(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년10월07일
(11) 등록번호 10-2310752
(24) 등록일자 2021년10월01일

(51) 국제특허분류(Int. Cl.)
H04N 19/11 (2014.01) *H04N 19/139* (2014.01)
H04N 19/174 (2014.01) *H04N 19/513* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/11 (2015.01)
H04N 19/139 (2015.01)
(21) 출원번호 10-2017-7034125
(22) 출원일자(국제) 2016년05월20일
심사청구일자 2021년02월23일
(85) 번역문제출일자 2017년11월24일
(65) 공개번호 10-2018-0013918
(43) 공개일자 2018년02월07일
(86) 국제출원번호 PCT/US2016/033553
(87) 국제공개번호 WO 2016/196043
국제공개일자 2016년12월08일
(30) 우선권주장
62/168,396 2015년05월29일 미국(US)
15/158,741 2016년05월19일 미국(US)
(56) 선행기술조사문헌
LI B et al: "Non-SCCE1: Unification of intra BC and inter modes", JCTVC-R0100, 20 June 2014.
Xu X et al: "On unification of intra block copy and inter-picture motion compensation", JCTVC-Q0132, 3 April 2014.

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
세레긴 바딤
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
라파카 크리쉬나칸트
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 11 항 심사관 : 김영태

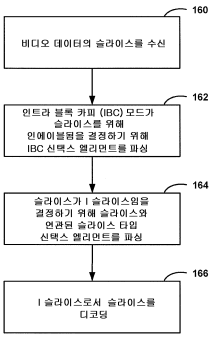
(54) 발명의 명칭 **슬라이스 레벨 인트라 블록 카피 및 기타 비디오 코딩 개선**

(57) 요 약

비디오 데이터를 디코딩하기 위한 디바이스로서, 상기 비디오 데이터를 저장하도록 구성된 메모리 및 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은 상기 비디오 데이터의 슬라이스를 수신하고, 인트라 블록 카피 (IBC) 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신텍스 엘리먼트를 파싱하고, 상

(뒷면에 계속)

대 표 도 - 도6



기 슬라이스가 I 슬라이스임을 결정하기 위해 상기 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하고, 그리고 인트라 예측 코딩 모드를 사용하여 슬라이스의 모든 블록을 디코딩하는 것에 의해 슬라이스를 I 슬라이스로서 디코딩하도록 구성된다.

(52) CPC특허분류
H04N 19/174 (2015.01)
H04N 19/521 (2015.01)
H04N 19/70 (2015.01)

(72) 발명자
시에 청-데
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
조쉬 라잔 랙스맨
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
왕 예-쿠이
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
팡 차오
미국 90292 캘리포니아주 마리나 델 레이 비아 마리나 4169 아파트먼트 307에이
카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명 세 서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터의 현재 화상의 슬라이스를 수신하는 단계;

인트라 블록 카피 (IBC) 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하는 단계 (162);

비디오 데이터의 상기 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수를 표시하는 신택스 엘리먼트를 수신하는 단계;

상기 참조 화상 리스트 내의 상기 참조 화상들의 수를 표시하는 상기 신택스 엘리먼트가 1 과 동일한 것 및 상기 IBC 모드가 상기 슬라이스에 대해 인에이블된다는 상기 결정에 응답하여, 참조 화상 리스트 수정 정보를 수신하지 않고, 상기 참조 화상 리스트 내의 제 1 엔트리로서 상기 슬라이스를 포함하는 상기 현재 화상을 추가함으로써, 상기 비디오 데이터의 상기 슬라이스에 대한 상기 참조 화상 리스트를 구축하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

시퀀스 파라미터 세트 (SPS) 에서 상기 IBC 신택스 엘리먼트를 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

디코딩하는 것은,

상기 비디오 데이터의 상기 현재 화상의 제 2 슬라이스를 수신하는 단계 (160; 170);

인트라 블록 카피 (IBC) 모드가 상기 제 2 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하는 단계 (162);

상기 제 2 슬라이스가 I 슬라이스임을 결정하기 위해 상기 제 2 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하는 단계 (164);

상기 제 2 슬라이스를 I 슬라이스로서 디코딩하는 단계 (166) 로서, 상기 제 2 슬라이스를 I 슬라이스로서 디코딩하는 단계는 적어도 하나의 인트라 예측 코딩 모드를 사용하여 상기 제 2 슬라이스의 모든 블록들을 디코딩하는 단계를 포함하는, 상기 제 2 슬라이스를 I 슬라이스로서 디코딩하는 단계 (166) 를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 3 항에 있어서,

상기 제 2 슬라이스를 I 슬라이스로서 디코딩하는 단계는 인트라 예측만을 사용하여 상기 제 2 슬라이스의 블록들을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 IBC 신택스 엘리먼트는 curr_pic_as_ref_enabled_flag 신택스 엘리먼트를 포함하고, 상기 IBC 모드가 상

기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하는 단계는, 상기 curr_pic_as_ref_enabled_flag 신택스 엘리먼트의 값이 1 과 동일함을 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 3 항에 있어서,

상기 슬라이스 타입 신택스 엘리먼트는 slice_type 신택스 엘리먼트를 포함하고, 상기 제 2 슬라이스가 I 슬라이스임을 결정하기 위해 상기 제 2 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하는 단계는, 상기 slice_type 신택스 엘리먼트의 값이 2 와 동일함을 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 7

비디오 데이터를 디코딩하기 위한 장치로서,

상기 비디오 데이터의 현재 화상의 슬라이스를 수신하는 수단;

인트라 블록 카피 (IBC) 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하는 수단;

상기 비디오 데이터의 상기 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수를 표시하는 신택스 엘리먼트를 수신하는 수단;

상기 참조 화상 리스트 내의 상기 참조 화상들의 수가 1 과 동일한 값인 것 및 상기 IBC 모드가 상기 슬라이스에 대해 인에이블된다는 상기 결정에 응답하여, 참조 화상 리스트 수정 정보를 수신하지 않고, 상기 비디오 데이터의 상기 슬라이스에 대한 상기 참조 화상 리스트 내의 제 1 엔트리로서 상기 슬라이스를 포함하는 상기 현재 화상을 추가함으로써, 상기 참조 화상 리스트를 구축하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 8

제 7 항에 있어서,

상기 IBC 신택스 엘리먼트는 curr_pic_as_ref_enabled_flag 신택스 엘리먼트를 포함하고, 상기 IBC 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하는 수단은, 상기 curr_pic_as_ref_enabled_flag 신택스 엘리먼트의 값이 1 과 동일함을 결정하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 9

제 8 항에 있어서,

상기 비디오 데이터의 상기 현재 화상의 제 2 슬라이스를 수신하는 수단;

인트라 블록 카피 (IBC) 모드가 상기 제 2 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하는 수단;

상기 제 2 슬라이스가 I 슬라이스임을 결정하기 위해 상기 제 2 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하는 수단;

상기 제 2 슬라이스를 I 슬라이스로서 디코딩하는 수단으로서, 상기 제 2 슬라이스를 I 슬라이스로서 디코딩하는 수단은 적어도 하나의 인트라 예측 코딩 모드를 사용하여 상기 제 2 슬라이스의 모든 블록들을 디코딩하는 수단을 포함하는, 상기 제 2 슬라이스를 I 슬라이스로서 디코딩하는 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 10

제 9 항에 있어서,

상기 슬라이스 타입 신택스 엘리먼트는 slice_type 신택스 엘리먼트를 포함하고, 상기 제 2 슬라이스가 I 슬라

이스임을 결정하기 위해 상기 제 2 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하는 수단은, 상기 slice_type 신택스 엘리먼트의 값이 2 와 동일함을 결정하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 11

하나 이상의 프로세서들에 의해 실행될 경우, 상기 하나 이상의 프로세서들로 하여금, 제 1 항 내지 제 6 항 중 어느 한 항에 기재된 비디오 데이터를 디코딩하는 방법을 수행하게 하는 명령들을 저장하는 컴퓨터 판독가능 저장 매체.

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

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

기 술 분 야

[0001] 본원은 2015년 5월 29일자로 출원된 U.S. 가특허출원 62/168,396 의 혜택을 주장하며, 그의 전체 내용은 참조에 의해 본원에 포함된다.

[0002] 기술 분야

[0003] 본 개시는 비디오 코딩에 관한 것이다.

배 경 기 술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기 (PDA) 들, 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, 전자책 단말기, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은, MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding), ITU-T H.265 에 의해 정의되는 표준들, HEVC (High Efficiency Video Coding) 표준, 및 그러한 표준들의 확장들에서 설명된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은, 그러한 비디오 압축기법들을 구현함으로써 보다 효율적으로 디지털 비디오 정보를 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은, 비디오 시퀀스들에 내재하는 중복성 (redundancy) 을 감소시키거나 또는 제거하기 위해 공간 (인트라-화상) 예측 및/또는 시간 (인터-화상) 예측을 수행한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (즉, 비디오 프레임, 또는 비디오 프레임의 일부) 는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 트리 블록들, 코딩 유닛 (CU) 들, 및/또는 코딩 노드들로도 지칭될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 화상에서 이웃 블록들에 있는 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오블록들은 동일한 화상에서의 이웃 블록들에 있는 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수도 있고, 참조 화상들은 참조 프레임들로 지칭될 수도 있다.

[0006] 공간 또는 시간 예측은 코딩될 블록을 위한 예측 블록을 낳는다. 잔차 데이터는 코딩될 원래 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터와, 코딩된 블록과 예측 블록 사이의 차이를 표시하는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터

는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 변환 계수들을 낳을 수도 있으며, 다음으로 이들은 양자화될 수도 있다. 초기에 2차원 행렬로 배열된, 양자화된 변환 계수들은,변환 계수들의 1차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩이 적용되어 훨씬 더 많은 압축을 달성할 수도 있다.

발명의 내용

[0007] 개요

[0008] 본 개시는 기존의 비디오 코딩 기술을 향상시킬 수도 있는 기술, 그리고 보다 구체적으로는 기존의 인트라 블록 카피 (IBC)-관련 비디오 코딩 기술을 개선할 수도 있는 기술을 설명한다. 본 개시의 기술들은, 일부 코딩 시나리오들에서, 예를 들어, 슬라이스 레벨에서 IBC를 인에이블함으로써 IBC의 사용을 더 세밀하게 제어할 수 있다.

[0009] 하나의 예에서는, 비디오 데이터를 디코딩하기 위한 방법은, 상기 비디오 데이터의 슬라이스를 수신하는 단계; 인트라 블록 카피 (IBC) 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 (IBC 신택스 엘리먼트를 파싱하는 단계; 상기 슬라이스가 I 슬라이스임을 결정하기 위해 상기 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하는 단계; 및 I 슬라이스로서 상기 슬라이스를 디코딩하는 단계를 포함하고, I 슬라이스로서 상기 슬라이스를 디코딩하는 단계는 적어도 하나의 인트라 예측 코딩 모드를 사용하여 상기 슬라이스의 모든 블록들을 디코딩하는 단계를 포함한다.

[0010] 또 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는, 상기 비디오 데이터를 저장하도록 구성된 메모리 및 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은 상기 비디오 데이터의 슬라이스를 수신하고; 인트라 블록 카피 (IBC) 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하고; 상기 슬라이스가 I 슬라이스임을 결정하기 위해 상기 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하고; 그리고 I 슬라이스로서 상기 슬라이스를 디코딩하도록 구성되고, I 슬라이스로서 상기 슬라이스를 디코딩하기 위하여, 상기 하나 이상의 프로세서들은 적어도 하나의 인트라 예측 코딩 모드를 사용하여 상기 슬라이스의 모든 블록들을 디코딩하도록 구성된다.

[0011] 다른 예에서, 비디오 데이터를 디코딩하기 위한 장치는 상기 비디오 데이터의 슬라이스를 수신하는 수단; 인트라 블록 카피 (IBC) 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하는 수단; 상기 슬라이스가 I 슬라이스임을 결정하기 위해 상기 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하는 수단; 및 I 슬라이스로서 상기 슬라이스를 디코딩하는 수단을 포함하고, I 슬라이스로서 상기 슬라이스를 디코딩하는 수단은 적어도 하나의 인트라 예측 코딩 모드를 사용하여 상기 슬라이스의 모든 블록들을 디코딩하는 수단을 포함한다.

[0012] 다른 예에서, 컴퓨터 판독가능 저장 매체는 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들로 하여금, 상기 비디오 데이터의 슬라이스를 수신하고; 인트라 블록 카피 (IBC) 모드가 상기 슬라이스에 대해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트를 파싱하고; 상기 슬라이스가 I 슬라이스임을 결정하기 위해 상기 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하고; 그리고 I 슬라이스로서 상기 슬라이스를 디코딩하게 하는 명령을 저장하고, I 슬라이스로서 상기 슬라이스를 디코딩하기 위하여, 상기 하나 이상의 프로세서들은 적어도 하나의 인트라 예측 코딩 모드를 사용하여 상기 슬라이스의 모든 블록들을 디코딩한다.

[0013] 본 개시의 하나 이상의 예들의 상세들은 첨부 도면 및 아래의 설명에 제시되어 있다. 다른 특징, 목적 및 이점들은 상세한 설명, 도면, 및 청구항들로부터 분명해질 것이다.

도면의 간단한 설명

[0014] 도 1 은 본 개시에 기재된 기법들을 이용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2는 본 개시의 기법들에 따라 현재 화상 내의 비디오 데이터의 현재 블록을 예측하기 위한 현재 화상 내의 비디오 데이터의 일 예의 예측 블록을 예시하는 개념도이다.

도 3 는 본 개시에 기재된 기술들을 구현할 수도 있는 일 예의 비디오 인코더를 예시하는 블록도이다.

도 4 은 본 개시에 기재된 기술들을 구현할 수도 있는 일 예의 비디오 디코더를 예시하는 블록도이다.

도 5는 본 개시의 기술들에 따른 비디오 데이터를 인코딩하는 방법을 보여주는 플로우차트이다.

도 6은 본 개시의 기술들에 따른 비디오 데이터를 디코딩하는 방법을 보여주는 플로우차트이다.

도 7은 본 개시의 기술들에 따른 비디오 데이터를 디코딩하는 방법을 보여주는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0015] 상세한 설명

[0016] 비디오 시퀀스는 일반적으로 화상들의 시퀀스로서 표현된다. 통상적으로, 블록 기반 코딩 기법들은 개개의 화상들의 각각을 코딩하는데 사용된다. 즉, 각각의 화상은 블록들로 분할되고, 블록들의 각각은 개별적으로 코딩된다. 비디오 데이터의 블록을 코딩하는 것은 일반적으로, 블록을 위한 예측 값을 형성하는 것 및 원래 블록과 예측 값 사이의 차이를 나타내는 잔차 값을 코딩하는 것을 수반한다. 특히, 비디오 데이터의 원래 블록은 픽셀 값들의 행렬을 포함하고, 예측 값은 예측 픽셀 값들의 행렬을 포함한다. 잔차 값은 원래 블록의 픽셀 값과 예측된 픽셀 값 사이의 픽셀 별 차이 (pixel-by-pixel difference) 에 대응하므로, 예측 값에 잔차 값을 가산하는 것은 원래 값에 근사한다.

[0017] 비디오 데이터의 블록을 위한 예측 기법들은 일반적으로, 인트라 예측 또는 인터 예측으로서 카테고리화된다. 인트라 예측, 또는 공간 예측은 일반적으로, 예측되는 블록과 동일한 화상들에서 이전에 코딩된 블록들의 부분인 이웃하는 픽셀 값들로부터 블록을 예측하는 것을 수반한다. 인터 예측, 또는 시간 예측은 일반적으로, 이전에 코딩된 화상들 (예를 들어, 프레임들 또는 슬라이스들) 의 픽셀 값들로부터 블록을 예측하는 것을 수반한다.

[0018] 원격 데스크톱, 원격 게이밍, 무선 디스플레이, 자동차 인포테인먼트 (infotainment), 클라우드 컴퓨팅 및 다른 것들과 같은 많은 응용들이 일상 생활에 있어서 일상화되고 있다. 이들 응용들에 있어서 비디오 콘텐츠는 보통, 자연 콘텐츠, 텍스트, 인공 그래픽, 및 다른 콘텐츠들의 조합이다. 텍스트 및 인공 그래픽 영역들에서, 반복된 패턴들 (이를테면, 문자 (character), 아이콘, 심볼 등) 이 종종 존재한다.

[0019] 인트라 블록 카피 (줄여서 IntraBC 또는 IBC) 그리고 때때로 인트라 모션 보상 (IntraMC 또는 IMC) 으로 지칭될 수도 있는 동일한 화상 내의 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측은, 비디오 코더로 하여금 리던던시 (redundancy) 를 제거하고, 인트라 프레임 코딩 효율성을 개선하는 것을 가능하게 할 수도 있는 기법이다. 통상적인 인트라 예측 코딩에서, 비디오 코더들 (예를 들어, 비디오 인코더 및 비디오 디코더) 은, 현재 비디오 블록의 예측을 위해 동일한 화상 내의 비디오 데이터의 현재 블록의 바로 상부 또는 하부에 있거나 또는 직접 수평적으로 라인을 이루는 이전에 재구성된 비디오 데이터의 블록들을 이용한다.

즉, 비디오 데이터의 화상 또는 프레임이 2-D 그리드 상에 부과되면, 비디오 데이터의 각각의 블록은 x 값 및 y 값들의 고유 범위를 차지한다. 따라서, 일부 비디오 코더들은, x 값들의 동일한 (즉, 현재 비디오 블록과 수직으로 라인을 이루는) 세트, 또는 y 값들의 동일한 (즉, 현재 비디오 블록과 수평적으로 라인을 이루는) 세트만을 공유하는 동일한 화상에서 이전에 코딩된 비디오 데이터의 블록들에 기초하여 비디오 데이터의 현재 블록을 예측할 수도 있다.

[0020] 비디오 코더는, 반드시 비디오 데이터의 현재 블록의 바로 상부 또는 좌측 (또는 바로 우측 또는 하부) 에 있을 필요가 있는 것은 아닌 동일한 프레임 또는 화상 내에 있는 비디오 데이터의 이전에 재구성된 블록으로부터 현재 비디오 블록을 예측하는 것이 유리할 수도 있다. 예측 세트에 더 많은 비디오 블록들을 포함시킴으로써, 비디오 코더는 현재 비디오 블록의 더 정확한 예측을 달성함으로써, 코딩 효율을 증가시킬 수도 있다.

[0021] 일반적으로, 본 개시는, IBC 모드로서 지칭될 수도 있는 동일한 화상 내에 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드를 포함하는 비디오 데이터를 코딩하기 위한 기법들을 설명한다. 본 개시의 IBC 기법들은 비디오 데이터의 현재 블록을 위한 비디오 데이터의 예측 블록을 식별하는 단계를 포함할 수도 있다. 비디오 데이터의 예측 블록은, 예를 들어, 비디오 데이터의 현재 블록과 동일한 화상 내의 비디오 데이터의 재구성된 블록에 대응할 수도 있다. 비디오 데이터의 예측 블록은 화상 내에 있는 의도된 영역 내로부터 나올 수도 있다. 의도된 영역은 또한 IBC 예측 영역으로 지칭될 수도 있다. 의도된 영역은 예를 들어, 비디오 데이터의 현재 블록의 위, 오른쪽 위, 왼쪽 위 및/또는 왼쪽 영역에 대응할 수도 있다. 비디오 데이터의 예측 블록은 현재 비디오 블록의 바로 위 또는 바로 왼쪽 중 어느 한쪽에 있는 것으로 한정되지 않고, 결과적으로 현재 블록에 대하여 예측 블록을 식별하는데 이용된 벡터는 반드시 1차원 벡터인 것은 아니다. 대신에, 비디오 데이터의 예측 블록을 식별 또는 결정하기 위하여, 비디오 코더는, 비디오 데이터의 현재 블록에 대해 수평 변위 성분 및 수직 변위 성분을 포함하는 2차원 벡터를 정의하는 하나 이상의 신택스 엘리먼트들을 코딩할 수도 있다. 2차원 벡터는 블록 벡터, 오프셋 벡터, 또는 모션 벡터로 지칭

될 수도 있고, 예를 들어, 현재 블록의 상단 좌측 코너에 관하여 예측 블록을 식별하는데 이용될 수도 있다.

[0022] 본 개시는 현존하는 비디오 코딩 기술을 향상시킬 수도 있는 기술, 그리고 보다 구체적으로는 기존의 IBC-관련 비디오 코딩 기술을 개선할 수도 있는 기술을 설명한다. 본 개시의 기술들은, 일부 코딩 시나리오들에서, 예를 들어, 슬라이스 레벨에서 IBC를 인에이블함으로써 IBC의 사용을 더 세밀하게 제어할 수 있다. 제안된 기술은 임의의 비트 깊이, 크로마 샘플링 포맷 및/또는 기타 등등과 함께 사용될 수도 있다.

[0023] 여기서 사용된, 용어 "비디오 코더" 는 총칭적으로 비디오 인코더 및 비디오 디코더 양자 모두를 지칭한다. 본 개시에서, 용어 "비디오 코딩" 또는 "코딩" 은 총칭적으로 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다. 따라서, 다르게 언급되지 않으면, 코딩과 관련하여 설명된 기법들은 비디오 인코더 또는 비디오 디코더 중 어느 하나에 의해 수행될 수도 있다는 것이 가정되야 한다. 본원의 일부에서, 어떤 기법들은 비디오 디코딩 또는 비디오 디코더와 관련하여 설명될 수도 있다. 하지만, 그러한 기법들은 비디오 인코딩에 적용가능하지 않거나 또는 비디오 인코더에 의해 수행되지 않을 수도 있다고 가정되어서는 안된다. 예를 들어, 그러한 기법들은 비디오 데이터를 어떻게 인코딩하는지를 결정하는 부분으로서 수행될 수도 있거나 또는 비디오 인코더에서 비디오 디코딩 루프의 부분으로서 수행될 수도 있다.

[0024] 본 개시에서 이용된 바처럼, 용어 현재 블록은, 이미 코딩되었거나 또는 아직 코딩되야 하는 블록과는 대조적으로, 현재 코딩되고 있는 블록을 지칭한다. 마찬가지로, 현재 코딩 유닛, 예측 유닛, 또는 변환 유닛은, 현재 코딩되고 있는 코딩 유닛, 예측 유닛, 또는 변환 유닛을 지칭한다.

[0025] 도 1은 본 개시에 설명된 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1에 도시된 바처럼, 시스템 (10) 은, 목적지 디바이스 (14) 에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크탑 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 전화기 핸드셋 이를테면 소위 "스마트" 폰들, 소위 "스마트" 패드, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 재생기들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 어느 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 갖추어질 수도 있다.

[0026] 목적지 디바이스 (14) 는, 링크 (16) 를 통해 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 는, 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 링크 (16) 는, 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 직접 송신할 수 있게 하기 위한 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는, 무선 통신 프로토콜 등의 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 이를테면, 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는, 로컬 영역 네트워크, 와이드 영역 네트워크, 또는 인터넷과 같은 글로벌 네트워크 등의 패킷 기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터, 스위치, 기지국, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 통신을 가능하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0027] 다르게는, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (26) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (26) 로부터 액세스될 수도 있다. 저장 디바이스 (26) 는, 하드 드라이브, 블루레이 디스크, DVD, CD-ROM, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체 등의 다양한 분산형 또는 로컬적으로 액세스되는 데이터 저장 매체 중 어느 것을 포함할 수도 있다. 다른 예에서, 저장 디바이스 (26) 는, 소스 디바이스 (12) 에 의해 생성되는 인코딩된 비디오를 유지할 수도 있는, 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는, 스트리밍 또는 다운로드를 통해 저장 디바이스 (26) 로부터 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는, 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 예시적인 파일 서버들은, (예를 들어, 웹사이트용) 웹 서버, FTP 서버, NAS (network attached storage) 디바이스, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는, 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은, 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 양자 모두의 조합을 포함할 수도 있다. 저장 디바이스 (26) 로부터

인코딩된 비디오 데이터의 송신은, 스트리밍 송신, 다운로드 송신 또는 양자 모두의 조합일 수도 있다.

[0028] 본 개시의 기법들은 무선 애플리케이션들 또는 세팅들에 반드시 한정되는 것은 아니다. 그 기법들은, 공중 경유 (over-the-air) 텔레비전 브로드캐스트, 케이블 텔레비전 송신, 위성 텔레비전 송신, 예를 들어 인터넷을 통한 스트리밍 비디오 송신, 데이터 저장 매체에의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들 중 어느 것을 지원하는 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅 및/또는 화상 통화등의 애플리케이션들을 지원하기 위하여 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0029] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22) 를 포함한다. 일부 경우들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 소스, 이를테면 비디오 캡처 디바이스, 예를 들어, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스, 및/또는 소스 비디오로서 컴퓨터 그래픽스 데이터를 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 그러한 소스들의 조합을 포함할 수도 있다. 일 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 하지만, 본 개시에 설명된 기법들은, 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 응용들에 적용될 수도 있다.

[0030] 캡처되거나, 미리 캡처되거나 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안적으로), 디코딩 및/또는 플레이백을 위해, 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 나중 액세스를 위해 저장 디바이스 (26) 에 저장될 수도 있다.

[0031] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16) 를 통해 통신되거나, 또는 저장 디바이스 (26) 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩함에 있어서, 비디오 디코더 (30) 와 같은 비디오 디코더에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 신택스 엘리먼트들을 포함할 수도 있다. 그러한 신택스 엘리먼트들은, 통신 매체 상에서 송신되거나, 저장 매체 상에 저장되거나, 또는 파일 서버에 저장되는 인코딩된 비디오 데이터와 함께 포함될 수도 있다.

[0032] 디스플레이 디바이스 (32) 는, 목적지 디바이스 (14) 와 통합되거나 또는 그 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는, 통합된 디스플레이 디바이스를 포함할 수도 있고 또한 외부 디스플레이 디바이스와 인터페이스 접속하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 어느 것을 포함할 수도 있다.

[0033] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최근에 완성된 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 압축 표준에 따라 동작할 수도 있다. 다르게는, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 다르게는 MPEG4, Part 10, AVC (Advanced Video Coding) 으로도 지칭되는, ITU-T H.264 표준과 같은 다른 사유 (proprietary) 또는 산업 표준들 또는 그러한 표준들의 확장들에 따라 동작할 수도 있다. 하지만, 본 개시의 기법들은 임의의 특정 코딩 표준에 한정되지 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 를 포함한다.

[0034] 3D, 멀티뷰, 스케일러블 및 스크린 콘텐츠에 대한 확장과 같은 HEVC 에 대한 다양한 확장들이 현재 개발 중에 있다. 추가적으로, HEVC 에 대한 범위 확장 (Range Extension), 즉 HEVC RExt 이, 또한 JCT-VC 에 의해 개발되고 있다. 이하에서 RExt WD7 로도 지칭되는, 범위 확장들의 최근 작업 초안 (WD) 는 http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1005-v4.zip 로부터 입수가능하다.

[0035] 본 명세서에서, JCTVC-Q1003에서와 같이 HEVC 규격 텍스트는 종종 HEVC 버전 1으로 지칭된다. 본 개시의 기술은 설명의 용이성을 위해 HEVC 용어를 이용할 수도 있다. 그러나, 본 개시의 기술은 HEVC에 한정되는 것으로 가정되어서는 안되고, 실제로, 본 개시의 기술이 차세대 표준 뿐만 아니라 HEVC의 확장들을 포함하여 HEVC에 대한 후속 표준에서 구현될 수도 있다는 것이 명시적으로 고려된다.

[0036] 비록 도 1에 도시되지는 않았지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통 데이터 스트림 또는 분리된 데이터 스트림들에서 오디오 및 비디오 양자 모두의 인코딩을 핸들링 (handling) 하기 위하여 적절한 MUX-DEMUX 유닛들 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜 또는 다른 프로토콜들 이를테면 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0037] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 반도체 (ASIC), 필드 프로그램가능 게이트 어레이 (FPGA), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합과 같은 다양한 적합한 인코더 회로 중 어느 것으로서 구현될 수도 있다. 그 기법들이 부분적으로 소프트웨어로 구현될 때, 디바이스는 적합한 비일시적 컴퓨터 판독가능 매체에 그 소프트웨어를 위한 명령들을 저장하고 본 개시의 기법들을 수행하기 위하여 하나 이상의 프로세서들을 이용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있는데, 이들 중 어느 쪽이 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다.

[0038] 위에 도입된 대로, JCT-VC 는 HEVC 표준, ITU-T H.265의 개발을 최근 완료했다. HEVC 는 예컨대, ITU-TH.264/AVC 에 따른 기존 디바이스들에 비하여 비디오 코딩 디바이스들의 여러 추가적인 능력들을 가능하게 한다. 예를 들어, H.264 는 9개의 인트라 예측 인코딩 모드들을 제공하지만, HEVC 는 무려 35개의 인트라 예측 인코딩 모드들을 지원한다.

[0039] HEVC 및 다른 비디오 코딩 규격들에서, 비디오 시퀀스는 통상적으로 일련의 화상 (picture) 들을 포함한다. 화상들은 또한, "프레임" 들로 지칭될 수도 있다. 화상은, S_L, S_{Cb} 및 S_{Cr} 로 표기되는, 3개의 샘플 어레이들을 포함할 수도 있다. S_L 는 루마 샘플들의 2차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 크로미넌스 샘플들의 2차원 어레이이다. S_{Cr} 는 Cr 크로미넌스 샘플들의 2차원 어레이이다. 크로미넌스 샘플들은 또한, 본원에서 "크로마" 샘플들로 지칭될 수도 있다. 다른 사례들에서, 화상은 단색 (monochrome) 일 수도 있고, 루마 샘플들의 어레이만을 포함할 수도 있다.

[0040] 화상의 인코딩된 표현을 생성하기 위하여, 비디오 인코더 (20) 는 코딩 트리 유닛 (CTU) 들의 세트를 생성할 수도 있다. CTU 들의 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2개의 대응 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 단색 화상들 또는 3개의 분리된 색 평면들을 갖는 화상들에서, CTU 는 단일 코딩 트리 블록 및 그 코딩 트리 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로 지칭될 수도 있다. HEVC 의 CTU 들은 대체로, H.264/AVC 와 같은 다른 표준들의 매크로블록들에 유사할 수도 있다. 하지만, CTU 가 특정 크기로 반드시 한정되는 것은 아니고 하나 이상의 코딩 유닛들 (CU) 들을 포함할 수도 있다. 슬라이스는 래스터 스캔 순서에서 연속적으로 순서화된 정수 개의 CTU를 포함할 수도 있다.

[0041] 코딩된 CTU 를 생성하기 위하여, 비디오 인코더 (20) 는 CTU 의 코딩 트리 블록들에 대한 쿼드 트리 파티셔닝을 회귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들을 분할할 수도 있으며, 따라서 이는 "코딩 트리 유닛"들이라고 부른다. 코딩 블록은 샘플들의 NxN 블록일 수도 있다. CU 는 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 화상의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 그 코딩 블록들의 샘플들을 코딩하는데 사용된 신택스 구조를 포함할 수도 있다. 단색 화상들 또는 3개의 분리된 색 평면들을 갖는 화상들에서, CU 는 단일 코딩 블록 및 그 코딩 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다.

[0042] 비디오 인코더 (20) 는 하나 이상의 예측 블록들로 CU 의 코딩 블록을 파티셔닝할 수도 있다. 예측 블록은, 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비정사각형) 블록이다. CU 의 예측 유닛 (PU) 는 루마 샘플들의 예측 블록, 크로마 샘플들의 2개 대응하는 예측 블록들, 및 예측 블록들을 예측하는데 사용된 신택스 구조들을 포함할 수도 있다. 단색 화상들 또는 3개의 분리된 색 평면들을 포함하는 화상들에

서, PU 는 단일 예측 블록 및 그 예측 블록을 예측하는데 이용된 신택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는, 예측 루마, Cb, 및 Cr 블록들을, CU 의 각각의 PU 의 루마, Cb, 및 Cr 예측 블록들을 위해 생성할 수도 있다.

[0043] 비디오 인코더 (20) 는 PU 에 대해 예측 블록을 생성하는데 인트라 예측 또는 인터 예측을 사용할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인트라 예측을 사용하면, 비디오 인코더 (20) 는 PU 와 연관된 화상의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인터 예측을 사용하면, 비디오 인코더 (20) 는 PU 와 연관된 화상 외의 하나 이상의 화상들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0044] 비디오 인코더 (20) 가, CU 의 하나 이상의 PU들에 대해 예측 루마, Cb, 및 Cr 블록들을 생성한 후에, 비디오 인코더 (20) 는 CU 를 위한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에 있는 각각의 샘플은 CU 의 예측 루마 블록들 중 하나에 있는 루마 샘플과 CU 의 원래 루마 코딩 블록에 있는 대응하는 샘플 사이의 차이를 표시한다. 또한, 비디오 인코더 (20) 는 CU 를 위한 Cb 잔차 블록을 생성할 수도 있다. 또한, CU 의 Cb 잔차 블록에 있는 각각의 샘플은 CU 의 예측 Cb 블록들 중 하나에 있는 Cb 샘플과 CU 의 원래 Cb 코딩 블록에 있는 대응하는 샘플 사이의 차이를 표시할 수도 있다. 또한, 비디오 인코더 (20) 는 CU 를 위한 Cr 잔차 블록을 생성할 수도 있다. 또한, CU 의 Cr 잔차 블록에 있는 각각의 샘플은 CU 의 예측 Cr 블록들 중 하나에 있는 Cr 샘플과 CU 의 원래 Cr 코딩 블록에 있는 대응하는 샘플 사이의 차이를 표시할 수도 있다.

[0045] 게다가, 비디오 인코더 (20) 는 CU 의 루마, Cb 및 Cr 잔차 블록들을 하나 이상의 루마, Cb 및 Cr 변환 블록들로 분해하기 위하여 쿼드 트리 파티셔닝을 이용할 수도 있다. 변환 블록은, 동일한 변환이 적용되는 샘플들의 직사각형 (예를 들어, 정사각형 또는 비정사각형) 블록이다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2개 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용된 신택스 구조들을 포함할 수도 있다. 따라서, CU 의 각 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 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. 단색 화상들 또는 3개의 분리된 색 평면들을 갖는 화상들에서, TU 는 단일 변환 블록 및 그 변환 블록의 샘플들을 변환하는데 사용된 신택스 구조들을 포함할 수도 있다.

[0046] 비디오 인코더 (20) 는 TU 를 위한 루마 계수 블록을 생성하기 위하여 TU 의 루마 변환 블록에 하나 이상의 변환들을 적용할 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 비디오 인코더 (20) 는 TU 를 위한 Cb 계수 블록을 생성하기 위하여 TU 의 Cb 변환 블록에 하나 이상의 변환들을 적용할 수도 있다. 비디오 인코더 (20) 는 TU 를 위한 Cr 계수 블록을 생성하기 위하여 TU 의 Cr 변환 블록에 하나 이상의 변환들을 적용할 수도 있다.

[0047] 계수 블록 (예를 들어, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후에, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 일반적으로 양자화는, 변환 계수들이 양자화되어 그 변환 계수들을 나타내는데 사용된 데이터의 양을 감소시킬 수 있으며, 추가 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 는 계수 블록을 양자화한 후에, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 신택스 엘리먼트들에 대해 콘텍스트 적응형 이진 산술 코딩 (CABAC) 을 수행할 수도 있다.

[0048] 비디오 인코더 (20) 는, 코딩된 화상 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 NAL 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은 NAL 유닛에서의 데이터의 타입 표시 및 에뮬레이션 방지 비트와 함께 필요에 따라 산재된 RBSP 의 형태로 그 데이터를 포함하는 바이트를 포함하는 신택스 구조이다. NAL 유닛들의 각각은, NAL 유닛 헤더를 포함하고, RBSP 를 캡슐화 (encapsulate) 한다. NAL 유닛 헤더는, NAL 유닛 타입 코드를 표시하는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 명시되는 NAL 유닛 타입 코드는 NAL 유닛의 타입을 표시한다. RBSP 는 NAL 유닛 내에 캡슐화되는 정수 개의 바이트들을 포함하는 신택스 구조일 수도 있다. 일부 사례들에서, RBSP 는 제로 비트들을 포함한다.

[0049] 상이한 타입들의 NAL 유닛들은 상이한 타입들의 RBSP 들을 캡슐화할 수도 있다. 예를 들어, 제 1 타입의 NAL 유닛은, PPS 를 위해 RBSP 를 캡슐화할 수도 있고, 제 2 타입의 NAL 유닛은 코딩된 슬라이스를 위해 RBSP 를 캡슐화할 수도 있고, 제 3 타입의 NAL 유닛은, SEI 메시지들을 위해 RBSP 를 캡슐화할 수도 있고 기타 등등

이다. (파라미터 세트들 및 SEI 메시지들을 위한 RBSP 들과는 반대로) 비디오 코딩 데이터를 위해 RBSP 를 캡슐화하는 NAL 유닛들은 VCL NAL 유닛들로 지칭될 수도 있다.

[0050] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 또한, 비디오 디코더 (30) 는 비트스트림으로부터 신택스 엘리먼트들을 획득하기 위하여 비트스트림을 파싱할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 화상들을 재구성할 수도 있다. 비디오 데이터를 재구성하기 위한 프로세스는 일반적으로, 비디오 인코더 (20) 에 의해 수행되는 프로세스에 상반될 수도 있다. 또한, 비디오 디코더 (30) 는 현재 CU 의 TU 들과 연관된 계수 블록들을 역 양자화할 수도 있다. 비디오 디코더 (30) 는 현재 CU 의 TU 들과 연관된 변환 블록들을 재구성하기 위하여 계수 블록들에 대해 역 변환들을 수행할 수도 있다. 비디오 디코더 (30) 는 현재 CU 의 TU 들의 변환 블록들의 대응하는 샘플들에 현재 CU 의 PU 들을 위한 예측 블록들의 샘플들을 추가하는 것에 의해 현재 CU 의 코딩 블록들을 재구성할 수도 있다. 화상의 각 CU 에 대해 코딩 블록들을 재구성하는 것에 의해, 비디오 디코더 (30) 는 화상을 재구성할 수도 있다.

[0051] 최근에는 모션과 함께 그래픽 및 텍스트와 같은 스크린 콘텐츠 재료를 위한 새로운 코딩 툴에 대한 연구가 요청되었고, 스크린 콘텐츠의 코딩 효율을 향상시키는 기술이 제안되었다. 일부 코딩 시나리오에서는 새로운 전용 코딩 도구들로 스크린 콘텐츠의 특성을 활용하여 코딩 효율의 개선이 획득될 수도 있다는 증거가 있기 때문에, CfP (Call for Proposals) 가 스크린 콘텐츠 코딩 (SCC) 를 위한 특정 도구들을 포함하는 고효율 비디오 코딩 (HEVC) 표준의 가능하게는 개발중인 향후 확장의 타겟으로 이슈되었다. 이 CfP의 사용 사례와 요구사항은 MPEG 문서 N14174에 기재되어 있다. 제 17 회 JCT-VC 미팅 동안, SCC 테스트 모델 (SCM) 이 확립되었다. SCC 의 최근 작업 초안 (WD) 는 http://phenix.int-evry.fr/jct/doc_end_user/documents/20_Geneva/wg11/JCTVC-T1005-v2.zip 로부터 입수가능하다.

[0052] 현재 SCC에서, IBC 시그널링은 현재 화상을 참조 화상 리스트에 추가함으로써 인터 예측 시그널링과 통합된다. 현재 슬라이스를 디코딩하기 전에, 비디오 디코더 (30) 는 현재 화상을 장기 참조 화상으로서 마킹한다. 그 다음, 현재 화상의 디코딩이 완료된 후에 현재 화상은 단기 참조 화상으로 다시 변환된다. 병합/AMVP 시그널링, AMVP 도출 및 MVD 코딩을 포함하는 시그널링 및 코딩 방법은 인터 케이스 (inter case) 에서와 동일하며, IBC 모드에 대한 모션 벡터는 정수 모션 벡터일 것이 요구된다는 차이가 있다. IBC 블록은 대응 참조 화상을 체크함으로써 종래의 인터 블록과 구별될 수 있다. 현재 화상 만이 참조 화상으로서 사용되면, 현재 블록은 IBC 블록이다. 그렇지 않으면, 현재 블록은 인터 블록이다. SPS 레벨 신택스 엘리먼트 curr_pic_as_ref_enabled_flag 는 현재 화상의 이미 코딩된 부분이 현재 화상의 블록을 코딩하기 위한 참조 화상으로서 사용될 수 있는지 여부를 표시하기 위해 사용될 수도 있다. 즉, 비디오 디코더 (30) 는 신택스 엘리먼트 curr_pic_as_ref_enabled_flag 를 수신할 수도 있고, 신택스 엘리먼트 curr_pic_as_ref_enabled_flag 가 1 과 동일한 것에 응답하여, 비디오 디코더 (30) 는 IBC를 사용하여 SPS와 연관된 슬라이스들의 일부 블록들을 디코딩할 수도 있다.

[0053] 도 2는, 본 개시에 따라 동일한 화상 내의 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드에 따라, 예를 들어 본 개시의 기법들에 따른 IBC 모드에 따라, 현재 화상 (103) 내의 비디오 데이터 (102) 의 현재 블록을 예측하기 위한 일 예의 기법을 예시하는 개념도이다. 도 2는 현재 화상 (103) 내의 예측 블록 비디오 블록 (104) 을 예시한다. 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, 본 개시의 기법들에 따른 IBC 모드에 따라 현재 비디오 블록 (102) 을 예측하기 위하여 예측 비디오 블록 (104) 을 이용할 수도 있다.

[0054] 비디오 인코더 (20) 는, 비디오 데이터의 이전에 재구성된 블록들의 세트로부터 현재 비디오 블록 (102) 을 예측하기 위한 예측 비디오 블록 (104) 을 선택한다. 비디오 인코더 (20) 는, 또한 인코딩된 비디오 비트스트림에 포함된 비디오 데이터를 역 양자화 및 역 변환하고, 비디오 데이터의 재구성된 블록들을 예측하는데 이용된 예측 블록들과 결과적인 잔차 블록들을 합산하는 것에 의해, 비디오 데이터의 블록들을 재구성한다. 도 2의 예에서, "의도된 구역" (intended area) 또는 "래스터 구역" (raster area) 으로도 지칭될 수도 있는 화상 (103) 내의 의도된 영역 (108) 은 이전에 재구성된 비디오 블록들의 세트를 포함한다. 비디오 인코더 (20) 는, 아래에서 더 자세하게 설명되어 있는 바처럼, 다양한 방식들로 화상 (103) 내의 의도된 영역 (108) 을 정의할 수도 있다. 비디오 인코더 (20) 는, 의도된 영역 (108) 내의 다양한 비디오 블록들에 기초하여 현재 비디오 블록 (102) 을 예측하고 코딩하는 상대적인 효율성 및 정확성의 분석에 기초하여 의도된 영역 (108) 에 있는 비디오 블록들 중에서 현재 비디오 블록 (102) 을 예측하기 위하여 예측 비디오 블록 (104) 을 선택할 수도

있다.

[0055] 비디오 인코더 (20) 는 현재 비디오 블록 (102) 에 대해 예측 비디오 블록 (104) 의 위치 또는 변위를 나타내는 2차원 벡터 (106) 를 결정한다. 2차원 블록 벡터 (106) 는, 현재 비디오 블록 (102) 에 대해 예측 비디오 블록 (104) 의 수평 및 수직 변위를 각각 나타내는 수평 변위 성분 (112) 및 수직 변위 성분 (110) 을 포함한다. 비디오 인코더 (20) 는, 인코딩된 비디오 비트스트림에서, 2차원 블록 벡터 (106) 를 식별 또는 정의하는, 예를 들어, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 을 정의하는, 하나 이상의 신택스 엘리먼트들을 포함할 수도 있다. 비디오 디코더 (30) 는 하나 이상의 신택스 엘리먼트들을 디코딩하여 2차원 블록 벡터 (106) 를 결정하고, 결정된 벡터를 이용하여 현재 비디오 블록 (102) 을 위한 예측 비디오 블록 (104) 을 식별할 수도 있다.

[0056] 일부 예들에서, 2차원 블록 벡터 (106) 의 공간 해상도는 정수 픽셀 해상도일 수 있다, 예를 들어, 정수 픽셀 해상도를 갖는 것으로 제약될 수 있다. 그러한 예들에서, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 의 공간 해상도는 정수 픽셀 해상도일 수도 있다. 그러한 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 비디오 블록 (102) 을 위한 예측자를 결정하기 위하여 예측 비디오 블록 (104) 의 픽셀 값들을 보간할 필요가 없다.

[0057] 다른 예들에서, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 중의 하나 또는 양자 모두의 해상도는 서브 픽셀 (sub-pixel) 일 수 있다. 예를 들어, 성분들 (110 및 112) 중의 하나는 정수 픽셀 해상도를 가질 수도 있는 반면, 다른 것은 서브 픽셀 해상도를 갖는다. 일부 예들에서, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 의 양자 모두의 해상도는 서브 픽셀일 수 있지만, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 은 상이한 해상도를 가질 수도 있다.

[0058] 일부 예들에서, 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 수평 변위 성분 (112) 및 수직 변위 성분 (110) 의 해상도를 특정 레벨, 예를 들어, 블록 레벨, 슬라이스 레벨 또는 화상 레벨 적응화에 기초하여 적응시킨다. 예를 들어, 비디오 인코더 (20) 는, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 이 정수 픽셀 해상도인지 또는 정수 픽셀 해상도가 아닌지를 표시하는 플래그를 슬라이스 레벨에서, 예를 들어, 슬라이스 헤더에서 시그널링할 수도 있다. 플래그가 수평 변위 성분 (112) 및 수직 변위 성분 (110) 이 정수 픽셀 해상도가 아니라고 표시하면, 비디오 디코더 (30) 는 해상도가 서브 픽셀 해상도라고 추론할 수도 있다. 일부 예들에서, 반드시 플래그인 것은 아닌 하나 이상의 신택스 엘리먼트들이 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 집합적 또는 개별적 해상도들을 표시하기 위하여 비디오 데이터의 각각의 슬라이스 또는 다른 유닛에 대해 송신될 수도 있다.

[0059] 또 다른 예들에서, 플래그 또는 신택스 엘리먼트 대신에, 비디오 인코더 (20) 는 해상도 콘텍스트 정보로부터 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 해상도에 기초하여 설정될 수도 있고, 비디오 디코더 (30) 는 해상도 콘텍스트 정보로부터 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 해상도를 추론할 수도 있다. 예들로서, 해상도 콘텍스트 정보는, 현재 비디오 블록 (102) 을 포함하는 화상 또는 화상들의 시퀀스를 위한, 색공간 (예를 들어, YUV, RGB 등), 특정 컬러 포맷 (예를 들어, 4:4:4, 4:2:2, 4:2:0 등), 프레임 크기, 프레임 레이트, 또는 양자화 파라미터 (QP) 를 포함할 수도 있다. 적어도 일부 예들에서, 비디오 코더는, 이전에 코딩된 프레임들 또는 화상들에 관련된 정보에 기초하여, 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 해상도를 결정할 수도 있다. 이런 식으로, 수평 변위 성분 (112) 의 해상도 및 수직 변위 성분 (110) 에 대한 해상도는 사전 정의 또는 시그널링될 수도 있거나, 다른 사이드 정보 (예를 들어, 해상도 콘텍스트 정보) 로부터 추론될 수도 있거나, 또는 이미 코딩된 프레임들에 기초할 수도 있다.

[0060] 현재 비디오 블록 (102) 은 CU, 또는 CU 의 PU 일 수도 있다. 일부 예들에서, 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, IBC 에 따라 예측되는 CU 를 다수의 PU 들로 스플릿할 수도 있다. 그러한 예들에서, 비디오 코더는, CU 의 PU 들의 각각에 대해 각각의 (예를 들어, 상이한) 2차원 벡터 (106) 를 결정할 수도 있다. 예를 들어, 비디오 코더는 2Nx2N CU 를 2개의 2NxN PU들, 2개의 Nx2N PU들, 또는 4개의 NxN PU 들로 스플리팅할 수도 있다. 다른 예들로서, 비디오 코더는 2Nx2N CU 를 ((N/2)xN + (3N/2)xN) PU들, ((3N/2)xN + (N/2)xN) PU들, (Nx(N/2) + Nx(3N/2)) PU들, (Nx(3N/2) + Nx(N/2)) PU들, 4개의 (N/2)x2N PU들, 또는 4개의 2Nx(N/2) PU들로 스플리팅할 수도 있다. 일부 예들에서, 비디오 코더는 2Nx2N PU 를 이용하여 2Nx2N CU 를 예측할 수도 있다.

[0061] 현재 비디오 블록 (102) 은 루마 비디오 블록, 또는 루마 비디오 블록에 대응하는 크로마 비디오 블록일 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 단지 루마 비디오 블록들을 위한 2차원 벡터들 (106) 을 정의

하는 하나 이상의 신택스 엘리먼트들을 인코딩된 비디오 비트스트림으로 인코딩할 수도 있다. 그러한 예들에서, 비디오 디코더 (30) 는, 루마 블록을 위해 시그널링되는 2차원 벡터에 기초하여 루마 블록에 대응하는 하나 이상의 크로마 블록들의 각각에 대해 2차원 벡터들 (106) 을 도출할 수도 있다.

[0062] 컬러 포맷, 예를 들어, 컬러 샘플링 포맷 또는 크로마 샘플링 포맷에 따라, 비디오 코더는 루마 비디오 블록에 비해 대응하는 크로마 비디오 블록들을 다운샘플링할 수도 있다. 컬러 포맷 4:4:4 은 다운샘플링을 포함하지 않으며, 크로마 블록들이 루마 블록과 수평 및 수직 방향들에서 동일한 수의 샘플들을 포함한다는 것을 의미한다. 컬러 포맷 4:2:2 는 수평 방향에서 크로마가 다운샘플링되어, 루마 블록에 비해 크로마 블록들에서 수평 방향에 절반의 샘플들이 있다는 것을 의미한다. 컬러 포맷 4:2:0 는 수평 및 수직 방향들에서 크로마가 다운샘플링되며, 루마 블록에 비해 크로마 블록들에서 수평 및 수직 방향들에 절반의 샘플들이 있다는 것을 의미한다.

[0063] 비디오 코더들이 대응하는 루마 블록들을 위한 벡터들 (106) 에 기초하여 크로마 비디오 블록들을 위한 벡터들 (106) 을 결정하는 예들에서, 비디오 코더들은 루마 벡터를 수정할 필요가 있을 수도 있다. 예를 들어, 루마 벡터 (106) 는 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 이 홀수 개의 픽셀들인 정수 해상도를 갖고, 컬러 포맷이 4:2:2 또는 4:2:0 이면, 컨버팅된 루마 벡터는 대응하는 크로마 블록에서 정수 픽셀 위치를 가리키지 않을 수도 있다. 그러한 예들에서, 비디오 코더들은 대응하는 크로마 블록을 예측하기 위한 크로마 벡터로서 사용하기 위해 루마 벡터를 스케일링할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는, 의도된 영역 (108) 을 정의하거나, 또는 컨버팅된 벡터를 스케일링하여, 크로마 블록을 예측하는데 이용된 컨버팅된 루마 벡터 (106) 가, 재구성되지 않은 또는 인루프 필터링된 예측 크로마 블록들을 가리키지 않을 수도 있다. 본 개시에서, 크로마 블록을 위한 벡터의 스케일링은 명시적으로 언급되지 않지만, 그러한 스케일링이 일어나지 않는 것으로 가정되지 않아야 한다. 크로마 블록을 위한 벡터의 스케일링은, 본 개시에 설명된 예들에서 명시적으로 설명되지 않더라도 일어날 수도 있다 (하지만, 모든 경우에 반드시 필요한 것은 아닐 수도 있다).

[0064] 본 개시는 IBC 를 위해 I-슬라이스를 가능하게 하는 기술을 설명한다. HEVC에 대한 SCC 확장의 현재 초안 텍스트 규격에서, I-슬라이스는 IBC가 인에이블될 때 (예를 들어, 신택스 엘리먼트 curr_pic_as_ref_enabled_flag가 1 과 동일할 때) 사용될 수 없다. 특히, curr_pic_as_ref_enabled_flag가 1 과 동일할 때, slice_type 신택스 엘리먼트의 값은 2 와 동일할 수 없고, 0, 1, 및 2 의 slice_type 값이 각각 B, P 및 I 슬라이스에 대응한다. 이것은 몇가지 잠재적인 단점을 제공한다. 예를 들어, 종래의 인트라 코딩된 슬라이스 (참조를 위해 현재 화상을 포함하여 어느 화상도 사용하지 않고서 코딩됨) 는, 일부 시나리오들에서, 예를 들어 아래에서 설명되는 바와 같이 일부 시그널링 오버헤드를 세이브 (save) 하는 데 유용할 수도 있는, slice_type 신택스 엘리먼트를 통해 시그널링될 수 없다.

[0065] 본 개시는 IBC가 인에이블되는 경우에도 I-슬라이스 사용을 가능하게 하는 기술을 설명한다. IBC가 인에이블될 때 I 슬라이스 사용을 가능하게 하는 것은, 일부 예들에서, curr_pic_as_ref_enabled_flag 가 1과 동일한 경우 (IBC 가 인에이블됨을 표시함) 에도, slice_type 신택스 엘리먼트의 값이 2와 동일할 수 있게 한다 (주어진 슬라이스의 코딩 타입이 I 슬라이스임을 표시함). 이러한 경우에, I 슬라이스는, 참조를 위해, 현재 화상을 포함한 어느 화상도 사용하지 않고서 코딩되고, 모든 코딩된 블록들을 위해 정상 인트라 예측 모드들을 사용하여 코딩된 종래의 인트라 코딩된 슬라이스이다. 바꾸어 말하면, 종래의 I 슬라이스의 어떤 블록도 인터 예측 또는 IBC를 사용하여 코딩되지 않고, 종래의 I 슬라이스에 대해 사용되는 유일한 코딩 모드는 인트라 모드이다.

[0066] 비디오 인코더 (20) 는, 예를 들어, 비디오 데이터의 슬라이스를 I 슬라이스로서 인코딩하고, IBC 모드가 슬라이스에 대해 인에이블됨을 표시하기 위한 값을 갖는 IBC 신택스 엘리먼트 (예를 들어, curr_pic_as_ref_enabled_flag) 를 생성하도록 구성됨으로써 그러한 기능을 구현하도록 구성될 수도 있다. 비디오 인코더 (20) 는 추가적으로 슬라이스 타입 신택스 엘리먼트 (예를 들어, slice_type) 를 생성하고 그 신택스 엘리먼트를 슬라이스가 I 슬라이스임을 표시하는 값으로 설정할 수도 있다. 비디오 인코더 (20) 는 예를 들어 SPS 에 IBC 신택스 엘리먼트를 포함할 수도 있거나 및/또는 슬라이스 헤더에 슬라이스 타입 신택스 엘리먼트를 포함할 수도 있다.

[0067] 비디오 디코더 (30) 는, 예를 들어, 비디오 데이터의 슬라이스를 수신하고, IBC 모드가 슬라이스를 위해 인에이블됨을 결정하기 위해 IBC 신택스 엘리먼트 (예를 들어, curr_pic_as_ref_enabled_flag) 를 파싱하도록 구성됨으로써 그러한 기능성을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 가

curr_pic_as_ref_enabled_flag의 값이 1과 동일한 것으로 결정하면, 비디오 디코더 (30) 는 IBC 모드가 인에이블됨을 결정할 수도 있는 반면, curr_pic_as_ref_enabled_flag가 0과 동일하면, 비디오 디코더 (30) 는 IBC 모드가 디스에이블됨을 결정할 수도 있다. 비디오 디코더 (30) 는 또한 슬라이스가 I 슬라이스이고 I 슬라이스로서 디코딩될 것인지를 결정하기 위해 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱할 수도 있다. 슬라이스를 I 슬라이스로서 디코딩하기 위해, 비디오 디코더 (30) 는 인트라 예측만을 사용하여 그리고 인터 예측 모드를 사용하지 않고서 그리고 IBC 모드를 사용하지 않고서 슬라이스의 블록들을 디코딩할 수도 있다.

[0068] 대안적으로, 본 개시는 다른 슬라이스 유형을 도입하는 것을 제안한다. 예를 들어, IBC 슬라이스는 현재 화상을 제외한 어느 다른 참조 화상도 갖지 않고서 IBC 모드의 사용을 표시하는데 사용될 수도 있다. 즉, IBC 슬라이스에 대하여, 비디오 코더는 인트라 모드 또는 IBC 모드를 사용하여, 그러나 인터 모드를 사용하지 않고서 IBC 슬라이스의 블록을 코딩할 수도 있다. 이러한 예에서, 신택스 엘리먼트 slice_type의 값은 0, 1, 2 또는 3과 동일할 수도 있으며 새로 허용된 값인 3은 IBC 슬라이스를 명시한다. 아래의 표 1에 나타낸 바처럼, 다른 값들의 의미는 변경되지 않은 채로 남을 수도 있다. slice_type 이 IBC 슬라이스로 설정되면, P 및 B 슬라이스에 대한 인터 모드들을 위한 일부 시그널링은 피할 수도 있다. 예를 들어, 비디오 디코더 (30) 가 슬라이스가 IBC 슬라이스라고 결정하면, 정보, 이를테면 cu_skip_flag 신택스 엘리먼트, CU 헤더에서의 pred_mode_flag 신택스 엘리먼트, 슬라이스 헤더에서의 참조 화상 관리 정보 및 다른 그러한 정보는 인코딩된 비디오 데이터의 비트스트림에 포함될 필요가 없다.

slice_type	slice_type 의 이름
0	B (B 슬라이스)
1	P (P 슬라이스)
2	I (I 슬라이스)
3	IBC (IBC 슬라이스)

표 1

[0069]

[0070] 위에서 도입된 기술에 따라, 비디오 디코더 (30)는 비디오 데이터의 슬라이스를 수신하고 비디오 데이터의 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하도록 구성될 수도 있다. 슬라이스가 IBC 슬라이스를 표시하는 값을 갖는 슬라이스 타입 신택스 엘리먼트 (예를 들어, slice_type이 3 과 동일) 에 응답하여, 비디오 디코더 (30) 는 슬라이스를 IBC 슬라이스로서 디코딩할 수도 있으며, 이것은 IBC 모드를 이용하여 슬라이스의 일부 블록들 및/또는 인트라 모드를 사용하여 슬라이스의 일부 블록들을 디코딩하는 것을 포함할 수도 있다. 비디오 디코더 (30) 는 다른 슬라이스를 I 슬라이스 (예를 들어, slice_type 이 2와 동일), P 슬라이스 (slice_type이 1과 동일) 또는 B 슬라이스 (slice_type이 0과 동일) 로서 디코딩할 수도 있다. 다른 유형의 슬라이스들을 디코딩하는 것은 IBC 슬라이스들로서 디코딩된 슬라이스들에 대해 수신되지 않은 일부 상이한 신택스 엘리먼트들을 수신하는 것을 수반할 수도 있고, 그리고 마찬가지로, IBC 슬라이스를 디코딩하는 것은 다른 슬라이스들에 대해 수신되지 않은 일부 상이한 신택스 엘리먼트들을 수신하는 것을 수반할 수도 있다.

[0071] 이 개시는 또한 슬라이스 레벨 IBC 제어 플래그에 대한 기술을 설명한다. 기존의 기술에 있어서, IBC가 인에이블되는지 여부는 SPS 플래그 curr_pic_as_ref_enabled_flag 에 의해서만 제어된다 (즉, 이에 의해서 IBC의 인에이블이 결정된다). 가령, 슬라이스 레벨에서 IBC 를 보다 세밀하게 제어할 수 있다면 몇가지 잠재적인 이점이 있을 수도 있다. 일례로서, 슬라이스 레벨 IBC 제어는 현재 슬라이스 내의 블록들이 현재 화상을 참조로서 사용할 수 있는지 여부를 명시할 수도 있다. IBC 모드는 인루프 필터링 (in-loop filtering) 전에 샘플들의 추가 저장을 필요로 할 수도 있다. IBC가 슬라이스 레벨에서 디스에이블될 때, 스마트 디코더는 이러한 추가 샘플을 저장하는 것을 피함으로써, 평균 대역폭과 평균 전력 사용량을 줄일 수도 있다. 또 다른 예로서, 슬라이스 레벨 IBC 제어는 IBC를 사용하지 않는 슬라이스가 시간 참조 화상에 할당된 더 많은 참조 인덱스들을 가질 수 있게 하며, 이는 어떤 시나리오에서 잠재적으로 개선된 코딩 효율성을 위해 더 많은 가중치가 가중된 예측에서 시그널링되는 것을 허용한다.

[0072] 슬라이스 레벨 IBC 제어를 인에이블하기 위한 기술이 이제 더 상세히 설명될 것이다. 비디오 인코더 (20) 는 IBC가 인에이블될 때 (예를 들어, curr_pic_as_ref_enabled_flag가 1과 동일) 슬라이스 헤더에서 슬라이스

레벨 IBC 제어 플래그, 예를 들어, slice_curr_pic_as_ref_enabled_flag를 시그널링할 수도 있다. 부가적으로, 비디오 디코더 (30) 는 시그널링되지 않으면 slice_curr_pic_as_ref_enabled_flag의 값이 0과 동일하다고 추론 (즉, 명시적 시그널링의 수신 없이 결정) 할 수 있다. 슬라이스 레벨 제어 플래그는 P 슬라이스들 및 B 슬라이스들에 대해서만 시그널링될 수도 있고, I 슬라이스에 대해 0 의 값을 갖는 것으로 추론될 수도 있다. 이 슬라이스 레벨 IBC 플래그는, 예를 들어, 현재 화상이 슬라이스의 블록을 예측하기 위한 참조 화상으로서 사용되는지의 여부를 표시할 수도 있다. 다른 구현에서, 슬라이스 레벨 IBC 플래그는 현재 화상이 자기 참조를 위해 현재 화상에 대한 참조 화상 리스트에 추가되는지 여부를 표시할 수도 있고, 이 경우 현재 화상은 슬라이스의 블록을 예측하기 위한 참조 화상으로서 사용을 위해 이용 가능하지만, 일부 슬라이스에 대해서는, 실제로 예측에 사용되지 않을 수도 있다. 부가적으로, slice_curr_pic_as_ref_enabled_flag 가 디스에이블되면 (즉, 값이 0과 동일한 것으로 명시적으로 시그널링되거나 또는 0과 동일한 것으로 추론되면, 비디오 디코더 (30)는 그 특정 슬라이스에 대해 IBC 모드를 사용하지 않고, (IBC에 대한 I-슬라이스를 인에이블하는 것에 관해서 위에서 설명한 바와 같이) I-슬라이스 타입 시그널링이 인에이블될 수 있다.

[0073] slice_curr_pic_as_ref_enabled_flag 신택스 엘리먼트의 의미는 다음과 같을 수도 있다:

[0074] 1과 동일한 slice_curr_pic_as_ref_enabled_flag 는, 현재 슬라이스를 디코딩할 때 현재 화상이 현재 화상의 참조 화상 리스트에 포함될 수도 있음을 명시한다. 0과 동일한 slice_curr_pic_as_ref_enabled_flag 는, 현재 슬라이스를 디코딩할 때 현재 화상이 현재 화상의 참조 화상 리스트에 포함될 수 없음을 명시한다. 존재하지 않을 때, slice_curr_pic_as_ref_enabled_flag 의 값은 0 과 동일한 것으로 추론된다.

[0075] 일반적으로, slice_curr_pic_as_ref_enabled_flag 는 slice_curr_pic_as_ref_enabled_flag 이후의 슬라이스 헤더 신택스에서, 슬라이스 헤더 레벨 아래의 신택스 구조, 의미론 및 디코딩 프로세스에서, 예를 들어, 참조 화상 리스트 구축, DPB 관리 등에서 curr_pic_as_ref_enabled_flag 를 대체할 수 있다. 신택스 엘리먼트 curr_pic_as_ref_enabled_flag 는 slice_curr_pic_as_ref_enabled_flag 시그널링에 대한 게이팅 플래그로서 취급될 수 있다. 예를 들어, NumPicTotalCurr의 값은 슬라이스 레벨 제어 플래그 slice_curr_pic_as_ref_enabled_flag 에 기초하여 도출될 수도 있다.

[0076] 또한, (위에서 설명한대로 I-슬라이스를 인에이블하는 기술이 적용되지 않을 때) 다음 조건이 적용될 수도 있다:

[0077] - 현재 화상이 BLA 또는 CRA 화상일 때, NumPicTotalCurr 의 값은 slice_curr_pic_ref_enabled_flag 과 동일할 것이다.

[0078] 다른 대안에서, 비디오 디코더 (30) 는 명시적인 시그널링 없이 slice_curr_pic_as_ref_enabled_flag 의 값을 도출하도록 구성될 수도 있다. 도출의 일부로서, 비디오 디코더 (30) 는 참조 화상 리스트 RefPicList0 및 RefPicList1을 슬라이스 헤더에서 시그널링되는 최대 수의 참조 화상들 (예를 들어, RefPicList0 및 RefPicList1 각각에 대한 num_ref_idx_l0_active_minus1 및 num_ref_idx_l1_active_minus1) 까지 체크할 수도 있고, 참조 화상이 현재 화상과 동일하지 않으면, 비디오 디코더 (30) 는 slice_curr_pic_as_ref_enabled_flag 를 0 과 동일한 것으로 도출할 수도 있다. 그렇지 않으면, 비디오 디코더 (30) 는 slice_curr_pic_as_ref_enabled_flag 가 1과 동일한 것으로 도출할 수도 있다. 도출된 플래그는, 예를 들어, 디코딩 프로세스 및 다른 관련된 사용에서 시그널링된 슬라이스 레벨 IBC 플래그 대신에 사용될 수도 있다.

[0079] 전술한 슬라이스 레벨 IBC 제어 플래그를 구현하기 위해, 비디오 디코더 (30) 는 비디오 데이터의 슬라이스를 수신하고 슬라이스 레벨 신택스 엘리먼트 (예를 들어, slice_curr_pic_as_ref_enabled_flag) 를 파싱하여 IBC 모드가 슬라이스에 대해 인에이블되는 것을 결정하도록 구성될 수도 있다. 비디오 디코더 (30) 는 비디오 데이터의 슬라이스에 대한 참조 화상 리스트를 구축하도록 구성될 수도 있고, 참조 화상 리스트에 포함된 화상의 수는 슬라이스 레벨 신택스 엘리먼트가 IBC 가 인에이블되는지 또는 디스에이블되는지를 표시하는지에 의존할 수도 있다. 비디오 디코더 (30) 는 슬라이스가 P 슬라이스 또는 B 슬라이스 중 하나인 것을 결정하는 것에 응답하여 수행되는 슬라이스에 대해 IBC 모드가 인에이블됨을 결정하기 위하여 슬라이스 레벨 신택스 엘리먼트를 파싱할 수도 있다. I 슬라이스인 것으로 결정되는 슬라이스에 대해, 비디오 디코더 (30) 는 IBC 모드가 디스에이블됨을 표시하기 위해 슬라이스 레벨 신택스 엘리먼트의 값을 추론할 수도 있다. 비디오 디코더 (30) 는 비디오 데이터의 슬라이스에 대한 참조 화상 리스트를 구축할 수도 있고, IBC 모드가 인에이블됨을 표시하는 슬라이스 레벨 신택스 엘리먼트에 응답하여, 슬라이스를 포함하는 화상을 참조 화상 리스트에 추가할 수도 있다.

[0080] 본 개시는 또한 슬라이스 레벨 IBC 제어 플래그 및 slice_type의 사용에 대한 잠재적 제약을 설명한다. 아래에 도입되는 제약 또는 양태 중 어느 것이 이 섹션 또는 본 개시에 있는 다른 섹션들에서의 다른 제약들 중 하나 이상과 조합하여 또는 독립적으로 적용될 수 있다.

[0081] 일부 예에서, 비트스트림 적합성은 다음 요구사항을 가질 수도 있다:

[0082] ● currPic 만 참조에 이용 가능할 때 slice_curr_pic_as_ref_enabled_flag 의 값은 P 및 B 슬라이스에 대해 0과 동일하지 않아야 한다.

[0083] ● nal_unit_type 이 BLA_W_LP 내지 RSV_IRAP_VCL23 를 포함한 범위의 값을 가질 때 (즉, 화상이 IRAP 화상일 때), slice_curr_pic_as_ref_enabled_flag 의 값은 P 및 B 슬라이스에 대해 0과 동일해서는 안된다. RefPicList0는 현재 화상만 참조하는 하나 이상의 엔트리들을 포함할 것이다.

[0084] ● 대안적으로, currPic 만이 참조에 이용 가능할 때 (즉, DPB가 현재 화상인 하나의 참조 화상만을 포함할 때) slice_curr_pic_as_ref_enabled_flag 는 P 및 B 슬라이스에 대해 시그널링되어서는 안되며, 그 값은 다음과 같이 추론된다

[0085] 디코딩 프로세스에 대한 변경이 이제 논의될 것이다. 디코딩 프로세스는 밑줄친 텍스트와 함께 나타낸 바와 같이 변경될 수도 있다. 나타낸 변경들은 따로따로 또는 공동으로 적용될 수도 있다.

[0086] 현재 화상 내의 적어도 하나의 슬라이스가 1과 동일한 slice_curr_pic_as_ref_enabled_flag 를 가질 때, DPB 내의 화상 저장 버퍼가 현재 화상에 할당되고, 현재 화상은 "장기 참조에 사용되는 것으로" 마킹된다.

[0087] 8.4, 8.5, 8.6, 8.7 섹션에 있는 프로세스는 모든 신택스 구조 계층에서 신택스 엘리먼트들을 사용한 디코딩 프로세스를 명시한다. 비트스트림 적합성의 요구 사항은 화상의 코딩된 슬라이스가 화상의 모든 코딩 트리 유닛에 대해 슬라이스 세그먼트 데이터를 포함함으로써, 슬라이스로의 화상의 분할, 슬라이스 세그먼트로의 슬라이스의 분할 및 코딩 트리 유닛들로의 슬라이스 세그먼트들의 분할 각각이 화상의 파티셔닝을 형성하는 것이다.
현재 화상 내의 적어도 하나의 슬라이스가 1과 동일한 slice_curr_pic_as_ref_enabled_flag 를 가질 때, 인루프 필터링 전에 현재 화상의 디코딩된 샘플 값들이 현재 화상에 대해 할당된 화상 저장 버퍼 내에 저장된다.

[0088] 주 2 - 현재 화상에서 적어도 하나의 슬라이스가 1과 동일한 slice_curr_pic_as_ref_enabled_flag 를 가질 때, 인루프 필터링 후 현재 화상의 샘플 값을 저장하기 위해 DPB의 일부로 카운팅되지 않은 하나의 추가 화상 저장 버퍼가 필요하다.

[0089] 본 개시는 또한 콜로케이팅된 화상 제약 (collocated picture constraint) 을 설명한다. 현재 WD 에서는, 현재 화상을 콜로케이팅된 화상으로서 사용할 수 없다는 제약이 있다. 이러한 제약의 이유는, 콜로케이팅된 화상이 현재 화상이라면, 모션 예측에 사용될 콜로케이팅된 블록에 대해 아직 할당된 모션 필드가 존재하지 않는다는 것이다.

[0090] 이 제약은, 예를 들어, 현재 화상의 처리 전에 현재 화상의 모션 필드를 초기화함으로써 제거될 수 있다. 예를 들어, 현재 화상은 인트라 코딩된 것으로서 모든 블록을 갖게 초기화될 수 있으며, 이는 모션 필드 정보가 없음을 의미한다. 대안적으로, 모션 필드는 디폴트 모션 필드로 초기화 될 수 있고, 여기서 디폴트 모션 필드는 인코더 및 디코더 양자 모두에 대해 알려져 있는 것, 예를 들어 RefPicList0 또는 RefPicList1 또는 양자 모두와 연관된 적어도 하나의 인터 방향에서의 제로 모션 (제로 참조 인덱스를 갖는 제로 모션 벡터) 이다. 또한 슬라이스에 할당된 디폴트 모션 필드는 슬라이스 유형에 의존할 수 있다. 예를 들어, I-슬라이스에 대하여 해당 슬라이스에서의 모든 블록이 인트라 모드로 초기화되고, P-슬라이스에 대하여, RefPicList0 관련 모션 필드가 제로 모션으로 설정되고, B-슬라이스에 대하여 제로 모션이 RefPicList0 및 RefPicList1 과 연관된 양방향에서 할당된다.

[0091] 또한, 위에서 언급된 제약 (현재의 화상이 콜로케이트된 화상으로 사용될 수 없음) 은 인코더 제약 또는 비트스트림 제약의 형태로 명시되고, 이것은 일부 인코더에 의해 위반될 수도 있으며, 이는 비트스트림을 잠재적으로 손상시킬 수도 있다.

[0092] 본 개시는 예를 들어, 제약을 만족시키지 않는 신택스 엘리먼트들의 값이 비트스트림에서 시그널링될 수 없도록 신택스를 수정함으로써 이 제약을 더욱 엄격하게 하는 기술을 설명한다.

[0093] 첫째, 슬라이스 헤더에 시그널링된 TMVP 인에이블 플래그는 슬라이스가 현재 화상만을 참조 화상(들) 로서 갖는 경우 디스에이블되어야하는데, 이는 현재 화상이 콜로케이트된 화상으로서 사용될 수 없기 때문이다. 하나

보다 많은 참조 화상들이 있을 수 있으며, 이들 모두는 현재 화상이 될 수 있다.

[0094] 대안적으로, TMVP 인에이블 플래그 시그널링은 슬라이스에서 사용되는 참조 화상들의 개수 (예를 들어, num_ref_idx_l0_active_minus1 및 num_ref_idx_l1_active_minus1) 의 시그널링 후에 이동 및 배치될 수 있고, 모든 참조 화상들이 현재 화상인지 여부를 결정하기 위해 체크가 수행될 수 있다. 모든 참조 화상이 현재 화상인 경우, TMVP 인에이블 플래그는 시그널링되지 않고 디스에이블되는 것으로 (즉, 0과 동일한 것으로) 도출된다.

[0095] 또 다른 대안에서, 모든 참조 화상이 현재 화상인지 여부를 도출하는 대신에, 플래그는 그러한 경우를 표시하기 위해 슬라이스 헤더에서 시그널링될 수 있다.

[0096] 또한, collocated_ref_idx의 시그널링을 위해, 현재 화상을 가리키는 참조 인덱스는 collocated_ref_idx에 의해 참조되는 것으로부터 배제 될 수 있고, collocated_ref_idx의 값은 원하는 콜로케이트된 화상 전에 존재하는 현재 참조 화상을 가리키는 참조 인덱스의 수 만큼 감소된다. 유사한 방식으로, 파싱 후에, 파싱된 collocated_ref_idx는 시그널링된 collocated_ref_idx 이전에 현재의 참조 화상을 가리키는 참조 인덱스들의 수만큼 증가된다.

[0097] 일례에 따르면, RefPicList = {Pic0, Curr, Pic1, Curr, Pic2} 를 가정한다.

[0098] Pic0을 콜로케이트된 화상으로서 만들기 위해, collocated_ref_idx = 0 이 시그널링된다.

[0099] Pic1 을 콜로케이트된 화상으로서 만들기 위해 collocated_ref_idx = 1 이 시그널링되고 collocated_ref_idx가 1씩 증분되어 collocated_ref_idx를 2와 동일하게 만든다.

[0100] Pic2 를 콜로케이트된 화상으로서 만들기 위해, collocated_ref_idx = 2 이 시그널링되고, collocated_ref_idx가 2씩 증분되어 collocated_ref_idx를 4와 동일하게 만든다.

[0101] 예를 들어, 원하는 collocated_ref_idx가 N 과 동일하다고 가정하면, 인코딩 부분은 다음 의사 코드로 구현될 수도 있다.

[0102]
```
collocated_ref_idx = N;
```

[0103]
```
for( i = 0; i < N; i++ )
```

[0104]
```
{
```

[0105]
```
 if( RefPicList[i] is the current picture )
```

[0106]
```
 collocated_ref_idx--;
```

[0107]
```
}
```

[0108] 예를 들어, 디코딩 부분은 다음 의사 코드로 구현될 수도 있고, 여기서 collocated_ref_idx는 업데이트될 파싱된 값이다.

[0109]
```
for( i = 0; i <= collocated_ref_idx; i++ )
```

[0110]
```
{
```

[0111]
```
 if( RefPicList[i] is the current picture )
```

[0112]
```
 collocated_ref_idx++;
```

[0113]
```
}
```

[0114] 제공된 예에서, collocated_ref_idx의 값은 디코더에 의해 증분될 것이기 때문에, collocated_ref_idx 는 현재 화상 (Curr) 과 동일할 수 없으며, 현재 화상이 콜로케이트된 화상이 되는 것을 불가능하게 한다.

[0115] 또한, collocated_ref_idx가 시그널링되는 조건은 collocated_ref_idx 가 하나의 값만 가질 수있는 경우를 제외하도록 수정될 수 있다. 이 경우, RefPicList0 및 RefPiclList1에 포함되는 참조 화상의 수가 카운트될 수 있고 (여기서는 numCurrRef0 및 numCurrRef1 로 표기됨), 참조 화상의 수 (예를 들어, num_ref_idx_l0_active_minus1 및 num_ref_idx_l1_active_minus1) 는 RefPicList0 및 RefPicList1에 대한 현재 화상과 동일한 참조 인덱스들의 수만큼 감소될 수 있다. 이 수정된 참조 인덱스 수는, 예를 들어 collocated_ref_idx를 시그널링함에 있어서, 디코딩 프로세스에서 슬라이스 헤더에서의 시그널링된 참조 인덱스

수 대신에 사용될 수 있다.

[0116] 예를 들어, 다음과 같이 구현이 달성될 수 있다.

[0117] numCurrRef0 = 0;

[0118] for(i = 0; i <= num_ref_idx_l0_active_minus1; i++)

[0119] {

[0120] if(RefPicList0[i] is the current picture)

[0121] numCurrRef0++;

[0122] }

[0123] numCurrRef1 = 0;

[0124] for(i = 0; i <= num_ref_idx_l1_active_minus1; i++)

[0125] {

[0126] if(RefPicList1[i] is the current picture)

[0127] numCurrRef1++;

[0128] }

[0129] 콜로케이트된 참조 인덱스는 가능한 콜로케이트된 인덱스의 수가 1보다 큰 경우에만 시그널링된다, 예를 들어

if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && (num_ref_idx_l0_active_minus1 - numCurrRef0) > 0) || (!collocated_from_l0_flag && (num_ref_idx_l1_active_minus1 - numCurrRef1) > 0))	
collocated_ref_idx	ue(v)
}	

[0130] 밑줄친 부분은 현재 초안 텍스트 규격 위에 추가된다.

[0131] 예: RefPicList0 = {Curr, Pic0, Curr}

[0132] num_ref_idx_l0_active_minus1은 2와 동일하고, numCurrRef0은 2와 동일하다 (참조 화상 리스트에 2개의 현재 참조 화상이 있기 때문이다).

[0133] num_ref_idx_l0_active_minus1 - numCurrRef0 은 0 과 동일하고, collocated_ref_idx 이 시그널링되는 것이 아니라, 0 과 동일한 것으로 추론되지만, 실제 값은 1 이다 (제로 인덱스는 현재 화상을 가리킨다).

[0134] 0 으로의 추론 후, 위에서 설명된 콜로케이트된 인덱스 수정이 사용된다

[0135] for(int i = 0; i <= collocated_ref_idx; i++)

[0136] {

[0137] if(RefPicList0[i] is the current picture)

[0138] collocated_ref_idx++;

[0139] }

[0140] 이는 제 1 참조 화상이 참조 화상 리스트 내의 현재 화상과 동일하기 때문에 하나의 증분이 일어날 것이므로, 1 과 동일한 collocated_ref_idx를 생성할 것이다.

[0141] 이 예에서, collocated_ref_idx 시그널링은 현재 초안 텍스트 규격과 비교하여 세이브되며,

collocated_ref_idx가 시그널링되지만, 1 과 동일할 필요가 있다. 콜로케이팅된 화상 제약들 및 콜로케이팅된 참조 인덱스 시그널링과 관련하여 여기에 설명된 임의의 기술은 독립적으로 또는 임의의 조합으로 사용될 수도 있다.

[0142] 또한, 본 개시는 활성 참조 화상의 수에 기초하여 리스트 수정을 디스에이블하는 기술을 설명한다. 현재 WD에서 ref_pic_list_modification 프로세스는 NumPicTotalCurr> 1 및 lists_modification_present_flag = 1 일 때 호출될 수도 있다. 본 개시는, num_ref_idx_l0_active_minus1 = 0 (즉, 슬라이스를 디코딩하는데 사용될 수도 있는 참조 화상 리스트 0에 대한 최대 참조 인덱스가 0 과 동일함) 이고 슬라이스 레벨 IBC 제어 플래그에 대해 전술한 슬라이스 레벨 IBC 제어 플래그가 1 과 동일할 때, 리스트 수정 정보는 중복적 (redundant) 이고 추론될 수 있다는 것을 제안한다. 다시 말해서, 일부 구현들에서, 리스트 수정 프로세스를 필요로 하지 않고서 IBC 모드가 인에이블될 때 현재 화상은 항상 참조 화상 리스트에 추가될 수도 있다.

[0143] 본 개시는 num_ref_idx_l0_active_minus1 = 0 및 slice_curr_pic_as_ref_enabled_flag = 1 일 때 리스트 수정 정보를 시그널링하는 기술을 설명한다. 리스트 수정 프로세스는 curPic을 RefPicList0 에서의 첫번째 엔트리로서 포함하는 것으로 추론할 수도 있다. 다시 말해서, 일부 구현들에서, 리스트 수정 프로세스를 필요로 하지 않고서 IBC 모드가 인에이블될 때 현재 화상은 항상 참조 화상 리스트에 추가될 수도 있다.

[0144] 예를 들어, 비디오 인코더 (20) 가 IBC 슬라이스에 대해 IBC가 인 에이블링된다고 결정하면, 슬라이스에 대한 참조 화상 리스트 내의 유일한 참조 화상은 그 슬라이스를 포함하는 현재 화상이다. 따라서, 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수가 1과 동일하다고 결정하는 것에 응답하여, 그리고 슬라이스에 대해 IBC 모드가 인에이블되는 것에 응답하여, 비디오 인코더 (20) 는 참조 화상 리스트 수정을 디스에이블하고 인코딩된 비트스트림 내의 슬라이스에 대한 참조 화상 리스트 수정 정보를 포함하지 않도록 구성될 수도 있다. 유사하게, 비디오 디코더 (30) 는 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상의 수를 표시하는 신택스 엘리먼트 (예를 들어, num_ref_idx_l0_active_minus1) 를 수신하도록 구성될 수도 있고, 신택스 엘리먼트가 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수가 1과 동일 (예를 들어 num_ref_idx_l0_active_minus1은 0과 동일) 함을 표시하는 것에 응답하여, 그리고 IBC 모드가 슬라이스에 대해 인에이블됨 (예를 들어, slice_curr_pic_as_ref_enabled_flag = 1 및/또는 curr_pic_as_ref_enabled_flag = 1) 에 응답하여, 비디오 디코더 (30) 은 참조 화상 리스트 수정을 디스에이블할 수도 있다. 비디오 디코더 (30) 는 예를 들어, 참조 화상 리스트 수정 정보를 수신하지 않고서 참조 화상 리스트를 구축함으로써 참조 화상 리스트 수정을 디스에이블할 수도 있다. 일례에서, 참조 화상 리스트 수정을 디스에이블하는 것은 참조 화상 리스트 수정 정보와 연관된 정보 (예를 들어, 신택스 엘리먼트) 가 비트스트림에서 시그널링되지 않음을 의미할 수도 있으며, 따라서 잠재적으로 압축을 개선하고 디코더 측 복잡성을 감소시킨다.

[0145] 위의 예들은 RefPicList0 에 대해 설명되었지만, 위의 기술은 RefPicList1에도 마찬가지로 동일하게 적용가능할 수도 있음을 이해해야 한다.

slice_segment_header( ) {	기술자
first_slice_segment_in_pic_flag	**u(1)**
.....	
if(curr_pic_as_ref_enabled_flag && (slice_type = = P || slice_type = = B))	
slice_curr_pic_as_ref_enabled_flag	
.....	
if(lists_modification_present_flag && NumPicTotalCurr > 1 && !(slice_curr_pic_as_ref_enabled_flag && num_ref_idx_l0_active_minus1 == 0))	
ref_pic_lists_modification( )	
....	
byte_alignment( )	
}	

[0146]

[0147] 제 1 예시적 구현에서, 현재 초안 규격 (curr_pic_as_ref_enabled_flag) 에서처럼 SPS 레벨 IBC 제어 플래그가

1 과 동일하고 num_ref_idx_l0_active_minus1 = 0 일 때 (즉, 슬라이스를 디코딩하는데 사용될 수도 있는 참조 화상 리스트 0 에 대한 최대 참조 인덱스가 1 과 동일할 때) 리스트 수정은 시그널링되지 않을 수도 있다. 리스트 수정 프로세스는 curPic을 RefPicList0 에서의 첫번째 엔트리로서 포함하는 것으로 추론된다.

[0148] 위의 내용은 RefPicList1 에도 확장될 수도 있다.

slice_segment_header( ) {	기술자
first_slice_segment_in_pic_flag	**u(1)**
……	
if(curr_pic_as_ref_enabled_flag && (slice_type = = P || slice_type = = B))	
slice_curr_pic_as_ref_enabled_flag	
…….	
if(lists_modification_present_flag && NumPicTotalCurr > 1 && && !(curr_pic_as_ref_enabled_flag && num_ref_idx_l0_active_minus1 == 0))	
ref_pic_lists_modification( )	
….	
byte_alignment( )	
}	

[0149]

[0150] 본 개시는 또한 후 처리로서 인루프 필터링을 구현하기 위한 기술을 설명한다. 즉, 일부 코딩 시나리오에서 인루프 필터로서 적용되는 필터는 다른 코딩 시나리오에서 포스트 루프 필터로서 적용될 수도 있다. IBC 모드가 사용될 때, IBC 예측을 위해 사용되는 현재 화상의 참조 샘플들은 필터링되지 않는다. 즉, 디블로킹 필터 및 SAO와 같은 인루프 필터는 IBC 참조 샘플에 적용되지 않을 수도 있다. 그러나, 인루프 필터링은 규칙적 예측을 위해 재구성된 샘플에 여전히 적용될 수도 있다. 이러한 기술은 여러 이유로 전반적인 코딩 효율과 디바이스 성능을 향상시킬 수도 있다. 일례로서, 필터링은 그래픽 콘텐츠에 대한 좋은 예측자가 아닐 수도 있으므로, 필터링과 연관된 계산상의 복잡성은 특정 유형의 비디오 콘텐츠에 대해 최소의 혜택 또는 아무런 혜택도 제공하지 않을 수도 있다. 또 다른 예로, (IBC 및 규칙적 인터 모드를 위해) 현재 화상에 대해 필터링되지 않은 샘플 및 필터링된 샘플을 저장해야 할 필요성 때문에, 인루프 필터링은 샘플들을 쓰고 읽기 위한 대역폭을 증가시키고 필요한 저장 용량을 증가시킬 수도 있다.

[0151] 일례에서, 비디오 디코더 (30) 는 화상이 출력 또는 디스플레이될 때에만 인루프 필터를 적용하고, IBC 및 규칙적 인터 모드 양자 모두에 대해 참조 화상과 연관된 필터링되지 않은 픽셀을 사용할 수도 있다. 그러한 예에 대해, 비디오 디코더 (30) 는 IBC 모드를 사용하여 비디오 데이터의 블록을 재구성하고, 블록을 포함하는 화상을 출력하도록 구성될 수도 있고, 출력된 화상에서의 그 블록은 하나 이상의 루프 필터를 사용하여 필터링된다. 그러나, 비디오 디코더 (30) 는 미래의 블록에 대한 참조 화상으로서, 블록의 필터링되지 않은 버전을 갖는 화상을 저장할 수도 있다.

[0152] 이것은 필터링되지 않은 픽셀만이 저장될 필요가 있으므로 대역폭을 감소시킬 수도 있고 필터에 의해 제거될 수도 있는 상세들을 보존하여 더 양호하게 그래픽 콘텐츠를 예측할 수도 있다. 그러나, 디스플레이를 위해 출력되는 화상들에 인루프 필터가 여전히 적용될 수도 있다.

[0153] IBC 제어 플래그 또는 새로운 플래그는 인루프 필터가 포스트 필터 또는 출력 화상에 적용되는 필터로서 사용되어야 하는지를 도출하는데 사용될 수 있다. 예를 들어, 비디오 디코더 (30) 는 IBC 모드가 인에이블됨을 결정하는 것에 응답하여 루프 필터 동작이 인루프 또는 포스트 루프에 적용되어야 하는지를 표시하는 신택스 엘리먼트를 수신하도록 구성될 수도 있다. 또한, IBC 모드가 사용 중일 때 새로운 플래그가 시그널링될 수 있다. 이러한 플래그는 예를 들어, 비트 스트림에 존재하지 않을 때 0과 동일한 것으로 도출될 수도 있다.

[0154] 다른 예에서, IBC 모드는 필터링된 이미지가 이용 가능할 때 필터링된 (예를 들어, 디블록 필터링된, SAO 필터링된 또는 ALF 필터링된) 샘플을 사용할 수 있다. 필터에 대해, 필터링되는 블록 외부의 일부 픽셀이 필요하지만 이용불가일 수도 있다. 그러한 외부 픽셀이 이용 가능한 경우, 즉 이미 재구성된 경우, 필터링된 픽셀들은 IBC 모드에서 예측에 사용될 수도 있다. 필터링에 필요한 적어도 하나의 픽셀이 이용 불가인 경우,

필터링되지 않은 샘플들이 IBC 예측에 사용된다. 이 경우, IBC 예측을 위해 저장될 필터링되지 않은 픽셀의 양이 감소되기 때문에 대역폭이 감소될 수도 있다. 대안적으로, IBC 는 필터링된 샘플들만을 참조할 수 있다; 이 경우, 필터링되지 않은 샘플들 (예를 들어, 아직 재구성되지 않은 적어도 하나의 픽셀을 필요로 하는 샘플들) 은 IBC 예측을 위해 저장될 필요가 없을 수도 있다. 이러한 제한은, 예를 들어, 예측 블록을 도출하는데 사용된 MV가 그러한 샘플을 포함하지 않는 인코더 또는 비트 스트림 제약에 의해 달성될 수 있다.

[0155] 위의 예에서, 이웃 CU가 예측에 사용될 수 있기 때문에 IBC 예측과 함께 사용될 필터링되거나 또는 필터링되지 않은 픽셀에 대한 체크가 복잡할 수도 있다. 도출 프로세스를 단순화하기 위해, 현재 블록을 포함하는 현재 CTU의 필터링되지 않은 샘플들이 IBC 예측을 위해 사용될 수 있고, 현재 CTU 외부에 위치된 필터링된 샘플이, 이용 가능한 경우, IBC 예측을 위해 사용된다. 이 경우, 현재의 CTU 샘플들이 캐시에 저장될 수 있고 인코더 (20) 또는 디코더 (30) 의 외부 메모리에 저장될 필요가 없기 때문에, 필터링되지 않은 픽셀의 저장과 연관된 대역폭이 감소될 수도 있다.

[0156] IBC 예측을 위해 필터링된 샘플 또는 필터링되지 않은 샘플 간의 선택이 적용되는지 여부를 표시하기 위해 플래그가 도입될 수 있다. 이 플래그는 IBC 모드가 사용될 때 시그널링될 수 있으며 비트스트림에 존재하지 않을 경우 디스에이블되는 것으로 도출될 수 있다.

[0157] 대안적으로, IBC 예측을 위해 필터링된 또는 필터링되지 않은 샘플들 사이의 선택을 표시하기 위한 시그널링은 (CTU, 슬라이스, 타일의 단위에서) 지연을 기준으로 이루어질 수 있다. 예를 들어, 지연 값 이전의 블록들로부터의 IBC 예측은 필터링된 샘플들만을 사용할 수도 있고 지연 값 이후에는 필터링되지 않은 샘플들을 사용할 수도 있다. 위의 예들에서 설명된 양태들은 독립적으로 또는 다른 설명된 방법들과 임의의 조합으로 사용될 수 있다.

[0158] 본 개시는 대역폭 감소를 낳을 수 있는 기술을 설명한다. IBC 모드의 기존 구현에서, 비디오 디코더는 예측 샘플을 필터링하지 않는다; 그래서, IBC 모드가 인에이블되는 경우, 필터링된 샘플들에 추가하여, 별도의 필터링되지 않은 샘플들도 저장될 필요가 있으며, 이는 IBC 모드가 사용되는 경우 적어도 저장 대역폭이 배가되는 것을 낳는다.

[0159] 본 개시의 기술에 따르면, 필터링되지 않은 샘플들에 의해 야기된 대역폭 증가를 감소시키기 위해, 필터링되지 않은 샘플들을 저장하는 대신, 비디오 디코더 (30) 는 필터링된 샘플과 필터링되지 않은 샘플 간의 차이를 저장할 수도 있다. 필터 연산 (예 : 디블록, SAO) 은 작은 값 변화만을 야기하므로, 필터링된 샘플과 필터링되지 않은 샘플 사이의 차이는 저장을 위한 전체 비트 심도를 필요로 하지 않을 수도 있으며, 이는 IBC 모드와 함께 사용하기 위한 필터링되거나 필터링되지 않은 샘플들 양자 모두를 저장하는데 필요한 저장 대역폭을 감소시키는데 도움이 될 수도 있다. 또한, 일례에서, 이웃들로부터의 예측 및 가변 길이 코딩과 같은 무손실 압축일 수 있는 일부 압축 방식이 대역폭 및 저장 공간을 더 감소시키기 위해 적용될 수 있다. 이러한 구현에서, 비디오 디코더 (30) 는 필터링된 재구성된 블록을 생성하기 위해 비디오 데이터의 재구성된 블록에 대해 루프 필터링 동작을 수행하고, 필터링된 재구성된 블록과 재구성된 블록 사이의 차이를 표시하는 차이 값을 저장하고, 저장된 차이 값에 적어도 부분적으로 기초하여 블록을 예측하도록 구성될 수도 있다.

[0160] 본 개시는 또한 예측 샘플들에 필터링을 적용하는 기술을 설명한다. 상기 섹션에서 언급한 바와 같이, IBC 모드에 대한 현재의 설계에서, 예측 샘플들은 현재 화상의 필터링되지 않은 재구성된 (사전 SAO 또는 디블로킹) 샘플들이다. 이러한 필터링되지 않은 샘플들은 디스플레이하기 위해 전송된 필터링된 샘플에 추가하여 저장될 필요가 있다. 이 섹션에서는, 필터링되지 않은 샘플을 추가로 저장하기 위한 필요성을 피하기 위하여 일부 방법이 제안된다.

[0161] 본 개시의 제 1 기술에 따르면, 비디오 코더 (예를 들어, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 재구성된 샘플 대신에 예측 샘플에 SAO 및 디블로킹을 적용할 수도 있다. 예시적인 구현은 아래에 제공된다. 예시적인 구현는 현재의 설계를 본 개시에서 제안된 설계의 일례와 비교한다.

[0162] 현재 설계: 최종 재구성 = Filtered (Pred + 잔차)

[0163] 제안된 설계: 최종 재구성= Filtered (Pred) + 잔차.

[0164] 식중에서:

[0165] Filtered : 일방 또는 양자 모두의 프로세스, SAO 및 디블로킹을 나타낸다Pred : 필터링되지 않은 예측 샘플들을 나타낸다.

[0166] 위에서 설명된 제 1 기술과 관련하여, 비디오 코더는 SAO 및 디블록킹을 임의의 모드 (예를 들어, 인터, IBC, 인트라) 의 예측 샘플에 적용할 수도 있다. 제 2 기술에 따르면, 비디오 코더는 IBC 모드의 예측 샘플에 SAO 및 디블로킹을 적용하기 위해 전술한 기술들을 적용할 수도 있다. 이 기술은 JCTVC-T0045 에 설명된 기술에 추가하여 적용될 수도 있다. T0045에서, 참조로서 IBC 에 사용될 수 있는 CTU 에 대하여 필터링이 디스에이블된다. 이는 객관적이고 주관적인 품질 저하를 초래할 수도 있다. 따라서, 본 개시는 IBC 모드에 대해서만 예측 샘플에 필터링을 적용하기 위한 기술을 도입한다.

[0167] 예시적인 구현이 이제 설명될 것이다. 본 개시는 IBC 모드에 대한 재구성된 샘플 대신에 예측 샘플에 SAO 및 디블록킹을 적용하기 위한 기술을 설명한다. 또한, CTU가 IBC에 대한 참조 구역으로서 사용될 경우, 비트 스트림에서 플래그가 시그널링될 수도 있으며, 이 경우 플래그는 그 CTU 에 대한 인루프 후처리 (디블록킹 및 SAO) 를 스위치 오프하여, 추가적인 저장을 피한다.

[0168] 현재 설계:

[0169] 최종 재구성 = Filtered (Pred + 잔차)

[0170] 제안된 설계:

[0171] IBC 모드를 위한 최종 재구성= Filtered (Pred) + 잔차.

[0172] 비 IBC 모드를 위한 최종 재구성= Filtered (Pred + 잔차)

[0173] 식중에서:

[0174] Filtered : 일방 또는 양자 모두의 프로세스, SAO 및 디블로킹을 나타낸다Pred : 필터링되지 않은 예측 샘플들을 나타낸다.

[0175] 전술한 바와 같이 예측 샘플에 필터링을 적용하는 일부로서, 비디오 디코더 (30) 는 비디오 데이터의 블록에 대한 예측 블록을 결정하고, 예측 블록을 필터링하여 필터링된 예측 블록을 생성하고, 잔차 데이터를 필터링된 예측 블록에 추가하여 재구성된 블록을 생성하도록 구성될 수도 있다. 예측 블록을 필터링하기 위해, 비디오 디코더 (30) 는 SAO 필터링, 디블록 필터링 또는 ALF 필터링 중 하나 이상을 수행할 수도 있다. 비디오 디코더 (30) 는 IBC 모드를 사용하여 코딩되는 비디오 데이터의 블록에 응답하여 필터링된 예측 블록을 생성하기 위해 예측 블록을 필터링할 수도 있다. IBC 모드외의 코딩 모드에서 코딩된 비디오 데이터의 제 2 블록에 대해, 비디오 디코더 (30) 는 잔차 데이터를 제 2 블록에 대한 제 2 예측 블록에 추가하여 제 2 재구성된 블록을 생성하고 제 2 재구성된 블록을 필터링할 수도 있다.

[0176] 도 3은 본 개시에 기재된 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 비디오 인코더 (20) 는, 비디오 슬라이스들 내의 비디오 블록들의 인트라 코딩 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은, 정해진 비디오 프레임 또는 화상 내의 비디오에서 공간적 중복성을 감소 또는 제거하기 위하여 공간적 예측에 의거한다. 인터 코딩은 비디오 시퀀스의 인접 프레임들 또는 화상들 내의 비디오에서 시간적 중복성을 감소 또는 제거하기 위하여 시간적 예측에 의거한다. 인트라 모드 (I 모드) 는 여러 공간 기반 압축 모드들 중 어느 것을 나타낼 수도 있다. 인터 모드들, 이를테면 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 은, 여러 시간 기반 압축 모드들 중 어느 것을 나타낼 수도 있다.

[0177] 도 3의 예에서, 비디오 인코더 (20) 는 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 처리 유닛 (41), 합산기 (50), 변환 처리 유닛 (52), 양자화 유닛 (54), 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 처리 유닛 (41) 은 모션 추정 유닛 (MEU; 42), 모션 보상 유닛 (MCU; 44) 및 인트라 예측 처리 유닛 (46), 및 IBC 유닛 (48) 을 포함한다. 설명을 용이하게 하기 위해 도 3에는 개별적으로 도시되어 있지만, MEU (42), MCU (44), 인트라 예측 처리 유닛 (46) 및 IBC 유닛 (48) 은 실제로 고도로 통합될 수도 있음을 이해해야 한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역 양자화 유닛 (58), 역 변환 처리 유닛 (60), 합산기 (62), 필터 유닛 (64), 및 디코딩된 화상 버퍼 (DPB; 66) 를 포함한다.

[0178] 도 3에 도시된 바처럼, 비디오 인코더 (20) 는 비디오 데이터를 수신하고 그 수신된 비디오 데이터를 비디오 데이터 메모리 (33) 에 저장한다. 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 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, 예를 들어, 인트라- 또는 인터-코딩 모드들에서 비디오 인코더 (20) 에 의한 비디오 데이터를 인코딩하는데 이용하기 위한 참조 비디오 데이터를 저장하는 참조 화상 메모리일 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66) 는 동기 DRAM (SDRAM) 을 포함

하는, 동적 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 타입의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중의 어느 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 함께 온칩 (on-chip) 될 수도 있거나, 또는 그러한 컴포넌트들에 대해 오프칩 (off-chip) 될 수도 있다.

[0179] 분할 유닛 (35) 은 비디오 데이터 메모리 (33) 로부터 비디오 데이터를 취출하고, 예측 처리 유닛 (41) 과 함께, 비디오 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한, 예를 들어, LCU 및 CU 들의 쿼드트리 구조에 따른, 비디오 블록 파티셔닝 뿐만 아니라 슬라이스들, 타일들 또는 다른 더 큰 유닛들로의 파티셔닝을 포함할 수도 있다. 설명을 위해 따로따로 도시되었지만, 파티셔닝 유닛 (35) 및 예측 처리 유닛 (41) 은 고도로 통합될 수도 있고, 파티셔닝 유닛 (35) 및 예측 처리 유닛 (41) 양자 모두는 비디오의 화상을 파티셔닝하는 방법을 결정하는 프로세스의 양태들을 수행할 수도 있다. 비디오 인코더 (20) 는 일반적으로, 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 슬라이스는, 다수의 비디오 블록들로 (그리고 가능하게는 타일들로 지칭되는 비디오 블록들의 세트들로) 분할될 수도 있다. 예측 처리 유닛 (41) 은, 복수의 가능한 코딩 모들 중 하나, 이를테면 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나를, 에러 결과 (예를 들어, 코딩 레이트 및 왜곡의 레벨) 에 기초하여 현재 비디오 블록에 대해 선택할 수도 있다. 예측 처리 유닛 (41) 은 결과적인 인트라 또는 인터 코딩 블록을 합산기 (50) 에 제공하여 잔차 블록 데이터를 생성하고 합산기 (62) 에 제공하여 참조 화상으로서의 사용을 위해 인코딩된 블록을 재구성할 수도 있다.

[0180] 예측 처리 유닛 (41) 내의 인트라 예측 처리 유닛 (46) 은 코딩될 현재 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 대해 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수도 있다. 예측 처리 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 하나 이상의 참조 화상들에서의 하나 이상의 예측 블록들에 대해 현재 비디오 블록의 인터 예측 코딩을 수행해 시간적 압축을 제공한다. 모션 추정 유닛 (42) 에 의해 수행되는, 모션 추정은, 비디오 블록들을 위한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 예를 들어, 모션 벡터는, 참조 화상 내의 예측 블록에 대해 현재 비디오 프레임 또는 화상 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다.

[0181] 예측 블록은, 절대 차이의 합 (sum of absolute difference; SAD), 제곱 차이의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 화소 차이에 관하여, 코딩될 비디오 블록의 PU에 밀접하게 매칭하는 것으로 구해진 블록이다. 일부 예들에서, 비디오 인코더 (20) 는 DPB (66) 에 저장된 참조 화상들의 서브 정수 픽셀 위치 (sub-integer pixel position) 들을 위한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 화상의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 보간할 수도 있다. 그러므로, 모션 추정 유닛 (42) 은, 전 (full) 픽셀 위치들 그리고 분수 픽셀 위치들에 대해 모션 탐색을 수행하고 분수 픽셀 정밀도로 모션 벡터를 출력할 수도 있다.

[0182] 모션 추정 유닛 (42) 은, PU 의 위치와 참조 화상의 예측 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서 비디오 블록의 PU를 위한 모션 벡터를 계산한다. 참조 화상은, 제 1 참조 화상 리스트 (리스트 0) 또는 제 2 참조 화상 리스트 (리스트 1) 로부터 선택될 수도 있고, 이들의 각각은 DPB (66) 에 저장된 하나 이상의 참조 화상들을 식별한다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0183] 모션 보상 유닛 (44) 에 의해 수행된 모션 보상은, 모션 추정에 의해 결정된 모션 벡터에 기초한 예측 블록의 페칭 (fetching) 또는 생성을 수반할 수도 있으며, 가능하게는 서브-픽셀 정밀도로 보간을 수행한다. 현재 비디오 블록의 PU를 위한 모션 벡터의 수신시에, 모션 보상 유닛 (44) 은, 모션 벡터가 참조 화상 리스트들 중 하나에서 가리키는 예측 블록을 로케이팅할 수도 있다. 비디오 인코더 (20) 는, 코딩되는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여, 픽셀 차이 값들을 형성함으로써, 잔차 비디오 블록을 형성한다. 픽셀 차이 값들은, 블록을 위한 잔차 데이터를 형성하고, 루마 및 크로마 차이 성분들 양자 모두를 포함할 수도 있다. 합산기 (50) 는 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 모션 보상 유닛 (44) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0184] 일부 예들에서, IBC 유닛 (48) 은 2차원 벡터들을 생성하고 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 대해 위에서 설명된 것과 유사한 방식으로 예측 블록들을 페칭할 수도 있지만, 그 예측 블록들은 현재 블록과 동

일한 화상 또는 프레임이다. 다른 예들에서, IBC 유닛 (48) 은, 전체적으로 또는 부분적으로, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 을 이용하여, 여기에 기재된 기법들에 따라 IBC 예측을 위한 그러한 기능들을 수행할 수도 있다. 어느 경우든, IBC 에 대해, 예측 블록은, SAD, SSD, 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 픽셀 차이의 측면에서, 코딩될 블록과 밀접하게 매치하는 것으로 구해진 블록일 수도 있고 그 블록의 식별은 서브 정수 픽셀 위치들을 위한 값들의 계산을 포함할 수도 있다.

[0185] 예측 처리 유닛 (41) 이 인트라 예측, 인터 예측, 또는 IBC 를 통해 현재 비디오 블록을 위한 예측 블록을 생성한 후에, 비디오 인코더 (20) 는 현재 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 잔차 블록에서 잔차 비디오 데이터는 하나 이상의 TU 들에 포함될 수도 있고 변환 처리 유닛 (52) 에 적용될 수도 있다. 변환 처리 유닛 (52) 은 DCT (discrete cosine transform) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 잔차 변환 계수들로 잔차 비디오 데이터를 변환한다. 변환 처리 유닛 (52) 은 잔차 비디오 데이터를 픽셀 도메인으로부터 변환 도메인, 이를테면 주파수 도메인으로 변환할 수도 있다.

[0186] 변환 처리 유닛 (52) 은 양자화 유닛 (54) 에 결과적인 변환 계수들을 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킨다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화도 (degree of quantization) 는 양자화 파라미터를 조절함으로써 수정될 수도 있다. 일부의 예들에서, 다음으로, 양자화 유닛 (54) 은 양자화된 변환 계수들을 포함하는 행렬의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56) 이 스캔을 수행할 수도 있다.

[0187] 양자화 다음에, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 CAVLC (context adaptive variable length coding), CABAC (context adaptive binary arithmetic coding), SBAC (syntax-based context-adaptive binary arithmetic coding), PIPE (probability interval partitioning entropy) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩 다음에, 인코딩된 비트스트림은, 비디오 디코더 (30) 로 송신되거나 또는 비디오 디코더 (30) 에 의한 나중의 송신 또는 취출을 위해 보관될 수도 있다.

엔트로피 인코딩된 유닛 (56) 은 또한, 모션 벡터들 그리고 코딩되고 있는 현재 비디오 슬라이스를 위한 다른 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0188] 역 양자화 유닛 (58) 및 역 변환 처리 유닛 (60) 은 역 양자화 및 역 변환을 각각 적용하여, 참조 화상의 참조 블록으로서 나중에 사용하기 위해 픽셀 도메인에서 잔차 블록을 재구성한다. 모션 보상 유닛 (44) 은, 참조 화상 리스트들 중의 하나 내의 참조 화상들 중의 하나의 참조 화상의 예측 블록에 잔차 블록을 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한 하나 이상의 보간 필터들을 그 재구성된 잔차 블록에 적용하여, 모션 추정에 사용하기 위한 서브 정수 픽셀 값들을 산출할 수도 있다. 합산기 (62) 는 재구성된 잔차 블록을 모션 보상 유닛 (344) 에 의해 생성되는 모션 보상된 예측 블록에 가산하여 재구성된 블록을 생성한다.

[0189] 필터 유닛 (64) 은 재구성된 블록 (예를 들면, 합산기 (62) 의 출력) 을 필터링하고, 그 필터링된 재구성된 블록을 참조 블록으로서 사용하기 위해 DPB (66) 에 저장한다. 참조 블록은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 참조 블록으로서 사용되어 후속 비디오 프레임 또는 화상에서 블록을 인터 예측할 수도 있다. 필터 유닛 (64) 은 디블로킹 필터링, 샘플 적응 오프셋 (SAO) 필터링, 적응 루프 필터링 (ALF), 또는 다른 유형의 루프 필터링 중 하나 이상을 적용할 수도 있다. 필터 유닛 (64) 은 재구성된 비디오로부터 블로키니스 (blockiness) 아티팩트를 제거하기 위해 필터 블록 경계에 디블로킹 필터링을 적용할 수도 있고, 전체 코딩 품질을 향상시키기 위해 다른 유형의 필터링을 적용할 수도 있다. 추가적인 루프 필터들 (인 루프 또는 포스트 루프) 이 또한 사용될 수도 있다.

[0190] 비디오 인코더 (20) 는 본 개시의 기술을 수행할 수도 있는 비디오 인코더의 하나의 그러나 유일한 것은 아닌 예를 나타낸다. 비디오 인코더 (20), 예를 들어, IBC 유닛 (48) 은, 예를 들어, 비디오 데이터의 슬라이스를 I 슬라이스로서 인코딩할 수도 있다. 비디오 인코더 (20) 는, 예를 들어, 코딩된 비디오 시퀀스에 대해 여러 코딩 패스들을 수행하고, I 슬라이스로서 슬라이스를 코딩하는 것이 바람직한 레이트-왜곡 트레이드오프와 같은 바람직한 코딩 특성을 제공한다는 것을 결정할 수도 있다. 슬라이스에 대해, 비디오 인코더 (20), 예를 들어, IBC 유닛 (48) 및 엔트로피 인코딩 유닛 (56) 은 또한 IBC 모드가 슬라이스에 대해 인에이블됨을 표시하기 위해 IBC 신택스 엘리먼트를 생성할 수도 있다. 슬라이스에 대하여, 비디오 인코더 (20), 예를 들어, IBC 유닛 (48) 은 IBC 모드를 사용하여 슬라이스의 적어도 하나의 블록을 인코딩할 수도 있고, 비디오 인코더

(20), 예를 들어, 인트라 예측 처리 유닛 (46) 은, 인트라 모드를 사용하여 슬라이스의 적어도 하나의 블록을 인코딩할 수도 있다. 비디오 인코더 (20), 예를 들어 엔트로피 인코딩 유닛 (56) 은 SPS 내에 IBC 신택스 엘리먼트를 포함할 수도 있다. 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수가 1과 동일한 것을 결정하는 것에 응답하여 그리고 슬라이스에 대해 IBC 모드가 인에이블됨에 응답하여, 비디오 인코더 (20), 예를 들어 예측 처리 유닛 (41) 은 참조 화상 리스트 수정을 디스에이블할 수도 있다. 참조 화상 리스트 수정을 디스에이블하는 것의 일부로서, 비디오 인코더 (20) 는 인코딩된 비트스트림 내의 슬라이스에 대한 참조 화상 리스트 수정 정보를 포함하지 않을 수도 있다.

[0191] 도 4는 본 개시에 기재된 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 4의 예에서, 비디오 디코더 (30) 는, 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 예측 처리 유닛 (81), 역 양자화 유닛 (86), 역 변환 처리 유닛 (88), 합산기 (90), 필터 유닛 (92), 및 DPB (94) 를 포함한다. 예측 처리 유닛 (81) 은 모션 보상 유닛 (82), 인트라 예측 유닛 (84) 및 IBC 유닛 (85) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서, 도 3 로부터의 비디오 인코더 (20) 에 대해 설명된 인코딩 패스에 일반적으로 상반되는 디코딩 패스를 수행할 수도 있다.

[0192] 디코딩 프로세스 동안, 비디오 디코더 (30) 는, 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 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 컴포넌트들에 의해 디코딩될 비디오 데이터, 이를테면 인코딩된 비디오 비트스트림을 저장할 수도 있다. 비디오 데이터 메모리 (78) 에 저장된 비디오 데이터는 예를 들어 링크 (16) 를 통해, 저장 디바이스 (26) 로부터, 또는 카메라와 같은 로컬 비디오 소스로부터, 또는 물리적 데이터 저장 매체에 액세스함으로써, 획득될 수도 있다. 비디오 데이터 메모리 (78) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 화상 버퍼 (CPB) 를 형성할 수도 있다. DPB (94) 는, 예를 들어, 인트라- 또는 인터-코딩 모드들에서 비디오 디코더 (30) 에 의한 비디오 데이터를 디코딩하는데 이용하기 위한 참조 비디오 데이터를 저장하는 참조 화상 메모리일 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94) 는 DRAM, SDRAM, MRAM, RRAM, 또는 다른 타입의 메모리 디바이스들 등의 다양한 메모리 디바이스들 중의 어느 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 함께 온칩 (on-chip) 일 수도 있거나, 또는 그러한 컴포넌트들에 대해 오프칩 (off-chip) 일 수도 있다.

[0193] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 양자화된 계수들, 모션 벡터들, 및 다른 신택스 엘리먼트들을 생성하기 위하여 비디오 데이터 메모리 (78) 에 저장된 비디오 데이터를 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 예측 처리 유닛 (81) 에 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.

[0194] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩될 때, 예측 처리 유닛 (81) 의 인트라 예측 처리 유닛 (84) 은, 현재 프레임 또는 화상의 이전에 디코딩된 블록들로부터 시그널링된 인트라 예측 모드 및 데이터에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터 코딩된 것으로서 코딩될 때, 예측 처리 유닛 (81) 의 모션 보상 유닛 (82) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 블록들을 생성한다. 예측 블록들은 참조 화상 리스트들 중의 하나의 리스트 내의 참조 화상들 중의 하나의 참조 화상로부터 생성될 수도 있다. 비디오 디코더 (30) 는 DPB (94) 에 저장된 참조 화상들에 기초하여 디폴트 구축 기법들을 이용하여 참조 프레임 리스트들, 리스트 0 및 리스트 1 을 구축할 수도 있다.

[0195] 모션 보상 유닛 (82) 및/또는 IBC 유닛 (85) 은, 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱 (parsing) 하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 정보를 결정하고, 그 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록을 위한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82) 은 수신된 신택스 엘리먼트들의 일부를 사용하여 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예를 들어, 인트라 또는 인터 예측 또는 IBC), 슬라이스를 위한 참조 화상 리스트들의 하나 이상을 위한 구축 정보, 슬라이스의 각 인터 인코딩된 비디오 블록을 위한 모션 벡터들, 슬라이스의 각 인터 코딩된 비디오 블록을 위한 인터 예측 상태, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0196] 모션 보상 유닛 (82) 및/또는 IBC 유닛 (85) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (82) 및/또는 IBC 유닛 (85) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 이용되는 보간 필터들을 이용하여 참조 블록들의 서브 정수 픽셀들을 위한 보간된 값들을 계산할 수도 있다. 이 경우에, 모션 보상 유닛 (82) 및/또는 IBC 유닛 (85) 은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 이용된 보간 필터들을 결정하고, 그 보간 필터들을 이용하여 예측 블록들을 생성할 수도 있다.

[0197] 역 양자화 유닛 (86) 은 비트스트림에서 제공되고 엔트로피 디코딩 유닛 (80) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉 탈양자화한다. 역 양자화 프로세스는, 양자화의 정도, 그리고, 마찬가지로, 적용되어야 하는 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서 각 비디오 블록을 위해 비디오 인코더 (20) 에 의해 산출된 양자화 파라미터의 이용을 포함할 수도 있다. 역 변환 처리 유닛 (88) 은, 픽셀 도메인에서 잔차 블록들을 생성하기 위해 변환 계수들에, 역 변환, 예를 들어, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 적용한다.

[0198] 예측 처리 유닛이, 예를 들어 인트라 또는 인터 예측을 사용하여 현재 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30) 는 역 변환 처리 유닛 (88) 으로부터의 잔차 블록을 모션 보상 유닛 (82) 에 생성된 대응하는 예측 블록과 합산함으로써 재구성된 비디오 블록을 형성한다. 합산기 (90) 는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 필터 유닛 (92) 은, 예를 들어 디블로킹 필터링, SAO 필터링, ALF 필터링, 또는 다른 유형의 필터링 중 하나 이상을 사용하여 재구성된 비디오 블록을 필터링한다. (코딩 루프내 또는 코딩 루프 후 중 어느 일방의) 다른 루프 필터들이 또한 픽셀 천이들을 매끄럽게 하거나 또는 다른 방법으로 비디오 품질을 향상시키는데 사용될 수도 있다. 다음으로, 주어진 프레임 또는 화상에서 디코딩된 비디오 블록들은 DPB (94) 에 저장되고, 이는 후속 모션 보상을 위해 사용된 참조 화상들을 저장한다. DPB (94) 는 또한, 도 1의 디스플레이 디바이스 (32) 등의 디스플레이 디바이스 상에 나중에 표출하기 위해 디코딩된 비디오를 저장하는 추가 메모리의 부분일 수도 있거나 또는 이로부터 분리될 수도 있다.

[0199] 비디오 디코더 (30) 는 본 개시의 기술을 수행할 수도 있는 비디오 디코더의 하나의 그러나 유일한 것은 아닌 예를 나타낸다. 비디오 디코더 (30), 예를 들어, 비디오 데이터 메모리 (78) 및 엔트로피 디코딩 유닛 (80) 은 비디오 데이터의 슬라이스를 수신할 수도 있다. 비디오 디코더 (30), 예를 들어, 엔트로피 디코딩 유닛 (80) 및 예측 처리 유닛 (81) 은 IBC 신택스 엘리먼트를 파싱하여 IBC 모드가 슬라이스에 대해 인에이블되는 것을 결정할 수도 있다. 비디오 디코더 (30), 예를 들어, 엔트로피 디코딩 유닛 (80) 및 예측 처리 유닛 (81) 은 슬라이스가 I 슬라이스인지를 결정하기 위해 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱할 수도 있다. 비디오 디코더 (30), 예를 들어 인트라 예측 처리 유닛 (84) 및 IBC 유닛 (85) 은 I 슬라이스로서 슬라이스를 디코딩할 수도 있다. 슬라이스를 I 슬라이스로서 디코딩하기 위해, 비디오 디코더 (30) 는 인트라 예측만을 사용하여 그리고 인터 예측 모드 또는 IBC 모드를 사용하지 않고서 슬라이스의 블록들을 디코딩할 수도 있다. 비디오 디코더 (30), 예를 들어 엔트로피 디코딩 유닛 (80) 은 SPS 내에 IBC 신택스 엘리먼트를 수신할 수도 있다. 슬라이스의 적어도 하나의 블록에 대해 IBC를 수행하기 위해, 비디오 디코더 (30), 예를 들어, IBC 유닛 (85) 은 참조 화상으로서 슬라이스를 포함하는 현재 화상을 사용할 수도 있다.

[0200] 비디오 디코더 (30), 예를 들어, 예측 처리 유닛 및 DPB (94) 는 예를 들어, 비디오 데이터의 슬라이스에 대한 참조 화상 리스트를 구축할 수도 있다. 참조 화상 리스트에 포함된 화상의 수는 인에이블되는 IBC 에 의존할 수도 있다. 비디오 디코더 (30), 예를 들어, 예측 처리 유닛 (81) 은, 비디오 데이터의 슬라이스에 대한 참조 화상 리스트를 구축할 수도 있다. IBC 모드가 인에이블되는 것에 응답하여, 비디오 디코더 (30), 예를 들어 예측 처리 유닛 및 DPB (94) 는 슬라이스를 포함하는 화상을 참조 화상 리스트에 추가할 수도 있다.

[0201] 비디오 디코더 (30), 예를 들어, 엔트로피 디코딩 유닛 (80) 및 예측 처리 유닛 (81) 은, 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상의 수를 표시하는 신택스 엘리먼트를 수신할 수도 있다. 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수를 표시하는 신택스 엘리먼트가 1과 동일함에 응답하여, 그리고 슬라이스에 대해 IBC 모드가 인에이블되는 것에 응답하여, 비디오 디코더 (30), 예를 들어 예측 처리 유닛 및 DPB (94) 는 참조 화상 리스트 내의 제 1 엔트리로서 슬라이스를 포함하는 현재 화상을 포함하는 비디오 데이터의 슬라이스에 대한 참조 화상 리스트를 구축할 수도 있다.

[0202] 비디오 디코더 (30), 예를 들어, 엔트로피 디코딩 유닛 (80) 및 예측 처리 유닛 (81) 은, 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상의 수를 표시하는 신택스 엘리먼트를 수신할 수도 있다. 신택스 엘리먼트가 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수가 1과 동일한 것을 표시하는 것에 응답하여 그리고 슬라이스에 대해 그 IBC 모드가 인에이블됨에 응답하여, 비디오 디코더 (30),

예를 들어 예측 처리 유닛 (81) 은 참조 화상 리스트 수정을 디스에이블할 수도 있다. 참조 화상 리스트 수정을 디스에이블하는 것의 일부로서, 비디오 디코더 (30), 예를 들어, 예측 처리 유닛 (81) 은, 참조 화상 리스트 수정 정보를 수신하지 않고서 참조 화상 리스트를 구축할 수도 있다.

[0203] 도 5는 본 개시의 기술들에 따른 비디오 데이터를 인코딩하는 방법을 보여주는 플로우차트이다. 도 5는 일반적 비디오 인코더에 관하여 설명될 것이다. 도 5의 예에서, 비디오 인코더는 I 슬라이스로서 비디오 데이터의 슬라이스를 인코딩한다 (150). 슬라이스에 대하여, 비디오 인코더는 또한 IBC 모드가 슬라이스에 대해 인에이블됨을 표시하기 위해 IBC 신택스 엘리먼트를 생성한다 (152). 슬라이스에 대하여, 비디오 인코더는 IBC 모드를 사용하여 슬라이스의 적어도 하나의 블록을 인코딩할 수도 있고 인트라 모드를 사용하여 슬라이스의 적어도 하나의 블록을 인코딩할 수도 있다. 비디오 인코더는 슬라이스와 연관된 SPS 에 포함시키기 위해 IBC 신택스 엘리먼트를 생성할 수도 있다. 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수가 1과 동일한 것을 결정하는 것에 응답하여 그리고 슬라이스에 대해 IBC 모드가 인에이블됨에 응답하여, 비디오 인코더는 참조 화상 리스트 수정을 디스에이블할 수도 있다. 참조 화상 리스트 수정을 디스에이블하는 것의 일부로서, 비디오 인코더는 인코딩된 비트스트림 내의 슬라이스에 대한 참조 화상 리스트 수정 정보를 포함하지 않을 수도 있다.

[0204] 도 6은 본 개시의 기술들에 따른 비디오 데이터를 인코딩하는 방법을 보여주는 플로우차트이다. 도 6은 일반적 비디오 디코더에 관하여 설명될 것이다. 일반 비디오 디코더는, 예를 들어, 비디오 디코더 (30) 에 대응할 수도 있지만, 본 개시의 기술은 임의의 특정 유형의 비디오 디코더에 제한되지 않는다. 비디오 디코더는 비디오 데이터의 슬라이스를 수신한다 (160). 비디오 디코더는 IBC 신택스 엘리먼트를 파싱하여 IBC 모드가 슬라이스에 대해 인에이블됨을 결정한다 (162). 비디오 디코더는 상기 슬라이스가 I 슬라이스임을 결정하기 위해 상기 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱한다 (164). 비디오 디코더는 슬라이스를 I 슬라이스로서 디코딩한다 (166). 슬라이스를 I 슬라이스로서 디코딩하기 위해, 비디오 디코더는 예를 들어, 인트라 예측 코딩 모드를 사용하여 슬라이스의 모든 블록을 디코딩할 수도 있다. 비디오 디코더는 SPS 에서 IBC 신택스 엘리먼트를 수신할 수도 있다. 슬라이스를 I 슬라이스로서 디코딩하기 위해, 비디오 디코더는 인트라 예측만을 사용하여 슬라이스의 블록을 디코딩한다. 즉, 슬라이스를 I 슬라이스로서 디코딩하기 위해서, 비디오 디코더는 인터 예측 모드를 사용하지 않고서 그리고 IBC 모드를 사용하지 않고서 슬라이스의 블록을 디코딩한다.

[0205] 예를 들어, IBC 신택스 엘리먼트는 전술한 curr_pic_as_ref_enabled_flag 일 수도 있고, IBC 신택스 엘리먼트를 파싱하여 IBC 모드가 슬라이스에 대해 인에이블됨을 결정하기 위하여, 비디오 디코더는 curr_pic_as_ref_enabled_flag의 값을 1과 동일한 것으로 결정할 수도 있다. 슬라이스 타입 신택스 엘리먼트는, 예를 들어 전술한 slice_type 신택스 엘리먼트일 수도 있고, 슬라이스가 I 슬라이스임을 결정하기 위해 슬라이스와 연관된 슬라이스 타입 신택스 엘리먼트를 파싱하기 위해, 비디오 디코더는 slice_type 신택스 엘리먼트의 값을 2와 동일한 것으로 결정할 수도 있다.

[0206] 도 6의 예에 따르면, 비디오 디코더는 또한 비디오 데이터의 제 2 슬라이스를 수신하고, 비디오 데이터의 제 2 슬라이스에 대한 참조 화상 리스트 내의 참조 화상의 수를 표시하는 신택스 엘리먼트를 수신할 수도 있고, 비디오 데이터의 제 2 슬라이스에 대한 참조 화상 리스트의 참조 화상의 수를 표시하는 신택스 엘리먼트가 1과 동일함에 응답하여, 그리고 IBC 모드가 제 2 슬라이스에 대해 인 에이블되는 것에 응답하여, 비디오 디코더는 슬라이스를 포함하는 현재 화상을 포함하는 비디오 데이터의 제 2 슬라이스에 대한 참조 화상 리스트를 구축할 수도 있다. 다른 예에서, 비디오 디코더는 비디오 데이터의 제 2 슬라이스를 수신하고, 비디오 데이터의 제 2 슬라이스에 대한 참조 화상 리스트를 구축할 수도 있고, IBC 모드가 인에이블되는 것에 응답하여, 제 2 슬라이스를 포함하는 화상을 참조 화상 리스트에 추가할 수도 있다.

[0207] 도 7은 본 개시의 기술들에 따른 비디오 데이터를 인코딩하는 방법을 보여주는 플로우차트이다. 도 7은 일반적 비디오 디코더에 관하여 설명될 것이다. 도 7의 기술은 도 6의 기술과 함께 사용될 수 있거나 또는 독립적으로 사용될 수도 있다. 일반 비디오 디코더는, 예를 들어, 비디오 디코더 (30) 에 대응할 수도 있지만, 본 개시의 기술은 임의의 특정 유형의 비디오 디코더에 제한되지 않는다. 비디오 디코더는 비디오 데이터의 슬라이스를 수신한다 (170). 비디오 디코더는 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상의 수를 표시하는 신택스 엘리먼트를 수신한다 (172). 비디오 데이터의 슬라이스에 대한 참조 화상 리스트 내의 참조 화상들의 수를 표시하는 신택스 엘리먼트가 1과 동일함에 응답하여, 그리고 IBC 모드가 제 2 슬라이스에 대해 인에이블되는 것에 응답하여, 비디오 디코더는 제 2 슬라이스에 대해 참조 화상 리스트에 대한 참조 화상 리스트 수정을 디스에이블한다 (174). 참조 화상 리스트 수정을 디스에이블하기 위해,

비디오 디코더는 예를 들어, 참조 화상 리스트 수정 정보를 수신하지 않고서 참조 화상 리스트를 구축하고, 참조 화상 리스트 내의 제 1 엔트리로서 슬라이스를 포함하는 현재 화상을 추가할 수도 있다.

[0208] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 그 기능들은 컴퓨터 판독가능 매체 상에 하나 이상의 명령들 또는 코드로서 저장되거나 송신될 수도 있고 하드웨어 기반 처리 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예를 들면, 통신 프로토콜에 따라, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체는 일반적으로, (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는, 본 개시에서 설명된 기법들의 구현을 위해 명령들, 코드 및/또는 데이터 구조들을 취출하기 위하여 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0209] 제한이 아닌 예로서, 이러한 컴퓨터-판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 희망하는 프로그램 코드를 저장하기 위해 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 라인 (DSL), 또는 적외선, 전파 (radio), 및 마이크로파와 같은 무선 기술을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되면, 그 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 전파, 및 마이크로파와 같은 무선 기술은 매체의 정의 내에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속, 캐리어 파, 신호 또는 다른 일시적 매체를 포함하는 것이 아니라, 대신에 비일시적, 유형의 저장 매체로 향하게 된다는 것이 이해되야 한다. 여기에 사용된, 디스크 (disk) 및 디스크 (disc) 는 CD (compact disc), 레이저 디스크 (laser disc), 광 디스크 (optical disc), DVD (digital versatile disc), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (Blu-ray disc) 를 포함하며, 여기서, 디스크 (disk) 는 보통 데이터를 자기적으로 재생하지만, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 또한, 상기의 조합은 컴퓨터 판독 가능 매체의 범위 내에 포함되어야 한다.

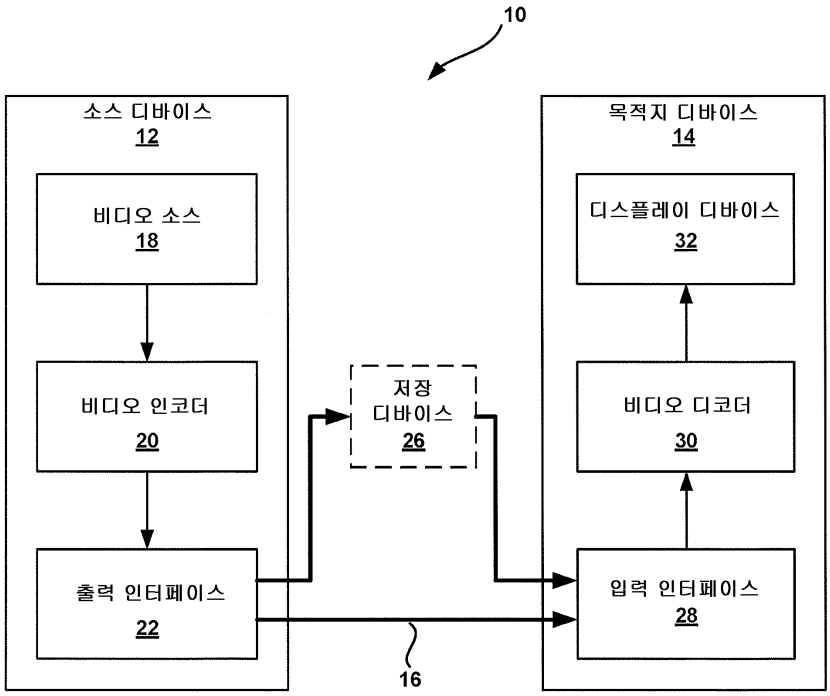
[0210] 명령들은, 고정된 기능 및/또는 프로그램 가능 처리 회로를 포함하는, 하나 이상의 프로세서, 이를테면 하나 이상의 DSP (digital signal processor), 범용 마이크로프로세서, ASIC (application specific integrated circuit), FPGA (field programmable logic array), 또는 다른 등가 집적 또는 이산 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에 사용된 용어 "프로세서" 는 전술한 구조 중 임의의 것 또는 본원에 설명된 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 추가로, 일부 양태들에서, 여기서 설명된 기능은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나 또는 결합된 코덱에 포함될 수도 있다. 또한, 그 기법들은 하나 이상의 회로 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0211] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들면, 칩 세트) 를 포함하여, 광범위하게 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들 또는 유닛들이, 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적인 양태들을 강조하기 위하여 본 개시에 설명되었지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하는 것은 아니다. 오히려, 상술된 바처럼, 다양한 유닛들이 코덱 하드웨어 유닛에 결합될 수도 있거나, 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상술된 하나 이상의 프로세서들을 포함하는 연동적인 (interoperative) 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

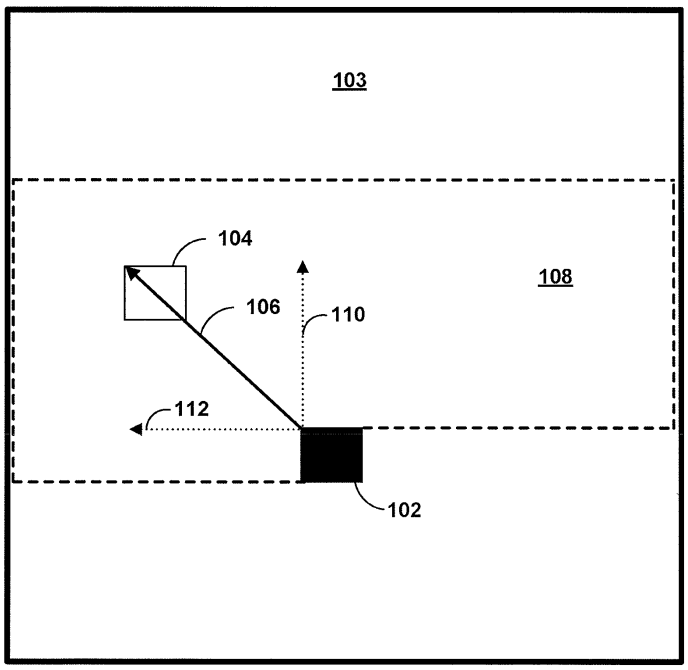
[0212] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

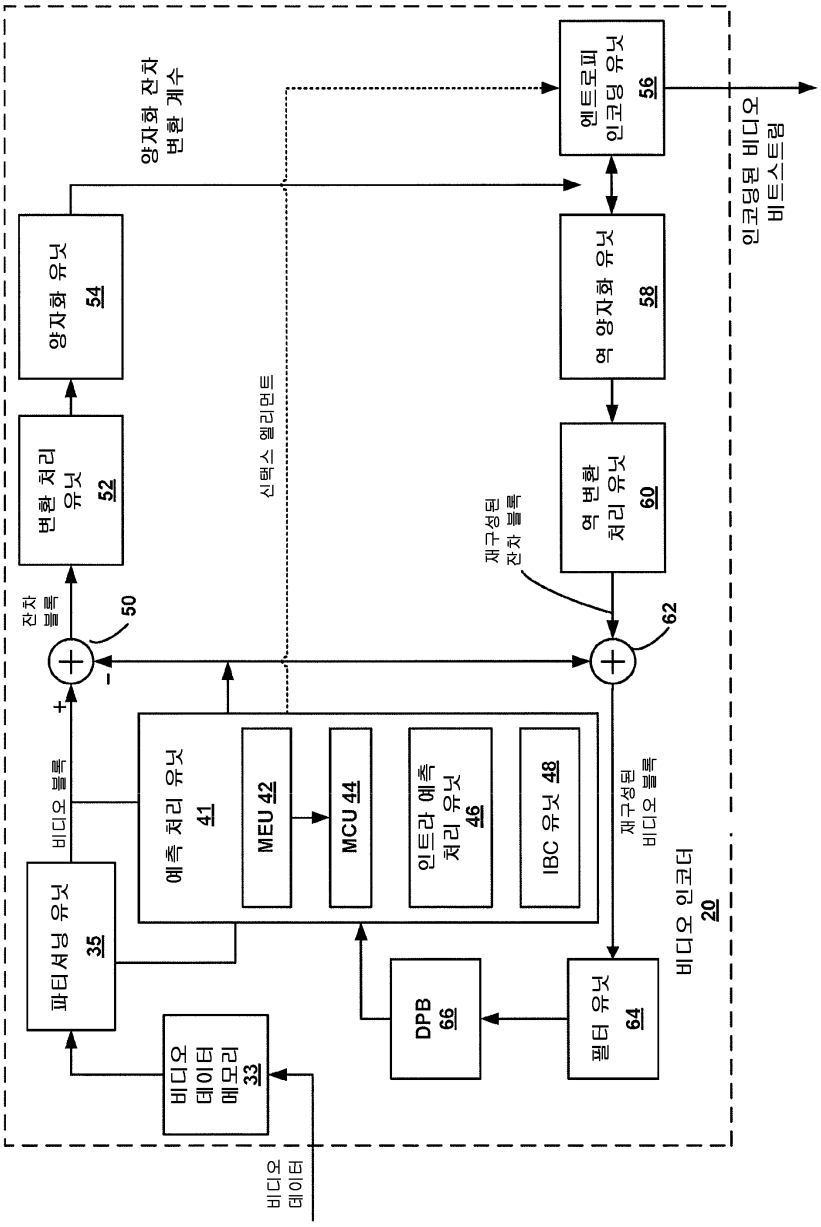
도면1


10
소스 디바이스
12
비디오 소스
18
비디오 인코더
20
출력 인터페이스
22
저장
디바이스
26
16
목적지 디바이스
14
디스플레이 디바이스
32
비디오 디코더
30
입력 인터페이스
28

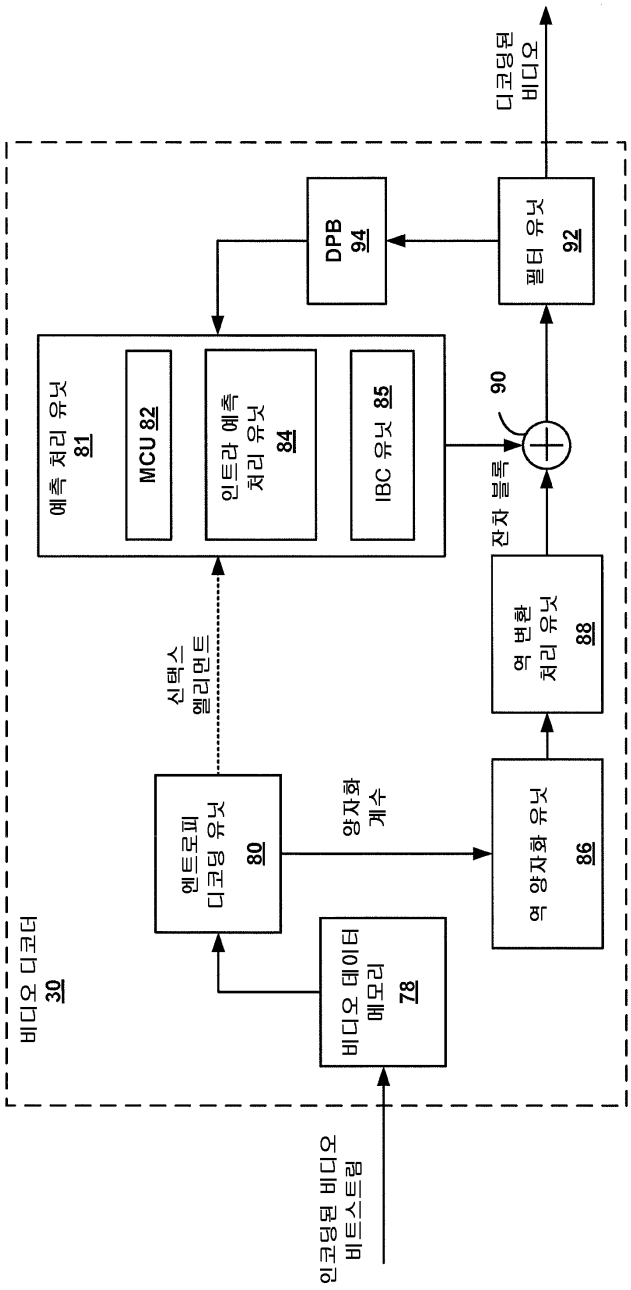
도면2


103
104
106
110
108
112
102

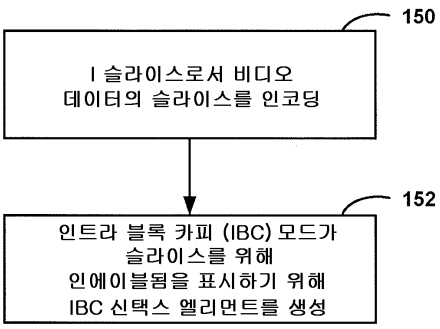
도면3


양자화 잔차
변환 계수
엔트로피
인코딩 유닛
56
인코딩된 비디오
비트스트림
양자화 유닛
54
역 양자화 유닛
58
변환 처리
유닛
52
신택스 엘리먼트
재구성된
잔차 블록
역 변환
처리 유닛
60
잔차
블록
50
62
+
-
비디오 블록
예측 처리 유닛
41
MEU 42
MCU 44
인트라 예측
처리 유닛
46
IBC 유닛 48
재구성된
비디오 블록
파티셔닝 유닛
35
DPB
66
필터 유닛
64
비디오 인코더
20
비디오
데이터
메모리
33
비디오
데이터

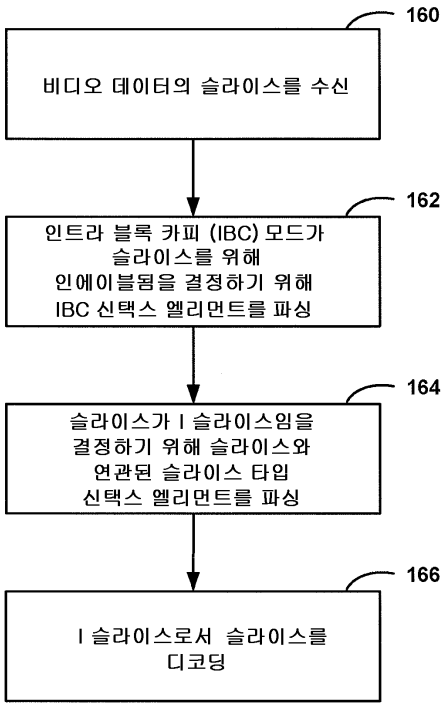
도면4


디코딩된 비디오
DPB 94
필터 유닛 92
예측 처리 유닛 81
MCU 82
인트라 예측 처리 유닛 84
IBC 유닛 85
90
잔차 블록
역 변환 처리 유닛 88
신택스 엘리먼트
엔트로피 디코딩 유닛 80
양자화 계수
역 양자화 유닛 86
비디오 데이터 메모리 78
비디오 디코더 30
인코딩된 비디오 비트스트림

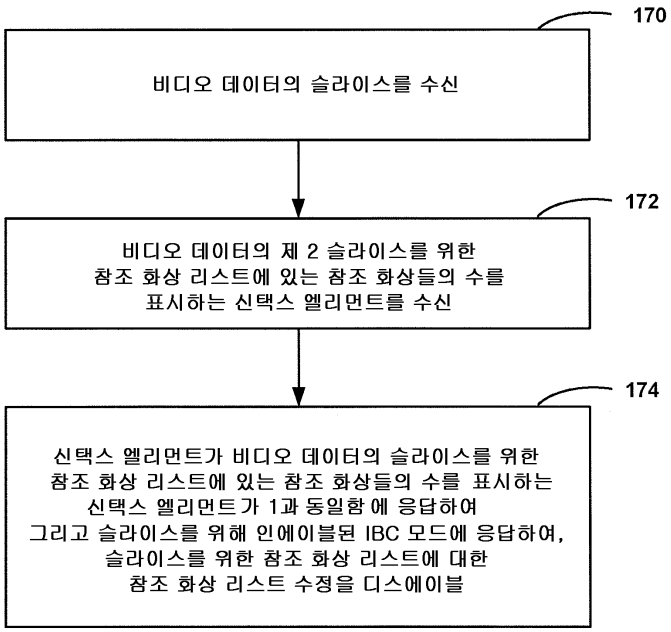
도면5


150
I 슬라이스로서 비디오 데이터의 슬라이스를 인코딩
152
인트라 블록 카피 (IBC) 모드가 슬라이스를 위해 인에이블됨을 표시하기 위해 IBC 신택스 엘리먼트를 생성

도면6


160
비디오 데이터의 슬라이스를 수신
162
인트라 블록 카피 (IBC) 모드가
슬라이스를 위해
인에이블됨을 결정하기 위해
IBC 신택스 엘리먼트를 파싱
164
슬라이스가 I 슬라이스임을
결정하기 위해 슬라이스와
연관된 슬라이스 타입
신택스 엘리먼트를 파싱
166
I 슬라이스로서 슬라이스를
디코딩

도면7


170
비디오 데이터의 슬라이스를 수신
172
비디오 데이터의 제 2 슬라이스를 위한
참조 화상 리스트에 있는 참조 화상들의 수를
표시하는 신택스 엘리먼트를 수신
174
신택스 엘리먼트가 비디오 데이터의 슬라이스를 위한
참조 화상 리스트에 있는 참조 화상들의 수를 표시하는
신택스 엘리먼트가 1과 동일함에 응답하여
그리고 슬라이스를 위해 인에이블된 IBC 모드에 응답하여,
슬라이스를 위한 참조 화상 리스트에 대한
참조 화상 리스트 수정을 디스에이블