



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년01월06일
(11) 등록번호 10-2348541
(24) 등록일자 2022년01월04일

(51) 국제특허분류(Int. Cl.)
H04N 19/186 (2014.01) H04N 19/11 (2014.01)
H04N 19/176 (2014.01) H04N 19/184 (2014.01)
H04N 19/33 (2014.01) H04N 19/44 (2014.01)
(52) CPC특허분류
H04N 19/186 (2015.01)
H04N 19/11 (2015.01)
(21) 출원번호 10-2016-7026049
(22) 출원일자(국제) 2015년03월13일
심사청구일자 2020년02월26일
(85) 번역문제출일자 2016년09월21일
(65) 공개번호 10-2016-0132866
(43) 공개일자 2016년11월21일
(86) 국제출원번호 PCT/US2015/020514
(87) 국제공개번호 WO 2015/138943
국제공개일자 2015년09월17일
(30) 우선권주장
61/953,573 2014년03월14일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
D. Marpe, et al. MB-adaptive residual colour
transform for 4:4:4 coding. JVT of ISO/IEC
MPEG and ITU-T VCEG. JVT-R071, Jan. 15, 2006,
pp.1-11*
JP2011091732 A
W02007034601 A1
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
켈컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하
우스 드라이브 5775
(72) 발명자
장 리
미국 92121-1714 캘리포니아주 샌디에고 모어하
우스 드라이브 5775
천 지안레
미국 92121-1714 캘리포니아주 샌디에고 모어하
우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 50 항

심사관 : 이상래

(54) 발명의 명칭 블록 적응적 컬러-공간 변환 코딩

(57) 요약

일반적으로, 이 개시물은 컬러-공간 변환 프로세스를 이용하여 비디오 블록들을 코딩하기 위한 기법들을 설명한다. 비디오 인코더 또는 비디오 디코더와 같은 비디오 코더는 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정할 수도 있고, 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 선택스 엘리먼트의 값을 설정

(뒷면에 계속)

대표도 - 도9



할 수도 있다. 비디오 코더는 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용할 수도 있다. 비디오 코더는 코딩 유닛의 선택스 엘리먼트를 디코딩할 수도 있다. 비디오 코더는 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정할 수도 있다. 비디오 코더는 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용할 수도 있다.

(52) CPC특허분류

HO4N 19/176 (2015.01)

HO4N 19/184 (2015.01)

HO4N 19/33 (2015.01)

HO4N 19/44 (2015.01)

HO4N 19/70 (2015.01)

(72) 발명자

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

솔레 로할스 호엘

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

김 우식

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장

61/981,645 2014년04월18일 미국(US)

62/062,637 2014년10월10일 미국(US)

14/656,516 2015년03월12일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

비디오 데이터의 코딩 유닛의 선택스 엘리먼트를 디코딩하는 단계로서, 상기 선택스 엘리먼트는 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하고, 상기 코딩 유닛은 쿼트트리 데이터 구조의 노드를 포함하는, 상기 선택스 엘리먼트를 디코딩하는 단계;

상기 선택스 엘리먼트의 값이 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하는 단계; 및

상기 선택스 엘리먼트가 상기 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여, 상기 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하는 단계를 포함하고,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라 코딩될 때, 그리고 상기 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 단계는 상기 선택스 엘리먼트의 상기 값을 추론하는 단계를 포함하거나, 또는

상기 선택스 엘리먼트는 상기 코딩 유닛이 팔레트 모드로 코딩될 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 단계는 상기 선택스 엘리먼트의 상기 값을 추론하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 선택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 코딩 유닛은 인트라-펄스 코드 변조 (intra-pulse code modulation; IPCM) 모드 외의 모드에서 코딩되고, 상기 선택스 엘리먼트는 상기 IPCM 모드 외의 모드를 이용하는 코딩 유닛들에 대해서만 시그널링되는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

삭제

청구항 6

삭제

청구항 7

제 1 항에 있어서,

상기 선택스 엘리먼트에 대한 1 의 값은 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 상기 제 1 선택스 엘리먼트는 상기 코딩 유닛과 연관되고, 상기 방법은,

제 2 선택스 엘리먼트를 디코딩하는 단계로서, 상기 제 2 선택스 엘리먼트는 픽처 파라미터 세트 (PPS) 와 연관되고, 상기 제 2 선택스 엘리먼트는 컬러-공간 변환이 상기 PPS 에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시하는, 상기 제 2 선택스 엘리먼트를 디코딩하는 단계; 및

상기 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정하는 단계를 더 포함하고,

상기 제 1 선택스 엘리먼트를 디코딩하는 단계는, 상기 제 2 선택스 엘리먼트가 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 상기 제 1 선택스 엘리먼트를 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 컬러-공간 역변환 프로세스를 적용하는 단계는, 고정된 컬러-공간 역변환 행렬을 적용하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 10

제 9 항에 있어서,

상기 고정된 컬러-공간 역변환 행렬은 YCoCg 행렬인, 비디오 데이터를 디코딩하는 방법.

청구항 11

제 9 항에 있어서,

상기 고정된 컬러-공간 역변환 행렬은 YCoCg-R 행렬인, 비디오 데이터를 디코딩하는 방법.

청구항 12

제 1 항에 있어서,

상기 선택스 엘리먼트가 상기 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여,

컬러-공간 역변환 프로세스를 상기 비디오 데이터에 적용하기 전에, 하나 이상의 컴포넌트들의 값들을 이용하여 인트라-코딩된 잔차 블록을 생성하는 단계; 및

상기 컬러-공간 역변환 프로세스를 상기 비디오 데이터의 상기 인트라-코딩된 잔차 블록에 적용하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 13

비디오 디코딩 디바이스로서,

비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

비디오 데이터의 코딩 유닛의 선택스 엘리먼트를 디코딩하는 것으로서, 상기 선택스 엘리먼트는 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하고, 상기 코딩 유닛은 쿼드트리 데이터 구조의 노드를 포함하는, 상기 선택스 엘리먼트를 디코딩하고;

상기 선택스 엘리먼트의 값이 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하고; 그리고

상기 선택스 엘리먼트가 상기 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여, 상기 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하도록 구성되고,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라 코딩될 때, 그리고 상기 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 단계는 상기 선택스 엘리먼트의 상기 값을 추론하는 단계를 포함하거나, 또는

상기 선택스 엘리먼트는 상기 코딩 유닛이 팔레트 모드로 코딩될 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 단계는 상기 선택스 엘리먼트의 상기 값을 추론하는 단계를 포함하는, 비디오 디코딩 디바이스.

청구항 14

제 13 항에 있어서,

상기 선택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 디코딩 디바이스.

청구항 15

제 13 항에 있어서,

상기 코딩 유닛은 인트라-펄스 코드 변조 (IPCM) 모드 외의 모드에서 코딩되고, 상기 선택스 엘리먼트는 상기 IPCM 모드 외의 모드를 이용하는 코딩 유닛들에 대해서만 시그널링되는, 비디오 디코딩 디바이스.

청구항 16

제 13 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 비디오 디코딩 디바이스.

청구항 17

삭제

청구항 18

삭제

청구항 19

제 13 항에 있어서,

상기 선택스 엘리먼트에 대한 1 의 값은 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 비디오 디코딩 디바이스.

청구항 20

제 13 항에 있어서,

상기 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 상기 제 1 선택스 엘리먼트는 상기 코딩 유닛과 연관되고, 상기 하나 이상의 프로세서들은,

제 2 선택스 엘리먼트를 디코딩하는 것으로서, 상기 제 2 선택스 엘리먼트는 픽처 파라미터 세트 (PPS) 와 연관되고, 상기 제 2 선택스 엘리먼트는 컬러-공간 변환이 상기 PPS 에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시하는, 상기 제 2 선택스 엘리먼트를 디코딩하고; 그리고

상기 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정하도록 추가로 구성되고,

상기 하나 이상의 프로세서들이 상기 제 1 선택스 엘리먼트를 디코딩하도록 구성되는 것은, 상기 하나 이상의 프로세서들이 상기 제 2 선택스 엘리먼트가 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 상기 제 1 선택스 엘리먼트를 디코딩하도록 구성되는 것을 포함하는, 비디오 디코딩 디바이스.

청구항 21

제 13 항에 있어서,

상기 하나 이상의 프로세서들이 상기 컬러-공간 역변환 프로세스를 적용하도록 구성되는 것은, 상기 하나 이상의 프로세서들이 고정된 컬러-공간 역변환 행렬을 적용하도록 구성되는 것을 포함하는, 비디오 디코딩 디바이스.

청구항 22

제 21 항에 있어서,

상기 고정된 컬러-공간 역변환 행렬은 YCoCg 행렬인, 비디오 디코딩 디바이스.

청구항 23

제 21 항에 있어서,

상기 고정된 컬러-공간 역변환 행렬은 YCoCg-R 행렬인, 비디오 디코딩 디바이스.

청구항 24

제 13 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 선택스 엘리먼트가 상기 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여,

컬러-공간 역변환 프로세스를 상기 비디오 데이터에 적용하기 전에, 하나 이상의 컴포넌트들의 값들을 이용하여 잔차 블록을 생성하고; 그리고

상기 컬러-공간 역변환 프로세스를 상기 비디오 데이터의 상기 잔차 블록에 적용하도록 추가로 구성되는, 비디오 디코딩 디바이스.

청구항 25

비디오 디코딩 장치로서,

비디오 데이터의 코딩 유닛의 선택스 엘리먼트를 디코딩하기 위한 수단으로서, 상기 선택스 엘리먼트는 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하고, 상기 코딩 유닛은 쿼드트리 데이터 구조의 노드를 포함하는, 상기 선택스 엘리먼트를 디코딩하기 위한 수단;

상기 선택스 엘리먼트의 값이 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하기 위한 수단; 및

상기 선택스 엘리먼트가 상기 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 상기 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하기 위한 수단

포함하고,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라 코딩될 때, 그리고 상기 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 것은 상기 선택스 엘리먼트의 상기 값을 추론하는 것을 포함하거나, 또는

상기 선택스 엘리먼트는 상기 코딩 유닛이 팔레트 모드로 코딩될 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 것은 상기 선택스 엘리먼트의 상기 값을 추론하는 것을 포함하는, 비디오 디코딩 장치.

청구항 26

제 25 항에 있어서,

상기 코딩 유닛은 인트라-펄스 코드 변조 (IPCM) 모드 외의 모드에서 코딩되고, 상기 선택스 엘리먼트는 상기 IPCM 모드 외의 모드를 이용하는 코딩 유닛들에 대해서만 시그널링되는, 비디오 디코딩 장치.

청구항 27

제 25 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 비디오 디코딩 장치.

청구항 28

삭제

청구항 29

삭제

청구항 30

제 25 항에 있어서,

상기 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 상기 제 1 선택스 엘리먼트는 상기 코딩 유닛과 연관되고, 상기 장치는,

제 2 선택스 엘리먼트를 디코딩하기 위한 수단으로서, 상기 제 2 선택스 엘리먼트는 픽처 파라미터 세트 (PPS) 와 연관되고, 상기 제 2 선택스 엘리먼트는 컬러-공간 변환이 상기 PPS 에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시하는, 상기 제 2 선택스 엘리먼트를 디코딩하기 위한 수단; 및

상기 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정하기 위한 수단을 더 포함하고,

상기 제 1 선택스 엘리먼트를 디코딩하기 위한 수단은 상기 제 2 선택스 엘리먼트가 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 상기 제 1 선택스 엘리먼트를 디코딩하기 위한 수단을 포함하는, 비디오 디코딩 장치.

청구항 31

제 25 항에 있어서,

상기 컬러-공간 역변환 프로세스를 적용하기 위한 수단은 고정된 컬러-공간 역변환 행렬을 적용하기 위한 수단을 포함하는, 비디오 디코딩 장치.

청구항 32

명령들을 포함하는 컴퓨터-판독가능 저장 매체로서,

상기 명령들은, 실행될 경우, 비디오 디코딩 디바이스의 하나 이상의 프로세서들로 하여금,

비디오 데이터의 코딩 유닛의 선택스 엘리먼트를 디코딩하게 하는 것으로서, 상기 선택스 엘리먼트는 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하고, 상기 코딩 유닛은 쿼드트리 데이터 구조의 노드를 포함하는, 상기 선택스 엘리먼트를 디코딩하게 하고;

상기 선택스 엘리먼트의 값이 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하게 하고; 그리고

상기 선택스 엘리먼트가 상기 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여, 상기 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하게 하고,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라 코딩될 때, 그리고 상기 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 것은 상기 선택스 엘리먼트의 상기 값을 추론하는 것을 포함하거나, 또는

상기 선택스 엘리먼트는 상기 코딩 유닛이 팔레트 모드로 코딩될 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하고, 상기 선택스 엘리먼트는 상기 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않고, 상기 선택스 엘리먼트를 디코딩하는 것은 상기 선택스 엘리먼트의 상기 값을 추론하는 것을 포함하는, 컴퓨터-판독가능 저장 매체.

청구항 33

제 32 항에 있어서,

상기 코딩 유닛은 인트라-펄스 코드 변조 (IPCM) 모드 외의 모드에서 코딩되고, 상기 선택스 엘리먼트는 상기 IPCM 모드 외의 모드를 이용하는 코딩 유닛들에 대해서만 시그널링되는, 컴퓨터-판독가능 저장 매체.

청구항 34

제 32 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 컴퓨터-판독가능 저장 매체.

청구항 35

삭제

청구항 36

삭제

청구항 37

제 32 항에 있어서,

상기 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 상기 제 1 선택스 엘리먼트는 상기 코딩 유닛과 연관되고, 상기 실행된 명령들은 추가로, 상기 하나 이상의 프로세서들로 하여금,

제 2 선택스 엘리먼트를 디코딩하게 하는 것으로서, 상기 제 2 선택스 엘리먼트는 픽처 파라미터 세트 (PPS) 와 연관되고, 상기 제 2 선택스 엘리먼트는 컬러-공간 변환이 상기 PPS 에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시하는, 상기 제 2 선택스 엘리먼트를 디코딩하게 하고; 그리고

상기 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정하게 하고,

상기 하나 이상의 프로세서들로 하여금, 상기 제 1 선택스 엘리먼트를 디코딩하게 하는 상기 실행된 명령들은, 상기 하나 이상의 프로세서들에 의해 실행될 경우, 상기 하나 이상의 프로세서들로 하여금, 상기 제 2 선택스 엘리먼트가 컬러-공간 변환이 상기 PPS 에 대응하는 상기 비디오 데이터에 대하여 인에이블된다는 것을 표시하

는 것으로 결정하는 것에 응답하여 상기 제 1 선택스 엘리먼트를 디코딩하게 하는 명령들을 포함하는, 컴퓨터-판독가능 저장 매체.

청구항 38

제 32 항에 있어서,

상기 하나 이상의 프로세서들로 하여금, 상기 컬러-공간 역변환 프로세스를 적용하게 하는 상기 실행된 명령들은, 상기 하나 이상의 프로세서들에 의해 실행될 경우, 상기 하나 이상의 프로세서들로 하여금, 고정된 컬러-공간 역변환 행렬을 적용하게 하는 명령들을 포함하는, 컴퓨터-판독가능 저장 매체.

청구항 39

비디오 데이터를 인코딩하는 방법으로서,

코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하는 단계로서, 상기 코딩 유닛은 쿼드트리 데이터 구조의 노드를 포함하는, 상기 컬러-공간 변환을 이용할 것인지 여부를 결정하는 단계;

상기 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 컬러-공간 변환의 상기 이용을 표시하도록 상기 코딩 유닛의 선택스 엘리먼트의 값을 설정하는 단계; 및

상기 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용하는 단계를 포함하고,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라 코딩될 때, 그리고 상기 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 시그널링되지 않거나, 또는

상기 선택스 엘리먼트는 상기 코딩 유닛이 팔레트 모드로 코딩될 때에 시그널링되지 않는, 비디오 데이터를 인코딩하는 방법.

청구항 40

제 39 항에 있어서,

상기 선택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 41

제 39 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라-펄스 코드 변조 (IPCM) 모드 외의 모드를 이용하여 코딩될 때에 시그널링되는, 비디오 데이터를 인코딩하는 방법.

청구항 42

제 39 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에만 시그널링되는, 비디오 데이터를 인코딩하는 방법.

청구항 43

삭제

청구항 44

삭제

청구항 45

제 39 항에 있어서,

상기 선택스 엘리먼트에 대한 1 의 값은 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 비디오 데이터를 인코딩하는 방법.

청구항 46

제 39 항에 있어서,

상기 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 상기 제 1 선택스 엘리먼트는 상기 코딩 유닛과 연관되고, 상기 방법은,

픽처 파라미터 세트를 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하는 단계;

상기 픽처 파라미터 세트를 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 제 2 선택스 엘리먼트의 값을 설정하는 단계로서, 상기 제 2 선택스 엘리먼트는 상기 픽처 파라미터 세트와 연관되고, 상기 제 2 선택스 엘리먼트는 컬러-공간 변환이 상기 픽처 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정되는, 상기 제 2 선택스 엘리먼트의 값을 설정하는 단계; 및

컬러-공간 변환 프로세스를 상기 픽처 파라미터 세트의 상기 비디오 데이터에 적용하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 47

제 39 항에 있어서,

상기 컬러-공간 변환 프로세스를 적용하는 단계는 컬러-공간 변환 행렬을 적용하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 48

제 47 항에 있어서,

상기 컬러-공간 변환 행렬은 YCoCg 행렬인, 비디오 데이터를 인코딩하는 방법.

청구항 49

제 47 항에 있어서,

상기 컬러-공간 변환 행렬은 YCoCg-R 행렬인, 비디오 데이터를 인코딩하는 방법.

청구항 50

제 39 항에 있어서,

상기 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여,

컬러-공간 변환 프로세스를 상기 비디오 데이터에 적용하기 전에, 하나 이상의 컴포넌트들의 값들을 이용하여 잔차 블록을 생성하는 단계; 및

상기 컬러-공간 변환 프로세스를 상기 비디오 데이터의 상기 잔차 블록에 적용하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 51

비디오 인코딩 디바이스로서,

비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하는 것으로서, 상기 코딩 유닛은 쿼드트리 데이터 구조의 노드를 포함하는, 상기 컬러-공간 변환을 이용할 것인지 여부를 결정하고;

상기 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 컬러-공간 변환의 상기 이용을 표시하도록 상기 코딩 유닛의 선택스 엘리먼트의 값을 설정하고; 그리고

상기 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용하도록 구성되고,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라 코딩될 때, 그리고 상기 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 시그널링되지 않거나, 또는

상기 선택스 엘리먼트는 상기 코딩 유닛이 팔레트 모드로 코딩될 때에 시그널링되지 않는, 비디오 인코딩 디바이스.

청구항 52

제 51 항에 있어서,

상기 선택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 인코딩 디바이스.

청구항 53

제 51 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛이 인트라-펄스 코드 변조 (IPCM) 모드 외의 모드를 이용하여 코딩될 때에 시그널링되는, 비디오 인코딩 디바이스.

청구항 54

제 51 항에 있어서,

상기 선택스 엘리먼트는 상기 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에만 시그널링되는, 비디오 인코딩 디바이스.

청구항 55

삭제

청구항 56

삭제

청구항 57

제 51 항에 있어서,

상기 선택스 엘리먼트에 대한 1 의 값은 상기 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는, 비디오 인코딩 디바이스.

청구항 58

제 51 항에 있어서,

상기 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 상기 제 1 선택스 엘리먼트는 상기 코딩 유닛과 연관되고, 상기 하나 이상의 프로세서들은,

픽처 파라미터 세트를 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하고;

상기 픽처 파라미터 세트를 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 제 2 선택스 엘리먼트의 값을 설정하는 것으로서, 상기 제 2 선택스 엘리먼트는 상기 픽처 파라미터 세트와 연관되고, 상기 제 2 선택스 엘리먼트는 컬러-공간 변환이 상기 픽처 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정되는, 상기 제 2 선택스 엘리먼트의 값을 설정하고; 그리고

컬러-공간 변환 프로세스를 상기 픽처 파라미터 세트의 상기 비디오 데이터에 적용하도록 추가로 구성되는, 비디오 인코딩 디바이스.

청구항 59

제 51 항에 있어서,

상기 하나 이상의 프로세서들이 상기 컬러-공간 변환 프로세스를 적용하도록 구성되는 것은, 상기 하나 이상의 프로세서들이 컬러-공간 변환 행렬을 적용하도록 구성되는 것을 포함하는, 비디오 인코딩 디바이스.

청구항 60

제 59 항에 있어서,

상기 컬러-공간 변환 행렬은 YCoCg 행렬인, 비디오 인코딩 디바이스.

청구항 61

제 59 항에 있어서,

상기 컬러-공간 변환 행렬은 YCoCg-R 행렬인, 비디오 인코딩 디바이스.

청구항 62

제 51 항에 있어서,

상기 하나 이상의 프로세서들은, 상기 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여,

컬러-공간 변환 프로세스를 상기 비디오 데이터에 적용하기 전에, 하나 이상의 컴포넌트들의 값들을 이용하여 잔차 블록을 생성하고; 그리고

상기 컬러-공간 변환 프로세스를 상기 비디오 데이터의 상기 잔차 블록에 적용하도록 추가로 구성되는, 비디오 인코딩 디바이스.

발명의 설명

기술 분야

[0001] 이 출원은 2014 년 3 월 14 일자로 출원된 미국 가출원 제 61/953,573 호, 2014 년 4 월 18 일자로 출원된 미국 가출원 제 61/981,645 호, 및 2014 년 10 월 10 일자로 출원된 미국 가출원 제 62/062,637 호의 이익을 주장하고, 그 각각의 전체 내용은 참조로 본원에 편입된다.

[0002] 이 개시물은 비디오 코딩에 관한 것으로, 더욱 구체적으로, 컬러-공간 변환을 이용한 비디오 코딩에 관한 것이다.

배경 기술

[0003] 디지털 비디오 기능들은, 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말 (personal digital assistant; PDA) 들, 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임 임용 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 및 기타 등등을 포함하는 광범위한 디바이스들 내에 편입될 수 있다.

디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, 진보된 비디오 코딩 (Advanced Video Coding; AVC) 에 의해 정의된 표준들, 현재 개발 중에 있는 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, 및 이러한 표준들의 확장들에서 설명된 것들과 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신하고, 수신하고, 인코딩하고, 디코딩하고, 및/또는 저장할 수도 있다.

[0004] 비디오 코딩 기법들은 비디오 시퀀스들에 내재된 중복성을 감소시키거나 제거하기 위한 공간적 (인트라-픽처 (intra-picture)) 예측 및/또는 시간적 (인터-픽처 (inter-picture)) 예측을 포함한다. 블록-기반 비디오 코딩을 위하여, 비디오 슬라이스 (예컨대, 비디오 프레임 또는 비디오 프레임의 부분) 는 비디오 블록들로 파티셔닝 (partitioning) 될 수도 있으며, 이 비디오 블록들은 또한, 트리블록 (treeblock) 들, 코딩 유닛 (coding unit; CU) 들 및/또는 코딩 노드들로서 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처에서의 이웃하는 블록들 내의 참조 샘플들에 대한 공간적 예측을 이용하여 인코딩된다.

픽처의 인터-코딩된 (P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽처에서의 이웃하는 블록들 내의 참

조 샘플들에 대한 공간적 예측, 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들로서 지칭될 수도 있고, 참조 픽처들은 참조 프레임들로서 지칭될 수도 있다.

[0005] 공간적 또는 시간적 예측은 코딩되어야 할 블록에 대한 예측 블록을 낳는다. 잔차 데이터 (residual data)는 코딩되어야 할 원래의 블록과 예측 블록과의 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터와, 코딩된 블록 및 예측 블록 사이의 차이를 표시하는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위하여, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어 잔차 변환 계수들을 낳을 수도 있고, 그 후 이들은 양자화될 수도 있다. 초기에 2 차원 어레이로 배치된 양자화된 변환 계수들은 변환 계수들의 1 차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩은 훨씬 더 많은 압축을 달성하기 위하여 적용될 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0006] 일반적으로, 이 개시물은 컬러-공간 변환 프로세스를 이용하여 비디오 블록들을 코딩하기 위한 기법들을 설명한다. 비디오 인코더 또는 비디오 디코더와 같은 비디오 코더는 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정할 수도 있다. 비디오 코더는 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 선택스 엘리먼트 (syntax element)의 값을 설정할 수도 있다. 비디오 코더는 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용할 수도 있다. 비디오 코더는 코딩 유닛의 선택스 엘리먼트를 디코딩할 수도 있고, 여기서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시한다. 비디오 코더는 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정할 수도 있다. 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여, 비디오 코더는 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용할 수도 있다.

[0007] 하나의 예에서는, 비디오 데이터를 디코딩하기 위한 방법이 설명되어 있다. 방법은 코딩 유닛의 선택스 엘리먼트를 디코딩하는 단계를 포함하고, 여기서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시한다. 방법은 또한, 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하는 단계, 및 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하는 단계를 포함한다.

[0008] 또 다른 예에서, 디바이스는 코딩 유닛의 선택스 엘리먼트를 디코딩하도록 구성된 비디오 디코더를 포함하고, 여기서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시한다. 게다가, 디바이스는 또한, 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하고, 그리고 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하도록 구성된다.

[0009] 또 다른 예에서, 디바이스는 코딩 유닛의 선택스 엘리먼트를 디코딩하기 위한 수단으로서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하는, 상기 코딩 유닛의 선택스 엘리먼트를 디코딩하기 위한 수단, 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하기 위한 수단, 및 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하기 위한 수단을 포함한다.

[0010] 또 다른 예에서, 컴퓨터-판독가능 저장 매체는, 실행될 경우, 프로세서로 하여금, 코딩 유닛의 선택스 엘리먼트를 디코딩하게 하는 명령들을 저장하고, 여기서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시한다. 명령들은 또한, 실행될 경우, 프로세서로 하여금, 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정하게 하고, 그리고 선택스

엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용하게 한다.

[0011] 또 다른 예에서는, 비디오 테이터를 인코딩하기 위한 방법이 설명되어 있다. 방법은 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하는 단계, 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 신택스 엘리먼트의 값을 설정하는 단계, 및 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용하는 단계를 포함한다.

[0012] 또 다른 예에서, 디바이스는, 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하고, 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 신택스 엘리먼트의 값을 설정하고, 그리고 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용하도록 구성된 비디오 인코더를 포함한다.

[0013] 또 다른 예에서, 디바이스는 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하기 위한 수단, 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 신택스 엘리먼트의 값을 설정하기 위한 수단, 및 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용하기 위한 수단을 포함한다.

[0014] 또 다른 예에서, 컴퓨터-판독가능 저장 매체는, 실행될 경우, 프로세서로 하여금, 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정하게 하고, 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 신택스 엘리먼트의 값을 설정하게 하고, 그리고 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용하게 하는 명령들을 저장한다.

[0015] 개시물의 하나 이상의 예들의 세부 사항들은 동반된 도면들 및 이하의 설명에서 기재된다. 개시물의 다른 특징들, 목적들, 및 장점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0016] 도 1 은 이 개시물에서 설명된 기법들을 사용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 예시하는 블록도이다.

도 3 은 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 예시하는 블록도이다.

도 4 는 현재의 개시물의 하나 이상의 기법들에 따라 35 개의 HEVC 예측 모드들을 예시하는 개념도이다.

도 5 는 현재의 개시물의 하나 이상의 기법들에 따라 병합 및 진보된 모션 벡터 예측 (advanced motion vector prediction; AMVP) 모드들에 대한 공간적으로 이웃하는 모션 벡터 후보들을 예시하는 개념도이다.

도 6 은 현재의 개시물의 하나 이상의 기법들에 따라 인트라 블록 복사 (block copy; BC) 예를 예시하는 개념도이다.

도 7 은 현재의 개시물의 하나 이상의 기법들에 따라, 인트라 8x8 블록에 대한 타겟 블록 및 참조 샘플의 예를 예시하는 개념도이다.

도 8 은 현재의 개시물의 하나 이상의 기법들에 따라 인코딩 기법을 예시하는 흐름도이다.

도 9 는 현재의 개시물의 하나 이상의 기법들에 따라 디코딩 기법을 예시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0017] 일부의 예들에서, 이 개시물은 스크린 콘텐츠 코딩에 관련되고, 여기서, 높은 크로마 샘플링 포맷 (chroma sampling format) 4:4:4 가 이용된다. 일부의 예들에서, 이 개시물은 또한, 아마도 높은 비트 심도 (bit depth), (8 비트 초과), 높은 크로마 샘플링 포맷 4:4:4 의 지원을 포함하는 범위 확장들 (RCEx) 을 위하여 적용가능하다. 일부의 예들에서, 이 개시물은 또한, 크로마 샘플링 포맷이 4:2:2 인 것과 같은 다른 컬러 포맷들에 적용가능하다. 더욱 구체적으로, 이 개시물에서는, 컬러-공간 변환에 관련된 다수의 상이한 기법들이 설명되어 있다.

[0018] 비디오 코딩 표준들은 그 스케일러블 비디오 코딩 (Scalable Video Coding; SVC) 및 멀티뷰 비디오 코딩

(Multiview Video Coding; MVC) 확장들을 포함하는, ITU-T H.261, ISO/IEC MPEG-1 비주얼 (Visual), ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 및 ITU-T H.264 (또한, ISO/IEC MPEG-4 AVC 로서 알려짐) 를 포함한다.

- [0019] 새로운 비디오 코딩 표준, 즉, 고효율 비디오 코딩 (High-Efficiency Video Coding; HEVC) 의 설계는 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 및 ISO/IEC 모션 픽처 전문가 그룹 (Motion Picture Experts Group; MPEG) 의 비디오 코딩에 관한 합동 협력 팀 (Joint Collaboration Team on Video Coding; JCT-VC) 에 의해 완결되었다.
- [0020] HEVC 에서, 슬라이스에서의 최대 코딩 유닛은 코딩 트리 블록 (coding tree block; CTB) 으로 칭해진다. CTB 는 그 노드들이 코딩 유닛들인 쿼드-트리 (quad-tree) 를 포함한다. CTB 의 사이즈는 HEVC 주요 프로파일에서 16x16 으로부터 64x64 까지의 범위일 수 있지만, 8x8 CTB 사이즈들이 지원될 수 있다. 코딩 유닛 (CU) 은 CTB 와 동일한 사이즈일 수 있고 8x8 만큼 작을 수 있다. 각각의 CU 는 하나의 모드로 코딩된다. CU 가 인터 코딩될 때, CU 는 2 개의 예측 유닛 (prediction unit; PU) 들로 추가로 파티셔닝될 수도 있거나, 추가의 파티셔닝들이 적용되지 않을 때에 단일 PU 가 될 수도 있다. 2 개의 PU 들이 하나의 CU 에서 존재할 때, 각각의 PU 는 절반의 사이즈의 직사각형들, 또는 CU 의 사이즈의 1/4 또는 3/4 와 동일한 사이즈를 갖는 2 개의 직사각형들일 수 있다.
- [0021] CU 가 인터 코딩될 때, 모션 정보의 하나의 세트는 각각의 PU 에 대해 존재한다. 게다가, 각각의 PU 는 모션 정보의 세트를 유도하기 위하여 고유의 인터-예측 모드로 코딩된다. HEVC 에서, 최소 PU 사이즈들은 8x4 및 4x8 이다.
- [0022] HEVC 는 예측 잔차를 코딩하기 위하여 4x4, 8x8, 16x16, 및 32x32 의 사이즈들을 갖는 4 개의 변환 유닛 (transform unit; TU) 들을 특징한다. CTB 는 4 개 이상의 TU 들로 재귀적으로 파티셔닝될 수도 있다. TU 들은 이산 코사인 변환 (discrete cosine transform; DCT) 함수들과 유사한 정수-기반 함수들을 이용한다. 게다가, 인트라 코딩된 영역에 속하는 4x4 루마 변환 블록들은 이산 사인 변환 (discrete sine transform; DST) 함수로부터 유도되는 정수 변환을 이용하여 변환된다. 크로마 (chroma) 는 루마 (luma) 와 동일한 TU 사이즈들을 이용한다.
- [0023] 현재의 HEVC 표준에서는, 각각의 예측 유닛 (PU) 의 루마 컴포넌트에 대하여, 인트라 예측 방법이 도 4 에 대하여 이하에서 설명된 바와 같이, 33 개의 각도 예측 모드들 (2 로부터 34 까지 인덱싱됨), DC 모드 (1 로 인덱싱됨), 및 평면 모드 (0 으로 인덱싱됨) 와 함께 사용된다.
- [0024] 상기 35 개의 인트라 모드들에 추가하여, 'I-PCM' 으로 명명된 모드가 하나 더 HEVC 에 의해 또한 채용된다. I-PCM 모드에서는, 예측, 변환, 양자화, 및 엔트로피 코딩이 바이패스 (bypass) 되는 반면, 예측 샘플들은 미리 정의된 수의 비트들에 의해 코딩된다. I-PCM 모드의 주요 목적은 신호가 다른 모드들에 의해 효율적으로 코딩될 수 없을 때의 상황을 처리하기 위한 것이다.
- [0025] 현재의 HEVC 표준에서는, 이용가능한 2 개의 인터 예측 모드들이 있다. 하나의 이러한 모드는 병합 모드이고 (스킵 (skip) 은 병합의 특수한 경우로서 고려됨), 제 2 의 이러한 모드는 예측 유닛 (PU) 에 대한 진보된 모션 벡터 예측 (AMVP) 모드이다.
- [0026] AMVP 및 병합 모드의 양자에서는, 모션 벡터 (motion vector; MV) 후보 리스트가 다수의 모션 벡터 예측자 (motion vector predictor) 들에 대하여 유지된다. 현재의 PU 의 모션 벡터 (들) 뿐만 아니라, 병합 모드에서의 참조 인덱스들도 MV 후보 리스트로부터 하나의 후보를 취함으로써 생성된다.
- [0027] MV 후보 리스트는 병합 모드에 대한 5 개에 이르는 후보들과, AMVP 모드에 대한 2 개의 후보들만을 포함할 수도 있다. 병합 후보는 모션 정보의 세트, 예컨대, 참조 픽처 리스트들 (리스트 0 및 리스트 1) 및 참조 인덱스들의 양자에 대응하는 모션 벡터들을 포함할 수도 있다. 병합 후보가 병합 인덱스에 의해 식별될 경우, 참조 픽처들은 현재의 블록들의 예측을 위하여 이용될 뿐만 아니라, 연관된 모션 벡터들이 결정된다. 그러나, AMVP 모드 하에서는, AMVP 후보가 모션 벡터만을 포함하므로, 리스트 0 또는 리스트 1 의 어느 하나로부터의 각각의 잠재적인 예측 방향에 대해, 참조 인덱스가 MV 후보 리스트에 대한 MVP 인덱스와 함께, 명시적으로 시그널링될 필요가 있다. AMVP 모드에서는, 예측된 모션 벡터들이 추가로 정제 (refine) 될 수 있다.
- [0028] 병합 후보는 모션 정보의 완전한 세트에 대응하는 반면, AMVP 후보는 특정 예측 방향 및 참조 인덱스에 대하여 단지 하나의 모션 벡터를 포함할 수도 있다. 양자의 모드들에 대한 후보들은 동일한 공간적 및 시간적으로

이웃하는 블록들로부터 유사하게 유도된다.

[0029] 도 5 에 대하여 이하에서 설명된 바와 같이, 공간적 MV 후보들은 특정 PU (PU0) 에 대하여, 도 5 상에서 도시된 이웃하는 블록들로부터 유도된다. 그러나, 블록들로부터 후보들을 생성하는 방법들은 병합 및 AMVP 모드들에 대하여 상이하다.

[0030] 병합 모드에서는, 4 개에 이르는 공간적 MV 후보들이 숫자들로 도 5 (a) 상에서 도시된 순서들로 유도될 수 있다. 순서는 다음과 같다: 도 5 (a) 에서 도시된 바와 같이, 좌측 (0), 상부 (1), 상부 우측 (2), 하부 좌측 (3), 및 상부 좌측 (4).

[0031] AMVP 모드에서는, 이웃하는 블록들이 2 개의 그룹들로 분할된다: 도 5 (b) 상에서 도시된 바와 같이, 블록 0 및 1 을 포함하는 좌측 그룹, 및 블록들 2, 3, 및 4 를 포함하는 상부 그룹. 각각의 그룹에 대하여, 시그널링된 참조 인덱스에 의해 표시된 것과 동일한 참조 픽처를 참조하는 이웃하는 블록에서의 잠재적인 후보는 그룹의 최종 후보를 형성하기 위하여 선택되어야 할 최고 우선순위를 가진다. 모든 이웃하는 블록들은 동일한 참조 픽처를 가리키는 모션 벡터를 포함하지 않는 것이 가능하다. 그러므로, 이러한 후보를 구할 수 없는 경우, 제 1 이용가능한 후보는 최종 후보를 형성하기 위하여 스케일링될 것이고, 이에 따라, 시간적 거리 차이들이 보상될 수 있다.

[0032] 도 6 에 대하여 이하에서 설명된 바와 같이, 인트라 블록-복사 (BC) 가 SCC 내에 포함되었다. 인트라 BC 의 예는 도 6 에서 도시되고, 여기서, 현재의 CU 는 현재의 픽처/슬라이스의 이미 디코딩된 블록으로부터 예측된다. 현재의 인트라 BC 블록 사이즈는 8x8 로부터 64x64 까지의 범위인 CU 사이즈만큼 클 수 있지만, 일부의 애플리케이션들에서는, 추가의 제약들이 적용될 수도 있다.

[0033] 일부의 예들에서는, 잔차 신호들에 대한 인-루프 (in-loop) 컬러-공간 변환 프로세스가 4:4:4 크로마 포맷으로 된 시퀀스 (sequence) 들에 대하여 이용될 수 있다. 이 방법은 RGB/YUV 크로마 포맷으로 된 예측 에러 신호들을 차선의 컬러-공간에서의 예측 에러 신호들로 변환한다. 이 추가적인 단계로, 컬러 컴포넌트들 사이의 상관은 추가로 감소될 수 있다. 변환 행렬은 특이값 분해 (singular-value-decomposition; SVD) 에 의해 각각의 코딩 유닛에 대한 픽셀 샘플 값들로부터 유도된다. 컬러-공간 변환은 인트라 및 인터 모드 양자의 예측 에러에 적용된다.

[0034] 일부의 예들에서, 컬러-공간 변환 프로세스가 인터 모드에 적용될 때, 잔차는 먼저, 유도된 변환 행렬로 상이한 도메인으로 변환된다. 컬러-공간 변환 후에, DCT/DST, 양자화, 및 엔트로피 코딩과 같은 기존의 코딩 단계들이 순서대로 수행된다.

[0035] 일부의 예들에서, 컬러-공간 변환 프로세스가 인트라 모드에 적용될 때, 예측 및 현재의 블록은 먼저, 유도된 변환 행렬로 각각 상이한 도메인으로 변환된다. 컬러-공간 변환 후에, 현재의 블록과 그 예측자 사이의 잔차는 DCT/DST 로 추가로 변환되고, 양자화되고, 그리고 엔트로피 코딩된다.

[0036] 순방향 연산 (forward operation) 에서, 컬러-공간 변환 행렬은 다음과 같이 3 개의 평면들 G, B, 및 R 에 적용된다:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} P \\ Q \\ S \end{bmatrix}$$

[0037]

[0038] 결과적인 값들은 HEVC 규격 (specification) 의 범위 내에서 클리핑 (clipping) 되는데, 이것은 일부의 예들에서, 값들이 $\sqrt{3}$ 배까지 확대되기 때문이다. 역 연산 (inverse operation) 에서는, 컬러-공간 변환 행렬이 다음과 같이 3 개의 컴포넌트들 P', Q', 및 R' 에 적용된다.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^t \begin{bmatrix} P' \\ Q' \\ S' \end{bmatrix} = \begin{bmatrix} G' \\ B' \\ R' \end{bmatrix}$$

[0039]

[0040] 도 7 에 대하여 이하에서 설명된 바와 같이, 변환 행렬은 참조 샘플 값들로부터 유도될 수도 있다. 상이한 참조 샘플들은 인트라의 경우 및 인터의 경우에 대해 사용될 수 있다. 인트라 코딩된 블록의 경우에 대하여, 타겟 블록 및 참조 샘플들은 도 7 에서 도시되어 있다. 이 예에서, 타겟 블록은 8x8 크로스헤칭된

샘플들로 구성되고, 참조 샘플들은 스트라이프 표시되고 점 표시된 샘플들이다.

[0041] 인터 코딩된 블록의 경우에 대하여, 행렬 유도를 위한 참조 샘플들은 모션 보상을 위한 참조 샘플들과 동일하다. 시프트 연산 (shift operation) 을 실현하기 위하여, AMP 블록에서의 참조 샘플들은 샘플들의 수가 2 의 거듭제곱 (power-of-two) 이 되도록 서브-샘플링된다. 예를 들어, 12x16 블록에서의 참조 샘플들의 수는 2/3 만큼 감소된다.

[0042] 현재의 개시물의 기법들에 따르면, 컬러-공간 변환 프로세스는 각각의 CU 에 대하여 적용될 수도 있다. 그러므로, 변환 프로세스가 호출되는지 여부를 시그널링할 필요성이 없다. 게다가, 인코더 및 디코더 측들의 양자는 변환 행렬을 시그널링하기 위한 오버헤드 (overhead) 를 회피하기 위하여 동일한 방법으로 변환 행렬을 유도한다.

[0043] 현재의 개시물의 기법들에 따르면, 컬러-공간 변환 행렬들과 같은 컬러-공간 변환 프로세스들이 이용된다. 하나의 이러한 행렬은 YCbCr 변환 행렬이고, 이것은:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

[0044] 순방향:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

[0045] 역:

[0046] 또 다른 이러한 행렬은 YCoCg 변환 행렬이고, 이것은:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

[0047] 순방향:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

[0048] 역:

[0049] 또 다른 이러한 행렬은 YCoCg-R 행렬이고, 이것은 2 배로 (by a factor of two) Co 및 Cg 컴포넌트들을 스케일링하는 YCoCg 행렬의 수정된 버전이다. 리프팅 (lifting) 기법을 이용함으로써, 순방향 및 역 변환은 다음의 수학식들에 의해 달성될 수 있다:

$$\begin{aligned} Co &= R - B \\ t &= B + \lfloor Co/2 \rfloor \\ Cg &= G - t \\ Y &= t + \lfloor Cg/2 \rfloor \end{aligned}$$

[0050] 순방향:

$$\begin{aligned} t &= Y - \lfloor Cg/2 \rfloor \\ G &= Cg + t \\ B &= t - \lfloor Co/2 \rfloor \\ R &= B + Co \end{aligned}$$

[0051] 역:

[0052] 상기 수학식들 및 행렬들에서, 순방향 변환들은 (예컨대, 비디오 인코더에 의해) 인코딩 프로세스 동안에 수행될 수도 있고, 역 변환들은 (예컨대, 비디오 디코더에 의해) 디코딩 프로세스에서 수행될 수도 있다.

[0053] 전통적인 비디오 코딩에서는, 이미지들이 연속-톤 (continuous-tone) 을 가지고 공간적으로 평활한 것으로 가정된다. 이 가정들에 기초하여, 자연적 콘텐츠를 갖는 비디오들에 대하여 양호한 성능을 보여주었던 다양한 툴들 (예컨대, 블록-기반 변환, 필터링 등) 이 개발되었다. 그러나, 어떤 애플리케이션들 (예컨대, 원격 데스크톱, 협력 작업 디스플레이들, 및 무선 디스플레이들) 에서는, 컴퓨터 생성된 스크린 콘텐츠가 압축되어야 할 지배적인 콘텐츠일 수도 있다. 이 타입의 콘텐츠는 이산-톤 (discrete-tone) 을 가지며 예리한 라인들

및 높은-콘트라스트의 객체 경계들을 특징으로 하는 경향이 있다. 연속-톤 및 평활화의 가정은 더 이상 적용되지 않을 수도 있고, 이에 따라, 전통적인 비디오 코딩 기법들이 효율적으로 작동하지 않을 수도 있다.

[0054] 팔레트 모드 코딩 (palette mode coding) 은 상기 결점들을 해소하기 위하여 이용될 수도 있다. 팔레트 코딩 기법들의 예들은 2013 년 4 월 10 일자로 출원된 미국 가출원 제 61/810,649 호에서 설명되어 있다. 각각의 CU 에 대하여, 현재의 CU 에서 가장 지배적인 픽셀 값들을 포함하는 팔레트가 유도될 수도 있다. 팔레트의 사이즈 및 엘리먼트들이 먼저 송신된다. 송신 후에, CU 에서의 픽셀들은 어떤 스캔 순서에 따라 인코딩된다. 각각의 로케이션 (location) 에 대하여, 플래그, palette_flag 와 같은 신택스 엘리먼트는 픽셀 값이 팔레트에 있는지 ("실행 모드") 또는 아닌지 ("픽셀 모드") 를 표시하기 위하여 먼저 송신된다.

[0055] "실행 모드" 에서는, 팔레트 인덱스가 시그널링될 수도 있고, "실행" 이 뒤따를 수도 있다. 실행은 현재 코딩되고 있는 픽셀과 동일한 팔레트 인덱스 값을 가지는 스캔 순서에서의 연속 픽셀들의 수를 표시하는 신택스 엘리먼트이다. 스캔 순서에서 바로 연속하는 다수의 픽셀들이 동일한 팔레트 인덱스 값을 가질 경우, "실행 모드" 는 palette_flag 와 같은 신택스 엘리먼트에 의해 표시될 수도 있다. 현재의 픽셀과 동일한 팔레트 인덱스 값을 가지는 현재의 픽셀을 뒤따르는 픽셀들의 수와 동일한 카운터 값이 결정될 수도 있고, 실행은 카운터 값과 동일하게 설정된다. palette_flag 뿐만 아니라 팔레트 인덱스도 "실행" 에 의해 커버되는 다음의 위치들에 대해 송신될 필요가 없는데, 이것은 이들이 모두 동일한 픽셀 값을 가지기 때문이다. 디코더 측 상에서는, 현재의 픽셀에 대한 최초 팔레트 인덱스 값만이 디코딩될 것이고, 결과는 "실행" 신택스 엘리먼트에서 표시된 픽셀들의 "실행" 에서 각각의 픽셀에 대하여 복제될 것이다.

[0056] "픽셀 모드" 에서는, 픽셀 샘플 값이 이 위치에 대하여 송신된다. palette_flag 와 같은 신택스 엘리먼트가 "픽셀 모드" 를 표시할 경우, 팔레트 인덱스 값은 디코딩되고 있는 하나의 픽셀에 대해서만 결정된다.

[0057] 기존의 기법들은 다양한 문제들을 겪을 수도 있다. 예를 들어, 시퀀스 특성들 및 로컬 다이버시티 (local diversity) 를 참작하지 않는 컬러-공간 변환이 호출될 수도 있다. 그러므로, 코딩 성능은 차선일 수도 있다. 또 다른 예에서는, 디코더 복잡성을 상당히 증가시키는 변환 행렬의 유도가 디코더에서 요구될 수도 있다. 또한, 변환 행렬은 공간적 재구성된 픽셀들 또는 인터-코딩된 PU 들의 예측자 중의 어느 하나를 이용하여 유도될 수도 있다. 그러나, 변환 행렬의 효율은 PU 사이즈가 상대적으로 작거나, 예측이 매우 정확하지 않거나, 이웃하는 픽셀들이 이용불가능할 때에 감소될 수도 있다. 이 개시물의 기법들은 이 문제들 중의 하나 이상을 해소할 수도 있다.

[0058] 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은 비디오 코더는 이 개시물에서 설명된 기법들을 실행할 수도 있다. 비디오 코더는 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정할 수도 있다. 비디오 코더는 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 신택스 엘리먼트의 값을 설정할 수도 있다. 비디오 코더는 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 행렬을 적용할 수도 있다. 비디오 코더는 코딩 유닛의 신택스 엘리먼트를 디코딩할 수도 있고, 여기서, 신택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시한다. 비디오 코더는 신택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정할 수도 있다. 비디오 코더는 신택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 행렬을 적용할 수도 있다. 디코더 측 상에서는, 코딩 모드에 관계 없이, 컬러-공간 순방향 변환 프로세스를 수행할 필요가 없다. 일부의 예들에서, 컬러-공간 역변환 프로세스는 고정될 수도 있다.

[0059] 이 개시물은 인-루프 컬러-공간 변환의 코딩 성능을 개선시킬 수도 있으며 디코더 복잡성을 감소시킬 수도 있는 기법들을 설명한다. 도 1 은 스크린 콘텐츠 코딩을 위한 기법들을 사용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이고, 여기서, 높은 크로마 샘플링 포맷이 이용된다. 도 1 에서 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 더 이후의 시간에 디코딩되어야 할 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 컴퓨터-판독가능 매체 (16) 를 통해 비디오 데이터를 목적지 디바이스 (14) 에 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 (set-top) 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들 중의 임의의 것을 포함할 수도 있다. 일부의 경우들에는, 소스 디바이스 (12) 및 목적지 디바이스 (14) 가 무선 통신을 위해 갖추어질 수도 있다.

- [0060] 목적지 디바이스 (14) 는 컴퓨터-관독가능 매체 (16) 를 통해, 디코딩되어야 할 인코딩된 비디오 데이터를 수신할 수도 있다. 컴퓨터-관독가능 매체 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 컴퓨터-관독가능 매체 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 직접 송신하는 것을 가능하게 하기 위한 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조될 수도 있고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 라디오 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 가능하게 하기 위해 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.
- [0061] 일부의 예들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, 블루-레이 (Blu-ray) 디스크들, DVD 들, CD-ROM 들, 플래시 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적당한 디지털 저장 매체들과 같은, 분산되거나 국소적으로 액세스된 다양한 데이터 저장 매체들 중의 임의의 것을 포함할 수도 있다. 추가의 예에서, 저장 디바이스는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 저장 디바이스로부터 저장된 비디오 데이터를 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있으며 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 일 예의 파일 서버들은 (예컨대, 웹사이트를 위한) 웹 서버, FTP 서버, 네트워크 연결 저장 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터를 액세스할 수도 있다. 이것은, 파일 서버 상에 저장된 인코딩된 비디오 데이터를 액세스하기 위해 적당한 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 그 조합일 수도 있다.
- [0062] 이 개시물의 기법들은 무선 애플리케이션들 또는 세팅들로 반드시 제한되는 것은 아니다. 기법들은 오버-더-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, HTTP 를 통한 동적 적응 스트리밍 (dynamic adaptive streaming over HTTP; DASH) 과 같은 인터넷 스트리밍 비디오 송신들, 데이터 저장 매체 상으로 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중의 임의의 것의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부의 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 및/또는 화상 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위하여 일방향 (one-way) 또는 양방향 (two-way) 비디오 송신을 지원하도록 구성될 수도 있다.
- [0063] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 이 개시물에 따르면, 소스 디바이스 (12) 의 비디오 인코더 (20) 는 컬러-공간 변환 프로세스를 이용하여 비디오 블록들을 코딩하기 위한 기법들을 적용하도록 구성될 수도 있다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 외부 카메라와 같은 외부 비디오 소스 (18) 로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하는 것이 아니라, 외부 디스플레이 디바이스와 인터페이스 접속할 수도 있다.
- [0064] 도 1 의 예시된 시스템 (10) 은 하나의 예에 불과하다. 컬러-공간 변환 프로세스를 이용하여 비디오 블록들을 코딩하기 위한 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로, 이 개시물의 기법들은 비디오 인코딩 디바이스에 의해 수행되지만, 기법들은 또한, "CODEC" 으로서 전형적으로 지칭된 비디오 인코더/디코더에 의해 수행될 수도 있다. 또한, 이 개시물의 기법들은 또한, 비디오 프리프로세서 (video preprocessor) 에 의해 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 소스 디바이스 (12) 가 목적지 디바이스 (14) 로의 송신을 위한 코딩된 비디오 데이터를

생성하는 이러한 코딩 디바이스들의 예들에 불과하다. 일부의 예들에서, 디바이스들 (12, 14) 은, 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 이 때문에, 시스템 (10) 은 예컨대, 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 또는 화상 통화를 위하여, 비디오 디바이스들 (12, 14) 사이의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0065] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 공급 인터페이스를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (18) 는 소스 비디오로서, 또는 라이브 비디오 (live video), 아카이빙된 비디오 (archived video), 및 컴퓨터-생성된 비디오의 조합으로서, 컴퓨터 그래픽-기반 (computer graphics-based) 데이터를 생성할 수도 있다. 일부의 경우들에는, 비디오 소스 (18) 가 비디오 카메라일 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14) 가 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 위에서 언급된 바와 같이, 이 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에 있어서, 캡처된, 프리-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 다음으로, 인코딩된 비디오 정보는 출력 인터페이스 (22) 에 의해 컴퓨터-판독가능 매체 (16) 상으로 출력될 수도 있다.

[0066] 컴퓨터-판독가능 매체 (16) 는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 순시적 매체 (transient medium) 들, 또는 하드 디스크, 플래시 드라이브, 콤팩트 디스크, 디지털 비디오 디스크, 블루-레이 디스크, 또는 다른 컴퓨터-판독가능 매체들과 같은 저장 매체들 (즉, 비-일시적 저장 매체들) 을 포함할 수도 있다. 일부의 예들에서, 네트워크 서버 (도시되지 않음) 는 예컨대, 네트워크 송신을 통해, 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신할 수도 있으며 인코딩된 비디오 데이터를 목적지 디바이스 (14) 에 제공할 수도 있다. 유사하게, 디스크 스탬핑 (disc stamping) 설비와 같은 매체 생산 설비의 컴퓨팅 디바이스는 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신할 수도 있고, 인코딩된 비디오 데이터를 포함하는 디스크를 생산할 수도 있다. 그러므로, 컴퓨터-판독가능 매체 (16) 는 다양한 예들에서, 다양한 형태들의 하나 이상의 컴퓨터-판독가능 매체들을 포함하는 것으로 이해될 수도 있다.

[0067] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터-판독가능 매체 (16) 로부터 정보를 수신한다. 컴퓨터-판독가능 매체 (16) 의 정보는, 블록들 및 다른 코딩된 유닛들, 예컨대, GOP 들의 특성들 및/또는 프로세싱을 설명하는 선택스 엘리먼트들을 포함하는 선택스 정보로서, 비디오 인코더 (20) 에 의해 정의되며 또한, 비디오 디코더 (30) 에 의해 이용되는 상기 선택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관 (cathode ray tube; CRT), 액정 디스플레이 (liquid crystal display; LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode; OLED) 디스플레이, 또는 또 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중의 임의의 것을 포함할 수도 있다.

[0068] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 개발 중에 있는 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 코딩 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HEVC Test Model; HM) 을 준수할 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MPEG-4, Part 10, 진보된 비디오 코딩 (AVC), 또는 이러한 표준들의 확장들로서 대안적으로 지칭된 ITU-T H.264 표준과 같은 다른 독점적 또는 산업 표준들에 따라 동작할 수도 있다. 그러나, 이 개시물의 기법들은 임의의 특정한 코딩 표준으로 제한되지는 않는다. 비디오 코딩 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 을 포함한다. 도 1 에서 도시되지 않았지만, 일부의 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통의 데이터 스트림 또는 별도의 데이터 스트림들에서 오디오 및 비디오 양자의 인코딩을 처리하기 위한 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol; UDP) 과 같은 다른 프로토콜들을 준수할 수도 있다.

[0069] ITU-T H.264/MPEG-4 (AVC) 표준은 합동 비디오 팀 (Joint Video Team; JVT) 으로서 알려진 집합적 파트너십의 산물로서의 ISO/IEC 동화상 전문가 그룹 (MPEG) 과 함께 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 에 의해 공식화되었다. 일부의 양태들에서, 이 개시물에서 설명된 기법들은 일반적으로 H.264 표준을 준수하는 디바이스들에 적용될 수도 있다. H.264 표준은, H.264 표준 또는 H.264 규격, 또는 H.264/AVC 표준 또는 규격으로서 본원에서 지칭될 수도 있는, 2005 년 3월자의 ITU-T 연구 그룹에 의한 ITU-T 추천안 H.264, Advanced Video

Coding for generic audiovisual services (일반적인 시청각 서비스들을 위한 진보된 비디오 코딩) 에서 설명되어 있다. 합동 비디오 팀 (JVT) 은 H.264/MPEG-4 AVC 에 대한 확장들에 대해 계속 작업하고 있다.

[0070] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (digital signal processor; DSP) 들, 주문형 집적 회로 (application specific integrated circuit; ASIC) 들, 필드 프로그래밍가능 게이트 어레이 (field programmable gate array; FPGA) 들, 개별 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그 임의의 조합들과 같은 다양한 적당한 인코더 회로부 중의 임의의 것으로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현될 때, 디바이스는 소프트웨어를 위한 명령들을 적당한 비-일시적 컴퓨터-판독가능 매체 내에 저장할 수도 있고, 이 개시물의 기법들을 수행하기 위하여 하나 이상의 프로세서들을 이용하여 명령들을 하드웨어로 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 인코더들 또는 디코더들 중의 어느 하나는 조합된 인코더/디코더 (combined encoder/decoder; CODEC) 의 일부로서 개개의 디바이스 내에 통합될 수도 있다.

[0071] JCT-VC 는 HEVC 표준의 개발에 노력을 들이고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로서 지칭된 비디오 코더의 진화형 모델에 기초한다. HM 은 예컨대, ITU-T H.264/AVC 에 따라 기존의 디바이스들에 비해 비디오 코더들의 몇몇 추가적인 능력들을 추정한다. 예를 들어, H.264 가 9 개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM 은 33 개만큼 많은 인트라-예측 인코딩 모드들을 제공할 수도 있다.

[0072] 일반적으로, HM의 작업 모델은, 비디오 프레임 또는 픽처가 루마 및 크로마 샘플들의 양자를 포함하는 최대 코딩 유닛 (LCU) 들 또는 트리블록들의 시퀀스로 분할될 수도 있다는 것을 설명한다. 비트스트림 내의 선택스 데이터는 픽셀들의 수의 측면에서 최대 코딩 유닛인 LCU 에 대한 사이즈를 정의할 수도 있다. 슬라이스는 코딩 순서에서의 다수의 연속 트리블록들을 포함한다. 비디오 프레임 또는 픽처는 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 쿼드트리 (quadtree) 에 따라 코딩 유닛 (CU) 들로 분할될 수도 있다. 일반적으로, 쿼드트리 데이터 구조는 CU 당 하나의 노드를 포함하고, 루트 노드 (root node) 는 트리블록에 대응한다. CU 가 4 개의 서브-CU 들로 분할될 경우, CU 에 대응하는 노드는 4 개의 리프 노드 (leaf node) 들을 포함하고, 4 개의 리프 노드들의 각각은 서브-CU 들 중의 하나에 대응한다.

[0073] 쿼드트리 데이터 구조의 각각의 노드는 대응하는 CU 에 대한 선택스 데이터를 제공할 수도 있다. 예를 들어, 쿼드트리에서의 노드는, 노드에 대응하는 CU 가 서브-CU 들로 분할되는지 여부를 표시하는 분할 플래그 (split flag) 를 포함할 수도 있다. CU 에 대한 선택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU 가 서브-CU 들로 분할되는지 여부에 종속될 수도 있다. CU 가 추가로 분할되지 않을 경우, 그것은 리프-CU 로서 지칭된다. 이 개시물에서는, 원래의 리프-CU 의 명시적 분할이 없더라도, 리프-CU 의 4 개의 서브-CU 들은 또한 리프-CU 들로서 지칭될 것이다. 예를 들어, 16x16 사이즈에서의 CU 가 추가로 분할되지 않을 경우, 16x16 CU 가 결코 분할되지 않았지만, 4 개의 8x8 서브-CU 들이 또한 리프-CU 들로서 지칭될 것이다.

[0074] CU 가 사이즈 구분 (size distinction) 을 가지지 않는다는 것을 제외하고는, CU 는 H.264 표준의 매크로블록 (macroblock) 과 유사한 목적을 가진다. 예를 들어, 트리블록은 4 개의 자식 노드 (child node) 들 (또한, 서브-CU 들로서 지칭됨) 로 분할될 수도 있고, 각각의 자식 노드는 궁극적으로 부모 노드 (parent node) 일 수도 있고, 또 다른 4 개의 자식 노드들로 분할될 수도 있다. 쿼드트리의 리프 노드로서 지칭된, 최종적인 분할되지 않은 자식 노드는 리프-CU 로서 또한 지칭된 코딩 노드를 포함한다. 코딩된 비트스트림과 연관된 선택스 데이터는 최대 CU 심도로서 지칭된, 트리블록이 분할될 수도 있는 최대 횟수를 정의할 수도 있고, 또한, 코딩 노드들의 최소 사이즈를 정의할 수도 있다. 따라서, 비트스트림은 또한, 최소 코딩 유닛 (smallest coding unit; SCU) 을 정의할 수도 있다. 이 개시물은 HEVC 의 문맥에서의 CU, PU, 또는 TU 중의 임의의 것, 또는 다른 표준들의 문맥에서의 유사한 데이터 구조들 (예컨대, H.264/AVC 에서의 매크로블록들 및 그 서브-블록들) 을 지칭하기 위하여 용어 "블록" 을 이용한다.

[0075] CU 는 코딩 노드와, 코딩 노드와 연관된 예측 유닛 (PU) 들 및 변환 유닛 (TU) 들을 포함한다. CU 의 사이즈는 코딩 노드의 사이즈에 대응하고, 형상에 있어서 정사각형이어야 한다. CU 의 사이즈는 8x8 픽셀들로부터, 64x64 픽셀들 이상의 최대치를 갖는 트리블록의 사이즈까지의 범위일 수도 있다. 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다. CU 와 연관된 선택스 데이터는 예를 들어, 하나 이상의 PU 들로의 CU 의 파티셔닝을 설명할 수도 있다. 파티셔닝 모드들은 CU 가 스킵 (skip) 또는 직접 모드 인코딩되는지, 인트라-예측 모드 인코딩되는지, 또는 인터-예측 모드 인코딩되는지 여부의 사이에서 상이할 수도 있다. PU 들은 형상에 있어서 비-정사각형 (non-square) 이 되도록 파티셔닝될 수도 있다. CU 와 연

관된 선택스 데이터는 또한, 예를 들어, 쿼드트리에 따른 하나 이상의 TU 들로의 CU 의 파티셔닝을 설명할 수도 있다. TU 는 형상에 있어서 정사각형 또는 비-정사각형 (예컨대, 직사각형) 일 수 있다.

[0076] HEVC 표준은 상이한 CU 들에 대해 상이할 수도 있는, TU 들에 따른 변환들을 허용한다. TU 들은 전형적으로, 파티셔닝된 LCU 에 대해 정의된 소정의 CU 내에서의 PU 들의 사이즈에 기초하여 사이즈가 정해지지만, 이것은 항상 그러하지는 않을 수도 있다. TU 들은 전형적으로 동일한 사이즈이거나 PU 들보다 더 작다. 일부의 예들에서, CU 에 대응하는 잔차 샘플들은 "잔차 쿼드 트리" (residual quad tree; RQT) 로서 알려진 쿼드트리 구조를 이용하여 더 작은 유닛들로 재분할될 수도 있다. RQT 의 리프 노드들은 변환 유닛 (TU) 들로서 지칭될 수도 있다. TU 들과 연관된 픽셀 차이 값들은 양자화될 수도 있는 변환 계수들을 생성하기 위하여 변환될 수도 있다.

[0077] 리프-CU 는 하나 이상의 예측 유닛 (PU) 들을 포함할 수도 있다. 일반적으로, PU 는 대응하는 CU 의 전부 또는 부분에 대응하는 공간적인 영역을 나타내고, PU 에 대한 참조 샘플을 추출 (retrieve) 하기 위한 데이터를 포함할 수도 있다. 또한, PU 는 예측에 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라-모드 인코딩될 때, PU 에 대한 데이터는 PU 에 대응하는 TU 에 대한 인트라-예측 모드를 설명하는 데이터를 포함할 수도 있는 잔차 쿼드트리 (RQT) 내에 포함될 수도 있다. 또 다른 예로서, PU 가 인터-모드 인코딩될 때, PU 는 PU 에 대한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. PU 에 대한 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터에 대한 해상도 (예컨대, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 참조 픽처, 및/또는 모션 벡터에 대한 참조 픽처 리스트 (예컨대, List 0, List 1, 또는 List C) 를 설명할 수도 있다.

[0078] 하나 이상의 PU 들을 가지는 리프-CU 는 또한, 하나 이상의 변환 유닛 (TU) 들을 포함할 수도 있다. 위에서 논의된 바와 같이, 변환 유닛들은 RQT (또한, TU 쿼드트리 구조로서 지칭됨) 를 이용하여 특정될 수도 있다. 예를 들어, 분할 플래그는 리프-CU 가 4 개의 변환 유닛들로 분할되는지 여부를 표시할 수도 있다. 다음으로, 각각의 변환 유닛은 추가의 서브-TU 들로 추가로 분할될 수도 있다. TU 가 추가로 분할되지 않을 때, 그것은 리프-TU 로서 지칭될 수도 있다. 일반적으로, 인트라 코딩을 위하여, 리프-CU 에 속하는 모든 리프-TU 들은 동일한 인트라 예측 모드를 공유한다. 즉, 동일한 인트라-예측 모드는 리프-CU 의 모든 TU 들에 대한 예측된 값들을 계산하기 위하여 일반적으로 적용된다. 인트라 코딩을 위하여, 비디오 인코더는 TU 에 대응하는 CU 의 부분과 원래의 블록과의 사이의 차이로서, 인트라 예측 모드를 이용하여 각각의 리프-TU 에 대한 잔차 값을 계산할 수도 있다. TU 는 반드시 PU 의 사이즈로 제한되는 것은 아니다. 이에 따라, TU 들은 PU 보다 더 크거나 더 작을 수도 있다. 인트라 코딩을 위하여, PU 는 동일한 CU 에 대한 대응하는 리프-TU 와 코로케이트 (collocate) 될 수도 있다. 일부의 예들에서, 리프-TU 의 최대 사이즈는 대응하는 리프-CU 의 사이즈에 대응할 수도 있다.

[0079] 또한, 리프-CU 들의 TU 들은 또한, 잔차 쿼드트리 (RQT) 들로서 지칭된 개개의 쿼드트리 데이터 구조들과 연관될 수도 있다. 즉, 리프-CU 는 리프-CU 가 어떻게 TU 들로 파티셔닝되는지를 표시하는 쿼드트리를 포함할 수도 있다. TU 쿼드트리의 루트 노드는 일반적으로 리프-CU 에 대응하는 반면, CU 쿼드트리의 루트 노드는 일반적으로 트리블록 (또는 LCU) 에 대응한다. 분할되지 않은 RQT 의 TU 들은 리프-TU 들로서 지칭된다. 일반적으로, 이와 다르게 언급되지 않으면, 이 개시물은 리프-CU 및 리프-TU 를 각각 지칭하기 위하여 용어들 CU 및 TU 를 이용한다.

[0080] 비디오 시퀀스는 전형적으로 일련의 비디오 프레임들 또는 픽처들을 포함한다. 픽처들의 그룹 (group of pictures; GOP) 은 일반적으로 비디오 픽처들 중의 일련의 하나 이상을 포함한다. GOP 는, GOP 내에 포함된 다수의 픽처들을 설명하는, GOP 의 헤더, 픽처들 중의 하나 이상의 픽처의 헤더, 또는 다른 곳에서의 선택스 데이터를 포함할 수도 있다. 픽처의 각각의 슬라이스는 개개의 슬라이스에 대한 인코딩 모드를 설명하는 슬라이스 선택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 전형적으로 비디오 데이터를 인코딩하기 위하여 개별적인 비디오 슬라이스들 내의 비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정된 또는 변동되는 사이즈들을 가질 수도 있고, 특정된 코딩 표준에 따라 사이즈에 있어서 상이할 수도 있다.

[0081] 예로서, HM 은 다양한 PU 사이즈들에 있어서의 예측을 지원한다. 특정한 CU 의 사이즈가 $2N \times 2N$ 인 것으로 가정하면, HM 은 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들에 있어서의 인트라-예측과, $2N \times 2N$, $2N \times N$, $N \times 2N$, 또는 $N \times N$ 의 대칭적인 PU 사이즈들에 있어서의 인터-예측을 지원한다. HM 은 또한, $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에 있어서의 인터-예측을 위한 비대칭적 파티셔닝을 지원한다. 비대칭적 파티셔닝에서는, CU 의

하나의 방향이 파티셔닝되지 않는 반면, 다른 방향은 25 % 및 75 % 로 파티셔닝된다. 25 % 파티션에 대응하는 CU 의 부분은 "n" 과, 그 다음으로, "상부 (Up)", "하부 (Down)", "좌측 (Left)", 또는 "우측 (Right)" 의 표시에 의해 표시된다. 이에 따라, 예를 들어, "2NxN" 는 상부의 2Nx0.5N PU 및 하부의 2Nx1.5N PU 로 수평으로 파티셔닝되는 2Nx2N CU 를 지칭한다.

[0082] 이 개시물에서, "NxN" 및 "N 대 (by) N" 은 수직 및 수평 차원들의 측면에서의 비디오 블록의 픽셀 차원들, 예를 들어, 16x16 픽셀들 또는 16 대 16 픽셀들을 지칭하기 위하여 상호 교환가능하게 이용될 수도 있다. 일반적으로, 16x16 블록은 수직 방향에서의 16 개의 픽셀들 ($y = 16$) 및 수평 방향에서의 16 개의 픽셀들 ($x = 16$) 을 가질 것이다. 마찬가지로, NxN 블록은 일반적으로, 수직 방향에서의 N 개의 픽셀들 및 수평 방향에서의 N 개의 픽셀들을 가지며, 여기서, N 은 음이 아닌 정수 (nonnegative integer) 값을 나타낸다. 블록 내의 픽셀들은 행 (row) 들 및 열 (column) 들로 배열될 수도 있다. 또한, 블록들은 수직 방향에서와 동일한 수의 픽셀들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, 블록들은 NxM 픽셀들을 포함할 수도 있으며, 여기서, M 은 반드시 N 과 동일하지는 않다.

[0083] CU 의 PU 들을 이용한 인트라-예측 또는 인터-예측 코딩에 후속하여, 비디오 인코더 (20) 는 CU 의 TU 들에 대한 잔차 데이터를 계산할 수도 있다. PU 들은 공간 도메인 (또한 픽셀 도메인으로서 지칭됨) 에서 예측 픽셀 데이터를 생성하는 방법 또는 모드를 설명하는 신택스 데이터를 포함할 수도 있고, TU 들은 변환, 예컨대, 이산 코사인 변환 (DCT), 정수 변환, 웨이블렛 변환 (wavelet transform), 또는 개념적으로 유사한 변환의 잔차 비디오 데이터로의 적용에 후속하는 변환 도메인에서 계수들을 포함할 수도 있다. 잔차 데이터는 인코딩되지 않은 픽처의 픽셀들과 PU 들에 대응하는 예측 값들과의 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔차 데이터를 포함하는 TU 들을 형성할 수도 있고, 그 다음으로, CU 에 대한 변환 계수들을 생성하기 위하여 TU 들을 변환할 수도 있다.

[0084] 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로, 계수들을 나타내기 위하여 이용된 데이터의 양을 가능한 대로 감소시키기 위하여 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도 (bit depth) 를 감소시킬 수도 있다. 예를 들어, n -비트 값은 양자화 동안에 m -비트 값으로 버림 (round down) 될 수도 있고, 여기서, n 은 m 보다 더 크다.

[0085] 양자화에 후속하여, 비디오 인코더는 변환 계수들을 스캔 (scan) 하여, 양자화된 변환 계수들을 포함하는 2 차원 행렬로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 더 높은 에너지 (그리고 이에 따라, 더 낮은 주파수) 계수들을 어레이의 전방에 배치하고 더 낮은 에너지 (그리고 이에 따라, 더 높은 주파수) 계수들을 어레이의 후방에 배치하도록 설계될 수도 있다. 일부의 예들에서, 비디오 인코더 (20) 는 양자화된 변환 계수들을 스캔하여, 엔트로피 인코딩될 수 있는 직렬화된 벡터 (serialized vector) 를 생성하기 위하여 미리 정의된 스캔 순서를 사용할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캔 (adaptive scan) 을 수행할 수도 있다. 1 차원 벡터를 형성하기 위하여 양자화된 변환 계수들을 스캔한 후, 비디오 인코더 (20) 는 예컨대, 컨텍스트-적응 가변 길이 코딩 (context-adaptive variable length coding; CAVLC), 컨텍스트-적응 2 진 산술 코딩 (context-adaptive binary arithmetic coding; CABAC), 신택스-기반 컨텍스트-적응 2 진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 간격 파티셔닝 엔트로피 (Probability Interval Partitioning Entropy; PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라 1 차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30) 에 의한 이용을 위한 인코딩된 비디오 데이터와 연관된 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0086] CABAC 를 수행하기 위하여, 비디오 인코더 (20) 는 컨텍스트 모델 내의 컨텍스트를 송신되어야 할 심볼에 배정할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 비-제로 (non-zero) 인지 아닌지의 여부와 관련 있을 수도 있다. CAVLC 를 수행하기 위하여, 비디오 인코더 (20) 는 송신되어야 할 심볼에 대한 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드 (codeword) 들은, 상대적으로 더 짧은 코드들이 더 확률이 큰 심볼들에 대응하는 반면, 더 긴 코드들은 더 확률이 작은 심볼들에 대응하도록 구성될 수도 있다. 이러한 방법으로, VLC 의 이용은 예를 들어, 송신되어야 할 각각의 심볼에 대한 동일-길이 코드워드들을 이용하는 것에 비해 비트 절감을 달성할 수도 있다. 확률 결정은 심볼에 배정된 컨텍스트에 기초할 수도 있다.

[0087] 이 개시물의 기법들에 따르면, 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은 비디오 코더는 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정할 수도 있다. 비디오 코더는 컬러-공간 변환의 이용

을 표시하도록 코딩 유닛의 선택스 엘리먼트의 값을 설정할 수도 있다. 비디오 코더는 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 적용할 수도 있다. 비디오 코더는 코딩 유닛의 선택스 엘리먼트를 디코딩할 수도 있고, 여기서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시한다. 비디오 코더는 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 결정할 수도 있다. 비디오 코더는 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 적용할 수도 있다. 디코더 측 상에서는, 코딩 모드에 관계 없이, 컬러-공간 순방향 변환 프로세스를 수행할 필요가 없다. 일부의 예들에서, 컬러-공간 역변환 프로세스는 고정될 수도 있다.

[0088] 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은 비디오 코더는 도 1 내지 도 9 에 관련하여 설명된 바와 같은 기법들 중의 임의의 것을 수행할 수도 있다. CU-레벨 적응적 컬러-공간 변환 방식, 또는 컬러-공간 변환이 제안되고, 여기서, 각각의 CU 는 컬러-공간 변환을 인에이블하거나 디스에이블하는 것의 어느 하나를 선택할 수 있다. 게다가, CU-레벨 플래그는 컬러-공간 변환의 이용을 표시하기 위하여 도입된다. 플래그는 시그널링될 수도 있고, CU-레벨 플래그의 시그널링 방법은 코딩 모드들 및 다른 정보에 종속적일 수 있다.

[0089] 다양한 선택스 엘리먼트들은 현재의 개시물의 기법들과 정렬하여 이용될 수도 있다. 이 선택스 엘리먼트들은 다음을 포함할 수도 있다:

표 1

seq_parameter_set_rbsp() {	디스크립터
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(sps_max_sub_layers_minus1)	
...	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_present_flag	u(1)
if(sps_extension_present_flag) {	
for(i = 0; i < 1; i++)	
sps_extension_flag[i]	u(1)
sps_extension_7bits	u(7)
if(sps_extension_flag[0]) {	
transform_skip_rotation_enabled_flag	u(1)
transform_skip_context_enabled_flag	u(1)
intra_block_copy_enabled_flag	u(1)
implicit_rdpem_enabled_flag	u(1)
explicit_rdpem_enabled_flag	u(1)
extended_precision_processing_flag	u(1)
intra_smoothing_disabled_flag	u(1)
high_precision_offsets_enabled_flag	u(1)
fast_rice_adaptation_enabled_flag	u(1)
cabac_bypass_alignment_enabled_flag	u(1)
color_transform_enabled_flag (new)	u(1)
}	
if(sps_extension_7bits)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
}	
} rbsp_trailing_bits()	
}	

[0090]

coding_unit(x0, y0, log2CbSize) {	디스크립터
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != 1)	
cu_skip_flag [x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(intra_block_copy_enabled_flag)	
intra_bc_flag [x0][y0]	ae(v)
if(slice_type != 1 && !intra_bc_flag[x0][y0])	
pred_mode_flag	ae(v)
if(CuPredMode[x0][y0] != MODE_INTRA intra_bc_flag[x0][y0] log2CbSize == MinCbLog2SizeY)	
part_mode	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag && !intra_bc_flag[x0][y0] && log2CbSize >= Log2MinIpcmCbSizeY && log2CbSize <= Log2MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else if(intra_bc_flag[x0][y0]) {	
mvd_coding(x0, y0, 2)	
if(PartMode == PART_2NxN)	
mvd_coding(x0, y0 + (nCbS / 2), 2)	
else if(PartMode == PART_Nx2N)	
mvd_coding(x0 + (nCbS / 2), y0, 2)	
else if(PartMode == PART_NxN) {	
mvd_coding(x0 + (nCbS / 2), y0, 2)	
mvd_coding(x0, y0 + (nCbS / 2), 2)	
mvd_coding(x0 + (nCbS / 2), y0 + (nCbS / 2), 2)	
}	
} else {	
pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : nCbS	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
prev_intra_luma_pred_flag [x0 + i][y0 + j]	ae(v)
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx [x0 + i][y0 + j]	ae(v)
else	

[0091]

rem_intra_luma_pred_mode[x0 + i][y0 + j]	ae(v)
if(ChromaArrayType == 3)	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
intra_chroma_pred_mode[x0 + i][y0 + j]	ae(v)
else if(ChromaArrayType != 0)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else {	
...	
}	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]) (CuPredMode[x0][y0] == MODE_INTRA && intra_bc_flag[x0][y0]))	
rqt_root_cbf	ae(v)
if(rqt_root_cbf) {	
if(color_transform_enabled_flag) (new)	
color_transform_flag[x0][y0] (new)	ae(v)
MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

[0092]

[0093]

신택스 엘리먼트 color_transform_enabled_flag 가 1 과 동일할 때, 인-루프 컬러-공간 변환 프로세스는 디코딩 프로세스에서 호출될 수도 있다. 신택스 엘리먼트 color_transform_enabled_flag 가 0 과 동일할 때, 인-루프 컬러-공간 변환은 적용되지 않을 수도 있다. 신택스 엘리먼트가 존재하지 않을 때, 신택스 엘리먼트 color_transform_enabled_flag 의 값은 0 과 동일한 것으로 추론될 수도 있다.

[0094]

신택스 엘리먼트 color_transform_flag[x0][y0] 가 1 과 동일할 때, 현재의 코딩 유닛은 컬러-공간 변환으로 코딩될 수도 있다. 신택스 엘리먼트 color_transform_flag[x0][y0] 가 0 과 동일할 때, 현재의 코딩 유닛은 컬러-공간 변환 없이 코딩된다. 신택스 엘리먼트가 존재하지 않을 때, 신택스 엘리먼트 color_transform_flag 의 값은 0 과 동일한 것으로 추론될 수도 있다. 어레이 인덱스들 x0 및 y0 는 픽처의 상부-좌측 루마 샘플에 관련하여 고려된 코딩 블록의 상부-좌측 루마 샘플의 로케이션 (x0, y0) 을 특정할 수도 있다.

[0095]

하나의 예에서, 수정된 YCoCg 변환은 인코딩 및 디코딩 프로세스에서 적용될 수도 있다. 수정된 YCoCg 변환은 다음과 같이 정의될 수도 있다:

[0096]

순방향:
$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

[0097]

역: temp = Y-Cg;

[0098]

$$G = (Y + Cg + offset) \gg 2;$$

[0099]

$$B = (temp - Co + offset) \gg 2;$$

[0100]

$$R = (temp + Co + offset) \gg 2; \text{ 여기서, offset (오프셋) 은 2 와 동일하다.}$$

[0101]

대안적으로, 수정된 YCoCg 변환은 다음과 같이 정의될 수도 있다:

[0102]

순방향:
$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/2 & 1 & 1/2 \\ 1 & 0 & -1 \\ -1/2 & 1 & -1/2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- [0103] 역: $\text{temp} = Y - C_g;$
- [0104] $G = (Y + C_g + \text{offset}) \gg 1;$
- [0105] $B = (\text{temp} - C_o + \text{offset}) \gg 1;$
- [0106] $R = (\text{temp} + C_o + \text{offset}) \gg 1;$ 여기서, offset (오프셋)은 1 또는 0과 동일하다.
- [0107] 컬러 변환으로 코딩되는 CU 들 또는 블록들에 의해 이용된 대응하는 양자화 파라미터는 ($dQP + 2$)와 동일한 것으로 추론될 수도 있는 반면, 컬러 변환 없이 CU 들 또는 블록들에 의해 이용된 것은 dQP 와 동일한 것으로 추론될 수도 있다. 동시에, 비트-심도는 양자화 및 변환 프로세스들의 양자에서 1만큼 추가로 증가될 수도 있다.
- [0108] 대안적으로, 2개의 단계들 순방향/역 변환은 변경되지 않은 원래의 YCoCg 변환과 정규화 프로세스로 적용된다. 순방향 변환에 대하여, 원래의 YCoCg 순방향 변환이 먼저 적용된다. 다음으로, 각각의 컴포넌트 i , 즉, Y , C_o , 및 C_g 에 대하여, 컴포넌트는 $(i * \text{forward_factor} + \text{offset}) \gg \text{BIT_PRECISION}$ 로 재설정되고, 여기서, BIT_PRECISION은 무부호 정수 (unsigned integer)이고, forward_factor는 BIT_PRECISION에 종속적이다. 하나의 예에서, forward_factor는 $1/\sqrt{6} * (1 \ll \text{BITS_TRANS}) + 0.5$ 와 동일하고, BIT_PRECISION은 15와 동일하고, 오프셋은 $(1 \ll (\text{BIT_PRECISION} - 1))$ 과 동일하다.
- [0109] 역 변환에 대하여, 원래의 YCoCg 역 변환이 먼저 적용된다. 다음으로, 각각의 컴포넌트 i , 즉, Y , C_o , 및 C_g 에 대하여, 컴포넌트는 $(i * \text{backward_factor} + \text{offset}) \gg \text{BIT_PRECISION}$ 로 재설정되고, 여기서, BIT_PRECISION은 순방향 변환에서 이용된 것과 동일하고, backward_factor는 BIT_PRECISION에 종속적이다. 하나의 예에서, backward_factor는 $\sqrt{6}/4 * (1 \ll \text{BITS_TRANS}) + 0.5$ 와 동일하고, 오프셋은 $(1 \ll (\text{BIT_PRECISION} - 1))$ 과 동일하다.
- [0110] 대안적으로/추가로, CU-레벨 플래그는 3개의 컬러 컴포넌트들 (즉, cbf_luma, cbf_cb, 및 cbf_cr)에서의 코딩된 블록 플래그들 중의 적어도 하나가 1과 동일할 때에만 시그널링될 수도 있다. 대안적으로, 컬러-공간 변환 후의 수정된 잔차는, 컬러-공간 변환을 갖는 CU 들 또는 블록들에서의 잔차의 범위가 컬러-공간 변환을 갖지 않는 CU 들/블록들에서의 잔차의 그것과 확실히 동일하게 하기 위하여 추가로 수정될 수도 있다. 하나의 예에서는, 클립 동작이 적용된다.
- [0111] 컬러-공간 변환 행렬은 재구성된 픽셀들로부터 독립적일 수도 있다. 그 대신에, 그것은 손실 또는 무손실 코딩 모드에 종속적일 수도 있다. 하나의 예에서, CU가 손실 모드로 코딩될 때, YCoCg가 적용될 수도 있는 반면, CU가 무손실 모드로 코딩될 때에는, YCoCg-R이 적용된다. 또한, YCoCg-R이 적용될 때, C_o 및 C_g 컴포넌트들의 비트 심도는 1만큼 증가될 수도 있다. 또 다른 예에서, 변환 행렬은 인트라, 인터/인트라 BC 모드들에 종속적일 수 있다. 이 경우, 각각의 모드에 대한 미리 정의된 행렬은 인코더 및 디코더 양자에서 특정될 수도 있고, 동일한 행렬은 CU-레벨 플래그가 1일 때에 코딩 모드에 따라 인코더 및 디코더 양자에서 이용될 수도 있다. 또한, CU-레벨 플래그는 3개의 컬러 컴포넌트들 (즉, cbf_luma, cbf_cb, 및 cbf_cr)에서의 코딩된 블록 플래그들 중의 적어도 하나가 1과 동일할 때에만 시그널링될 수도 있다.
- [0112] 현재의 개시물의 기법들은 또한, 변환 행렬 유도 및 시그널링 방법을 이룰 때면 프레임-레벨에서 포함할 수도 있다. 하나의 예에서, 각각의 프레임에 대하여, 변환 행렬은 원래의 이미지에서의 컬러 컴포넌트들에 걸친 통계에 기초하여 유도될 수도 있다. 변환 행렬은 비트스트림으로 코딩될 수 있고, 디코더로 직접적으로 송신될 수 있다. 대안적으로, 연속으로 코딩된 프레임들 사이의 변환 행렬 예측이 적용될 수도 있다. 예를 들어, 하나의 플래그는 행렬이 이전의 프레임에서 이용된 것과 동일한지 여부를 표시하기 위하여 송신될 수도 있다. 대안적으로, 변환 행렬은 원래의 프레임들의 세트에 기초하여 유도될 수 있고, 비트스트림에서 시그널링될 수 있다.
- [0113] 또 다른 예에서는, 변환 행렬들의 세트가 PPS와 같은 하이-레벨 선택스에서 유도되고 시그널링된다. 하나의 예에서, 이미지는 특성들에 기초하여 다수의 영역들로 분류되고, 각각의 영역에 대하여, 변환 행렬이 유도된다. 유도된 행렬들은 하이-레벨 선택스에서 코딩될 수 있고 시그널링될 수 있다. 선택된 변환 행렬의 인덱스는 CU/TU/PU 레벨에서 추가로 시그널링된다. 대안적으로, 선택된 변환 행렬의 인덱스는 슬라이스 헤더에서, 또는 각각의 타일 영역에 대하여 시그널링된다. 대안적으로, 변환 행렬은 슬라이스 헤더에서, 또는 각각의 타일 영역에 대하여 유도될 수도 있다.
- [0114] 루마 및 크로마 컴포넌트들의 비트-심도가 상이할 때, 모든 컬러 컴포넌트들의 비트-심도는 동일한 것으로 먼저

수정될 수도 있고, 그 다음으로, 컬러 변환이 나중에 적용될 수도 있다. 대안적으로, 비트스트림은, 루마 및 크로마 컴포넌트들의 비트-심도가 코딩된 비트스트림에서 상이할 때, 컬러 변환이 디스에이블 (즉, 0 과 동일한 `color_transform_enabled_flag`) 될 것이라는 제약을 준수할 수도 있다.

[0115] 상이한 QP 들은 컬러 변환으로, 또는 컬러 변환 없이 코딩되는 CU 들/블록들에 적용될 수도 있거나, 및/또는 양자화 및 변환 프로세스에서 이용된 비트-심도는 이러한 비-정규화된 컬러 변환으로 인해 수정될 수 있다. 하나의 예에서, 상기 수정된 YCoCg 변환이 이용될 때, 다음의 단계들 (즉, 디블록킹 필터 프로세스 이전의 스케일링, 변환, 및 어레이 구성 프로세스) 은 `color_transform_flag` 에 기초하여 추가로 수정된다.

[0116] 하나의 예에서, 수정된 YCoCg 변환이 문단 식별 번호 <0095> 내지 <0100> 에서 설명된 바와 같이 이용될 때, 대응하는 양자화 파라미터 유도 프로세스 및 역 변환 프로세스가 이에 따라 수정된다. 이 프로세스에 대한 입력들은 현재의 픽처의 상부-좌측 루마 샘플에 관련하여 현재의 루마 코딩 블록의 상부-좌측 샘플을 특정하는 루마 로케이션 (`xCb`, `yCb`) 을 포함할 수도 있다. 이 프로세스에서는, 변수 `QpY`, 루마 양자화 파라미터 `Qp'Y`, 및 크로마 양자화 파라미터들 `Qp'Cb` 및 `Qp'Cr` 이 유도될 수도 있다. 루마 로케이션 (`xQg`, `yQg`) 은 현재의 픽처의 상부 좌측 루마 샘플에 관련하여 현재의 양자화 그룹의 상부-좌측 루마 샘플을 특정할 수도 있다. 수평 및 수직 위치들 `xQg` 및 `yQg` 는 각각 `xCb-(xCb&((1<<Log2MinCuQpDeltaSize)-1))` 및 `yCb-(yCb&((1<<Log2MinCuQpDeltaSize)-1))` 과 동일하게 설정될 수도 있다. 양자화 그룹의 루마 사이즈, `Log2MinCuQpDeltaSize` 는 동일한 `qPY_PRED` 를 공유하는 코딩 트리 블록 내부의 최소 영역의 루마 사이즈를 결정할 수도 있다.

[0117] 변수 `QpY` 는 다음과 같이 유도될 수도 있다:

[0118]
$$QpY = ((qPY_PRED + CuQpDeltaVal + 52 + 2 * QpBdOffsetY) \% (52 + QpBdOffsetY)) - QpBdOffsetY$$

[0119] 루마 양자화 파라미터 `Qp'Y` 는 다음과 같이 유도될 수도 있다:

[0120]
$$Qp'Y = QpY + QpBdOffsetY$$

[0121] 선택스 엘리먼트 `ChromaArrayType` 가 0 과 동일하지 않을 때, 다음이 적용될 수도 있다:

[0122] 변수들 `qPiCb` 및 `qPiCr` 은 다음과 같이 유도될 수도 있다:

[0123]
$$qPiCb = Clip3(-QpBdOffsetC, 57, QpY + pps_cb_qp_offset + slice_cb_qp_offset + CuQpAdjValCb)$$

[0124]
$$qPiCr = Clip3(-QpBdOffsetC, 57, QpY + pps_cr_qp_offset + slice_cr_qp_offset + CuQpAdjValCr)$$

[0125] 선택스 엘리먼트 `ChromaArrayType` 가 1 과 동일할 때, 변수들 `qPCb` 및 `qPCr` 은 각각 `qPiCb` 및 `qPiCr` 과 동일한 인덱스 `qPi` 에 기초하여 이하의 표에서 특정된 바와 같이 `QpC` 의 값과 동일하게 설정될 수도 있다. 그렇지 않으면, 변수들 `qPCb` 및 `qPCr` 은 각각 `qPiCb` 및 `qPiCr` 과 동일한 인덱스 `qPi` 에 기초하여 $Min(qPi, 51)$ 과 동일하게 설정된다.

표 2

qPi	< 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
Qp _c	= qP _i	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6

[0126]

[0127] `Cb` 및 `Cr` 컴포넌트들에 대한 크로마 양자화 파라미터들, `Qp'Cb` 및 `Qp'Cr` 은 다음과 같이 유도될 수도 있다:

[0128]
$$Qp'Cb = qPCb + QpBdOffsetC$$

[0129]
$$Qp'Cr = qPCr + QpBdOffsetC$$

[0130] `color_transform_flag` 가 1 과 동일할 때, 다음이 적용될 수도 있다:

[0131]
$$Qp'Y = (Qp'Y + 8)$$

[0132] `ChromaArrayType` ≠ 0 일 때, $Qp'Cb = (Qp'cb + 8)$ 이고 $Qp'Cr = (Qp'cr + 8)$ 이다

- [0133] $BitDepthY = (BitDepthY + 2)$
- [0134] $BitDepthC = (BitDepthC + 2)$
- [0135] 이 프로세스에 대한 입력들은 현재의 픽처의 상부 좌측 루마 샘플에 관련하여 현재의 루마 변환 블록의 상부-좌측 샘플을 특징하는 루마 로케이션 ($xTbY$, $yTbY$), 현재의 변환 블록의 사이즈를 특징하는 변수 $nTbS$, 현재의 블록의 컬러 컴포넌트를 특징하는 변수 $cIdx$, 및 양자화 파라미터를 특징하는 변수 qP 를 포함할 수도 있다.
이 프로세스의 출력은 엘리먼트들 $d[x][y]$ 을 갖는 스케일링된 변환 계수들의 $(nTbS) \times (nTbS)$ 어레이 d 를 포함할 수도 있다.
- [0136] 변수들 $\log_2 TransformRange$, $bdShift$, $coeffMin$, 및 $coeffMax$ 는 다음과 같이 유도될 수도 있다:
- [0137] $color_transform_flag = 1$ 일 때, 다음이 적용된다:
- [0138] $CoeffCTMinY = -(1 \ll (extended_precision_processing_flag ? Max(15, BitDepthY+6):15))$
- [0139] $CoeffCTMinC = -(1 \ll (extended_precision_processing_flag ? Max(15, BitDepthC+6):15))$
- [0140] $CoeffCTMaxY = (1 \ll (extended_precision_processing_flag ? Max(15, BitDepthY+6):15)) - 1$
- [0141] $CoeffCTMaxC = (1 \ll (extended_precision_processing_flag ? Max(15, BitDepthC+6):15)) - 1$
- [0142] $cIdx = 0$ 일 경우,
- [0143] $\log_2 TransformRange = extended_precision_processing_flag ? Max(15, BitDepthY+6):15$
- [0144] $bdShift = BitDepthY + \log_2(nTbS) + 10 - \log_2 TransformRange$
- [0145] $coeffMin = (color_transform_flag ? CoeffCTMinY : CoeffMinY)$
- [0146] $coeffMax = (color_transform_flag ? CoeffCTMaxY : CoeffMaxY)$
- [0147] 그렇지 않으면,
- [0148] $\log_2 TransformRange = extended_precision_processing_flag ? Max(15, BitDepthC+6):15$
- [0149] $bdShift = BitDepthC + \log_2(nTbS) + 10 - \log_2 TransformRange$
- [0150] $coeffMin = (color_transform_flag ? CoeffCTMinC : CoeffMinC)$
- [0151] $coeffMax = (color_transform_flag ? CoeffCTMaxC : CoeffMaxC)$
- [0152] 리스트 $levelScale[]$ 는 $k = 0..5$ 인 $levelScale[k] = \{ 40, 45, 51, 57, 64, 72 \}$ 로서 특정될 수도 있다.
- [0153] $x = 0..nTbS - 1$, $y = 0..nTbS - 1$ 인 스케일링된 변환 계수들 $d[x][y]$ 의 유도를 위하여, 다음이 적용될 수도 있다:
- [0154] 스케일링 인수 $m[x][y]$ 는 다음과 같이 유도될 수도 있다:
- [0155] 다음의 조건들 중의 하나 이상이 참 (true) 일 경우, $m[x][y]$ 은 16 과 동일하게 설정된다:
- [0156] $scaling_list_enabled_flag = 0$.
- [0157] $transform_skip_flag[xTbY][yTbY] = 1$ 이고 $nTbS > 4$ 이다.
- [0158] 그렇지 않으면, 다음이 적용된다:
- [0159] $m[x][y] = ScalingFactor[sizeId][matrixId][x][y]$ (8-283)
- [0160] 선택스 엘리먼트 $sizeId$ 가 $(nTbS) \times (nTbS)$ 와 동일한 양자화 행렬의 사이즈에 대해 특정되고, $matrixId$ 가 각각

sizeId, CuPredMode[xTbY][yTbY], 및 cIdx에 대하여 표 7-4 에서 특정될 때, 다음으로, 스케일링된 변환 계수 $d[x][y]$ 는 다음과 같이 유도될 수도 있다:

[0161]
$$d[x][y] = \text{Clip3}(\text{coeffMin}, \text{coeffMax}, ((\text{TransCoeffLevel}[xTbY][yTbY][cIdx][x][y] * \text{levelScale}[qP\%6] << (qP/6)) + (1 << (\text{bdShift} - 1))) >> \text{bdShift})$$

[0162] color_transform_flag 가 1 과 동일할 때, 다음이 적용될 수도 있다:

[0163] $\text{BitDepthY} = \text{BitDepthY} - 2$

[0164] $\text{BitDepthC} = \text{BitDepthC} - 2$

[0165] 하나의 예에서, 컬러-공간 변환이 인트라 모드들에 적용될 때, 잔차는 변환된 컬러-공간 도메인에서 유도된다. 즉, 인트라 샘플 예측 프로세스를 호출하기 전에, 현재의 CU 의 이웃하는 픽셀들은 수정된 순방향 YCoCg 또는 YCoCg-R 변환으로 또 다른 차선의 컬러-공간으로 먼저 변환될 수도 있다. 다음으로, 수정된 이웃하는 픽셀들은 현재의 CU 의 예측자를 유도하기 위하여 이용될 수도 있다. 잔차 (즉, 예측 에러) 는 차선의 도메인에서 현재의 CU 및 현재의 CU 의 이웃하는 픽셀들로부터 유도된다. 잔차는 인트라-컴포넌트 잔차 예측, 변환, 양자화, 및 엔트로피 코딩과 같이, 기존의 코딩 흐름과 동일한 방법으로 코딩된다. 엔트로피 코딩한 후, 변환 계수들은 역 YCoCg 또는 YCoCg-R 로 추가로 수정될 수도 있다. 다음으로, 수정된 변환 계수들은 현재의 CU 의 재구성된 픽셀들을 유도하기 위하여 예측자에 추가될 수도 있다. 재구성 프로세스가 호출된 후, 역 YCoCg 또는 YCoCg-R 변환은 수정된 이웃하는 픽셀들에 적용될 수도 있다. 그러므로, 디코더 측에서는, 재구성 프로세스를 호출하기 전에, 하나의 추가적인 프로세스가 적용될 수도 있고, 여기서, 역 컬러-공간 변환이 유도된 변환 계수들에 적용된다.

[0166] 대안적으로, 컬러-공간 변환이 인트라 모드들에 적용될 때, 변환된 컬러 도메인에서 현재의 블록으로부터 예측자를 감산함으로써 잔차를 생성하는 대신에, 잔차는 컬러-공간 변환 전, 컬러-공간 변환 후의 픽셀들을 이용하여 생성될 수도 있다. 이 경우, 루마 및 크로마 인트라 모드들은 동일한 모드로 설정될 수 있다. 대안적으로, 크로마 모드의 시그널링은 스킵될 수도 있다. 대안적으로, 컬러 변환 플래그는 루마 및 크로마 인트라 모드들이 동일할 때에만 시그널링될 수도 있다.

[0167] 위에서 설명된 바와 같이, 일부의 예들에서, 무손실 및 손실 코딩 방법들은 동일한 행렬, 예를 들어, YCoCg-R 을 공유할 수도 있다. 추가적으로 또는 대안적으로, YCoCg-R 변환이 손실 코딩에 적용될 때, 컬러 변환을 갖는 블록들의 크로마 컴포넌트들의 비트-심도는 컬러 변환을 갖지 않는 블록들에 비해 1 만큼 증가될 수도 있다. 컬러 변환을 갖는 블록들의 루마 컴포넌트 QP 는 컬러 변환을 갖지 않는 블록들의 루마 컴포넌트 QP 마이너스 (minus) 4 와 동일할 수도 있다. 하나의 예에서, 컬러 변환을 갖는 블록들의 2 개의 크로마 컴포넌트들 QP 는 컬러 변환을 갖지 않는 블록들의 크로마/루마 컴포넌트 QP 플러스 (plus) 2 와 동일할 수도 있다. 하나의 예에서, 컬러 변환을 갖는 블록들의 Cg 크로마 컴포넌트들은 컬러 변환을 갖지 않는 블록들의 루마/크로마 컴포넌트 QP 플러스 2 와 동일할 수도 있다. 컬러 변환을 갖는 블록들의 Co 크로마 컴포넌트들은 컬러 변환을 갖지 않는 블록들의 루마/크로마 컴포넌트 QP 플러스 3 과 동일할 수도 있다.

[0168] 이러한 예들에서, YCoCg-R 이 손실 및 무손실 모드의 양자에 대하여 이용될 때, 현재의 블록이 손실 있게 코딩될 경우, 루마 및 크로마의 비트-심도는 적응적 컬러 변환을 인에이블하지 않으면서 코딩된 그러한 블록들에 대하여 동일할 수도 있다. 이 경우, 손실 코딩이 적용될 때, YCoCg-R 의 순방향 변환 후에는, 순방향 YCoCg-R 변환으로 인해 증가된 비트-심도를 감소시키고 비트-심도 정밀도를 Y 컴포넌트의 비트-심도 정밀도와 동일하게 하기 위하여, Co 및 Cg 컴포넌트들은 N-비트 우측 시프트에 의해 스케일링될 수도 있다. 게다가, 역 YCoCg-R 변환을 수행하기 전에, Co 및 Cg 컴포넌트는 N-비트 좌측 시프트로 수정될 수도 있다. 하나의 예에서, N 은 1 과 동일하다.

[0169] 하나의 예에서, 루마 및 크로마 컴포넌트들의 입력 비트-심도가 동일한 것으로 가정될 수도 있다. 손실 코딩 모드에 대하여, 시프트 연산들은 순방향 변환 후에, 그리고 역 변환 전에 2 개의 크로마 컴포넌트들 (즉, Cg, Co) 에 적용될 수도 있다. 하나의 예에서는, 다음의 프로세스가 순서대로 적용된다:

$$\begin{aligned} Co &= R - B \\ t &= B + \lfloor Co/2 \rfloor \\ Cg &= G - t \\ Y &= t + \lfloor Cg/2 \rfloor \end{aligned}$$

[0170] 1. 순방향 YCoCg-R 은 변경되지 않는다:

[0171] 2. 현재의 블록이 손실 모드에서 코딩될 경우, 다음이 추가로 적용될 수도 있다:

[0172] $C_g = C_g \gg 1$;

[0173] $C_o = C_o \gg 1$;

[0174] 또 다른 예에서, 오프셋은 우측 시프트 프로세스에서 고려될 수도 있다. 예를 들어, 상기 하이라이트된 수학적식은 다음에 의해 대체될 수 있다:

[0175] $C_g = (C_g + 1) \gg 1$;

[0176] $C_o = (C_o + 1) \gg 1$;

[0177] 역 YCoCg-R 변환을 호출하기 전에, 다음이 적용될 수도 있다:

[0178] 현재의 블록이 손실 모드에서 코딩될 경우, 다음이 적용될 수도 있다:

[0179] $C_g = C_g \ll 1$;

[0180] $C_o = C_o \ll 1$;

$$t = Y - \lfloor C_g/2 \rfloor$$

$$G = C_g + t$$

$$B = t - \lfloor C_o/2 \rfloor$$

$$R = B + C_o$$

[0181] 그리고 역 YCoCg-R 은 변경되지 않고 유지될 수도 있다: $R = B + C_o$.

[0182] 이러한 예들에서, 적응적 컬러 변환을 이용하는 변환 블록들에 대한 잔차 수정 프로세스는 ChromaArrayType 이 3 과 동일할 때에 호출될 수도 있다. 이 프로세스에 대한 입력들은 블록 사이즈를 특징하는 변수 blkSize, 엘리먼트들 $r_Y[x][y]$ 을 갖는 루마 잔차 샘플들 r_Y 의 (blkSize)x(blkSize) 어레이, 엘리먼트들 $r_{Cb}[x][y]$ 을 갖는 크로마 잔차 샘플들 r_{Cb} 의 (blkSize)x(blkSize) 어레이, 및 엘리먼트들 $r_{Cr}[x][y]$ 을 갖는 크로마 잔차 샘플들 r_{Cr} 의 (blkSize)x(blkSize) 어레이를 포함할 수도 있다. 이 프로세스에 대한 출력들은 루마 잔차 샘플들의 수정된 (blkSize)x(blkSize) 어레이 r_Y , 크로마 잔차 샘플들의 수정된 (blkSize)x(blkSize) 어레이 r_{Cb} , 및 크로마 잔차 샘플들의 수정된 (blkSize)x(blkSize) 어레이 r_{Cr} 를 포함할 수도 있다.

[0183] $x = 0..blkSize - 1$, $y = 0..blkSize - 1$ 인 잔차 샘플들 r_Y , r_{Cb} , 및 r_{Cr} 의 (blkSize)x(blkSize) 어레이들은 다음과 같이 수정될 수도 있다:

[0184] $cu_transquant_bypass_flag = 0$ 일 경우, $r_{Cb}[x][y] = r_{Cb}[x][y] \ll 1$ 이고 $r_{Cr}[x][y] = r_{Cr}[x][y] \ll 1$ 이다

[0185] $tmp = r_Y[x][y] - (r_{Cb}[x][y] \gg 1)$

[0186] $r_Y[x][y] = tmp + r_{Cb}[x][y]$

[0187] $r_{Cb}[x][y] = tmp - (r_{Cr}[x][y] \gg 1)$

[0188] $r_{Cr}[x][y] = r_{Cb}[x][y] + r_{Cr}[x][y]$

[0189] 대안적으로, 'cu_transquant_bypass_flag 가 0 과 동일할 경우, $r_{Cb}[x][y] = r_{Cb}[x][y] \ll 1$ 이고 $r_{Cr}[x][y] = r_{Cr}[x][y] \ll 1$ 이다' 는 ' $r_{Cb}[x][y] = r_{Cb}[x][y] \ll (1 - cu_transquant_bypass_flag)$ 이고 $r_{Cr}[x][y] = r_{Cr}[x][y] \ll (1 - cu_transquant_bypass_flag)$ 이다' 에 의해 대체될 수 있다.

[0190] 컬러-변환 행렬을 공유하는 손실 코딩 방법 및 무손실 코딩 방법의 또 다른 예에서, 루마 및 크로마 컴포넌트들의 입력 비트-심도는 상이할 수도 있다. 이러한 예에서, 루마 및 크로마 컴포넌트들의 입력 비트 심도들이 상이하고 YCoCg-R 이 손실 및 무손실 코딩 모드들의 양자에 대하여 이용될 때, 역 변환 전에, Y, Co, Cg 컴포넌트의 값들 중의 적어도 하나는 모든 3 개의 컴포넌트들이 동일한 비트-심도 정밀도를 가지도록 하기 위하여 먼저, N 비트들로 시프트될 수도 있다. 게다가, 또 하나의 비트가 2 개의 크로마 컴포넌트들 (즉, Co, 및 Cg)

로 시프트될 수도 있다. 모든 3 개의 컴포넌트들의 예측 값들 중의 적어도 하나는 순방향 YCoCg-R 변환을 호출하기 전에 동일한 비트-심도 정밀도를 가지도록 수정될 수도 있다. 그리고 순방향 변환 후에, 2 개의 크로마 컴포넌트들은 1 비트로 우측 시프트될 수도 있다. 무손실 코딩 모드들에 대해서는, 상이한 비트-심도가 고려되지 않을 수도 있고, 그러므로, 적응적 컬러-공간 변환이 디스에이블될 수도 있다. 하나의 예에서, 비트스트림은, 루마 및 크로마 컴포넌트들의 비트-심도가 코딩된 비트스트림에서 상이할 때, 컬러 변환이 디스에이블 (즉, 0 과 동일한 color_transform_enabled_flag) 될 것이라는 제약을 포함할 수도 있다.

[0191] 이 예에서, 적응적 컬러 변환을 이용하는 변환 블록들에 대한 잔차 수정 프로세스는 ChromaArrayType 이 3 과 동일할 때에 호출될 수도 있다. 이 프로세스에 대한 입력들은 블록 사이즈를 특정하는 변수 blkSize, 엘리먼트들 $rY[x][y]$ 을 갖는 루마 잔차 샘플들 rY 의 $(blkSize) \times (blkSize)$ 어레이, 엘리먼트들 $rCb[x][y]$ 을 갖는 크로마 잔차 샘플들 rCb 의 $(blkSize) \times (blkSize)$ 어레이, 및 엘리먼트들 $rCr[x][y]$ 을 갖는 크로마 잔차 샘플들 rCr 의 $(blkSize) \times (blkSize)$ 어레이를 포함할 수도 있다. 이 프로세스에 대한 출력들은 루마 잔차 샘플들의 수정된 $(blkSize) \times (blkSize)$ 어레이 r_Y , 크로마 잔차 샘플들의 수정된 $(blkSize) \times (blkSize)$ 어레이 r_{Cb} , 및 크로마 잔차 샘플들의 수정된 $(blkSize) \times (blkSize)$ 어레이 r_{Cr} 를 포함할 수도 있다.

[0192] 변수들 ΔBD_Y 및 ΔBD_C 은 다음과 같이 유도될 수도 있다:

[0193] $BitDepthMax = \max(BitDepthY, BitDepthC)$

[0194] $\Delta BD_Y = cu_transquant_bypass_flag ? 0 : BitDepthMax - BitDepthY$

[0195] $\Delta BD_C = cu_transquant_bypass_flag ? 0 : BitDepthMax - BitDepthC$

[0196] $O_Y = cu_transquant_bypass_flag || (BitDepthMax == BitDepthY) ? 0 : 1 << (\Delta BD_Y - 1)$

[0197] $O_C = cu_transquant_bypass_flag || (BitDepthMax == BitDepthC) ? 0 : 1 << (\Delta BD_C - 1)$

[0198] $x = 0..blkSize - 1, y = 0..blkSize - 1$ 인 잔차 샘플들 r_Y, r_{Cb} , 및 r_{Cr} 의 $(blkSize) \times (blkSize)$ 어레이들은 다음과 같이 수정될 수도 있다:

[0199] 잔차 샘플들 $rY[x][y]$, $rCb[x][y]$, 및 $rCr[x][y]$ 은 다음과 같이 수정될 수도 있다:

[0200] $rY[x][y] = rY[x][y] << \Delta BD_Y$

[0201] $rCb[x][y] = rCb[x][y] << (\Delta BD_C + 1 - cu_transquant_bypass_flag)$

[0202] $rCr[x][y] = rCr[x][y] << (\Delta BD_C + 1 - cu_transquant_bypass_flag)$

[0203] $tmp = rY[x][y] - (rCb[x][y] >> 1)$

[0204] $rY[x][y] = tmp + rCb[x][y]$

[0205] $rCb[x][y] = tmp - (rCr[x][y] >> 1)$

[0206] $rCr[x][y] = rCb[x][y] + rCr[x][y]$

[0207] $rY[x][y] = (rY[x][y] + O_Y) >> \Delta BD_Y$

[0208] $rCb[x][y] = (rCb[x][y] + O_C) >> \Delta BD_C$

[0209] $rCr[x][y] = (rCr[x][y] + O_C) >> \Delta BD_C$

[0210] 상기 예들에서, 함수 \max 는 2 개의 변수들 사이의 더 큰 값을 유도하기 위하여 이용될 수도 있다. 대안적으로, 라운딩 오프셋 (rounding offset) 들 O_Y 및 O_C 중의 적어도 하나는 0 과 동일하게 설정될 수도 있다.

[0211] 대안적으로, 무손실 코딩에 대해서도, 루마 및 크로마 컴포넌트들의 비트-심도가 상이할 때에는, 컴포넌트들이 필요할 경우에 동일한 비트-심도 정밀도로 시프트될 수도 있다. 인코더 측에서는, 순방향 변환 후에, 원래의 비트-심도 정밀도로 다시 시프트될 필요가 없을 수도 있다. 다시 말해서, 3 개의 컴포넌트들의 코딩된 잔차는 그것이 무손실 코딩인 것을 확실하게 하기 위하여 동일한 비트-심도일 수도 있다. 손실 코딩과는 상이하게, 입력들은 무손실 코딩을 위한 역방향 YCoCg-R 변환을 호출하기 전에 수정되지 않을 수도 있다. 우측 시프트는 출력들이 원래의 입력들과 동일한 비트-심도에 있는 것을 확실하게 하기 위하여 여전히 필요할 수

도 있다.

[0212] 이 예에서, 적응적 컬러 변환을 이용하는 변환 블록들에 대한 잔차 수정 프로세스는 ChromaArrayType 이 3 과 동일할 때에 호출될 수도 있다. 이 프로세스에 대한 입력들은 블록 사이즈를 특정하는 변수 blkSize, 엘리먼트들 $rY[x][y]$ 을 갖는 루마 잔차 샘플들 rY 의 $(blkSize) \times (blkSize)$ 어레이, 엘리먼트들 $rCb[x][y]$ 을 갖는 크로마 잔차 샘플들 rCb 의 $(blkSize) \times (blkSize)$ 어레이, 및 엘리먼트들 $rCr[x][y]$ 을 갖는 크로마 잔차 샘플들 rCr 의 $(blkSize) \times (blkSize)$ 어레이를 포함할 수도 있다. 이 프로세스에 대한 출력들은 루마 잔차 샘플들의 수정된 $(blkSize) \times (blkSize)$ 어레이 r_Y , 크로마 잔차 샘플들의 수정된 $(blkSize) \times (blkSize)$ 어레이 r_{Cb} , 및 크로마 잔차 샘플들의 수정된 $(blkSize) \times (blkSize)$ 어레이 r_{Cr} 를 포함할 수도 있다.

[0213] 변수들 ΔY 및 ΔC 은 다음과 같이 유도될 수도 있다:

[0214] $BitDepthMax = \max(BitDepthY, BitDepthC)$

[0215] $\Delta Y = BitDepthMax - BitDepthY$

[0216] $\Delta C = BitDepthMax - BitDepthC$

[0217] $OY = (BitDepthMax == BitDepthY) ? 0 : 1 \ll (\Delta Y - 1)$

[0218] $OC = (BitDepthMax == BitDepthC) ? 0 : 1 \ll (\Delta C - 1)$

[0219] $x = 0..blkSize - 1, y = 0..blkSize - 1$ 인 잔차 샘플들 r_Y, r_{Cb} , 및 r_{Cr} 의 $(blkSize) \times (blkSize)$ 어레이들은 다음과 같이 수정될 수도 있다:

[0220] cu 트랜스퀀트 바이패스 플래그 (transquant bypass flag) 가 0 과 동일할 때, 잔차 샘플들 $rY[x][y]$, $rCb[x][y]$, 및 $rCr[x][y]$ 은 다음과 같이 수정될 수도 있다:

[0221] $rY[x][y] = rY[x][y] \ll \Delta Y$

[0222] $rCb[x][y] = rCb[x][y] \ll (\Delta C + 1 - cu_transquant_bypass_flag)$

[0223] $rCr[x][y] = rCr[x][y] \ll (\Delta C + 1 - cu_transquant_bypass_flag)$

[0224] $tmp = rY[x][y] - (rCb[x][y] \gg 1)$

[0225] $rY[x][y] = tmp + rCb[x][y]$

[0226] $rCb[x][y] = tmp - (rCr[x][y] \gg 1)$

[0227] $rCr[x][y] = rCb[x][y] + rCr[x][y]$

[0228] $rY[x][y] = (rY[x][y] + OY) \gg \Delta Y$

[0229] $rCb[x][y] = (rCb[x][y] + OC) \gg \Delta C$

[0230] $rCr[x][y] = (rCr[x][y] + OC) \gg \Delta C$

[0231] 대안적으로, O_Y 및 O_C 중의 하나는 0 과 동일할 수도 있다.

[0232] 비디오 인코더 (20) 는 예컨대, 프레임 헤더, 블록 헤더, 슬라이스 헤더, 또는 GOP 헤더에서, 블록-기반 선택스 데이터, 프레임-기반 선택스 데이터, 및 GOP-기반 선택스 데이터와 같은 선택스 데이터를 비디오 디코더 (30) 로 추가로 전송할 수도 있다. GOP 선택스 데이터는 개개의 GOP 에서의 프레임들의 수를 기술할 수도 있고, 프레임 선택스 데이터는 대응하는 프레임을 인코딩하기 위하여 이용된 인코딩/예측 모드를 표시할 수도 있다.

[0233] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각, 적용가능한 경우, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적 회로 (ASIC) 들, 필드 프로그래밍가능한 게이트 어레이 (FPGA) 들, 개별 로직 회로부, 소프트웨어, 하드웨어, 펌웨어 또는 그 임의의 조합들과 같은 다양한 적당한 인코더 또는 디코더 회로부 중의 임의의 것으로서 구현될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 인코더들 또는 디코더들 중의 어느 하나는 조합된 비디오 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 셀룰러 전화와 같은 무선 통신 디바이스를

포함할 수도 있다.

- [0234] 도 2 는 컬러-공간 변환 프로세스를 이용하여 비디오 블록들을 인코딩하기 위한 기법들을 구현할 수도 있는 비디오 인코더 (20) 의 예를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 소정의 비디오 프레임 또는 픽처 내의 비디오에서의 공간적 중복성을 감소시키거나 제거하기 위하여 공간적 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내의 비디오에 있어서의 시간적 중복성을 감소시키거나 제거하기 위하여 시간적 예측에 의존한다. 인트라-모드 (I 모드) 는 몇몇 공간 기반 코딩 모드들 중의 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 과 같은 인터-모드들은 몇몇 시간 기반 코딩 모드들 중의 임의의 것을 지칭할 수도 있다.
- [0235] 도 2 에서 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩되어야 할 비디오 프레임 내의 현재의 비디오 블록을 수신한다. 도 2 의 예에서, 비디오 인코더 (20) 는 모드 선택 유닛 (40), 참조 픽처 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 모드 선택 유닛 (40) 은 궁극적으로, 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라 예측 유닛 (46), 및 파티션 유닛 (48) 을 포함한다. 비디오 블록 재구성을 위하여, 비디오 인코더 (20) 는 또한, 역양자화 유닛 (58), 역변환 유닛 (60), 및 합산기 (62) 를 포함한다. 디블록킹 필터 (deblocking filter; 도 2 에서 도시되지 않음) 는 또한, 블록 경계들을 필터링하여 재구성된 비디오로부터 블록화 아티팩트 (blockiness artifact) 들을 제거하기 위하여 포함될 수도 있다. 희망하는 경우, 디블록킹 필터는 전형적으로 합산기 (62) 의 출력을 필터링할 것이다. (루프 내의 또는 루프 이후의) 추가적인 필터들은 또한, 디블록킹 필터에 추가하여 이용될 수도 있다. 이러한 필터들은 간결함을 위하여 도시되어 있지 않지만, 희망하는 경우, (인-루프 (in-loop) 필터로서) 합산기 (50) 의 출력을 필터링할 수도 있다.
- [0236] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩되어야 할 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 다수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 시간적 예측을 제공하기 위하여, 하나 이상의 참조 프레임들 내의 하나 이상의 블록들에 관련하여 수신된 비디오 블록의 인터-예측 코딩을 수행한다. 인트라 예측 유닛 (46) 은 대안적으로, 공간적 예측을 제공하기 위하여, 코딩되어야 할 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 관련하여 수신된 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다. 비디오 인코더 (20) 는 예컨대, 비디오 데이터의 각각의 블록에 대한 적절한 코딩 모드를 선택하기 위하여, 다수의 코딩 패스 (coding pass) 들을 수행할 수도 있다.
- [0237] 또한, 파티션 유닛 (48) 은 이전의 코딩 패스들에서의 이전의 파티셔닝 방식들의 평가에 기초하여, 비디오 데이터의 블록들을 서브-블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛 (48) 은 초기에, 프레임 또는 슬라이스를 LCU 들로 파티셔닝할 수도 있고, 레이트-왜곡 (rate-distortion) 분석 (예컨대, 레이트-왜곡 최적화) 에 기초하여 LCU 들의 각각을 서브-CU 들로 파티셔닝할 수도 있다. 모드 선택 유닛 (40) 은 서브-CU 들로의 LCU 의 파티셔닝을 표시하는 쿼드트리 데이터 구조를 추가로 생성할 수도 있다. 쿼드트리의 리프-노드 CU 들은 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다.
- [0238] 모드 선택 유닛 (40) 은 예컨대, 여러 결과들에 기초하여 코딩 모드들 중의 하나, 인트라 또는 인터를 선택할 수도 있고, 결과적인 인트라-코딩된 또는 인터-코딩된 블록을, 잔차 블록 데이터를 생성하기 위하여 합산기 (50) 에, 그리고 참조 프레임으로서의 이용을 위한 인코딩된 블록을 재구성하기 위하여 합산기 (62) 에 제공한다. 모드 선택 유닛 (40) 은 또한, 모션 벡터들, 인트라-모드 표시자들, 파티션 정보, 및 다른 이러한 신택스 정보와 같은 신택스 엘리먼트들을 엔트로피 인코딩 유닛 (56) 에 제공한다.
- [0239] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적인 목적들을 위하여 별도로 예시되어 있다. 모션 추정 유닛 (42) 에 의해 수행된 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 예를 들어, 모션 벡터는 현재의 프레임 (또는 다른 코딩된 유닛) 내에서 코딩되고 있는 현재의 블록에 대한 참조 프레임 (또는 다른 코딩된 유닛) 내의 예측 블록에 관련하여 현재의 비디오 프레임 또는 픽처 내의 비디오 블록의 PU 의 변위를 표시할 수도 있다. 예측 블록은, 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는 픽셀 차이의 측면에서, 코딩되어야 할 블록과 근접하게 일치하는 것으로 구해지는 블록이다. 일부의 예들에서, 비디오 인코더 (20) 는 참조 픽처 메모리 (64) 내에 저장된 참조 픽처들의 정수-미만 (sub-integer) 픽셀 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는

참조 픽처의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 보간할 수도 있다. 그러므로, 모션 추정 유닛 (42) 은 전체 픽셀 위치들 및 분수 픽셀 위치들에 관련하여 모션 검색을 수행할 수도 있고, 분수 픽셀 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0240] 모션 추정 유닛 (42) 은 PU 의 위치를 참조 픽처의 예측 블록의 위치와 비교함으로써, 인터-코딩된 슬라이스에 서의 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 참조 픽처는 제 1 참조 픽처 리스트 (List 0) 또는 제 2 참조 픽처 리스트 (List 1) 로부터 선택될 수도 있고, 이들의 각각은 참조 픽처 메모리 (64) 내에 저장된 하나 이상의 참조 픽처들을 식별한다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0241] 모션 보상 유닛 (44) 에 의해 수행된 모션 보상은 모션 추정 유닛 (42) 에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페치 (fetch) 하거나 생성하는 것을 수반할 수도 있다. 또한, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 일부의 예들에서 기능적으로 통합될 수도 있다. 현재의 비디오 블록의 PU 에 대한 모션 벡터를 수신할 시에, 모션 보상 유닛 (44) 은 모션 벡터가 참조 픽처 리스트들 중의 하나에서 가리키는 예측 블록을 로케이팅할 수도 있다. 합산기 (50) 는 이하에서 논의된 바와 같이, 코딩되고 있는 현재의 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 일반적으로, 모션 추정 유닛 (42) 은 루마 컴포넌트들에 관련하여 모션 추정을 수행하고, 모션 보상 유닛 (44) 은 크로마 컴포넌트들 및 루마 컴포넌트들의 양자에 대한 루마 컴포넌트들에 기초하여 계산된 모션 벡터들을 이용한다. 모드 선택 유닛 (40) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 이용을 위한 비디오 블록들 및 비디오 슬라이스와 연관된 신텍스 엘리먼트들을 생성할 수도 있다.

[0242] 인트라 예측 유닛 (46) 은 위에서 설명된 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 수행된 인터-예측에 대한 대안으로서, 현재의 블록을 인트라-예측할 수도 있다. 특히, 인트라 예측 유닛 (46) 은 현재의 블록을 인코딩하기 위하여 이용하기 위한 인트라-예측 모드를 결정할 수도 있다. 일부의 예들에서, 인트라 예측 유닛 (46) 은 예컨대, 별도의 인코딩 패스들 동안에 다양한 인트라-예측 모드들을 이용하여 현재의 블록을 인코딩할 수도 있고, 인트라 예측 유닛 (46) (또는 일부의 예들에서, 모드 선택 유닛 (40)) 은 테스트된 모드들로부터 이용하기 위한 적절한 인트라-예측 모드를 선택할 수도 있다.

[0243] 예를 들어, 인트라 예측 유닛 (46) 은 다양한 테스트된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 이용하여 레이트-왜곡 값들을 계산할 수도 있고, 테스트된 모드들 중에서 최상의 레이트-왜곡 특성들을 가지는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로, 인코딩된 블록과, 인코딩된 블록을 생성하기 위하여 인코딩되었던 원래의 인코딩되지 않은 블록과의 사이의 왜곡 (또는 에러) 의 양뿐만 아니라, 인코딩된 블록을 생성하기 위하여 이용된 비트레이트 (즉, 비트들의 수) 를 결정한다. 인트라 예측 유닛 (46) 은 어느 인트라-예측 모드가 블록에 대한 최상의 레이트-왜곡 값을 나타내는지 결정하기 위하여 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비 (ratio) 들을 계산할 수도 있다.

[0244] 블록에 대한 인트라-예측 모드를 선택한 후, 인트라 예측 유닛 (46) 은 블록에 대한 선택된 인트라-예측 모드를 표시하는 정보를 엔트로피 인코딩 유닛 (56) 에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 선택된 인트라-예측 모드를 표시하는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 복수의 인트라-예측 모드 인덱스 표들 및 복수의 수정된 인트라-예측 모드 인덱스 표들 (또한 코드워드 맵핑 표 (codeword mapping table) 들로서 지칭됨) 을 포함할 수도 있는 송신된 비트스트림 구성 데이터에서, 다양한 블록들에 대한 인코딩 컨텍스트들의 정의들과, 컨텍스트들의 각각에 대해 이용하기 위한 가장 가능성 있는 인트라-예측 모드, 인트라-예측 모드 인덱스 표, 및 수정된 인트라-예측 모드 인덱스 표의 표시들을 포함할 수도 있다.

[0245] 비디오 인코더 (20) 는 코딩되고 있는 원래의 비디오 블록으로부터 모드 선택 유닛 (40) 으로부터의 예측 데이터를 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50) 는 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 프로세싱 유닛 (52) 은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 유닛 (52) 은 DCT 와 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 웨이블릿 변환들, 정수 변환 (integer transform) 들, 서브-대역 변환 (sub-band transform) 들 또는 다른 타입들의 변환들이 또한 이용될 수 있다. 여하튼, 변환 프로세싱 유닛 (52) 은 변환을 잔차 블록에 적용하여, 잔차 변환 계수들의 블록을 생성한다. 변환은 잔차 정보를 픽셀 값 도메인으로부터, 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다. 변환 프로세싱 유닛 (52) 은 결과적인 변환 계수들을 양자화 유닛 (54) 으로 전송할 수도 있

다. 양자화 유닛 (54) 은 비트 레이트를 추가로 감소시키기 위하여 변환 계수들을 양자화한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화도 (degree of quantization) 는 양자화 파라미터를 조절함으로써 수정될 수도 있다. 일부의 예들에서, 다음으로, 양자화 유닛 (54) 은 양자화된 변환 계수들을 포함하는 행렬의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56) 은 스캔을 수행할 수도 있다.

[0246] 양자화에 후속하여, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트-적응 가변 길이 코딩 (context adaptive variable length coding; CAVLC), 컨텍스트 적응 2 진 산술 코딩 (CABAC), 선택스-기반 컨텍스트-적응 2 진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 간격 파티셔닝 엔트로피 (probability interval partitioning entropy; PIPE) 코딩 또는 또 다른 엔트로피 코딩 기법을 수행할 수도 있다. 컨텍스트-기반 엔트로피 코딩의 경우, 컨텍스트는 이웃하는 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 코딩에 후속하여, 인코딩된 비트스트림은 또 다른 디바이스 (예컨대, 비디오 디코더 (30)) 에 송신될 수도 있거나, 더 이후의 송신 또는 취출을 위해 아카이빙될 수도 있다.

[0247] 이 개시물의 기법들에 따르면, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 현재의 개시물의 하나 이상의 기법들을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정할 수도 있다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 선택스 엘리먼트의 값을 추가로 설정할 수도 있다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 추가로 적용할 수도 있다.

[0248] 역양자화 유닛 (58) 및 역변환 유닛 (60) 은 예컨대, 참조 블록으로서의 더 이후의 이용을 위한, 픽셀 도메인에서의 잔차 블록을 재구성하기 위하여 역양자화 및 역변환을 각각 적용한다. 모션 보상 유닛 (44) 은 잔차 블록을 참조 픽처 메모리 (64) 의 프레임들 중의 하나의 프레임의 예측 블록에 추가함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한, 모션 추정 시에 이용하기 위한 정수-미만 픽셀 값들을 계산하기 위하여 하나 이상의 보간 필터들을 재구성된 잔차 블록에 적용할 수도 있다. 합산기 (62) 는 참조 픽처 메모리 (64) 에서의 저장을 위한 재구성된 비디오 블록을 생성하기 위하여, 재구성된 잔차 블록을 모션 보상 유닛 (44) 에 의해 생성된 모션 보상된 예측 블록에 가산한다. 재구성된 비디오 블록은 추후의 비디오 프레임에서 블록을 인터-코딩하기 위하여, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 참조 블록으로서 이용될 수도 있다.

[0249] 이러한 방식으로, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 현재의 개시물의 하나 이상의 기법들을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛을 위해 컬러-공간 변환을 이용할 것인지 여부를 결정할 수도 있다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여 컬러-공간 변환의 이용을 표시하도록 코딩 유닛의 선택스 엘리먼트의 값을 추가로 설정할 수도 있다. 일부의 예들에서, 선택스 엘리먼트는 1-비트 플래그를 포함한다. 일부의 예들에서, 선택스 엘리먼트는 코딩 유닛이 인트라-펄스 코드 변조 (intra-pulse code modulation; IPCM) 모드 외의 모드를 이용하여 코딩될 때에 시그널링된다. 다른 예들에서, 선택스 엘리먼트는 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에만 시그널링된다. 일부의 예들에서, 선택스 엘리먼트에 대한 1 의 값은 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시한다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛을 인코딩함에 있어서 컬러-공간 변환 프로세스를 추가로 적용할 수도 있다.

[0250] 일부의 예들에서, 선택스 엘리먼트는 시그널링되지 않을 수도 있다. 예를 들어, 선택스 엘리먼트는 코딩 유닛이 인트라 코딩될 때, 그리고 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 시그널링되지 않을 수도 있다. 또 다른 예에서, 선택스 엘리먼트는 코딩 유닛이 팔레트 모드로 코딩될 때에 시그널링되지 않을 수도 있다.

[0251] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 시퀀스 파라미터 세트에 대한 컬러-공간 변환을 이용할 것인지 여부를 추가로 결정할 수도 있다. 시퀀스 파라미터 세트에 대한 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛은

신택스 엘리먼트의 값을 설정할 수도 있다. 제 2 신택스 엘리먼트는 시퀀스 파라미터 세트와 연관될 수도 있고, 컬러-공간 변환이 시퀀스 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정될 수도 있다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛은 컬러-공간 변환 프로세스를 시퀀스 파라미터 세트의 비디오 데이터에 추가로 적용할 수도 있다.

[0252] 일부의 예들에서, 신택스 엘리먼트는 제 1 신택스 엘리먼트이고, 제 1 신택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 픽처 파라미터 세트를 위해 컬러-공간 변환을 이용할 것인지 여부를 추가로 결정할 수도 있다. 픽처 파라미터 세트를 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20)의 엔트로피 인코딩 유닛은 신택스 엘리먼트의 값을 설정할 수도 있다. 제 2 신택스 엘리먼트는 픽처 파라미터 세트와 연관될 수도 있고, 컬러-공간 변환이 픽처 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정될 수도 있다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛은 컬러-공간 변환 프로세스를 픽처 파라미터 세트의 비디오 데이터에 추가로 적용할 수도 있다.

[0253] 일부의 예들에서, 신택스 엘리먼트는 제 1 신택스 엘리먼트이고, 제 1 신택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 슬라이스 헤더를 위해 컬러-공간 변환을 이용할 것인지 여부를 추가로 결정할 수도 있다. 슬라이스 헤더를 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20)의 엔트로피 인코딩 유닛은 신택스 엘리먼트의 값을 설정할 수도 있다. 제 2 신택스 엘리먼트는 슬라이스 헤더와 연관될 수도 있고, 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정될 수도 있다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛은 컬러-공간 변환 프로세스를 슬라이스 헤더의 비디오 데이터에 추가로 적용할 수도 있다.

[0254] 일부의 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 표시하는 픽처 파라미터 세트의 데이터를 추가로 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 표시하는 시퀀스 파라미터 세트의 데이터를 인코딩할 수도 있다. 또 다른 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 표시하는 슬라이스 헤더의 데이터를 인코딩할 수도 있다. 일부의 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 코딩 유닛의 복수의 영역들의 각각에 대한 컬러-공간 역변환 프로세스를 유도할 수도 있고, 여기서, 영역들은 하이 레벨 신택스에 기초하여 분할된다. 일부의 예들에서, 컬러-공간 역변환 프로세스는 고정될 수도 있다.

[0255] 일부의 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 표시하는 변환 유닛 (TU) 레벨에서의 데이터를 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 표시하는 예측 유닛 (PU) 레벨에서의 데이터를 인코딩할 수도 있다. 또 다른 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 표시하는 슬라이스 헤더의 데이터를 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 표시하는 보충 강화 정보 (supplemental enhancement information; SEI) 메시지의 데이터를 인코딩할 수도 있다.

[0256] 일부의 예들에서, 신택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하는 제 1 신택스 엘리먼트이다. 이러한 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 코딩 유닛에서의 제 1 픽셀에 대한 팔레트 인덱스의 값을 결정할 수도 있다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 제 1 픽셀을 바로 뒤따르는 스캔 순서에서의 하나 이상의 다음 픽셀들에 대한 팔레트 인덱스들의 값들을 추가로 결정할 수도 있다. 팔레트 인덱스들의 값들을 결정하는 것은, 픽셀이 제 1 픽셀에 대한 팔레트 인덱스의 값과는 상이한 값을 갖는 팔레트 인덱스를 가질 때까지 팔레트 인덱스들의 값들을 결정하는 것을 포함할 수도 있다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 다음 픽셀들에 대하여 결정된 팔레트 인덱스 값들의 수를 추가로 결정할 수도 있다. 팔레트 인덱스 값들의 수에 기초하여, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 제 1 모드 또는 제 2 모드의 어느 하나를 이용하여 팔레트 인덱스를 인코딩할 수도 있다. 팔레트 인덱스 값들의 수가 1 보다 더 클 때, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 제 1 모드가 팔레트 인덱스를 인코딩하기 위하여 이용되었다는 것을 표시하기 위하여 제 2 신택스 엘리먼트의 값을 설정할 수도 있고, 제 3 신택스 엘리먼트의 값을 팔레트 인덱스 값들의 수와 동일하게 설정할 수도 있다. 팔레트 인덱스 값들의 수가 1 이하일 때, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 제 2 모드가 팔레트 인덱스를 인코딩하기 위하여 이용되었다는 것을 표시하기 위하여 제 2 신택스 엘리

먼트의 값을 설정할 수도 있다. 일부의 예들에서, 제 1 모드는 실행 모드이고, 제 2 모드는 픽셀 모드이다.

[0257] 일부의 예들에서, 선택스 엘리먼트를 인코딩하는 것은, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하는 보충 강화 정보 (SEI) 메시지의 데이터를 인코딩하는 것을 포함할 수도 있다. 일부의 예들에서, 컬러-공간 역변환 프로세스를 적용하는 것은, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 이 고정된 컬러-공간 역변환 행렬을 적용하는 것을 포함할 수도 있다. 일부의 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg 행렬이다. 다른 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg-R 행렬이다.

[0258] 일부의 예들에서, 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 비디오 데이터에 적용하기 전에 하나 이상의 컴포넌트들의 값을 이용하여 잔차 블록을 추가로 생성할 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 비디오 데이터의 잔차 블록에 추가로 적용할 수도 있다.

[0259] 도 3 은 비디오 블록들의 일부가 컬러-공간 변환 프로세스를 이용하여 인코딩되었던 비디오 블록들을 디코딩하기 위한 기법들을 구현할 수도 있는 비디오 디코더 (30) 의 예를 예시하는 블록도이다. 도 3 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (70), 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 역양자화 유닛 (76), 역변환 유닛 (78), 참조 픽처 메모리 (82), 및 합산기 (80) 를 포함한다. 일부의 예들에서, 비디오 디코더 (30) 는 비디오 인코더 (20) (도 2) 에 대하여 설명된 인코딩 패스와 일반적으로 상반되는 디코딩 패스를 수행할 수도 있다. 모션 보상 유닛 (72) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있는 반면, 인트라 예측 유닛 (74) 은 엔트로피 디코딩 유닛 (70) 으로부터 수신된 인트라-예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.

[0260] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 선택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 양자화된 계수들, 모션 벡터들 또는 인트라-예측 모드 표시자들, 및 다른 선택스 엘리먼트들을 생성하기 위하여 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (70) 은 모션 벡터들 및 다른 선택스 엘리먼트들을 모션 보상 유닛 (72) 으로 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 선택스 엘리먼트들을 수신할 수도 있다.

[0261] 이 개시물의 기법들에 따르면, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 현재의 개시물의 하나 이상의 기법들을 수행할 수도 있다. 예를 들어, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 코딩 유닛의 선택스 엘리먼트를 디코딩할 수도 있다. 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시할 수도 있다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 추가로 결정할 수도 있다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 추가로 적용할 수도 있다. 디코더 측 상에서는, 코딩 모드에 관계 없이, 컬러-공간 순방향 변환 프로세스를 수행할 필요가 없다. 일부의 예들에서, 컬러-공간 역변환 프로세스는 고정될 수도 있다.

[0262] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 인트라 예측 유닛 (74) 은 시그널링된 인트라 예측 모드와, 현재의 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B, P, 또는 GPB) 슬라이스로서 코딩될 때, 모션 보상 유닛 (72) 은 모션 벡터들과, 엔트로피 디코딩 유닛 (70) 으로부터 수신된 다른 선택스 엘리먼트들에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 픽처 리스트들 중의 하나 내의 참조 픽처들 중의 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는 참조 픽처 메모리 (82) 내에 저장된 참조 픽처들에 기초하여, 디폴트 구성 (default construction) 기법들을 이용하여 참조 프레임 리스트들, List 0 및 List 1 을 구성할 수도 있다. 모션 보상 유닛 (72) 은 모션 벡터들 및 다른 선택스 엘리먼트들을 파싱 (parsing) 함으로써 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 디코딩되고 있는 현재의 비디오 블록에 대한 예측 블록들을 생성하기 위하여 예측 정보를 이용한다. 예를 들어, 모션 보상 유닛 (72) 은 비디오 슬라이스의 비디오 블록들을 코딩하기 위해 이용된 예측 모드 (예컨대, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 타입

(예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 픽처 리스트들 중의 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재의 비디오 슬라이스에서의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위하여, 수신된 선택스 엘리먼트들의 일부를 이용한다.

[0263] 모션 보상 유닛 (72) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (72) 은 참조 블록들의 정수-미만 픽셀들에 대한 보간된 값들을 계산하기 위하여, 비디오 블록들의 인코딩 동안에 비디오 인코더 (20) 에 의해 이용된 바와 같은 보간 필터들을 이용할 수도 있다. 이 경우, 모션 보상 유닛 (72) 은 수신된 선택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 이용된 보간 필터들을 결정할 수도 있고, 예측 블록들을 생성하기 위하여 보간 필터들을 이용할 수도 있다.

[0264] 역양자화 유닛 (76) 은, 비트스트림에서 제공되며 엔트로피 디코딩 유닛 (70) 에 의해 디코딩된 양자화된 변환 계수들을 역양자화, 즉, 탈양자화 (de-quantize) 한다. 역양자화 프로세스는 적용되어야 할 양자화도와, 마찬가지로, 역양자화도를 결정하기 위하여, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 디코더 (30) 에 의해 계산된 양자화 파라미터 QP_V 의 이용을 포함할 수도 있다.

[0265] 역변환 유닛 (78) 은 픽셀 도메인에서 잔차 블록들을 생성하기 위하여, 역변환, 예컨대, 역 DCT, 역정수 변환, 또는 개념적으로 유사한 역변환 프로세스를 변환 계수들에 적용한다.

[0266] 모션 보상 유닛 (72) 이 모션 벡터들 및 다른 선택스 엘리먼트들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30) 는 역변환 유닛 (78) 으로부터의 잔차 블록들을 모션 보상 유닛 (72) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (80) 는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 희망하는 경우, 디블록킹 필터는 또한, 블록화 아티팩트들을 제거하기 위하여 디코딩된 블록들을 필터링하도록 적용될 수도 있다. (코딩 루프 내에 또는 코딩 루프 이후 중의 어느 하나에서의) 다른 루프 필터들은 또한, 픽셀 천이 (pixel transition) 들을 평활화하거나, 또는 이와 다르게 비디오 품질을 개선시키기 위하여 이용될 수도 있다. 다음으로, 소정의 프레임 또는 픽처에서의 디코딩된 비디오 블록들은, 후속 모션 보상을 위하여 이용된 참조 픽처들을 저장하는 참조 픽처 메모리 (82) 내에 저장된다. 참조 픽처 메모리 (82) 는 또한, 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에서의 나중의 제시를 위해 디코딩된 비디오를 저장한다.

[0267] 이러한 방식으로, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 현재의 개시물의 하나 이상의 기법들을 수행할 수도 있다. 예를 들어, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 코딩 유닛의 선택스 엘리먼트를 디코딩할 수도 있다. 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시할 수도 있다. 일부의 예들에서, 선택스 엘리먼트는 1-비트 플래그를 포함할 수도 있다. 일부의 예들에서, 코딩 유닛은 인트라-펄스 코드 변조 (IPCM) 모드 외의 모드에서 코딩되고, 선택스 엘리먼트는 IPCM 모드 외의 모드를 이용하는 코딩 유닛들에 대해서만 시그널링된다. 일부의 예들에서, 선택스 엘리먼트는 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시한다. 일부의 예들에서, 선택스 엘리먼트에 대한 1 의 값은 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시한다. 디코더 측 상에서는, 코딩 모드에 관계 없이, 컬러-공간 순방향 변환 프로세스를 수행할 필요가 없다.

[0268] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 선택스 엘리먼트의 값이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시하는지 여부를 추가로 결정할 수도 있다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 선택스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여 코딩 유닛을 디코딩함에 있어서 컬러-공간 역변환 프로세스를 추가로 적용할 수도 있다. 일부의 예들에서, 컬러-공간 역변환 프로세스는 고정될 수도 있다.

[0269] 일부의 예들에서, 선택스 엘리먼트는 컬러-공간 변환이 코딩 유닛을 인코딩하기 위하여 이용되지 않았다는 것을 표시할 수도 있다. 예를 들어, 선택스 엘리먼트는 코딩 유닛이 인트라 코딩될 때, 그리고 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에, 컬러-공간 변환이 코딩 유닛을 인코딩하기 위하여 이용되지 않았다는 것을 표시할 수도 있다. 또 다른 예에서, 선택스 엘리먼트는 코딩 유닛이 팔레트 모드로 코딩될 때에, 컬러-공간 변환이 코딩 유닛을 인코딩하기 위하여 이용되지 않았다는 것을 표시할 수도 있다. 이 예들에서, 선택스 엘리먼트는 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않을 수도 있고, 선택스 엘리먼트를 디코딩하는 것은 선택스 엘리먼트의 값을 추론하는 것을 포함할 수도 있다.

- [0270] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 2 선택스 엘리먼트를 추가로 디코딩할 수도 있다. 제 2 선택스 엘리먼트는 시퀀스 파라미터 세트 (sequence parameter set; SPS)와 연관될 수도 있고, 컬러-공간 변환이 SPS에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시할 수도 있다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 SPS에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정할 수도 있다. 이러한 예들에서, 제 1 선택스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)이, 제 2 선택스 엘리먼트가 컬러-공간 변환이 시퀀스 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 제 1 선택스 엘리먼트를 디코딩하도록 구성되는 것을 포함할 수도 있다.
- [0271] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 2 선택스 엘리먼트를 추가로 디코딩할 수도 있다. 제 2 선택스 엘리먼트는 픽처 파라미터 세트 (picture parameter set; PPS)와 연관될 수도 있고, 컬러-공간 변환이 PPS에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시할 수도 있다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 PPS에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정할 수도 있다. 이러한 예들에서, 제 1 선택스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)이, 제 2 선택스 엘리먼트가 컬러-공간 변환이 픽처 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 제 1 선택스 엘리먼트를 디코딩하도록 구성되는 것을 포함할 수도 있다.
- [0272] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 2 선택스 엘리먼트를 추가로 디코딩할 수도 있다. 제 2 선택스 엘리먼트는 슬라이스 헤더와 연관될 수도 있고, 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시할 수도 있다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정할 수도 있다. 이러한 예들에서, 제 1 선택스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)이, 제 2 선택스 엘리먼트가 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 제 1 선택스 엘리먼트를 디코딩하도록 구성되는 것을 포함할 수도 있다.
- [0273] 일부의 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 컬러-공간 역변환 프로세스를 표시하는 픽처 파라미터 세트 (PPS)의 데이터를 추가로 디코딩할 수도 있다. 다른 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 컬러-공간 역변환 프로세스를 표시하는 시퀀스 파라미터 세트 (SPS)의 데이터를 디코딩할 수도 있다. 또 다른 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 컬러-공간 역변환 프로세스를 표시하는 슬라이스 헤더의 데이터를 디코딩할 수도 있다. 일부의 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 코딩 유닛의 복수의 영역들의 각각에 대한 컬러-공간 역변환 프로세스를 유도할 수도 있고, 여기서, 영역들은 하이 레벨 선택스에 기초하여 분할된다.
- [0274] 일부의 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 컬러-공간 역변환 프로세스를 표시하는 변환 유닛 (TU) 레벨에서의 데이터를 디코딩할 수도 있다. 다른 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 컬러-공간 역변환 프로세스를 표시하는 예측 유닛 (PU) 레벨에서의 데이터를 디코딩할 수도 있다. 일부의 다른 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 슬라이스 헤더의 데이터로부터 컬러-공간 역변환 프로세스를 유도할 수도 있다. 또 다른 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 컬러-공간 역변환 프로세스를 표시하는 보충 강화 정보 (SEI) 메시지의 데이터를 디코딩할 수도 있다.
- [0275] 일부의 예들에서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하는 제 1 선택스 엘리먼트이다. 이러한 예들에서, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 2 선택스 엘리먼트의 값을 추가로 결정할 수도 있다. 선택스 엘리먼트는 제 1 모드 또는 제 2 모드가 코딩 유닛에서의 픽셀에 대한 팔레트 인덱스를 디코딩하기 위하여 이용되는지 여부를 표시할 수도 있다. 제 2 선택스 엘리먼트의 결정된 값에 기초하여, 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 제 1 모드 또는

제 2 모드의 어느 하나를 이용하여 팔레트 인덱스를 디코딩할 수도 있다. 제 1 모드를 이용할 때, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 팔레트 인덱스의 값을 결정할 수도 있고, 현재 디코딩되고 있는 픽셀을 바로 뒤따르는 스캔 순서에서의 픽셀들의 수를 표시하는 제 3 인덱스 엘리먼트의 값을 결정할 수도 있고, 스캔 순서에서의 다음 N 개의 픽셀들에 대한 팔레트 인덱스의 값을 결정하는 결과를 복제할 수도 있고, N 은 제 3 인덱스 엘리먼트의 값과 동일하다. 일부의 예들에서, 제 1 모드는 실행 모드이다. 제 2 모드를 이용할 때, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 팔레트 인덱스의 값을 결정할 수도 있고 픽셀에 대한 픽셀 샘플 값을 출력할 수도 있고, 여기서, 픽셀 샘플 값은 팔레트 인덱스의 값과 동일하다. 일부의 예들에서, 제 2 모드는 픽셀 모드이다.

[0276] 일부의 예들에서, 인덱스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하는 보충 강화 정보 (SEI) 메시지의 데이터를 디코딩하도록 구성되는 것을 포함한다. 일부의 예들에서, 컬러-공간 역변환 프로세스를 적용하는 것은, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 이 고정된 컬러-공간 역변환 행렬을 적용하도록 구성되는 것을 포함한다. 일부의 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg 행렬이다. 다른 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg-R 행렬이다.

[0277] 일부의 예들에서, 인덱스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 비디오 데이터에 적용하기 전에, 하나 이상의 컴포넌트들의 값들을 이용하여 잔차 블록을 생성할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 비디오 데이터의 잔차 블록에 적용할 수도 있다.

[0278] 도 4 는 현재의 개시물의 하나 이상의 기법들에 따라 35 개의 HEVC 예측 모드들을 예시하는 개념도이다. 현재의 HEVC 에서는, 각각의 예측 유닛 (PU) 의 루마 컴포넌트에 대하여, 인트라 예측 방법이 도 4 에 대하여 설명된 바와 같이, 33 개의 각도 예측 모드들 (2 로부터 34 까지 인덱싱됨), DC 모드 (1 로 인덱싱됨), 및 평면 모드 (0 으로 인덱싱됨) 와 함께 사용된다.

[0279] 상기 35 개의 인트라 모드들에 추가하여, 'I-PCM' 으로 명명된 모드가 하나 더 HEVC 에 의해 또한 채용된다. I-PCM 모드에서는, 예측, 변환, 양자화, 및 엔트로피 코딩이 우회되는 반면, 예측 샘플들은 미리 정의된 수의 비트들에 의해 코딩된다. I-PCM 모드의 주요 목적은 신호가 다른 모드들에 의해 효율적으로 코딩될 수 없을 때의 상황을 처리하기 위한 것이다.

[0280] 도 5 는 현재의 개시물의 하나 이상의 기법들에 따라 병합 및 진보된 모션 벡터 예측 (AMVP) 모드들에 대한 공간적으로 이웃하는 모션 벡터 후보들을 예시하는 개념도이다. 도 5 에 대하여 설명된 바와 같이, 공간적 MV 후보들은 특정 PU (PU0) 에 대하여 도 5 상에서 도시된 이웃하는 블록들로부터 유도되지만, 블록들로부터 후보들을 생성하는 방법들은 병합 및 AMVP 모드들에 대해 상이하다.

[0281] 병합 모드에서는, 4 개에 이르는 공간적 MV 후보들이 숫자들로 도 5 (a) 상에서 도시된 순서들로 유도될 수도 있고, 순서는 다음과 같다: 도 5 (a) 에서 도시된 바와 같이, 좌측 (0), 상부 (1), 상부 우측 (2), 하부 좌측 (3), 및 상부 좌측 (4).

[0282] AMVP 모드에서는, 이웃하는 블록들이 2 개의 그룹들로 분할된다: 도 5 (b) 상에서 도시된 바와 같이, 블록 0 및 1 로 구성되는 좌측 그룹과, 블록들 2, 3, 및 4 로 구성되는 상부 그룹. 각각의 그룹에 대하여, 시그널링된 참조 인덱스에 의해 표시된 것과 동일한 참조 픽처를 참조하는 이웃하는 블록에서의 잠재적인 후보는 그룹의 최종 후보를 형성하기 위하여 선택되어야 할 최고 우선순위를 가진다. 모든 이웃하는 블록들은 동일한 참조 픽처를 가리키는 모션 벡터를 포함하지 않는 것이 가능하다. 그러므로, 이러한 후보가 구해지지 않을 경우, 제 1 이용가능한 후보는 최종 후보를 형성하기 위하여 스케일링될 것이고, 이에 따라, 시간적 거리 차이들이 보상될 수 있다.

[0283] 도 6 은 현재의 개시물의 하나 이상의 기법들에 따라 인트라 블록 복사 (BC) 예를 예시하는 개념도이다. 도 6 에 대하여 설명된 바와 같이, 인트라 블록-복사 (BC) 가 RExt 내에 포함되었다. 인트라 BC 의 예는 도 6 에서와 같이 도시되고, 여기서, 현재의 CU 는 현재의 픽처/슬라이스의 이미 디코딩된 블록으로부터 예측된다. 현재의 인트라 BC 블록 사이즈는 8x8 로부터 64x64 까지의 범위인 CU 사이즈만큼 클 수 있지만, 일부의 애플리케이션들에서는, 추가의 제약들이 추가로 적용될 수도 있다.

[0284] 도 7 은 현재의 개시물의 하나 이상의 기법들에 따라, 인트라 8x8 블록에 대한 타겟 블록 및 참조 샘플의 예를

예시하는 개념도이다. 도 7에 대하여 이하에서 설명된 바와 같이, 변환 행렬은 참조 샘플 값들로부터 유도된다. 상이한 참조 샘플들은 인트라의 경우 및 인터의 경우에 대해 사용된다. 인트라 블록의 경우에 대하여, 타겟 블록 및 참조 샘플들은 도 7에서 도시되어 있다. 이 도면에서, 타겟 블록은 8x8 크로스해칭된 샘플들로 구성되고, 참조들은 스트라이프 표시되고 점 표시된 샘플들이다.

[0285] 인터 블록의 경우에 대하여, 행렬 유도를 위한 참조 샘플들은 모션 보상을 위한 참조 샘플과 동일하다. 시프트 연산을 실현하기 위하여, AMP 블록에서의 참조 샘플들은 샘플들의 수가 2의 거듭제곱이 되도록 서브-샘플링된다. 예를 들어, 12x16 블록에서의 참조 샘플들의 수는 2/3로 감소된다.

[0286] 현재의 개시물의 일부의 기법들에 따르면, 컬러-공간 변환 프로세스가 적용될 수도 있다. 이러한 예들에서는, 변환 프로세스가 호출되는지 여부는 시그널링될 필요가 없다. 게다가, 인코더 및 디코더 측들 양쪽 모두는 변환 행렬을 시그널링하기 위한 오버헤드를 회피하기 위하여 동일한 방법으로 변환 행렬을 유도할 수도 있다.

[0287] 도 8은 현재의 블록을 인코딩하기 위한 일 예의 방법을 예시하는 플로우차트이다. 현재의 블록은 현재의 CU 또는 현재의 CU의 부분을 포함할 수도 있다. 비디오 인코더 (20) (도 1 및 도 2)에 대해 설명되었지만, 다른 디바이스들은 도 8의 방법과 유사한 방법을 수행하도록 구성될 수도 있다는 것을 이해해야 한다.

[0288] 이 예에서, 비디오 인코더 (20)는 초기에 현재의 블록을 예측한다 (150). 예를 들어, 비디오 인코더 (20)는 현재의 블록에 대한 하나 이상의 예측 유닛 (PU)들을 계산할 수도 있다. 다음으로, 비디오 인코더 (20)는 예컨대, 변환 유닛 (TU)을 생성하기 위하여, 현재의 블록에 대한 잔차 블록을 계산할 수도 있다 (152). 잔차 블록을 계산하기 위하여, 비디오 인코더 (20)는 원래의 코딩되지 않은 블록과, 현재의 블록에 대한 예측된 블록과의 사이의 차이를 계산할 수도 있다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환을 이용할 것인지 여부를 결정할 수도 있다 (154). 비디오 인코더 (20)가 컬러-공간 변환을 이용하는 것으로 결정하는지 여부에 기초하여, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 이에 따라, 선택스 엘리먼트의 값을 설정할 수도 있다 (156). 일부의 예들에서, 선택스 엘리먼트는 1-비트 플래그를 포함한다. 일부의 예들에서, 선택스 엘리먼트는 코딩 유닛이 인트라-펄스 코드 변조 (IPCM) 모드 외의 모드를 이용하여 코딩될 때에 시그널링된다. 다른 예들에서, 선택스 엘리먼트는 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에만 시그널링된다. 일부의 예들에서, 선택스 엘리먼트에 대한 1의 값은 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시한다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 컬러-공간 변환 프로세스를 잔차 블록에 적용할 수도 있다 (158). 다음으로, 비디오 인코더 (20)는 잔차 블록의 계수들을 변환하고 양자화할 수도 있다 (160). 다음으로, 비디오 인코더 (20)는 잔차 블록의 양자화된 변환 계수들을 스캔할 수도 있다 (162). 스캔 동안에, 또는 스캔에 후속하여, 비디오 인코더 (20)는 계수들을 엔트로피 인코딩할 수도 있다 (164). 예를 들어, 비디오 인코더 (20)는 CAVLC 또는 CABAC를 이용하여 계수들을 인코딩할 수도 있다. 다음으로, 비디오 인코더 (20)는 블록의 엔트로피 코딩된 데이터를 출력할 수도 있다 (166).

[0289] 일부의 예들에서, 선택스 엘리먼트는 시그널링되지 않을 수도 있다. 예를 들어, 선택스 엘리먼트는 코딩 유닛이 인트라 코딩될 때, 그리고 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에 시그널링되지 않을 수도 있다. 또 다른 예에서, 선택스 엘리먼트는 코딩 유닛이 팔레트 모드로 코딩될 때에 시그널링되지 않을 수도 있다.

[0290] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 시퀀스 파라미터 세트에 대한 컬러-공간 변환을 이용할 것인지 여부를 추가로 결정할 수도 있다. 시퀀스 파라미터 세트에 대한 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20)의 엔트로피 인코딩 유닛은 선택스 엘리먼트의 값을 설정할 수도 있다. 제 2 선택스 엘리먼트는 시퀀스 파라미터 세트와 연관될 수도 있고, 컬러-공간 변환이 시퀀스 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정될 수도 있다. 비디오 인코더 (20)의 엔트로피 인코딩 유닛은 컬러-공간 변환 프로세스를 시퀀스 파라미터 세트의 비디오 데이터에 추가로 적용할 수도 있다.

[0291] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56)은 픽처 파라미터 세트를 위해 컬러-공간 변환을 이용할 것인지 여부를 추가로 결정할 수도 있다. 픽처 파라미터 세트를 위해

컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛은 선택스 엘리먼트의 값을 설정할 수도 있다. 제 2 선택스 엘리먼트는 픽처 파라미터 세트와 연관될 수도 있고, 컬러-공간 변환이 픽처 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정될 수도 있다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛은 컬러-공간 변환 프로세스를 픽처 파라미터 세트의 비디오 데이터에 추가로 적용할 수도 있다.

[0292] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 슬라이스 헤더를 위해 컬러-공간 변환을 이용할 것인지 여부를 추가로 결정할 수도 있다. 슬라이스 헤더를 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛은 선택스 엘리먼트의 값을 설정할 수도 있다. 제 2 선택스 엘리먼트는 슬라이스 헤더와 연관될 수도 있고, 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하도록 설정될 수도 있다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛은 컬러-공간 변환 프로세스를 슬라이스 헤더의 비디오 데이터에 추가로 적용할 수도 있다.

[0293] 일부의 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 표시하는 픽처 파라미터 세트의 데이터를 추가로 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 표시하는 시퀀스 파라미터 세트의 데이터를 인코딩할 수도 있다. 또 다른 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 표시하는 슬라이스 헤더의 데이터를 인코딩할 수도 있다. 일부의 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛의 복수의 영역들의 각각에 대한 컬러-공간 역변환 프로세스를 유도할 수도 있고, 여기서, 영역들은 하이 레벨 선택스에 기초하여 분할된다.

[0294] 일부의 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 표시하는 변환 유닛 (TU) 레벨에서의 데이터를 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 표시하는 예측 유닛 (PU) 레벨에서의 데이터를 인코딩할 수도 있다. 또 다른 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 표시하는 슬라이스 헤더의 데이터를 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 표시하는 보충 강화 정보 (SEI) 메시지의 데이터를 인코딩할 수도 있다.

[0295] 일부의 예들에서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하는 제 1 선택스 엘리먼트이다. 이러한 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 코딩 유닛에서의 제 1 픽셀에 대한 팔레트 인덱스의 값을 결정할 수도 있다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 제 1 픽셀을 바로 뒤따르는 스캔 순서에서의 하나 이상의 다음 픽셀들에 대한 팔레트 인덱스들의 값들을 추가로 결정할 수도 있다. 팔레트 인덱스들의 값들을 결정하는 것은, 픽셀이 제 1 픽셀에 대한 팔레트 인덱스의 값과는 상이한 값을 갖는 팔레트 인덱스를 가질 때까지 팔레트 인덱스들의 값들을 결정하는 것을 포함할 수도 있다. 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 다음 픽셀들에 대하여 결정된 팔레트 인덱스 값들의 수를 추가로 결정할 수도 있다. 팔레트 인덱스 값들의 수에 기초하여, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 제 1 모드 또는 제 2 모드의 어느 하나를 이용하여 팔레트 인덱스를 인코딩할 수도 있다. 팔레트 인덱스 값들의 수가 1 보다 더 클 때, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 제 1 모드가 팔레트 인덱스를 인코딩하기 위하여 이용되었다는 것을 표시하기 위하여 제 2 선택스 엘리먼트의 값을 설정할 수도 있고, 제 3 선택스 엘리먼트의 값을 팔레트 인덱스 값들의 수와 동일하게 설정할 수도 있다. 팔레트 인덱스 값들의 수가 1 이하일 때, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 제 2 모드가 팔레트 인덱스를 인코딩하기 위하여 이용되었다는 것을 표시하기 위하여 제 2 선택스 엘리먼트의 값을 설정할 수도 있다. 일부의 예들에서, 제 1 모드는 실행 모드이고, 제 2 모드는 픽셀 모드이다.

[0296] 일부의 예들에서, 선택스 엘리먼트를 인코딩하는 것은, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하는 보충 강화 정보 (SEI) 메시지의 데이터를 인코딩하는 것을 포함할 수도 있다. 일부의 예들에서, 컬러-공간 역변환 프로세스를 적용하는 것은, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 이 고정된 컬러-공간 역변환 행렬을 적용하는 것을 포함할 수도 있다. 일부의 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg 행렬이다. 다른 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg-R 행렬이다.

- [0297] 일부의 예들에서, 코딩 유닛을 위해 컬러-공간 변환을 이용하는 것으로 결정하는 것에 응답하여, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 비디오 데이터에 적용하기 전에 하나 이상의 컴포넌트들의 값들을 이용하여 잔차 블록을 추가로 생성할 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (56) 은 컬러-공간 변환 프로세스를 비디오 데이터의 잔차 블록에 추가로 적용할 수도 있다.
- [0298] 도 9 는 비디오 데이터의 현재의 블록을 디코딩하기 위한 일 예의 방법을 제시하는 플로우차트이다. 현재의 블록은 현재의 CU 또는 현재의 CU 의 부분을 포함할 수도 있다. 비디오 디코더 (30) (도 1 및 도 3) 에 대해 설명되었지만, 다른 디바이스들은 도 9 의 방법과 유사한 방법을 수행하도록 구성될 수도 있다는 것을 이해해야 한다.
- [0299] 비디오 디코더 (30) 는 현재의 블록에 대한 예측된 블록을 계산하기 위하여, 예컨대, 인트라-예측 또는 인터-예측 모드를 이용하여 현재의 블록을 예측할 수도 있다 (200). 비디오 디코더 (30) 는 또한, 현재의 블록에 대응하는 잔차 블록의 계수들에 대한 엔트로피 코딩된 데이터와 같은, 현재의 블록에 대한 엔트로피 코딩된 데이터를 수신할 수도 있다 (202). 비디오 디코더 (30) 는 잔차 블록의 계수들을 재생성하기 위하여 엔트로피 코딩된 데이터를 엔트로피 디코딩할 수도 있다 (204). 다음으로, 비디오 디코더 (30) 는 양자화된 변환 계수들의 블록을 생성하기 위하여, 재생성된 계수들을 역스캔할 수도 있다 (206). 다음으로, 비디오 디코더 (30) 는 계수들을 역변환하고 역양자화할 수도 있다 (208). 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 현재의 블록에 대한 신택스 엘리먼트를 디코딩할 수도 있다 (210). 일부의 예들에서, 신택스 엘리먼트는 1-비트 플래그를 포함할 수도 있다. 일부의 예들에서, 코딩 유닛은 인트라-필스 코드 변조 (IPCM) 모드 외의 모드에서 코딩되고, 신택스 엘리먼트는 IPCM 모드 외의 모드를 이용하는 코딩 유닛들에 대해서만 시그널링된다. 일부의 예들에서, 신택스 엘리먼트는 코딩 유닛의 변환 유닛에서 비-제로 계수들이 있을 때에 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시한다. 일부의 예들에서, 신택스 엘리먼트에 대한 1 의 값은 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었다는 것을 표시한다.
- [0300] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 그 신택스 엘리먼트에 대한 값을 결정할 수도 있고, 신택스 엘리먼트에 대한 값은 현재의 블록이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시할 수도 있다 (212). 컬러-공간 변환이 인코딩 프로세스에서 이용된 것으로 결정하는 것에 응답하여, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 잔차 블록을 재생성하기 위하여 컬러-공간 역변환 프로세스를 적용할 수도 있다 (214). 비디오 디코더 (30) 는 궁극적으로, 예측된 블록 및 잔차 블록을 합성함으로써 현재의 블록을 디코딩할 수도 있다 (216). 디코더 측 상에서는, 코딩 모드에 관계 없이, 컬러-공간 순방향 변환 프로세스를 수행할 필요가 없을 수도 있다. 일부의 예들에서, 컬러-공간 역변환 프로세스는 고정될 수도 있다.
- [0301] 일부의 예들에서, 신택스 엘리먼트는 컬러-공간 변환이 코딩 유닛을 인코딩하기 위하여 이용되지 않았다는 것을 표시할 수도 있다. 예를 들어, 신택스 엘리먼트는 코딩 유닛이 인트라 코딩될 때, 그리고 코딩 유닛 내의 예측 유닛의 루마 예측 모드 및 크로마 예측 모드가 상이할 때에, 컬러-공간 변환이 코딩 유닛을 인코딩하기 위하여 이용되지 않았다는 것을 표시할 수도 있다. 또 다른 예에서, 신택스 엘리먼트는 코딩 유닛이 팔레트 모드로 코딩될 때에, 컬러-공간 변환이 코딩 유닛을 인코딩하기 위하여 이용되지 않았다는 것을 표시할 수도 있다. 이 예들에서, 신택스 엘리먼트는 비디오 데이터를 포함하는 수신된 비트스트림에서 존재하지 않을 수도 있고, 신택스 엘리먼트를 디코딩하는 것은 신택스 엘리먼트의 값을 추론하는 것을 포함할 수도 있다.
- [0302] 일부의 예들에서, 신택스 엘리먼트는 제 1 신택스 엘리먼트이고, 제 1 신택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 2 신택스 엘리먼트를 추가로 디코딩할 수도 있다. 제 2 신택스 엘리먼트는 시퀀스 파라미터 세트 (SPS) 와 연관될 수도 있고, 컬러-공간 변환이 SPS 에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시할 수도 있다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 2 신택스 엘리먼트의 값이 컬러-공간 변환이 SPS 에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정할 수도 있다. 이러한 예들에서, 제 1 신택스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 이, 제 2 신택스 엘리먼트가 컬러-공간 변환이 시퀀스 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 제 1 신택스 엘리먼트를 디코딩하도록 구성되는 것을 포함할 수도 있다.
- [0303] 일부의 예들에서, 신택스 엘리먼트는 제 1 신택스 엘리먼트이고, 제 1 신택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 2 신택스 엘

리먼트를 추가로 디코딩할 수도 있다. 제 2 선택스 엘리먼트는 픽처 파라미터 세트 (PPS) 와 연관될 수도 있고, 컬러-공간 변환이 PPS 에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시할 수도 있다.

비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 PPS 에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정할 수도 있다. 이러한 예들에서, 제 1 선택스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 이, 제 2 선택스 엘리먼트가 컬러-공간 변환이 픽처 파라미터 세트에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 제 1 선택스 엘리먼트를 디코딩하도록 구성되는 것을 포함할 수도 있다.

[0304] 일부의 예들에서, 선택스 엘리먼트는 제 1 선택스 엘리먼트이고, 제 1 선택스 엘리먼트는 코딩 유닛과 연관된다. 일부의 이러한 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 2 선택스 엘리먼트를 추가로 디코딩할 수도 있다. 제 2 선택스 엘리먼트는 슬라이스 헤더와 연관될 수도 있고, 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블되는지 여부를 표시할 수도 있다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 2 선택스 엘리먼트의 값이 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는지 여부를 결정할 수도 있다. 이러한 예들에서, 제 1 선택스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 이, 제 2 선택스 엘리먼트가 컬러-공간 변환이 슬라이스 헤더에 대응하는 비디오 데이터에 대하여 인에이블된다는 것을 표시하는 것으로 결정하는 것에 응답하여 제 1 선택스 엘리먼트를 디코딩하도록 구성되는 것을 포함할 수도 있다.

[0305] 일부의 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 표시하는 픽처 파라미터 세트 (PPS) 의 데이터를 추가로 디코딩할 수도 있다. 다른 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 표시하는 시퀀스 파라미터 세트 (SPS) 의 데이터를 디코딩할 수도 있다. 또 다른 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 표시하는 슬라이스 헤더의 데이터를 디코딩할 수도 있다. 일부의 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 코딩 유닛의 복수의 영역들의 각각에 대한 컬러-공간 역변환 프로세스를 유도할 수도 있고, 여기서, 영역들은 하이 레벨 선택스에 기초하여 분할된다.

[0306] 일부의 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 표시하는 변환 유닛 (TU) 레벨에서의 데이터를 디코딩할 수도 있다. 다른 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 표시하는 예측 유닛 (PU) 레벨에서의 데이터를 디코딩할 수도 있다. 일부의 다른 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 슬라이스 헤더의 데이터로부터 컬러-공간 역변환 프로세스를 유도할 수도 있다. 또 다른 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 표시하는 보충 강화 정보 (SEI) 메시지의 데이터를 디코딩할 수도 있다.

[0307] 일부의 예들에서, 선택스 엘리먼트는 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되지 않았다는 것을 표시하는 제 1 선택스 엘리먼트이다. 이러한 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 2 선택스 엘리먼트의 값을 추가로 결정할 수도 있다. 선택스 엘리먼트는 제 1 모드 또는 제 2 모드가 코딩 유닛에서의 픽셀에 대한 팔레트 인덱스를 디코딩하기 위하여 이용되는지 여부를 표시할 수도 있다. 제 2 선택스 엘리먼트의 결정된 값에 기초하여, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 제 1 모드 또는 제 2 모드의 어느 하나를 이용하여 팔레트 인덱스를 디코딩할 수도 있다. 제 1 모드를 이용할 때, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 팔레트 인덱스의 값을 결정할 수도 있고, 현재 디코딩되고 있는 픽셀을 바로 뒤따르는 스캔 순서에서의 픽셀들의 수를 표시하는 제 3 선택스 엘리먼트의 값을 결정할 수도 있고, 스캔 순서에서의 다음 N 개의 픽셀들에 대한 팔레트 인덱스의 값을 결정하는 결과를 복제할 수도 있고, N 은 제 3 선택스 엘리먼트의 값과 동일하다. 일부의 예들에서, 제 1 모드는 실행 모드이다. 제 2 모드를 이용할 때, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 팔레트 인덱스의 값을 결정할 수도 있고 픽셀에 대한 픽셀 샘플 값을 출력할 수도 있고, 여기서, 픽셀 샘플 값은 팔레트 인덱스의 값과 동일하다. 일부의 예들에서, 제 2 모드는 픽셀 모드이다.

[0308] 일부의 예들에서, 선택스 엘리먼트를 디코딩하는 것은, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 이 코딩 유닛이 컬러-공간 변환을 이용하여 인코딩되었는지 여부를 표시하는 보충 강화 정보 (SEI) 메시지의 데이터를 디코딩하도록 구성되는 것을 포함한다. 일부의 예들에서, 컬러-공간 역변환 프로세스를 적용하는 것은, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 이 고정된 컬러-공간 역변환 행렬을 적용하도록 구성

되는 것을 포함한다. 일부의 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg 행렬이다. 다른 예들에서, 고정된 컬러-공간 변환 행렬은 YCoCg-R 행렬이다.

[0309] 일부의 예들에서, 신텍스 엘리먼트가 코딩 유닛이 컬러-공간 변환을 이용하여 코딩되었다는 것을 표시하는 것으로 결정하는 것에 응답하여, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 비디오 데이터에 적용하기 전에, 하나 이상의 컴포넌트들의 값들을 이용하여 잔차 블록을 생성할 수도 있다. 이러한 예들에서, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (70) 은 컬러-공간 역변환 프로세스를 비디오 데이터의 잔차 블록에 적용할 수도 있다.

[0310] 예에 따라서는, 본원에서 설명된 기법들 중의 임의의 것의 어떤 액트 (act) 들 또는 이벤트 (event) 들이 상이한 시퀀스에서 수행될 수 있거나, 추가될 수도 있거나, 병합될 수도 있거나, 또는 모두 배제 (예를 들어, 모든 설명된 액트들 또는 이벤트들이 기법들의 실시를 위해 필요한 것은 아님) 될 수도 있다는 것을 인식해야 한다. 또한, 어떤 예들에서는, 액트들 또는 이벤트들이 순차적인 것이 아니라, 예를 들어, 멀티-스레딩된 (multi-threaded) 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해 동시에 수행될 수도 있다.

[0311] 하나 이상의 예들에서는, 설명된 기능들이 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현될 경우, 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 컴퓨터-판독가능 매체를 통해 송신될 수도 있고, 하드웨어-기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터-판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터-판독가능 저장 매체들, 또는 예컨대, 통신 프로토콜에 따라 하나의 장소로부터 또 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비-일시적 유형의 컴퓨터-판독가능 저장 매체들, 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 이 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 리트리빙하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독가능 매체를 포함할 수도 있다.

[0312] 제한이 아닌 예로서, 이러한 컴퓨터-판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 회망하는 프로그램 코드를 저장하기 위해 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속은 컴퓨터-판독가능 매체로 적절하게 칭해진다. 예를 들어, 동축 케이블, 광섬유 케이블, 트위스트 페어 (twisted pair), 디지털 가입자 회선 (digital subscriber line; DSL), 또는 적외선, 라디오 (radio), 및 마이크로파 (microwave) 와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신될 경우, 동축 케이블, 광섬유 케이블, 트위스트 페어, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하는 것이 아니라, 그 대신에, 비-일시적인, 유형의 저장 매체들에 관한 것이라는 것을 이해해야 한다. 본원에서 이용된 바와 같은 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (compact disc; CD), 레이저 디스크 (laser disc), 광학 디스크 (optical disc), 디지털 다기능 디스크 (digital versatile disc; DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (Blu-ray disc) 를 포함하고, 여기서 디스크 (disk) 들은 통상 데이터를 자기적으로 재생하는 반면, 디스크 (disc) 들은 데이터를 레이저들로 광학적으로 재생한다. 상기의 조합들은 컴퓨터-판독가능 매체들의 범위 내에 또한 포함되어야 한다.

[0313] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적된 또는 별도의 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 이용된 바와 같은 용어 "프로세서" 는 상기한 구조, 또는 본원에서 설명된 기법들의 구현을 위해 적당한 임의의 다른 구조 중의 임의의 것을 지칭할 수도 있다. 게다가, 일부의 양태들에서는, 본원에서 설명된 기능성이 인코딩 및 디코딩을 위해 구성되거나 조합된 코덱 내에 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에서 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

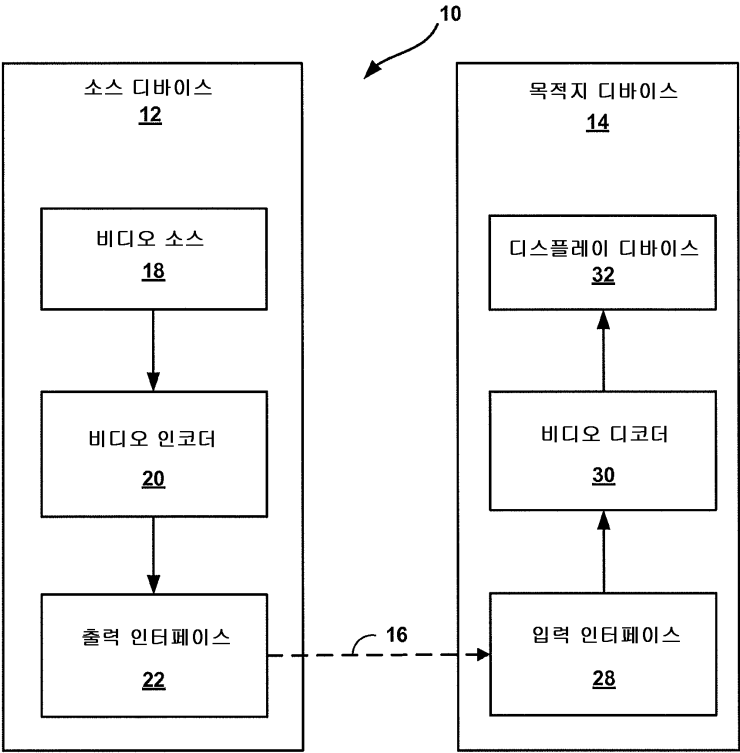
[0314] 이 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들어, 칩셋) 를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위하여 이 개시물에서 설명되어 있지만, 상이

한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛들은 코텍 하드웨어 유닛 내에 조합될 수도 있거나, 적당한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는 상호동작하는 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

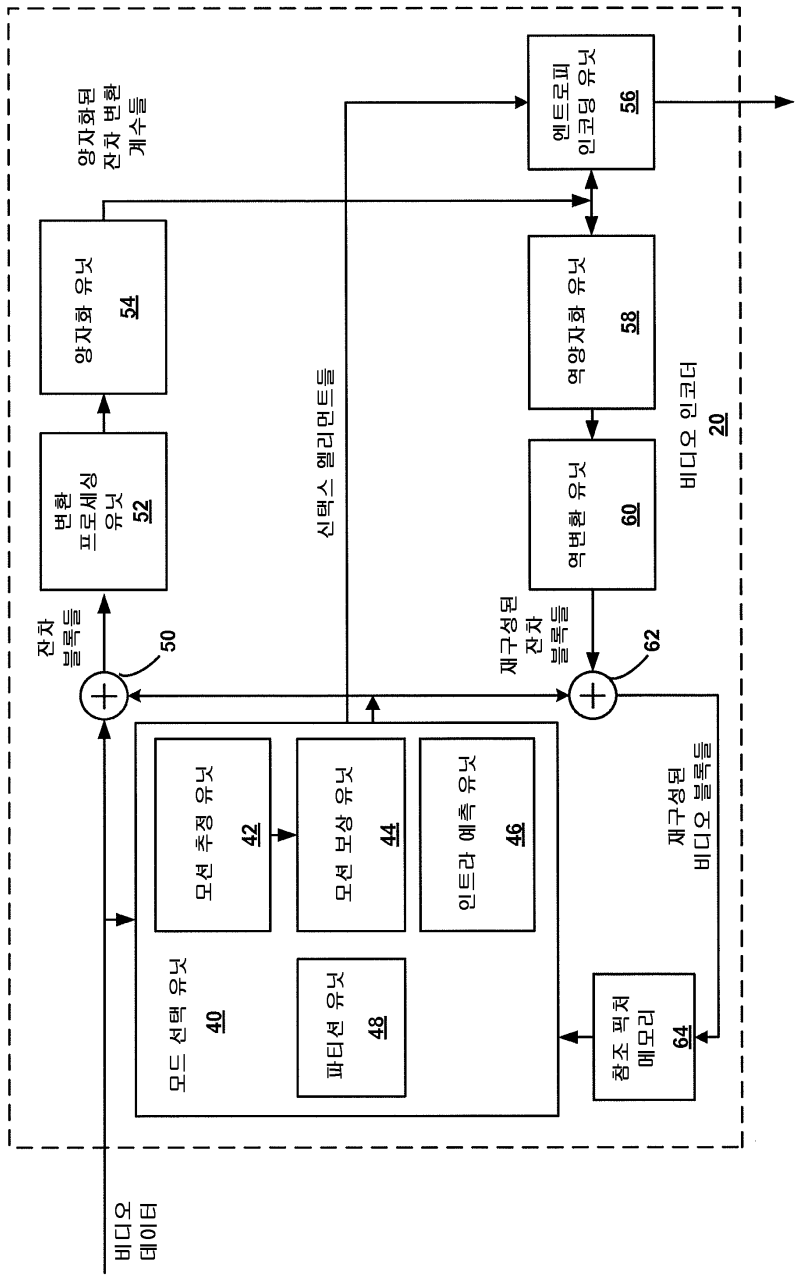
[0315] 개시물의 다양한 예들이 설명되었다. 설명된 시스템들, 동작들, 또는 기능들의 임의의 조합이 구상된다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

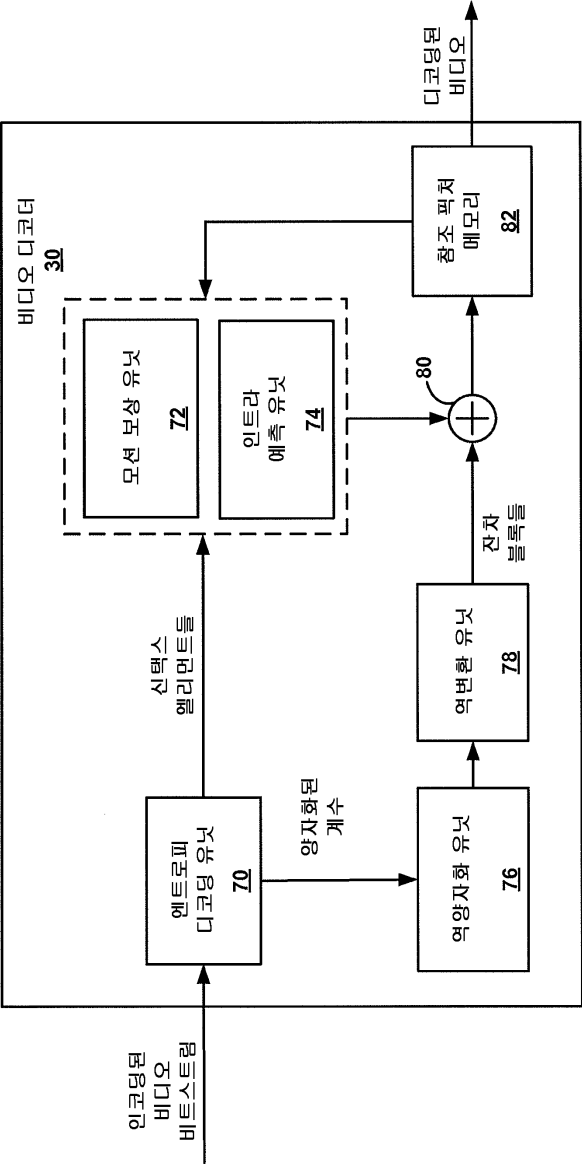
도면1



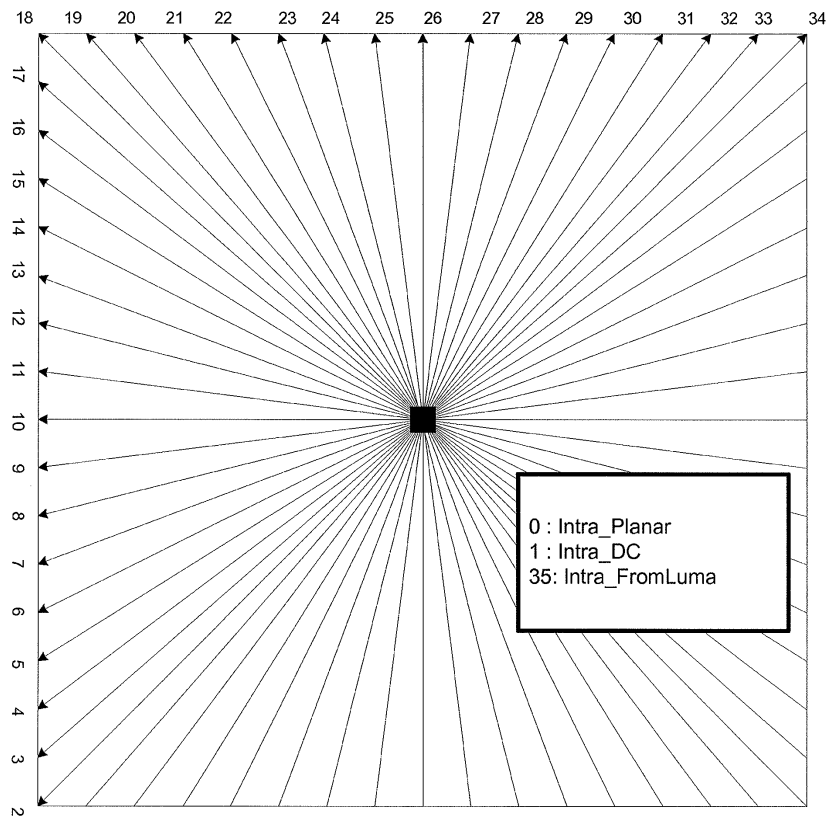
도면2



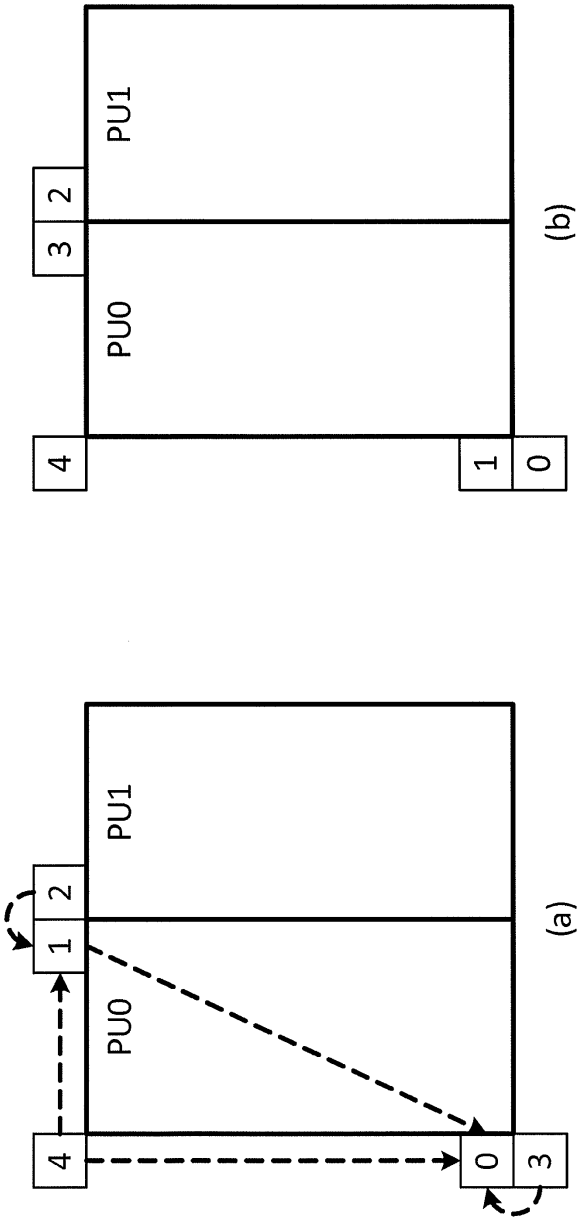
도면3



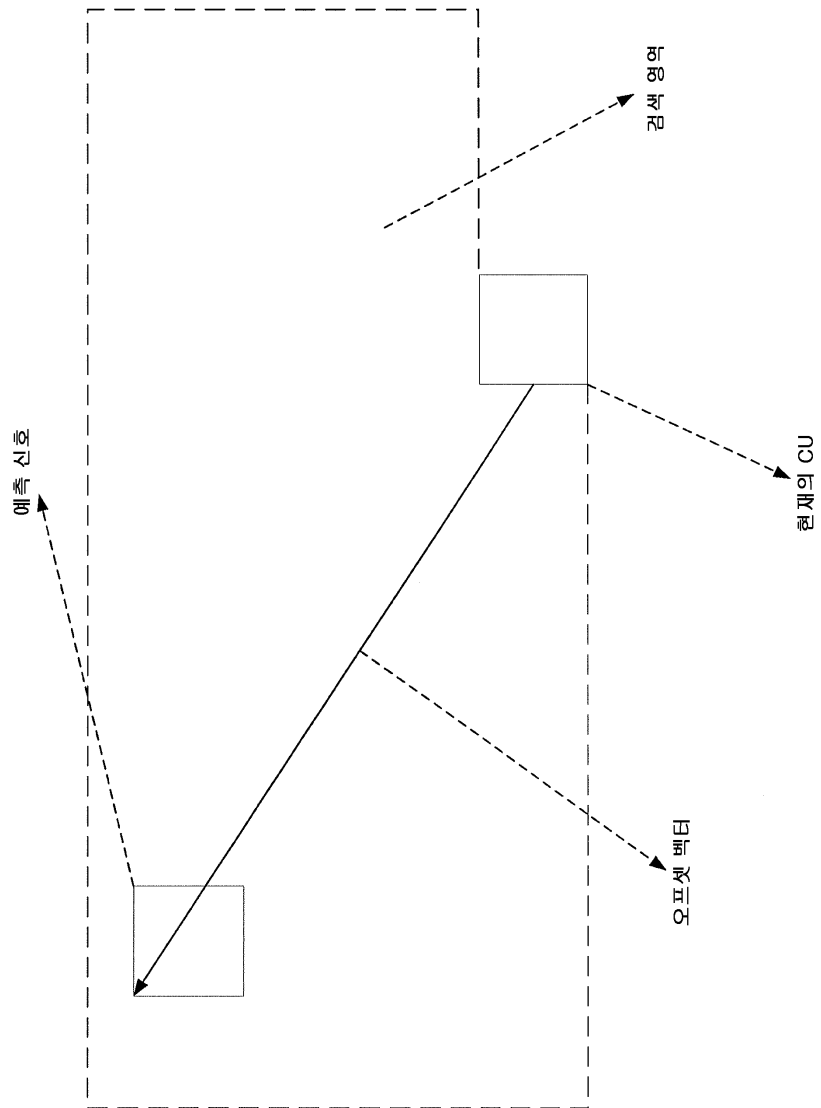
도면4



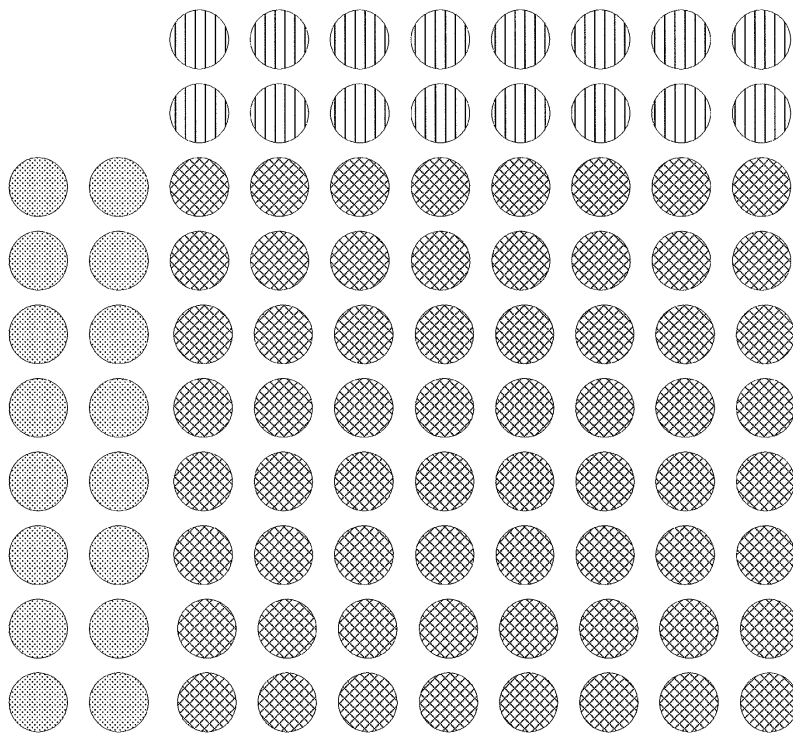
도면5



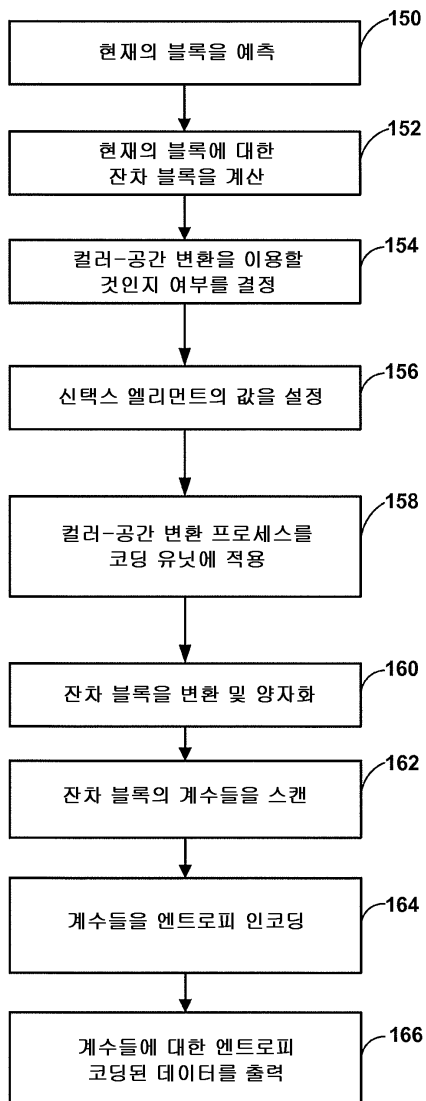
도면6



도면7



도면8



도면9

