



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2024년10월28일

(11) 등록번호 10-2723082

(24) 등록일자 2024년10월24일

(51) 국제특허분류(Int. Cl.)

H04N 19/583 (2014.01) H04N 19/167 (2014.01)

H04N 19/176 (2014.01) H04N 19/423 (2014.01)

H04N 19/70 (2014.01)

(52) CPC특허분류

H04N 19/583 (2015.01)

H04N 19/167 (2015.01)

(21) 출원번호 10-2023-7004482(분할)

(22) 출원일자(국제) 2016년01월26일

심사청구일자 2023년03월09일

(85) 번역문제출일자 2023년02월07일

(65) 공개번호 10-2023-0022282

(43) 공개일자 2023년02월14일

(62) 원출원 특허 10-2017-7020610

원출원일자(국제) 2016년01월26일

심사청구일자 2021년01월07일

(86) 국제출원번호 PCT/US2016/014857

(87) 국제공개번호 WO 2016/123068

국제공개일자 2016년08월04일

(30) 우선권주장

62/107,964 2015년01월26일 미국(US)

(뒷면에 계속)

(56) 선행기술조사문헌

KR1020040108056 A*

KR1020100015456 A*

KR1020140096130 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

퀄컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

리우 홍빈

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

천 잉

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(뒷면에 계속)

(74) 대리인

특허법인코리어나

전체 청구항 수 : 총 14 항

심사관 : 황수진

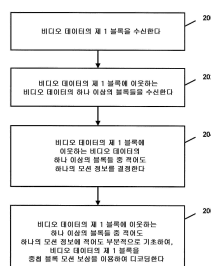
(54) 발명의 명칭 비디오 코딩을 위한 중첩 모션 보상

(57) 요약

일 예에서, 비디오 데이터를 디코딩하는 방법은 비디오 데이터의 제 1 블록을 수신하는 단계를 포함할 수도 있다. 비디오 데이터의 제 1 블록은 예측 유닛의 서브-블록일 수도 있다. 본 방법은 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 수신하는 단계를 포함할 수도 있다. 본 방법은 비

(뒷면에 계속)

대표도 - 도11



디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보를 결정하는 단계를 포함할 수도 있다. 본 방법은 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 모션 정보에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 디코딩하는 단계를 포함할 수도 있다.

(52) CPC특허분류

H04N 19/176 (2015.01)

H04N 19/423 (2015.01)

H04N 19/70 (2015.01)

(30) 우선권주장

62/116,631 2015년02월16일 미국(US)

15/005,934 2016년01월25일 미국(US)

(72) 발명자

천 지안레

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

리 상

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

비디오 데이터의 디코딩하는 방법으로서,

비디오 데이터의 제 1 코딩 유닛을 수신하는 단계로서, 비디오 데이터의 상기 제 1 코딩 유닛은 제 1 서브-블록을 포함하는 복수의 서브-블록들을 포함하는 예측 유닛을 포함하는, 상기 제 1 코딩 유닛을 수신하는 단계;

비디오 데이터의 제 2 코딩 유닛을 수신하는 단계로서, 상기 제 2 코딩 유닛은 비디오 데이터의 상기 제 1 서브-블록에 이웃하는 하나 이상의 서브-블록들을 포함하는 예측 유닛을 포함하는, 상기 제 2 코딩 유닛을 수신하는 단계;

비디오 데이터의 상기 제 1 서브-블록이 중첩 블록 모션 보상 모드에 따라 인코딩되었는지 여부를 나타내는 값을 갖는 신텍스 엘리먼트를 수신하는 단계로서, 상기 신텍스 엘리먼트의 값은 제 1 값 또는 제 2 값이고, 상기 제 1 값은 비디오 데이터의 상기 제 1 서브-블록이 상기 중첩 블록 모션 보상 모드에 따라 인코딩되었음을 나타내고, 상기 제 2 값은 비디오 데이터의 상기 제 1 서브-블록이 상기 중첩 블록 모션 보상 모드에 따라 인코딩되지 않았음을 나타내는, 상기 신텍스 엘리먼트를 수신하는 단계;

비디오 데이터의 상기 제 1 서브-블록이 상기 제 1 값과 동일한 상기 신텍스 엘리먼트의 값에 기초하여 상기 중첩 블록 모션 보상 모드에 따라 인코딩되었음을 결정하는 단계;

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 적어도 하나의 모션 정보를 결정하는 단계; 및

중첩 블록 모션 보상을 이용하여, 비디오 데이터의 상기 제 1 서브-블록에 이웃하는 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 상기 모션 정보에 적어도 부분적으로 기초하여 비디오 데이터의 상기 제 1 서브-블록을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 적어도 하나가 디코딩되지 않았는지 여부를 결정하는 단계; 및

디코딩되지 않은 것이 결정된 상기 하나 이상의 서브-블록들 중 상기 적어도 하나에 대응하는 모션 정보를 이용함이 없이, 비디오 데이터의 상기 제 1 서브-블록을 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 서브-블록을 디코딩하는 것에 의해, 비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 상기 모션 정보를 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 대한 제 1 예측 블록을 발생시키는 단계로서, 상기 제 1 예측 블록은 비디오 데이터의 상기 제 1 서브-블록에 대한 각각의 픽셀 위치에 대한 값을 포함하는, 상기 제 1 예측 블록을 발생시키는 단계;

비디오 데이터의 상기 제 1 코딩 유닛의 사이즈에 기초하여, 하나 이상의 가중된 값들을 상기 제 1 예측 블록의 하나 이상의 값들에 할당하는 단계; 및

상기 제 1 예측 블록의 상기 하나 이상의 값들에 할당된 상기 하나 이상의 가중된 값들을 이용하여, 비디오 데이터의 상기 제 1 서브-블록을 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 4 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 적어도 하나의 상기 모션 정보에 기초하여 제 1 이웃 예측 블록을 발생시키는 단계;

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 사이즈에 기초하여, 하나 이상의 가중된 값들을 상기 제 1 이웃 예측 블록의 하나 이상의 값들에 할당하는 단계; 및

상기 제 1 이웃 예측 블록의 상기 하나 이상의 값들에 할당된 상기 하나 이상의 가중된 값들을 이용하여, 비디오 데이터의 상기 제 1 서브-블록을 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 5 항에 있어서,

수정된 제 1 예측 블록을 생성하기 위해, 상기 제 1 이웃 예측 블록의 상기 하나 이상의 값들에 할당된 상기 하나 이상의 가중된 값들에 기초하여, 상기 제 1 예측 블록의 하나 이상의 값들을 수정하는 단계; 및

상기 수정된 제 1 예측 블록을 이용하여, 비디오 데이터의 상기 제 1 서브-블록을 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 5 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 다른 하나의 모션 정보에 기초하여 제 2 이웃 예측 블록을 생성하는 단계;

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 다른 하나의 사이즈에 기초하여, 하나 이상의 가중된 값들을 상기 제 2 이웃 예측 블록의 하나 이상의 값들에 할당하는 단계; 및

수정된 제 1 예측 블록을 생성하기 위해, 상기 제 1 및 제 2 이웃 예측 블록들의 하나 이상의 값들에 할당된 상기 하나 이상의 가중된 값들에 기초하여 상기 제 1 및 제 2 이웃 예측 블록들의 하나 이상의 값들을 수정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

비디오 데이터를 코딩하기 위한 장치로서,

비디오 데이터의 제 1 코딩 유닛을 수신하는 수단으로서, 비디오 데이터의 상기 제 1 코딩 유닛은 제 1 서브-블록을 포함하는 복수의 서브-블록들을 포함하는 예측 유닛을 포함하는, 상기 제 1 코딩 유닛을 수신하는 수단;

비디오 데이터의 제 2 코딩 유닛을 수신하는 수단으로서, 상기 제 2 코딩 유닛은 비디오 데이터의 상기 제 1 서브-블록에 이웃하는 하나 이상의 서브-블록들을 포함하는 예측 유닛을 포함하는, 상기 제 2 코딩 유닛을 수신하는 수단;

비디오 데이터의 상기 제 1 서브-블록이 중첩 블록 모션 보상 모드에 따라 코딩되었는지 여부를 나타내는 값을 갖는 선택스 엘리먼트를 수신하는 수단으로서, 상기 선택스 엘리먼트의 값은 제 1 값 또는 제 2 값이고, 상기 제 1 값은 비디오 데이터의 상기 제 1 서브-블록이 상기 중첩 블록 모션 보상 모드에 따라 코딩되었음을 나타내고, 상기 제 2 값은 비디오 데이터의 상기 제 1 서브-블록이 상기 중첩 블록 모션 보상 모드에 따라 코딩되지 않았음을 나타내는, 상기 선택스 엘리먼트를 수신하는 수단;

비디오 데이터의 상기 제 1 서브-블록이 상기 제 1 값과 동일한 상기 선택스 엘리먼트의 값에 기초하여 상기 중첩 블록 모션 보상 모드에 따라 코딩되었음을 결정하는 수단;

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 적어도 하나의 모션 정보를 결정하는 수단; 및

중첩 블록 모션 보상을 이용하여, 비디오 데이터의 상기 제 1 서브-블록에 이웃하는 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 상기 모션 정보에 적어도 부분적으로 기초하여 비디오 데이터의 상기 제 1 서브-블록을 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 9

제 8 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 적어도 하나가 코딩되지 않았는지 여부를 결정하는 수단;

코딩되지 않은 것이 결정된 상기 하나 이상의 서브-블록들 중 상기 적어도 하나에 대응하는 모션 정보를 이용함이 없이, 비디오 데이터의 상기 제 1 서브-블록을 코딩하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 10

제 8 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 서브-블록을 코딩하는 것에 의해, 비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 상기 모션 정보를 결정하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 11

제 8 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 대한 제 1 예측 블록을 발생시키는 수단으로서, 상기 제 1 예측 블록은 비디오 데이터의 상기 제 1 서브-블록에 대한 각각의 픽셀 위치에 대한 값을 포함하는, 상기 제 1 예측 블록을 발생시키는 수단;

비디오 데이터의 상기 제 1 코딩 유닛의 사이즈에 기초하여, 하나 이상의 가중된 값들을 상기 제 1 예측 블록의 하나 이상의 값들에 할당하는 수단; 및

상기 제 1 예측 블록의 상기 하나 이상의 값들에 할당된 상기 하나 이상의 가중된 값들을 이용하여, 비디오 데이터의 상기 제 1 서브-블록을 코딩하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 12

제 11 항에 있어서,

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 적어도 하나의 상기 모션 정보에 기초하여 제 1 이웃 예측 블록을 발생시키는 수단;

비디오 데이터의 상기 제 1 서브-블록에 이웃하는 비디오 데이터의 상기 하나 이상의 서브-블록들 중 상기 적어도 하나의 사이즈에 기초하여, 하나 이상의 가중된 값들을 상기 제 1 이웃 예측 블록의 하나 이상의 값들에 할당하는 수단; 및

상기 제 1 이웃 예측 블록의 상기 하나 이상의 값들에 할당된 상기 하나 이상의 가중된 값들을 이용하여, 비디오 데이터의 상기 제 1 서브-블록을 코딩하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 13

제 11 항에 있어서,

수정된 제 1 예측 블록을 생성하기 위해, 상기 제 1 이웃 예측 블록의 상기 하나 이상의 값들에 할당된 상기 하나 이상의 가중된 값들에 기초하여, 상기 제 1 예측 블록의 하나 이상의 값들을 수정하는 수단; 및

상기 수정된 제 1 예측 블록을 이용하여, 비디오 데이터의 상기 제 1 서브-블록을 코딩하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 14

비일시적 컴퓨터 판독가능 저장 매체로서, 실행될 때 하나 이상의 프로세서들로 하여금 제 1 항 내지 제 7 항 중 어느 한 항의 방법을 수행하도록 하게 하는 명령들이 저장되어 있는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2015년 1월 26일에 출원된 미국 가특허 출원번호 제 62/107,964호, 및 2015년 2월 16일에 출원된 미국 가특허 출원번호 제 62/116,631호의 이익을 주장하며, 이의 각각은 본원에 전체적으로 참조로 포함된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 코딩에 관한 것으로, 좀더 구체적으로는, 블록 기반 비디오 코딩에서의 모션 보상에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은, 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDA들), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트폰들", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), ITU-T H.265, HEVC (High Efficiency Video Coding) 에 의해 정의된 표준들, 및 이런 표준들의 확장판들에서 설명되는 것들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이런 비디오 압축 기법들을 구현함으로써, 디지털 비디오 정보를 좀더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거하기 위해 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 수행한다. 블록-기반 비디오 코딩에 있어, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 일부) 는 비디오 블록들로 파티셔닝될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 픽처들은 프레임들로 지칭될 수 있으며, 참조 픽처들은 참조 프레임들로서 지칭될 수도 있다.

[0006] 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 초대한다. 잔차 데이터는 코딩될 원래 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라서 인코딩되며, 잔차 데이터는 코딩된 블록과 예측 블록 사이의 차이를 나타낸다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라서 인코딩된다. 추가적인 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 그 결과 잔차 계수들이 되고, 이 잔차 계수들은 그후 양자화될 수도 있다. 처음에 2차원 어레이로 배열된, 양자화된 계수들은 계수들의 1차원 벡터를 발생하기 위해 스캐닝될 수도 있으며, 엔트로피 코딩이 더욱 더 많은 압축을 달성하기 위해 적용될 수도 있다.

발명의 내용

과제의 해결 수단

[0007] 본 개시물의 기법들은 블록-기반 비디오 코딩에 관한 것이다. 예를 들어, 본 개시물에서 설명하는 기법들은 비디오 데이터의 블록을 중첩 블록 모션 보상 (OBMC) 을 이용하여 인코딩 또는 디코딩하는 하나 이상의 기법들

을 포함할 수도 있다.

[0008] 일 예에서, 본 개시물은 비디오 데이터의 제 1 블록을 수신하는 단계로서, 비디오 데이터의 제 1 블록은 예측 유닛의 서브-블록인, 상기 제 1 블록을 수신하는 단계; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 수신하는 단계; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보를 결정하는 단계; 및 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 모션 정보에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법을 기술한다.

[0009] 다른 예에서, 본 개시물은 비디오 데이터를 저장하도록 구성된 메모리; 및 메모리와 통신하는 비디오 코더를 포함하며, 상기 비디오 코더가, 비디오 데이터의 제 1 블록을 메모리에 저장하는 것으로서, 상기 비디오 데이터의 제 1 블록이 예측 유닛의 서브-블록인, 상기 제 1 블록을 메모리에 저장하고; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 메모리에 저장하고; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보를 결정하고; 그리고 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 모션 정보에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 코딩하도록 구성된, 비디오 데이터를 코딩하는 디바이스를 기술한다.

[0010] 다른 예에서, 본 개시물은 비디오 데이터의 제 1 블록을 수신하는 수단으로서, 상기 비디오 데이터의 제 1 블록이 예측 유닛의 서브-블록인, 상기 제 1 블록을 수신하는 수단; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 수신하는 수단; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보를 결정하는 수단; 및 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 모션 정보에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하는 장치를 기술한다.

[0011] 다른 예에서, 본 개시물은 명령들을 저장하고 있는 비일시성 컴퓨터-판독가능 저장 매체를 기술하며, 상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 데이터의 제 1 블록을 메모리에 저장하게 하는 것으로서, 상기 비디오 데이터의 제 1 블록이 예측 유닛의 서브-블록인, 상기 제 1 블록을 메모리에 저장하게 하고; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 메모리에 저장하게 하고; 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보를 결정하게 하고; 그리고 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 모션 정보에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 코딩하게 한다.

[0012] 본 개시물의 하나 이상의 예들의 세부 사항들은 첨부도면 및 아래의 설명에서 개시된다. 본 개시물의 다른 특징들, 목적들, 및 이점들은 설명 및 도면들로부터, 그리고 청구범위로부터 명백히 알 수 있을 것이다.

도면의 간단한 설명

[0013] 도 1은 본 개시물에서 설명하는 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2는 인터 예측 모드 비디오 코딩을 위한 파티션 모드들을 나타낸다.

도 3a 및 도 3b는 병합 및 진보된 모션 벡터 예측 (AMVP) 비디오 코딩 모드들에 대한 공간 이웃하는 모션 벡터 (MV) 후보들을 각각 나타낸다.

도 4a는 시간 모션 벡터 예측 (TMVP) 후보의 일 예를 나타낸다.

도 4b는 모션 벡터 스케일링의 일 예를 나타낸다.

도 5는 ITU-T H.263 비디오 코딩 표준에서 사용될 때에 중첩 블록 모션 보상 (OBMC)의 일 예를 나타낸다.

도 6a 및 도 6b는 HEVC 또는 다른 비디오 코딩 표준들에서 사용될 수도 있는 예측 유닛 (PU)-기반의 OBMC의 일 예를 나타낸다.

도 7a 내지 도 7c는 PU에서의 서브-PU들 및 서브-블록들을 나타낸다.

도 8a 내지 도 8b는 OBMC가 적용될 수도 있는 서브-블록들을 나타낸다.

도 9는 본 개시물에서 설명된 OBMC 기법들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는

블록도이다.

도 10 은 본 개시물에서 설명된 OBMC 기법들을 구현할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 11 은 본 개시물의 OMBC 비디오 코딩을 위한 기법들에 따른, 비디오 데이터를 디코딩하는 예시적인 프로세스를 예시하는 플로우차트이다.

도 12 는 본 개시물의 OMBC 비디오 코딩을 위한 기법들에 따른, 비디오 데이터를 인코딩하는 예시적인 프로세스를 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

- [0014] 본 개시물의 기법들은 일반적으로 블록 기반의 비디오 코딩에서의 모션 보상에 관한 것이다. 본 개시물의 기법들은 기존 비디오 코덱 또는 비디오 코딩 표준 (예컨대, ITU-T H.265, HEVC) 에 적용될 수도 있거나, 또는 미래 비디오 코덱 또는 비디오 코딩 표준에 적용될 수도 있다.
- [0015] 본원에서 사용될 때, 용어 "콘텐츠" 의 경우에는 용어 "비디오" 로 변경될 수도 있으며, 용어 "비디오" 의 경우에는 용어 "콘텐츠" 로 변경될 수도 있다. 이것은 용어들 "콘텐츠" 또는 "비디오" 가 화법의 형용사, 명사, 또는 다른 품사로서 사용되고 있는지 여부에 관계없이 적용된다. 예를 들어, "콘텐츠 코더" 에 대한 참조는 또한 "비디오 코더" 에 대한 참조를 포함하며, "비디오 코더" 에 대한 참조는 또한 "콘텐츠 코더" 에 대한 참조를 포함한다. 이와 유사하게, "콘텐츠" 에 대한 참조는 또한 "비디오" 에 대한 참조를 포함하며, "비디오" 에 대한 참조는 또한 "콘텐츠" 에 대한 참조를 포함한다.
- [0016] 본원에서 사용될 때, "콘텐츠" 는 임의 종류의 콘텐츠를 지칭한다. 예를 들어, "콘텐츠" 는 비디오, 스크린 콘텐츠, 이미지, 임의의 그래픽 콘텐츠, 임의의 디스플레이가능한 콘텐츠, 또는 그에 대응하는 임의의 데이터 (예컨대, 비디오 데이터, 스크린 콘텐츠 데이터, 이미지 데이터, 그래픽 콘텐츠 데이터, 디스플레이가능한 콘텐츠 데이터, 및 기타 등등) 을 지칭할 수도 있다.
- [0017] 본원에서 사용될 때, 용어 "비디오" 는 스크린 콘텐츠, 가동 콘텐츠 (movable content), 시퀀스로 제시될 수도 있는 복수의 이미지들, 또는 그에 대응하는 임의의 데이터 (예컨대, 스크린 콘텐츠 데이터, 가동 콘텐츠 데이터, 비디오 데이터, 이미지 데이터, 및 기타 등등) 을 지칭할 수도 있다.
- [0018] 본원에서 사용될 때, 용어 "이미지" 는 단일 이미지, 하나 이상의 이미지들, 비디오에 대응하는 복수의 이미지들 중 하나 이상의 이미지들, 비디오에 대응하지 않는 복수의 이미지들 중 하나 이상의 이미지들, 비디오에 대응하는 복수의 이미지들 (예컨대, 비디오에 대응하는 이미지들의 모두 또는 비디오에 대응하는 이미지들의 모두 미만), 단일 이미지의 서브-부분, 단일 이미지의 복수의 서브-부분들, 복수의 이미지들에 대응하는 복수의 서브-부분들, 하나 이상의 그래픽스 프리미티브들, 이미지 데이터, 그래픽 데이터 등을 지칭할 수도 있다.
- [0019] 본원에서 사용될 때, "모션 정보" 는 모션 벡터 정보, 또는 좀더 간단히, 모션 벡터를 지칭하거나 또는 아니면 포함할 수도 있다. 다른 예들에서, "모션 정보" 는 모션 벡터 정보와는 상이한 모션 정보를 지칭하거나 또는 아니면 포함할 수도 있다. 또 다른 예들에서, "모션 정보" 는 하나 이상의 모션 벡터들 및 임의의 다른 모션 관련된 정보, 예컨대, 사용될 참조 픽처 리스트(들) 을 식별하는 예측 방향 및 참조 픽처 리스트(들) 에서 참조 픽처(들) 을 식별하는 하나 이상의 참조 인덱스들을 지칭할 수도 있다. 본원에서 사용될 때, "모션 정보의 세트" 또는 기타 등등은 "모션 정보" 를 지칭할 수도 있다. 이와 유사하게, "모션 정보" 는 "모션 정보의 세트" 또는 기타 등등을 지칭할 수도 있다.
- [0020] 본원에서 사용될 때, "이웃", "이웃하는 블록", "이웃 블록", 및 기타 등등은 적어도 하나의 측면/경계선 상에서 비디오 데이터의 다른 블록에 이웃하는 비디오의 블록을 지칭한다. 예를 들어, 비디오 데이터의 현재의 블록은 4개의 측면들: 좌측, 우측, 상부, 및 저부를 가질 수도 있다. 비디오 데이터의 현재의 블록에 이웃하는 비디오 데이터의 블록은 비디오의 현재의 블록의 좌측, 우측, 상부, 또는 저부 경계선과 경계를 이루는 비디오 데이터의 임의의 블록을 포함할 수도 있다. 다른 예로서, 제 2 블록에 이웃하는 제 1 블록은 제 2 블록의 경계선 (예컨대, 좌측 경계선, 우측 경계선, 상부 경계선, 또는 저부 경계선) 을 공유한다.
- [0021] 도 1 은 본 개시물의 기법들을 이용할 수도 있는 예시적인 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 사용될 때, 용어 "비디오 코더" 는 비디오 인코더들 및 비디오 디코더들 양쪽을 포괄적으로 지칭한다. 본 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 비디오 인코딩 또는 비디오 디코딩을 포괄적으로

지칭할 수도 있다. 비디오 코딩 시스템 (10) 의 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물에서 설명된 여러 예들에 따른, 비디오 데이터의 블록을 중첩 블록 모션 보상을 이용하여 인코딩 또는 디코딩하는 기법들을 수행하도록 구성될 수도 있는 디바이스들의 예를 나타낸다.

[0022] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 본 개시물에서 설명하는 기법들에 따라서 동작하도록 구성될 수도 있다. 비디오 디코더 (30) 는 본원에서 설명되는 비디오 인코더 (20) 의 프로세스와는 일반적으로 반대인 프로세스를 수행하도록 구성될 수도 있다. 이와 유사하게, 비디오 인코더 (20) 는 본원에서 설명되는 비디오 디코더 (30) 의 프로세스와는 일반적으로 반대인 프로세스를 수행하도록 구성될 수도 있다.

[0023] 도 1 에 나타낸 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 및 목적지 디바이스 (14) 를 포함한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 발생한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치로서 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 발생된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치로서 지칭될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.

[0024] 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예컨대, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 자동차용 컴퓨터들, 또는 기타 등등을 포함한, 광범위한 디바이스들을 포함할 수도 있다.

[0025] 목적지 디바이스 (14) 는 소스 디바이스 (12) 로부터 채널 (16) 을 통해서 인코딩된 비디오 데이터를 수신할 수도 있다. 채널 (16) 은 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시키는 것이 가능한 하나 이상의 매체들 또는 디바이스들 및/또는 임의 유형의 매체를 포함할 수도 있다. 일 예에서, 채널 (16) 은 소스 디바이스 (12) 로 하여금, 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 무선 통신 프로토콜과 같은 통신 표준에 따라서, 인코딩된 비디오 데이터를 변조할 수도 있으며, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 와 같은, 패킷-기반 네트워크의 일부를 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.

[0026] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터, 소스 디바이스 (12) 에 의해 발생하는 인코딩된 비디오 데이터를 저장하도록 구성될 수도 있는 저장 디바이스 (26) 로 출력될 수도 있다. 채널 (16) 로부터 분리된 것으로 도시되지만, 다른 예들에서 채널 (16) 이 저장 디바이스 (26) 에 통신가능하게 커플링될 수도 있음을 알 수 있다. 이 예에서, 목적지 디바이스 (14) 는 저장 디바이스 (26) 상에 저장된 인코딩된 비디오 데이터에 액세스할 수도 있다. 예를 들면, 인코딩된 비디오 데이터는 입력 인터페이스 (28) 에 의해 저장 디바이스 (26) 로부터 액세스될 수도 있다.

[0027] 저장 디바이스 (26) 는 하드 드라이브, 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 로컬 액세스되는 데이터 저장 매체들 중 임의의 데이터 저장 매체를 포함할 수도 있다. 일부 예들에서, 저장 디바이스 (26) 는 소스 디바이스 (12) 에 의해 발생된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 이러한 예들에서, 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신하는 것이 가능한 임의 종류의 서버일 수도 있다. 예시적인 파일 서버들은 예를 들어, (예컨대, 웹사이트용) 웹 서버들, 파일 전송 프로토콜 (FTP) 서버들, NAS (network attached storage) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.

[0028] 목적지 디바이스 (14) 는 예를 들어, 인터넷 접속을 포함한, 임의의 표준 접속 (예컨대, 임의의 표준 데이터 접속 또는 그 밖의 방법) 을 통해서, 인코딩된 비디오 데이터에 액세스할 수도 있다. 데이터 접속들의 예시적인 유형들은 무선 채널 (예컨대, Wi-Fi 접속), 유선 채널 (예컨대, DSL, 케이블 모뎀, 등), 또는 저장 디바이스 (26) (예컨대, 파일 서버) 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 임의의 양자의 조합을

포함할 수도 있다. 저장 디바이스 (26) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다.

[0029] 본 개시물의 기법들은 무선 애플리케이션들 또는 환경들에 한정되지 않는다. 이 개시물의 기법들은 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 인코딩된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중 임의의 애플리케이션의 지원 하에, 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0030] 도 1 에 예시된 비디오 코딩 시스템 (10) 은 단지 일 예이며, 본 개시물의 기법들은 인코딩 디바이스와 디코딩 디바이스 사이의 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 환경들 (예컨대, 비디오 인코딩 또는 비디오 디코딩) 에 적용할 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 추출되어, 네트워크 등을 통해서 스트리밍된다. 비디오 인코딩 디바이스는 데이터를 인코딩하여 메모리에 저장할 수도 있거나, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 추출하여 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은 서로 통신하지 않지만 간단히 데이터를 메모리로 인코딩하거나 및/또는 메모리로부터 데이터를 추출하여 디코딩하는 디바이스들에 의해 수행된다.

[0031] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 일부 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에-캡처된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하는 비디오 공급 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 이런 비디오 데이터의 소스들의 조합을 포함할 수도 있다.

[0032] 비디오 인코더 (20) 는 비디오 소스 (18) 로부터의 비디오 데이터를 인코딩하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 캡처된, 사전-캡처된, 또는 컴퓨터-발생된 비디오 데이터 (또는, 임의의 다른 데이터) 를 인코딩하도록 구성될 수도 있다. 일부 예들에서, 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 출력 인터페이스 (22) 를 통해서 목적지 디바이스 (14) 로 직접 송신한다. 다른 예들에서, 인코딩된 비디오 데이터는 또한 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14) 에 의한 추후 액세스를 위해 저장 매체 (예컨대, 저장 디바이스 (26)) 상으로 저장될 수도 있다.

[0033] 도 1 의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함한다. 입력 인터페이스 (28) 는 인코딩된 비디오 데이터를 채널 (16) 을 통해서 수신할 수도 있다. 채널 (16) 을 통해서 통신되거나, 또는 저장 디바이스 (26) 에 의해 제공되는 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩할 때에, 비디오 디코더 (30) 와 같은 비디오 디코더에 의해 사용하기 위한, 비디오 인코더 (20) 에 의해 발생하는 다양한 신호스 엘리먼트들을 포함할 수도 있다. 이런 신호스 엘리먼트들은 통신 매체 상으로 송신되거나, 저장 매체 상에 저장되거나, 또는 파일 서버에 저장된 인코딩된 비디오 데이터와 함께 포함될 수도 있다.

[0034] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 또는 그 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하며, 또한 외부 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32) 는 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디스플레이 디바이스를 포함할 수도 있다.

[0035] 본 개시물의 기법들은 설명의 용이성을 위해 HEVC 전문용어 또는 다른 비디오 표준 전문용어를 사용할 수도 있다. 그러나, 본 개시물의 기법들이 HEVC 또는 다른 비디오 표준들에 한정되지 않음을 알 수 있다. 본 개시물의 기법들은 HEVC 및 그의 확장판들에 대한 후속 표준들 뿐만 아니라, 과거, 현재, 또는 미래의 어떤 다른 비디오 표준들에서 구현될 수도 있다.

[0036] 도 1 에 나타내지는 않지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 오디오 인코더 및

디코더와 각각 통합될 수도 있으며, 오디오 및 비디오 양쪽의 인코딩을 공통 데이터 스트림 또는 별개의 데이터 스트림들로 처리하기에 적합한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능한 경우, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대, 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0037] 본 개시물은 일반적으로 어떤 정보를 비디오 디코더 (30) 와 같은 또 다른 디바이스로 "시그널링하는" 또는 "송신하는" 비디오 인코더 (20) 를 참조할 수도 있다. 용어 "시그널링하는 것" 또는 "송신하는 것" 은 일반적으로, 압축된 비디오 데이터를 디코딩하는데 사용되는 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 이런 통신은 실시간 또는 거의-실시간으로 일어날 수도 있다. 대안적으로, 이런 통신은 어떤 기간에 걸쳐서 일어날 수도 있으며, 예컨대 인코딩 시에 선택스 엘리먼트들을 컴퓨터-판독가능 저장 매체에 인코딩된 비트스트림으로 저장할 때에 발생할 수도 있으며, 이 선택스 엘리먼트들은 그후 이 매체에 저장되어진 후 언제라도 디코딩 디바이스에 의해 추출될 수도 있다. 따라서, 비디오 디코더 (30) 는 어떤 정보를 "수신하는" 것으로서 지칭될 수도 있지만, 정보의 수신은 반드시 실시간 거의-실시간으로 일어날 필요는 없으며 저장 이후 언젠가 매체로부터 추출될 수도 있다.

[0038] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 이산 로직, 하드웨어, 또는 임의의 이들의 조합들과 같은, 다양한 적합한 회로 중 임의의 회로로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현되면, 디바이스는 소프트웨어용 명령들을 적합한 비일시성 컴퓨터-판독가능 저장 매체에 저장할 수도 있으며, 본 개시물의 기법들을 수행하기 위해 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 이용하여 실행할 수도 있다. (하드웨어, 소프트웨어, 하드웨어와 소프트웨어의 조합 등을 포함한) 전술한 것 중 임의의 것이 하나 이상의 프로세서들로 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 개별 디바이스에서 결합된 인코더/디코더 (CODEC) 의 일부분으로서 통합될 수도 있다.

[0039] HEVC 및 다른 비디오 코딩 표준들에서, 비디오 시퀀스는 일반적으로 일련의 픽처들을 포함한다. 픽처들은 또한 "프레임들" 로서 지칭될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 POC (picture order count) 를 이용하여 복수의 픽처들 (예컨대, 픽처들의 시퀀스) 에 대한 픽처의 디스플레이 순서를 식별하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 는 POC 값을 픽처에 할당하도록 구성될 수도 있다. 다수의 코딩된 비디오 시퀀스들이 비트스트림에 존재하는 예에서, 동일한 POC 값을 가진 픽처들은 디코딩 순서의 관점에서 서로 더 가까울 수도 있다. 픽처들의 POC 값들은 참조 픽처 리스트 구성, 예를 들어, HEVC 에서와 같은 참조 픽처 세트의 유도, 및 모션 벡터 스케일링에 이용될 수도 있다.

[0040] 일부 예들에서, 픽처는 3개의 샘플 어레이들을 포함할 수도 있다. 이런 예들에서, 픽처는 S_L , S_{Cb} 및 S_{Cr} 로 표기되는, 3개의 샘플 어레이들을 포함할 수도 있다. 이러한 예들에서, S_L 은 루마 샘플들의 2차원 어레이 (예컨대, 블록) 이고, S_{Cb} 는 Cb 색차 샘플들의 2차원 어레이 (예컨대, 블록) 이며, S_{Cr} 는 Cr 색차 샘플들의 2차원 어레이 (예컨대, 블록) 이다. 색차 샘플들은 또한 본원에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우, 픽처는 단색일 수도 있으며 단지 루마 샘플들의 어레이를 포함할 수도 있다.

[0041] 픽처의 인코딩된 표현을 발생시키기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛들 (CTU들) 의 세트를 발생시킬 수도 있다. CTU들의 세트는 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 $N \times N$ 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로서 지칭될 수도 있다. HEVC 의 CTU들은 H.264/AVC 와 같은, 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 특정의 사이즈에 반드시 제한되지 않으며, 하나 이상의 코딩 유닛들 (CU들) 을 포함할 수도 있다. 비디오 프레임 또는 픽처는 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 슬라이스는 래스터 스캔으로 연속적으로 순서화된 CTU들의 정수를 포함할 수도 있다. 코딩된 슬라이스는 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스의 슬라이스 헤더는 슬라이스에 관한 정보를 제공하는 선택스 엘리먼트들을 포함하는 선택스 구조일 수도 있다. 슬라이스 데이터는 슬라이스의 코딩된 CTU들을 포함할 수도 있다.

[0042] 일부 예들에서, CU 는 코딩 노드, 및 그 코딩 노드와 연관되는 하나 이상의 예측 유닛들 (PU들) 및 변환 유닛들 (TU들) 을 포함할 수도 있다. CU 의 사이즈는 코딩 노드의 사이즈에 대응할 수도 있으며 형상은 정사각형일

수도 있다. CU의 사이즈는 예를 들어, 8x8 픽셀들로부터 64x64 픽셀들 이상의 최대치를 갖는 트리 블록의 사이즈까지 이를 수도 있다. 각각의 CU는 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다.

CU와 연관되는 선택스 데이터는 예를 들어, 하나 이상의 PU들로의 CU의 파티셔닝을 기술할 수도 있다. 파티셔닝 모드들은 CU가 스킵되는지 또는 직접 모드 인코딩될지, 인트라-예측 모드 인코딩될지, 또는 인터-예측 모드 인코딩될지 여부의 사이에서 상이할 수도 있다. PU들은 정사각형 또는 비-정사각형의 형태로 파티셔닝될 수도 있다. CU와 연관되는 선택스 데이터는 또한 예를 들어, 쿼드트리에 따른 하나 이상의 TU들로의 CU의 파티셔닝을 기술할 수도 있다. TU는 정사각형 또는 비-정사각형의 형태일 수 있다.

[0043] 일반적으로, PU는 예측 프로세스에 관련되는 데이터를 포함할 수도 있다. 예를 들어, PU가 인트라-모드 인코딩될 때, PU는 그 PU에 대한 인트라-예측 모드를 기술하는 데이터를 포함할 수도 있다. 다른 예로서, PU가 인터-모드 인코딩될 때, PU는 그 PU에 대한 모션 벡터를 정의하는 데이터를 포함할 수도 있다. PU에 대한 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터에 대한 해상도 (예컨대, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 참조 픽처, 및/또는 모션 벡터에 대한 참조 픽처 리스트 (예컨대, List 0, List 1, 또는 List C)를 기술할 수도 있다.

[0044] 일반적으로, TU는 변환 및 양자화 프로세스들에 사용될 수도 있다. 하나 이상의 PU들을 갖는 주어진 CU는 또한 하나 이상의 변환 유닛들 (TU들)을 포함할 수도 있다. 예측 이후, 비디오 인코더 (20)는 PU에 대응하는 잔차 값들을 계산할 수도 있다. 잔차 값들은 엔트로피 코딩을 위한 직렬화된 변환 계수들을 생성하기 위해 변환 계수들로 변환되고, 양자화되고, 그리고 TU들을 이용하여 스캐닝될 수도 있는 픽셀 차이 값들을 포함할 수도 있다.

[0045] 본 개시물은 하나 이상의 샘플 블록들 및 하나 이상의 샘플들의 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들을 지칭하기 위해, 용어 "비디오 유닛", "비디오 블록", "코딩 블록", 또는 "블록"을 이용할 수도 있다. 비디오 유닛들 또는 블록들의 예시적인 유형들은 코딩 트리 유닛들 (CTU들), 코딩 유닛들 (CU들), 예측 유닛들 (PU들), 서브-PU들, 변환 유닛들 (TU들), 매크로블록들 (MB들), 매크로블록 파티션들, 서브-블록들 등을 포함할 수도 있다. 일부 예들에서, 서브-블록은 CTU의 서브-블록, CU의 서브-블록, PU의 서브-블록, TU의 서브-블록, 매크로블록의 서브-블록, 또는 서브-블록의 서브-블록일 수도 있다. 예를 들어, 서브-블록은 서브-PU들의 그룹을 포함할 수도 있다. 이러한 예들에서, 서브-블록은 CTU, CU, PU, TU, 또는 매크로블록보다 더 작을 수도 있다. 일부 예들에서, 서브-PU는 PU보다 작은 블록을 지칭할 수도 있다. 이러한 예에서, PU가 8x4이면, 서브-PU는 4x4일 수도 있다.

[0046] 일부 예들에서, 모션 정보의 세트는 비디오 데이터의 각각의 블록에 이용가능할 수도 있다. 모션 정보의 세트는 순방향 및 역방향 예측 방향들에 대한 모션 정보를 포함할 수도 있다. 순방향 및 역방향 예측 방향들은 양방향 예측 모드의 2개의 예측 방향들일 수도 있다. 순방향 및 역방향 예측 방향들은 단방향 예측 모드의 2개의 예측 방향들 중 하나일 수도 있다. 용어들 "순방향" 및 "역방향"은 반드시 기하학적 의미를 갖기 보다는, 그들은 예를 들어, 현재의 픽처의 참조 픽처 리스트 0 (RefPicList0) 및 참조 픽처 리스트 1 (RefPicList1)에 대응한다. 오직 하나의 참조 픽처 리스트가 픽처 또는 슬라이스에 이용가능한 경우, 단지 RefPicList0만이 이용가능하며 픽처 또는 슬라이스의 각각의 블록의 모션 정보는 순방향일 수도 있다.

[0047] 각각의 예측 방향 (예컨대, 순방향 또는 역방향)에 대해, 모션 정보는 예측 방향, 참조 인덱스, 및 모션 벡터를 포함할 수도 있다. 일부 예들에서, 간결성을 위해, 모션 벡터 자체는 연관된 참조 인덱스를 가진다고 가정되는 방법으로 참조될 수도 있다. 참조 인덱스는 현재의 참조 픽처 리스트 (예컨대, RefPicList0 또는 RefPicList1)에서 참조 픽처를 식별하기 위해 사용된다. 모션 벡터는 수평 및 수직 성분을 가질 수도 있다.

[0048] 본원에서 설명되는 비디오 블록들은 고정 또는 가변 사이즈들을 가질 수도 있으며, 규정된 코딩 표준에 따라서 사이즈가 상이할 수도 있다. 일 예로서, 특정의 CU의 사이즈는 2Nx2N일 수도 있다. 이러한 예에서, 비디오 인코더 (20)는 2Nx2N 또는 NxN의 사이즈들을 가진 PU들에 대해 인트라-예측을 수행하도록 구성될 수도 있으며, 그리고, 2Nx2N, 2NxN, Nx2N, 또는 NxN의 사이즈들을 가진 PU들에 대해서는 인터-예측을 수행하도록 구성될 수도 있다. 다른 예들에서, PU의 가용 사이즈들은 동일하거나 또는 상이할 수도 있다.

[0049] 본 개시물에서, "NxN" 및 "N 곱하기 N"은 수직 및 수평 치수들의 관점에서 비디오 블록의 픽셀 치수들, 예컨대, 16x16 픽셀들 또는 16 곱하기 16 픽셀들을 지칭하기 위해 상호교환가능하게 사용될 수도 있다. 일반적으로, 16x16 블록은 수직 방향으로 16개의 픽셀들 (y = 16) 및 수평 방향으로 16개의 픽셀들 (x = 16)을 가질 것이다. 이와 유사하게, NxN 블록은 일반적으로 수직 방향으로 N개의 픽셀들 및 수평 방향으로 N개

의 픽셀들을 가지며, 여기서 N 은 양의 정수 값을 나타낸다. 블록에서 픽셀들은 로우들 및 칼럼들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향에서와 같이 수평 방향에서 동일한 픽셀들의 개수를 반드시 가질 필요는 없다. 예를 들어, 블록들은 $N \times M$ 픽셀들을 포함할 수도 있으며, 여기서, M 은 N 이거나 N 과 동일하지 않으며, M 은 양의 정수 값이다.

[0050] 일부 예들에서, 본원에서 개시된 CU 의 구조는 H.265/HEVC 와 같은 비디오 코딩 표준에서 개시된 바와 같은 CU 의 구조를 지칭할 수도 있다. HEVC 에서, 슬라이스에서의 최대 코딩 유닛은 코딩 트리 블록 (CTB) 로 불린다. CTB 는 쿼드-트리를 포함하며, 그 쿼드-트리의 노드들은 코딩 유닛들이다. CTB 의 사이즈는 HEVC 메인 프로파일에서 16x16 로부터 64x64 까지의 범위이며; 일부 예들에서, 8x8 CTB 사이즈들이 또한 지원된다. CU 는 CTB 와 동일한 사이즈일 수 있으며 8x8 만큼 작을 수도 있다. 각각의 CU 는 하나의 모드로 코딩된다. CU 가 인터 예측 모드 코딩될 때, CU 는 2 또는 4 개의 예측 유닛들 (PU들) 로 추가로 파티셔닝되거나 또는 추가적인 파티션이 적용되지 않을 때에는 오직 하나의 PU (예컨대, 도 2 에 나타난 PART_2Nx2N) 가 될 수도 있다. 2개의 PU들이 하나의 CU 에 존재할 때, 그들은 절반 사이즈 직사각형들 (예컨대, 도 2 에 나타난 PART_2NxN 또는 PART_Nx2N) 또는 하나의 1/4 사이즈 및 나머지 3/4 사이즈를 가진 2개의 직사각형들 (예컨대, 도 2 에 나타난 PART_2NxN, PART_2NxN, PART_2NxN, PART_2NxN, PART_2NxN, PART_2NxN, PART_2NxN 및 PART_2NxN) 일 수도 있다. CU 가 인터 코딩될 때, 모션 정보의 하나의 세트가 각각의 PU 에 대해 존재한다. 또, 각각의 PU 는 모션 정보의 세트를 유도하기 위해 고유한 인터-예측 모드로 코딩된다.

[0051] 예컨대, ITU-T H.264/AVC 또는 다른 표준들에서의 매크로블록들을 참조하면, 각각의 인터 매크로블록 (MB) 은 일 예에 따라서 다음 4가지 상이한 방법들로 파티셔닝될 수도 있다: 하나의 16x16 MB 파티션, 2개의 16x8 MB 파티션들, 2개의 8x16 MB 파티션들, 또는 4개의 8x8 MB 파티션들. 하나의 MB 에서의 상이한 MB 파티션들은 각각의 방향에 대해 상이한 참조 인덱스 값들 (RefPicList0 또는 RefPicList1) 을 가질 수도 있다. MB 가 4개의 8x8 MB 파티션들로 파티셔닝되지 않는 예에서, MB 는 각각의 방향에서 각각의 MB 파티션에 대해 오직 하나의 모션 벡터만을 가질 수도 있다. MB 가 4개의 8x8 MB 파티션들로 파티셔닝되는 예에서, 각각의 8x8 MB 파티션은 서브-블록들로 추가로 파티셔닝될 수도 있으며, 서브-블록들의 각각은 각각의 방향에서 상이한 모션 벡터를 가질 수도 있다. 이러한 예에서, 8x8 MB 파티션은 하나 이상의 서브-블록들로 다음 4가지 상이한 방법들로 파티셔닝될 수도 있다: 하나의 8x8 서브-블록, 2개의 8x4 서브-블록들, 2개의 4x8 서브-블록들, 또는 4개의 4x4 서브-블록들. 각각의 서브-블록은 각각의 방향에서 상이한 모션 벡터를 가질 수도 있다. 이러한 예에서, 각각의 모션 벡터는 서브-블록과 같거나 또는 더 높은 레벨에 존재할 수도 있다.

[0052] 일부 예들에서, 본원에서 개시된 MB 의 구조는 ITU-T H.264/AVC 와 같은 비디오 코딩 표준에 개시된 바와 같은 MB 의 구조를 지칭할 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 B 슬라이스들에서의 스킵 또는 직접 모드에 대해 MB 또는 MB 파티션 레벨에서 시간 직접 모드를 이용하여 비디오 데이터를 코딩하도록 구성될 수도 있다. 각각의 MB 파티션에 대해, 현재의 블록의 RefPicList1[0] 에서 현재의 MB 파티션과 병치된 블록의 모션 벡터들이 모션 벡터들을 유도하기 위해 사용될 수도 있다. 병치된 블록에서의 각각의 모션 벡터는 POC 거리들에 기초하여 스케일링될 수도 있다. H.264/AVC 는 또한 공간 이웃들로부터 모션 정보를 예측하는데 사용될 수도 있는 공간 직접 모드를 포함한다.

[0053] 도 1 을 참조하면, 코딩된 CTU 를 발생시키기 위해, 비디오 인코더 (20) 는 코딩 트리 블록들을 코딩 블록들, 따라서 이름 "코딩 트리 유닛들" 로 분할하기 위해 CTU 의 코딩 트리 블록들에 관해 쿼드-트리 파티셔닝을 회귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 $N \times N$ (또는, $N \times M$) 블록일 수도 있다. CU 는 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 코딩 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 PU들로 파티셔닝할 수도 있다. PU 는 동일한 예측 기법이 적용되는 샘플들의 정사각형 또는 비-정사각형 블록일 수도 있다. CU 의 PU 는 루마 샘플들의 예측 블록, 픽처의 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용되는 선택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb, 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 발생시킬 수도 있다.

[0054] 비디오 인코더 (20) 는 PU 에 대한 예측 블록들을 발생시키기 위해 인트라 예측 또는 인터 예측을 이용할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 발생시키기 위해 인트라 예측을 이용하면, 비디오 인코

더 (20) 는 PU 와 연관되는 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 발생시킬 수도 있다.

[0055] 비디오 인코더 (20) 가 PU 의 예측 블록들을 발생하기 위해 인터 예측을 이용하면, 비디오 인코더 (20) 는 PU 와 연관되는 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여, PU 의 예측 블록들을 발생시킬 수도 있다. 비디오 인코더 (20) 는 PU 의 예측 블록들을 발생시키기 위해 단방향-예측 또는 양방향-예측을 이용할 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 발생시키기 위해 단방향-예측을 이용할 때, PU 는 단일 모션 벡터 (MV) 를 가질 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 발생시키기 위해 양방향-예측을 이용할 때, PU 는 2개의 MV들을 가질 수도 있다.

[0056] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들) 을 발생시킨 후, 비디오 인코더 (20) 는 CU 에 대한 잔차 블록들을 발생시킬 수도 있다. CU 의 잔차 블록에서의 각각의 샘플은 CU 의 PU 의 예측 블록에서의 샘플과 CU 의 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 발생시킬 수도 있다. CU 의 루마 잔차 블록에서의 각각의 샘플은 CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 원래 루마 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낸다. 게다가, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 발생시킬 수도 있다. CU 의 Cb 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 원래 Cb 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한 CU 에 대한 Cr 잔차 블록을 발생시킬 수도 있다. CU 의 Cr 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 원래 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다.

[0057] 비디오 인코더 (20) 는 쿼드-트리 파티셔닝을 이용하여, 잔차 블록들 (예컨대, 루마, Cb, 및 Cr 잔차 블록들) 을 하나 이상의 변환 블록들 (예컨대, 루마, Cb, 및 Cr 변환 블록들) 로 분해할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 블록일 수도 있다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용되는 선택스 구조들일 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관되는 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다.

[0058] 비디오 인코더 (20) 는 하나 이상의 변환들을 변환 블록에 적용하여, TU 에 대한 계수 블록을 발생시킬 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 예를 들어, 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용하여, TU 에 대한 루마 계수 블록을 발생시킬 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용하여, TU 에 대한 Cb 계수 블록을 발생시킬 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용하여, TU 에 대한 Cr 계수 블록을 발생시킬 수도 있다.

[0059] 일부 예들에서, CU 의 PU들을 이용한 인트라-예측 또는 인터-예측 코딩 이후, 인코더 (20) 는 CU 의 TU들에 대한 잔차 데이터를 계산할 수도 있다. PU들은 공간 도메인 (또한, 픽셀 도메인으로 지칭됨) 에서 픽셀 데이터를 포함할 수도 있으며, TU들은 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 잔차 비디오 데이터에 개념적으로 유사한 변환과 같은 변환의 적용 이후 변환 도메인에서 계수들을 포함할 수도 있다. 잔차 데이터는 미인코딩된 픽처의 픽셀들과 PU들에 대응하는 예측 값들 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔차 데이터를 포함하는 TU들을 형성하고, 그 후 그 TU들을 변환하여, 그 CU 에 대한 변환 계수들을 생성할 수도 있다.

[0060] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록, 또는 Cr 계수 블록) 을 발생시킨 후, 비디오 인코더 (20) 는 그 블록에서의 계수들을 양자화할 수도 있다. 양자화는 일반적으로 변환 계수들이 변환 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감소시키도록 양자화되어 추가적인 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들을 엔트로피 인코딩하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들에 관해 컨텍스트-적응 2진 산술 코딩 (CABAC) 을 수행할 수도 있다.

[0061] 일부 예들에서, 비디오 인코더 (20) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 발생하기 위해, 미리 정의된 스캔 순서 (예컨대, 수평, 수직, 또는 임의의 다른 스캔 순서) 를 이용하여, 양자화된 변환 계수들을 스캐닝할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캐닝을 수행할 수도 있다. 양자화된 변

환 계수들을 스캐닝하여 1차원 벡터를 형성한 후, 비디오 인코더 (20) 는 1차원 벡터를, 예컨대, 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 2진 산술 코딩 (CABAC), 신택스-기반 컨텍스트-적응 2진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라서, 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한 비디오 데이터를 디코딩할 때에 비디오 디코더 (30) 에 의해 사용하기 위한 인코딩된 비디오 데이터와 연관되는 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0062] CABAC 를 수행하기 위해, 비디오 인코더 (20) 는 컨텍스트 모델 내 컨텍스트를 송신될 심볼에 할당할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 비-제로인지 여부에 관련될 수도 있다. CAVLC 를 수행하기 위해, 비디오 인코더 (20) 는 송신될 심볼에 대해 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드들은 상대적으로 더 짧은 코드들이 더 가능성 있는 심볼들에 대응하지만, 더 긴 코드들이 덜 가능성 있는 심볼들에 대응하도록, 구성될 수도 있다. 이와 같이, VLC 의 사용은 예를 들어, 송신될 각각의 심볼에 대해 동일-길이 코드워드들을 사용하는 것을 넘어서 비트 절감을 달성할 수도 있다. 확률 결정은 그 심볼에 할당된 컨텍스트에 기초할 수도 있다.

[0063] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 발생하는 비트스트림을 수신할 수도 있다. 게다가, 비디오 디코더 (30) 는 비트스트림을 파싱하여, 비트스트림으로부터 신택스 엘리먼트들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하는 프로세스는 일반적으로 비디오 인코더 (20) 에 의해 수행되는 프로세스와 반대일 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재의 CU 의 인터-예측된 PU들에 대한 예측 블록들을 결정하기 위해 PU들의 MV들을 이용할 수도 있다. 이와 유사하게, 비디오 디코더 (30) 는 현재의 CU 의 PU들에 대한 인트라-예측된 블록들을 발생시킬 수도 있다. 게다가, 비디오 디코더 (30) 는 현재의 CU 의 TU들과 연관되는 변환 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 변환 계수 블록들에 관해 역변환들을 수행하여, 현재의 CU 의 TU들과 연관되는 변환 블록들을 복원할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 PU들에 대한 예측 블록들의 샘플들을, 현재의 CU 의 TU들의 변환 블록들의 역양자화 및 역변환으로부터 획득된 대응하는 잔차 값들에 가산함으로써, 현재의 CU 의 코딩 블록들을 복원할 수도 있다. 픽처의 각각의 CU 에 대해 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 그 픽처를 복원할 수도 있다.

[0064] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 본 개시물에서 설명되는 기법들에 따라서 동작하도록 구성될 수도 있다. 일반적으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 본원에서 설명되는 하나 이상의 기법들을 임의의 조합으로 수행하도록 구성될 수도 있다.

[0065] 본원에서 개시된 바와 같이, 본 개시물의 하나 이상의 기법들이 비디오 표준 (예컨대, AVC 및/또는 HEVC), 그의 확장판(들), 개발 중인 비디오 표준들, 또는 미래의 비디오 표준들 등과 함께 사용되거나 또는 아니면 이들을 향상시키기 때문에, AVC 및 HEVC 에 있어서의 블록 파티션 및 모션 예측에 관한 어떤 설계 양태들이 소개되거나 또는 아니면 본원에서 설명된다.

[0066] HEVC 에서의 모션 예측의 양태들이 이하 설명될 것이다. HEVC 표준에서는, PU 에 이용가능한 2개의 인터 예측 모드들: 병합 모드 (스킵은 병합의 특수한 경우로 간주된다) 및 진보된 모션 벡터 예측 (AMVP) 이 있다. AMVP 또는 병합 모드에서, 모션 벡터 (MV) 후보 리스트가 다수의 모션 벡터 예측자들에 대해 유지된다. 현재의 PU 의, 모션 벡터(들) 뿐만 아니라, 병합 모드에서의 참조 인덱스들은 MV 후보 리스트로부터 하나의 후보를 취함으로써 발생된다. MV 후보 리스트는 병합 모드에 대해 최대 5개의 후보들을 그리고 AMVP 모드에 대해 단지 2개의 후보들만을 포함한다. 병합 후보는 모션 정보의 세트, 예컨대, 참조 픽처 리스트들 및 참조 인덱스들 (리스트 0 및 리스트 1) 양쪽에 대응하는 모션 벡터들을 포함할 수도 있다. 병합 후보가 병합 인덱스에 의해 식별되면, 참조 픽처들이 현재의 블록들의 예측에 대해 사용될 뿐만 아니라, 연관된 모션 벡터들이 결정된다. 그러나, 리스트 0 또는 리스트 1로부터의 각각의 잠재적인 예측 방향에 대한 AMVP 모드 하에서, AMVP 후보가 단지 모션 벡터만을 포함하기 때문에, MV 후보 리스트로의 MVP 인덱스와 함께, 참조 인덱스가 명시적으로 시그널링될 필요가 있다. AMVP 모드에서, 예측된 모션 벡터들은 더 개선될 수 있다. 위에서 설명된 바와 같이, HEVC 에서의 병합 후보는 모션 정보의 풀 세트에 대응하며, 한편 HEVC 에서의 AMVP 후보는 특정의 예측 방향 및 참조 인덱스에 대해 단지 하나의 모션 벡터만을 포함한다. 병합 모드 및 AMVP 모드 양자에 대한 후보들이 동일한 공간 및 시간 이웃하는 블록들로부터 유사하게 유도된다.

[0067] 도 3a 및 도 3b 는 병합 모드 및 AMVP 모드에 대한 공간 이웃하는 MV 후보들을 각각 나타낸다. 공간 MV 후보들은 특정의 PU (나타낸 예에서는 PU_0) 에 대해 도 3a 및 도 3b 에 나타낸 이웃하는 블록들로부터 유도된다.

이하에서 좀더 자세하게 설명하는 바와 같이, 이웃하는 블록들로부터의 후보들의 발생은 병합 모드 및 AMVP 모드에 대해 상이하다.

[0068] 병합 모드에서, 최대 4개의 공간 MV 후보들이 도 3a 에 나타낸 순서, 즉, 좌측 (0), 상부 (1), 우상측 (2), 좌하측 (3), 및 좌상측 (4) 으로 유도될 수 있다. AMVP 모드에서, 이웃하는 블록들은 도 3b 에 나타낸 바와 같이, 2개의 그룹들: 블록들 0 및 1 로 이루어진 좌측 그룹, 및 블록들 2, 3, 및 4 로 이루어진 상부 그룹으로 분할된다. 각각의 그룹에 대해, 시그널링된 참조 인덱스에 의해 표시된 것과 동일한 참조 픽처를 참조하는 이웃하는 블록에서의 잠재적인 후보가 그룹의 최종 후보를 형성하도록 선택될 최고 우선순위 (priority) 를 갖는다. 모든 이웃하는 블록들이 동일한 참조 픽처를 가리키는 모션 벡터를 포함하지 않을 수 있다. 따라서, 이러한 후보가 발견될 수 없으면, 제 1 가용 후보가 최종 후보를 형성하기 위해 스케일링될 것이며, 따라서 일시적인 거리 차이들이 보상될 수 있다.

[0069] 시간 모션 벡터 예측자 (TMVP) 후보는, 이용가능하고 사용가능한 경우, MV 후보 리스트에 공간 모션 벡터 후보들 뒤에 추가된다. TMVP 후보에 대한 모션 벡터 유도의 프로세스는 병합 모드 및 AMVP 모드 양쪽에 대해 동일하다; 그러나, 병합 모드에서의 TMVP 후보에 대한 목표 참조 인덱스는 항상 0 으로 설정된다. TMVP 후보 유도를 위한 1차 블록 로케이션은 공간 이웃하는 후보들을 발생시키는데 사용되는 상부 및 좌측 블록들에 대한 바이어스를 보상하기 위해 (도 4a 에서 블록 "T" 로 나타낸) 병치된 PU 의 외부에 있는 우하측 블록이다. 그러나, 그 블록이 현재의 CTB 로우의 외부에 로케이트되거나 또는 모션 정보가 이용불가능하면, 블록은 PU 의 중심 블록으로 치환된다.

[0070] 도 4b 에 나타낸 바와 같이, TMVP 후보에 대한 모션 벡터는 슬라이스 레벨에 표시된, 병치된 픽처의 병치된 PU 로부터 유도된다. AVC 에서의 시간 직접 모드와 유사하게, TMVP 후보의 모션 벡터는 거리 차이들을 보상하도록 수행되는 모션 벡터 스케일링을 가질 수도 있다.

[0071] HEVC 에서의 모션 예측의 다른 양태들이 이하 설명될 것이다. 하나의 이러한 양태는 모션 벡터 스케일링이다. 모션 벡터들의 값은 그 프리젠테이션 시간에서의 픽처들의 거리에 비례한다고 가정된다. 모션 벡터는 2개의 픽처들, 즉, 참조 픽처와, 모션 벡터를 포함하는 픽처 (즉, 포함하는 픽처) 를 연관시킨다. 모션 벡터가 다른 모션 벡터를 예측하는데 이용될 때, 포함하는 픽처와 참조 픽처의 거리가 픽처 순서 카운트 (POC) 값들에 기초하여 계산된다.

[0072] 예측되는 모션 벡터에 대해, 그의 연관된 포함하는 픽처 및 참조 픽처 양자는 상이할 수도 있다. 따라서, (POC 에 기초한) 새로운 거리가 계산된다. 그리고, 모션 벡터가 이들 2개의 POC 거리들에 기초하여 스케일링된다. 공간 이웃하는 후보에 대해, 2개의 모션 벡터들에 대한 포함하는 픽처들은 동일한 반면, 참조 픽처들은 상이하다. HEVC 에서, 모션 벡터 스케일링이 공간 및 시간 이웃하는 후보들에 대해 TMVP 및 AMVP 양쪽에 적용된다.

[0073] HEVC 에서의 모션 예측의 다른 양태는 인공적인 모션 벡터 후보 발생이다. 모션 벡터 후보 리스트가 완료되지 않으면, 인공적인 모션 벡터 후보들이 발생되어 모든 후보들을 가질 때까지 리스트의 끝에 삽입된다. 병합 모드에서는, 2개의 유형들의 인공적인 MV 후보들: 오직 B-슬라이스들에 대해서만 유도되는 결합된 후보, 및 제 1 유형이 충분한 인공적인 후보들을 제공하지 않으면 오직 AMVP 에 대해서만 사용되는 제로 후보들이 존재한다. 후보 리스트에 이미 있고 필요한 모션 정보를 가지는 후보들의 각각의 쌍에 대해, 양방향 결합된 모션 벡터 후보들이 리스트 0 에서의 픽처를 참조하는 제 1 후보의 모션 벡터와 리스트 1 에서의 픽처를 참조하는 제 2 후보의 모션 벡터의 조합에 의해 유도된다.

[0074] HEVC 에서의 모션 예측의 다른 양태는 후보 삽입을 위한 프루닝 프로세스이다. 상이한 블록들로부터의 후보는 동일한 경우가 발생하여, 병합 모드 또는 AMVP 모드 후보 리스트의 효율을 감소시킬 수도 있다. 프루닝 프로세스는 동일한 후보들을 삽입하는 것을 피하기 위해 하나의 후보를 현재의 후보 리스트에서의 나머지 후보들과 비교한다.

[0075] 중첩 블록 모션 보상 (OBMC) 의 일반적인 컨셉이 이하 소개된다. OBMC 는 이전의 비디오 표준들에서 고려되었다. 중첩 블록 모션 보상 (OBMC) 은 H.263 의 개발 (development) 에서 제안되었다 (1995년 4월, 문서 권고안 H.263, ITU-T, Video Coding for Low Bitrate Communication). OBMC 는 현재의 16x16 매크로블록의 현재의 8x8 블록에 대해 수행될 수도 있으며, 2개의 연결된 이웃하는 8x8 블록들의 모션 벡터들이 도 5 에 나타낸 바와 같이 현재의 8x8 블록에 대해 사용될 수도 있다. 예를 들어, 현재의 매크로블록에서의 제 1 의 8x8 블록 (예컨대, 도 5 에서 숫자 "1" 로 식별된 블록) 에 대해, 그의 자신의 모션 벡터를 이용하여 발생된 예측

블록 이외에도, 상부 및 좌측 이웃하는 모션 벡터가 또한 2개의 추가적인 예측 블록들을 발생시키기 위해 적용된다. 이러한 방법으로, 현재의 8x8 블록에서의 각각의 픽셀은 3개의 예측 값들을 가지며, 이들 3개의 예측 값들의 가중 평균이 현재의 8x8 블록에서의 각각의 픽셀에 대한 최종 예측 값으로서 사용된다. 다른 예로서, 현재의 매크로블록에서의 제 2 의 8x8 블록 (예컨대, 도 5 에서 숫자 "2" 로 식별된 블록) 에 대해, 그의 자신의 모션 벡터를 이용하여 발생된 예측 블록 이외에도, 상부 및 우측 이웃하는 모션 벡터가 또한 2개의 추가적인 예측 블록들을 발생시키기 위해 적용될 수도 있다. 이러한 방법으로, 현재의 8x8 블록에서의 각각의 픽셀은 3개의 예측 값들을 가지며, 이들 3개의 예측의 가중 평균이 현재의 8x8 블록에서의 각각의 픽셀에 대한 최종 예측 값으로서 사용된다.

[0076] 도 5 는 OBMC 가 H.263 에서 어떻게 구현되었는지의 일 예를 나타낸다. 이웃하는 블록이 코딩되지 않거나 또는 인트라로서 코딩되면 (즉, 이웃하는 블록이 가용 모션 벡터를 가지지 않으면), 현재의 8x8 블록의 모션 벡터가 이웃하는 모션 벡터로서 사용된다. 다음으로, 현재의 매크로블록의 제 3 및 제 4 의 8x8 블록들 (예컨대, 도 5 에서 숫자 "3" 및 숫자 "4" 로 식별된 블록들) 을 참조하면, 하부 이웃하는 블록이 항상 사용되지 않는다 (즉, 가용하지 않은 것으로 간주된다). 다시 말해서, 현재의 매크로블록에 대해, 어떤 현재의 매크로블록 하부의 매크로블록들로부터의 모션 정보도 OBMC 를 이용하여 현재의 매크로블록의 픽셀들을 복원하는데 사용되지 않을 것이다. 대신, 블록 3 에 대해, 좌측 이웃으로부터의 모션 정보가 OBMC 에 사용될 수도 있으며, 블록 4 에 대해서는, 우측 이웃으로부터의 모션 정보가 OBMC 에 대해 사용될 수도 있으나, 블록들 3 및 4 하부의 블록들은 코딩되지 않는 것으로 가정되며 OBMC 에 사용되지 않는다.

[0077] OBMC 는 HEVC 에 대해 제안되었다. HEVC 에서, OBMC 는 또한 PU 경계를 평활화하기 위해 제안되었다. HEVC 에서의 OBMC 의 제안된 방법의 일 예가 도 6a 및 도 6b 에 도시된다. 도 6a 및 도 6b 에서, PU-기반의 OBMC 는 제 1 PU (예컨대, PU0) 를 나타내기 위해서는 백색 영역을, 그리고 제 2 PU (예컨대, PU1) 를 나타내기 위해서 해칭 영역을 이용하여 예시된다. HEVC 에 대해 제안된 바와 같이, CU 가 2개 이상의 PU들을 포함할 때, PU 경계 근처의 로우들 또는 칼럼들은 OBMC 에 의해 평활화된다. 제 1 PU 및 제 2 PU 에서 "A" 또는 "B" 로 표시된 픽셀들에 대해, 2개의 예측 값들은 제 1 PU (PU0) 및 제 2 PU (PU1) 의 모션 벡터들을 각각 적용함으로써 발생된다. HEVC 에 대해 제안된 바와 같이, 2개의 예측 값들의 가중 평균이 최종 예측값으로서 사용될 것이다.

[0078] HEVC 에서, 각각의 PU 는 모션 정보의 단지 하나의 세트를 포함한다. 그러나, 서브-PU 기반의 모션 예측/유도가 적용될 때, PU 는 서브-PU들로 분할될 수도 있으며, 여기서, 각각의 서브-PU 는 모션 정보의 하나의 세트를 포함한다. PU 가 다수의 서브-PU들로 분할될 때, 각각의 서브-PU 는 별개로 모션 보상될 수 있다. 예를 들어, 서브-PU 프로세스가 시간 모션 벡터 예측에 적용가능할 수도 있으며 이는 진보된 시간 모션 벡터 예측 (ATMVP) 으로 불린다.

[0079] 서브-블록들의 양태들이 이하 설명될 것이다. 디블록킹과 같은, 사후 필터링 프로세스들에서, 디코딩 프로세스들은 PU 의 구조를 전적으로 따르지 않고 프로세싱될 최소 블록들의 사이즈들을 따르는 방법으로 적용될 수 있다. 이러한 최소 프로세싱 블록들은 서브-블록들로 불린다. 서브-블록은 정사각형 또는 비-정사각형 형태일 수 있다. 서브-블록은 전형적으로 PU 에 속하며, 따라서 서브-블록 사이즈는 가장 작은 PU 의 사이즈보다 작거나 또는 동일하다.

[0080] 일 예가 서브-PU 와 서브-블록 사이의 관계를 나타내기 위해 도 7a 내지 도 7c 에 예시된다. 볼 수 있는 바와 같이, 서브-블록 사이즈는 서브-PU 사이즈보다 작거나 (도 7a), 동일하거나 (도 7b) 또는 더 클 수 있다 (도 7c).

[0081] 이전에 제안된 및 현재의 기법들은 일부 한계들을 가질 수도 있다. 일 예로서, HEVC 에 대한 기존 OBMC 방법들은 PU, 매크로블록, 또는 서브-매크로블록 경계에 대해 주로 수행된다. 다른 예로서, OBMC 로 현재의 PU 를 예측하기 위해 다른 CU들의 모션 정보를 이용하는 방법이 명확하지 않다. 다른 예로서, 현재의 PU 가 서브-PU들을 포함할 때, PU-기반의 OBMC 방법들은 효율적으로 작동하지 않을 수도 있다. 위에서 식별된 문제들 및 한계들을 잠재적으로 해결하기 위해, 본 개시물은 여러 기법들을 도입한다. 본 개시물의 기법들은 개별적으로 적용될 수도 있거나, 또는 대안적으로는, 이들의 임의의 조합이 적용될 수도 있다. 본원에서 사용될 때, 참조 정보는 본 개시물에서 모션 정보의 부분으로서 간주될 수도 있다.

[0082] 아래에서 설명되는 기법들이 중첩 블록 모션 보상 (OBMC) 에 사용된다. 일부 예들에서, 본원에서 설명된 바와 같이, OBMC 는 현재의 블록에 이웃하는 하나 이상의 블록들로부터의 모션 정보의 사용을 수반하며 현재의 블록의 예측 블록을 발생시키기 위해서 가중 평균을 수행하는 임의의 프로세스를 지칭할 수도 있다. 이러한

예들에서, OBMC 는 경계, 블록의 부분, 블록의 슬라이스, 또는 전체 블록에 대해 수행될 수도 있다. 다른 예들에서, 본원에서 설명된 바와 같이, OBMC 는 (i) 현재의 블록에 이웃하는 하나 이상의 블록들로부터의 모션 정보에 기초한 예측 블록(들), 및 (ii) 현재의 블록의 모션 정보에 기초한 예측 블록의 가중 평균의 사용을 수반하는 임의의 프로세스를 지칭할 수도 있다. 이러한 예들에서, OBMC 는 경계, 블록의 부분, 블록의 슬라이스, 또는 전체 블록에 대해 수행될 수도 있다. 다른 예들에서, 본원에서 설명된 바와 같이, OBMC 는 현재의 블록 및 하나 이상의 이웃하는 블록들로부터 다수의 모션 벡터들을 발생시키고 그들을 현재의 블록에 대해 결합하는 (예컨대, 평균화하는) 것을 지칭할 수도 있다.

[0083] 본 개시물의 제 1 기법은 서브-블록 기반의 OBMC 이다. 이 제 1 기법 하에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 PU 또는 CU (예컨대, 현재의 PU 또는 CU) 내 각각의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 PU 경계를 평활화하기 위해 OBMC 를 단지 수행하도록 구성되는 대신, PU 또는 CU 내 각각의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다. 위에서 설명한 바와 같이, 일부 예들에서, 서브-블록은 PU 의 서브-블록을 지칭할 수도 있으며; 그리고, 이러한 예들에서, 서브-블록 사이즈는 가장 작은 PU 의 사이즈보다 작거나 또는 동일할 수도 있다. 다른 예들에서, 서브-블록은 PU 의 서브-블록과는 상이한 어떤 것을 지칭할 수도 있다. 예를 들어, 서브-블록은 CU 의 서브-블록을 지칭할 수도 있다. 서브-블록들의 사이즈가 아래에서 좀더 자세하게 설명된다. 이 제 1 기법에 따르면, 그리고, 일부 예들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 인코딩되거나 및/또는 디코딩 중인 현재의 서브-블록에 이웃하는 서브-블록들의 모션 정보를 이용함으로써 OBMC 를 수행하도록 구성될 수도 있다.

[0084] 이 제 1 기법 및 다른 예들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 이웃하는 하나 이상의 서브-블록들이 디코딩되지 않은 CU 에 속한다고 결정하도록 구성될 수도 있다. 일부 예들에서, "디코딩되지 않은" CU 는 현재의 서브-블록이 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 디코딩되고 있을 때 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 CU 를 지칭할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 비디오 디코더 (30) 에 의해 수신된 CU 는 "디코딩되지 않은" CU 로서 지칭될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 스캔 순서에 따라서 복수의 블록들을 디코딩하도록 구성될 수도 있으며, 디코딩되지 않은 CU 는 디코딩되지 않은 CU 가 현재의 서브-블록 이후 언젠가 디코딩되도록 그 스캔 순서에 배치될 수도 있다. 하나 이상의 이웃하는 서브-블록들이 디코딩되지 않은 CU 에 속한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 디코딩되지 않은 CU 에 속하는 하나 이상의 이웃하는 서브-블록들이 서브-블록 기반의 OBMC 에 이용불가능하다고 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용불가능한 것으로 결정된 하나 이상의 이웃하는 서브-블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0085] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 이웃하는 하나 이상의 서브-블록들이 디코딩되지 않은 LCU 에 속한다고 결정하도록 구성될 수도 있다. 일부 예들에서, "디코딩되지 않은" LCU 는 현재의 서브-블록이 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 디코딩되고 있을 때 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 LCU 를 지칭할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 비디오 디코더 (30) 에 의해 수신된 LCU 는 "디코딩되지 않은" LCU 로서 지칭될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 스캔 순서에 따라서 복수의 블록들을 디코딩하도록 구성될 수도 있으며, 디코딩되지 않은 LCU 는 디코딩되지 않은 LCU 가 현재의 서브-블록 이후 언젠가 디코딩되도록 그 스캔 순서에 배치될 수도 있다. 하나 이상의 이웃하는 서브-블록들이 디코딩되지 않은 LCU 에 속한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 디코딩되지 않은 LCU 에 속하는 하나 이상의 이웃하는 서브-블록들이 서브-블록 기반의 OBMC 에 이용불가능하다고 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용불가능한 것으로 결정된 하나 이상의 이웃하는 서브-블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0086] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 이웃하는 하나 이상의 서브-블록들이 디코딩되지 않은 PU 에 속한다고 결정하도록 구성될 수도 있다. 일부 예들에서, "디코딩되지 않은" PU 는 현재의 서브-블록이 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 디코딩되고 있을 때 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 PU 를 지칭할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 비디오 디코더 (30) 에 의해 수신된 PU 는

"디코딩되지 않은" PU 로서 지칭될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 스캔 순서에 따라서 복수의 블록들을 디코딩하도록 구성될 수도 있으며, 디코딩되지 않은 PU 는 디코딩되지 않은 PU 가 현재의 서브-블록 이후 언젠가 디코딩되도록 그 스캔 순서에 배치될 수도 있다. 하나 이상의 이웃하는 서브-블록들이 디코딩되지 않은 PU 에 속한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 디코딩되지 않은 PU 에 속하는 하나 이상의 이웃하는 서브-블록들이 서브-블록 기반의 OBMC 에 이용불가능하다고 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용불가능한 것으로 결정된 하나 이상의 이웃하는 서브-블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0087]

다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 이웃하는 하나 이상의 서브-블록들이 디코딩되지 않은 서브-블록에 속한다고 결정하도록 구성될 수도 있다. 일부 예들에서, "디코딩되지 않은" 서브-블록은 현재의 서브-블록이 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 디코딩되고 있을 때 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 서브-블록을 지칭할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 에 의해 아직 디코딩되지 않은 비디오 디코더 (30) 에 의해 수신된 서브-블록은 "디코딩되지 않은" 서브-블록으로서 지칭될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 스캔 순서에 따라서 복수의 블록들을 디코딩하도록 구성될 수도 있으며, 디코딩되지 않은 서브-블록은 디코딩되지 않은 서브-블록이 현재의 서브-블록 이후 언젠가 디코딩되도록 그 스캔 순서에 배치될 수도 있다. 하나 이상의 이웃하는 서브-블록들이 디코딩되지 않은 서브-블록에 속한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 디코딩되지 않은 서브-블록에 속하는 하나 이상의 이웃하는 서브-블록들이 서브-블록 기반의 OBMC 에 이용불가능하다고 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용불가능한 것으로 결정된 하나 이상의 이웃하는 서브-블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0088]

이 제 1 기법에 따르면, 서브-블록들의 사이즈는 CU 에 대한 가장 작은 PU 사이즈와 동일할 수도 있다. 다른 예들에서, 서브-블록은 가장 작은 PU 사이즈보다 더 작은 사이즈를 가질 수도 있다. 예를 들어, NxN 은 가장 작은 모션 블록 사이즈를 나타낼 수도 있으며 서브-블록 사이즈 (예컨대, 서브-PU 사이즈) 는 NxN 과 동일할 수도 있으며, 여기서, N 은 양의 정수이다. 이러한 예에서, 가장 작은 PU 사이즈가 8x4 또는 4x8 이면, 서브-블록 사이즈는 4x4 와 동일할 수도 있다. 다른 예에서, NxM 은 가장 작은 모션 블록 사이즈를 나타낼 수도 있으며, 서브-블록 사이즈 (예컨대, 서브-PU 사이즈) 는 NxM 으로 설정될 수도 있으며, 여기서, M 및 N 은 양의 정수들이며 서로 동일하거나 또는 동일하지 않을 수도 있다. 이러한 예에서, 가장 작은 PU 사이즈가 8x8 이면, 서브-블록 사이즈는 4x4, 8x4, 또는 4x8 과 동일할 수도 있다.

[0089]

다른 예에서, 서브-블록의 사이즈는 또한 임의의 사전-정의된 값일 수도 있다. 사전-정의된 서브-블록 사이즈는 예를 들어, 가장 작은 PU 사이즈보다 클 수도 있다. 다른 예에서, 서브-블록들의 사이즈는 파라미터 세트에 시그널링될 수도 있다. 여전히, 다른 예에서, 서브-블록의 사이즈는 현재의 PU 또는 CU 사이즈 (즉, 현재의 서브-블록이 속하는 현재의 PU 또는 CU) 에 의존할 수도 있다. 여전히, 다른 예에서, 서브-블록의 사이즈는 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 OBMC 를 수행하도록 구성되는 블록 사이즈에 의존할 수도 있다. 여전히, 다른 예에서, 서브-블록의 사이즈는 현재의 PU 또는 CU 사이즈 (즉, 현재의 서브-블록이 속하는 현재의 PU 또는 CU) 뿐만 아니라 하나 이상의 이웃하는 PU들 또는 CU들 (즉, 현재의 서브-블록이 속하는 PU 또는 CU 에 이웃하는 하나 이상의 PU들 또는 CU들) 의 사이즈에 의존할 수도 있다.

[0090]

본 개시물의 제 2 기법은 서브-PU 기반의 OBMC 이다. 이 제 2 기법 하에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 서브-PU 와 코딩되는 PU 또는 CU (예컨대, 현재의 PU 또는 CU) 내 각각의 서브-PU 에 대해 OBMC 를 수행하도록 구성될 수도 있다. 각각의 서브-PU 는 예컨대, 모션 벡터 및 참조 픽처 리스트 인덱스를 포함한, 모션 정보의 그 자신의 세트를 포함할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 PU 또는 CU 내 하나 이상의 이웃하는 서브-PU들의 모션 정보를 이용하여 현재의 PU 또는 CU 내 하나 이상의 서브-PU들에 대해 OBMC 를 수행하도록 구성될 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 PU 또는 CU 의 외부에 있는 하나 이상의 이웃하는 블록들의 모션 정보와 함께, 현재의 PU 또는 CU 내 하나 이상의 이웃하는 서브-PU들의 모션 정보를 이용하여, 현재의 PU 또는 CU 내 하나 이상의 서브-PU들에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0091]

이 제 2 기법 및 일부 예들에 따르면, 각각의 서브-PU 에 대해, 그의 이웃하는 블록들은 동일한 PU 또는 CU (예컨대, 현재의 PU 또는 CU) 내 이웃하는 서브-PU들, 및 현재의 PU 또는 CU (즉, 현재의 PU 또는 현재의 CU) 의

외부에 있는 이웃하는 블록들일 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 블록이 현재의 서브-PU 와 동일한 사이즈를 가지면 현재의 PU 또는 CU 의 외부에 있는 (즉, 현재의 PU 의 외부에 있거나 또는 현재의 CU 의 외부에 있는) 이웃하는 블록을 이용하여 현재의 PU 또는 CU 의 현재의 서브-PU 에 대해 OBMC 를 수행하도록 구성될 수도 있다. 이러한 예들에서, 현재의 PU 또는 CU 에 이웃하는 블록이 모션 정보의 하나 보다 많은 세트를 포함하더라도, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 그 이웃하는 블록에 대한 모션 정보의 오직 하나의 세트를 이용하도록 구성될 수도 있다. 또한, 이러한 예들에서, 현재의 PU 또는 CU 의 외부에 있는 이웃하는 블록이 현재의 서브-PU 의 사이즈와는 상이한 사이즈를 가지면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-PU 의 사이즈와는 상이한 사이즈를 갖는 것으로 결정된 하나 이상의 이웃하는 블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-PU 에 대한 OBMC 를 수행하도록 구성될 수도 있다.

[0092] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 블록이 현재의 서브-PU 의 사이즈보다 작으면 현재의 PU 또는 CU 의 외부에 있는 이웃하는 블록을 이용하여 현재의 PU 또는 CU 의 현재의 서브-PU 에 대해 OBMC 를 수행하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-PU 에 대한 하나 이상의 서브-블록들을 이용함으로써 현재의 서브-PU 에 대한 OBMC 를 수행하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 PU 의 임의의 경계 서브-PU들을 서브-블록들로 분할하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 각각의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0093] 다른 예들에서, 현재의 PU 또는 CU 의 외부에 있는 임의의 픽셀에 인접하지 않은 서브-PU 는 단지 이웃하는 서브-PU들만을 이용하여 예측될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-PU 가 현재의 PU 또는 CU 의 외부에 있는 임의의 픽셀에 인접하지 않는다고 결정하도록 구성될 수도 있으며, 단지 이웃하는 서브-PU들만을 이용하여 현재의 서브-PU 에 대한 OBMC 를 수행하도록 구성될 수도 있다.

[0094] 또한, 본 개시물의 제 2 기법에 따르면, 그리고 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-PU 에의 하나 이상의 이웃하는 블록들이 디코딩되지 않은 CU 에 속한다고 결정하도록 구성될 수도 있다. 하나 이상의 이웃하는 블록들이 디코딩되지 않은 CU 에 속한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 디코딩되지 않은 CU 에 속하는 하나 이상의 이웃하는 블록들이 서브-PU 기반의 OBMC 에 이용불가능하다고 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용불가능한 것으로 결정된 하나 이상의 이웃하는 블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-PU 에 대해 OBMC 를 수행할 수도 있다.

[0095] 또한, 본 개시물의 제 2 기법에 따르면, 그리고 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-PU 에의 하나 이상의 이웃하는 블록들이 디코딩되지 않은 LCU 에 속한다고 결정하도록 구성될 수도 있다. 하나 이상의 이웃하는 블록들이 디코딩되지 않은 LCU 에 속한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 디코딩되지 않은 LCU 에 속하는 하나 이상의 이웃하는 블록들이 서브-PU 기반의 OBMC 에 이용불가능하다고 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용불가능한 것으로 결정된 하나 이상의 이웃하는 블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-PU 에 대해 OBMC 를 수행할 수도 있다.

[0096] 또한, 본 개시물의 제 2 기법에 따르면, 그리고 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-PU 에의 하나 이상의 이웃하는 블록들이 디코딩되지 않은 PU (예컨대, 인코딩된 PU) 에 속한다고 결정하도록 구성될 수도 있다. 하나 이상의 이웃하는 블록들이 디코딩되지 않은 PU 에 속한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 디코딩되지 않은 PU 에 속하는 하나 이상의 이웃하는 블록들이 서브-PU 기반의 OBMC 에 이용불가능하다고 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용불가능한 것으로 결정된 하나 이상의 이웃하는 블록들의 모션 정보를 이용하지 않음으로써 (예컨대, 배제함으로써) 현재의 서브-PU 에 대해 OBMC 를 수행할 수도 있다.

[0097] 본 개시물의 제 3 기법은 CU-경계 기반의 OBMC 이다. 이 제 3 기법 하에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 CU 경계들에 대한 OBMC 를 수행하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 또한 하나 이상의 PU 경계들에 대한 OBMC 를 수행하도

록 구성될 수도 있다.

[0098] 현재의 CU 의 상부 및 좌측 경계들을 참조하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, 현재의 CU 에 속하지 않지만 현재의 CU 의 상부 및/또는 좌측 경계 (예컨대, 에지) 와 경계를 이루는 하나 이상의 이웃하는 블록들의 모션 정보를 이용하여, 제 1 기법에 대해 본원에서 설명한 방법과 유사한 방법으로 추가적인 예측자들을 발생시키도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 CU 의 좌측에 이웃하는 블록(들) (예컨대, 이웃하는 CU) 및/또는 현재의 CU 의 상부에 이웃하는 블록(들) (예컨대, 이웃하는 CU) 로부터의 CU 경계 (예컨대, 현재의 CU 의 상부 및/또는 좌측 경계) 상의 블록들의 모션 정보의 모두 또는 일부를 이용함으로써, CU 경계 (예컨대, 현재의 CU 경계) 에 대한 OBMC 를 수행하도록 구성될 수도 있다.

[0099] 이러한 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 다음과 같이 CU 경계에 대한 OBMC 를 수행하도록 구성될 수도 있다: 상부 CU 경계 상에 로케이트된 현재의 CU 의 각각의 가장 작은 블록에 대해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 상부 CU 경계의 OBMC 를 위해 각각의 가장 작은 블록의 상부 이웃하는 블록의 모션 정보를 이용하도록 구성될 수도 있으며, 추가적인 예측자(들) 이 그 CU 를 가진 현재의 블록의 최종 예측 신호를 형성하기 위해 이용된다. 좌측 CU 경계 상에 로케이트된 현재의 CU 의 각각의 가장 작은 블록에 대해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 좌측 CU 경계의 OBMC 를 위해 각각의 가장 작은 블록의 좌측 이웃하는 블록의 모션 정보를 이용하도록 구성될 수도 있으며, 추가적인 예측자(들) 이 현재의 블록의 최종 예측 신호를 형성하기 위해 이용된다.

[0100] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 단지 좌측 CU 및/또는 상부 CU 경계들에 대해서만 OBMC 를 수행하도록 구성될 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 좌측 CU, 상부 CU, 우측 CU, 및/또는 저부 CU 경계들에 대해 OBMC 를 수행하도록 구성될 수도 있다.

이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 좌측 및 상부 CU 경계들에 대해 위에서 설명한 방법과 유사한 방법으로 우측 CU 및/또는 저부 CU 경계들에 대해 OBMC 를 수행하도록 구성될 수도 있다.

예를 들어, 우측 CU 경계 상에 로케이트된 현재의 CU 의 각각의 가장 작은 블록에 대해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 우측 CU 경계의 OBMC 를 위해 각각의 가장 작은 블록의 상부 이웃하는 블록의 모션 정보를 이용하도록 구성될 수도 있으며, 추가적인 예측자(들) 이 그 CU 를 가진 현재의 블록의 최종 예측 신호를 형성하기 위해 사용된다. 저부 CU 경계 상에 로케이트된 현재의 CU 의 각각의 가장 작은 블록에 대해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 저부 CU 경계의 OBMC 를 위해 각각의 가장 작은 블록의 저부 이웃하는 블록의 모션 정보를 이용하도록 구성될 수도 있으며, 추가적인 예측자(들) 이 그 CU 를 가진 현재의 블록의 최종 예측 신호를 형성하기 위해 사용된다.

[0101] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 CU 로부터의 모션 정보를 서브샘플링하거나 또는 압축하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 CU 의 서브샘플링된 또는 압축된 모션 정보를 이용하여 현재의 CU 의 하나 이상의 추가적인 예측자들을 발생시키도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 가장 작은 블록 사이즈 대신 서브-블록에 기초하여 이웃하는 CU 의 모션 정보를 서브샘플링하거나 또는 압축하도록 구성될 수도 있으며, 이웃하는 CU 가 개개의 MV들을 갖는 다수의 서브-블록들 (예컨대, PU들) 을 가지면 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 PU들의 하나의 그룹 (예컨대, 세트) 에 대해서는 단일 MV 를, 그리고 PU들의 다른 그룹에 대해서는 다른 단일 MV 를 이용하도록 구성될 수도 있다는 것을 의미한다.

[0102] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 본 개시물의 제 3 기법 및/또는 본 개시물의 다른 기법과 관련한 OBMC 를 디스에이블하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 CU 의 사이즈가 서브-블록 사이즈의 사이즈보다 작으면 (예컨대, 작을 때) 현재의 CU 내 PU들에 대해 OBMC 를 디스에이블하도록 구성될 수도 있다. 이러한 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 CU 의 사이즈를 결정하고 현재의 CU 의 결정된 사이즈를 서브-블록의 사이즈와 비교하여 현재의 CU 의 사이즈가 서브-블록의 사이즈보다 작은지 여부를 결정하도록 구성될 수도 있다. 현재의 CU 의 사이즈가 서브-블록 사이즈보다 작다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 CU 내 PU들에 대해 OBMC 를 디스에이블하도록 구성될 수도 있다.

[0103] 다른 예로서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 PU 경계가 서브-블록 경계 (예컨대, 임의의 서브-블록들을 포함하지 않는 경계) 가 아니면 (예컨대, 아닐 때) PU 경계에 대해 OBMC 를 디스에이블하도록 구성될 수도 있다. 이 예에서, 서브-블록은 PU 보다 크며 PU 경계는 서브-블록 내에 로케이트될 수도 있다.

PU 경계는 서브-블록 경계이거나 또는 아닐 수도 있다. 따라서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 인코딩 및/또는 디코딩 복잡성을 감소시키기 위해 그 PU 경계를 따른 OBMC 를 디스에이블하도록 (예컨대, 스킵하도록) 구성될 수도 있다. 다른 예로서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 우측 CU 및/또는 저부 CU 경계들에 대해 OBMC 를 디스에이블하도록 구성될 수도 있다.

[0104] 본 개시물의 제 4 기법은 상기 3개의 설명된 제 1, 제 2, 및 제 3 OBMC 기법들에 공통인 양태들을 포함한다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 서브-블록, 이웃하는 서브-PU, 또는 이웃하는 블록의 모션 벡터를 이용하여, 현재의 서브-블록, 현재의 서브-PU, 또는 현재의 블록의 추가적인 예측자를 각각 발생시키도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 추가적인 예측자를 이용하여, 현재의 서브-블록, 현재의 서브-PU, 또는 현재의 블록의 최종 예측 신호를 각각 형성하도록 구성될 수도 있다.

[0105] 이 제 4 기법 및 다른 예들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 OBMC 를 위해 현재의 서브-블록, 현재의 서브-PU, 또는 현재의 블록의 이웃들을 미리 정의된 개수까지 이용하도록 구성될 수도 있다. 이러한 예들에서, 미리 정의된 개수는 8개 또는 8개와는 상이한 개수일 수도 있다. 일 예에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 OBMC 를 위해 4개의 연결된 이웃 블록들 (예컨대, 상부, 하부, 좌측, 및 우측 이웃하는 서브-블록들) 의 모션 정보 (예컨대, 모션 벡터들) 를 이용하도록 구성될 수도 있다. 본원에서 사용될 때, "OBMC 를 위해" 를 지칭하는 것은 "OBMC 를 수행하기 위해" 를 지칭할 수도 있다.

[0106] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 서브-블록, 이웃하는 서브-PU, 또는 이웃하는 블록이 모션 정보의 다수의 세트들을 포함하면 (예컨대, 포함할 때) 모션 정보의 오직 하나의 세트만을 선택하도록 구성될 수도 있다. 예를 들어, 서브-블록 사이즈가 8x8 이고 모션 정보가 4x4 방식으로 저장되면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이러한 특징의 예에서 8x8 서브-블록에 대응하는 모션 정보의 하나의 세트를 단지 선택하도록 구성될 수도 있다.

[0107] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 기법들에 따라서 모션 정보의 단일 세트를 선택하도록 구성될 수도 있다. 일 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 서브-블록, 서브-PU, 또는 블록의 중심 픽셀에 기초하여 모션 정보의 세트를 선택하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 서브-블록, 서브-PU, 또는 블록의 중심 픽셀을 포함하는 서브-블록, 서브-PU, 또는 블록에 대한 모션 정보의 세트를 선택하도록 구성될 수도 있다. 그러므로, PU에 대한 모션 정보의 다수의 세트들이 존재하면, 중심 픽셀을 포함하는 블록에 대한 모션 정보의 단일 세트가 현재의 블록에 대한 OBMC 를 위해 사용될 수도 있다. 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 어느 모션 정보의 세트가 현재의 서브-블록, 서브-PU, 또는 블록에 인접한 가장 작은 모션 블록들에 속하는지에 기초하여 모션 정보의 세트를 선택하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록, 서브-PU, 또는 블록에 인접한 가장 작은 모션 블록들에 속하는 모션 정보의 세트를 선택하도록 구성될 수도 있다. 복수의 이러한 모션 블록들이 존재하면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 예를 들어, 래스터 스캔 순서에서 복수의 모션 블록 중 가장 빠른 (또는, 최초에-디코딩된) 모션 블록을 선택하도록 구성될 수도 있다.

[0108] 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 서브-블록들의 모든 모션 정보를 현재의 PU, CU, 또는 블록의 모션 정보와 비교하도록 구성될 수도 있다. 이 비교에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 PU, CU, 또는 블록의 모션 정보와 가장 유사한 모션 정보를 선택하도록 구성될 수도 있다. 이 비교에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 PU, CU, 또는 블록의 모션 정보와 가장 유사하지 않은 모션 정보를 선택하도록 구성될 수도 있다.

[0109] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 OBMC 를 위해 임의의 연결된 이웃하는 서브-블록들, 서브-PU들, 또는 블록들의 서브세트의 모션 정보를 이용하도록 구성될 수도 있다. 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 단지 현재의 블록의 좌측 및 우측 이웃들만을 이용하여 OBMC 를 수행하도록 구성될 수도 있다. 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록의 단지 상부 및 하부 이웃들을 이용하여 OBMC 를 수행하도록 구성될 수도 있다. 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록의 상부 또는 하부 이웃들 중 단지 하나 및 좌측 또는 우측 이웃들 중 하나를 이용하여 OBMC 를 수행하도록 구성될 수도 있다. 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록의 4개의 연결된 이웃들 중 단지 하나를 이용하여 OBMC 를 수행하도록 구

성될 수도 있다. 여전히, 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 PU 또는 CU 내 서브-블록의 상대적인 위치에 기초하여 이웃하는 블록들의 서브세트를 선택하도록 구성될 수도 있다.

[0110] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 서브-블록, 서브-PU, 또는 블록이 본원에서 설명된 하나 이상의 기법들에 따라서 이용불가능하다고 결정하도록 구성될 수도 있다. 예를 들어, 인트라 블록은 이용불가능한 것으로 간주될 수도 있거나, 현재의 슬라이스 또는 타일의 외부에 있는 (또는, 부분적으로 외부에 있는) 블록은 이용불가능한 것으로 간주될 수도 있거나, 파두 (wavefront) 가 사용될 때 이용불가능한 것으로 간주되는 블록은 이용불가능한 것으로 간주될 수도 있거나, 아직 디코딩되지 않은 (예컨대, 적어도 그의 모션 벡터들이 아직 디코딩되지 않은) 블록은 이용불가능한 것으로 간주될 수도 있거나, 및/또는 이웃하는 블록 또는 서브-블록은 이웃하는 블록 또는 서브-블록이 현재의 CTU 에 속하지 않을 때 이용불가능한 것으로 간주될 수도 있다.

[0111] 본 개시물의 제 5 기법에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 각각의 가용 이웃하는 서브-블록의 모션 정보 (예컨대, 모션 벡터들) 을 이용하여 현재의 서브-블록에 대해 OBMC 를 수행하여 예측 블록을 발생시키도록 구성될 수도 있다. 일부 예들에서, 예측 블록은 P_k 에 의해 표시될 수도 있으며, 여기서 N 은 현재의 서브-블록에 이웃하는 이웃하는 블록에 대한 (예컨대, 상부, 하부, 좌측 또는 우측과 동일한) 인덱스/표시를 나타낸다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 이웃하는 블록들의 예측 블록(들) 을 이용하여, P_k 에 의해 표시될 수도 있는, 현재의 서브-블록의 예측 블록을 업데이트하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록의 최종 예측 블록 (즉, 현재의 서브-블록의 업데이트된 예측 블록) 을 발생시키도록 구성될 수도 있으며, 그 예측 블록은 (i) 현재의 블록에 이웃하는 하나 이상의 블록들로부터의 모션 정보에 기초한 예측 블록(들), 및 (ii) 현재의 서브-블록의 모션 정보에 기초한 예측 블록의 가중 평균일 수도 있다.

[0112] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록에 하나 이상의 이웃하는 서브-블록들이 모션 정보의 다수의 세트들을 나타낼 수도 있는 하나 보다 많은 서브-PU 를 포함하는지 (예컨대, 포함하는 경우) 를 결정하도록 구성될 수도 있다. 이러한 예들에서, 현재의 서브-블록에의 하나 이상의 이웃하는 서브-블록들이 하나 보다 많은 서브-PU들을 포함한다고 결정하는 것에 기초하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 이웃하는 이러한 서브-PU들 중 하나 이상의 모션 정보를 이용하여 현재의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 이웃하지 않는 서브-PU 로부터의 모션 정보를 이용하지 않도록 또는 아니면 배제하도록 구성될 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 이웃하는 블록의 임의의 서브-PU 의 모션 정보를 이용하여, 이웃하는 블록의 임의의 서브-PU 가 현재의 서브-블록에 이웃하든 아니든, 현재의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0113] 본 개시물의 제 6 기법에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 정사각형 또는 비-정사각형 사이즈 (예컨대, 직사각형) 을 갖는 임의의 블록 (예컨대, PU 또는 CU) 에 대해 OBMC 를 수행하도록 구성될 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 특수 유형들 또는 사이즈들을 갖는 블록 (예컨대, CU 또는 PU) (예컨대, 미리 정의된 유형들 또는 사이즈들, HEVC 에서의 병합 모드로 코딩된 PU들, HEVC 와는 상이한 코딩 표준에서의 병합 모드와 유사한 모드로 코딩된 PU들, 또는 32x32 보다 큰 PU들) 에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0114] 본 개시물의 제 7 기법에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 각각의 가용 이웃하는 서브-블록, 서브-PU, 또는 블록의 모션 정보 (예컨대, 모션 벡터들) 를 이용하여 현재의 서브-블록, 서브-PU, 또는 블록에 대해 OBMC 를 수행하여 예측 블록을 발생시키도록 구성될 수도 있다. 일부 예들에서, 예측 블록은 P_k 에 의해 표시될 수도 있으며, 여기서, N 은 현재의 서브-블록, 서브-PU, 또는 블록에 이웃하는 이웃하는 블록에 대한 (예컨대, 상부, 하부, 좌측 또는 우측과 동일한) 인덱스/표시를 나타낸다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 이웃하는 블록들의 예측 블록(들) 을 이용하여, P_k 에 의해 표시될 수도 있는, 현재의 블록 (예컨대, 서브-블록, 서브-PU, 또는 블록) 의 예측 블록을 업데이트하도록 구성될 수도 있다.

[0115] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 가중 평균을 수행하여 P_k 를 업데이트하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 (i) 이웃하는

블록의 모션 정보 (예컨대, 모션 벡터(들))로부터의 예측 블록 P_N 의 하나 이상의 K 로우(들) 또는 칼럼(들), 및 (ii) 현재의 예측 블록 P_C 의 대응하는 로우(들) 또는 칼럼(들)의 가중 평균을 수행하여 P_C 를 업데이트하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 상이한 가중치들을 상이한 위치들 (예컨대, 상이한 로우(들) 또는 칼럼(들))에 할당하도록 구성될 수도 있다. 일부 예들에서, K는 1, 2, 3, 또는 4의 값과 동일하게 설정될 수도 있다. 하나 이상의 K 로우(들) 또는 칼럼(들)은 N으로 표시된 바와 같은 이웃하는 블록에 가장 가까운 로우(들) 또는 칼럼(들)이다. 예를 들어, 이웃하는 블록이 현재의 블록에 좌측-이웃하는 블록 또는 우측-이웃하는 블록이면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 (i) 이웃하는 블록의 모션 정보 (예컨대, 모션 벡터(들))로부터의 예측 블록 P_N 의 하나 이상의 K 칼럼(들), 및 (ii) 현재의 예측 블록 P_C 의 대응하는 칼럼(들)의 가중 평균을 수행하여, P_C 를 업데이트하도록 구성될 수도 있다. 다른 예로서, 이웃하는 블록이 현재의 블록에 상부-이웃하는 블록 또는 저부-이웃하는 블록이면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 (i) 이웃하는 블록의 모션 정보 (예컨대, 모션 벡터(들))로부터의 예측 블록 P_N 의 하나 이상의 K 로우(들), 및 (ii) 현재의 예측 블록 P_C 의 대응하는 로우(들)의 가중 평균을 수행하여 P_C 를 업데이트하도록 구성될 수도 있다.

[0116] 아래 테이블 1을 참조하면, x 및 y는 현재의 서브-블록, 서브-PU, 또는 블록의 좌상부 픽셀에 대해 수직 및 수평 좌표들을 나타낼 수도 있다. $P_C(x, y)$ 및 $P_N(x, y)$ 의 가중치는 $w1(x, y)$ 및 $w2(x, y)$ 로서 각각 표시된다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 가중 평균을 수행하여, 업데이트된 $P_C(x, y)$ 가 $w1(x, y) * P_C(x, y) + w2(x, y) * P_N(x, y)$ 와 동일하게 P_C 를 업데이트하도록 구성될 수도 있다. 현재의 서브-블록, 서브-PU, 또는 블록의 폭 및 높이는 W 및 H로 각각 표시될 수도 있다. 예를 들어, W-1은 현재의 서브-블록, 서브-PU, 또는 블록의 최종 칼럼 (예컨대, 가장 우측 칼럼)을 지칭할 수도 있다. 다른 예로서, H-1은 현재의 서브-블록, 서브-PU, 또는 블록의 최종 로우 (예컨대, 최하단의 로우)를 지칭할 수도 있다. 다른 예로서, H-2는 현재의 서브-블록, 서브-PU, 또는 블록의 끝으로부터 두번째 로우 (예컨대, 최종 로우 상부의 첫번째 로우로서 설명될 수도 있는, 두번째 최하단 로우)를 지칭할 수도 있다. N의 값들은 본문에서 설명되는 예측 블록 P_N 의 N을 지칭할 수도 있다. 가중치들 w1 및 w2의 일 예가 아래 테이블 1에 규정된다.

[0117] 테이블 1 (OBMC에서 사용되는 예시적인 가중치들):

중첩된 위치들		w1	w2
N = 상부 y = 0..W-1	[(0, y)]	[3/4]	[1/4]
	(1, y)	7/8	1/8
	(2, y)	15/16	1/16
	(3, y)	31/32	1/32
N = 하부 y = 0..W-1	[(H-1, y)]	[3/4]	[1/4]
	(H-2, y)	7/8	1/8
	(H-3, y)	15/16	1/16
	(H-4, y)	31/32	1/32
N = 좌측 x = 0..H-1	[(x, 0)]	[3/4]	[1/4]
	(x, 1)	7/8	1/8
	(x, 2)	15/16	1/16
	(x, 3)	31/32	1/32
N = 우측 x = 0..H-1	[(x, W-1)]	[3/4]	[1/4]
	(x, W-2)	7/8	1/8
	(x, W-3)	15/16	1/16
	(x, W-4)	31/32	1/32

[0118]

- [0119] 일부 예들에서, N 이 상부와 동일할 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 상부 4개의 로우들이 R_N 에 의해 업데이트되게 가중 평균을 수행하도록 구성될 수도 있다. 이와 유사하게, N 이 하부와 동일할 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 저부 4개의 로우들이 R_N 에 의해 업데이트되게 가중 평균을 수행하도록 구성될 수도 있다. 다른 예로서, N 이 좌측과 동일할 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 좌측 4개의 칼럼들이 R_N 에 의해 업데이트되게 가중 평균을 수행하도록 구성될 수도 있다. 이와 유사하게, N 이 우측과 동일할 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 우측 4개의 칼럼들이 R_N 에 의해 업데이트되게 가중 평균을 수행하도록 구성될 수도 있다.
- [0120] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 하나의 로우 또는 칼럼을 R_N 에 의해 업데이트하도록 구성될 수도 있다. 이러한 예들에 따른 위치들 및 가중치들은 위 테이블 1 에 [각괄호로 표시된다].
- [0121] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 2개의 로우들 또는 칼럼들을 R_N 에 의해 업데이트하도록 구성될 수도 있다. 이러한 예들에 따른 2개의 로우들 또는 칼럼들의 위치들 및 가중치들은 위 테이블 1 에서 [각괄호로 표시되며] 단일 밑줄로 표시된다 (예컨대, 각각의 N 값에 대해 첫번째/최상부 2개의 로우들).
- [0122] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 3개의 로우들 또는 칼럼들을 R_N 에 의해 업데이트하도록 구성될 수도 있다. 이러한 예들에 따른 3개의 로우들 또는 칼럼들의 위치들 및 가중치들은 위 테이블 1 에 [각괄호로 표시되며], 단일 밑줄로 표시되며, 이중 밑줄로 표시된다 (예컨대, 각각의 N 값에 대해 첫번째/최상부 3개의 로우들).
- [0123] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록, 서브-PU, 또는 블록의 사이즈에 기초하여 로우들의 수 (예컨대, 하나 이상의 로우들) 또는 칼럼들의 수 (예컨대, 하나 이상의 칼럼들) 를 선택하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록, 서브-PU, 또는 블록이 제 1 사이즈를 가지면 더 많은 로우들 또는 칼럼들을 선택하도록, 그리고, 현재의 서브-블록, 서브-PU, 또는 블록이 제 2 사이즈를 가지면 더 적은 로우들 또는 칼럼들을 선택하도록, 구성될 수도 있다. 예를 들어, 현재의 PU 사이즈가 8×4 또는 4×8 와 동일하거나 및/또는 현재의 PU 가 서브-PU 모드들로 코딩되면 (예컨대, 현재의 PU 가 하나 이상의 서브-PU들을 포함한다는 것을 의미하면), 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 단지 2개의 로우들 또는 칼럼들을 R_N 에 의해 업데이트하도록 구성될 수도 있다. 다른 예로서, 현재의 PU 사이즈가 8×4 또는 4×8 보다 크면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 4개의 로우들 또는 칼럼들을 R_N 에 의해 업데이트하도록 구성될 수도 있다.
- [0124] 또 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 P_C 의 모든 픽셀을 R_N 의 대응하는 픽셀에 의해 업데이트하도록 구성될 수도 있다.
- [0125] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 가산 및 시프트 동작들을 이용함으로써 임의의 가중 평균에 사용되는 가중치들을 구현하도록 구성될 수도 있다. 이러한 예들 또는 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 $w1(x, y) + w2(x, y)$ 가 1 과 동일하게 가중치들 (예컨대, 테이블 1 에 나타난 가중치들 또는 테이블 1 에 나타난 가중치들과는 상이한 가중치들) 을 구현하도록 구성될 수도 있다.
- [0126] 일부 예들에서, R_N 에 의해 업데이트되는 P_C 의 로우 또는 칼럼 개수는 현재의 예측 블록의 사이즈에 의존될 (예컨대, 기초할) 수도 있다. 이러한 예들에서, 현재의 예측 블록의 사이즈는 서브-블록, 서브-PU, PU, 또는 CU 의 사이즈에 대응할 수도 있다. 일부 예들에서, R_N 에 의해 업데이트된 P_C 의 로우 개수는 W 에 의존할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 H 가 더 클 (예컨대, 제 1 값을 가질) 때 더 많은 로우들을 업데이트하도록 구성될 수도 있다. 다른 예로서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 H 가 더 작을 (예컨대, 제 1 값보다 작은 제 2 값을 가질) 때 더 적은 로우들을 업데이트하도록 구성될 수도 있다. 다른 예로서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 W 가 더 클 (예컨대, 제 1 값을 가질) 때 칼럼들 로우들을 업데이트하도록 구성될 수도 있다. 다른 예로서, 비디오 인

코더 (20) 및/또는 비디오 디코더 (30) 는 W 가 더 작을 (예컨대, 제 1 값보다 작은 제 2 값을 가질) 때 더 적은 칼럼들을 업데이트하도록 구성될 수도 있다.

[0127] 본 개시물의 제 8 기법에 따르면, 다수의 이웃들이 R_N 으로 각각 표시되는 다수의 예측자들을 제공할 때, 예측자들 R_N 과 현재의 예측자 P_C 의 조합은 순서대로 하나씩 이루어질 수도 있다. 여기서, 예측자는 이웃하는 블록의 모션 정보를 이용하여 인터-예측에 의해 결정된 픽셀 값들의 블록으로 이루어질 수도 있는 예측 블록을 의미할 수도 있다. 예를 들어, 각각의 조합 후, 현재의 예측자 P_C 는 각각의 칼라 성분의 내부 프로세싱 비트 심도일 수도 있는, 조합 전 현재의 예측자의 비트 심도와 동일한 비트 심도로서 업데이트되어 유지될 수도 있다. 각각의 조합 동안, 각각의 픽셀에 대한 가중 프로세스는 값 r 을 사사오입하는 것을 포함할 수도 있으며, 그 결과 가중 프로세스는 $(a1(x, y) * P_C(x, y) + a2(x, y) * R_N(x, y) + r) / S$ 로 변경되며, 여기서, $a1/S$ 및 $a2/S$ 는 $w1$ 및 $w2$ 와 각각 동일하다. 일부 예들에서, r 은 0 또는 $S/2$ 로 설정될 수도 있다. 이 예에서, $a1$ 은 $w1$ 의 양자화된 값이며, $a2$ 는 $w2$ 의 양자화된 값이며, 그리고 S 는 정규화 인자이다.

[0128] 일부 예들에서, 4개의 예측자들이 이용가능한 순서는 상부, 좌측, 저부, 우측일 수도 있다. 다른 예들에서, 4개의 예측자들이 이용가능한 순서는 좌측, 상부, 저부, 우측일 수도 있다. 다른 예들에서, 4개의 예측자들이 이용가능한 순서는 상기 2개의 예들과는 상이한 임의의 순서일 수도 있다.

[0129] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃 블록이 모션 정보를 포함하지 않으면 (예컨대, 양을 때) 이것이 R_N 이 이용불가능하다는 것을 표시할 수도 있기 때문에 조합 프로세스를 스킵하도록 구성될 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 모든 예측자들 (예컨대, 모든 R_N) 을 수집하고 모든 예측자들을 이용하여 합동 (joint) 가중 예측 프로세스를 수행하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 R_N 이 이용불가능하다고 결정하도록 구성될 수 있으며, 이용불가능한 것으로 결정된 하나 이상의 R_N 을 P_C 와 동일하게 설정하도록 구성될 수도 있다.

[0130] 일부 예들에서, 각각의 조합 동안 (예컨대, 예측 블록(들) R_N 의 각각의 조합에 대해 가중 평균 프로세스를 수행하는 동안), 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 비트 심도가 증가함에 따라 현재의 예측자 P_C 를 업데이트하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 다음 조합이 프로세싱되기 전에 디-스케일링 (de-scaling) 동작을 수행함이 없이 현재의 예측자 P_C 를 업데이트하도록 구성될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 모든 조합들이 프로세싱된 후 디-스케일링 동작을 수행하도록 구성될 수도 있다. 디스케일링은 위에서 진술한 정규화 인자 S 에 의한 분할을 지칭하며, 이는 가중된 조합 값을 각각의 입력 예측의 동일한 레벨로 되게 할 수도 있다. 모든 조합들이 프로세싱된 후 디-스케일링함으로써, 비디오 디코더 (30) 는 조합들을 순차적으로 프로세싱하고 합동 가중 예측 프로세스를 이용함으로써 동일한 최종 예측 결과들을 획득하도록 구성될 수도 있다. 이러한 설계는 다수의 경계 OBMC 가 동일한 예측 블록에 적용될 때 구현에서의 유연성을 제공할 수도 있다.

[0131] 본 개시물의 제 9 기법에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 CU, PU, 또는 임의의 다른 블록 레벨에서 OBMC 를 적응적으로 인에이블하거나 또는 디스에이블하도록 (예컨대, 스위칭 온 또는 오프하도록) 구성될 수도 있다. 예를 들어, OBMC 가 특정의 블록 레벨에 대해 인에이블될 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 그 특정의 블록 레벨에 대해 본원에서 설명되는 하나 이상의 기법들을 이용하여 OBMC 를 수행하도록 구성될 수도 있다. 다른 예로서, OBMC 가 특정의 블록 레벨에 대해 디스에이블될 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 그 특정의 블록 레벨에 대해 본원에서 설명되는 하나 이상의 기법들을 이용하여 OBMC 를 수행하지 않도록 구성될 수도 있다.

[0132] 일부 예들에서, 비디오 인코더 (20) 는 OBMC 플래그를 시그널링함으로써 CU, PU, 또는 임의의 다른 블록 레벨에서 OBMC 를 적응적으로 인에이블하거나 또는 디스에이블하도록 (예컨대, 스위칭 온 또는 오프하도록) 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 OBMC 가 특정의 블록에 대해 인에이블되는지 또는 디스에이블되는지 여부를 나타내는 값을 가진 OBMC 플래그와 같은, OBMC 신택스 엘리먼트를 시그널링하도록 구성될 수도 있으며, 비디오 디코더 (30) 는 OBMC 가 특정의 블록에 대해 인에이블되는지 또는 디스에이블되는지 여부를 나타내는 값을 가진 OBMC 플래그와 같은, OBMC 신택스 엘리먼트를 수신하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 각각의 인터-코딩된 CU, PU, 또는 블록에 대한 OBMC 플래그를 시그널링하도록 구

성될 수도 있다. 일부 예들에서, OBMC 플래그는 2진수이며 2개의 값들 중 하나를 가질 수도 있다. OBMC 플래그의 제 1 값은 OBMC 가 인에이블되어 있다는 것을 표시할 수도 있으며, OBMC 플래그의 제 2 값은 OBMC 가 디스에이블되어 있다는 것을 표시할 수도 있다. 예를 들어, OBMC 플래그가 참일 (예컨대, 제 1 값에 대응하는 값을 가질) 때, OBMC 는 현재의 CU, PU, 또는 블록에 대해 적용된다 (예컨대, 인에이블된다). 다른 예로서, OBMC 플래그가 거짓일 (예컨대, 제 2 값에 대응하는 값을 가질) 때, OBMC 는 현재의 CU, PU, 또는 블록에 대해 적용되지 않는다 (예컨대, 디스에이블된다).

[0133] 이와 유사하게, 비디오 디코더 (30) 는 OBMC 플래그를 수신하고 OBMC 플래그의 값을 결정하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터 비디오 인코더 (20) 에 의해 발생하는 비트스트림에서 OBMC 플래그를 수신하도록 구성될 수도 있다. 일부 예들에서, OBMC 플래그에 대응하는 값은 디코딩 중인 비디오 데이터의 블록 (예컨대, CU, PU, 또는 다른 블록) 에 대한 OBMC 가 인에이블되는지 또는 디스에이블되는지 여부를 표시한다. 이러한 예들에서, 비디오 디코더 (30) 는 OBMC 가 특정의 블록에 대해 인에이블되는지 또는 디스에이블되는지 여부를 나타내는 값을 가진 OBMC 플래그를 수신하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 각각의 인터-코딩된 CU, PU, 또는 블록에 대한 OBMC 플래그를 수신하도록 구성될 수도 있다. OBMC 플래그의 제 1 값은 OBMC 가 인에이블되어 있다는 것을 표시할 수도 있으며, OBMC 플래그의 제 2 값은 OBMC 가 디스에이블되어 있다는 것을 표시할 수도 있다. 예를 들어, OBMC 플래그가 참일 (예컨대, 제 1 값에 대응하는 값을 가질) 때, 비디오 디코더 (30) 는 OBMC 가 현재의 CU, PU, 또는 블록에 대해 적용된다 (예컨대, 인에이블된다) 고 결정하도록 구성될 수도 있다. 다른 예로서, OBMC 플래그가 거짓일 (예컨대, 제 2 값에 대응하는 값을 가질) 때, 비디오 디코더 (30) 는 OBMC 가 현재의 CU, PU, 또는 블록에 대해 적용되지 않는다 (예컨대, 디스에이블된다) 고 결정하도록 구성될 수도 있다.

[0134] 일부 예들에서, 비디오 인코더 (20) 는 CU, PU, 또는 블록에 사용되는 파티션 유형(들), CU, PU, 또는 블록에 사용되는 예측 모드 (예컨대, 병합 모드 또는 AMVP 모드), 또는 CU, PU, 또는 블록의 사이즈에 기초하여, 단지 CU, PU, 또는 블록에 대한 OBMC 플래그를 시그널링하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 는 파티션 유형(들), 예측 모드, 및/또는 CU, PU, 또는 블록에 대응하는 사이즈를 결정하고, 파티션 유형(들), 예측 모드, 및/또는 CU, PU, 또는 블록에 대응하는 사이즈 중 하나 이상의 결정에 기초하여, OBMC 플래그를 시그널링하도록 구성될 수도 있다.

[0135] 일 예에서, 비디오 인코더 (20) 는 단지 CU, PU, 또는 블록이 2Nx2N 병합 모드로 코딩되지 않고 CU, PU, 또는 블록의 사이즈가 16x16 보다 작거나 또는 동일하면 CU, PU, 또는 블록에 대한 OBMC 플래그를 시그널링하도록 구성될 수도 있다. 이러한 예에서, 비디오 인코더 (20) 는 CU, PU, 또는 블록이 2Nx2N 병합 모드로 코딩되는지 여부 및 CU, PU, 또는 블록의 사이즈를 결정하고, 그의 결정(들) 에 기초하여 OBMC 플래그를 시그널링하도록 구성될 수도 있다.

[0136] 일부 예들에서, 비디오 인코더 (20) 는 CU, PU, 또는 블록에 대한 OBMC 플래그를 시그널링하지 않도록 구성될 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 는 (예컨대, 다음 4개의 단락들에서 다수의 예들을 포함하는 개시물의 단지 일 예로서 설명되는 바와 같은) 다른 시그널링된 정보에 기초하여, OBMC 플래그에 대응하는 값을 내재적으로 유도하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 OBMC 가 CU, PU, 또는 블록에 대해 적용된다 (예컨대, 인에이블된다) 는 것을 표시하는 OBMC 플래그의 값을 내재적으로 유도하도록 구성될 수도 있다. 다른 예로서, 비디오 디코더 (30) 는 OBMC 가 CU, PU, 또는 블록에 대해 적용되지 않는다 (예컨대, 디스에이블된다) 는 것을 표시하는 OBMC 플래그의 값을 내재적으로 유도하도록 구성될 수도 있다.

[0137] 본 개시물의 제 10 기법에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 CU, PU, 서브-블록, 또는 블록 경계에 대해 OBMC 를 적응적으로 인에이블하거나 또는 디스에이블하도록 (예컨대, 스위칭 온 또는 오프하도록) 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 (예컨대, 2개의) 이웃하는 블록들의 모션 정보 (예컨대, 모션 벡터들) 의 특성, 및/또는 하나 이상의 (예컨대, 2개의) 이웃하는 블록들의 모션 정보 (예컨대, 모션 벡터들) 에 기초한 예측된 샘플 값들의 특성에 기초하여, CU, PU, 서브-블록, 또는 블록 경계에서 내재적으로 OBMC 를 적응적으로 인에이블하거나 또는 디스에이블하도록 (예컨대, 스위칭 온 또는 오프하도록) 구성될 수도 있다.

[0138] 일부 예들에서, 블록 경계 (예컨대, CU, PU, 서브-블록, 또는 블록 경계) 에 대해 OBMC 를 적용할지 (예컨대, 인에이블할지) 여부는 현재의 블록과 경계를 공유하는 하나 이상의 (예컨대, 2개의) 이웃하는 블록들의 모션 벡터들 사이의 모션 벡터 차이에 기초할 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록과 경계를 공유하는 하나 이상의 (예컨대, 2개의) 이웃하는 블록들의 모션 벡터들 사

이의 모션 벡터 차이를 결정하도록 구성될 수도 있다. 모션 벡터 차이가 임계치 (예컨대, 16 개의 픽셀들 또는 임의의 다른 임계치) 보다 클 때, OBMC 가 경계(들) 에 대해 적용되지 않을 수도 있으며 (예컨대, 디스에이블될 수도 있으며); 그렇지 않으면, OBMC 가 경계(들) 에 대해 적용될 (예컨대, 인에이블될) 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 결정된 모션 벡터 차이를 미리 정의된 임계치와 비교하고 그 비교에 기초하여 하나 이상의 경계들에 대해 OBMC 를 인에이블하거나 또는 디스에이블하도록 구성될 수도 있다. 일부 예들에서, 임계치는 미리 정의된 값일 수도 있으며, 인코딩된 비디오 비트스트림으로 시그널링되거나 또는 되지 않을 수도 있다.

[0139] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 2개의 이웃하는 블록들의 모션 벡터들 사이의 모션 벡터 차이를 결정하도록 구성될 수도 있는 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 2개의 모션 벡터들 각각이 상이한 참조 프레임을 가리키면 (예컨대, 가리킬 때), 또는 적어도 하나의 모션 벡터가 이용불가능하면 (예컨대, 이용불가능할 때) OBMC 를 적용하지 않도록 (예컨대, 디스에이블하도록) 구성될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 2개 이상의 이웃하는 블록들의 모션 벡터들 사이의 모션 벡터 차이를 결정하도록 구성될 수도 있는 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 2개 이상의 모션 벡터들 각각이 상이한 참조 프레임을 가리키면 (예컨대, 가리킬 때), 또는 적어도 하나의 모션 벡터가 이용불가능하면 (예컨대, 이용불가능할 때) OBMC 를 적용하도록 구성될 수도 있다.

[0140] 다른 예들에서, 블록 경계 (예컨대, CU, PU, 서브-블록, 또는 블록 경계) 에 대해 OBMC 를 적용할지 (예컨대, 인에이블할지) 여부는 현재의 블록과 경계를 공유하는 2개의 이웃하는 블록들의 2개의 모션 벡터들 (예컨대, 2개의 이웃하는 블록들의 각각에 대해 하나의 모션 벡터) 양쪽이 정수 픽셀 위치들을 가리키는지 여부에 기초할 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록과 경계를 공유하는 2개의 이웃하는 블록들의 2개의 모션 벡터들 (예컨대, 2개의 이웃하는 블록들의 각각에 대해 하나의 모션 벡터) 양쪽이 정수 픽셀 위치들을 가리키는지 여부를 결정하도록 구성될 수도 있다. 일부 예들에서, 양쪽의 모션 벡터들이 정수 픽셀 위치들을 가리키면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 경계(들) 에 대해 OBMC 를 적용하지 않도록 (예컨대, 디스에이블하도록) 구성될 수도 있다. 다른 예들에서, 양쪽의 모션 벡터들이 정수 픽셀 위치들을 가리키지 않으면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 경계(들) 에 대해 OBMC 를 적용하도록 (예컨대, 인에이블하도록) 구성될 수도 있다.

[0141] 다른 예들에서, 블록 경계 (예컨대, CU, PU, 서브-블록, 또는 블록 경계) 에 대해 OBMC 를 적용할지 (예컨대, 인에이블할지) 여부는 현재의 블록과 경계를 공유하는 2개의 이웃하는 블록들의 2개의 모션 벡터들로부터 유도된 2개의 예측된 블록들의 샘플 값 차이에 기초할 수도 있다. 예를 들어, 2개의 예측된 블록들의 평균 절대 차이가 (예컨대, 80 과 같은) 어떤 임계치보다 클 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 그 경계에 대해 OBMC 를 적용하지 않도록 (예컨대, 디스에이블하도록) 구성될 수도 있다. 다른 예로서, 2개의 예측된 블록의 샘플들 사이의 최대 절대 차이가 (예컨대, 80, 88, 또는 96 과 같은) 어떤 임계치보다 클 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 그 경계에 대해 OBMC 를 적용하지 않도록 (예컨대, 디스에이블하도록) 구성될 수도 있다. 다른 예로서, 2개의 예측된 블록의 샘플들 사이의 최대 절대 차이가 (예컨대, 80, 88, 또는 96 과 같은) 어떤 임계치보다 작거나 또는 동일할 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 그 경계에 대해 OBMC 를 적용하도록 (예컨대, 인에이블하도록) 구성될 수도 있다. 일부 예들에서, 임계치는 미리 정의된 값일 수도 있으며, 인코딩된 비디오 비트스트림으로 시그널링되거나 또는 되지 않을 수도 있다. 이들 예들에서, 예측된 블록들의 사이즈는 (i) 상기 본 개시물의 제 4 기법에서 설명된 서브-블록 사이즈와 동일한 사이즈이거나, (ii), 미리 정의되거나, 또는 (iii) 비트스트림으로 시그널링될 수도 있다.

[0142] 본원에서 사용될 때, 경계에 대해 OBMC 를 디스에이블하거나 또는 적용하지 않는 것은 경계에 대해 OBMC 를 수행하지 않는다는 것을 지칭할 수도 있으며, 반대의 경우도 마찬가지이다. 이와 유사하게, 본원에서 사용될 때, 경계에 대해 OBMC 를 인에이블하거나 또는 적용하는 것은 경계에 대해 OBMC 를 수행하는 것을 지칭할 수도 있으며, 반대의 경우도 마찬가지이다. 본원에서 사용될 때, "X 에 대한 (for X)" 에 관련한 임의의 참조는 "X 에 대해 (to X)" 또는 "X 상에서 (on X)" 를 지칭할 수도 있다. 예를 들어, "블록에 대한 (for a block)" 에 대한 참조는 "블록에 대해 (to a block)" 또는 "블록 상에서 (on a block)" 를 지칭할 수도 있다. "X 에 대한 (for X)", "X 에 대해 (to X)", 또는 "X 상에서 (on X)" 에 관련한 임의의 참조는 "X 에 대한 (for X)", "X 에 대해 (to X)", 및 "X 상에서 (on X)" 중 임의의 것을 지칭할 수도 있음을 알 수 있다. 예를 들어, "서브-블록 상에서의" 에 대한 참조는 "서브-블록에 대한" 또는 "서브-블록에 대해" 를 유사하게 지칭할 수도 있다.

[0143] 본 개시물의 제 11 기법에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 CU, PU, 서브-블록, 임

의 다른 블록 레벨에서 OBMC 를 적응적으로 인에이블하거나 또는 디스에이블하도록 (예컨대, 스위칭 온 또는 오프하도록) 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 각각의 인터-코딩된 CU, PU, 서브-블록, 또는 블록에 대해 OBMC 를 적응적으로 인에이블하거나 또는 디스에이블하도록 (예컨대, 스위칭 온 또는 오프하도록) 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 본 개시물의 제 9 기법에 대해서 설명된 하나 이상의 기법들 및 본 개시물의 제 10 기법에 대해서 설명된 하나 이상의 기법들을 구현하는 OBMC 를 적응적으로 인에이블하거나 또는 디스에이블하도록 (예컨대, 스위칭 온 또는 오프하도록) 구성될 수도 있다. 예를 들어, OBMC 플래그가 CU 또는 PU 에 대해 시그널링될 때, 비디오 디코더 (30) 는 OBMC 플래그의 값에 기초하여, OBMC 를 수행하도록 (예컨대, 인에이블하거나 또는 적용하도록) 또는 수행하지 않도록 (예컨대, 디스에이블하거나 또는 적용하지 않도록) 구성될 수도 있다.

다른 예로서, OBMC 플래그가 CU 또는 PU 에 대해 시그널링되지 않을 때, 비디오 디코더 (30) 는 본 개시물의 제 10 기법에 대해서 위에서 설명한 하나 이상의 방법들을 이용하여, OBMC 를 수행하도록 (예컨대, 인에이블하거나 또는 적용하도록) 또는 수행하지 않도록 (예컨대, 디스에이블하거나 또는 적용하지 않도록) 구성될 수도 있다.

[0144] 본 개시물의 제 12 기법에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 모든 모션 보상된 (MC) 블록 경계들에 대해 OBMC 를 수행하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 블록 (예컨대, CU) 의 우측 및 저부 경계들을 제외한, 모든 모션 보상된 (MC) 블록 경계들에 대해 OBMC 를 수행하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 루마 및/또는 크로마 성분들에 대해 OBMC 를 수행하도록 구성될 수도 있다.

[0145] HEVC 에서, MC 블록은 PU 에 대응한다. PU 가 HEVC 에서의 ATMVP 모드로 코딩될 때, 각각의 서브-PU (4x4) 는 MC 블록이다. CU, PU, 또는 서브-PU 경계들을 균일한 방식으로 프로세싱하기 위해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 블록의 모든 MC 블록 경계들에 대해 서브-블록 레벨에서 본원에서 설명되는 하나 이상의 기법들에 따라서 OBMC 를 수행하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 도 8 에 예시된 바와 같이, 서브-블록 사이즈를 4x4 와 동일하게 설정하도록 구성될 수도 있다.

[0146] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 대해 OBMC 를 수행하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재의 서브-블록에 대한 하나 이상의 예측 블록들을 유도하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이용가능한 경우 4개의 연결된 이웃하는 서브-블록들 (예컨대, 현재의 서브-블록과 경계선을 공유하는 4개의 서브-블록들) 의 모션 벡터들을 이용하여 현재의 서브-블록에 대해 4개의 예측 블록들을 유도하도록 구성될 수도 있다. 다른 예로서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 4개의 연결된 이웃하는 서브-블록들 (예컨대, 현재의 서브-블록과 경계선을 공유하는 4개의 서브-블록들) 의 모션 벡터들이 이용가능하고 현재의 블록에 대한 현재의 모션 벡터와 동일하지 않으면, 4개의 연결된 이웃하는 서브-블록들 (예컨대, 현재의 서브-블록과 경계선을 공유하는 4개의 서브-블록들) 의 모션 벡터들을 이용하여, 현재의 서브-블록에 대해 4개의 예측 블록들을 유도하도록 구성될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 다수의 모션 벡터들 (예컨대, 아래에서 설명되는 바와 같이 4개의 예측 블록들의 각각과 연관된 하나의 모션 벡터, 또는 4개의 예측 블록들의 각각과 연관된 2개의 모션 벡터들) 에 기초하여, 이들 다수의 예측 블록들 (예컨대, 4개의 연결된 이웃하는 서브-블록들에 기초한 4개의 예측 블록들) 에 대해 가중 평균을 수행하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 가중 평균을 수행하여 현재의 서브-블록의 최종 예측 블록을 발생시키도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 다수의 모션 벡터들을 이용하여 다수의 예측 샘플들을 획득하고, 그 후 가중 평균을 샘플들에 적용하도록 구성될 수도 있다. 최종 예측 블록은 현재의 서브-블록에서의 픽셀들에 대한 예측된 픽셀들의 블록을 지칭할 수도 있다. 이 문단에서, 상기 예들이 단방향 예측에 적용되지만, 양방향 예측을 수반하는 예들에서는, 상기 예들에서의 예측 블록이 4개의 이웃하는 블록들의 각각으로부터의 2개의 모션 벡터들을 이용함으로써 발생될 수도 있다는 점에 유의한다.

[0147] 일부 예들에서, 이웃하는 서브-블록의 하나 이상의 모션 벡터들에 기초한 예측 블록은 P_N 으로서 표시될 수도 있으며, 여기서, N 은 이웃하는 서브-블록에 대한 인덱스이다. 즉, N 은 예측 블록이 상부, 하부, 좌측, 또는 우측 이웃하는 서브-블록인지 여부를 식별한다. 현재의 서브-블록의 하나 이상의 모션 벡터들에 기초한 예측 블록은 P_C 로서 표시될 수도 있다. 일부 예들에서, P_N 의 모든 픽셀이 P_C 에서 동일한 픽셀에 추가된다

(예컨대, R_N 의 4개의 로우들 및/또는 칼럼들이 R_C 에 추가된다). 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 R_N 에 대해 $\{1/4, 1/8, 1/16, 1/32\}$ 의 가중치 인자들을, 그리고 R_C 에 대해 $\{3/4, 7/8, 15/16, 31/32\}$ 의 가중치 인자들을 이용하도록 구성될 수도 있다. 작은 MC 블록들에 대해 (예컨대, PU 사이즈가 8×4 , 4×8 와 동일하거나, 또는 PU 가 ATMVP 모드로 코딩되는 경우) 계산 비용을 낮게 유지하기 위해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 R_N 의 2개의 로우들 및/또는 칼럼들을 R_C 에 단지 추가하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 R_N 에 대해 $\{1/4, 1/8\}$ 의 가중 인자들을, 그리고, R_C 에 대해 가중 인자 $\{3/4, 7/8\}$ 을 적용하도록 구성될 수도 있다. 수직 이웃하는 서브-블록 (예컨대, 상부 또는 저부) 의 하나 이상의 모션 벡터들에 기초하여 발생된 각각의 R_N 에 대해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 R_N 의 동일한 로우에서의 픽셀들을 R_C 에 동일한 가중 인자로 추가하도록 구성될 수도 있다. 수평 이웃하는 서브-블록 (예컨대, 좌측 또는 우측) 의 하나 이상의 모션 벡터들에 기초하여 발생된 각각의 R_N 에 대해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 R_N 의 동일한 칼럼에서의 픽셀들을 R_C 에 동일한 가중 인자로 추가하도록 구성될 수도 있다.

[0148] 도 9 는 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라 코딩은 주어진 비디오 프레임 또는 픽처 내 비디오에서 공간 리던던시를 감소시키거나 또는 제거하기 위해, 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접 프레임들 또는 픽처들 내 비디오에서 시간 리던던시를 감소시키거나 또는 제거하기 위해, 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 여러 공간 기반의 압축 모드들 중 임의의 압축 모드를 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향-예측 (B 모드) 과 같은 인터-모드들은 여러 시간-기반의 압축 모드들 중 임의의 모드를 지칭할 수도 있다.

[0149] 도 9 의 예에서, 비디오 인코더 (20) 는 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 프로세싱 유닛 (41) 은 모션 추정 유닛 (MEU) (42), 모션 보상 유닛 (MCU) (44), 및 인트라 예측 프로세싱 유닛 (46) 을 포함한다. 비디오 블록 복원을 위해, 비디오 인코더 (20) 는 또한 역양자화 유닛 (58), 역변환 프로세싱 유닛 (60), 합산기 (62), 필터 유닛 (64), 및 디코딩된 픽처 버퍼 (DPB) (66) 를 포함한다. 도 9 의 하나 이상의 구성요소들은 본 개시물에서 설명되는 하나 이상의 기법들을 수행할 수도 있다.

[0150] 도 9 에 나타낸 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고 수신된 비디오 데이터를 비디오 데이터 메모리 (33) 에 저장한다. 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 구성요소들에 의해 인코딩된 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (33) 에 저장된 비디오 데이터는 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. DPB (66) 는 예컨대, 인트라- 또는 인터-코딩 모드들에서 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩할 때에 사용하기 위해 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66) 는 동기식 DRAM (SDRAM) 을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은, 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 여러 예들에서, 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 다른 구성요소들과의 온칩, 또는 그들 구성요소들에 대한 오프-칩일 수도 있다.

[0151] 파티셔닝 유닛 (35) 은 비디오 데이터 메모리 (33) 로부터 비디오 데이터를 추출하고 그 비디오 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한 슬라이스들, 타일들, 또는 다른 더 큰 유닛들 뿐만 아니라, 예컨대, LCU들 및 CU들의 쿼드트리 구조에 따라서 파티셔닝하는 비디오 블록으로 파티셔닝하는 것을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 인코딩되는 비디오 슬라이스 내 비디오 블록들을 인코딩하는 구성요소들을 예시한다. 슬라이스는 다수의 비디오 블록들로 (그리고, 아마도, 타일들로서 지칭되는 비디오 블록들의 세트들로) 분할될 수도 있다. 예측 프로세싱 유닛 (41) 은 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨) 에 기초하여, 현재의 비디오 블록에 대해, 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나와 같은, 복수의 가능한 코딩 모드들 중 하나를 선택할 수도 있다. 예측 프로세싱 유닛 (41) 은 최종 인트라- 또는 인터-코딩된 블록을, 합산기 (50) 에 제공하여 잔차 블록 데이터를 발생하고, 그리

고 합산기 (62) 에 제공하여, 참조 픽처로서 사용하기 위한 인코딩된 블록을 재구성할 수도 있다.

- [0152] 예측 프로세싱 유닛 (41) 내의 인트라 예측 유닛 (46) 은 코딩될 현재의 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 대해, 현재의 비디오 블록의 인트라-예측 코딩을 수행하여, 공간 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 하나 이상의 참조 픽처들에서의 하나 이상의 예측 블록들에 대해, 현재의 비디오 블록의 인터-예측 코딩을 수행하여, 시간 압축을 제공한다.
- [0153] 모션 추정 유닛 (42) 은 비디오 시퀀스에 대한 미리 결정된 패턴에 따라서 비디오 슬라이스에 대한 인터-예측 모드를 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 그 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들 또는 B 슬라이스들로서 지시할 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적인 목적들을 위해 별개로 예시된다. 모션 추정 유닛 (42) 에 의해 수행되는 모션 추정은 모션 벡터들을 발생하는 프로세스이며, 이 프로세스는 비디오 블록들에 대한 모션을 추정한다. 모션 벡터는, 예를 들어, 참조 픽처 내 예측 블록에 대한, 현재의 비디오 프레임 또는 픽처 내 비디오 블록의 PU 의 변위를 나타낼 수도 있다.
- [0154] 예측 블록은 픽셀 차이의 관점에서 코딩되는 비디오 블록의 PU 에 가깝게 매칭하는 것으로 발견되는 블록이며, 이 픽셀 차이는 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 DPB (66) 에 저장된 참조 픽처들의 서브-정수 픽셀 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 픽처의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 내삽할 수도 있다. 따라서, 모션 추정 유닛 (42) 은 풀 픽셀 위치들 및 분수 픽셀 위치들에 대해, 모션 탐색을 수행하고, 분수 픽셀 정밀도를 가진 모션 벡터를 출력할 수도 있다.
- [0155] 모션 추정 유닛 (42) 은 PU 의 위치를 참조 픽처의 예측 블록의 위치와 비교함으로써 인터-코딩된 슬라이스에서의 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 참조 픽처는 제 1 참조 픽처 리스트 (List 0) 또는 제 2 참조 픽처 리스트 (List 1) 로부터 선택될 수도 있으며, 이 리스트들 각각은 DPB (66) 에 저장된 하나 이상의 참조 픽처들을 식별한다. 모션 추정 유닛 (42) 은 그 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.
- [0156] 모션 보상 유닛 (44) 에 의해 수행되는 모션 보상은 가능한 한, 서브-픽셀 정밀도까지 내삽들을 수행함으로써, 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 패치하거나 또는 발생시키는 것을 수반할 수도 있다. 현재의 비디오 블록의 PU 에 대한 모션 벡터를 수신하자마자, 모션 보상 유닛 (44) 은 모션 벡터가 참조 픽처 리스트들 중 하나에서 가리키는 예측 블록을 로케이트할 수도 있다. 비디오 인코더 (20) 는 코딩중인 현재의 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 픽셀 차이 값들은 블록에 대한 잔차 데이터를 형성하며, 루마 및 크로마 차이 성분들 양쪽을 포함할 수도 있다. 합산기 (50) 는 이 감산 연산을 수행하는 구성요소 또는 구성요소들을 나타낸다. 모션 보상 유닛 (44) 은 또한 비디오 슬라이스의 비디오 블록들을 디코딩할 때에 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관되는 신텍스 엘리먼트들을 발생시킬 수도 있다.
- [0157] 예측 프로세싱 유닛 (41) 이 현재의 비디오 블록에 대한 예측 블록을 인트라 예측 또는 인터 예측을 통해서 발생시킨 후, 비디오 인코더 (20) 는 현재의 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 잔차 블록에서의 잔차 비디오 데이터는 하나 이상의 TU들에 포함되어, 변환 프로세싱 유닛 (52) 에 적용될 수도 있다. 변환 프로세싱 유닛 (52) 은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 잔차 비디오 데이터를 잔차 변환 계수들로 변환한다. 변환 프로세싱 유닛 (52) 은 잔차 비디오 데이터를 픽셀 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다.
- [0158] 변환 프로세싱 유닛 (52) 은 최종 변환 계수들을 양자화 유닛 (54) 으로 전송할 수도 있다. 양자화 유닛 (54) 은 비트 레이트를 추가로 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 그 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 일부 예들에서, 양자화 유닛 (54) 은 그후 양자화된 변환 계수들을 포함하는 매트릭스의 스캐닝을 수행할 수도 있다. 이의 대안으로, 엔트로피 인코딩 유닛 (56) 이 그 스캐닝을 수행할 수도 있다.

- [0159] 양자화 이후, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 2진 산술 코딩 (CABAC), 신택스-기반 컨텍스트-적응 2진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩에 이어서, 인코딩된 비트스트림은 비디오 디코더 (30) 로 송신되거나, 또는 비디오 디코더 (30) 에 의한 추후 송신 또는 취출을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (56) 은 또한 코딩중인 현재의 비디오 슬라이스에 대한 모션 벡터들 및 다른 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.
- [0160] 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60) 은 역양자화 및 역변환을 각각 적용하여, 참조 픽처의 참조 블록으로 추후 사용을 위해 픽셀 도메인에서 잔차 블록을 복원한다. 모션 보상 유닛 (44) 은 잔차 블록을 참조 픽처 리스트들 중 하나 내 참조 픽처들 중 하나의 예측 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한 하나 이상의 내삽 필터들을 그 복원된 잔차 블록에 적용하여, 모션 추정 에 사용하기 위한 서브-정수 픽셀 값들을 계산할 수도 있다. 합산기 (62) 는 복원된 잔차 블록을 모션 보상 유닛 (44) 에 의해 발생하는 모션 보상된 예측 블록에 가산하여, 복원된 블록을 발생한다.
- [0161] 필터 유닛 (64) 은 복원된 블록 (예컨대, 합산기 (62) 의 출력) 을 필터링하고 참조 블록으로서의 사용을 위해, 필터링된 복원된 블록을 DPB (66) 에 저장한다. 참조 블록은 후속 비디오 프레임 또는 픽처에서 블록을 인터-예측하기 위해 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 참조 블록으로서 사용될 수도 있다. 필터 유닛 (64) 은 디블록킹 필터링, 샘플 적응 오프셋 (SAO) 필터링, 적응 루프 필터링 (ALF), 또는 다른 유형들의 루프 필터링 중 하나 이상을 적용할 수도 있다. 필터 유닛 (64) 은 블록 경계들을 필터링하여 복원된 비디오로부터 블록킹 현상 아티팩트들을 제거하기 위해 디블록킹 필터링을 적용할 수도 있으며, 전체 코딩 품질을 향상시키기 위해 다른 유형들의 필터링을 적용할 수도 있다. 추가적인 루프 필터들 (인 루프 또는 사후 루프) 이 또한 사용될 수도 있다.
- [0162] 비디오 인코더 (20) 는 서브-블록 기반의 OBMC, 서브-PU 기반의 OBMC, CU-경계 기반의 OBMC, 또는 이들의 임의의 조합을 수행하도록 구성된 비디오 인코더의 일 예를 나타낸다.
- [0163] 도 10 은 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 10 의 예에서, 비디오 디코더 (30) 는 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역양자화 유닛 (86), 역변환 프로세싱 유닛 (88), 합산기 (90), 및 DPB (94) 를 포함한다. 예측 프로세싱 유닛 (81) 은 모션 보상 유닛 (82) 및 인트라 예측 유닛 (84) 을 포함한다. 비디오 디코더 (30) 는 일부 예들에서, 도 9 로부터의 비디오 인코더 (20) 에 대해 설명된 인코딩 과정과는 일반적으로 반대인 디코딩 과정을 수행할 수도 있다. 도 10 의 하나 이상의 구성요소들은 본 개시물에서 설명되는 하나 이상의 기법들을 수행할 수도 있다.
- [0164] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관되는 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 는 수신된 인코딩된 비디오 비트스트림을 비디오 데이터 메모리 (78) 에 저장한다. 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 구성요소들에 의해 디코딩될, 인코딩된 비디오 비트스트림과 같은, 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (78) 에 저장된 비디오 데이터는 예를 들어, 저장 디바이스 (26) 로부터, 또는 카메라와 같은 로컬 비디오 소스로부터, 링크 (16) 를 통해서, 또는 물리적인 데이터 저장 매체들에 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (78) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (CPB) 를 형성할 수도 있다. DPB (94) 는 예컨대, 인트라- 또는 인터-코딩 모드들에서 비디오 디코더 (30) 에 의해 비디오 데이터를 디코딩할 때에 사용하기 위해 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94) 는 DRAM, SDRAM, MRAM, RRAM, 또는 다른 유형들의 메모리 디바이스들과 같은, 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 여러 예들에서, 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 다른 구성요소들과의 온칩, 또는 그들 구성요소들에 대한 오프-칩일 수도 있다.
- [0165] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 비디오 데이터 메모리 (78) 에 저장된 비디오 데이터를 엔트로피 디코딩하여, 양자화된 계수들, 모션 벡터들, 및 다른 신택스 엘리먼트들을 발생시킨다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 예측 프로세싱 유닛 (81) 으로 포워딩한다.

비디오 디코더 (30) 는 선택스 엘리먼트들을 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 수신할 수도 있다.

[0166] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81) 의 인트라 예측 유닛 (84) 은 시그널링된 인트라 예측 모드 및 현재의 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 발생할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B 또는 P) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81) 의 모션 보상 유닛 (82) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 모션 벡터들 및 다른 선택스 엘리먼트들에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 발생한다. 예측 블록들은 참조 픽처 리스트들 중 하나 내 참조 픽처들 중 하나로부터 발생할 수도 있다. 비디오 디코더 (30) 는 디폴트 구성 기법들을 이용하여, DPB (94) 에 저장된 참조 픽처들에 기초하여, 참조 프레임 리스트들, 즉, List 0 및 List 1 을 구성할 수도 있다.

[0167] 모션 보상 유닛 (82) 은 모션 벡터들 및 다른 선택스 엘리먼트들을 파악하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그리고, 그 예측 정보를 이용하여, 디코딩중인 현재의 비디오 블록에 대한 예측 블록들을 발생한다. 예를 들어, 모션 보상 유닛 (82) 은 그 수신된 선택스 엘리먼트들 중 일부를 이용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는 사용되는 예측 모드 (예컨대, 인트라- 또는 인터-예측), 인터-예측 슬라이스 유형 (예컨대, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 참조 픽처 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 다른 정보를 결정하여, 현재의 비디오 슬라이스에서의 비디오 블록들을 디코딩한다.

[0168] 모션 보상 유닛 (82) 은 또한 내삽 필터들에 기초하여 내삽을 수행할 수도 있다. 모션 보상 유닛 (82) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 사용되는 것과 같은 내삽 필터들을 이용하여, 참조 블록들의 서브-정수 픽셀들에 대해 내삽된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (82) 은 수신된 선택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 사용되는 내삽 필터들을 결정하고 그 내삽 필터들을 이용하여 예측 블록들을 발생할 수도 있다.

[0169] 역양자화 유닛 (86) 은 비트스트림으로 제공되어 엔트로피 디코딩 유닛 (80) 에 의해 디코딩되는 양자화된 변환 계수들을 역양자화한다, 즉 양자화 해제한다. 역양자화 프로세스는 양자화의 정도와, 마찬가지로, 적용되어야 하는 역양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다. 역변환 프로세싱 유닛 (88) 은 픽셀 도메인에서 잔차 블록들을 발생하기 위해 역변환, 예컨대, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역변환 프로세스를 변환 계수들에 적용한다.

[0170] 예측 프로세싱 유닛이 예를 들어, 인트라 또는 인터 예측을 이용하여 현재의 비디오 블록에 대한 예측 블록을 발생시킨 후, 비디오 디코더 (30) 는 역변환 프로세싱 유닛 (88) 으로부터의 잔차 블록들을 모션 보상 유닛 (82) 에 의해 발생된 대응하는 예측 블록들과 합산함으로써, 복원된 비디오 블록을 형성한다. 합산기 (90) 는 이 합산 동작을 수행하는 구성요소 또는 구성요소들을 나타낸다. 필터 유닛 (92) 은 예를 들어, 디블록킹 필터링, SAO 필터링, ALF 필터링, 또는 다른 유형들의 필터링 중 하나 이상을 이용하여, 복원된 비디오 블록을 필터링한다. (코딩 루프 중에 또는 코딩 루프 이후에) 다른 루프 필터들이 또한 픽셀 전환들 (pixel transitions) 을 평활화하거나 또는 아니면 비디오 품질을 향상시키기 위해 사용될 수도 있다. 주어진 프레임 또는 픽처에서의 디코딩된 비디오 블록들은 그후 DPB (94) 에 저장되며, 이 메모리는 후속 모션 보상을 위해 사용되는 참조 픽처들을 저장한다. DPB (94) 는 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에의 추후 프리젠테이션을 위해, 디코딩된 비디오를 저장하는 추가적인 메모리의 일부이거나 또는 그와 분리될 수도 있다.

[0171] 비디오 디코더 (30) 는 서브-블록 기반의 OBMC, 서브-PU 기반의 OBMC, CU-경계 기반의 OBMC, 또는 이들의 임의의 조합을 수행하도록 구성된 비디오 디코더의 일 예를 나타낸다. 예를 들어, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록, 즉, 이 예에서는, 예측 유닛의 서브-블록을 수신하도록 구성될 수도 있다. 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 수신하고, 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 적어도 하나의 모션 벡터를 결정하고, 그리고 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 적어도 하나의 모션 벡터에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 디코딩하도록 구성될 수도 있다.

- [0172] 도 11 은 본 개시물의 기법들에 따른, 비디오 데이터를 디코딩하는 예시적인 프로세스를 예시하는 플로우차트이다. 도 11 의 프로세스는 예시의 목적을 위해 비디오 디코더 (30) 에 의해 수행되는 것으로 일반적으로 설명되지만, 다양한 다른 프로세서들이 또한 도 11 에 나타난 프로세스를 실행할 수도 있다. 일부 예들에서, 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 및/또는 예측 프로세싱 유닛 (81) 은 도 11 에 나타난 하나 이상의 프로세스들을 수행할 수도 있다.
- [0173] 도 11 의 예에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록을 수신하도록 구성될 수도 있다 (200). 일부 예들에서, 비디오 데이터의 제 1 블록은 예측 유닛의 서브-블록이다. 일부 예들에서, 비디오 데이터의 제 1 블록은 예측 유닛의 정사각형 서브-블록 또는 비-정사각형 서브-블록이다.
- [0174] 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 수신하도록 구성될 수도 있다 (202). 일부 예들에서, 비디오의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들은 정사각형 서브-블록들, 비-정사각형 서브-블록들, 또는 적어도 하나의 정사각형 서브-블록과 적어도 하나의 비-정사각형 서브-블록의 조합이다.
- [0175] 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보를 결정하도록 구성될 수도 있다 (204). 일부 예들에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 블록을 디코딩함으로써 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 적어도 하나의 모션 벡터를 결정하도록 구성될 수도 있다. 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 모션 정보에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 디코딩하도록 구성될 수도 있다 (206).
- [0176] 일부 예들에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나가 디코딩되지 않는지 여부를 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 는 디코딩되지 않는 것으로 결정된 하나 이상의 블록들 중 적어도 하나의 블록에 대응하는 모션 정보를 이용함이 없이, 비디오 데이터의 제 1 블록을 디코딩하도록 구성될 수도 있다.
- [0177] 일부 예들에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 대한 제 1 예측 블록을 발생시키도록 구성될 수도 있다. 제 1 예측 블록은 비디오 데이터의 제 1 블록에 대한 각각의 픽셀 위치에 대한 값을 포함할 수도 있다. 비디오 디코더 (30) 는 비디오의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나에 대한 이웃 예측 블록을 발생시키도록 구성될 수도 있다. 이웃 예측 블록은 비디오의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나에 대한 각각의 픽셀 위치에 대한 값을 포함할 수도 있다. 비디오 디코더 (30) 는 이웃 예측 블록의 하나 이상의 값들에 기초하여 제 1 예측 블록의 하나 이상의 값들을 수정하여, 수정된 제 1 예측 블록을 발생시키도록 구성될 수도 있다. 비디오 디코더 (30) 는 수정된 제 1 예측 블록을 이용하여 비디오 데이터의 제 1 블록을 디코딩하도록 구성될 수도 있다.
- [0178] 일부 예들에서, 비디오 디코더 (30) 는 제 1 가중된 값을 제 1 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 디코더 (30) 는 제 2 가중된 값을 이웃 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 디코더 (30) 는 제 1 가중된 값 및 제 2 가중된 값에 기초하여 제 1 예측 블록의 하나 이상의 값들을 수정하여 수정된 제 1 예측 블록을 발생시키도록 구성될 수도 있다.
- [0179] 일부 예들에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 대한 제 1 예측 블록을 발생시키도록 구성될 수도 있다. 제 1 예측 블록은 비디오 데이터의 제 1 블록에 대한 각각의 픽셀 위치에 대한 값을 포함할 수도 있다. 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록의 사이즈에 기초하여, 하나 이상의 가중된 값들을 제 1 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 디코더 (30) 는 제 1 예측 블록의 하나 이상의 값들에 할당된 하나 이상의 가중된 값들을 이용하여 비디오 데이터의 제 1 블록을 디코딩하도록 구성될 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보에 기초하여, 제 1 이웃 예측 블록을 발생시키도록 구성될 수도 있다. 비디오 디코더 (30) 는 제 1 이웃 블록의 사이즈에 기초하여, 하나 이상의 가중된 값들을 제 1 이웃 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 디코더 (30) 는 제 1 이웃 예측 블록의 하나 이상의 값들에 할당된 하나 이상의 가중된 값들을 이용하여, 비디오 데이터의 제 1 블록을 디코딩하도록 구성될 수도 있다.
- [0180] 일부 예들에서, 비디오 디코더 (30) 는 제 1 이웃 예측 블록의 하나 이상의 값들에 할당된 하나 이상의 가중된

값들에 기초하여 제 1 예측 블록의 하나 이상의 값들을 수정하여 수정된 제 1 예측 블록을 발생시키고 수정된 제 1 예측 블록을 이용하여 비디오 데이터의 제 1 블록을 디코딩하도록 구성될 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보에 기초하여, 제 2 이웃 예측 블록을 발생시키도록 구성될 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 는 제 2 이웃 블록의 사이즈에 기초하여, 하나 이상의 가중된 값들을 제 2 이웃 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 디코더 (30) 는 제 1 및 제 2 이웃 예측 블록들의 하나 이상의 값들에 할당된 하나 이상의 가중된 값들에 기초하여 제 1 및 제 2 이웃 예측 블록들의 하나 이상의 값들을 수정하여 수정된 제 1 예측 블록을 발생시키도록 구성될 수도 있다.

[0181] 일부 예들에서, 비디오 디코더 (30) 는 비디오 데이터의 제 1 블록이 중첩 블록 모션 보상 모드에 따라서 인코딩되는지 여부를 나타내는 값을 가진 선택스 엘리먼트를 수신하도록 구성될 수도 있다. 예를 들어, 선택스 엘리먼트는 비디오 인코더 (예컨대, 비디오 인코더 (20)) 에 의해 발생된 비트스트림으로 수신될 수도 있다.

비디오 디코더 (30) 는 선택스 엘리먼트의 값에 기초하여, 비디오 데이터의 제 1 블록이 중첩 블록 모션 보상 모드에 따라서 인코딩된다고 결정하도록 구성될 수도 있다.

[0182] 도 12 는 본 개시물의 기법들에 따른, 비디오 데이터를 인코딩하는 예시적인 프로세스를 예시하는 플로우차트이다. 도 12 의 프로세스는 예시의 목적을 위해 비디오 코더에 의해 수행되는 것으로 일반적으로 설명되지만, 다양한 다른 프로세서들이 또한 도 12 에 나타난 프로세스를 실행할 수도 있다. 본원에서 사용될 때, 비디오 코더는 비디오 인코더 및/또는 비디오 디코더 (예컨대, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)) 를 지칭할 수도 있다. 일부 예들에서, 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 및/또는 엔트로피 인코딩 유닛 (56) 은 도 12 에 나타난 하나 이상의 프로세스들을 수행할 수도 있다.

일부 예들에서, 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 및/또는 예측 프로세싱 유닛 (81) 은 도 12 에 나타난 하나 이상의 프로세스들을 수행할 수도 있다.

[0183] 도 12 의 예에서, 비디오 코더는 비디오 데이터의 제 1 블록을 메모리에 저장하도록 구성될 수도 있다 (220).

일부 예들에서, 비디오의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들은 정사각형 서브-블록들, 비-정사각형 서브-블록들, 또는 적어도 하나의 정사각형 서브-블록과 적어도 하나의 비-정사각형 서브-블록의 조합이다.

[0184] 비디오 코더는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들을 메모리에 저장하도록 구성될 수도 있다 (222). 일부 예들에서, 비디오의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들은 정사각형 서브-블록들, 비-정사각형 서브-블록들, 또는 적어도 하나의 정사각형 서브-블록과 적어도 하나의 비-정사각형 서브-블록의 조합이다.

[0185] 비디오 코더는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보를 결정하도록 구성될 수도 있다 (224). 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 블록을 코딩함으로써, 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 적어도 하나의 모션 벡터를 결정하도록 구성될 수도 있다. 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록에 이웃하는 하나 이상의 블록들 중 적어도 하나의 모션 정보에 적어도 부분적으로 기초하여, 비디오 데이터의 제 1 블록을 중첩 블록 모션 보상을 이용하여 코딩하도록 구성될 수도 있다 (226).

[0186] 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나가 코딩되지 않는지 여부를 결정하도록 구성될 수도 있다. 이러한 예들에서, 비디오 코더는 코딩되지 않는 것으로 결정된 하나 이상의 블록들 중 적어도 하나의 블록에 대응하는 모션 정보를 이용하지 않으므로써, 비디오 데이터의 제 1 블록을 코딩하도록 구성될 수도 있다.

[0187] 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록에 대한 제 1 예측 블록을 발생시키도록 구성될 수도 있다. 제 1 예측 블록은 비디오 데이터의 제 1 블록에 대한 각각의 픽셀 위치에 대한 값을 포함할 수도 있다. 비디오 코더는 비디오의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나에 대한 이웃 예측 블록을 발생시키도록 구성될 수도 있다. 이웃 예측 블록은 비디오의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나에 대한 각각의 픽셀 위치에 대한 값을 포함할 수도 있다. 비디오 코더는 이웃 예측 블록의 하나 이상의 값들에 기초하여 제 1 예측 블록의 하나 이상의 값들을 수정하여 수정된 제 1 예측 블록을 발생시키도록 구성될 수도 있다. 비디오 코더는 수정된 제 1 예측 블록을 이용하여 비디오 데이터의 제 1 블록을 코딩하도록 구성될 수도 있다.

- [0188] 일부 예들에서, 비디오 코더는 제 1 가중된 값을 제 1 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 코더는 제 2 가중된 값을 이웃 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 코더는 제 1 가중된 값 및 제 2 가중된 값에 기초하여 제 1 예측 블록의 하나 이상의 값들을 수정하여 수정된 제 1 예측 블록을 발생시키도록 구성될 수도 있다.
- [0189] 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록에 대한 제 1 예측 블록을 발생시키도록 구성될 수도 있다. 제 1 예측 블록은 비디오 데이터의 제 1 블록에 대한 각각의 픽셀 위치에 대한 값을 포함할 수도 있다. 비디오 코더는 비디오 데이터의 제 1 블록의 사이즈에 기초하여, 하나 이상의 가중된 값들을 제 1 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 코더는 제 1 예측 블록의 하나 이상의 값들에 할당된 하나 이상의 가중된 값들을 이용하여 비디오 데이터의 제 1 블록을 코딩하도록 구성될 수도 있다. 이러한 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보에 기초하여, 제 1 이웃 예측 블록을 발생시키도록 구성될 수도 있다. 비디오 코더는 제 1 이웃 블록의 사이즈에 기초하여, 하나 이상의 가중된 값들을 제 1 이웃 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 코더는 제 1 이웃 예측 블록의 하나 이상의 값들에 할당된 하나 이상의 가중된 값들을 이용하여 비디오 데이터의 제 1 블록을 코딩하도록 구성될 수도 있다.
- [0190] 일부 예들에서, 비디오 코더는 제 1 이웃 예측 블록의 하나 이상의 값들에 할당된 하나 이상의 가중된 값들에 기초하여 제 1 예측 블록의 하나 이상의 값들을 수정하여 수정된 제 1 예측 블록을 발생시키고, 수정된 제 1 예측 블록을 이용하여 비디오 데이터의 제 1 블록을 코딩하도록 구성될 수도 있다. 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록에 이웃하는 비디오 데이터의 하나 이상의 블록들 중 적어도 하나의 모션 정보에 기초하여, 제 2 이웃 예측 블록을 발생시키도록 구성될 수도 있다. 이러한 예들에서, 비디오 코더는 제 2 이웃 블록의 사이즈에 기초하여, 하나 이상의 가중된 값들을 제 2 이웃 예측 블록의 하나 이상의 값들에 할당하도록 구성될 수도 있다. 비디오 코더는 제 1 및 제 2 이웃 예측 블록들의 하나 이상의 값들에 할당된 하나 이상의 가중된 값들에 기초하여 제 1 및 제 2 예측 블록들의 하나 이상의 값들을 수정하여 수정된 제 1 예측 블록을 발생시키도록 구성될 수도 있다.
- [0191] 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록이 중첩 블록 모션 보상 모드에 따라서 인코딩되는지 여부를 나타내는 값을 가진 선택스 엘리먼트를 수신하도록 구성될 수도 있다. 예를 들어, 선택스 엘리먼트는 비디오 인코더 (예컨대, 비디오 인코더 (20)) 에 의해 발생된 비트스트림으로 수신될 수도 있다. 비디오 코더는 선택스 엘리먼트의 값에 기초하여, 비디오 데이터의 제 1 블록이 중첩 블록 모션 보상 모드에 따라서 인코딩된다고 결정하도록 구성될 수도 있다.
- [0192] 일부 예들에서, 비디오 코더는 비디오 데이터의 제 1 블록이 중첩 블록 모션 보상 모드에 따라서 인코딩되는지 여부를 나타내는 값을 가진 선택스 엘리먼트를 발생시키도록 구성될 수도 있다. 이러한 예들에서, 비디오 코더는 선택스 엘리먼트를, 비디오 디코더 (예컨대, 비디오 디코더 (30)) 에 의해 수신될 수도 있는 비트스트림으로 송신하도록 구성될 수도 있다.
- [0193] 본원에서 설명되는 기법들 모두가 개별적으로 또는 조합하여 사용될 수도 있는 것으로 이해되어야 한다. 예를 들어, 비디오 인코더 (20) 및/또는 하나 이상의 그의 구성요소들 및 비디오 디코더 (30) 및/또는 하나 이상의 그의 구성요소들은 본 개시물에서 설명하는 기법들을 임의의 조합으로 수행할 수도 있다. 다른 예로서, 본원에서 설명되는 기법들은 비디오 코더로서 일반적으로 둘다 지칭될 수도 있는, 비디오 인코더 (20) (도 1 및 도 9) 및/또는 비디오 디코더 (30) (도 1 및 도 10) 에 의해 수행될 수도 있다. 이와 유사하게, 비디오 코딩은 적용가능한 경우, 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.
- [0194] 그 예에 따라서, 본원에서 설명되는 기법들 중 임의의 기법의 어떤 행위들 또는 이벤트들이 상이한 시퀀스로 수행될 수 있으며, 추가되거나, 병합되거나, 또는 모두 제외시킬 수도 있는 (예컨대, 모든 설명되는 행위들 또는 이벤트들이 기법들의 실시예에 필수적인 것은 아님) 것으로 인식되어야 한다. 더욱이, 어떤 예들에서, 행위들 또는 이벤트들은 순차적으로 보다는, 동시에, 예컨대, 멀티-쓰레드된 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해서 수행될 수도 있다. 게다가, 본 개시물의 어떤 양태들이 명료성의 목적들을 위해 단일 모듈 또는 유닛에 의해 수행되는 것으로 설명되지만, 본 개시물의 기법들은 비디오 코더와 연관되는 유닛들 또는 모듈들의 결합에 의해 수행될 수도 있는 것으로 이해되어야 한다.
- [0195] 본 개시물의 어떤 양태들이 예시의 목적을 위해, 하나 이상의 비디오 코딩 표준들에 대해, 공개되었거나, 개발 중이거나, 또는 기타 등등이든, 설명되었다. 그러나, 본 개시물에서 설명하는 기법들은 아직 개발되지 않은

다른 표준 또는 독점 비디오 코딩 프로세스들을 포함한, 다른 비디오 코딩 프로세스들에 유용할 수도 있다.

[0196] 본 개시물에 따르면, 용어 "또는"은 문맥이 달리 지시하지 않는 한 "및/또는"로서 중단될 (interrupted) 수도 있다. 게다가, "하나 이상의" 또는 "적어도 하나의" 또는 기타 등등과 같은 어구들이 본원에서 개시된 일부 특징들에 대해 사용되었지만 다른 것들에 대해서는 사용되지 않았을 수도 있다; 이러한 용어가 사용되지 않은 특징들은 문맥이 달리 지시하지 않는 한 그러한 의미를 내포하는 것으로 해석될 수도 있다.

[0197] 본 기법들의 여러 양태들의 특징의 조합들이 위에서 설명되지만, 이들 조합들은 단지 본 개시물에서 설명하는 기법들의 예를 예시하기 위해 제공된다. 따라서, 본 개시물의 기법들은 이들 예시적인 조합들에 한정되지 않아야 하며 본 개시물에서 설명하는 기법들의 여러 양태들의 임의의 상상가능한 조합을 포괄할 수도 있다.

[0198] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 또는 컴퓨터-판독가능 매체를 통해서 송신될 수도 있으며, 하드웨어-기반의 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터-판독가능 매체는 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터-판독가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라서 한 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방법으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비일시성 유형의 컴퓨터-판독가능 저장 매체, 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명하는 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독가능 매체를 포함할 수도 있다.

[0199] 일 예로서, 이에 한정하지 않고, 이런 컴퓨터-판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광디스크 스토리지, 자기디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터-판독가능 매체로 적절히 지칭된다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 무선 기술들, 예컨대 적외선, 라디오, 및 마이크로파를 이용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들 예컨대 적외선, 라디오, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체 및 데이터 저장 매체는 접속부들, 반송파들, 신호들, 또는 다른 일시성 매체를 포함하지 않고, 그 대신, 비-일시성 유형의 저장 매체로 송신되는 것으로 해석되어야 한다.

디스크 (disk) 및 디스크 (disc)는, 본원에서 사용할 때, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루-레이 디스크를 포함하며, 디스크들 (disks)은 데이터를 자기적으로 보통 재생하지만, 디스크들 (discs)은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 결합들이 또한 컴퓨터-판독가능 매체들의 범위 내에 포함되어야 한다.

[0200] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서"는, 본원에서 사용될 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다.

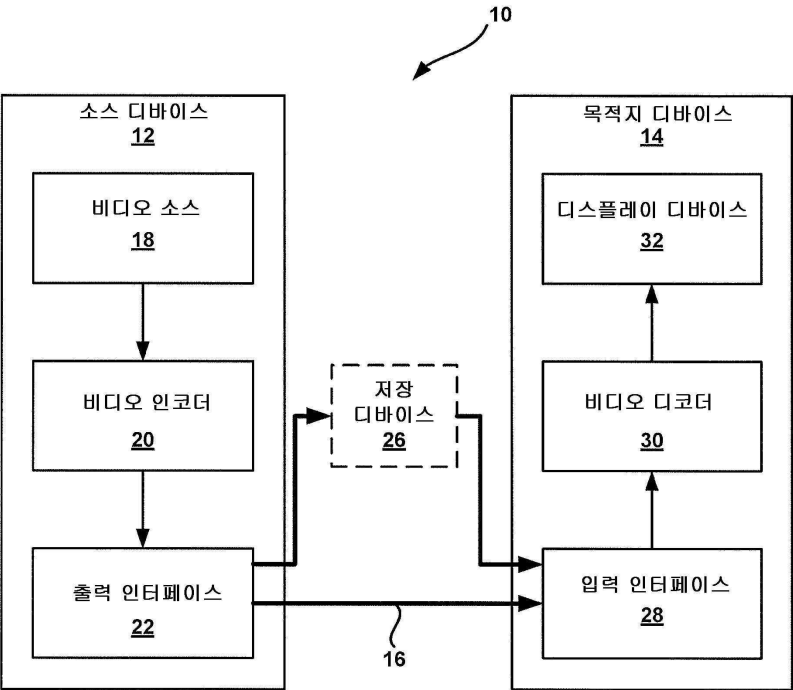
게다가, 일부 양태들에서, 본원에서 설명하는 기능은 전용 하드웨어 및/또는 인코딩 및 디코딩을 위해 구성되는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.

[0201] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트)를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 여러 구성요소들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 대신, 위에서 설명한 바와 같이, 여러 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

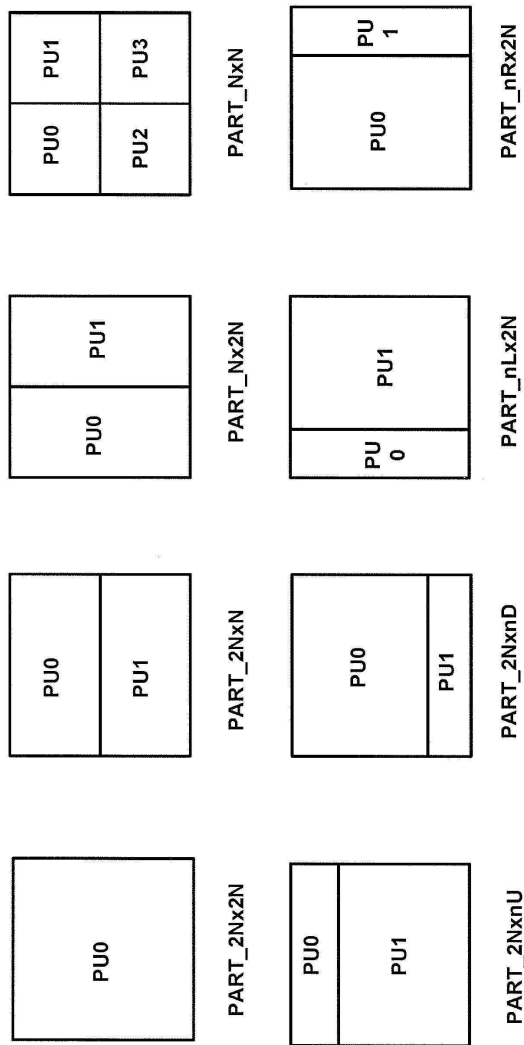
[0202] 여러 예들이 본원에서 설명되었다. 설명된 시스템들, 동작들, 기능들, 또는 예들의 임의의 조합이 고려된다. 이들 및 다른 예들은 다음 청구항들의 범위 이내이다.

도면

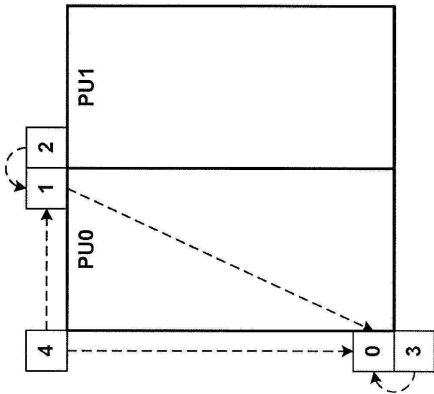
도면1



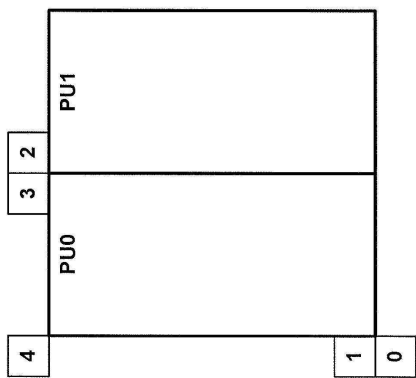
도면2



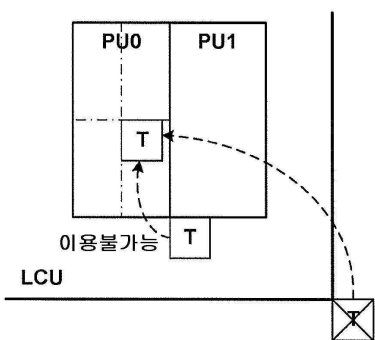
도면3a



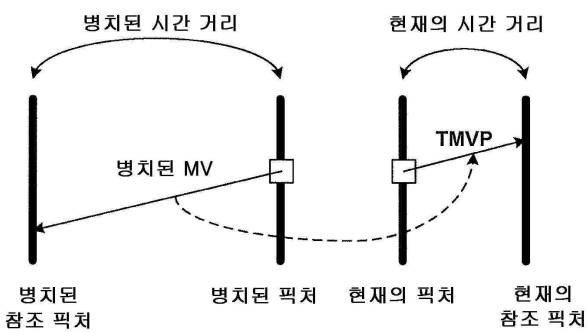
도면3b



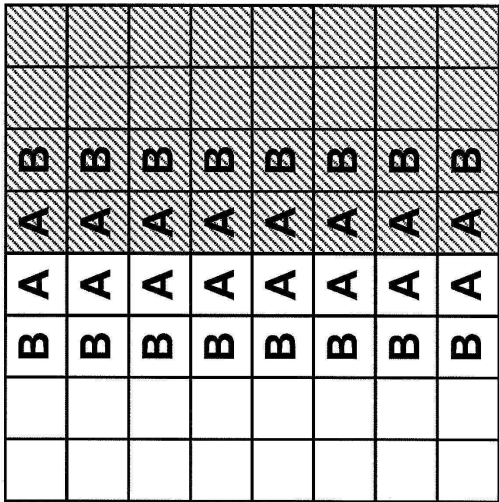
도면4a



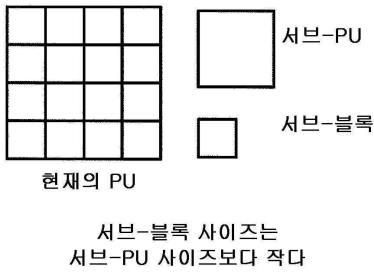
도면4b



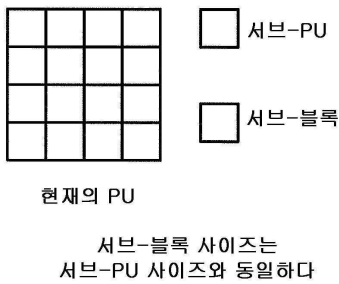
도면6b



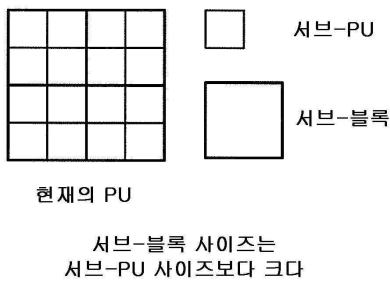
도면7a



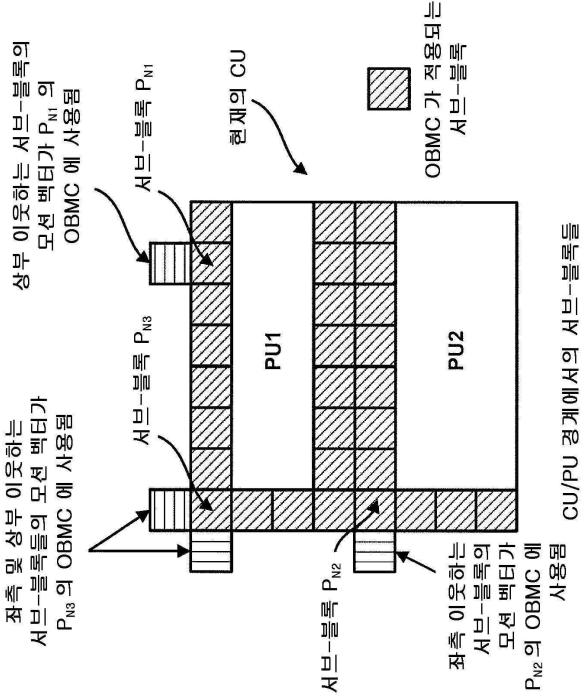
도면7b



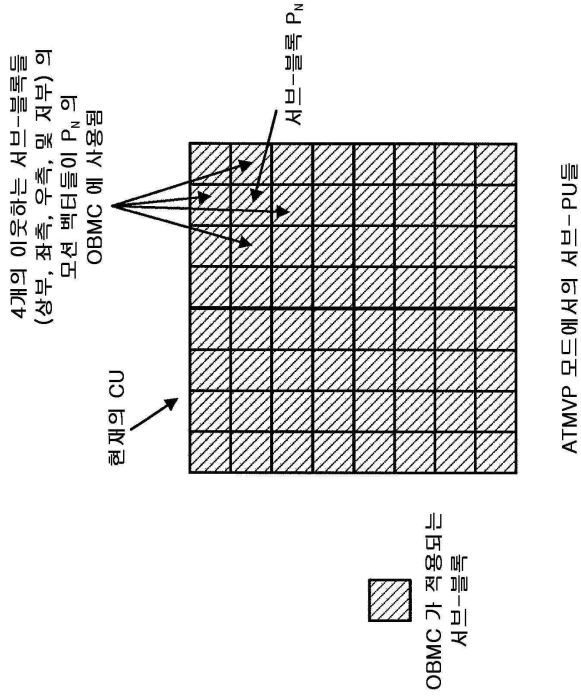
도면7c



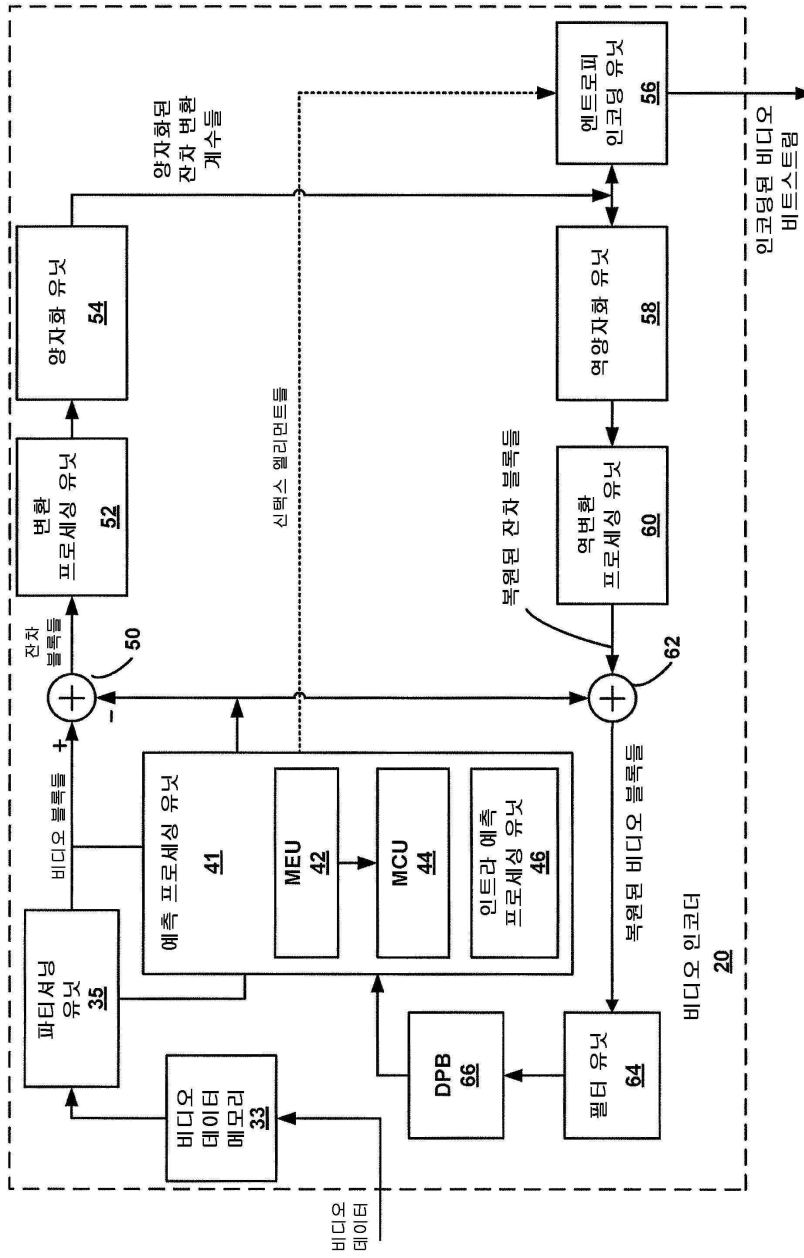
도면8a



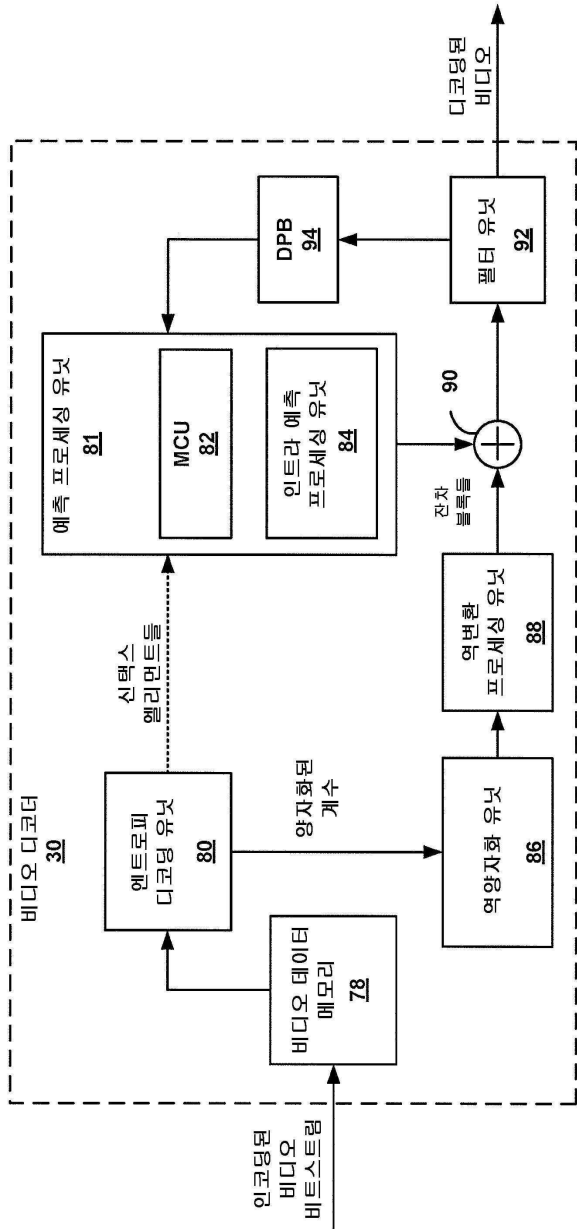
도면8b



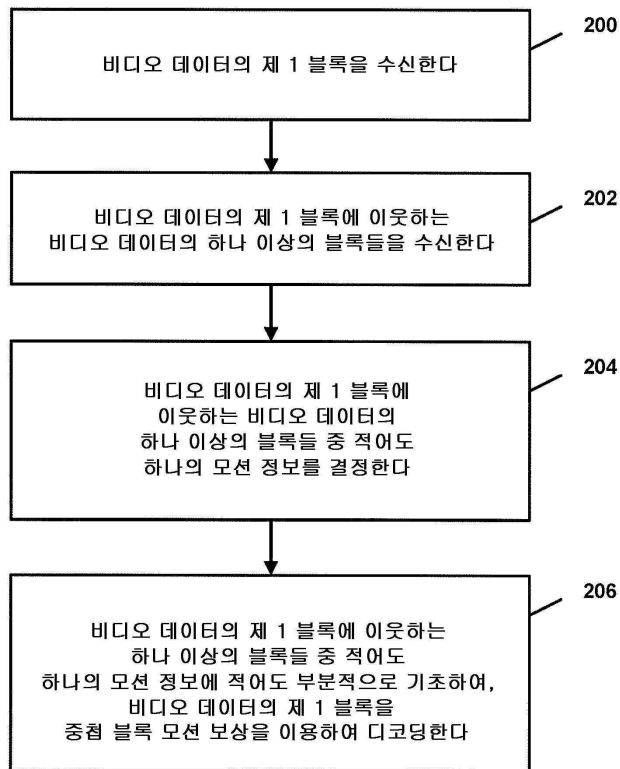
도면9



도면10



도면11



도면12

