



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0049366
(43) 공개일자 2024년04월16일

- (51) 국제특허분류(Int. Cl.)
H04N 19/543 (2014.01) H04N 19/119 (2014.01)
H04N 19/136 (2014.01) H04N 19/172 (2014.01)
H04N 19/70 (2014.01) H04N 19/86 (2014.01)
- (52) CPC특허분류
H04N 19/543 (2015.01)
H04N 19/119 (2015.01)
- (21) 출원번호 10-2024-7010050
- (22) 출원일자(국제) 2022년11월10일
심사청구일자 2024년03월26일
- (85) 번역문제출일자 2024년03월26일
- (86) 국제출원번호 PCT/US2022/049505
- (87) 국제공개번호 WO 2023/229632
국제공개일자 2023년11월30일
- (30) 우선권주장
63/345,329 2022년05월24일 미국(US)
(뒷면에 계속)

- (71) 출원인
텐센트 아메리카 엘엘씨
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
- (72) 발명자
가오 한
미국 94306 캘리포니아 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
자오 신
미국 94306 캘리포니아 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
(뒷면에 계속)
- (74) 대리인
유미특허법인

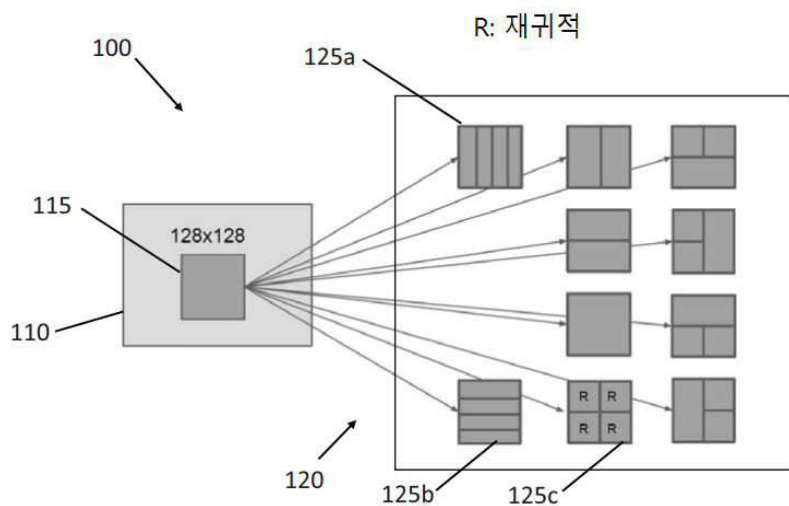
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 파티셔닝된 픽처의 예측의 비대칭 블렌딩을 위한 방법 및 장치

(57) 요약

비트스트림의 디코딩을 위한, 더 구체적으로는 디코딩을 위해 입력 이미지의 픽처 영역을 예측하기 위한 디코딩 디바이스가 제공된다. 픽처 영역은 파티셔닝 경계에 의해 적어도 제1 부분과 제2 부분으로 나뉘지고, 각 부분의 픽셀은 적절한 측정에 따라 예측된다. 그런 다음 블렌딩 마스크를 적용하여 예측된 픽셀에 대한 블렌딩된 영역을 생성하고, 픽셀을 수정하여 두 부분에 기반하여 픽처 영역의 완전한 예측을 생성한다. 블렌딩 마스크는 제1 임계 및 제2 임계에 기반하며, 이들은 파티셔닝 경계를 기준으로 정의되고 경계를 기준으로 비대칭 블렌딩을 생성하기 위해 서로 다른 값을 가질 수 있다. 적절한 임계 값에 기반하여, 픽처 영역의 예측, 조합 및 디코딩이 그 부분의 서로 다른 콘텐츠에 보다 적응적이다.

대표도 - 도1



(52) CPC특허분류

HOAN 19/136 (2015.01)

HOAN 19/172 (2015.01)

HOAN 19/70 (2015.01)

HOAN 19/86 (2015.01)

(72) 발명자

전 리엔-페이

미국 94306 캘리포니아주 팔로 알토 파크 블러바드
2747 텐센트 아메리카 엘엘씨 내

리우 산

미국 94306 캘리포니아주 팔로 알토 파크 블러바드
2747 텐센트 아메리카 엘엘씨 내

(30) 우선권주장

63/346,614 2022년05월27일 미국(US)

17/983,017 2022년11월08일 미국(US)

명세서

청구범위

청구항 1

디코딩 프로세스에서 픽처 영역을 예측하는 방법으로서,
 상기 픽처 영역을 예측하는 방법은 디코딩 디바이스의 적어도 하나의 프로세서에 의해 수행되며,
 상기 픽처 영역을 예측하는 방법은,
 적어도 제1 부분과 제2 부분으로 나뉜 픽처 영역을 포함하는 입력 이미지를 수신하는 단계;
 상기 픽처 영역을 상기 제1 부분과 상기 제2 부분으로 나누는 파티셔닝 경계를 식별하는 단계;
 상기 제1 부분에 대응하는 제1 임계(threshold)와 상기 제2 부분에 대응하는 제2 임계를 결정하는 단계;
 상기 제1 임계에 기반한 제1 블렌딩 마스크를 상기 제1 부분에 대한 예측된 픽셀에 적용하여 제1 블렌딩된 영역을 생성하고, 상기 제2 임계에 기반한 제2 블렌딩 마스크를 상기 제2 부분에 대한 예측된 픽셀에 적용하여 제2 블렌딩된 영역을 생성하는 단계 - 상기 제1 블렌딩 마스크는 상기 제2 블렌딩 마스크와 상이함 -; 및
 상기 제1 블렌딩된 영역과 상기 제2 블렌딩된 영역에 의해 수정된 상기 제1 부분과 상기 제2 부분을 포함하는 상기 픽처 영역에 대한 예측을 포함하여 상기 입력 이미지를 재구성하는 단계를 포함하는 픽처 영역을 예측하는 방법.

청구항 2

제1항에 있어서,
 상기 제1 임계와 상기 제2 임계는 각각 상기 파티셔닝 경계를 기준으로 정의되고,
 상기 제1 블렌딩 마스크 및 상기 제2 블렌딩 마스크 각각을 적용하는 것은,
 상기 제1 임계와 상기 제2 임계 중 적어도 하나로부터의 위치의 거리에 기반하여 가중치를 상기 픽처 영역에서 상기 위치에 있는 예측된 픽셀에 적용하는 것을 포함하는, 픽처 영역을 예측하는 방법.

청구항 3

제1항에 있어서,
 상기 제1 임계와 상기 제2 임계 중 적어도 하나에 대한 값은 상기 입력 이미지로부터 도출된 적어도 하나의 고려 사항(consideration)에 기반하고, 상기 적어도 하나의 고려 사항은 상기 픽처 영역을 둘러싼, 상기 입력 이미지의 적어도 하나의 샘플에 기반하는, 픽처 영역을 예측하는 방법.

청구항 4

제1항에 있어서,
 상기 제1 임계와 상기 제2 임계 중 적어도 하나에 대한 값은 복수의 후보 값 중 왜곡 비용이 가장 낮은 후보 값에 기반하는, 픽처 영역을 예측하는 방법.

청구항 5

제1항에 있어서,
 상기 제1 임계와 상기 제2 임계 각각은 복수의 선택 가능한 임계 값(threshold value)의 개개의 미리 정의된 리스트를 가지며, 상기 제1 임계와 상기 제2 임계 각각은 시그널링된 인덱스 쌍에서 지시된 상기 개개의 리스트 내 개개의 인덱스에 기반하여 결정되는, 픽처 영역을 예측하는 방법.

청구항 6

제1항에 있어서,

상기 제1 임계와 상기 제2 임계는 복수의 선택 가능한 임계 값 조합의 미리 정의된 리스트 내의 임계 값 조합에 대응하는 시그널링된 인덱스에 기반하여 결정되는, 픽처 영역을 예측하는 방법.

청구항 7

제1항에 있어서,

상기 제1 임계의 값과 상기 제2 임계의 값은 플래그에 따라 결정되어,

상기 플래그가 제1 논리 레벨일 때, 상기 제1 임계는 제1 값으로 설정되고 상기 제2 임계는 제2 값으로 설정되며,

상기 플래그가 제2 논리 레벨일 때, 상기 제1 임계는 상기 제2 값으로 설정되고 상기 제2 임계는 상기 제1 값으로 설정되는, 픽처 영역을 예측하는 방법.

청구항 8

제1항에 있어서,

상기 파티셔닝 경계는 각도 파라미터 및 오프셋 파라미터에 따라 기하학적으로 정의되는, 픽처 영역을 예측하는 방법.

청구항 9

제1항에 있어서,

상기 파티셔닝 경계는 미리 정의된 웨지 파티션 세트의 웨지 파티션에 따라 정의되는, 픽처 영역을 예측하는 방법.

청구항 10

제1항에 있어서,

상기 제1 임계의 값과 상기 제2 임계의 값이 같을 때, 상기 블렌딩은 대칭 적응적 블렌딩이고,

상기 제1 임계의 값과 상기 제2 임계의 값이 같지 않을 때, 상기 블렌딩은 비대칭 적응적 블렌딩인, 픽처 영역을 예측하는 방법.

청구항 11

디코딩 디바이스로서,

신호를 수신하도록 구성된 적어도 하나의 통신 모듈;

컴퓨터 프로그램 코드를 저장하도록 전기적으로 구성된 적어도 하나의 비휘발성 메모리; 및

상기 적어도 하나의 통신 모듈 및 상기 적어도 하나의 비휘발성 메모리에 작동 가능하게 연결되는 적어도 하나의 프로세서

를 포함하고,

상기 적어도 하나의 프로세서는 상기 컴퓨터 프로그램 코드에 의해 명령된 대로 작동하도록 구성되고, 상기 컴퓨터 프로그램 코드는:

상기 적어도 하나의 프로세서 중 적어도 하나가 상기 적어도 하나의 통신 모듈을 통해, 적어도 제1 부분과 제2 부분으로 나뉜 픽처 영역을 포함하는 입력 이미지를 수신하도록 구성된 입력 코드;

상기 적어도 하나의 프로세서 중 적어도 하나가 상기 픽처 영역을 상기 제1 부분과 상기 제2 부분으로 나누는 파티셔닝 경계를 식별하게 하도록 구성된 파티셔닝 코드;

상기 적어도 하나의 프로세서 중 적어도 하나가 상기 제1 부분에 대응하는 제1 임계 및 상기 제2 부분에 대응하

는 제2 임계를 결정하게 하도록 구성된 임계 코드;

상기 적어도 하나의 프로세서 중 적어도 하나가 상기 제1 임계에 기반한 제1 블렌딩 마스크를 상기 제1 부분에 대한 예측된 픽셀에 적용하여 제1 블렌딩된 영역을 생성하고, 상기 제2 임계에 기반한 제2 블렌딩 마스크를 상기 제2 부분에 대한 예측된 픽셀에 적용하여 제2 블렌딩된 영역을 생성하게 하도록 구성된 블렌딩 코드 - 상기 제1 블렌딩 마스크는 상기 제2 블렌딩 마스크와 상이함 -; 및

상기 적어도 하나의 프로세서 중 적어도 하나가 상기 제1 블렌딩된 영역과 상기 제2 블렌딩된 영역에 의해 수정된 상기 제1 부분과 상기 제2 부분을 포함하는 상기 픽처 영역에 대한 예측을 포함하여 상기 입력 이미지를 재구성하게 하도록 구성된 재구성 코드

를 포함하는, 디코딩 디바이스.

청구항 12

제11항에 있어서,

상기 제1 임계와 상기 제2 임계는 각각 상기 파티셔닝 경계를 기준으로 정의되고,

상기 제1 블렌딩 마스크 및 상기 제2 블렌딩 마스크 각각을 적용하는 것은,

상기 제1 임계와 상기 제2 임계 중 적어도 하나로부터의 위치의 거리에 기반하여 가중치를 상기 픽처 영역에서 상기 위치에 있는 예측된 픽셀에 적용하는 것

을 포함하는, 디코딩 디바이스.

청구항 13

제11항에 있어서,

상기 제1 임계와 상기 제2 임계 중 적어도 하나에 대한 값은 상기 입력 이미지로부터 도출된 적어도 하나의 고려 사항에 기반하고, 상기 적어도 하나의 고려 사항은 상기 픽처 영역을 둘러싼, 상기 입력 이미지의 적어도 하나의 샘플에 기반하는, 디코딩 디바이스.

청구항 14

제11항에 있어서,

상기 제1 임계와 상기 제2 임계 중 적어도 하나에 대한 값은 복수의 후보 값 중 왜곡 비용이 가장 낮은 후보 값에 기반하는, 디코딩 디바이스.

청구항 15

제11항에 있어서,

상기 제1 임계와 상기 제2 임계 각각은 복수의 선택 가능한 임계 값의 개개의 미리 정의된 리스트를 갖고, 상기 제1 임계와 상기 제2 임계 각각은 시그널링된 인덱스 쌍에서 지시된 상기 개개의 리스트 내 개개의 인덱스에 기반하여 결정되는, 디코딩 디바이스.

청구항 16

제11항에 있어서,

상기 제1 임계와 상기 제2 임계는 복수의 선택 가능한 임계 값 조합의 미리 정의된 리스트 내의 임계 값 조합에 대응하는 시그널링된 인덱스에 기반하여 결정되는, 디코딩 디바이스.

청구항 17

제11항에 있어서,

상기 제1 임계의 값과 상기 제2 임계의 값은 플래그에 따라 결정되어,

상기 플래그가 제1 논리 레벨일 때, 상기 제1 임계는 제1 값으로 설정되고 상기 제2 임계는 제2 값으로 설정되

며,

상기 플래그가 제2 논리 레벨일 때, 상기 제1 임계는 상기 제2 값으로 설정되고 상기 제2 임계는 상기 제1 값으로 설정되는, 디코딩 디바이스.

청구항 18

제11항에 있어서,

상기 파티셔닝 경계는 각도 파라미터 및 오프셋 파라미터에 따라 기하학적으로 정의되는, 디코딩 디바이스.

청구항 19

제11항에 있어서,

상기 파티셔닝 경계는 미리 정의된 웨지 파티션 세트의 웨지 파티션에 따라 정의되는, 디코딩 디바이스.

청구항 20

적어도 하나의 프로세서가 디코딩 프로세스에서 픽처 영역을 예측하는 방법을 수행하기 위한 명령어를 기록한, 컴퓨터가 판독 가능한 비일시적 기록 매체로서,

상기 픽처 영역을 예측하는 방법은,

적어도 제1 부분과 제2 부분으로 나뉜 픽처 영역을 포함하는 입력 이미지를 수신하는 단계;

상기 픽처 영역을 상기 제1 부분과 상기 제2 부분으로 나누는 파티셔닝 경계를 식별하는 단계;

상기 제1 부분에 대응하는 제1 임계와 상기 제2 부분에 대응하는 제2 임계를 결정하는 단계;

상기 제1 임계에 기반한 제1 블렌딩 마스크를 상기 제1 부분에 대한 예측된 픽셀에 적용하여 제1 블렌딩된 영역을 생성하고, 상기 제2 임계에 기반한 제2 블렌딩 마스크를 상기 제2 부분에 대한 예측된 픽셀에 적용하여 제2 블렌딩된 영역을 생성하는 단계 - 상기 제1 블렌딩 마스크는 상기 제2 블렌딩 마스크와 상이함 -; 및

상기 제1 블렌딩된 영역과 상기 제2 블렌딩된 영역에 의해 수정된 상기 제1 부분과 상기 제2 부분을 포함하는 상기 픽처 영역에 대한 예측을 포함하여 상기 입력 이미지를 재구성하는 단계

를 포함하는, 컴퓨터가 판독 가능한 비일시적 기록 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2022년 5월 24일에 출원된 미국 임시 특허 출원 번호 제63/345,329호, 2022년 5월 27일에 출원된 미국 임시 특허 출원 번호 제63/346,614호, 2022년 11월 8일에 출원된 미국 특허 출원 번호 제17/983,017호에 기반하고 이들에 대한 우선권을 주장하는 바이며, 각각의 내용은 그 전체가 본 명세서에 참조로 포함된다.

[0002] 본 개시는 일반적으로 비디오 인코딩 및 디코딩에 관한 것이며, 특히 비디오 비트스트림 인코딩의 예측된 파티셔닝된 블록에 비대칭 블렌딩을 적용하기 위한 방법 및 장치에 관한 것이다.

배경 기술

[0003] 비디오 인코딩 및 디코딩은 일반적으로 연결된 디바이스 및 디지털 미디어의 확산과 함께 널리 사용된다. AV1(AOMedia Video 1)은 인터넷을 통한 비디오 전송을 위해 설계된 개방형 비디오 코딩 포맷이다. AV1 프로젝트의 많은 컴포넌트는 이전 연구 노력에서 나온 것이다. AV1은 이전 코덱인 VP9와 같은 기존 솔루션에 비해 개선된 솔루션이지만, 보안 문제가 여전히 존재한다. 그러므로 추가적인 개선이 필요하다.

발명의 내용

[0004] 본 개시의 특정 실시예에 따르면, 디코딩 프로세스에서 픽처 영역(picture area)을 예측하는 방법이 제공된다. 이 방법은 디코딩 디바이스의 적어도 하나의 프로세서에 의해 수행된다. 이 방법은 적어도 제1 부분(part)과 제2 부분으로 나뉜 픽처 영역을 포함하는 입력 이미지를 수신하는 단계를 포함한다. 이 방법은 픽처 영역을 제1

부분과 제2 부분으로 나누는(divide) 파티셔닝 경계(partitioning boundary)를 식별하는 단계를 더 포함한다. 이 방법은 제1 부분에 대응하는 제1 임계(threshold) 및 제2 부분에 대응하는 제2 임계를 결정하는 단계를 더 포함한다. 이 방법은 제1 임계에 기반한 제1 블렌딩 마스크(blending mask)를 제1 부분에 대한 예측된 픽셀에 적용하여 제1 블렌딩된 영역(blended region)을 생성하고, 제2 임계에 기반한 제2 블렌딩 마스크를 제2 부분에 대한 예측된 픽셀에 적용하여 제2 블렌딩된 영역을 생성하는 단계를 더 포함하며, 제1 블렌딩 마스크는 제2 블렌딩 마스크와 상이하다. 이 방법은 제1 블렌딩된 영역과 제2 블렌딩된 영역에 의해 수정된 제1 부분과 제2 부분을 포함하는 픽처 영역에 대한 예측을 포함하여 입력 이미지를 재구성하는 단계를 더 포함한다.

[0005] 본 개시의 다른 실시예에 따르면, 디코딩 디바이스가 제공된다. 인코딩 디바이스는 신호를 수신하도록 구성된 적어도 하나의 통신 모듈, 컴퓨터 프로그램 코드를 저장하도록 전기적으로 구성된 적어도 하나의 비휘발성 메모리, 및 적어도 하나의 통신 모듈과 적어도 하나의 비휘발성 메모리에 작동 가능하게 연결된 적어도 하나의 프로세서를 포함한다. 적어도 하나의 프로세서는 컴퓨터 프로그램 코드에 의해 명령된 대로 작동하도록 구성된다. 컴퓨터 프로그램 코드는 적어도 하나의 프로세서 중 적어도 하나가 적어도 하나의 통신 모듈을 통해, 적어도 제1 부분과 제2 부분으로 나뉜 픽처 영역을 포함하는 입력 이미지를 수신하도록 구성되는 입력 코드를 포함한다. 컴퓨터 프로그램 코드는 적어도 하나의 프로세서 중 적어도 하나가 픽처 영역을 제1 부분과 제2 부분으로 나누는 파티셔닝 경계를 식별하게 하도록 구성된 파티셔닝(partitioning) 코드를 더 포함한다. 컴퓨터 프로그램 코드는 적어도 하나의 프로세서 중 적어도 하나가 제1 부분에 대응하는 제1 임계 및 제2 부분에 대응하는 제2 임계를 결정하게 하도록 구성된 임계 코드를 더 포함한다. 컴퓨터 프로그램 코드는 적어도 하나의 프로세서 중 적어도 하나가, 제1 임계에 기반한 제1 블렌딩 마스크를 제1 부분에 대한 예측된 픽셀에 적용하여 제1 블렌딩된 영역을 생성하고, 제2 임계에 기반한 제2 블렌딩 마스크를 제2 부분에 대한 예측된 픽셀에 적용하여 제2 블렌딩된 영역을 생성하게 하도록 구성된 블렌딩 코드를 더 포함하고, 제1 블렌딩 마스크는 제2 블렌딩 마스크와 상이하다. 컴퓨터 프로그램 코드는 적어도 하나의 프로세서 중 적어도 하나가 제1 블렌딩된 영역과 제2 블렌딩된 영역에 의해 수정된 제1 부분과 제2 부분을 포함하는 픽처 영역에 대한 예측을 포함하여 입력 이미지를 재구성하게 하도록 구성된 재구성 코드를 더 포함한다.

[0006] 본 개시의 또 다른 실시예에 따르면, 컴퓨터가 관독 가능한 비밀시적 기록 매체가 제공된다. 기록 매체에는 디코딩 프로세스에서 픽처 영역을 예측하는 방법을 수행하기 위해 적어도 하나의 프로세서에 의해 실행 가능한 명령어가 기록되어 있다. 이 방법은 적어도 제1 부분과 제2 부분으로 나뉜 픽처 영역을 포함하는 입력 이미지를 수신하는 단계를 포함한다. 이 방법은 픽처 영역을 제1 부분과 제2 부분으로 나누는 파티셔닝 경계를 식별하는 단계를 더 포함한다. 이 방법은 제1 부분에 대응하는 제1 임계 및 제2 부분에 대응하는 제2 임계를 결정하는 단계를 더 포함한다. 이 방법은 제1 임계에 기반한 제1 블렌딩 마스크를 제1 부분에 대한 예측된 픽셀에 적용하여 제1 블렌딩된 영역을 생성하고, 제2 임계에 기반한 제2 블렌딩 마스크를 제2 부분에 대한 예측된 픽셀에 적용하여 제2 블렌딩된 영역을 생성하는 단계를 더 포함하며, 제1 블렌딩 마스크는 제2 블렌딩 마스크와 상이하다. 이 방법은 제1 블렌딩된 영역과 제2 블렌딩된 영역에 의해 수정된 제1 부분과 제2 부분을 포함하는 픽처 영역에 대한 예측을 포함하여 입력 이미지를 재구성하는 단계를 더 포함한다.

[0007] 추가 측면은 다음의 설명에서 부분적으로 설명될 것이며, 부분적으로는 설명으로부터 명백해질 것이며, 또는 본 개시의 제시된 실시예의 실시예에 의해 실현될 수 있다.

도면의 간단한 설명

[0008] 본 개시의 특정한 예시적인 실시예의 특징, 측면 및 장점은 첨부 도면을 참조하여 아래에서 설명될 것이며, 여기서 유사한 참조 번호는 유사한 엘리먼트를 나타낸다.

도 1은 예시적인 실시예에 따른 AV1 트리 파티셔닝의 예시적인 예를 도시한다.

도 2는 예시적인 실시예에 따른 쿼드-트리 플러스 이진 트리 구조를 사용하여 블록을 분배하는 예시적인 예를 도시한다.

도 3은 예시적인 실시예에 따른 삼진 트리를 사용하는 블록 파티셔닝 구조의 예시적인 예를 도시한다.

도 4는 예시적인 실시예에 따른 공간적 모션 벡터 예측자를 도출하는 예시적인 작동을 도시한다.

도 5는 예시적인 실시예에 따른 시간적 모션 벡터 예측자의 예시적인 작동을 도시한다.

도 6은 예시적인 실시예에 따른 시간적 모션 예측자를 도출하기 위한 미리 정의된 블록 위치의 예시적인 세트를 도시한다.

- 도 7은 예시적인 실시예에 따른 단일 인터 예측 블록을 통해 모션 벡터 후보를 생성하는 예시적인 작동을 도시한다.
- 도 8은 예시적인 실시예에 따른 복합 인터 예측 블록을 통해 모션 벡터 후보를 생성하는 예시적인 작동을 도시한다.
- 도 9는 예시적인 실시예에 따른 모션 벡터 후보 बैं크를 업데이트하는 예시적인 작동을 도시한다.
- 도 10은 예시적인 실시예에 따른 모션 벡터 예측 리스트 구성 프로세스를 도시하는 흐름도이다.
- 도 11은 예시적인 실시예에 따른 복합 인터 예측 모드의 예시적인 작동을 도시한다.
- 도 12는 예시적인 실시예에 따른 시간적 보간 예측 모드의 예시적인 작동을 도시한다.
- 도 13은 예시적인 실시예에 따른 양자 매칭 기반 디코더측 모션 벡터 리파인먼트의 예시적인 작동을 도시한다.
- 도 14는 예시적인 실시예에 따른 모션 벡터 차이를 갖는 병합 모드의 예시적인 사용 사례를 도시한다.
- 도 15는 예시적인 실시예에 따른 지오메트리 파티셔닝 모드 예측의 예시적인 작동을 도시한다.
- 도 16은 예시적인 실시예에 따른 지오메트리 파티셔닝 모드 예측에서 마스크 생성의 예시적인 작동을 도시한다.
- 도 17은 예시적인 실시예에 따른 특수 복합 예측 모드에서 웨지 기반 예측을 위한 예시적인 코드북을 도시한다.
- 도 18은 예시적인 실시예에 따른 비대칭 블렌딩 생성의 예시적인 작동을 도시한다.
- 도 19는 예시적인 실시예에 따른 비대칭/대칭 블렌딩 마스크에 대한 적응적 임계 선택의 예시적인 작동을 도시한다.
- 도 20은 여기에 설명된 시스템 및/또는 방법의 실시예가 구현될 수 있는 디바이스의 예시적인 컴포넌트의 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0009] 예시적인 실시예에 대한 다음의 상세한 설명은 첨부 도면을 참조한다. 서로 다른 도면에서의 동일한 참조 번호는 동일하거나 유사한 엘리먼트를 식별할 수 있다.
- [0010] 전술한 개시는 예시 및 설명을 제공하지만, 개시된 정확한 형태로의 구현을 제한하거나 철저히 수행하려는 의도는 아니다. 위의 개시에 비추어 수정 및 변형이 가능하거나 구현의 실행을 통해 획득될 수 있다. 또한, 한 실시예의 하나 이상의 특징 또는 컴포넌트는 다른 실시예(또는 다른 실시예의 하나 이상의 특징)에 통합되거나 조합될 수 있다. 추가적으로, 아래 제공된 작동의 흐름도 및 설명에서, 하나 이상의 작동이 생략될 수 있고, 하나 이상의 작동이 추가될 수 있으며, 하나 이상의 작동이 동시에(적어도 부분적으로) 수행될 수 있으며, 하나 이상의 작동의 순서가 전환될 수 있음을 이해할 수 있다.
- [0011] 여기에 설명된 시스템 및/또는 방법은 서로 다른 형태의 하드웨어, 펌웨어, 또는 하드웨어와 소프트웨어의 조합으로 구현될 수 있다는 것이 명백할 것이다. 이러한 시스템 및/또는 방법을 구현하는 데 사용되는 실제 특수 제어 하드웨어 또는 소프트웨어 코드는 구현을 제한하지 않는다. 따라서, 시스템 및/또는 방법의 작동 및 행동 (behavior)은 특정 소프트웨어 코드를 참조하지 않고 여기에서 설명된다. 소프트웨어 및 하드웨어는 여기에서의 설명에 기반하여 시스템 및/또는 방법을 구현하도록 설계될 수 있다는 것이 이해된다.
- [0012] 특정 특징의 조합이 청구범위에 인용되거나 및/또는 명세서에 개시되어 있더라도, 이러한 조합은 가능한 구현의 개시를 제한하려는 의도는 아니다. 실제로, 이들 특징 중 다수는 청구 범위에 구체적으로 언급되지 않거나 및/또는 명세서에 개시되지 않은 방식으로 조합될 수 있다. 아래에 나열된 각 종속항은 단 하나의 청구항에만 직접적으로 의존할 수 있지만, 가능한 구현의 개시에는 청구항 세트의 다른 모든 청구항과 조합된 각 종속항이 포함된다.
- [0013] 명시적으로 설명하지 않는 한 여기에 사용된 엘리먼트, 동작(act) 또는 명령어는 중요하거나 필수적인 것으로 해석되어서는 안 된다. 또한, 여기에서 사용된 관사 "a" 및 "an"은 하나 이상의 항목을 포함하도록 의도되었으며, "하나 이상"과 상호 교환적으로 사용될 수 있다. 하나의 항목만 의도된 경우, "하나"라는 용어 또는 이와 유사한 언어가 사용된다. 또한, 여기에서 사용된 용어 "가지다", "갖다", "갖는", "포함하다", "포함하는" 등의 용어는 개방형 용어로 의도된다. 또한, "~에 기반하여"라는 문구는 달리 명시적으로 언급되지 않는 한 "적어도

부분적으로 ~에 기반하여"를 의미하도록 의도된다. 또한, "[A]와 [B] 중 적어도 하나" 또는 "[A] 또는 [B] 중 적어도 하나"와 같은 표현은 A만을 포함하거나, B만을 포함하거나, A와 B를 모두 포함하는 것으로 이해되어야 한다.

- [0014] 현재 인터넷을 통한 미디어 접근성이 확산되면서 네트워크 부하를 줄이기 위해 비디오 인코딩이 더욱 중요해졌다. 비디오 인코딩 및 디코딩을 위한 방법 및 장치가 개시된다.
- [0015] 인코딩 및 디코딩에서 블렌딩 마스크 또는 웨지 가중치 마스크(wedge weighting mask)는 대칭 블렌딩(symmetrical blending)을 사용할 수 있으며: 파티셔닝 경계 간의 가중치 임계는 같다. 이는 모든 콘텐츠 유형에 적합하지 않을 수 있다. 예를 들어, 예측된 이미지의 한 부분이 동질적이지만 다른 부분이 객체를 묘사하면, 동질적인 부분의 블렌딩은 객체를 포함하는 부분보다 더 선명할 수 있다. 여기서 개선(improvement)이 바람직하다.
- [0016] 개시된 방법 및 장치에서, 미리 정의된 대칭(즉, 하나의 임계) 블렌딩 설계 대신에, 설계는 파티셔닝 경계 주위에 상이한 블렌딩 임계: 예를 들어 2개의 미리 정의된 임계를 가질 수 있다. 블렌딩 마스크 또는 웨지 가중치 마스크는 이 두 임계에 기반하여 미리 또는 즉시(on-the-fly) 계산될 수 있다. 결과적인 비대칭 블렌딩 설계는 예를 들어 VVC(Versatile Video Coding) 및 후속 코덱의 GPM(Geometric Partitioning Mode)뿐만 아니라 AV1, AV2 및 후속 코덱의 웨지 기반 예측을 보완하는 데 사용될 수 있다.
- [0017] 도 1은 예시적인 실시예에 따른 AV1 파티션 트리(100)의 예시적인 예를 도시한다. 이미지(110)에 대한 파티션 트리(100)에서, 이미지(110)의 부분(115)(VP9/AV1 용어로 슈퍼블록(superblock)이라고 함)은, 각각 처리될 수 있는 다양한 파티션 패턴(예: 125a, 125b, 125c)에 따라 슈퍼블록(115)을 파티셔닝하는, 10방향(ten-way) 구조(120)로 확장된다. 직사각형의 파티셔닝을 이용한 파티션 패턴은 더 이상 세분화될 수 없지만, 파티션 패턴(125c)은 슈퍼블록(115)과 동일한 방식으로 파티셔닝될 수 있는 정사각형 패턴만으로 구성되어, 재귀적인 파티셔닝을 이룰 수 있다.
- [0018] 이러한 프로세스의 파티션 또는 블록을 코딩 트리 유닛(coding tree unit, CTU)이라고 지칭할 수도 있고, CTU가 집합적으로 표현하는 픽셀 또는 픽셀 데이터 유닛의 그룹을 코딩 트리 블록(coding tree block, CTB)이라고 지칭할 수 있다. 단일 CTU는 다수의 CTB를 나타낼 수 있으며, 여기서 각 CTB는 정보의 서로 다른 컴포넌트를 나타낸다(예: 밝기 정보에 대한 CTB 및 "빨간색", "녹색" 및 "파란색" 팩터(factor)와 같은 서로 다른 색상 컴포넌트에 대한 다수의 CTB).
- [0019] AV1은 VP9의 64×64 픽셀 슈퍼블록과 비교하여, 시작 슈퍼블록(115)의 가능한 최대 크기를 예를 들어 128×128 픽셀로 증가시킨다. 또한, 10방향 구조(120)는 VP9에는 존재하지 않았던 4:1 및 1:4 직사각형 파티션 패턴(125a, 125b)을 포함한다. 또한 특정 경우에 2×2 크로마 인터 예측이 가능해진다는 의미에서, AV1은 8×8 픽셀 레벨 아래의 파티션 사용에 더 많은 유연성을 추가한다.
- [0020] HEVC(High Efficiency Video Coding)에서는 다양한 로컬 특성에 적응하기 위해 코딩 트리라고 불리는 쿼드트리(quadtree) 구조를 이용하여 코딩 트리 유닛을 코딩 유닛(coding unit, CU)으로 분할(split)할 수 있다. 인터 픽처(시간적) 예측 또는 인트라 픽처(공간적) 예측을 이용하여 픽처 영역을 코딩할지의 여부에 대한 판정은 CU 레벨에서 이루어질 수 있다. 각 CU는 PU 분할 유형에 따라 1개, 2개 또는 4개의 예측 유닛(prediction unit, PU)으로 추가로 분할될 수 있다. 하나의 PU 내에서, 동일한 예측 프로세스가 적용될 수 있으며 관련 정보는 PU 단위로 디코더로 전송될 수 있다. PU 분할 유형에 기반하여 예측 프로세스를 적용하여 잔차 블록을 획득한 후, CU는 CU에 대한 코딩 트리와 같은 다른 쿼드트리 구조에 따라 변환 유닛(transform unit, TU)으로 파티셔닝될 수 있다. HEVC 구조에는 CU, PU 및 TU를 포함한 다수의 파티션 개념이 있다. HEVC에서는 CU 또는 TU가 정사각형 모양일 수 있고, PU는 인터 예측된 블록에 대해 정사각형 모양 또는 직사각형 모양일 수 있다. HEVC에서는 하나의 코딩 블록이 4개의 정사각형 서브 블록으로 추가로 분할될 수 있으며, 각 서브 블록, 즉 TU에 대해 변환이 수행될 수 있다. 각 TU는 RQT(Residual Quad-Tree)라고 하는 더 작은 TU로 재귀적으로(쿼드트리 분할을 이용하여) 추가로 분할될 수 있다. 픽처 경계에서, HEVC는 크기가 픽처 경계에 맞을 때까지 블록이 쿼드-트리 분할을 유지할 수 있도록 암시적 쿼드-트리 분할을 사용할 수 있다.
- [0021] 도 2는 예시적인 실시예에 따른 쿼드-트리 플러스 이진 트리(quad-tree plus binary tree, QTBT) 구조(210)를 이용하여 CTU(220)를 분배하는(portioning) 예시적인 예를 도시한다. QTBT 구조(210)는 쿼드트리 노드와 이진 트리 노드를 모두 포함한다. 도 2에서 실선은 브랜치(branch)와 리프(leaf), 그리고 노드(211a)와 같은 쿼드트리 노드에서의 나누기(division)에 따른 대응하는 블록 나누기를 지시하고(indicate), 점선은 브랜치와 리프 그리고 노드(211b)와 같은 이진 트리 노드에서의 나누기에 따른 대응하는 블록 나누기를 지시한다.

- [0022] 이진 트리 노드에서의 분할(splitting)은 대응하는 블록을 같은 크기의 두 서브 블록으로 나눈다. 각각의 분할 (즉, 리프가 아님) 이진 트리 노드(예: 노드 211b)에 대해, 플래그 또는 기타 표시(indicia)를 사용하여 어떤 분할 유형(즉, 수평 또는 수직)이 사용되는지를 지시할 수 있으며, 예를 들어 0은 수평 분할을 지시하고, 1은 수직 분할을 지시한다. 쿼드트리 노드(예: 노드(211a))에서의 분할은 대응하는 블록을 수평 및 수직으로 모두 같은 크기의 4개의 서브 블록으로 나누며, 따라서 분할 유형을 지시하는 플래그가 생략될 수 있다.
- [0023] 또한, QTBT 체계(scheme)는 루마와 크로마가 별도의 QTBT 구조를 가질 수 있는 유연성을 지원한다. P 슬라이스 및 B 슬라이스의 경우, 하나의 CTU에 있는 루마 CTB 및 크로마 CTB는 동일한 QTBT 구조를 공유할 수 있다. 그러나, I 슬라이스의 경우, 루마 CTB는 QTBT 구조에 의해 CU로 파티셔닝될 수 있고, 크로마 CTB는 상이한 QTBT 구조에 의해 크로마 CU로 파티셔닝될 수 있다. 이는 I 슬라이스의 CU가 루마 컴포넌트의 코딩 블록 또는 2개의 크로마 컴포넌트의 코딩 블록을 포함할 수 있고, P 슬라이스 또는 B 슬라이스의 CU가 세 가지 색상 컴포넌트의 코딩 블록을 모두 포함할 수 있음을 의미한다.
- [0024] HEVC에서는 모션 보상의 메모리 액세스를 줄이기 위해 작은 블록에 대한 인터 예측을 제한하므로, 4×8 및 8×4 블록에 대해서는 이중 예측(bi-prediction)이 지원되지 않고 4×4 블록에 대해서는 인터 예측이 지원되지 않는다. 특정 실시예에서 구현된 QTBT에서는 이러한 제한이 제거된다.
- [0025] HEVC에서는 다양한 로컬 특성에 적응하기 위해 코딩 트리라고 불리는 쿼드트리를 이용하여 CTU를 CU로 분할할 수 있다. 인터 픽처(시간적) 예측 또는 인트라 픽처(공간적) 예측을 이용하여 픽처 영역을 코딩할지의 여부에 대한 판정은 CU 레벨에서 이루어질 수 있다. 각 CU는 PU 분할 유형에 따라 1개, 2개 또는 4개의 PU로 추가로 분할될 수 있다. 하나의 PU 내에서, 동일한 예측 프로세스가 적용될 수 있으며 관련 정보는 PU 단위로 디코더로 전송될 수 있다. PU 분할 유형에 기반하여 예측 프로세스를 적용하여 잔차 블록을 획득한 후, CU는 CU에 대한 코딩 트리과 마찬가지로 다른 쿼드트리 구조에 따라 변환 유닛(transform unit, TU)으로 파티셔닝될 수 있다. 따라서 HEVC 구조는 CU, PU 및 TU를 포함하는 다수의 파티션 개념을 가질 수 있다.
- [0026] 도 2에 도시된 실시예에 따르면, QTBT 구조(210)는 다수의 파티션 유형의 개념을 제거하며, 즉, 이는 CU, PU 및 TU 개념의 분리를 제거하고 CU 파티션 모양에 대해 더 많은 유연성을 지원한다. QTBT 블록 구조에서, CU는 정사각형 또는 직사각형 모양을 가질 수 있다. 도 2에 도시된 바와 같이, 코딩 트리 유닛(CTU)(220)은 먼저 QTBT 구조(210)의 쿼드트리 노드(211a)에 따라 파티셔닝될 수 있다. 쿼드트리 노드(211a)의 브랜치는 이진 트리 노드(예: 노드(211b, 211c)) 또는 다른 쿼드트리 노드(예: 노드(211d))에 따라 추가로 파티셔닝될 수 있다. 이진 트리 분할에서는 대칭(symmetric) 수평 분할과 대칭 수직 분할의 두 가지 분할 유형이 있을 수 있다. 이진 트리 리프 노드는 코딩 유닛(CU)으로 지정(designate)될 수 있으며, 이 세그멘테이션(segmentation)은 임의의 추가 파티셔닝 없이 예측 및 변환 처리에 사용될 수 있다. 이는 QTBT 코딩 블록 구조에서 CU, PU, TU가 동일한 블록 크기를 가질 수 있음을 의미한다.
- [0027] 특정 실시예에서, CU는 서로 다른 색상 컴포넌트의 코딩 블록(coding block, CB)을 포함할 수 있으며(예: 4:2:0 크로마 포맷의 P 슬라이스 및 B 슬라이스의 경우에, 하나의 CU는 하나의 루마 CB 및 두 개의 크로마 CB를 포함할 수 있음), 또는 다르게는 단일 컴포넌트의 CB를 포함할 수 있다(예: I 슬라이스의 경우, 하나의 CU는 하나의 루마 CB 또는 두 개의 크로마 CB를 포함할 수 있음).
- [0028] QTBT 파티셔닝 체계에 대해 다음 파라미터가 정의된다.
- [0029] - CTU 크기: 쿼드트리의 루트 노드 크기, HEVC와 동일한 개념
- [0030] - MinQTSIZE: 허용되는 최소 쿼드트리 리프 노드 크기
- [0031] - MaxBTSIZE: 허용되는 최대 이진 트리 루트 노드 크기
- [0032] - MaxBTDepth: 허용되는 최대 이진 트리 깊이
- [0033] - MinBTSIZE: 허용되는 최소 이진 트리 리프 노드 크기
- [0034] QTBT 파티셔닝 구조의 예시적인 구현에서, CTU(220) 크기는 2개의 대응하는 64×64 크로마 샘플 블록을 갖는 128×128 루마 샘플로 설정될 수 있고, MinQTSIZE는 16×16으로 설정되며, MaxBTSIZE는 64×64로 설정되고, MinBTSIZE(너비 및 높이 모두에 대해)는 4×4로 설정되며, MaxBTDepth는 4로 설정된다.
- [0035] 이러한 구현에서, 쿼드트리 파티셔닝은 쿼드트리 루트 노드(211a)로 표현되는 CTU(220)에 적용되어 쿼드트리 리프 노드(211b, 211c, 211d, 211e)를 생성한다. 쿼드트리 리프 노드(211b, 211c, 211d, 211e)는 16×16(즉,

MinQTSize)부터 128×128 (즉, CTU 크기)까지의 크기를 가질 수 있다. 리프 쿼드트리 노드 크기가 128×128 이면, 크기가 MaxBTSIZE(예: 64×64)를 초과하므로 이진 트리에 의해 더 이상 분할되지 않는다. 그렇지 않으면, 리프 쿼드트리 노드는 QTBT 파티셔닝 구조(210)에 의해 추가로 파티셔닝될 수 있다. 따라서 쿼드트리 리프 노드(211b)는 이진 트리 깊이가 0인 이진 트리의 루트 노드으로도 처리될 수 있다.

[0036] 이진 트리 깊이가 MaxBTDepth(즉, 4)에 도달할 때, 더 이상 분할이 고려되지 않는다. 이진 트리 노드의 너비가 MinBTSIZE(즉, 4)와 같을 때, 더 이상 수평 분할이 고려되지 않는다. 마찬가지로, 이진 트리 노드의 높이가 MinBTSIZE와 같을 때, 더 이상 수직 분할이 고려되지 않는다.

[0037] 분할이 중단되면, QTBT 파티셔닝 구조(210)의 최종 리프 노드(예: 리프 노드(211f))는 예측 및 변환 처리에 의해 추가로 처리될 수 있다. 특정 실시예에서, 최대 CTU 크기는 256×256 루마 샘플이다.

[0038] 도 3은 예시적인 실시예에 따른 VVC 다중 유형 트리(Multi-type-tree, MTT) 구조와 같은 삼진 트리(ternary tree)를 사용하는 블록 파티셔닝 구조의 예시적인 예를 도시한다. 이진 트리 노드에 사용된 것과 유사한 플래그 또는 표시를 사용하여 파티셔닝 구조에 삼진 트리의 사용을 추가하면, 위의 QTBT 파티셔닝 구조로 이네이블된(enabled) 파티션외에도 수직(310) 및 수평(320) 중앙측 트리플 트리 파티션 모두를 이네이블한다. 트리플 트리 파티셔닝은 쿼드트리 및 이진 트리 파티셔닝을 보완하며: 트리플 트리 파티셔닝은 쿼드트리 또는 이진 트리 파티션으로 나뉘는 블록의 중앙에 있는 객체를 캡처할 수 있다. 트리플 트리 파티션의 너비와 높이는 각각 2의 거듭제곱이 될 수 있으므로 추가 변환이 필요하지 않다.

[0039] 이론적으로 트리 순회(traversal)의 복잡도는 T^D 이며, 여기서 T는 분할 유형의 수를 나타내고 D는 트리의 깊이이다. 이와 같이, 복잡도 감소의 이유로 트리는 2레벨 트리(D=2)일 수 있다.

[0040] 도 4는 예시적인 실시예에 따라 공간적 모션 벡터 예측자를 도출하는 예시적인 작동을 도시한다. 공간적 모션 벡터 예측자(spatial motion vector predictor, SVMVP)는 그 자체로 모션 벡터의 형태를 취할 수도 있고 그렇지 않으면 모션 벡터를 포함할 수 있다. SVMVP는 현재 블록(410)에 이웃하는 블록으로부터 도출될 수 있다. 보다 구체적으로, SVMVP는 상단(top) 측 및 왼쪽 측에서 현재 블록(410)에 인접하거나 가까운 공간적 이웃 블록(420)으로부터 도출될 수 있다. 예를 들어, 도 4에서, 블록은 현재 블록(410) 바로 위의 3개의 블록 행에 있으면, 또는 현재 블록(410) 바로 왼쪽의 3개의 블록 열에 있으면, 또는 현재 블록(410)의 상단에 바로 인접한 행의 바로 왼쪽이나 오른쪽에 있으면, 공간적 이웃 블록(420)이다. 공간적 이웃 블록(420)은 현재 블록(410)보다 작은 정규 크기(regular size)일 수 있다. 예를 들어, 도 4에서 현재 블록(410)은 32×32 블록이고, 각 공간적 이웃 블록(420)은 8×8 블록이다.

[0041] 공간적 이웃 블록(420)은 현재 블록과 동일한 참조 프레임 인덱스와 연관된 하나 이상의 모션 벡터(motion vector, MV)를 찾기 위해 검사될 수 있다. 예를 들어 검사(examination) 순서에 따라 라벨이 지정된 도 4에 도시된 블록 세트에 따라 루마 블록에 대해 공간적 이웃 블록이 검사될 수 있다. 즉: (1) 상단 인접 행을 왼쪽에서 오른쪽으로 확인한다(check). (2) 왼쪽 인접 열을 상단에서 하단(bottom)으로 확인한다. (3) 우측 상단의 이웃 블록을 확인한다. (4) 왼쪽 상단 블록 이웃 블록을 확인한다. (5) 제1 상단 비인접(non-adjacent) 행을 왼쪽에서 오른쪽으로 확인한다. (6) 제1 왼쪽 비인접 열을 상단에서 하단으로 확인한다. (7) 제2 상단 비인접 행을 왼쪽에서 오른쪽으로 확인한다. (8) 제2 왼쪽 비인접 열을 상단에서 하단으로 확인한다.

[0042] "인접한" 블록(즉, 블록 세트 1-3의 블록)으로부터 도출된 "인접한" 공간적 MV 예측자에 대한 후보는 여기서 추가로 설명될 시간적 모션 벡터 예측자(temporal motion vector predictor, TMVP)인 시간적 MV 예측자들에 대한 후보보다 먼저 MV 예측자 리스트에 포함될 수 있으며, "비인접" 블록(외부 블록으로도 알려져 있으며, 즉 블록 세트 4-8의 블록)으로부터 도출된 "비인접" 공간적 MV 예측자에 대한 후보는 시간적 MV 예측자에 대한 후보 이후에 MV 예측자 리스트에 추가된다.

[0043] 일 실시예에서, 각 SVMVP 후보는 현재 블록과 동일한 참조 픽처를 갖는다. 예를 들어, 현재 블록(410)이 하나의 참조 픽처를 갖는다고 가정하자. MV 후보가 현재 블록의 참조 픽처와 동일한 단일 참조 픽처를 가지면, 이 MV 후보는 MV 예측자 리스트에 포함될 수 있다. 마찬가지로, MV 후보가 복수의 참조 픽처를 갖고 그 중 하나의 참조 픽처가 현재 블록의 참조 픽처와 동일하면, 이 MV 후보는 MV 예측자 리스트에 포함될 수 있다. 다만, 현재 블록(410)이 복수의 참조 픽처를 가지면, MV 후보는 MV 후보가 현재 블록(410)의 참조 픽처 각각에 대해 동일한 대응하는 참조 픽처를 갖는 경우에만 MV 예측자 리스트에 포함될 수 있다.

[0044] 도 5는 예시적인 실시예에 따른, 시간적 모션 벡터 예측자(temporal motion vector predictor, TMVP) 세트의 예시적인 작동을 도시한다. TMVP는 참조 프레임에서 동일 위치에 있는(collocated) 블록을 이용하여 도출될 수

있다. TMVP를 생성하기 위해, 먼저 하나 이상의 참조 프레임의 하나 이상의 MV가 개개의 참조 프레임과 연관된 참조 인덱스와 함께 저장될 수 있다. 이후, 현재 프레임의 8×8 블록 각각에 대해, 8×8 블록을 통과하는 궤적을 갖는 참조 프레임의 MV 벡터를 식별하여 참조 프레임 인덱스와 함께 시간적 MV 버퍼에 저장할 수 있다. 단일 참조 프레임을 사용하는 인터 예측의 경우, 참조 프레임이 "앞쪽(forward)" 참조 프레임인지 "뒤쪽(backward)" 참조 프레임인지(즉, 각각 현재 프레임보다 프레임 시퀀스에서 더 늦거나 더 빠른) 여부에 관계없이, MV는 미래 프레임의 시간적 모션 벡터 예측을 수행하기 위해 8×8 유닛으로 저장된다. 복합 인터 예측(compound inter prediction)의 경우, "앞쪽" 참조 프레임의 MV는 미래 프레임의 시간적 모션 벡터 예측을 수행하기 위해 8×8 유닛으로 저장될 수 있다.

[0045] TMVP를 생성하는 프로세스에 대한 예시적인 실시예는 다음과 같은 작동을 따를 수 있다. 이 예에서, 초기 참조 프레임(510)의 참조 모션 벡터(550)(MVref라고도 라벨링됨)는 초기 참조 프레임(510)으로부터 그 자체가 초기 참조 프레임(510)의 참조 프레임인 나중(later) 참조 프레임(540)을 가리킨다. 그렇게 함으로써, 현재 프레임(520)의 8×8 블록(570)(회색 점으로 음영 처리됨)을 통과한다. MVref(550)는 현재 블록(570)과 연관된 시간적 MV 버퍼에 저장될 수 있다. 시간적 MV 예측자(500)를 도출하기 위한 모션 투영(projection) 프로세스 동안, 나중 참조 프레임(예: 프레임(530, 540))은 미리 정의된 순서로 스캐닝될 수 있다. 예를 들어, AV1 표준에서 정의된 프레임 라벨을 사용하면 스캐닝 순서는: LAST_FRAME, BWDREF_FRAME, ALTREF_FRAME, ALTREF2_FRAME 및 LAST2_FRAME이 될 수 있다. 일 실시예에서, (스캐닝 순서에서) 더 높은 인덱스 참조 프레임으로부터의 MV는 (스캐닝 순서에서) 더 낮은 인덱스 참조 프레임에 의해 할당된 이전에 식별된 MV를 대체하지 않는다.

[0046] 마지막으로, 미리 정의된 블록 좌표가 주어지면, 시간적 MV 버퍼에 저장된 연관된 MV가 식별되어 현재 블록(570)에 투영되어, 현재 블록(570)으로부터 인접한 참조 프레임(530)을 가리키는 시간적 MV 예측자(560)(또한 MV0으로 라벨링됨)를 도출할 수 있다.

[0047] 도 6은 예시적인 실시예에 따른, 16×16 블록의 시간적 모션 예측자를 도출하기 위한 미리 정의된 블록 위치(600)의 예시적인 세트를 도시한다. 유효한 시간적 MV 예측자에 대해 최대 7개의 블록을 확인할 수 있다. 도 6에서 블록은 B0-B6으로 라벨링되어 있다. 도 4를 참조하여 언급한 바와 같이, 시간적 MV 예측자에 대한 후보는 인접한 공간적 MV 예측자에 대한 후보보다 이후에 그리고 비인접 공간적 MV 예측자에 대한 후보보다 먼저 확인되어 제1 MVP 리스트에 배치될 수 있다. 이후, MV 예측자(MV predictor, MVP) 도출을 위해, 모든 공간적 MVP 후보 및 시간적 MVP 후보를 풀링(pool)할 수 있으며, 각 후보에는 공간적 이웃 블록 및 시간적 이웃 블록을 스캐닝하는 동안 결정된 가중치가 할당될 수 있다. 연관된 가중치에 기반하여, 후보를 정렬하고(sort) 순위를 매길 수 있으며, 최대 4명의 후보자를 식별하여 제2 MVP 리스트에 배치할 수 있다. MVP의 제2 리스트는 동적 참조 리스트(dynamic reference list, DRL)라고도 하며, 동적 MV 예측 모드에서 추가로 사용될 수 있다.

[0048] DRL이 가득 차지 않으면, 추가 검색(extra search)이 수행될 수 있으며, 결과로 나오는 추가(extra) MVP 후보는 DRL을 채우는 데 사용된다. 추가 MVP 후보에는 예를 들어 전역(global) MV, 제로(zero) MV, 스케일링 없는 조합된 복합 MV 등이 포함될 수 있다. DRL에 있는 인접 SMVP 후보, TMVP 후보 및 비인접 SMVP 후보가 다시 재정렬(reorder)될 수 있다. AV1과 AVM은 모두 예를 들어, 각 후보의 가중치에 기반하여 재정렬을 가능하게 한다. 현재 블록과 후보 블록의 중첩된 영역(overlapped area)에 따라 후보의 가중치가 미리 정의될 수 있다.

[0049] 도 7은 예시적인 실시예에 따른 단일 인터 예측 블록을 통해 신규 MV 후보를 생성하는 예시적인 작동을 도시한다. 이웃 블록의 참조 프레임이 현재 블록의 참조 프레임과 상이하지만 MV들이 동일한 방향에 있을 때, 현재 블록의 모션 벡터에 대해 MVP를 형성하기 위해 시간적 스케일링 알고리즘을 활용하여 그의 MV를 그 참조 프레임에 맞게 스케일링할 수 있다. 도 7의 예에서, 현재 프레임(701)에서 현재 블록(710)의 이웃 블록(750)으로부터의 모션 벡터(740)(도 7에서 mv1로 라벨링됨)는 참조 프레임(703)에서 동일 위치에 있는 이웃 블록(760)을 가리킨다. 모션 벡터(740)는, 시간적 스케일링을 이용하여, 다른 참조 프레임(702)에서 동일 위치에 있는 현재 블록(720)을 가리키는 현재 블록(710)의 모션 벡터(730)(또한 도 7에서 mv0으로 라벨링됨)에 대한 MVP를 도출하기 위해 활용될 수 있다.

[0050] 도 8은 예시적인 실시예에 따른 복합 예측 블록을 통해 신규 MV 후보를 생성하는 예시적인 작동을 도시한다. 도 8의 예에서, 구성된 MV(860, 870)는 현재 프레임(802)의 현재 블록(810)의 개개의 서로 다른 이웃 블록(820, 830)에서부터 참조 프레임(803 및 801)으로 가리킨다. 구성된 MV(860, 870)(또한 도 8에서 mv2 및 mv3으로 라벨링됨)의 참조 프레임(803 및 801)은 현재 블록(810)에 대한 것과 동일할 수 있다. 복합 인터 예측은 도 7에서와 같이 결정될 수 있는 현재 블록(710)의 구성된 MV(840, 850)(또한 도 8에서 mv0 및 mv1로 라벨링됨)에 대한 MVP를 도출할 수 있다.

- [0051] 도 9는 예시적인 실시예에 따라 모션 벡터 후보 뱅크(bank)(920)를 업데이트하는 예시적인 작동을 도시한다. 이 뱅크(920)는 처음에 CWG-B023에서 제안되었으며, 이는 여기에 그 전체가 포함된다.
- [0052] 뱅크 업데이트 프로세스는 슈퍼블록(910)에 기반할 수 있다. 즉, 각 슈퍼블록(예: 슈퍼블록(910a))이 코딩된 후, 슈퍼블록 내부의 각 코딩 블록에 의해 사용되는 제1 후보 MV 세트(예를 들어, 처음 64개의 후보)가 뱅크(920)에 추가될 수 있다. 업데이트하는 동안, 푸르닝(pruning)도 포함될 수 있다.
- [0053] 슈퍼블록에 대한 참조 MV 후보 스캐닝이 완료된 후, 후보 리스트에 열린(open) 슬롯이 있으면, 코덱은 추가적인 MV 후보에 대해 (매칭되는 참조 프레임 유형을 갖는 버퍼에서) MV 후보 뱅크(920)를 참조할 수 있다. 버퍼의 끝에서 뒤로 거꾸로 이동하여, 뱅크 버퍼의 MV가 리스트에 아직 존재하지 않으면 후보 리스트에 추가될 수 있다. 보다 구체적으로, 각각의 버퍼는 단일 인터 모드 및 복합 인터 모드를 각각 커버하는, 단일 또는 한 쌍의 참조 프레임에 대응하는 고유한 참조 프레임 유형에 대응할 수 있다. 모든 버퍼의 크기는 동일할 수 있다. 가득 찬 버퍼에 신규 MV가 추가될 때, 신규 MV를 위한 공간을 확보하기 위해 기존 MV가 제거될 수 있다.
- [0054] 코딩 블록은 AV1 참조 MV 리스트 생성을 통해 획득한 것 외에, MV 후보 뱅크(920)를 참조하여 참조 MV 후보를 수집할 수 있다. 슈퍼블록을 코딩한 후, MV 뱅크는 슈퍼블록의 코딩 블록에 의해 사용된 MV로 업데이트될 수 있다.
- [0055] AV1을 사용하면 프레임을 타일(tile)로 나눌 수 있으며, 각 타일에는 복수의 슈퍼블록이 포함된다. 각 타일은 서로 다른 프로세서에서 병렬로 처리될 수 있다. 후보 뱅크와 관련하여, 각 타일은 타일 내의 모든 슈퍼블록에 의해 활용되는 독립적인 MV 후보 뱅크를 가질 수 있다. 각 타일의 인코딩 시작 시, 대응하는 뱅크가 비워진다. 이후 그 타일 내의 각 슈퍼블록을 인코딩하는 동안 뱅크의 MV가 MV 참조 후보로 사용될 수 있다. 각 슈퍼블록을 인코딩한 후, 뱅크는 위에서 설명한 바와 같이 업데이트될 수 있다.
- [0056] 뱅크 업데이트 및 참조를 위한 뱅크 업데이트 및 참조 프로세스의 특정 실시예는 여기에서 나중에 설명될 것이다.
- [0057] 도 10은 예시적인 실시예에 따라 임의의 비디오 입력에 대한 모션 벡터 예측 리스트 구성의 프로세스를 도시하는 흐름도이다. 인접 SMVP, TMVP, 비인접 SMVP 후보는 예를 들어 도 4 및 도 5를 참조하여 앞서 논의된 프로세스에 의해, 각각 S1010, S1020, S1030에서 생성될 수 있다. 다음으로, 후보들은 예를 들어 도 6을 참조하여 앞서 논의된 프로세스에 의해 S1040에서 정렬되거나 재정렬될 수 있다. E또한 MVP 후보는 예를 들어 도 7 및 도 8을 참조하여 앞서 논의된 프로세스에 의해 S1050에서 도출될 수 있다. 필요하다면, 추가 MVP 후보는 예를 들어 도 6을 참조하여 앞서 논의된 프로세스에 의해 S1060에서 추가 검색에 의해 결정될 수 있거나, 예를 들어 도 9를 참조하여 앞서 논의된 프로세스에 의해 S1070에서 참조 뱅크로부터 검색될 수 있다.
- [0058] 도 11은 예시적인 실시예에 따른 복합 인터 예측 모드의 예시적인 작동을 도시한다.
- [0059] 복합 인터 모드는 복수의 서로 다른 참조 프레임의 가설(hypotheses)을 조합하는 것에 의해 블록의 예측을 생성할 수 있다. 예를 들어, 도 11의 예에서, 현재 프레임(1110)의 블록(1111)은 이웃 참조 프레임(1120a, 1120b)의 모션 벡터(1130a, 1130b)(또한 도 11에서 mv0, mv1로 라벨링됨)에 의해 예측된다. 이웃 참조 프레임(1120a, 1120b)은 바로 이웃(즉, 시퀀스에서 현재 프레임(1110) 바로 앞과 뒤의 프레임)일 수 있지만, 이것이 필수 요건은 아니다. 각 블록에 대한 모션 정보 컴포넌트(예: 모션 벡터(1130a, 1130b))는 오버헤드로서 비트스트림에서 송신될 수 있다.
- [0060] 그러나 모션 벡터는 일반적으로 이력 모션 벡터 또는 공간적 및 시간적 이웃으로부터의 예측자를 이용하여 잘 예측될 수 있지만, 모션 정보에 사용되는 바이트는 많은 콘텐츠와 애플리케이션에서 여전히 상당히 중요할 수 있다.
- [0061] 도 12는 예시적인 실시예에 따른 시간적 보간 예측(temporal interpolated prediction, TIP) 모드의 예시적인 작동을 도시한다.
- [0062] 도 12의 예에서, 참조 프레임(1220a, 1220b)의 정보는 간단한 보간 프로세스를 이용하여 현재 프레임(1210)과 동일한 시간 인스턴스로 조합되고 투영될 것이다. 다수의 TIP 모드가 지원될 수 있다. 하나의 TIP 모드에서는 보간된 프레임, 즉 "TIP 프레임"(1210')이 추가적 참조 프레임으로 사용될 수 있다. 현재 프레임(1210)의 코딩 블록은 TIP 프레임(1210')을 직접 참조하고 단일 인터 예측 모드의 오버헤드 비용(cost)만으로 두 개의 서로 다른 참조에서 나오는 정보를 활용할 수 있다. 다른 TIP 모드에서, TIP 프레임(1210')은 임의의 다른 전통적인 코딩 단계를 건너뛰면서 현재 프레임(1210)에 대한 디코딩 프로세스의 출력으로 직접 할당될 수 있다. 이 모드는

특히 비트레이트가 낮은 애플리케이션에 대해 상당한 코딩 및 단순화 이점을 제공할 수 있다.

[0063] FRUC(Frame Rate Up Conversion)와 같이 두 참조 프레임 사이에서 프레임을 보간하는 기존 기술이 있지만, 복잡도와 압축 품질 간의 적절한 균형을 달성하는 것이 신규 코딩 도구를 설계할 때 중요한 제약이 될 수 있다. 위에서 개시된 이 방법은 간단하며, 임의의 추가적인 모션 검색을 수행할 필요 없이 참조 프레임에서 이미 사용 가능한 모션 정보를 재사용한다. 시뮬레이션 결과는 이 간단한 방법이 복잡도가 낮은 구현으로 좋은 품질을 달성할 수 있음을 보여준다.

[0064] 도 12의 예에서, TIP 모드 작동은 현재 프레임(1210)에 대응하는 TIP 프레임(1210')을 생성하는 것에 의해 시작된다. 그 다음, TIP 프레임(1210')은 현재 프레임(1210)에 대한 추가 참조 프레임으로 사용되거나 현재 프레임(1210)에 대한 디코더의 재구성된 출력으로 직접 할당될 수 있다. 디코더 측에서, TIP 모드에서 코딩된 블록은 즉시 생성될 수 있으므로, 디코더에서 전체 TIP 프레임(1210')을 생성할 필요가 없어 디코딩 시간과 처리를 절약한다. 이는 디코더의 원패스 디코딩 파이프라인과도 호환되므로 하드웨어 구현에 좋다.

[0065] 프레임 레벨 TIP 모드는 신택스 엘리먼트를 사용하여 지시될 수 있다. tip_frame_mode 파라미터에 대한 값으로 지시되는 모드의 예가 다음 표에 나와 있다.

표 1

tip 모드 예시

[0066]

tip_frame_mode	의미
0	이 프레임에서 TIP 모드를 디스에이블함
1	TIP 프레임을 추가적 참조 프레임으로 사용하며, 일반적으로 현재 프레임을 코딩
2	TIP 프레임을 직접 출력하며, 현재 프레임을 코딩하지 않음

[0067] 두 프레임 사이의 중간 프레임을 보간하기 위한 간단한 보간 방법이 개시되며, 이는 이용 가능한 참조로부터 모션 벡터를 완전히 재사용할 수 있다. 사소한 수정 후에 TMVP(temporal motion vector predictor) 프로세스에도 동일한 모션 벡터가 사용될 수 있다. 이 프로세스에는 세 가지 작동이 포함될 수 있다: 1. 수정된 TMVP 필드의 투영을 통해 TIP 프레임에 대한 대략적인(coarse) 모션 벡터 필드를 생성한다. 2. 홀(hole)을 채우고 평활화 작동을 사용하여 대략적인 모션 벡터 필드를 리파인한다(refine). 3. 리파인된 모션 벡터 필드를 사용하여 TIP 프레임을 생성한다. 디코더 측에서는 TIP 모드로 코딩된 블록이 전체 TIP 프레임을 생성하지 않고도 즉시 생성될 수 있다.

[0068] 그러나, 본 개시에서 논의된 다른 특징과 조합하여 다른 적합한 보간 방법이 대체될 수 있으며, 이는 본 개시의 범위 내에 있다는 점에 유의한다.

[0069] 도 13은 예시적인 실시예에 따른 양자 매칭 기반 디코더측 모션 벡터 리파인먼트의 예시적인 작동을 도시한다. VVC(Versatile Video Coding)는 이전에 디코딩된 픽처를 2개의 참조 픽처 리스트(1320a, 1320b)로 배포할 수 있다. 이들 이전에 디코딩된 픽처들은 현재 픽처(1310)를 예측하기 위한 참조로서 사용될 수 있다. 도 13의 예에서는 디스플레이 순서에 따라 현재 픽처(1310) 이전의 참조 픽처가 '과거' 참조 픽처 리스트(1320a)에 할당될 수 있으며, 현재 픽처(1310)보다 나중의 참조 픽처가 "미래" 참조 픽처 리스트(1320b)에 할당될 수 있다. 각 리스트(도시되지 않음)에 대한 대응하는 참조 픽처 인덱스는 각 리스트의 어떤 픽처가 현재 픽처(1310)의 현재 블록(1311)을 예측하는 데 사용되는지를 지시한다. 양방향 예측(bidirectional prediction)의 경우, 과거 참조 픽처 리스트(1320a)와 미래 참조 픽처 리스트(1320b)에 대한 개개의 MV(1331a, 1331b)를 사용하여 예측된 2개의 예측 블록(1321a, 1321b)이 조합되어 단일(single) 예측 신호를 획득할 수 있다.

[0070] 병합 모드로 모션 정보를 코딩하면, 참조 픽처 인덱스 및 이웃 블록의 MV가 현재 블록(1311)에 그대로 적용될 수 있다. 그러나 이는 현재 블록(1311)을 정확하게 예측하지 못할 수 있다.

[0071] 디코더측 정보만을 포함하는 것에 의해, 병합 모드 코딩된 블록의 정밀도를 증가시키기 위해 디코더측 모션 벡터 리파인먼트(decoder-side motion vector refinement, DMVR) 알고리즘이 사용될 수 있다. DMVR 알고리즘이 블록(1311, 1321a, 1321b)에 적용될 때, 병합 모드로부터 도출된 MV(1331a, 1331b)가 DMVR에 대한 "초기" MV로 설정될 수 있다.

[0072] 그러면 DMVR은 블록 매칭에 의해 초기 MV(1331a, 1331b)를 추가로 리파인할 수 있다. 두 참조 픽처 모두에서, 초기 MV가 가리키는 블록(1321a, 1321b) 주변의 후보 블록을 검색하여 양자 매칭(bilateral match)을 수행할 수

있다. 최상의 매칭된(best-matched) 블록(1323a, 1323b)은 최종 예측된 신호를 생성하는 데 사용될 수 있으며, 이러한 신규 예측 블록(1323a, 1323b)을 가리키는 신규 MV(1333a, 1333b)는 각각 초기 MV(1331a, 1331b)에 대응하는 "리파인된" MV로 설정될 수 있다. VVC에서 채택한 양자 매칭 기반 방법, 양방향 템플릿 매칭 기반 방법, 템플릿 매칭과 같은, DMVR에 적합한 많은 블록 매칭 방법이 연구되어 왔다.

[0073] 양자 매칭 기반 DMVR에서, 초기 MV가 가리키는 블록 쌍(1321a, 1321b)을 초기 블록 쌍으로 정의할 수 있다. 초기 블록 쌍(1321a, 1321b)의 왜곡 비용(distortion cost)은 초기 비용으로 계산될 수 있다. 초기 블록 쌍(1321a, 1321b) 주변의 블록들은 DMVR 후보 블록 쌍으로 사용될 수 있다. 각 블록 쌍은 과거 참조 픽처 리스트(1320a)의 참조 픽처로부터의 하나의 예측된 블록과 미래 참조 픽처 리스트(1320b)의 참조 픽처로부터의 하나의 예측된 블록을 포함할 수 있다.

[0074] DMVR 후보 블록 쌍의 왜곡 비용을 측정하고 비교할 수 있다. 가장 낮은 왜곡 비용을 갖는 DMVR 후보 블록 쌍은 참조 픽처 사이에서 가장 유사한 두 개의 블록을 포함하기 때문에, 이 블록 쌍(즉, 블록(1323a, 1323b))은 현재 블록(1311)의 최상의 예측자로 가정될 수 있다. 따라서, 블록 쌍(1323a, 1323b)은 최종 이중 예측된 신호(final bi-predicted signal)를 생성하는 데 사용될 수 있다. 대응하는 MV(1333a, 1333b)는 리파인된 MV로 표시될 수 있다. 모든 DMVR 후보 블록 쌍이 초기 블록 쌍(1321a, 1321b)보다 더 큰 왜곡 비용을 가지면, 초기 블록(1321a, 1321b)은 이중 예측에 사용될 수 있고, 리파인된 MV(1333a, 1333b)는 초기 MV(1331a, 1331b)와 함께 설정될 수 있다.

[0075] 왜곡 비용 계산을 단순화하기 위해, SAD(sum of absolute difference)는 왜곡 메트릭(metric)으로 사용될 수 있으며 DMVR 검색 프로세스에서 루마 왜곡만 고려할 수 있다. SAD는 계산 복잡도를 더욱 줄이기 위해 후보 블록 쌍의 짝수 행 사이에서 평가될 수 있다.

[0076] 도 13의 예에서, 각 참조 픽처의 점선 블록(1321a, 1321b)은 초기 블록 쌍을 지시한다. 회색 블록(1323a, 1323b)은 최상의 매칭되는 블록 쌍을 지시하며, 이는 다른 DMVR 후보 블록 쌍 및 초기 블록 쌍(1321a, 1321b)에 비해 SAD 비용이 가장 낮은 블록 쌍일 수 있다. 초기 MV(1331a, 1331b)는 리파인된 MV(1333a, 1333b)를 생성하기 위해 리파인될 수 있고, 최종 이중 예측된 신호는 최상의 매칭되는 블록 쌍(1323a, 1323b)을 사용하여 생성될 수 있다. 초기 MV(1331a, 1331b)는 전체 샘플 위치를 가리킬 필요가 없다는 점에 유의하며, 이는 병합 모드로부터 도출될 수 있고 그에 따라 최대 1/16 분수 샘플 MV 정밀도가 지원될 수 있기 때문이다.

[0077] 리파인된 MV와 대응하는 초기 MV 사이의 차이(도 13에서 $\Delta MV(1335a, 1335b)$ 로 라벨링됨)는 정수 또는 분수일 수 있기 때문에, 리파인된 MV는 분수 픽셀 위치(fractional pixel position)를 가리킬 수 있다. 이 경우, 중간 검색된 블록과 최종 예측된 블록은 DMVR 보간 프로세스에 의해 생성될 수 있다.

[0078] 일부 실시예에서, 블록 레벨 양자 매칭 기반 DMVR은 TMVP 생성된 모션 필드의 위에 수행될 수 있다. 이러한 프로세스의 예가 이제 본 명세서에서 이전에 설명된 개념을 참조하여 설명된다.

[0079] 프로세스는 각 8×8 블록에 대한 TIP의 일부로 생성되는 모션 필드로 시작될 수 있다. 모션 필드는 픽처와 같은 2차원 공간에 투영된 3차원 모션의 표현(representation)이며, 일반적으로 각각 대응하는 포인트의 움직임(movement)을 설명하는 하나 이상의 모션 벡터로 정의된다. 여기서, 모션 필드는 두 개의 참조 픽처를 가리키는 두 개의 모션 벡터(MV0, MV1)를 포함할 수 있다. 모션 벡터(MV0, MV1)는 DMVR 프로세스의 시작 포인트로 사용될 수 있다. 보다 구체적으로, 모션 벡터가 가리키는 참조 픽처 내의 대응하는 예측자가 생성될 수 있다. 이 작동에서는 보간, 쌍선형(bilinear) 등과 같은 필터를 사용하여 입력을 필터링할 수 있다. 이후, 모션 벡터 주변의 후보 예측자가 생성될 수 있다. 이러한 예측자는, 루마 샘플의 수에 대응하는 정수값인, 미리 정의된 검색 범위 N을 통해 검색될 수 있다. 검색 정밀도(search precision)는 K로 정의되며, 이는 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8에서 루마 샘플의 수(지원되는 가장 높은 MV 정밀도까지)까지의 분수일 수 있다. 다음 작동에서는 모든 후보 예측자 간의 양자 매칭이 수행될 수 있으며, 왜곡 비용이 가장 낮은 예측자들의 위치가 이 8×8 블록에 대한 리파인된 위치로 결정될 수 있다. 왜곡 비용은 SAD, SATD, SSE, 서브샘플링된 SAD, 평균 제거된 SAD 등일 수 있지만 이에 제한되지는 않는다.

[0080] 각 8×8 블록에 대해 리파인된 위치(리파인된 모션 벡터)를 획득한 후, TIP 프로세스를 수행할 수 있다. 보다 구체적으로, TIP 프레임은 DMVR 리파인된 모션 벡터 필드를 이용하여 생성될 수 있다. 생성된 프레임은 예측을 위한 참조로 사용될 수도 있고, 예측에 직접 사용될 수도 있다.

[0081] 디코더 측에서, 블록이 TIP로 또는 TIP 모드에 의해 코딩될 때, TIP 예측자와 DMVR 리파인먼트는 전체 프레임을 생성하지 않고 즉시 수행될 수 있다. 일부 실시예에서, 서브 블록 레벨 양자 매칭 기반 DMVR은 TMVP 생성 모션

필드의 위에 수행될 수 있다. 보다 구체적으로, 8×8 TIP 블록 세트 각각에 대해, 추가적인 분할이 수행될 수 있다. 이러한 추가적 분할로 인해 각 8×8 블록으로부터 4개의 4×4 서브 블록이 생성될 수 있다. 각 서브 블록은 TIP에 대한 리파인된 모션 필드를 획득하기 위해 양자 매칭 기반 DMVR 검색을 수행할 수 있다. 추가적으로, 서브 블록 레벨 양자 매칭 기반 DMVR이 TMVP 생성된 모션 필드 및 광학 플로 리파인먼트의 위에 수행될 수 있다. 보다 구체적으로, 각각의 8×8 TIP 블록에 대해, 추가적인 분할이 수행될 수 있다. 예를 들어, 각 8×8 TIP 블록은 4개의 4×4 서브 블록으로 분할될 수 있으며, 여기서 광학 플로 리파인먼트를 먼저 적용하여 모션 벡터를 리파인한 다음 양자 매칭 기반 DMVR 검색을 추가로 적용하여 TIP에 대한 모션 필드를 리파인한다.

[0082] 일부 실시예에서, 서브 블록 레벨 양자 매칭 기반 DMVR은 TMVP 생성된 모션 필드 및 광학 플로 리파인먼트의 위에 수행될 수 있다. 예를 들어, 각 8×8 TIP 블록에 대해, 추가적인 분할 작동을 통해 4개의 4×4 서브 블록을 생성할 수 있으며, 양자 매칭 기반 DMVR 리파인먼트를 적용하여 모션 벡터를 리파인한 다음 광학 플로 리파인먼트를 추가로 적용하여 TIP에 대한 모션 필드를 리파인한다. 일부 실시예에서, TMVP 생성된 TIP 모션 필드를 리파인하기 위해 다중 스테이지(multi-stage) DMVR이 사용될 수 있다. 예를 들어, 생성되는 초기 모션 필드를 리파인하기 위해 제1 블록 레벨 DMVR이 사용될 수 있다. 리파인된 MV는 제2 스테이지의 시작 포인트로 사용될 수 있다. 제2 스테이지에서는 모션 필드를 더욱 리파인하기 위해 서브 블록 레벨 DMVR이 수행될 수 있다. 추가적인 이러한 스테이지는 본 개시의 범위 내에 있다.

[0083] 다른 실시예에서, TIP 모션 필드는 명시적으로 시그널링된 MV 차이 및/또는 보정(correction)을 사용할 수 있다. 임의의 레벨, 예를 들어 코딩 블록 그룹, 코딩 블록 또는 서브 블록 레벨에서 시작하여, 하나 이상의 모션 벡터 차이(motion vector difference, MVD)가 비트스트림으로 시그널링될 수 있다. 비트스트림은 디코더에 의해 파싱되어 TIP 모션 필드의 보정으로 사용될 수 있다. 블록이 TIP 모드로 인코딩되면, 블록에 대한 대응하는 모션 필드는 TMVP 기반의 방법을 사용하여 생성될 수 있다. 다음으로, 파싱된 MVD를 모션 필드에 추가함으로써, 블록이 8×8이거나 더 작으면, 파싱된 MVD에 의해 블록의 MV를 보정할 수 있다. 블록이 8×8보다 크면, 각 8×8 서브 블록의 각 MV가 파싱된 MVD에 추가될 수 있다.

[0084] 일부 실시예에서, 모션 보상을 위해 2개의 참조 픽처를 사용하여 TIP가 적용될 때, 선택된 참조 픽처와 연관된 모션 필드를 보정하기 위해 MVD가 시그널링될 수 있다. 예를 들어, MVD는 미래 참조 픽처 리스트에 대해서는 시그널링되지만 과거 참조 픽처 리스트에 대해서는 시그널링되지 않을 수 있으며, 그 반대일 수도 있다. 참조 픽처가 시그널링되는 추가적인 MVD를 요구하는 선택은 추가로 시그널링되거나 암시적으로 도출될 수 있다.

[0085] 도 14는 예시적인 실시예에 따른, 모션 벡터 차이(motion vector difference, MVD)를 이용한 병합 모드의 예시적인 사용 사례를 도시한다. 병합 모드는 일반적으로 현재 코딩 유닛(CU)에 의해 생성된 샘플을 예측하기 위해 암시적으로 도출된 모션 정보와 함께 사용될 수 있다. 모션 벡터 차이를 이용한 병합 모드는 플래그를 사용하여 MMVD가 CU에 사용된다고 시그널링할 수 있다. MMVD 플래그는 스킵(skip) 플래그가 송신된 후에 송신될 수 있다. MMVD에서는 병합 후보가 선택된 후, 시그널링된 MVD 정보에 의해 추가로 리파인될 수 있다. 추가적인 정보는 병합 후보 플래그, 모션 매그니튜드(magnitude)를 명시하는(specify) 인덱스, 모션 방향을 지시하는 인덱스를 포함할 수 있다. MMVD 모드에서는 병합 리스트의 처음 두 후보 중 하나를 선택하여 MV 기준(MV basis)으로 사용할 수 있다. 병합 후보 플래그는 어떤 후보가 사용될 것인지를 시그널링할 수 있다.

[0086] 이 작동은 모션 매그니튜드 정보를 명시하고 시작 포인트로부터의 미리 정의된 오프셋을 지시하는 거리 인덱스를 사용할 수 있다. 시작 MV의 수평 컴포넌트 또는 수직 컴포넌트에 오프셋을 추가할 수 있다. 거리 인덱스와 미리 정의된 오프셋의 관계 예시가 표 2에 명시되어 있다.

표 2

[0087] 거리 인덱스와 미리 정의된 오프셋의 관계

거리 IDX	0	1	2	3	4	5	6	7
오프셋(루마 샘플의 단위 위로)	1/4	1/2	1	2	4	8	16	32

[0088] 방향 인덱스는 시작 포인트를 기준으로 MVD의 방향을 나타낼 수 있다. 방향 인덱스는 표 3에 도시된 바와 같이 네 가지 방향 중 하나를 나타낼 수 있다.

표 3

[0089] 방향 인덱스에 의해 명시된 MV 오프셋의 부호

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

[0090] MVD 부호(sign)의 의미는 시작 MV의 정보에 따라 달라질 수 있다. 시작 MV가 단일 예측(uni-prediction) MV이거나 두 참조 픽처 리스트가 모두 현재 픽처의 동일한 축을 가리키는 이중 예측 MV일 때(즉, 두 참조의 POC(picture order count)가 모두 현재 픽처의 POC보다 크거나 둘다 현재 픽처의 POC보다 작음), 표 3의 부호는 시작 MV에 추가된 MV 오프셋의 부호를 명시할 수 있다. 시작 MV가 현재 픽처의 서로 다른 축을 가리키는 두 MV를 갖는 이중 예측 MV이고(즉, 하나의 참조의 POC는 현재 픽처의 POC보다 크고 다른 참조의 POC는 현재 픽처의 POC보다 작음), 제1 참조 픽처 리스트의 POC의 차이가 제2 참조 픽처 리스트의 POC의 차이보다 클 때, 표 3의 부호는 시작 MV에 대한 제1 리스트 MV 컴포넌트에 추가된 MV 오프셋의 부호를 명시할 수 있으며, 제2 리스트 MV의 부호는 반대 값을 가질 수 있다. 그렇지 않고, 제2 리스트의 POC의 차이가 제1 리스트의 POC의 차이보다 크면, 표 3의 부호는 시작 MV에 대한 제2 리스트 MV 컴포넌트에 추가된 MV 오프셋의 부호를 명시할 수 있으며, 제1 리스트 MV에 대한 부호는 반대 값을 가질 수 있다.

[0091] MVD는 각 방향에서 POC의 차이에 따라 스케일링될 수 있다. 두 리스트의 POC의 차이가 동일하면, 스케일링이 생략될 수 있다. 그렇지 않고, 한 리스트의 POC의 차이가 다른 리스트의 POC의 차이보다 크면, POC 차이가 더 작은 리스트에 대한 MVD가 스케일링될 수 있다. 시작 MV가 단일 예측되면, MVD는 이용 가능한 MV에 추가될 수 있다.

[0092] 단방향(unidirectional) 예측 및 양방향 예측 모드 MVD 시그널링 외에도, 양방향 MVD 시그널링을 위한 대칭 MVD 모드도 적용될 수 있다. 대칭 MVD 모드에서는 두 참조 픽처 리스트의 참조 픽처 인덱스와 미래 참조 픽처 리스트의 MVD를 포함하는 모션 정보를 시그널링하지 않고 도출한다.

[0093] 구체적인 구현에서, 대칭 MVD 모드의 디코딩 프로세스는 다음과 같을 수 있다.

[0094] 슬라이스 레벨에서, 변수 BiDirPredFlag, RefIdxSymL0 및 RefIdxSymL1은 다음과 같이 도출될 수 있다: mvd_l1_zero_flag가 1이면, BiDirPredFlag는 0으로 설정된다. 그렇지 않고, 과거 참조 픽처 리스트 L0의 최근 접(nearest) 참조 픽처와 미래 참조 픽처 리스트 L1의 최근접 참조 픽처가 앞쪽 및 뒤쪽 참조 픽처 쌍(forward and backward pair of reference pictures) 또는 뒤쪽 및 앞쪽 참조 픽처 쌍(backward and forward pair of reference pictures)을 형성하면, BiDirPredFlag는 1로 설정되고, L0 참조 픽처 및 L1 참조 픽처는 모두 단기(short-term) 참조 픽처이다. 그렇지 않으면 BiDirPredFlag가 0으로 설정된다.

[0095] CU 레벨에서, 대칭 모드가 사용되는지의 여부를 지시하는 대칭 모드 플래그는 CU가 이중 예측 코딩되고 BiDirPredFlag가 1이면, 명시적으로 시그널링될 수 있다. 대칭 모드 플래그가 참(true)일 때, mvp_l0_flag, mvp_l1_flag 및 MVD0는 명시적으로 시그널링될 수 있으며, 다른 시그널은 생략할 수 있다. L0 및 L1에 대한 참조 인덱스는 각각 참조 픽처 쌍과 같게 설정되고, MVD1은 (-MVD0)과 같게 설정될 수 있다.

[0096] 일부 실시예에서, 인터 프레임에서 각 코딩된 블록에 대해, 현재 블록의 모드가 스킵 모드가 아니라 인터 코딩된 모드이면, 단일 참조 모드 또는 복합 참조 모드가 현재 블록에 사용되는지를 지시하기 위해 또 다른 플래그가 시그널링될 수 있다. 단일 참조 모드에서는 하나의 모션 벡터에 의해 예측 블록이 생성될 수 있고, 복합 참조 모드에서는 2개의 모션 벡터로부터 도출된 2개의 예측 블록을 가중 평균하여 예측 블록이 생성될 수 있다.

[0097] 단일 참조 모드의 경우, 예시적인 구현의 신택스에 따라 다음과 같은 특정 모드가 시그널링될 수 있다:

[0098] NEARMV - DRL(Dynamic Reference List) 인덱스에 지시되는 리스트 내 모션 벡터 예측자(motion vector predictor, MVP) 중 하나를 사용한다.

[0099] NEWMV - DRL 인덱스에 의해 시그널링된 리스트 내 MVP(motion vector predictor) 중 하나를 참조로 사용하고, 델타(delta)를 MVP에 적용한다.

[0100] GLOBALMV - 프레임 레벨 전역 모션 파라미터에 기반하여 모션 벡터를 사용한다.

[0101] 복합 참조 모드의 경우, 예시적인 구현의 신택스에 따라 다음과 같은 특정 모드가 시그널링될 수 있다:

[0102] NEAR_NEARMV - DRL 인덱스에 의해 시그널링된 리스트 내 MVP(motion vector predictor) 중 하나를 사용한다.

- [0103] NEAR_NEWMV - DRL 인덱스에 의해 시그널링된 리스트 내 MVP(motion vector predictor) 중 하나를 참조로 사용하고, 제2 MV에 대한 델타 MV를 송신한다.
- [0104] NEW_NEARMV - DRL 인덱스에 의해 시그널링된 리스트 내 MVP(motion vector predictor) 중 하나를 참조로 사용하고 제1 MV에 대한 델타 MV를 송신한다.
- [0105] NEW_NEWMV - DRL 인덱스에 의해 시그널링된 리스트 내 MVP(motion vector predictor) 중 하나를 참조로 사용하고, 두 MV에 대한 델타 MV를 송신한다.
- [0106] GLOBAL_GLOBALMV - 프레임 레벨 전역 모션 파라미터에 기반하여 각 참조로부터의 MV를 사용한다.
- [0107] 일부 실시예에서, 작동은 1/8 픽셀 모션 벡터 정밀도(또는 정확도)를 허용할 수 있으며, 예시적인 구현에서 L0 또는 L1의 모션 벡터 차이를 시그널링하기 위해 다음 선택스가 사용될 수 있다:
- [0108] mv_joint는 모션 벡터 차이의 어떤 컴포넌트가 0이 아닌지를 명시한다.
- [0109] 0은 수평 또는 수직 방향을 따라 0이 아닌(non-zero) MVD가 없음을 지시한다.
- [0110] 1은 수평 방향을 따라서만 0이 아닌 MVD가 있음을 지시한다.
- [0111] 2는 수직 방향을 따라서만 0이 아닌 MVD가 있음을 지시한다.
- [0112] 3은 수평 방향 및 수직 방향 모두를 따라 0이 아닌 MVD가 있음을 지시한다.
- [0113] mv_sign은 모션 벡터 차이가 양수 또는 음수인지를 명시한다.
- [0114] mv_class는 모션 벡터 차이의 클래스를 명시한다. 표 4에서 볼 수 있는 바와 같이, 클래스가 높을수록 모션 벡터 차이의 매그니튜드가 더 크다는 것을 지시할 수 있다.

표 4

- [0115] 모션 벡터 차이에 대한 매그니튜드 클래스

MV 클래스	MVD의 매그니튜드
MV_CLASS_0	(0, 2]
MV_CLASS_1	(2, 4]
MV_CLASS_2	(4, 8]
MV_CLASS_3	(8, 16]
MV_CLASS_4	(16, 32]
MV_CLASS_5	(32, 64]
MV_CLASS_6	(64, 128]
MV_CLASS_7	(128, 256]
MV_CLASS_8	(256, 512]
MV_CLASS_9	(512, 1024]
MV_CLASS_10	(1024, 2048]

- [0116] mv_bit는 모션 벡터 차이와 각 MV 클래스의 시작 매그니튜드 사이의 오프셋의 정수 부분을 명시한다.
- [0117] mv_fr은 모션 벡터 차이의 처음 2개의 분수 비트(fractional bit)를 명시한다.
- [0118] mv_hp는 모션 벡터 차이의 세 번째 분수 비트를 명시한다.
- [0119] NEW_NEARMV 모드 및 NEAR_NEWMV 모드의 경우, MVD의 정밀도는 연관된 클래스 및 MVD의 매그니튜드에 따라 달라질 수 있다. 예를 들어, 분수 MVD는 MVD 매그니튜드가 1픽셀보다 작거나 같은 경우에만 허용될 수 있다. 또한, 연관된 MV 클래스의 값이 MV_CLASS_1보다 크거나 같을 때, 하나의 MVD 값만 허용될 수 있으며, 각 MV 클래스의 MVD 값은 MV 클래스 1(MV_CLASS_1), 2(MV_CLASS_2), 3(MV_CLASS_3), 4(MV_CLASS_4) 또는 5(MV_CLASS_5)에 대해 4, 8, 16, 32, 64로 도출된다
- [0120] 위의 실시예에 따른 각 MV 클래스에서 허용되는 MVD 값이 표 5에 예시되어 있다.

표 5

[0121] 각 MV 매그니튜드 클래스의 적응적 MVD

MV 클래스	MVD의 매그니튜드
MV_CLASS_0	(0, 1], {2}
MV_CLASS_1	{4}
MV_CLASS_2	{8}
MV_CLASS_3	{16}
MV_CLASS_4	{32}
MV_CLASS_5	{64}
MV_CLASS_6	{128}
MV_CLASS_7	{256}
MV_CLASS_8	{512}
MV_CLASS_9	{1024}
MV_CLASS_10	{2048}

[0122] 또한, 현재 블록이 NEW_NEARMV 모드 또는 NEAR_NEWMV 모드로 코딩되면, mv_joint 또는 mv_class를 시그널링하기 위해 하나의 콘텍스트가 사용될 수 있다. 그렇지 않으면, mv_joint 또는 mv_class를 시그널링하기 위해 다른 콘텍스트가 사용될 수 있다.

[0123] 두 개의 참조 리스트에 대한 MVD가 공동으로 시그널링되는지의 여부를 지시하기 위해 JOINT_NEWMV로 명명된 신규 인터 코딩 모드가 적용될 수 있다. 인터 예측 모드가 JOINT_NEWMV 모드와 같으면, L0 및 L1에 대한 MVD가 공동으로 시그널링될 수 있다. 보다 구체적으로, Joint_mvd로 명명된 하나의 MVD만 시그널링되어 디코더로 전송될 수 있으며, L0 및 L1에 대한 델타 MV는 Joint_mvd로부터 도출될 수 있다.

[0124] JOINT_NEWMV 모드는 NEAR_NEARMV 모드, NEAR_NEWMV 모드, NEW_NEARMV 모드, NEW_NEWMV 모드, 및 GLOBAL_GLOBALMV 모드와 함께 시그널링될 수 있다. 추가적인 콘텍스트를 추가할 필요가 없다. JOINT_NEWMV 모드가 시그널링되고 두 참조 프레임과 현재 프레임 간의 POC 거리가 상이할 때, MVD는 POC 거리에 기반하여 L0 또는 L1에 대해 스케일링될 수 있다. td0을 L0과 현재 프레임 사이의 POC 거리라고 하고, td1을 L1과 현재 프레임 사이의 POC 거리라고 하자. td0이 td1보다 크거나 같으면, joint_mvd가 L0에 대해 직접 사용될 수 있으며, L1에 대한 mvd는 수식(1)에 기반하여 Joint_mvd로부터 도출될 수 있다:

[0125]
$$derived_mvd = \frac{td1}{td0} * joint_mvd \tag{1}$$

[0126] 그렇지 않고, td1이 td0보다 크거나 같으면, joint_mvd가 L1에 대해 직접 사용될 수 있고, L0에 대한 mvd는 수식(2)에 기반하여 joint_mvd로부터 도출될 수 있다:

[0127]
$$derived_mvd = \frac{td0}{td1} * joint_mvd \tag{2}$$

[0128] (위 수식 중 하나에 따라 td0과 td1이 같으면, derived_mvd = joint_mvd: 그래서 joint_mvd는 L0 및 L1 모두에 대해 도출된 MVD로 직접 사용될 수 있으며, 이 경우 스케일링이 발생하지 않음이 명백해질 것이다.)

[0129] 여기서 AMVDMV 모드로 명명된 인터 코딩 모드를 단일 참조 사례에 이용할 수 있다. AMVDMV 모드에서는 적응적 MVD(adaptive MVD, AMVD) 해상도(resolution)가 신호 MVD에 적용된다.

[0130] AMVD가 공동(joint) MVD 코딩 모드에 적용되는지의 여부를 지시하기 위해 플래그(여기서는 amvd_flag로 라벨링됨)가 JOINT_NEWMV 모드 하에서 추가될 수 있다. 이는 공동 AMVD 코딩이라고 불릴 수 있다. 공동 AMVD 코딩에서는 두 참조 프레임에 대한 MVD가 공동으로 시그널링될 수 있으며 MVD의 정밀도는 MVD 매그니튜드에 의해 암시적으로 결정될 수 있다. 그렇지 않으면, 2개(또는 2개 이상)의 참조 프레임에 대한 MVD를 공동으로 시그널링하고 MVD 코딩을 적용할 수 있다.

[0131] CWG-C012에서 처음 제안된 적응적 모션 벡터 해상도(adaptive motion vector resolution, AMVR)는 7개의 MV 정밀도 값(8, 4, 2, 1, 1/2, 1/4, 1/8)을 지원한다. 각 예측 블록에 대해, AVM(adaptive motion vector) 인코더는 지원되는 모든 정밀도 값을 검색하고 최상의 정밀도를 디코더에 시그널링할 수 있다.

[0132] 인코더 런타임(run-time)을 감소시키기 위해, 2개의 정밀도 세트가 지원될 수 있다. 각 정밀도 세트에는 미리 정의된 정밀도 4개가 포함될 수 있다. 정밀도 세트는 프레임의 최대 정밀도의 값에 기반하여 프레임 레벨에서

적응적으로 선택될 수 있다. 표준 AV1과 마찬가지로 최대 정밀도는 프레임 헤더에서 시그널링될 수 있다. 다음 표에는 지원되는 정밀도 값이 프레임 레벨 최대 정밀도에 따라 요약되어 있다.

표 6

두 세트에서 지원되는 MV 정밀도

프레임 레벨 최대 정밀도	지원되는 MV 정밀도
1/8	1/8, 1/2, 1, 4
1/4	1/4, 1, 4, 8

[0133]

AV1과 관련된 AOMedia AVM 저장소(repository)는 프레임의 MV가 서브 픽셀 정밀도를 포함하는지의 여부를 지시하는 프레임 레벨 플래그를 제공한다. 특정 실시예에서, AMVR은 cur_frame_force_integer_mv 플래그의 값이 0인 경우에만 이네이블될 수 있다. 블록의 정밀도가 최대 정밀도보다 낮으면, 모션 모델과 보간 필터는 시그널링되지 않고 활성화되지 않은(unactuated) 상태로 유지될 수 있다. 블록의 정밀도가 최대 정밀도보다 낮으면, 적용 가능한 모션 모델은 트랜스레이션(translation) 모션 모델로 추론될 수 있으며, 적용 가능한 보간 필터는 '정규(regular)' 필터로 추론될 수 있다. 블록의 정밀도가 4-펠(pe1) 또는 8-펠이면, 인터-인트라 모드는 시그널링되지 않은 상태로 유지될 수 있으며 0으로 추론될 수 있다.

[0134]

[0135]

도 15는 예시적인 실시예에 따른 지오메트리 파티셔닝 모드(geometric partitioning mode, GPM) 예측의 예시적인 작동을 도시한다. 이 작동은 인터 픽처 예측된 코딩 유닛(coding unit, CU)에 중점을 둔다. 현재 CU(1510)에 GPM이 적용될 때, 현재 CU(1510)는 파티셔닝 경계에 의해 두 부분(part)(1510a, 1510b)으로 분할될 수 있다. 파티셔닝 경계의 위치(location)는 각도 파라미터 ϕ 와 오프셋 파라미터 ρ 에 의해 수학적으로 정의될 수 있다. 이러한 파라미터는 양자화되어 GPM 파티셔닝 인덱스 조회 테이블로 조합될 수 있다. 현재 CU(1510)의 GPM 파티셔닝 인덱스는 비트스트림으로 코딩될 수 있다. 전체적으로, $k, l \in \{3...6\}$ 인 $w \times h = 2k \times 2l$ (루마 샘플의 관점에서) 크기를 갖는 CU(1510)에 대해 총 64개의 파티셔닝 모드가 사용될 수 있다. 좁은 CU에는 일반적으로 기하학적으로 분리된 패턴이 포함되어 있지 않기 때문에, 종횡비(aspect ratio)가 4:1보다 크거나 1:4보다 작은 CU(1510)에서는 GPM의 적용이 디스에이블될 수 있다.

[0136]

2개의 GPM 파티션에는 현재 CU(1510)에서 대응하는 부분을 예측하는 데 사용될 수 있는 개별 모션 정보가 포함되어 있다. 각 CU 부분(1510a, 1510b)에 대해 단방향 모션 보상된 예측(motion-compensated prediction, MCP)이 적용될 수 있으므로, GPM에서 MCP의 필요한 메모리 대역폭은 일반 양방향 MCP의 것과 같다. 모션 정보 코딩을 단순화하고 GPM에 대한 가능한 조합을 감소시키기 위해, 모션 정보는 병합 모드를 사용하여 코딩될 수 있다. GPM 병합 후보 리스트는 단방향 모션 정보만이 포함되는 것을 보장하기 위해 기존의 병합 후보 리스트로부터 도출될 수 있다.

[0137]

도 15의 예에서, 현재 CU(1510)의 오른쪽 부분(1510a)은 제1 참조 픽처(1530a)(도 15에서 P0으로도 라벨링됨)로부터 제1 모션 벡터(1520a)(또한 도 15에서 MV0으로 라벨링됨)에 의해 예측되며, 왼쪽 부분(1510b)은 제2 참조 픽처(1530b)(또한 도 15에서 P1로 라벨링됨)로부터 제2 모션 벡터(1520b)(또한 도 15에서 MV1로 라벨링됨)에 의해 예측된다.

[0138]

CU의 각 부분이 예측되면, 완전한(complete) CU에 대한 예측이 블렌딩 프로세스에 의해 생성될 수 있다.

[0139]

도 16은 예시적인 실시예에 따른 GPM 예측에서의 블렌딩의 예시적인 작동을 도시한다. 블렌딩 마스크는 CU의 개개의 예측된 부분에 적용하기 위해 행렬(1600a, 1600b)과 같은 행렬의 형태를 취할 수 있다. 도 16의 예에서 행렬(1600a, 1600b)은 각각 0에서 8 사이의 값 범위에 있는 가중치를 포함한다. 즉, W0와 W1은 각각 제1 행렬 및 제2 행렬(1600a, 1600b)을 지시하며, J는 $w \times h$ 크기로 구성된 1의 행렬을 나타내고, $W0 + W1 = 8J$ 이다. 블렌딩 행렬의 가중치는 샘플 위치(sample location)와 파티셔닝 경계 사이의 변위에 따라 달라질 수 있다. 블렌딩 행렬 도출의 계산 복잡도는 매우 낮으므로, 이러한 행렬은 디코더 측에서 즉시 생성될 수 있다.

[0140]

행렬을 적용하면, 완전한 CU에 대한 예측이 수식(3)에 기반하여 결정될 수 있다:

[0141]

$$PG = (W0 \cdot P0 + W1 \cdot P1 + 4) \gg 3 \tag{3}$$

[0142]

여기서 W0 및 W1은 각각 제1 행렬 및 제2 행렬(1600a, 1600b)을 지시하고, P0 및 P1은 각각 제1 참조 픽처 및 제2 참조 픽처(1530a, 1530b)를 지시하며, PG는 생성된 예측을 지시한다.

[0143] 그런 다음 원본 신호로부터 생성된 예측을 빼서 잔차를 생성할 수 있다. 잔차는 예를 들어 VVC 변환, 양자화 및 엔트로피 코딩 엔진 또는 다른 적합한 코딩 엔진을 사용하여 변환되고 양자화되어 비트스트림으로 코딩될 수 있다. 디코더 측에서는 잔차를 생성된 예측에 추가하여 신호를 재구성할 수 있다. 잔차가 무시할 수 있으면, "스킵 모드"가 적용될 수 있으며, 여기서 잔차는 인코더에 의해 삭제되고, 생성된 예측은 디코더에 의해 재구성된 신호로 직접 사용된다.

[0144] 도 17은 예시적인 실시예에 따른 특수 복합 예측 모드에서 웨지 기반 예측을 위한 예시적인 코드북을 도시한다. 웨지 기반 예측은 AV1에서 구현될 수 있으며, 인터-인터(Inter-Inter)와 인터-인트라 조합(Inter-Intra Combination) 모두에 사용될 수 있다.

[0145] 복합 웨지 예측에서, 움직이는 객체의 경계를 온-그리드 블록 파티션(on-grid block partition)으로 근사화하는 것이 어려운 경우가 많다. 따라서, 특정 실시예에서, 코딩 유닛이 이러한 방식으로 추가로 파티셔닝되도록 선택할 때, 비트스트림에서 웨지 인덱스를 시그널링하기 위해 16개의 가능한 웨지 파티션의 미리 정의된 코드북이 사용될 수 있다. 수평적이거나, 수직적이거나 또는 기울기가 ±2 또는 ±0.5인 파티션 방향을 포함하는 16진 모양 코드북(16-ary shape codebook)이 설계될 수 있다. 도 17의 예에서, 2개의 코드북(1710, 1720)은 정사각형 및 직사각형 블록을 위해 각각 설계된다.

[0146] 종종 두 개의 예측자를 직접 병치하여(juxtaposing) 생성되는 허위 고주파수 성분을 완화하기 위해, 부드러운 절벽 모양(soft-cliff-shaped)의 2D 웨지 마스크를 사용하여 의도된 파티션 주위의 에지(edge)를 매끄럽게 만들 수 있다. 예를 들어, $m(i, j)$ 는 에지 주위에서 0.5에 가까울 수 있으며, 양쪽 끝에서 점진적으로 이진 가중치로 변환될 수 있다.

[0147] 전술한 블렌딩은 파티셔닝 경계 주위의 블렌딩 간격(blending interval)을 정의하는 임계 θ 를 활용할 수 있다. 이 간격 내에 마스크를 적용하여 블렌딩된 영역을 생성할 수 있다. 마스크가 정의될 수 있으며, 그에 따라 영역(area)이 블렌딩된다. 파티션 경계로부터 $d(x_c, y_c)$ 거리에 있는 각 위치(position)(x_c, y_c)의 가중치에 따라, 램프 함수를 이용하여; 예를 들어, 수식 (4)에 의해:

$$\omega_{x_c, y_c} = \begin{cases} 0 & d(x_c, y_c) \leq -\theta \\ \frac{8}{2\theta} (d(x_c, y_c) + \theta) & -\theta < d(x_c, y_c) < \theta \\ 8 & d(x_c, y_c) \geq \theta, \end{cases} \quad (4)$$

[0148] 고정된 임계 θ 를 사용하는 것은 최적이지 아닐 수 있으며, 이는 고정된 블렌딩 영역 너비가 항상 다양한 유형의 비디오 콘텐츠에 대해 최상의 블렌딩 품질을 제공하는 것은 아니기 때문이다. 예를 들어, 화면 비디오 콘텐츠에는 일반적으로 강한 질감과 날카로운 에지가 포함되어 있으며, 이는 에지 정보를 보존하기 위해 좁은 블렌딩된 영역(즉, 작은 임계)을 참조한다. 카메라로 캡처한 콘텐츠의 경우, 일반적으로 블렌딩이 필요하며; 그러나 블렌딩된 영역 너비는 여러 팩터, 예를 들어, 움직이는 객체의 실제 경계 및 두 파티션의 모션 구별성에 따라 달라진다. 또한 서로 다른 CU 부분마다 임계 요건이 서로 다를 수 있다.

[0150] 도 18은 예시적인 실시예에 따른 비대칭 블렌딩 생성의 예시적인 작동을 도시한다. 여기에 설명된 비대칭 블렌딩 마스크의 실시예는 VVC의 지오메트리 파티셔닝 모드 예측, AV1의 웨지 기반 예측, 또는 기타 유사한 인코딩 포맷 및/또는 기술에 적용될 수 있다.

[0151] 도 18의 예에서는 제1 임계 θ_1 과 제2 임계 θ_2 가 정의되며, 여기서 θ_1 은 파티셔닝 경계 B로부터 한 방향에서의 거리를 반영하는 유효 음수 값을 갖고, θ_2 는 B로부터 다른 방향에서의 거리를 반영하는 유효 양수 값을 갖는다. 그런 다음 특정 위치의 가중치는 예를 들어 수식 (5)에 의해 임계로부터 계산될 수 있다:

$$\omega_{x_c, y_c} = \begin{cases} 0 & d(x_c, y_c) \leq \theta_1 \\ \frac{8}{2\theta_1} (d(x_c, y_c) + \theta_1) & \theta_1 < d(x_c, y_c) \leq 0 \\ \frac{8}{2\theta_2} (d(x_c, y_c) + \theta_2) & 0 < d(x_c, y_c) < \theta_2 \\ 8 & d(x_c, y_c) \geq \theta_2, \end{cases} \quad (5)$$

- [0153] 예제 수식 (4)에 설명된 바와 같이, 위치 (x_c, y_c) 에서 파티셔닝 경계 B까지의 변위 $d(x_c, y_c)$ 가 θ_1 보다 작거나 같으면, 위치는 B를 기준으로 임계 θ_1 외부에 있고, 가중치는 0이 사용된다. $d(x_c, y_c)$ 가 θ_2 보다 크거나 같으면, 위치는 B를 기준으로 임계 θ_2 외부에 있고 전체 가중치(예: 이 예에서는 8)가 사용된다. $d(x_c, y_c)$ 가 θ_1 과 θ_2 사이에 있으면, 0과 8 사이의 램프 가중치 값이 사용된다.
- [0154] 다른 적합한 가중치 값과 램프 공식을 갖는 다른 수식이 경험적으로, 정성적으로 또는 임의로 결정될 수 있다. θ_1 과 θ_2 의 절대값이 같을 때, 블렌딩은 대칭 적응적 블렌딩과 유사하게 작동하는 반면, θ_1 과 θ_2 가 같지 않을 때, 비대칭 적응적 블렌딩이 발생한다.
- [0155] 특정 실시예에서, 블렌딩 마스크는 2개의 임계를 사용하는 웨지 기반 예측 설계에 기반하여 계산될 수 있다. 이러한 실시예에서, 파티셔닝 경계 B 근처의 마스크 가중치는 절반 값(예: 32)과 동등하며(equivalent), 어느 한 쪽 극단에서 점차적으로 이진 가중치(예: 0 및 64)로 변환된다. 이러한 실시예에서, 그레이디언트(gradient)는 미리 정의된 임계에 기반할 수 있으며, 이는 예를 들어 임계가 클수록 마스크에 대한 변환이 덜 선명해지도록 마스크를 변경한다.
- [0156] 서로 다른 블렌딩 임계에 대응하기 위한 부분 선택(part selection)이 명시적으로 시그널링될 수 있다. 예를 들어, 이진 부분 선택 플래그는 두 가지 가능한 할당; 제1 CU 부분에 대응하는 파티션 경계의 제1 측에 임계 θ_1 (예를 들어, 더 작은 임계일 수 있고 결과적으로 더 날카로운 블렌딩이 발생함)이 할당되고, 제2 CU 부분에 대응하는 파티션 경계의 제2 측에는 임계 θ_2 (예를 들어, 더 큰 임계일 수 있으며 결과적으로 더 무디거나(blunter) 더 부드러운 블렌딩이 발생함)가 할당되는 제1 할당 및 제1 할당의 반대인 제2 할당 중 하나를 시그널링할 수 있다.
- [0157] 서로 다른 블렌딩 임계에 대응하기 위한 부분 선택은 미리 정의된 방법에 의해 암시적으로 도출될 수도 있다. 다양한 각도, 오프셋, 웨지 인덱스 또는 기타 파라미터를 선택하는 것에 의해 선택이 이루어질 수 있다. 이러한 기타 파라미터에는 매그니튜드, 각 부분의 대응하는 모션 벡터의 방향, 각 부분의 예측 모드 유형 또는 이웃 채 구성된 샘플에 기반한 것이 포함될 수 있다.
- [0158] 도 19는 예시적인 실시예에 따른 비대칭/대칭 블렌딩 마스크에 대한 적응적 임계 선택의 예시적인 작동을 도시한다. 이 실시예에서, 2개의 임계 θ_1 및 θ_2 는 블렌딩 마스크(1900)를 생성하는 데 사용된다. 임계는 대응하는 인덱스, 시그널링된 값일 수 있으며 또는 암시적으로 도출될 수 있다. 이러한 임계는 동일하거나 일치할 수도 있고 상이한 값일 수도 있다. 임계 사이의 간격(1910)은 θ_1 및 θ_2 가 동일하거나 유사할 수 있도록, 또는 어느 하나가 임의의 원하는 정도로 다른 것보다 더 클 수 있도록, 적응적으로 시프트될 수 있다.
- [0159] 임계는 별도로 시그널링될 수 있으며 각각 비트스트림 및 컨텍스트 모델에서 고유한 신택스 엘리먼트를 가질 수 있다. θ_1 및 $(\theta_2 - \theta_1)$ 이 시그널링되거나 θ_2 및 $(\theta_1 - \theta_2)$ 가 시그널링되고, 나머지 임계가 도출될 수 있도록 임계는 다르게는 차등적으로 시그널링될 수 있다. 임계는 $\theta_1 : \theta_2 = 1 : 2$ 와 같이 미리 정의된 비율을 가질 수도 있으므로, θ_1 (또는 θ_2)만 시그널링되면 된다.
- [0160] 임계는 미리 정의된 리스트에서 선택될 수도 있다. 예를 들어, 가능한 임계 값(threshold value)으로는 {0.5, 1, 2, 4, 8}과 같은 리스트가 사용될 수 있다. 리스트를 이용하여, 대응하는 임계 값의 인덱스를 시그널링할 수 있다. 디코더에서는 미리 정의된 리스트와 파싱된 인덱스에 기반하여, θ_1 및 θ_2 에 대한 값을 얻을 수 있다. 임계 θ_1 및 θ_2 에 대한 값 조합은 {(1,1), (1,2), (2,1), (1, 4), (4, 1), ...}과 같은 미리 정의된 리스트에서 선택하도록 제공될 수도 있으며, 선택된 조합에 대해 미리 정의된 리스트의 인덱스가 시그널링될 수 있다. 일부 경우에는 각 임계에 미리 정의된 자체 리스트가 있다. 일 예에서, θ_1 에 대해 미리 정의된 리스트는 {0.5, 1, 2, 4, 8}일 수 있는 반면, θ_2 에 대해 미리 정의된 리스트는 {0.25, 0.5, 1, 2, 4}일 수 있다. 각 임계에 대한 개별 인덱스가 시그널링될 수 있다.
- [0161] 임계에 대해 하나 이상의 미리 정의된 리스트가 사용될 때, 각각의 미리 정의된 리스트의 서브 세트가 보다 구체적으로 사용될 수 있다. 또한, 미리 정의된 임계는 모든 임계 후보의 각 블록에 사용될 수도 있다. 각각의 미

리 정의된 리스트의 서브 세트는 현재 블록의 인코딩 및 디코딩 모두에 존재할 수 있는 코딩된 정보에 의해 결정될 수 있다. 현재 블록의 코딩된 정보는 이웃하는 재구성된 샘플, 블록 크기, 예측 모드 또는 미리 정의된 임계 리스트의 서브 세트를 생성하기 위한 기타 관련 정보를 포함할 수 있다.

- [0162] 특정 실시예에서, 템플릿 매칭에 의해 최상의 후보가 선택될 수 있다. 템플릿은 각 참조 프레임으로부터의 예측자의 왼쪽 상단 주변 샘플을 사용할 수 있으며 미리 정의된 임계에 기반하여 생성될 수 있다. 생성된 템플릿은 현재 블록의 왼쪽 상단 주변 샘플과 비교될 수 있다. 왜곡 비용이 가장 낮은 후보가 GPM 또는 웨지 기반 예측에 사용될 수 있다.
- [0163] 후보들은 또한 템플릿 매칭에 기반하여 재정렬될 수 있고, 가장 낮은 왜곡에 따른 상위 N개의 후보들이 사용될 수 있다. 최종적으로 선택된 임계는 시그널링된/파싱된 인덱스에 따라 달라질 수 있다. N의 값은 미리 정의되거나 상위 레벨 선택으로 시그널링될 수 있다. N이 1일 때, 인덱스는 시그널링에 사용되지 않음을 유의한다.
- [0164] 특정 실시예에서, 코딩된 정보로부터 도출된 콘텐츠를 사용하여 두 임계의 엔트로피 코딩이 수행될 수 있다. 코딩된 정보는 이웃 블록으로부터 선택된 임계일 수 있다.
- [0165] 위의 개시에 따르면, 미리 정의된 대칭(즉, 하나의 임계) 블렌딩 설계 대신에, 설계는 파티셔닝 경계 주위에서 서로 다른 블렌딩 임계; 예를 들어, 도 18에 도시된 바와 같이 미리 정의된 2개의 임계 θ_1 및 θ_2 를 가질 수 있다. 블렌딩 마스크 또는 웨지 가중치 마스크는 이 두 임계에 기반하여 미리 또는 즉시 계산될 수 있다. 특정 코덱의 임계 정의에 기반하여, 임계 값은 변위를 지시하기 위해 음수 값(VVC의 GPM에서 볼 수 있음) 또는 양수 값(AV1 및 AV2에 대한 웨지 기반 예측에서 볼 수 있음)으로 정의될 수 있다.
- [0166] 제안된 방법은 개별적으로 사용될 수도 있고 임의의 순서로 조합되어 사용될 수도 있다. 또한, 방법(또는 실시예), 인코더 및 디코더 각각은 처리 회로(예: 하나 이상의 프로세서 또는 하나 이상의 집적 회로)에 의해 구현될 수 있다. 일 예에서, 하나 이상의 프로세서는 컴퓨터가 판독 가능한 비일시적 매체에 저장된 프로그램을 실행한다.
- [0167] 도 20은 여기에서 설명된 시스템 및/또는 방법의 실시예가 구현될 수 있는 디바이스 또는 시스템(2000)의 예시적인 컴포넌트의 도면이다. 예시적인 시스템(2000)은 개인용 컴퓨터, 모바일 디바이스, 컴퓨터 클러스터, 서버, 임베디드 디바이스, ASIC, 마이크로컨트롤러, 또는 코드를 실행할 수 있는 임의의 다른 디바이스와 같은 다양한 시스템 중 하나일 수 있다. 버스(2010)는 모든 컴포넌트가 서로 통신할 수 있도록 예시적인 시스템(2000)을 함께 연결한다. 버스(2010)는 프로세서(2020), 메모리(2030), 저장 컴포넌트(2040), 입력 컴포넌트(2050), 출력 컴포넌트(2060) 및 인터페이스 컴포넌트를 연결한다.
- [0168] 프로세서(2020)는 단일 프로세서, 내부에 다중 프로세서를 갖는 프로세서, 프로세서의 클러스터(2개 이상) 및/또는 분산 처리일 수 있다. 프로세서는 메모리(2030)와 저장 컴포넌트(2040) 모두에 저장된 명령어를 실행한다. 프로세서(2020)는 공유된 언리얼 엔진 도출된 데이터 캐시(Unreal Engine Derived Data Cache)를 수정하기 위한 작동을 수행하는 계산 디바이스로 작동한다. 메모리(2030)는 하나 이상의 CPU와 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 임의의 메모리 디바이스에 대한 빠른 저장 및 검색이 가능해질 수 있다. 저장 컴포넌트(2040)는 HDD, SSD, 자기 테이프 또는 임의의 다른 장기 저장 포맷과 같은 임의의 장기 저장소(longer term storage) 중 하나일 수 있다.
- [0169] 입력 컴포넌트(2050)는 카메라, 핸드헬드 컨트롤러, 게임 패드, 키보드, 마우스 또는 모션 캡처 장비와 같은 입력 캡처 장비와 같은 사용자 인터페이스 컴포넌트로부터의 신호 또는 임의의 파일 유형일 수 있다. 출력 컴포넌트(2060)는 처리된 정보를 통신 인터페이스(2070)로 출력한다. 통신 인터페이스는 사용자 또는 다른 컴퓨팅 시스템과 같은 다른 관찰자에게 정보를 디스플레이할 수 있는 화면과 같은 기타 통신 디바이스 또는 스피커일 수 있다.
- [0170] 전술한 개시는 예시 및 설명을 제공하지만, 개시된 정확한 형태로 구현을 제한하거나 총망라하려는 의도는 아니다. 위의 개시에 비추어 수정 및 변형이 가능하거나 구현의 실행을 통해 획득될 수 있다.
- [0171] 일부 실시예는 임의의 가능한 기술 세부 통합 레벨에서 시스템, 방법 및/또는 컴퓨터가 판독 가능한 매체와 관련될 수 있다. 또한, 위에 설명된 위의 컴포넌트 중 하나 이상은 컴퓨터가 판독 가능한 매체에 저장되고 적어도 하나의 프로세서에 의해 실행 가능한 명령어로 구현될 수 있다(및/또는 적어도 하나의 프로세서를 포함할 수 있음). 컴퓨터가 판독 가능한 매체는 프로세서가 작동을 수행하게 하기 위한, 컴퓨터가 판독 가능한 프로그램 명령어를 저장하는, 컴퓨터가 판독 가능한 비일시적 저장 매체(또는 미디어)를 포함할 수 있다.

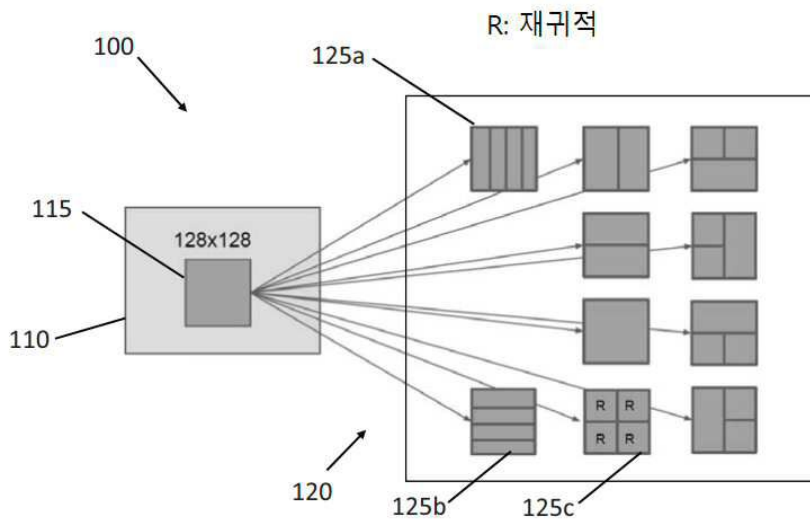
- [0172] 컴퓨터가 판독 가능한 저장 매체는 명령어 실행 디바이스에 의해 사용되는 명령어를 유지하고 저장할 수 있는 유형의 디바이스일 수 있다. 컴퓨터가 판독 가능한 저장 매체는 예를 들어 전자 저장 디바이스, 자기 저장 디바이스, 광학 저장 디바이스, 전자기 저장 디바이스, 반도체 저장 디바이스 또는 이들의 임의의 적절한 조합일 수 있지만 이에 제한되지는 않는다. 컴퓨터가 판독 가능한 저장 매체의 보다 구체적인 예의 비포괄적인 리스트는 다음: 휴대용 컴퓨터 디스켓, 하드 디스크, 랜덤 액세스 메모리(random access memory, RAM), 읽기 전용 메모리(read-only memory, ROM), 소거 가능 프로그래밍 가능한 읽기 메모리 전용 메모리(erasable programmable read-only memory, EPROM 또는 플래시 메모리), 정적 랜덤 액세스 메모리(static random access memory, SRAM), 휴대용 콤팩트 디스크 읽기 전용 메모리(portable compact disc read-only memory, CD-ROM), 디지털 다용도 디스크(digital versatile disk, DVD), 메모리 스틱, 플로피 디스크, 펀치 카드 또는 명령어가 기록된 홈의 용기 구조와 같은 기계적 인코딩된 디바이스, 및 전술한 것의 임의의 적합한 조합을 포함한다. 여기에서 사용된 컴퓨터가 판독 가능한 저장 매체는 전파 또는 기타 자유롭게 전파되는 전자기파, 도파관 또는 기타 전송 매체를 통해 전파되는 전자기파(예: 광섬유 케이블을 통과하는 광 펄스) 또는 와이어(wire)를 통해 전송되는 전기 신호와 같은 일시적인 신호 그 자체로 해석되어서는 안된다.
- [0173] 여기에서 설명된 컴퓨터가 판독 가능한 프로그램 명령어는 컴퓨터가 판독 가능한 저장 매체로부터 개개의 컴퓨팅/처리 디바이스로 다운로드될 수 있거나 네트워크, 예를 들어 인터넷, 근거리 통신망, 광역 네트워크 및/또는 무선 네트워크를 통해 외부 컴퓨터 또는 외부 저장 디바이스로 다운로드될 수 있다. 네트워크는 구리 전송 케이블, 광 전송 섬유, 무선 전송, 라우터, 방화벽, 스위치, 게이트웨이 컴퓨터 및/또는 에지 서버를 포함할 수 있다. 각 컴퓨팅/처리 디바이스의 네트워크 어댑터 카드 또는 네트워크 인터페이스는 네트워크로부터 컴퓨터가 판독 가능한 프로그램 명령어를 수신하고, 개개의 컴퓨팅/처리 디바이스 내 컴퓨터가 판독 가능한 저장 매체에 저장하기 위해 컴퓨터가 판독 가능한 프로그램 명령어를 전달한다.
- [0174] 작동을 수행하기 위한 컴퓨터가 판독 가능한 프로그램 코드/명령어는 어셈블러 명령어, 명령어 세트 아키텍처(instruction-set-architecture, ISA) 명령어, 기계 명령어, 기계 종속 명령어, 마이크로코드, 펌웨어 명령어, 상태 설정 데이터, 집적 회로를 위한 구성 데이터, 또는 Smalltalk, C++ 등과 같은 객체 지향 프로그래밍 언어와 "C" 프로그래밍 언어 또는 유사한 프로그래밍 언어와 같은 절차적 프로그래밍 언어를 포함하여 하나 이상의 프로그래밍 언어의 임의의 조합으로 작성된 소스 코드 또는 객체 코드일 수 있다. 컴퓨터가 판독 가능한 프로그램 명령어는 사용자 컴퓨터에서 전체적으로, 사용자 컴퓨터에서 부분적으로, 독립형 소프트웨어 패키지로써, 사용자 컴퓨터에서 부분적으로 그리고 원격 컴퓨터에서 부분적으로, 또는 원격 컴퓨터나 서버에서 전체적으로 실행될 수 있다. 후자의 시나리오에서, 원격 컴퓨터는 근거리 통신망(local area network, LAN)이나 광역 네트워크(wide area network, WAN)를 포함한 임의의 유형의 네트워크를 통해 사용자의 컴퓨터에 연결될 수 있거나, 이 연결은 외부 컴퓨터(예를 들어, 인터넷 서비스 공급자를 사용하여 인터넷을 통해) 이루어질 수 있다. 일부 실시예에서, 측면이나 작동을 수행하기 위해, 예를 들어 프로그래밍 가능한 논리 회로, 필드 프로그래밍 가능한 게이트 어레이(field-programmable gate array, FPGA) 또는 프로그래밍 가능한 논리 어레이(programmable logic array, PLA)를 포함하는 전자 회로는 컴퓨터가 판독 가능한 프로그램 명령어의 상태 정보를 활용하여 컴퓨터가 판독 가능한 프로그램 명령어를 실행하여 전자 회로를 개인화할 수 있다.
- [0175] 이러한 컴퓨터가 판독 가능한 프로그램 명령어는 범용 컴퓨터, 특수 목적 컴퓨터의 프로세서 또는 기계를 생산하기 위한 기타 프로그래밍 가능한 데이터 처리 장치에 제공될 수 있으므로, 컴퓨터의 프로세서 또는 기타 프로그래밍 가능한 데이터 처리 장치를 통해 실행되는 명령어는 흐름도 및/또는 블록도 블록 또는 블록들에서 명시된 작동을 구현한다. 이들 컴퓨터가 판독 가능한 프로그램 명령어는 또한 컴퓨터, 프로그래밍 가능한 데이터 처리 장치, 및/또는 기타 디바이스에 특정 방식으로 작동하도록 지시할 수 있는 컴퓨터가 판독 가능한 저장 매체에 저장될 수 있으므로, 명령어가 저장된, 컴퓨터가 판독 가능한 저장 매체는 흐름도 및/또는 블록도 블록 또는 블록들에서 명시된 작동의 측면을 구현하는 명령어를 포함하는 제품으로 구성된다.
- [0176] 컴퓨터가 판독 가능한 프로그램 명령어는 컴퓨터, 기타 프로그래밍 가능한 데이터 처리 장치, 또는 다른 디바이스에 로딩되어 일련의 작동 단계가 컴퓨터, 기타 프로그래밍 가능한 장치 또는 다른 디바이스에서 수행되어 컴퓨터 구현 프로세스를 생성하게 할 수 있으므로, 컴퓨터, 기타 프로그래밍 가능한 장치 또는 다른 디바이스에서 실행되는 명령어는 흐름도 및/또는 블록도 블록 또는 블록들에서 명시된 작동을 구현한다.
- [0177] 도면의 흐름도 및 블록도는 다양한 실시예에 따른 시스템, 방법 및 컴퓨터가 판독 가능한 매체의 가능한 구현의 아키텍처, 기능 및 작동을 예시한다. 이와 관련하여, 흐름도 또는 블록도의 각 블록은 명시된 논리 연산(들)을 구현하기 위한 하나 이상의 실행 가능한 명령어를 포함하는 명령어의 부분, 세그먼트 또는 모듈을 나타낼 수 있다. 방법, 컴퓨터 시스템 및 컴퓨터가 판독 가능한 매체는 도면에 도시된 것보다 추가적인 블록, 더 적은 수의

블록, 상이한 블록, 또는 다르게 배열된 블록을 포함할 수 있다. 일부 대안적인 구현에서, 블록에 표시된 작동은 도면에 표시된 순서를 벗어나 발생할 수 있다. 예를 들어, 연속적으로 도시된 2개의 블록은 실제로 동시에 또는 실질적으로 동시에 실행될 수 있거나, 관련된 기능에 따라 블록이 때때로 역순으로 실행될 수 있다. 또한, 블록도 및/또는 흐름도 예시의 각 블록과 블록도 및/또는 흐름도 예시의 블록 조합은 명시된 작동 또는 동작을 수행하거나, 특수 목적의 하드웨어와 컴퓨터 명령어의 조합을 수행하는 특수 목적의 하드웨어 기반 시스템에 의해 구현될 수 있다는 점에 유의할 것이다.

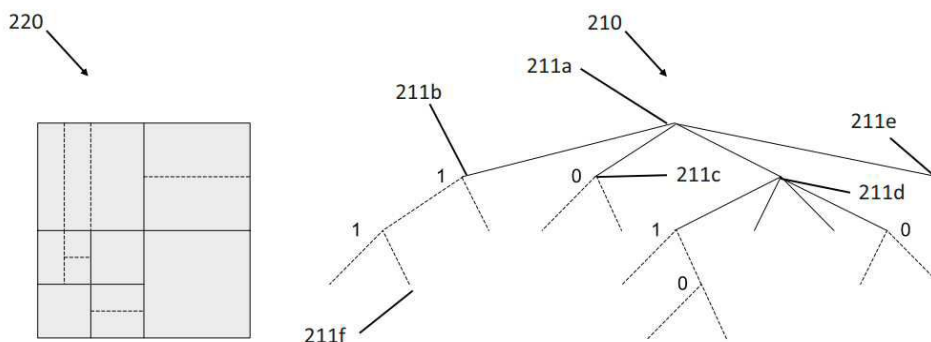
[0178] 여기에 설명된 시스템 및/또는 이 방법은 서로 다른 형태의 하드웨어, 펌웨어, 또는 하드웨어와 소프트웨어의 조합으로 구현될 수 있다는 것이 명백할 것이다. 이러한 시스템 및/또는 방법을 구현하는 데 사용되는 실제 특수 제어 하드웨어 또는 소프트웨어 코드는 구현을 제한하지 않는다. 따라서, 시스템 및/또는 방법의 작동 및 행동은 특정 소프트웨어 코드를 참조하지 않고 여기에서 설명되었으며 - 소프트웨어 및 하드웨어는 여기에서의 설명에 기반하여 시스템 및/또는 방법을 구현하도록 설계될 수 있다는 것이 이해된다.

도면

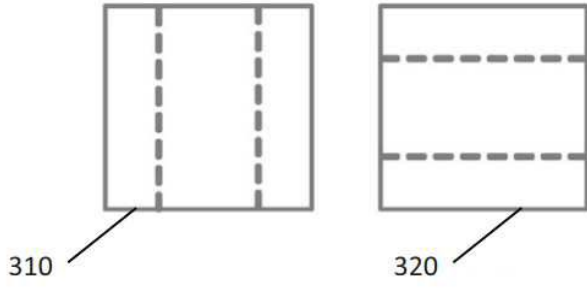
도면1



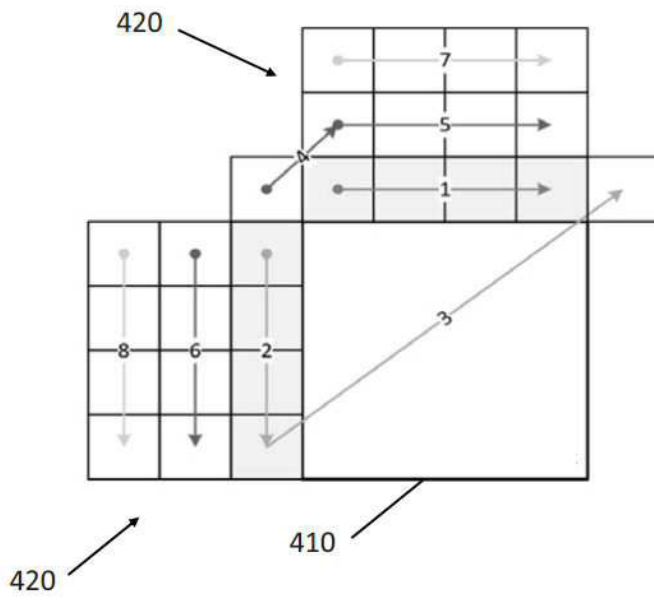
도면2



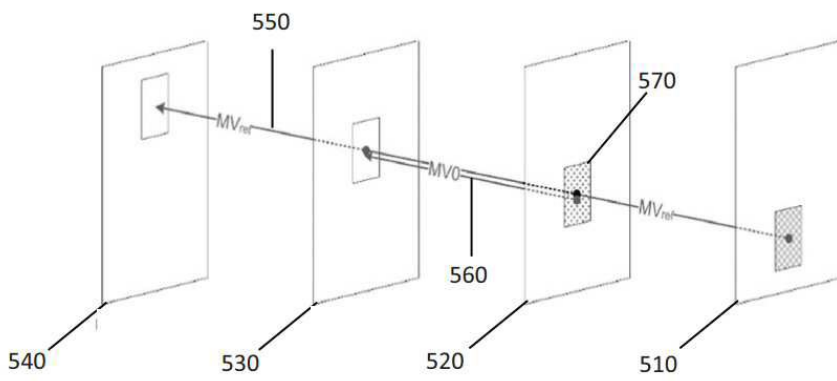
도면3



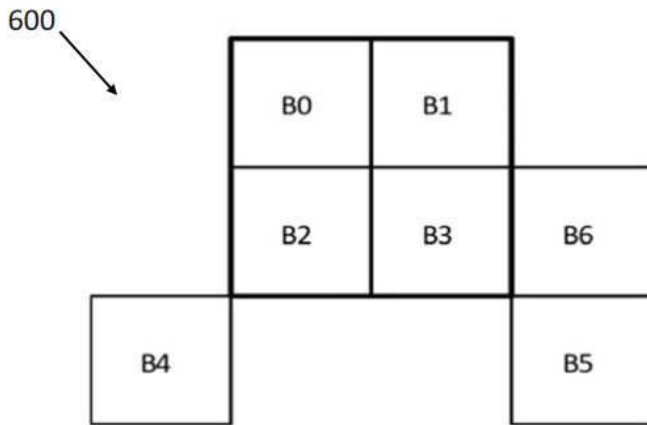
도면4



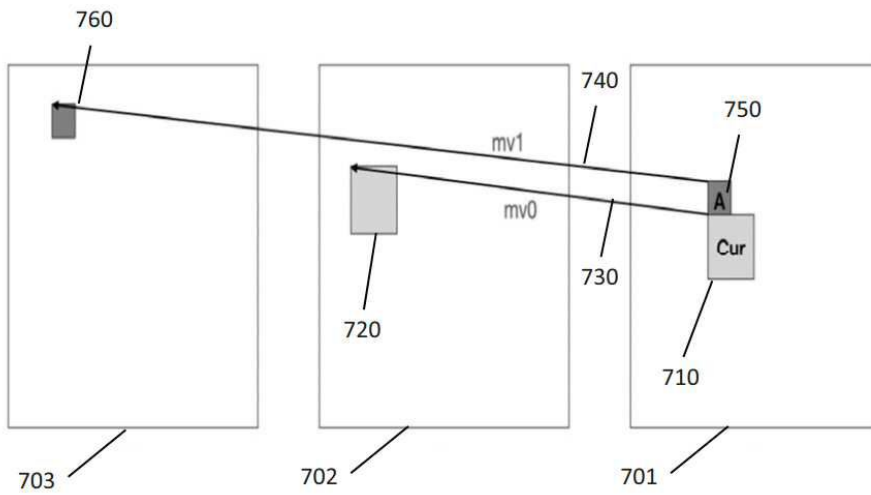
도면5



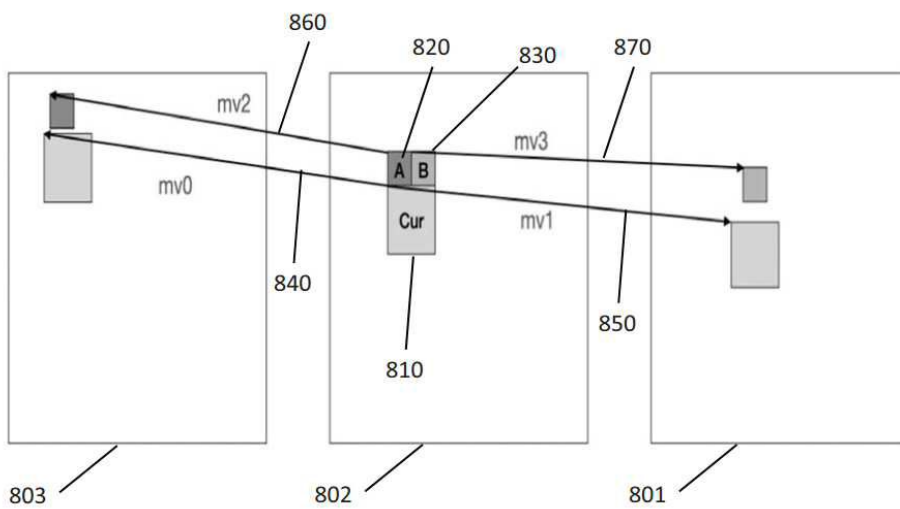
도면6



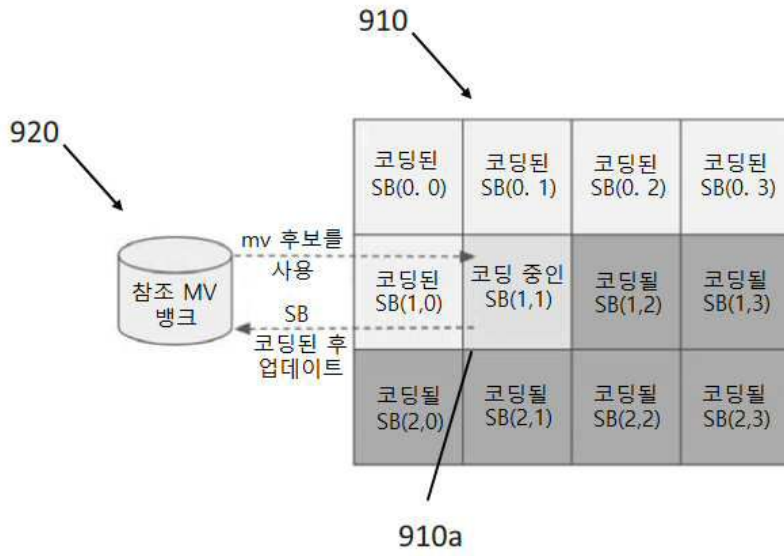
도면7



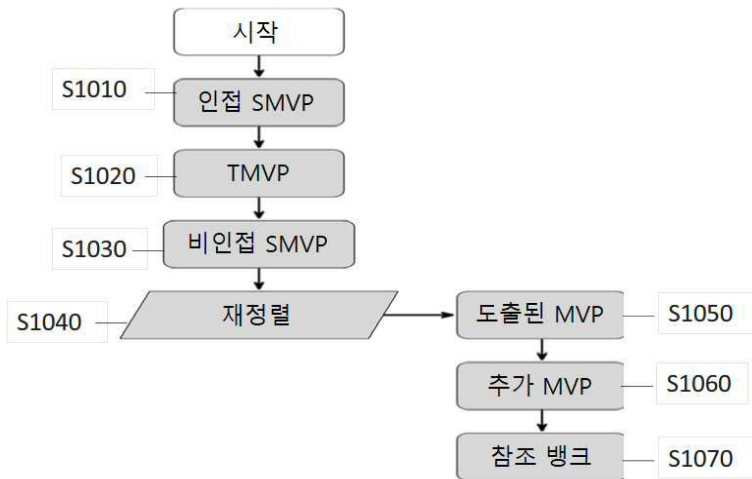
도면8



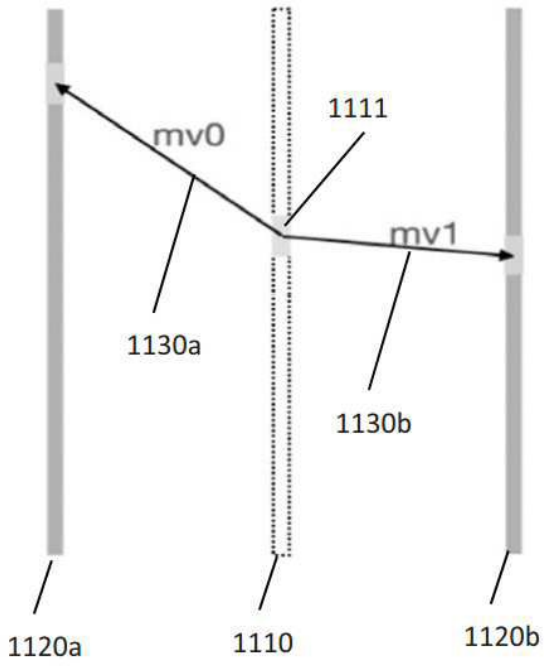
도면9



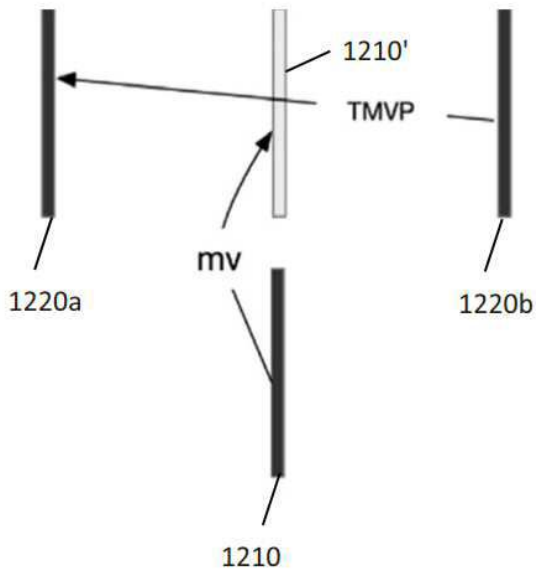
도면10



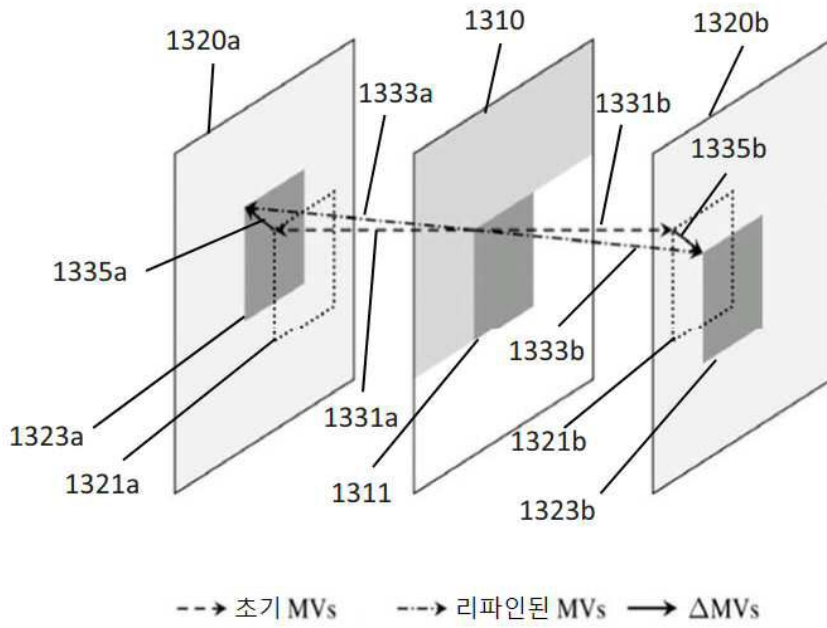
도면11



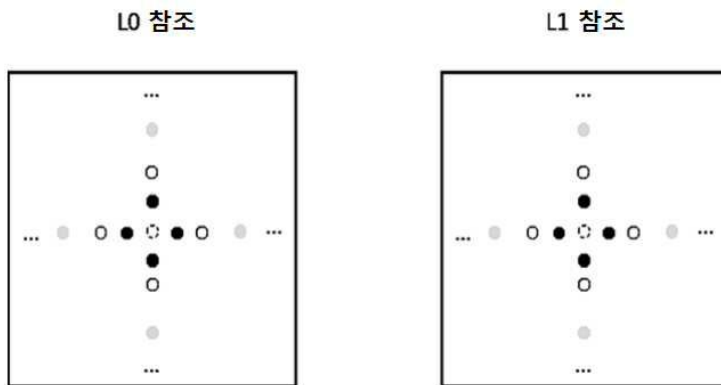
도면12



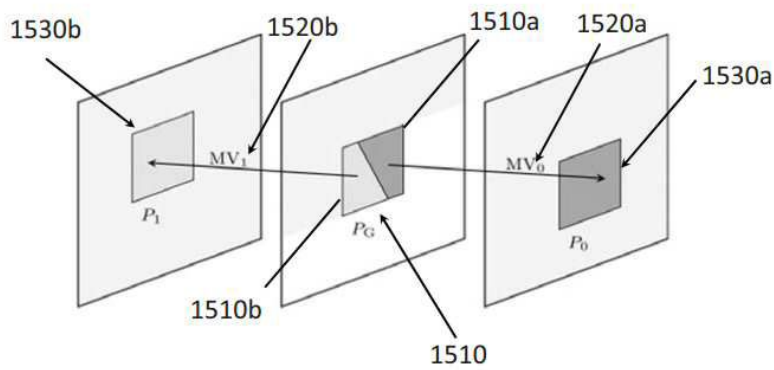
도면13



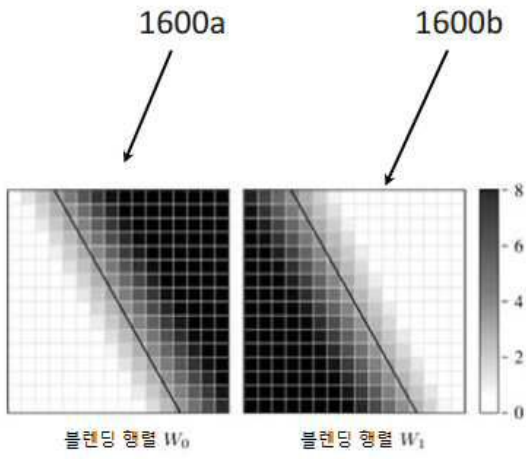
도면14



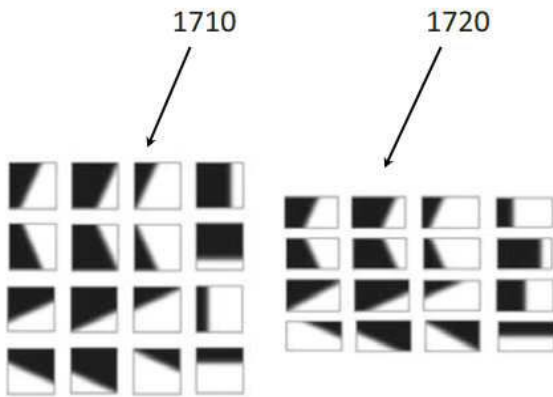
도면15



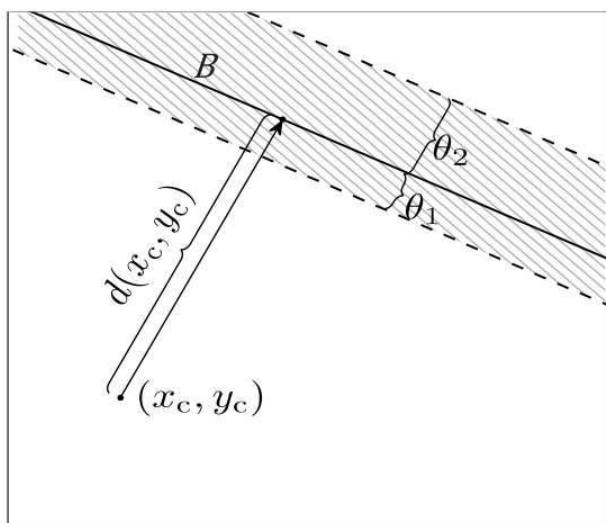
도면16



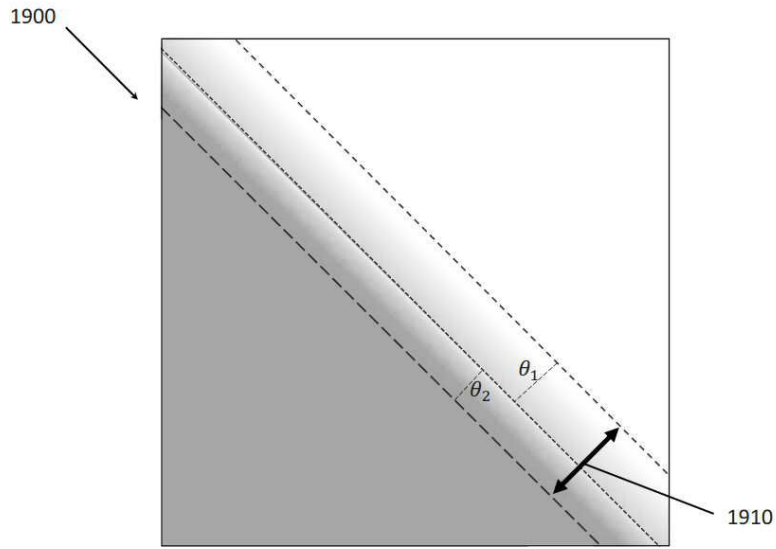
도면17



도면18



도면19



도면20

