적용됨을 표시하고; 그렇지 않으면(즉, 플래그가 1과 같다면), 이는 제2 코드워드 표가 픽처에 대해 적용됨을 표시한다.

[0142] 다른 실시예에서, 전체 비디오 시퀀스의 인코딩/디코딩 동안 MVR 오프셋 크기를 이진화하기 위해 항상 하나의 코드워드 표를 사용하는 것이 제안된다. 일례로, MVR 크기의 이진화를 위해 제1 코드워드 표를 항상 사용하는 것이 제안된다. 다른 예에서, MVR 크기의 이진화를 위해 제2 코드워드 표를 항상 사용하는 것이 제안된다. 다른 방법에서, 모든 MVR 크기의 이진화를 위해 하나의 고정된 코드워드 표(예컨대, 제2 코드워드 표)을 사용하는 것이 제안된다.

[0143] 다른 방법에서, 시그널링 없이 그때그때(on-the-fly) MVR 오프셋 크기에 대한 최적의 코드워드를 적응적으로 설계하기 위해 하나의 통계 기반 이진화 방법이 적용될 수 있다. 최적의 코드워드를 결정하기 위해 사용되는 통계는 다수의 이전에 코딩된 픽처, 슬라이스 및/또는 코딩 블록 상에서 수집되는 MVR 오프셋 크기의 확률 분포일 수 있지만 이에 제한되는 것은 아닐 수 있다. 코드워드는 다양한 주파수 레벨에서 재결정/업데이트될 수 있다. 예를 들어, 업데이트는 CU가 GPM-MVR 모드에서 코딩될 때마다 수행될 수 있다. 다른 예에서, 업데이트는, GPM-MVR 모드에서 코딩된 다수, 예컨대 8 또는 16개의 CU가 존재할 때마다 재결정 및/또는 업데이트될 수 있다.

[0144] 다른 방법에서, 코드워드의 하나의 새로운 세트를 재설계하는 대신에, 제안된 통계 기반 방법은 또한, 더 많이 사용되는 크기에 더 짧은 코드워드를 할당하고 덜 사용되는 크기에 더 긴 코드워드를 할당하기 위해 동일한 세트의 코드워드에 기초하여 MVR 크기 값을 재정렬하기 위해 사용될 수 있다. 다음 표를 예로서 취하여, 통계가 픽처 레벨에서 수집된다고 가정하면, "사용량" 열은 이전에 코딩된 픽처에서 GPM-MVR 코딩 블록에 의해 사용되

는 상이한 MVR 오프셋 크기의 대응하는 퍼센티지를 표시한다. "사용량" 열 내의 값에 따르면, 동일한 이진화 방법(즉, 절단된 단항 코드워드)을 사용하여, 인코더/디코더는 이들의 사용량을 기초로 MVR 크기 값을 정렬할 수 있고; 그 후, 인코더/디코더는 가장 짧은 코드워드(즉, "1")를 가장 빈번하게 사용되는 MVR 크기(즉, 1 픽셀)에 할당하고, 두 번째로 짧은 코드워드(즉, "01")를 두 번째로 빈번하게 사용되는 MVR 크기(즉, ½ 픽셀)에 할당할 수 있으며, …, 가장 긴 코드워드(즉, "0000001" 및 "0000000")을 2개의 가장 드물게 사용되는 MVR 크기(즉, 16 픽셀 및 32 픽셀)에 할당할 수 있다. 이에 따라, 그러한 재정렬 방식에 의해, 동일한 세트의 코드워드는 MVR 크기의 통계 분포의 동적 변화를 수용하도록 자유롭게 재정렬될 수 있다.

MVR 오프셋	사용량	이진화
1/4 픽셀	15%	001
1/2 픽셀	20%	01
1 픽셀	30%	1
2 픽셀	10%	0001
4 픽셀	9%	00001
8 픽셀	6%	000001
16 픽셀	5%	0000001
32 픽셀	5%	0000000

[0145]

[0146] **GPM-MVR 레이트 왜곡 최적화를 위한 인코더 가속 로직**

[0147] 제안된 GPM-MVR 방식의 경우, 각각의 GPM 파티션에 대한 최적의 MVR을 결정하기 위해, 인코더는 각각의 GPM 파티션의 레이트 왜곡 비용을 여러 번 테스트할 필요가 있을 수 있는데, 이러한 테스트는 각각, 적용되고 있는 MVR 값을 변화시킨다. 이는 GPM 모드의 인코딩 복잡도를 상당히 증가시킬 수 있다. 인코딩 복잡도 문제를 해결하기 위해, 다음의 고속 인코딩 로직이 이 섹션에서 제안된다:

[0148] 첫째, VVC 및 AVS3에 적용되는 쿼드/2진/삼진 트리 블록 파티션 구조로 인해, 하나의 동일한 코딩 블록이 레이트 왜곡 최적화(RDO: rate-distortion optimization) 프로세스 동안 체크될 수 있는데, 이들 각각은 하나의 상이한 파티션 경로를 통해 분할된다. 현재 VTM/HPM 인코더 구현에서, 상이한 블록 파티션 조합을 통해 하나의 동일한 CU가 획득될 때마다, 다른 인터 및 인트라 코딩 모드와 함께 GPM 및 GPM-MVR 모드가 항상 테스트된다. 일반적으로 말하자면, 상이한 파티션 경로에 대해, 하나의 CU의 이웃 블록만이 상이할 수 있지만, 이는 하나의 CU가 선택할 최적의 코딩 모드에 대해 비교적 작은 영향을 가져야 한다. 이러한 고려사항에 기초하여, 적용되는 GPM RDO의 총 수를 감소시키기 위해, 하나의 CU의 RD 비용이 처음으로 체크될 때 GPM 모드가 선택되는지 여부의 결정을 저장하는 것이 제안된다. 그 후, 동일한 CU가 RDO 프로세스에 의해(다른 파티션 경로에 의해) 다시 체크될 때, 처음에 CU에 대해 GPM이 선택되는 경우에만, (GPM-MVR을 포함하는) GPM의 RD 비용이 체크된다. 하나의 CU의 초기 RD 체크에 대해 GPM이 선택되지 않는 경우, 다른 파티션 경로를 통해 동일한 CU가 달성될 때 (GPM-MVR이 없는) GPM만이 테스트된다. 다른 방법에서, 하나의 CU의 초기 RD 체크에 대해 GPM이 선택되지 않는 경우, 다른 파티션 경로를 통해 동일한 CU가 달성될 때 GPM과 GPM-MVR 모두 테스트되지 않는다.

[0149] 둘째, GPM-MVR 모드에 대한 GPM 파티션의 수를 감소시키기 위해, 하나의 CU의 RD 비용이 처음으로 체크될 때 최소 RD 비용 없이 처음 *M*개의 GPM 파티션 모드를 유지하는 것이 제안된다. 그 후, 동일한 CU가 RDO 프로세스에 의해(다른 파티션 경로에 의해) 다시 체크될 때, 그러한 *M*개의 GPM 파티션 모드만이 GPM-MVR 모드에 대해 테스트된다.

[0150] 셋째로, 각각의 GPM 파티션에 대해, 하나의 초기 RDO 프로세스에 대해 테스트되는 GPM 파티션의 수를 감소시키기 위해, 2개의 GPM 파티션에 대해 상이한 단방향 예측 병합 후보를 사용할 때 절대 차 합(SAD) 값을 먼저 계산하는 것이 제안된다. 그런 다음, 하나의 특정 파티션 모드 하에서 각각의 GPM 파티션에 대해, 가장 작은 SAD 값을 갖는 최상의 단방향 예측 병합 후보를 선택하고, 2개의 GPM 파티션에 대한 최상의 단방향 예측 병합 후보

의 SAD 값의 합과 같은 파티션 모드의 대응하는 SAD 값을 계산한다. 그런 다음, 후속 RD 프로세스에 대해, 이전 단계에 대한 최상의 SAD 값을 갖는 처음 *N*개의 파티션 모드만이 GPM-MVR 모드에 대해 테스트된다.

[0151] **명시적 움직임 시그널링을 이용한 기하학적 파티션**

[0152] 이 섹션에서는, GPM 모드의 2개의 단방향 MV가 인코더로부터 디코더로 명시적으로 시그널링되는 정규 인터 모드의 양방향 예측으로 GPM 모드를 확장하기 위한 다수의 방법이 제안된다.

[0153] 제1 솔루션(솔루션 1)에서, GPM 모드의 2개의 단방향 MV를 시그널링하기 위해 양방향 예측의 기존의 움직임 시그널링을 완전히 재사용하는 것이 제안된다. 표 8은 제안된 방식의 수정된 신택스 표를 예시하며, 여기서 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다. 표 8에 도시된 바와 같이, 솔루션에서, L0 및 L1 움직임 정보를 시그널링하는 모든 기존의 신택스 엘리먼트는 2개의 GPM 파티션의 단방향 MV를 각각 표시하기 위해 완전히 재사용된다. 추가로, L0 MV는 항상 제1 GPM 파티션과 연관되고 L1 MV는 항상 제2 GPM 파티션과 연관된다고 가정된다. 다른 한편으로, 표 8에서, 인터 예측 신택스, 즉 *inter_pred_idc*의 값이 GPM 플래그(즉, *gpm_flag*)의 존재를 컨디셔닝하는 데 사용될 수 있도록 *inter_pred_idc*가 *gpm_flag*에 앞서 시그널링된다. 구체적으로, 플래그 *gpm_flag*는 *inter_pred_idc*가 PRED_BI(즉, 양방향 예측)와 같고 *inter_affine_flag*와 *sym_mvd_flag* 둘 다 0과 같을 때(즉, CU가 아핀(affine) 모드에 의해서도 SMVD 모드에 의해서도 코딩되지 않을 때)만 시그널링될 필요가 있다. 플래그 *gpm_flag*가 시그널링되지 않을 때, 그 값은 항상 0인 것으로 추론된다(즉, GPM 모드가 디세이블됨). *gpm_flag*가 1일 때, 다른 신택스 엘리먼트 *gpm_partition_idx*는 현재 CU에 대한 (총 64개의 GPM 파티션 중) 선택된 GPM 모드를 표시하도록 추가로 시그널링된다.

[0154] **표 8** 솔루션 1의 움직임 시그널링에 대한 수정된 신택스 표(옵션 1)

if(sh_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_6param_affine_enabled_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
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	
}	

[0155]

if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(ph_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 {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = −MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = −MvdL0[x0][y0][1]	
} 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(inter_pred_idc[x0][y0] = = PRED_BI && !inter_affine_flag[x0][y0] && !sym_mvd_flag[x0][y0] && cbWidth >= 8 && cbHeight >= 8 && cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth) && cbWidth < 128 && cbHeight < 128) {	
gpm_flag[x0][y0]	***ae(v)***
if(gpm_partition_flag[x0][y0]) {	
gpm_partition_idx[x0][y0]	***ae(v)***
}	

[0156]

[0157] 다른 방법에서, 플래그 *gpm_flag*의 값이 다른 인터 신택스 엘리먼트가 존재할 필요가 있는지 여부를 결정하는 데 사용될 수 있도록 *gpm_flag*의 시그널링을 다른 인터 시그널링 신택스 엘리먼트 앞에 배치하는 것이 제안된다. 표 9는 그러한 방법이 적용될 때의 대응하는 신택스 표를 예시하며, 여기서 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다. 알 수 있는 바와 같이, *gpm_flag*가 먼저 표 9에서 시그널링된다. gpm_flag가 1과 같을 때, *inter_pred_idc*, *inter_affine_flag* and *sym_mvd_flag*의 대응하는 시그널링이 우회될 수 있다. 대신에, 3개의 신택스 엘리먼트의 대응하는 값은 각각 PRED_BI, 0 및 0으로서 추론될 수 있다.

[0158] **표 9** 솔루션 1의 움직임 시그널링에 대한 수정된 신택스 표(옵션 2)

if(sh_slice_type == B && cbWidth >= 8 && cbHeight >= 8 && cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth) && cbWidth < 128 && cbHeight < 128) {	
gpm_flag[x0][y0]	***ae(v)***
if(gpm_partition_flag[x0][y0]) {	
gpm_partition_idx[x0][y0]	***ae(v)***
}	
if(sh_slice_type == B && ***!gpm_flag[x0][y0]***)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag ***&& !gpm_flag[x0][y0]*** && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_6param_affine_enabled_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI ***&& !gpm_flag[x0][y0]*** && !inter_affine_flag[x0][y0] && RefIdxSymL0 > −1 && RefIdxSymL1 > −1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
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	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	

[0159]

if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(ph_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 {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = −MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = −MvdL0[x0][y0][1]	
} 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	
}	

[0160]

[0161] 표 8과 표 9 모두에서, SMVD 모드는 GPM 모드와 조합될 수 없다. 다른 예에서, 현재 CU가 GPM 모드에 의해 코딩될 때 SMVD 모드를 허용하는 것이 제안된다. 이러한 조합이 허용될 때, SMVD의 동일한 설계를 따름으로써, 2개의 GPM 파티션의 MVD는 대칭적인 것으로 가정되어, 제1 GPM 파티션의 MVD만이 시그널링될 필요가 있고 제2 GPM 파티션의 MVD는 제1 MVD에 대해 항상 대칭적이다. 이러한 방법이 적용될 때, *gpm_flag*에 대한 *sym_mvd_flag*의 대응하는 시그널링 조건이 제거될 수 있다.

[0162] 위에서 예시된 바와 같이, 제1 솔루션에서는, L0 MV가 제1 GPM 파티션에 대해 사용되고 L1 MV가 제2 GPM 파티션에 대해 사용될 것으로 항상 가정한다. 이러한 설계는, 이러한 방법이 2개의 GPM 파티션의 MV가 하나의 동일한 예측 리스트(L0 또는 L1)로부터 오는 것을 금지한다는 점에서 최적이 아닐 수 있다. 이러한 문제를 해결하기 위해, 하나의 대안적인 GPM-EMS 방식인 솔루션 2가 표 10에 예시된 바와 같은 시그널링 설계로 제안된다. 표 10에서, 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다. 표 10에 도시된 바와 같이, 플래그 gpm_flag가 먼저 시그널링된다. 플래그가 1과 같을 때(즉, GPM이 인에이블될 때), 선택된 GPM 모드를 특정하기 위해 신택스 gpm_partition_idx가 시그널링된다. 그런 다음, 제1 GPM 파티션의 MV가 나오는 대응하는 예측 리스트를 표시하기 위해 하나의 추가 플래그 gpm_pred_dir_flag0이 시그널링된다. 플래그 gpm_pred_dir_flag0이 1과 같을 때, 이는 제1 GPM 파티션의 MV가 L1에서 나옴을 표시하고; 그렇지 않으면(플래그가 0과 같으면), 이는 제1 GPM 파티션의 MV가 L0에서 나옴을 표시한다. 그 후, 기존의 신택스 엘리먼트 ref_idx_l0, mvp_l0_flag 및 mvd_coding()은 제1 GPM 파티션의 참조 픽처 인덱스, mvp 인덱스 및 MVD의 값을 시그널링하기 위해 이용된다. 다른 한편으로, 제1 파티션과 유사하게, 제2 GPM 파티션의 대응하는 예측 리스트를 선택하기 위해 다른 신택스 엘리먼트 gpm_pred_dir_flag1이 도입되고, 그 다음에 제2 GPM 파티션의 MV를 도출하기 위해 사용될 기존의 신택스 엘리먼트 ref_idx_l1, mvp_l1_flag 및 mvd_coding()이 이어진다.

[0163] **표 10** 솔루션 2의 움직임 시그널링에 대한 수정된 신택스 표

*if(sh_slice_type == B && cbWidth >= 8 && cbHeight >= 8 && cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth) && cbWidth < 128 && cbHeight < 128) {*	
gpm_flag[x0][y0]	*ae(v)*
if(gpm_partition_flag[x0][y0]) {	
gpm_partition_idx[x0][y0]	*ae(v)*
gpm_pred_dir_flag0[x0][y0]	*ae(v)*
ref_idx_l0[x0][y0]	*ae(v)*
mvd_coding(x0, y0, 0, 0)	
mvp_l0_flag[x0][y0]	*ae(v)*
gpm_pred_dir_flag1[x0][y0]	*ae(v)*
ref_idx_l1[x0][y0]	*ae(v)*
mvd_coding(x0, y0, 0, 0)	
mvp_l1_flag[x0][y0]	*ae(v)*
} else {	
if(sh_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_6param_affine_enabled_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > −1 && RefIdxSymL1 > −1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
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)

[0164]

} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(ph_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 {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} 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	
}	
}	

[0165]

[0166] 마지막으로, GPM 모드가 (분할 에지 상의 혼합 샘플을 제외하고) 2개의 단방향 예측 파티션으로 구성된다면, 양방향 예측을 위해 특별히 설계되는 VVC 및 AVS3에서의 일부 기존의 코딩 툴, 예컨대 양방향 광 흐름, 디코더 측 움직임 벡터 미세조정(DMVR: decoder-side motion vector refinement) 및 CU 가중치를 이용한 양방향 예측(BCW)은 제안된 GPM-EMS 방식이 하나의 인터 CU에 대해 인에이블될 때 자동으로 우회될 수 있다. 예컨대, 제안된 GPM-EMS 중 하나가 하나의 CU에 대해 인에이블될 때, BCW가 GPM 모드에 적용될 수 없다면, 시그널링 오버헤드를 감소시키기 위해 CU에 대해 대응하는 BCW 가중치가 추가로 시그널링될 필요가 없다.

[0167] **GPM-MVR과 GPM-EMS의 조합**

[0168] 이 섹션에서, 하나의 CU와 기하학적 구조 파티션에 대해 GPM-MVR과 GPM-EMS를 조합하는 것이 제안된다. 구체적으로, 2개의 GPM 파티션의 단방향 예측 MV를 시그널링하기 위해 병합 기반 움직임 시그널링 또는 명시적 시그널링 중 하나만이 적용될 수 있는 GPM-MVR 또는 GPM-EMS와 달리, 제안된 방식에서, 이는 1) GPM-MVR 기반 움직임 시그널링을 사용하는 하나의 파티션 및 GPM-EMS 기반 움직임 시그널링을 사용하는 다른 파티션; 또는 2) GPM-MVR 기반 움직임 시그널링을 사용하는 2개의 파티션; 또는 3) GPM-EMS 기반 움직임 시그널링을 사용하는 2개의 파티션; 또는 3} GPM-EMS 기반 움직임 시그널링을 사용하는 2개의 파티션을 가능하게 한다. 표 4의 GPM-MVR 시그널링 및 표 10의 GPM-EMS를 사용하여, 표 11은 제안된 GPM-MVR 및 GPM-EMS가 조합된 후의 대응하는 신택스 표를 보여준다. 표 11에서, 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다. 표 11에 도시된 바와 같이, 2개의 추가 신택스 엘리먼트 gpm_merge_flag0 및 gpm_merge_flag1이 파티션 #1 및 파티션 #2에 대해 각각 도입

되며, 이는 대응하는 파티션이 GPM-MVR 기반 병합 시그널링 또는 GPM-EMS 기반 명시적 시그널링을 사용하는 것을 특정한다. 플래그가 1인 경우, 이는 GPM 단방향 예측 움직임이 merge_gpm_idxX, gpm_mvr_partIdxX_enabled_flag, gpm_mvr_partIdxX_direction_idx 및 gpm_mvr_partIdxX_distance_idx를 통해 시그널링될 파티션에 대해 GPM-MVR 기본 시그널링이 인에이블된다는 것을 의미하며, 여기서 X = 0, 1이다. 그렇지 않으면, 플래그가 0이라면, 이는 파티션의 단방향 예측 움직임이 신택스 엘리먼트 gpm_pred_dir_flagX, ref_idx_lX, mvp_lX_flag 및 mvd_lX를 사용하여 GPM-EMS 방식에 의해 명시적으로 시그널링될 것임을 의미하며, 여기서 X = 0, 1이다.

[0169] **표 11** GPM-MVR과 GPM-EMS의 조합을 이용하는 GPM 모드에 대한 제안된 신택스 표

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
……	
if(!ciip_flag[x0][y0]) {	
gpm_partition_idx[x0][y0]	***ae(v)***
gpm_merge_flag0[x0][y0]	***ae(v)***
if(gpm_merge_flag0[x0][y0]) {	
merge_gpm_idx0[x0][y0]	***ae(v)***
gpm_mvr_partIdx0_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_directoin_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx0_distance_idx[x0][y0]	***ae(v)***
}	
}else{	
gpm_pred_dir_flag0[x0][y0]	***ae(v)***
ref_idx_l0[x0][y0]	***ae(v)***
mvd_coding(x0, y0, 0, 0)	
mvp_l0_flag[x0][y0]	***ae(v)***
}	
gpm_merge_flag1[x0][y0]	***ae(v)***
if(gpm_merge_flag1[x0][y0]) {	
merge_gpm_idx1[x0][y0]	***ae(v)***
gpm_mvr_partIdx1_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_directoin_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx1_distance_idx[x0][y0]	***ae(v)***
}	
}else{	
gpm_pred_dir_flag1[x0][y0]	***ae(v)***
ref_idx_l1[x0][y0]	***ae(v)***
mvd_coding(x0, y0, 0, 0)	
mvp_l1_flag[x0][y0]	***ae(v)***
}	
……	
}	

[0170]

[0171] **템플릿 매칭과 GPM-MVR의 조합**

[0172] 이 섹션에서, 템플릿 매칭과 GPM-MVR을 조합하기 위한 상이한 솔루션이 제공된다.

[0173] 방법 1에서, 하나의 CU가 GPM 모드에서 코딩될 때, 대응하는 파티션의 단방향 움직임이 템플릿 매칭에 의해 추가로 미세조정되는지 여부를 각각 표시하는 2개의 GPM 파티션에 대한 2개의 별개의 플래그를 시그널링하는 것이 제안된다. 플래그가 인에이블될 때, 현재 CU의 좌측 및 최상부 이웃의 재구성된 샘플을 사용하여 템플릿이 생성되고; 그런 다음, "템플릿 매칭" 섹션에서 소개된 것과 동일한 프로시저에 따라 템플릿과 템플릿의 참조 샘플 간의 차이를 최소화함으로써 파티션의 단방향 움직임이 미세조정될 것이다. 그렇지 않으면(플래그가 디세이블

되면), 템플릿 매칭이 파티션에 적용되지 않고, GPM-MVR이 추가로 적용될 수 있다. 예로서 표 5의 GPM-MVR 시그널링 방법을 사용하면, 표 12는 GPM-MVR이 템플릿 매칭과 조합되는 경우의 대응하는 신택스 표를 예시한다. 표 12에서, 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다.

[0174] **표 12** 템플릿 매칭과 GPM-MVR을 조합하는 제안된 방법의 신택스 엘리먼트(방법 1)

[0175]

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
.....	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
gpm_tm_enable_flag0[x0][y0]	***ae(v)***
if(!gpm_tm_enable_flag0) {	
gpm_mvr_partIdx0_enable_flag[x0][y0]	***ae(v)***
}	
if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_direc to in_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx0_distance_idx[x0][y0]	***ae(v)***
}	
gpm_tm_enable_flag1[x0][y0]	***ae(v)***
if(!gpm_tm_enable_flag1) {	
gpm_mvr_partIdx1_enable_flag[x0][y0]	***ae(v)***
}	
if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_direction_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx1_distance_idx[x0][y0]	***ae(v)***
}	
}	
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
.....	
}	

[0176]

[0177] 표 12에 도시된 바와 같이, 제안된 방식에서, 2개의 GPM 파티션에 대해 움직임이 미세조정되는지 여부를 표시하기 위해 2개의 추가 플래그 gpm_tm_enable_flag0 및 gpm_tm_enable_flag1이 먼저 각각 시그널링된다. 플래그가 1일 때, 이는, TM이 하나의 파티션의 단방향 MV를 개선하기 위해 적용됨을 표시한다. 플래그가 0일 때, GPM-MVR이 GPM 파티션에 적용되는지 여부를 각각 표시하기 위해 하나의 플래그(gpm_mvr_partIdx0_enable_flag 또는 gpm_mvr_partIdx1_enable_flag)가 추가로 시그널링된다. 하나의 GPM 파티션의 플래그가 1과 같은 경우, (신택스 엘리먼트 gpm_mvr_partIdx0_distance_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) 거리 인덱스 및 (신택스 엘리먼트 gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) 방향 인덱스가 MVR의 방향을 특정하도록 시그널링된다. 그 후, 2개의 GPM 파티션에 대한 단방향 MV를 식별하도록 기존의 신택스 merge_gpm_idx0 및 merge_gpm_idx1이 시그널링된다. 한편, 표 5에 적용되는 시그널링 조건과 유사하게, 2개의 GPM 파티션의 예측에 사용되는 결과적인 MV가 동일하지 않음을 보장하기 위해 다음의 조건이 적용될 수 있다.

[0178] 먼저, gpm_tm_enable_flag0 및 gpm_tm_enable_flag1의 두 값 모두가 1과 같을 때(즉, TM이 2개의 GPM 파티션 모두에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값은 동일할 수 없다;

[0179] 둘째로, gpm_tm_enable_flag0 및 gpm_tm_enable_flag1 중 하나가 1이고 다른 하나가 0일 때, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0180] 그렇지 않으면, 즉 gpm_tm_enable_flag0과 gpm_tm_enable_flag1 모두가 1과 같다면: 첫째, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 0과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 모두에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일할 수 없고; 둘

째, gpm_mvr_partIdx0_enable_flag가 1과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 인에이블되고) 그리고 gpm_mvr_partIdx1_enable_flag가 0과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되고; 셋째, gpm_mvr_partIdx0_enable_flag가 0과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 디세이블되고) gpm_mvr_partIdx1_enable_flag가 1과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되고; 넷째, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 1과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 모두에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되는지 여부에 대한 결정은 2개의 GPM 파티션에 적용되는 (gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx0_distance_idx, 그리고 gpm_mvr_partIdx1_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같이) MVR의 값에 의존한다. 2개의 MVR의 값이 같다면, merge_gpm_idx0 및 merge_gpm_idx1은 동일한 것이 허용되지 않는다. 그렇지 않으면(2개의 MVR의 값이 동일하지 않다면), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0181] 위의 방법 1에서, TM 및 MVR은 GPM에 배타적으로 적용된다. 그러한 방식에서, TM 모드의 미세조정된 MV에 추가로 MVR을 적용하는 것은 금지된다. 따라서 GPM에 대한 더 많은 MV 후보를 더 제공하기 위해, TM 미세조정 MV 상에 MVR 오프셋의 적용을 가능하게 하는 방법 2가 제안된다. 표 13은 GPM-MVR이 템플릿 매칭과 조합될 때 대응하는 신택스 표를 예시한다. 표 13에서, 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다.

[0182] **표 13** 템플릿 매칭과 GPM-MVR을 조합하는 제안된 방법의 신택스 엘리먼트(방법 2)

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
.....	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
gpm_tm_enable_flag0[x0][y0]	***ae(v)***
gpm_mvr_partIdx0_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_direc to in_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx0_distance_idx[x0][y0]	***ae(v)***
}	
gpm_tm_enable_flag1[x0][y0]	***ae(v)***
gpm_mvr_partIdx1_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_direction_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx1_distance_idx[x0][y0]	***ae(v)***
}	
}	
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
.....	
}	

[0183]

[0184] 표 13에 도시된 바와 같이, 표 12와 상이하게, gpm_tm_enable_flag0 및 gpm_tm_enable_flag1 상의 gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 시그널링 조건이 제거된다. 따라서 하나의 GPM 파티션의 단방향 움직임을 미세조정하기 위해 TM이 적용되는지 여부에 관계없이, MV 미세조정은 항상 GPM 파티션의 MV에 적용되는 것이 허용된다. 이전과 유사하게, 2개의 GPM 파티션의 결과적인 MV가 동일하지 않음을 보장하기 위해 다음의 조건이 적용되어야 한다.

[0185] 먼저, gpm_tm_enable_flag0 및 gpm_tm_enable_flag1 중 하나가 1이고 다른 하나가 0일 때, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0186] 그렇지 않으면, 즉 gpm_tm_enable_flag0과 gpm_tm_enable_flag1 모두가 1과 같거나, 두 플래그 모두가 0과 같다면: 첫째, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 0과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 모두에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일할

수 없고; 둘째, gpm_mvr_partIdx0_enable_flag가 1과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 인에이블되고) 그리고 gpm_mvr_partIdx1_enable_flag가 0과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되고; 셋째, gpm_mvr_partIdx0_enable_flag가 0과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 디세이블되고) gpm_mvr_partIdx1_enable_flag가 1과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되고; 넷째, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 1과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 모두에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되는지 여부에 대한 결정은 2개의 GPM 파티션에 적용되는 (gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx0_distance_idx, 그리고 gpm_mvr_partIdx1_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같이) MVR의 값에 의존한다. 2개의 MVR의 값이 같다면, merge_gpm_idx0 및 merge_gpm_idx1은 동일한 것이 허용되지 않는다. 그렇지 않으면(2개의 MVR의 값이 동일하지 않다면), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0187] 위의 2개의 방법에서, TM이 각각의 GPM 파티션에 적용되는지 여부를 표시하기 위해 2개의 별개의 플래그가 시그널링될 필요가 있다. 추가된 시그널링은, 특히 낮은 비트 레이트에서 추가 오버헤드로 인해 전체 코딩 효율을 감소시킬 수 있다. 시그널링 오버헤드를 감소시키기 위해, 추가 시그널링을 도입하는 대신에, TM 기반 단방향 MV를 GPM 모드의 단방향 MV 후보 리스트에 삽입하기 위한 방법 3이 제안된다. TM 기반 단방향 MV는 "템플릿 매칭" 섹션에서 설명된 것과 동일한 TM 프로세스에 후속하여 그리고 초기 MV로서 GPM의 원래의 단방향 MV를 사용하여 생성된다. 그러한 방식에 의해, 인코더로부터 디코더로 여분의 제어 플래그를 추가로 시그널링할 필요가 없다. 대신에, 디코더는 하나의 MV가 비트스트림으로부터 수신된 대응하는 병합 인덱스(즉, merge_gpm_idx0 및 merge_gpm_idx1)를 통해 TM에 의해 미세조정되는지 여부를 식별할 수 있다. 정규 GPM MV 후보(즉, 비-TM) 및 TM 기반 MV 후보를 배열하기 위한 상이한 방법이 존재할 수 있다. 하나의 방법에서, TM 기반 MV 후보를 MV 후보 리스트의 시작부에 배치하고, 이어서 비-TM 기반 MV 후보를 배치하는 것이 제안된다. 다른 방법에서, 처음에 비-TM 기반 MV 후보를 시작부에 두고, 이어서 TM 기반 후보를 두는 것이 제안된다. 다른 방법에서, TM 기반 MV 후보 및 비-TM 기반 MV 후보를 인터리빙된 방식으로 두는 것이 제안된다. 예를 들어, 이는 처음 N개의 비-TM 기반 후보를 배치할 수 있고; 그 다음에 모든 TM 기반 후보; 마지막으로, 나머지 비-TM 기반 후보를 배치할 수 있다. 다른 예에서, 이는 처음 N개의 TM 기반 후보를 배치할 수 있고; 그 다음에 모든 비-TM 기반 후보; 마지막으로, 나머지 TM 기반 후보를 배치할 수 있다. 다른 예에서, 비-TM 기반 후보 및 TM 기반 후보를 하나씩, 즉 하나의 비-TM 기반 후보, 하나의 TM 기반 후보 등을 배치하는 것이 제안된다.

[0188] 방법 1에서, 2개의 GPM 템플릿 플래그가 GPM-MVR 플래그 전에 시그널링된다. 구체적으로, 이러한 설계에서, GPM-MVR은 0과 같은 하나의 파티션의 GPM 템플릿 플래그를 먼저 시그널링함으로써 하나의 주어진 GPM 파티션에 대해서만 인에이블될 수 있다. GPM 템플릿 플래그가 적절한 콘텍스트 모델을 사용하여 코딩될 수 있지만, 이는 GPM-MVR 모드 상에서 시그널링 패널티를 야기할 것이다. 이러한 문제를 해결하기 위해, 본 개시내용의 일 실시예에서, GPM-TM 모드를 시그널링하기 전에 먼저 GPM-MVR 모드를 시그널링하는 것이 제안된다. 구체적으로, 이러한 방법에서, GPM-MVR 플래그는 먼저, GPM-MVR이 파티션에 적용되는지 여부를 표시하기 위해 각각의 GPM 파티션에 대해 시그널링된다. 플래그가 1과 같을 때, MVR 신택스 엘리먼트 gpm_mvr_partIdx0_distance_idx/gpm_mvr_partIdx1_distance_idx 및 gpm_mvr_partIdx0_dierction_idx/gpm_mvr_partIdx1_direction_idx는 파티션의 MVR 크기 및 방향의 대응하는 값을 특정하도록 추가로 시그널링된다 그렇지 않으면, 파티션의 GPM-MVR 플래그가 거짓과 같을 때, (좌측 및 최상부 이웃의 재구성된 샘플을 사용하여 파티션의 MV를 미세조정하는) GPM-TM 모드가 적용되는지 여부를 표시하기 위해 GPM-TM 플래그가 시그널링될 것이다. 표 22는 위의 시그널링 방법이 적용될 때의 대응하는 신택스 표를 예시하여, 여기서 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다.

[0189] 표 22

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
……	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
gpm_mvr_partIdx0_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_directoin_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx0_distance_idx[x0][y0]	***ae(v)***
} else {	
gpm_tm_enable_flag0[x0][y0]	***ae(v)***
}	
gpm_mvr_partIdx1_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_direction_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx1_distance_idx[x0][y0]	***ae(v)***
} else {	
gpm_tm_enable_flag1[x0][y0]	***ae(v)***
}	
}	
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
……	
}	

[0190]

[0191] 추가로, GPM 병합 인덱스 사이의 시그널링 리던던시를 제거하기 위해, 다음의 조건이 적용되어야 한다.

[0192] 먼저, gpm_tm_enable_flag0 및 gpm_tm_enable_flag1 중 하나가 1이고 다른 하나가 0일 때, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0193] 둘째, gpm_tm_enable_flag0과 gpm_tm_enable_flag1 둘 다 1과 같을 때, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0194] 다른 예에서, gpm_tm_enable_flag0과 gpm_tm_enable_flag1의 값 모두가 1과 같을 때, merge_gpm_idx0과 merge_gpm_idx1의 값은 동일한 것이 허용되지 않는다.

[0195] 셋째, gpm_tm_enable_flag0과 gpm_tm_enable_flag1이 0과 같을 때, 상이한 조건이 적용된다. gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 0과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 둘 모두에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값은 동일할 수 없다. gpm_mvr_partIdx0_enable_flag가 1과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 인에이블되고) gpm_mvr_partIdx1_enable_flag가 0과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다. gpm_mvr_partIdx0_enable_flag가 0과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 디세이블되고) gpm_mvr_partIdx1_enable_flag가 1과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다. gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 1과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 모두에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되는지 여부에 대한 결정은 2개의 GPM 파티션에 적용되는 (gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx0_distance_idx, 그리고 gpm_mvr_partIdx1_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같은) MVR의 값에 의존한다. 2개의 MVR의 값이 같다면, merge_gpm_idx0 및 merge_gpm_idx1은 동일한 것이 허용되지 않는다. 그렇지 않으면(2개의 MVR의 값이 동일하지 않다면), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0196] 다른 방법에서, 2개의 별개의 GPM-TM 플래그를 사용하는 대신에, 2개의 GPM 파티션에 대한 템플릿 매칭의 인에

이블/디세이블을 공동으로 제어하기 위한 하나의 단일 플래그가 제안된다. 플래그가 참인 경우, 이는 2개의 GPM 파티션의 2개의 단방향 MV가 템플릿 매칭 방식에 의해 템플릿(즉, 좌측 및 최상부 이웃의 재구성된 샘플)과 그 대응하는 참조 샘플 간의 차의 최소화에 기초하여 미세조정될 필요가 있음을 의미한다. 구체적으로, 방법 4와 유사하게, GPM-MVR이 하나의 특정 GPM 파티션에 적용되는지 여부를 표시하기 위해, 하나의 GPM CU에 대해 2개의 GPM-MVR 플래그가 먼저 시그널링된다. 각각의 파티션의 GPM-MVR 플래그가 참과 같을 때, MVR 크기 및 MVR 방향은 다음에서 파티션에 대해 추가로 시그널링된다. 게다가, 2개의 GPM 파티션의 두 GPM-MVR 플래그 모두가 거짓과 같을 때, GPM-TM이 2개의 GPM 파티션 모두에 적용되는지 여부를 표시하기 위해 GPM-TM 플래그가 추가로 시그널링될 것이다. 표 23은 그러한 설계가 적용될 때의 GPM 모드의 대응하는 신택스 표를 예시하며, 여기서 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다.

[0197] 표 23

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
……	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
gpm_mvr_partIdx0_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_directoin_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx0_distance_idx[x0][y0]	***ae(v)***
}	
gpm_mvr_partIdx1_enable_flag[x0][y0]	***ae(v)***
if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_direction_idx[x0][y0]	***ae(v)***
gpm_mvr_partIdx1_distance_idx[x0][y0]	***ae(v)***
}	
if(!gpm_mvr_partIdx0_enable_flag[x0][y0] && !gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_tm_enable_flag[x0][y0]	***ae(v)***
}	
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
……	
}	

[0198]

[0199] 다른 실시예에서, 2개의 GPM-MVR 플래그를 시그널링하기 전에 2개의 GPM 파티션에 대한 GPM-TM 플래그를 시그널링하는 것이 제안된다. 대응하여, GPM-TM 플래그의 값이 0과 같을 때(즉, GPM-TM이 2개의 GPM 파티션에 적용되지 않을 때) GPM-MVR 플래그만이 시그널링되도록 2개의 GPM-MVR 플래그의 존재를 컨디셔닝하기 위해 GPM-TM의 값이 사용될 수 있다. 표 24는 그러한 시그널링 방식이 적용될 때의 GPM 모드의 대응하는 신택스 표를 예시하며, 여기서 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다.

[0200] 표 24

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
.....	
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
gpm_tm_enable_flag[x0][y0]	*ae(v)*
if(gpm_tm_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_enable_flag[x0][y0]	*ae(v)*
}	
if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
gpm_mvr_partIdx0_direc to_in_idx[x0][y0]	*ae(v)*
gpm_mvr_partIdx0_distance_idx[x0][y0]	*ae(v)*
}	
if(gpm_tm_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_enable_flag[x0][y0]	*ae(v)*
}	
if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
gpm_mvr_partIdx1_direction_idx[x0][y0]	*ae(v)*
gpm_mvr_partIdx1_distance_idx[x0][y0]	*ae(v)*
}	
merge_gpm_idx0[x0][y0]	ae(v)
merge_gpm_idx1[x0][y0]	ae(v)
.....	
}	

[0201]

[0202] 추가로, 2개의 방법 모두에 대해, GPM 병합 인덱스 사이의 시그널링 리던던시를 제거하기 위해, 다음의 조건이 적용되어야 한다.

[0203] 먼저, gpm_tm_enable_flag가 1일 때, merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다. 둘째, gpm_tm_enable_flag가 0와 같을 때, 상이한 조건이 적용될 수 있다. 예를 들어, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 0과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 둘 모두에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값은 동일할 수 없다. 게다가, gpm_mvr_partIdx0_enable_flag가 1과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 인에이블되고) gpm_mvr_partIdx1_enable_flag가 0과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 디세이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다. 더욱이, gpm_mvr_partIdx0_enable_flag가 0과 같고(즉, GPM-MVR이 제1 GPM 파티션에 대해 디세이블되고) gpm_mvr_partIdx1_enable_flag가 1과 같을 때(즉, GPM-MVR이 제2 GPM 파티션에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다. 추가로, gpm_mvr_partIdx0_enable_flag 및 gpm_mvr_partIdx1_enable_flag의 두 값 모두가 1과 같을 때(즉, GPM-MVR이 2개의 GPM 파티션 모두에 대해 인에이블될 때), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용되는지 여부에 대한 결정은 2개의 GPM 파티션에 적용되는 (gpm_mvr_partIdx0_direction_idx 및 gpm_mvr_partIdx0_distance_idx, 그리고 gpm_mvr_partIdx1_direction_idx 및 gpm_mvr_partIdx1_distance_idx에 의해 표시된 바와 같이) MVR의 값에 의존한다. 2개의 MVR의 값이 같다면, merge_gpm_idx0 및 merge_gpm_idx1은 동일한 것이 허용되지 않는다. 그렇지 않으면(2개의 MVR의 값이 동일하지 않다면), merge_gpm_idx0 및 merge_gpm_idx1의 값이 동일한 것이 허용된다.

[0204] 다른 예에서, gpm_tm_enable_flag가 1과 같을 때, merge_gpm_idx0과 merge_gpm_idx1의 값은 동일한 것이 허용되지 않는다.

[0205] 템플릿 매칭 방식이 GPM 모드에 적용될 때, 각각의 GPM 파티션에 대한 최적의 단방향 MV를 식별하기 위해 계산상 광범위한 움직임 추정을 수행함으로써 인코더와 디코더 모두에 대해 추가 복잡도가 필요하게 된다. 이러한 무시할 수 없는 복잡도 증가는 GPM 모드를 낮은 비디오 지연이 요구되는 라이브 비디오 스트리밍, 비디오 컨퍼

런스 및 비디오 게이밍과 같은 특정 저급 인코더 또는 특정 비디오 애플리케이션에 대해 실현 가능하지 않게 할 수 있다. 이러한 고려사항에 기초하여, 시퀀스 레벨, 픽처/슬라이스 레벨, 코딩 블록 그룹 레벨 등과 같은 특정한 높은 코딩 레벨에서 하나의 제어 플래그를 부가하여 그 레벨 하에서 CU의 GPM-TM 모드를 적응적으로 인에이블 또는 디세이블하는 것이 제안된다. 제안된 적응이 픽처 레벨에서 수행된다고 가정하면, 표 25는 픽처 헤더에서 시그널링된 대응하는 신택스 엘리먼트를 예시하며, 여기서 새롭게 추가된 신택스 엘리먼트는 굵은 이탤릭체이다.

[0206] 표 25

[0207]

pic ture_head er_struc ture( ) {	De scriptor
.....	
if(sps_mmvd_full pel_on ly_en a bled_flag)	
ph_mmvd_full pel_on ly_flag	u(1)
pres enceFlag = 0	
if(!pps_rpl_in fo_in_ph_flag) /* This con di tion is in ten tion al ly not merged in to the next, to avoid pos si ble in ter pre ta tion of RplsIdx[i] not hav ing a spec i fied val ue. */	
pres enceFlag = 1	
else if(num_ref_en tries[1][RplsIdx[1]] > 0)	
pres enceFlag = 1	
if(pres enceFlag) {	
ph_mvd_l1_zero_flag	u(1)
if(sps_bdof_con trol_pre sent_in_ph_flag)	
ph_bdof_dis a bled_flag	u(1)
if(sps_dmvr_con trol_pre sent_in_ph_flag)	
ph_dmvr_dis a bled_flag	u(1)

[0208]

}	
if(!pps_rpl_in fo_in_ph_flag || num_ref_en tries[1][RplsIdx[1]] > 0) {	
if(sps_dmvd_en a ble_flag) {	
ph_gpm_tm_en a ble_flag	***u(1)***
}	
}	
if(sps_prof_con trol_pre sent_in_ph_flag)	
ph_prof_dis a bled_flag	u(1)
.....	
}	

[0209] 위의 신택스 표 25에서, 플래그 sps_dmvd_enable_flag는 템플릿 매칭이 비디오 시퀀스의 코딩에 대해 인에이블되는지 여부를 표시하는 시퀀스 레벨 제어 플래그이고, ph_gpm_tm_enable_flag는 GPM-TM이 픽처 내의 CU에 적용될 수 있는지 여부를 표시하는 데 사용되는 제안된 GPM-TM 제어 플래그이다.

[0210] **움직임 벡터 프루닝(pruning)을 이용한 GPM 후보 리스트 구성**

[0211] 도입부에서 논의된 바와 같이, 2개의 기하학적 파티션의 MV를 획득하기 위해, 하나의 단방향 예측 후보 리스트가 먼저 정규 병합 후보 리스트 생성 프로세스로부터 직접 도출된다. 각각의 GPM MV의 예측 방향의 선택이 대응하는 병합 인덱스의 패리티에 기반한다고 가정하면, 2개의 기하학적 파티션의 MV는 동일할 수 있는데, 이는 CU의 기하학적 파티션이 비-파티션 경우에 비해 어떠한 추가 이익도 제공할 수 없기 때문에 명백하게 의미가 맞지 않는다. 이러한 리던던시를 피하기 위해, 하나의 GPM CU의 단방향 예측 MV 후보 리스트를 생성할 때 움직임 벡터 프루닝을 적용하는 것이 제안되며, 그러한 하나의 MV는 단지, 리스트 내의 기존의 후보 중 어떠한 후보와도 동일하지 않을 때 그리고 그러한 경우에만 리스트에 추가될 수 있다. 다른 방식에서, 2개의 MV를 비교할 때 하나의 MV 임계치가 적용되도록 추가로 제안된다. 구체적으로, 이러한 방법에 의해, (수평 방향 및 수직 방향 각각에서의) 2개의 MV의 차이가 하나의 MV 임계치보다 작을 때, 2개의 MV가 동일한 것으로 간주되고; 그렇지 않으면(한 방향에서의 MV 차이가 MV 임계치보다 크거나 같다면), 2개의 MV는 동일하지 않은 것으로 간주된다. 하나의 방법에서, 모든 블록 크기에 대해 하나의 고정된 MV 임계치를 사용하는 것이 제안된다. 다른 방법에서,

더 큰 MV 임계치가 더 큰 CU에 대해 사용되는 한편 더 작은 MV 임계치가 작은 CU에 사용되도록 코딩 블록의 크기에 기초하여 MV 임계치의 값을 결정하는 것이 제안된다. 일부 예에서, 블록 내의 샘플의 수가 N < 64일 때 MV 임계치의 값은 ¼ 픽셀로 설정되고; 64 <= N < 256일 때 MV 임계치의 값은 ½ 픽셀로 설정되며; N >= 256일 때 MV 임계치의 값은 1 픽셀로 설정된다.

[0212] **개선된 단방향 MV 후보 리스트 구성**

[0213] 정규 병합 모드의 MV 후보 리스트로부터 단방향 MV 후보를 도출하기 위한 하나의 개선된 후보 리스트 구성 방법이 제안된다.

[0214] 먼저, 패리티 기반 단방향 MV가 획득된다. 기존의 GPM 설계와 동일하게, 다수의 단방향 MV가 먼저 정규 병합 후보 리스트 생성 프로세스로부터 도출된다. 예를 들어, 도 5에 도시된 바와 같이, n은 GPM MV 후보 리스트에서 단방향 움직임의 인덱스로서 표시된다. X가 n의 패리티와 동일한, n번째 병합 후보의 LX 움직임 벡터가 후보 리스트에서 n번째 단방향 MV로서 사용된다. n번째 확장된 병합 후보의 LX 움직임 벡터가 존재하지 않는 경우, 동일한 후보의 L(1 - X) MV가 대신 선택될 것이다.

[0215] 일부 예에서, 패리티 방지 기반 단방향 MV가 획득된다. 단방향 MV 후보 리스트가 가득 차 있지 않을 때, 정규 병합 후보 리스트의 양방향 예측 MV로부터 도출되는 추가 단방향 MV가 단방향 MV 후보 리스트에 더 추가된다. 구체적으로, 정규 병합 후보 리스트에서 병합 인덱스(n)를 갖는 각각의 양방향 예측 MV에 대해, 양방향 예측 MV의 L(1-X) MV가 단방향 MV 후보 리스트에 더 추가되며, 여기서 X는 n의 패리티와 같다.

[0216] 일부 예에서, 쌍별 평균 단방향 MV가 획득된다. 단방향 MV 후보 리스트가 가득 차 있지 않을 때, 기존의 단방향 MV 후보 리스트의 하나의 참조 픽처 리스트(L0 및 L1) 내의 처음 2개의 단방향 후보를 평균함으로써 하나 이상의 쌍별 평균 후보가 리스트에 추가된다. 일부 예에서, 쌍별 평균 단방향 MV는, 패리티 방지 기반 단방향 MV가 획득되어 단방향 MV 후보 리스트에 추가된 후에 획득된다. 일부 다른 예에서, 쌍별 평균 단방향 MV는, 패리티 방지 기반 단방향 MV를 획득하고 패리티 방지 기반 단방향 MV를 단방향 MV 후보 리스트에 추가하기 전에 획득될 수 있다.

[0217] 예를 들어, 하나의 주어진 참조 픽처 리스트(L0/L1)에 대해, 인코더/디코더는 제1 MV 후보가 p0Cand로서 정의되고 제2 L0 MV 후보가 p1Cand로서 정의된다고 가정할 수 있다. 2개의 MV 후보가 동일한 참조 픽처를 가리킨다면, 동일한 기준으로 지향되는 2개의 MV 후보를 평균함으로써 하나의 쌍별 평균 후보가 생성되고; 그렇지 않으면, 2개의 MV 후보가 상이한 참조 픽처를 가리킬 때, p0Cnad 및 p1Cand를 평균함으로써 쌍별 MV의 크기가 계산되고, 제1 MV 후보의 참조 픽처가 결과적인 쌍별 평균 MV의 참조 픽처로서 선택된다. 다른 예에서, 2개의 MV 후보가 상이한 참조 픽처를 가리킬 때, 제2 MV 후보의 참조 픽처가 결과적인 쌍별 평균 MV의 참조 픽처로서 선택된다.

[0218] 더욱이, 0의 단방향 MV가 획득된다. 단방향 MV 후보 리스트가 여전히 가득 차 있지 않은 경우, (현재 픽처의 L0/L1 참조 픽처 리스트 내의 상이한 참조 픽처로 지향되는) 0의 단방향 MV는 리스트의 최대 길이에 도달할 때까지 단방향 MV 후보 리스트에 주기적으로 추가된다.

[0219] 추가로, 위의 단방향 MV 후보 리스트 생성 방식에서, MV 프루닝(pruning) 프로세스가 추가로 적용되어 리스트로부터 중복 MV 후보를 제거할 수 있다. 하나 이상의 실시예에서, 디폴트 MV 프루닝 방법은, 하나의 MV가 단지, 리스트 내의 기존의 후보 중 어떠한 후보와도 동일하지 않을 때 그리고 그러한 경우에만 리스트에 추가될 수 있도록 적용된다. 일부 다른 실시예에서, "움직임 벡터 프루닝을 이용한 GPM 후보 리스트 구성" 섹션에서 제안된 바와 같은 대안적인 MV 프루닝 방법이 적용되며, 여기서 2개의 MV가 동일한지 여부를 결정하는 데 사용되는 MV 임계치는 현재 CU의 블록 크기에 의존한다.

[0220] 도 9는 사용자 인터페이스(960)와 결합된 컴퓨팅 환경(또는 컴퓨팅 디바이스)(910)을 도시한다. 컴퓨팅 환경(910)은 데이터 처리 서버의 일부일 수 있다. 일부 실시예에서, 컴퓨팅 디바이스(910)는 본 개시내용의 다양한 예에 따라 앞서 설명된 바와 같은 다양한 방법 또는 프로세스(이를테면, 인코딩/디코딩 방법 또는 프로세스) 중 임의의 것을 수행할 수 있다. 컴퓨팅 환경(910)은 프로세서(920), 메모리(940) 및 I/O 인터페이스(950)를 포함할 수 있다.

[0221] 프로세서(920)는 통상적으로, 컴퓨팅 환경(910)의 전체 동작, 이를테면 디스플레이, 데이터 획득, 데이터 통신 및 이미지 처리와 연관된 동작을 제어한다. 프로세서(920)는 위에서 설명된 방법의 단계 모두 또는 일부를 수행하기 위한 명령을 실행하기 위한 하나 이상의 프로세서를 포함할 수 있다. 더욱이, 프로세서(1020)는 프로세서(920)와 다른 컴포넌트 간의 상호 작용을 가능하게 하는 하나 이상의 모듈을 포함할 수 있다. 프로세서는 중

앙 처리 유닛(CPU: Central Processing Unit), 마이크로프로세서, 단일 칩 머신, GPU 등일 수 있다.

[0222] 메모리(940)는 컴퓨팅 환경(910)의 동작을 지원하기 위해 다양한 타입의 데이터를 저장하도록 구성된다. 메모리(940)는 미리 결정된 소프트웨어(942)를 포함할 수 있다. 이러한 데이터의 예는 컴퓨팅 환경(910) 상에서 동작되는 임의의 애플리케이션 또는 방법에 대한 명령, 비디오 데이터세트, 이미지 데이터 등을 포함한다. 메모리(940)는 임의의 타입의 휘발성 또는 비휘발성 메모리 디바이스, 또는 이들의 조합, 이를테면 정적 랜덤 액세스 메모리(SRAM: static random access memory), 전기적으로 소거 가능한 프로그래밍 가능 판독 전용 메모리(EEPROM: electrically erasable programmable read-only memory), 소거 가능한 프로그래밍 가능 판독 전용 메모리(EPROM: erasable programmable read-only memory), 프로그래밍 가능 판독 전용 메모리(PROM: programmable read-only memory), 판독 전용 메모리(ROM: read-only memory), 자기 메모리, 플래시 메모리, 자기 또는 광 디스크를 사용함으로써 구현될 수 있다.

[0223] I/O 인터페이스(950)는 키보드, 클릭 휠, 버튼 등과 같은 주변 인터페이스 모듈과 프로세서(920) 사이에 인터페이스를 제공한다. 버튼은 홈 버튼, 스캔 시작 버튼 및 스캔 중지 버튼을 포함할 수 있지만 그에 제한되는 것은 아니다. I/O 인터페이스(950)는 인코더 및 디코더와 결합될 수 있다.

[0224] 일부 실시예에서, 앞서 설명된 방법을 수행하기 위해 컴퓨팅 환경(910) 내의 프로세서(920)에 의해 실행 가능한, 이를테면 메모리(940)에 포함된 복수의 프로그램을 포함하는 비-일시적 컴퓨터 판독 가능 저장 매체가 또한 제공된다. 예를 들어, 비-일시적 컴퓨터 판독 가능 저장 매체는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광학 데이터 저장 디바이스 등일 수 있다.

[0225] 비-일시적 컴퓨터 판독 가능 저장 매체는, 하나 이상의 프로세서를 갖는 컴퓨팅 디바이스에 의한 실행을 위한 복수의 프로그램을 내부에 저장하며, 여기서 복수의 프로그램은 하나 이상의 프로세서에 의해 실행될 때, 컴퓨팅 디바이스로 하여금, 앞서 설명된 움직임 예측을 위한 방법을 수행하게 한다.

[0226] 일부 실시예에서, 컴퓨팅 환경(910)은 위의 방법을 수행하기 위한 하나 이상의 주문형 집적 회로(ASIC), 디지털 신호 프로세서(DSP), 디지털 신호 처리 디바이스(DSPD), 프로그래밍 가능 로직 디바이스(PLD), 필드 프로그래밍 가능 게이트(FPGA), 그래픽 처리 유닛(GPU: graphical processing unit), 제어기, 마이크로컨트롤러, 마이크로프로세서, 또는 다른 전자 컴포넌트로 구현될 수 있다.

[0227] 도 8은 본 개시내용의 일례에 따른, GPM에서 비디오 블록을 디코딩하기 위한 방법을 예시하는 흐름도이다.

[0228] 단계(801)에서, 프로세서(920)는 비디오 블록을 제1 기하학적 파티션 및 제2 기하학적 파티션으로 파티셔닝할 수 있다.

[0229] 단계(802)에서, 프로세서(920)는 복수의 정규 병합 후보를 추가함으로써 GPM의 단방향 MV 후보 리스트를 구성할 수 있 있다.

[0230] 복수의 정규 병합 후보는 정규 병합 후보 리스트 생성 프로세스로부터 도출될 수 있다. 예를 들어, 도 5에 도시된 바와 같이, X가 n의 패리티와 동일한, n번째 병합 후보의 LX 움직임 벡터가 후보 리스트에서 n번째 단방향 MV로서 사용된다.

[0231] 단계(803)에서, 프로세서(920)는 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로 정규 병합 후보 리스트의 하나 이상의 양방향 예측 MV로부터 도출된 하나 이상의 추가 단방향 MV를 단방향 MV 후보 리스트에 추가함으로써 제1 업데이트된 단방향 MV 후보 리스트를 구성할 수 있다.

[0232] 일부 예에서, 하나 이상의 추가 단방향 MV는 제1 참조 픽처 리스트에서 홀수 병합 인덱스를 갖는 하나 이상의 MV 후보 및 제2 참조 픽처 리스트에서 짝수 병합 인덱스를 갖는 하나 이상의 MV를 획득함으로써 도출될 수 있다. 예를 들어, 제1 참조 픽처 리스트는 L0일 수 있고 제2 참조 픽처 리스트는 L1일 수 있다. 추가 단방향 MV는 L0에서 홀수 병합 인덱스를 갖는 후보 및 L1에서 짝수 병합 인덱스를 갖는 후보를 포함할 수 있다. 즉, 정규 병합 후보 리스트에서 병합 인덱스(n)를 갖는 각각의 양방향 예측 MV에 대해, 양방향 예측 MV의 L(1-X) MV가 단방향 MV 후보 리스트에 더 추가되며, 여기서 X는 n의 패리티와 같다. 일부 예에서, 제1 참조 픽처 리스트는 L1일 수 있고 제2 참조 픽처 리스트는 L0일 수 있다.

[0233] 일부 예에서, 프로세서(920)는 단계(804)에 도시된 바와 같이, 하나 이상의 쌍별 평균 후보를 제1 업데이트된 단방향 MV 후보 리스트에 추가함으로써 제2 업데이트된 단방향 MV 후보 리스트를 구성할 수 있다.

[0234] 예를 들어, 프로세서(920)는 처음 2개의 단방향 MV 후보가 동일한 참조 픽처를 지시한다는 결정에 대한 응답으

로, 제1 참조 픽처 리스트 또는 제2 참조 픽처 리스트에서 처음 2개의 단방향 MV 후보를 획득하고 처음 2개의 단방향 MV 후보를 평균함으로써 쌍별 평균 후보를 획득할 수 있다.

[0235] 일부 예에서, 쌍별 평균 후보는 처음 2개의 단방향 MV 후보가 상이한 참조 픽처를 지시한다는 결정에 대한 응답으로, 처음 2개의 단방향 MV 후보를 평균함으로써 쌍별 평균 후보의 크기를 결정하고 첫 번째 단방향 MV 후보의 참조 픽처를 쌍별 평균 후보의 참조 픽처로서 결정함으로써 획득될 수 있다.

[0236] 일부 예에서, 쌍별 평균 후보는, 처음 2개의 단방향 MV 후보를 평균함으로써 쌍별 평균 후보의 크기를 결정하고, 처음 2개의 단방향 MV 후보가 상이한 참조 픽처를 지시한다는 결정에 대한 응답으로 두 번째 단방향 MV 후보의 참조 픽처를 쌍별 평균 후보의 참조 픽처로서 결정함으로써 획득될 수 있다.

[0237] 일부 예에서, 프로세서(920)는 단계(805)에 도시된 바와 같이, 제2 업데이트된 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로, 최대 길이에 도달할 때까지 0의 단방향 MV를 제2 업데이트된 단방향 MV 후보 리스트에 주기적으로 추가하는 단계를 포함할 수 있다.

[0238] 일부 예에서, 단방향 MV 후보 리스트를 구성할 때, 중복 후보는 단방향 MV 후보 리스트로부터 제거될 수 있다.

[0239] 예를 들어, 프로세서(920)는 추가 단방향 MV가 단방향 MV 후보 리스트 내의 후보와 동일하다는 결정에 대한 응답으로 추가 단방향 MV를 단방향 MV 후보 리스트에 추가하는 것을 생략할 수 있다.

[0240] 일부 예에서, 추가 단방향 MV가 단방향 MV 후보 리스트 내의 후보와 동일한지 여부를 결정하기 위해 MV 임계치가 사용된다. 예를 들어, 추가 단방향 MV와 단방향 MV 후보 리스트 내의 후보 간의 차이가 MV 임계치보다 더 작다는 결정에 대한 응답으로, 추가 단방향 MV는 단방향 MV 후보 리스트 내의 후보와 동일한 것으로 결정되며, MV 임계치는 고정 임계치 또는 비디오 블록의 블록 크기에 기초한 변수이다.

[0241] 더욱이, 프로세서(920)는 쌍별 평균 후보가 제1 업데이트된 단방향 MV 후보 리스트 내의 후보와 동일하다는 결정에 대한 응답으로 쌍별 평균 후보를 제1 업데이트된 단방향 MV 후보 리스트에 추가하는 것을 생략할 수 있다. 일부 예에서, 쌍별 평균 후보와 제1 업데이트된 단방향 MV 후보 리스트 내의 후보 간의 차이가 MV 임계치보다 더 작다는 결정에 대한 응답으로, 쌍별 평균 후보는 후보와 동일한 것으로 결정되며, MV 임계치는 고정 임계치 또는 비디오 블록의 블록 크기에 기초한 변수이다.

[0242] 단계(806)에서, 프로세서(920)는 제1 기하학적 파티션에 대한 단방향 MV 및 제2 기하학적 파티션에 대한 단방향 MV를 생성할 수 있다.

[0243] 일부 예에서, GPM에서 비디오 블록을 디코딩하기 위한 장치가 제공된다. 이 장치는 프로세서(920) 및 프로세서에 의해 실행 가능한 명령을 저장하도록 구성된 메모리(940)를 포함하며; 프로세서는 명령의 실행 시에, 도 8에 예시된 바와 같은 방법을 수행하도록 구성된다.

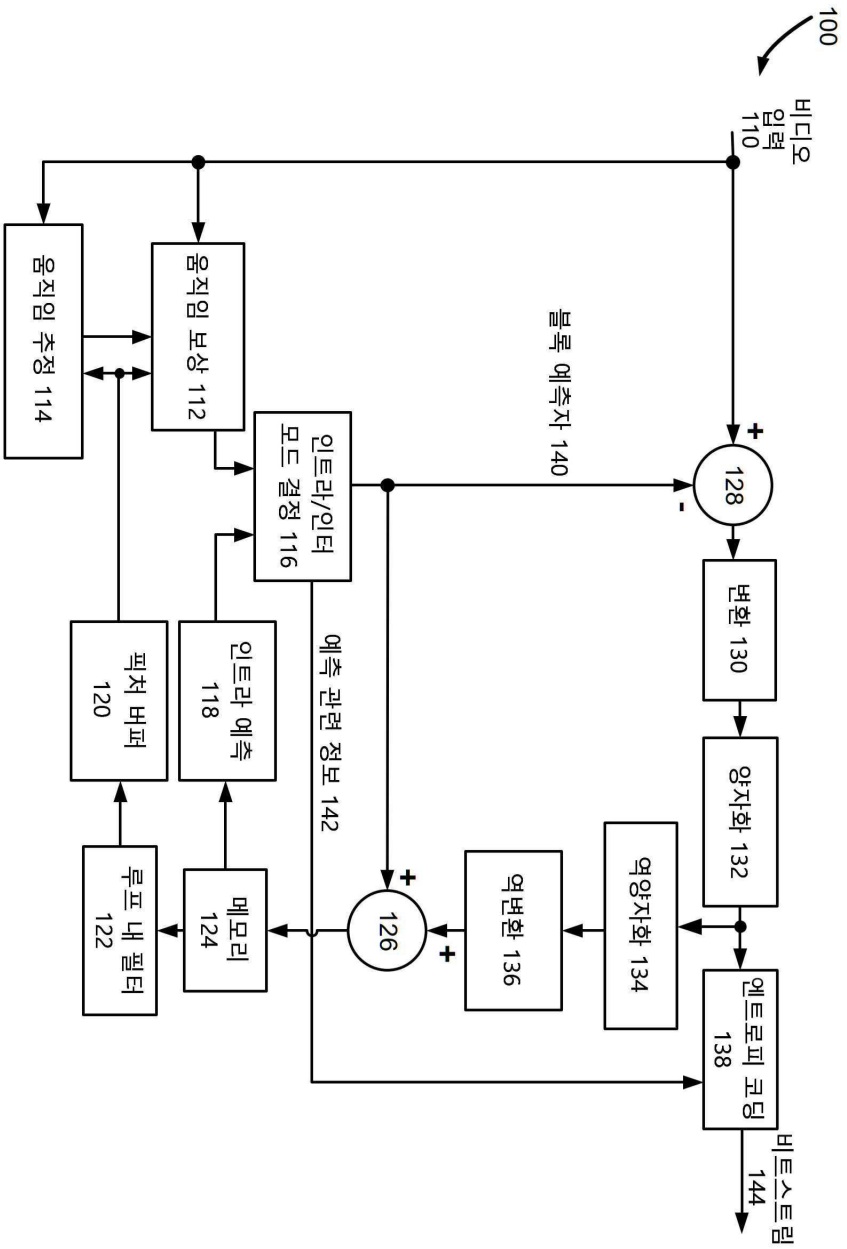
[0244] 일부 다른 예에서, 명령이 저장된 비-일시적 컴퓨터 판독 가능 저장 매체가 제공된다. 명령이 프로세서(920)에 의해 실행될 때, 명령은 프로세서로 하여금, 도 8에 예시된 바와 같은 방법을 수행하게 한다.

[0245] 본 개시내용의 다른 예는, 명세서의 고려 및 여기에 개시되는 본 개시내용의 실시로부터 당해 기술분야에서 통상의 지식을 가진 자들에게 자명할 것이다. 본 출원은 본 개시내용의 일반 원리들을 따르는 그리고 당해 기술분야에 알려진 또는 통상적인 실시 내에서 본 개시내용으로부터의 그러한 출발들을 포함하는 본 개시내용의 임의의 변형들, 사용들 또는 적응들을 커버하는 것으로 의도된다. 명세서 및 예는 단지 예시인 것으로 간주되는 것으로 의도된다.

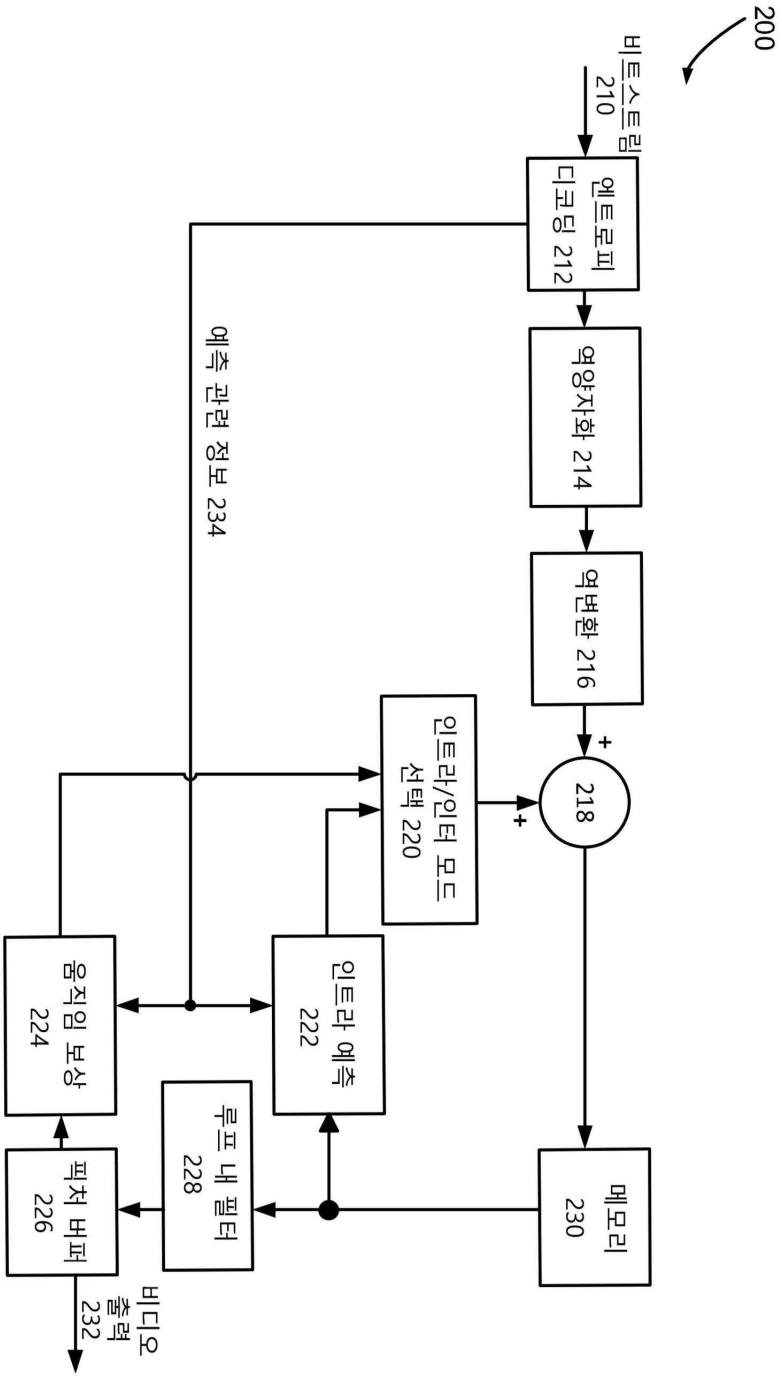
[0246] 본 개시내용은 위에서 설명되고 첨부 도면들에 예시된 정확한 예로 제한되지 않으며, 본 개시내용의 범위를 벗어나지 않으면서 다양한 수정들 및 변경들이 이루어질 수 있다고 인식될 것이다.

도면

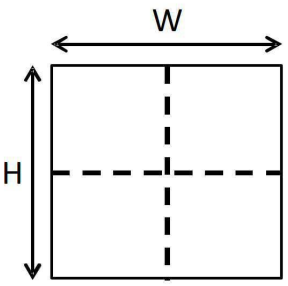
도면1


100
비디오 입력 110
움직임 추정 114
움직임 보상 112
인트라/인터 모드 결정 116
블록 예측자 140
128
변환 130
양자화 132
엔트로피 코딩 138
비트스트림 144
픽처 버퍼 120
인트라 예측 118
예측 관련 정보 142
루프 내 필터 122
메모리 124
126
역변환 136
역양자화 134

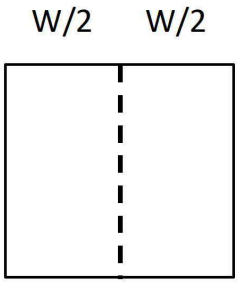
도면2


200
비트스트림 210
엔트로피 디코딩 212
역양자화 214
역변환 216
+
+
218
인트라/인터 모드 선택 220
예측 관련 정보 234
움직임 보상 224
인트라 예측 222
루프 내 필터 228
픽처 버퍼 226
메모리 230
비디오 출력 232

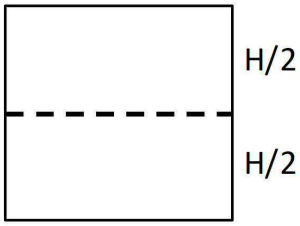
도면3a


W
H

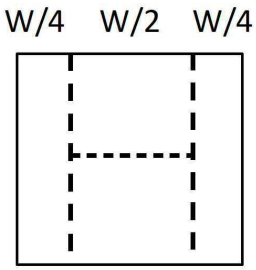
도면3b



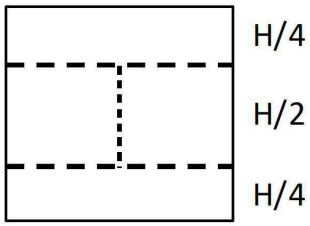
도면3c



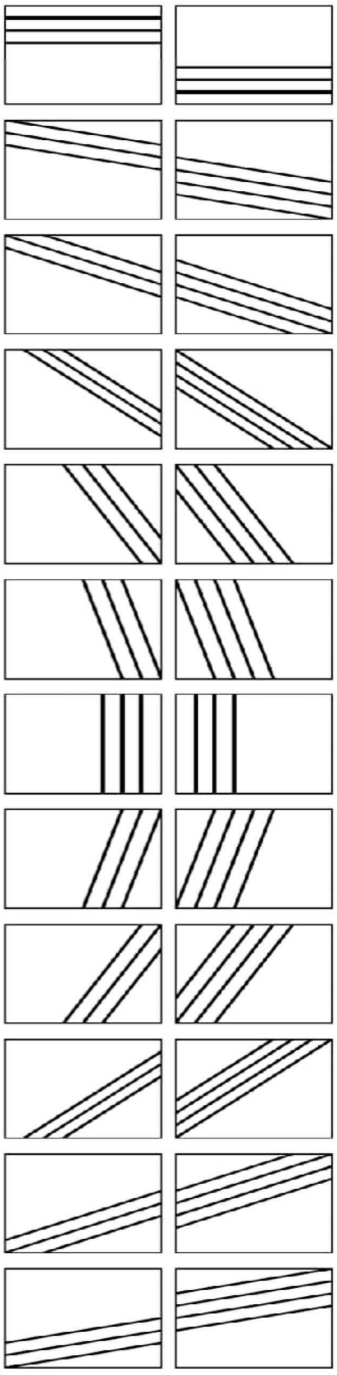
도면3d



도면3e



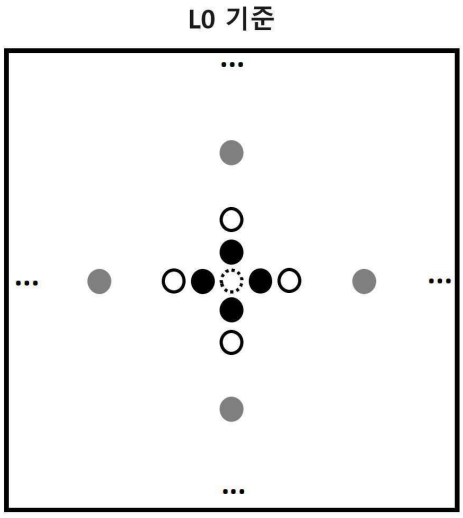
도면4



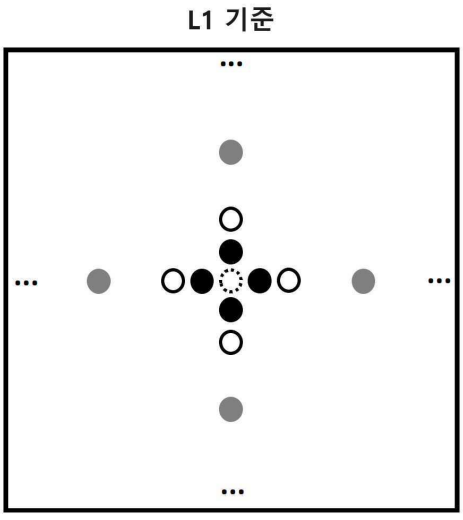
도면5

	L0 MV	L1 MV
병합 인덱스 0	X	
병합 인덱스 1		X
병합 인덱스 2	X	
병합 인덱스 3		X
병합 인덱스 4	X	

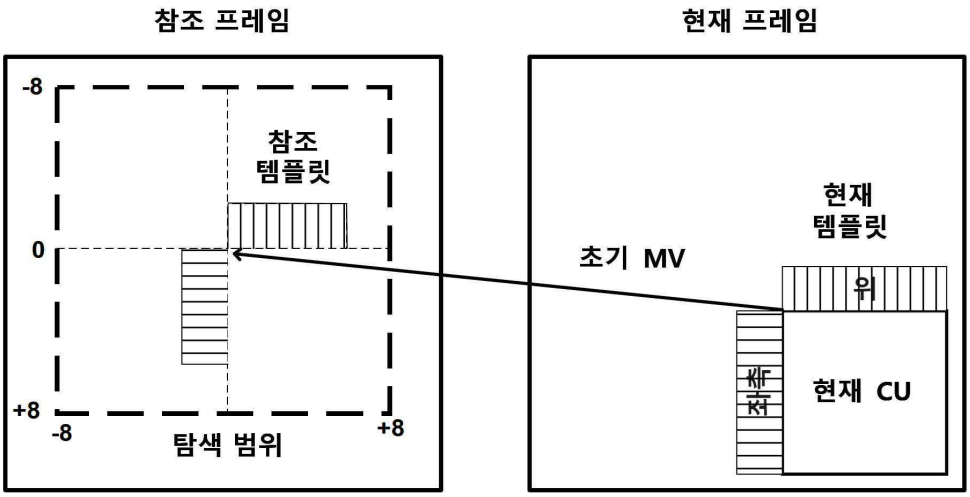
도면6a



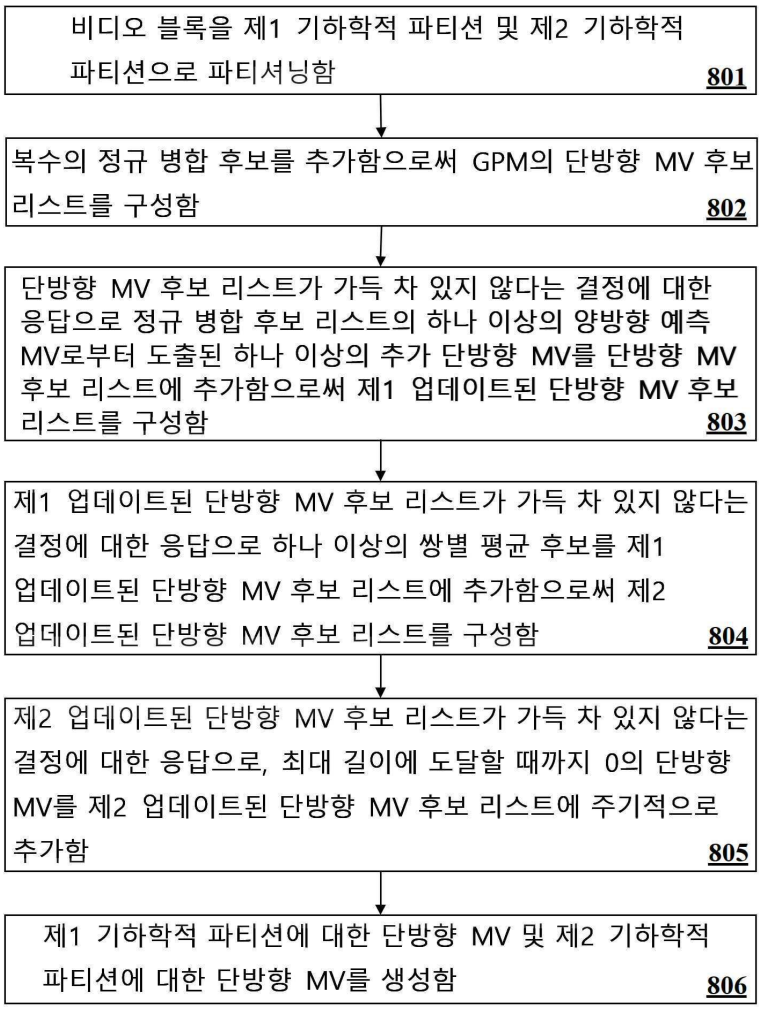
도면6b



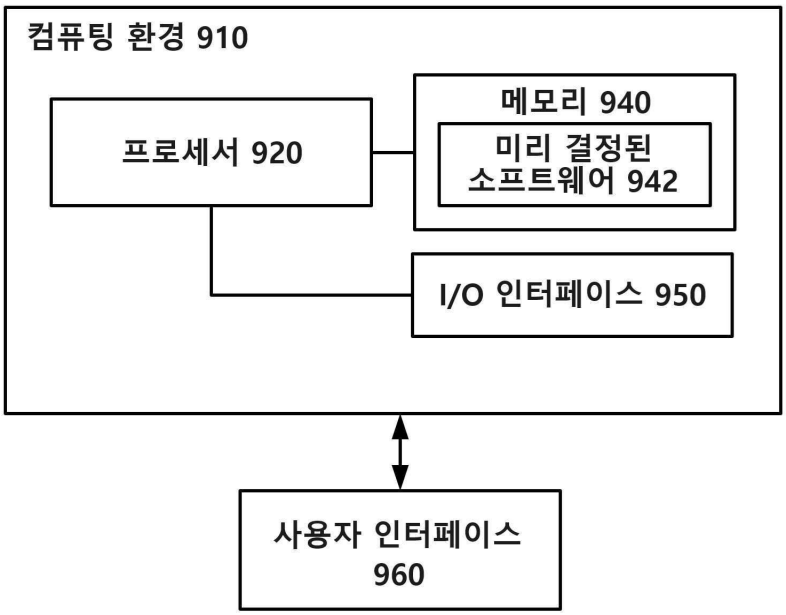
도면7


참조 프레임
현재 프레임
-8
0
+8
-8
+8
참조
템플릿
탐색 범위
초기 MV
현재
템플릿
위
왼쪽
현재 CU

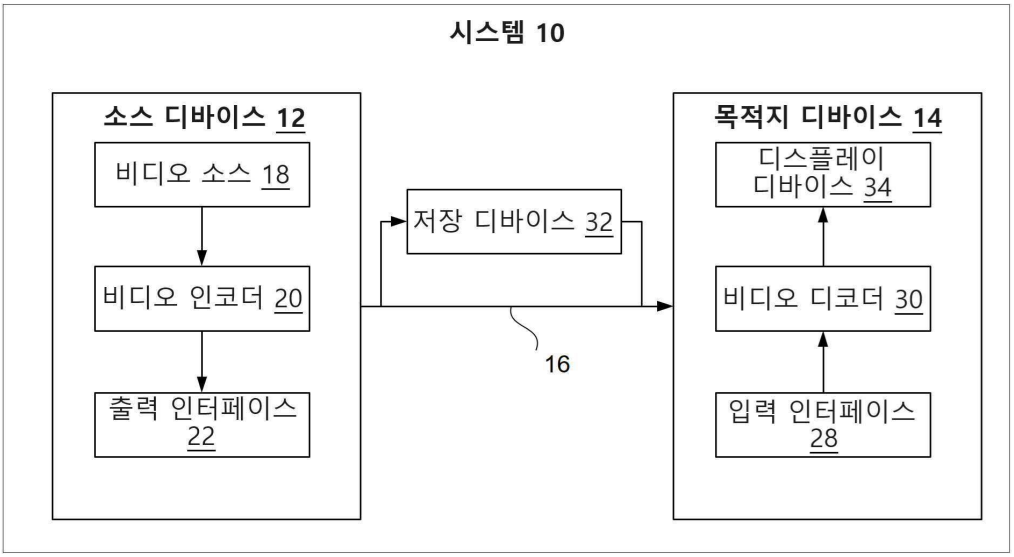
도면8


비디오 블록을 제1 기하학적 파티션 및 제2 기하학적 파티션으로 파티셔닝함
801
복수의 정규 병합 후보를 추가함으로써 GPM의 단방향 MV 후보 리스트를 구성함
802
단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로 정규 병합 후보 리스트의 하나 이상의 양방향 예측 MV로부터 도출된 하나 이상의 추가 단방향 MV를 단방향 MV 후보 리스트에 추가함으로써 제1 업데이트된 단방향 MV 후보 리스트를 구성함
803
제1 업데이트된 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로 하나 이상의 쌍별 평균 후보를 제1 업데이트된 단방향 MV 후보 리스트에 추가함으로써 제2 업데이트된 단방향 MV 후보 리스트를 구성함
804
제2 업데이트된 단방향 MV 후보 리스트가 가득 차 있지 않다는 결정에 대한 응답으로, 최대 길이에 도달할 때까지 0의 단방향 MV를 제2 업데이트된 단방향 MV 후보 리스트에 주기적으로 추가함
805
제1 기하학적 파티션에 대한 단방향 MV 및 제2 기하학적 파티션에 대한 단방향 MV를 생성함
806

도면9


컴퓨팅 환경 910
프로세서 920
메모리 940
미리 결정된 소프트웨어 942
I/O 인터페이스 950
사용자 인터페이스 960

도면10


시스템 10
소스 디바이스 12
비디오 소스 18
비디오 인코더 20
출력 인터페이스 22
저장 디바이스 32
16
목적지 디바이스 14
디스플레이 디바이스 34
비디오 디코더 30
입력 인터페이스 28