



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0133970
(43) 공개일자 2021년11월08일

- | | |
|--|---|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>HO4N 19/44</i> (2014.01) <i>HO4N 19/105</i> (2014.01)
 <i>HO4N 19/117</i> (2014.01) <i>HO4N 19/132</i> (2014.01)
 <i>HO4N 19/157</i> (2014.01) <i>HO4N 19/159</i> (2014.01)
 <i>HO4N 19/176</i> (2014.01) <i>HO4N 19/186</i> (2014.01)
 <i>HO4N 19/82</i> (2014.01)</p> <p>(52) CPC특허분류
 <i>HO4N 19/45</i> (2015.01)
 <i>HO4N 19/105</i> (2015.01)</p> <p>(21) 출원번호 10-2021-7028099
 (22) 출원일자(국제) 2020년03월06일
 심사청구일자 없음
 (85) 번역문제출일자 2021년09월01일
 (86) 국제출원번호 PCT/US2020/021544
 (87) 국제공개번호 WO 2020/185619
 국제공개일자 2020년09월17일</p> <p>(30) 우선권주장
 62/815,936 2019년03월08일 미국(US)
 (뒷면에 계속)</p> | <p>(71) 출원인
 헬컴 인코포레이티드
 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775</p> <p>(72) 발명자
 판 더 아우베라 게르트
 미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
 레이 밥파디트야
 미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
 (뒷면에 계속)</p> <p>(74) 대리인
 특허법인코리아나</p> |
|--|---|

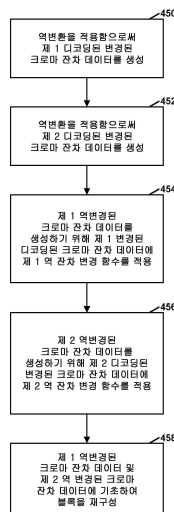
전체 청구항 수 : 총 41 항

(54) 발명의 명칭 비디오 코딩에서의 결합된 잔차 코딩

(57) 요약

비디오 데이터를 디코딩하는 디바이스는 비디오 데이터의 현재 블록에 대한 인터 예측 블록 및 인트라 예측 블록을 생성하고; 생성될 예측 블록의 각각의 샘플에 대해; 예측 블록에서의 샘플의 위치에 따라 샘플에 대한 제 1 가중치를 결정하고; 예측 블록에서의 샘플의 위치에 따라 샘플에 대한 제 2 가중치를 결정하고; 가중된 인터 예측 샘플을 생성하기 위해 인터 예측 블록에서의 위치의 샘플에 제 1 가중치를 적용하고; 가중된 인트라 예측 샘플을 생성하기 위해 인트라 예측 블록에서의 위치의 샘플에 제 2 가중치를 적용하고; 및 가중된 인터 예측 샘플 및 가중된 인트라 예측 샘플을 사용하여 예측 블록에서의 위치의 샘플의 값을 계산하고; 및 예측 블록을 사용하여 현재 블록을 코딩하도록 구성된 프로세서를 포함한다.

대표도 - 도11



(52) CPC특허분류

HO4N 19/117 (2015.01)
HO4N 19/132 (2015.01)
HO4N 19/157 (2015.01)
HO4N 19/159 (2015.01)
HO4N 19/176 (2015.01)
HO4N 19/186 (2015.01)
HO4N 19/82 (2015.01)

(30) 우선권주장

62/866,450 2019년06월25일 미국(US)
16/810,680 2020년03월05일 미국(US)

(72) 발명자

라마수브라모니안 아다르쉬 크리쉬난

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

코반 무하메드 제이드

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

팜 반 루옹

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터의 블록의 변환 계수들의 제 1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성하는 단계;

상기 블록의 변환 계수들의 제 2 세트를 상기 변환 도메인으로부터 상기 샘플 도메인으로 변환하기 위해 상기 역변환을 적용함으로써 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성하는 단계;

상기 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 제 1 역 잔차 변경 함수를 적용하는 단계;

상기 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하는 단계로서, 상기 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제 2 디코딩된 변경된 크로마 잔차 데이터는 제 2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하는 단계; 및

상기 제 1 역 변경된 크로마 잔차 데이터 및 상기 제 2 역 변경된 크로마 잔차 데이터에 기초하여 상기 비디오 데이터의 상기 블록을 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 블록을 재구성하는 단계는,

상기 블록의 제 1 재구성된 크로마 데이터를 획득하기 위해 상기 제 1 역 변경된 크로마 잔차 데이터를 제 1 예측된 크로마 데이터에 가산하는 단계; 및

상기 블록의 제 2 재구성된 크로마 데이터를 획득하기 위해 상기 제 2 역 변경된 크로마 잔차 데이터를 제 2 예측된 크로마 데이터에 가산하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수 또는 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계; 및

상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수 또는 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 3 항에 있어서,

상기 제 1 역 잔차 변경 함수 또는 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계는, 상기 블록의 하나 이상의 특성들에 기초하여, 상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수 또는 상기 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계를 포함하고,

상기 제 2 역 잔차 변경 함수 또는 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계는, 상기 블록의 하나 이상의 특성들에 기초하여, 상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수를 적용할지 여부

를 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 4 항에 있어서,

상기 블록의 상기 특성들은: 상기 블록을 포함하는 슬라이스 또는 타일이 인트라 유형인지 인터 유형인지 여부, 현재 픽처 참조가 상기 블록과 함께 사용되는지 여부, 이중 또는 공유 코딩 트리가 상기 블록과 함께 사용되는지 여부, 상기 블록의 차원들, 또는 상기 블록의 중형비 중 하나 이상을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 3 항에 있어서,

상기 제 1 역 잔차 변경 함수 또는 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계는, 비트스트림에서 시그널링된 데이터에 기초하여, 상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수 또는 상기 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계를 포함하고,

상기 제 2 역 잔차 변경 함수 또는 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계는, 상기 비트스트림에서 시그널링된 상기 데이터에 기초하여, 상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 제 1 역 잔차 변경 함수는 $invF1(x,y) = (resCr(x,y)' + B10) / M10 + (resCb(x,y)' + B11) / M11$ 이고,

상기 제 2 역 잔차 변경 함수는 $invF2(x,y) = (resCr(x,y)' + B20) / M20 - (resCb(x,y)' + B21) / M21$ 이며,

여기서 $resCb(x,y)'$ 는 상기 제 1 디코딩된 변경된 크로마 잔차 데이터이고, $resCr(x,y)'$ 는 상기 제 2 디코딩된 변경된 크로마 잔차 데이터이고, B10, B11, B20 및 B21은 라운딩 항이고, M10, M11, M20 및 M21은 정규화 인자 인, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수를 적용하기 전에 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용하는 단계; 및

상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수를 적용하기 전에 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 역 CRS 프로세스를 적용하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용하는 단계; 및

상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기

역 CRS 프로세스를 적용하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 10

비디오 데이터를 인코딩하는 방법으로서,

제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하는 단계;

상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 단계로서, 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 단계는, 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용한 후, 상기 제 1 변경된 크로마 잔차 데이터를 샘플 도메인으로부터 변환 도메인으로 변환하기 위해 상기 제 1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하는 단계를 포함하는, 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 단계;

제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 블록의 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용하는 단계; 및

상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 단계로서,

상기 제1 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제2 크로마 잔차 데이터는 제 2 크로마 성분과 연관되며,

상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 단계는, 상기 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용한 후, 상기 제 2 변경된 크로마 잔차 데이터를 상기 샘플 도메인으로부터 상기 변환 도메인으로 변환하기 위해 상기 제 2 변경된 크로마 잔차 데이터에 상기 순방향 변환을 적용하는 단계를 포함하는, 상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 11

제 10 항에 있어서,

상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수 또는 제 3 잔차 변경 함수를 적용할지 여부를 결정하는 단계; 및

상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수 또는 제 4 잔차 변경 함수를 적용할지 여부를 결정하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 12

제 11 항에 있어서,

상기 제 1 잔차 변경 함수 또는 제 3 잔차 변경 함수를 적용할지 여부를 결정하는 단계는, 상기 블록의 하나 이상의 특성들에 기초하여, 상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수 또는 상기 제 3 잔차 변경 함수를 적용할지 여부를 결정하는 단계를 포함하고,

상기 제 2 잔차 변경 함수 또는 제 4 잔차 변경 함수를 적용할지 여부를 결정하는 단계는, 상기 블록의 상기 하나 이상의 특성들에 기초하여, 상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수 또는 상기 제 4 잔차 변경 함수를 적용할지 여부를 결정하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 13

제 12 항에 있어서,

상기 블록의 상기 특성들은: 상기 블록을 포함하는 슬라이스 또는 타일이 인트라 유형인지 인터 유형인지 여부, 현재 픽처 참조가 상기 블록과 함께 사용되는지 여부, 이중 또는 공유 코딩 트리가 상기 블록과 함께 사용되는지 여부, 상기 블록의 차원들, 또는 상기 블록의 중형비 중 하나 이상을 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 14

제 11 항에 있어서,

비트스트림에서, 상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 잔차 변경 함수 또는 상기 제 3 잔차 변경 함수가 적용되었는지 여부를 나타내는 데이터를 시그널링하는 단계; 및

상기 비트스트림에서, 상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수가 적용되었는지 여부를 나타내는 데이터를 시그널링하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 15

제 10 항에 있어서,

상기 제 1 잔차 변경 함수는 $F1(x,y) = (resCb(x,y) + A10)/N10 - (resCr(x,y) + A11)/N11$ 이고,

상기 제 2 잔차 변경 함수는 $F2(x,y) = (resCb(x,y) + A20)/N20 + (resCr(x,y) + A21)/N21$ 이며,

여기서 $resCb(x,y)$ 는 상기 제 1 크로마 잔차 데이터이고, $resCr(x,y)$ 는 상기 제 2 크로마 잔차 데이터이고, $A10$, $A11$, $A20$ 및 $A21$ 은 라운딩 항이고, $N10$, $N11$, $N20$ 및 $N21$ 은 정규화 인자인, 비디오 데이터를 인코딩하는 방법.

청구항 16

제 10 항에 있어서,

상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용한 후에 상기 제 1 변경된 크로마 잔차 데이터에 CRS 프로세스를 적용하는 단계; 및

상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수를 적용한 후에 상기 제 2 변경된 크로마 잔차 데이터에 상기 CRS 프로세스를 적용하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 17

제 10 항에 있어서,

상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 1 크로마 잔차 데이터에 CRS 프로세스를 적용하는 단계; 및

상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 2 크로마 잔차 데이터에 상기 CRS 프로세스를 적용하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 18

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하는 메모리; 및

회로로 구현된 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 블록의 변환 계수들의 제 1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성하고;

상기 블록의 변환 계수들의 제 2 세트를 상기 변환 도메인으로부터 상기 샘플 도메인으로 변환하기 위해 상기 역변환을 적용함으로써 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성하며;

상기 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 제 1 역 잔차 변경 함수를 적용하고;

상기 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하는 것으로서, 상기 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제 2 디코딩된 변경된 크로마 잔차

데이터는 제 2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하며; 및

상기 제 1 역 변경된 크로마 잔차 데이터 및 상기 제 2 역 변경된 크로마 잔차 데이터에 기초하여 상기 비디오 데이터의 상기 블록을 재구성하도록 구성된, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 19

제 18 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 블록을 재구성하는 것의 일부로서,

상기 하나 이상의 프로세서들이,

상기 블록의 제 1 재구성된 크로마 데이터를 획득하기 위해 상기 제 1 역 변경된 크로마 잔차 데이터를 제 1 예측된 크로마 데이터에 가산하고;

상기 블록의 제 2 재구성된 크로마 데이터를 획득하기 위해 상기 제 2 역 변경된 크로마 잔차 데이터를 제 2 예측된 크로마 데이터에 가산하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 20

제 18 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 역 잔차 변경 함수 또는 제 3 역 잔차 변경 함수를 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정하고; 및

상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 역 잔차 변경 함수 또는 제 4 역 잔차 변경 함수를 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 21

제 20 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 제 1 역 잔차 변경 함수 또는 상기 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하는 것의 일부로서, 상기 하나 이상의 프로세서들이 상기 블록의 하나 이상의 특성들에 기초하여, 상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수 또는 상기 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하도록 구성되고,

상기 하나 이상의 프로세서들은, 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하는 것의 일부로서, 상기 하나 이상의 프로세서들이 상기 블록의 상기 하나 이상의 특성들에 기초하여, 상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 22

제 21 항에 있어서,

상기 블록의 상기 특성들은: 상기 블록을 포함하는 슬라이스 또는 타일이 인트라 유형인지 인터 유형인지 여부, 현재 픽처 참조가 상기 블록과 함께 사용되는지 여부, 이중 또는 공유 코딩 트리가 상기 블록과 함께 사용되는지 여부, 상기 블록의 차원들, 또는 상기 블록의 총비트 중 하나 이상을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 23

제 20 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 제 1 역 잔차 변경 함수 또는 상기 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하는 것의 일부로서, 상기 하나 이상의 프로세서들이 비트스트림에서 시그널링된 데이터에

기초하여, 상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수 또는 상기 제 3 역 잔차 변경 함수를 적용할지 여부를 결정하도록 구성되고,

상기 하나 이상의 프로세서들은, 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하는 것의 일부로서, 상기 하나 이상의 프로세서들이 상기 비트스트림에서 시그널링된 상기 데이터에 기초하여, 상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수를 적용할지 여부를 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 24

제 18 항에 있어서,

상기 제 1 역 잔차 변경 함수는 $invF1(x,y) = (resCr(x,y)' + B10) / M10 + (resCb(x,y)' + B11) / M11$ 이고,

상기 제 2 역 잔차 변경 함수는 $invF2(x,y) = (resCr(x,y)' + B20) / M20 - (resCb(x,y)' + B21) / M21$ 이며,

여기서 $resCb(x,y)'$ 는 상기 제 1 디코딩된 변경된 크로마 잔차 데이터이고, $resCr(x,y)'$ 는 상기 제 2 디코딩된 변경된 크로마 잔차 데이터이고, B10, B11, B20 및 B21은 라운딩 항이고, M10, M11, M20 및 M21은 정규화 인자인, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 25

제 18 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수를 적용하기 전에 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용하고; 및

상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수를 적용하기 전에 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 역 CRS 프로세스를 적용하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 26

제 18 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 상기 제 1 역 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용하고; 및

상기 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 제 2 역 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 상기 역 CRS 프로세스를 적용하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 27

제 18 항에 있어서,

디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 28

제 18 항에 있어서,

상기 디바이스는 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스, 또는 셋탑 박스 중 하나 이상을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 29

비디오 데이터를 인코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하는 메모리; 및

회로로 구현된 하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하고;

상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 것으로서, 상기 하나 이상의 프로세서들은 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 상기 하나 이상의 프로세서들이, 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용한 후, 상기 제 1 변경된 크로마 잔차 데이터를 샘플 도메인으로부터 변환 도메인으로 변환하기 위해 상기 제 1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하도록 구성되는, 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하며;

제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 블록의 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용하고; 및

상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 것으로서,

상기 제1 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제2 크로마 잔차 데이터는 제 2 크로마 성분과 연관되며,

상기 하나 이상의 프로세서들은, 상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 상기 하나 이상의 프로세서들이, 상기 제 2 잔차 변경 함수를 상기 제 2 크로마 잔차 데이터에 적용한 후, 상기 제 2 변경된 크로마 잔차 데이터를 상기 샘플 도메인으로부터 상기 변환 도메인으로 변환하기 위해 상기 제 2 변경된 크로마 잔차 데이터에 상기 순방향 변환을 적용하도록 구성되는, 상기 제 2 변경된 크로마 잔차 데이터를 인코딩하도록 구성된, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 30

제 29 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수 또는 제 3 잔차 변경 함수를 적용할지 여부를 결정하고; 및

상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수 또는 제 4 잔차 변경 함수를 적용할지 여부를 결정하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 31

제 30 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 제 1 잔차 변경 함수 또는 상기 제 3 잔차 변경 함수를 적용할지 여부를 결정하는 것의 일부로서, 상기 하나 이상의 프로세서들이 상기 블록의 하나 이상의 특성들에 기초하여, 상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수 또는 상기 제 3 잔차 변경 함수를 적용할지 여부를 결정하도록 구성되고,

상기 하나 이상의 프로세서들은, 상기 제 2 잔차 변경 함수 또는 상기 제 4 잔차 변경 함수를 적용할지 여부를 결정하는 것의 일부로서, 상기 하나 이상의 프로세서들이 상기 블록의 상기 하나 이상의 특성들에 기초하여, 상

기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수 또는 상기 제 4 잔차 변경 함수를 적용할지 여부를 결정하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 32

제 31 항에 있어서,

상기 블록의 상기 특성들은: 상기 블록을 포함하는 슬라이스 또는 타일이 인트라 유형인지 인터 유형인지 여부, 현재 픽처 참조가 상기 블록과 함께 사용되는지 여부, 이중 또는 공유 코딩 트리가 상기 블록과 함께 사용되는지 여부, 상기 블록의 차원들, 또는 상기 블록의 중형비 중 하나 이상을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 33

제 30 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

비트스트림에서, 상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 잔차 변경 함수 또는 상기 제 3 잔차 변경 함수가 적용되었는지 여부를 나타내는 데이터를 시그널링하고; 및

상기 비트스트림에서, 상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 역 잔차 변경 함수 또는 상기 제 4 역 잔차 변경 함수가 적용되었는지 여부를 나타내는 데이터를 시그널링하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 34

제 29 항에 있어서,

상기 제 1 잔차 변경 함수는 $F1(x,y) = (resCb(x,y) + A10)/N10 - (resCr(x,y) + A11)/N11$ 이고,

상기 제 2 잔차 변경 함수는 $F2(x,y) = (resCb(x,y) + A20)/N20 + (resCr(x,y) + A21)/N21$ 이며,

여기서 $resCb(x,y)$ 는 상기 제 1 크로마 잔차 데이터이고, $resCr(x,y)$ 는 상기 제 2 크로마 잔차 데이터이고, A10, A11, A20 및 A21은 라운딩 항이고, N10, N11, N20 및 N21은 정규화 인자인, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 35

제 29 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용한 후에 상기 제 1 변경된 크로마 잔차 데이터에 CRS 프로세스를 적용하고; 및

상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수를 적용한 후에 상기 제 2 변경된 크로마 잔차 데이터에 상기 CRS 프로세스를 적용하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 36

제 29 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 1 크로마 잔차 데이터에 CRS 프로세스를 적용하고; 및

상기 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수를 적용하는 것의 일부로서 상기 제 2 크로마 잔차 데이터에 상기 CRS 프로세스를 적용하도록 구성되는, 비

디오 데이터를 인코딩하기 위한 디바이스.

청구항 37

제 29 항에 있어서,

상기 디바이스는 카메라, 컴퓨터, 또는 모바일 디바이스 중 하나 이상을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 38

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 비디오 데이터의 블록의 변환 계수들의 제 1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성하는 수단;

상기 블록의 변환 계수들의 제 2 세트를 상기 변환 도메인으로부터 상기 샘플 도메인으로 변환하기 위해 상기 역변환을 적용함으로써 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성하는 수단;

상기 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 제 1 역 잔차 변경 함수를 적용하는 수단;

상기 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하는 수단으로서, 상기 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제 2 디코딩된 변경된 크로마 잔차 데이터는 제 2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하는 수단; 및

상기 제 1 역 변경된 크로마 잔차 데이터 및 상기 제 2 역 변경된 크로마 잔차 데이터에 기초하여 상기 비디오 데이터의 상기 블록을 재구성하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 39

비디오 데이터를 인코딩하기 위한 디바이스로서,

제 1 변경된 크로마 잔차 데이터를 생성하기 위해 상기 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하는 수단;

상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 수단으로서, 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 수단은, 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용한 후, 상기 제 1 변경된 크로마 잔차 데이터를 샘플 도메인으로부터 변환 도메인으로 변환하기 위해 상기 제 1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하는 수단을 포함하는, 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하는 수단;

제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 블록의 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용하는 수단; 및

상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 수단으로서,

상기 제 1 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제 2 크로마 잔차 데이터는 제 2 크로마 성분과 연관되며,

상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 수단은, 상기 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용한 후, 상기 제 2 변경된 크로마 잔차 데이터를 상기 샘플 도메인으로부터 상기 변환 도메인으로 변환하기 위해 상기 제 2 변경된 크로마 잔차 데이터에 상기 순방향 변환을 적용하는 수단을 포함하는, 상기 제 2 변경된 크로마 잔차 데이터를 인코딩하는 수단을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 40

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금,

비디오 데이터의 블록의 변환 계수들의 제 1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변

환을 적용함으로써 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성하게 하고;

상기 블록의 변환 계수들의 제 2 세트를 상기 변환 도메인으로부터 상기 샘플 도메인으로 변환하기 위해 상기 역변환을 적용함으로써 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성하게 하며;

상기 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 1 디코딩된 변경된 크로마 잔차 데이터에 제 1 역 잔차 변경 함수를 적용하게 하고;

상기 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 상기 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하게 하는 것으로서, 상기 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제2 디코딩된 변경된 크로마 잔차 데이터는 제 2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하게 하며; 및

상기 제 1 역 변경된 크로마 잔차 데이터 및 상기 제 2 역 변경된 크로마 잔차 데이터에 기초하여 상기 비디오 데이터의 상기 블록을 재구성하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 41

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금,

제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하게 하고;

상기 제 1 변경된 크로마 잔차 데이터를 인코딩하게 하는 것으로서, 상기 하나 이상의 프로세서들로 하여금 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하게 하는 것의 일부로서, 상기 명령들의 실행은 상기 하나 이상의 프로세서들로 하여금, 상기 제 1 크로마 잔차 데이터에 상기 제 1 잔차 변경 함수를 적용한 후, 상기 제 1 변경된 크로마 잔차 데이터를 샘플 도메인으로부터 변환 도메인으로 변환하기 위해 상기 제 1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하게 하는, 상기 제 1 변경된 크로마 잔차 데이터를 인코딩하게 하며;

제 2 변경된 크로마 잔차 데이터를 생성하기 위해 상기 블록의 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용하게 하고; 및

상기 제 2 변경된 크로마 잔차 데이터를 인코딩하게 하는 것으로서,

상기 제1 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 상기 제2 크로마 잔차 데이터는 제 2 크로마 성분과 연관되며,

상기 하나 이상의 프로세서들로 하여금 상기 제 2 변경된 크로마 잔차 데이터를 인코딩하게 하는 것의 일부로서, 상기 명령들의 실행은 상기 하나 이상의 프로세서들로 하여금, 상기 제 2 크로마 잔차 데이터에 상기 제 2 잔차 변경 함수를 적용한 후, 상기 제 2 변경된 크로마 잔차 데이터를 상기 샘플 도메인으로부터 상기 변환 도메인으로 변환하기 위해 상기 제 2 변경된 크로마 잔차 데이터에 상기 순방향 변환을 적용하게 하는, 상기 제 2 변경된 크로마 잔차 데이터를 인코딩하게 하는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 이 출원은 2020년 3월 5일자로 출원된 미국 특허 출원 제 16/810,680 호, 2019년 3월 8일자로 출원된 미국 가특허출원 제 62/815,936 호, 및 2019년 6월 25일자로 출원된 미국 가특허출원 제 62/866,450 호의 이익을 주장하고, 그것들 각각의 전체 내용은 참조에 의해 통합된다.

[0002] 본 개시는 비디오 인코딩 및 비디오 디코딩에 관한 것이다.

배경 기술

[0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인용 디지털 보조기 (PDA) 들, 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털

카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 텔레컨퍼런싱 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC), ITU-T H.265, 고효율 비디오 코딩 (HEVC) 표준에 의해 정의된 표준들, 및 그러한 표준들의 확장들에서 설명된 기법들과 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 그러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 코딩 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소 또는 제거하기 위해 공간 (인트라 픽처) 예측 및/또는 시간 (인터 픽처) 예측을 포함한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (예를 들어, 비디오 픽처 또는 비디오 픽처의 부분) 는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 코딩 트리 유닛들 (CTU들), 코딩 유닛들 (CU들) 및/또는 코딩 노드들로도 또한 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃 블록들에서의 레퍼런스 샘플들에 대한 공간 예측을 사용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처에 있어서 이웃하는 블록들에서의 참조 샘플들에 대한 공간적 예측 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들로 지칭될 수도 있고, 참조 픽처들은 레퍼런스 프레임들로 지칭될 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0005] 일반적으로, 본 개시는 비디오 코딩에서의 결합된 잔차 코딩을 위한 기법들을 설명한다. 본 명세서에 설명된 바와 같이, 비디오 인코더 및 비디오 디코더는 비디오 데이터의 블록들을 인코딩 및 디코딩하기 위해 결합된 잔차 코딩을 수행할 수 있다. 결합된 잔차 코딩을 수행하기 위해, 비디오 인코더는 블록의 Cb 잔차 데이터에 제1 잔차 변경 함수를 적용하고 블록의 Cr 잔차 데이터에 제2 잔차 변경 함수를 적용할 수 있다. 제1 잔차 변경 함수는 블록의 원래 Cb 잔차 데이터와 블록의 Cr 잔차 데이터 양자 두를 기반으로 블록의 Cb 잔차 데이터를 변경한다. 유사하게, 제2 잔차 변경 함수는 블록의 원래 Cr 잔차 데이터와 블록의 Cb 잔차 데이터 양자 두에 기초하여 블록의 Cr 잔차 데이터를 변경한다. 비디오 디코더는 블록의 디코딩된 Cb 및 Cr 잔차 데이터에 역 잔차 변경 함수를 적용한다. 잔차 변경 함수를 적용함으로써, 비디오 인코더는 더 낮은 값을 사용하여 Cb 및 Cr 크로마 잔차 데이터를 나타낼 수 있을 수 있다. 더 낮은 값은 일반적으로 더 적은 비트를 사용하여 인코딩될 수 있다. 따라서, 잔차 변경 함수의 적용은 더 큰 코딩 효율(예를 들어, 더 큰 압축)을 초래할 수 있다.

[0006] 일례에서, 본 개시물은 비디오 데이터를 디코딩하는 방법을 설명하고, 이 방법은 비디오 데이터의 블록의 변환 계수들의 제1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제1 디코딩된 변경된 크로마 잔차 데이터를 생성하는 단계; 블록의 변환 계수들의 제2 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제2 디코딩된 변경된 크로마 잔차 데이터를 생성하는 단계; 제1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제1 디코딩된 변경된 크로마 잔차 데이터에 제1 역 잔차 변경 함수를 적용하는 단계; 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하는 단계로서, 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 제2 디코딩된 변경된 크로마 잔차 데이터는 제2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하는 단계; 및 제1 역 변경된 크로마 잔차 데이터 및 제2 역 변경된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성하는 단계를 포함한다.

[0007] 다른 예에서, 본 개시물은 비디오 데이터를 인코딩하는 방법을 설명하고, 상기 방법은: 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하는 단계; 제1 변경된 크로마 잔차 데이터를 인코딩하는 단계로서, 제1 변경된 크로마 잔차 데이터를 인코딩하는 단계는, 제1 잔차 변경 함수를 제1 크로마 잔차 데이터에 적용한 후, 제1 변경된 크로마 잔차 데이터를 샘플 도

메인에서 변환 도메인으로 변환하기 위해 제1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하는 단계를 포함하는, 상기 제1 변경된 크로마 잔차 데이터를 인코딩하는 단계; 제2 변경된 크로마 잔차 데이터를 생성하기 위해 블록의 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용하는 단계; 및 제2 변경된 크로마 잔차 데이터를 인코딩하는 단계를 포함하고, 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관되며, 제2 변경된 크로마 잔차 데이터를 인코딩하는 단계는, 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용한 후, 샘플 도메인으로부터 변환 도메인으로 제2 변경된 크로마 잔차 데이터를 변환하기 위해 제2 변경된 크로마 잔차 데이터에 순방향 변환을 적용하는 단계를 포함한다.

[0008] 다른 예에서, 본 개시물은 비디오 데이터를 디코딩하는 디바이스를 설명하고, 이 디바이스는 비디오 데이터를 저장하기 위한 메모리; 및 회로로 구현된 하나 이상의 프로세서들을 포함하며, 그 하나 이상의 프로세서들은 비디오 데이터의 블록의 변환 계수들의 제1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제1 디코딩된 변경된 크로마 잔차 데이터를 생성하고; 블록의 변환 계수들의 제2 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제2 디코딩된 변경된 크로마 잔차 데이터를 생성하며; 제1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제1 디코딩된 변경된 크로마 잔차 데이터에 제1 역 잔차 변경 함수를 적용하고; 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하는 것으로서, 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 제2 디코딩된 변경된 크로마 잔차 데이터는 제2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하고; 및 제1 역 변경된 크로마 잔차 데이터 및 제2 역 변경된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성하도록 구성된다.

[0009] 다른 예에서, 본 개시물은 비디오 데이터를 인코딩하는 디바이스를 설명하고, 이 디바이스는 비디오 데이터를 저장하기 위한 메모리; 및 회로로 구현된 하나 이상의 프로세서들을 포함하며, 하나 이상의 프로세서들은 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하고; 제1 변경된 크로마 잔차 데이터를 인코딩하는 것으로서, 하나 이상의 프로세서들은, 제1 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 하나 이상의 프로세서들이 제1 잔차 변경 함수를 제1 크로마 잔차 데이터에 적용한 후, 제1 변경된 크로마 잔차 데이터를 샘플 도메인에서 변환 도메인으로 변환하기 위해 제1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하도록 구성되는, 상기 제1 변경된 크로마 잔차 데이터를 인코딩하며; 제2 변경된 크로마 잔차 데이터를 생성하기 위해 블록의 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용하고; 및 제2 변경된 크로마 잔차 데이터를 인코딩하도록 구성되며, 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관되며, 하나 이상의 프로세서들은, 제2 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 하나 이상의 프로세서들이 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용한 후, 샘플 도메인으로부터 변환 도메인으로 제2 변경된 크로마 잔차 데이터를 변환하기 위해 제2 변경된 크로마 잔차 데이터에 순방향 변환을 적용하도록 구성된다.

[0010] 다른 예에서, 본 개시물은 비디오 데이터를 디코딩하는 디바이스를 설명하고, 이 디바이스는 비디오 데이터의 블록의 변환 계수들의 제1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제1 디코딩된 변경된 크로마 잔차 데이터를 생성하는 수단; 블록의 변환 계수들의 제2 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제2 디코딩된 변경된 크로마 잔차 데이터를 생성하는 수단; 제1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제1 디코딩된 변경된 크로마 잔차 데이터에 제1 역 잔차 변경 함수를 적용하는 수단; 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하는 수단으로서, 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 제2 디코딩된 변경된 크로마 잔차 데이터는 제2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하는 수단; 및 제1 역 변경된 크로마 잔차 데이터 및 제2 역 변경된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성하는 수단을 포함한다.

[0011] 다른 예에서, 본 개시물은 비디오 데이터를 인코딩하는 디바이스를 설명하고, 이 디바이스는: 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하는 수단; 제1 변경된 크로마 잔차 데이터를 인코딩하는 수단으로서, 제1 변경된 크로마 잔차 데이터를 인코딩하는 수단은, 제1 잔차 변경 함수를 제1 크로마 잔차 데이터에 적용한 후, 제1 변경된 크로마 잔차 데이터를 샘플 도메인에서 변환 도메인으로 변환하기 위해 제1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하는 수단을 포함하는, 상기 제1 변경된 크로마 잔차 데이터를 인코딩하는 수단; 제2 변경된 크로마 잔차 데이터를 생

성하기 위해 블록의 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용하는 수단; 및 제2 변경된 크로마 잔차 데이터를 인코딩하는 수단을 포함하고, 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관되며, 제2 변경된 크로마 잔차 데이터를 인코딩하는 수단은, 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용한 후, 샘플 도메인으로부터 변환 도메인으로 제2 변경된 크로마 잔차 데이터를 변환하기 위해 제2 변경된 크로마 잔차 데이터에 순방향 변환을 적용하는 수단을 포함한다.

[0012] 다른 예에서, 본 개시물은 명령들을 저장한 컴퓨터 판독가능 저장 매체를 설명하고, 이 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 데이터의 블록의 변환 계수들의 제1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제1 디코딩된 변경된 크로마 잔차 데이터를 생성하게 하고; 블록의 변환 계수들의 제2 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제2 디코딩된 변경된 크로마 잔차 데이터를 생성하게 하며; 제1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제1 디코딩된 변경된 크로마 잔차 데이터에 제1 역 잔차 변경 함수를 적용하게 하고; 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용하게 하는 것으로서, 제 1 디코딩된 변경된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 제2 디코딩된 변경된 크로마 잔차 데이터는 제2 크로마 성분과 연관되는, 상기 제 2 역 잔차 변경 함수를 적용하게 하고; 및 제1 역 변경된 크로마 잔차 데이터 및 제2 역 변경된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성하게 한다.

[0013] 다른 예에서, 본 개시물은 명령들을 저장한 컴퓨터 판독가능 저장 매체를 설명하고, 이 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금, 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하게 하고; 제1 변경된 크로마 잔차 데이터를 인코딩하게 하는 것으로서, 하나 이상의 프로세서들로 하여금 제1 변경된 크로마 잔차 데이터를 인코딩하게 하는 것의 일부로서, 그 명령들의 실행은 하나 이상의 프로세서들로 하여금, 제1 잔차 변경 함수를 제1 크로마 잔차 데이터에 적용한 후, 제1 변경된 크로마 잔차 데이터를 샘플 도메인에서 변환 도메인으로 변환하기 위해 제1 변경된 크로마 잔차 데이터에 순방향 변환을 적용하게 하는, 상기 제1 변경된 크로마 잔차 데이터를 인코딩하게 하며; 제2 변경된 크로마 잔차 데이터를 생성하기 위해 블록의 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용하게 하며; 및 제2 변경된 크로마 잔차 데이터를 인코딩하게 하고, 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관되며, 하나 이상의 프로세서들로 하여금 제2 변경된 크로마 잔차 데이터를 인코딩하게 하는 것의 일부로서, 그 명령들의 실행은 하나 이상의 프로세서들로 하여금, 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용한 후, 샘플 도메인으로부터 변환 도메인으로 제2 변경된 크로마 잔차 데이터를 변환하기 위해 제2 변경된 크로마 잔차 데이터에 순방향 변환을 적용하게 한다.

[0014] 하나 이상의 예들의 상세들은 첨부 도면들 및 이하의 설명에 기재된다. 다른 피쳐들, 목적들, 및 이점들은 설명, 도면들, 및 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0015] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록 다이어그램이다.

도 2 는 인루프 루마 재성형기 (reshaper) 를 사용한 인트라 슬라이스 재구성의 예를 예시하는 블록도이다.

도 3 은 인루프 루마 재성형기를 사용한 인터 슬라이스 재구성의 예를 예시하는 블록도이다.

도 4 는 크로마 잔차 스케일링을 사용한 인트라 모드 및 인터 모드 재구성의 예를 예시하는 블록도이다.

도 5a 및 도 5b 는 예시적인 쿼드트리 이진 트리 (QIBT) 구조, 및 대응하는 코딩 트리 유닛 (CTU) 을 예시하는 개념적 다이어그램들이다.

도 6 은 본 개시의 기법들을 수행할 수도 있는 예시의 비디오 인코더를 도시하는 블록 다이어그램이다.

도 7 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더를 예시하는 블록 다이어그램이다.

도 8 은 현재의 블록을 인코딩하는 예시의 방법을 도시하는 플로우차트이다.

도 9 는 비디오 데이터의 현재 블록을 디코딩하는 예시적인 방법을 나타내는 플로우차트이다.

도 10 은 본 개시의 하나 이상의 기법들에 따른, 비디오 인코더의 예시적인 동작을 나타내는 플로우차트이다.

도 11 은 본 개시의 하나 이상의 기법들에 따른, 비디오 디코더의 예시적인 동작을 나타내는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

- [0016] 비디오 데이터는 종종 YCbCr 색 공간으로 표현된다. 즉, 픽처의 각 픽셀은 루마 (Y) 성분 및 두 개의 크로마 성분들 (Cb 및 Cr) 로 표현될 수 있다. 비디오 데이터를 YCbCr 색 공간으로 표현하는 것은 RGB (Red Green Blue) 색 공간 고유의 중복성을 줄일 수 있다. 그러나, Cb 와 Cr 값들 사이에는 상당한 상관 (correlation) 이 남아 있을 수 있다. 이러한 상관을 감소시키는 것은 더 나은 코딩 효율성을 가능하게 할 수 있다.
- [0017] 본 개시물은 코딩 효율을 개선할 수 있는 크로스-성분 상관 (cross-component correlation) 을 감소시킬 수 있는 기법들을 설명한다. 본 명세서에 설명된 바와 같이, 비디오 인코더는 잔차 크로마 데이터에 잔차 변경 함수 (residual modification function: RMF) 를 적용할 수 있다. 비디오 디코더는 디코딩된 잔차 크로마 데이터에 역 RMF 를 적용할 수 있다. 역 RMF는 비디오 인코더에 의해 적용된 RMF의 효과를 적어도 부분적으로 역전시킬 수 있다. 본 명세서에서 설명되는 바와 같이, 크로마 성분에 대한 잔차 크로마 데이터에 적용되는 RMF는 그 크로마 성분 및 다른 크로마 성분에 대한 잔차 크로마 데이터를 사용할 수 있다. 따라서, 변경된 크로마 잔차 데이터의 두 세트는 여전히 단일 블록의 상이한 크로마 성분들에 대해 인코딩되지만, 비디오 인코더는 서로의 관점에서 그 크로마 성분들에 대한 크로마 잔차 데이터를 변경한다. 이러한 RMF를 적용하는 것은 코딩 효율을 향상시킬 수 있다. 예를 들어, 그러한 RMF의 적용은 크로마 잔차 데이터의 값을 감소시킬 수 있기 때문에 그러한 RMF를 적용하는 것은 비디오 데이터를 인코딩하는 데 필요한 비트의 수를 감소시킬 수 있다.
- [0018] 일 예에서, 비디오 인코더는 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용할 수 있다. 비디오 인코더는 또한 제1 변경된 크로마 잔차 데이터를 인코딩할 수 있다. 제 1 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 비디오 인코더는 제 1 잔차 변경 함수를 제 1 크로마 잔차 데이터에 적용한 후, 제 1 변경된 크로마 잔차 데이터를 샘플 도메인으로부터 변환 도메인으로 변환하기 위해 제 1 변경된 크로마 잔차 데이터에 순방향 변환을 적용할 수 있다. 추가적으로, 비디오 인코더는 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 블록의 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용할 수 있다. 비디오 인코더는 또한 제2 변경된 크로마 잔차 데이터를 인코딩할 수 있다. 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관된다. 제 2 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 비디오 인코더는 제 2 잔차 변경 함수를 제 2 크로마 잔차 데이터에 적용한 후, 제 2 변경된 크로마 잔차 데이터를 샘플 도메인으로부터 변환 도메인으로 변환하기 위해 제 2 변경된 크로마 잔차 데이터에 순방향 변환을 적용할 수 있다.
- [0019] 유사하게, 비디오 디코더는 비디오 데이터의 블록의 변환 계수들의 제 1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성할 수 있다. 추가적으로, 비디오 디코더는 블록의 변환 계수들의 제 2 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성할 수 있다. 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성한 후에, 비디오 디코더는 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 디코딩된 변경된 크로마 잔차 데이터에 제 1 역 잔차 변경 함수를 적용할 수 있다. 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성한 후에, 비디오 디코더는 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용할 수도 있다. 제1 디코딩된 변경된 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 디코딩된 변경된 크로마 잔차 데이터는 제2 크로마 성분과 연관된다. 비디오 디코더는 제 1 역 변경된 크로마 잔차 데이터 및 제 2 역 변경된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성할 수 있다.
- [0020] 도 1 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (100) 을 나타내는 블록도이다. 본 개시의 기법들은 일반적으로 비디오 데이터를 코딩 (인코딩 및/또는 디코딩) 하는 것과 관련된다. 일반적으로, 비디오 데이터는 비디오를 프로세싱하기 위한 임의의 데이터를 포함한다. 따라서, 비디오 데이터는 원시, 인코딩되지 않은 비디오, 인코딩된 비디오, 디코딩된 (예를 들어, 복원된) 비디오, 및 비디오 메타데이터, 예컨대 시그널링 데이터를 포함할 수도 있다.
- [0021] 도 1 에 도시된 바와 같이, 시스템 (100) 은, 이 예에서 목적지 디바이스 (116) 에 의해 디코딩 및 디스플레이

될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (102) 를 포함한다. 특히, 소스 디바이스 (102) 는 비디오 데이터를 컴퓨터 판독가능 매체 (110) 를 통해 목적지 디바이스 (116) 에 제공한다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋탑 박스들, 전화기 핸드셋들, 예컨대 스마트폰들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 무선 통신을 위해 장비될 수도 있고, 따라서 무선 통신 디바이스들로 지칭될 수도 있다.

[0022] 도 1 의 예에서, 도 1 의 예에서, 소스 디바이스 (102) 는 비디오 소스 (104), 메모리 (106), 비디오 인코더 (200), 및 출력 인터페이스 (108) 를 포함한다. 목적지 디바이스 (116) 는 입력 인터페이스 (122), 비디오 디코더 (300), 메모리 (120), 및 디스플레이 디바이스 (118) 를 포함한다. 본 개시에 따르면, 소스 디바이스 (102) 의 비디오 인코더 (200) 및 목적지 디바이스 (116) 의 비디오 디코더 (300) 는 결합된 잔차 코딩 (CRC) 을 위한 본 개시의 기법들을 적용하도록 구성될 수 있다. 따라서, 소스 디바이스 (102) 는 비디오 인코딩 디바이스의 예를 나타내는 한편, 목적지 디바이스 (116) 는 비디오 디코딩 디바이스의 예를 나타낸다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102) 는 외부 카메라와 같은 외부 비디오 소스로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (116) 는 통합된 디스플레이 디바이스를 포함하기 보다는, 외부 디스플레이 디바이스와 인터페이스로 접속할 수도 있다.

[0023] 도 1 에 나타낸 바와 같은 시스템 (100) 은 도 1 에 도시된 바와 같은 시스템 (100) 은 하나의 예일 뿐이다. 일반적으로, 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스는 결합된 잔차 코딩을 위한 본 명세서에 기재된 기법들을 수행할 수 있다. 소스 디바이스 (102) 및 목적지 디바이스 (116) 는 소스 디바이스 (102) 가 목적지 디바이스 (116) 로의 송신을 위해 코딩된 비디오 데이터를 생성하는 그러한 코딩 디바이스들의 예들 일 뿐이다. 본 개시는 데이터의 코딩 (인코딩 및/또는 디코딩) 을 수행하는 디바이스로서 "코딩" 디바이스를 참조한다. 따라서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 코딩 디바이스들, 특히 각각 비디오 인코더 및 비디오 디코더의 예들을 나타낸다. 일부 예들에 있어서, 소스 디바이스 (102) 및 목적지 디바이스 (116) 는, 소스 디바이스 (102) 및 목적지 디바이스 (116) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (100) 은 예를 들면, 비디오 스트리밍, 비디오 플레이어백, 비디오 브로드캐스팅, 또는 비디오 텔레포니를 위해, 소스 디바이스 (102) 와 목적지 디바이스 (116) 간의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0024] 일반적으로, 비디오 소스 (104) 는 비디오 데이터 (즉, 원시, 코딩되지 않은 비디오 데이터) 의 소스를 나타내며 픽처들에 대한 데이터를 인코딩하는 비디오 인코더 (200) 에 비디오 데이터의 순차적인 일련의 픽처들 (또한 "프레임들" 로도 지칭됨) 을 제공한다. 소스 디바이스 (102) 의 비디오 소스 (104) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스 (video feed interface) 를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (104) 는 컴퓨터 그래픽 기반 데이터를 소스 비디오로서, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합으로서 생성할 수도 있다. 각각의 경우에 있어서, 비디오 인코더 (200) 는 캡처된, 미리-캡처된, 또는 컴퓨터-생성된 비디오 데이터를 인코딩한다. 비디오 인코더 (200) 는 픽처들을 수신된 순서 (때때로 "디스플레이 순서" 로 지칭됨) 로부터 코딩을 위한 코딩 순서로 재배열할 수도 있다. 비디오 인코더 (200) 는 인코딩된 비디오 데이터를 포함하는 비트스트림을 생성할 수도 있다. 소스 디바이스 (102) 는 그 후 예를 들어 목적지 디바이스 (116) 의 입력 인터페이스 (122) 에 의한 수신 및/또는 취출을 위해 출력 인터페이스 (108) 를 통해 컴퓨터 판독가능 매체 (110) 상으로 인코딩된 비디오 데이터를 출력할 수도 있다.

[0025] 소스 디바이스 (102) 의 메모리 (106) 및 목적지 디바이스 (116) 의 메모리 (120) 는 범용 메모리들을 나타낸다. 일부 예에서, 메모리들 (106, 120) 은 원시 비디오 데이터, 예를 들어, 비디오 소스 (104) 로부터의 원시 비디오 및 비디오 디코더 (300) 로부터의 원시, 디코딩된 비디오 데이터를 저장할 수도 있다. 추가적으로 또는 대안적으로, 메모리들 (106, 120) 은 예를 들어, 각각 비디오 인코더 (200) 및 비디오 디코더 (300) 에 의해 실행가능한 소프트웨어 명령들을 저장할 수도 있다. 메모리 (106) 및 메모리 (120) 는 이 예에서 비디오 인코더 (200) 및 비디오 디코더 (300) 와 별도로 도시되지만, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 또한 기능적으로 유사하거나 또는 동등한 목적들을 위한 내부 메모리들을 포함할 수도 있음을 이해해야 한다. 또한, 메모리들 (106, 120) 은 예를 들어, 비디오 인코더 (200) 로부터 출력되고 비디오 디코더

(300) 에 입력되는 인코딩된 비디오 데이터를 저장할 수도 있다. 일부 예들에서, 메모리들 (106, 120) 의 부분들은 예를 들어, 원시, 디코딩된, 및/또는 인코딩된 비디오 데이터를 저장하기 위해 하나 이상의 비디오 버퍼들로서 할당될 수도 있다.

[0026] 컴퓨터 판독가능 매체 (110) 는 인코딩된 비디오 데이터를 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로 전송할 수 있는 임의의 타입의 매체 또는 디바이스를 나타낼 수도 있다. 하나의 예에서, 컴퓨터 판독가능 매체 (110) 는, 소스 디바이스 (102) 로 하여금, 인코딩된 비디오 데이터를 직접 목적지 디바이스 (116) 에 실시간으로, 예를 들어 무선 주파수 네트워크 또는 컴퓨터 기반 네트워크를 통해 송신할 수 있게 하기 위한 통신 매체를 나타낸다. 무선 통신 프로토콜과 같은 통신 표준에 따라, 출력 인터페이스 (108) 는 인코딩된 비디오 데이터를 포함하는 송신 신호를 변조할 수도 있고, 입력 인터페이스 (122) 는 수신된 송신 신호를 복조할 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 예컨대 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예컨대 인터넷의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (102) 로부터 목적지 디바이스 (116) 로의 통신을 가능하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0027] 일부 예들에 있어서, 컴퓨터 판독가능 매체 (110) 는 저장 디바이스 (112) 를 포함할 수도 있다. 소스 디바이스 (102) 는 출력 인터페이스 (108) 로부터 저장 디바이스 (112) 로 인코딩된 데이터를 출력할 수도 있다. 유사하게, 목적지 디바이스 (116) 는 입력 인터페이스 (122) 를 통해 저장 디바이스 (112) 로부터의 인코딩된 데이터에 액세스할 수도 있다. 저장 디바이스 (112) 는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체와 같은 다양한 분산된 또는 로컬 액세스된 데이터 저장 매체 중 임의의 것을 포함할 수도 있다.

[0028] 일부 예들에서, 컴퓨터 판독가능 매체 (110) 는 소스 디바이스 (102) 에 의해 생성된 인코딩된 비디오 데이터를 저장할 수도 있는 파일 서버 (114) 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 소스 디바이스 (102) 는, 소스 디바이스 (102) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 (114) 또는 다른 중간 저장 디바이스로 인코딩된 비디오 데이터를 출력할 수도 있다. 목적지 디바이스 (116) 는 스트리밍 또는 다운로드를 통해 파일 서버 (114) 로부터의 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버 (114) 는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (116) 에 송신할 수 있는 임의의 타입의 서버 디바이스일 수도 있다. 파일 서버 (114) 는 (예를 들어, 웹 사이트를 위한) 웹 서버, 파일 전송 프로토콜 (FTP) 서버, 콘텐츠 전달 네트워크 디바이스, 또는 NAS (network attached storage) 디바이스를 나타낼 수도 있다. 목적지 디바이스 (116) 는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 파일 서버 (114) 로부터 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 (114) 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한, 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, 디지털 가입자 라인 (DSL), 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 파일 서버 (114) 및 입력 인터페이스 (122) 는 스트리밍 송신 프로토콜, 다운로드 송신 프로토콜 또는 이들의 조합에 따라 동작하도록 구성될 수도 있다.

[0029] 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 무선 송신기/수신기, 모뎀들, 유선 네트워킹 컴포넌트들 (예를 들어, 이더넷 카드들), 다양한 IEEE 802.11 표준들 중 임의의 것에 따라 동작하는 무선 통신 컴포넌트들, 또는 다른 물리적 컴포넌트들을 나타낼 수도 있다. 출력 인터페이스 (108) 및 입력 인터페이스 (122) 가 무선 컴포넌트들을 포함하는 예들에서, 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 4G, 4G-LTE (Long-Term Evolution), LTE 어드밴스드, 5G 등과 같은 셀룰러 통신 표준에 따라, 인코딩된 비디오 데이터와 같은 데이터를 전송하도록 구성될 수도 있다. 출력 인터페이스 (108) 가 무선 송신기를 포함하는 일부 예들에 있어서, 출력 인터페이스 (108) 및 입력 인터페이스 (122) 는 IEEE 802.11 사양, IEEE 802.15 사양 (예컨대, ZigBee™), Bluetooth™ 표준 등과 같은 다른 무선 표준들에 따라, 인코딩된 비디오 데이터와 같은 데이터를 전송하도록 구성될 수도 있다. 일부 예들에 있어서, 소스 디바이스 (102) 및/또는 목적지 디바이스 (116) 는 개별 SoC (system-on-a-chip) 디바이스들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (102) 는 비디오 인코더 (200) 및/또는 출력 인터페이스 (108) 에 기인한 기능성을 수행하기 위한 SoC 디바이스를 포함할 수도 있고, 목적지 디바이스 (116) 는 비디오 디코더 (300) 및/또는 입력 인터페이스 (122) 에 기인한 기능성을 수행하기 위한 SoC 디바이스를 포함할 수도 있다.

[0030] 본 개시의 기법들은 오버-더-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔

레비전 송신들, 인터넷 스트리밍 비디오 송신들, 예컨대 DASH (dynamic adaptive streaming over HTTP), 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들 중 임의의 것을 지원하여 비디오 코딩에 적용될 수도 있다.

[0031] 목적지 디바이스 (116) 의 입력 인터페이스 (122) 는 컴퓨터 관독가능 매체 (110) (예를 들어, 통신 매체, 저장 디바이스 (112), 파일 서버 (114) 등) 로부터 인코딩된 비디오 비트스트림을 수신한다. 인코딩된 비디오 비트스트림은 비디오 블록들 또는 다른 코딩된 유닛들 (예를 들어, 슬라이스들, 픽처들, 픽처들의 그룹들, 시퀀스들 등) 의 프로세싱 및/또는 특징들을 기술하는 값들을 갖는 신택스 엘리먼트들과 같은, 비디오 디코더 (300) 에 의해 또한 사용되는, 비디오 인코더 (200) 에 의해 정의된 시그널링 정보를 포함할 수도 있다. 디스플레이 디바이스 (118) 는 디코딩된 비디오 데이터의 디코딩된 픽처들을 사용자에게 디스플레이한다. 디스플레이 디바이스 (118) 는 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디스플레이 디바이스를 포함할 수도 있다.

[0032] 도 1 에 도시되지는 않았지만, 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 각각 오디오 인코더 및/또는 오디오 디코더와 통합될 수도 있고, 공통 데이터 스트림에서 오디오 및 비디오 양자 모두를 포함하는 멀티플렉싱된 스트림들을 핸들링하기 위해, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및/또는 소프트웨어를 포함할 수도 있다. 적용가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대 사용자 데이터그램 프로토콜 (UDP) 에 따를 수도 있다.

[0033] 비디오 인코더 (200) 및 비디오 디코더 (300) 각각은 다양한 적합한 인코더 및/또는 디코더 회로부, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들 중 임의의 것으로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어에서 구현되는 경우, 디바이스는 적합한, 비일시적 컴퓨터 관독가능 매체에 소프트웨어에 대한 명령들을 저장하고, 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있는데, 이들 중 어느 하나는 개별의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다. 비디오 인코더 (200) 및/또는 비디오 디코더 (300) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 이를 테면 셀룰러 전화기를 포함할 수도 있다.

[0034] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 고 효율 비디오 코딩 (HEVC) 으로도 또한 지칭되는 ITU-T H.265 와 같은 비디오 코딩 표준 또는 그에 대한 확장들, 예컨대 멀티-뷰 및/또는 스케일러블 비디오 코딩 확장들에 따라 동작할 수도 있다. 대안적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 공동 탐구 테스트 모델 (JEM) 또는 다용도 비디오 코딩 (VVC) 으로도 또한 지칭되는 ITU-T H.266 과 같은 다른 독점 또는 산업 표준들에 따라 동작할 수도 있다. VVC 표준의 최근 초안은 Bross 등의 "Versatile Video Coding (Draft 4)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 13th Meeting: Marrakech, MA, 9-18 January 2019, JVET-M1001-v5 (이하 "VVC Draft 4") 에 설명되어 있다. 그러나, 본 개시의 기법들은 임의의 특정 코딩 표준에 한정되지 않는다.

[0035] 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 픽처들의 블록 기반 코딩을 수행할 수도 있다. 용어 "블록" 은 일반적으로 프로세싱될 (예를 들어, 인코딩될, 디코딩될, 또는 다르게는 인코딩 및/또는 디코딩 프로세스에서 사용될) 데이터를 포함하는 구조를 지칭한다. 예를 들어, 블록은 루미넌스 및/또는 크로미넌스 데이터의 샘플들의 2 차원 행렬을 포함할 수도 있다. 일반적으로, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 YUV (예를 들어, Y, Cb, Cr) 포맷으로 표현된 비디오 데이터를 코딩할 수도 있다. 즉, 픽처의 샘플들에 대한 적색, 녹색, 및 청색 (RGB) 데이터를 코딩하는 것보다는, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 및 크로미넌스 컴포넌트들을 코딩할 수도 있고, 여기서 크로미넌스 컴포넌트들은 적색 색조 및 청색 색조 크로미넌스 컴포넌트들 양자 모두를 포함할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 인코딩 이전에 수신된 RGB 포맷팅된 데이터를 YUV 표현으로 변환하고, 비디오 디코더 (300) 는 YUV 표현을 RGB 포맷으로 변환한다. 대안적으로, 프리- 및 포스트-프로세싱 유닛들 (미도시) 이 이들 변환들을 수행할 수도 있다.

[0036] 본 개시는 일반적으로 픽처의 데이터를 인코딩 또는 디코딩하는 프로세스를 포함하는 픽처들의 코딩 (예를

들어, 인코딩 및 디코딩) 을 참조할 수도 있다. 유사하게, 본 개시는, 예를 들어, 예측 및/또는 잔차 코딩과 같은, 블록들에 대한 데이터를 인코딩 또는 디코딩하는 프로세스를 포함하는 픽처의 블록들의 코딩을 참조할 수도 있다. 인코딩된 비디오 비트스트림은 일반적으로 코딩 결정들 (예를 들어, 코딩 모드들) 및 픽처들의 블록들로의 파티셔닝을 나타내는 신택스 엘리먼트들에 대한 일련의 값들을 포함한다. 따라서, 픽처 또는 블록을 코딩하는 것에 대한 참조들은 일반적으로 픽처 또는 블록을 형성하는 신택스 엘리먼트들에 대한 코딩 값들로서 이해되어야 한다.

[0037] HEVC 는 코딩 유닛들 (CU들), 예측 유닛들 (PU들), 및 변환 유닛들 (TU들) 을 포함한 다양한 블록들을 정의한다. HEVC 에 따르면, 비디오 코더 (예컨대 비디오 인코더 (200)) 는 쿼드트리 구조에 따라 코딩 트리 유닛 (CTU) 을 CU들로 파티셔닝한다. 즉, 비디오 코더는 CTU들 및 CU들을 4 개의 동일한, 오버랩하지 않는 정사각형들로 파티셔닝하고, 쿼드트리의 각각의 노드는 0 개 또는 4 개의 자식 노드들을 갖는다. 자식 노드들이 없는 노드들은 "리프 노드들" 로 지칭될 수도 있고, 그러한 리프 노드들의 CU들은 하나 이상의 PU들 및/또는 하나 이상의 TU들을 포함할 수도 있다. 비디오 코더는 PU들 및 TU들을 추가로 파티셔닝할 수도 있다. 예를 들어, HEVC 에서, 잔차 쿼드트리 (RQT) 는 TU들의 파티셔닝을 나타낸다. HEVC 에서, PU들은 인터-예측 데이터를 나타내는 한편, TU들은 잔차 데이터를 나타낸다. 인트라-예측되는 CU들은 인트라-모드 표시와 같은 인트라-예측 정보를 포함한다.

[0038] 다른 예로서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 JEM 또는 VVC 에 따라 동작하도록 구성될 수도 있다. JEM 또는 VVC 에 따르면, 비디오 코더 (이를 테면 비디오 인코더 (200)) 는 픽처를 복수의 코딩 트리 유닛들 (CTU들) 로 파티셔닝한다. 비디오 인코더 (200) 는 쿼드트리 바이너리 트리 (QTBT) 구조 또는 멀티타입 트리 (MTT) 구조와 같은 트리 구조에 따라 CTU 를 파티셔닝할 수도 있다. QTBT 구조는 HEVC 의 CU들, PU들, 및 TU들 간의 분리와 같은 다중 파티션 타입들의 개념들을 제거한다. QTBT 구조는 2 개의 레벨들: 쿼드트리 파티셔닝에 따라 파티셔닝된 제 1 레벨, 및 이진 트리 파티셔닝에 따라 파티셔닝된 제 2 레벨을 포함한다. QTBT 구조의 루트 노드는 CTU 에 대응한다. 바이너리 트리들의 리프 노드들은 코딩 유닛들 (CU들) 에 대응한다.

[0039] MTT 파티셔닝 구조에서, 블록들은 쿼드트리 (QT) 파티션, 바이너리 트리 (BT) 파티션, 및 하나 이상의 타입들의 트리플 트리 (TT) 파티션들을 사용하여 파티셔닝될 수도 있다. 트리플 트리 파티션은 블록이 3 개의 서브-블록들로 스플리팅되는 파티션이다. 일부 예들에서, 트리플 트리 파티션은 센터를 통해 원래의 블록을 분할하지 않고 블록을 3 개의 서브-블록들로 분할한다. MTT (예를 들어, QT, BT, 및 TT) 에서의 파티셔닝 타입들은 대칭적 또는 비대칭적일 수도 있다.

[0040] 일부 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 루미넌스 및 크로미넌스 성분들의 각각을 나타내기 위해 단일 QTBT 또는 MTT 구조를 사용할 수도 있는 한편, 다른 예들에서, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 2 개 이상의 QTBT 또는 MTT 구조들, 이를 테면 루미넌스 성분을 위한 하나의 QTBT/MTT 구조 및 양자의 크로미넌스 성분들을 위한 다른 QTBT/MTT 구조 (또는 각각의 크로미넌스 성분들을 위한 2 개의 QTBT/MTT 구조들) 를 사용할 수도 있다.

[0041] 비디오 인코더 (200) 및 비디오 디코더 (300) 는 HEVC 당 쿼드트리 파티셔닝, QTBT 파티셔닝, MTT 파티셔닝, 또는 다른 파티셔닝 구조들을 사용하도록 구성될 수도 있다. 설명의 목적을 위해, 본 개시의 기법들의 기재는 QTBT 파티셔닝에 대하여 제시된다. 그러나, 본 개시의 기법들은 또한, 쿼드트리 파티셔닝, 또는 다른 타입들의 파티셔닝도 물론 사용하도록 구성된 비디오 코더들에 적용될 수도 있음을 이해해야 한다.

[0042] 본 개시는 수직 및 수평 차원들에 관하여 (CU 또는 다른 비디오 블록과 같은) 블록의 샘플 차원들을 지칭하기 위해 "NxN" 및 "N 바이 N", 예를 들어 16x16 샘플들 또는 16 바이 16 샘플들을 상호교환가능하게 사용할 수도 있다. 일반적으로, 16x16 CU 는 수직 방향에서 16 샘플들 (y = 16) 그리고 수평 방향에서 16 샘플들 (x = 16) 을 가질 것이다. 마찬가지로, NxN CU 는 일반적으로 수직 방향에서 N 샘플들 및 수평 방향에서 N 샘플들을 갖고, 여기서 N 은 음이 아닌 정수 값을 나타낸다. CU 에서의 샘플들은 행들 및 열들로 배열될 수도 있다. 더욱이, CU들은 수직 방향에서와 동일한 수의 샘플들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, CU들은 N×M 개의 샘플들을 포함할 수도 있고, 여기서 M 은 반드시 N 과 동일한 것은 아니다.

[0043] 비디오 인코더 (200) 는 예측 및/또는 잔차 정보를 나타내는 CU들에 대한 비디오 데이터, 및 다른 정보를 인코딩한다. 예측 정보는 CU 에 대한 예측 블록을 형성하기 위하여 CU 가 어떻게 예측될지를 표시한다. 잔차 정보는 일반적으로 인코딩 이전의 CU 의 샘플들과 예측 블록 사이의 샘플 별 (sample-by-sample) 차이들을

나타낸다.

- [0044] CU 를 예측하기 위해, 비디오 인코더 (200) 는 일반적으로 인터 예측 또는 인트라 예측을 통해 CU 에 대한 예측 블록을 형성할 수도 있다. 인터 예측은 일반적으로 이전에 코딩된 픽처의 데이터로부터 CU 를 예측하는 것을 지칭하는 반면, 인트라 예측은 일반적으로 동일한 픽처의 이전에 코딩된 데이터로부터 CU 를 예측하는 것을 지칭한다. 인터 예측을 수행하기 위해, 비디오 인코더 (200) 는 하나 이상의 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 비디오 인코더 (200) 는 일반적으로 CU 와 레퍼런스 블록 사이의 차이들의 관점에서, CU 에 밀접하게 매칭하는 레퍼런스 블록을 식별하기 위해 모션 탐색을 수행할 수도 있다. 비디오 인코더 (200) 는 절대 차이의 합 (sum of absolute difference; SAD), 제곱 차이들의 합 (sum of squared differences; SSD), 평균 절대 차이 (mean absolute difference; MAD), 평균 제곱 차이들 (mean squared differences; MSD), 또는 레퍼런스 블록이 현재 CU 에 밀접하게 매칭하는지 여부를 결정하기 위한 다른 그러한 차이 계산들을 사용하여 차이 메트릭을 계산할 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 단방향 예측 또는 양방향 예측을 사용하여 현재 CU 를 예측할 수도 있다.
- [0045] JEM 및 VVC 의 일부 예들은 또한, 인터-예측 모드로 고려될 수도 있는 아핀 모션 보상 모드를 제공한다. 아핀 모션 보상 모드에서, 비디오 인코더 (200) 는 줌 인 또는 아웃, 회전, 원근 모션 (perspective motion), 또는 다른 불규칙한 모션 타입들과 같은 비-병진 모션을 나타내는 2 개 이상의 모션 벡터들을 결정할 수도 있다.
- [0046] 인트라 예측을 수행하기 위해, 비디오 인코더 (200) 는 예측 블록을 생성하기 위해 인트라 예측 모드를 선택할 수도 있다. JEM 및 VVC 의 일부 예들은 다양한 방향 모드들 뿐만 아니라 평면 모드 및 DC 모드를 포함하여 67 개의 인트라 예측 모드들을 제공한다. 일반적으로, 비디오 인코더 (200) 는 현재 블록의 샘플들을 예측할 현재 블록 (예를 들어, CU 의 블록) 에 대한 이웃하는 샘플들을 기술하는 인트라 예측 모드를 선택한다. 그러한 샘플들은 일반적으로, 비디오 인코더 (200) 가 래스터 스캔 순서로 (왼쪽에서 오른쪽으로, 상단에서 하단으로) CTU들 및 CU들을 코딩하는 것을 가정하여, 현재 블록과 동일한 픽처에서 현재 블록의 상측, 상측 및 좌측에, 또는 좌측에 있을 수도 있다.
- [0047] 비디오 인코더 (200) 는 현재 블록에 대한 예측 모드를 나타내는 데이터를 인코딩한다. 예를 들어, 인터-예측 모드들에 대해, 비디오 인코더 (200) 는 다양한 이용가능한 인터-예측 모드들 중 어느 것이 사용되는지를 나타내는 데이터 뿐만 아니라, 대응하는 모드에 대한 모션 정보를 인코딩할 수도 있다. 단방향 또는 양방향 인터-예측을 위해, 예를 들어, 비디오 인코더 (200) 는 어드밴스드 모션 벡터 예측 (AMVP) 또는 병합 모드를 사용하여 모션 벡터들을 인코딩할 수도 있다. 비디오 인코더 (200) 는 유사한 모드들을 사용하여 아핀 모션 보상 모드에 대한 모션 벡터들을 인코딩할 수도 있다.
- [0048] 블록의 인트라-예측 또는 인터-예측과 같은 예측에 후속하여, 비디오 인코더 (200) 는 블록에 대한 잔차 데이터를 계산할 수도 있다. 잔차 블록과 같은 잔차 데이터는 대응하는 예측 모드를 사용하여 형성되는, 블록과 블록에 대한 예측 블록 사이의 샘플 별 차이들을 나타낸다. 비디오 인코더 (200) 는 샘플 도메인 대신에 변환 도메인에서 변환된 데이터를 생성하기 위해, 잔차 블록에 하나 이상의 변환들을 적용할 수도 있다. 예를 들어, 비디오 인코더 (200) 는 이산 코사인 변환 (DCT), 정수 변환, 웨이브릿 변환, 또는 개념적으로 유사한 변환을 잔차 비디오 데이터에 적용할 수도 있다. 추가적으로, 비디오 인코더 (200) 는 MDNSST (mode-dependent non-separable secondary transform), 신호 의존적 변환, Karhunen-Loeve 변환 (KLT) 등과 같은 제 1 변환에 후속하는 2 차 변환을 적용할 수도 있다. 비디오 인코더 (200) 는 하나 이상의 변환들의 적용에 이어 변환 계수들을 생성한다.
- [0049] 상기 언급된 바와 같이, 변환 계수들을 생성하기 위한 임의의 변환들에 이어, 비디오 인코더 (200) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로, 변환 계수들이 그 변환 계수들을 나타내는데 사용된 데이터의 양을 가능하게는 감소시키도록 양자화되어 추가 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스를 수행함으로써, 비디오 인코더 (200) 는 변환 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, 비디오 인코더 (200) 는 양자화 동안 n-비트 값을 m-비트 값으로 라운딩 다운할 수도 있고, 여기서 n 은 m 보다 크다. 일부 예들에서, 양자화를 수행하기 위해, 비디오 인코더 (200) 는 양자화될 값의 비트단위 우측-시프트를 수행할 수도 있다.
- [0050] 양자화에 후속하여, 비디오 인코더 (200) 는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 포함한 2 차원 행렬로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 벡터의 전방에 더 높은 에너지 (및 따라서 더 낮은 주파수) 변환 계수들을 배치하고 벡터의 후방에 더 낮은 에너지 (및 따라서 더 높은 주파수) 변환 계수들을 배치

하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (200) 는 양자화된 변환 계수들을 스캔하기 위해 미리정의된 스캔 순서를 활용하여 직렬화된 벡터를 생성한 후, 벡터의 양자화된 변환 계수들을 엔트로피 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (200) 는 적응적 스캔을 수행할 수도 있다. 1 차원 벡터를 형성하기 위해 양자화된 변환 계수들을 스캐닝한 후, 비디오 인코더 (200) 는, 예를 들어, 컨텍스트-적응적인 산술 코딩 (CABAC) 에 따라, 1 차원 벡터로 변환 계수들을 표현하는 신택스 엘리먼트들을 엔트로피 인코딩할 수 있다. 비디오 인코더 (200) 는 또한, 비디오 데이터를 디코딩하는데 있어서 비디오 디코더 (300) 에 의한 사용을 위해 인코딩된 비디오 데이터와 연관된 메타데이터를 기술하는 신택스 엘리먼트들에 대한 값들을 엔트로피 인코딩할 수도 있다.

[0051] CABAC 을 수행하기 위해, 비디오 인코더 (200) 는 송신될 심볼에 컨텍스트 모델 내의 컨텍스트를 할당할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃 값들이 제로 값인지 여부와 관련될 수도 있다. 확률 결정은 심볼에 할당된 컨텍스트에 기초할 수도 있다.

[0052] 비디오 인코더 (200) 는 신택스 데이터, 예컨대 블록 기반 신택스 데이터, 픽처 기반 신택스 데이터, 및 시퀀스 기반 신택스 데이터를, 비디오 디코더 (300) 에, 예를 들어, 픽처 헤더, 블록 헤더, 슬라이스 헤더, 또는 다른 신택스 데이터, 예컨대 시퀀스 파라미터 세트 (SPS), 픽처 파라미터 세트 (PPS), 또는 비디오 파라미터 세트 (VPS) 에서 추가로 생성할 수도 있다. 비디오 디코더 (300) 는 마찬가지로 대응하는 비디오 데이터를 디코딩하는 방법을 결정하기 위해 그러한 신택스 데이터를 디코딩할 수도 있다.

[0053] 이러한 방식으로, 비디오 인코더 (200) 는 인코딩된 비디오 데이터, 예를 들어 픽처의 블록들 (예를 들어, CU들) 로의 파티셔닝을 기술하는 신택스 엘리먼트들 및 블록들에 대한 예측 및/또는 잔차 정보를 포함하는 비트스트림을 생성할 수도 있다. 궁극적으로, 비디오 디코더 (300) 는 비트스트림을 수신하고 인코딩된 비디오 데이터를 디코딩할 수도 있다.

[0054] 일반적으로, 비디오 디코더 (300) 는 비트스트림의 인코딩된 비디오 데이터를 디코딩하기 위해 비디오 인코더 (200) 에 의해 수행되는 것과 상반되는 프로세스를 수행한다. 예를 들어, 비디오 디코더 (300) 는 비디오 인코더 (200) 의 CABAC 인코딩 프로세스와 실질적으로 유사하지만, 상반되는 방식으로 CABAC 을 사용하여 비트스트림의 신택스 엘리먼트들에 대한 값들을 디코딩할 수도 있다. 신택스 엘리먼트들은 픽처의 CTU들로의 파티셔닝 정보, 및 QTBT 구조와 같은 대응하는 파티션 구조에 따른 각각의 CTU 의 파티셔닝을 정의하여, CTU 의 CU들을 정의할 수도 있다. 신택스 엘리먼트들은 비디오 데이터의 블록들 (예를 들어, CU들) 에 대한 예측 및 잔차 정보를 추가로 정의할 수도 있다.

[0055] 비디오 디코더 (300) 는 블록에 대한 잔차 블록을 재생하기 위해 블록의 양자화된 변환 계수들을 역 양자화 및 역 변환할 수도 있다. 비디오 디코더 (300) 는 시그널링된 예측 모드 (인트라- 또는 인터-예측) 및 관련된 예측 정보 (예를 들어, 인터-예측을 위한 모션 정보) 를 사용하여 블록에 대한 예측 블록을 형성한다. 비디오 디코더 (300) 는 그 후 예측 블록과 잔차 블록을 (샘플 별 단위로) 결합하여 원래의 블록을 재생할 수도 있다. 비디오 디코더 (300) 는 블록의 경계들을 따라 시각적 아티팩트들을 감소시키기 위해 디블로킹 프로세스를 수행하는 것과 같은 추가적인 프로세싱을 수행할 수도 있다.

[0056] 상기 언급된 바와 같이, 비디오 인코더 (200) 및 비디오 디코더 (300) 는 신택스 엘리먼트들의 값들에 CABAC 인코딩 및 디코딩을 적용할 수 있다. 신택스 엘리먼트에 CABAC 인코딩을 적용하기 위해, 비디오 인코더 (200) 는 신택스 엘리먼트의 값을 이진화하여 "빈들" 로서 지칭되는 일련의 하나 이상의 비트들을 형성할 수도 있다. 또한, 비디오 인코더 (200) 는 코딩 컨텍스트를 식별할 수도 있다. 코딩 컨텍스트는 특정 값들을 갖는 빈들의 확률들을 식별할 수 있다. 실례로, 코딩 컨텍스트는 0-값의 빈을 코딩하는 0.7 의 확률 및 1-값의 빈을 코딩하는 0.3 의 확률을 나타낼 수도 있다. 코딩 컨텍스트를 식별한 후에, 비디오 인코더 (200) 는 간격을 하위 서브-간격 (lower sub-interval) 및 상위 서브-간격 (upper sub-interval) 으로 분할할 수도 있다. 서브-간격들 중 하나는 값 0 과 연관될 수도 있고, 다른 서브-간격은 값 1 과 연관될 수도 있다. 서브-간격들의 폭들은 식별된 코딩 컨텍스트에 의해 연관된 값들에 대해 표시된 확률들에 비례할 수도 있다. 신택스 엘리먼트의 빈이 하위 서브-간격과 연관된 값을 가지는 경우에, 인코딩된 값은 하위 서브-간격의 하위 경계와 동일할 수도 있다. 신택스 엘리먼트의 동일 빈이 상위 서브-간격과 연관된 값을 가지는 경우에, 인코딩된 값은 상위 서브-간격의 하위 경계와 동일할 수도 있다. 신택스 엘리먼트의 다음 빈을 인코딩하기 위해, 비디오 인코더 (200) 는 인코딩되는 비트의 값과 연관된 서브-간격인 간격으로 이들 단계들을 반복할 수도 있다. 비디오 인코더 (200) 가 다음 빈에 대해 이들 단계들을 반복할 때, 비디오 인코더 (200) 는 인코딩되는 빈들의 실제 값들 및 식별된 코딩 컨텍스트에 의해 표시된 확률들에 기초하여 변경된 확률들을 이용할

수도 있다.

[0057] 비디오 디코더 (300) 가 신택스 엘리먼트의 값에 대해 CABAC 디코딩을 수행할 때, 비디오 디코더 (300) 는 코딩 컨텍스트를 식별할 수도 있다. 비디오 디코더 (300) 는 그 다음, 간격을 하위 서브-간격과 상위 서브-간격으로 분할할 수도 있다. 서브-간격들 중 하나는 값 0 과 연관될 수도 있고, 다른 서브-간격은 값 1 과 연관될 수도 있다. 서브-간격들의 폭들은 식별된 코딩 컨텍스트에 의해 연관된 값들에 대해 표시된 확률들에 비례할 수도 있다. 인코딩된 값이 하위 서브-간격 내인 경우에, 비디오 디코더 (300) 는 하위 서브-간격과 연관된 값을 갖는 빈을 디코딩할 수도 있다. 인코딩된 값이 상위 서브-간격 내인 경우에, 비디오 디코더 (300) 는 상위 서브-간격과 연관된 값을 갖는 빈을 디코딩할 수도 있다. 신택스 엘리먼트의 다음 빈을 디코딩하기 위해, 비디오 디코더 (300) 는 인코딩된 값을 값을 포함하는 서브-간격인 간격으로 이들 단계들을 반복할 수도 있다. 비디오 디코더 (300) 가 다음 빈에 대해 이들 단계들을 반복할 때, 비디오 디코더 (300) 는 디코딩되는 빈들 및 식별된 코딩 컨텍스트에 의해 표시된 확률들에 기초하여 변경된 확률들을 이용할 수도 있다. 비디오 디코더 (300) 는 그 다음, 신택스 엘리먼트의 값을 복원하기 위해 빈들을 이진화해제할 수도 있다.

[0058] 일부 경우들에서, 비디오 인코더 (200) 는 바이패스 CABAC 코딩을 이용하여 빈들을 인코딩할 수 있다. 빈에 대해 정규 CABAC 코딩을 수행하기보다는 빈에 대해 바이패스 CABAC 코딩을 수행하는 것이 계산적으로 덜 비용이 들 수도 있다. 또한, 바이패스 CABAC 코딩을 수행하는 것은 더 높은 정도의 병렬화 (parallelization) 및 스트루풋 (throughput) 을 허용할 수도 있다. 바이패스 CABAC 코딩을 이용하여 인코딩된 빈들은 "바이패스 빈들" 로서 지칭될 수도 있다. 바이패스 빈들을 함께 그룹핑하는 것은 비디오 인코더 (200) 및 비디오 디코더 (300) 의 스트루풋을 증가시킬 수도 있다. 바이패스 CABAC 코딩 엔진은 단일 사이클에서 수개의 빈들을 코딩 가능할 수도 있는 반면, 정규 CABAC 코딩 엔진은 한 사이클에서 오직 단일 빈만을 코딩 가능할 수도 있다. 이들 바이패스 CABAC 코딩 엔진은 그 바이패스 CABAC 코딩 엔진이 컨텍스트들을 선택하지 않고 양 심볼들 (0 및 1) 에 대해 1/2 의 확률을 가정할 수도 있기 때문에 더 단순할 수도 있다. 결과적으로, 바이패스 CABAC 코딩에서, 간격들은 직접 절반으로 분할된다.

[0059] 또한, 상술한 바와 같이, 비트스트림은 비디오 데이터 및 연관된 데이터의 인코딩된 화상의 표현을 포함할 수도 있다. 연관된 데이터는 파라미터 세트를 포함할 수도 있다. 네트워크 추상화 계층 (NAL) 유닛들은 비디오 파라미터 세트 (VPS) 들, 시퀀스 파라미터 세트 (SPS) 들, 및 픽처 파라미터 세트 (PPS) 들에 대해 원시 바이트 시퀀스 페이로드 (RBSP) 를 캡슐화할 수 있다. VPS 는 제로 이상의 전체 코딩된 비디오 시퀀스 (CVS) 에 적용되는 신택스 엘리먼트들을 포함하는 신택스 구조이다. SPS 는 또한 제로 이상의 전체 CVS 에 적용되는 신택스 엘리먼트들을 포함하는 신택스 구조이다. SPS 는 SPS 가 활성화될 때 활성화된 VPS 를 식별하는 신택스 엘리먼트를 포함할 수도 있다. 따라서, VPS 의 신택스 엘리먼트는 SPS 의 신택스 엘리먼트보다 더 일반적으로 적용 가능할 수도 있다. PPS 는 제로 이상의 코딩된 픽처들에 적용되는 신택스 엘리먼트들을 포함하는 신택스 구조이다. PPS 는 PPS 가 활성화될 때 활성화된 SPS 를 식별하는 신택스 엘리먼트를 포함할 수도 있다. 슬라이스의 슬라이스 헤더는 슬라이스 세그먼트가 코딩되고 있을 때 활성화된 PPS 를 표시하는 신택스 엘리먼트를 포함할 수도 있다.

[0060] 컬러 비디오는 멀티미디어 시스템들에서 필수적인 역할을 하며, 여기서, 다양한 컬러 공간들은 컬러를 효율적으로 나타내기 위해 사용된다. 컬러 공간은 다중의 컴포넌트들을 사용한 수치 값들로 컬러를 명시한다. 대중적인 컬러 공간은 RGB 컬러 공간이며, 여기서, 컬러는 3원색 컴포넌트 값들: 적색, 녹색 및 청색의 조합으로서 표현된다.

[0061] 컬러 비디오 압축을 위해, YCbCr 색 공간이 널리 사용되었다. A. Ford 및 A. Roberts, "색상 공간 변환", University of Westminster, London, Tech. Rep., Aug. 1998 을 참조하라. YCbCr 은 RGB 컬러 공간으로부터 선형 변환을 통해 변환될 수 있고, 상이한 컴포넌트들 간의 리던던시, 즉, 크로스-컴포넌트 리던던시는 YCbCr 컬러 공간에서 현저히 감소된다. YCbCr 의 하나의 이점은, Y 신호가 루미넌스 (루마) 정보를 전달할 때 흑백 TV 와의 역방향 호환성이다. 부가적으로, 크로미넌스 (크로마) 대역폭은, RGB 에서의 서브샘플링보다 현저하게 더 적은 주관적 임팩트를 갖는 4:2:0 크로마 샘플링 포맷에서 Cb 및 Cr 컴포넌트들을 서브샘플링함으로써 감소될 수 있다. 이들 이점들 때문에, YCbCr 은 비디오 압축에 있어서 가장 중요한 컬러 공간이 되었다. 비디오 압축에 있어서 사용되는 YCoCg 와 같은 다른 컬러 공간들이 또한 존재한다. 본 문서에 있어서, 사용된 실제 컬러 공간에 무관하게, 비디오 압축 스킴에 있어서 3개의 컬러 컴포넌트들을 나타내기 위해 Y, Cb, Cr 이 사용된다. 비록 크로스 컴포넌트 리던던시가 YCbCr 컬러 공간에서 현저히 감소되더라도, 3 개 컬러 컴포넌트들 간의 상관은 여전히 존재한다. 특히 4:4:4 크로마 포맷 비디오의 코딩을 위

해, 상관을 추가로 감소시킴으로써 비디오 코딩 성능을 개선하기 위해, 다양한 방법들이 연구되었다. B. C. Song, Y. G. Lee, 및 N. H. Kim, "Block adaptive inter-color compensation algorithm for RGB 4:4:4 video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 10, pp. 1447-1451, Oct. 2008 에 기술된 바와 같이, 스케일 인자 및 오프셋은 재구성된 루마 신호로부터 크로마 신호를 예측하기 위해 각 블록에 사용된다.

[0062] W. Pu, W.-S. Kim, J. Chen, J. Sole, M. Karczewicz, "RCE1: Descriptions and Results for Experiments 1, 2, 3, and 4", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-00202, Geneva, Nov. 2013 에서, 크로스-컴포넌트 예측 (Cross-Component Prediction: CCP) 가 잔차 도메인에서 수행된다. 크로마 잔차 신호는 인코더 측에서 재구성된 루마 잔차 신호를 사용하여 다음과 같이 예측되고:

[0063]
$$\Delta r_c(x, y) = r_c(x, y) - (\alpha \times r'_l(x, y)) \gg 3 \quad (1)$$

[0064] 크로마 잔차 신호는 디코더 측에서 다음과 같이 디코더측에서 보상되며:

[0065]
$$r'_c(x, y) = \Delta r'_c(x, y) + (\alpha \times r'_l(x, y)) \gg 3 \quad (2)$$

[0066] 여기서, $r_c(x, y)$ and $r'_c(x, y)$ 는 위치 (x, y) 에서의 원래의 및 재구성된 크로마 잔차 샘플들을 나타낸다. $\Delta r_c(x, y)$ 및 $\Delta r'_c(x, y)$ 는 크로스 성분 예측 후 최종 크로마 잔차 샘플들을 나타낸다. $r'_l(x, y)$ 는 재구성된 루마 잔차 샘플 값이다. 가중 인자 α 는 HEVC의 각 크로마 변환 유닛에 대한 비트 스트림에서 명시적으로 시그널링된다. 이 크로스 성분 잔차 예측 방법은 인트라 예측 잔차 및 인터 예측 잔차 모두에 대해 수행되는, 4:4:4 크로마 샘플링 형식 비디오 코딩을 위한 HEVC 표준의 형식 및 범위 확장에 채택되었다. 이 크로스 성분 잔차 예측 방법은 4:2:0 비디오 형식 코딩에도 제안되었지만 이 크로스 성분 잔차 예측 방법은 제한된 코딩 이점 때문에 채택되지 않는다. 루마를 사용하여 크로마 성분을 예측하는 것 외에도, 예측이 두 크로마 성분 사이에 적용될 수 있다는 것, 즉 재구성된 조기 코딩된 Cb 잔차가 Cr 잔차를 예측하는 데 사용된다는 것도 제안되었다. A. Khairat, T. Nguyen, M. Siekmann, D. Marpe, "Non-RCE1: Extended Adaptive Inter-Component Prediction", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-00150, Geneva, Nov. 2013 을 참조하라. 2016년 7월 28일자로 공개된 미국 특허 공개 번호 2016/0219283, 2016년 9월 22일자로 공개된 2016/0277762, 2018년 3월 15일자로 공개된 2018/0077426, 2018년 6월 21일자로 공개된 2018/0176594, 2019년 1월 3일자로 공개된 2019/0007688 은 또한 교차 구성 예측의 양태들을 설명한다.

[0067] 4:2:0 크로마 비디오 코딩에 있어서, 리니어 모델 (Linear Model; LM) 예측 모드로 명명된 방법이 HEVC 표준의 개발 동안 잘 연구되었다. J. Chen, V. Seregin, W.-J. Han, J.-S. Kim, B.-M. Joen, "CE6.a.4: Chroma intra prediction by reconstructed luma samples", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-E266, Geneva, 16-23 March, 2011을 참조하라. LM 예측 모드에 있어서, 크로마 샘플들은 다음과 같은 선형 모델을 사용함으로써 동일한 블록의 복원된 루마 샘플들에 기초하여 예측된다:

[0068]
$$pred_c(i, j) = \alpha \cdot rec_l(i, j) + \beta \quad (3)$$

[0069] 여기서, $pred_c(i, j)$ 는 블록에서의 크로마 샘플들의 예측을 나타내고, $rec_l(i, j)$ 는 동일한 블록의 다운 샘플링된 재구성된 루마 샘플들을 나타낸다. 파라미터들 α 및 β 는 현재 블록 주변의 이웃한 복원된 루마 및 크로마 샘플들 간의 회귀 (regression) 에러를 최소화함으로써 도출된다.

[0070]
$$E(\alpha, \beta) = \sum_i (y_i - (\alpha \cdot x_i + \beta))^2 \quad (4)$$

[0071] 파라미터들 α 및 β 는 다음과 같이 풀린다:

$$\alpha = \frac{N \sum x_i \cdot y_i - \sum x_i \cdot \sum y_i}{N \sum x_i \cdot x_i - \sum x_i \cdot \sum x_i} \quad (5)$$

$$\beta = (\sum y_i - \alpha \cdot \sum x_i) / N \quad (6)$$

[0072]

[0073] 여기서 x_i 는 다운 샘플링된 재구성된 루마 참조 샘플이며, y_i 는 재구성된 크로마 참조 샘플이며, N 은 사용된 참조 샘플의 수이다.

[0074] 인루프 재성형기가 제 13 차 JVET 회의에서 채택되었다. J. Lainema, "CE7 관련: Joint coding of chrominance residuals," 13th JVET Meeting, Marrakech, MA, Jan. 2019, JVET-M0305 (hereinafter, "JVET-M0305") and X. Xiu, P. T. Lu, P. Hanhart, R. Vanam, Y. He, Y. Ye, T. Lu, F. Pu, P. Yin, W. Husak, T. Chen, "Description of SDR, HDR, and 360° video coding technology proposal by InterDigital Communications and Dolby Laboratories", JVET-J0015, San Diego, USA, April 2018 을 참조하라.

[0075] 인루프 루마 재성형기는 한 쌍의 룩업 테이블(LUT)로 구현되지만, 그 두 LUT 중 하나만이 (구분적 선형 모델링에 의해) 시그널링될 필요가 있으며, 이는 다른 하나는 시그널링된 LUT로부터 계산될 수 있기 때문이다. 각 LUT는 10 비트 비디오용 1차원, 10비트, 1024 엔트리 매핑 테이블 (1D-LUT) 이다. 하나의 LUT는 입력 루마 코드 값 Y_i (원래 도메인) 을 변경된 값 $Y_r: Y_r = FwdLUT[Y_i]$ (재성형된 도메인) 으로 매핑하는 순방향 LUT

($FwdLUT$) 이다. 다른 LUT는 변경된 코드 값 Y_r (재성형된 도메인) 을 \hat{Y}_i (원래 도메인) 로 맵핑하는 역

LUT ($InvLUT$) 이다: $\hat{Y}_i = InvLUT[Y_r]$ (\hat{Y}_i 는 Y_i 의 재구성 값을 나타냄).

[0076] 도 2 는 인루프 루마 재성형기 (reshaper) 를 사용한 인트라 슬라이스 재구성의 예를 예시하는 블록도이다. 인트라 슬라이스의 경우, $InvLUT$ 만이 디코더에서 적용된다 (도 2). 도 2 에서, 회색 음영 블록은 재성형된 도메인의 신호들을 나타낸다: 루마 잔차, 예측된 인트라 루마, 및 재구성된 인트라 루마. 더욱 구체적으로, 도 2 의 예에 도시된 바와 같이, 비디오 디코더(300)는 CABAC 디코딩(CABAC⁻¹으로 표시됨), 역양자화(Q⁻¹로 표시됨) 및 역변환(T⁻¹로 표시됨) 을 수행할 수 있다 (150). 추가적으로, 비디오 디코더 (300) 는 루마 잔차 샘플들 (Y_{res}) 을 인트라 예측 샘플들 (Y'_{pred}) 에 가산함으로써 인트라 예측 (152) 및 재구성 (154) 을 수행할 수 있다. 게다가, 비디오 디코더 (300) 는 역 재성형 동작을 수행할 수 있다 (156). 비디오 디코더 (300) 는 역 재성형 동작의 출력에 하나 이상의 루프 필터(LF)를 적용할 수 있다 (158). 결과는 디코딩된 픽처 버퍼(DPB) 에 저장될 수 있다 (160).

[0077] 도 3 은 인루프 루마 재성형기를 사용한 인터 슬라이스 재구성의 예를 예시하는 블록도이다. 인터 슬라이스의 경우(도 3), $FwdLUT$ 그리고 $InvLUT$ 양자 모두가 적용된다. LUT 들은 인트라 슬라이스와 인터 슬라이스 양자 모두에 대해 루프 필터링 전에 적용된다. 도 3 에서, 회색 음영 블록은 재성형된 도메인의 신호들을 나타낸다: 루마 잔차, 예측된 인트라 루마, 및 재구성된 인트라 루마. 이 문서에서 용어 "슬라이스" 는 NAL 유닛에서 시그널링되는 코딩 트리 유닛들의 집합을 나타내기 위해 사용되지만, 그 논의는 또한 타일 그룹(VVC에서 사용) 과 같은 픽처들의 다른 공간적 파티셔닝에 적용된다는 것이 이해되어야 한다.

[0078] 인터 슬라이스(또는 타일 그룹)의 경우, $FwdLUT$ 는 원래 도메인의 모션 보상된 값들을 재성형된 도메인 ($FwdLUT[Y_{pred}]$) 에 매핑한다. $InvLUT$ 는 그런 다음 재성형된 도메인의 인터 재구성된 값을 원래 도메인의 인터 재구성된 값 (***) 에 매핑한다. 도 3 의 예에서, 인트라 예측은 항상 슬라이스 타입에 관계없이 재성형된 영역에서 수행된다. 이와 같은 구성으로, 선행 TU의 재구성 직후 인트라 예측이 시작될 수 있다.

[0079] 도 3 의 예에서, 비디오 디코더(300)는 CABAC 디코딩(CABAC⁻¹으로 표시됨), 역양자화(Q⁻¹로 표시됨) 및 역변환

(T^{-1} 로 표시됨) 을 수행할 수 있다 (170). 추가적으로, 비디오 디코더 (300) 는 블록에 대한 인트라 예측 (172) 또는 모션 보상 (184) 및 순방향 재성형 프로세스를 수행할 수 있다. 비디오 디코더 (300) 는 루마 잔차 샘플들 (Y_{res}) 을 예측 샘플들 (Y'_{pred}) 에 가산함으로써 재구성 (176) 을 수행할 수 있다. 도 3 의 예에서, 인트라 예측을 사용하여 블록을 예측하면, 예측 샘플(Y'_{pred})는 인트라 예측을 사용하여 예측된다 (172).

비디오 디코더(300)는 재성형된 도메인에서 인트라 예측을 수행한다. 그러나, 비디오 디코더(300)는 재성형되지 않은 도메인에서 인트라 예측을 수행한다. 따라서, 블록이 인트라 예측을 사용하여 예측되면, 비디오 디코더(300)는 모션 보상(184)에 의해 생성된 예측된 샘플들을 재성형된 도메인으로 변환하기 위해 모션 보상 (184) 및 순방향 재성형 동작(174)을 수행한다.

[0080] 게다가, 재구성 (176) 후에, 비디오 디코더 (300) 는 역 재성형 동작을 수행할 수 있다 (178). 비디오 디코더 (300) 는 역 재성형 동작의 출력에 하나 이상의 루프 필터(LF)를 적용할 수 있다 (180). 결과는 디코딩된 픽처 버퍼(DPB) 에 저장될 수 있다 (182). DPB로부터의 데이터는 후속 블록에서의 순방향 재성형 (174) 을 위해 후속적으로 사용될 수 있는 모션 보상(184)을 위해 사용될 수 있다.

[0081] 재성형기 모델 선택은 16개의 조각으로 구분적 선형 (piece-wise linear: PWL) 모델을 시그널링할 수 있다. 1024-엔트리 매핑 테이블은 16-조각 PWL 모델을 사용하여 선형 보간에 의해 구성된다. 개념적으로 PWL 모델은 다음과 같이 구현될 수 있다. x_1 , x_2 를 두 개의 입력 피벗 포인트라고 하고 y_1 , y_2 를 한 조각에 대한 그들의 대응하는 출력 피벗 포인트라고 하자 x_1 과 x_2 사이의 임의의 입력 값 x 에 대한 출력 값 y 는 다음 방정식으로 보간될 수 있다:

[0082]
$$y = ((y_2 - y_1) / (x_2 - x_1)) * (x - x_1) + y_1$$

[0083] 고정 소수점 구현에서 위의 방정식은 다음과 같이 다시 작성할 수 있다:

[0084]
$$y = ((m * x + 2^{FP_PREC-1}) \gg FP_PREC) + c$$

[0085] 여기서 m 은 스칼라, c 는 오프셋, FP_PREC 는 정밀도를 특정하는 상수 값이다.

[0086] 도 4 는 크로마 잔차 스케일링 (CRS) 을 사용한 인트라 모드 및 인트라 모드 재구성의 예를 예시하는 블록도이다. 크로마 잔차 스케일링 또는 CRS(도 4) 는 크로마 신호와의 루마 신호 상호작용을 보상한다. 크로마 잔차 스케일링은 TU 수준에서 적용된다.

[0087] 인코더 측:
$$C_{ResScale} = C_{Res} * C_{Scale} = C_{Res} / C_{ScaleInv} \quad (7)$$

[0088] 디코더 측:
$$C_{Res} = C_{ResScale} / C_{Scale} = C_{ResScale} * C_{ScaleInv} \quad (8)$$

[0089] C_{Res} 는 원래 크로마 잔차 신호이고 $C_{ResScale}$ 는 스케일링된 크로마 잔차 신호이다. C_{Scale} 는 FwdLUT 를 사용하여 계산되는 스케일링 인자이고, 그 역수 $C_{ScaleInv}$ 로 변환되어, 디코더 측에서 나눗셈 대신 곱셈을 수행하여, 구현 복잡성을 감소시킨다. 도 4 에서, 회색 음영 블록은 재성형된 도메인의 신호들을 나타낸다: 루마 잔차, 예측된 인트라 루마, 및 재구성된 인트라 루마. PWL 모델 하에서, 각 루마 PWL 범위의 임의의 입력 루마 값에 대해, 동일한 크로마 스케일 값이 사용된다. 따라서 루마 종속 크로마 스케일은 각 루마 코드 값 대신 각 루마 PWL 범위에 대해 계산되도록 단순화되고, $LUTcScaleInv[pieceIdx]$ (16-조각 입도) 에 저장된다.

인코더 측과 디코더 측 양자 모두에서의 스케일링 동작들은 다음 방정식을 사용하여 고정 소수점 정수 산술로 구현된다:

[0090]
$$c' = \text{sign}(c) * ((\text{abs}(c) * s + 2^{CSCALE_FP_PREC-1}) \gg CSCALE_FP_PREC)$$

[0091] 여기서 c 는 크로마 잔차, s 는 $cScaleInv[pieceIdx]$ 로부터의 크로마 잔차 스케일링 인자, $pieceIdx$ 는 TU의 대응하는 평균 루마 값에 의해 결정되며, $CSCALE_FP_PREC$ 는 정밀도를 특정하기 위한 상수 값이다.

[0092] $C_{ScaleInv}$ 의 값은 다음 단계들로 계산된다:

[0093] 1. 인트라 모드를 사용하는 경우, 인트라 예측된 루마 값들의 평균을 계산; 인트라 모드가 사용되는 경우, 순방향으로 재성형된 인트라 예측 루마 값의 평균을 계산. 즉, 평균 루마 값 $avgY'_{TU}$ 는 재성형된 도메인에서 계산

된다.

[0094] 2. $avgY'_{TU}$ 가 속하는 역 매핑 PWL에서 루마 범위의 인덱스 idx 를 구함.

[0095] 3. $C_{ScaleInv} = cScaleInv[idx]$

[0096] 도 4 의 예에서, 비디오 디코더(300)는 CABAC 디코딩(CABAC⁻¹), 역양자화(Q⁻¹) 및 역변환(T⁻¹) 을 수행할 수 있다 (240). 블록(240)의 출력은 비디오 데이터의 현재 블록에 대한 루마 잔차 데이터(Y_{res}) 및 스케일링된 크로마 데이터(C_{resScale}) 을 포함할 수 있다. 블록이 인트라 모드를 사용하여 인코딩되면, 비디오 디코더 (300) 는 현재 블록에 대한 인트라 예측 프로세스 (242) 를 수행하여 루마 예측 데이터 (Y'_{pred}) 를 생성할 수 있다.

대안적으로, 블록이 인터 모드를 사용하여 인코딩되면, 비디오 디코더 (300) 는 현재 블록에 대한 모션 보상 (252) 및 순방향 재성형 프로세스를 수행하여 루마 예측 데이터 (Y'_{pred}) 를 생성할 수 있다. 비디오 디코더 (300) 는 루마 잔차 데이터(Y_{res}) 및 예측 데이터(Y'_{pred}) 에 기초하여 루마 샘플들을 재구성하는 루마 재구성 프로세스(246)를 수행할 수 있다. 비디오 디코더 (300) 는 후속 블록들의 인트라 예측을 위해 재구성된 루마 데이터를 사용할 수 있다.

[0097] 또한, 도 4 의 예에서, 비디오 디코더 (300) 는 현재 블록에 대한 재구성된 루마 데이터에 대해 역 재성형 프로세스 (247) 를 수행할 수 있다. 비디오 디코더 (300) 는 역 재성형 동작의 출력에 하나 이상의 루프 필터 (LF)를 적용할 수 있다 (248). 비디오 디코더 (300) 는 그 결과들을 디코딩된 픽처 버퍼에 저장할 수 있다 (250). 디코딩된 픽처 버퍼의 데이터는 비디오 데이터의 후속 블록에 대한 모션 보상(252)을 위해 사용될 수 있다. 도 4 의 예에서, 비디오 디코더(300)는 또한 현재 변환 유닛에 대한 평균 루마 값(avgY'_{TU}) 및 크로마 잔차 스케일링 인자(C_{scaleInv}) 를 결정할 수 있다 (254). 비디오 디코더 (300) 는 위에서 설명된 바와 같이 크로마 잔차 스케일링 인자를 결정할 수 있다.

[0098] 도 4 의 예에서, 비디오 디코더 (300) 는 현재 블록에 대한 스케일링된 크로마 잔차 데이터를 스케일링하여 현재 블록에 대한 크로마 잔차 데이터 (C_{res}) 를 생성하는 크로마 잔차 스케일링 프로세스를 수행할 수 있다 (256). 비디오 디코더 (300) 는 스케일링된 크로마 잔차 데이터의 샘플들에 크로마 잔차 스케일링 팩터를 곱함으로써 크로마 잔차 스케일링 프로세스를 수행할 수 있다. 또한, 도 4 의 예에서, 비디오 디코더 (300) 는 현재 블록에 대한 크로마 예측 데이터 (C_{pred}) 를 생성하기 위해 인트라 예측 프로세스 (257) 또는 모션 보상 프로세스 (258) 를 수행할 수 있다. 비디오 디코더 (300) 는 현재 블록에 대한 크로마 잔차 데이터(C_{res}) 및 크로마 예측 데이터에 기초하여 현재 블록의 크로마 데이터를 재구성하는 재구성 프로세스를 수행할 수 있다 (259). 비디오 디코더 (300) 는 역 재성형 동작의 출력에 하나 이상의 루프 필터(LF)를 적용할 수 있다 (260). 비디오 디코더 (300) 는 그 결과의 데이터를 디코딩된 픽처 버퍼에 저장할 수 있다 (262).

[0099] Cb-Cr 크로마 잔차의 공동 코딩 방법은 JVET-M0305에 설명되어 있다. 이 방법에서는 단일 공동 잔차 블록을 사용하여 동일한 변환 유닛에서 Cb 및 Cr 블록 양자 모두의 잔차를 설명한다. 인코더 측에서, 알고리즘은 변환 및 양자화 프로세스에 대한 입력으로 양의 Cb 잔차 및 음의 Cr 잔차의 평균을 사용한다. 모드가 활성화되면, 단일 잔여 블록이 디코딩된다. Cr 블록의 잔차는 디코딩된 공동 잔차를 무효화하여 생성된다. 디코더 측에서, 공동 잔차 모드가 활성화되면, 표시된 공동 잔차가 Cb 예측 블록에 추가되고 Cr 예측 블록에서 공제된다. 단일 잔차가 두 블록의 잔차를 나타내기 위해 사용되기 때문에, 공동 색차 잔차 모드가 활성화될 때 크로마 양자화 파라미터(QP) 오프셋 파라미터는 2만큼 감소된다.

[0100] 결합 잔차 모드는 Cb 및 Cr 모두에 대한 코딩된 블록 플래그(cbf)가 참인 경우 비트스트림의 플래그로 표시된다. 공동 잔차에 대한 비트스트림 선택스 및 디코딩 프로세스는 VTM-3의 Cb 잔차의 그것들을 따른다.

J. Chen, Y. Ye, S. Kim, "Algorithm description for Versatile Video Coding and Test Model 3 (VTM3)," 12th JVET Meeting, Macau SAR, CN, Oct. 2018, JVET-L1002 를 참조하라.

[0101] 핵심 실험 테스트 CE7.1(H. Schwarz, M. Coban, C. Auyeung, "Description of Core Experiment 7 (CE7): Quantization and coefficient coding," 13th JVET Meeting, Marrakech, MA, Jan. 2019, JVET-M1027) 에서, 이 방법은 다음과 같이 인루프 크로마 잔차 스케일러 (CRS) 와 통합된다:

- [0102] • 인코더 측:
- [0103] • Cb 잔차: $(resCb(x,y) - resCr(x,y)) / 2 \rightarrow CRS \rightarrow T \rightarrow Q (Qp + jointCbCrQpOffset)$
- [0104] • Cr 잔차: 0
- [0105] • 디코더 측:
- [0106] • Cb 잔차: $invQ \rightarrow invT \rightarrow invCRS \rightarrow resCb(x,y)'$
- [0107] • Cr 잔차: $- resCb(x,y)'$
- [0108] 여기서, CRS 는 순방향 크로마 잔차 스케일러이며, invCRS 는 역 크로마 잔차 스케일러이고, T 는 순방향 변환이고, Q 는 순방향 양자화이며, invT 는 역변환이고, invQ 는 역양자화이다.
- [0109] 모델 기반 크로스 성분 잔차 예측은 비디오 디코더(300)에서 증가된 구현 비용을 암시하는 디코더 측에서의 모델 파라미터 추정을 요구하거나, 코딩 효율에 부정적인 영향을 미치고 비트스트림으로부터 파라미터를 디코딩하는 것을 요구하는, 모델 파라미터가 시그널링되어야 한다.
- [0110] 두 크로마 잔차를 비트스트림으로 코딩되는 단일 잔차로 결합하는 공동 Cb-Cr 코딩 방법은 두 크로마 채널의 비대칭 프로세싱으로 인해 크로마 아티팩트를 겪을 수 있다. 예를 들어, JVET-M0305에서, Cr 잔차는 공동 잔차의 역으로 가정되며, 이는 대략적인 근사치이다. 또한 이 방법은 변환 후 공동 잔차에 적용되는 양자화 파라미터의 튜닝과 조인트 잔차를 적절하게 코딩하고 코딩 이득을 달성하기 위해 레이트-왜곡 최적화를 위해 인코더 측에서 람다 파라미터의 튜닝을 요구한다. 양자화 파라미터 오프셋이 시그널링될 수 있다. 그러나 정확한 값을 결정하는 것은 비디오 인코더(200)에 대한 추가적인 부담일 수 있다.
- [0111] 본 개시는 위에 설명된 문제들 중 하나 이상을 해결할 수도 있는 기법들을 설명한다. 본 개시물의 기법들은 모델 기반 크로스 성분 잔차 예측의 모델 파라미터들의 시그널링 및 디코딩을 회피할 수 있고 위에서 설명된 공동 Cb-Cr 코딩 방법에서 양자화 파라미터의 튜닝을 회피할 수 있다. 그럼에도 불구하고, 본 개시물의 기법은 Cb 및 Cr 샘플 간의 상관을 감소시킬 수 있고, 그에 의해 코딩 효율 이득을 제공할 수 있다. 아래 및 본 개시의 다른 곳에서 제공된 예는 독립적으로 적용될 수 있거나 하나 이상의 예가 함께 적용될 수 있다.
- [0112] 본 개시물의 다음 섹션은 결합된 잔차 코딩(CRC)을 위한 기법들을 설명한다. 결합된 잔차 코딩을 위한 기법에 따라, 차원들(폭 및 높이)을 갖는 Cb 및 Cr 잔차 블록($resCb(x,y)$ 및 $resCr(x,y)$)은 잔차 변경 함수(RMF) ($F1(x,y)$ 및 $F2(x,y)$)로 프로세싱되어 아래의 표 1에 예시된 바와 같이 비디오 인코더(200)에서 후속적으로 순방향 변환(T), 양자화(Q) 및 엔트로피 코딩(E)되는 변경된 잔차를 획득한다.

표 1

[0113]

표 1

<ul style="list-style-type: none"> • 인코더 <ul style="list-style-type: none"> • $resCb(x,y) = origCb(x,y) - predCb(x,y)$ • $resCr(x,y) = origCr(x,y) - predCr(x,y)$ • $resCb(x,y) \rightarrow F1(x,y) \rightarrow T \rightarrow Q(Qp + offsetQp1) \rightarrow E$ • $resCr(x,y) \rightarrow F2(x,y) \rightarrow T \rightarrow Q(Qp + offsetQp2) \rightarrow E$ • 디코더: <ul style="list-style-type: none"> • $invE \rightarrow invQ(Qp + offsetQp1) \rightarrow invT \rightarrow resCb(x,y)' \rightarrow invF1(x,y) \rightarrow resCb(x,y)''$ • $invE \rightarrow invQ(Qp + offsetQp2) \rightarrow invT \rightarrow resCr(x,y)' \rightarrow invF2(x,y) \rightarrow resCr(x,y)''$ • $resCb(x,y) = predCb(x,y) + resCb(x,y)'$ • $resCr(x,y) = predCr(x,y) + resCr(x,y)'$

[0114] 표 1의 비디오 디코더(300)에서, 엔트로피 디코딩(invE), 역양자화(invQ), 역변환(invT) 후, 역 RMF 들

($\text{invF1}(x,y)$ 및 $\text{invF2}(x,y)$) 이 디코딩된 잔차 ($\text{resCb}(x,y)'$ 및 $\text{resCr}(x,y)'$) 에 적용되어, 재구성 ($\text{recCb}(x,y)$ 및 $\text{recCr}(x,y)$) 을 획득하기 위해 예측 ($\text{predCb}(x,y)$ 및 $\text{predCr}(x,y)$) 에 가산되는 잔차 ($\text{resCb}(x,y)''$ 및 $\text{resCr}(x,y)''$) 을 획득한다.

[0115] 일 예에서, 함수 F1 및 F2는 각각 감산 및 합산이고, 역 함수 invF1 및 invF2 는 각각 합산 및 감산이다. 정규화 인수 $1/N1$, $1/N2$, $1/M1$, $1/M2$ 및 라운딩 항 $A1$, $A2$, $B1$, $B2$ 는 다음 함수의 일부일 수 있다:

[0116] i) $F1(x,y) = (\text{resCb}(x,y) - \text{resCr}(x,y) + A1) / N1$ (식 1.a.i)

[0117] ii) $F2(x,y) = (\text{resCb}(x,y) + \text{resCr}(x,y) + A2) / N2$ (식 1.a.ii)

[0118] iii) $\text{invF1}(x,y) = (\text{resCr}(x,y)' + \text{resCb}(x,y)' + B1) / M1$ (식 1.a.iii)

[0119] iv) $\text{invF2}(x,y) = (\text{resCr}(x,y)' - \text{resCb}(x,y)' + B2) / M2$ (식 1.a.iv)

[0120] 한 예에서 $N1 = N2 = 2$, $A1 = A2 = 1$ (또는 0), $M1 = M2 = 1$, $B1 = B2 = 0$ 이다. 다른 예에서 $N1 = N2 = M1 = M2 = \sqrt{2}$ 이며, 인자 ($1/\sqrt{2}$) 는 대략 (181/256) 과 같으며 이 경우 $A1 = A2 = B1 = B2 = 128$ (또는 0) 이다. 일부 예들에서, 잔차 블록들 ($\text{resCb}(x,y)$, $\text{resCr}(x,y)$, $\text{resCb}(x,y)'$ 또는 $\text{resCr}(x,y)'$) 의 모든 값들은 0일 수 있거나 하나 이상의 잔차 블록들은 위 식에서 생략될 수 있다.

[0121] 다른 예에서, 함수 F1 및 F2는 각각 합산 및 감산이고, 역 함수 invF1 및 invF2 는 각각 합산 및 감산이다. 정규화 인수 $1/N1$, $1/N2$, $1/M1$, $1/M2$ 및 라운딩 항 $A1$, $A2$, $B1$, $B2$ 는 다음 함수의 일부일 수 있다:

[0122] i) $F1(x,y) = (\text{resCb}(x,y) + \text{resCr}(x,y) + A1) / N1$ (식 1.b.i)

[0123] ii) $F2(x,y) = (\text{resCb}(x,y) - \text{resCr}(x,y) + A2) / N2$ (식 1.b.ii)

[0124] iii) $\text{invF1}(x,y) = (\text{resCb}(x,y)' + \text{resCr}(x,y)' + B1) / M1$ (식 1.b.iii)

[0125] iv) $\text{invF2}(x,y) = (\text{resCb}(x,y)' - \text{resCr}(x,y)' + B2) / M2$ (식 1.b.iv)

[0126] 한 예에서 $N1 = N2 = 2$, $A1 = A2 = 1$ (또는 0), $M1 = M2 = 1$, $B1 = B2 = 0$ 이다. 다른 예에서 $N1 = N2 = M1 = M2 = \sqrt{2}$ 이며, 인자 ($1/\sqrt{2}$) 는 대략 (181/256) 과 같으며 이 경우 $A1 = A2 = B1 = B2 = 128$ (또는 0) 이다. 일부 예들에서, 잔차 블록들 ($\text{resCb}(x,y)$, $\text{resCr}(x,y)$, $\text{resCb}(x,y)'$ 또는 $\text{resCr}(x,y)'$) 의 모든 값들은 0일 수 있거나 하나 이상의 잔차 블록들은 위 식에서 생략될 수 있다.

[0127] 일부 예들에서, 함수 F1 및 F2는 각각 동일하지 않은 정규화 인자들을 갖는 감산 및 합산이고, 역 함수 invF1 및 invF2 는 각각 동일하지 않은 정규화 인자들을 갖는 합산 및 감산이다. 정규화 인자 $1/N10$, $1/N11$, $1/N20$, $1/N21$, $1/M10$, $1/M11$, $1/M20$, $1/M21$ 및 라운딩 항 $A10$, $A11$, $A20$, $A21$, $B10$, $B11$, $B20$, $B21$ 은 다음 함수들의 일부일 수 있다:

[0128] i) $F1(x,y) = (\text{resCb}(x,y) + A10)/N10 - (\text{resCr}(x,y) + A11)/N11$ (식 1.c.i)

[0129] ii) $F2(x,y) = (\text{resCb}(x,y) + A20)/N20 + (\text{resCr}(x,y) + A21)/N21$ (식 1.c.ii)

[0130] iii) $\text{invF1}(x,y) = (\text{resCr}(x,y)' + B10)/M10 + (\text{resCb}(x,y)' + B11)/M11$ (식 1.c.iii)

[0131] iv) $\text{invF2}(x,y) = (\text{resCr}(x,y)' + B20)/M20 - (\text{resCb}(x,y)' + B21)/M21$ (식 1.c.iv)

[0132] 한 예에서, 방정식 1.c.i, 1.c.ii, 1.c.iii 및 1.c.iv에서 $N10 = N11 = 1$ 및 $A10 = A11 = 0$; $N20 = N21 = 2$ 및 $A20 = A21 = 0$; $M10 = 1$ 및 $B10 = 0$; $M11 = 2$ 및 $B11 = 1$; $M20 = 1$ 및 $B20 = 0$; $M21 = 2$ 및 $B21 = 0$ 이다. 따라서 이 예에서 방정식 1.c.i, 1.c.ii, 1.c.iii 및 1.c.iv는 다음과 같이 다시 작성할 수 있다:

[0133] i) $F1(x,y) = \text{resCb}(x,y) - \text{resCr}(x,y)$ (식 1.c.v)

[0134] ii) $F2(x,y) = (\text{resCb}(x,y) + \text{resCr}(x,y)) / 2$ (식 1.c.vi)

[0135] (1) $\text{invF1}(x,y) = \text{resCr}(x,y)' + (\text{resCb}(x,y)' + 1) / 2$ (식 1.c.vii)

[0136] (2) $\text{invF2}(x,y) = \text{resCr}(x,y)' - \text{resCb}(x,y)' / 2$ (식 1.c.viii)

[0137] 잔차 블록들 ($\text{resCb}(x,y)$, $\text{resCr}(x,y)$, $\text{resCb}(x,y)'$ 또는 $\text{resCr}(x,y)'$) 의 모든 값들은 0일 수 있거나 하나 이상의 잔차 블록들은 위 식에서 생략될 수 있다.

[0138] 본 개시의 일부 예에서, CRS 프로세스는 RMF와 함께 적용될 수 있다. 예를 들어, CRS 프로세스가 RMF와 함

계 적용되는 예에서, RMF $F1(x,y)$ 및 $F2(x,y)$ 는 CRS가 잔차 $resCb(x,y)$ 및 $resCr(x,y)$ 를 스케일링한 후에 (예를 들어, 비디오 인코더(200)에 의해) 적용될 수 있다. 역으로, 이 예에서, 역 RMF $invF1(x,y)$ 및 $invF2(x,y)$ 는 $invCRS$ 스케일링 프로세스 전에 (예를 들어, 비디오 디코더(300)에 의해) 적용된다.

[0139] CRS 프로세스가 RMF와 함께 적용되는 다른 예에서, RMF $F1(x,y)$ 및 $F2(x,y)$ 는 CRS가 잔차 $resCb(x,y)$ 및 $resCr(x,y)$ 를 스케일링하기 전에 비디오 인코더(200)에 의해 적용될 수 있다. 반대로, 이 예에서, 역 RMF $invF1(x,y)$ 및 $invF2(x,y)$ 는 $invCRS$ 스케일링 프로세스 후에 비디오 디코더(300)에 의해 적용된다.

[0140] 일부 예들에서, 정규화 인자 $1/N$, $1/M$, 및 위에서 정의된 반올림 항 A , B 는 순방향 변환 프로세스 T 의 일부인 변환 계수 스케일링 프로세스와, 및 역으로 역변환 프로세스 $invT$ 의 일부인 역변환 계수 스케일링과 통합될 수 있다. 즉, 정규화 인자와 스케일링 항이 결합될 수 있다. 일부 그러한 예들에서, CRS 스케일링 프로세스는 (예를 들어, 비디오 인코더 (200) 또는 비디오 디코더 (300)에 의해) 적용되지 않는다. 정규화 인자 및 라운딩 항이 순방향 변환 프로세스 T 의 일부인 변환 계수 스케일링 프로세스와 통합되는 일부 예에서, CRS 스케일링 프로세스는 RMF $F1(x,y)$ 및 $F2(x,y)$ 전에 적용되고, 역으로 역 RMF $invF1(x,y)$ 및 $invF2(x,y)$ 는 $invCRS$ 스케일링 프로세스 전에 적용된다.

[0141] 정규화 인자 및 라운딩 항이 순방향 변환 프로세스 T 의 일부인 변환 계수 스케일링 프로세스와 통합되는 일부 예에서, RMF $F1(x,y)$ 및 $F2(x,y)$ 는 CRS가 잔차 ($resCb(x,y)$ 및 $resCr(x,y)$)를 스케일링하기 전에 적용될 수 있고, 역으로 역 RMF ($invF1(x,y)$ 및 $invF2(x,y)$)는 $invCRS$ 스케일링 프로세스 후에 적용될 수 있다. 이러한 경우에, CRS 스케일링은 또한 순방향 변환 프로세스 T 의 일부인 변환 계수 스케일링 프로세스와, 및 역으로 역변환 프로세스 $invT$ 의 일부인 역변환 계수 스케일링과 통합될 수 있다.

[0142] 대안적으로, 정규화 인자 및 라운딩 항이 순방향 변환 프로세스 T 의 일부인 변환 계수 스케일링 프로세스와 통합되는 예들에서, 정규화 인자 $1/N$, $1/M$, 및 위에서 정의된 라운딩 항 A , B 는 변경된 양자화 파라미터 값(표 1의 $offsetQp$)으로 완전히 또는 부분적으로 대체될 수 있다.

[0143] 더욱이, 본 개시물의 일부 예들에서, RMF들 ($F1(x,y)$, $F2(x,y)$)는 순방향 변환 프로세스 T 이후 및 양자화 Q 이전에 (예를 들어, 비디오 인코더(200)에 의해) 적용될 수 있다. 역으로, 역 RMF ($invF1(x,y)$, $invF2(x,y)$)는 역양자화 $invQ$ 이후 및 역변환 프로세스 $invT$ 전에 (예를 들어, 비디오 디코더(300)에 의해) 적용될 수 있다.

[0144] 이 예에서 변환 스킵 모드는 결합 잔차 코딩과 함께 적용될 수 있다. 따라서, 일부 예들에서, 비디오 인코더 (200)는 순방향 변환(예를 들어, DCT, DST 등)을 적용하지 않고 RMF들을 적용한다. 유사하게, 일부 예들에서, 비디오 디코더 (300)는 역 변환(예를 들어, 역 DCT, 역 DST 등)을 적용하지 않고 역 RMF들을 적용한다.

[0145] 상술한 바와 유사한 방식으로, 크로마 잔차 블록 $resCb(x,y)$ 및 $resCr(x,y)$ 에 대응하는 루마 잔차 블록 $resY(x,y)$ 가 함께 결합될 수 있다. 예를 들어, 하나의 그러한 예에서, RMF $F3(x,y)$ 는 (예를 들어, 비디오 인코더(200)에 의해) $resCb(x,y)$ 와 함께 $resY(x,y)$ 에 적용된다. 추가적으로, RMF $F4(x,y)$ 는 (예를 들어, 비디오 인코더(200)에 의해) $resCr(x,y)$ 와 함께 $resY(x,y)$ 에 적용된다. 더욱이, 이 예에서, 역으로 $F3$ 에 대한 역 RMF(즉, $invF3(x,y)$) 및 $F4$ 에 대한 역 RMF(즉, $invF4(x,y)$)가 (예를 들어, 비디오 디코더(300)에 의해) 적용된다. 한 예에서, $F3$, $F4$, $invF3$ 및 $invF4$ 는 다음과 같이 표현될 수 있다:

[0146]
$$F3 = (A1 * resY + B1 * resCB + D1) / N1$$

[0147]
$$F4 = (A2 * resY + B2 * resCR + D2) / N2$$

[0148]
$$invF3 = (A3 * F3 - B3 * F4 + D3) / N3$$

[0149]
$$invF4 = (A4 * F3 - B4 * F4 + D4) / N4$$

[0150] 크로마 잔차 블록에 대응하는 루마 잔차 블록이 함께 결합되는 다른 예에서, RMF $F5(x,y)$ 는 (예를 들어, 비디오 인코더(200)에 의해) $resCb(x,y)$ 및 $resCr(x,y)$ 와 함께 $resY(x,y)$ 에 적용된다. 역으로, $invF5(x,y)$ 가 (예를 들어, 비디오 디코더(300)에 의해) 적용된다. 한 예에서, $F5$ 및 $invF5$ 는 다음과 같이 표현될 수 있다:

[0151]
$$F5 = (A1 * resY + B1 * resCB + C1 * resCR + D1) / N1$$

[0152] $invF5 = (A3 * F5 + D3) / N3$

[0153] 예를 들어, 4:2:0 크로마 비디오 포맷의 경우, RMF F3, F4, F5는 resY(x,y) 잔차 블록의 다운샘플링(필터링 유무에 관계없이) 또는 resCb(x,y) 및 resCr(x,y) 블록들의 업샘플링(보간 여부에 관계없이)을 포함할 수 있다. 본 개시물의 CRC 기법은 또한 4:4:4 비디오 포맷 등에 적용될 수 있으며, 이 경우 다운샘플링 또는 업샘플링이 필요하지 않을 수 있다.

[0154] 본 개시의 CRC 틀은 SPS, PPS, 슬라이스 또는 타일 그룹 헤더 등에서 바이너리 플래그를 시그널링함으로써 가능해질 수 있다. 또한, CRC 플래그(crc_flag)는 변환 유닛 선택스 또는 잔차 코딩 선택스에서 시그널링되어 변환 유닛에 대한 CRC 틀을 가능하게 할 수 있다. 따라서, 비디오 인코더(200)는 비트스트림에서 CRC 플래그를 시그널링할 수도 있고 비디오 디코더(300)는 비트스트림으로부터 CRC 플래그를 획득할 수도 있다.

[0155] 일부 예들에서, CRC 플래그의 시그널링은 특정 기준의 발생에 의존한다. 예를 들어, 일 예에서, Cb 잔차 블록(cbf_cb)의 코딩된 블록 플래그(cbf)가 참이면 crc_flag가(예를 들어, 비디오 인코더(200)에 의해) 시그널링될 수도 있다. 잔차 블록의 cbf는 잔차 블록의 변환 계수 값이 모두 0인지 여부를 나타낸다. 변환 스킵 모드가 적용되는 경우, 잔차 블록의 변환 계수는 샘플 도메인에 있을 수 있다. 제 2 예에서, Cr 잔차 블록(cbf_cr)의 코딩된 블록 플래그(cbf)가 참이면 crc_flag가(예를 들어, 비디오 인코더(200)에 의해) 시그널링될 수도 있다. 제 3 예에서, cbf_cb가 참이거나 cbf_cr이 참이면 crc_flag가(예를 들어, 비디오 인코더(200)에 의해) 시그널링될 수도 있다. 또 다른 예에서, cbf_cb 및 cbf_cr이 참이면 crc_flag가(예를 들어, 비디오 인코더(200)에 의해) 시그널링될 수 있다.

[0156] cbf_cb가 참이고 cbf_cr도 참인 경우 crc_flag가 시그널링되는 예에서, 두 잔차들(식 1.a.i) 및 1.a.ii))에서 정의된 F1(x,y) 및 F2(x,y)) 양자 모두 또는 두 잔차 중 하나만이 F1(x,y) 또는 F2(x,y)를 0으로 강제함으로써 시그널링되는지 여부를 나타내는 제 2 플래그가(예를 들어, 비디오 인코더(200)에 의해) 각 변환 유닛에 대해 시그널링될 수 있다. 강제하는 것은 고정되거나(예를 들어, F1(x,y)가 아닌 F2(x,y)만 0으로 강제될 수 있음), 또는 가변적일 수 있다(F1(x,y) 또는 F2(x,y)는 0으로 강제될 수 있는 경우). 후자의 경우, 어떤 잔차가 0으로 강제되는지를 나타내기 위해 다른 플래그가 시그널링될 수 있다. 그렇게 하는 한 가지 동기는 하나의 잔차를 0으로 강제하는 것이 레이트 측면에서 저렴할 수 있지만 더 많은 왜곡을 유발할 수 있다는 것이다. 따라서, 비디오 인코더(200)는 레이트-왜곡 트레이드오프의 관점에서 자신을 튜닝하는 유연성을 가질 수 있다. 다른 예에서, 이러한 플래그(시그널링된 잔차의 수, 그리고 어느 잔차가 0으로 강제되는지)는 변환 유닛 레벨 대신 SPS, PPS, 슬라이스 또는 타일 그룹 헤더에서 시그널링될 수 있다.

[0157] 일부 예들에서, crc_flag는(예를 들어, 비디오 인코더(200) 및 비디오 디코더(300)에 의해) 컨텍스트 코딩되거나 바이패스 코딩될 수도 있다. 다중 컨텍스트는 cbf_cb 또는 cbf_cr 값, 인접 또는 현재 블록 모드(인트라, 인터, 현재 픽처 참조 등), 변환 스킵 플래그, 블록 차원 특성(폭, 높이, 비율), 이중 또는 공유 코딩 트리가 사용되는지 여부 등에 의존하는 컨텍스트와 함께 사용될 수 있다. 또한 cbf_cr 이후에 crc 플래그가 파싱되는 경우 cbf_cr이 참인지 여부에 따라 crc 플래그에 대해 다른 컨텍스트가 적용될 수 있다. cbf_cr 이전에 crc 플래그가 파싱되는 경우 crc 플래그의 값에 따라 cbf_cr에 대한 상이한 컨텍스트가 적용될 수 있다. crc 플래그 값에 따라 새로운 컨텍스트가 정의될 수 있거나, crc 플래그 값에 따라 cbf_cr 컴포넌트에 대한 기존 컨텍스트가 변경될 수 있다.

[0158] 일부 예들에서, 샘플들을 코딩하기 위해 하나 이상의 CRC 방법들이 이용가능할 수 있다(예를 들어, 하나의 방법은 RMF 들 (F1_a(x,y) 및 F2_a(x,y))를 수반할 수 있고 다른 방법은 RMF 들 (F1_b(x,y) 및 F2_b(x,y))를 수반할 수 있다. 코더에 의해 사용될 CRC 방법의 결정은 다음의 예들 중 하나 이상에 의해(예를 들어, 비디오 인코더(200) 및 비디오 디코더(300)에 의해) 이루어질 수도 있다. 일 예에서, 이용 가능한(또는 가능한) CRC 방법의 목록은 원래 Cb 및 Cr 성분의 고역 통과 필터링된 버전들 사이의 상관을 분석함으로써 결정될 수 있다. 이 분석은 오프라인에서 발생할 수 있으며 어떤 CRC 방법들이 압축 효율 이득을 제공하는 지를 결정하기 위해 다양한 콘텐츠를 기반으로 할 수 있다. 일부 예에서, 이용 가능한(또는 가능한) CRC 방법의 목록은 픽처 텍스처 또는 블록 텍스처를 분석하고 그 특징들을 그 목록에 매핑함으로써 결정될 수 있다. 상이한 CRC 방법은 상이한 RMF를 사용할 수 있다. 사용될 CRC 방법은 사용될 CRC 방법을 특징하는 비트스트림의 표시(예를 들어, 플래그 또는 CRC 방법의 목록에 대한 인덱스)에 의해 결정될 수 있다. 대안적으로, 비디오 인코더(200) 및 비디오 디코더(300)는 블록의 하나 이상의 특성들에 기초하여 CRC 방법을 결정할 수도 있다. 이러한 특성에는 슬라이스 또는 타일 그룹 유형이 인트라 또는 인터 유형인지 여부, CPR(현재 픽처 참조), 이중 또는 공유 코딩 트리가 사용되는지 여부, 잔차 블록 차원 또는 중형비 또는 블록의 기타 특성이 포함될 수 있다.

만 이에 국한되지 않는다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는, 그 결정에 기초하여, 결정된 CRC 방법을 사용하여 비디오의 하나 이상의 컴포넌트들을 코딩할 수도 있다.

[0159] 더욱이, 본 개시물의 일부 예들에서, 순방향 및 역 양자화 프로세스들, Q 및 invQ에 의해 사용되는 양자화 파라미터 값 Q_p 는 표 1에 예시된 바와 같이 $offsetQp1$ 및 $offsetQp2$ 값들에 의해 변경될 수 있다. 이러한 $offsetQp$ 값은 디폴트 값이거나 비트스트림(SPS, PPS, APS(Adaptation Parameter Set), 슬라이스 또는 타일 그룹 헤더, 유닛 헤더 등)에서 시그널링될 수 있다. 디폴트 값 또는 시그널링된 값은 슬라이스 또는 타일 그룹 유형이 인트라 또는 인터 유형인지 여부, CPR(현재 화상 참조), 이중 또는 공유 코딩 트리가 사용되는지 여부, 잔차 블록 차원 또는 중첩비 등에 따라 달라질 수 있다. 그 시그널링은 플래그가 결합된 잔여 코딩 틀이 가능해지는지 또는 불가능해지는지를 나타내는지 여부에 따라 달라질 수 있다.

[0160] 일부 예에서 CRC 방법은 특정 특성을 가진 블록에 대해 허용되지 않을 수 있다. 예를 들어, CRC 방법은 샘플의 임계 수(예를 들어, 16)보다 적은 블록에 대해 허용되지 않을 수 있다. 다른 예들에서, 폭, 높이, 중첩비, 이웃 블록 특성, cbf_flags 또는 다른 선택스 엘리먼트의 시그널링된 값 등을 포함하지만 이에 제한되지 않는 블록의 다른 특성은 CRC 방법이 허용되지 않을 수 있는 경우에 블록을 특정하는 데 사용될 수 있다.

[0161] 본 개시는 일반적으로 선택스 엘리먼트들과 같은, 소정의 정보를 "시그널링" 하는 것을 참조할 수도 있다. 용어 "시그널링" 은 일반적으로 인코딩된 비디오 데이터를 디코딩하는데 사용된 선택스 엘리먼트들 및/또는 다른 데이터에 대한 값들의 통신을 지칭할 수도 있다. 즉, 비디오 인코더 (200) 는 비트스트림에서 선택스 엘리먼트들에 대한 값들을 시그널링할 수도 있다. 일반적으로, 시그널링은 비트스트림에서 값을 생성하는 것을 지칭한다. 상기 언급된 바와 같이, 소스 디바이스 (102) 는 목적지 디바이스 (116) 에 의한 추후 취출을 위해 저장 디바이스 (112) 에 선택스 엘리먼트를 저장할 때 발생할 수도 있는 바와 같이, 비실시간으로 또는 실질적으로 실시간으로 비트스트림을 목적지 디바이스 (116) 로 전송할 수도 있다.

[0162] 도 5a 및 도 5b 는 예시적인 쿼드트리 이진 트리 (QTBT) 구조 (290), 및 대응하는 코딩 트리 유닛 (CTU) (292) 을 예시하는 개념적 다이어그램이다. 실선들은 쿼드트리 스플릿팅을 나타내고, 점선들은 바이너리 트리 스플릿팅을 나타낸다. 이진 트리의 각각의 분할된 (즉, 비-리프) 노드에서, 어느 분할 타입 (즉, 수평 또는 수직) 이 사용되는지를 나타내기 위해 하나의 플래그가 시그널링되며, 여기서 0 은 수평 분할을 표시하고 1 은 수직 분할을 표시한다. 쿼드트리 스플릿팅에 대해, 스플릿팅 타입을 표시할 필요는 없는데, 이는 쿼드트리 노드들이 동일한 사이즈를 가진 4 개의 서브-블록들로 수평으로 및 수직으로 블록을 스플릿팅하기 때문이다. 이에 따라, QTBT 구조 (290) 의 영역 트리 레벨 (즉, 제 1 레벨) (즉, 실선들) 에 대한 선택스 엘리먼트들 (이를 테면 분할 정보) 및 QTBT 구조 (290) 의 예측 트리 레벨 (즉, 제 2 레벨) (즉, 점선들) 에 대한 선택스 엘리먼트들 (이를 테면 분할 정보) 을, 비디오 인코더 (200) 가 인코딩할 수도 있고, 비디오 디코더 (300) 가 디코딩할 수도 있다. QTBT 구조 (290) 의 중간 리프 노드들에 의해 표현된 CU들에 대한, 예측 및 변환 데이터와 같은, 비디오 데이터를, 비디오 인코더 (200) 가 인코딩할 수도 있고, 비디오 디코더 (300) 가 디코딩할 수도 있다.

[0163] 일반적으로, 도 5b 의 CTU (292) 는 제 1 및 제 2 레벨들에서 QTBT 구조 (290) 의 노드들에 대응하는 블록들의 사이즈들을 정의하는 파라미터들과 연관될 수도 있다. 이들 파라미터들은 CTU 사이즈 (샘플들에서 CTU (292) 의 사이즈를 나타냄), 최소 쿼드트리 사이즈 (MinQTSIZE, 최소 허용된 쿼드트리 리프 노드 사이즈를 나타냄), 최대 바이너리 트리 사이즈 (MaxBTSIZE, 최대 허용된 바이너리 트리 루트 노드 사이즈를 나타냄), 최대 바이너리 트리 깊이 (MaxBTDEPTH, 최대 허용된 바이너리 트리 깊이를 나타냄), 및 최소 바이너리 트리 사이즈 (MinBTSIZE, 최소 허용된 바이너리 트리 리프 노드 사이즈를 나타냄) 를 포함할 수도 있다.

[0164] CTU 에 대응하는 QTBT 구조의 루트 노드는 QTBT 구조의 제 1 레벨에서 4 개의 자식 노드들을 가질 수도 있고, 이들의 각각은 쿼드트리 파티셔닝에 따라 파티셔닝될 수도 있다. 즉, 제 1 레벨의 노드들은 리프 노드들 (자식 노드들이 없음) 이거나 또는 4 개의 자식 노드들을 갖는다. QTBT 구조 (290) 의 예는 그러한 노드들을 브랜치들에 대한 실선들을 갖는 부모 노드 및 자식 노드들을 포함하는 것으로서 나타낸다. 제 1 레벨의 노드들이 최대 허용된 바이너리 트리 루트 노드 사이즈 (MaxBTSIZE) 보다 더 크지 않으면, 그들은 개개의 바이너리 트리들에 의해 추가로 파티셔닝될 수 있다. 하나의 노드의 바이너리 트리 스플릿팅은 스플릿팅으로부터 발생하는 노드들이 최소 허용된 바이너리 트리 리프 노드 사이즈 (MinBTSIZE) 또는 최대 허용된 바이너리 트리 깊이 (MaxBTDEPTH) 에 도달할 때까지 반복될 수 있다. QTBT 구조 (290) 의 예는 그러한 노드들을 브랜치들에 대한 점선들을 갖는 것으로서 나타낸다. 바이너리 트리 리프 노드는, 임의의 추가 파티셔닝 없이, 예측 (예를 들어, 인트라 픽처 또는 인터 픽처 예측) 및 변환을 위해 사용되는 코딩 유닛 (CU) 으로 지칭된다.

상기 논의된 바와 같이, CU들은 또한, "비디오 블록들" 또는 "블록들" 로 지칭될 수도 있다.

[0165] QTBT 파티셔닝 구조의 하나의 예에서, CTU 사이즈는 128×128 (루마 샘플들 및 2 개의 대응하는 64×64 크로마 샘플들) 로서 설정되고, MinQTSIZE 는 16×16 으로서 설정되고, MaxBTSIZE 는 64×64 로서 설정되고, (폭 및 높이 양자 모두에 대한) MinBTSIZE 는 4 로서 설정되고, 그리고 MaxBTDepth 는 4 로서 설정된다. 퀴드트리 파티셔닝은 퀴드트리 리프 노드들을 생성하기 위해 먼저 CTU 에 적용된다. 퀴드트리 리프 노드들은 16×16 (즉, MinQTSIZE) 으로부터 128×128 (즉, CTU 사이즈) 까지의 사이즈를 가질 수도 있다. 퀴드트리 리프 노드가 128x128 인 경우, 사이즈가 MaxBTSIZE (즉, 이 예에서는 64x64) 를 초과하기 때문에 그것은 바이너리 트리 에 의해 추가로 스플릿팅되지 않을 것이다. 그렇지 않으면, 퀴드트리 리프 노드는 바이너리 트리 에 의해 추가로 파티셔닝될 것이다. 따라서, 퀴드트리 리프 노드는 또한 바이너리 트리 에 대한 루트 노드이고 바이너리 트리 심도를 0 으로서 갖는다. 바이너리 트리 심도가 MaxBTDepth (이 예에서는 4) 에 도달할 때, 추가의 스플릿팅이 허용되지 않는다. 바이너리 트리 노드가 MinBTSIZE (이 예에서는 4) 와 동일한 폭을 가질 때, 그것은 추가의 수직 스플릿팅이 허용되지 않음을 암시한다. 유사하게, 높이가 MinBTSIZE 와 동일한 이진 트리 노드는 그 이진 트리 노드에 대해 추가의 수평 분할이 허용되지 않음을 암시한다. 위에 언급된 바와 같이, 이진 트리의 리프 노드들은 CU들로 지칭되고, 추가 파티셔닝 없이 예측 및 변환에 따라 추가로 프로세싱된다.

[0166] 도 6 은 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 인코더 (200) 를 예시하는 블록 다이어그램이다. 도 6 은 설명을 목적으로 제공되고, 본 개시에서 폭넓게 예시화 및 설명된 바와 같이 기법들의 한정으로 고려되어서는 안된다. 설명의 목적들을 위해, 본 개시는 HEVC 비디오 코딩 표준 및 개발 중인 H.266 비디오 코딩 표준과 같은 비디오 코딩 표준들의 컨텍스트에서 비디오 인코더 (200) 를 설명한다. 그러나, 본 개시의 기법들은 이들 비디오 코딩 표준들에 한정되지 않으며, 일반적으로 비디오 인코딩 및 디코딩에 적용가능하다.

[0167] 도 6 의 예에서, 비디오 인코더 (200) 는 비디오 데이터 메모리 (230), 모드 선택 유닛 (202), 잔차 생성 유닛 (204), 변환 프로세싱 유닛 (206), 양자화 유닛 (208), 역 양자화 유닛 (210), 역 변환 프로세싱 유닛 (212), 재구성 유닛 (214), 필터 유닛 (216), 디코딩된 픽처 버퍼 (DPB) (218), 및 엔트로피 인코딩 유닛 (220) 을 포함한다. 또한, 도 6 의 예에서, 비디오 인코더 (200) 는 잔차 변경 유닛 (205) 및 역 잔차 변경 유닛 (213) 을 포함할 수 있다. 비디오 데이터 메모리 (230), 모드 선택 유닛 (202), 잔차 생성 유닛 (204), 잔차 변경 유닛 (205), 변환 프로세싱 유닛 (206), 양자화 유닛 (208), 역 양자화 유닛 (210), 역 변환 프로세싱 유닛 (212), 역 잔차 변경 유닛 (213), 재구성 유닛 (214), 필터 유닛 (216), DPB (218), 및 엔트로피 인코딩 유닛 (220) 중 임의의 것 또는 전부는 하나 이상의 프로세서들에서 또는 프로세싱 회로부에서 구현될 수도 있다. 더욱이, 비디오 인코더 (200) 는 이들 및 다른 기능들을 수행하기 위해 부가 또는 대안의 프로세서들 또는 프로세싱 회로부들을 포함할 수도 있다.

[0168] 비디오 데이터 메모리 (230) 는 비디오 인코더 (200) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 인코더 (200) 는 예를 들어, 비디오 소스 (104) (도 1) 로부터 비디오 데이터 메모리 (230) 에 저장된 비디오 데이터를 수신할 수도 있다. DPB (218) 는 비디오 인코더 (200) 에 의한 후속 비디오 데이터의 예측에 사용하기 위해 레퍼런스 비디오 데이터를 저장하는 참조 픽처 메모리로서 작용할 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동기식 동적 랜덤 액세스 메모리 (SDRAM) 을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (230) 및 DPB (218) 는 동일한 메모리 디바이스 또는 개별 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (230) 는 예시된 바와 같이 비디오 인코더 (200) 의 다른 컴포넌트들과 온-칩이거나, 또는 그 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0169] 본 개시에서, 비디오 데이터 메모리 (230) 에 대한 참조는 이처럼 구체적으로 기재되지 않으면 비디오 인코더 (200) 내부의 메모리 또는 이처럼 구체적으로 기재되지 않으면 비디오 인코더 (200) 외부의 메모리로 제한되는 것으로 해석되지 않아야 한다. 오히려, 비디오 데이터 메모리 (230) 에 대한 참조는 비디오 인코더 (200) 가 인코딩을 위해 수신하는 비디오 데이터 (예를 들어, 인코딩될 현재 블록에 대한 비디오 데이터) 를 저장하는 레퍼런스 메모리로서 이해되어야 한다. 도 1 의 메모리 (106) 는 도 1 의 메모리 (106) 는 또한 비디오 인코더 (200) 의 다양한 유닛들로부터의 출력들의 일시적 저장을 제공할 수도 있다.

[0170] 도 3 의 다양한 유닛들은 도 6 의 다양한 유닛들은 비디오 인코더 (200) 에 의해 수행되는 동작들의 이해를 돕

기 위해 예시된다. 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하며, 수행될 수 있는 동작들에 대해 미리설정된다. 프로그래밍가능 회로들은 다양한 태스크들을 수행하도록 프로그래밍될 수 있는 회로들을 지칭하고, 수행될 수 있는 동작들에 있어서 유연한 기능성을 제공한다. 예를 들어, 프로그래밍가능 회로들은, 프로그래밍가능 회로들이 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 (예를 들어, 파라미터들을 수신하거나 또는 파라미터들을 출력하기 위해) 소프트웨어 명령들을 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작들의 타입들은 일반적으로 불변이다. 일부 예들에 있어서, 유닛들 중 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그래밍가능) 일 수도 있고, 일부 예들에 있어서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.

[0171] 비디오 인코더 (200) 는 프로그래밍가능 회로들로부터 형성된, 산술 로직 유닛들 (ALU들), 기본 함수 유닛들 (EFU들), 디지털 회로들, 아날로그 회로들, 및/또는 프로그래밍가능 코어들을 포함할 수도 있다. 비디오 인코더 (200) 의 동작들이 프로그래밍가능 회로들, 메모리 (106) 에 의해 실행되는 소프트웨어를 사용하여 수행되는 예들에서, 메모리 (106) (도 1) 는 비디오 인코더 (200) 가 수신하고 실행하는 소프트웨어의 오브젝트 코드를 저장할 수도 있거나, 또는 (도시되지 않은) 비디오 인코더 (200) 내의 다른 메모리가 그러한 명령들을 저장할 수도 있다.

[0172] 비디오 데이터 메모리 (230) 는 수신된 비디오 데이터를 저장하도록 구성된다. 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 비디오 데이터의 픽처를 추출하고 비디오 데이터를 잔차 생성 유닛 (204) 및 모드 선택 유닛 (202) 에 제공할 수도 있다. 비디오 데이터 메모리 (230) 에서의 비디오 데이터는 인코딩되어야 하는 원시 비디오 데이터일 수도 있다.

[0173] 모드 선택 유닛 (202) 은 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라-예측 유닛 (226) 을 포함한다. 모드 선택 유닛 (202) 은 다른 예측 모드들에 따라 비디오 예측을 수행하기 위해 추가적인 기능 유닛들을 포함할 수도 있다. 예를로서, 모드 선택 유닛 (202) 은 팔레트 유닛, 인트라-블록 카피 유닛 (이는 모션 추정 유닛 (222) 및/또는 모션 보상 유닛 (224) 의 부분일 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다.

[0174] 모드 선택 유닛 (202) 은 일반적으로 인코딩 파라미터들의 조합들 및 그러한 조합들에 대한 결과의 레이트-왜곡 값들을 테스트하기 위해 다중 인코딩 패스들을 조정한다. 인코딩 파라미터들은 CTU들의 CU들로의 파티셔닝, CU들에 대한 예측 모드들, CU들의 잔차 데이터에 대한 변환 타입들, CU들의 잔차 데이터에 대한 양자화 파라미터들 등을 포함할 수도 있다. 모드 선택 유닛 (202) 은 궁극적으로 다른 테스트된 조합들보다 우수한 레이트-왜곡 값들을 갖는 인코딩 파라미터들의 조합을 선택할 수도 있다.

[0175] 비디오 인코더 (200) 는 비디오 데이터 메모리 (230) 로부터 추출된 픽처를 일련의 CTU들로 파티셔닝하고, 슬라이스 내에 하나 이상의 CTU들을 캡슐화할 수도 있다. 모드 선택 유닛 (202) 은 상기 설명된 HEVC 의 쿼드트리 구조 또는 QTBT 구조와 같은, 트리 구조에 따라 픽처의 CTU 를 파티셔닝할 수도 있다. 상기 설명된 바와 같이, 비디오 인코더 (200) 는 트리 구조에 따라 CTU 를 파티셔닝하는 것으로부터 하나 이상의 CU들을 형성할 수도 있다. 그러한 CU 는 일반적으로 "비디오 블록" 또는 "블록" 으로도 또한 지칭될 수도 있다.

[0176] 일반적으로, 모드 선택 유닛 (202) 은 또한 그의 컴포넌트들 (예를 들어, 모션 추정 유닛 (222), 모션 보상 유닛 (224), 및 인트라-예측 유닛 (226)) 을 제어하여 현재 블록 (예를 들어, 현재 CU, 또는 HEVC 에서, PU 및 TU 의 오버랩하는 부분) 에 대한 예측 블록을 생성한다. 현재 블록의 인트라-예측을 위해, 모션 추정 유닛 (222) 은 하나 이상의 참조 픽처들 (예를 들어, DPB (218) 에 저장된 하나 이상의 이전에 코딩된 픽처들) 에서 하나 이상의 근접하게 매칭하는 레퍼런스 블록들을 식별하기 위해 모션 탐색을 수행할 수도 있다. 특히, 모션 추정 유닛 (222) 은, 예를 들어, 절대 차이의 합 (SAD), 제곱 차이들의 합 (SSD), 평균 절대 차이 (MAD), 평균 제곱 차이들 (MSD) 등에 따라, 잠재적 레퍼런스 블록이 현재 블록에 얼마나 유사한지를 나타내는 값을 계산할 수도 있다. 모션 추정 유닛 (222) 은 일반적으로 고려되는 레퍼런스 블록과 현재 블록 사이의 샘플 별 차이들을 사용하여 이들 계산들을 수행할 수도 있다. 모션 추정 유닛 (222) 은 현재 블록에 가장 근접하게 매칭하는 레퍼런스 블록을 표시하는, 이러한 계산들로부터 야기되는 최저 값을 갖는 레퍼런스 블록을 식별할 수도 있다.

[0177] 모션 추정 유닛 (222) 은 현재 픽처에서의 현재 블록의 포지션에 대한 참조 픽처들에서의 레퍼런스 블록들의 포지션들을 정의하는 하나 이상의 모션 벡터들 (MV들) 을 형성할 수도 있다. 모션 추정 유닛 (222) 은 그 후 모션 벡터들을 모션 보상 유닛 (224) 에 제공할 수도 있다. 예를 들어, 단방향 인트라-예측에 대해, 모션 추

정 유닛 (222) 은 단일 모션 벡터를 제공할 수도 있는 반면, 양방향 인터-예측에 대해, 모션 추정 유닛 (222) 은 2 개의 모션 벡터들을 제공할 수도 있다. 그 후, 모션 보상 유닛 (224) 은 모션 벡터들을 사용하여 예측 블록을 생성할 수도 있다. 예를 들어, 모션 보상 유닛 (224) 은 모션 벡터를 사용하여 레퍼런스 블록의 데이터를 추출할 수도 있다. 다른 예로서, 모션 벡터가 분수 샘플 정밀도를 갖는다면, 모션 보상 유닛 (224) 은 하나 이상의 보간 필터들에 따라 예측 블록에 대한 값들을 보간할 수도 있다. 또한, 양방향 인터-예측에 대해, 모션 보상 유닛 (224) 은 개별의 모션 벡터들에 의해 식별된 2 개의 레퍼런스 블록들에 대한 데이터를 추출하고, 예를 들어 샘플 별 평균화 또는 가중된 평균화를 통해 추출된 데이터를 결합할 수도 있다.

[0178] 다른 예로서, 인트라-예측, 또는 인트라-예측 코딩에 대해, 인트라-예측 유닛 (226) 은 현재 블록에 이웃하는 샘플들로부터 예측 블록을 생성할 수도 있다. 예를 들어, 방향성 모드들에 대해, 인트라-예측 유닛 (226) 은 일반적으로 이웃 샘플들의 값들을 수학적으로 결합하고 현재 블록에 걸쳐 정의된 방향에서 이들 계산된 값들을 파플레이트하여 예측 블록을 생성할 수도 있다. 또 다른 예로서, DC 모드에 대해, 인트라-예측 유닛 (226) 은 현재 블록에 대한 이웃 샘플들의 평균을 계산하고 예측 블록을 생성하여 예측 블록의 각각의 샘플에 대해 이러한 결과의 평균을 포함할 수도 있다.

[0179] 모드 선택 유닛 (202) 은 예측 블록을 잔차 생성 유닛 (204) 에 제공한다. 잔차 생성 유닛 (204) 은 비디오 데이터 메모리 (230) 로부터의 현재 블록의 원시, 코딩되지 않은 버전 및 모드 선택 유닛 (202) 으로부터의 예측 블록을 수신한다. 잔차 생성 유닛 (204) 은 현재 블록과 예측 블록 사이의 샘플 별 차이들을 계산한다. 결과의 샘플 별 차이들은 현재 블록에 대한 잔차 블록을 정의한다. 일부 예들에서, 잔차 생성 유닛 (204) 은 또한 잔차 차분 펄스 코드 변조 (residual differential pulse code modulation; RDPCM) 를 사용하여 잔차 블록을 생성하기 위해 잔차 블록에서의 샘플 값들 사이의 차이들을 결정할 수도 있다. 일부 예들에서, 잔차 생성 유닛 (204) 은 이진 감산 (binary subtraction) 을 수행하는 하나 이상의 감산 회로들을 사용하여 형성될 수도 있다.

[0180] 모드 선택 유닛 (202) 이 CU들을 PU들로 파티셔닝하는 예들에서, 각각의 PU 는 루마 예측 유닛 및 대응하는 크로마 예측 유닛들과 연관될 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 다양한 사이즈를 갖는 PU들을 지원할 수도 있다. 상기 나타낸 바와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있고 PU 의 사이즈는 PU 의 루마 예측 유닛의 사이즈를 지칭할 수도 있다. 특정 CU 의 사이즈가 $2N_x \times 2N_y$ 임을 가정하면, 비디오 인코더 (200) 는 인트라-예측을 위해 $2N_x \times 2N_y$ 또는 $N_x \times N_y$ 의 PU 사이즈들을 지원하고, 인터-예측을 위해 $2N_x \times 2N_y$, $2N_x \times N_y$, $N_x \times 2N_y$, $N_x \times N_y$, 기타 등등의 대칭적인 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 또한, 인터-예측을 위해 $2N_x \times nU$, $2N_x \times nD$, $nL \times 2N_y$, 및 $nR \times 2N_y$ 의 PU 사이즈에 대한 비대칭적 파티셔닝을 지원할 수도 있다.

[0181] 모드 선택 유닛이 CU 를 PU들로 추가로 파티셔닝하지 않는 예들에서, 각각의 CU 는 루마 코딩 블록 및 대응하는 크로마 코딩 블록들과 연관될 수도 있다. 위에서와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있다. 비디오 인코더 (200) 및 비디오 디코더 (300) 는 $2N_x \times 2N_y$, $2N_x \times N_y$, 또는 $N_x \times 2N_y$ 의 CU 크기들을 지원할 수도 있다.

[0182] 인트라-블록 카피 모드 코딩, 아핀 모드 코딩 및 선형 모델 (LM) 모드 코딩과 같은 다른 비디오 코딩 기법들에 대해, 몇몇 예들로서, 모드 선택 유닛 (202) 은, 코딩 기법들과 연관된 개별 유닛들을 통해, 인코딩되는 현재 블록에 대한 예측 블록을 생성한다. 팔레트 모드 코딩과 같은 일부 예에서, 모드 선택 유닛 (202) 은 예측 블록을 생성하지 않을 수도 있고, 대신에 선택된 팔레트에 기초하여 블록을 재구성하는 방식을 표시하는 신텍스 엘리먼트들을 생성할 수도 있다. 이러한 모드들에서, 모드 선택 유닛 (202) 은 이들 신텍스 엘리먼트들을 인코딩되도록 엔트로피 인코딩 유닛 (220) 에 제공할 수도 있다.

[0183] 상기 설명된 바와 같이, 잔차 생성 유닛 (204) 은 현재 블록 및 대응하는 예측 블록에 대한 비디오 데이터를 수신한다. 잔차 생성 유닛 (204) 은 그 후 현재 블록에 대한 잔차 블록을 생성한다. 잔차 블록을 생성하기 위해, 잔차 생성 유닛 (204) 은 현재 블록과 예측 블록 사이의 샘플 별 차이들을 계산한다.

[0184] 본 개시물의 하나 이상의 기법들에 따르면, 잔차 변경 유닛 (205) 은 잔차 생성 유닛 (204) 에 의해 생성된 잔차 크로마 데이터에 RMF들을 적용할 수도 있다. 예를 들어, 잔차 변경 유닛 (205) 은 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용할 수 있다. 잔차 변경 유닛 (205) 은 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 블록의 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용할 수 있다. 이 예에서, 제1 크로마 잔차 데이터는 제1 크로마 성분 (예를 들어, Cb) 과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분 (예를 들어, Cr) 과 연관된

다. 잔차 변경 유닛(205)은 잔차 루마 데이터에 RMF를 적용하지 않는다.

- [0185] 변환 프로세싱 유닛 (206) 은 잔차 블록에 하나 이상의 변환들을 적용하여 변환 계수들의 블록 (본 명세서에서는 "변환 계수 블록" 으로 지칭됨) 을 생성한다. 변환 프로세싱 유닛 (206) 은 다양한 변환들을 잔차 블록에 적용하여 변환 계수 블록을 형성할 수 있다. 예를 들어, 변환 프로세싱 유닛 (206) 은 이산 코사인 변환 (DCT), 방향성 변환, Karhunen-Loeve 변환 (KLT), 또는 개념적으로 유사한 변환을 잔차 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 대한 다중 변환들, 예를 들어 1 차 변환 및 2 차 변환, 이를 테면 회전 변환을 수행할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (206) 은 잔차 블록에 변환들을 적용하지 않는다.
- [0186] 양자화 유닛 (208) 은 양자화된 변환 계수 블록을 생성하기 위해 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 유닛 (208) 은 현재 블록과 연관된 양자화 파라미터 (QP) 값에 따라 변환 계수 블록의 변환 계수들을 양자화할 수도 있다. 비디오 인코더 (200) 는 (예컨대, 모드 선택 유닛 (202) 을 통해) CU 와 연관된 QP 값을 조정함으로써 현재 블록과 연관된 변환 계수 블록들에 적용되는 양자화도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서, 양자화된 변환 계수들은 변환 프로세싱 유닛 (206) 에 의해 생성된 오리지널 변환 계수들보다 더 낮은 정밀도를 가질 수도 있다.
- [0187] 역 양자화 유닛 (210) 및 역 변환 프로세싱 유닛 (212) 은, 각각, 양자화된 변환 계수 블록에 역 양자화 및 역 변환들을 적용하여, 변환 계수 블록으로부터 잔차 블록을 복원할 수도 있다. 본 개시물의 하나 이상의 예들에 따르면, 역 잔차 변경 유닛 (213) 은 역변환 프로세싱 유닛 (212) 에 의해 생성된 디코딩된 변경된 크로마 잔차 데이터에 역 RMF들을 적용할 수 있다. 역 잔차 변경 유닛(213)에 의해 적용된 역 RMF는 잔차 변경 유닛(205)에 의해 적용된 RMF의 역일 수 있다. 따라서, 역 잔차 변경 유닛(213)은 잔차 블록을 생성할 수 있다. 역 잔차 변경 유닛(213)은 잔차 루마 데이터에 역 RMF를 적용하지 않는다.
- [0188] 재구성 유닛 (214) 은 모드 선택 유닛 (202) 에 의해 생성된 예측 블록 및 재구성된 잔차 블록에 기초하여 (잠재적으로 어느 정도의 왜곡을 가짐에도 불구하고) 현재 블록에 대응하는 재구성된 블록을 생성할 수도 있다. 예를 들어, 재구성 유닛 (214) 은 재구성된 잔차 블록의 샘플들을, 모드 선택 유닛 (202) 에 의해 생성된 예측 블록으로부터의 대응하는 샘플들에 가산하여 재구성된 블록을 생성할 수도 있다.
- [0189] 필터 유닛 (216) 은 재구성된 블록에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (216) 은 CU들의 에지들을 따라 블록화 아티팩트들 (blockiness artifacts) 을 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (216) 의 동작들은 일부 예들에서 스킵될 수도 있다.
- [0190] 비디오 인코더 (200) 는 DPB (218) 에 재구성된 블록들을 저장한다. 예를 들어, 필터 유닛 (216) 의 동작들이 필요하지 않은 예들에서, 재구성 유닛 (214) 은 재구성된 블록들을 DPB (218) 에 저장할 수도 있다. 필터 유닛 (216) 의 동작들이 필요한 예들에서, 필터 유닛 (216) 은 필터링된 재구성된 블록들을 DPB (218) 에 저장할 수도 있다. 모션 추정 유닛 (222) 및 모션 보상 유닛 (224) 은 재구성된 (및 잠재적으로 필터링된) 블록들로부터 형성된 DPB (218) 로부터 참조 픽처를 취출하여, 후속 인코딩된 픽처들의 블록들을 인터-예측할 수도 있다. 또한, 인트라-예측 유닛 (226) 은 현재 픽처에서의 다른 블록들을 인트라-예측하기 위해 현재 픽처의 DPB (218) 에서 재구성된 블록들을 사용할 수도 있다.
- [0191] 일반적으로, 엔트로피 인코딩 유닛 (220) 은 비디오 인코더 (200) 의 다른 기능 컴포넌트들로부터 수신된 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 양자화 유닛 (208) 으로부터 양자화된 변환 계수 블록들을 엔트로피 인코딩할 수도 있다. 다른 예로서, 엔트로피 인코딩 유닛 (220) 은 모드 선택 유닛 (202) 으로부터 예측 신택스 엘리먼트들 (예를 들어, 인트라-예측에 대한 인트라-모드 정보 또는 인터-예측에 대한 모션 정보) 을 엔트로피 인코딩할 수도 있다. 엔트로피 인코딩 유닛 (220) 은 엔트로피-인코딩된 데이터를 생성하기 위해, 비디오 데이터의 다른 예인, 신택스 엘리먼트들에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (220) 은 컨텍스트-적응적 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 신택스 기반 컨텍스트-적응적 이진 산술 코딩 (SBAC) 동작, 화률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-골롬 인코딩 동작, 또는 다른 타입의 엔트로피 인코딩 동작을 데이터에 대해 수행할 수도 있다. 일부 예들에서, 엔트로피 인코딩 유닛 (220) 은 신택스 엘리먼트들이 엔트로피 인코딩되지 않는 바이패스 모드에서 동작할 수도 있다.
- [0192] 비디오 인코더 (200) 는 픽처 또는 슬라이스의 블록들을 재구성하는데 필요한 엔트로피 인코딩된 신택스 엘리먼트

트들을 포함하는 비트스트림을 출력할 수도 있다. 특히, 엔트로피 인코딩 유닛 (220) 이 비트스트림을 출력할 수도 있다.

- [0193] 상기 설명된 동작들은 블록과 관련하여 설명된다. 그러한 설명은 루마 코딩 블록 및/또는 크로마 코딩 블록들에 대한 동작들인 것으로 이해되어야 한다. 상기 설명된 바와 같이, 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 CU 의 루마 및 크로마 컴포넌트들이다. 일부 예들에서, 루마 코딩 블록 및 크로마 코딩 블록들은 PU 의 루마 및 크로마 컴포넌트들이다.
- [0194] 일부 예들에서, 루마 코딩 블록에 대해 수행되는 동작들은 크로마 코딩 블록들에 대해 반복될 필요가 없다. 하나의 예로서, 크로마 블록들에 대한 모션 벡터 (MV) 및 참조 픽처를 식별하기 위해 루마 코딩 블록에 대한 MV 및 참조 픽처를 식별하는 동작들이 반복될 필요는 없다. 오히려, 루마 코딩 블록에 대한 MV 는 크로마 블록들에 대한 MV 를 결정하도록 스케일링될 수도 있고, 참조 픽처는 동일할 수도 있다. 다른 예로서, 인트라-예측 프로세스는 루마 코딩 블록들 및 크로마 코딩 블록들에 대해 동일할 수도 있다.
- [0195] 비디오 인코더 (200) 는 비디오 데이터를 저장하도록 구성된 메모리, 및 회로에서 구현된 하나 이상의 프로세싱 유닛들을 포함하는 비디오 데이터를 인코딩하도록 구성된 디바이스의 예를 나타내며, 그 하나 이상의 프로세싱 유닛들은 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하고; 제1 변경된 크로마 잔차 데이터를 인코딩하고; 제2 변경된 크로마 잔차 데이터를 생성하기 위해 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용하고; 그리고 제2 변경된 크로마 잔차 데이터를 인코딩하도록 구성되며, 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관된다.
- [0196] 도 7 는 본 개시의 기법들을 수행할 수도 있는 예시적인 비디오 디코더 (300) 를 예시하는 블록 다이어그램이다. 도 7 은 설명을 목적으로 제공되고, 본 개시에서 폭넓게 예시화 및 설명된 바와 같이 기법들에 대해 한정하는 것은 아니다. 설명의 목적들을 위해, 본 개시는 JEM, VVC, 및 HEVC 의 기법들에 따르는 비디오 디코더 (300) 를 설명한다. 그러나, 본 개시의 기법들은 다른 비디오 코딩 표준들로 구성되는 비디오 코딩 디바이스들에 의해 수행될 수도 있다.
- [0197] 도 7 의 예에서, 비디오 디코더 (300) 는, 코딩된 픽처 버퍼 (CPB) 메모리 (320), 엔트로피 디코딩 유닛 (302), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 재구성 유닛 (310), 필터 유닛 (312), 및 디코딩된 픽처 버퍼 (DPB) (314) 를 포함한다. 본 개시물의 하나 이상의 기법들에 따르면, 비디오 디코더 (300) 는 또한 역 잔차 변경 유닛 (309) 을 포함한다. CPB 메모리 (320), 엔트로피 디코딩 유닛 (302), 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 역 잔차 변경 유닛 (309), 재구성 유닛 (310), 필터 유닛 (312), 및 DPB (314) 의 어느 것 또는 전부는 하나 이상의 프로세서들에서 또는 프로세싱 회로에서 구현될 수도 있다. 더욱이, 비디오 디코더 (300) 는 이들 및 다른 기능들을 수행하기 위해 부가 또는 대안의 프로세서들 또는 프로세싱 회로부를 포함할 수도 있다.
- [0198] 예측 프로세싱 유닛 (304) 은 모션 보상 유닛 (316) 및 인트라-예측 유닛 (318) 을 포함한다. 예측 프로세싱 유닛 (304) 은 다른 예측 모드들에 따라 예측을 수행하기 위해 부가 유닛들을 포함할 수도 있다. 예를 들어, 예측 프로세싱 유닛 (304) 은 팔레트 유닛, 인트라-블록 카피 유닛 (모션 보상 유닛 (316) 의 일부를 형성할 수도 있음), 아핀 유닛, 선형 모델 (LM) 유닛 등을 포함할 수도 있다. 다른 예들에서, 비디오 디코더 (300) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.
- [0199] CPB 메모리 (320) 는, 비디오 디코더 (300) 의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. CPB 메모리 (320) 에 저장된 비디오 데이터는, 예를 들어 컴퓨터 판독가능 매체 (110) (도 1) 로부터 획득될 수도 있다. CPB 메모리 (320) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터 (예를 들어, 신택스 엘리먼트들) 를 저장하는 CPB 를 포함할 수도 있다. 또한, CPB 메모리 (320) 는 비디오 디코더 (300) 의 다양한 유닛들로부터의 출력들을 나타내는 일시적 데이터와 같은, 코딩된 픽처의 신택스 엘리먼트들 이외의 비디오 데이터를 저장할 수도 있다. DPB (314) 는 일반적으로, 인코딩된 비디오 비트스트림의 후속 데이터 또는 픽처들을 디코딩할 때 레퍼런스 비디오 데이터로서 비디오 디코더 (300) 가 출력 및/또는 사용할 수도 있는 디코딩된 픽처들을 저장한다. CPB 메모리 (320) 및 DPB (314) 는 다양한 메모리 디바이스들, 예컨대 동기식 동적 랜덤 액세스 메모리 (SDRAM) 를 포함한 DRAM, 자기저항성 RAM (MRAM), 저항성 RAM (RRAM) 과 같은 다양한 메모리 디바이스들, 또는 다른 타입들의 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. CPB 메모리 (320) 및 DPB (314) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, CPB 메모리 (320) 는 비디오 디코더 (300) 의

다른 컴포넌트들과 온-칩이거나, 또는 그 컴포넌트들에 대해 오프-칩일 수도 있다.

- [0200] 추가적으로 또는 대안적으로, 일부 예들에서, 비디오 디코더 (300) 는 메모리 (120) (도 1) 로부터 코딩된 비디오 데이터를 추출할 수도 있다. 즉, 메모리 (120) 는 CPB 메모리 (320) 로 상기 논의된 바와 같이 데이터를 저장할 수도 있다. 마찬가지로, 메모리 (120) 는, 비디오 디코더 (300) 의 기능의 일부 또는 전부가 비디오 디코더 (300) 의 프로세싱 회로부에 의해 실행되도록 소프트웨어에서 구현될 때, 비디오 디코더 (300) 에 의해 실행될 명령들을 저장할 수도 있다.
- [0201] 도 7 에 도시된 다양한 유닛들은 비디오 디코더 (300) 에 의해 수행되는 동작들의 이해를 돕기 위해 예시된다. 이 유닛들은 고정 기능 회로들, 프로그래밍가능 회로들, 또는 이들의 조합으로서 구현될 수도 있다. 도 6 과 유사하게, 고정 기능 회로들은 특정 기능성을 제공하는 회로들을 지칭하고, 수행될 수 있는 동작들에 대해 미리 설정된다. 프로그래밍가능 회로들은 다양한 태스크들을 수행하도록 프로그래밍될 수 있는 회로들을 지칭하고, 수행될 수 있는 동작들에 있어서 유연한 기능성을 제공한다. 예를 들어, 프로그래밍가능 회로들은, 프로그래밍가능 회로들이 소프트웨어 또는 펌웨어의 명령들에 의해 정의된 방식으로 동작하게 하는 소프트웨어 또는 펌웨어를 실행할 수도 있다. 고정 기능 회로들은 (예를 들어, 파라미터들을 수신하거나 또는 파라미터들을 출력하기 위해) 소프트웨어 명령들을 실행할 수도 있지만, 고정 기능 회로들이 수행하는 동작들의 타입들은 일반적으로 불변이다. 일부 예들에 있어서, 유닛들 중 하나 이상은 별개의 회로 블록들 (고정 기능 또는 프로그래밍가능) 일 수도 있고, 일부 예들에 있어서, 하나 이상의 유닛들은 집적 회로들일 수도 있다.
- [0202] 비디오 디코더 (300) 는 프로그래밍가능 회로들로부터 형성된, ALU들, EFU들, 디지털 회로들, 아날로그 회로들, 및/또는 프로그래밍가능 코어들을 포함할 수도 있다. 비디오 디코더 (300) 의 동작들이 프로그래밍가능 회로들 상에서 실행하는 소프트웨어에 의해 수행되는 예들에서, 온-칩 또는 오프-칩 메모리는 비디오 디코더 (300) 가 수신하고 실행하는 소프트웨어의 명령들 (예를 들어, 오브젝트 코드) 을 저장할 수도 있다.
- [0203] 엔트로피 디코딩 유닛 (302) 은 인코딩된 비디오 데이터를 CPB 메모리 (320) 로부터 수신하고, 비디오 데이터를 엔트로피 디코딩하여 선택스 엘리먼트들을 재생할 수도 있다. 예측 프로세싱 유닛 (304), 역 양자화 유닛 (306), 역 변환 프로세싱 유닛 (308), 역 잔차 변경 유닛 (309), 복원 유닛 (310), 및 필터 유닛 (312) 은 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.
- [0204] 일반적으로, 비디오 디코더 (300) 는 블록 별 (block-by-block) 단위로 픽처를 재구성한다. 비디오 디코더 (300) 는 개별적으로 각각의 블록에 대해 재구성 동작을 수행할 수도 있다 (여기서 현재 재구성되는, 즉 디코딩되는 블록은 "현재 블록" 으로 지칭될 수도 있음).
- [0205] 엔트로피 디코딩 유닛 (302) 은 양자화 파라미터 (QP) 및/또는 변환 모드 표시(들)와 같은 변환 정보 뿐만 아니라, 양자화된 변환 계수 블록의 양자화된 변환 계수들을 정의하는 선택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 역 양자화 유닛 (306) 은 양자화된 변환 계수 블록과 연관된 QP 를 사용하여, 양자화도 및 유사하게, 역 양자화 유닛 (306) 이 적용할 역 양자화도를 결정할 수도 있다. 역 양자화 유닛 (306) 은 예를 들어, 양자화된 변환 계수들을 역 양자화하기 위해 비트단위 좌측-시프트 동작을 수행할 수도 있다. 따라서, 역 양자화 유닛 (306) 은 변환 계수들을 포함하는 변환 계수 블록을 형성할 수도 있다.
- [0206] 역 양자화 유닛 (306) 이 변환 계수 블록을 형성한 후, 역 변환 프로세싱 유닛 (308) 은 현재 블록과 연관된 잔차 블록을 생성하기 위해 변환 계수 블록에 하나 이상의 역 변환들을 적용할 수도 있다. 본 개시물의 하나 이상의 예들에 따르면, 역변환 프로세싱 유닛 (308) 은 디코딩된 변경된 크로마 잔차 데이터를 생성하기 위해 변환 계수 블록에 하나 이상의 역변환들을 적용할 수 있다. 일 예에서, 역변환 프로세싱 유닛 (308) 은 역 DCT, 역 정수 변환, 역 Karhunen-Loeve 변환 (KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 변환 계수 블록에 적용할 수도 있다. 현재 블록은 루마 성분 및 2개의 크로마 성분을 포함할 수 있기 때문에, 역 변환 프로세싱 유닛(308)은 현재 블록에 대한 디코딩된 루마 잔차 데이터, 제1 디코딩된 변경된 크로마 잔차 데이터, 및 제2 디코딩된 변경된 크로마 잔차 데이터를 생성할 수 있다.
- [0207] 본 개시물의 하나 이상의 기법들에 따르면, 역 잔차 변경 유닛 (309) 은 디코딩된 변경된 크로마 잔차 데이터에 역 RMF들을 적용할 수 있다. 따라서, 역 잔차 변경 유닛 (309) 은 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 디코딩된 변경된 크로마 잔차 데이터 (예를 들어, Cb 잔차 데이터) 에 제 1 역 잔차 변경 함수를 적용할 수 있다. 또한, 역 잔차 변경 유닛 (309) 은 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 디코딩된 변경된 크로마 잔차 데이터 (예를 들어, Cr 잔차 데이터) 에 제 2 역 잔차 변경 함수를 적용할 수 있다.

- [0208] 또한, 예측 프로세싱 유닛 (304) 은 엔트로피 디코딩 유닛 (302) 에 의해 엔트로피 디코딩된 예측 정보 신택스 엘리먼트들에 따라 예측 블록을 생성한다. 예를 들어, 예측 정보 신택스 엘리먼트들이 현재 블록이 인터-예측됨을 표시하면, 모션 보상 유닛 (316) 은 예측 블록을 생성할 수도 있다. 이 경우에, 예측 정보 신택스 엘리먼트들은 레퍼런스 블록을 추출할 DPB (314) 에서의 참조 픽처 뿐만 아니라 현재 픽처에서의 현재 블록의 위치에 대한 참조 픽처에서의 레퍼런스 블록의 위치를 식별하는 모션 벡터를 표시할 수도 있다. 모션 보상 유닛 (316) 은 일반적으로 모션 보상 유닛 (224) (도 6) 에 대하여 설명된 것과 실질적으로 유사한 방식으로 인터-예측 프로세스를 수행할 수도 있다.
- [0209] 다른 예로서, 예측 정보 신택스 엘리먼트들이 현재 블록이 인트라-예측됨을 표시하면, 인트라-예측 유닛 (318) 은 예측 정보 신택스 엘리먼트들에 의해 표시된 인트라-예측 모드에 따라 예측 블록을 생성할 수도 있다. 다시, 인트라-예측 유닛 (318) 은 일반적으로 인트라-예측 유닛 (226) (도 6) 에 대하여 설명된 것과 실질적으로 유사한 방식으로 인트라-예측 프로세스를 수행할 수도 있다. 인트라-예측 유닛 (318) 은 DPB (314) 로부터 현재 블록에 대한 이웃 샘플들의 데이터를 추출할 수도 있다.
- [0210] 재구성 유닛(310)은 예측 블록 및 잔차 블록(예를 들어, 역 잔차 변경 유닛(309)에 의해 생성된 잔차 크로마 데이터의 블록)을 사용하여 현재 블록을 재구성할 수 있다. 예를 들어, 재구성 유닛 (310) 은 잔차 블록의 샘플들을 예측 블록의 대응하는 샘플들에 가산하여 현재 블록을 재구성할 수도 있다. 본 개시물의 기법들에 따르면, 잔차 블록은 역 잔차 변경 유닛 (309) 에 의해 생성된 역 변경된 크로마 잔차 데이터일 수 있다.
- [0211] 필터 유닛 (312) 은 재구성된 블록에 대해 하나 이상의 필터 동작들을 수행할 수도 있다. 예를 들어, 필터 유닛 (312) 은 재구성된 블록들의 에지들을 따라 블록크니스 아티팩트를 감소시키기 위해 디블록킹 동작들을 수행할 수도 있다. 필터 유닛 (312) 의 동작들이 모든 예들에서 반드시 수행되는 것은 아니다.
- [0212] 비디오 디코더 (300) 는 DPB (314) 에 재구성된 블록들을 저장할 수도 있다. 예를 들어, 필터 유닛 (312) 의 동작들이 수행되지 않는 예들에서, 재구성 유닛 (310) 은 재구성된 블록들을 DPB (314) 에 저장할 수도 있다. 필터 유닛 (312) 의 동작들이 수행되는 예들에서, 필터 유닛 (312) 은 필터링된 재구성된 블록들을 DPB (314) 에 저장할 수도 있다. 상기 논의된 바와 같이, DPB (314) 는 예측 프로세싱 유닛 (304) 에, 인트라-예측을 위한 현재 픽처 및 후속 모션 보상을 위한 이전에 디코딩된 픽처들의 샘플들과 같은 레퍼런스 정보를 제공할 수도 있다. 또한, 비디오 디코더 (300) 는 도 1 의 디스플레이 디바이스 (118) 와 같은 디스플레이 이 디바이스 상의 후속 프리젠테이션을 위해 DPB 로부터 디코딩된 픽처들을 출력할 수도 있다.
- [0213] 이러한 방식으로, 비디오 디코더 (300) 는 비디오 데이터를 저장하도록 구성된 메모리, 및 회로에서 구현된 하나 이상의 프로세싱 유닛들을 포함하는 비디오 디코딩 디바이스의 예를 나타내고, 그 하나 이상의 프로세싱 유닛들은 제1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 디코딩된 변경된 크로마 잔차 데이터에 제 1 역 잔차 변경 함수를 적용하고; 제2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제2 디코딩된 변경된 크로마 잔차 데이터에 제2 역 잔차 변경 함수를 적용하는 것으로서, 제1 디코딩된 변경된 크로마 잔차 데이터는 제1 크로마 성분과 연관되고 제2 디코딩된 변경된 크로마 잔차 데이터는 제2 크로마 성분과 연관되는, 상기 제2 역 잔차 변경 함수를 적용하며; 및 제1 역 변경된 크로마 잔차 데이터 및 제2 역 변경된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성하도록 구성된다.
- [0214] 도 8 은 현재의 블록을 인코딩하는 예시의 방법을 도시하는 플로우차트이다. 현재 블록은 현재 CU 를 포함할 수도 있다. 비디오 인코더 (200) (도 1 및 도 6) 에 대해 설명되지만, 다른 디바이스들이 도 8 의 방법과 유사한 방법을 수행하도록 구성될 수도 있음을 이해해야 한다.
- [0215] 이 예에서, 비디오 인코더 (200) 는 초기에 현재 블록을 예측한다 (350). 예를 들어, 비디오 인코더 (200) 는 현재 블록에 대한 예측 블록을 형성할 수도 있다. 비디오 인코더 (200) 는 그 후 현재 블록에 대한 잔차 블록을 계산할 수 있다 (352). 다시 말해서, 비디오 인코더 (200) 는 현재 블록에 대한 루마 잔차 데이터 및 크로마 잔차 데이터를 계산할 수 있다. 잔차 블록을 계산하기 위해, 비디오 인코더 (200) 는 원래의, 인코딩되지 않은 블록의 샘플들과 현재 블록에 대한 예측 블록의 대응하는 샘플들 사이의 차이를 계산할 수 있다. 예를 들면, Cb 잔차 블록을 계산하기 위해, 비디오 인코더 (200) 는 원래의, 인코딩되지 않은 블록의 Cb 샘플들과 현재 블록에 대한 예측 블록의 대응하는 Cb 샘플들 사이의 차이를 계산할 수 있다.
- [0216] 본 개시물의 하나 이상의 기법들에 따르면, 비디오 인코더 (200) 는 변경된 크로마 잔차 데이터를 생성하기 위해 현재 블록의 크로마 잔차 데이터에 잔차 변경 함수들을 적용할 수 있다 (353). 비디오 인코더 (200) 는 그 후 잔차 블록을 변환하여 변환 계수들을 생성하고 잔차 블록의 변환 계수들을 양자화할 수 있다 (354).

즉, 비디오 인코더 (200) 는 그 다음 변경된 크로마 잔차 데이터를 주파수 도메인과 같은 변환 도메인으로 변환하는 순방향 변환을 적용하여 변환 계수들을 생성할 수 있다. 비디오 인코더 (200) 는 그 후 변환 계수들을 양자화할 수 있다. 다른 예들에서, 비디오 인코더 (200) 는 순방향 변환을 적용한 후에 잔차 변경 함수들을 적용할 수 있다.

[0217] 다음으로, 비디오 인코더 (200) 는 잔차 블록의 양자화된 변환 계수들을 스캔할 수도 있다 (356). 스캔 동안 또는 스캔에 후속하여, 비디오 인코더 (200) 는 변환 계수들을 엔트로피 인코딩할 수도 있다 (358). 예를 들어, 비디오 인코더 (200) 는 CAVLC 또는 CABAC 를 사용하여 변환 계수들을 인코딩할 수도 있다. 비디오 인코더 (200) 는 그 후 블록의 엔트로피 인코딩된 데이터를 출력할 수도 있다 (360).

[0218] 도 9 는 비디오 데이터의 현재 블록을 디코딩하는 예시적인 방법을 나타내는 플로우차트이다. 현재 블록은 현재 CU 를 포함할 수도 있다. 비디오 디코더 (300) (도 1 및 도 7) 와 관련하여 설명되었지만, 다른 디바이스들이 도 9 의 방법과 유사한 방법을 수행하도록 구성될 수도 있음을 이해해야 한다.

[0219] 비디오 디코더 (300) 는 현재 블록에 대응하는 잔차 블록의 변환 계수들에 대한 엔트로피 코딩된 예측 정보 및 엔트로피 코딩된 데이터와 같은, 현재 블록에 대한 엔트로피 코딩된 데이터를 수신할 수도 있다 (370). 비디오 디코더 (300) 는 엔트로피 인코딩된 데이터를 엔트로피 디코딩하여 현재 블록에 대한 예측 정보를 결정하고 잔차 블록의 변환 계수들을 재생할 수도 있다 (372). 비디오 디코더 (300) 는 현재 블록에 대한 예측 블록을 계산하기 위해, 예를 들어 현재 블록에 대한 예측 정보에 의해 표시된 바와 같이 인트라- 또는 인터-예측 모드를 사용하여, 현재 블록을 예측할 수도 있다 (374). 비디오 디코더 (300) 는 그 후 양자화된 변환 계수들의 블록을 생성하기 위해 재생된 변환 계수들을 역 스캔할 수 있다 (376). 비디오 디코더 (300) 는 그 후 잔차 블록을 생성하기 위해 변환 계수들을 역 양자화 및 역 변환할 수도 있다 (378). 비디오 디코더 (300) 는 현재 블록의 각각의 컬러 성분에 대해 동작들 (372-378) 을 수행할 수 있다. 따라서, 비디오 디코더 (300) 는 현재 블록에 대한 Y 잔차 블록, Cb 잔차 블록, 및 Cr 잔차 블록을 생성할 수 있다. Cb 잔차 블록은 제1 디코딩된 잔차 크로마 데이터일 수 있고 Cr 잔차 블록은 제2 디코딩된 잔차 크로마 데이터일 수 있다. 다른 예들에서, Cr 잔차 블록은 제 1 디코딩된 잔차 크로마 데이터로 간주될 수도 있고 Cb 잔차 블록은 제 2 디코딩된 잔차 크로마 데이터로 간주될 수 있다. 본 개시물의 하나 이상의 기법들에 따르면, 비디오 디코더 (300) 는 역 잔차 변경 함수를 디코딩된 잔차 크로마 데이터에 적용할 수 있다 (379). 비디오 디코더 (300) 는 이 개시물의 다른 곳에서 제공된 예들의 임의의 것에 따라 디코딩된 잔차 크로마 데이터에 역 잔차 변경 함수를 적용할 수 있다. 다른 예들에서, 비디오 디코더 (300) 는 역변환을 적용하기 전에 역 잔차 변경 함수들을 적용할 수 있다.

[0220] 비디오 디코더 (300) 는 예측 블록 및 잔차 블록을 조합함으로써 결국 현재 블록을 디코딩할 수도 있다 (380). 예를 들어, 비디오 디코더 (300) 는, 컬러 성분들 각각에 대해, 예측 블록의 컬러 성분의 샘플들을 그 컬러 성분에 대한 잔차 블록의 대응하는 샘플들에 가산할 수 있다.

[0221] 도 10 은 본 개시의 하나 이상의 기법들에 따른, 비디오 인코더 (200) 의 예시적인 동작을 나타내는 플로우차트이다. 도 10 의 예에서, 비디오 인코더 (200) 는 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 비디오 데이터의 블록의 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용할 수 있다 (400). 예를 들어, 비디오 인코더 (200) 는 식 1.a.i, 1.b.i, 1.c.i, 1.c.v 중 임의의 것에 설명된 바와 같은 잔차 변경 함수, 또는 다른 잔차 변경 함수를 적용할 수 있다.

[0222] 또한, 도 10 의 예에서, 비디오 인코더 (200) 는 제1 변경된 크로마 잔차 데이터를 인코딩할 수 있다 (402). 제 1 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 비디오 인코더 (200) 는 샘플 도메인으로부터의 주파수 도메인과 같은 변환 도메인으로 제 1 변경된 크로마 잔차 데이터를 변환하는 순방향 변환 (예를 들어, 이산 코사인 변환, 이산 사인 변환, 또는 다른 유형의 변환) 을 적용할 수 있다 (404). 샘플 도메인에서의 데이터는 샘플의 값(예를 들어, 루마 값, Cb 값, Cr 값)을 나타낸다. 제 1 변경된 크로마 잔차 데이터에 순방향 변환을 적용함으로써, 비디오 인코더 (200) 는 변환 계수들의 제 1 블록을 생성할 수 있다. 더욱이, 제 1 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 비디오 인코더 (200) 는 변환 계수들의 제 1 블록에서의 변환 계수들을 양자화할 수 있다. 추가적으로, 비디오 인코더 (200) 는 변환 계수들의 양자화된 제 1 블록을 나타내는 선택스 엘리먼트들에 엔트로피 인코딩을 적용할 수 있다. 일부 예들에서, 비디오 인코더 (200) 는 순방향 변환의 적용을 스킵할 수 있다. 다시 말해서, 비디오 인코더 (200) 는 현재 블록과 함께 변환 스킵 모드를 사용할 수 있다. 설명의 편의를 위해, 본 개시물은 변환 스킵 모드가 현재 블록과 함께 사용되는 경우 잔차 데이터에 적용하기 위해 "변환 계수"라는 용어를 사용할 수 있다.

- [0223] 또한, 비디오 인코더 (200) 은 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 블록의 제 2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용할 수 있다 (406). 예를 들어, 비디오 인코더 (200) 는 식 1.a.ii, 1.b.ii, 1.c.ii, 1.c.vi 중 임의의 것에 설명된 바와 같은 잔차 변경 함수, 또는 다른 잔차 변경 함수를 적용할 수 있다.
- [0224] 비디오 인코더 (200) 는 제 2 변경된 크로마 잔차 데이터를 인코딩할 수 있다 (408). 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관된다. 제 2 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 비디오 인코더 (200) 는 샘플 도메인으로부터의 주파수 도메인과 같은 변환 도메인으로 제 2 변경된 크로마 잔차 데이터를 변환하는 순방향 변환 (예를 들어, 이산 코사인 변환, 이산 사인 변환, 또는 다른 유형의 변환) 을 적용할 수 있다 (410). 제 2 변경된 크로마 잔차 데이터에 순방향 변환을 적용함으로써, 비디오 인코더 (200) 는 변환 계수들의 제 2 블록을 생성할 수 있다. 더욱이, 제 2 변경된 크로마 잔차 데이터를 인코딩하는 것의 일부로서, 비디오 인코더 (200) 는 변환 계수들의 제 2 블록에서의 변환 계수들을 양자화할 수 있다. 추가적으로, 비디오 인코더 (200) 는 변환 계수들의 양자화된 제 2 블록을 나타내는 선택스 엘리먼트들에 엔트로피 인코딩을 적용할 수 있다. 일부 예들에서, 비디오 인코더 (200) 는 순방향 변환의 적용을 스킵할 수 있다.
- [0225] 일부 예들에서, 제 1 잔차 변경 함수를 적용하기 전에, 비디오 인코더 (200) 는 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 잔차 변경 함수 또는 제 3 잔차 변경 함수를 제 1 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 유사하게, 제 2 잔차 변경 함수를 적용하기 전에, 비디오 인코더 (200) 는 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 잔차 변경 함수 또는 제 4 잔차 변경 함수를 제 2 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 따라서, 비디오 인코더 (200) 는 제 1 잔차 크로마 데이터 및 제 2 잔차 크로마 데이터에 어떤 잔차 변경 함수를 적용할지를 결정할 수 있다.
- [0226] 비디오 인코더 (200) 는 다양한 방식들 중 하나로 어떤 잔차 변경 함수를 적용할지를 결정할 수 있다. 예를 들어, 일부 예들에서, 비디오 인코더 (200) 는, 블록의 하나 이상의 특성들에 기초하여, 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 잔차 변경 함수 또는 제 3 잔차 변경 함수를 제 1 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 또한, 비디오 인코더 (200) 는, 블록의 하나 이상의 특성들에 기초하여, 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 잔차 변경 함수 또는 제 4 잔차 변경 함수를 제 2 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 블록의 이러한 특성은 다음 중 하나 이상을 포함할 수 있다: 블록을 포함하는 슬라이스 또는 타일이 인트라 유형인지 인터 유형인지 여부, 현재 픽처 참조가 블록과 함께 사용되는지 여부, 이중 또는 공유 코딩 트리가 블록과 함께 사용되는지 여부, 블록의 차원들, 블록의 중첩비 또는 블록의 기타 특성. 예를 들어, 비디오 인코더 (200) 는 블록이 I 슬라이스에 있으면 제 1 잔차 변경 함수를 사용하거나 블록이 P 또는 B 슬라이스에 있으면 제 3 잔차 수정을 사용하기로 결정할 수 있다. I 슬라이스는 인트라 예측은 허용되지만 인터 예측은 허용되지 않는 슬라이스이다. P 슬라이스는 인트라 예측 및 단방향 인터 예측이 허용되지만 양방향 인터 예측이 허용되지 않는 슬라이스이다. B 슬라이스는 인트라 예측, 단방향 인터 예측, 및 양방향 인터 예측이 허용되는 슬라이스이다.
- [0227] 일부 예들에서, 비디오 인코더 (200) 는, 비트스트림에서, 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 잔차 변경 함수 또는 제 3 잔차 변경 함수가 제 1 크로마 잔차 데이터에 적용되었는지 여부를 나타내는 데이터를 시그널링할 수 있다. 유사하게, 비디오 인코더 (200) 는, 비트스트림에서, 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 잔차 변경 함수 또는 제 4 잔차 변경 함수가 제 2 크로마 잔차 데이터에 적용되었는지 여부를 나타내는 데이터를 시그널링할 수 있다. 예를 들어, 비트스트림에서 시그널링된 하나 이상의 선택스 엘리먼트는 어떤 잔차 변경 함수가 적용되었는지를 특정할 수 있다.
- [0228] 일부 예들에서, 비디오 인코더 (200) 는 잔차 변경 함수들을 적용하는 것과 함께 CRS 프로세스들을 적용할 수도 있다. 예를 들어, 일부 예들에서, 비디오 인코더 (200) 는, 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 역 잔차 변경 함수를 제 1 크로마 잔차 데이터에 적용한 후 제 1 역 변경된 크로마 잔차 데이터에 CRS 프로세스를 적용할 수 있다. 또, 이 예에서, 비디오 인코더 (200) 는 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 잔차 변경 함수를 제 2 크로마 잔차 데이터에 적용한 후에 제 2 변경된 크로마 잔차 데이터에 CRS 프로세스를 적용할 수 있다. 예를 들어, CRS 프로세스를 적용하기 위해, 비디오 인코더 (200) 는 본 개시물의 다른 곳에서 설명된 바와 같이 식 7을 적용할 수도 있다.
- [0229] 다른 예들에서, 비디오 인코더 (200) 는, 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 잔차 변경 함수를 제 1 크로마 잔차 데이터에 적용하는 것의 일부로서 제 1 크로마 잔차 데이터에 CRS 프로세스를 적용할 수

있다. 또, 이 예에서, 비디오 인코더 (200) 는 제 2 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 잔차 변경 함수를 제 2 크로마 잔차 데이터에 적용하는 것의 일부로서 제 2 크로마 잔차 데이터에 CRS 프로세스를 적용할 수 있다. 비디오 인코더 (200) 는 이 개시물의 다른 곳에서 제공된 예들의 임의의 것에 따라 제 1 및 제 2 잔차 변경 함수를 적용하는 것의 일부로서 CRS 프로세스를 적용할 수도 있다.

[0230] 도 11 은 본 개시의 하나 이상의 기법들에 따른, 비디오 디코더 (300) 의 예시적인 동작을 나타내는 플로우차트이다. 도 11 의 예에서, 비디오 디코더 (300) 는 비디오 데이터의 블록의 변환 계수들의 제 1 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제 1 디코딩된 변경된 크로마 잔차 데이터를 생성할 수 있다 (450). 예를 들어, 비디오 디코더 (300) 는 변환 계수들의 제 1 세트를 주파수 도메인으로부터 샘플 도메인으로 변환하는 역 DCT 변환을 적용할 수도 있다. 추가적으로, 비디오 디코더 (300) 는 블록의 변환 계수들의 제 2 세트를 변환 도메인으로부터 샘플 도메인으로 변환하기 위해 역변환을 적용함으로써 제 2 디코딩된 변경된 크로마 잔차 데이터를 생성할 수 있다 (452). 제1 디코딩된 변경된 크로마 잔차 데이터는 제1 크로마 성분 (예를 들어, Cb) 과 연관되고, 제2 디코딩된 변경된 크로마 잔차 데이터는 제2 크로마 성분 (예를 들어, Cr) 과 연관된다.

[0231] 또한, 도 11 의 예에서, 따라서, 비디오 디코더 (300) 은 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 디코딩된 변경된 크로마 잔차 데이터에 제 1 역 잔차 변경 함수를 적용할 수 있다. 예를 들어, 비디오 디코더 (300) 는 식 1.a.iii, 1.b.iii, 1.c.iii, 1.c.vii 중 임의의 것에 설명된 바와 같은 역 잔차 변경 함수, 또는 다른 역 잔차 변경 함수를 적용할 수 있다.

[0232] 비디오 디코더 (300) 은 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 디코딩된 변경된 크로마 잔차 데이터에 제 2 역 잔차 변경 함수를 적용할 수 있다. 예를 들어, 비디오 디코더 (300) 는 식 1.a.iv, 1.b.iv, 1.c.iv, 1.c.viii 중 임의의 것에 설명된 바와 같은 역 잔차 변경 함수, 또는 다른 역 잔차 변경 함수를 적용할 수 있다.

[0233] 또한, 비디오 디코더 (300) 는 제 1 역 변경된 크로마 잔차 데이터 및 제 2 역 변경된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성할 수 있다 (458). 예를 들어, 비디오 디코더 (300) 는 제 1 역 변경된 크로마 잔차 데이터의 샘플들을 제 1 크로마 성분에 대한 예측 블록의 대응하는 샘플들에 가산할 수 있다. 추가적으로, 이 예에서, 비디오 디코더 (300) 는 제 2 역 변경된 크로마 잔차 데이터의 샘플들을 제 2 크로마 성분에 대한 예측 블록의 대응하는 샘플들에 가산할 수도 있다. 다시 말해서, 비디오 디코더 (300) 는 블록의 제 1 재구성된 크로마 데이터를 획득하기 위해 제 1 역 변경된 크로마 잔차 데이터를 제 1 예측된 크로마 데이터에 가산하고, 블록의 제 2 재구성된 크로마 데이터를 획득하기 위해 제 2 역 변경된 크로마 잔차 데이터를 제 2 예측된 크로마 데이터에 가산할 수 있다.

[0234] 일부 예들에서, 제 1 역 잔차 변경 함수를 적용하기 전에, 비디오 디코더 (300) 는 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 역 잔차 변경 함수 또는 제 3 역 잔차 변경 함수를 제 1 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 유사하게, 제 2 역 잔차 변경 함수를 적용하기 전에, 비디오 디코더 (300) 는 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 역 잔차 변경 함수 또는 제 4 역 잔차 변경 함수를 제 2 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 따라서, 비디오 디코더 (300) 는 제 1 디코딩된 잔차 크로마 데이터 및 제 2 디코딩된 잔차 크로마 데이터에 어떤 잔차 변경 함수를 적용할지를 결정할 수 있다.

[0235] 비디오 디코더 (300) 는 다양한 방식들 중 하나로 어떤 역 잔차 변경 함수를 적용할지를 결정할 수 있다. 예를 들어, 일부 예들에서, 비디오 디코더 (300) 는, 블록의 하나 이상의 특성들에 기초하여, 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 역 잔차 변경 함수 또는 제 3 역 잔차 변경 함수를 제 1 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 또한, 비디오 디코더 (300) 는, 블록의 하나 이상의 특성들에 기초하여, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 역 잔차 변경 함수 또는 제 4 역 잔차 변경 함수를 제 2 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 블록의 이러한 특성은 다음 중 하나 이상을 포함할 수 있다: 블록을 포함하는 슬라이스 또는 타일이 인트라 유형인지 인터 유형인지 여부, 현재 픽처 참조가 블록과 함께 사용되는지 여부, 이중 또는 공유 코딩 트리가 블록과 함께 사용되는지 여부, 블록의 차원들, 블록의 중형비 또는 블록의 기타 특성. 예를 들어, 비디오 디코더 (300) 는 블록이 I 슬라이스에 있으면 제 1 역 잔차 변경 함수를 사용하거나 블록이 P 또는 B 슬라이스에 있으면 제 3 역 잔차 수정을 사용하기로 결정할 수 있다.

[0236] 일부 예들에서, 비디오 디코더 (300) 는, 비트스트림에서 시그널링된 데이터에 기초하여, 제 1 역 변경된 크로

마 잔차 데이터를 생성하기 위해 제 1 역 잔차 변경 함수 또는 제 3 역 잔차 변경 함수를 제 1 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 유사하게, 비디오 디코더 (300) 는, 비트스트림에서 시그널링된 데이터에 기초하여, 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 역 잔차 변경 함수 또는 제 4 역 잔차 변경 함수를 제 2 디코딩된 변경된 크로마 잔차 데이터에 적용할지 여부를 결정할 수 있다. 예를 들어, 비트스트림에서 시그널링된 하나 이상의 신텍스 엘리먼트는 어떤 역 잔차 변경 함수를 사용해야 하는지를 특정할 수 있다.

[0237] 일부 예들에서, 비디오 디코더 (300) 는 역 잔차 변경 함수들을 적용하는 것과 함께 CRS 프로세스들을 적용할 수도 있다. 예를 들어, 일부 예들에서, 비디오 디코더 (300) 는, 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 역 잔차 변경 함수를 제 1 디코딩된 변경된 크로마 잔차 데이터에 적용하기 전에 제 1 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용할 수 있다. 또, 이 예에서, 비디오 디코더 (300) 는 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 역 잔차 변경 함수를 제 2 디코딩된 변경된 크로마 잔차 데이터에 적용하기 전에 제 2 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용할 수 있다. 예를 들어, 역 CRS 프로세스를 적용하기 위해, 비디오 디코더 (300) 는 본 개시물의 다른 곳에서 설명된 바와 같이 식 8을 적용할 수도 있다.

[0238] 다른 예들에서, 비디오 디코더 (300) 는, 제 1 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 역 잔차 변경 함수를 제 1 디코딩된 변경된 크로마 잔차 데이터에 적용하는 것의 일부로서 제 1 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용할 수 있다. 또, 이 예에서, 비디오 디코더 (300) 는 제 2 역 변경된 크로마 잔차 데이터를 생성하기 위해 제 2 역 잔차 변경 함수를 제 2 디코딩된 변경된 크로마 잔차 데이터에 적용하는 것의 일부로서 제 2 디코딩된 변경된 크로마 잔차 데이터에 역 CRS 프로세스를 적용할 수 있다.

[0239] 다음의 열거된 단락들은 본 개시의 기법들의 예들의 비제한적 리스트를 제공한다.

[0240] 실시예 1A. 비디오 데이터를 디코딩하는 방법으로서, 제1 역 잔차 변경 함수를 제1 디코딩된 크로마 잔차 데이터(즉, 비디오 디코딩의 컨텍스트에서 제1 디코딩된 변경된 크로마 잔차 데이터)에 적용하여 제1 변경된 디코딩된 크로마 잔차 데이터(즉, 비디오 디코딩의 컨텍스트에서 제1 역 변경된 디코딩된 크로마 잔차 데이터)를 생성하는 단계; 제2 역 잔차 변경 함수를 제2 디코딩된 크로마 잔차 데이터(즉, 비디오 디코딩의 컨텍스트에서 제1 디코딩된 변경된 크로마 잔차 데이터)에 적용하여 제2 변경된 디코딩된 크로마 잔차 데이터(즉, 비디오 디코딩의 컨텍스트에서 제2 역 변경된 디코딩된 크로마 잔차 데이터)를 생성하는 단계로서, 제 1 디코딩된 크로마 잔차 데이터는 제 1 크로마 성분과 연관되고, 제 2 디코딩된 크로마 잔차 데이터는 제 2 크로마 성분과 연관되는, 상기 제2 변경된 디코딩된 크로마 잔차 데이터를 생성하는 단계; 및 제1 변경된 디코딩된 크로마 잔차 데이터 및 제2 변경된 디코딩된 크로마 잔차 데이터에 기초하여 비디오 데이터의 블록을 재구성하는 단계를 포함한다.

[0241] 실시예 2A: 실시예 1A 에 있어서, 블록을 재구성하는 단계는 블록의 제 1 재구성된 크로마 데이터를 획득하기 위해 제 1 변경된 디코딩된 크로마 잔차 데이터를 제 1 예측된 크로마 데이터에 가산하는 단계; 및 블록의 제 2 재구성된 크로마 데이터를 획득하기 위해 제 2 변경된 디코딩된 크로마 잔차 데이터를 제 2 예측된 크로마 데이터에 가산하는 단계를 포함한다.

[0242] 실시예 3A: 실시예 1A 내지 2A 중 어느 하나의 방법에서, 제1 역 잔차 변경 함수는: $invF1(x,y) = (resCr(x,y)' + resCb(x,y)' + B1) / M1$ 이고, 제 2 역 잔차 변경 함수는: $invF2(x,y) = (resCr(x,y)' - resCb(x,y)' + B2) / M2$ 이며, 여기서 $resCb(x,y)'$ 는 제 1 디코딩된 크로마 잔차 데이터이고, $resCr(x,y)'$ 는 제 2 디코딩된 크로마 잔차 데이터이고, B1 및 B2는 라운딩 항이고, 1/M1 및 1/M2는 정규화 인자이다.

[0243] 실시예 4A: 실시예 1A 내지 2A 중 어느 하나의 방법에서, 제1 역 잔차 변경 함수는: $invF1(x,y) = (resCb(x,y)' + resCr(x,y)' + B1) / M1$ 이고, 제 2 역 잔차 변경 함수는: $invF2(x,y) = (resCb(x,y)' - resCr(x,y)' + B2) / M2$ 이며, 여기서 $resCb(x,y)'$ 는 제 1 디코딩된 크로마 잔차 데이터이고, $resCr(x,y)'$ 는 제 2 디코딩된 크로마 잔차 데이터이고, B1 및 B2는 라운딩 항이고, 1/M1 및 1/M2는 정규화 인자이다.

[0244] 실시예 5A: 실시예 1A 내지 2A 중 어느 하나의 방법에서, 제1 역 잔차 변경 함수는: $invF1(x,y) = (resCr(x,y)' + B10) / M10 + (resCb(x,y)' + B11) / M11$ 이고, 제 2 역 잔차 변경 함수는: $invF2(x,y) = (resCr(x,y)' + B20) / M20 - resCb(x,y)' + B21) / M21$ 이며, 여기서 $resCb(x,y)'$ 는 제 1 디코딩된 크로마 잔차 데이터이고, $resCr(x,y)'$ 는 제 2 디코딩된 크로마 잔차 데이터이고, B10, B11, B20 및 B21은 라운딩 항이고, 1/M10, 1/M11, 1/M20 및 1/M21은 정규화 인자이다.

[0245] 실시예 6A: 실시예 1A 내지 5A 중 어느 하나의 방법에 있어서, 상기 방법은: 제1 변경된 디코딩된 크로마 잔차

데이터에 제1 역 크로마 잔차 스케일링(CRS) 프로세스를 적용하는 단계; 및 제2 역 CRS 프로세스를 제2 변경된 디코딩된 크로마 잔차 데이터에 적용하는 단계를 포함한다.

- [0246] 실시예 7A: 실시예 1A 내지 6A 중 어느 하나의 예에 있어서, 제1 역 잔차 변경 함수를 적용하는 단계는 제1 디코딩된 크로마 잔차 데이터를 역양자화한 후 그리고 역변환 프로세스를 적용하기 전에 제1 역 잔차 변경 함수를 적용하는 단계를 포함하고, 제2 역 잔차 변경 함수를 적용하는 단계는 제2 디코딩된 크로마 잔차 데이터를 역양자화한 후 그리고 역변환 프로세스를 적용하기 전에 제2 역 잔차 변경 함수를 적용하는 단계를 포함한다.
- [0247] 실시예 8A: 실시예 3A 내지 5A 중 어느 하나의 예에 있어서, $\text{resCr}(x,y)$ ' 또는 $\text{resCb}(x,y)$ '는 $\text{resY}(x,y)$ '로 치환되고, 여기서 $\text{resY}(x,y)$ '는 디코딩된 루마 잔차 데이터이다.
- [0248] 실시예 9A: 실시예 1A 내지 8A 중 어느 하나의 방법에 있어서, 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림으로부터, 제1 역 잔차 변경 함수 및 제2 역 잔차 변경 함수를 적용할지 여부를 나타내는 플래그를 획득하는 단계를 더 포함한다.
- [0249] 실시예 10A: 실시예 9A의 방법에 있어서, 제1 크로마 성분에 대한 코딩된 블록 플래그(CBF)가 참이고 제2 크로마 성분에 대한 CBF가 참이면 플래그가 시그널링된다.
- [0250] 실시예 11A: 실시예 10A의 방법에 있어서, 상기 플래그는 제1 플래그이고, 상기 방법은 비트스트림으로부터 제2 플래그를 획득하는 단계를 더 포함하고, 상기 제2 플래그는 변환 유닛에 대한 것이며, 상기 제2 플래그는 다음 중 어느 것이 적용되는지를 표시한다: (i) 제1 디코딩된 크로마 잔차 데이터 및 제2 디코딩된 크로마 잔차 데이터 모두가 시그널링되는 것, 및 (ii) 제1 디코딩된 크로마 잔차 데이터 및 제2 디코딩된 크로마 잔차 데이터 중 하나만 시그널링되는 것.
- [0251] 실시예 12A: 실시예 11A의 방법에 있어서, 비트스트림으로부터, 제1 디코딩된 크로마 잔차 데이터 및 제2 디코딩된 크로마 잔차 데이터 중 어느 것이 0으로 강제되는지를 나타내는 제3 플래그를 획득하는 단계를 더 포함하는, 방법.
- [0252] 실시예 13A: 실시예 9A 내지 12A 중 어느 한 예에 있어서, 제1 역 잔차 변경 함수 및 제2 역 잔차 변경 함수를 적용할지 여부를 표시하는 플래그는 제1 플래그이고, 상기 방법은: 비트스트림으로부터, 제1 크로마 성분에 대한 CBF 를 획득하는 단계; 제1 크로마 성분에 대한 CBF에 기초하여, 제1 플래그에 대한 컨텍스트를 선택하는 단계; 및 선택된 컨텍스트를 사용하여 제1 플래그를 엔트로피 디코딩하는 단계를 더 포함한다.
- [0253] 실시예 14A: 실시예 9A-12A 중 어느 하나의 예에 있어서, 상기 제1 역 잔차 변경 함수 및 상기 제2 역 잔차 변경 함수를 적용할지 여부를 표시하는 플래그는 제1 플래그이고, 상기 방법은: 제1 플래그에 기초하여, 제1 크로마 성분에 대한 CBF 에 대한 컨텍스트를 선택하는 단계; 및 그 선택된 컨텍스트를 사용하여 제1 크로마 성분에 대한 CBF 를 엔트로피 디코딩하는 단계를 더 포함한다.
- [0254] 실시예 15A: 실시예 1A-14A 중 어느 하나의 방법에 있어서, 복수의 이용가능한 제1 역 잔차 변경 함수 중에서 제1 역 잔차 변경 함수를 결정하는 단계; 및 복수의 이용 가능한 제2 역 잔차 변경 함수 중에서 제2 역 잔차 변경 함수를 결정하는 단계를 더 포함한다.
- [0255] 실시예 16A: 실시예 1A-15A 중 어느 하나의 방법에 있어서, 제1 역 잔차 변경 함수를 제1 디코딩된 크로마 잔차 데이터에 적용하기 전에: 양자화 파라미터 및 제1 양자화 파라미터 오프셋에 기초하여 제1 양자화된 크로마 변환 데이터를 역양자화하는 단계; 및 역 양자화된 제1 양자화된 크로마 변환 데이터에 역변환을 적용하여 제1 디코딩된 크로마 잔차 데이터를 생성하는 단계, 제2 역 잔차 변경 함수를 제2 디코딩된 크로마 잔차 데이터에 적용하기 이전에: 양자화 파라미터 및 제2 양자화 파라미터 오프셋에 기초하여 제2 양자화된 크로마 변환 데이터를 역양자화하는 단계; 및 역변환을 제2 역양자화된 제2 양자화된 크로마 변환 데이터에 적용하여 제2 디코딩된 크로마 잔차 데이터를 생성하는 단계를 더 포함한다.
- [0256] 실시예 17A: 실시예 16A의 방법에 있어서, 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림으로부터 제1 양자화 파라미터 오프셋 또는 제2 양자화 파라미터 오프셋 중 적어도 하나를 획득하는 단계를 더 포함하는, 방법.
- [0257] 실시예 18A: 실시예 1A-17A 중 어느 하나의 방법에 있어서, 블록의 하나 이상의 특성에 기초하여, 제1 역 잔차 변경 함수의 적용이 허용되지 않는지 여부를 결정하는 단계; 및 블록의 하나 이상의 특성에 기초하여, 제2 역 잔차 변경 함수의 적용이 허용되지 않는지 여부를 결정하는 단계를 더 포함한다.

- [0258] 실시예 1B: 비디오 데이터를 인코딩하는 방법으로서, 그 방법은 제 1 변경된 크로마 잔차 데이터를 생성하기 위해 제 1 크로마 잔차 데이터에 제 1 잔차 변경 함수를 적용하는 단계; 제1 변경된 크로마 잔차 데이터를 인코딩하는 단계; 제2 변경된 크로마 잔차 데이터를 생성하기 위해 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용하는 단계; 그리고 제2 변경된 크로마 잔차 데이터를 인코딩하는 단계를 포함하며, 제1 크로마 잔차 데이터는 제1 크로마 성분과 연관되고, 제2 크로마 잔차 데이터는 제2 크로마 성분과 연관된다.
- [0259] 실시예 2B: 실시예 1B에 있어서, 변경된 크로마 잔차 데이터를 인코딩하는 단계는 변환 데이터를 생성하기 위해 변경된 크로마 잔차 데이터에 순방향 변환을 적용하는 단계; 및 양자화된 변환 데이터를 생성하기 위해 변환 데이터를 양자화하는 단계를 포함한다.
- [0260] 실시예 3B: 실시예 1B-2B 중 어느 하나의 방법에서, 제1 잔차 변경 함수는: $F1(x,y) = (resCb(x,y) - resCr(x,y) + A1) / N1$ 이고, 제 2 잔차 변경 함수는: $F2(x,y) = (resCb(x,y) + resCr(x,y) + A2) / N2$ 이며, 여기서 $resCb(x,y)$ 는 제 1 크로마 잔차 데이터, $resCr(x,y)$ 는 제 2 크로마 잔차 데이터이고, $A1$ 및 $A2$ 는 라운딩 항이며, $1/N1$ 및 $1/N2$ 는 정규화 인자이다.
- [0261] 실시예 4B: 실시예 1B-2B 중 어느 하나의 방법에서, 제1 잔차 변경 함수는: $F1(x,y) = (resCb(x,y) + resCr(x,y) + A1) / N1$ 이고, 제 2 잔차 변경 함수는: $F2(x,y) = (resCb(x,y) - resCr(x,y) + A2) / N2$ 이며, 여기서 $resCb(x,y)$ 는 제 1 크로마 잔차 데이터, $resCr(x,y)$ 는 제 2 크로마 잔차 데이터이고, $A1$ 및 $A2$ 는 라운딩 항이며, $1/N1$ 및 $1/N2$ 는 정규화 인자이다.
- [0262] 실시예 5B: 실시예 1B-2B 중 어느 하나의 방법에서, 제1 잔차 변경 함수는: $F1(x,y) = (resCb(x,y) + A10) / N10 - (resCr(x,y) + A11) / N11$ 이고, 제 2 잔차 변경 함수는: $F2(x,y) = (resCb(x,y) + A20) / N20 + (resCr(x,y) + A21) / N21$ 이며, 여기서 $resCb(x,y)$ 는 제 1 크로마 잔차 데이터이고, $resCr(x,y)$ 는 제 2 크로마 잔차 데이터이고, $A10$, $A11$, $A20$, 및 $A21$ 는 라운딩 항이며, $1/N10$, $1/N11$, $1/N20$ 및 $1/N21$ 는 정규화 인자이다.
- [0263] 실시예 6B: 실시예 1B-5B 중 어느 하나의 방법에 있어서, 상기 방법은: 제1 크로마 잔차 데이터에 크로마 잔차 스케일링(CRS)을 적용하는 단계; 및 CRS를 제2 크로마 잔차 데이터에 적용하는 단계를 더 포함하고, 제1 잔차 변경 함수를 적용하는 단계는 CRS를 제1 크로마 잔차 데이터에 적용한 후 제1 잔차 변경 함수를 제1 크로마 잔차 데이터에 적용하는 단계를 포함하고, 제2 잔차 변경 함수를 적용하는 단계는 CRS를 제2 크로마 잔차 데이터에 적용한 후 제2 크로마 잔차 데이터에 제 2 잔차 변경 함수를 적용하는 단계를 포함한다.
- [0264] 실시예 7B: 실시예 1B-6B 중 어느 하나의 방법에서, 제1 잔차 변경 함수를 적용하는 단계는 제1 크로마 잔차 데이터를 양자화하기 전에 제1 크로마 잔차 데이터에 순방향 변환 프로세스를 적용하는 단계를 포함하고, 제2 잔차 변경 함수를 적용하는 단계는 제2 크로마 잔차 데이터를 양자화하기 전에 제2 크로마 잔차 데이터에 순방향 변환 프로세스를 적용하는 단계를 포함한다.
- [0265] 실시예 8B: 실시예 3B 내지 5B 중 어느 하나의 예에 있어서, $resCr(x,y)$ 또는 $resCb(x,y)$ 는 $resY(x,y)$ 로 치환되고, 여기서 $resY(x,y)$ 는 루마 잔차 데이터이다.
- [0266] 실시예 9B: 실시예 1B 내지 8B 중 어느 하나의 방법에 있어서, 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림에서, 제1 잔차 변경 함수 및 제2 잔차 변경 함수를 적용할지 여부를 나타내는 플래그를 시그널링하는 단계를 더 포함한다.
- [0267] 실시예 10B: 실시예 9B의 방법에 있어서, 제1 크로마 성분에 대한 코딩된 블록 플래그(CBF)가 참이고 제2 크로마 성분에 대한 CBF가 참이면 플래그가 시그널링된다.
- [0268] 실시예 11B: 실시예 10B의 방법에 있어서, 상기 플래그는 제1 플래그이고, 상기 방법은, 비트스트림에서, 제2 플래그를 포함하는 단계를 더 포함하고, 상기 제2 플래그는 변환 유닛에 대한 것이며, 상기 제2 플래그는 다음 중 어느 것이 적용되는지를 표시한다: (i) 제1 크로마 잔차 데이터 및 제2 크로마 잔차 데이터 모두가 시그널링되는 것, 및 (ii) 제1 크로마 잔차 데이터 및 제2 크로마 잔차 데이터 중 하나만 시그널링되는 것.
- [0269] 실시예 12B: 실시예 11B의 방법에 있어서, 비트스트림에서, 제1 크로마 잔차 데이터 및 제2 크로마 잔차 데이터 중 어느 것이 0으로 강제되는지를 나타내는 제3 플래그를 포함하는 단계를 더 포함하는, 방법.
- [0270] 실시예 13B: 실시예 9B 내지 12B 중 어느 한 예에 있어서, 제1 잔차 변경 함수 및 제2 잔차 변경 함수를 적용할지 여부를 표시하는 플래그는 제1 플래그이고, 상기 방법은: 비트스트림에서, 제1 크로마 성분에 대한 CBF를 포함하는 단계; 제1 크로마 성분에 대한 CBF에 기초하여, 제1 플래그에 대한 컨텍스트를 선택하는 단계; 및 선

택된 컨텍스트를 사용하여 제1 플래그를 엔트로피 인코딩하는 단계를 더 포함한다.

- [0271] 실시예 14B: 실시예 9B-12B 중 어느 하나의 예에 있어서, 상기 제1 잔차 변경 함수 및 상기 제2 잔차 변경 함수를 적용할지 여부를 표시하는 플래그는 제1 플래그이고, 상기 방법은: 제1 플래그에 기초하여, 제1 크로마 성분 에 대한 CBF 에 대한 컨텍스트를 선택하는 단계; 및 그 선택된 컨텍스트를 사용하여 제1 크로마 성분에 대한 CBF 를 엔트로피 인코딩하는 단계를 더 포함한다.
- [0272] 실시예 15B: 실시예 1B-14B 중 어느 하나의 방법에 있어서, 복수의 이용가능한 제1 잔차 변경 함수 중에서 제1 잔차 변경 함수를 결정하는 단계; 및 복수의 이용 가능한 제2 잔차 변경 함수 중에서 제2 잔차 변경 함수를 결정하는 단계를 더 포함한다.
- [0273] 실시예 16B: 실시예 1B-15B 중 어느 하나의 방법에 있어서, 제1 잔차 변경 함수를 제1 크로마 잔차 데이터에 적용한 후: 제1 크로마 변환 데이터를 생성하기 위해 변환을 제1 크로마 잔차 데이터에 적용하는 단계; 및 양자화 파라미터 및 제1 양자화 파라미터 오프셋에 기초하여 제1 크로마 변환 데이터를 양자화하는 단계; 및 제2 크로마 잔차 데이터에 제2 잔차 변경 함수를 적용한 후: 제2 크로마 잔차 데이터에 변환을 적용하여 제2 크로마 변환 데이터를 생성하는 단계; 및 양자화 파라미터 및 제2 양자화 파라미터 오프셋에 기초하여 제1 크로마 변환 데이터를 양자화하는 단계를 더 포함한다.
- [0274] 실시예 17B: 실시예 16B 의 방법에 있어서, 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림에서, 제1 양자화 파라미터 오프셋 또는 제2 양자화 파라미터 오프셋 중 적어도 하나를 시그널링하는 단계를 더 포함한다.
- [0275] 실시예 18B: 실시예 1B-17B 중 어느 하나의 방법에 있어서, 블록의 하나 이상의 특성에 기초하여, 제1 잔차 변경 함수의 적용이 허용되지 않는지 여부를 결정하는 단계; 및 블록의 하나 이상의 특성에 기초하여, 제2 잔차 변경 함수의 적용이 허용되지 않는지 여부를 결정하는 단계를 더 포함한다.
- [0276] 실시예 1C 비디오 데이터를 코딩하기 위한 디바이스로서, 그 디바이스는 실시예 1A-18A 또는 1B-18B 중 어느 한 예의 방법을 수행하기 위한 하나 이상의 수단을 포함한다.
- [0277] 실시예 2C 실시예 1C 의 디바이스로서, 그 하나 이상의 수단들은 회로로 구현되는 하나 이상의 프로세서들을 포함한다.
- [0278] 실시예 3C 실시예들 1C-2C 중 임의의 것의 디바이스에서, 비디오 데이터를 저장하기 위한 메모리를 더 포함한다.
- [0279] 실시예 4C 실시예들 1C-3C 중 임의의 것의 디바이스에서, 디코딩된 비디오 데이터를 디스플레이하도록 구성된 디스플레이를 더 포함한다.
- [0280] 실시예 5C 실시예들 1C-4C 중 임의의 것의 디바이스에서, 그 디바이스는 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스 또는 셋톱 박스 중 하나 이상을 포함한다.
- [0281] 실시예 6C 실시예들 1C-5C 중 임의의 것의 디바이스에서, 디바이스가 비디오 디코더를 포함한다.
- [0282] 실시예 7C 실시예들 1C-6C 중 임의의 것의 디바이스에서, 디바이스가 비디오 인코더를 포함한다.
- [0283] 실시예 8C 실행될 때 하나 이상의 프로세서들로 하여금 실시예들 1A-18A 또는 1B-18B 중 임의의 것의 방법을 수행하게 하는 명령들을 저장한 컴퓨터 판독가능 저장 매체.
- [0284] 예에 의존하여, 본 명세서에서 설명된 기법들의 임의의 특정 행위들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있고, 전체적으로 부가되거나 병합되거나 또는 제거될 수도 있음 (예를 들어, 설명된 모든 행위들 또는 이벤트들이 그 기법들의 실시를 위해 필수적인 것은 아님) 이 인식되어야 한다. 더욱이, 특정 예들에 있어서, 행위들 또는 이벤트들은 순차적인 것보다는, 예를 들어, 다중-스레딩된 프로세싱, 인터럽트 프로세싱, 또는 다중의 프로세서들을 통해 동시에 수행될 수도 있다.
- [0285] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 그 기능들은 컴퓨터 판독가능 매체 상의 하나 이상의 명령 또는 코드로서 저장되거나 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예를 들면, 통신 프로토콜에 따라, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체는 일반적으로, (1) 비일시적인 유형의 컴

퓨터 판독가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품이 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0286] 제한이 아닌 일 예로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 원하는 프로그램 코드를 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 커넥션이 컴퓨터 판독가능 매체로 적절히 명명된다. 예를 들어, 동축 케이블, 광섬유 케이블, 꼬임쌍선, 디지털 가입자 라인 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 소프트웨어가 송신된다면, 동축 케이블, 광섬유 케이블, 꼬임쌍선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속, 캐리어 파, 신호 또는 다른 일시적 매체를 포함하는 것이 아니라, 대신에 비일시적, 유형의 저장 매체에 관련된다는 것이 이해되어야 한다. 본 명세서에서 사용된 바와 같은 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서, 디스크 (disk) 는 통상적으로 데이터를 자기적으로 재생하지만 디스크 (disc) 는 레이저를 이용하여 데이터를 광학적으로 재생한다. 상기의 조합들이 또한, 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

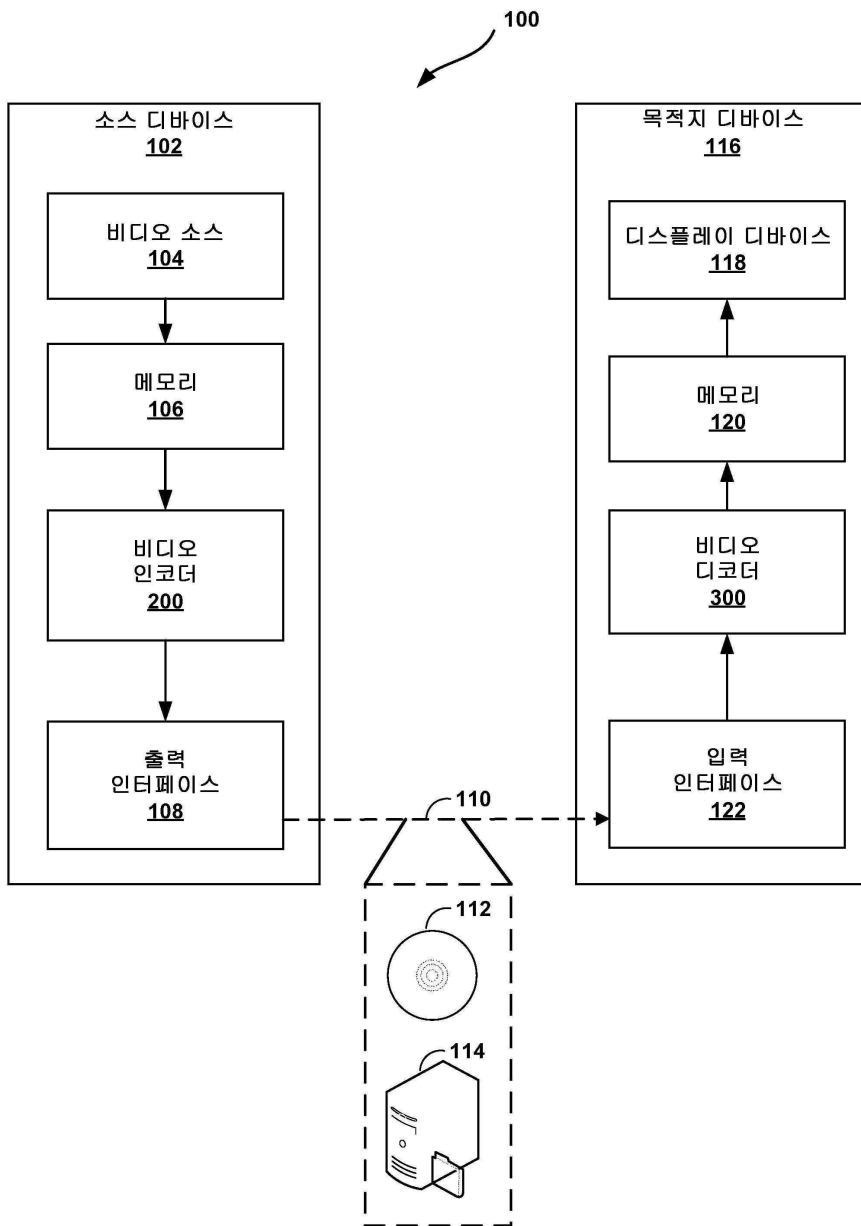
[0287] 명령들은 하나 이상의 프로세서, 예컨대 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 또는 다른 등가 집적 또는 이산 로직 회로부에 의해 실행될 수도 있다. 이에 따라, 본 명세서에서 사용된 바와 같은 용어들 "프로세서" 및 "프로세싱 회로부" 는 전술한 구조들 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 추가로, 일부 양태들에서, 본 명세서에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 통합될 수도 있다. 또한, 그 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0288] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들면, 칩 세트) 를 포함하는, 매우 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시에 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하는 것은 아니다. 오히려, 상기 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에서 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상기 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호동작가능한 하드웨어 유닛들의 콜렉션에 의해 제공될 수도 있다.

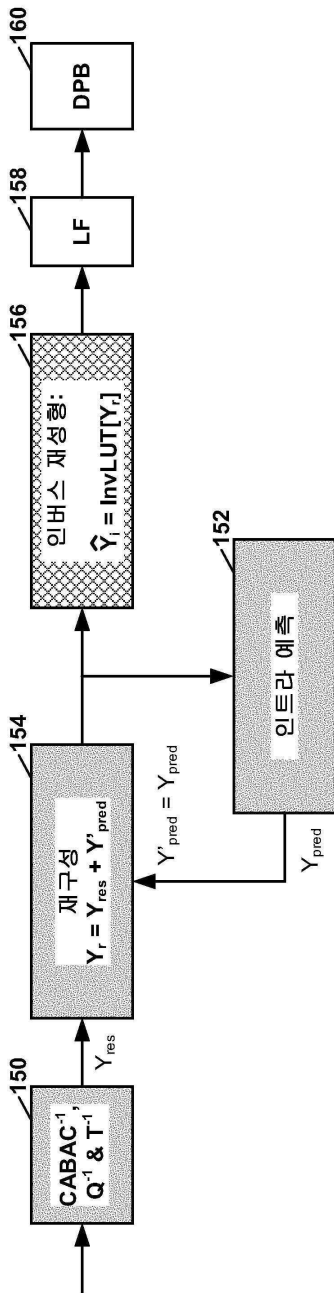
[0289] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

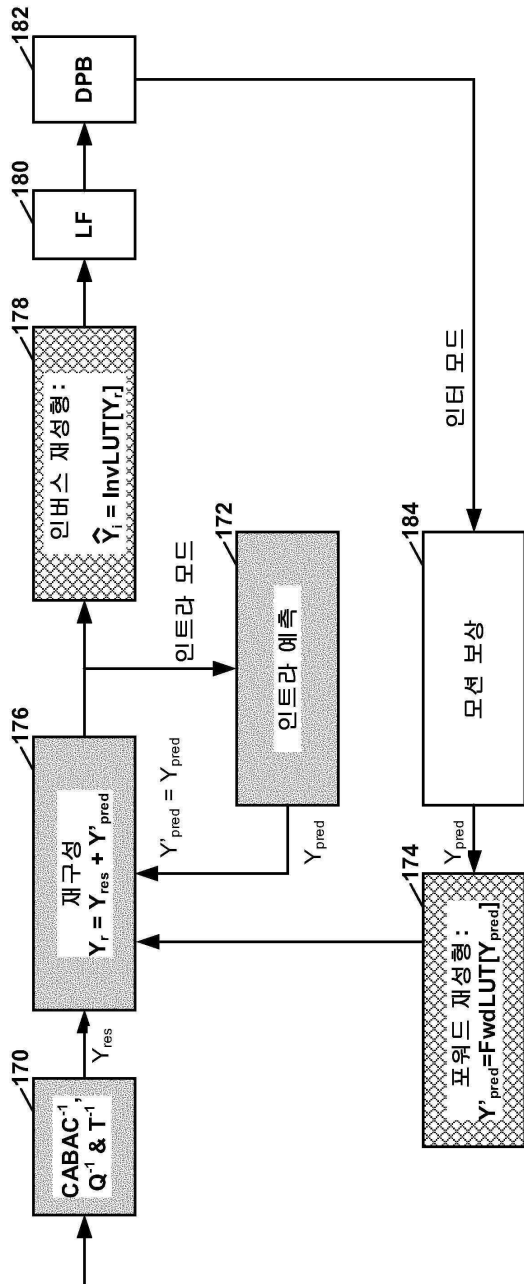
도면1



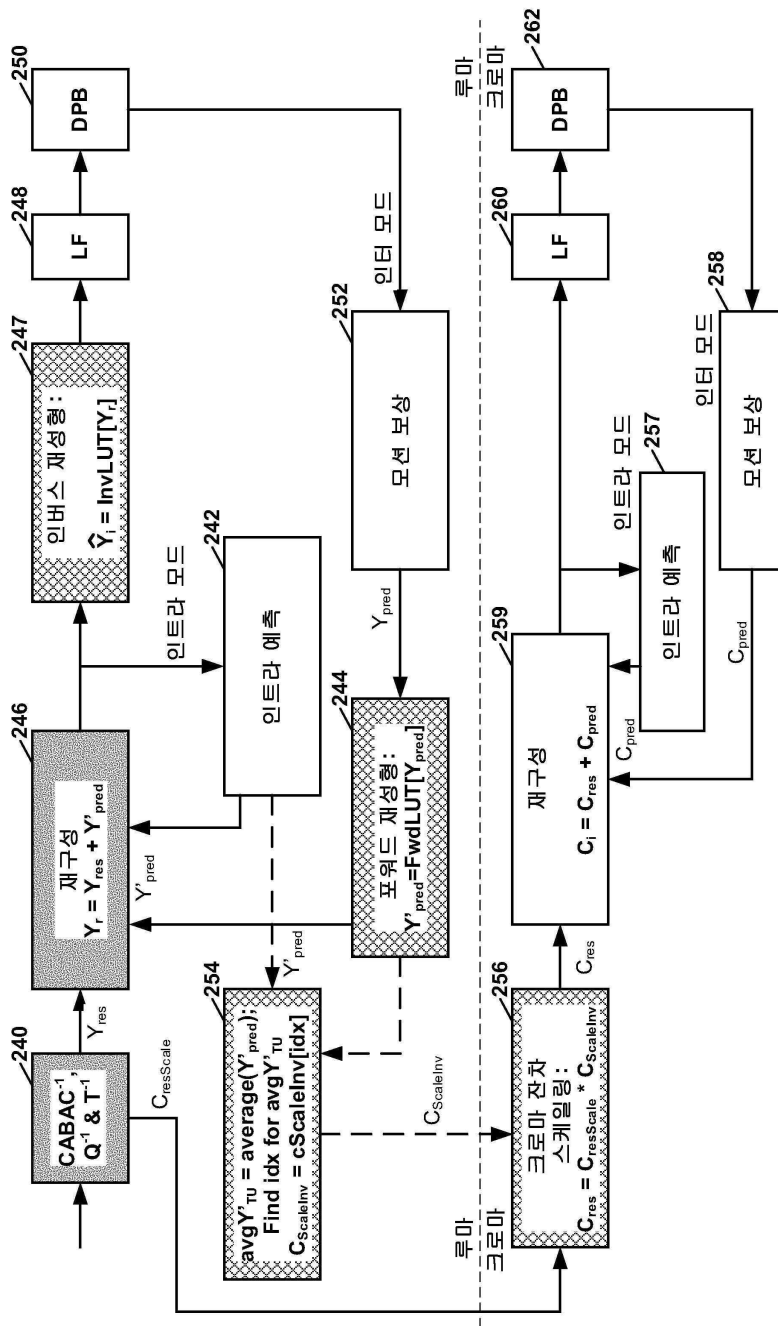
도면2



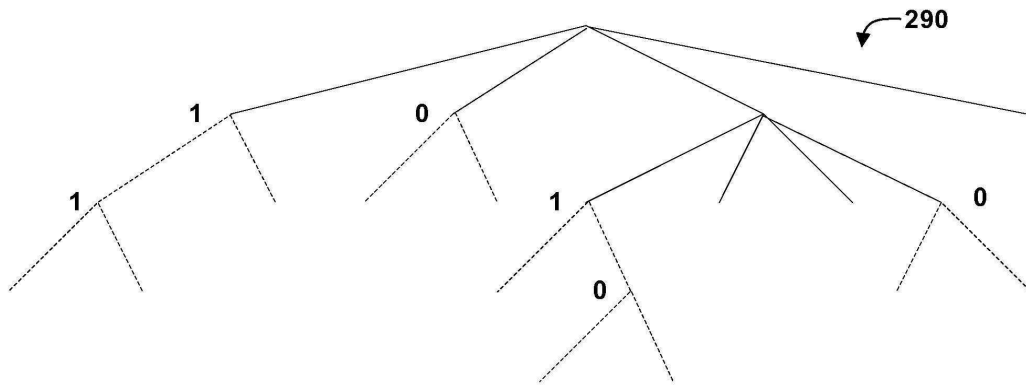
도면3



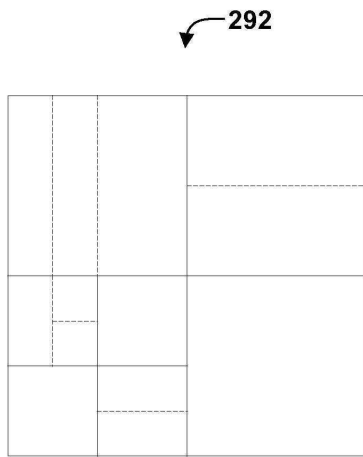
도면4



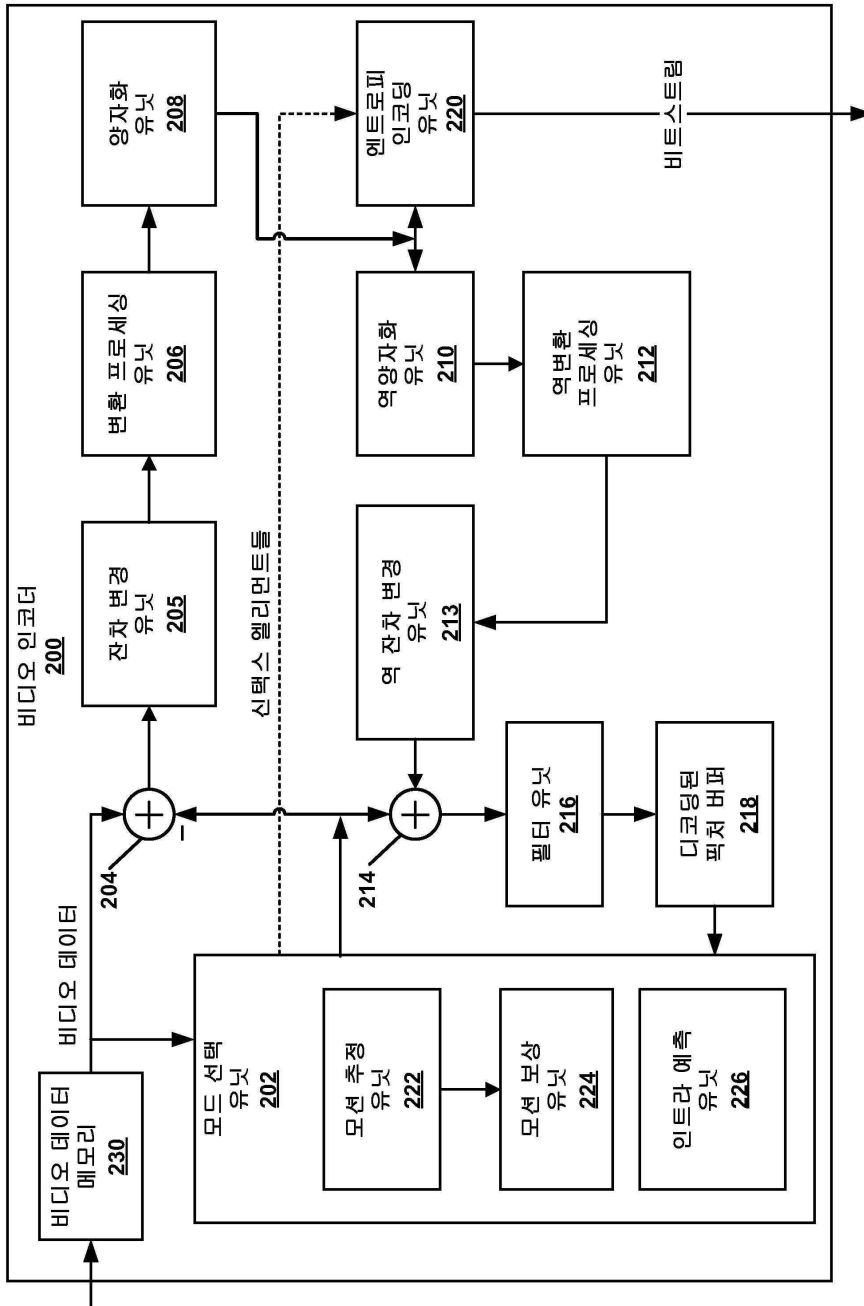
도면5a



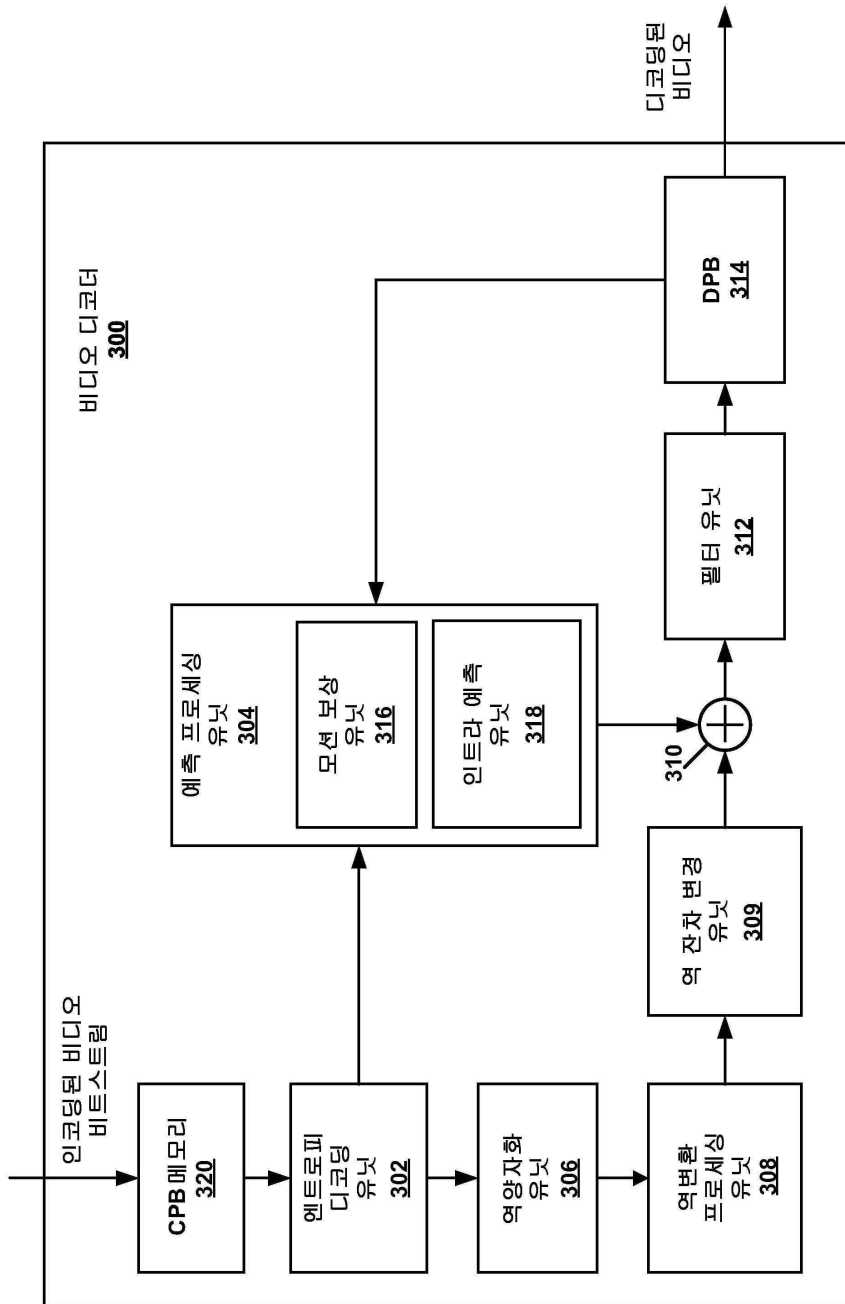
도면5b



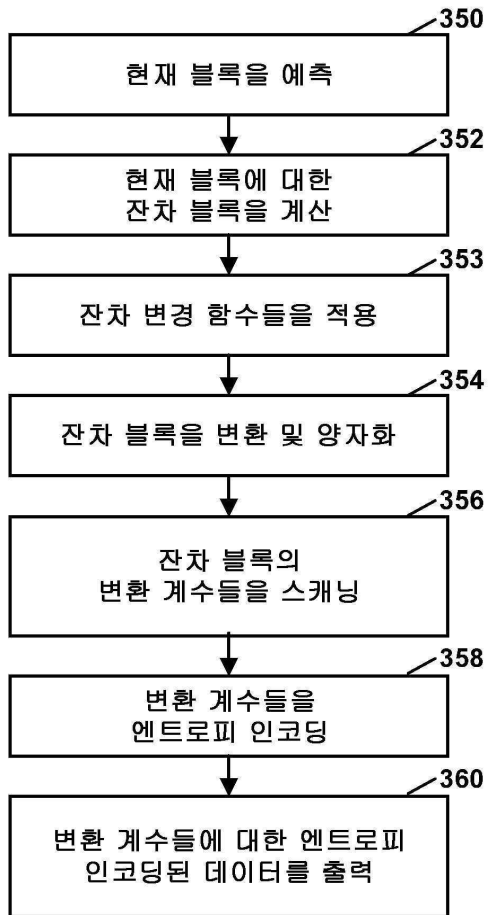
도면6



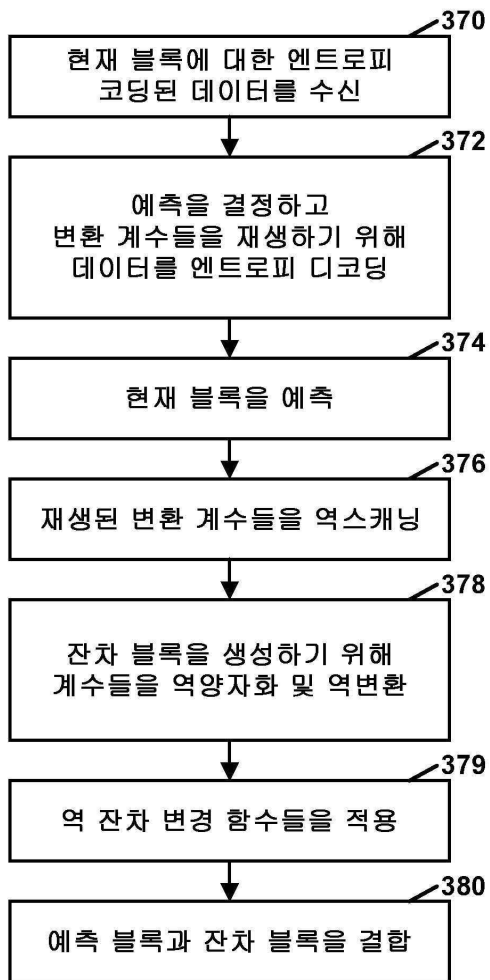
도면7



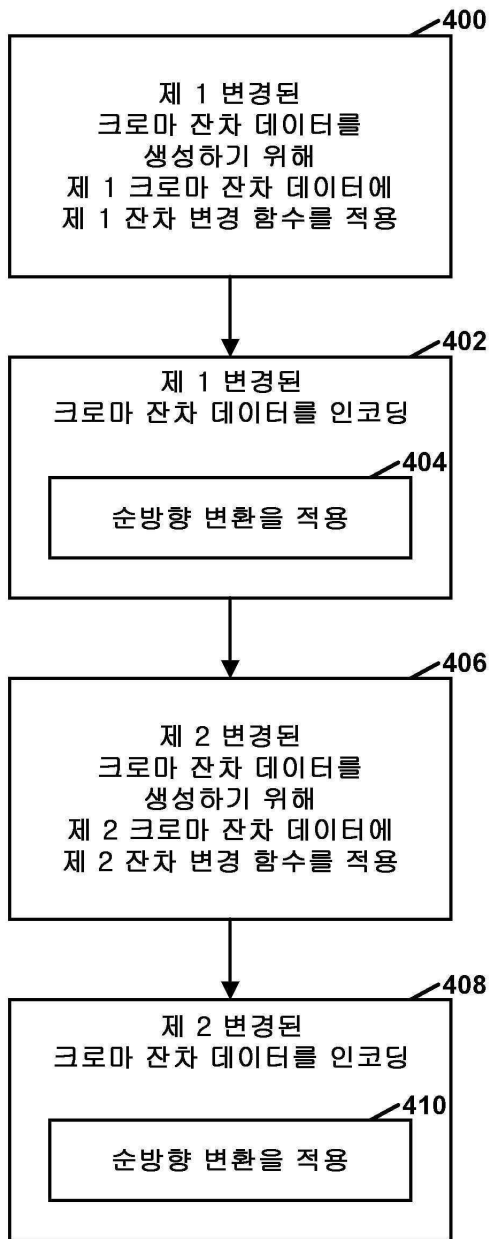
도면8



도면9



도면10



도면11

