



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년01월30일
(11) 등록번호 10-2493839
(24) 등록일자 2023년01월26일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/124 (2014.01) HO4N 19/12 (2014.01)
HO4N 19/176 (2014.01) HO4N 19/18 (2014.01)
HO4N 19/186 (2014.01) HO4N 19/44 (2014.01)
HO4N 19/60 (2014.01)
- (52) CPC특허분류
HO4N 19/124 (2015.01)
HO4N 19/12 (2015.01)
- (21) 출원번호 10-2017-7009327
- (22) 출원일자(국제) 2015년10월07일
심사청구일자 2020년09월21일
- (85) 번역문제출일자 2017년04월05일
- (65) 공개번호 10-2017-0066402
- (43) 공개일자 2017년06월14일
- (86) 국제출원번호 PCT/US2015/054488
- (87) 국제공개번호 WO 2016/057665
국제공개일자 2016년04월14일

- (73) 특허권자
켈컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
라파카 크리쉬나칸트
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- 장 리**
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
- (74) 대리인
특허법인코리아나

(30) 우선권주장
62/061,099 2014년10월07일 미국(US)
(뒷면에 계속)

(56) 선행기술조사문헌
KR1020140109933 A*
Li Zhang, et. al., "SCCE5 Test 3.2.1: In-loop color-space transform", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 18th Meeting: Sapporo, JP, 201*
*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 30 항

심사관 : 박상철

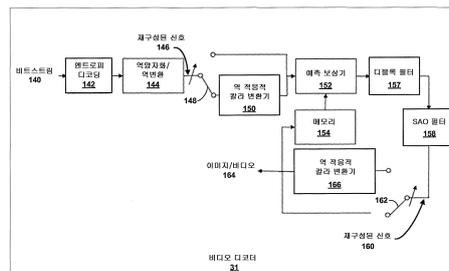
(54) 발명의 명칭 비디오 코딩에서의 적응적 컬러 변환을 위한 QP 도출 및 오프셋

(57) 요약

비디오 데이터를 디코딩하기 위한 디바이스는, 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정하고; 하나 이상의 블록들에 대해 양자화 파라미터를 결정하며; 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 양자화 파라미터를 결정하기 위해 상기 양자화 파라미터를 변경하고;

(뒷면에 계속)

대표도



및 변경된 양자화 파라미터에 기초하여 변환 계수들을 역양자화하도록 구성된다.

(52) CPC특허분류

HO4N 19/176 (2015.01)

HO4N 19/18 (2015.01)

HO4N 19/186 (2015.01)

HO4N 19/44 (2015.01)

HO4N 19/60 (2015.01)

(72) 발명자

조쉬 라잔 렉스맨

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장

62/064,347 2014년10월15일 미국(US)

14/876,574 2015년10월06일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터에서 오프셋 값을 수신하는 단계;

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하는 단계;

상기 하나 이상의 블록들에 대해 RGB 컬러 공간 양자화 파라미터를 결정하는 단계로서, 상기 RGB 컬러 공간 양자화 파라미터를 결정하는 단계는 YUV 컬러 공간 양자화 파라미터에 상기 오프셋 값을 가산하는 단계를 포함하는, 상기 RGB 컬러 공간 양자화 파라미터를 결정하는 단계;

상기 RGB 컬러 공간 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 RGB 컬러 공간 양자화 파라미터를 결정하기 위해 상기 RGB 컬러 공간 양자화 파라미터를 변경하는 단계; 및

변경된 상기 RGB 컬러 공간 양자화 파라미터에 기초하여 상기 하나 이상의 블록들로부터의 블록의 변환 계수들을 역양자화하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하는 단계는 상기 비디오 데이터에 대한 크로마 포맷을 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 2 항에 있어서,

4:4:4 의 크로마 포맷을 결정하는 것에 응답하여, 적응적 컬러 변환이 가능하게 된다고 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 2 항에 있어서,

4:4:4 이외의 크로마 포맷을 결정하는 것에 응답하여, 적응적 컬러 변환이 불가능하게 된다고 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하는 단계는 상기 비디오 데이터에서의 신택스 엘리먼트를 파싱하는 단계를 포함하고,

상기 신택스 엘리먼트는 적응적 컬러 변환이 가능하게 되는지 여부를 나타내는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 5 항에 있어서,

상기 신택스 엘리먼트는 화상 파라미터 세트 (PPS) 에서 시그널링되는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 임계값은 제로이고,

상기 변경된 양자화 파라미터의 값은 제로 이상인, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 변경된 RGB 컬러 공간 양자화 파라미터는 51 플러스 상기 오프셋 값 이하인, 비디오 데이터를 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

무선 통신 디바이스의 수신기에서 상기 비디오 데이터를 수신하는 단계;

상기 무선 통신 디바이스의 메모리에 상기 비디오 데이터를 저장하는 단계; 및

상기 무선 통신 디바이스의 하나 이상의 프로세서들상에서 상기 비디오 데이터를 프로세싱하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 10

제 1 항에 있어서,

무선 통신 디바이스는 전화기 핸드셋을 포함하고,

상기 무선 통신 디바이스의 수신기에서 상기 비디오 데이터를 수신하는 단계는, 무선 통신 표준에 따라, 상기 비디오 데이터를 포함하는 신호를 복조하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 11

제 1 항에 있어서,

상기 오프셋 값이 상기 양자화 파라미터에 가산되어야 한다는 것을 나타내는 플래그를 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 12

제 1 항에 있어서,

역양자화된 상기 변환 계수들에 기초하여, 상기 비디오 데이터에 대한 레지듀얼 블록을 결정하는 단계;

예측 블록을 결정하는 단계;

비디오 데이터의 재구성된 블록을 결정하기 위해 예측 블록에 상기 레지듀얼 블록을 가산하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 13

제 1 항에 있어서,

상기 비디오 데이터를 디코딩하는 방법은 비디오 인코딩 프로세스의 부분으로서 수행되는, 비디오 데이터를 디코딩하는 방법.

청구항 14

비디오 데이터를 디코딩하기 위한 디바이스로서,

비디오 데이터 메모리;

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터에서 오프셋 값을 수신하며

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하고;

상기 하나 이상의 블록들에 대해 RGB 컬러 공간 양자화 파라미터를 결정하는 것으로서, 상기 RGB 컬러 공간 양자화 파라미터를 결정하는 것은 YUV 컬러 공간 양자화 파라미터에 상기 오프셋 값을 가산하는 것을 포함하는, 상기 RGB 컬러 공간 양자화 파라미터를 결정하며,;

상기 RGB 컬러 공간 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 RGB 컬러 공간 양자화 파라미터를 결정하기 위해 상기 RGB 컬러 공간 양자화 파라미터를 변경하고; 및

변경된 상기 RGB 컬러 공간 양자화 파라미터에 기초하여 상기 하나 이상의 블록들로부터의 블록의 변환 계수들을 역양자화하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 15

제 14 항에 있어서,

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하기 위해, 상기 하나 이상의 프로세서들은 또한 상기 비디오 데이터에 대한 크로마 포맷을 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 16

제 15 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

4:4:4 의 크로마 포맷을 결정하는 것에 응답하여, 적응적 컬러 변환이 가능하게 된다고 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 17

제 15 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

4:4:4 이외의 크로마 포맷을 결정하는 것에 응답하여, 적응적 컬러 변환이 불가능하게 된다고 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 18

제 14 항에 있어서,

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하기 위해, 상기 하나 이상의 프로세서들은 또한 상기 비디오 데이터에서의 선택스 엘리먼트를 파악하도록 구성되고,

상기 선택스 엘리먼트는 적응적 컬러 변환이 가능하게 되는지 여부를 나타내는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 19

제 18 항에 있어서,

상기 선택스 엘리먼트는 화상 파라미터 세트 (PPS) 에서 시그널링되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 20

제 14 항에 있어서,

상기 임계값은 제로이고,

상기 변경된 양자화 파라미터의 값은 제로 이상인, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 21

제 14 항에 있어서,

상기 변경된 RGB 컬러 공간 양자화 파라미터는 51 플러스 상기 오프셋 값 이하인, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 22

제 14 항에 있어서,

상기 디바이스는 무선 통신 디바이스를 포함하고,

상기 디바이스는 상기 비디오 데이터를 수신하도록 구성된 수신기를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 23

제 22 항에 있어서,

상기 무선 통신 디바이스는 전화기 핸드셋을 포함하고,

상기 수신기는 무선 통신 표준에 따라, 상기 비디오 데이터를 포함하는 신호를 복조하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 24

제 22 항에 있어서,

상기 하나 이상의 프로세서들은 또한 상기 오프셋 값이 상기 양자화 파라미터에 가산되어야 한다는 것을 나타내는 플래그를 수신하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 25

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

역양자화된 상기 변환 계수들에 기초하여, 상기 비디오 데이터에 대한 레지듀얼 블록을 결정하고;

예측 블록을 결정하며;

비디오 데이터의 재구성된 블록을 결정하기 위해 예측 블록에 상기 레지듀얼 블록을 가산하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 26

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 또한 상기 비디오 데이터를 디코딩하도록 구성되고,

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

청구항 27

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 또한 상기 비디오 데이터를 인코딩하도록 구성되고,

상기 디바이스는 상기 비디오 데이터를 획득하도록 구성된 카메라를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 28

제 14 항에 있어서,

상기 디바이스는,

집적회로;

마이크로프로세서; 또는

무선 통신 디바이스

중 적어도 하나를 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 29

비디오 데이터를 디코딩하기 위한 장치로서,

상기 비디오 데이터에서 오프셋 값을 수신하는 수단;

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하는 수단;

상기 하나 이상의 블록들에 대해 RGB 컬러 공간 양자화 파라미터를 결정하는 수단으로서, 상기 RGB 컬러 공간 양자화 파라미터를 결정하는 것은 YUV 컬러 공간 양자화 파라미터에 상기 오프셋 값을 가산하는 것을 포함하는, 상기 RGB 컬러 공간 양자화 파라미터를 결정하는 수단;

상기 RGB 컬러 공간 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 RGB 컬러 공간 양자화 파라미터를 결정하기 위해 상기 RGB 컬러 공간 양자화 파라미터를 변경하는 수단; 및

변경된 상기 RGB 컬러 공간 양자화 파라미터에 기초하여 상기 하나 이상의 블록들로부터의 블록의 변환 계수들을 역양자화하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 30

명령들을 저장하는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 때, 상기 하나 이상의 프로세서들로 하여금:

비디오 데이터에서 오프셋 값을 수신하게 하고;

상기 비디오 데이터의 하나 이상의 블록들에 대해, 적응적 컬러 변환이 가능하게 된다고 결정하게 하고;

상기 하나 이상의 블록들에 대해 RGB 컬러 공간 양자화 파라미터를 결정하게 하는 것으로서, 상기 RGB 컬러 공간 양자화 파라미터를 결정하는 것은 YUV 컬러 공간 양자화 파라미터에 상기 오프셋 값을 가산하는 것을 포함하는, 상기 RGB 컬러 공간 양자화 파라미터를 결정하게 하며;

상기 RGB 컬러 공간 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 RGB 컬러 공간 양자화 파라미터를 결정하기 위해 상기 RGB 컬러 공간 양자화 파라미터를 변경하게 하며; 및

변경된 상기 RGB 컬러 공간 양자화 파라미터에 기초하여 상기 하나 이상의 블록들로부터의 블록의 변환 계수들을 역양자화하게 하는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

- [0001] 본 출원은
- [0002] 2014년 10월 7일자로 출원된 미국 임시 특허 출원 제62/061,099호, 및
- [0003] 2014년 10월 15일자로 출원된 미국 임시 특허 출원 제62/064,347호
- [0004] 의 이익을 주장하며, 본 명세서에 그것의 전체 내용이 전체로 포함된다.
- [0005] 본 개시는 비디오 인코딩 또는 비디오 디코딩과 같은 비디오 코딩에 관한 것이다.

배경 기술

[0006] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말기들 (personal digital assistants, PDA들), 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (Advanced Video Coding, AVC) 에 의해 규정된 표준들, 현재 개발중인 고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장본들에 기재된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현하는 것에 의해 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0007] 비디오 압축 기법들은 공간적 (인트라-화상) 예측 및/또는 시간적 (인터-화상) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분) 가 비디오 블록들로 파티셔닝될 수도 있으며, 그 비디오 블록들은 또한 트리블록들 (treeblocks), 코딩 유닛들 (coding units, CU들) 및/또는 코딩 노드들이라고 지칭될 수도 있다. 화상의 인트라 코딩된 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측을 사용하여 인코딩된다. 화상의 인터 코딩된 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 화상들에서의 참조 샘플들에 관한 시간적 예측을 사용할 수도 있다. 화상들은 프레임들이라고 지칭될 수도 있고, 참조 화상들은 참조 프레임들이라고 지칭될 수도 있다.

[0008] 공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록 (predictive block) 을 야기한다. 레지듀얼 데이터는 코딩될 원래의 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터 코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터, 및 코딩된 블록과 예측 블록 사이의 차이를 나타내는 레지듀얼 데이터에 따라 인코딩된다. 인트라 코딩된 블록은 인트라 코딩 모드 및 레지듀얼 데이터에 따라 인코딩된다. 추가 압축을 위해, 레지듀얼 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 결과적으로 레지듀얼 변환 계수들을 야기하며, 그 계수들은 그 다음에 양자화될 수도 있다. 처음에는 2차원 어레이로 배열된 양자화된 변환 계수들은 변환 계수들의 1차원 벡터를 생성하기 위하여 스캐닝될 수도 있고, 엔트로피 코딩이 훨씬 더 많은 압축을 달성하기 위해 적용될 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0009] 본 개시는 컬러-공간 변환 코딩이 사용되는 경우의 양자화 파라미터들을 결정하는 것에 관련된 기법들을 기술하고, 또한, 본 개시는 비디오 데이터의 인코딩된 비트스트림에서, 컬러-공간 변환 코딩이 사용되는 경우 양자화 파라미터들을 시그널링하기 위해 사용되는 여러 신택스 엘리먼트들을 생성 및 파싱하는 기법들을 기술한다.

[0010] 하나의 예에서, 비디오 데이터를 디코딩하는 방법은 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정하는 단계; 하나 이상의 블록들에 대해 양자화 파라미터를 결정하는 단계; 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 양자화 파라미터를 결정하기 위해 양자화 파라미터를 변경하는 단계; 및 변경된 양자화 파라미터에 기초하여 변환 계수들을 역양자화하는 단계를 포함한다.

[0011] 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는 비디오 데이터 메모리 및 하나 이상의 프로세서들을 포함하고, 그 하나 이상의 프로세서들은 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정하고; 하나 이상의 블록들에 대해 양자화 파라미터를 결정하며; 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 양자화 파라미터를 결정하기 위해 양자화 파라미터를 변경하고; 및 변경된 양자화 파라미터에 기초하여 변환 계수들을 역양자화하도록 구성된다.

[0012] 다른 예에서, 비디오 데이터를 디코딩하기 위한 장치는 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정하는 수단; 하나 이상의 블록들에 대해 양자화 파라미터를 결정하는 수단; 양자

화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 양자화 파라미터를 결정하기 위해 양자화 파라미터를 변경하는 수단; 및 변경된 양자화 파라미터에 기초하여 변환 계수들을 역양자화하는 수단을 포함한다.

[0013] 다른 예에서, 명령들을 저장하는 컴퓨터 관독가능 저장 매체로서, 상기 명령들은 하나 이상의 프로세서들에 의해 실행될 때 하나 이상의 프로세서들로 하여금: 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정하게 하고; 하나 이상의 블록들에 대해 양자화 파라미터를 결정하게 하며; 양자화 파라미터의 값이 임계값 아래에 있는 것에 응답하여, 변경된 양자화 파라미터를 결정하기 위해 양자화 파라미터를 변경하게 하며; 및 변경된 양자화 파라미터에 기초하여 변환 계수들을 역양자화하게 한다.

[0014] 본 개시의 하나 이상의 예들의 상세들이 첨부 도면들 및 다음의 설명에서 진술된다. 본 개시의 다른 특징들, 목적들, 및 장점들은 상세한 설명, 도면들, 및 청구범위로부터 명확하게 될 것이다.

도면의 간단한 설명

[0015] 도 1 은 본 개시에서 기술된 기법들을 이용할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템을 도시하는 블록도이다.

도 2a 내지 도 2c 는 비디오 데이터에 대한 상이한 샘플 포맷들을 도시하는 개념도들이다.

도 3 은 4:2:0 샘플 포맷에 따라 포맷된 16x16 코딩 유닛을 도시하는 개념도이다.

도 4 은 4:2:2 샘플 포맷에 따라 포맷된 16x16 코딩 유닛을 도시하는 개념도이다.

도 5 는 본 개시의 하나 이상의 기법들에 따른, 인트라 8x8 블록에 대한 타겟 블록 및 참조 샘플의 예를 도시하는 개념도이다.

도 6 은 본 개시에 기술된 기법들을 구현할 수도 있는 예시의 비디오 인코더를 도시하는 블록도이다.

도 7 은 본 개시에 기술된 기법들을 구현할 수도 있는 예시의 비디오 디코더를 도시하는 블록도이다.

도 8 은 본 개시의 하나 이상의 양태들에 따른 컬러 변환을 사용하여 RGB 컬러 공간을 갖는 비디오 데이터를 제 2 컬러 공간을 갖는 비디오 데이터로 변환하는 기법들을 이용할 수도 있는 비디오 인코더의 다른 예를 도시하는 블록도이다.

도 9 는 본 개시의 하나 이상의 양태들에 따른 역 컬러 변환을 사용하여 제 1 컬러 공간을 갖는 비디오 데이터의 블록을 제 2 의, RGB 컬러 공간을 갖는 비디오 데이터의 블록으로 역으로 변환하는 기법들을 이용할 수도 있는 비디오 디코더의 다른 예를 도시하는 블록도이다.

도 10 은 본 개시의 기법들에 따른 예시의 비디오 디코딩 방법을 도시하는 플로우차트이다.

도 11 은 본 개시의 기법들에 따른 예시의 비디오 인코딩 방법을 도시하는 플로우차트이다.

도 12 는 본 개시의 기법들에 따른 예시의 비디오 디코딩 방법을 도시하는 플로우차트이다.

도 13 은 본 개시의 기법들에 따른 예시의 비디오 인코딩 방법을 도시하는 플로우차트이다.

도 14 는 본 개시의 기법들에 따른 예시의 비디오 디코딩 방법을 도시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 개시는 적응적 컬러 변환 양자화 파라미터 도출들에 관련된 기법들을 기술한다. 본 개시는 적응적 컬러 변환이 사용되는 경우의 양자화 파라미터 도출에 관련된 여러 문제들을 식별하고 이들 문제들을 다루는 해결책들을 제안한다. 본 개시는 최근에 완결된 고효율 비디오 코딩 (HEVC) 표준의 떠오르는 스크린 콘텐츠 코딩 (SCC) 확장들 및 범위 확장들 (RExt) 에 관련된 기법들을 포함하는 비디오 코딩 기법들을 기술한다. SCC 및 범위 확장들은 4:4:4, 4:2:2, 4:2:0, 4:0:0 등과 같은 상이한 크로마 샘플링 포맷들 및/또는 높은 비트 깊이 (예를 들어, 8 비트 초과) 를 잠재적으로 지원하도록 설계되고 있고, 따라서 베이스 HEVC 표준에 포함되지 않은 새로운 코딩 툴들을 포함하도록 설계되고 있다.

[0017] 하나의 그러한 코딩 툴은 컬러 공간 변환 코딩이다. 컬러 공간 변환 코딩에서, 비디오 인코더는 더 양호한 코딩 품질 (예를 들어, 더 양호한 레이트-왜곡 트레이드오프) 을 달성하기 위해 제 1 컬러 공간 (예를 들어, YCbCr) 으로부터 제 2 컬러 공간 (예를 들어, RGB) 으로 레지듀얼 데이터를 변환할 수도 있다. 레지듀얼 데이터의 컬러 공간에 관계없이, 비디오 인코더는 통상 레지듀얼 데이터를 변환 계수들로 변환하고 그 변환 계수

들을 양자화한다. 비디오 디코더는 변환 계수들을 역양자화하고, 레지듀얼 데이터를 재구성하기 위해 변환 계수들을 역변환하는 역의 (reciprocal) 프로세스들을 수행한다. 비디오 인코더는, 비디오 데이터의 인코딩된 비트스트림에의 포함을 위해, 변환 계수 레벨들을 양자화함에 있어서 사용되는 스케일링의 양을 나타내는 양자화 파라미터를 생성한다. 비디오 디코더는 비디오 인코더에 의해 사용된 양자화 파라미터를 결정하기 위해 비트스트림을 파싱한다. 양자화 파라미터는 또한 디블록 필터링과 같은 다른 비디오 코딩 프로세스들에 의해 사용될 수도 있다.

[0018] 본 개시는 컬러 공간 변환 코딩이 사용되는 경우의 양자화 파라미터들을 결정하는 것에 관련된 기법들을 기술하고, 또한, 본 개시는 인코더로부터 디코더로 비디오 데이터의 인코딩된 비트스트림의 부분으로서, 컬러 공간 변환 코딩이 사용되는 경우의 양자화 파라미터들을 시그널링하기 위한 기법들을 기술한다.

[0019] 도 1 은 IBC 모드에서 블록들을 코딩하기 위한 기법들 및 병렬 프로세싱을 위한 기법들을 포함하는, 본 개시에서 기술된 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 도시하는 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 이후의 시간에 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화 핸드셋들, 이른바 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한 매우 다양한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 무선 통신을 위해 구비될 수도 있다.

[0020] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 링크 (16) 를 통해 수신할 수도 있다. 링크 (16) 는 소스 디바이스 (12) 로부터의 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 링크 (16) 는 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접 실시간으로 송신하는 것을 가능하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 통신 표준, 이를테면 무선 통신 프로토콜에 따라 변조되고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 이를테면 무선 주파수 (radio frequency, RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 이를테면 국부 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0021] 대안적으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (17) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (17) 로부터 액세스될 수도 있다. 저장 디바이스 (17) 는 하드 드라이브, 블루 레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산형 또는 국부적으로 액세스되는 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다.

[0022] 추가의 예에서, 저장 디바이스 (17) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 유지할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 저장된 비디오 데이터를 저장 디바이스 (17) 로부터 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, 네트워크 부속 스토리지 (network attached storage, NAS) 디바이스들, 또는 국부 디스크 드라이브를 포함한다.

[0023] 목적지 디바이스 (14) 는 인터넷 접속을 포함한 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 (예컨대, Wi-Fi 와 같은 무선 로컬 영역 네트워크 표준 또는 LTE 또는 다른 셀룰러 통신 표준과 같은 무선 전기통신 표준을 포함하는 무선 표준에 따라 변조되는) 무선 채널, 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 양쪽 모두의 조합을 포함할 수도 있다. 저장 디바이스 (17) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.

[0024] 본 개시의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 반드시 제한되지는 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비

전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 중 임의의 것의 지원 하의 비디오 코딩, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0025] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22) 를 포함한다. 일부 경우들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오를 담고 있는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 피드 인터페이스, 및/또는 컴퓨터 그래픽 데이터를 소스 비디오로서 생성하는 컴퓨터 그래픽 시스템과 같은 소스, 또는 그런 소스들의 조합을 포함할 수도 있다. 하나의 예로서, 비디오 소스 (18) 가 비디오 카메라인 경우, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 이른바 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시에서 설명되는 기법들은 일반적으로 비디오 코딩에 적용 가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0026] 캡처된, 사전 캡처된 (pre-captured), 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안으로) 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 디코딩 및/또는 플레이백을 위한 나중의 액세스를 위해 저장 디바이스 (17) 에 저장될 수도 있다.

[0027] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 구비한다. 일부 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16) 를 통해 통신되거나 또는 저장 디바이스 (17) 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩함에 있어서, 비디오 디코더, 이를테면 비디오 디코더 (30) 에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 신택스 엘리먼트들을 포함할 수도 있다. 이러한 신택스 엘리먼트들은 통신 매체 상에서 송신되는, 저장 매체 상에 저장되는, 또는 파일 서버에 저장되는 인코딩된 비디오 데이터에 포함될 수도 있다.

[0028] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나, 또는 목적지 디바이스 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는 통합형 디스플레이 디바이스를 포함할 수도 있고 또한 외부 디스플레이 디바이스와 인터페이싱하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (liquid crystal display, LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode, OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0029] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC 와 같은 비디오 압축 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HM) 에 따를 수도 있다. "HEVC 워킹 드래프트 10" 또는 "HEVC WD 10" 으로서 지칭되는, HEVC 표준의 워킹 드래프트는 Bross et al., "Editors' proposed corrections to HEVC version 1," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 13th Meeting, Incheon, KR, April 2013 에 기술되어 있다. 다른 HEVC 드래프트 사양은 http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-01003-v2.zip 으로부터 이용가능하다. 본 개시에 기술된 기법들은 또한 현재 개발 중에 있는 HEVC 표준의 확장들에 따라 동작할 수도 있다.

[0030] 대안적으로 또는 추가적으로, 비디오 인코더 (20) 와 비디오 디코더 (30) 는, 다르게는 MPEG-4, 파트 10, 고급 비디오 코딩 (AVC) 이라고 지칭되는 ITU-T H.264 표준과 같은 다른 독점 또는 업계 표준들, 또는 그런 표준들의 확장본들에 따라 동작할 수도 있다. 그러나, 본 개시의 기법들은 임의의 특정 코딩 표준으로 제한되지 않는다. 비디오 압축 표준들의 다른 예들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 그리고 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장본들을 포함한 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC 로도 알려짐) 를 포함한다.

[0031] HEVC 의 설계는 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group, VCEG) 및 ISO/IEC 동 화상 전문가 그룹 (Motion Picture Experts Group, MPEG) 의 JCT-VC 에 의해 최근에 완결되었다. HEVC REExt 로서 지

칭되는, HEVC 에 대한 범위 확장들이 또한 JCT-VC 에 의해 개발되고 있다. 이하에서 REExt WD7 으로서 지칭되는, 범위 확장들의 최근의 워킹 드래프트 (WD) 는 http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1005-v4.zip 로부터 이용가능하다.

[0032] 이 개시는 일반적으로 HEVC 버전 1 또는 베이스 HEVC 로서 최근에 완결된 HEVC 사양 텍스트를 참조할 것이다. 범위 확장 사양은 HEVC 의 버전 2 가 될 수도 있다. 모션 벡터 예측과 같은 많은 코딩 툴들에 대해, HEVC 버전 1 및 범위 확장 사양은 기술적으로 유사하다. 따라서, 본 개시가 HEVC 버전 1 에 대한 변경들을 기술할 때마다, 동일한 변경들이 또한 일부 추가적인 코딩 툴들에 대하여, 베이스 HEVC 사양을 일반적으로 포함하는 범위 확장 사양에 적용될 수도 있다. 또한, 일반적으로 HEVC 버전 1 모듈들이 또한 HEVC 범위 확장을 구현하는 디코더에 통합될 수도 있다는 것이 가정될 수 있다.

[0033] 모션을 갖는 텍스트 및 그래픽과 같은 스크린-콘텐츠 자료를 위한 새로운 코딩 툴들이 현재 개발 중에 있고, HEVC 의 미래의 버전을 포함하는 미래의 비디오 코딩 표준들에의 포함을 위해 고려되고 있다. 이들 새로운 코딩 툴들은 스크린 콘텐츠에 대한 코딩 효율을 잠재적으로 향상시킨다. 코딩 효율에서의 상당한 향상들이 신규한 전용 코딩 툴들을 갖는 스크린 콘텐츠의 특징들을 활용함으로써 획득될 수도 있다는 증거가 존재하기 때문에, 제안 요청 (Call for Proposals: CfP) 이 가능하게는 SCC 를 위한 특정의 툴들을 포함하는 HEVC 표준의 미래의 확장들을 개발하는 것을 목표로 발생되었다. 회사들 및 조직들은 이러한 요청에 응답하여 제안들을 제출하도록 초대되었다. 이러한 CfP 의 사용 케이스들 및 요건들이 MPEG 문서 N14174 에 기술되어 있다. 17th JCT-VC 회의 동안, SCC 테스트 모델 (SCM) 이 확립된다. 최근의 SCC 워킹 드래프트 (WD) 는 JCTVC-U1005 이고 http://phenix.int-evry.fr/jct/doc_end_user/documents/21_Warsaw/wg11/JCTVC-U1005-v1.zip 에서 이용가능하다.

[0034] 이 개시는 소스 디바이스 (12) 의 비디오 인코더 (20) 가 이들 현재 또는 미래의 표준들 중 임의의 것에 따라 비디오 데이터를 인코딩하도록 구성될 수도 있다는 것을 고려한다. 유사하게, 이 개시는 또한 목적지 디바이스 (14) 의 비디오 디코더 (30) 가 이들 현재 또는 미래의 표준들 중 임의의 것에 따라 비디오 데이터를 디코딩하도록 구성될 수도 있다는 것을 고려한다.

[0035] 도 1 에 도시되지는 않았지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통의 데이터 스트림 또는 별개의 데이터 스트림들에서 오디오 및 비디오 양자 모두의 인코딩을 다루기 위해 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜들에 따를 수도 있다.

[0036] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 반도체들 (ASICs), 필드 프로그램가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들과 같은 임의의 다양한 적합한 인코더 회로로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어에서 구현되는 경우, 디바이스는 적합한, 비일시적 컴퓨터 판독가능 매체에 그 소프트웨어에 대한 명령들을 저장하고 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 그들 중 어느 것은 각각의 디바이스에서 결합된 인코더/디코더 (코덱) 의 부분으로서 통합될 수도 있다.

[0037] 위에서 도입된 바와 같이, JCT-VC 는 HEVC 표준의 개발을 최근에 완결했다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HEVC Test Model, HM) 이라고 지칭되는 비디오 코딩 디바이스의 진화 모델에 기초하였다. HM은, 예컨대, ITU-T H.264/AVC에 따른 현존 디바이스들에 비해 비디오 코딩 디바이스들의 몇몇 추가적인 능력들을 상정한다. 예를 들어, H.264가 아홉 개의 인트라 예측 인코딩 모드들을 제공하는 반면, HM은 서른다섯 개 정도의 인트라 예측 인코딩 모드들을 제공할 수도 있다.

[0038] HEVC 및 다른 비디오 코딩 사양들에서, 비디오 시퀀스는 일련의 화상들을 통상 포함한다. 화상들은 "프레임들"이라고 또한 지칭될 수도 있다. 화상이 S_L , S_{Cb} 및 S_{Cr} 로 표시되는 세 개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 색차 샘플들의 2차원 어레이이다. S_{Cr} 은 Cr 색차 샘플들의 2차원 어레이이다. 색차 샘플들은 본원에서 "크로마" 샘플들이라고 또한 지칭될 수도 있다. 다른 예들에서, 화상이 모노크롬일 수도 있고 루마 샘플들의 어레이만을 포함할 수도 있다.

[0039] 화상의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛들 (coding tree units, CTU) 의

세트를 생성할 수도 있다. 각각의 CTU 는 루마 샘플들의 코딩 트리 블록 (coding tree block), 크로마 샘플들의 두 개의 대응 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들을 포함할 수도 있다. 모노크롬 화상들 또는 세 개의 별개의 컬러 평면들을 갖는 화상들에서, CTU 는 단일 코딩 트리 블록과 그 코딩 트리 블록의 샘플들을 코딩하는데 사용된 선택스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 "트리 블록" 또는 "최대 코딩 유닛 (largest coding unit, LCU)"이라고 또한 지칭될 수도 있다. HEVC의 CTU들은 다른 표준들, 이를테면 H.264/AVC의 매크로블록들과 대체로 유사할 수도 있다. 그러나, CTU가 특정 사이즈로 반드시 제한되는 것은 아니고 하나 이상의 코딩 유닛들 (CU들) 을 포함할 수도 있다. 슬라이스는 래스터 스캔 순서로 연속적으로 순서화된 정수 개수의 CTU 들을 포함할 수도 있다.

[0040] 코딩된 CTU 를 생성하기 위해, 비디오 인코더 (20) 는 CTU의 코딩 트리 블록들에 대해 퀴드트리 파티셔닝을 재귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들로 분할할 수도 있으며, 그래서 그 이름이 "코딩 트리 유닛들"이다. 코딩 블록은 샘플들의 NxN 블록일 수도 있다. CU 는 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 화상의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 두 개의 대응하는 코딩 블록들과, 그 코딩 블록들의 샘플들을 코딩하는데 사용된 선택스 구조들을 포함할 수도 있다. 모노크롬 화상들 또는 세 개의 별개의 컬러 평면들을 갖는 화상들에서, CU 는 단일 코딩 블록 및 그 코딩 블록의 샘플들을 코딩하는데 사용되는 선택스 구조들을 포함할 수도 있다.

[0041] 비디오 인코더 (20) 는 CU의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비정사각형) 블록이다. CU의 예측 유닛 (PU) 은 루마 샘플들의 예측 블록, 크로마 샘플들의 대응하는 두 개의 예측 블록들, 및 예측 블록들을 예측하는데 사용된 선택스 구조들을 포함할 수도 있다. 모노크롬 화상들 또는 세 개의 별개의 컬러 평면들을 갖는 화상들에서, PU 는 단일 예측 블록과 그 예측 블록을 예측하는데 사용된 선택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb, 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.

[0042] 비디오 인코더 (20) 는 PU 에 대한 예측 블록들을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인트라 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 화상의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인터 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 화상 이외의 하나 이상의 화상들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0043] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU 들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성한 후, 비디오 인코더 (20) 는 CU 에 대한 루마 레지듀얼 블록을 생성할 수도 있다. CU 의 루마 레지듀얼 블록에서의 각 샘플은 CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 원래의 루마 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낸다. 또, 비디오 인코더 (20) 는 CU 에 대한 Cb 레지듀얼 블록을 생성할 수도 있다. CU 의 Cb 레지듀얼 블록에서의 각 샘플은 CU 의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 원래의 Cb 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한 CU 에 대한 Cr 레지듀얼 블록을 생성할 수도 있다. CU 의 Cr 레지듀얼 블록에서의 각 샘플은 CU 의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 원래의 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다.

[0044] 더욱이, 비디오 인코더 (20) 는 CU 의 루마, Cb, 및 Cr 레지듀얼 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해하기 위해 퀴드-트리 파티셔닝을 사용할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형 (예를 들어, 정사각형 또는 비정사각형) 블록이다. CU의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 두 개의 대응하는 변환 블록들, 및 그 변환 블록 샘플들을 변환하는데 사용된 선택스 구조들을 포함할 수도 있다. 따라서, CU의 각각의 TU는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 루마 변환 블록은 CU의 루마 레지듀얼 블록의 서브블록일 수도 있다. Cb 변환 블록은 CU의 Cb 레지듀얼 블록의 서브블록일 수도 있다. Cr 변환 블록은 CU의 Cr 레지듀얼 블록의 서브블록일 수도 있다. 모노크롬 화상들 또는 세 개의 별개의 컬러 평면들을 갖는 화상들에서, TU 는 단일 변환 블록과 그 변환 블록의 샘플들을 변환하는데 사용된 선택스 구조들을 포함할 수도 있다.

[0045] 비디오 인코더 (20) 는 TU 에 대한 루마 계수 블록을 생성하기 위해 TU 의 루마 변환 블록에 하나 이상의 변환들을 적용할 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라 양

일 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cb 계수 블록을 생성하기 위해 TU 의 Cb 변환 블록에 하나 이상의 변환들을 적용할 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cr 계수 블록을 생성하기 위해 TU 의 Cr 변환 블록에 하나 이상의 변환들을 적용할 수도 있다.

[0046] 계수 블록 (예를 들어, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후, 비디오 인코더 (20) 는 그 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로 변환 계수들이 가능하게는 변환 계수들을 표현하는데 사용되는 데이터의 양을 감소시키기 위해 양자화되어, 추가의 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들에 대해 컨텍스트-적응형 이진 산술 코딩 (CABAC) 을 수행할 수도 있다.

[0047] 비디오 인코더 (20) 는 코딩된 화상들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 그 비트스트림은 NAL 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은 그 NAL 유닛 내의 데이터의 타입의 표시를 포함하는 신택스 구조 및 에플레이션 방지 비트들이 필요에 따라 산재되어 있는 RBSP 의 형태로 그 데이터를 포함하는 바이트들이다. NAL 유닛들 각각은 NAL 유닛 헤더를 포함하고 RBSP 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 타입 코드를 나타내는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 타입 코드는 NAL 유닛의 타입을 나타낸다. RBSP 는 NAL 유닛 내에 캡슐화되는 정수 개수의 바이트들을 포함하는 신택스 구조일 수도 있다. 일부 예들에서, RBSP 는 제로 비트들을 포함한다.

[0048] 상이한 타입들의 NAL 유닛들은 상이한 타입들의 RBSP 들을 캡슐화할 수도 있다. 예를 들어, 제 1 타입의 NAL 유닛은 PPS 에 대한 RBSP 를 캡슐화할 수도 있고, 제 2 타입의 NAL 유닛은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있고, 제 3 타입의 NAL 유닛은 SEI 메시지들에 대한 RBSP 를 캡슐화할 수도 있는 등이다. (파라미터 세트들 및 SEI 메시지들에 대한 RBSP 들과 대조되는) 비디오 코딩 데이터에 대한 RBSP 들을 캡슐화하는 NAL 유닛들은 VCL NAL 유닛들로서 지칭될 수도 있다.

[0049] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 또, 비디오 디코더 (30) 는 그 비트스트림으로부터 신택스 엘리먼트들을 획득하기 위해 그 비트스트림을 파싱할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 화상들을 재구성할 수도 있다. 비디오 데이터를 재구성하는 프로세스는 일반적으로 비디오 인코더 (20) 에 의해 수행된 프로세스에 역일 수도 있다. 또, 비디오 디코더 (30) 는 현재의 CU 의 TU 들과 연관된 계수 블록들을 역 양자화할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 TU 들과 연관된 변환 블록들을 재구성하기 위해 계수 블록들에 대해 역변환들을 수행할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 TU 들의 변환 블록들의 대응하는 샘플들에 현재의 CU 의 PU 들에 대한 예측 블록들의 샘플들을 가산함으로써 현재의 CU 의 코딩 블록들을 재구성할 수도 있다. 화상의 각각의 CU 에 대한 코딩 블록들을 재구성함으로써, 비디오 디코더 (30) 는 화상을 재구성할 수도 있다.

[0050] 크로마 포맷으로서도 지칭될 수도 있는 비디오 샘플링 포맷은 CU 내에 포함된 루마 샘플들의 수에 대한 CU 내에 포함된 크로마 샘플들의 수를 정의할 수도 있다. 크로마 컴포넌트들에 대한 비디오 샘플링 포맷에 의존하여, U 및 V 컴포넌트들의, 샘플들의 수에 의한 사이즈가 Y 컴포넌트의 사이즈와 동일하거나 상이할 수도 있다. HEVC 표준에서, chroma_format_idc 로 불리는 값은 루마 컴포넌트에 대해, 크로마 컴포넌트들의 상이한 샘플링 포맷들을 나타내기 위해 정의된다. HEVC 에서, chroma_format_idc 는 SPS 에서 시그널링된다. 표 1 은 chroma_format_idc 의 값들과 연관된 크로마 포맷들 사이의 관계를 도시한다.

표 1

[0051]

chroma_format_idc	chroma format	SubWidthC	SubHeightC
0	모노크롬	-	-
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

[0052] 표 1: HEVC 에서 정의된 상이한 크로마 포맷들

[0053] 표 1 에서, 변수들 SubWidthC 및 SubHeightC 는 루마 컴포넌트에 대한 샘플들의 수와 각각의 크로마 컴포넌트에

대한 샘플들의 수 사이의 수평 및 수직 샘플링 레이트 비를 나타내기 위해 사용될 수 있다. 표 1 에서 기술된 크로마 포맷들에서, 2 개의 크로마 컴포넌트들은 동일한 샘플링 레이트를 갖는다. 따라서, 4:2:0 샘플링에서, 2 개의 크로마 어레이들 각각은 루마 어레이의 높이의 반 및 폭의 반을 갖는 반면, 4:2:2 샘플링에서는, 2 개의 크로마 어레이들 각각은 루마 어레이의 동일한 높이 및 폭의 반을 갖는다. 4:4:4 샘플링에서, 2 개의 크로마 어레이들 각각은 루마 어레이와 동일한 높이 및 폭을 가질 수도 있거나, 일부 예들에서는, 3 개의 컬러 평면들은 모두 모노크롬 샘플링된 화상들로서 별개로 프로세싱될 수도 있다.

[0054] 표 1 의 예에서, 4:2:0 포맷의 경우, 루마 컴포넌트에 대한 샘플링 레이트는 수평 및 수직 방향들 양자 모두에 대해 크로마 컴포넌트들의 샘플링 레이트의 두 배이다. 결과적으로, 4:2:0 포맷에 따라 포맷된 코딩 유닛의 경우, 루마 컴포넌트에 대한 샘플들의 어레이의 폭 및 높이는 크로마 컴포넌트들에 대한 샘플들의 각 어레이의 두 배이다. 유사하게, 4:2:2 포맷에 따라 포맷된 코딩 유닛의 경우, 루마 컴포넌트에 대한 샘플들의 어레이의 폭은 각각의 크로마 컴포넌트에 대한 샘플들의 어레이의 폭의 두 배이지만, 루마 컴포넌트에 대한 샘플들의 어레이의 높이는 각각의 크로마 컴포넌트에 대한 샘플들의 어레이의 높이와 동일하다. 4:4:4 포맷에 따라 포맷된 코딩 유닛의 경우, 루마 컴포넌트에 대한 샘플들의 어레이는 각각의 크로마 컴포넌트에 대한 샘플들의 어레이와 동일한 폭 및 높이를 갖는다. YUV 컬러 공간에 더하여, 비디오 데이터는 RGB 컬러 공간에 따라 정의될 수 있다는 것을 유의해야 한다. 이러한 방식으로, 여기에 기술된 크로마 포맷들은 YUV 또는 RGB 컬러 공간에 적용될 수도 있다. RGB 크로마 포맷들은 통상 적색 샘플들의 수, 녹색 샘플들의 수 및 청색 샘플들의 수가 동일하도록 샘플링된다. 따라서, 여기에 사용된 바와 같은 용어 "4:4:4 크로마 포맷" 은 샘플들의 수가 모든 컬러 컴포넌트들에 대해 동일한 YUV 컬러 공간 또는 RGB 컬러 공간을 지칭할 수도 있다.

[0055] 도 2a 내지 도 2c 는 비디오 데이터를 위한 상이한 샘플 포맷들을 예시하는 개념도들이다. 도 2a 는 4:2:0 샘플 포맷을 예시하는 개념도이다. 도 2a 에 예시된 바처럼, 4:2:0 샘플 포맷에 대하여, 크로마 성분들은 루마 성분의 크기의 1/4 이다. 따라서, 4:2:0 샘플 포맷에 따라 포맷된 CU 에 대하여, 크로마 성분의 샘플마다 4개의 루마 샘플들이 있다. 도 2b 는 4:2:2 샘플 포맷을 예시하는 개념도이다. 도 2b 에 예시된 바처럼, 4:2:2 샘플 포맷에 대하여, 크로마 성분들은 루마 성분의 크기의 1/2 이다. 따라서, 4:2:2 샘플 포맷에 따라 포맷된 CU 에 대하여, 크로마 성분의 샘플마다 2개의 루마 샘플들이 있다. 도 2c 는 4:4:4 샘플 포맷을 예시하는 개념도이다. 도 2c 에 예시된 바처럼, 4:4:4 샘플 포맷에 대하여, 크로마 성분들은 루마 성분의 크기와 동일하다. 따라서, 4:4:4 샘플 포맷에 따라 포맷된 CU 에 대하여, 크로마 성분의 샘플마다 1 개의 루마 샘플이 있다.

[0056] 도 3은 4:2:0 샘플 포맷에 따라 포맷된 16x16 코딩 유닛의 예를 예시하는 개념도이다. 도 3은 CU 내의 루마 샘플들에 대한 크로마 샘플들의 상대적인 위치를 예시한다. 위에 설명된 바처럼, CU 는 통상적으로, 수평 및 수직 루마 샘플들의 수에 따라 정의된다. 따라서, 도 3에 예시된 바처럼, 4:2:0 샘플 포맷에 따라 포맷된 16x16 CU 는 루마 성분들의 16x16 샘플들 및 각각의 크로마 성분을 위한 8x8 샘플들을 포함한다. 또한, 위에 설명된 바처럼, CU 는 더 작은 CU들로 파티셔닝될 수도 있다. 예를 들어, 도 3에 예시된 CU 는 4개의 8x8 CU 들로 파티셔닝될 수도 있고, 여기서 각각의 8x8 CU 는 루마 성분을 위한 8x8 샘플들 및 각각의 크로마 성분을 위한 4x4 샘플들을 포함한다.

[0057] 도 4는 4:2:2 샘플 포맷에 따라 포맷된 16x16 코딩 유닛의 예를 예시하는 개념도이다. 도 4는 CU 내의 루마 샘플들에 대한 크로마 샘플들의 상대적인 위치를 예시한다. 위에 설명된 바처럼, CU 는 통상적으로, 수평 및 수직 루마 샘플들의 수에 따라 정의된다. 따라서, 도 4에 예시된 바처럼, 4:2:2 샘플 포맷에 따라 포맷된 16x16 CU 는 루마 성분들의 16x16 샘플들 및 각각의 크로마 성분을 위한 8x16 샘플들을 포함한다. 또한, 위에 설명된 바처럼, CU 는 더 작은 CU들로 파티셔닝될 수도 있다. 예를 들어, 도 4에 예시된 CU 는 4개의 8x8 CU 들로 파티셔닝될 수도 있고, 여기서 각각의 CU 는 루마 성분을 위한 8x8 샘플들 및 각각의 크로마 성분을 위한 4x8 샘플들을 포함한다.

[0058] 본 개시에 기술된 기법들에 따르면, 레지듀얼 신호들 (즉, 레지듀얼 블록들) 에 대한 인-루프 컬러 공간 변환은 4:4:4 크로마 포맷에서의 시퀀스들에 대해 제안된다; 그러나, 그 기법들은 4:4:4 포맷에 제한되지 않는다. 인-루프 컬러 공간 변환 프로세스는 RGB/YUV 크로마 포맷에서의 예측 에러 신호들 (즉, 레지듀얼 신호들) 을 서브-최적의 컬러 공간에서의 그것들로 변환한다. 인-루프 컬러 공간 변환은 또한 컬러 컴포넌트들 사이의 상관을 감소시킬 수 있다. 변환 매트릭스는 특이값 분해 (SVD) 에 의해 각 CU 에 대한 화소 샘플 값들로부터 도출될 수도 있다. 컬러 공간 변환은 인트라 모드 및 인터 모드 양자 모두의 예측 에러에 적용될 수도 있다.

[0059] 컬러 공간 변환이 인터 모드에 적용되는 경우, 레지듀얼은 먼저 도출된 변환 매트릭스로 상이한 도메인으로 변환된다. 컬러 공간 변환 후, DCT/DST, 양자화, 및 엔트로피 코딩과 같은 코딩 단계들이 순서대로 수행된다.

[0060] 컬러 공간 변환이 인트라 모드를 사용하여 코딩된 CU 에 적용되는 경우, 예측 및 현재의 블록은 먼저 각각 도출된 변환 매트릭스로 상이한 도메인으로 변환된다. 컬러 공간 변환 후에, 현재의 블록과 그 현재의 블록에 대한 예측기 사이의 레지듀얼은 또한 DCT/DST 로 변환되고, 양자화되며, 엔트로피 코딩된다.

[0061] 비디오 인코더 (20) 와 같은 비디오 인코딩 디바이스는 순방향 동작을 수행하며, 여기서 변환 값들 a, b, c, d, e, f, g, h, 및 i 를 포함하는 컬러 공간 변환 매트릭스는 다음과 같이 컬러 컴포넌트들 P, Q, 및 S 에 대한 값들을 도출하기 위해 3 개의 평면들 G, B, 및 R 에 적용된다:

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

[0062]

[0063] 결과의 값들은 값들이 최악의 경우에 $\sqrt{3}$ 배까지 확대될 수도 있기 때문에 HEVC 사양의 범위 내에 클립핑될 수도 있다. 비디오 디코더 (30) 과 같은 비디오 디코딩 디바이스는 역 동작을 수행하며, 여기서 변환 값들 a', b', c', d', e', f', g', h', 및 i' 를 포함하는 컬러 공간 변환 매트릭스는 다음과 같이 3 개의 평면들 G', B', 및 R' 을 도출하기 위해 3개의 컬러 컴포넌트들 P', Q', 및 S' 에 적용된다:

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

[0064]

[0065] 도 5 는 본 개시의 하나 이상의 기법들에 따른, 인트라 8x8 블록에 대한 타겟 블록 및 참조 샘플의 예를 도시하는 개념도이다. 변환 매트릭스는 참조 샘플 값들로부터 특이값 분해 (SVD) 를 사용하여 도출될 수도 있다.

비디오 코딩 디바이스 (예를 들어, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 인트라 케이스 및 인터 케이스에 대해 상이한 참조 샘플들을 사용할 수도 있다. 인트라 코딩된 블록의 케이스의 경우, 타겟 블록들 및 참조 샘플들은 도 5 에 도시된 바일 수도 있다. 도 5 에서, 타겟 블록은 8x8 크로스해치된 샘플들 (94) 로 이루어지고, 상측 참조 샘플들 (96) 은 스트라이프된 것으로 도시되며, 좌측 참조 샘플들 (98) 은 점으로 도시된다.

[0066] 인터 코딩된 블록의 케이스의 경우, 매트릭스 도출에 대한 참조 샘플들은 모션 보상을 위한 참조 샘플들과 동일할 수도 있다. 진보된 모션 예측 (AMP) 블록에서의 참조 샘플들은 참조 샘플들의 수가 감소되도록 서브 샘플링될 수도 있다. 예를 들어, 12x16 블록에서의 참조 샘플들의 수는 2/3 만큼 감소된다.

[0067] 상기 예들 중 일부에서, 컬러 공간 변환 프로세스는 항상 적용될 수도 있다. 따라서, 컬러 공간 변환 프로세스가 호출되는지 여부를 시그널링할 필요가 없을 수도 있다. 또, 비디오 인코더 (20) 및 비디오 디코더 (30) 양자 모두는 변환 매트릭스를 시그널링하기 위한 오버헤드를 피하기 위해 변환 매트릭스를 도출하는 동일한 방법을 사용할 수도 있다.

[0068] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 여러 컬러 공간 변환 매트릭스들을 사용할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 상이한 컬러 공간들에 대해 상이한 컬러 공간 변환 매트릭스들을 적용할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 RGB 컬러 공간으로부터 YCbCr 컬러 공간으로 및 본래 상태로 샘플 값들을 변환하기 위해 한 쌍의 YCbCr 변환 매트릭스들을 사용할 수도 있다. 다음의 식들은 YCbCr 변환 매트릭스들의 하나의 예시의 세트를 보여준다:

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

[0069] 순방향:

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

[0070] 역방향 :

[0071] 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 RGB 컬러 공간으로부터 YCoCg 컬러 공간으로 및 본래 상태로 샘플 값들을 변환하기 위해 한 쌍의 YCoCg 변환 매트릭스들을 사용할 수도 있다. 다음의 식들은 YCoCg 변환 매트릭스들의 하나의 예시의 세트를 보여준다:

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

[0072] 순방향:

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

[0073] 역방향:

[0074] 다른 그러한 매트릭스는 2 의 팩터로 Co 및 Cg 컴포넌트들을 스케일링하는 YCoCg 매트릭스의 개정가능 버전인 YCoCg-R 매트릭스일 수도 있다. 리프팅 기법을 사용함으로써, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다음의 식들에 의해 순방향 및 역방향 변환을 달성할 수도 있다:

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

[0075] 순방향:

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

[0076] 역방향:

[0077] 상기 식들 및 매트릭스들에서, 순방향 변환들은 (예를 들어, 비디오 인코더에 의해) 인코딩 프로세스 전에 수행될 수도 있다. 역으로, 역방향 변환들은 (예를 들어, 비디오 디코더에 의해) 디코딩 프로세스 후에 수행될 수도 있다. 비디오 인코더 (20) 는 다른 비디오 데이터를 예측하는데 사용하기 위해 인코딩된 날짜를 재구성하기 위해 디코딩 루프를 포함한다는 것을 유지해야 한다. 이에 따라, 비디오 디코더 (30) 와 같이, 비디오 인코더 (20) 의 디코딩 루프는 또한 역 변환들을 수행할 수도 있다.

[0078] 본 개시의 기법들은 잠재적으로 하나 이상의 문제들 및 더 구체적으로는, cu_residual_act_flag 가 인에이블되는 경우의 QP 도출에 대한 잠재적인 문제들을 다룬다. 예를 들어, 기존의 솔루션들에 따르면, 적응적 컬러 변환이 가능하게 되는 경우, 스케일링 및 변환 프로세스 동안, -5 의 QP 오프셋이 루마 및 Cb 크로마 컴포넌트에 대해 가산되고, -3 이 Cr 크로마 컴포넌트에 대해 가산된다. 그러나, Qp 의 결과적인 값은 허용된 Qp 범위를 언더플로우 (underflow) 할 수도 있다. 예를 들어, 현재의 테스트 모델에서, 결과적인 Qp 는 HEVC 에 의해 허용된 범위가 0 과 51 사이에 있을 때 -5 까지 언더플로우할 수도 있는 것이 가능하다. 본 개시는 또한 적응적 컬러 변환이 가능하게 되는 경우 적응적 QP 오프셋들을 시그널링하기 위한 기법들을 기술한다.

[0079] 스케일링 및 변환 프로세스의 일부가 이하에 진술된다.

[0080] 8.6.2 스케일링 및 변환 프로세스

[0081] -----

[0082] 양자화 파라미터 qp 는 현재의 테스트 모델에서 다음과 같이 도출된다,

[0083] cIdx 가 0 과 동일한 경우,

[0084] $qp = Qp'_Y + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0)$ (8-261)

[0085] - 그렇지 않은 경우, cIdx 가 1 과 동일한 경우,

[0086] $qp = Qp'_{Cb} + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0)$ (8-262)

[0087] - 그렇지 않은 경우 (cIdx 가 2 와 동일),

[0088] $qP = Qp'_{cr} + (cu_residual_act_flag[xTbY][yTbY] ? -3 : 0)$ (8-263)

[0089] 여기서, cIdx 는 현재의 블록의 컬러 컴포넌트를 특정하고, cu_residual_act_flag 는 적응적 컬러 변환이 현재의 코딩 유닛의 레지듀얼 샘플들에 적용되는지 여부를 특정한다.

[0090] 본 개시는 위에서 소개된 문제들을 다룰 수도 있는 여러 기법들을 기술한다. 다음의 기법들 각각은 다른 것들 중 하나 이상과 별개로 또는 결합하여 구현될 수도 있다. 본 개시의 하나의 기법에 따르면, 비디오 디코더 (30) 는 섹션 8.6.2 식 8-261, 8-262, 8-263 - 0, $51 + QpBdOffsetY$ 인 HEVC Qp 범위에 대한 (적응적 컬러 변환이 가능하게 된 경우에 오프셋이 가산된 후) 스케일링 및 변환 프로세스로부터 결과적인 Qp 들을 클립핑할 수도 있다. 본 개시의 다른 기법에 따르면, 비디오 인코더 (20) 는 적응적 컬러 변환이 가능하게 된 경우 섹션 8.6.2 (스케일링 및 변환 프로세스) 에서 적용될 Qp 오프셋을 비디오 디코더 (30) 로 시그널링할 수도 있다. Qp 오프셋의 이러한 시그널링은 VPS, SPS, PPS, 슬라이스 헤드 또는 그것의 확장과 같은 여러 입도 레벨들에서 행해질 수도 있다. Qp 오프셋은 모든 컴포넌트들 (루마 + 크로마) 또는 컴포넌트들의 단지 일부 (예를 들어, 크로마) 에 대해 시그널링될 수도 있다.

[0091] 본 개시의 다른 기법에 따르면, 비디오 인코더 (20) 는 적응적 컬러 변환이 가능하게 되는 경우 QP 오프셋이 섹션 8.6.2 (스케일링 및 변환 프로세스) 에서 적용되어야 하는지 여부를 나타내는 플래그를 비디오 디코더 (30) 로 시그널링할 수도 있다. 플래그의 이러한 시그널링은 VPS, SPS, PPS, 슬라이스 헤드 또는 그것의 확장과 같은 여러 입도 레벨들에서 행해질 수 있다. 플래그의 시그널링은 모든 컴포넌트들 (루마 + 크로마) 또는 컴포넌트들의 단지 일부 (예를 들어, 크로마) 에 대해 시그널링될 수 있다.

[0092] 위에서 소개된 기법들의 예시의 구현들이 이제 더욱 상세히 기술될 것이다. 본 개시의 하나의 기법에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC Qp 의 범위 내까지 Qp 들을 클립핑하도록 구성될 수 있다. 적응적 컬러 변환이 사용되는 경우 HEVC 에서 사용되는 것과 동일한 것으로 허용된 Qp 범위를 유지하기 위해, 본 개시는 HEVC Qp 범위의 그것으로 Qp 값들의 범위를 클립핑하는 기법들을 기술한다. 테스트 모델에 대한 제안된 변경들은 이하에서 이탤릭체로 되어 있다.

[0093] 8.6.2 스케일링 및 변환 프로세스

[0094] -----

[0095] - cIdx 가 0 과 동일한 경우,

[0096] $qP = Clip3(0, 51 + QpBdOffsetY,$

[0097] $Qp'Y + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0))$

[0098] ChromaArrayType 이 0 과 동일하지 않은 경우,

[0099] - cIdx 가 1 과 동일한 경우,

[0100] $qP = Clip3(0, 51 +$

[0101] $QpBdOffsetC, Qp'_{cb} + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0))$ (8-262)

[0102] - 그렇지 않은 경우 (cIdx 가 2 와 동일한 경우),

[0103] $qP = Clip3(0, 51 + QpBdOffsetC, Qp'_{cr}$

[0104] $+ (cu_residual_act_flag[xTbY][yTbY] ? -3 : 0)$ (8-262)

[0105] ***Flexible signalling of QP offset for adaptive color transform***

표 2

[0106] pps_scc_extension () {	디스크립터
...	
if(residual_adaptive_colour_transform_enabled_flag){	
pps_slice_act_qp_offsets_present_flag	
pps_act_y_qp_offset	se(v)

<i>pps_act_cb_qp_offset</i>	
<i>pps_act_cr_qp_offset</i>	
}	
...	
}	

표 3

[0107]

slice_segment_header() {	디스크립터
first_slice_segment_in_pic_flag	u(1)
.....	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
}	
if(chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(pps_slice_act_qp_offsets_present_flag){	
slice_act_y_qp_offset	u(1)
slice_act_cb_qp_offset	
slice_act_cr_qp_offset	
}	
.....	

[0108] 루마 및 크로마 컴포넌트에 대한 QP 값들의 범위를 클리핑하는 것이 제안된다.

[0109] cIdx 가 0 과 동일한 경우,

[0110] $qP = Clip3(0, 51 + QpBdOffset_y,$

[0111] $Qp'Y + (cu_residual_act_flag[xTbY][yTbY] ? pps_act_y_qp_offset +$

[0112] $slice_act_y_qp_offset : 0))$

[0113] ChromaArrayType 이 0 과 동일하지 않은 경우,

[0114] cIdx 가 1 과 동일한 경우,

[0115] $qP = Clip3(0, 51 +$

[0116] $QpBdOffset_c, Qp'_{cb} + (cu_residual_act_flag[xTbY][yTbY] ? pps_act_$

[0117] $pps_cb_qp_offset + slice_act_cb_qp_offset : 0))$ (8-262)

[0118] 그렇지 않은 경우 (cIdx 가 2 와 동일함),

[0119] $qP = Clip3(0, 51 + QpBdOffset_c, Qp'_{cr}$

[0120] $+ (cu_residual_act_flag[xTbY][yTbY] ? pps_act_pps_cr_qp_offset +$

[0121] $slice_act_cr_qp_offset : 0)$ (8-263)

[0122] **pps_act_y_qp_offset**, **pps_act_cb_qp_offset** 및 **pps_act_cr_qp_offset** 은 각각 섹션 8.6.2 에서 도출된 루마, cb 및 cr 양자화 파라미터 qP 에 대한 오프셋들을 특정한다. **pps_act_y_qp_offset**, **pps_cb_qp_offset** 및 **pps_cr_qp_offset** 은 -12 내지 + 12 의 범위에 있을 것이다. ChromaArrayType 이 0 과 동일한 경우, **pps_act_cb_qp_offset** 및 **pps_act_cr_qp_offset** 은 디코딩 프로세스에서 사용되지 않고, 디코더들은 그들의 값을 무시할 것이다.

[0123] 1 과 동일한 **pps_slice_act_qp_offsets_present_flag** 는 slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset 가 슬라이스 헤더 내에 존재한다는 것을 특징한다. 0 과 동일한 pps_slice_act_qp_offsets_present_flag 는 slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset 가 슬라이스 헤더 내에 존재하지 않는다는 것을 특징한다. 존재하지 않는 경우, cu_chroma_qp_offset_enabled_flag 의 값은 0 과 동일한 것으로 추론된다.

[0124] **slice_act_y_qp_offset**, **slice_cb_qp_offset** 및 **slice_cr_qp_offset** 은 각각 섹션 8.6.2 에서 도출된 루마, cb 및 cr 양자화 파라미터 qP 에 대한 오프셋들을 특징한다. slice_act_y_qp_offset, slice_cb_qp_offset 및 slice_cr_qp_offset 의 값들은 -12 내지 +12 의 범위에 있을 것이다. ChromaArrayType 이 0 과 동일한 경우, slice_act_cb_qp_offset 및 slice_act_cr_qp_offset 은 디코딩 프로세스에서 사용되지 않고, 디코더들은 그들의 값을 무시할 것이다.

[0125] 적응적 컬러 변환에 대한 QP 오프셋의 존재를 시그널링하는 기법들이 이제 기술될 것이다. 위의 기법 (1) 에서 논의된 바와 같이, 적응적 컬러 변환이 가능하게 되는 경우의 고정된 네거티브 QP 오프셋은 더 높은 Qp 들에서 Qp 범위를 좁힌다. 예를 들어, 적응적 컬러 변환이 가능하게 되는 경우의 현재의 정의에 의해, 일부 시나리오들에서 타겟 비트레이트를 충족하기 위해 필요한 $46 + QpBdOffset_Y$ 를 통해 QP 들에 도달하는 것은 불가능하다. 아래의 솔루션에서, Qp 오프셋이 가산될지 여부를 나타내기 위해 플래그를 시그널링하는 것이 제안된다. 테스트 모델에 대한 제안된 변경들은 노랑색 텍스트로 강조된다.

표 4

[0126]	slice_segment_header() {	디스크립터
	first_slice_segment_in_pic_flag	u(1)
	
	slice_qp_delta	se(v)
	if(pps_slice_chroma_qp_offsets_present_flag) {	
	slice_cb_qp_offset	se(v)
	slice_cr_qp_offset	se(v)
	}	
	if(chroma_qp_offset_list_enabled_flag)	
	cu_chroma_qp_offset_enabled_flag	u(1)
	if(residual_adaptive_colour_transform_enabled_flag)	
	slice_act_qp_offset_present_flag	u(1)
	

[0127] 1 과 동일한 **slice_act_qp_offset_present_flag** 는 1 과 동일한 cu_residual_act_flag 와 함께 코딩을 위해 적용된다는 것을 특징한다. 0 과 동일한 slice_act_qp_offset_present_flag 는 1 과 동일한 cu_residual_act_flag 와 함께 코딩을 위해 적용되지 않는다는 것을 특징한다. 존재하지 않는 경우, cu_chroma_qp_offset_enabled_flag 의 값은 0 과 동일한 것으로 추론된다.

[0128] cIdx 가 0 과 동일한 경우,

[0129] $qP = Clip3(0, 51 + QpBdOffset_Y,$

[0130] $Qp'Y + (cu_residual_act_flag[xTbY][yTbY] \&\& slice_act_qp_offset_present$
 [0131] $_flag? -5 : 0))$

[0132] ChromaArrayType 이 0 과 동일하지 않는 경우,

[0133] cIdx 가 1 과 동일한 경우,

[0134] $qP = Clip3(0, 51 +$

[0135] $QpBdOffset_c, Qp'_{cb} + (cu_residual_act_flag[xTbY][yTbY] \&\& slice_act_qp_$

[0136] $offset_present_flag? -5 : 0))$ (8-262)

[0137] 그렇지 않은 경우 (cIdx 가 2 와 동일),
 [0138] $qP = Clip3(0, 51 + QpBdOffset_c ,$
 [0139] $Qp'_{cr} + (cu_residual_act_flag[xTbY][yTbY] \&\& slice_act_qp_offset$
 [0140] $_present_flag? -3 : 0)$ (8-263)

[0141] 적응적 컬러 변환을 위한 QP 오프셋의 다른 예시의 구현이 이제 기술될 것이다. 이러한 개시는 다음을 제안한다:

[0142] a) 시퀀스 파라미터 세트 대신에 화상 파라미터 세트에서 적응적 컬러 변환 인에이블드 플래그를 시그널링. 이것은 잠재적으로 적응적 컬러 변환의 사용을 화상 레벨에서 적용시킬 수 있는 것으로부터 이익이 된다.

[0143] b) 비트스트림 제한이 크로마 포맷이 4:4:4 가 아닌 경우 적응적 컬러 변환을 불가능하게 하기 위해 제안된다. 하나의 예에서, 이러한 제한은 적응적 컬러 변환 인에이블 플래그 (residual_adaptive_colour_transform_enabled_flag) 에 대해 적용되도록 제안된다.

[0144] 예시의 신택스 및 시맨틱스가 아래에 상세화된다.

표 5

[0145]	pps_scc_extension () {	디스크립터
	...	
	residual_adaptive_colour_transform_enabled_flag	u(1)
	if(residual_adaptive_colour_transform_enabled_flag){	
	pps_slice_act_qp_offsets_present_flag	
	pps_act_y_qp_offset	se(v)
	pps_act_cb_qp_offset	
	pps_act_cr_qp_offset	
	}	
	...	
	}	

표 6

[0146]	slice_segment_header () {	디스크립터
	first_slice_segment_in_pic_flag	u(1)
	
	slice_qp_delta	se(v)
	if(pps_slice_chroma_qp_offsets_present_flag) {	
	slice_cb_qp_offset	se(v)
	slice_cr_qp_offset	se(v)
	}	
	if(chroma_qp_offset_list_enabled_flag)	
	cu_chroma_qp_offset_enabled_flag	u(1)
	if(pps_slice_act_qp_offsets_present_flag){	
	slice_act_y_qp_offset	se(v)
	slice_act_cb_qp_offset	se(v)
	slice_act_cr_qp_offset	se(v)
	}	
	

[0147] 1 과 동일한 residual_adaptive_colour_transform_enabled_flag 는 적응적 컬러 변환이 PPS 를 참조하는 화상들에 대한 디코딩 프로세스에서 레지듀얼에 적용될 수도 있다는 것을 특정한다. 0 과 동일한

residual_adaptive_colour_transform_enabled_flag 는 적응적 컬러 변환이 PPS 를 참조하는 화상들에 대한 레지듀얼에 적용되지 않는다는 것을 특징한다. 존재하지 않는 경우, residual_adaptive_colour_transform_enabled_flag 의 값은 0 과 동일한 것으로 추론된다.

- [0148] chroma_format_idc 가 3 과 동일하지 않는 경우,
- [0149] residual_adaptive_colour_transform_enabled_flag 는 0 과 동일할 것이다.
- [0150] pps_act_y_qp_offset, pps_act_cb_qp_offset 및 pps_act_cr_qp_offset 는 각각 섹션 8.6.2 에서 도출된 루마, cb 및 cr 양자화 파라미터 qp 에 대한 오프셋들을 특징한다. pps_act_y_qp_offset, pps_cb_qp_offset 및 pps_cr_qp_offset 의 값들은 -12 내지 +12 의 범위에 있을 것이다. ChromaArrayType 이 0 과 동일한 경우, pps_act_cb_qp_offset 및 pps_act_cr_qp_offset 는 디코딩 프로세스에서 사용되지 않고, 디코더들은 그들의 값을 무시할 것이다.
- [0151] 1 과 동일한 pps_slice_act_qp_offsets_present_flag 는 slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset 이 슬라이스 헤더에 존재한다는 것을 특징한다. 0 과 동일한 pps_slice_act_qp_offsets_present_flag 는 slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset 이 슬라이스 헤더에 존재하지 않는다는 것을 특징한다. 존재하지 않는 경우, cu_chroma_qp_offset_enabled_flag 의 값은 0 과 동일한 것으로 추론된다.
- [0152] slice_act_y_qp_offset, slice_cb_qp_offset 및 slice_cr_qp_offset 는 각각 섹션 8.6.2 에서 도출된 루마, cb 및 cr 양자화 파라미터 qp 에 대한 오프셋들을 특징한다. slice_act_y_qp_offset, slice_cb_qp_offset 및 slice_cr_qp_offset 의 값들은 -12 내지 +12 의 범위에 있을 것이다. ChromaArrayType 이 0 과 동일한 경우, slice_act_cb_qp_offset 및 slice_act_cr_qp_offset 는 디코딩 프로세스에서 사용되지 않고, 디코더들은 그들의 값을 무시할 것이다.
- [0153] 도 6은 본 개시에 기재된 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오를 포스트 프로세싱 엔티티 (27) 로 출력하도록 구성될 수도 있다. 포스트 프로세싱 엔티티 (27) 는, 비디오 인코더 (20) 로부터 인코딩된 비디오 데이터를 프로세싱할 수도 있는, 미디어 인식 네트워크 엘리먼트 (MANE) 또는 스플라이싱/편집 디바이스 등의 비디오 엔티티의 일 예를 나타내도록 의도된다. 일부 사례들에서, 포스트 프로세싱 엔티티 (27) 는 MANE 와 같은 네트워크 엔티티의 일 예일 수도 있지만, 다른 사례들에서는, 포스트 프로세싱 엔티티 (27) 는 인코더 (20) 의 부분으로 고려될 수도 있다. 예를 들어, 일부 비디오 인코딩 시스템들에서, 포스트 프로세싱 엔티티 (27) 및 비디오 인코더 (20) 는 분리된 디바이스들의 부분들일 수도 있는 한편, 다른 사례들에서는, 포스트 프로세싱 엔티티 (27) 에 대해 설명된 기능은, 비디오 인코더 (20) 를 포함하는 동일한 디바이스에 의해 수행될 수도 있다. 또 다른 예들에서, 포스트 프로세싱 엔티티 (27) 는 도 1의 저장 디바이스 (17) 의 부분으로서 구현될 수도 있다.
- [0154] 비디오 인코더 (20) 는, 비디오 슬라이스들 내의 비디오 블록들의 인트라-, 인터-, 및 IMC 코딩을 수행할 수도 있다. 인트라 코딩은, 정해진 비디오 프레임 또는 화상 내의 비디오에서 공간적 중복성을 감소 또는 제거하기 위하여 공간적 예측에 의거한다. 인터 코딩은 비디오 시퀀스의 인접 프레임들 또는 화상들 내의 비디오에서 시간적 중복성을 감소 또는 제거하기 위하여 시간적 예측에 의거한다. 인트라 모드 (I 모드) 는 여러 공간 기반 압축 모드들 중 어느 것을 나타낼 수도 있다. 인터 모드들, 이를테면 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 은, 여러 시간 기반 압축 모드들 중 어느 것을 나타낼 수도 있다. 위에 설명된 바처럼, IMC 코딩 모드들은, 비디오 데이터의 프레임으로부터 공간적 리던던시를 제거할 수도 있지만, 통상적인 인트라 모드들과는 달리, IMC 코딩 코드들은, 인트라 예측 코딩 모드들에 의거하기 보다는, 프레임 내의 더 큰 탐색 구역에서 예측 블록들을 로케이팅하고 오프셋 벡터들로 예측 블록들을 나타내는데 사용될 수도 있다.
- [0155] 도 6 의 예에서, 비디오 인코더 (20) 는, 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 필터 유닛 (63), 디코딩된 화상 버퍼 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 프로세싱 유닛 (41) 은 모션 추정 유닛 (42), 모션 보상 유닛 (44) 및 인트라 예측 프로세싱 유닛 (46) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역 양자화 유닛 (58), 역 변환 프로세싱 유닛 (60), 및 합산기 (62) 를 포함한다. 필터 유닛 (64) 은 디블록킹 필터, ALF (adaptive loop filter) 및 SAO (sample adaptive offset) 필터 등의 하나 이상의 루프 필터들을 나타내도록 의도된다. 필터 유닛 (63) 이 인루프 필터인 것으로서 도 6에 도시되어 있지만, 다른 구성들에서, 필터 유닛 (63) 은 포스트 루프 필터로서 구현될 수도 있다.

- [0156] 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (33) 에 저장된 비디오 데이터는, 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. 디코딩된 화상 버퍼 (64) 는, 예를 들어, 인트라-, 인터- 또는 IMC 코딩 모드들에서 비디오 인코더 (20) 에 의한 비디오 데이터를 인코딩하는데 사용하기 위한 참조 비디오 데이터를 저장하는 참조 화상 메모리일 수도 있다. 비디오 데이터 메모리 (33) 및 디코딩된 화상 버퍼 (64) 는 동기 DRAM (SDRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 타입의 메모리 디바이스들을 포함하는, 동적 랜덤 액세스 메모리 (SDRAM) 등의 다양한 메모리 디바이스들 중의 어느 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (33) 및 디코딩된 화상 버퍼 (64) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 함께 온칩 (on-chip) 될 수도 있거나, 또는 그러한 컴포넌트들에 대해 오프칩 (off-chip) 될 수도 있다.
- [0157] 도 6에 도시된 바처럼, 비디오 인코더 (20) 는 비디오 데이터를 수신하고 그 비디오 데이터를 비디오 데이터 메모리 (33) 에 저장한다. 파티셔닝 유닛 (35) 은 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한, 예를 들어, LCU 및 CU 들의 쿼드트리 구조에 따른, 비디오 블록 파티셔닝 뿐만 아니라 슬라이스들, 타일들 또는 다른 더 큰 유닛들로의 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로, 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 슬라이스는, 다수의 비디오 블록들로 (그리고 가능하게는 타일들로 지칭되는 비디오 블록들의 세트들로) 분할될 수도 있다. 예측 프로세싱 유닛 (41) 은, 복수의 가능한 코딩 모드들 중 하나, 이를테면 복수의 인트라 코딩 모드들 중 하나, 복수의 인터 코딩 모드들 중 하나, 또는 복수의 IMC 코딩 모드들 중 하나를 현재 비디오 블록을 위해 여러 결과 (예를 들어, 코딩 레이트 및 왜곡의 레벨) 에 기초하여 선택할 수도 있다. 예측 프로세싱 유닛 (41) 은 결과적인 인트라, 인터 또는 IMC 코딩된 블록을 합산기 (50) 에 제공하여 레지듀얼 블록 데이터를 생성하고 합산기 (62) 에 제공하여 참조 화상으로서의 사용을 위해 인코딩된 블록을 재구성할 수도 있다.
- [0158] 예측 프로세싱 유닛 (41) 내의 인트라 예측 프로세싱 유닛 (46) 은 코딩될 현재 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 대해 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 하나 이상의 참조 화상들에서의 하나 이상의 예측 블록들에 대해 현재 비디오 블록의 인터 예측 코딩을 수행해 시간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 또한, 동일한 화상에서의 하나 이상의 예측 블록들에 대해 현재 비디오 블록의 IMC 코딩을 수행해 공간적 압축을 제공할 수도 있다.
- [0159] 모션 추정 유닛 (42) 은, 비디오 시퀀스에 대해 미리결정된 패턴에 따라 비디오 슬라이스를 위한 IMC 모드 또는 인터 예측 모드를 결정하도록 구성될 수도 있다. 미리결정된 패턴은, P 슬라이스들, B 슬라이스들 또는 GPB 슬라이스들로서 시퀀스에서 비디오 슬라이스들을 표기할 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적인 목적을 위해 따로따로 예시되어 있다. 모션 추정 유닛 (42) 에 의해 수행되는, 모션 추정은, 비디오 블록들을 위한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 예를 들어, 모션 벡터는, 참조 화상 내의 예측 블록에 대해 현재 비디오 프레임 또는 화상 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다. IMC 코딩의 경우에, IMC 에서 오프셋 벡터로도 지칭될 수도 있는 모션 벡터는, 현재 비디오 프레임 내의 예측 블록에 대하여 현재 비디오 프레임 또는 화상 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다.
- [0160] 예측 블록은, 절대 차이의 합 (sum of absolute difference; SAD), 제곱 차이의 합 (sum of square difference; SSD), 또는 다른 차이 매트릭들에 의해 결정될 수도 있는, 픽셀 차이에 관하여, 코딩될 비디오 블록의 PU에 밀접하게 매치하는 것으로 구해진 블록이다. 일부 예들에서, 비디오 인코더 (20) 는 디코딩된 화상 버퍼 (64) 에 저장된 참조 화상들의 서브 정수 픽셀 위치들을 위한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 화상의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 보관할 수도 있다. 그러므로, 모션 추정 유닛 (42) 은, 전 (full) 픽셀 위치들 그리고 분수 픽셀 위치들에 대해 모션 탐색을 수행하고 분수 픽셀 정밀도로 모션 벡터를 출력할 수도 있다.
- [0161] 모션 추정 유닛 (42) 은, PU 의 위치와 참조 화상의 예측 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서 비디오 블록의 PU를 위한 모션 벡터를 계산한다. 참조 화상은, 제 1 참조 화상 리스트 (리스트 0) 또는 제 2 참조 화상 리스트 (리스트 1) 로부터 선택될 수도 있고, 이들의 각각은 디코딩된 화상 버퍼 (64) 에 저장된 하나 이상의 참조 화상들을 식별한다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유

닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0162] 본 개시의 일부 기법들에 따르면, IMC 모드를 이용하여 비디오 블록을 코딩할 때, 모션 추정 유닛 (42) 은 비디오 블록의 루마 성분을 위해 모션 벡터, 또는 오프셋 벡터를 결정할 수도 있고, 루마 성분을 위한 오프셋 벡터에 기초하여 비디오 블록의 크로마 성분을 위한 오프셋 벡터를 결정할 수도 있다. 또 다른 예에서, IMC 모드를 이용하여 비디오 블록을 코딩할 때, 모션 추정 유닛 (42) 은 비디오 블록의 크로마 성분을 위한 모션 벡터, 또는 오프셋 벡터를 결정할 수도 있고, 크로마 성분을 위한 오프셋 벡터에 기초하여 비디오 블록의 루마 성분을 위한 오프셋 벡터를 결정할 수도 있다. 따라서, 비디오 인코더 (20) 는 비트스트림에서 하나의 오프셋 벡터만을 시그널링할 수도 있고, 이로부터 비디오 블록의 크로마 및 루마 성분들 양자 모두를 위한 오프셋 벡터들이 결정될 수도 있다.

[0163] 모션 보상 유닛 (44) 에 의해 수행된 모션 보상은, 모션 추정에 의해 결정된 모션 벡터에 기초한 예측 블록의 페칭 (fetching) 또는 생성을 수반할 수도 있으며, 가능하게는 서브-픽셀 정밀도로 보간을 수행한다. 보간 필터링은 알려진 픽셀 샘플들로부터 추가적인 픽셀 샘플들을 생성할 수도 있으며, 따라서, 비디오 블록을 코딩 하는데 사용될 수도 있는 후보 예측 블록들의 수를 잠재적으로 증가시킨다. 현재 비디오 블록의 PU를 위한 모션 벡터의 수신시에, 모션 보상 유닛 (44) 은, 모션 벡터가 참조 화상 리스트들 중 하나에서, 또는 IMC 코딩의 경우에, 코딩되고 있는 화상 내에서, 가리키는 예측 블록을 로케이팅할 수도 있다. 비디오 인코더 (20) 는, 코딩되는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여, 픽셀 차이 값들을 형성함으로써, 레지듀얼 비디오 블록을 형성한다. 픽셀 차이 값들은, 블록을 위한 레지듀얼 데이터를 형성하고, 루마 및 크로마 차이 성분들 양자 모두를 포함할 수도 있다. 합산기 (50) 는 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 모션 보상 유닛 (44) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 신텍스 엘리먼트들을 생성할 수도 있다.

[0164] 인트라 예측 프로세싱 유닛 (46) 은, 상술된 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 수행되는 인트라 예측 및 IMC 에 대한 대안으로서, 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 프로세싱 유닛 (46) 은 현재 블록을 인코딩하는데 이용할 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 프로세싱 유닛 (46) 은, 예를 들어, 별도의 인코딩 패스들 동안에, 다양한 인트라 예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 프로세싱 유닛 (46) (또는, 일부 예들에서, 모드 선택 유닛 (40)) 은 테스트된 모드들로부터 이용할 적절한 인트라 예측 모드를 선택할 수도 있다. 예를 들어, 인트라 예측 프로세싱 유닛 (46) 은 다양한 테스트된 인트라 예측 모드들을 위해 레이트 왜곡 분석을 이용하여 레이트 왜곡 값들을 산출하고, 테스트된 모드들 중에서 최상의 레이트 왜곡 특성들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록과, 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러) 의 양 뿐만 아니라 인코딩된 블록을 생성하는데 이용된 비트 레이트 (즉, 비트들의 수) 를 결정한다. 인트라 예측 프로세싱 유닛 (46) 은 그 왜곡들로부터 비 (ratio) 및 여러 인코딩된 블록들을 위한 레이트들을 산출하여 어느 인트라 예측 모드가 블록을 위한 최상의 레이트 왜곡 값을 나타내는지 결정할 수도 있다.

[0165] 어느 경우든, 블록을 위한 인트라 예측 모드를 선택한 후에, 인트라 예측 프로세싱 유닛 (46) 은 엔트로피 인코딩 유닛 (56) 에 그 블록을 위해 선택된 인트라 예측 모드를 나타내는 정보를 제공할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 본 개시의 기법들에 따라 선택된 인트라-예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 송신된 비트스트림에서 구성 데이터를 포함할 수도 있고, 이는 복수의 인트라 예측 모드 인덱스 테이블들 및 복수의 수정된 인트라 예측 모드 인덱스 테이블들 (코드워드 맵핑 테이블이라고도 한다), 다양한 블록들을 위한 인코딩 콘텍스트들의 정의들, 그리고 콘텍스트들의 각각을 위해 사용할 최고 확률 인트라 예측 모드의 표시들, 인트라 예측 모드 인덱스 테이블, 및 수정된 인트라 예측 모드 인덱스 테이블을 포함할 수도 있다.

[0166] 예측 프로세싱 유닛 (41) 이 (예를 들어, 인트라-예측, 인트라-예측 또는 IMC 를 통해) 현재 비디오 블록을 위한 예측 블록을 생성한 후에, 비디오 인코더 (20) 는 현재 비디오 블록으로부터 예측 블록을 감산함으로써 레지듀얼 비디오 블록을 형성한다. 레지듀얼 블록에서 레지듀얼 비디오 데이터는 하나 이상의 TU 들에 포함될 수도 있고 변환 프로세싱 유닛 (52) 에 적용될 수도 있다. 변환 프로세싱 유닛 (52) 은 DCT (discrete cosine transform) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 레지듀얼 변환 계수들로 레지듀얼 비디오 데이터를 변환한다. 변환 프로세싱 유닛 (52) 은 레지듀얼 비디오 데이터를 픽셀 도메인으로부터 변환

도메인, 이를테면 주파수 도메인으로 변환할 수도 있다.

[0167] 변환 프로세싱 유닛 (52) 은 양자화 유닛 (54) 에 결과적인 변환 계수들을 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킨다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 깊이를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 다음으로, 일부 예들에서, 양자화 유닛 (54) 은 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56) 은 스캔을 수행할 수도 있다.

[0168] 양자화 다음에, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 CAVLC (context adaptive variable length coding), CABAC (context adaptive binary arithmetic coding), SBAC (syntax-based context-adaptive binary arithmetic coding), PIPE (probability interval partitioning entropy) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩 다음에, 인코딩된 비트스트림은, 비디오 디코더 (30) 로 송신되거나 또는 비디오 디코더 (30) 에 의한 나중의 송신 또는 추출을 위해 보관될 수도 있다.

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

[0169] 역 양자화 유닛 (58) 및 역 변환 프로세싱 유닛 (60) 은 역 양자화 및 역 변환을 각각 적용하여, 참조 화상의 참조 블록으로서 나중에 사용하기 위해 픽셀 도메인에서 레지듀얼 블록을 재구성한다. 모션 보상 유닛 (44) 은, 참조 화상 리스트들 중의 하나 내의 참조 화상들 중의 하나의 예측 블록에 레지듀얼 블록을 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한 하나 이상의 보간 필터들을 그 재구성된 레지듀얼 블록에 적용하여, 모션 추정에 사용하기 위한 서브 정수 픽셀 값들을 산출할 수도 있다. 보간 필터링은 알려진 픽셀 샘플들로부터 추가적인 픽셀 샘플들을 생성할 수도 있으며, 따라서, 비디오 블록을 코딩하는데 사용될 수도 있는 후보 예측 블록들의 수를 잠재적으로 증가시킨다. 합산기 (62) 는 재구성된 레지듀얼 블록을 모션 보상 유닛 (44) 에 의해 생성되는 모션 보상된 예측 블록에 가산하여 참조 블록을 생성해 디코딩된 화상 버퍼 (64) 에 저장한다. 참조 블록은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 참조 블록으로서 사용되어 후속 비디오 프레임 또는 화상에서 블록을 인터 예측할 수도 있다.

[0170] 도 7은 본 개시에 기재된 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 7 의 예에서, 비디오 디코더 (30) 는, 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역 양자화 유닛 (86), 역 변환 프로세싱 유닛 (88), 합산기 (90), 필터 유닛 (91), 및 디코딩된 화상 버퍼 (92) 를 포함한다. 예측 프로세싱 유닛 (81) 은, 모션 보상 유닛 (82) 및 인트라 예측 프로세싱 유닛 (84) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서, 도 6 로부터의 비디오 인코더 (20) 에 대해 설명된 인코딩 패스에 일반적으로 상반되는 디코딩 패스를 수행할 수도 있다.

[0171] 디코딩 프로세스 동안, 비디오 디코더 (30) 는, 비디오 인코더 (20) 로부터 비디오 데이터, 예를 들어 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 선택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더 (30) 는 네트워크 엔티티 (29) 로부터 비디오 데이터를 수신하고 그 비디오 데이터를 비디오 데이터 메모리 (78) 에 저장할 수도 있다. 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될 비디오 데이터, 이를테면 인코딩된 비디오 비트스트림을 저장할 수도 있다. 비디오 데이터 메모리 (78) 에 저장된 비디오 데이터는, 예를 들어, 저장 디바이스 (17) 로부터, 예를 들어, 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체에 액세스하는 것에 의해, 획득될 수도 있다. 비디오 데이터 메모리 (78) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 화상 버퍼를 형성할 수도 있다. 따라서, 도 7 에 따로 도시되어 있지만, 비디오 데이터 메모리 (78) 및 디코딩된 화상 버퍼 (92) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 비디오 데이터 메모리 (78) 및 디코딩된 화상 버퍼 (92) 는 동기 DRAM (SDRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 타입의 메모리 디바이스들을 포함하는, 동적 랜덤 액세스 메모리 (SDRAM) 등의 다양한 메모리 디바이스들 중의 어느 것에 의해 형성될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 함께 온칩 (on-chip) 될 수도 있거나, 또는 그러한 컴포넌트들에 대해 오프칩 (off-chip) 될 수도 있다.

[0172] 네트워크 엔티티 (29) 는, 예를 들어, 서버, MANE, 비디오 에디터/스플라이서, 또는 전송된 기법들의 하나 이상을 구현하도록 구성된 다른 그러한 디바이스를 포함할 수도 있다. 네트워크 엔티티 (29) 는 비디오 인코더 (20) 와 같은 비디오 인코더를 포함할 수도 있거나 또는 포함하지 않을 수도 있다. 본 개시에 설명된 기법

들의 일부는, 네트워크 엔티티 (29) 가 인코딩된 비디오 비트스트림을 비디오 디코더 (30) 로 송신하기 전에 네트워크 엔티티 (29) 에 의해 구현될 수도 있다. 일부 비디오 디코딩 시스템들에서, 네트워크 엔티티 (29) 및 비디오 디코더 (30) 는 별개의 디바이스들의 부분들일 수도 있는 한편, 다른 사례들에서는, 네트워크 엔티티 (29) 에 대해 설명된 기능은, 비디오 디코더 (30) 를 포함하는 동일한 디바이스에 의해 수행될 수도 있다. 네트워크 엔티티 (29) 는 일부 경우들에서 도 1의 저장 디바이스 (17) 의 일 예일 수도 있다.

[0173] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 양자화된 계수들, 모션 벡터들, 및 다른 신택스 엘리먼트들을 생성하기 위하여 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 예측 프로세싱 유닛 (81) 에 포워드한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.

[0174] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81) 의 인트라 예측 프로세싱 유닛 (84) 은, 현재 프레임 또는 화상의 이전에 디코딩된 블록들로부터 시그널링된 인트라 예측 모드 및 데이터에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터 코딩된 (즉, B, P 또는 GPB) 슬라이스로서 코딩될 때 또는 블록이 IMC 코딩될 때, 예측 프로세싱 유닛 (81) 의 모션 보상 유닛 (82) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 블록들을 생성한다. 인터 예측을 위해, 예측 블록들은 참조 화상 리스트들 중의 하나의 리스트 내의 참조 화상들 중의 하나의 화상로부터 생성될 수도 있다. 비디오 디코더 (30) 는 디코딩된 화상 버퍼 (92) 에 저장된 참조 화상들에 기초하여 디폴트 (default) 구성 기법들을 이용하여 참조 프레임 리스트들, 리스트 0 및 리스트 1 을 구성할 수도 있다. IMC 코딩을 위해, 예측 블록들은, 예측되고 있는 블록들과 동일한 화상으로부터 생성될 수도 있다.

[0175] 모션 보상 유닛 (82) 은, 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱 (parsing) 하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 정보를 결정하고, 그 예측 정보를 사용하여, 디코딩되고 있는 현재 비디오 블록을 위한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82) 은 수신된 신택스 엘리먼트들의 일부를 사용하여 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예를 들어, 인트라 또는 인터 예측), 인터 예측 슬라이스 타입 (예를 들어, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스를 위한 참조 화상 리스트들의 하나 이상을 위한 구성 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록을 위한 모션 벡터들, 슬라이스의 각각의 인터 코딩된 비디오 블록을 위한 인터 예측 상태, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0176] 모션 보상 유닛 (82) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (82) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 이용되는 보간 필터들을 이용하여 참조 블록들의 서브 정수 픽셀들을 위한 보간된 값들을 계산할 수도 있다. 이 경우에, 모션 보상 유닛 (82) 은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 이용된 보간 필터들을 결정하고, 그 보간 필터들을 이용하여 예측 블록들을 생성할 수도 있다.

[0177] 본 개시의 일부 기법들에 따르면, IMC 모드를 이용하여 비디오 블록을 코딩할 때, 모션 보상 유닛 (82) 은 비디오 블록의 루마 성분을 위한, 모션 벡터, 또는 오프셋 벡터를 결정할 수도 있고, 루마 성분을 위한 모션 벡터에 기초하여 비디오 블록의 크로마 성분을 위한 모션 벡터를 결정할 수도 있다. 또 다른 예에서, IMC 모드를 이용하여 비디오 블록을 코딩할 때, 모션 보상 유닛 (82) 은 비디오 블록의 크로마 성분을 위한, 모션 벡터, 또는 오프셋 벡터를 결정할 수도 있고, 크로마 성분을 위한 모션 벡터에 기초하여 비디오 블록의 루마 성분을 위한 모션 벡터를 결정할 수도 있다. 따라서, 비디오 디코더 (30) 는 비트스트림에서 하나의 오프셋 벡터만을 수신할 수도 있고, 이로부터 비디오 블록의 크로마 및 루마 성분들 양자 모두를 위한 오프셋 벡터들이 결정될 수도 있다.

[0178] IMC 모드를 이용하여 비디오 블록을 디코딩할 때, 모션 보상 유닛 (82) 은, 예를 들어, 크로마 성분을 위한 오프셋 벡터를 결정하기 위해 루마 성분을 위한, IMC 모드를 위한 오프셋 벡터로 지칭되는 모션 벡터를 수정할 수도 있다. 모션 보상 유닛 (82) 은, 예를 들어, 비디오 블록을 위한 샘플링 포맷에 기초하여 그리고 오프셋 벡터가 가리키는 서브픽셀 위치의 정밀도에 기초하여 루마 블록의 오프셋 벡터의 x 성분 및 y 성분 중의 하나 또는 양자 모두를 수정할 수도 있다. 예를 들어, 비디오 블록이 4:2:2 샘플링 포맷을 이용하여 코딩되면, 모션 보상 유닛 (82) 은 크로마 성분을 위한 오프셋 벡터를 결정하기 위하여 루마 오프셋 벡터의, y 성분이 아니라, x 성분만을 수정할 수도 있다. 도 4로부터 알 수 있듯이, 4:2:2 샘플링 포맷에서, 크로마 블록들 및 루마 블록들은 수직 방향에서 동일한 수의 샘플들을 가지며, 따라서 y 성분의 수정을 잠재적으로 불필요하게 만

는다. 모션 보상 유닛 (82) 은, 크로마 예측 블록을 로케이팅하는데 사용될 때, 루마 오프셋 벡터가 (예를 들어, 현재 블록을 포함하는 현재 화상의 크로마 샘플에 있는 서브픽셀 위치에서) 크로마 샘플이 없는 위치를 가리키면, 루마 오프셋 벡터만을 수정할 수도 있다. 루마 오프셋 벡터는, 크로마 예측 블록을 로케이팅하는데 사용될 때, 크로마 샘플이 존재하는 위치를 가리키면, 모션 보상 유닛 (82) 은 루마 오프셋 벡터를 수정하지 않을 수도 있다.

[0179] 다른 예들에서, 비디오 블록이 4:2:0 샘플링 포맷을 이용하여 코딩되면, 모션 보상 유닛 (82) 은 크로마 성분을 위한 오프셋 벡터를 결정하기 위하여 루마 오프셋 벡터의 x 성분 및 y 성분 중의 어느 하나 또는 양자 모두를 수정할 수도 있다. 도 3으로부터 알 수 있듯이, 4:2:0 샘플링 포맷에서, 크로마 블록들 및 루마 블록들은 수직 방향 및 수평 방향 양자 모두에서 동일한 수의 샘플들을 가진다. 모션 보상 유닛 (82) 은, 크로마 예측 블록을 로케이팅하는데 사용될 때, 루마 오프셋 벡터가 (예를 들어, 현재 블록을 포함하는 현재 화상의 크로마 샘플에 있는 서브픽셀 위치에서) 크로마 샘플이 없는 위치를 가리키면, 루마 오프셋 벡터만을 수정할 수도 있다. 루마 오프셋 벡터는, 크로마 예측 블록을 로케이팅하는데 사용될 때, 크로마 샘플이 존재하는 위치를 가리키면, 모션 보상 유닛 (82) 은 루마 오프셋 벡터를 수정하지 않을 수도 있다.

[0180] 모션 보상 유닛 (82) 은, 수정된 오프셋 벡터로도 지칭되는 수정된 모션 벡터를 생성하기 위하여 루마 오프셋 벡터를 수정할 수도 있다. 모션 보상 유닛 (82) 은, 크로마 예측 블록을 로케이팅하는데 사용될 때, 크로마 블록에 사용된 수정된 오프셋 벡터가 더 낮은 해상도 서브픽셀 위치 또는 정수 픽셀 위치를 가리키도록 서브픽셀 위치를 가리키는 루마 오프셋 벡터를 수정할 수도 있다. 하나의 예로서, 1/8 픽셀 위치를 가리키는 루마 오프셋 벡터는 1/4 픽셀 위치를 가리키도록 수정될 수도 있고, 1/4 픽셀 위치를 가리키는 루마 오프셋 벡터는 1/2 픽셀 위치를 가리키도록 수정될 수도 있는 등이다. 다른 예들에서, 모션 보상 유닛 (82) 은, 수정된 오프셋 벡터가 크로마 참조 블록을 로케이팅하기 위한 정수 픽셀 위치를 항상 가리키도록 루마 오프셋 벡터를 수정할 수도 있다. 더 낮은 해상도 서브픽셀 위치 또는 정수 픽셀 위치를 가리키도록 루마 오프셋 벡터를 수정하는 것은 일부 보간 필터링의 필요성을 제거하거나 및/또는 임의의 필요한 보간 필터링의 복잡성을 감소시킬 수도 있다.

[0181] 도 3 및 도 4를 참조하여 상단 좌측 샘플이 위치 (0, 0) 에 위치되어 있다고 가정하면, 비디오 블록은 홀수 및 짝수 x 위치들 양자 모두 그리고 홀수 및 짝수 y 위치들 양자 모두에서 루마 샘플들을 갖는다. 4:4:4 샘플링 포맷에서, 비디오 블록은 또한, 홀수 및 짝수 x 위치들 양자 모두 및 홀수 및 짝수 y 위치들 양자 모두에서 크로마 샘플들을 갖는다. 따라서, 4:4:4 샘플링 포맷에 대해, 모션 보상 유닛은 루마 예측 블록 및 크로마 예측 블록 양자 모두를 로케이팅하는데 동일한 오프셋 벡터를 사용할 수도 있다. 4:2:2 샘플링 포맷에 대해, 도 4에 도시된 바처럼, 비디오 블록은 홀수 및 짝수 y 위치들 양자 모두에서 그러나 짝수 x 위치들에서만 크로마 샘플들을 갖는다. 따라서, 4:2:2 샘플링 포맷에 대해, 루마 오프셋 벡터가 홀수 x 위치를 가리키면, 모션 보상 유닛 (82) 은, 수정된 오프셋 벡터가 보간의 필요 없이 현재 블록의 크로마 블록을 위한 참조 크로마 블록을 로케이팅하는데 사용될 수 있도록 짝수 x 위치를 가리키는 수정된 오프셋 벡터를 생성하기 위하여 루마 오프셋 벡터의 x 성분을 수정할 수도 있다. 모션 보상 유닛 (82) 은, 예를 들어, 가장 가까운 짝수 x 위치로 라운딩 업 또는 라운딩 다운 중 어느 하나에 의해, 즉 가장 가까운 좌측 x 위치 또는 가장 가까운 우측 x 위치를 가리키도록 x 성분을 변경하는 것에 의해, x 성분을 수정할 수도 있다. 루마 오프셋 벡터가 이미 짝수의 x 위치를 가리키면, 수정이 필요하지 않을 수도 있다.

[0182] 4:2:0 샘플링 포맷에 대해, 도 3에 도시된 바처럼, 비디오 블록은 짝수 y 위치들에서만 그리고 짝수 x 위치들에서만 크로마 샘플들을 갖는다. 따라서, 4:2:0 샘플링 포맷에 대해, 루마 오프셋 벡터가 홀수 x 위치 또는 홀수 y 위치를 가리키면, 모션 보상 유닛 (82) 은, 수정된 오프셋 벡터가 보간의 필요 없이 현재 블록의 크로마 블록을 위한 참조 크로마 블록을 로케이팅하는데 사용될 수 있도록 짝수 x 위치를 가리키는 수정된 오프셋 벡터를 생성하기 위하여 루마 오프셋 벡터의 x 성분 또는 y 성분을 수정할 수도 있다. 모션 보상 유닛 (82) 은, 예를 들어, 가장 가까운 짝수 x 위치로 라운딩 업 또는 라운딩 다운 중 어느 하나에 의해, 즉 가장 가까운 좌측 x 위치 또는 가장 가까운 우측 x 위치를 가리키도록 x 성분을 변경하는 것에 의해, x 성분을 수정할 수도 있다. 모션 보상 유닛 (82) 은, 예를 들어, 가장 가까운 짝수 y 위치로 라운딩 업 또는 라운딩 다운 중 어느 하나에 의해, 즉 가장 가까운 상부 y 위치 또는 가장 가까운 하부 y 위치 중 어느 하나를 가리키도록 y 성분을 변경하는 것에 의해, y 성분을 수정할 수도 있다. 루마 오프셋 벡터가 이미 짝수 x 위치 및 짝수 y 위치를 가리키면, 수정이 필요하지 않을 수도 있다.

[0183] 역 양자화 유닛 (86) 은 비트스트림에서 제공되고 엔트로피 디코딩 유닛 (80) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉 탈양자화한다. 역 양자화 프로세스는, 양자화의 정도, 그리고, 마찬가지로, 적용되

어야 하는 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서 각 비디오 블록을 위해 비디오 인코더 (20)에 의해 산출된 양자화 파라미터의 이용을 포함할 수도 있다. 역 변환 프로세싱 유닛 (88)은, 픽셀 도메인에서 레지듀얼 블록들을 생성하기 위해 변환 계수들에, 역 변환, 예를 들어, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 적용한다.

[0184] 모션 보상 유닛 (82)이 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 블록을 위한 예측 블록을 생성한 후에, 비디오 디코더 (30)는 역 변환 프로세싱 유닛 (88)으로부터의 레지듀얼 블록들과 모션 보상 유닛 (82)에 의해 생성된 대응하는 예측 블록들을 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (90)는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 원하는 경우, (코딩 루프내 또는 코딩 루프 후의) 루프 필터들이 또한 픽셀 천이들을 매끄럽게 하거나 또는 다른 방법으로 비디오 품질을 향상시키는데 사용될 수도 있다. 필터 유닛 (91)은 디블록킹 필터, ALF (adaptive loop filter) 및 SAO (sample adaptive offset) 필터 등의 하나 이상의 루프 필터들을 나타내도록 의도된다. 필터 유닛 (91)이 인루프 필터인 것으로서 도 7에 도시되어 있지만, 다른 구성들에서, 필터 유닛 (91)은 포스트 루프 필터로서 구현될 수도 있다. 다음으로, 정해진 프레임 또는 화상에서 디코딩된 비디오 블록들은 디코딩된 화상 버퍼 (92)에 저장되고, 이는 후속 모션 보상을 위해 사용된 참조 화상들을 저장한다. 디코딩된 화상 버퍼 (92)는, 또한 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에 나중에 표출하기 위해 디코딩된 비디오를 저장하는 메모리의 부분일 수도 있거나, 또는 그러한 메모리로부터 분리될 수도 있다.

[0185] 도 8은 본 개시의 하나 이상의 하나 이상의 양태들에 따른 컬러 변환을 사용하여 RGB 컬러 공간을 갖는 비디오 데이터를 제 2 컬러 공간을 갖는 비디오 데이터의 블록들로 변환하기 위한 기법들을 이용할 수도 있는 다른 예시의 비디오 인코더 (21)를 도시하는 블록도이다.

[0186] 도 8은 비디오 인코더 (20)의 더욱 상세화된 버전을 도시한다. 비디오 인코더 (21)는 비디오 인코더 (20) (도 1) 또는 비디오 인코더 (20) (도 2)의 예일 수도 있다. 도 8의 예는 본 개시의 기법들을 구현하기 위한 2가지 가능한 예를 도시한다. 제 1 구현에서, 비디오 인코더 (21)는 하나 이상의 컬러 변환 중의 컬러 변환을 사용하여 제 1 컬러 공간을 갖는 입력 비디오 신호의 제 1 블록을 제 2 컬러 공간을 갖는 제 2 블록으로 적응적으로 변환한다. 제 2 도시된 예는 동일한 기법들을 수행하지만, 입력 신호에 대해서라기 보다, 레지듀얼 비디오 데이터의 블록들에 대해 컬러 변환을 수행한다.

[0187] 도 8의 예에서, 비디오 인코더 (21)는 스위치들 (101, 105, 113, 121)의 상태들에 기초하여 비디오 데이터의 예측 및 레지듀얼 블록들에 대해 컬러 변환들을 수행하는 것으로서 도시된다. 스위치들 (101, 105, 113, 및 121)이 대안적인 위치로 스위칭되는 경우, 비디오 인코더 (21)는 예측 및/또는 레지듀얼 비디오 데이터의 블록들을 변환하기 보다는, 모션 추정, 및 모션 예측을 수행하기 전에 RGB 컬러 공간을 갖는 오리지널 신호의 비디오 데이터의 블록들에 대한 제 2 컬러 공간을 갖는 비디오 데이터의 블록들로의 컬러 변환들을 수행하도록 구성된다.

[0188] 도 8에 도시된 바와 같은 레지듀얼 비디오 데이터의 블록들에 대한 컬러 변환들을 수행하는 하나의 예시의 프로세스가 이제 더욱 상세히 기술된다. 도 8의 예에서, 오리지널 신호 (100)는 (스위치 (101)의 경로에 후속하는) 예측 프로세싱 유닛 (104)으로 전달된다. 예측 프로세싱 유닛 (104)은 참조 화상 메모리 (122)로부터의 하나 이상의 참조 화상들로부터 데이터를 수신할 수도 있다. 예측 프로세싱 유닛 (104)은 비디오 데이터의 예측 블록을 생성하고, 레지듀얼 신호 (124)를 생성하기 위해 오리지널 신호 (100)로부터의 비디오 데이터의 예측 블록을 결합한다. 이러한 예에서, 적응적 컬러 변환기 (106)는 RGB 컬러 공간으로부터의 비디오 데이터의 예측 블록 및 레지듀얼 블록을 제 2 컬러 공간을 갖는 비디오의 제 2 예측 블록 및 제 2 레지듀얼 블록으로 변환한다. 일부 예들에서, 비디오 인코더 (21)는 비용 함수에 기초하여 제 2 컬러 공간 및 컬러 변환을 선택할 수도 있다.

[0189] 변환/양자화 유닛 (108)은 제 2 컬러 공간을 갖는 제 2 비디오 블록에 대해 변환 (예를 들어, 이산 코사인 변환 (DCT) 또는 다른 타입의 변환)을 수행할 수도 있다. 또, 변환/양자화 유닛 (108)은 제 2 비디오 블록 (즉, 변환된 레지듀얼 비디오 블록)을 양자화할 수도 있다. 엔트로피 인코더 (110)는 양자화된 레지듀얼 비디오 블록을 엔트로피 인코딩할 수도 있다. 엔트로피 인코더 (110)는 비디오 디코더, 예를 들어 비디오 디코더 (30)에 의한 디코딩을 위해 양자화된 레지듀얼 비디오 블록을 포함하는 비트스트림을 출력할 수도 있다.

[0190] 역양자화/역변환 유닛 (112)은 또한 양자화된, 변환된 계수 및/또는 레지듀얼 비디오 블록들을 수신할 수도 있고, 변환된 계수 및 레지듀얼 비디오 블록들을 역으로 변환 및 역양자화할 수도 있다. 역양자화된, 역으로

변환된 비디오 블록들은 여전히 이 시점에서 제 2 컬러 공간을 가질 수도 있다. 역양자화/역변환의 결과는 재구성된 레지듀얼 신호 (126) 이다. 역 적응적 컬러 변환기 (114) 는 적응저 컬러 변환기 (106) 에 의해 수행된 변환과 연관된 역 컬러 변환에 기초하여 재구성된 레지듀얼 신호를 역으로 컬러 변환할 수도 있다. 결과의 역으로 적응적 컬러 변환된 계수 및/또는 레지듀얼 비디오 블록들은 이 시점에서 RGB 컬러 공간을 가질 수도 있다.

[0191] 레지듀얼 비디오 블록에의 역 컬러 변환의 적용에 후속하여, 예측 보상기 (116) 는 레지듀얼 비디오 블록에 예측 블록을 다시 가산할 수도 있다. 디블록 필터 (118) 는 결과의 블록을 디블록킹할 수도 있다. SAO 필터 (120) 는 SAO 필터링을 수행할 수도 있다. 참조 화상 메모리 (122) 는 그 후 미래의 사용을 위해 결과의 재구성된 신호 (128) 를 저장할 수도 있다.

[0192] 레지듀얼 비디오 데이터의 블록이라기 보다, 입력 신호 (즉, 인코딩되지 않은 비디오 데이터) 의 비디오 블록을 컬러 변환하기 위해, 스위치 (101) 가 대안적인 위치로 돌러지고, 적응적 변환기 (102) 는 입력 비디오 블록을 하나 이상의 컬러 변환들 중의 컬러 변환을 사용하여 RGB 컬러 공간을 갖는 비디오 블록으로부터 제 2 컬러 공간으로 컬러 변환한다. 예측 프로세싱 유닛 (104) 에 의한 예측은 상술된 바와 같이 진행되지만, 그 결과는 적응적 컬러 변환기 (106) 에 의해 컬러 변환되기 보다는, 스위치 (105) 가 (도 8 에 도시된 위치와 비교하여) 대안적인 위치에 있기 때문에 직접 변환/양자화 유닛 (108) 으로 피딩될 수도 있다.

[0193] 변환/양자화 유닛 (108), 엔트로피 인코더 (110), 및 역양자화/역 변환 유닛 (112) 은 각각은 레지듀얼 비디오 블록을 컬러 변환하는 것에 대해 상술된 바와 같이 동작할 수도 있고, 재구성된 신호 (126) 가 생성되며, 또한 제 2 컬러 공간에 있다. 재구성된 신호 (126) 는 스위치 (113) 를 통해 예측 보상기 (116) 로 피딩된다. 스위치 (113) 는 도 8 에 도시된 위치에 대해 대안적인 위치에 있고, 역 적응적 컬러 변환기 (114) 는 바이패싱된다. 예측 보상기 (116), 디블록 필터 (118), 및 SAO 필터 (120) 는 재구성된 신호 (128) 를 생성하기 위해 레지듀얼 비디오 블록을 컬러 변환하는 것에 대해 상술된 바와 같이 동작할 수도 있다. 그러나, 상술된 재구성된 신호 (128) 와 달리, 이 예에서는, 재구성된 신호 (128) 의 블록은, RGB 컬러 공간보다는 제 2 컬러 공간을 여전히 가질 수도 있다. 제

[0194] 재구성된 신호 (128) 는 도 8 에 도시된 위치에 대해 대안적인 위치에 있는 스위치 (121) 를 통해 역 적응적 컬러 변환기 (130) 으로 피딩될 수도 있다. 역 적응적 컬러 변환기 (130) 는 재구성된 신호 (128) 의 블록들을 RGB 컬러 공간을 갖는 블록들로 역으로 컬러 변환할 수도 있고, 참조 화상 메모리 (122) 는 그 블록들을 장래의 참조를 위해 참조 화상의 블록들로서 저장할 수도 있다.

[0195] 상술된 바와 같이, 비디오 인코더 (21) 는 제 2 컬러 공간으로 RGB 컬러 공간을 갖는 비디오 데이터의 제 1 블록을 변환하기 위해 하나 이상의 컬러 공간들의 변환을 선택할 수도 있다. 일부 예들에서, 비디오 인코더 (21) 는 컬러 변환들 각각과 연관된 레이트-왜곡 비용들을 적응적으로 계산함으로써 컬러 변환을 선택한다. 예를 들어, 비디오 인코더 (21) 는 CU 또는 CU 의 블록에 대한 최저 연관된 왜곡 비용을 갖는 복수의 컬러 변환들의 컬러 변환을 선택할 수도 있다. 비디오 인코더 (21) 는 최저 연관된 왜곡 비용을 갖는 선택된 컬러 변환을 나타내는 인덱스 신택스 엘리먼트 또는 다른 신택스 데이터를 시그널링할 수도 있다.

[0196] 일부 예들에서, 비디오 인코더 (21) 는 컬러 변환과 연관된 왜곡 (예를 들어, 충실도의 손실) 뿐 아니라 컬러 변환에 의한 비트레이트 (예를 들어, 달성된 압축) 사이의 트레이드오프를 설명하는 라그랑주 비용 함수를 이용할 수도 있다. 일부 예들에서, 라그랑주 비용은 $L = D + \lambda R$, 여기서 L 은 라그랑주 비용이고, D 는 왜곡이며, λ 는 라그랑주 승수, 및 R 은 비트레이트이다. 일부 예들에서, 비디오 인코더 (21) 는 라그랑주 비용을 최소화하는 복수의 컬러 변환들의 컬러 변환을 나타내는 인덱스 신택스 엘리먼트를 시그널링할 수도 있다.

[0197] 일부 고성능 고충실도 비디오 코딩 애플리케이션들 또는 구성들에서, 왜곡은 비트레이트를 최소화하는 것보다 많이 최소화되어야 한다. 그러한 경우들에서, RGB 컬러 공간으로부터 제 2 컬러 공간으로 비디오 데이터를 변환하는 경우, 비디오 인코더 (21) 는 컬러 변환, 및 최소 왜곡을 야기하는 컬러 공간을 선택할 수도 있다. 비디오 인코더 (21) 는 최소 왜곡을 야기하는 선택된 컬러 변환 또는 컬러 공간을 나타내는 인덱스 신택스 엘리먼트를 시그널링할 수도 있다.

[0198] 일부 다른 경우들에서, 비디오 인코더 (21) 는 RGB 비디오 데이터의 블록의 컬러 컴포넌트들 각각과 제 2 컬러 공간의 블록의 컬러 컴포넌트들 사이의 상관에 기초하여 RGB 컬러 공간의 블록들을 제 2 컬러 공간으로 변환하는 비용을 계산할 수도 있다. 최저 연관된 비용을 갖는 컬러 변환은 입력 신호의 RGB 컬러 컴포넌트들과 가

장 근접하게 상관되는 컬러 컴포넌트들을 갖는 컬러 변환일 수도 있다. 비디오 인코더 (21) 는 그것의 컬러 컴포넌트들과 RGB 컬러 컴포넌트들 사이의 최고의 상관을 갖는 선택된 컬러 변환을 나타내는 인덱스 신택스 엘리먼트를 시그널링할 수도 있다.

[0199] 일부 경우들에서, 비디오 인코더 (21) 는 비디오 데이터의 상이한 CU 들, LCU 들, CTU 들, 또는 다른 유닛들에 대해 상이한 컬러 변환들을 선택할 수도 있다. 즉, 단일의 화상에 대해, 비디오 인코더 (21) 는 상이한 컬러 공간들과 연관된 상이한 컬러 변환들을 선택할 수도 있다. 다수의 상이한 컬러 변환들을 선택하는 것은 더 양호하게 코딩 효율을 최적화하고 레이트 왜곡을 감소시킬 수도 있다. 비디오 인코더 (21) 가 다수의 변환들 중 어느 변환을 현재의 블록에 대해 선택했는지를 나타내기 위해, 비디오 인코더 (21) 는 선택된 컬러 변환에 대응하는 인덱스 값을 시그널링할 수도 있다. 비디오 인코더 (21) 는 비디오의 제 1 블록, CTU, CU, PU, 및 TU 중 하나 이상에서 인덱스 값을 시그널링할 수도 있다.

[0200] 그러나, 일부 경우들에서, 비디오 인코더 (21) 는 하나의 또는 복수의 블록들, 또는 CVS 로서 지칭되는 코딩된 화상들의 시퀀스에 적용되어야 하는 단일의 컬러 변환을 결정할 수도 있다. 단지 하나의 컬러 변환이 선택되는 경우에, 각각의 블록에 대해, 비디오 인코더 (21) 는 플래그 신택스 엘리먼트를 시그널링할 수도 있다. 플래그 신택스 엘리먼트의 하나의 값은 비디오 인코더 (21) 가 현재의 블록에 또는 CVS 에서의 모든 화상들에 단일의 변환을 적용했다는 것을 나타낼 수도 있다. 플래그 신택스 엘리먼트의 다른 값은 현재의 블록에 어떠한 변환도 적용되지 않았다는 것을 나타낸다. 비디오 인코더 (21) 는 예를 들어 상술된 비용-기반 기준들을 사용하여 개별적인 기초로 화상의 블록들 각각에 컬러 변환을 적용할지 여부를 결정할 수도 있다.

[0201] 일부 예들에서, 비디오 인코더 (21) 는 복수의 블록들 중 각각의 블록에 복수의 역 컬러 변환들의 미리 정의된 컬러 변환을 적용할지 여부를 결정한다. 예를 들어, 비디오 인코더 (21) 및 비디오 디코더 (31) 는 디폴트 미리 정의된 컬러 변환/역 컬러 변환을 이용할 수도 있다. 복수의 블록들 중 각각의 블록에 미리 정의된 컬러 변환을 적용하기로 결정하는 것에 응답하여, 비디오 인코더 (21) 는 미리 정의된 컬러 변환이 비디오 데이터의 복수의 블록들 중 각각의 블록에 적용되었다는 것을 나타내는 데이터를 디코딩하지 않고 미리 정의된 컬러 변환을 사용하여 복수의 블록들 각각을 변환할 수도 있다.

[0202] 역의 방식으로, 비디오 디코더 (31) 는 복수의 블록들 중 각각의 블록에 복수의 역 컬러 변환들 중 미리 정의된 역 컬러 변환을 적용할지 여부를 결정하도록 구성될 수도 있다. 복수의 블록들 중 각각의 블록에 미리 정의된 역 컬러 변환을 적용하기로 결정하는 것에 응답하여, 비디오 디코더 (31) 는 미리 정의된 컬러 변환이 비디오 데이터의 복수의 블록들 중 각각의 블록에 적용되었다는 것을 나타내는 데이터를 디코딩하지 않고 미리 정의된 컬러 변환을 사용하여 복수의 블록들 각각을 역으로 변환할 수도 있다.

[0203] 본 개시의 컬러 변환들은 비디오 데이터의 블록에 대한 아이덴티티 변환, 차분 변환, 가중 차분 변환, DCT, YCbCr 변환, YCgCo 변환, 및 YCgCo-R 변환을 포함할 수도 있지만, 이들에 반드시 제한되지는 않는다. 비디오 인코더 (21) 와 같은, 본 개시의 기법들에 따라 구성된 비디오 코더는 하나 이상의 이들 변환들 및/또는 그들의 역들 뿐만 아니라 Adobe RGB, sRGB, scRGB, Rec. 709, Rec. 2020, Adobe Wide Gamut RGB, ProPhoto RGB, CMYK, Pantone, YIQ, YDbDr, YPbPr, xvYCC, ITU BT.601, ITU BT.709, HSV, 및 여기에 구체적으로 기술되지 않은 다른 컬러 공간들, 컬러 공간들, 및/또는 크로마 샘플링 포맷들로/로부터의 변환들과 같은 다른 변환들을 적용할 수도 있다.

[0204] RGB 컬러 공간을 갖는 비디오 데이터의 블록에 컬러 변환을 적용하기 위해, 비디오 인코더 (21) 는 컬러 변환 매트릭스를 갖는 RGB 화소의 적색, 녹색, 및 청색 컬러 컴포넌트들을 포함하는 3 x 1 매트릭스를 곱할 수도 있다. 그 곱셈의 결과는 제 2 컬러 공간을 갖는 화소이다. 비디오 코더는 제 2 컬러 공간에서의 화소들의 제 2 블록을 생성하기 위해 비디오 블록의 각 화소에 컬러 변환 매트릭스를 적용할 수도 있다. 여러 컬러 변환들이 이제 더 상세히 기술된다.

[0205] 일부 예들에서, 비디오 인코더 (21) 는 아이덴티티 변환 매트릭스 또는 역 아이덴티티 변환 매트릭스를 적용할 수도 있다. 아이덴티티 변환 매트릭스는:

[0206]
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[0207] 을 포함하고, 비디오 디코더 (30) 가 적용할 수도 있는 역 변환 매트릭스는:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[0208]

[0209] 을 포함한다.

[0210] 비디오 코더가 아이덴티티 변환을 적용하는 경우, 결과의 화소 값은 입력 화소 값과 동일하며, 즉, 아이덴티티 변환을 적용하는 것은 전혀 컬러 변환을 적용하지 않는 것과 등가이다. 비디오 인코더 (21) 는 비디오 블록들의 RGB 컬러 공간을 유지하는 것이 요구될 때 아이덴티티 변환을 선택할 수도 있다.

[0211] 다른 예에서, 비디오 인코더 (21) 는 차분 변환 매트릭스를 적용할 수도 있다. 차분 변환 매트릭스는:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

[0212]

[0213] 를 포함한다.

[0214] 비디오 디코더 (30) 는:

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

[0215]

[0216] 를 포함하는 리시프로컬 (reciprocal), 역 차분 매트릭스를 적용할 수도 있다.

[0217] 다른 예에서, 비디오 인코더 (21) 는 가중 차분 변환 또는 역 가중 차분 변환을 적용하도록 구성될 수도 있다. 가중 차분 변환 매트릭스는:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -\alpha_1 & 1 \\ 1 & -\alpha_2 & 0 \end{bmatrix}$$

[0218]

[0219] 을 포함하고, 비디오 디코더 (31) 가 적용할 수도 있는 역 가중 차분 매트릭스는:

$$\begin{bmatrix} \alpha_2 & 0 & 1 \\ 1 & 0 & 0 \\ \alpha_1 & 1 & 0 \end{bmatrix}$$

[0220]

[0221] 을 포함한다.

[0222] 가중 차분 변환들에서, α_1 및 α_2 는 비디오 코더가 조정할 수도 있는 파라미터들이다. 일부 예들에서, 비디오 인코더 (20) 는 다음의 식들에 따라 파라미터들 α_1 및 α_2 를 계산할 수도 있다.

[0223] $\alpha_1 = \text{cov}(G, B) / \text{var}(G)$, 및

[0224] $\alpha_2 = \text{cov}(G, R) / \text{var}(G)$.

[0225] 비디오 인코더 (21) 는 여러 예들에서 코딩된 비디오 비트스트림에서 α_1 및 α_2 의 값들을 시그널링할 수도 있다.

[0226] 이들 식들에서, R 은 적색 컬러채널에 대응하고, G 는 녹색 컬러 채널에 대응하며, B 는 RGB 컬러 공간의 청색 컬러 채널에 대응한다. 차분 변환 식들에서, "cov()" 는 공분산 함수이고, "var()" 는 분산 함수이다.

[0227] R, G, 및 B 의 값들을 결정하기 위해, 인코더 또는 디코더는 공분산 및 분산 함수들이 인코더 또는 디코더에 의해 계산될 때 동일한 결과 또는 가중치를 갖는 것을 보장하기 위해 참조 화소들의 세트를 이용할 수도 있다. 일부 예들에서, 특정의 참조 화소들은 (예를 들어, 코딩된 비디오 비트스트림에서의 선택스 엘리먼트들로서) 코딩된 비디오 비트스트림에서 시그널링될 수도 있다. 다른 예들에서, 인코더 및 디코더는 소정의 참조 화

소들을 사용하도록 사전 프로그래밍될 수도 있다.

[0228] 일부 예들에서, 비디오 인코더 (21) 는 차분 변환을 사용하여 블록들을 변환할 때 a_1 및 a_2 의 값들을 제한 또는 제약할 수도 있다. 비디오 코더는 정수들 또는 다이어딕 (dyadic) 수들, 예를 들어 1/2, 1/4, 1/8, 등 .. 의 세트로 a_1 및 a_2 의 값들을 제약할 수도 있다. 다른 예들에서, 비디오 코더는 다이어딕 수, 예를 들어 1/8, 2/8, 3/8, ..., 8/8 를 갖는 분수의 값들로 a_1 및 a_2 를 제한할 수도 있다. 다이어딕 수 또는 다이어딕 함수는 2 의 멱인 분모를 갖는 유리수이고, 여기서 분자는 정수이다. a_1 및 a_2 의 값들을 제한하는 것은 a_1 및 a_2 를 코딩하는 비트스트림 효율을 개선할 수도 있다.

[0229] 다른 예들에서, 비디오 인코더 (21) 는 DCT 변환을 사용하여 제 2 블록을 생성하기 위해 RGB 컬러 공간을 갖는 블록을 변환하도록 구성될 수도 있다. DCT 는 상이한 주파수들 및 진폭들의 사인파들의 합으로서 샘플들을 표현하도록 블록의 샘플들을 변환한다. DCT 변환 또는 역 변환은 코사인 함수들의 합에 의해 데이터 포인트들의 유한 시퀀스로 및로부터 화소를 변환할 수도 있다. DCT 변환 매트릭스는:

$$\begin{bmatrix} 0.5774 & 0.5774 & 0.5774 \\ 0.7071 & 0 & -0.7071 \\ 0.4082 & -0.8156 & 0.4082 \end{bmatrix}$$

[0230] 에 대응한다.
[0231] 에 대응한다.

[0232] 역의 방식에서, 비디오 디코더 (31) 는 DCT 를 사용하여 변환된 블록들에 역 변환을 적용하여 원래의 샘플들로 다시 블록들을 되돌리도록 구성될 수도 있다. 역 DCT 변환 매트릭스는:

$$\begin{bmatrix} 0.5774 & 0.7071 & 0.4082 \\ 0.5774 & 0 & -0.8156 \\ 0.5774 & -0.7071 & 0.4082 \end{bmatrix}$$

[0233] 에 대응한다.
[0234] 에 대응한다.

[0235] 비디오 인코더 (21) 는 또한 YCbCr 컬러 공간을 갖는 블록을 생성하기 위해 RGB 컬러 공간을 갖는 블록에 YCbCr 변환을 적용할 수도 있다. 상술된 바와 같이, YCbCr 컬러 공간은 루마 (Y) 컴포넌트 뿐아니라, 청색 색차 (Cb) 및 적색 색차 (Cr) 컴포넌트들을 포함한다. YCbCr 변환 매트릭스는:

$$\begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix}$$

[0236] 에 대응할 수도 있다.
[0237] 에 대응할 수도 있다.

[0238] 비디오 디코더 (31) 는 YCbCr 컬러 공간을 갖는 블록을 RGB 컬러 공간을 갖는 블록으로 변환하기 위해 역 YCbCr 변환을 적용하도록 구성될 수도 있다. 역 YCbCr 변환 매트릭스는:

$$\begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix}$$

[0239] 에 대응할 수도 있다.
[0240] 에 대응할 수도 있다.

[0241] 비디오 인코더 (21) 는 또한 YCgCo 컬러 공간을 갖는 블록을 생성하기 위해 RGB 컬러 공간을 갖는 블록에 YCgCo 변환을 적용할 수도 있다. YCgCo 컬러 공간은 루마 (Y) 컴포넌트 뿐아니라, 녹색 색차 (Cg) 및 오렌지색 색차 (Co) 컴포넌트들을 포함한다. YCgCo 변환 매트릭스는:

$$\begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix}$$

[0242] 에 대응할 수도 있다.
[0243] 에 대응할 수도 있다.

[0244] 비디오 디코더 (31) 는 YCgCo 컬러 공간을 갖는 블록을 RGB 컬러 공간을 갖는 블록으로 변환하기 위해 역 YCgCo

변환을 적용하도록 구성될 수도 있다. 역 YCgCo 변환 매트릭스는:

$$\begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix}$$

[0245]

에 대응할 수도 있다.

[0246]

[0247] 비디오 인코더 (21) 는 또한 YCgCo-R 컬러 공간을 갖는 블록을 생성하기 위해 RGB 컬러 공간을 갖는 블록에 YCgCo-R 변환을 적용하도록 구성될 수도 있다. YCgCo-R 컬러 공간은 루마 (Y) 컴포넌트 뿐만 아니라, 녹색 색차 (Cg) 및 오렌지색 색차 (Co) 컴포넌트들을 포함한다. 그러나, 상술된 YCgCo 변환과 달리, YCgCo-R 변환은 가역적이며, 예를 들어, YCgCo-R 변환은 예를 들어 라운딩 에러들로 인해 임의의 왜곡을 생성하지 않을 수도 있다.

[0247]

[0248]

YCbCr 변환 매트릭스는:

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

[0249]

에 대응할 수도 있다.

[0250]

[0251] 비디오 디코더 (31) 는 역 YCgCo-R 변환을 적용하도록 구성될 수도 있다. YCgCo-R 역 변환은 YCgCo-R 컬러 공간을 갖는 블록들을 RGB 컬러 공간을 갖는 블록들로 역으로 변환한다. 역 YCgCo-R 변환 매트릭스는:

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

[0252]

에 대응할 수도 있다.

[0253]

[0254] 여기에 기술된 컬러 변환들 중 임의의 것을 적용하기 위해, 비디오 인코더 (21) 는 유연한 파라미터들을 갖는 리프팅 (lifting) 스킴을 구현할 수도 있다. 리프팅 스킴은 이산 웨이블릿 변환을 리프팅 스텝들로서 또는 래더 (ladder) 구조들로서 지칭되는 간단한 필터링 스텝들의 유한 시퀀스로 분해하는 기법이다. 비디오 인코더 (21) 는 코딩된 비디오 비트스트림에서 파라미터들을 시그널링할 수도 있거나, 비디오 인코더 (21) 는 동일한 방식으로 파라미터들을 도출할 수도 있다. 리프팅 스킴의 하나의 예는 다음과 같다:

$$\begin{aligned} R' &= R + [aB] \\ B' &= B + [bR'] \\ G' &= G + [cB'] \\ R'' &= R' + [dG'] \end{aligned}$$

[0255]

[0256] 여기서, a, b, c, 및 d 는 상술된 바와 같은 파라미터들이다. 이러한 리프팅 스킴에서, R, G, 및 B 는 각각 적색, 녹색, 및 청색 컬러 채널들 또는 샘플들이다. 가중된 차분 변환에 대해 상술된 a 파라미터들과 마찬가지로, a, b, c, 및 d 의 값들은 예를 들어 부호들이 단지 포지티브 또는 네거티브일 수 있도록 제한 또는 한정될 수도 있다. 일부 경우들에서,

$$\begin{aligned} R''' &= [eR'' + f] \\ B''' &= [gB' + h] \\ G''' &= [iG' + j] \end{aligned}$$

[0257]

[0258] 과 같은 리프팅 스킴에서의 추가적인 스텝들이 존재할 수도 있으며, 여기서 f, g, h, i, 및 j 는 파라미터들이다. 리프팅 스킴을 사용할 때 뿐만 아니라, 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 3 개의 컴포넌트들의 출력 깊이를 정규화할 수 있고, R''', B''', 및 G''' 는 각 컴포넌트에 대해 반드시 동일하지는 않을 수도 있는 미리 결정된 비트 깊이 내에서 정규화될 수 있다.

[0258]

- [0259] 도 9 는 본 개시의 하나 이상의 양태들에 따른 역 컬러 변환을 사용하여 제 1 컬러 공간을 갖는 비디오 데이터를 제 2 의, RGB 컬러 공간을 갖는 비디오 데이터로 역으로 변환하는 기법들을 이용할 수도 있는 다른 예시의 비디오 디코더 (31) 를 도시하는 블록도이다.
- [0260] 도 9 는 도 1 및 도 7 의 비디오 디코더 (30) 에 대한 비디오 디코더 (31) 의 더욱 상세한 버전을 도시한다. 실제로, 일부 예들에서, 비디오 디코더 (31) 는 비디오 디코더 (30) (도 7) 및/또는 비디오 디코더 (30) (도 1) 의 더욱 특정한 예로 고려될 수도 있다. 도 9 의 예는 본 개시의 기법들을 구현하는 2 개의 가능한 예들을 도시한다. 제 1 구현에서, 비디오 디코더 (31) 는 복수의 역 컬러 변환들 중의 역 컬러 변환을 사용하여 제 1 컬러 공간 (예를 들어, 난-RGB 컬러 공간) 으로부터의 입력 비디오 신호의 블록을 제 2 의, RGB 컬러 공간을 갖는 제 2 블록으로 적응적으로 역으로 변환한다. 제 2 도시된 예는 동일한 기법들을 수행하지만, 입력 신호에 대해서라기 보다, 레지듀얼 비디오 데이터의 블록들에 대해 역 컬러 변환을 수행한다
- [0261] 도 9 의 예에서, 비디오 디코더 (31) 는 웨이 (way) 스위치들 (145, 및 156) 이 현재 스위칭되어 있기 때문에 레지듀얼 비디오 데이터 예의 블록들에 대해 역 컬러 변환들을 수행하는 것으로 도시된다. 스위치들 (145 및 156) 이 대안적인 위치로 스위칭되는 경우, 비디오 디코더 (31) 는 레지듀얼 비디오 데이터의 블록들을 역으로 변환하기보다, 제 1 표현을 갖는 입력 비디오 데이터의 블록들을 제 2 의, RGB 컬러 공간을 갖는 비디오 데이터의 블록들로 역으로 컬러 변환하도록 구성된다.
- [0262] 도 9 에 도시된 레지듀얼 비디오 데이터의 블록들에 대해 역 컬러 변환들을 수행하는 프로세스가 이제 상세히 기술된다. 도 9 의 예에서, (입력 신호로서도 지칭되는) 인코딩된 입력 비트스트림 (140) 은 엔트로피 디코딩 유닛 (142) 으로 전달된다. 엔트로피 디코딩 유닛 (142) 은 제 1 컬러 공간을 갖는 레지듀얼 비디오 데이터의 양자화된 블록을 생성하기 위해 비트스트림 (140) 을 엔트로피 디코딩할 수도 있다. 예를 들어, 엔트로피 디코딩 유닛 (142) 은 비트스트림 (140) 에 포함된 특정의 선택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 역양자화/역변환 유닛 (144) 은 변환 계수 블록을 역양자화할 수도 있다. 추가적으로, 역양자화/역변환 유닛 (144) 은 레지듀얼 비디오 데이터를 포함하는 변환 블록을 결정하기 위해 변환 계수 블록에 역변환을 적용할 수도 있다. 따라서, 역양자화/역변환 유닛 (144) 은 비트스트림 (140) 의 엔트로피 디코딩된 비디오 데이터의 블록들을 역양자화 및 역으로 변환할 수도 있다. 비디오 디코더 (31) 가 레지듀얼 데이터의 블록들을 역으로 컬러 변환하도록 구성되는 경우, 스위치 (148) 는 역 적응적 컬러 변환기 (150) 로 제 1 컬러 공간을 갖는 레지듀얼 비디오 데이터의 블록을 피딩한다. 이러한 방식으로, 역 적응적 컬러 변환기 (150) 는 TU 의 변환 블록을 수신할 수도 있다.
- [0263] 역 적응적 컬러 변환기 (150) 는 제 1 컬러 공간을 갖는 비디오 데이터의 블록을 제 2 의, RGB 컬러 공간을 갖는 비디오 데이터의 제 2 블록으로 적응적으로 역으로 변환할 수도 있다. 예를 들어, 역 적응적 컬러 변환기 (150) 는 TU 의 변환 블록에 적용할 역변환을 선택할 수도 있다. 이러한 예에서, 역 적응적 컬러 변환기 (150) 는 변환 블록을 제 1 컬러 공간으로부터 RGB 컬러 공간으로 변환하기 위해 변환 블록에 선택된 역변환을 적용할 수도 있다. 예측 보상 유닛 (152) 은 참조 화상 메모리 (154) 로부터의 참조 화상을 결합할 수도 있다. 예를 들어, 예측 보상 유닛 (152) 은 CU 의 TU 의 변환 블록을 수신할 수도 있다. 이러한 예에서, 예측 보상 유닛 (152) 은 CU 에 대한 코딩 블록을 결정할 수도 있다. 이러한 예에서, CU 의 코딩 블록의 각 샘플은 변환 블록에서의 샘플 및 CU 의 PU 에 대한 예측 블록에서의 대응하는 샘플의 합과 동일할 수도 있다. 디블록 필터 (157) 는 결합된, 재구성된 이미지를 디블록킹할 수도 있다. SAO 필터 유닛 (158) 은 적용가능한 경우 추가적인 SAO 필터링을 수행할 수도 있다.
- [0264] SAO 필터 유닛 (158) 의 출력은 재구성된 신호 (160) 이다. 비디오 디코더 (31) 가 레지듀얼 비디오 데이터의 블록들을 역으로 컬러 변환하도록 구성되는 경우, 스위치 (162) 는 참조 화상으로서 장래의 사용을 위해 참조 화상 메모리 (154) 로 재구성된 신호 (160) 를 피딩한다. 비디오 디코더 (31) 는 또한 이미지/비디오 (164) 로서 재구성된 신호 (160) 를 출력할 수도 있다.
- [0265] 비디오 디코더 (31) 가 레지듀얼 비디오 데이터의 블록들과 대조적인 오리지널 입력 신호의 블록들을 역으로 컬러 변환하도록 구성되는 예들에서, 엔트로피 디코딩 유닛 (142) 및 역양자화/역변환 유닛 (144) 은 이전에 기술된 방식으로 동작한다. 스위치 (148) 는 대안적인 위치에 있고, 예측 보상 유닛 (152) 으로 직접 재구성된 레지듀얼 신호를 피딩한다. 이 시점에서, 예측 보상 유닛 (152) 으로 제공된 레지듀얼 블록은 RGB 컬러 공간이라기 보다는, 여전히 제 1 컬러 공간에 있다.
- [0266] 예측 보상 유닛 (152) 은 오리지널 이미지의 블록을 재구성할 수도 있고, 참조 화상 메모리 (154) 로부터의 화상들의 하나 이상의 블록들과 레지듀얼 블록을 결합할 수도 있다. 디블록 필터 (157) 및 SAO 필터 유닛

(158) 은 비디오 데이터의 레지듀얼 블록들을 역으로 변환하는 것에 대해 상술된 바와 같이 동작할 수도 있다.

SAO 필터 유닛 (158) 의 출력은 재구성된 신호 (160) 이며, 그것의 블록들은 여전히 제 1 컬러 공간에 있으며, RGB 컬러 공간을 갖지 않을 수도 있다 (예를 들어, 블록들은 아이덴티티 변환이 사용되었던 경우에 RGB 컬러 공간을 여전히 가질 수도 있다).

[0267] 재구성된 신호 (160) 는 도 9 에 도시된 위치에 비해 대안적인 위치에 있는 스위치 (162) 를 통해 역 적응적 컬러 변환기 (166) 로 피딩될 수도 있다. 역 적응적 컬러 변환기 (166) 는 하나 이상의 역 컬러 변환들 중의 역 컬러 변환을 사용하여 제 1 컬러 공간을 갖는 재구성된 신호의 블록을 제 2, RGB 컬러 공간을 갖는 비디오 데이터의 제 2 블록으로 역으로 컬러 변환할 수도 있다. 일부 예들에서, 디코더 (31) 가 사용하는 특정의 역 변환은 비트스트림 (140) 에서 시그널링될 수도 있다. 역 적응적 컬러 변환기 (166) 는 장래의 저장 및 참조 화상으로서의 사용을 위해 참조 화상 메모리 (154) 로 뿐만아니라 이미지/비디오 (164) 로서의 출력을 위해 제 2 컬러 공간을 갖는 제 2 블록을 피딩할 수도 있다.

[0268] 도 10 은 본 개시의 기법들에 따른 비디오 데이터를 디코딩하는 방법의 예를 도시한다. 도 10 의 기법들은 일반 비디오 디코더에 대해 기술될 것이다. 일반 비디오 디코더는, 예를 들어, 도 7 의 비디오 디코더 (30) 또는 도 9 의 비디오 디코더 (31) 에 대응할 수도 있지만, 도 10 의 기법들은 비디오 디코더의 임의의 특정의 타입에 제한되지 않는다. 비디오 인코더들이 통상적으로 인코딩 프로세스의 부분으로서 비디오 디코딩을 수행하기 때문에, 도 10 의 기법들은 또한 도 6 의 비디오 인코더 (20) 및 도 8 의 비디오 인코더 (21) 와 같은 비디오 인코더에 의해 수행될 수도 있다. 비디오 인코더 (20) 는 예를 들어 도 10 의 기법들이 구현될 수도 있는, 디코딩 루프의 부분을 형성하는 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60) 을 포함한다. 따라서, 도 10 의 기법들이 비디오 디코더에 대해 설명될 것이지만, 이러한 비디오 디코더는 비디오 인코더의 부분일 수도 있다는 것이 이해되어야 한다.

[0269] 도 10 의 예에서, 비디오 디코더는 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정한다 (210). 일부 예들에서, 비디오 디코더는 적응적 컬러 변환이 가능하게 되는지 여부를 나타내는 신택스 엘리먼트를 수신함으로써 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정할 수도 있다. 신택스 엘리먼트는 예를 들어 PPS 에서 또는 다른 레벨에서 수신될 수도 있다. 수신된 신택스 엘리먼트를 파싱함으로써, 비디오 디코더는 적응적 컬러 변환이 가능하게 되는지 불가능하게 되는지 여부를 결정할 수 있다. 다른 예들에서, 비디오 디코더는 비디오 데이터에 대한 크로마 포맷을 결정함으로써 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정할 수도 있다. 예를 들어, 비디오 데이터에 대한 크로마 포맷이 4:4:4 라고 결정하는 것에 응답하여, 비디오 디코더는 적응적 컬러 변환이 가능하게 된다고 결정할 수도 있다. 비디오 데이터에 대한 크로마 포맷이 4:4:4 이 외라고 결정하는 것에 응답하여, 비디오 디코더는 적응적 컬러 변환이 불가능하게 된다고 결정할 수도 있다.

[0270] 가능하게 되는 적응적 컬러 변환을 갖는 비디오 데이터의 경우, 비디오 디코더는 하나 이상의 블록들에 대해 양자화 파라미터를 결정할 수도 있다 (212). 양자화 파라미터의 값이 임계값 아래에 있다는 것에 응답하여, 비디오 디코더는 변경된 양자화 파라미터를 결정하기 위해 양자화 파라미터를 변경할 수도 있다 (214). 임계값은 예를 들어 제로일 수도 있고, 변경된 양자화 파라미터의 값은 제로 이상일 수도 있다. 변경된 양자화 파라미터는 51 플러스 오프셋 값 이하일 수도 있다. 양자화 파라미터를 변경하기 위해, 비디오 디코더는 양자화 파라미터에 오프셋 값을 가산할 수도 있다. 비디오 디코더는 양자화 파라미터에 오프셋 값이 가산되어야 하는지 여부를 나타내는 플래그를 수신할 수도 있다.

[0271] 비디오 디코더는 예를 들어 신택스 엘리먼트로서 오프셋 값을 수신할 수도 있다. 오프셋 값은 (적응적 컬러 변환이 블록에 대해 가능하게 되는 경우) 양자화 파라미터에 대한 오프셋일 수도 있다. 비디오 디코더는 변경된 양자화 파라미터에 기초하여 변환 계수들을 역양자화할 수도 있다 (216).

[0272] 도 11 은 본 개시의 기법들에 따른 비디오 데이터를 인코딩하는 방법의 예를 도시한다. 도 11 의 기법들은 일반 비디오 인코더에 대해 기술될 것이다. 일반 비디오 인코더는, 예를 들어, 도 6 의 비디오 인코더 (20) 또는 도 8 의 비디오 인코더 (21) 에 대응할 수도 있지만, 도 11 의 기법들은 비디오 인코더의 임의의 특정의 타입에 제한되지 않는다. 비디오 인코더는 비디오 데이터에 대한 크로마 샘플링 포맷을 선택한다 (220). 제 1 크로마 샘플링 포맷인 크로마 샘플링 포맷에 응답하여, 비디오 인코더는 적응적 컬러 변환이 가능하게 되는지를 나타내는 신택스 엘리먼트를 생성한다 (222). 제 1 크로마 샘플링 포맷 이외인 크로마 샘플링 포맷에 응답하여, 비디오 인코더는 적응적 컬러 변환 없이 비디오 데이터를 인코딩한다 (224). 제 1 크로마

샘플링 포맷은 예를 들어 4:4:4 크로마 샘플링 포맷일 수도 있다.

- [0273] 도 12 는 본 개시의 기법들에 따른 비디오 데이터를 디코딩하는 방법의 예를 도시한다. 도 12 의 기법들은 일반 비디오 디코더에 대해 기술될 것이다. 일반 비디오 디코더는, 예를 들어, 도 7 의 비디오 디코더 (30) 또는 도 9 의 비디오 디코더 (31) 에 대응할 수도 있지만, 도 12 의 기법들은 비디오 디코더의 임의의 특성의 타입에 제한되지 않는다. 일반 비디오 디코더는 또한 일부 예들에서 비디오 인코더의 디코딩 루프에 대응할 수도 있다.
- [0274] 도 12 의 예에서, 비디오 디코더는, 비디오 데이터에 대한 크로마 샘플링 포맷에 기초하여, 적응적 컬러 변환이 비디오 데이터의 하나 이상의 블록들에 대해 가능하게 된다고 결정한다 (230). 비디오 디코더는 예를 들어 크로마 샘플링 포맷이 4:4:4 샘플링 포맷이라고 결정함으로써 적응적 컬러 변환이 비디오 데이터의 하나 이상의 블록들에 대해 가능하게 된다고 결정할 수도 있다. 비디오 디코더는 적응적 컬러 변환이 가능하게 된다고 결정하는 것에 기초하여 하나 이상의 블록들에 대한 양자화 파라미터를 결정할 수도 있고 (232), 결정된 양자화 파라미터에 기초하여 변환 계수들을 역양자화할 수도 있다 (234).
- [0275] 비디오 디코더는 또한 예를 들어 비디오 데이터의 하나 이상의 제 2 블록들에 대해 비디오 블록들에 대한 크로마 샘플링 포맷이 4:4:4 이외의 크로마 샘플링 포맷이라는 것을 결정하고, 4:4:4 이외인 크로마 샘플링 포맷에 기초하여, 제 2 의 하나 이상의 블록들에 대해 적응적 컬러 변환이 불가능하게 되는 것을 결정할 수도 있다. 비디오 디코더는, 예를 들어, 적응적 컬러 변환이 불가능하게 되는지 여부를 나타내기 위해, 크로마 샘플링 포맷의 표시 이외의 신택스 엘리먼트를 수신하지 않고, 제 2 의 하나 이상의 블록들에 대해 적응적 컬러 변환이 불가능하게 된다고 결정할 수도 있다.
- [0276] 도 13 은 본 개시의 기법들에 따른 비디오 데이터를 인코딩하는 방법의 예를 도시한다. 도 13 의 기법들은 일반 비디오 인코더에 대해 기술될 것이다. 일반 비디오 디코더는, 예를 들어, 도 6 의 비디오 인코더 (20) 또는 도 8 의 비디오 인코더 (21) 에 대응할 수도 있지만, 도 13 의 기법들은 비디오 인코더의 임의의 특성의 타입에 제한되지 않는다. 비디오 인코더는 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 그 블록들을 인코딩하는데 사용된다고 결정한다 (240). 비디오 인코더는 비디오 데이터의 제 1 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터를 결정한다 (242). 비디오 인코더는 그 양자화 파라미터에 기초하여 변환 계수들을 양자화한다 (244). 비디오 인코더는, 화상 파라미터 세트에의 포함을 위해, 비디오 데이터의 제 2 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터와 비디오 데이터의 제 2 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터 사이의 차이를 나타내는 하나 이상의 오프셋 값들을 생성한다 (246).
- [0277] 도 14 는 본 개시의 기법들에 따른 비디오 데이터를 디코딩하는 방법의 예를 도시한다. 도 14 의 기법들은 일반 비디오 디코더에 대해 기술될 것이다. 일반 비디오 디코더는, 예를 들어, 도 7 의 비디오 디코더 (30) 또는 도 9 의 비디오 디코더 (31) 에 대응할 수도 있지만, 도 14 의 기법들은 비디오 디코더의 임의의 특성의 타입에 제한되지 않는다. 도 14 의 예에서, 비디오 디코더는 비디오 데이터의 하나 이상의 블록들에 대해 적응적 컬러 변환이 가능하게 된다고 결정한다 (250). 적응적 컬러 변환이 가능하게 되는 것에 응답하여, 비디오 디코더는 화상 파라미터 세트에서 하나 이상의 오프셋 값들을 수신한다 (252). 비디오 디코더는 하나 이상의 오프셋 값들 중 제 1 오프셋 값에 기초하여 제 1 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터를 결정하고 (254), 변경된 양자화 파라미터에 기초하여 변환 계수들을 역양자화한다 (256). 하나 이상의 오프셋 값들은 제 1 컬러 컴포넌트에 대한 오프셋 값, 제 2 컬러 컴포넌트에 대한 오프셋 값, 및 제 3 컬러 컴포넌트에 대한 오프셋 값을 포함할 수도 있다.
- [0278] 비디오 디코더는 제 2 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터를 결정할 수도 있다. 하나 이상의 오프셋 값들 중 제 1 오프셋 값에 기초하여 제 1 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터를 결정하기 위해, 비디오 디코더는 제 2 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터에 하나 이상의 오프셋 값들 중 제 1 오프셋 값을 가산함으로써 제 2 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터를 제 1 컬러 공간의 제 1 컬러 컴포넌트에 대한 양자화 파라미터로 변환할 수도 있다.
- [0279] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 그 기능들은 컴퓨터 판독가능 매체 상에 하나 이상의 명령들 또는 코드로서 저장되거나 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 예를 들면, 통신 프로토콜에 따라, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체 또는 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독

가능 저장 매체를 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체는 일반적으로, (1) 비밀시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는, 본 개시에서 설명된 기법들의 구현을 위해 명령들, 코드 및/또는 데이터 구조들을 추출하기 위하여 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0280] 비한정적 예로서, 그러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령 또는 데이터 구조의 형태로 원하는 프로그램 코드를 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 라인 (DSL), 또는 적외선, 전파 (radio), 및 마이크로파와 같은 무선 기술을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되면, 그 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 전파, 및 마이크로파와 같은 무선 기술은 매체의 정의 내에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속, 캐리어 파, 신호 또는 다른 일시적 매체를 포함하는 것이 아니라, 대신에 비밀시적, 유형의 저장 매체를 포함한다는 것이 이해되어야 한다. 여기에 사용된, 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, DVD (digital versatile disc), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서, 디스크 (disk) 는 보통 데이터를 자기적으로 재생하지만, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 또한, 상기의 조합은 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

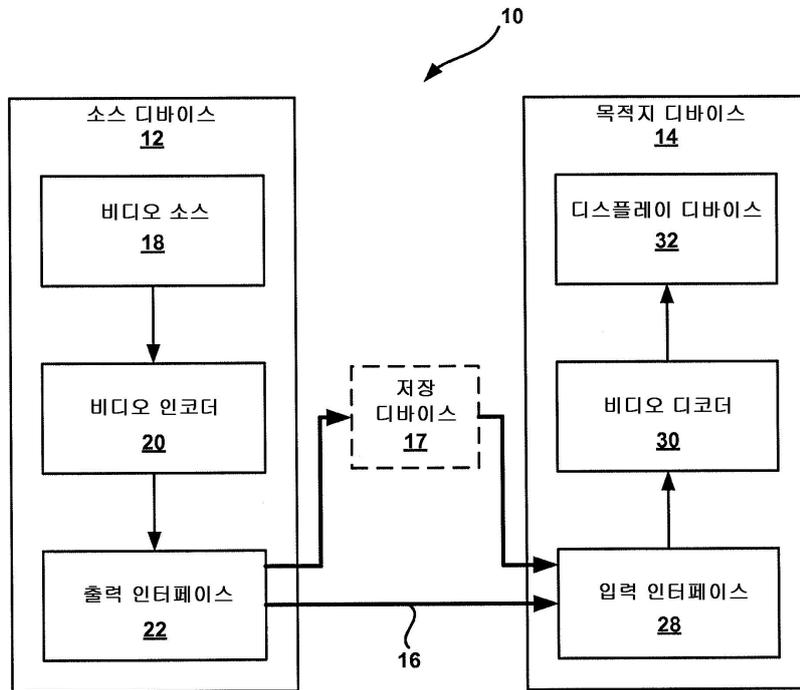
[0281] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 DSP 들, 범용 마이크로프로세서들, ASIC 들, FPGA 들, 또는 다른 동등한 집적 또는 이산 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에 사용된 용어 "프로세서" 는 전술한 구조 중 임의의 것 또는 본원에 설명된 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 추가로, 일부 양태들에서, 여기서 설명된 기능은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나 또는 결합된 코텍에 포함될 수도 있다. 또한, 그 기법들은 하나 이상의 회로 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0282] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들면, 칩 세트) 를 포함하여, 광범위하게 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들 또는 유닛들이, 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적인 양태들을 강조하기 위하여 본 개시에 설명되었지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하는 것은 아니다. 오히려, 상술된 바처럼, 다양한 유닛들이 코텍 하드웨어 유닛에 결합될 수도 있거나, 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상술된 하나 이상의 프로세서들을 포함하는 연동적인 (interoperative) 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

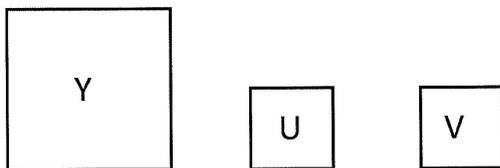
[0283] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

도면1

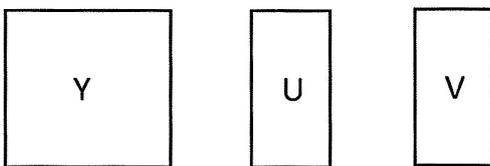


도면2a



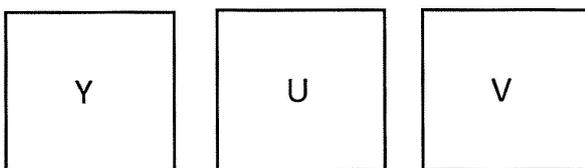
4:2:0 샘플 포맷

도면2b



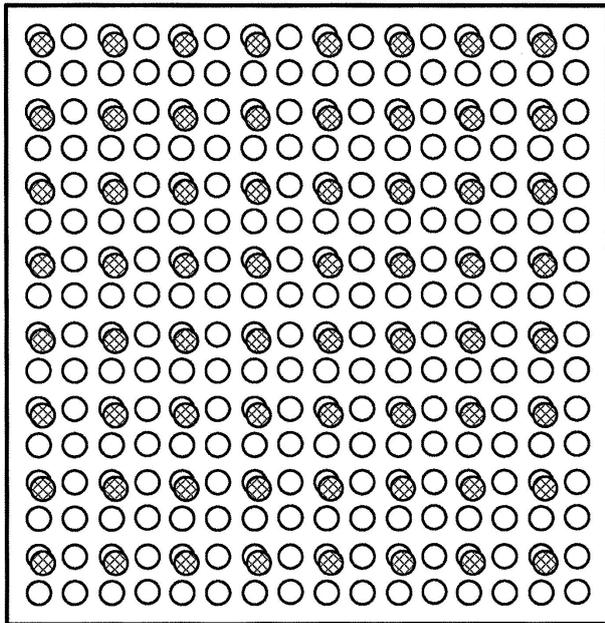
4:2:2 샘플 포맷

도면2c



4:4:4 샘플 포맷

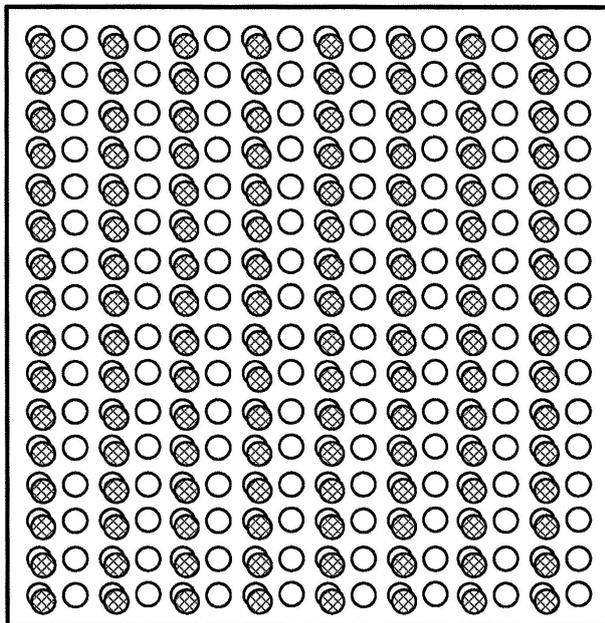
도면3



4:2:0 샘플 포매팅을 갖는 16 x 16 CU

- 루마 샘플
- ⊗ 크로마 샘플들

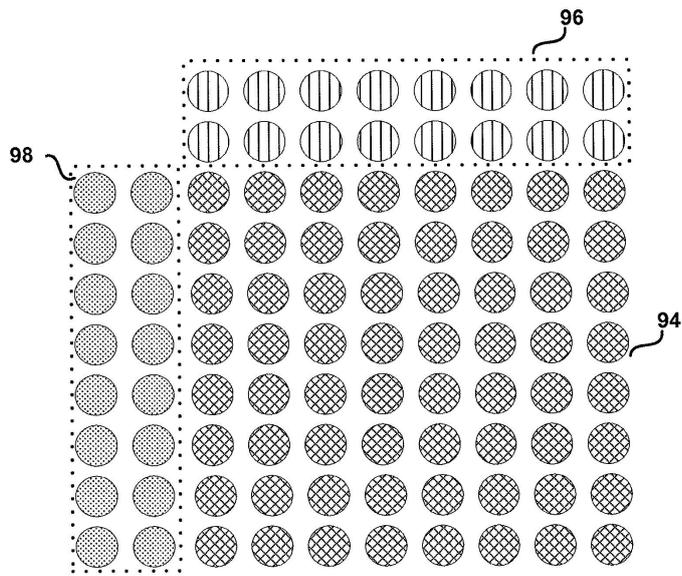
도면4



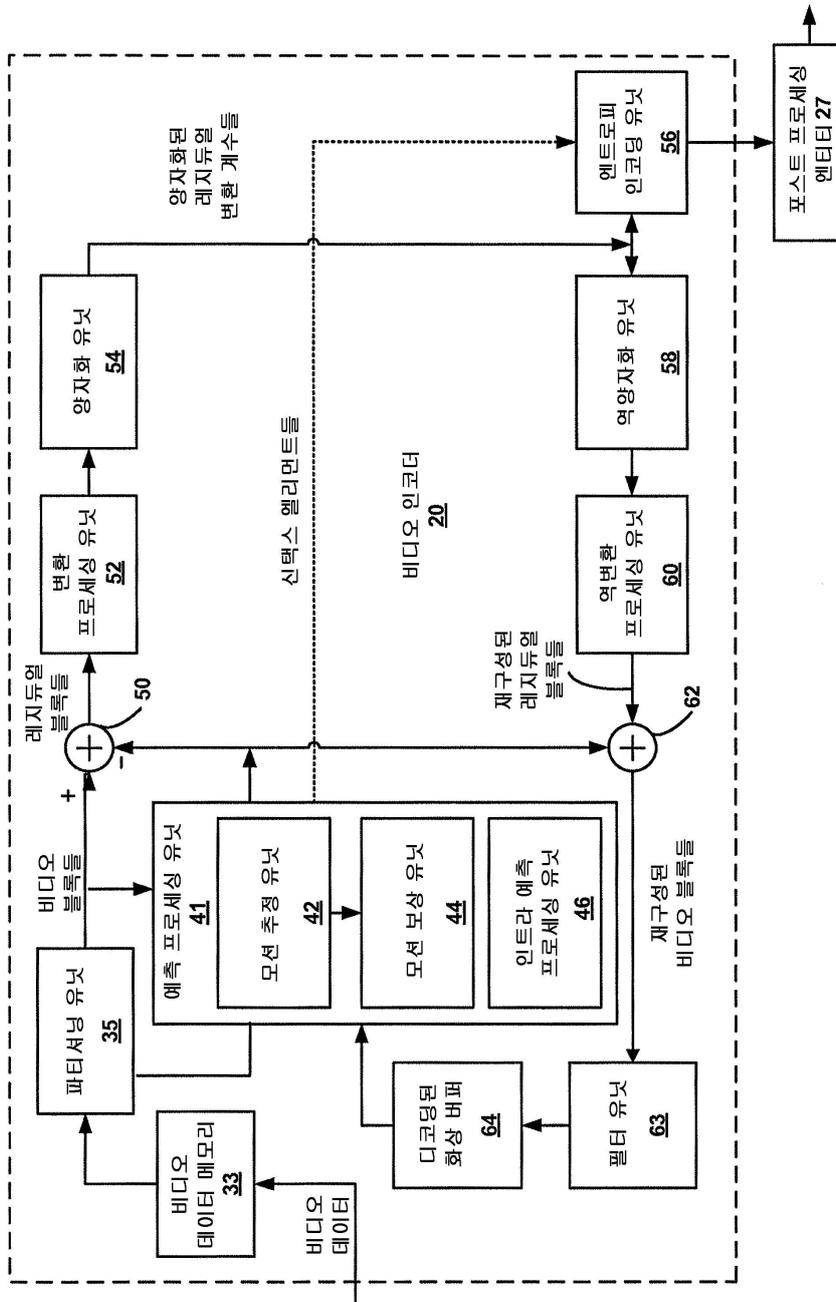
4:2:2 샘플 포매팅을 갖는 16 x 16 CU

- 루마 샘플
- ⊗ 크로마 샘플들

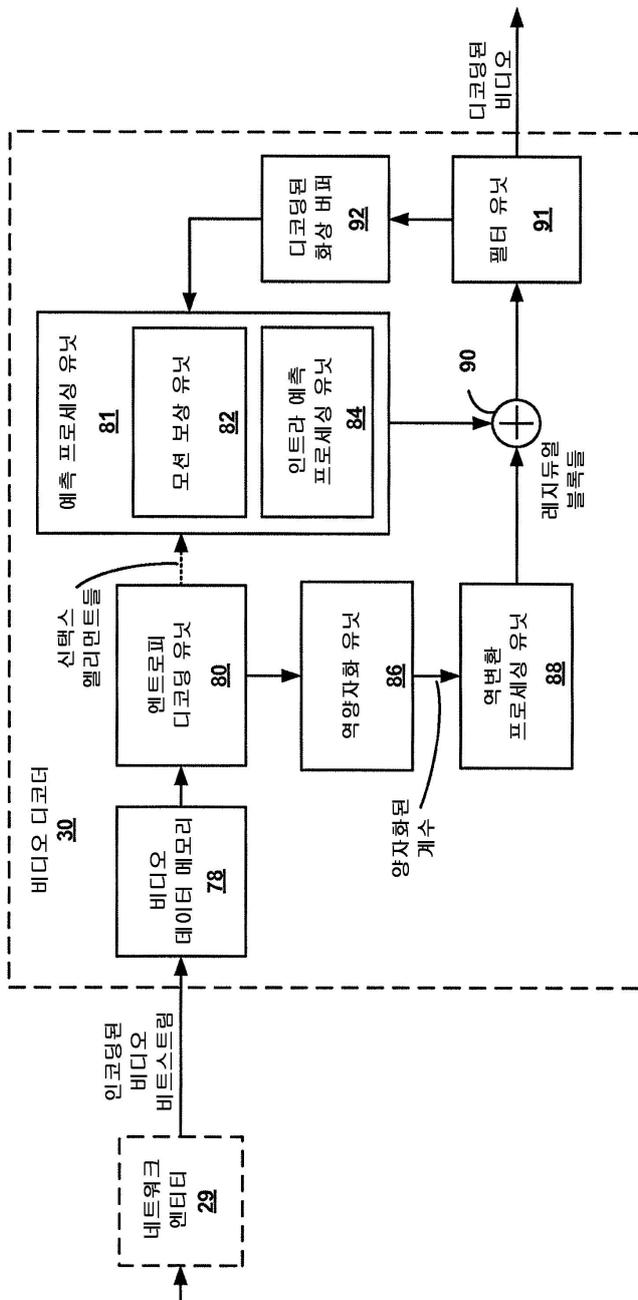
도면5



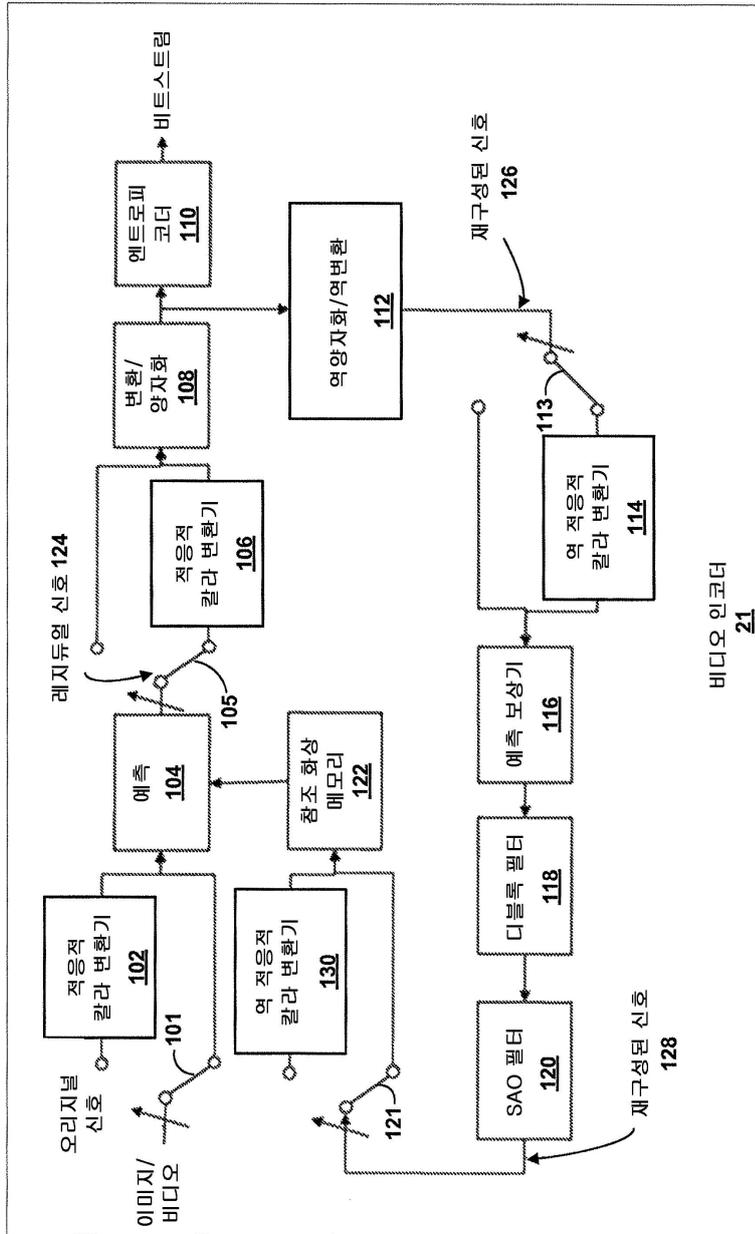
도면6



도면7

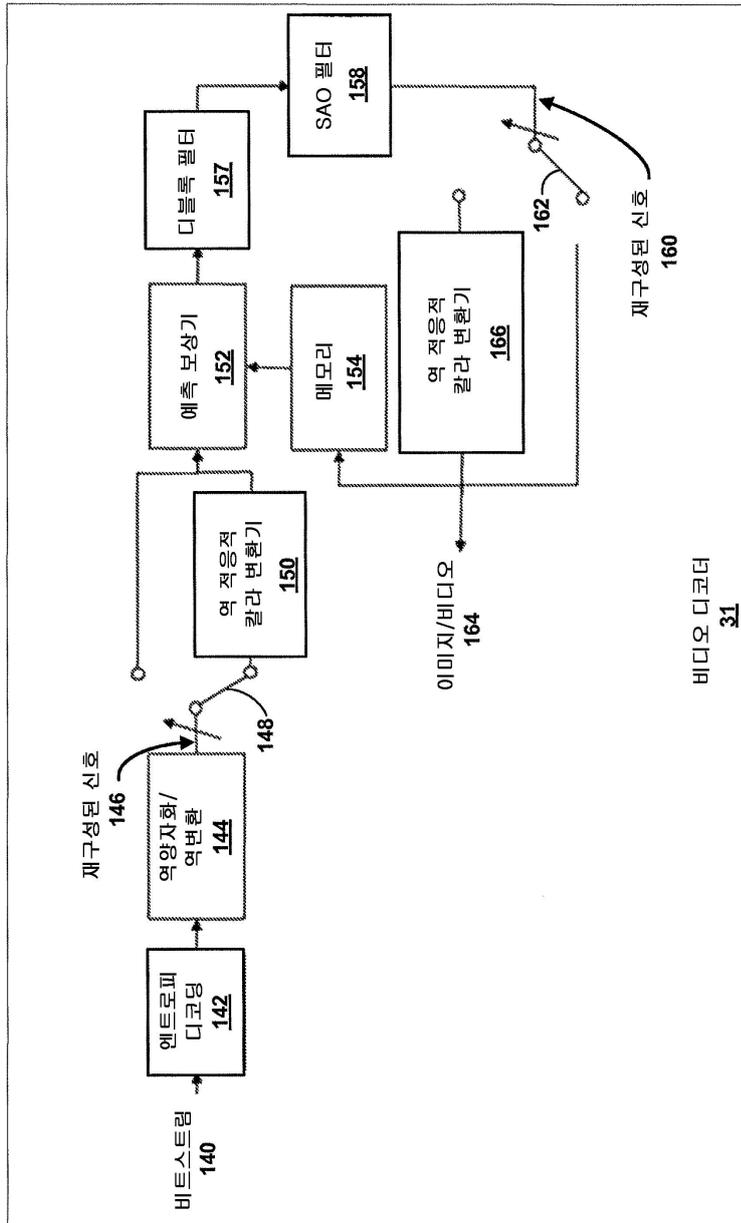


도면8

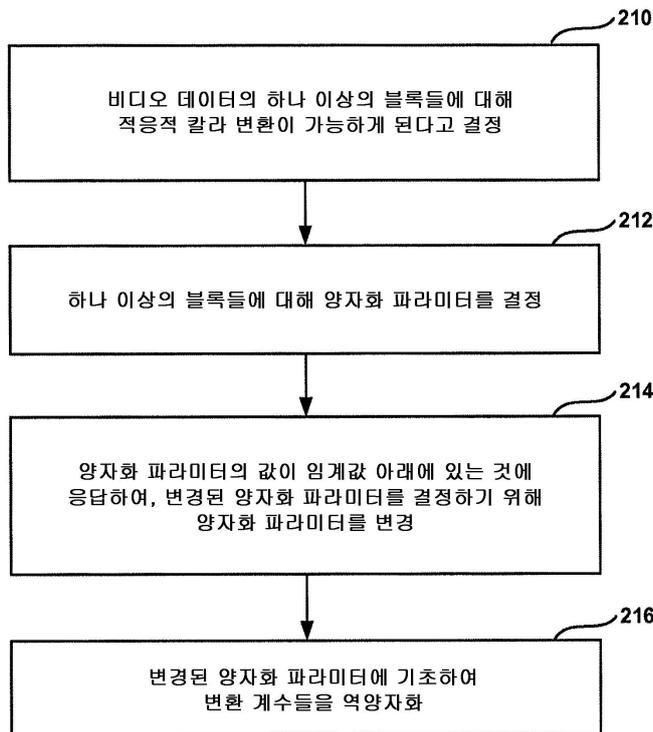


비디오 인코더
21

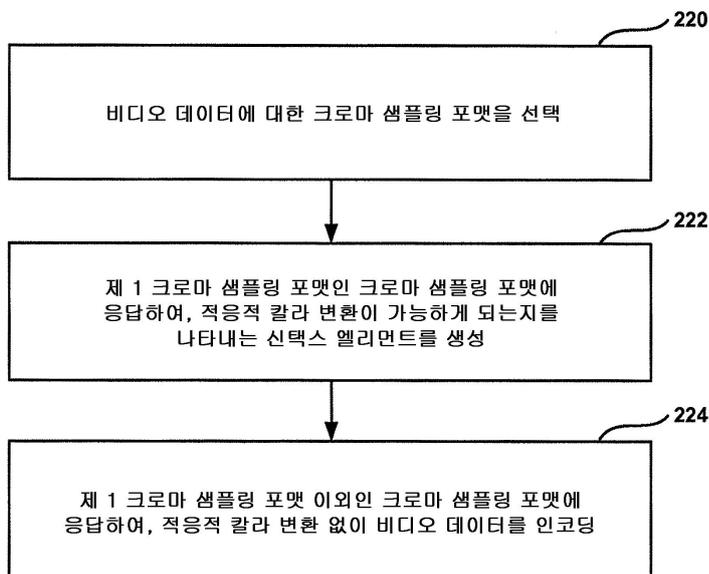
도면9



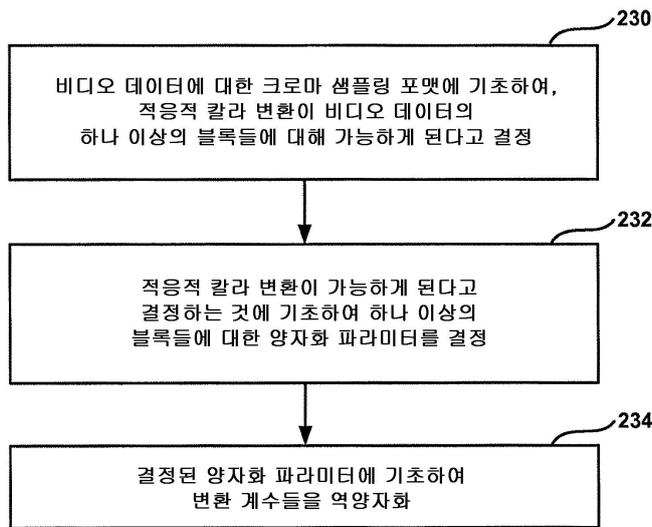
도면10



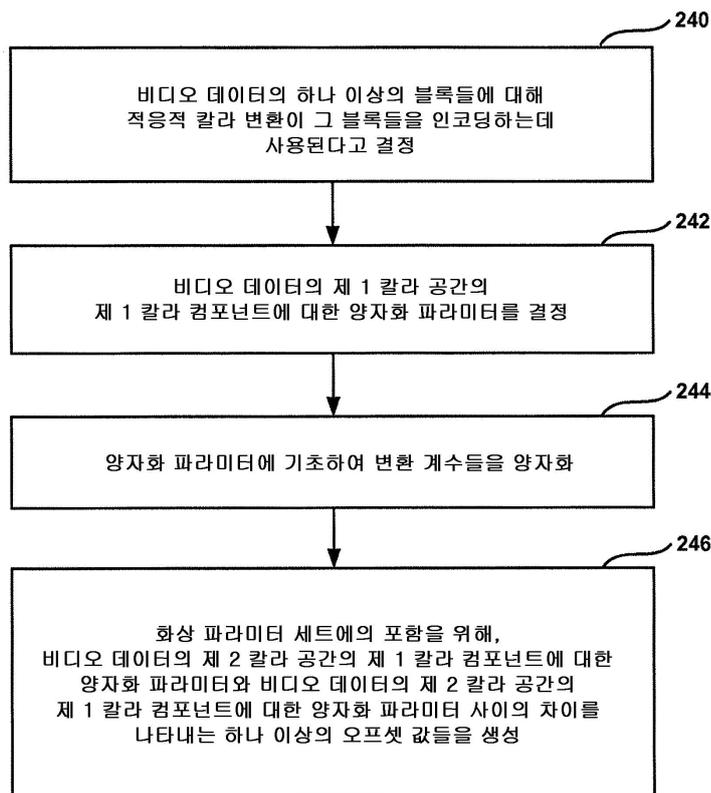
도면11



도면12



도면13



도면14

