(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년04월05일
(11) 등록번호 10-2383619
(24) 등록일자 2022년04월01일

(51) 국제특허분류(Int. Cl.)
H04N 19/105 (2014.01) *H04N 19/11* (2014.01)
H04N 19/139 (2014.01) *H04N 19/159* (2014.01)
H04N 19/174 (2014.01) *H04N 19/176* (2014.01)
H04N 19/503 (2014.01) *H04N 19/52* (2014.01)
H04N 19/573 (2014.01) *H04N 19/58* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/105 (2015.01)
H04N 19/11 (2015.01)
(21) 출원번호 10-2016-7028879
(22) 출원일자(국제) 2015년03월20일
심사청구일자 2020년03월04일
(85) 번역문제출일자 2016년10월17일
(65) 공개번호 10-2016-0135306
(43) 공개일자 2016년11월25일
(86) 국제출원번호 PCT/US2015/021866
(87) 국제공개번호 WO 2015/143395
국제공개일자 2015년09월24일
(30) 우선권주장
61/969,022 2014년03월21일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
Siu-Leong Yu et al., "New Intra Prediction using Intra-Macroblock Motion Compensation", JVT-C151, 2002-04-30.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
리 샹
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
팡 차오
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 34 항 심사관 : 김영태

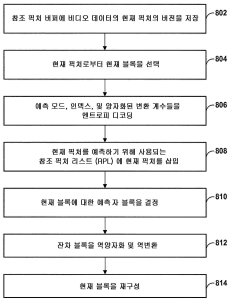
(54) 발명의 명칭 **비디오 코딩을 위한 참조로서 현재 픽처의 이용**

(57) 요 약

비디오 데이터를 인코딩 또는 디코딩하는 일 예시적인 방법은, 비디오 코더에 의해 그리고 참조 픽처 버퍼에서, 현재 픽처를 예측하기 위해 사용된 참조 픽처 리스트 (RPL) 에서의 현재 픽처를 포함하는, 비디오 데이터의 현재 픽처의 버전을 저장하는 단계, 및, 비디오 코더에 의해 그리고 RPL 에 기초하여, 참조 픽처 버퍼에 저장된 현재

(뒷면에 계속)

대 표 도 - 도8



픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 블록을 코딩하는 단계를 포함한다.

(52) CPC특허분류
H04N 19/139 (2015.01)
H04N 19/159 (2015.01)
H04N 19/174 (2015.01)
H04N 19/176 (2015.01)
H04N 19/503 (2015.01)
H04N 19/52 (2015.01)
H04N 19/573 (2015.01)
H04N 19/58 (2015.01)
H04N 19/70 (2015.01)

(72) 발명자
천 잉
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
왕 예-쿠이
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장
62/000,437 2014년05월19일 미국(US)
14/663,155 2015년03월19일 미국(US)

명 세 서

청구범위

청구항 1

비디오 데이터를 인코딩 또는 디코딩하는 방법으로서,

비디오 코더에 의해 그리고 참조 픽처 버퍼에서, 상기 비디오 데이터의 현재 픽처의 재구성된 블록들의 셋트를 저장하는 단계;

상기 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 내의 상기 현재 픽처에 인덱스 값을 할당하는 단계;

상기 현재 픽처의 현재 블록의 모션 정보는 상기 참조 픽처 리스트 내의 상기 현재 픽처의 상기 인덱스 값 및 상기 현재 픽처 내의 참조 블록의 위치를 특정한다고 결정하는 단계로서, 상기 참조 블록은 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 블록인, 상기 결정하는 단계;

상기 결정에 응답하여, 상기 참조 픽처 버퍼에 저장된 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 상기 참조 블록으로부터 예측자 블록을 형성하는 단계; 및

상기 비디오 코더에 의해 그리고 상기 RPL 에 기초하여, 잔차 픽셀 데이터와 비디오 데이터의 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 상기 참조 블록으로부터 형성된 상기 예측자 블록의 픽셀 값들의 합에 기초하여 상기 현재 픽처에서의 비디오 데이터의 상기 현재 블록의 픽셀 값들을 재구성하는 단계를 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 비디오 코더에 의해 그리고 상기 현재 블록의 코딩 후에, 코딩된 상기 현재 블록으로 상기 참조 픽처 버퍼에서의 상기 현재 픽처의 재구성된 블록들의 상기 셋트를 업데이트하는 단계; 및

상기 비디오 코더에 의해 그리고 상기 RPL 에 기초하여, 제 2 블록에 대한 잔차 픽셀 데이터와 상기 참조 픽처 버퍼에 저장된 상기 현재 픽처의 재구성된 블록들의 업데이트된 상기 셋트 중으로부터 형성된 제 2 예측자 블록의 픽셀 값들의 합에 기초하여 상기 현재 픽처에서의 비디오 데이터의 상기 제 2 블록의 픽셀 값들을 재구성하는 단계를 더 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

비디오 코더에 의해, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 신택스 엘리먼트를 코딩하는 단계; 및

상기 신택스 엘리먼트에 기초하여 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 에서 상기 비디오 데이터의 상기 현재 픽처에 상기 인덱스 값을 할당하도록 결정하는 단계를 더 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 4

제 3 항에 있어서,

상기 신택스 엘리먼트를 코딩하는 단계는, 상기 현재 픽처에 의해 참조되는 비디오 파라미터 셋트 (VPS), 상기 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS), 또는 상기 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (PPS) 에서 상기 신택스 엘리먼트를 코딩하는 단계를 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 5

제 4 항에 있어서,

상기 방법은, 상기 현재 블록이 인트라 블록 카피 (인트라 BC) 를 이용하여 코딩되는지 여부를 나타내는 신택스 엘리먼트를 코딩하는 것을 포함하지 않는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 현재 슬라이스에 대한 병치된 참조 인덱스를 나타내는 신택스 엘리먼트는 상기 현재 픽처가 아닌 픽처를 나타내는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 예측자 블록은 상기 현재 슬라이스에 포함되는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 RPL 에서의 각각의 엔트리는 각각의 인덱스 값을 가지며,

상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 내의 상기 현재 픽처에 상기 인덱스 값을 할당하는 단계는 적어도,

장기 참조 픽처 셋트 (RPS) 에서의 픽처들의 인덱스 값들보다 적은 인덱스 값을 상기 현재 픽처에 할당하는 것;

장기 참조 픽처 셋트 (RPS) 에서의 픽처들의 인덱스 값들보다 큰 인덱스 값을 상기 현재 픽처에 할당하는 것; 또는

고정된 인덱스 값을 상기 현재 픽처에 할당하는 것

에 의해, 하나 이상의 참조 픽처 셋트 (RPS) 들에 기초하여 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 을 구성하는 단계를 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 현재 슬라이스는 인트라 슬라이스이며,

상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 내의 상기 현재 픽처에 상기 인덱스 값을 할당하는 단계는, 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 이 상기 현재 픽처를 오직 포함하도록 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 을 구성하는 단계를 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 10

제 1 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 현재 슬라이스는 인터 슬라이스이며,

상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 내의 상기 현재 픽처에 상기 인덱스 값을 할당하는 단계는, 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 이 상기 현재 픽처 및 비디오 데이터의 하나 이상의 다른 픽처들을 포함하도록 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 을 구성하는 단계를 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 11

제 1 항에 있어서,

상기 비디오 코더에 의해 그리고 상기 현재 픽처의 상기 현재 블록을 코딩하기 전에, 상기 비디오 데이터의 상기 현재 픽처를 장기 참조 픽처로서 마킹하는 단계; 및

상기 비디오 코더에 의해 그리고 상기 현재 픽처의 상기 현재 블록을 코딩한 후에, 상기 비디오 데이터의 상기 현재 픽처를 단기 참조 픽처로서 마킹하는 단계를 더 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 12

제 1 항에 있어서,

비디오 데이터의 상기 현재 블록과 비디오 데이터의 상기 예측자 블록 사이의 변위를 나타내는 벡터의 표현을 인코딩된 비디오 비트스트림에서 인코딩하는 단계를 더 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 13

제 1 항에 있어서,

비디오 데이터의 상기 현재 블록과 비디오 데이터의 상기 예측자 블록 사이의 변위를 나타내는 벡터를 인코딩된 비디오 비트스트림에 기초하여 결정하는 단계를 더 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 14

비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더로서,

상기 비디오 데이터의 하나 이상의 픽처들을 저장하도록 구성된 참조 픽처 버퍼; 및

상기 참조 픽처 버퍼에서, 상기 비디오 데이터의 현재 픽처의 재구성된 블록들의 셋트를 저장하고;

상기 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 내의 상기 현재 픽처에 인덱스 값을 할당하고;

상기 현재 픽처의 현재 블록의 모션 정보는 상기 참조 픽처 리스트 내의 상기 현재 픽처의 상기 인덱스 값 및 상기 현재 픽처 내의 참조 블록의 위치를 특정한다고 결정하며, 상기 참조 블록은 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 블록이고;

상기 결정에 응답하여, 상기 참조 픽처 버퍼에 저장된 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 상기 참조 블록으로부터 예측자 블록을 형성하고; 및

상기 비디오 코더에 의해 그리고 상기 RPL 에 기초하여, 잔차 픽셀 데이터와 비디오 데이터의 상기 예측자 블록의 픽셀 값들의 합에 기초하여 상기 현재 픽처에서의 비디오 데이터의 상기 현재 블록의 픽셀 값들을 재구성하도록

구성된 하나 이상의 프로세서들을 포함하는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 15

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 현재 블록의 코딩 후에, 코딩된 상기 현재 블록으로 상기 참조 픽처 버퍼에서의 상기 현재 픽처의 재구성된 블록들의 상기 셋트를 업데이트하고; 그리고

상기 RPL 에 기초하여, 제 2 블록에 대한 잔차 픽셀 데이터와 상기 참조 픽처 버퍼에 저장된 상기 현재 픽처의 재구성된 블록들의 업데이트된 상기 셋트 중으로부터 형성된 제 2 예측자 블록의 픽셀 값들의 합에 기초하여 상기 현재 픽처에서의 비디오 데이터의 상기 제 2 블록의 픽셀 값들을 재구성하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 16

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 신택스 엘리먼트를 코딩하고; 및

상기 신택스 엘리먼트에 기초하여 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 에서 상기 비디오 데이터의 상기 현재 픽처에 상기 인덱스 값을 할당하도록 결정하도록

구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 17

제 16 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 현재 픽처에 의해 참조되는 비디오 파라미터 셋트 (VPS), 상기 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS), 또는 상기 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (PPS) 에서 상기 신택스 엘리먼트를 코딩하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 18

제 17 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 현재 블록이 인트라 블록 카피 (인트라 BC) 를 이용하여 코딩되는지 여부를 나타내는 신택스 엘리먼트를 코딩하도록 구성되지 않는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 19

제 14 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 현재 슬라이스에 대한 병치된 참조 인덱스를 나타내는 신택스 엘리먼트는 상기 현재 픽처가 아닌 픽처를 나타내는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 20

제 14 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 예측자 블록은 상기 현재 슬라이스에 포함되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 21

제 14 항에 있어서,

상기 RPL 에서의 각각의 엔트리는 각각의 인덱스 값을 가지며,

상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 에서 상기 현재 픽처를 삽입하기 위해, 상기 하나 이상의 프로세서들은, 적어도,

장기 참조 픽처 셋트 (RPS) 에서의 픽처들의 인덱스 값들보다 적은 인덱스 값을 상기 현재 픽처에 할당하는 것;

장기 참조 픽처 셋트 (RPS) 에서의 픽처들의 인덱스 값들보다 큰 인덱스 값을 상기 현재 픽처에 할당하는 것; 또는

고정된 인덱스 값을 상기 현재 픽처에 할당하는 것

에 의해, 하나 이상의 참조 픽처 셋트 (RPS) 들에 기초하여 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 을 구성하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 22

제 14 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 현재 슬라이스는 인트라 슬라이스이며,

상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 내의 상기 현재 픽처에 상기 인덱스 값을 할당하기 위해, 상기 하나 이상의 프로세서들은, 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 이 상기 현재 픽처를 오직 포함하도록 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 을 구성하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 23

제 14 항에 있어서,

상기 현재 블록은 상기 현재 픽처의 현재 슬라이스에 포함되고, 상기 현재 슬라이스는 인터 슬라이스이며,

상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 내의 상기 현재 픽처에 상기 인덱스 값을 할당하기 위해, 상기 하나 이상의 프로세서들은, 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 이 상기 현재 픽처 및 비디오 데이터의 하나 이상의 다른 픽처들을 포함하도록 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 을 구성하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 24

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 현재 픽처의 상기 현재 블록을 코딩하기 전에, 상기 비디오 데이터의 상기 현재 픽처를 장기 참조 픽처로서 마킹하고; 그리고

상기 현재 픽처의 상기 현재 블록을 코딩한 후에, 상기 비디오 데이터의 상기 현재 픽처를 단기 참조 픽처로서 마킹하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 25

제 14 항에 있어서,

상기 비디오 코더는 비디오 인코더이고,

상기 하나 이상의 프로세서들은 또한, 비디오 데이터의 상기 현재 블록과 비디오 데이터의 상기 예측자 블록 사이의 변위를 나타내는 벡터의 표현을 인코딩된 비디오 비트스트림에서 인코딩하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 26

제 14 항에 있어서,

상기 비디오 코더는 비디오 디코더이고,

상기 하나 이상의 프로세서들은 또한, 비디오 데이터의 상기 현재 블록과 비디오 데이터의 상기 예측자 블록 사이의 변위를 나타내는 벡터를 인코딩된 비디오 비트스트림에 기초하여 결정하도록 구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 27

비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스로서,

참조 픽처 버퍼에서, 상기 비디오 데이터의 현재 픽처의 재구성된 블록들의 셋트를 저장하는 수단;

상기 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 내의 상기 현재 픽처에 인덱스 값을 할당하는 수단;

상기 현재 픽처의 현재 블록의 모션 정보는 상기 참조 픽처 리스트 내의 상기 현재 픽처의 상기 인덱스 값 및 상기 현재 픽처 내의 참조 블록의 위치를 특정한다고 결정하는 수단으로서, 상기 참조 블록은 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 블록인, 상기 결정하는 수단;

상기 결정에 응답하여, 상기 참조 픽처 버퍼에 저장된 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 상기 참조 블록으로부터 예측자 블록을 형성하는 수단; 및

상기 RPL 에 기초하여, 잔차 픽셀 데이터와 비디오 데이터의 상기 예측자 블록의 픽셀 값들의 합에 기초하여 상기 현재 픽처에서의 비디오 데이터의 현재 블록의 픽셀 값들을 재구성하는 수단을 포함하는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스.

청구항 28

제 27 항에 있어서,

상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 신택스 엘리먼트를 코딩하는 수단; 및

상기 신택스 엘리먼트에 기초하여 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 에서 상기 비디오 데이터의 상기 현재 픽처에 상기 인덱스 값을 할당할지 여부를 결정하는 수단을 더 포함하는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스.

청구항 29

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 비디오 코더의 하나 이상의 프로세서들로 하여금,

참조 픽처 버퍼에서, 상기 비디오 데이터의 현재 픽처의 재구성된 블록들의 셋트를 저장하게 하고;

상기 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 내의상기 현재 픽처에 인덱스 값을 할당하게 하며;

상기 현재 픽처의 현재 블록의 모션 정보는 상기 참조 픽처 리스트 내의 상기 현재 픽처의 상기 인덱스 값 및 상기 현재 픽처 내의 참조 블록의 위치를 특정한다고 결정하게 하고, 상기 참조 블록은 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 블록이며;

상기 결정에 응답하여, 상기 참조 픽처 버퍼에 저장된 상기 현재 픽처의 재구성된 블록들의 상기 셋트 중의 상기 참조 블록으로부터 예측자 블록을 형성하게 하고;

상기 RPL 에 기초하여, 잔차 픽셀 데이터와 비디오 데이터의 상기 예측자 블록의 픽셀 값들의 합에 기초하여 상기 현재 픽처에서의 비디오 데이터의 현재 블록의 픽셀 값들을 재구성하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 30

제 29 항에 있어서,

실행될 때, 비디오 코더의 하나 이상의 프로세서들로 하여금,

상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 신택스 엘리먼트를 코딩하게 하고; 및

상기 신택스 엘리먼트에 기초하여 상기 현재 픽처를 예측하기 위해 사용된 상기 RPL 에서 상기 비디오 데이터의 상기 현재 픽처에 상기 인덱스 값을 할당할지 여부를 결정하게 하는

명령들을 더 저장하는, 컴퓨터 판독가능 저장 매체.

청구항 31

제 3 항에 있어서,

상기 신택스 엘리먼트를 코딩하는 단계는,

상기 비디오 코더에 의해 그리고 상기 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 1 신택스 엘리먼트를 코딩하는 단계; 및

상기 비디오 코더에 의해 그리고 상기 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 2 신택스 엘리먼트를 코딩하는 단계

를 포함하는, 비디오 데이터를 인코딩 또는 디코딩하는 방법.

청구항 32

제 16 항에 있어서,

상기 신택스 엘리먼트를 코딩하기 위해, 상기 하나 이상의 프로세서들은,

상기 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 1 신택스 엘리먼트를 코딩하고; 및

상기 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 2 신택스 엘리먼트를 코딩하도록

구성되는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 비디오 코더.

청구항 33

제 28 항에 있어서,

상기 신택스 엘리먼트를 코딩하는 수단은,

상기 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 1 신택스 엘리먼트를 코딩하는 수단; 및

상기 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 2 신택스 엘리먼트를 코딩하는 수단

을 포함하는, 비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스.

청구항 34

제 30 항에 있어서,

상기 하나 이상의 프로세서들로 하여금 상기 신택스 엘리먼트를 코딩하게 하는 상기 명령들은, 상기 하나 이상의 프로세서들로 하여금,

상기 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 1 신택스 엘리먼트를 코딩하게 하고; 및

상기 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (SPS) 에서, 상기 비디오 데이터의 픽처들이 상기 픽처들 자신들을 예측하기 위해 사용된 RPL 들에서 인덱스 값들이 할당되는 것이 허용된다는 것을 나타내는 제 2 신택스 엘리먼트를 코딩하게 하는

명령들을 포함하는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기 술 분 야

[0001] 본 출원은 2014년 3월 21일자로 출원된 미국 가출원 제 61/969,022 호, 및 2014년 5월 19일자로 출원된 미국 가출원 제 62/000,437 호의 이익을 주장하며, 그들의 각각의 전체 내용들이 본원에 참조에 의해 그 전체가 통합된다.

[0002] 본 개시는 비디오 코딩에 관한 것이다.

배 경 기 술

[0003] 디지털 비디오 능력들은, 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDAs), 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 "스마트 폰들", 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등 포함한, 광범위한 디바이스들에 포함될 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), 현재 개발중인 HEVC (High Efficiency Video Coding) 표준, 및 이런 표준들의 확장판들에 의해 정의된 표준들에서 기술되는 것들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현함으로써 디지털 비디오 정보를 보다 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 압축 기법들은 비디오 시퀀스들에 고유한 리던던시 (redundancy) 를 감소시키거나 제거하기 위해 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 수행한다. 블록-기반 비디오 코딩에 있어서, 비디오 슬라이스 (즉, 비디오 픽처 또는 비디오 픽처의 부분) 는, 트리블록들, 코딩 유닛들 (CUs) 및또는 코딩 노드들로서도 지칭될 수도 있는, 비디오 블록들로 파티셔닝될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 블록들은 동일한 픽처에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다.

[0005] 공간 또는 시간 예측은 코딩될 블록에 대한 예측적 블록을 초래한다. 잔차 (residual) 데이터는 코딩될 오리지날 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 있어, 잔차 변환 계수들을 야기하고, 이들은 그 후 양자화될 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0006] 일반적으로, 이 개시물은 비디오 코딩을 위해 인트라-예측을 수행하는 기술들을 설명한다. 보다 상세하게는, 이 개시물은 현재 픽처의 하나 이상의 블록들을 코딩할 때 참조 픽처 (reference picture) 로서 현재 픽처 (current picture) 를 이용하는 예시적인 기술들을 설명한다. 예를 들면, 현재 픽처는 인트라 블록 카핑 (인트라 BC) 을 이용하여 현재 픽처의 하나 이상의 블록들을 코딩할 때 참조 픽처로서 사용될 수도 있다.

[0007] 하나의 예에서, 비디오 데이터를 인코딩 또는 디코딩하는 방법은, 비디오 코더에 의해 그리고 참조 픽처 버퍼에서, 비디오 데이터의 현재 픽처의 버전 (version) 을 저장하는 단계; 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (reference picture list; RPL) 에서 현재 픽처의 표시 (indication) 를 삽입하는 단계; 및 비디오 코더에 의해 그리고 RPL 에 기초하여, 참조 픽처 버퍼에 저장된 현재 픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 제 1 블록을 코딩하는 단계를 포함한다. 일부 예들에서, 예측자 블록은 대안적으로 예측 블록으로서 지칭될 수도 있다.

[0008] 다른 예에서, 비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스는, 비디오 데이터의 하나 이상의 픽처들을 저장하도록 구성된 참조 픽처 버퍼, 및 하나 이상의 프로세서들을 포함한다. 이 예에서, 하나 이상의 프로세서들은, 참조 픽처 버퍼에서, 비디오 데이터의 현재 픽처의 버전을 저장하고; 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 에서 현재 픽처의 표시를 삽입하며; 그리고, 비디오 코더에 의해 그리고 RPL 에 기초하여, 참조 픽처 버퍼에 저장된 현재 픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 제 1 블록을 코딩하도록 구성된다.

[0009] 또 다른 예에서, 비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스는, 참조 픽처 버퍼에서, 비디오 데이터의 현재 픽처의 버전을 저장하는 수단; 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 에서 현재 픽처의 표시를 삽입하는 수단; 및, RPL 에 기초하여, 참조 픽처 버퍼에 저장된 현재 픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 제 1 블록을 코딩하는 수단을 포함한다.

[0010] 또 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행될 때 비디오 코더의 하나 이상의 프로세서들로 하여금, 참조 픽처 버퍼에서, 비디오 데이터의 현재 픽처의 버전을 저장하게 하고; 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 에서 현재 픽처의 표시를 삽입하게 하며; 그리고, RPL 에 기초하여, 참조 픽처 버퍼에 저장된 현재 픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 제 1 블록을 코딩하게 하는 명령들을 저장한다.

[0011] 본 개시의 하나 이상의 양태들의 상세들이 첨부하는 도면들 및 이하의 설명에서 설명된다. 이 개시물에서 설명된 기술들의 다른 특징들, 목적들 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구범위로부터 명백해 질 것이다.

도면의 간단한 설명

[0012] 도 1 은 본 개시의 기술들을 구현할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 도시하는 블록도이다.

도 2 는 본 개시의 하나 이상의 기술들에 따라, 픽처들의 예시적인 비디오 시퀀스를 나타내는 개념도이다.

도 3 은 본 개시물에서 설명된 인트라 블록 카피를 위한 기술들을 이용할 수도 있는 비디오 인코더의 일 예를 나타내는 블록도이다.

도 4 는 본 개시의 하나 이상의 기술들에 따라 사용될 수도 있는 각도 인트라-예측 모드들의 일 예를 나타낸다.

도 5 는 본 개시물에서 설명된 기술들을 구현할 수도 있는 비디오 디코더 (30) 의 일 예를 나타내는 블록도이다.

도 6 은 본 개시의 하나 이상의 기술들에 따른, 인트라 블록 카핑 프로세스의 일 예를 나타내는 도이다.

도 7 은 본 개시의 하나 이상의 기술들에 따른, 동일한 픽처에 포함된 예측자 블록에 기초하여 픽처의 비디오 데이터의 블록을 인코딩하기 위한 비디오 인코더의 예시적인 동작들을 나타내는 흐름도이다.

도 8 은 본 개시의 하나 이상의 기술들에 따른, 동일한 픽처에 포함된 예측자 블록에 기초하여 픽처의 비디오 데이터의 블록을 디코딩하기 위한 비디오 디코더의 예시적인 동작들을 나타내는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0013] 비디오 시퀀스는 일반적으로 픽처들의 시퀀스로서 표현된다. 통상적으로, 블록-기반 코딩 기법들이 개별 픽처들의 각각을 코딩하기 위해 사용된다. 즉, 각 픽처는 블록들로 분할되고, 블록들의 각각은 개별적으로 코딩된다. 비디오 데이터의 블록을 코딩하는 것은 블록에서의 픽셀들에 대한 예측된 값들을 형성하는 것 및 잔차 값들을 코딩하는 것을 일반적으로 수반한다. 예측 값들은 하나 이상의 예측 블록들에서의 픽셀 샘플들을 이용하여 형성된다. 잔차 값들은 오리지날 블록의 픽셀들과 예측된 픽셀 값들 사이의 차이들을 나타낸다. 구체적으로, 비디오 데이터의 오리지날 블록은 픽셀 값들의 어레이를 포함하고, 예측된 블록은 예측된 픽셀 값들의 어레이를 포함한다. 잔차 값들은 오리지날 블록의 픽셀 값들과 예측된 픽셀 값들 사이의 픽셀-바이-픽셀 차이들을 나타낸다.

[0014] 비디오 데이터의 블록에 대한 예측 기법들은 일반적으로 인트라-예측 (intra-prediction) 및 인터-예측 (inter-prediction) 으로서 분류된다. 인트라-예측, 또는 공간 예측은 임의의 참조 픽처로부터의 예측을 포함하지

않고, 대신에, 블록은 이웃하는 (neighboring), 이전에 코딩된 블록들의 픽셀 값들로부터 예측된다. 인터-예측, 또는 시간 예측은, 하나 이상의 참조 픽처 리스트들 (RPLs) 로부터 선택된 하나 이상의 이전에 코딩된 참조 픽처들 (예컨대, 프레임들 또는 슬라이스들) 의 픽셀 값들로부터 블록을 예측하는 것을 일반적으로 수반한다. 비디오 코더는 RPL 들에 포함된 픽처들을 저장하도록 구성된 하나 이상의 참조 픽처 버퍼들을 포함할 수도 있다.

[0015] 원격 데스크탑, 원격 게이밍, 무선 디스플레이들, 차량용 인포테인먼트, 클라우드 컴퓨팅 등과 같은 많은 애플리케이션들은 일상 생활들에서 통상적이 되고 있다. 이들 애플리케이션들에서의 비디오 콘텐츠는 통상적으로 자연적인 내용, 텍스트, 인공적인 그래픽 등의 결합들이다. 텍스트 및 인공적인 그래픽 영역에서, (문자들, 아이콘들, 심볼들 등과같은) 반복된 패턴들이 종종 존재한다. 인트라 블록 카핑 (Intra BC) 은, 비디오 코더로 하여금 이러한 리던던시를 제거하고 인트라-픽처 코딩 효율을 향상시키는 것을 가능하게 할 수도 있는 기법이다. 일부 경우들에서, 인트라 BC 는 다르게는 인트라 모션 보상 (MC) 으로서 지칭될 수도 있다.

[0016] 몇몇 인트라 BC 기법들에 따르면, 비디오 코더들은, 현재 블록의 예측을 위해 동일한 픽처에서 비디오 데이터의 (코딩될) 현재 블록 바로 위 또는 수평으로 바로 일직선에 있는, 비디오 데이터의 현재 블록과 동일한 픽처 내의, 이전에 코딩된 비디오 데이터의 블록들을 이용할 수도 있다. 달리 말하면, 비디오 데이터의 픽처가 2-D 격자 상에 부과되는 경우에, 비디오 데이터의 각각의 블록은 x-값들 및 y-값들의 고유한 범위를 점유할 것이다. 따라서, 일부 비디오 코더들은, 동일한 x-값들의 셋트 (즉, 현재 블록과 수직으로 일치) 또는 동일한 y-값들의 셋트 (즉, 현재 블록과 수평으로 일치) 만을 공유하는 이전에 코딩된 비디오 데이터의 블록들에 기초하여 비디오 데이터의 현재 블록을 예측할 수도 있다.

[0017] Pang 등의 "Non-RCE3: Intra Motion Compensation with 2-D MVs", 문서: JCTVC-N0256, JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14차 회의: 오스트리아 비엔나, 2013년 7월 25일 - 8월 2일 (이하, "JCTVC-N0256") 에서 다른 인트라 BC 기법들이 설명된다. 비엔나에서의 JCT-VC 회의 (2013년 7월) 에서, 인트라 BC 는 고 효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 범위 확장 표준에서 채택되었다. JCTVC-N0256 에 따르면, 비디오 코더는 현재 블록과 동일한 픽처 내의 예측 블록을 식별하는 2-차원 모션 벡터를 결정할 수도 있다. 일부 예들에서, 모션 벡터는 블록 벡터, 오프셋 벡터, 또는 변위 벡터로서 또한 지칭될 수도 있다. 어느 경우에도, 2-차원 모션 벡터는 수평 변위 성분 및 수직 변위 성분을 가지며, 이들의 각각은 제로 또는 비-제로일 수도 있다. 수평 변위 성분은 비디오 데이터의 예측적 블록, 또는 예측 블록과 비디오 데이터의 현재 블록 사이의 수평 변위를 나타내고, 수직 변위 성분은 비디오 데이터의 예측 블록과 비디오 데이터의 현재 블록 사이의 수직 변위를 나타낸다. 인트라 BC 에 대해, 예측 블록의 픽셀들은 코딩되고 있는 블록 (즉, 현재 블록) 에서의 대응하는 픽셀들에 대한 예측 샘플들로서 사용될 수도 있다. 비디오 코더는 추가적으로, 비디오 데이터의 현재 블록과 예측 블록에 기초하여 비디오 데이터의 잔차 블록을 결정하고, 2-차원 모션 벡터 및 비디오 데이터의 잔차 블록을 코딩할 수도 있다.

[0018] 일부 예들에서, 인트라 BC 는 특히 스크린 콘텐츠 코딩을 위해 효율적인 코딩 툴 (tool) 일 수도 있다. 실례로, 일부 예들에서, 인트라 BC 를 이용하여 블록들을 코딩하는 것은 인터 또는 인트라 코딩을 이용하여 블록들을 코딩하는 것보다 더 작은 비트스트림을 초래할 수도 있다. 상기 논의된 바와 같이, 인트라 BC 는 인터-유사 코딩 툴이다 (픽처에 대한 픽셀 값들이 픽처에서의 다른 픽셀 값들로부터 예측됨을 의미한다). 하지만, 일부 예들에서, 실제 설계에서 선호되지 않을 수도 있는, 인트라 BC 에 적용되는 하나 이상이 제약들로 인해, 인트라 BC 를 종래의 인트라 픽처들 내로 통합하는 것은 어려울 수도 있다. 일부 예들에서, 제약들은, 비제한적으로, 예측자 블록이 코딩될 현재 블록과 동일한 슬라이스 (slice) 또는 타일 (tile) 내에 있는 것, 예측자 블록이 코딩될 현재 블록과 중첩하지 않는 것, 예측자 블록에서의 모든 픽셀들이 재구성되는 것, 예측자 블록이 (예를 들어, Rapaka 등의 "On parallel processing capability of intra block copy," 문서: JCTVC-S0220, JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 19차 회의: 프랑스 스트라스부르그, 2014년 10월 17-24 에서 기술된 바와 같이 병렬화 구현에 관한 고려로 인해) 어떤 구역 내에 있는 것, 및, 제약된 인트라 예측이 가능하게 될 때, 예측자 블록은 종래의 인터 모드를 이용하여 코딩되는 임의의 픽셀을 포함하지 않는 것을 포함한다. 부가적으로, 일부 예들에서, 종래의 인트라 및 인터 프레임들에 대한 하드웨어 아키텍처는 (예를 들어, 픽처 내부의 블록 카피를 초래하는 인트라 BC 로 인해) 수정 없이 인트라 BC 에 대해 재사용되지 않을 수도 있다. 이와 같이, 비디오 코더로 하여금, 현재 인트라 BC 에 적용되는 제약들의 일부 또는 전부를 유지하면서, 그리고 하드웨어 아키텍처에 대한 (중요한) 수정 없이, 인트라 BC 에 의해 제공되는 효율성들을 얻는 것이 가능하게 하는 것이 바람직할 수도 있다.

[0019] 이 개시물의 하나 이상의 기술들에 따라서, 종래의 인트라 예측 기법들을 이용하여 현재 픽처에서의 샘플들에

기초하여 현재 픽처의 블록을 예측하는 것과는 대조적으로, 비디오 코더는 종래의 인터 예측과 유사한 기법들을 이용하여 현재 픽처에서의 샘플들에 기초하여 현재 픽처에서 블록을 예측하도록 인트라 BC 를 수행할 수도 있다. 실례로, 비디오 코더는 현재 픽처를 예측하기 위해 사용되는 참조 픽처 리스트 (RPL) 에서 현재 픽처를 포함하고, 현재 픽처의 버전을 참조 픽처 버퍼에 저장하며, 참조 픽처 버퍼에 저장된 현재 픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 블록을 코딩할 수도 있다. 이러한 방식으로, 비디오 코더는, 인트라 BC 에 현재 적용되는 제약들의 일부 또는 전부를 유지하면서, 인트라 BC 에 의해 제공되는 효율성들을 얻을 수도 있다. 또한 이러한 방식으로, 비디오 코더는, 심각한 수정 없이 인트라 BC 에 대해 종래의 인트라 및 인터 프레임들에 대한 하드웨어 아키텍처를 재사용할 수도 있다.

[0020] 이 개시물은 현재 픽처의 부분들 (portions) 을 예측할 때 참조 픽처로서 현재 픽처를 이용하는 것에 관련된 예시적인 기술들을 설명한다. 이해를 돕기 위해, 가능하게는 (8 비트보다 높은) 하이 비트 심도, 4:4:4 및 4:2:2 를 포함하는 하이 크로마 샘플링 포맷의 지원을 포함하는, 고 효율 비디오 코딩 (HEVC) 비디오 코딩 표준에 대한 범위 확장들 (REXt) 에 관해 예시적인 기술들이 설명된다. 그 기술들은 또한 스크린 콘텐츠 코딩을 위해 적용가능할 수도 있다. 기술들은 범위 확장들 또는 스크린 콘텐츠 코딩에 제한되지 않고, 표준들 기반 또는 비-표준들 기반 비디오 코딩을 포함하는 비디오 코딩 기법들에 일반적으로 적용가능할 수도 있음을 이해하여야 한다. 또한, 이 개시물에서 설명된 기술들은 미래에 개발되는 표준들의 부분이 될 수도 있다. 달리 말하면, 이 개시물에서 설명된 기술들은 이전에 개발된 비디오 코딩 표준들, 현재 개발 중인 비디오 코딩 표준들, 그리고 다가오는 비디오 코딩 표준들에 적용가능할 수도 있다.

[0021] 도 1 은 이 개시물의 기술들을 구현할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템 (10) 을 나타내는 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 이후의 시간에 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 컴퓨터 판독가능 매체 (16) 를 통해 목적지 디바이스 (14) 로 비디오 데이터를 제공한다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화기 핸드셋들, 이른바 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스들 등을 포함한 임의의 다양한 범위의 디바이스들을 포함할 수도 있다. 몇몇 경우들에서, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 무선 통신을 위해 장비될 수도 있다.

[0022] 목적지 디바이스 (14) 는 컴퓨터 판독가능 매체 (16) 를 통해 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 컴퓨터 판독가능 매체 (16) 는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 컴퓨터 판독가능 매체 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접 실시간으로 송신하는 것을 가능하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 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 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0023] 일부 예들에서, 인코딩된 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 로부터 저장 디바이스 (32) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 목적지 디바이스 (14) 의 입력 인터페이스에 의해 저장 디바이스 (32) 로부터 액세스될 수도 있다. 저장 디바이스 (32) 는 하드 드라이브, 블루 레이 디스크들, DVD 들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 임의의 다양한 분포되거나 국소적으로 액세스되는 데이터 저장 매체들을 포함할 수도 있다. 다른 예에서, 저장 디바이스 (32) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수도 있다.

[0024] 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 저장 디바이스 (32) 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. 예의 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, 네트워크 어태치드 스토리지 (network attached storage; NAS) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다. 목적지 디바이스 (14) 는 임의의 표준 데이터 접속, 이를테면 인터넷 접속

을 통해, 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽 모두의 조합들을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.

[0025] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 제한되지 않는다. 그 기법들은, 임의의 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 이를테면, DASH (dynamic adaptive streaming over HTTP) 의 지원 하의 비디오 코딩, 데이터 저장 매체상으로 인코딩되는 디지털 비디오, 데이터 저장 매체상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 몇몇 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 전화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0026] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (31) 를 포함한다. 본 개시에 따르면, 소스 디바이스 (12) 의 비디오 인코더 (20) 는 비디오 코딩에서 변환을 수행하는 기법들을 적용하도록 구성될 수도 있다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 외부 카메라와 같은 외부 비디오 소스 (18) 로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하기보다는 외부 디스플레이 디바이스와 인터페이싱할 수도 있다.

[0027] 도 1 의 도시된 시스템 (10) 은 단순히 하나의 예일 뿐이다. 비디오 코딩에서 향상된 인트라 블록 카피 시그널링을 위한 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 본 개시의 기법들은 일반적으로 비디오 인코딩 또는 디코딩 디바이스에 의해 수행되지만, 그 기법들은 또한 비디오 코덱에 의해 수행될 수도 있다. 게다가, 본 개시의 기법들은 또한 비디오 프리프로세서에 의해 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소스 디바이스 (12) 가 목적지 디바이스 (14) 로의 송신을 위한 코딩된 비디오 데이터를 생성하는 그러한 코딩 디바이스들의 단순한 예들일 뿐이다. 일부 예들에서, 디바이스들 (12, 14) 은 디바이스들 (12,14) 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 하는 실질적으로 대칭적인 방식으로 동작할 수도 있다. 이리하여, 시스템 (10) 은 예를 들어 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 화상 전화를 위해 비디오 디바이스들 (12, 14) 사이의 단방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0028] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 컨텐츠 제공자로부터 비디오를 수신하는 비디오 피드 인터페이스와 같은 비디오 캡처 디바이스를 포함할 수도 있다. 다른 대안으로서, 비디오 소스 (18) 는 소스 비디오로서 컴퓨터 그래픽-기반 데이터, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합을 생성할 수도 있다. 일부 경우들에서, 비디오 소스 (18) 가 비디오 카메라인 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 상술된 바와 같이, 그러나, 본 개시에 기술된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에, 캡처된, 프리-캡처된, 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 그 인코딩된 비디오 정보는 그 후 컴퓨터 판독가능 매체 (16) 상으로 출력 인터페이스 (22) 에 의해 출력될 수도 있다.

[0029] 컴퓨터 판독가능 매체 (16) 는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 일시적인 매체들, 또는 하드 디스크, 플래시 드라이브, 컴팩트 디스크, 디지털 비디오 디스크, 블루-레이 디스크, 또는 다른 컴퓨터 판독가능 매체들과 같은 저장 매체들 (즉, 비일시적인 저장 매체들) 을 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (도시하지 않음) 는 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신하고 예를 들어 네트워크 송신을 통해 목적지 디바이스 (14) 로 그 인코딩딘 비디오 데이터를 제공할 수도 있다. 유사하게, 디스크 스탬핑 설비와 같은 매체 생산 설비의 컴퓨팅 디바이스는 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신하고 그 인코딩된 비디오 데이터를 포함하는 디스크를 제조할 수도 있다. 따라서, 컴퓨터 판독가능 매체 (16) 는 여러 예들에서 여러 형태들의 하나 이상의 컴퓨터 판독가능 매체들을 포함하는 것으로 이해될 수도 있다.

[0030] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 판독가능 매체 (16) 또는 저장 디바이스 (32) 로부터

정보를 수신한다. 컴퓨터 판독가능 매체 (16) 또는 저장 디바이스 (32) 의 정보는 블록들 및 다른 코딩된 유닛들, 예컨대, GOP 들의 특성들 및/또는 프로세싱을 기술하는 신택스 엘리먼트들 (syntax elements) 을 포함하는, 비디오 디코더 (30) 에 의해 또한 사용되는, 비디오 인코더 (20) 에 의해 정의된 신택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (31) 는 사용자에게 디코딩된 비디오 데이터를 디스플레이하며, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 임의의 다양한 디스플레이 디바이스들을 포함할 수도 있다.

[0031] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 반도체들 (ASICs), 필드 프로그램가능 게이트 어레이들 (FPGAs), 이산 로직 회로, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들과 같은 임의의 다양한 적합한 인코더 또는 디코더 회로로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어로 구현되는 경우, 디바이스는 적합한 비일시적 컴퓨터 판독가능 매체에 소프트웨어를 위한 명령들을 저장하고, 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이들 중 어느 것은 결합된 비디오 인코더/디코더 (코덱) 의 부분으로서 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 을 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 예컨대 셀룰러 전화를 포함할 수도 있다.

[0032] 도 1 에 도시되지는 않지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통 데이터 스트림 또는 별개의 데이터 스트림들에서의 오디오 및 비디오 양자의 인코딩을 다루기 위해 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 와 같은 다른 프로토콜들에 따를 수도 있다.

[0033] 본 개시는 일반적으로 비디오 디코더 (30) 와 같은 다른 디바이스로 소정의 정보를 "시그널링하는" 비디오 인코더 (20) 를 참조할 수도 있다. 그러나, 비디오 인코더 (20) 는 비디오 데이터의 여러 인코딩된 부분들과 소정의 신택스 엘리먼트들을 연관시킴으로써 정보를 시그널링할 수도 있다. 즉, 비디오 인코더 (20) 는 비디오 데이터의 여러 인코딩된 부분들의 헤더들로 소정의 신택스 엘리먼트들을 저장함으로써 데이터를 "시그널링" 할 수도 있다. 일부 경우들에서, 그러한 신택스 엘리먼트들은 비디오 디코더 (30) 에의해 수신되고 디코딩되기 전에 인코딩되고 저장 (예를 들어, 저장 디바이스 (32) 에 저장) 될 수도 있다. 따라서, 용어 "시그널링 (signaling)" 은 일반적으로, 그러한 통신이 실시간으로 또는 비실시간으로, 또는 예컨데, 인코딩 시에 매체에 신택스 엘리먼트들을 저장하고, 그 후 이러한 매체에 저장된 후 임의의 시간에 디코딩 디바이스에 의해 취출될 수도 있는 때에 발생할 수도 있는 시간 구간을 경과하여 발생하는지에 관계없이, 압축된 비디오 데이터를 디코딩하기 위한 신택스 또는 다른 데이터의 통신을 지칭할 수도 있다.

[0034] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC 표준과 같은 비디오 압축 표준에 따라 동작할 수도 있다. 본 개시의 기술들은 임의의 특정의 코딩 표준에 제한되지 않는 한편, 그 기술들은 HEVC 표준에, 특히 REXt 확장판과 같은 HEVC 표준의 확장판들에 관련될 수도 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 으로서 지칭되는 비디오 코딩 디바이스의 모델에 기초한다. HM 은, 예컨대, ITU-T H.264/AVC에 따른 현존 디바이스들에 비해 비디오 코딩 디바이스들의 몇몇 부가적인 능력들을 상정한다. 예를 들어, H.264가 9 개의 인트라 예측 인코딩 모드들을 제공하는 반면, HM은 35 개나 되는 인트라 예측 인코딩 모드들을 제공할 수도 있다.

[0035] 일반적으로, HM 의 작업 모델은 비디오 픽처가 루마 및 크로마 샘플들 양자를 포함하는 트리블록들의 시퀀스 또는 최대 코딩 유닛들 (LCU) 로 분할될 수도 있다고 기술한다. 비트스트림 내의 신택스 데이터는 픽셀들의 수의 면에서 최대 코딩 유닛인 LCU 에 대한 사이즈를 정의할 수도 있다. 슬라이스는 다수의 연속적인 코딩 트리 유닛들 (CTUs) 을 포함한다. CTU 들 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2 개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하기 위해 사용되는 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처 또는 3 개의 별개의 칼라 평면들을 갖는 픽처에서, CTU 는 코딩 트리 블록의 샘플들을 코딩하기 위해 사용되는 단일의 코딩 트리 블록 및 신택스 구조들을 포함할 수도 있다.

[0036] 비디오 픽처는 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 쿼드트리에 따라 코딩 유닛들 (CUs) 로 분할될 수도 있다. 일반적으로, 쿼드트리 데이터 구조는 CU 당 하나의 노드를 포함하며, 루트 노드가 트리블록에 대응한다. CU가 네 개의 서브 CU들로 분할되면, 그 CU에 대응하는 노드는 네 개의

리프 (leaf) 노드들을 포함하며, 그 리프 노드들의 각각은 서브 CU들 중 하나에 대응한다. CU 는 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2 개의 대응하는 코딩 블록들, 및 그 코딩 블록들의 샘플들을 코딩하기 위해 사용되는 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처 또는 3 개의 별개의 칼라 평면들을 갖는 픽처에서, CU 는 단일의 코딩 블록 및 그 코딩 블록의 샘플들을 코딩하기 위해 사용되는 신택스 구조들을 포함할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다.

[0037] 쿼드트리 데이터 구조의 각각의 노드는 대응하는 CU 에 대해 신택스 데이터를 제공할 수도 있다. 예를 들어, 쿼드트리에서의 노드가, 그 노드에 대응하는 CU가 서브 CU들로 분할되는지의 여부를 표시하는 분할 플래그를 포함할 수도 있다. CU에 대한 신택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU가 서브 CU들로 분할되는지의 여부에 의존할 수도 있다. CU가 더 분할되지 않으면, 그것은 리프 CU라고 지칭된다. 본 개시물에서, 리프 CU의 네 개의 서브 CU들은 또한 원래의 리프 CU의 명시적인 분할이 없는 경우에도 리프 CU들이라고 지칭될 것이다. 예를 들어, 16x16 사이즈의 CU가 더 분할되지 않는다면, 네 개의 8x8 서브 CU들은 또한 16x16 CU가 전혀 분할되지 않았더라도 리프 CU들이라고 지칭될 것이다.

[0038] CU 는 CU 가 사이즈 구별을 갖지 않는 것을 제외하고, H.264 표준의 매크로블록과 유사한 목적을 갖는다. 예를 들어, 트리블록은 (서브-CU 들로서도 지칭되는) 4 개의 차일드 노드들로 분할될 수도 있고, 각각의 차일드 노드는 차례로 페어런트 노드일 수도 있고 다른 4 개의 차일드 노드들로 분할될 수도 있다. 쿼드트리의 리프 노드로서 지칭되는 마지막 분할되지 않은 차일드 노드는 리프-CU 로서도 지칭되는 코딩 노드를 포함한다. 코딩된 비트스트림과 연관된 신택스 데이터는 최대 CU 깊이로서 지칭되는, 트리블록이 분할될 수도 있는 최대 회수를 정의할 수도 있고, 또한 코딩 노드들의 최소 사이즈를 정의할 수도 있다. 이에 따라, 비트스트림은 또한 최소 코딩 유닛 (SCU) 을 정의할 수도 있다. 본 개시는 HEVC 의 컨텍스트에서 CU, PU, 또는 TU 의 임의의 것, 또는 다른 표준들의 컨텍스트에서 유사한 데이터 구조들 (예를 들어, H.264/AVC 에서의 매크로블록들 및 그것의 서브 블록들) 을 지칭하기 위해 용어 "블록" 을 사용한다.

[0039] CU 는 코딩 노드 및 그 코딩 노드와 연관된 예측 유닛들 (PUs) 및 변환 유닛들 (TUs) 을 포함한다. CU의 사이즈는 코딩 노드의 사이즈에 대응하고, 형상이 정사각형이어야만 한다. CU의 사이즈는 8x8 픽셀들로부터 최대 64x64 픽셀들 또는 그 이상을 갖는 트리블록의 사이즈까지의 범위일 수도 있다. 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다.

[0040] 일반적으로, PU 는 대응하는 CU 의 전부 또는 일부에 대응하는 공간 영역을 나타내고, 그 PU 에 대한 참조 샘플을 취출하는 데이터를 포함할 수도 있다. 게다가, PU 는 예측과 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라 모드 인코딩되는 경우, PU 에 대한 데이터는 PU 에 대응하는 TU 에 대한 인트라 예측 모드를 기술하는 데이터를 포함할 수도 있는 잔차 쿼드트리 (RQT) 에 포함될 수도 있다. 다른 예로서, PU 가 인터 모드 인코딩되는 경우, PU 는 PU 에 대한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형, 또는 비정사각형) 블록일 수도 있다. CU 의 PU 는 픽처의 루마 샘플들의 예측 블록, 크로마 샘플들의 2 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하기 위해 사용되는 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처 또는 3 개의 별개의 칼라 평면들을 갖는 픽처에서, PU 는 단일의 예측 블록 및 그 예측 블록 샘플들을 예측하기 위해 사용되는 신택스 구조들을 포함할 수도 있다.

[0041] TU 들은 변환, 예를 들어 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 잔차 비디오 데이터에 대한 개념적으로 유사한 변환의 적용에 후속하여 변환 도메인에서 계수들을 포함할 수도 있다. 잔차 데이터는 PU 들에 대응하는 예측 값들과 인코딩되지 않은 픽처의 픽셀들 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔차 데이터를 포함하는 TU 들을 형성하고, 그 후 CU 에 대한 변환 계수들을 생성하기 위해 TU 들을 변환할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 그 변환 블록 샘플들을 변환하기 위해 사용되는 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처 또는 3 개의 별개의 칼라 평면들을 갖는 픽처에서, TU 는 단일의 변환 블록 및 그 변환 블록 샘플들을 변환하기 위해 사용되는 신택스 구조들을 포함할 수도 있다.

[0042] 변환에 후속하여, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로 변화 계수들이 가능하게는 그 계수들을 표현하기 위해 사용되는 데이터의 양을 감소시키기 위해 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 깊

이를 감소시킬 수도 있다. 예를 들어, n-비트 값은 양자화 동안 m-비트 값으로 라운드 다운될 수도 있고, 여기서 n 은 m 보다 크다.

[0043] 비디오 인코더 (20) 는 변환 계수들을 스캐닝하여, 양자화된 변환 계수들을 포함하는 2차원 매트릭스로부터 1차원 벡터를 발생할 수도 있다. 스캐닝은 어레이의 앞부분에 더 높은 에너지 (따라서, 더 낮은 주파수) 계수들을 배치하고, 그리고 어레이의 뒷부분에 더 낮은 에너지 (따라서, 더 높은 주파수) 계수들을 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 발생하기 위해, 미리 정의된 스캐닝 순서를 이용하여, 양자화된 변환 계수들을 스캐닝할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캐닝을 수행할 수도 있다.

[0044] 양자화된 변환 계수들을 스캐닝하여 1차원 벡터를 형성한 후, 비디오 인코더 (20) 는 예컨대, 컨텍스트-적응 가변 길이 코딩 (CAVLC), 컨텍스트-적응 이진 산술 코딩 (CABAC), 신택스-기반 컨텍스트-적응 2진 산술 코딩 (SBAC), 확률 구간 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라서, 1차원 벡터를 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.

[0045] 비디오 인코더 (20) 는 블록-기반 신택스 데이터, 픽처-기반의 신택스 데이터, 및 픽처들의 그룹 (GOP)-기반 신택스 데이터와 같은 신택스 데이터를, 비디오 디코더 (30) 로, 예컨대, 픽처 헤더, 블록 헤더, 슬라이스 헤더, 또는 GOP 헤더로 추가로 전송할 수도 있다. GOP 신택스 데이터는 각각의 GOP 에서의 다수의 픽처들을 기술할 수도 있으며, 픽처 신택스 데이터는 대응하는 픽처를 인코딩하는데 사용되는 인코딩/예측 모드를 나타낼 수도 있다.

[0046] 비디오 디코더 (30) 는, 코딩된 비디오 데이터를 획득 시, 비디오 인코더 (20) 에 관해 설명된 인코딩 패스 (pass) 에 대해 일반적으로 상호역의 관계에 있는 디코딩 패스를 수행할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 획득할 수도 있다. 비디오 디코더 (30) 는 비트스트림에 포함된 데이터를 이용하여 원래의, 인코딩되지 않은 비디오 시퀀스를 재구성할 수도 있다.

[0047] 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 공간적 예측에 의존하여, 주어진 비디오 픽처 내의 비디오에서 공간적 리던던시를 감소시키거나 또는 제거한다. 인터 코딩은 시간적 예측에 의존하여, 비디오 시퀀스의 인접한 픽처들 내의 비디오에서 시간적 리던던시를 감소시키거나 또는 제거한다. 인트라 모드 (I 모드) 는 여러 공간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 과 같은 인터 모드들은 수개의 시간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다.

[0048] 일부 예들에서, 스크린 콘텐츠를 코딩할 때 등의 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 비디오 슬라이스들 내의 블록들을 코딩하기 위해 인트라 블록 카핑 (인트라 BC) 을 수행할 수도 있다. 인트라 BC 를 수행하기 위해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재 블록의 예측을 위해 현재 블록과 동일한 픽처에 있는 이전에 코딩된 비디오 데이터의 블록들을 이용할 수도 있다. 상기 논의된 바와 같이, 인트라 BC 는 인터-유사 코딩 툴이다 (픽처에 대한 픽셀 값들이 벡터에 의해 표시된 예측자 블록에 기초하여 예측됨을 의미한다). 하지만, 종래의 인트라 예측 픽처들에서의 인트라 BC 의 이용은 복잡성을 증가시키고 및/또는 인트라 BC 의 코딩 효율을 감소시킬 수도 있다. 추가적으로, 이 개시물의 기술들 이전에, 인트라 BC 를 이용한 비디오 코더들의 구현들은, (예를 들어, 인트라 BC 가 픽처 내부의 블록 카피를 초래하는 것으로 인해) 종래의 인트라- 및 인터-예측을 수행하기 위해 사용되는 것들에 비해 인트라 BC 를 수행하기 위한 추가적인 툴들로 구성될 필요가 있을 것이다. 이와 같이, 비디오 코더로 하여금, 현재 인트라 BC 에 적용되는 제약들의 일부 또는 전부를 유지하면서, 그리고 하드웨어 아키텍처에 대한 (중요한) 수정 없이, 인트라 BC 에 의해 제공되는 효율성들을 얻는 것이 가능하게 하는 것이 바람직할 수도 있다.

[0049] 이 개시물의 하나 이상의 기술들에 따라서, 종래의 인트라 예측 기법들을 이용하여 현재 픽처의 블록을 예측하기 위해 인트라 BC 를 수행하는 것과는 대조적으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 종래의 인터 예측과 유사한 기법들을 이용하여 인트라 BC 를 수행할 수도 있다. 실례로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재 픽처의 블록들을 예측하기 위해 사용되는 참조 픽처 리스트 (RPL) 에 현재 픽처의 표시를 삽입하고, 현재 픽처의 버전 (즉, 인스턴스) 을 참조 픽처 버퍼에 저장하며, 참조 픽처 버퍼에 저장된 현재 픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 블록

을 코딩할 수도 있다. 이러한 방식으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, 인트라 BC 에 현재 적용되는 제약들의 일부 또는 전부를 유지하면서, 인트라 BC 에 의해 제공되는 효율성들을 얻을 수도 있다. 또한 이러한 방식으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, 심각한 수정 없이 인트라 BC 에 대해 종래의 인트라 및 인터 프레임들에 대한 하드웨어 아키텍처를 재사용할 수도 있다.

[0050] 도 2 는, 표시 순서로, 픽처들 34, 35A, 36A, 38A, 35B, 36B, 38B, 및 35C 를 포함하는 예시적인 비디오 시퀀스 (33) 를 나타내는 개념도이다. 이들 픽처들의 하나 이상은 P-슬라이스들, B-슬라이스들, 또는 I-슬라이스들을 포함할 수도 있다. 일부 경우들에서, 비디오 시퀀스 (33) 는 픽처들의 그룹 (GOP) 으로서 지칭될 수도 있다. 픽처 (39) 는 비디오 시퀀스 (33) 후에 발생하는 시퀀스에 대한 디스플레이 순서에서 제 1 픽처이다. 도 2 는 일반적으로, 비디오 시퀀스에 대한 예시적인 예측 구조를 나타내고, 오직 상이한 인터-예측된 슬라이스 유형들을 인코딩하기 위해 사용되는 픽처 참조들을 나타내도록만 의도된다. 실제 비디오 시퀀스는 상이한 슬라이스 유형들을 상이한 디스플레이 순서로 포함하는 보다 많은 또는 보다 적은 비디오 픽처들을 포함할 수도 있다.

[0051] 블록-기반 비디오 코딩에 있어서, 비디오 시퀀스 (33) 에 포함된 비디오 픽처들의 각각은 비디오 블록들 또는 코딩 유닛 (CU) 들로 파티셔닝될 수도 있다. 비디오 픽처의 각각의 CU 는 하나 이상의 예측 유닛 (PU) 들을 포함할 수도 있다. 일부 예들에서, 픽처 내의 PU 들을 예측하기 위해 이용가능한 예측 방법들은 픽처 유형에 의존할 수도 있다. 하나의 예로서, 인트라-예측된 픽처 (I-픽처) 의 슬라이스들에서의 비디오 블록들 또는 PU 들은 인트라-예측 모드들 (즉, 동일한 픽처에서의 이웃하는 블록들에 대한 공간적 예측) 을 이용하여 예측될 수도 있다. 다른 예로서, 인터-예측된 픽처 (B-픽처 또는 P-픽처) 의 슬라이스들에서의 비디오 블록들 또는 PU 들은 인터 또는 인트라-예측 모드들 (즉, 동일한 픽처에서 이웃하는 블록들에 대한 공간 예측 또는 다른 참조 픽처들에 대한 시간적 예측) 을 이용하여 예측될 수도 있다. 달리 말하면, I-픽처는 I-슬라이스들을 포함할 수도 있고, P-픽처는 I-슬라이스들 및 P-슬라이스들 양자 모두를 포함할 수도 있고, B-픽처는 I-슬라이스들, P-슬라이스들, 및 B-슬라이스들을 포함할 수도 있다.

[0052] P-슬라이스의 비디오 블록들은 참조 픽처 리스트에서 식별된 참조 픽처로부터 단-방향 예측 코딩을 이용하여 인코딩될 수도 있다. B-슬라이스의 비디오 블록들은 다수의 참조 픽처 리스트들에서 식별된 다수의 참조 픽처로부터 양-방향 예측 코딩을 이용하여 인코딩될 수도 있다.

[0053] 도 2 의 예에서, 제 1 픽처 (34) 는 I-픽처로서 인트라-모드 코딩에 대해 지정된다. 다른 예들에서, 제 1 픽처 (34) 는 선행하는 시퀀스의 제 1 픽처를 참조하여, 예컨대 P-픽처, 또는 B-픽처로서, 인터-모드 코딩으로 코딩될 수도 있다. 비디오 픽처들 (35A-35C) (총칭하여 "비디오 픽처들 (35)") 은 과거 픽처 및 미래의 픽처를 참조하여 양방향-예측을 이용하여 B-픽처들로서 코딩을 위해 지정된다. 도 2 의 예에서 나타낸 바와 같이, 픽처 (35A) 는 픽처 (34) 및 픽처 (36A) 로부터 비디오 픽처 (35A) 로의 화살표들에 의해 표시되는 바와 같이, 제 1 픽처 (34) 및 픽처 (36A) 를 참조하여 B-픽처로서 인코딩될 수도 있다. 도 2 의 예에서, 제 1 픽처 (34) 및 픽처 (36A) 는 픽처 (35A) 의 블록들의 예측 동안 사용된 참조 픽처 리스트들에 포함될 수도 있다. 픽처들 (35B 및 35C) 은 유사하게 인코딩된다.

[0054] 비디오 픽처들 (36A-36B) (총칭하여 "비디오 픽처들 (36)") 은 과거 픽처를 참조하여 단-방향 예측을 이용하여 P-픽처들, 또는 B-픽처들로서 코딩을 위해 지정될 수도 있다. 도 2 의 예에서 나타낸 바와 같이, 픽처 (36A) 는 픽처 (34) 로부터 비디오 픽처 (36A) 로의 화살표에 의해 표시되는 바와 같이, 제 1 픽처 (34) 를 참조하여 P-픽처, 또는 B-픽처로서 인코딩될 수도 있다. 픽처 (36B) 는 유사하게, 픽처 (38A) 로부터 비디오 픽처 (36B) 로의 화살표에 의해 표시되는 바와 같이, 픽처 (38A) 를 참조하여 P-픽처, 또는 B-픽처로서 인코딩된다.

[0055] 비디오 픽처들 (38A-38B) (총칭하여 "비디오 픽처들 (38)") 은 동일한 과거 픽처를 참조하여 단-방향 예측을 이용하여 P-픽처들, 또는 B-픽처들로서 코딩을 위해 지정될 수도 있다. 도 2 의 예에서 나타낸 바와 같이, 픽처 (38A) 는 픽처 (36A) 로부터 비디오 픽처 (38A) 로의 2 개의 화살표들에 의해 표시되는 바와 같이, 픽처 (36A) 를 2 번 참조하여 인코딩될 수도 있다. 픽처 (38B) 는 유사하게 인코딩된다.

[0056] 일부 예들에서, 픽처들의 각각은 픽처들이 출력될 순서를 나타내는 고유 값 (즉, 특정 비디오 시퀀스, 예컨대, 디코딩 순서에서 순시 디코더 리프레쉬 (IDR) 픽처를 뒤따르는 픽처들의 시퀀스에 고유한) 을 할당받을 수도 있다. 이 고유 값은 픽처 순서 카운트 (picture order count; POC) 로서 지칭될 수도 있다. 일부 예들에서, 픽처들이 출력될 순서는 픽처들이 코딩되는 순서와는 상이할 수도 있다. 실례로, 픽처 (35A) 가 픽처 (36A) 전에 출력될 수도 있는 한편, 픽처 (36A) 가 픽처 (35A) 전체 코딩될 수도 있다.

[0057] 이 개시물의 하나 이상의 기술들에 따라서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 현재 픽처에서의 블록들을 예측하기 위해 사용되는 참조 픽처 리스트 (RPL) 에 현재 픽처를 삽입함으로써 인트라 BC 를 수행할 수도 있다. 실례로, 도 2 의 예에서, 비디오 코더는 픽처 (35A) 에서의 블록들을 예측하기 위해 사용된 RPL 들에, 픽처 (34) 및 픽처 (36A) 의 표시와 함께, 픽처 (35A) 의 표시를 삽입할 수도 있다. 비디오 코더는 그 다음, 픽처 (35A) 의 블록들을 코딩할 때 참조 픽처로서 픽처 (35A) 를 이용할 수도 있다.

[0058] 도 3 은 이 개시물에서 기술된 인트라 블록 카피를 위한 기법들을 이용할 수도 있는 비디오 인코더 (20) 의 일 예를 나타내는 블록도이다. 비디오 인코더 (20) 는 예시의 목적을 위해 하지만 다른 코딩 표준들에 관한 이 개시의 제한 없이 HEVC 코딩의 상황에서 설명될 것이다. 더욱이, 비디오 인코더 (20) 는 HEVC 의 범위 확장들에 따라서 기법들을 구현하도록 구성될 수도 있다.

[0059] 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라- 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 주어진 비디오 픽처 내에서 비디오에서의 공간적 리던던시를 감소 또는 제거하기 위해 공간적 예측에 의존한다. 인터-코딩은, 비디오 시퀀스의 인접 픽처들 내에서 비디오에서의 시간적 리던던시를 감소 또는 제거하기 위해 또는 다른 뷰들에서의 비디오와의 리던던시를 감소 도는 제거하기 위해 시간적 예측 또는 인터-뷰 예측에 의존한다.

[0060] 도 3 의 예에서, 비디오 인코더 (20) 는, 비디오 데이터 메모리 (40), 예측 프로세싱 유닛 (42), 참조 픽처 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 프로세싱 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함할 수도 있다. 예측 프로세싱 유닛 (42) 은 다시, 모션 추정 유닛 (44), 모션 보상 유닛 (46), 및 인트라-예측 유닛 (48) 을 포함한다. 비디오 블록 재구성들을 위해, 비디오 인코더 (20) 는 또한, 양자화 프로세싱 유닛 (58), 역 변환 프로세싱 유닛 (60), 및 합산기 (62) 를 또한 포함한다. 디블록킹 필터 (도 3 에는 미도시) 는 또한 재구성된 비디오로부터 블록키니스 아티팩트들 (blockiness artifacts) 을 제거하기 위해 블록 경계들을 필터링하기 위해 포함될 수도 있다. 원하는 경우, 디블록킹 필터는 합산기 (62) 의 출력을 통상적으로 필터링할 것이다. 디블록킹 필터에 추가하여 (루프 또는 포스트 루프에서) 추가적인 루프 필터들이 또한 사용될 수도 있다.

[0061] 비디오 데이터 메모리 (40) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (40) 에 저장된 비디오 데이터는, 예를 들어 비디오 소스 (18) 로부터 획득될 수도 있다. 참조 픽처 메모리 (64) 는 (예컨대, 인트라- 또는 인터-예측 코딩 모드들로서도 지칭되는, 인트라- 또는 인터-코딩 모드들에서) 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩하는데 사용하기 위한 참조 비디오 데이터를 저장하는 디코딩 픽처 버퍼 (DPB) 의 하나의 예이다. 비디오 데이터 메모리 (40) 및 참조 픽처 메모리 (64) 는 동기식 DRAM (SDRAM), 자기저항식 RAM (MRAM), 저항식 RAM (RRAM) 을 포함하는 동적 랜덤 액세스 메모리 (DRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 임의의 다양한 메모리 디바이스들에 의해 형성될 수도 있다. 비디오 데이터 메모리 (40) 및 참조 픽처 메모리 (64) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 여러 예들에서, 비디오 데이터 메모리 (40) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 함께 온-칩일 수도 있거나, 그들 컴포넌트들에 대해 오프-칩일 수도 있다.

[0062] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩될 비디오 픽처 또는 슬라이스를 수신한다. 그 픽처 또는 슬라이스는 다수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (44) 및 모션 보상 유닛 (46) 은 시간적 압축을 제공하기 위해 또는 인터-뷰 예측을 제공하기 위해 하나 이상의 참조 픽처들 내의 하나 이상의 블록들에 대해 수신된 비디오 블록의 인터 예측 코딩을 수행한다. 인트라 예측 유닛 (48) 은 대안적으로 공간적 압축을 제공하기 위해 코딩될 블록과 동일한 픽처 또는 슬라이스 내의 하나 이상의 이웃 블록들에 대해 수신된 비디오 블록의 인트라 예측 코딩을 수행할 수도 있다. 비디오 인코더 (20) 는 (예를 들어 비디오 데이터의 각 블록에 대한 적절한 코딩 모드를 선택하기 위해) 다수의 코딩 패스들을 수행할 수도 있다.

[0063] 게다가, 파티션 유닛 (미도시) 은 이전의 코딩 패스들에서의 이전의 파티셔닝 스킴들의 평가에 기초하여 비디오 데이터의 블록들을 서브 블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛은 초기에 픽처 또는 슬라이스를 LCU들로 파티셔닝하고, 레이트 왜곡 분석 (예컨대, 레이트 왜곡 최적화) 에 기초하여 그 LCU들의 각각을 서브 CU들로 파티셔닝할 수도 있다. 예측 프로세싱 유닛 (42) 은 LCU의 서브 CU들로의 파티셔닝을 나타내는 쿼드트리 데이터 구조를 추가로 생성할 수도 있다. 쿼드트리의 리프 노드 CU들은 하나 이상의 PU들과 하나 이상의 TU들을 포함할 수도 있다.

[0064] 예측 프로세싱 유닛 (42) 은 예를 들어 에러 결과들에 기초하여 코딩 모드들 중 하나, 인트라 또는 인터를 선택할 수도 있고, 참조 픽처로서 사용을 위해 인코딩된 블록을 재구성하기 위해 합산기 (62) 로 그리고 잔차 블록 데이터를 생성하기 위해 합산기 (50) 로 결과의 인트라 또는 인터 코딩된 블록을 제공한다. 예측 프로세싱 유닛 (42) 은 또한 엔트로피 코딩 유닛 (56) 으로 모션 벡터들, 인트라 모드 표시자들, 파티션 정보, 및 다른 그러한 신택스 정보와 같은 신택스 엘리먼트들을 제공한다.

[0065] 모션 추정 유닛 (44) 및 모션 보상 유닛 (46) 은 고도로 집적될 수도 있지만, 개념적 목적으로 별개로 도시된다. 모션 추정 유닛 (44) 에 의해 수행되는 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어, 현재의 픽처 (또는 다른 코딩된 유닛) 내에서 코딩되고 있는 현재의 블록에 대해 참조 픽처 (또는 다른 코딩된 유닛) 내의 예측 블록에 대한 현재 비디오 픽처 내의 비디오 블록의 PU의 변위 (displacement) 를 나타낼 수도 있다. 예측 블록은 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 픽셀 차이의 관점에서 코딩될 비디오 블록과 밀접하게 매칭하는 것으로 발견되는 블록이다. 몇몇 예들에서, 비디오 인코더 (20) 는 참조 픽처 메모리 (64) 에 저장된 참조 픽처들의 서브 정수 (sub-integer) 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 보간할 수도 있다. 그러므로, 모션 추정 유닛 (44) 은 풀 (full) 픽셀 위치들 및 분수 픽셀 위치들에 대한 모션 검색을 수행하여 분수 픽셀 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0066] 모션 추정 유닛 (44) 은 PU의 위치와 참조 픽처의 예측 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽처는 참조 픽처 메모리 (64) 에 저장된 하나 이상의 참조 픽처들을 식별하는 하나 이상의 참조 픽처 리스트 (RPL) 들로부터 선택될 수도 있다. 모션 추정 유닛 (44) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 과 모션 보상 유닛 (46) 으로 전송한다. 일부 예들에서, 모션 추정 유닛 (44) 은 선택된 참조 픽처의 표시를 엔트로피 인코딩 유닛 (56) 에 전송할 수도 있다.

[0067] 일부 예들에서, 모션 추정 유닛 (44) 은, 출력 순서에서 현재 픽처 전에 있는 (즉, 현재 픽처보다 더 낮은 POC 값들을 갖는) 비디오 데이터의 픽처들을 포함할 수도 있는 제 1 RPL (List 0) 및 출력 순서에서 현재 픽처 후에 있는 (즉, 현재 픽처보다 더 높은 POC 값들을 갖는) 비디오 데이터의 픽처들을 포함할 수도 있는 제 2 RPL (List 1) 을 발생시킬 수도 있다. 일부 예들에서, RPL 들은 단기 (short-term) 및 장기 (long-term) 픽처들을 포함할 수도 있고, 이들은, 장기 픽처들이 단기 픽처들보다 더 길게 참조 픽처 메모리 (64) 에 저장될 수도 있다는 점에서 구별될 수도 있다. 일부 예들에서, 참조 픽처 메모리 (64) 에서의 픽처는 예컨대 장기, 단기 등으로 마킹될 수도 있다.

[0068] 일부 예들에서, 모션 추정 유닛 (44) 은 하나 이상의 참조 픽처 셋트들 (reference picture sets; RPSs) 에 기초하여 하나 이상의 RPL 들을 발생시킬 수도 있다. 하나 이상의 RPS 들은, 1) 단기 RPS 전 또는 RefPicSetStCurrBefore 으로서 지칭되는, 출력 순서에서 현재 픽처 전에 있고 현재 픽처를 예측하기 위해 이용가능한 단기 픽처들의 하나 이상의 셋트들, 2) 단기 RPS 후 또는 efPicSetStCurrAfter 로서 지칭되는, 출력 순서에서 현재 픽처 후에 있고 현재 픽처를 예측하기 위해 이용가능한 단기 픽처들의 셋트, 3) 단기 RPS 이용불가능 또는 RefPicSetStFoll 로서 지칭되는, 현재 픽처를 예측하기 위해 이용가능하지 않지만 코딩 순서에서 후속하는 픽처를 코딩하기 위해 이용될 수도 있는 단기 픽처들의 셋트, 4) 장기 RPS 이용가능 또는 RefPicSetLtCurr 로서 지칭되는, 현재 픽처를 예측하기 위해 이용가능한 장기 픽처들의 셋트, 및/또는 5) 장기 RPS 이용불가능 또는 RefPicSetLtFoll 로서 지칭되는, 현재 픽처를 예측하기 위해 이용불가능한 장기 픽처들의 셋트를 포함할 수도 있다.

[0069] 일부 예들에서, 모션 추정 유닛 (44) 은 특정 순서로 하나 이상의 RPL 들에서 하나 이상의 RPS 들에서의 픽처들을 포함할 수도 있다. 실례로, RPL 을 발생시키기 위해, 모션 추정 유닛 (44) 은 먼저 현재 픽처를 예측하기 위해 이용가능한 단기 RPS 들 (예컨대, RefPicSetStCurrBefore 및/또는 RefPicSetStCurrAfter) 를 포함할 수도 있고, 이어서, 현재 픽처를 예측하기 위해 이용가능한 장기 RPS 들 (예컨대, RefPicSetLtCurr) 에서의 픽처들을 포함할 수도 있다. 일부 예들에서, RPL 에서의 각각의 엔트리는, RPL 에서 더 일찍 포함된 픽처들 (예컨대, 단기 RPS 들에서의 픽처들) 이 RPL 에서 더 나중에 포함된 픽처들 (예컨대, 장기 RPS 들에서의 픽처들) 보다 더 낮은 인덱스 값들을 가지도록 하는 인덱스 값을 가질 수도 있다.

[0070] 상기 논의된 바와 같이, 모션 추정 유닛 (44) 은 선택된 참조 픽처의 표시를 엔트로피 인코딩 유닛 (56) 에 전송할 수도 있다. 일부 예들에서, 모션 추정 유닛 (44) 은 RPL 내의 선택된 참조 픽처의 인덱스 값을 전송함

으로써 그 표시를 전송할 수도 있다.

[0071] 일부 예들에서, 모션 추정 유닛 (44) 은 비디오 디코더로 하여금 하나 이상의 이전 RPL 들로부터 하나 이사으이 현재 RPL 들을 예측하는 것을 가능하게 하기 위한 정보를 출력할 수도 있다. 실례로, 모션 추정 유닛 (44) 은 비디오 디코더로 하여금현재 슬라이스에 대한 하나 이상의 RPL 들을 발생시키기 위해 이전 슬라이스에 대한 하나 이상의 RPL 들을 수정하는 것을 가능하게 하는 하나 이상의 신택스 엘리먼트들을 출력하게 할 수도 있다.

[0072] 이 개시물의 하나 이상의 기술들에 따라서, 참조 픽처들로서 다른 픽처들을 이용하도록 인터-예측을 제한하는 것과는 대조적으로, 모션 추정 유닛 (44) 은 현재 픽처에 포함된 비디오 데이터의 블록들을 예측하기 위해 참조 픽처로서 현재 픽처를 이용할 수도 있다. 예를 들어, 모션 추정 유닛 (44) 은 참조 픽처 메모리 (64) 에 현재 픽처의 버전을 저장할 수도 있다. 일부 예들에서, 모션 추정 유닛 (44) 은 고정된 값으로 초기화된 픽셀 값들로 현재 픽처의 초기화된 버전을 저장할 수도 있다. 일부 예들에서, 그 고정 값은 현재 픽처의 샘플들의 비트 심도 (depth) 에 기초할 수도 있다. 실례로, 고정 값은 1<<(비트심도 -1) 일 수도 있다. 일부 예들에서, 모션 추정 유닛 (44) 은 현재 픽처의 임의의 블록들을 코딩하기 전에 현재 픽처의 초기화된 버전을 저장할 수도 있다. 현재 픽처의 초기화된 버전을 저장함으로써, 모션 추정 유닛 (44) 은 예측 블록들에 대한 탐색 (즉, 탐색 구역) 을 이미 재구성된 블록들로 제한하도록 요구되지 않을 수도 있다. 반면, 모션 추정 유닛 (44) 이 현재 픽처의 초기화된 버전을 저장하지 않는 경우에, 예측 블록들에 대한 탐색은 예를 들어 디코더/인코더 미스매치를 회피하기 위해 이미 재구성된 블록들로 제한될 수도 있다.

[0073] 예측 프로세싱 유닛 (42) 은 현재 픽처에 대한 하나 이상의 RPL 들을 발생시킬 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은 현재 픽처에 대해 현재 픽처 RPL 을 포함시킬 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 RPL 내의 특정 위치에 현재 픽처를 포함할 수도 있다. 하나의 예로서, 예측 프로세싱 유닛 (42) 은 장기 RPS 에서의 픽처들 전에 RPL 에서 현재 픽처를 삽입할 수도 있다. 예를 들어, 예측 프로세싱 유닛 (42) 은 장기 RPS 로부터의 픽처들의 인덱스 값들보다 더 적은 인덱스 값으로 RPL 에서 현재 픽처를 삽입할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 장기 RPS 에서의 픽처들 바로 전에 RPL 에서 현재 픽처를 삽입할 수도 있다.

[0074] 다른 예로서, 예측 프로세싱 유닛 (42) 은 장기 RPS 로부터의 픽처들을 삽입한 후에 RPL 에서 현재 픽처를 삽입할 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은 장기 RPS 에서의 픽처들의 인덱스 값들보다 더 큰 인덱스 값으로 RPL 에서 현재 픽처를 삽입할 수도 있다. 일부 예들에서, 모션 추정 유닛 (44) 은 장기 RPS 로부터의 픽처들을 삽입한 바로 후에 RPL 에서 현재 픽처를 삽입할 수도 있다.

[0075] 다른 예로서, 예측 프로세싱 유닛 (42) 은 고정 위치에서 RPL 에 현재 픽처를 삽입할 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은 고정된 인덱스 값으로 RPL 에서 현재 픽처를 삽입할 수도 있다. 일부 예들에서, 고정된 인덱스 값은 -1, 또는 num_ref_idx_l1_active_minus1+1 일 수도 있다. 이러한 예들의 일부에서, 모션 추정 유닛 (44) 은 현재 블록이 인트라 BC 를 이용하여 코딩되는 것을 나타내는 플래그 (즉, intra_bc_flag) 를 코딩하지 않을 수도 있다.

[0076] 일부 예들에서, 모션 벡터들이 시간적 모션 벡터 예측 (temporal motion vector prediction; TMVP) 을 이용하여 예측되는 경우 등에서, 모션 추정 유닛 (44) 은 현재 픽처가 그 자신이 병치된 (collocated) 픽처로서 사용되지 않도록 하나 이상의 제약들을 적용할 수도 있다. 실례로, 모션 추정 유닛 (44) 은, X 가 collocated_from_l0_flag 와 동일한 경우에 RefPicListX[collocated_ref_idx] 가 현재 픽처가 아니도록 TMVP 에 대해 사용되는 병치된 픽처의 참조 인덱스를 특정하는 신택스 엘리먼트 (예컨대, collocated_ref_idx) 를 코딩할 수도 있다.

[0077] 상기 논의된 바와 같이, 비디오 데이터의 현재 픽처의 비디오 데이터의 블록을 인코딩할 때, 모션 추정 유닛 (44) 은 현재 블록에 밀접하게 매칭되는 예측적 블록을 선택할 수도 있다. 이 개시물의 하나 이상의 기술들에 따라서, 다른 픽처들의 블록들을 탐색하는 것과는 대조적으로 (또는 그것에 추가하여), 모션 추정 유닛 (44) 은 현재 픽처의 현재 블록에 대한 예측적 블록으로서의 사용을 위해 현재 픽처에서 위치된 블록을 선택할 수도 있다. 예를 들어, 모션 추정 유닛 (44) 은, 현재 픽처를 포함하는, 하나 이상의 참조 픽처들을 포함하는 픽처들에 대해 탐색을 수행할 수도 있다. 각 픽처에 대해, 모션 추정 유닛 (44) 은, 예컨대 SAD (sum of absolute differences), SSD (sum of squared differences), MAD (mean absolute difference), MSD (mean squared difference) 등을 이용하여, 예측된 블록이 현재 블록에 얼마나 잘 매칭하는지를 반영하는 탐색 결과들을 산출할 수도 있다. 그 다음, 모션 추정 유닛 (44) 은 현재 블록에 대해 최상의 매치를 갖는 픽처에서의 블록을 식별하고, (현재 픽처일 수도 있는) 픽처 및 블록의 위치를 예측 프로세싱 유닛 (42) 에 대해 표시할 수

도 있다. 이러한 방식으로, 모션 추정 유닛 (44) 은, 예컨대 예측된 블록이 현재 픽처, 즉, 예측되고 있는 현재 블록과 동일한 픽처에 포함되는 것을 모션 추정 유닛 (44) 이 결정할 때, 인트라 BC 를 수행할 수도 있다.

[0078] 일부 예들에서, 모션 추정 유닛 (44) 은 현재 픽처에서 예측적 블록에 대한 탐색을 제한할 수도 있다. 실례로, 현재 블록이 현재 슬라이스에 위치되는 경우에, 모션 추정 유닛 (44) 은 예측 블록에 대한 탐색을 현재 슬라이스의 이전에 코딩된 영역들 (예컨대, 현재 슬라이스에서의 현재 블록의 위 및/또는 좌측 영역들) 로 제한할 수도 있다. 이와 같이, 일부 예들에서, 현재 픽처의 이전 슬라이스들의 이전에 코딩된 영역들은 인트라 BC 를 수행하기 위한 예측적 블록들로서 사용되지 않을 수도 있다. 하지만, 상기 논의된 바와 같이, 일부 예들에서, 현재 픽처의 이전 슬라이스들의 이전에 코딩된 영역들은 인트라 BC 를 수행하기 위한 예측적 블록들로서 사용될 수도 있다. 일부 예들에서, 모션 추정 유닛 (44) 은 인터 참조 픽처들에서의 모션 탐색에서와 유사한 제한들 (예컨대, 모션 추정의 유사한 탐색 범위) 을 이용하여 예측적 블록에 대한 탐색을 제한할 수도 있다.

[0079] 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 현재 픽처가 현재 픽처를 예측하기 위해 사용된 RPL 에서 존재할 수도 있는지 여부를 나타내기 위해 하나 이상의 신택스 엘리먼트들을 인코딩하게 할 수도 있다. 하나의 예로서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 비디오 데이터의 픽처들이 그 자신드을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는지 여부를 나타내는 단일 신택스 엘리먼트를 인코딩하게 할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 현재 픽처에 의해 참조되는 비디오 파라미터 셋트 (VPS), 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS), 또는 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (PPS) 에서 단일 신택스 엘리먼트를 포함하게 할 수도 있다.

[0080] 또 다른 예로서, 예측 프로세싱 유닛 (42) 은 엔트로피 디코딩 유닛 (70) 으로 하여금 현재 픽처가 그 현재 픽처를 예측하기 위해 사용된 RPL 에서 존재할 수도 있는지 여부를 나타내기 위해 다수의 신택스 엘리먼트들을 인코딩하게 할 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 비디오 데이터의 픽처들이 비디오 데이터의 각각의 픽처들을 예측하기 위해 사용된 (즉, 그 자신들을 예측하기 위해 사용된) RPL 들에서 존재할 수도 있는지 여부를 나타내는 제 1 신택스 엘리먼트를 인코딩할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 현재 픽처에 의해 참조되는 VPS, 현재 픽처에 의해 참조되는 SPS, 또는 현재 픽처 (또는 현재 픽처의 슬라이스) 에 의해 참조되는 PPS 에서 제 1 신택스 엘리먼트를 포함하게 할 수도 있다. 일부 예들에서, 신택스 엘리먼트가, 비디오 데이터의 픽처들이 그 자신들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는 것을 나타내는 경우 등에, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 비디오 데이터의 현재 픽처가 현재 슬라이스를 예측하기 위해 사용된 RPL 에서 존재할 수도 있는지 여부를 나타내는 제 2 신택스 엘리먼트를 인코딩하게 할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 현재 슬라이스의 슬라이스 헤더에 제 2 신택스 엘리먼트를 포함하게 할 수도 있다.

[0081] 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 블록이 인트라 블록 카피 (Intra Block Copy; Intra BC) 를 이용하여 코딩되는지 여부를 나타내는 신택스 엘리먼트를 인코딩하게 하지 않을 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 이 개시물의 기술들에 따라서 인트라 BC 를 이용하여 예측되는 블록들의 코딩 유닛 신택스에서 intra_bc_flag 를 인코딩하게 하지 않을 수도 있다.

[0082] 일부 예들에서, 인트라 BC 를 이용하여 B-슬라이스 및 P-슬라이스에서 블록들을 코딩하는 것에 추가하여, 예측 프로세싱 유닛 (42) 은 현재 픽처의 I-슬라이스의 블록들을 인코딩하기 위해 현재 픽처를 포함하는 하나 이상의 RPL 들을 구성할 수도 있다. 일부 이러한 예들에서, 하나 이상의 RPL 들은 오직 현재 픽처를 포함하는 것으로 가정될 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 현재 픽처가 현재 픽처에 포함된 I-슬라이스들의 참조 픽처로서 사용될 수도 있는지 여부를 나타내기 위한 신택스 엘리먼트를 인코딩하게 할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 현재 픽처에 의해 참조되는 VPS, 현재 픽처에 의해 참조되는 SPS, 또는 현재 픽처에 의해 참조되는 PPS, 또는 현재 I-슬라이스의 슬라이스 헤더에서 제 1 신택스 엘리먼트를 포함하게 할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 여전히 AMVP 및 병합의 일방 또는 양방을 이용할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금, 디코더가 고정 값, 예컨대, 0 으로서 타겟 참조 인덱스를 도출하도록, I-슬라이스에 대해 AMVP 에 대한 타겟 참조 인덱스를 시그널링하게 하지 않을 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 I-슬라이스에 대해 AMVP 에 대한 타겟 참조 인덱스를 시그널링하게 할 수도 있지만, 타겟 참조 인덱스의

값은 고정 값, 예컨대 0 으로 제약될 수도 있다.

[0083] 상기 논의된 바와 같이, 참조 픽처 메모리 (64) 에 의해 저장된 픽처들은 단기, 장기, 다른 마킹으로서 마킹될 수도 있고, 및/또는 마킹되지 않을 수도 있다. 일부 예들에서, 현재 슬라이스가 I-슬라이스이고 인트라 BC 가 인에이블될 때 등의 경우에, 예측 프로세싱 유닛 (42) 은 현재 픽처를 장기 또는 단기 중 어느 일방으로서 마킹할 수도 있다. 일부 예들에서, 현재 슬라이스가 I-슬라이스이고 인트라 BC 가 인에이블될 때 등의 경우에, 예측 프로세싱 유닛 (42) 은 현재 픽처를 장기 또는 단기 중 어느 일방으로서 마킹하지 않을 수도 있다.

[0084] 일부 예들에서, 예측 프로세싱 유닛 (42) 은 현재 픽처를 코딩하기 전에 현재 픽처를 장기로서 마킹하고 현재 픽처를 코딩한 후에 현재 픽처를 단기로서 마킹할 수도 있다. 이러한 예들의 일부에서, 현재 슬라이스가 I-슬라이스인 경우 등에서, 예측 프로세싱 유닛 (42) 은, ("장기" 로서 마킹된) 인트라 BC 참조를 순전히 참조하는 후보들 또는 ("단기" 로서 마킹된) 인터 참조들을 참조하는 다른 후보들을 포함하도록 병합 후보 리스트 (merge candidate list) 를 생성할 수도 있다. 이러한 방식으로, 예측 프로세싱 유닛 (42) 은 인트라 BC 후보들 및 보통의 (인터-예측) 후보들 양자를 모두 포함하도록 후보 리스트를 생성할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 인트라 BC 참조 및 인터-예측 참조로부터 병합 후보를 양방향으로 예측할 수도 있다. 이러한 예들의 일부에서, 현재 슬라이스가 인터 코딩된 슬라이스 (예컨대, B-슬라이스 또는 P-슬라이스) 인 경우 등에서, 예측 프로세싱 유닛 (42) 은 현재 블록이 인트라 BC 로 코딩되는지 여부를 나타내는 플래그 (intra_bc_flag) 를 나타내는 추가적인 플래그를 생략할 수도 있다.

[0085] 일부 예들에서, 현재 프레임이 추가적인 참조 픽처라고 가정함으로써 인트라 BC 가 인에이블되고 시간적 모션 벡터 예측 (TMVP) 이 인에이블되는 경우 등에서, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금 타겟 병합 참조 픽처가 RPL 에서 현재 픽처 또는 제 1 픽처인지 여부를 나타내는 신택스 엘리먼트 (예컨대, RefPicList0[0] 또는 RefPicList1[0] 중 어느 일방) 를 코딩하게 할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은, 참조들의 장기 (인트라 BC) 카테고리 또는 참조들의 단기 카테고리에 대한 타겟 인덱스를 도출 또는 시그널링하는 것, 및 병치된 블록의 참조 픽처의 카테고리에 기초하여 상이한 타겟 병합 인덱스를 적용하는 것에 의해 참조 픽처를 식별할 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은 참조들의 장기 (인트라 BC) 카테고리 또는 참조들의 단기 카테고리에 대한 타겟 인덱스를 시그널링 또는 도출하는 것에 의해 참조 픽처를 식별할 수도 있고, 병치된 블록의 참조 픽처의 카테고리에 기초하여 상이한 타겟 병합 인덱스가 적용된다.

[0086] 상기 논의된 바와 같이, 예측 프로세싱 유닛 (42) 은 비디오 데이터의 현재 블록과 비디오 데이터의 예측자 블록 사이의 변위를 나타내는 모션 벡터를 결정하고, 그 결정된 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (46) 에 출력할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 정수 정밀도를 갖는 모션 벡터를 결정할 수도 있다. 이러한 예들에서, 현재 픽처가 장기 참조 픽처로서 마킹되는 경우 등에서, 예측 프로세싱 유닛 (42) 은 현재 픽처를 예측하기 위해 보통의 장치 참조 픽처들 (즉, 현재 픽처가 아닌 장기 참조 픽처들) 을 이용하지 않을 수도 있다. 또한, 이러한 예들에서, 예측 프로세싱 유닛 (42) 은, 예를 들어 인트라 BC 모션 벡터가 공간적 및 시간적 이웃 블록들의 인트라 BC 모션 벡터들에만 기초하여 예측됨에 따라, 어떤 변경도 없이 어드밴스드 모션 벡터 예측 (advanced motion vector prediction; AMVP) 또는 병합 관련 디코딩 프로세스를 HEVC 버전 1 에서와 같이 이용할 수도 있다.

[0087] 일부 예들에서, 예측 프로세싱 유닛 (42) 은 상이한 레벨들의 정밀도를 갖는 모션 벡터를 결정할 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은 정수 정밀도, 디폴트 정밀도, 또는 가장 미세한 모션 정밀도 (예컨대, HEVC 에서 1/4 픽셀 ("pel") 정밀도) 로 모션 벡터를 결정할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은, 예컨대, 현재 픽처에 의해 참조되는 SPS 또는 VPS 에서, 코딩된 인트라 BC 모션 벡터들의 정밀도를 나타내는 신택스 엘리먼트를 인코딩할 수도 있다. 일부 예들에서, 인트라 BC 모션 벡터들의 정밀도는 픽처 레벨에서 적응적일 수도 있고, 예측 프로세싱 유닛 (42) 은 엔트로피 인코딩 유닛 (56) 으로 하여금, 예컨대, 현재 블록에 의해 참조되는 슬라이스 또는 PPS 에서, 코딩된 인트라 BC 모션 벡터들의 정밀도를 나타내는 신택스 엘리먼트를 인코딩하게 할 수도 있다.

[0088] 일부 예들에서, 예측 프로세싱 유닛 (42) 은 인트라 BC 모션 벡터들의 정밀도 레벨에 대해 보상하기 위한 하나 이상의 동작들을 수행할 수도 있다. 하나의 예로서, 블록들을 참조 픽처 메모리 (64) 에 저장하기 전에, 예측 프로세싱 유닛 (42) 은, 가장 미세한 정밀도가 1/4-pel 일 때 2 만큼, 또는 좌측 시프트 후에 +/-2 등 라운딩의 임의의 다른 수단에 의해, 인트라 BC 로 인코딩된 각 블록의 모션 벡터의 x- 및 y-성분들을 좌측 시프트할 수도 있다. 다른 예들로서, 정수의 정밀도를 갖는 인트라 BC 모션 벡터들로 현재 슬라이스를 코딩할 때, 예

측 프로세싱 유닛 (42) 은 가장 미세한 정밀도가 1/4-pel 일 때 2 만큼 등에 의해 각 인트라 BC 코딩된 블록의 모션 벡터가 우측 시프트되는 방식으로 병치된 픽처를 프로세싱할 수도 있다. 일부 예들에서, 현재 슬라이스가 가장 미세한 정밀도를 갖는 인트라 BC 모션 벡터들로 코딩될 때 등의 경우에, 예측 프로세싱 유닛 (42) 은 모션 벡터 우측 시프트의 상기 프로세스를 적용하지 않을 수도 있다.

[0089] 일부 예들에서, 현재 픽처가 장기 참조 픽처로서 마킹되는 경우에, 예측 프로세싱 유닛 (42) 은 현재 픽처를 예측하기 위해 여전히 보통의 장기 참조 픽처들을 이용할 수도 있다. 보통의 장기 참조 픽처를 참조하는 모션 벡터 및 현재 픽처를 참조하는 모션 벡터가 병합 또는 AMVP 동안 서로로부터 예측하는 것을 방지하기 위해, 예측 프로세싱 유닛 (42) 은 병합 또는 AMVP 프로세스 동안 보통의 장기 참조 픽처와 현재 픽처를 구분할 수도 있다. 실례로, 예측 프로세싱 유닛 (42) 은, 참조 픽처의 픽처 순서 카운트 (POC) 값이 현재 픽처의 POC 와 동일한지 여부를 결정함으로써 현재 픽처로부터 보통의 장기 참조 픽처를 구분할 수도 있다.

[0090] 모션 보상 유닛 (46) 에 의해 수행되는 모션 보상은 모션 추정 유닛 (44) 에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페치하는 것 또는 생성하는 것을 수반할 수도 있다. 다시, 모션 추정 유닛 (44) 및 모션 보상 유닛 (46) 은 일부 예들에서 기능적으로 통합될 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터를 수신 시, 모션 보상 유닛 (46) 은 참조 픽처 리스트 (RPL) 들 중 하나에서 모션 벡터가 가리키는 예측 블록을 로케이팅할 수도 있다. 합산기 (50) 는 이하에서 논의되는 바와 같이, 코딩되고 있는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 대체로, 모션 추정 유닛 (44) 은 루마 성분들에 관하여 모션 추정을 수행하고, 모션 보상 유닛 (46) 은 크로마 성분들 및 루마 성분들 양쪽 모두에 대해 루마 성분들에 기초하여 계산된 모션 벡터들을 사용한다. 예측 프로세싱 유닛 (42) 은 또한 비디오 슬라이스의 비디오 블록들을 디코딩하는데 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0091] 인트라 예측 유닛 (48) 은 위에서 설명된 바와 같이, 모션 추정 유닛 (44) 및 모션 보상 유닛 (46) 에 의해 수행된 인터 예측에 대한 대안으로서 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 유닛 (48) 은 현재의 블록을 인코딩하는데 사용하기 위한 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 유닛 (48) 은 예컨대, 개별 인코딩 패스들 동안에 다양한 인트라 예측 모드들을 사용하여 블록들을 인코딩할 수도 있고, 인트라 예측 유닛 (46) 은 복수의 인트라 예측 모드들로부터 사용할 적절한 인트라 예측 모드를 선택할 수도 있다.

[0092] 예를 들어, 인트라 예측 유닛 (48) 은 다양한 테스트된 인트라 예측 모드들에 대한 레이트 왜곡 분석을 사용하여 레이트 왜곡 값들을 계산할 수도 있고, 테스트된 모드들 중에서 최선의 레이트 왜곡 특성들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록과 인코딩된 블록을 생성하기 위해 인코딩되었던 오리지날, 인코딩되지 않은 블록 사이의 왜곡 (또는 에러) 의 양 뿐만아니라 인코딩된 블록을 생성하는데 사용된 비트레이트 (다시 말하면, 비트들의 수) 를 결정한다. 인트라 예측 유닛 (48) 은 여러 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 계산하여 어떤 인트라 예측 모드가 그 블록에 대한 최선의 레이트 왜곡 값을 나타내는지를 결정할 수도 있다.

[0093] 일부 예들에서, 인트라 예측 유닛 (48) 에 의한 사용을 위해 이용가능한 복수의 인트라-예측 모드들은 평면형 예측 모드, DC 예측 모드, 및 하나 이상의 각도 예측 모드들을 포함할 수도 있다. 선택된 모드에 관계없이, 인트라 예측 유닛 (48) 은 항상 현재 블록에 인접한 재구성된 블록들에 기초하여 현재 블록을 예측할 수도 있다. 하나의 예로서, 평면형 예측 모드를 이용할 때, 인트라 예측 유닛 (48) 은 결정된 수평 및 수직 예측들을 평균화함으로써 현재 블록을 예측할 수도 있다. 일부 예들에서, 인트라 예측 유닛 (48) 은, (현재 블록을 예측할 때 우측 이웃하는 블록의 샘플들이 재구성되지 않을 수도 있음에 따라) 좌측 이웃하는 블록 및 상부-우측 이웃하는 블록에 기초하여 수평 예측들을 결정하고, (현재 블록을 예측할 때 하부 이웃하는 블록의 샘플들이 재구성되지 않을 수도 있음에 따라) 상부 이웃하는 블록 및 하부-좌측 이웃하는 블록에 기초하여 수직 예측들을 결정할 수도 있다.

[0094] 다른 예로서, DC 예측 모드를 이용할 때, 인트라 예측 유닛 (48) 은 상수 값으로 현재 블록의 샘플들을 예측할 수도 있다. 일부 예들에서, 상수 값은 좌측-이웃하는 블록에서의 샘플들 및 상부 이웃하는 블록에서의 샘플들의 평균을 나타낼 수도 있다. 다른 예로서, 하나 이상의 각도 예측 모드들 중 하나를 이용할 때, 인트라 예측 유닛 (48) 은 예측 방향에 의해 표시되는 이웃하는 블록으로부터의 샘플들에 기초하여 현재 블록의 샘플들을 예측할 수도 있다. 도 4 는 인트라 예측 유닛 (48) 에 의해 사용될 수도 있는 각도 인트라-예측 모드들

의 일 예를 나타낸다. 도 4 에 의해 도시된 화살표들은 예측 방향 (즉, 현재 블록으로부터 멀리 연장됨) 을 나타낸다.

[0095] 일부 예들에서, 복수의 인트라-예측 모드들의 각각은 대응하는 모드 인덱스를 가질 수도 있고, 이는 인트라 예측 유닛 (48) 에 의해 시그널링 (즉, 비디오 디코더로) 될 수도 있다. 도 4 에서 도시된 숫자들은 상기 논의된 각도 인트라-예측 모드들에 대응하는 모드 인덱스들이다. 도 4 에 의해 도시된 모드 인덱스들에 추가하여, 평면형 모드는 0 의 인덱스를 가질 수도 있고, DC 모드는 1 의 인덱스를 가질 수도 있다. 상기 논의된 바와 같이 그리고 이 개시물의 하나 이상의 기술들에 따라서, 인트라 모드 인덱스에 기초하여 현재 프레임에서의 블록을 코딩하는 것과는 대조적으로, 모션 추정 유닛 (44) 은 모션 벡터에 의해 식별된 현재 프레임에서의 예측자 블록에 기초하여 현재 프레임에서의 블록을 예측할 수도 있다.

[0096] 비디오 인코더 (20) 는 코딩되고 있는 오리지날 비디오 블록으로부터 예측 프로세싱 유닛 (42) 으로부터의 예측 데이터를 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50) 는 이러한 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0097] 변환 프로세싱 유닛 (52) 은 잔차 블록에 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 유닛 (52) 은 DCT 와 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 웨이블릿 변환들, 정수 변환들, 서브 밴드 변환들 또는 다른 유형들의 변환들이 또한 이용될 수 있다. 어느 경우든지, 변환 프로세싱 유닛 (52) 은 잔차 블록에 변환을 적용하여 잔차 변환 계수들의 블록을 생성한다. 변환은 잔차 정보를 픽셀값 도메인으로부터 변환 도메인, 이를테면 주파수 도메인으로 변환할 수도 있다.

[0098] 변환 프로세싱 유닛 (52) 은 결과적인 변환 계수들을 양자화 프로세싱 유닛 (54) 에 전송할 수도 있다. 양자화 프로세싱 유닛 (54) 은 비트 레이트를 더욱 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 변환 계수들의 일부 또는 전부와 연관된 비트 깊이를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 일부 예들에서, 양자화 프로세싱 유닛 (54) 은 그 후 양자화된 변환 계수들을 포함하는 행렬의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56) 이 그 스캔을 수행할 수도 있다.

[0099] 양자화에 후속하여, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 이진 산술 코딩 (CABAC), 신택스-기반 컨텍스트 적응 이진 산술 코딩 (SBAC), 확률 구간 파티셔닝 엔트로피 (PIPE) 코딩, 또는 다른 엔트로피 인코딩 기법을 수행할 수도 있다. 컨텍스트 기반 엔트로피 코딩의 경우에, 컨텍스트는 이웃하는 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 코딩에 후속하여, 인코딩된 비트스트림은 다른 디바이스 (예를 들어, 비디오 디코더 (30)) 로 송신되거나 나중의 송신 또는 취출을 위해 아카이브될 수도 있다.

[0100] 역양자화 프로세싱 유닛 (58) 및 역변환 프로세싱 유닛 (60) 은, 예를 들어 참조 블록으로서 나중의 사용을 위해 픽셀 도메인에서 잔차 블록을 재구성하기 위해 각각 역양자화 및 역변환을 적용한다.

[0101] 모션 보상 유닛 (46) 은 모션 추정에서 사용하기 위한 서브-정수 (sub-integer) 픽셀 값들을 계산하기 위해 참조 블록에 하나 이상의 보간 필터들을 또한 적용할 수도 있다. 합산기 (62) 는 참조 픽처 메모리 (64) 에의 저장을 위해 재구성된 비디오 블록을 생성하기 위해 모션 보상 유닛 (46) 에 의해 생성된 모션 보상된 예측 블록에 재구성된 잔차 블록을 가산한다. 재구성된 비디오 블록은 후속하는 비디오 픽처에서의 블록을 인터-코딩하기 위해 참조 블록으로서 모션 추정 유닛 (44) 및 모션 보상 유닛 (46) 에 의해 사용될 수도 있다. 일부 예들에서, 현재 픽처를 예측하기 위한 참조 픽처로서 현재 픽처가 사용되는 경우 등에서, 모션 보상 유닛 (46) 및/또는 합산기 (62) 는 현재 픽처를 코딩하는 동안 규칙적인 간격들로 참조 픽처 메모리 (64) 에 의해 저장된 현재 픽처의 버전을 업데이트할 수도 있다. 하나의 예로서, 모션 보상 유닛 (46) 및/또는 합산기 (62) 는 현재 픽처의 각 블록을 코딩한 후에 참조 픽처 메모리 (64) 에 의해 저장된 현재 픽처의 버전을 업데이트할 수도 있다. 실례로, 현재 블록의 샘플들이 초기화된 값들로서 참조 픽처 메모리 (64) 에 저장되는 경우에, 모션 보상 유닛 (46) 및/또는 합산기 (62) 는 현재 블록에 대한 재구성된 샘플들로 참조 픽처 메모리 (64) 에 의해 저장된 현재 픽처의 현재의 샘플들을 업데이트할 수도 있다.

[0102] 필터링 유닛 (미도시) 은 다양한 필터링 프로세스들을 수행할 수도 있다. 예를 들어, 필터링 유닛은 디블록킹을 수행할 수도 있다. 즉, 필터링 유닛은 재구성된 비디오의 슬라이스 또는 프레임을 형성하는 복수의 재

구성된 비디오 블록들을 수신하고 슬라이스 또는 프레임으로부터 블록키니스 아티팩트들을 제거하기 위해 블록 경계들을 필터링할 수도 있다. 하나의 예에서, 필터링 유닛은 소위 비디오 블록의 "경계 강도 (boundary strength)" 를 평가한다. 비디오 블록의 경계 강도에 기초하여, 비디오 블록의 에지 픽셀들은, 하나의 비디오 블록으로부터의 천이 (transition) 가 뷰어 (viewer) 가 감지하기 보다 어렵게 되도록 인접 비디오 블록의 에지 픽셀들에 대해 필터링될 수도 있다.

[0103] 일부 예들에서, 모션 보상 유닛 (46) 및/또는 합산기 (62) 는, 필터링이 샘플들에 대한 필터링 (예컨대, 디블록킹 및/또는 SAO) 을 수행하기 전에 참조 픽처 메모리 (64) 에 의해 저장된 현재 픽처의 버전을 업데이트할 수도 있다. 실례로, 필터링 유닛은 필터링을 적용하기 전에 전체 픽처가 코딩될 때까지 기다릴 수도 있다. 이러한 방식으로, 모션 추정 유닛 (44) 은 필터링을 적용하기 전에 참조로서 현재 픽처를 이용할 수도 있다. 일부 예들에서, 필터링 유닛은 참조 픽처에 의해 저장된 현재 픽처의 버전이 업데이트됨에 따라 필터링을 수행할 수도 있다. 실례로, 필터링 유닛은 각 블록이 업데이트됨에 따라 필터링을 적용할 수도 있다. 이러한 방식으로, 모션 추정 유닛 (44) 은 필터링을 적용한 후에 참조로서 현재 픽처를 이용할 수도 있다.

[0104] 다수의 상이한 양태들 및 예들의 기법들이 본 개시에 기술되지만, 그 기법들의 여러 양태들 및 예들은 함께 또는 서로로부터 별개로 수행될 수도 있다. 즉, 그 기법들은 상술된 여러 양태들 및 예들에 엄격하게 제한되는 것이 아니라, 조합으로 사용되거나 함께 및/또는 별개로 수행될 수도 있다. 또한, 소정의 기법들이 (인트라 예측 유닛 유닛 (48), 모션 보상 유닛 (46), 또는 엔트로피 인코딩 유닛 (56) 과 같은) 비디오 인코더 (20) 의 소정의 유닛들로 돌려질 수도 있지만, 비디오 인코더 (20) 의 하나 이상의 다른 유닛들이 또한 그러한 기법들을 수행하는 것을 담당할 수도 있다는 것이 이해되어야 한다.

[0105] 이러한 방식으로, 비디오 인코더 (20) 는 이 개시에서 설명된 하나 이상의 예시적인 기술들을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 현재 프레임에서 포함된 예측자 블록을 이용하여 현재 프레임에서 비디오 데이터의 블록을 코딩하도록 구성될 수도 있다. 비디오 인코더 (20) 는, 예를 들어 인트라 BC 를 이용하여 현재 픽처의 하나 이상의 블록들을 코딩할 목적을 위해, VPS/SPS/PPS 를 참조하는 픽처가 픽처 그 자신의 참조 픽처 리스트에서 존재할 수도 있는지 여부를 나타내는 신택스 엘리먼트를 포함하는 비트스트림을 출력하도록 더 구성될 수도 있다. 즉, 블록이 인트라 BC 모드를 이용하여 코딩될 때, 비디오 인코더 (20) 는 (신택스 엘리먼트가 현재 픽처가 그 자체로 참조 픽처 리스트에 포함될 수 있는 것을 나타내는 것을 가정하면) 예를 들어 인덱스 값이 픽처 그 자신에 대응하도록 참조 픽처 리스트 내로의 인덱스 값을 이용하여, 블록에 대한 참조 픽처가 블록을 포함하는 픽처인 것을 시그널링할 수도 있다. 비디오 인코더 (20) 는 인트라 BC 모드를 이용하여 코딩되는 블록의 모션 정보에서 이 인덱스 값을 포함할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 의 하드웨어 아키텍처는 현재 픽처의 현재 블록을 예측하기 위해 참조 픽처로서 현재 픽처를 이용하기 위해 구체적으로 적응되지 않을 수도 있다.

[0106] 도 5 는 이 개시물에서 설명된 기술들을 구현할 수도 있는 비디오 디코더 (30) 의 일 예를 나타내는 블록도이다. 다시, 비디오 디코더 (30) 는 예시의 목적을 위해 HEVC 코딩의 맥락에서 설명될 것이지만, 다른 코딩 표준들에 관해서도 이 개시의 제한은 없다. 더욱이, 비디오 디코더 (30) 는 범위 확장에 따라서 기술들을 구현하도록 구성될 수도 있다.

[0107] 도 5 의 예에서, 비디오 디코더 (30) 는 비디오 데이터 메모리 (69), 엔트로피 디코딩 유닛 (70), 예측 프로세싱 유닛 (71), 역 양자화 프로세싱 유닛 (76), 역 변환 프로세싱 유닛 (78), 합산기 (80), 및 참조 픽처 메모리 (82) 를 포함할 수도 있다. 예측 프로세싱 유닛 (71) 은 모션 보상 유닛 (72) 및 인트라 예측 유닛 (74) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서, 도 3 으로부터의 비디오 인코더 (20) 에 대해 설명된 인코딩 패스에 대해 일반적으로 상호 역인 디코딩 패스를 수행할 수도 있다.

[0108] 비디오 데이터 메모리 (69) 는 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될, 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (69) 에 저장된 비디오 데이터는 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해 예를 들어 저장 디바이스 (34) 로부터, 예를 들어 카메라와 같은 로컬 비디오 소스로부터, 또는 물리적 데이터 저장 매체들에 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (69) 는 인코딩된 비디오 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (CPB) 를 형성할 수도 있다.

[0109] 참조 픽처 메모리 (82) 는 (예를 들어 인트라- 또는 인터-코딩 모드들에서) 비디오 디코더 (30) 에 의해 비디오 데이터를 디코딩하는데 사용하기 위한 참조 비디오 데이터를 저장하는 디코딩된 픽처 버퍼 (DPB) 의 하나의 예이다. 비디오 데이터 메모리 (69) 및 참조 픽처 메모리 (82) 는 동기식 DRAM (SDRAM), 자기저항식 RAM

(MRAM), 저항식 RAM (RRAM) 을 포함하는 동적 랜덤 액세스 메모리 (DRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 임의의 다양한 메모리 디바이스들에 의해 형성될 수도 있다. 비디오 데이터 메모리 (69) 및 참조 픽처 메모리 (82) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 여러 예들에서, 비디오 데이터 메모리 (69) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 함께 온-칩일 수도 있거나, 그들 컴포넌트들에 대해 오프-칩일 수도 있다.

[0110] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 표현하는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 비트스트림을 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들 또는 인트라 예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (70) 은 모션 보상 유닛 (72) 으로 모션 벡터들 및 다른 신택스 엘리먼트들을 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.

[0111] 일부 예들에서, 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩되는 경우, 인트라 예측 유닛 (74) 은 시그널링된 인트라 예측 모드 및 현재의 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 일부 예들에서, 비디오 픽처가 인터 코딩된 (즉, B 또는 P) 슬라이스로서 코딩되는 경우, 모션 보상 유닛 (72) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 그 예측 블록들은 참조 픽처 리스트 (RPL) 들 중 하나 내의 참조 픽처들 중 하나로부터 생성될 수도 있다. 예측 프로세싱 유닛 (71) 은 참조 픽처 메모리 (82) 에 저장된 참조 픽처들에 기초하여 구성 (construction) 기법들을 이용하여 RPL 들 (List0 및 List1) 을 구성 (construct) 할 수도 있다.

[0112] 이 개시물의 하나 이상의 기술들에 따라서, 참조 픽처드로서 다른 픽처들을 이용하도록 인터-예측을 제한하는 것과는 대조적으로, 비디오 디코더 (30) 는 현재 픽처에 포함된 비디오 데이터의 블록드을 예측하기 위해 참조 픽처로서 현재 픽처를 이용할 수도 있다. 예를 들어, 예측 프로세싱 유닛 (71) 은 예측 프로세싱 유닛 (71) 에 현재 픽처의 버전을 저장할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 고정된 값으로 초기화된 픽셀 값들로 현재 프레임의 초기화된 버전을 저장할 수도 있다. 일부 예들에서, 그 고정 값은 현재 픽처의 샘플들의 비트 심도에 기초할 수도 있다. 실례로, 고정 값은 1<<(비트심도 -1) 일 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 현재 픽처의 임의의 블록들을 인코딩하기 전에 현재 픽처의 초기화된 버전을 저장할 수도 있다. 현재 픽처의 초기화된 버전을 저장함으로써, 예측 프로세싱 유닛 (71) 은 아직 재구성되지 않은 예측적 블록들을 이용할 수도 있다. 반면, 예측 프로세싱 유닛 (71) 이 현재 픽처의 초기화된 버전을 저장하지 않는 경우에, 이미 재구성된 블록들만이 예측자 블록들로서 사용될 수도 있다 (즉, 디코더/인코더 미스매치를 회피하기 위해).

[0113] 상기 논의된 바와 같이, 예측 프로세싱 유닛 (72) 은 현재 픽처에 대한 하나 이상의 RPL 들을 발생시킬 수도 있다. 실례로, 예측 프로세싱 유닛 (71) 은 현재 픽처에 대해 현재 픽처 RPL 을 포함시킬 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (42) 은 RPL 내의 특정 위치에서 현재 픽처를 포함할 수도 있다. 하나의 예로서, 예측 프로세싱 유닛 (71) 은 장기 RPS 에서의 픽처들 전에 RPL 에서 현재 픽처를 삽입할 수도 있다. 예를 들어, 예측 프로세싱 유닛 (71) 은 장기 RPS 로부터의 픽처들의 인덱스 값들보다 더 적은 인덱스 값으로 RPL 에서 현재 픽처를 삽입할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 장기 RPS 에서의 픽처들 바로 전에 RPL 에서 현재 픽처를 삽입할 수도 있다.

[0114] 다른 예로서, 예측 프로세싱 유닛 (71) 은 장기 RPS 로부터의 픽처들을 삽입한 후에 RPL 에서 현재 픽처를 삽입할 수도 있다. 실례로, 예측 프로세싱 유닛 (71) 은 장기 RPS 에서의 픽처들의 인덱스 값들보다 더 큰 인덱스 값으로 RPL 에서 현재 픽처를 삽입할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 장기 RPS 로부터의 픽처들을 삽입한 바로 후에 RPL 에서 현재 픽처를 삽입할 수도 있다.

[0115] 다른 예로서, 예측 프로세싱 유닛 (71) 은 고정 위치에서 RPL 에 현재 픽처를 삽입할 수도 있다. 실례로, 예측 프로세싱 유닛 (71) 은 고정된 인덱스 값으로 RPL 에서 현재 픽처를 삽입할 수도 있다. 일부 예들에서, 고정된 인덱스 값은 -1, 또는 num_ref_idx_l1_active_minus1+1 일 수도 있다. 이러한 예들의 일부에서, 예측 프로세싱 유닛 (71) 은 현재 블록이 인트라 BC 를 이용하여 코딩되는 것을 나타내는 플래그 (즉, intra_bc_flag) 를 엔트로피 디코딩 유닛 (70) 으로부터 수신하지 않을 수도 있다.

[0116] 일부 예들에서, 모션 벡터들이 시간적 모션 벡터 예측 (TMVP) 을 이용하여 예측되는 경우 등에서, 예측 프로세싱 유닛 (71) 은 현재 픽처가 그 자신이 병치된 픽처로서 사용되지 않도록 하나 이상의 제약들을 적용할 수도

있다. 실례로, 예측 프로세싱 유닛 (71) 은, X 가 collocated_from_l0_flag 와 동일한 경우에 RefPicListX[collocated_ref_idx] 가 현재 픽처가 아니도록 TMVP 에 대해 사용되는 병치된 픽처의 참조 인덱스를 특정하는 신택스 엘리먼트 (예컨대, collocated_ref_idx) 를 엔트로피 디코딩 유닛 (70) 으로부터 수신할 수도 있다.

[0117] 상기 논의된 바와 같이, 비디오 디코더 (30) 는 예측적 블록에 기초하여 비디오 데이터의 현재 픽처의 비디오 데이터의 블록을 디코딩할 수도 있다. 이 개시물의 하나 이상의 기술들에 따라서, 모션 보상 유닛 (72) 은 현재 픽처의 현재 블록에 대한 예측적 블록으로서의 사용을 위해 현재 픽처에서 위치된 블록을 선택할 수도 있다. 특히, 예측 프로세싱 유닛 (71) 은, 현재 블록에 대해, 현재 픽처를 포함하는 RPL 을 구성할 수도 있고, 모션 보상 유닛 (72) 은 RPL 에서의 인덱스를 나타내는 현재 블록에 대한 모션 파라미터들을 수신할 수도 있다. 일부 예들에서, 인덱스는 RPL 에서 현재 픽처를 식별할 수도 있다. 이것이 발생할 때, 모션 보상 유닛 (72) 은 현재 블록에 대해 모션 벡터에 의해 식별된 위치에서 현재 픽처 그 자신으로부터 예측된 블록을 추출하기 위해 모션 파라미터들에 포함된 모션 벡터를 이용할 수도 있다. 이러한 방식으로, 모션 보상 유닛 (72) 은 인트라 BC 를 수행할 수도 있다.

[0118] 일부 예들에서, 예측 프로세싱 유닛 (71) 은, 현재 픽처를 예측하기 위해 현재 픽처가 RPL 에서 존재할 수도 있는지 여부를 나타내기 위한 하나 이상의 신택스 엘리먼트들을 엔트로피 디코딩 유닛 (70) 으로부터 수신할 수도 있다. 하나의 예로서, 예측 프로세싱 유닛 (71) 은 비디오 데이터의 픽처들이 그 자신들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는지 여부를 나타내는 단일 신택스 엘리먼트를 수신할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 현재 픽처에 의해 참조되는 비디오 파라미터 셋트 (VPS), 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS), 또는 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (PPS) 로부터 단일 신택스 엘리먼트를 수신할 수도 있다.

[0119] 또 다른 예로서, 예측 프로세싱 유닛 (71) 은, 엔트로피 인코딩 유닛 (56) 으로부터, 현재 픽처가 그 현재 픽처를 예측하기 위해 사용된 RPL 에서 존재할 수도 있는지 여부를 나타내기 위한 다수의 신택스 엘리먼트들을 수신할 수도 있다. 실례로, 예측 프로세싱 유닛 (71) 은 비디오 데이터의 픽처들이 그 자신들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는지 여부를 나타내는 제 1 신택스 엘리먼트를 수신할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 현재 픽처에 의해 참조되는 VPS, 현재 픽처에 의해 참조되는 SPS, 또는 현재 픽처에 의해 참조되는 PPS 로부터 제 1 신택스 엘리먼트를 수신할 수도 있다. 일부 예들에서, 신택스 엘리먼트가, 비디오 데이터의 픽처들이 그 자신들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는 것을 나타내는 경우 등에, 예측 프로세싱 유닛 (71) 은 비디오 데이터의 현재 픽처가 현재 슬라이스를 예측하기 위해 사용된 RPL 에서 존재할 수도 있는지 여부를 나타내는 제 2 신택스 엘리먼트를 수신할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 현재 슬라이스의 슬라이스 헤더로부터 제 2 신택스 엘리먼트를 수신할 수도 있다.

[0120] 일부 예들에서, 예측 프로세싱 유닛 (71) 은 블록이 인트라 블록 카피 (Intra BC) 를 이용하여 코딩되는지 여부를 나타내는 신택스 엘리먼트를 수신하지 않을 수도 있다. 실례로, 예측 프로세싱 유닛 (71) 은 이 개시물의 기술들에 따라서 인트라 BC 를 이용하여 예측되는 블록들의 코딩 유닛 신택스에서 intra_bc_flag 를 수신하지 않을 수도 있다.

[0121] 일부 예들에서, 인트라 BC 를 이용하여 B-슬라이스들 및 P-슬라이스들에서 블록들을 코딩하는 것에 추가하여, 예측 프로세싱 유닛 (71) 은 현재 픽처의 I-슬라이스의 블록들을 디코딩하기 위해 현재 픽처를 포함하는 하나 이상의 RPL 들을 구성할 수도 있다. 일부 이러한 예들에서, 하나 이상의 RPL 들은 오직 현재 픽처를 포함하는 것으로 가정될 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 현재 픽처가 현재 픽처에 포함된 I-슬라이스들의 참조 픽처로서 사용될 수도 있는지 여부를 나타내는 신택스 엘리먼트를 수신할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 현재 픽처에 의해 참조되는 VPS, 현재 픽처에 의해 참조되는 SPS, 또는 현재 픽처에 의해 참조되는 PPS, 또는 현재 I-슬라이스의 슬라이스 헤더로부터 신택스 엘리먼트를 수신할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 여전히 AMVP 및 병합의 일방 또는 양방을 이용할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은, 인코더로부터 I-슬라이스에 대한 AMVP 에 대한 타겟 참조 인덱스를 수신하지 않을 수도 있고, 고정 값, 예컨대, 0 으로서 타겟 참조 인덱스를 도출할 수도 있다.
일부 예들에서, 예측 프로세싱 유닛 (71) 은 I-슬라이스에 대해 AMVP 에 대한 타겟 참조 인덱스를 수신할 수도 있지만, 타겟 참조 인덱스의 값은 고정 값, 예컨대 0 으로 제약될 수도 있다.

[0122] 참조 픽처 메모리 (82) 에 의해 저장된 픽처들은 단기, 장기, 다른 마킹으로서 마킹될 수도 있고, 및/또는 마킹

되지 않을 수도 있다. 일부 예들에서, 현재 슬라이스가 I-슬라이스이고 인트라 BC 가 인에이블될 때 등의 경우에, 예측 프로세싱 유닛 (71) 은 현재 픽처를 장기 또는 단기 중 어느 일방으로서 마킹할 수도 있다. 일부 예들에서, 현재 슬라이스가 I-슬라이스이고 인트라 BC 가 인에이블될 때 등의 경우에, 예측 프로세싱 유닛 (71) 은 현재 픽처를 장기 또는 단기 중 어느 일방으로서 마킹하지 않을 수도 있다.

[0123] 일부 예들에서, 예측 프로세싱 유닛 (71) 은 현재 픽처를 코딩하기 전에 현재 픽처를 장기로서 마킹하고 현재 픽처를 코딩한 후에 현재 픽처를 단기로서 마킹할 수도 있다. 이러한 예들의 일부에서, 현재 슬라이스가 I-슬라이스인 경우 등에서, 예측 프로세싱 유닛 (71) 은, ("장기" 로서 마킹된) 인트라 BC 참조를 순전히 참조하는 후보들 또는 ("단기" 로서 마킹된) 인터 참조들을 참조하는 다른 후보들을 포함하도록 병합 후보 리스트를 생성할 수도 있다. 이러한 방식으로, 예측 프로세싱 유닛 (71) 은 인트라 BC 후보들 및 보통의 (인터-예측) 후보들 양자를 모두 포함하도록 후보 리스트를 생성할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 인트라 BC 참조 및 인터-예측 참조로부터 병합 후보를 양방향으로 예측할 수도 있다. 이러한 예들의 일부에서, 현재 슬라이스가 인터 코딩된 슬라이스 (예컨대, B-슬라이스 또는 P-슬라이스) 인 경우 등에서, 예측 프로세싱 유닛 (71) 은 현재 블록이 인트라 BC 로 코딩되는지 여부를 나타내는 플래그 (intra_bc_flag) 를 나타내는 추가적인 플래그를 수신하지 않을 수도 있다.

[0124] 일부 예들에서, 현재 프레임이 추가적인 참조 픽처라고 가정함으로써 인트라 BC 가 인에이블되고 시간적 모션 벡터 예측 (TMVP) 이 인에이블되는 경우 등에서, 예측 프로세싱 유닛 (71) 은 타겟 병합 참조 픽처가 RPL 에서 현재 픽처 또는 제 1 픽처인지 여부를 나타내는 신택스 엘리먼트 (예컨대, RefPicList0[0] 또는 RefPicList1[0] 중 어느 일방) 를 수신할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은, 참조들의 장기 (인트라 BC) 카테고리 또는 참조들의 단기 카테고리에 대한 타겟 인덱스를 도출 또는 수신하는 것, 및 병치된 블록의 참조 픽처의 카테고리에 기초하여 상이한 타겟 병합 인덱스를 적용하는 것에 의해 참조 픽처를 식별할 수도 있다. 실례로, 예측 프로세싱 유닛 (71) 은 참조들의 장기 (인트라 BC) 카테고리 또는 참조들의 단기 카테고리에 대한 타겟 인덱스를 시그널링 또는 도출하는 것에 의해 참조 픽처를 식별할 수도 있고, 병치된 블록의 참조 픽처의 카테고리에 기초하여 상이한 타겟 병합 인덱스가 적용된다.

[0125] 예측 프로세싱 유닛 (71) 은 비디오 데이터의 현재 블록과 비디오 데이터의 예측자 블록 사이의 변위를 나타내는 모션 벡터를 결정할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 인코딩된 비디오 비트스트림에서 수신된 하나 이상의 신택스 엘리먼트들에 기초하여 모션 벡터를 결정할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은 정수 정밀도로 모션 벡터를 결정할 수도 있다. 이러한 예들에서, 현재 픽처가 장기 참조 픽처로서 마킹되는 경우 등에서, 예측 프로세싱 유닛 (71) 은 현재 픽처를 예측하기 위해 보통의 장치 참조 픽처들 (즉, 현재 픽처가 아닌 장기 참조 픽처들) 을 이용하지 않을 수도 있다. 또한, 이러한 예들에서, 예측 프로세싱 유닛 (71) 은, 예를 들어 인트라 BC 모션 벡터가 공간적 및 시간적 이웃 블록들의 인트라 BC 모션 벡터들에만 기초하여 예측됨에 따라, 어떤 변경도 없이 어드밴스드 모션 벡터 예측 (AMVP) 또는 병합 관련 디코딩 프로세스를 HEVC 버전 1 에서와 같이 이용할 수도 있다.

[0126] 일부 예들에서, 예측 프로세싱 유닛 (71) 은 상이한 레벨들의 정밀도로 모션 벡터를 결정할 수도 있다. 실례로, 예측 프로세싱 유닛 (71) 은 정수 정밀도, 디폴트 정밀도, 또는 가장 미세한 모션 정밀도 (예컨대, HEVC 에서 1/4 pel 정밀도) 로 모션 벡터를 결정할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (71) 은, 예컨대, 현재 픽처에 의해 참조되는 SPS 또는 VPS 에서, 코딩된 인트라 BC 모션 벡터들의 정밀도를 나타내는 신택스 엘리먼트를 수신할 수도 있다. 일부 예들에서, 인트라 BC 모션 벡터들의 정밀도는 픽처 레벨에서 적응적일 수도 있고, 예측 프로세싱 유닛 (71) 은 예컨대, 현재 블록에 의해 참조되는 슬라이스 또는 PPS 에서, 코딩된 인트라 BC 모션 벡터들의 정밀도를 나타내는 신택스 엘리먼트를 수신할 수도 있다.

[0127] 일부 예들에서, 예측 프로세싱 유닛 (71) 은 인트라 BC 모션 벡터들의 정밀도 레벨에 대해 보상하기 위한 하나 이상의 동작들을 수행할 수도 있다. 하나의 예로서, 블록들을 참조 픽처 메모리 (82) 에 저장하기 전에, 예측 프로세싱 유닛 (71) 은, 가장 미세한 정밀도가 1/4-pel 일 때 2 만큼, 또는 좌측 시프트 후에 +/-2 등 라운딩의 임의의 다른 수단에 의해, 인트라 BC 로 코딩된 각 블록의 모션 벡터를 좌측 시프트할 수도 있다. 다른 예들로서, 정수의 정밀도를 갖는 인트라 BC 모션 벡터들로 현재 슬라이스를 코딩할 때, 예측 프로세싱 유닛 (71) 은 가장 미세한 정밀도가 1/4-pel 일 때 2 만큼 등에 의해 각 인트라 BC 코딩된 블록의 모션 벡터가 우측 시프트되는 방식으로 병치된 픽처를 프로세싱할 수도 있다. 일부 예들에서, 현재 슬라이스가 가장 미세한 정밀도를 갖는 인트라 BC 모션 벡터들로 코딩될 때 등의 경우에, 예측 프로세싱 유닛 (71) 은 모션 벡터 우측 시프트의 상기 프로세스를 적용하지 않을 수도 있다.

[0128] 일부 예들에서, 현재 픽처가 장기 참조 픽처로서 마킹되는 경우에, 예측 프로세싱 유닛 (71) 은 현재 픽처를 예측하기 위해 여전히 보통의 장기 참조 픽처들을 이용할 수도 있다. 보통의 장기 참조 픽처를 참조하는 모션 벡터 및 현재 픽처를 참조하는 모션 벡터가 병합 또는 AMVP 동안 서로로부터 예측하는 것을 방지하기 위해, 예측 프로세싱 유닛 (71) 은 병합 또는 AMVP 프로세스 동안 보통의 장기 참조 픽처와 현재 픽처를 구분할 수도 있다.

[0129] 모션 보상 유닛 (72) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱함으로써 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 디코딩되고 있는 현재의 비디오 블록에 대한 예측 블록들을 생성하기 위해 그 예측 정보를 사용한다. 예를 들어, 모션 보상 유닛 (72) 은 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용되는 예측 모드 (예를 들어, 인트라- 또는 인터-예측), 인터 예측 슬라이스 타입 (예를 들어, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 참조 픽처 리스트들의 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 현재의 비디오 슬라이스 내의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위해 수신된 신택스 엘리먼트들의 일부를 사용한다.

[0130] 모션 보상 유닛 (72) 은 또한 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (72) 은 참조 블록들의 서브-정수 픽셀들에 대한 보간된 값들을 계산하기 위해 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 사용된 바와 같은 보간 필터들을 사용할 수도 있다. 이러한 경우에, 모션 보상 유닛 (72) 은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 사용된 보간 필터들을 결정하고 예측 블록들을 생성하기 위해 그 보간 필터들을 사용할 수도 있다.

[0131] 역양자화 프로세싱 유닛 (76) 은 비트스트림에서 제공되고 엔트로피 디코딩 유닛 (70) 에 의해 디코딩된 양자화된 변환 계수들을 역양자화, 즉 탈양자화한다. 역양자화 프로세스는 양자화의 정도, 및 마찬가지로 적용되어야 하는 역양자화의 정도를 결정하기 위해 비디오 슬라이스 내의 각 비디오 블록에 대해 비디오 디코더 (30) 에 의해 계산된 양자화 파라미터 QP_Y 의 사용을 포함할 수도 있다.

[0132] 역변환 프로세싱 유닛 (78) 은 픽셀 도메인에서 잔차 블록들을 생성하기 위해 변환 계수들에 역변환, 예를 들어 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역변환 프로세스를 적용한다. 비디오 디코더 (30) 는 역변환 프로세싱 유닛 (78) 으로부터의 잔차 블록들을 모션 보상 유닛 (72) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (80) 는 이러한 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0133] 비디오 디코더 (30) 는, 일부 예들에서, 상기 설명된 비디오 인코더 (20) 의 필터링 유닛과 유사하게 구성될 수도 있는 필터링 유닛을 포함할 수도 있다. 예를 들어, 필터링 유닛은 인코딩된 비트스트림으로부터 비디오 데이터를 디코딩 및 재구성할 때 디블록킹, SAO, 또는 다른 필터링 동작들을 수행하도록 구성될 수도 있다.

[0134] 그 기법들의 다수의 상이한 양태들 및 예들이 본 개시에서 기술되지만, 그 기법들의 여러 양태들 및 예들은 함께 또는 서로로부터 별개로 수행될 수도 있다. 즉, 그 기법들은 상술된 여러 양태들 및 예들에 엄격하게 제한되지 않아야 하고, 조합으로 사용되거나 함께 및/또는 별개로 수행될 수도 있다. 또한, 소정의 기법들이 비디오 디코더 (30) 의 소정의 유닛들로 돌려질 수도 있지만, 비디오 디코더 (30) 의 하나 이상의 다른 유닛들이 또한 그러한 기법들을 수행하는 것을 담당할 수도 있다는 것이 이해되어야 한다.

[0135] 이러한 방식으로, 비디오 디코더 (30) 는 이 개시에서 설명된 하나 이상의 예시적인 기술들을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는, 예를 들어, 인트라 BC 모드를 이용하여 현재 픽처의 하나 이상의 블록들을 코딩할 목적을 위해, PPS 를 참조하는 픽처가 픽처 그 자신의 참조 픽처 리스트에서 존재할 수도 있는지 여부를 나타내는 신택스 엘리먼트를 포함하는 비트스트림을 수신하도록 구성될 수도 있다. 즉, 비디오 디코더 (30) 는 현재 픽처가 그 자신에 대한 참조 픽처 리스트에서 포함될 수 있는 것을 나타내는 신택스 엘리먼트에 대한 값을 디코딩할 수도 있다. 따라서, 블록이 인트라 BC 모드를 이용하여 코딩될 때, 비디오 디코더 (30) 는, 예를 들어, 인덱스 값이 픽처 그 자신에 대응하도록 참조 픽처 리스트 내로의 인덱스 값을 이용하여, 블록에 대한 참조 픽처가 블록을 포함하는 픽처인 것을 결정할 수도 있다. 비디오 디코더 (30) 는 인트라 BC 모드를 이용하여 코딩되는 블록의 모션 정보로부터 이 인덱스 값을 디코딩할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 의 하드웨어 아키텍처는 현재 픽처의 현재 블록을 예측하기 위해 참조 픽처로서 현재 픽처를 이용하기 위해 구체적으로 적응되지 않을 수도 있다.

[0136] 도 6 은 이 개시의 하나 이상의 기술들에 따른, 인트라 블록 카핑 프로세스의 일 예를 나타내는 도이다. 하

나의 예시적인 인트라-예측 프로세스에 따르면, 비디오 인코더 (20) 는, 예를 들어 비디오 데이터의 이전에 코딩된 및 재구성된 블록들의 셋트로부터, 예측자 비디오 블록을 선택할 수도 있다. 도 6 의 예에서, 재구성된 구역 (108) 은 이전에 코딩된 및 재구성된 비디오 블록들의 셋트를 포함한다. 재구성된 구역 (108) 에서의 블록들은 비디오 디코더 (30) 에 의해 디코딩된 그리고 재구성된 그리고 재구성된 구역 메모리 (92) 에 저장된 블록들, 또는 비디오 인코더 (20) 의 재구성 루프에서 디코딩된 그리고 재구성된 그리고 재구성된 구역 메모리 (64) 에서 저장된 블록들을 나타낼 수도 있다. 현재 블록 (102) 은 코딩될 비디오 데이터의 현재 블록을 나타낸다. 예측자 블록 (104) 은, 현재 블록 (102) 의 인트라 BC 예측을 위해 사용되는, 현재 블록 (102) 과 동일한 픽처에서의, 재구성된 비디오 블록을 나타낸다.

[0137] 예시적인 인트라-예측 프로세스에서, 비디오 인코더 (20) 는, 잔차 신호와 함께, 현재 블록 (102) 에 대한 예측자 브록 (104) 의 위치를 나타내는, 모션 벡터 (106) 를 결정하고 인코딩할 수도 있다. 실례로, 도 6 에 의해 예시된 바와 같이, 모션 벡터 (106) 는 현재 블록 (102) 의 상부-좌측 코너에 대한 예측자 블록 (104) 의 상부-좌측 코너의 위치를 나타낼 수도 있다. 상기 논의된 바와 같이, 모션 벡터 (106) 는 또한 오프셋 벡터, 변위 벡터, 또는 블록 벡터 (BV) 로서 지칭될 수도 있다. 비디오 디코더 (30) 는 현재 블록을 디코딩하기 위해 인코딩된 정보를 이용할 수도 있다.

[0138] 하나의 예시적인 예로서, 현재 픽처를 디코딩하기 전에, 비디오 디코더 (30) 는 현재 픽처의 재구성된 샘플들을 1<<(비트심도-1) 로 초기화할 수도 있다. 비디오 디코더 (30) 는 그 다음, 참조 픽처 메모리 (82) 와 같은 참조 픽처 버퍼에서 현재 픽처의 버전을 저장하고, 현재 픽처를 장기 참조로서 마킹할 수도 있다. 비디오 디코더 (30) 는 그 다음, 참조 픽처 리스트 (RPL) 에 현재 픽처를 포함하고, RPL 에서 현재 픽처에 대해 참조 인덱스 (예컨대, (참조 리스트 ListX 에서) IdxCur) 를 할당할 수도 있다.

[0139] 비디오 디코더 (30) 는, RPL 에 기초하여, 참조 픽처 메모리 (82) 에서 저장된 현재 픽처의 버전에 포함된 예측 블록에 기초하여 현재 픽처에서 비디오 데이터의 블록을 디코딩할 수도 있다. 달리 말하면, 현재 픽처의 블록을 디코딩할 때, 비디오 디코더 (30) 는 현재 픽처, 즉, (ListX 에서) 참조 인덱스 IdxCur 를 갖는 참조로부터 블록을 예측할 수도 있다. 이 경우에, 비디오 디코더 (30) 는 HEVC 버전 1 에서의 것과 동일한 MV 코딩 방법을 이용할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는, 예를 들어 비디오 디코더 (30) 가 고정 값을 갖는 현재 픽처를 초기화하였기 때문에, MV 의 값을 제한하지 않을 수도 있다. 비디오 디코더 (30) 는 (예컨대, 비디오 디코더가 블록을 디코딩하는 것을 마쳤기 때문에) 초기화된 값들을 대체하기 위해 현재 픽처 버퍼 (예컨대, 참조 픽처 메모리 (82)) 에 블록의 재구성된 샘플들을 기입할 수도 있다. 이 예에서, 비디오 디코더 (30) 는 블록을 디코딩한 후에 재구성된 샘플들에 디블록킹, SAO 또는 임의의 다른 필터링 동작을 적용하지 않는다. 달리 말하면, 비디오 디코더 (30) 는 디블록킹 및 SAO 를 적용하기 전에 참조로서 현재 픽처를 이용할 수도 있다.

[0140] 전체 픽처를 코딩한 후에, 비디오 디코더 (30) 는, HEVC 버전 1 에서 기술된 것들과 동일한 방식으로 디블록킹, SAO 및 픽처 마킹과 같은 다른 동작들을 적용할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 현재 픽처에서의 블록을 참조할 때 종래의 참조 픽처들과 동일한 MV 의 정밀도 (예컨대, 1/4-픽셀 정밀도) 를 유지할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 는 HEVC 버전 1 에서 정의된 보간 필터를 이용할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 (HEVC 버전 1 에서 정의된 보간 필터에 추가하여 또는 그 대신에) 쌍선형 보간 필터와 같은 다른 보간 필터들을 이용할 수도 있다. 일부 예들에서, MV 가 현재 픽처에서의 블록을 참조할 때 등의 경우에, 비디오 디코더 (30) 는 MV 의 정밀도를 정수-픽셀로 제한할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 상기 설명된 바와 같이현재 픽처를 수반하는 하나 이상의 참조 픽처 관리 기법들을 수행할 수도 있다.

[0141] 상기 논의된 바와 같이, 비디오 인코더 (20) 는 현재 픽처가 그 현재 픽처를 예측하기 위해 사용된 RPL 에서 존재할 수도 있는지 여부를 나타내기 위한 하나 이상의 신택스 엘리먼트들을 인코딩하고 그 신택스 엘리먼트를 시그널링할 수도 있다. 실례로, 비디오 디코더 (30) 는 PPS 를 참조하는 픽처가 픽처 그 자신의 참조 픽처 리스트에서 존재할 수도 있는지 여부를 비디오 디코더 (30) 에 대해 표시하기 위해 **curr_pic_as_ref_enabled_flag** 를 시그널링할 수도 있다. 하나의 예로서, 비디오 인코더 (20) 는 PPS 를 참조하는 픽처가 픽처 그 자신의 참조 픽처 리스트에서 존재할 수도 있는 것을 나타내기 위해 1 과 동일한 **curr_pic_as_ref_enabled_flag** 를 시그널링할 수도 있다. 다른 예로서, 비디오 인코더 (20) 는 PPS 를 참조하는 픽처가 픽처 그 자신의 참조 픽처 리스트에서 존재하지 않는 것을 나타내기 위해 0 과 동일한 **curr_pic_as_ref_enabled_flag** 를 시그널링할 수도 있다. 또 다른 예로서, 비디오 인코더 (20) 는 **curr_pic_as_ref_enabled_flag** 를 시그널링하지 않을 수도 있다. 일부 예들에서, 신택스 엘리먼트 **curr_pic_as_ref_enabled_flag** 가 존재하지 않을 때, 비디오 디

코더 (30) 는 0 과 동일할 **curr_pic_as_ref_enabled_flag** 의 값을 추론할 수도 있다 (즉, 명시적 표시 또는 시그널 없이 결정할 수도 있다). 일부 예들에서, 비디오 인코더 (20) 는 (curr_pic_as_ref_enabled_flag ? 1 : 0) 과 동일하게 변수 **NumAddRefPic** 를 설정할 수도 있다. 아래의 표에서, curr_pic_as_ref_enabled 는 이 신택스 엘리먼트가 픽처 파라미터 셋트에 부가되고 있는 것을 나타내기 위해 이탤릭체로 표시된다. 다른 예들에서, 이 신택스 엘리먼트는 다른 데이터 구조들, 예컨대, 시퀀스 파라미터 셋트 (SPS), 비디오 파라미터 셋트 (VPS), 슬라이스 헤더 등에 추가적으로 또는 대안적으로 부가될 수도 있다.

pic_parameter_set_rbsp( ) {	디스크립터
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
dependent_slice_segments_enabled_flag	u(1)
output_flag_present_flag	u(1)
num_extra_slice_header_bits	u(3)
sign_data_hiding_enabled_flag	u(1)
cabac_init_present_flag	u(1)
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
init_qp_minus26	se(v)
constrained_intra_pred_flag	u(1)
transform_skip_enabled_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
if(cu_qp_delta_enabled_flag)	
diff_cu_qp_delta_depth	ue(v)
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
weighted_pred_flag	u(1)
weighted_bipred_flag	u(1)
transquant_bypass_enabled_flag	u(1)
tiles_enabled_flag	u(1)
entropy_coding_sync_enabled_flag	u(1)
if(tiles_enabled_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
column_width_minus1[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height_minus1[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
pps_loop_filter_across_slices_enabled_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	

[0142]

}	
pps_scaling_list_data_present_flag	u(1)
if(pps_scaling_list_data_present_flag)	
scaling_list_data( )	
lists_modification_present_flag	u(1)
log2_parallel_merge_level_minus2	ue(v)
slice_segment_header_extension_present_flag	u(1)
pps_extension_present_flag	u(1)
if(pps_extension_present_flag) {	
for(i = 0; i < 1; i++)	
pps_extension_flag[i]	u(1)
pps_extension_7bits	u(7)
}	
if(pps_extension_flag[0]) {	
if(transform_skip_enabled_flag)	
log2_max_transform_skip_block_size_minus2	ue(v)
cross_component_prediction_enabled_flag	u(1)
chroma_qp_adjustment_enabled_flag	u(1)
if(chroma_qp_adjustment_enabled_flag) {	
diff_cu_chroma_qp_adjustment_depth	ue(v)
chroma_qp_adjustment_table_size_minus1	ue(v)
for(i = 0; i <= chroma_qp_adjustment_table_size_minus1; i++) {	
cb_qp_adjustment[i]	se(v)
cr_qp_adjustment[i]	se(v)
}	
}	
log2_sao_offset_scale_luma	ue(v)
log2_sao_offset_scale_chroma	ue(v)
curr_pic_as_ref_enabled_flag	*u(1)*
}	
if(pps_extension_7bits)	
while(more_rbsp_data( ))	
pps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

[0143]

[0144] 상기 논의된 바와 같이, 비디오 디코더 (30) 는 현재 픽처를 포함할 수도 있는 하나 이상의 참조 픽처 리스트들을 구성할 수도 있다. 실례로, 일부 예들에서, 비디오 디코더 (30) 는 각각의 인터 코딩된 슬라이스에 대해 디코딩 프로세스의 시작부에서 이하의 프로세스를 유발할 수도 있다. 일부 예들에서, P 슬라이스를 디코딩할 때 등의 경우에, 비디오 디코더 (30) 는 단일 참조 픽처 리스트 RefPicList0 를 구성할 수도 있다. 일부 예들에서, B 슬라이스를 디코딩할 때 등의 경우에, 비디오 디코더 (30) 는 RefPicList0 에 추가하여 제 2 독립적 참조 픽처 리스트 RefPicList1 를 추가로 구성할 수도 있다.

[0145] 각 슬라이스에 대한 디코딩 프로세스의 시작부에서, 비디오 디코더 (30) 는 다음과 같이 참조 픽처 리스트들 RefPicList0 그리고, B 슬라이스들에 대해, RefPicList1 을 도출할 수도 있다 (여기서, 이탤릭체의 텍스트는 표준의 현재 시맨틱들에 대한 추가들을 나타낸다):

[0146] 비디오 디코더 (30) 는 Max(num_ref_idx_l0_active_minus1 + 1, NumPocTotalCurr + *NumAddRefPic*) 와 동일하게 변수 NumRpsCurrTempList0 를 설정하고 리스트 RefPicListTemp0 를 다음과 같이 구성할 수도 있고, *여기서, currPic 는 현재 픽처이다*:

[0147] rIdx = 0

[0148] while(rIdx < NumRpsCurrTempList0) {

[0149] for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++, i++)

[0150] RefPicListTemp0[rIdx] = RefPicSetStCurrBefore[i]

[0151] *if(curr_pic_as_ref_enabled_flag)*

[0152] *RefPicListTemp0[rIdx] = currPic*

[0153] for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0; rIdx++, i++)

[0154] RefPicListTemp0[rIdx] = RefPicSetStCurrAfter[i]

[0155] for(i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++)

[0156] RefPicListTemp0[rIdx] = RefPicSetLtCurr[i]

[0157] }

[0158] 비디오 디코더 (30) 는 RefPicList0 를 다음과 같이 구성할 수도 있고, *여기서, currPic 는 현재 픽처이다*:

[0159] for(rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++)

[0160] RefPicList0[rIdx] = ref_pic_list_modification_flag_l0 ?

[0161] RefPicListTemp0[list_entry_l0[rIdx]] : RefPicListTemp0[rIdx]

[0162] 일부 예들에서, 슬라이스가 B 슬라이스일 때 등의 경우에, 비디오 디코더 (30) 는 변수 NumRpsCurrTempList1 를 Max(num_ref_idx_l1_active_minus1 + 1, NumPocTotalCurr + *NumAddRefPic*) 와 동일하게 설정하고 리스트 RefPicListTemp1 를 다음과 같이 구성할 수도 있다:

[0163] rIdx = 0

[0164] while(rIdx < NumRpsCurrTempList1) {

[0165] for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++, i++)

[0166] RefPicListTemp1[rIdx] = RefPicSetStCurrAfter[i]

[0167] *if(curr_pic_as_ref_enabled_flag)*

[0168] *RefPicListTemp0[rIdx] = currPic*

[0169] for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1; rIdx++, i++)

[0170] RefPicListTemp1[rIdx] = RefPicSetStCurrBefore[i]

[0171] for(i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++)

[0172] RefPicListTemp1[rIdx] = RefPicSetLtCurr[i]

[0173] }

[0174] 일부 예들에서, 슬라이스가 B 슬라이스일 때 등의 경우에, 비디오 디코더 (30) 는 리스트 RefPicList1 를 다음과 같이 구성할 수도 있다:

[0175] for(rIdx = 0; rIdx <= num_ref_idx_l1_active_minus1; rIdx++)

[0176] RefPicList1[rIdx] = ref_pic_list_modification_flag_l1 ?

[0177] RefPicListTemp1[list_entry_l1[rIdx]] : RefPicListTemp1[rIdx]

[0178] 도 7 은 본 개시의 하나 이상의 기술들에 따른, 동일한 픽처에 포함된 예측자 블록에 기초하여 픽처의 비디오 데이터의 블록을 인코딩하기 위한 비디오 인코더의 예시적인 동작들을 나타내는 흐름도이다. 도 7 의 기술들은 도 1 및 도 3 에 예시된 비디오 인코더 (20) 와 같은 하나 이상의 비디오 인코더들에 의해 수행될 수도 있다. 예시의 목적을 위해, 도 7 의 기술들은 비디오 인코더 (20) 의 맥락에서 설명되지만, 비디오 인코더 (20) 의 것과는 상이한 구성들을 갖는 비디오 인코더들이 도 7 의 기술들을 수행할 수도 있다.

[0179] 이 개시의 하나 이상의 기술들에 따라서, 비디오 인코더 (20) 는 비디오 데이터의 현재 픽처의 버전을 참조 픽처 버퍼에 저장할 수도 있다 (702). 실례로, 예측 프로세싱 유닛 (42) 은 참조 픽처 메모리 (64) 에 현재 픽처의 초기화된 버전을 저장할 수도 있다.

[0180] 비디오 인코더 (20) 는 현재 블록을 선택할 수도 있고 (704), 현재 블록에 대한 예측 모드를 결정할 수도 있다 (706). 비디오 인코더 (20) 는, 예컨대 BD-레이트를 이용하여 테스트될 수도 있는 다양한 상이한 모드들 중

에서 예측 모드를 선택할 수도 있고, 비디오 인코더 (20) 는 최선의 BD-레이트 성능을 산출하는 모드를 선택할 수도 있다. 도 7 의 예에서, 비디오 인코더 (20) 는 인트라 BC 를 이용하여 현재 블록을 인코딩하도록 결정할 수도 있다.

[0181] 비디오 인코더 (20) 는 현재 블록에 대한 예측자 블록을 결정할 수도 있다 (708). 실례로, 비디오 인코더 (20) 는 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 픽셀 차이의 관점에서 현재 블록에 밀접하게 매칭하는 것으로 발견되는 현재 블록에서의 블록으로서 예측 블록을 결정할 수도 있다.

[0182] 비디오 인코더 (20) 는 현재 픽처를 예측하기 위해 사용되는 참조 픽처 리스트 (RPL) 에 현재 픽처를 삽입할 수도 있다 (710). 일부 예들에서, 비디오 인코더 (20) 는 비디오 데이터의 픽처들이 그 자신들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는지 여부를 나타내는 신택스 엘리먼트 (예컨대, curr_pic_as_ref_enabled_flag) 를 인코딩할 수도 있다. 상기 논의된 바와 같이, 일부 예들에서, 비디오 인코더 (20) 는, 장기 RPS 에서 픽처들의 인덱스 값들보다 더 적은 인덱스 값, 장기 RPS 에서 픽처들의 인덱스 값들보다 더 큰 인덱스 값, 또는 고정된 인덱스 값으로 RPL 에서 현재 픽처를 삽일할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 현재 픽처를 예측하기 위해 사용된 RPL 이 오직 현재 픽처를 포함하도록 현재 픽처를 예측하기 위해 사용되는 RPL 을 구성할 수도 있다. 예를 들어, 현재 블록이 현재 인트라 슬라이스 (예컨대, I-슬라이스) 에 포함되는 경우에, 비디오 인코더 (20) 는, 현재 인트라 슬라이스의 블록들을 예측하기 위해 사용된 RPL 이 오직 현재 픽처를 포함하도록 현재 인트라 슬라이스의 블록들을 예측하도록 사용되는 RPL 을 구성할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는, 현재 픽처를 예측하기 위해 사용된 RPL 이 현재 픽처 및 비디오 데이터의 하나 이상의 다른 픽처들을 포함하도록 현재 픽처를 예측하기 위해 사용되는 RPL 을 구성할 수도 있다. 예를 들어, 현재 블록이 현재 인터 슬라이스 (예컨대, B-슬라이스 또는 P-슬라이스) 에 포함되는 경우에, 비디오 인코더 (20) 는, 현재 인트라 슬라이스의 블록들을 예측하기 위해 사용된 RPL 이 현재 픽처 및 비디오 데이터의 하나 이상의 다른 픽처들을 포함하도록 현재 인터 슬라이스의 블록들을 예측하도록 사용되는 RPL 을 구성할 수도 있다.

[0183] 비디오 인코더 (20) 는 RPL 에서의 현재 픽처의 인덱스를 결정할 수도 있다 (712). 실례로, 현재 픽처가 고정된 인덱스 값으로 RPL 내로 삽입되는 경우에, 비디오 인코더 (20) 는 RPL 에서 현재 픽처의 인덱스가 고정된 인덱스 값이라고 결정할 수도 있다.

[0184] 비디오 인코더 (20) 는 현재 블록에 대한 잔차 블록을 계산할 수도 있다 (714). 실례로, 합산기 (50) 는 잔차 블록을 계산하기 위해 결정된 예측자 블록의 샘플들로부터 현재 블록의 샘플들을 감산할 수도 있다.

[0185] 비디오 인코더 (20) 는 잔차 블록을 양자화 및 변환할 수도 있다 (716). 실례로, 비디오 인코더 (20) 의 변환 프로세싱 유닛 (52) 은 잔차 블록에 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한할 수도 있다. 비디오 인코더 (20) 의 양자화 프로세싱 유닛 (54) 은 비트 레이트를 추가적으로 감소시키기 위해 변환 계수들을 양자화할 수도 있다.

[0186] 비디오 인코더 (20) 는 예측 모드, 인덱스, 및 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다 (718). 실례로, 엔트로피 인코딩 유닛 (56) 은 현재 블록의 예측 모드, (RPL 에서 현재 블록의 인덱스일 수도 있는) 현재 블록에 대한 참조 픽처의 인덱스, 및 양자화된 변환 계수들을 인트로피 인코딩할 수도 있다. 이러한 방식으로, 비디오 인코더 (20) 는 인트라 BC 를 수행할 수도 있다.

[0187] 일부 예들에서, 비디오 인코더 (20) 는 도 7 에 예시된 순서로 예시적인 동작들을 수행할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 도 7 에 예시된 순서와는 다른 순서로 예시적인 동작들을 수행할 수도 있다. 실례로, 일부 예들에서, 비디오 인코더 (20) 는 현재 픽처로부터 현재 블록을 선택 (704) 하기 전에 현재 픽처를 예측하기 위해 사용된 RPL 에서 현재 픽처를 삽입 (710) 할 수도 있다. 예를 들어, 현재 블록이 현재 슬라이스에 포함되는 경우에, 비디오 인코더 (20) 는 현재 슬라이스의 임의의 블록들을 인코딩하기 전에 현재 픽처를 예측하기 위해 사용된 RPL 에서 현재 픽처를 삽입할 수도 있다.

[0188] 도 8 은 본 개시의 하나 이상의 기술들에 따른, 동일한 픽처에 포함된 예측자 블록에 기초하여 픽처의 비디오 데이터의 블록을 디코딩하기 위한 비디오 디코더의 예시적인 동작들을 나타내는 흐름도이다. 도 8 의 기술들은 도 1 및 도 5 에 예시된 비디오 디코더 (30) 와 같은 하나 이상의 비디오 디코더들에 의해 수행될 수도 있다. 예시의 목적을 위해, 도 8 의 기술들은 비디오 디코더 (30) 의 맥락 내에서 설명되지만, 비디오 디코더 (30) 의 것과는 상이한 구성들을 갖는 비디오 디코더들이 도 8 의 기술들을 수행할 수도 있다.

[0189] 이 개시의 하나 이상의 기술들에 따라서, 비디오 디코더 (30) 는 비디오 데이터의 현재 픽처의 버전을 참조 픽처 버퍼에 저장할 수도 있다 (802). 실례로, 예측 프로세싱 유닛 (71) 은 참조 픽처 메모리 (82) 에 현재 픽처의 초기화된 버전을 저장할 수도 있다.

[0190] 비디오 디코더 (30) 는 현재 픽처로부터 현재 블록을 선택할 수도 있고 (804), 현재 블록에 대한 예측 모드, 참조 픽처의 인덱스, 양자화된 변환 계수들을 엔트로피 디코딩할 수도 있다 (806). 실례로, 엔트로피 디코딩 유닛 (70) 은 현재 블록에 대한 예측 모드가 인트라 BC 라는 것을 나타내는 하나 이상의 신택스 엘리먼트들 및 현재 블록을 예측하기 위해 사용된 참조 픽처의 인덱스 (예컨대, ref_idx_lX) 를 엔트로피 디코딩할 수도 있다.
도 8 의 예에서, 현재 블록을 예측하기 위해 사용된 참조 픽처는 현재 픽처일 수도 있다.

[0191] 비디오 디코더 (30) 는 현재 픽처를 예측하기 위해 사용된 참조 픽처 리스트 (RPL) 에 현재 픽처를 삽입할 수도 있다 (808). 일부 예들에서, 비디오 디코더 (30) 는 하나 이상의 신택스 엘리먼트들의 존재/값에 기초하여 RPL 에서 현재 픽처를 삽입할지 여부를 결정할 수도 있다. 실례로, 비디오 디코더 (30) 는 비디오 데이터의 픽처들이 그 자신들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는지 여부를 나타내는 신택스 엘리먼트의 값 (예컨대, **curr_pic_as_ref_enabled_flag**) 에 기초하여 RPL 에서 현재 픽처를 삽입할지 여부를 결정할 수도 있다. 상기 논의된 바와 같이, 일부 예들에서, 비디오 디코더 (30) 는 장기 RPS 에서 픽처들의 인덱스 값들보다 더 적은 인덱스 값, 장기 RPS 에서 픽처들의 인덱스 값들보다 더 큰 인덱스 값, 또는 고정된 인덱스 값으로 RPL 에서 현재 픽처를 삽일할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 현재 픽처를 예측하기 위해 사용된 RPL 이 오직 현재 픽처를 포함하도록 현재 픽처를 예측하기 위해 사용되는 RPL 을 구성할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는, 현재 픽처를 예측하기 위해 사용된 RPL 이 현재 픽처 및 비디오 데이터의 하나 이상의 다른 픽처들을 포함하도록 현재 픽처를 예측하기 위해 사용되는 RPL 을 구성할 수도 있다.

[0192] 비디오 디코더 (30) 는 현재 블록에 대한 예측자 블록을 결정할 수도 있다 (810). 실례로, 예측 프로세싱 유닛 (71) 은, RPL 에서 현재 픽처의 인덱스를 참조할 수도 있는 참조 픽처의 인덱스, 및 현재 블록과 예측자 블록 사이의 변위를 나타내는 모션 벡터에 기초하여 현재 블록의 예측자 블록을 결정할 수도 있다.

[0193] 비디오 디코더 (30) 는 잔차 블록을 역-양자화 및 역-변환할 수도 있다 (812). 실례로, 역 양자화 프로세싱 유닛 (76) 은 엔트로피 디코딩 유닛 (70) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉, 양자화해제할 수도 있다. 역 변환 프로세싱 유닛 (78) 은 픽셀 도메인에서의 잔차 블록들을 생성하기 위해 변환 계수들에 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 적용할 수도 있다.

[0194] 비디오 디코더 (30) 는 현재 블록을 재구성할 수도 있다 (814). 실례로, 합산기 (80) 는 현재 블록을 재구성하기 위해 예측자 블록에 잔차 블록을 가산할 수도 있다. 비디오 디코더 (30) 는, 현재 블록을 재구성한 후에, 코딩된 현재 블록을 포함한 현재 블록의 업데이트된 버전으로 참조 픽처 버퍼에서 현재 픽처의 버전을 업데이트할 수도 있다. 실례로, 합산기 (80) 는, 예컨대, 후속 블록으로 하여금 예측 블록의 일부 또는 전부로서 현재 블록의 하나 이상의 재구성된 샘플들을 사용하는 것을 가능하게 하도록, 참조 픽처 메모리 (82) 에 현재 블록의 재구성된 샘플들을 저장할 수도 있다. 이러한 방식으로, 비디오 디코더 (30) 는 인트라 BC 를 수행할 수도 있다.

[0195] 일부 예들에서, 비디오 디코더 (30) 는 도 8 에 예시된 순서로 예시적인 동작들을 수행할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 도 8 에 예시된 순서와는 다른 순서로 예시적인 동작들을 수행할 수도 있다.
실례로, 일부 예들에서, 비디오 디코더 (30) 는 현재 픽처로부터 현재 블록을 선택 (804) 하기 전에 현재 픽처를 예측하기 위해 사용된 RPL 에서 현재 픽처를 삽입 (808) 할 수도 있다. 예를 들어, 현재 블록이 현재 슬라이스에 포함되는 경우에, 비디오 디코더 (30) 는 현재 슬라이스의 임의의 블록들을 인코딩하기 전에 현재 픽처를 예측하기 위해 사용된 RPL 에서 현재 픽처를 삽입할 수도 있다.

[0196] 이하의 넘버링된 예들은 본 개시의 하나 이상의 양태들을 예시할 수도 있다:

[0197] 1. 비디오 데이터를 인코딩 또는 디코딩하는 방법으로서, 이 방법은, 비디오 코더에 의해 그리고 참조 픽처 버퍼에서, 비디오 데이터의 현재 픽처의 버전을 저장하는 단계; 현재 픽처의 블록들의 예측 동안 사용된 참조 픽처 리스트 (RPL) 에서 현재 픽처의 표시를 삽입하는 단계; 및 비디오 코더에 의해 그리고 RPL 에 기초하여, 참조 픽처 버퍼에 저장된 현재 픽처의 버전에 포함된 비디오 데이터의 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 제 1 블록을 코딩하는 단계를 포함한다.

[0198] 2. 제 1 항의 방법은, 비디오 코더에 의해 그리고 제 1 블록의 코딩 후에, 코딩된 제 1 블록을 포함하는 현

재 픽처의 업데이트된 버전으로 참조 픽처 버퍼에서의 현재 픽처의 버전을 업데이트하는 단계; 및 비디오 코더에 의해 그리고 RPL 에 기초하여, 참조 픽처 버퍼에 저장된 현재 픽처의 업데이트된 버전에 포함된 예측자 블록에 기초하여 현재 픽처에서의 비디오 데이터의 제 2 블록을 코딩하는 단계를 더 포함한다.

[0199] 3. 제 1 항의 방법은, 비디오 코더에 의해, 비디오 데이터의 픽처들이 그들 자신 픽처들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는지 여부를 나타내는 신택스 엘리먼트를 코딩하는 단계; 및 신택스 엘리먼트에 기초하여 현재 픽처를 예측하기 위해 사용된 RPL 에서 비디오 데이터의 현재 픽처를 포함하도록 결정하는 단계를 더 포함한다.

[0200] 4. 제 3 항의 방법에 있어서, 신택스 엘리먼트를 코딩하는 단계는, 현재 픽처에 의해 참조되는 비디오 파라미터 셋트 (VPS), 현재 픽처에 의해 참조되는 시퀀스 파라미터 셋트 (SPS), 또는 현재 픽처에 의해 참조되는 픽처 파라미터 셋트 (PPS) 에서 신택스 엘리먼트를 코딩하는 단계를 포함한다.

[0201] 5. 제 4 항에 있어서, 신택스 엘리먼트는 제 1 신택스 엘리먼트이고, 제 1 블록은 현재 픽처의 현재 슬라이스에 포함되며, 상기 방법은, 비디오 코더에 의해 그리고 비디오 데이터의 픽처들이 픽처들 그 자신들을 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는 것을 나타내는 제 1 신택스 엘리먼트, 비디오 데이터의 현재 픽처가 현재 슬라이스를 예측하기 위해 사용된 RPL 들에서 존재할 수도 있는지 여부를 나타내는 제 2 신택스 엘리먼트를 코딩하는 단계를 더 포함하며, 제 2 신택스 엘리먼트는 현재 슬라이스의 헤더에 포함되고, 현재 픽처를 예측하기 위해 사용된 RPL 에 비디오 데이터의 현재 픽처를 삽입하기 위한 결정은 제 2 신택스 엘리먼트에 더 기초한다.

[0202] 6. 제 5 항에 있어서, 제 2 신택스 엘리먼트를 코딩하는 단계는, 현재 슬라이스를 예측하기 위해 사용된 RPL 을 수정하는 것에 관련된 다른 신택스 엘리먼트들 전에 현재 스라이스의 헤더에 제 2 신택스 엘리먼트를 코딩하는 단계를 포함한다.

[0203] 7. 제 4 항에 있어서, 상기 방법은, 제 1 블록이 인트라 블록 카피 (인트라 BC) 를 이용하여 코딩되는지 여부를 나타내는 신택스 엘리먼트를 코딩하는 것을 포함하지 않는다.

[0204] 8. 제 1 항에 있어서, 블록은 현재 픽처의 현재 슬라이스에 포함되고, 현재 슬라이스에 대한 병치된 참조 인덱스를 나타내는 신택스 엘리먼트는 현재 픽처가 아닌 픽처를 나타낸다.

[0205] 9. 제 1 항에 있어서, 블록은 현재 픽처의 현재 슬라이스에 포함되고, 예측자 블록은 현재 슬라이스에 포함된다.

[0206] 10. 제 1 항에 있어서, RPL 에서의 각각의 엔트리는 인덱스 값을 가지며, 현재 픽처를 예측하기 위해 사용된 RPL 에서 현재 픽처를 삽입하는 것은 적어도, 장기 RPS 에서의 픽처들의 인덱스 값들보다 적은 인덱스 값으로 RPL 에서 현재 픽처를 삽입하는 것; 장기 RPS 에서의 픽처들의 인덱스 값들보다 큰 인덱스 값으로 RPL 에서 현재 픽처를 삽입하는 것; 또는 고정된 인덱스 값으로 RPL 에서 현재 픽처를 삽입하는 것에 의해, 하나 이상의 참조 픽처 셋트 (RPS) 들에 기초하여 현재 픽처를 예측하기 위해 사용된 RPL 을 구성하는 것을 포함한다.

[0207] 11. 제 1 항에 있어서, 블록은 현재 픽처의 현재 슬라이스에 포함되고, 현재 슬라이스는 인트라 슬라이스이며, 현재 픽처를 예측하기 위해 사용된 RPL 에서 현재 픽처를 삽입하는 것은, 현재 픽처를 예측하기 위해 사용된 RPL 이 현재 픽처를 오직 포함하도록 현재 픽처를 예측하기 위해 사용된 RPL 을 구성하는 것을 포함한다.

[0208] 12. 제 1 항에 있어서, 블록은 현재 픽처의 현재 슬라이스에 포함되고, 현재 슬라이스는 인터 슬라이스이며, 현재 픽처를 예측하기 위해 사용된 RPL 에서 현재 픽처를 삽입하는 것은, 현재 픽처를 예측하기 위해 사용된 RPL 이 현재 픽처 및 비디오 데이터의 하나 이상의 다른 픽처들을 포함하도록 현재 픽처를 예측하기 위해 사용된 RPL 을 구성하는 것을 포함한다.

[0209] 13. 제 1 항의 방법은, 비디오 코더에 의해 그리고 현재 픽처의 블록을 코딩하기 전에, 비디오 데이터의 현재 픽처를 장기 참조 픽처로서 마킹하는 단계; 및 비디오 코더에 의해 그리고 현재 픽처의 블록을 코딩한 후에, 비디오 데이터의 현재 픽처를 단기 참조 픽처로서 마킹하는 단계를 더 포함한다.

[0210] 14. 제 1 항에 있어서, 블록을 코딩하는 것은 블록을 인코딩하는 것을 포함하고, 상기 방법은, 비디오 데이터의 블록과 비디오 데이터의 예측자 블록 사이의 변위를 나타내는 벡터의 표현을 인코딩된 비디오 비트스트림에서 인코딩하는 단계를 더 포함한다.

[0211] 15. 제 1 항에 있어서, 블록을 코딩하는 것은 블록을 디코딩하는 것을 포함하고, 상기 방법은, 비디오 데이

터의 블록과 비디오 데이터의 예측자 블록 사이의 변위를 나타내는 벡터를 인코딩된 비디오 비트스트림에 기초하여 결정하는 단계를 더 포함한다.

[0212] 16. 제 1 항에 있어서, 참조 픽처 버퍼에서 현재 픽처의 버전을 저장하는 단계는, 비디오 코더에 의해 그리고 참조 픽처 버퍼에서, 고정된 값으로 초기화된 샘플 픽셀 값들로 현재 픽처의 버전을 저장하는 단계를 포함한다.

[0213] 17. 비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스로서, 상기 디바이스는, 비디오 데이터의 하나 이상의 픽처들을 저장하도록 구성된 참조 픽처 버퍼; 및 제 1 항 내지 제 16 항의 예들의 임의의 조합의 방법을 수행하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0214] 18. 비디오 데이터를 인코딩 또는 디코딩하기 위한 디바이스로서, 상기 디바이스는, 제 1 항 내지 제 16 항의 예들의 임의의 조합의 방법을 수행하기 위한 수단을 포함한다.

[0215] 19. 명령들을 저장한 컴퓨터 판독가능 저장 매체로서, 상기 명령들은, 실행될 때, 비디오 코더의 하나 이상의 프로세서들로 하여금, 제 1 항 내지 제 16 항의 예들의 임의의 조합의 방법을 수행하게 한다.

[0216] 이 개시의 어떤 양태들은 예시의 목적을 위해 HEVC 의 표준에 대해 설명되었다. 하지만, 이 개시물에서 설명된 기술들은, 현재 개발 중인 H.266 비디오 코딩 표준과 같은 아직 개발되지 않은 특허의 비디오 코딩 프로세스들 또는 다른 표준을 포함하는 다른 비디오 코딩 프로세스들에 대해 유용할 수도 있다.

[0217] 이 개시물에서 설명된 바와 같은 비디오 코더는 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 유사하게, 비디오 코딩 유닛은 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 마찬가지로, 비디오 코딩은 적용가능하면 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.

[0218] 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 실시에 필수적인 것은 아닌) 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.

[0219] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다.

[0220] 이런 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0221] 비제한적인 예로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 이를테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다.

[0222] 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비일시적, 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc) 는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들

은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들로써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

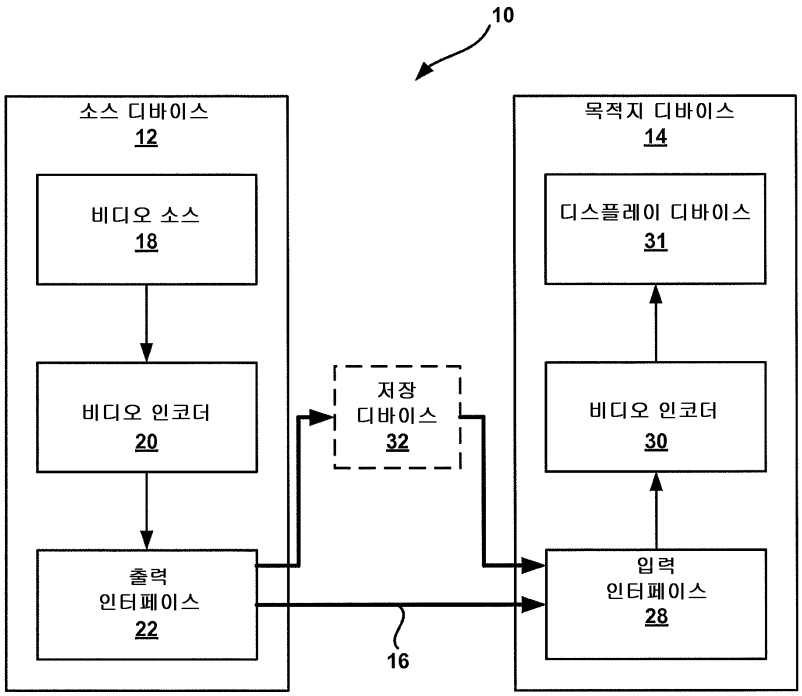
[0223] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 "프로세서"라는 용어는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 나타낼 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0224] 본 개시물의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC들의 셋트 (예컨대, 칩 셋) 를 포함하는 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

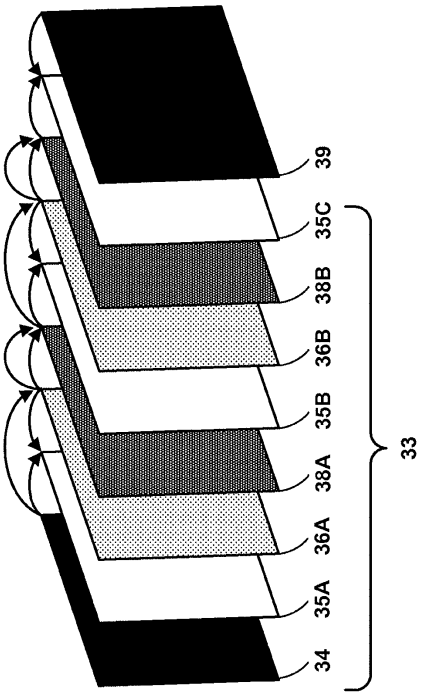
[0225] 다양한 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

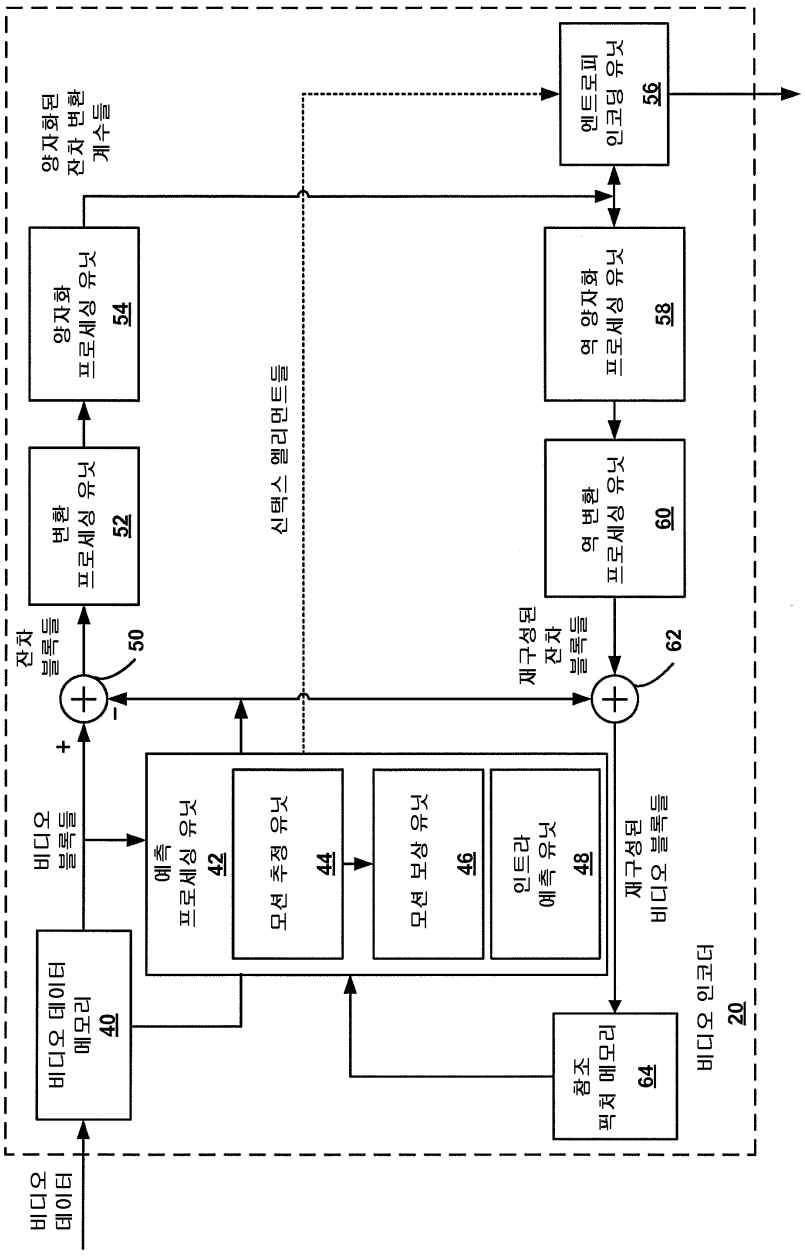
도면1



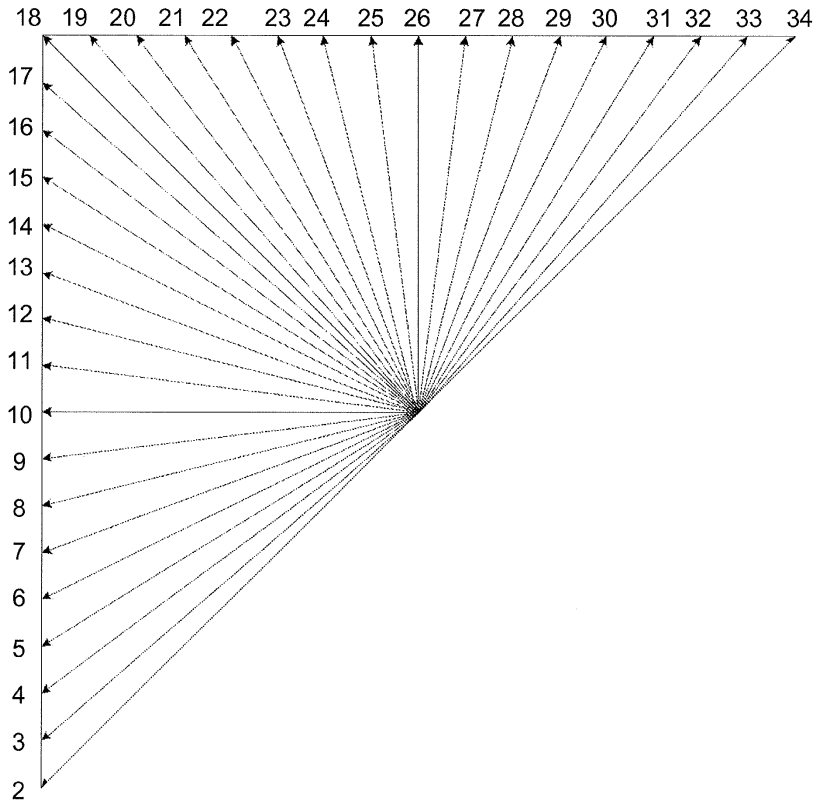
도면2



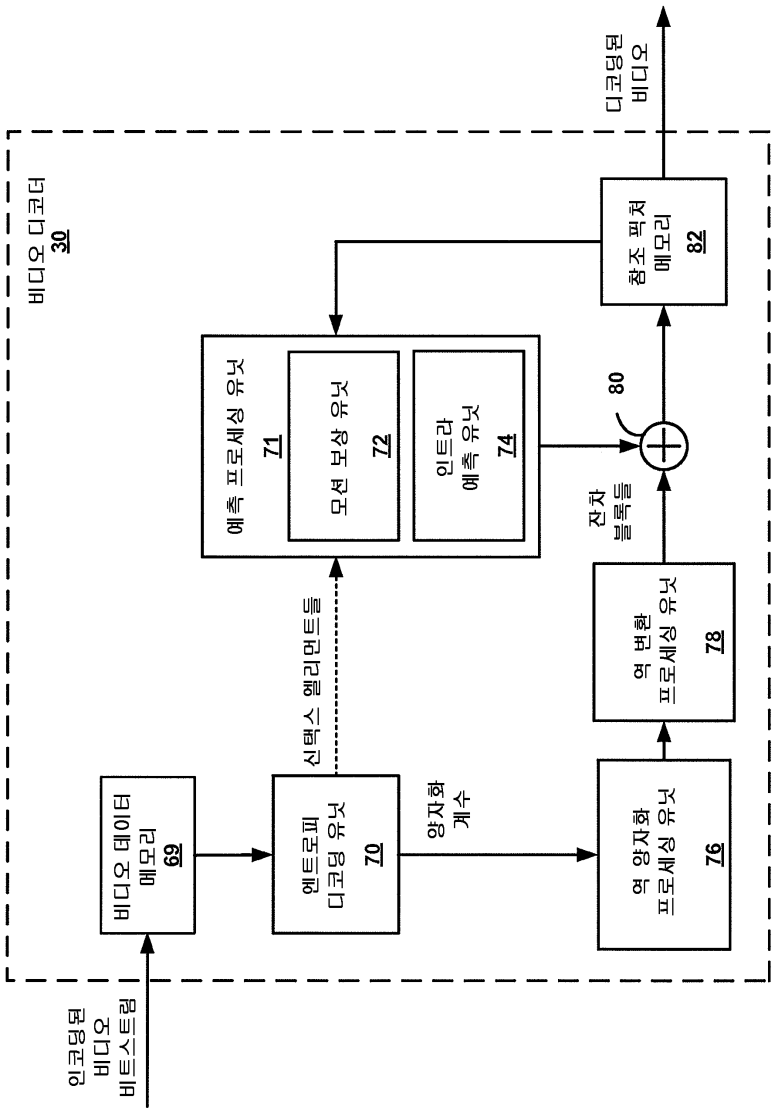
도면3


비디오 데이터
비디오 데이터 메모리 40
비디오 블록들
+
-
50
잔차 블록들
변환 프로세싱 유닛 52
양자화 프로세싱 유닛 54
양자화된 잔차 변환 계수들
신택스 엘리먼트들
엔트로피 인코딩 유닛 56
역 양자화 프로세싱 유닛 58
역 변환 프로세싱 유닛 60
재구성된 잔차 블록들
62
예측 프로세싱 유닛 42
모션 추정 유닛 44
모션 보상 유닛 46
인트라 예측 유닛 48
재구성된 비디오 블록들
참조 픽처 메모리 64
비디오 인코더 20

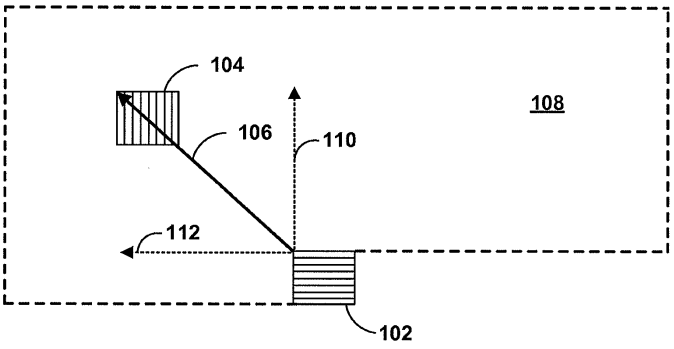
도면4


18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2

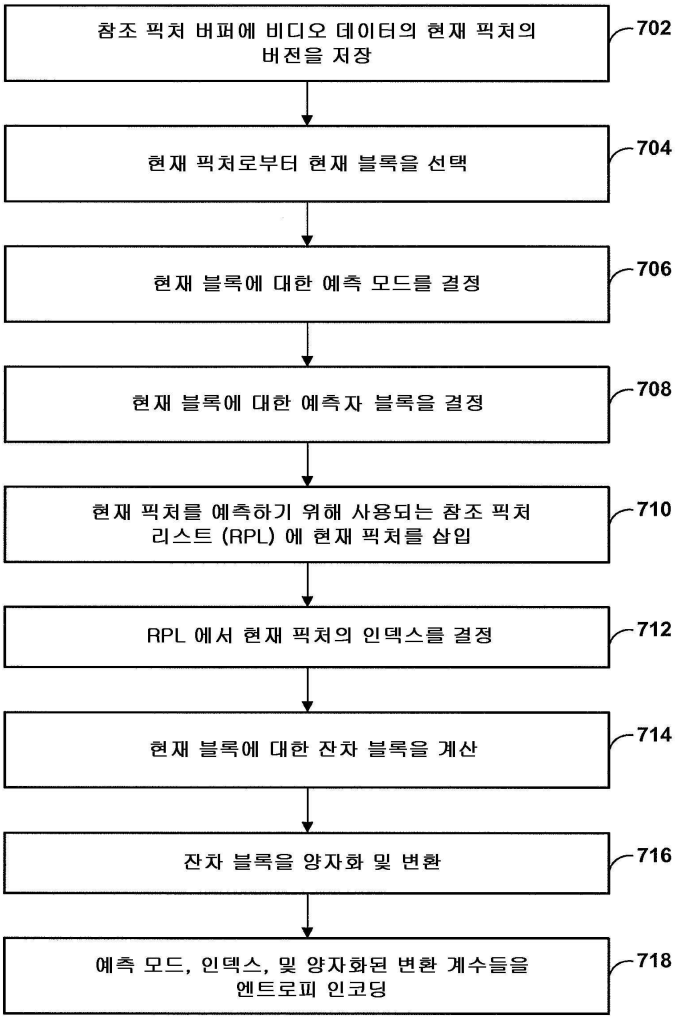
도면5


디코딩된 비디오
비디오 디코더 30
참조 픽처 메모리 82
예측 프로세싱 유닛 71
모션 보상 유닛 72
인트라 예측 유닛 74
80
잔차 블록들
역 변환 프로세싱 유닛 78
신택스 엘리먼트들
비디오 데이터 메모리 69
엔트로피 디코딩 유닛 70
양자화 계수
역 양자화 프로세싱 유닛 76
인코딩된 비디오 비트스트림

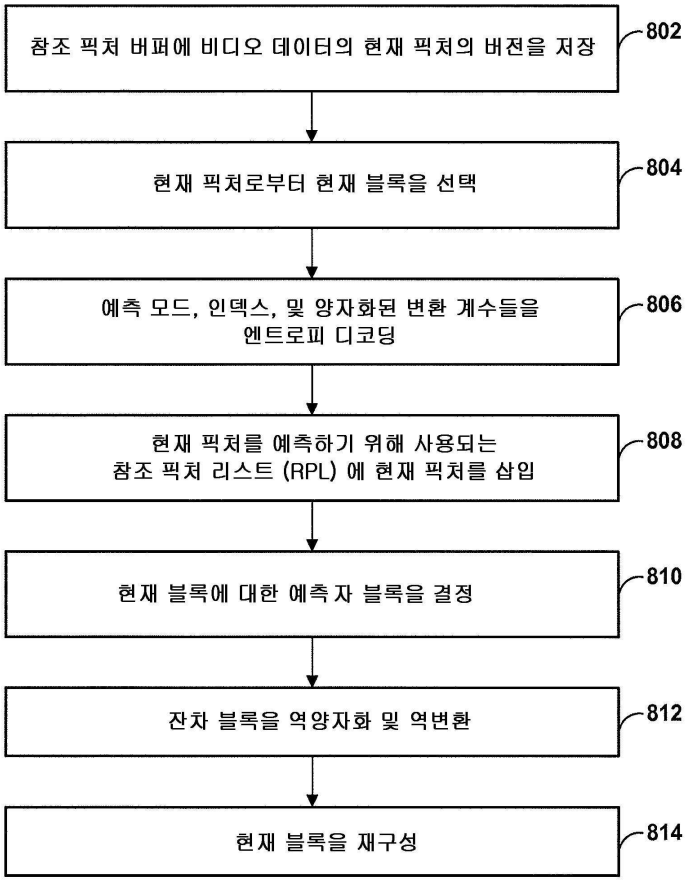
도면6


104
106
110
108
112
102

도면7


참조 픽처 버퍼에 비디오 데이터의 현재 픽처의 버전을 저장
702
현재 픽처로부터 현재 블록을 선택
704
현재 블록에 대한 예측 모드를 결정
706
현재 블록에 대한 예측자 블록을 결정
708
현재 픽처를 예측하기 위해 사용되는 참조 픽처 리스트 (RPL) 에 현재 픽처를 삽입
710
RPL 에서 현재 픽처의 인덱스를 결정
712
현재 블록에 대한 잔차 블록을 계산
714
잔차 블록을 양자화 및 변환
716
예측 모드, 인덱스, 및 양자화된 변환 계수들을 엔트로피 인코딩
718

도면8


참조 픽처 버퍼에 비디오 데이터의 현재 픽처의 버전을 저장
802
현재 픽처로부터 현재 블록을 선택
804
예측 모드, 인덱스, 및 양자화된 변환 계수들을 엔트로피 디코딩
806
현재 픽처를 예측하기 위해 사용되는 참조 픽처 리스트 (RPL) 에 현재 픽처를 삽입
808
현재 블록에 대한 예측자 블록을 결정
810
잔차 블록을 역양자화 및 역변환
812
현재 블록을 재구성
814