



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0134375
(43) 공개일자 2021년11월09일

- (51) 국제특허분류(Int. Cl.)
H04N 19/186 (2014.01) H04N 19/117 (2014.01)
H04N 19/124 (2014.01) H04N 19/14 (2014.01)
H04N 19/176 (2014.01) H04N 19/593 (2014.01)
H04N 19/61 (2014.01) H04N 19/70 (2014.01)
H04N 19/98 (2014.01)
- (52) CPC특허분류
H04N 19/186 (2015.01)
H04N 19/117 (2015.01)
- (21) 출원번호 10-2021-7031818
(22) 출원일자(국제) 2020년02월28일
심사청구일자 없음
(85) 번역문제출일자 2021년10월01일
(86) 국제출원번호 PCT/US2020/020513
(87) 국제공개번호 WO 2020/180737
국제공개일자 2020년09월10일
(30) 우선권주장
62/813,728 2019년03월04일 미국(US)
62/817,546 2019년03월12일 미국(US)
- (71) 출원인
알리바바 그룹 홀딩 리미티드
케이만군도, 그랜드 케이만, 피오박스 847, 원 캐
피탈 플레이스 4층
(72) 발명자
예, 안
미국 캘리포니아 94402 샌 마테오 스위트 400 사
우스 엘 카미노 리얼 400 알리바바 그룹 리갈 디
파트먼트
첸, 지에
미국 캘리포니아 94402 샌 마테오 스위트 400 사
우스 엘 카미노 리얼 400 알리바바 그룹 리갈 디
파트먼트
랴오, 룰링
미국 캘리포니아 94402 샌 마테오 스위트 400 사
우스 엘 카미노 리얼 400 알리바바 그룹 리갈 디
파트먼트
(74) 대리인
특허법인 광장리앤코

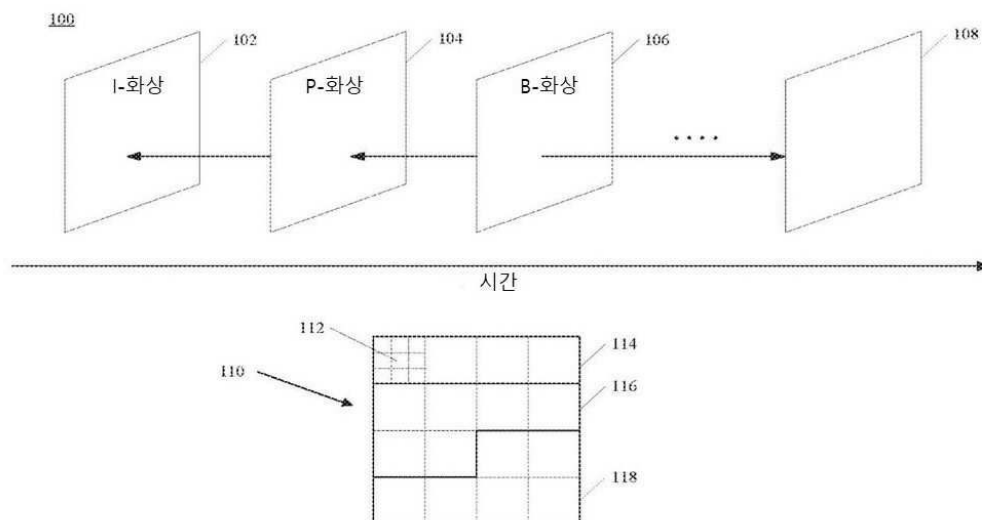
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 비디오 콘텐츠를 처리하기 위한 방법 및 시스템

(57) 요약

본 개시의 실시예들은 비디오 콘텐츠를 처리하기 위한 시스템 및 방법을 제공한다. 상기 방법은, 화상과 연관된 크롬 블록 및 루마 블록을 수신하는 단계; 루마 블록과 연관된 루마 스케일링 정보를 결정하는 단계; 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하는 단계; 및 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

H04N 19/124 (2015.01)

H04N 19/14 (2015.01)

H04N 19/176 (2015.01)

H04N 19/593 (2015.01)

H04N 19/61 (2015.01)

H04N 19/70 (2015.01)

H04N 19/98 (2015.01)

명세서

청구범위

청구항 1

비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법으로서,
 화상(picture)과 연관된 크롬 블록과 루마(luma) 블록을 수신하는 단계;
 상기 루마 블록과 연관된 루마 스케일링 정보를 결정하는 단계;
 상기 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하는 단계; 및
 상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리하는 단계를 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 2

제1항에 있어서,
 상기 루마 스케일링 정보에 기초하여 상기 크로마 스케일링 인자를 결정하는 단계는,
 상기 루마 스케일링 정보에 기초하여 상기 루마 블록의 루마 스케일링 인자를 결정하는 단계;
 상기 루마 스케일링 인자의 값에 기초하여 상기 크로마 스케일링 인자를 결정하는 단계를 더 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 3

제2항에 있어서,
 상기 루마 스케일링 인자의 값에 기초하여 상기 크로마 스케일링 인자를 결정하는 단계는,
 상기 크로마 스케일링 인자를 상기 루마 스케일링 인자의 값과 동일하게 설정하는 단계를 더 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 4

제1항에 있어서,
 상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리하는 단계는,
 제1 조건이 충족되는지 결정하는 단계; 및
 상기 제1 조건이 충족된다는 결정에 응답하여, 상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리하는 것; 또는
 상기 제1 조건이 충족되지 않는다는 결정에 응답하여, 상기 크로마 스케일링 인자를 이용한 상기 크로마 블록의 처리를 우회(bypass)하는 것
 중 하나를 수행하는 단계를 더 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 5

제4항에 있어서,
 상기 제1 조건은,
 비-제로 잔차(non-zero residuals)를 가지지 않는 상기 화상과 연관된 타깃 코딩 유닛; 또는
 비-제로 크로마 잔차(non-zero chroma residuals)를 가지지 않는 상기 화상과 연관된 타깃 변환 유닛 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 6

제5항에 있어서,

비-제로 잔차를 가지지 않는 상기 타깃 코딩 유닛은 상기 타깃 코딩 유닛의 제1 코딩된 블록 플래그의 값에 기초하여 결정되고, 및

비-제로 크로마 잔차를 가지지 않는 상기 타깃 변환 유닛은 제1 크로마 성분에 대한 제2 코딩된 블록 플래그 및 상기 타깃 변환 유닛의 제2 루마 크로마 성분에 대한 제3 코딩된 블록 플래그의 값에 기초하여 결정되는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 7

제6항에 있어서,

상기 제1 코딩된 블록 플래그의 값은 0이고; 및

상기 제2 코딩된 블록 플래그 및 상기 제3 코딩된 블록 플래그의 값은 0인, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 8

제1항에 있어서,

상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리하는 단계는,

상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록의 잔차를 처리하는 단계를 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

청구항 9

비디오 콘텐츠 처리 장치로서,

명령어의 세트를 저장하는 메모리; 및

상기 메모리에 커플링되고 상기 장치로 하여금,

화상과 연관된 크롬 블록과 루마 블록을 수신하고;

상기 루마 블록과 연관된 루마 스케일링 정보를 결정하고;

상기 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하고; 및

상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리하는 것을 수행하게 하는 상기 명령어의 세트를 실행하도록 구성되는 프로세서를 포함하는, 비디오 콘텐츠 처리 장치.

청구항 10

제9항에 있어서,

상기 루마 스케일링 정보에 기초하여 상기 크로마 스케일링 인자를 결정함에 있어서, 상기 프로세서는 상기 장치로 하여금,

상기 루마 스케일링 정보에 기초하여 상기 루마 블록의 루마 스케일링 인자를 결정하고;

상기 루마 스케일링 인자의 값에 기초하여 상기 크로마 스케일링 인자를 결정하는 것을 더 수행하게 하는 상기 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

청구항 11

제10항에 있어서,

상기 루마 스케일링 인자의 값에 기초하여 상기 크로마 스케일링 인자를 결정함에 있어서, 상기 프로세서는 상기 장치로 하여금,

상기 크로마 스케일링 인자를 상기 루마 스케일링 인자의 값과 동일하게 설정하는 것을 더 수행하게 하는 상기 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

청구항 12

제9항에 있어서,

상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리함에 있어서, 상기 프로세서는 상기 장치로 하여금,

상기 제1 조건이 충족되는지 결정하고; 및

상기 제2 조건이 충족된다는 결정에 응답하여, 상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리하는 것; 또는

상기 제2 조건이 충족되지 않는다는 결정에 응답하여, 상기 크로마 스케일링 인자를 이용한 상기 크로마 블록의 처리를 우회하는 것

중 하나를 수행하는 것을 더 수행하게 하는 상기 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

청구항 13

제12항에 있어서,

상기 제1 조건은,

비-제로 잔차를 가지지 않는 상기 화상과 연관된 타깃 코딩 유닛; 또는

비-제로 크로마 잔차를 가지지 않는 상기 화상과 연관된 타깃 변환 유닛을 포함하는, 비디오 콘텐츠 처리 장치.

청구항 14

제13항에 있어서,

비-제로 잔차를 가지지 않는 상기 타깃 코딩 유닛은 상기 타깃 코딩 유닛의 제1 코딩된 블록 플래그의 값에 기초하여 결정되고, 및

비-제로 크로마 잔차를 가지지 않는 상기 타깃 변환 유닛은 제1 크로마 성분에 대한 제2 코딩된 블록 플래그 및 상기 타깃 변환 유닛의 제2 크로마 성분에 대한 제3 코딩된 블록 플래그의 값에 기초하여 결정되는, 비디오 콘텐츠 처리 장치.

청구항 15

제14항에 있어서,

상기 제1 코딩된 블록 플래그의 값이 0이고,

상기 제2 코딩된 블록 플래그 및 상기 제3 코딩된 블록 플래그의 값이 0인, 비디오 콘텐츠 처리 장치.

청구항 16

제9항에 있어서,

상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리함에 있어서, 상기 프로세서는 상기 장치로 하여금,

상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록의 잔차를 처리하는 것을 더 수행하게 하는 상기 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

청구항 17

명령어의 세트를 저장하는 비-일시적 컴퓨터-판독 가능 저장 매체로서, 상기 명령어 세트는 디바이스가 비디오 콘텐츠를 처리하기 위한 방법을 수행하도록 상기 디바이스의 하나 이상의 프로세서에 의해 실행될 수 있으며,

상기 방법은,

화상과 연관된 크롬 블록과 루마 블록을 수신하는 단계;

상기 루마 블록과 연관된 루마 스케일링 정보를 결정하는 단계;

상기 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하는 단계; 및

상기 크로마 스케일링 인자를 이용하여 상기 크로마 블록을 처리하는 단계를 포함하는, 비-일시적 컴퓨터-판독 가능 저장 매체.

발명의 설명

기술 분야

- [0001] 본 개시는 2019년 3월 4일에 출원된 미국 가출원 번호 제62/813,728호 및 2019년 3월 12일에 출원된 미국 가출원 번호 제62/817,546호에 대한 우선권을 주장하며, 두 문헌 모두 전체가 참조로서 본 명세서에 포함된다.
- [0002] 본 개시는 일반적으로 비디오 처리, 특히 크로마 스케일링(chroma scaling)으로 인-루프 루마 매핑(in-loop luma mapping)을 수행하는 방법 및 시스템에 관한 것이다.

배경 기술

- [0003] 비디오 코딩 시스템은 디지털 비디오 신호를 압축하기 위해, 예를 들어 소비되는 저장 공간을 감소시키거나 이러한 신호와 연관되는 전송 대역폭 소비를 감소시키기 위해 종종 사용된다. 온라인 비디오 스트리밍, 비디오 컨버전싱, 또는 비디오 모니터링과 같은 비디오 압축의 다양한 어플리케이션에서 인기를 얻고 있는 HD(high-definition) 비디오(예를 들어, 해상도가 1920x1080 픽셀인 비디오)의 경우, 비디오 데이터의 압축 효율을 증가시킬 수 있는 비디오 코딩 툴 개발에 대한 필요성이 지속적으로 존재한다.
- [0004] 예를 들어, 많은 어플리케이션 시나리오(예를 들어, 보안, 교통, 환경 모니터링 등)에서 비디오 모니터링 어플리케이션이 점점 더 광범위하게 사용되고, 모니터링 디바이스의 개수 및 해상도는 계속해서 빠르게 증가하고 있다. 많은 비디오 모니터링 어플리케이션 시나리오에서, 더 많은 정보를 캡처하고 이러한 정보를 캡처하기 위해 프레임 당 픽셀 수가 더 많은 HD 비디오를 사용자에게 제공하는 것이 선호된다. 하지만, HD 비디오 비트스트림은 높은 비트율을 가질 수 있고, 이는 높은 전송 대역폭 및 큰 저장 공간을 요구한다. 예를 들어, 평균 1920x1080의 해상도를 가지는 모니터링 비디오 스트림은 실시간 전송을 위해 4Mbps만큼 높은 대역폭을 요구할 수 있다. 또한, 비디오 모니터링은 일반적으로 7x24 연속적으로 모니터링하기 때문에, 비디오 데이터를 저장하고자 할 경우, 저장 시스템에 큰 문제를 야기할 수 있다. 따라서, HD 비디오의 높은 대역폭 및 큰 저장 공간(storage)에 대한 요구는 비디오 모니터링의 대규모 배포(deployment)에 있어서 주요한 제한 사항이 되어 왔다.

발명의 내용

- [0005] 본 개시의 실시예들은 비디오 콘텐츠를 처리하는 방법을 제공한다. 방법은, 화상과 연관된 크롬 블록(chrome block)과 루마 블록(luma block)을 수신하는 단계; 루마 블록과 연관된 루마 스케일링 정보(luma scaling information)를 결정하는 단계; 루마 스케일링 정보에 기초하여 크로마 스케일링 인자(chroma scaling factor)를 결정하는 단계; 및 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계를 포함할 수 있다.
- [0006] 본 개시의 실시예들은 비디오 콘텐츠를 처리하는 장치를 제공한다. 장치는, 명령어의 세트를 저장하는 메모리; 및 메모리에 커풀링되는 프로세서를 포함할 수 있고, 프로세서는 장치로 하여금, 화상과 연관된 크롬 블록과 루마 블록을 수신하고; 루마 블록과 연관된 루마 스케일링 정보를 결정하고; 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하고; 및 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 것을 수행하게 하는 명령어의 세트를 실행하도록 구성된다.
- [0007] 본 개시의 실시예들은 디바이스가 비디오 콘텐츠를 처리하는 방법을 수행하도록, 디바이스의 하나 이상의 프로세서에 의해 실행될 수 있는 명령어의 세트를 저장하는 비-일시적 컴퓨터-판독 가능한 저장 매체를 제공한다. 방법은, 화상과 연관된 크롬 블록과 루마 블록을 수신하는 단계; 루마 블록과 연관된 루마 스케일링 정보를 결정하는 단계; 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하는 단계; 및 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계를 포함한다.

도면의 간단한 설명

[0008]

본 개시의 실시예들 및 다양한 양태들은 다음의 상세한 설명 및 첨부되는 도면에 나타나 있다. 도면에 도시되는 다양한 특징들은 축척에 따라 도시된 것은 아니다.

도 1은 본 개시의 일부 실시예에 따른 예시적인 비디오 시퀀스의 구조를 나타낸다.

도 2a는 본 개시의 일부 실시예에 따른 예시적인 인코딩 프로세스의 개략도를 나타낸다.

도 2b는 본 개시의 일부 실시예에 따른 다른 예시적인 인코딩 프로세스의 개략도를 나타낸다.

도 3a는 본 개시의 일부 실시예에 따른 예시적인 디코딩 프로세스의 개략도를 나타낸다.

도 3b는 본 개시의 일부 실시예에 따른 다른 예시적인 디코딩 프로세스의 개략도를 나타낸다.

도 4는 본 개시의 일부 실시예에 따른, 비디오를 인코딩 또는 디코딩하기 위한 예시적인 장치의 블록도를 나타낸다.

도 5는 본 개시의 일부 실시예에 따른, 예시적인 LMCS(luma mapping with chroma scaling) 프로세스의 개략도를 나타낸다.

도 6은 본 개시의 일부 실시예에 따른, LMCS 구분적 선형 모델에 대한 타일 그룹 수준 선택스 표를 나타낸다.

도 7은 본 개시의 일부 실시예에 따른, LMCS 구분적 선형 모델에 대한 다른 타일 그룹 수준 선택스 표를 나타낸다.

도 8은 본 개시의 일부 실시예에 따른 코딩 트리 유닛 선택스 구조의 표이다.

도 9는 본 개시의 일부 실시예에 따른 듀얼 트리 파티션 선택스 구조의 표이다.

도 10은 본 개시의 일부 실시예에 따른, 루마 예측 블록의 평균화의 예시적인 단순화를 나타낸다.

도 11은 본 개시의 일부 실시예에 따른 코딩 트리 유닛 선택스 구조의 표이다.

도 12는 본 개시의 일부 실시예에 따른, 타일 그룹 수준에서 LMCS 구분적 선형 모델의 수정된 시그널링을 위한 선택스 요소들의 표이다.

도 13은 본 개시의 일부 실시예에 따른, 비디오 콘텐츠를 처리하기 위한 방법의 흐름도이다.

도 14는 본 개시의 일부 실시예에 따른, 비디오 콘텐츠를 처리하기 위한 방법의 흐름도이다.

도 15는 본 개시의 일부 실시예에 따른, 비디오 콘텐츠를 처리하기 위한 다른 방법의 흐름도이다.

도 16은 본 개시의 일부 실시예에 따른, 비디오 콘텐츠를 처리하기 위한 다른 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0009]

이제 예시적인 실시예들이 상세하게 참조될 것이며, 그 예시들은 첨부되는 도면에 나타나 있다. 이하의 설명은 첨부되는 도면을 참조하며, 상이한 도면에서 동일한 번호는 달리 제시하지 않는 한, 동일하거나 유사한 요소를 나타낸다. 이하의 예시적인 실시예들에 대한 설명에서 기술되는 구현예들은 본 발명과 일치하는 모든 구현예를 나타내는 것은 아니다. 이들은 오히려 첨부되는 청구항에 기술되는 것과 같은 본 발명과 관련된 양태들과 일치하는 장치와 방법의 예시일 뿐이다. 달리 명시되지 않는 한, “또는”이라는 용어는 실행 불가능할 경우를 제외하고 모든 가능한 조합들을 포함한다. 예를 들어, 만약에 구성 요소가 A 또는 B를 포함할 수 있다고 명시되어 있는 경우, 달리 특정하게 명시되어 있거나 불가능하지 않는 한, 그 구성 요소는 A, 또는 B, 또는 A와 B를 포함할 수 있다. 두 번째 예시로서, 만약 구성 요소가 A, B, 또는 C를 포함할 수 있다고 명시되어 있는 경우, 달리 특정하게 명시되어 있거나 불가능하지 않는 한, 그 컴포넌트는 A, 또는 B, 또는 C, 또는 A와 B, 또는 A와 C, 또는 B와 C, 또는 A 및 B 및 C를 포함할 수 있다.

[0010]

비디오는 시각 정보를 저장하기 위해 시간적 시퀀스로 배열되는 정적 화상(또는 “프레임”)의 세트이다. 비디오 캡처 디바이스(예를 들어, 카메라)는 시간적 시퀀스에서의 이러한 화상들을 캡처하고 저장하는 데에 사용될 수 있으며, 비디오 재생 디바이스(예를 들어, 텔레비전, 컴퓨터, 스마트폰, 태블릿 컴퓨터, 비디오 플레이어, 또는 디스플레이 기능을 갖는 임의의 최종-사용자 단말기)는 시간적 시퀀스에서의 이러한 화상들을 디스플레이하는 데에 사용될 수 있다. 또한, 일부 어플리케이션에서, 비디오 캡처링 디바이스는 예컨대 모니터링, 컨퍼런

싱, 또는 라이브 방송을 위해 캡처된 비디오를 실시간으로 비디오 재생 디바이스(예를 들어, 모니터를 갖는 컴퓨터)로 전송할 수 있다.

[0011] 이러한 어플리케이션에 의해 요구되는 저장 공간과 전송 대역폭을 감소시키기 위해, 비디오는 저장 및 전송 전에 압축될 수 있고 디스플레이 전에 압축 해제(decompress)될 수 있다. 압축 및 압축 해제는 프로세서(예를 들어, 일반 컴퓨터의 프로세서) 또는 특화된 하드웨어에 의해 실행되는 소프트웨어에 의해 구현될 수 있다. 압축을 위한 모듈은 일반적으로 “인코더”로 지칭되고, 압축 해제를 위한 모듈은 일반적으로 “디코더”로 지칭된다. 인코더와 디코더는 집합적으로 “코덱(codec)”으로 지칭될 수 있다. 인코더와 디코더는 다양한 적절한 하드웨어, 소프트웨어, 또는 이들의 조합 중 임의의 것으로 구현될 수 있다. 예를 들어, 인코더와 디코더의 하드웨어 구현은 하나 이상의 마이크로 프로세서, 디지털 신호 프로세서(DSPs), 주문형 집적 회로(ASICs), 필드-프로그램 가능한 게이트 어레이(FPGAs), 이산 논리, 또는 이들의 임의의 조합과 같은 회로망을 포함할 수 있다. 인코더와 디코더의 소프트웨어 구현은 프로그램 코드, 컴퓨터-실행 가능한 명령어, 펌웨어, 또는 컴퓨터-판독 가능한 매체에 고정되는 임의의 적절한 컴퓨터-구현되는 알고리즘이나 프로세스를 포함할 수 있다. 비디오 압축과 압축 해제는 MPEG-1, MPEG-2, MPEG-4, H.26x 시리즈 등과 같은 다양한 알고리즘이나 표준에 의해 구현될 수 있다. 일부 어플리케이션에서, 코덱은 제1 코딩 표준으로부터의 비디오를 압축 해제하고, 압축 해제된 비디오를 제2 코딩 표준을 이용하여 재-압축할 수 있으며, 이 경우 이러한 코덱은 “트랜스코더”로 지칭될 수 있다.

[0012] 비디오 인코딩 프로세스는 화상을 재구성(reconstruct)하는 데에 사용될 수 있는 유용한 정보를 식별 및 유지하고, 재구성에 중요하지 않은 정보는 무시할 수 있다. 무시되는 중요하지 않은 정보가 완전히 재구성될 수 없는 경우, 이러한 인코딩 프로세스는 “손실 압축(lossy)”으로 지칭될 수 있다. 그렇지 않을 경우에는, “무손실 압축(lossless)”으로 지칭될 수 있다. 대부분의 인코딩 프로세스는 손실 압축인데, 이는 요구되는 저장 공간과 전송 대역폭을 감소시키기 위한 절충안(tradeoff)이다.

[0013] 인코딩되는 화상(“현재 화상”으로 지칭됨)의 유용한 정보는 참조 화상(reference picture)(예를 들어, 이전에 인코딩되어 재구성된 화상)에 대한 변화를 포함한다. 이러한 변화는 픽셀의 위치 변화, 광도 변화, 또는 색상 변화를 포함할 수 있으며, 이들 중 위치 변화가 가장 중시된다. 객체를 표현하는 픽셀 그룹의 위치 변화는 참조 화상과 현재 화상 사이의 객체의 모션을 반영할 수 있다.

[0014] 다른 화상을 참조 하지 않고 코딩되는 화상(즉, 이 화상은 그 자신의 참조 화상임)은 “I-화상”으로 지칭된다. 참조 화상으로서 이전 화상을 이용하여 코딩되는 화상은 “P-화상”으로 지칭된다. 참조 화상으로서 이전 화상과 미래 화상 모두를 이용하여 코딩되는 화상(즉, 참조가 “양-방향”임)은 “B-화상”으로 지칭된다.

[0015] 전술한 바와 같이, HD 비디오를 이용한 비디오 모니터링은 높은 대역폭과 큰 저장 공간이 요구된다는 문제에 직면한다. 이러한 문제를 해결하기 위해, 인코딩되는 비디오의 비트율(bitrate)이 감축될 수 있다. I-화상, P-화상, 및 B-화상 중에서는 I-화상의 비트율이 가장 높다. 대부분의 모니터링 비디오의 배경은 거의 정적이기 때문에, 인코딩되는 비디오의 전체 비트율을 감축하는 하나의 방법은 비디오 인코딩을 위해 I-화상을 더 적게 사용하는 것일 수 있다.

[0016] 하지만, 더 적은 I-화상을 이용한다는 개선책은 효과가 미미할 수 있는데, 이는 통상적으로 인코딩되는 비디오에서 I-화상이 지배적이지 않기 때문이다. 예를 들어, 통상적인 비디오 비트스트림에서 I-화상, B-화상 및 P-화상의 비율은 1:20:9일 수 있으며, 이때 I-화상은 총 비트율의 10% 미만만을 차지할 수 있다. 환언하면, 이러한 예시에서, 모든 I-화상이 제거된다 하더라도, 감축되는 비트율은 10%이하일 수 있다.

[0017] 본 개시는 비디오 모니터링에 대한 특징-기반 비디오 처리를 위한 방법, 장치, 및 시스템을 제공한다. 본 명세서에서 “특징(characteristic)”이라는 용어는 화상 내 비디오 콘텐츠와 연관된 콘텐츠 특징, 화상을 인코딩 또는 디코딩하는 모션 추정과 연관된 모션 특징, 또는 이 둘 모두를 지칭한다. 예를 들어, 콘텐츠 특성은 비디오 시퀀스의 하나 이상의 연속적 화상 내 픽셀들일 수 있고, 픽셀들은 화상 내 객체, 장면, 또는 환경적 이벤트 중 적어도 하나와 관련된다. 다른 예시로서, 모션 특징은 비디오 코딩 프로세스와 관련된 정보를 포함할 수 있으며, 그 예시는 추후에 상세하게 설명될 것이다.

[0018] 본 개시에서, 비디오 시퀀스의 화상을 인코딩할 때, 비디오 시퀀스의 화상의 하나 이상의 특징을 검출 및 분류하기 위해 특징 분류기(characteristic classifier)가 사용될 수 있다. 특징의 상이한 클래스는 인코딩을 위한 상이한 비트율과 더 연관되는 상이한 우선순위 수준과 연관될 수 있다. 상이한 우선순위 수준은 인코딩을 위한 상이한 파라미터와 연관될 수 있으며, 이는 결과적으로 상이한 인코딩 품질 수준을 야기할 수 있다. 우선순위 수준이 높을수록, 연관된 파라미터 세트가 유도할 수 있는 비디오의 품질이 더 향상된다. 이러한 특징-기반 비

디오 처리에 의해, 비트율은 큰 정보 손실을 야기하지 않으면서 모니터링 비디오를 위해 크게 감축될 수 있다. 또한, 본 개시의 실시예들은 상이한 어플리케이션 시나리오(예를 들어, 보안, 교통, 환경 모니터링 등)에 대한 우선순위 수준과 파라미터 세트 사이의 대응하는 관계를 맞춤화(customize)할 수 있고, 이로 인해 비디오 코딩 품질을 크게 향상시키고 대역폭 및 저장 공간에 대한 비용 또한 크게 절감할 수 있다.

[0019] 도 1은 본 개시의 일부 실시예에 따른 예시적인 비디오 시퀀스(100)의 구조를 나타낸다. 비디오 시퀀스(100)는 라이브 비디오 또는 캡처 및 보관된 비디오일 수 있다. 비디오(100)는 실세계(real-life) 비디오, 컴퓨터-생성된 비디오(예를 들어, 컴퓨터 게임 비디오), 또는 이들의 조합(예를 들어, 증강-현실 효과가 있는 실세계 비디오)일 수 있다. 비디오 시퀀스(100)는 비디오 캡처 디바이스(예를 들어, 카메라), 이전에 캡처된 비디오를 포함하는 비디오 아카이브(예를 들어, 저장 디바이스에 저장된 비디오 파일), 또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스(예를 들어, 비디오 방송 송수신기)로부터 입력될 수 있다.

[0020] 도 1에 도시된 바와 같이, 비디오 시퀀스(100)는 화상들(102, 104, 106 및 108)을 포함하는, 타임라인을 따라 시간적으로(temporally) 배열되는 일련의 화상을 포함할 수 있다. 화상들(102 내지 106)은 연속적이며, 화상들(106, 108) 사이에는 더 많은 화상들이 존재한다. 도 1에서, 화상(102)은 I-화상이며, 그 참조 화상은 화상(102) 자체이다. 화상(104)은 P-화상이고, 화살표로 표시된 바와 같이 그 참조 화상은 화상(102)이다. 화상(106)은 B-화상이고, 화살표로 표시된 바와 같이 그 참조 화상은 화상(104) 및 화상(108)이다. 일부 실시예에서, 화상(예를 들어, 화상(104))의 참조 화상은 바로 선행하는 또는 이어지는 화상이 아닐 수 있다. 예를 들어, 화상(104)의 참조 화상은 화상(102)에 선행하는 화상일 수 있다. 화상들(102 내지 106)의 참조 화상은 단지 예시일 뿐이며, 본 개시가 참조 화상들의 실시예들을 도 1에 도시된 예시들로 제한하는 것은 아님에 유의해야 한다.

[0021] 통상적으로, 작업(task)의 연산 복잡성 때문에 비디오 코덱은 전체 화상을 동시에 인코딩 또는 디코딩하지 않는다. 비디오 코덱은 그 대신, 화상을 기본 세그먼트로 분할(split)할 수 있고, 화상을 세그먼트별로 인코딩 또는 디코딩할 수 있다. 본 개시에서 이러한 기본 세그먼트는 기본 처리 유닛("BPU" : Basic Processing Unit)으로 지칭된다. 예를 들어, 도 1의 구조(110)는 비디오 시퀀스(100)의 화상(예를 들어, 화상(102 내지 108) 중 임의의 것)의 예시적인 구조를 도시한다. 구조(110)에서, 화상은 4x4 기본 처리 유닛으로 분할(divide)되며, 그 경계선이 점선으로 도시되어 있다. 일부 실시예에서, 기본 처리 유닛이 일부 비디오 코딩 표준(예를 들어, MPEG 패밀리, H.261, H.263, 또는 H.264/AVC)에서는 "매크로 블록(macroblocks)"으로, 또는 일부 다른 비디오 코딩 표준(예를 들어, H.265/HEVC 또는 H.266/VVC)에서는 "코딩 트리 유닛(CTU: Coding Tree Unit)으로 지칭될 수 있다. 기본 처리 유닛은 화상 내에서 128x128, 64x64, 32x32, 16x16, 4x8, 16x32와 같은 다양한 크기 또는 임의의 형상 및 픽셀 크기를 가질 수 있다. 기본 처리 유닛의 크기와 형상은 코딩 효율과 기본 처리 유닛에서 유지되어야 하는 디테일의 수준의 균형에 기초하여 화상에 대해 선택될 수 있다.

[0022] 기본 처리 유닛은 논리 유닛일 수 있으며, 논리 유닛은 컴퓨터 메모리(예를 들어, 비디오 프레임 버퍼)에 저장되는 상이한 유형의 비디오 데이터의 그룹을 포함할 수 있다. 예를 들어, 컬러 화상의 기본 처리 유닛은 무채색 밝기 정보(achromatic brightness information)를 나타내는 루마 성분(Y), 색상 정보를 나타내는 하나 이상의 크로마 성분(예를 들어, Cb 및 Cr), 및 루마 및 크로마 성분이 동일한 크기의 기본 처리 유닛을 구비할 수 있는 연관된 신택스 요소를 포함할 수 있다. 루마 및 크로마 성분은 일부 비디오 코딩 표준(예를 들어, H.265/HEVC 또는 H.266/VVC)에서는 "코딩 트리 블록(CTB: coding tree block)"으로 지칭될 수 있다. 기본 처리 유닛에 대해 수행되는 임의의 동작은 기본 처리 유닛의 루마 및 크로마 성분의 각각에 반복적으로 수행될 수 있다.

[0023] 비디오 코딩은 다수의 동작 단계를 구비하며, 그 예시가 도 2a 내지 도 2b, 그리고 도 3a 내지 도 3b에 상세하게 설명될 것이다. 각 단계에 대해, 기본 처리 유닛의 크기가 여전히 처리하기에는 너무 클 수 있기 때문에, 본 개시에서 "기본 처리 서브-유닛"으로 지칭되는 세그먼트로 더 분할될 수 있다. 일부 실시예에서, 기본 처리 서브-유닛은 일부 비디오 코딩 표준(예를 들어, MPEG 패밀리, H.261, H.263, 또는 H.264/AVC)에서는 "블록"으로, 또는 일부 다른 비디오 코딩 표준(예를 들어, H.265/HEVC 또는 H.266/VVC)에서는 "코딩 유닛(CU)"으로 지칭될 수 있다. 기본 처리 서브-유닛은 기본 처리 유닛과 동일하거나 더 작은 크기를 가질 수 있다. 기본 처리 유닛과 유사하게, 기본 처리 서브-유닛 또한 논리 유닛일 수 있으며, 논리 유닛은 컴퓨터 메모리(예를 들어, 비디오 프레임 버퍼)에 저장되는 상이한 유형의 비디오 데이터(예를 들어, Y, Cb, Cr, 및 연관된 신택스 요소)의 그룹을 포함할 수 있다. 기본 처리 서브-유닛에 수행되는 임의의 동작은 그 기본 처리 서브-유닛의 루마 및 크로마 성분의 각각에 대해 반복적으로 수행될 수 있다. 이러한 분할은 처리 필요성에 따라 추가적인 수준으로 수행될 수 있음에 유의해야 한다. 또한, 상이한 단계가 상이한 방식을 이용하여 기본 처리 유닛들을 분할할 수 있음에 유

의해야 한다.

- [0024] 예를 들어, 모드 결정 단계(그 예시가 도 2b에서 상세하게 설명될 것이다)에서, 인코더는 기본 처리 유닛에 대해 어떤 예측(prediction) 모드(예를 들어, 인트라-화상 예측 또는 인터-화상 예측)를 사용할지 결정할 수 있는데, 기본 처리 유닛은 이러한 결정을 하기에 너무 클 수 있다. 인코더는 기본 처리 유닛을 다수의 기본 처리 서브-유닛(예를 들어, H.265/HEVC 또는 H.266/VVC에서는 CU)으로 분할하고, 각각의 개별 기본 처리 서브-유닛에 대해 예측 유형을 결정할 수 있다.
- [0025] 또 다른 예시로서, 예측 단계(그 예시가 도 2a에서 상세하게 설명될 것이다)에서, 인코더는 기본 처리 서브-유닛(예를 들어, CU)의 수준에서 예측 동작을 수행할 수 있다. 그러나 일부 경우에는 기본 처리 서브-유닛이 처리 하기에 여전히 너무 클 수 있다. 인코더는 기본 처리 서브-유닛을 예측 동작이 수행될 수 있는 수준의 더 작은 세그먼트(예를 들어, H.265/HEVC 또는 H.266/VVC에서 "예측 블록(prediction block)" 또는 "PB"로 지칭됨)로 더 세분화할 수 있다.
- [0026] 또 다른 예시로서, 변환 단계(그 예시가 도 2a에서 상세하게 설명될 것이다)에서, 인코더는 잔차 기본 처리 서브-유닛(예를 들어, CU)에 대해 변환 동작을 수행할 수 있다. 그러나 일부 경우에는 기본 처리 서브-유닛이 처리하기에는 여전히 너무 클 수 있다. 인코더는 기본 서브-유닛을 변환 동작이 수행될 수 있는 수준의 더 작은 세그먼트(H.265/HEVC 또는 H.266/VVC에서 "변환 블록(transform block)" 또는 "TB"로 지칭됨)로 더 세분화할 수 있다. 동일한 기본 처리 서브-유닛의 분할 방식이 예측 단계와 변환 단계에서 상이할 수 있다. 예를 들어, H.265/HEVC 또는 H.266/VVC에서 동일한 CU의 예측 블록과 변환 블록은 상이한 크기와 개수를 가질 수 있다.
- [0027] 도 1의 구조(110)에서, 기본 처리 유닛(112)은 경계가 점선으로 도시되어 있는 3x3 기본 처리 서브-유닛으로 더 분할된다. 동일한 화상의 상이한 기본 처리 유닛이 상이한 방식으로 기본 처리 서브-유닛으로 분할될 수도 있다.
- [0028] 일부 구현예에서, 비디오 인코딩 및 디코딩에 병렬 처리 및 오류 복원 능력을 제공하기 위해, 화상의 영역에 대해, 인코딩 또는 디코딩 처리가 화상의 임의의 다른 영역으로부터의 어떤 정보에도 의존하지 않을 수 있도록, 화상은 처리를 위한 영역으로 분할될 수 있다. 환언하면, 화상의 각 영역이 독립적으로 처리될 수 있다. 이렇게 함으로써, 코덱은 화상의 상이한 영역을 병렬적으로 처리할 수 있고, 이로 인해 코딩 효율이 증가할 수 있다. 또한, 영역의 데이터가 처리 중에 손상되거나 네트워크 전송 중에 손실되는 경우, 코덱은 손상 또는 손실된 데이터에 의존하지 않고 동일한 화상의 다른 영역들을 정확하게 인코딩 또는 디코딩할 수 있어, 오류 복원 능력을 제공한다. 일부 비디오 코딩 표준에서, 화상은 상이한 유형의 영역으로 분할될 수 있다. 예를 들어, H.265/HEVC 및 H.266/VVC는 "슬라이스"와 "타일", 두 유형의 영역을 제공한다. 또한 비디오 시퀀스(100)의 상이한 화상들은 화상을 영역으로 세분화하기 위한 상이한 구획(partition) 방식을 가질 수 있음에 유의해야 한다.
- [0029] 예를 들어, 도 1에서 구조(110)는 구조(110) 내측에 실선으로 경계가 도시되어 있는 3개의 영역(114, 116, 118)으로 분할된다. 영역(114)은 4개의 기본 처리 유닛을 포함한다. 영역(116 및 118)들의 각각은 6개의 기본 처리 유닛을 포함한다. 도 1에서의 구조(110)의 기본 처리 유닛, 기본 처리 서브-유닛 및 영역들은 단지 예시일 뿐이며, 본 개시는 그 실시예를 제한하지 않음에 유의해야 한다.
- [0030] 도 2a는 본 개시의 일부 실시예에 따른 예시적인 인코딩 프로세스(200A)의 개략도를 나타낸다. 인코더는 프로세스(200A)에 따라 비디오 시퀀스(202)를 비디오 비트스트림(228)으로 인코딩할 수 있다. 도 1의 비디오 시퀀스(100)와 유사하게, 비디오 시퀀스(202)는 시간 순서로 배열되어 있는 화상들("원본 화상들(original pictures)"으로 지칭됨)의 세트를 포함할 수 있다. 도 1의 구조(110)와 유사하게, 비디오 시퀀스(202)의 각 원본 화상은 인코더에 의해 기본 처리 유닛, 기본 처리 서브-유닛, 또는 처리를 위한 영역으로 분할될 수 있다. 일부 실시예에서, 인코더는 비디오 시퀀스(202)의 각 원본 화상에 대하여 기본 처리 유닛 수준에서 프로세스(200A)를 수행할 수 있다. 예를 들어, 인코더는 반복적인 방식으로 프로세스(200A)를 수행할 수 있어, 인코더는 프로세스(200A)의 한 번의 반복에서 하나의 기본 처리 유닛(a basic processing unit)을 인코딩할 수 있다. 일부 실시예에서, 인코더는 비디오 시퀀스(202)의 각 원본 화상의 영역들(예를 들어, 영역들(114 내지 118))에 대해 병렬적으로 프로세스(200A)를 수행할 수 있다.
- [0031] 도 2a에서, 인코더는 예측 데이터(206) 및 예측된 BPU(208)를 생성하기 위해 비디오 시퀀스(202)의 원본 화상의 기본 처리 유닛("원본 BPU"로 지칭됨)을 예측 단계(204)로 공급할 수 있다. 인코더는 원본 BPU로부터 예측 BPU(208)를 감산(subtract)하여 잔차 BPU(210)를 생성할 수 있다. 인코더는 양자화된 변환 계수(216)를 생성하기 위해 잔차 BPU(210)를 변환 단계(212) 및 양자화 단계(214)로 공급할 수 있다. 인코더는 비디오 비트스트림

(228)을 생성하기 위해 예측 데이터(206) 및 양자화된 변환 계수(216)를 바이너리 코딩 단계(226)로 공급할 수 있다. 컴포넌트들(202, 204, 206, 208, 210, 212, 214, 216, 226, 및 228)은 "순방향 경로(forward path)"로 지칭될 수 있다. 프로세스(200A) 중에, 양자화 단계(214) 이후에 인코더는 재구성된 잔차 BPU(222)를 생성하기 위해 양자화된 변환 계수(216)를 역양자화 단계(218) 및 역변환 단계(220)로 공급할 수 있다. 인코더는 프로세스(200A)의 다음 반복에 대한 예측 단계(204)에서 사용되는 예측 참조(prediction reference)(224)를 생성하기 위해 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 가산(add)할 수 있다. 프로세스(200A)의 컴포넌트들(218, 220, 222, 및 224)은 "재구성 경로(reconstruction path)"로 지칭될 수 있다. 재구성 경로는 인코더와 디코더 모두가 예측에 대해 동일한 참조 데이터를 사용함을 보장하기 위해 사용될 수 있다.

[0032] 인코더는 (순방향 경로에서) 원본 화상의 각각의 원본 BPU를 인코딩하고 (재구성 경로에서) 원본 화상의 다음 원본 BPU를 인코딩하기 위한 예측 참조(224)를 생성하기 위해 프로세스(200A)를 반복적으로 수행할 수 있다. 인코더는 원본 화상의 모든 원본 BPU를 인코딩한 후에, 비디오 시퀀스(202) 내 다음 화상을 인코딩하도록 진행할 수 있다.

[0033] 프로세스(200A)를 참조하면, 인코더는 비디오 캡처 디바이스(예를 들어, 카메라)에 의해 생성되는 비디오 시퀀스(202)를 수신할 수 있다. 본 명세서에서, "수신하다"라는 용어는 입력 데이터에 대한 수신, 입력, 취득, 검색(retrieve), 획득, 판독, 접근, 또는 임의의 방식으로의 임의의 동작을 지칭할 수 있다.

[0034] 현재 반복에서의 예측 단계(204)에서 인코더는 원본 BPU와 예측 참조(224)를 수신하고, 예측 동작을 수행하여 예측 데이터(206)와 예측된 BPU(208)를 생성할 수 있다. 예측 참조(224)는 프로세스(200A)의 이전 반복의 재구성 경로로부터 생성될 수 있다. 예측 단계(204)의 목적은 원본 BPU를 재구성하기 위해 예측된 BPU(208)로서 사용될 수 있는 예측 데이터(206)를, 예측 데이터(206) 및 예측 참조(224)로부터 추출(extract)함으로써 정보 리던던시(redundancy)를 감소시키기 위함이다.

[0035] 이상적으로는, 예측된 BPU(208)가 원본 BPU와 동일할 수 있다. 그러나 비-이상적인 예측 및 재구성 동작 때문에, 예측된 BPU(208)는 일반적으로 원본 BPU와 약간 상이하다. 이러한 차이를 기록하기 위해, 인코더는 예측된 BPU(208)를 생성한 다음 이를 원본 BPU로부터 감산하여 잔차 BPU(210)를 생성할 수 있다. 예를 들어, 인코더는 원본 BPU의 대응하는 픽셀 값으로부터 예측된 BPU(208)의 픽셀 값(예를 들어, 그레이스케일 값 또는 RGB 값)을 감산할 수 있다. 잔차 BPU(210)의 각 픽셀은 원본 BPU와 예측된 BPU(208)의 대응하는 픽셀 간의 이러한 감산의 결과로서 잔차 값을 가질 수 있다. 예측 데이터(206)와 잔차 BPU(210)는 원본 BPU에 비해 더 적은 비트를 가질 수 있지만, 큰 품질 저하 없이 원본 BPU를 재구성하는 데에 사용될 수 있다. 따라서, 원본 BPU가 압축된다.

[0036] 잔차 BPU(210)를 더 압축하기 위해, 변환 단계(212)에서 인코더는 잔차 BPU(210)를 2-차원 "베이스 패턴(base pattern)"들의 세트에 분해함으로써, 잔차 BPU(210)의 공간 리던던시(spatial redundancy)를 감소시킬 수 있으며, 각각의 베이스 패턴은 "변형 계수(transform coefficient)"와 연관된다. 베이스 패턴은 동일한 크기(예를 들어, 잔차 BPU(210)의 크기)를 가질 수 있다. 각각의 베이스 패턴은 잔차 BPU(210)의 변동 주파수(variation frequency)(예를 들어, 밝기 변동 주파수) 성분을 나타낼 수 있다. 어떤 베이스 패턴도 임의의 다른 베이스 패턴의 임의의 결합(예를 들어, 선형 결합)으로부터 재생성될 수 없다. 환언하면, 분해는 잔차 BPU(210)의 변동들을 주파수 도메인으로 분해할 수 있다. 이러한 분해는, 베이스 패턴이 이산 푸리에 변환의 베이스 함수(예를 들어, 삼각함수)와 유사하고, 변환 계수가 베이스 함수와 연관된 계수와 유사하다는 점에서, 함수의 이산 푸리에 변환과 유사하다.

[0037] 상이한 변환 알고리즘은 상이한 베이스 패턴을 사용할 수 있다. 예를 들어, 변환 단계(212)에서는 이산 코사인 변환, 이산 사인 변환 등과 같은 다양한 변환 알고리즘이 사용될 수 있다. 변환 단계(212)에서의 변환은 가역적이다. 즉, 인코더는 변환의 역방향 동작("역변환(inverse transform)"으로 지칭됨)에 의해 잔차 BPU(210)를 복원할 수 있다. 예를 들어, 잔차 BPU(210)의 픽셀을 복원하기 위해 역변환은 베이스 패턴의 대응하는 픽셀의 값에 각각의 연관 계수를 곱하고, 가중합을 생성하기 위해 곱한 것을 가산하는 것일 수 있다. 비디오 코딩 표준에 있어서, 인코더와 디코더 모두 동일한 변환 알고리즘(따라서 동일한 베이스 패턴)을 사용할 수 있다. 따라서, 인코더는, 디코더가 인코더로부터 베이스 패턴을 수신하지 않고 잔차 BPU(210)를 재구성할 수 있는 변환 계수만 기록할 수 있다. 잔차 BPU(210)에 비해 변환 계수는 더 적은 비트 수를 가질 수 있지만, 큰 품질 저하 없이 잔차 BPU(210)를 재구성하는 데에 사용될 수 있다. 따라서, 잔차 BPU(210)는 더 압축된다.

[0038] 인코더는 양자화 단계(214)에서 변환 계수를 더 압축할 수 있다. 변환 프로세스에서, 상이한 베이스 패턴은 상이한 변동 주파수(예를 들어, 밝기 변동 주파수)를 나타낼 수 있다. 일반적으로 저주파수 변동이 육안으로 더

잘 인식되므로, 인코더는 디코딩에 큰 품질 저하를 일으키지 않고 고주파수 변동 정보를 무시할 수 있다. 예를 들어, 양자화 단계(214)에서 인코더는 각각의 변환 계수를 정수 값("양자화 파라미터"로 지칭됨)으로 나누고 몫을 가장 가까운 정수로 반올림함으로써 양자화된 변환 계수를 생성할 수 있다. 이러한 연산(operation) 이후에, 고주파수 베이스 패턴의 일부 변환 계수는 0으로 변환(convert)될 수 있고, 저주파수 베이스 패턴의 변환 계수는 더 작은 정수로 변환될 수 있다. 인코더는 0의 값의 양자화된 변환 계수(216)를 무시할 수 있으며, 이로 인해 변환 계수는 더 압축된다. 양자화 프로세스 또한 가역적이며, 여기서 양자화된 변환 계수(216)가 양자화의 역방향 동작("역양자화(inverse quantization)"로 지칭됨)에서 변환 계수로 재구성될 수 있다.

[0039] 인코더가 반올림 연산에서 이러한 나눗셈의 나머지를 무시하기 때문에, 양자화 단계(214)는 손실이 있을 수 있다(lossy). 통상적으로, 양자화 단계(214)는 프로세스(200A) 중 정보 손실에 가장 많이 기여할 수 있다. 정보 손실이 클수록 양자화된 변환 계수(216)가 필요로 할 수 있는 비트 수는 더 적다. 상이한 수준의 정보 손실을 획득하기 위해 인코더는 상이한 값의 양자화 파라미터 또는 양자화 프로세스의 임의의 다른 파라미터를 사용할 수 있다.

[0040] 바이너리 코딩 단계(226)에서, 인코더는, 예를 들어, 엔트로피 부호화, 가변 길이 부호화, 산술 부호화, 허프만(Huffman) 부호화, 컨텍스트-적응적 이진 산술 부호화(context-adaptive binary arithmetic coding), 또는 임의의 다른 무손실 또는 손실 압축 알고리즘과 같은 바이너리 코딩 기술을 사용하여 예측 데이터(206)와 양자화된 변환 계수(216)를 인코딩할 수 있다. 일부 실시예에서, 예측 데이터(206) 및 양자화된 변환 계수(216) 이외에도, 인코더는, 예를 들어 예측 단계(204)에서 사용되는 예측 모드, 예측 동작의 파라미터, 변환 단계(212)에서의 변환 유형, 양자화 프로세스의 파라미터(예를 들어, 양자화 파라미터), 인코더 제어 파라미터(예를 들어, 비트율 제어 파라미터) 등과 같은 바이너리 코딩 단계(226)에서의 다른 정보를 인코딩할 수 있다. 인코더는 바이너리 코딩 단계(226)의 출력 데이터를 사용하여 비디오 비트스트림(228)을 생성할 수 있다. 일부 실시예에서, 비디오 비트스트림(228)은 네트워크 전송을 위해 추가적으로 패킷화(packetize)될 수 있다.

[0041] 프로세스(200A)의 재구성 경로를 참조하면, 인코더는 역양자화 단계(218)에서, 양자화된 변환 계수(216)에 역양자화를 수행하여 재구성된 변환 계수를 생성할 수 있다. 역변환 단계(220)에서, 인코더는 재구성된 변환 계수에 기초하여 재구성된 잔차 BPU(222)를 생성할 수 있다. 인코더는 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 가산하여 프로세스(200A)의 다음 반복에서 사용될 예측 참조(224)를 생성할 수 있다.

[0042] 비디오 시퀀스(202)를 인코딩하는 데에 프로세스(200A)의 다른 변형이 사용될 수 있음에 유의해야 한다. 일부 실시예에서, 인코더에 의해 프로세스(200A)의 단계들이 다른 순서로 수행될 수 있다. 일부 실시예에서, 프로세스(200A)의 하나 이상의 단계는 단일 단계로 결합될 수 있다. 일부 실시예에서, 프로세스(200A)의 단일 단계는 다수의 단계로 분할될 수 있다. 예를 들어, 변환 단계(212)와 양자화 단계(214)는 단일 단계로 결합될 수 있다. 일부 실시예에서, 프로세스(200A)는 추가적인 단계를 포함할 수 있다. 일부 실시예에서, 프로세스(200A)는 도 2a에 도시된 하나 이상의 단계를 생략할 수 있다.

[0043] 도 2b는 본 개시의 일부 실시예에 따른 다른 예시적인 인코딩 프로세스(200B)의 개략도를 나타낸다. 프로세스(200B)는 프로세스(200A)로부터 수정될 수 있다. 예를 들어, 프로세스(200B)는 하이브리드 비디오 코딩 표준(예를 들어, H.26x 시리즈)에 따른 인코더에 의해 사용될 수 있다. 프로세스(200A)와 비교하면, 프로세스(200B)의 순방향 경로는 추가적으로 모드 결정 단계(230)를 포함하며, 예측 단계(204)를 공간 예측 단계(2042)와 시간 예측 단계(2044)로 분할한다. 프로세스(200B)의 재구성 경로는 추가적으로 루프 필터 단계(232) 및 버퍼(234)를 포함한다.

[0044] 일반적으로, 예측 기술은 공간 예측과 시간 예측의 두 가지로 분류될 수 있다. 공간 예측(예를 들어, 인트라-화상 예측 또는 "인트라 예측")은 동일한 화상에서 하나 이상의 이미 코딩된 이웃하는 BPU로부터의 픽셀을 이용하여 현재 BPU를 예측할 수 있다. 즉, 공간 예측에서 예측 참조(224)는 이웃하는 BPU를 포함할 수 있다. 공간 예측은 화상의 내재적인(inherent) 공간 리던던시를 감소시킬 수 있다. 시간 예측(예를 들어, 인터-화상 예측 또는 "인터 예측")은 하나 이상의 이미 코딩된 화상으로부터의 영역을 사용하여 현재 BPU를 예측할 수 있다. 즉, 시간 예측에서 예측 참조(224)는 코딩된 화상을 포함할 수 있다. 시간 예측은 화상들의 내재적인 시간 리던던시(temporal redundancy)를 감소할 수 있다.

[0045] 프로세스(200B)를 참조하면, 인코더는 순방향 경로에서, 공간 예측 단계(2042) 및 시간 예측 단계(2044)에서 예측 동작을 수행한다. 예를 들어, 공간 예측 단계(2042)에서 인코더는 인트라 예측을 수행할 수 있다. 인코딩되는 화상의 원본 BPU에 대하여, 예측 참조(224)는 동일한 화상에서 (순방향 경로에서) 인코딩되고 (재구성 경로에서) 재구성되었던 하나 이상의 이웃하는 BPU를 포함할 수 있다. 인코더는 이웃하는 BPU를 외삽(extrapolate)

함으로써 예측된 BPU(208)를 생성할 수 있다. 외삽 기술은 예를 들어 선형 외삽 또는 내삽법(interpolation), 다항식 외삽 또는 내삽법 등을 포함할 수 있다. 일부 실시예에서, 인코더는 예컨대 예측된 BPU(208)의 각 픽셀에 대해 대응하는 픽셀의 값을 외삽함으로써, 픽셀 수준에서 외삽을 수행할 수 있다. 외삽에 사용되는 이웃하는 BPU는 원본 BPU에 대해 수직 방향(예를 들어, 원본 BPU의 상부), 수평 방향(예를 들어, 원본 BPU의 좌측), 대각선 방향(예를 들어, 원본 BPU의 좌측 하단, 우측 하단, 좌측 상단, 우측 상단), 또는 사용되는 비디오 코딩 표준에서 정의되는 임의의 방향과 같이 다양한 방향에 위치될 수 있다. 인트라 예측의 경우, 예측 데이터(206)는 예를 들어 사용되는 이웃하는 BPU의 위치(예를 들어, 좌표), 사용되는 이웃하는 BPU의 크기, 외삽의 파라미터, 사용되는 이웃하는 BPU의 원본 BPU에 대한 방향 등을 포함할 수 있다.

[0046] 또 다른 예시로서, 인코더는 시간 예측 단계(2044)에서 인트라 예측을 수행할 수 있다. 현재 화상의 원본 BPU에 대해, 예측 참조(224)는 (순방향 경로에서) 인코딩되고 (재구성 경로에서) 재구성되었던 하나 이상의 화상("참조 화상"으로 지칭됨)을 포함할 수 있다. 일부 실시예에서, 참조 화상은 BPU 마다 인코딩 및 재구성될 수 있다. 예를 들어, 인코더는 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 가산하여 재구성된 BPU를 생성할 수 있다. 동일한 화상의 모든 재구성된 BPU가 생성되면, 인코더는 참조 화상으로서 재구성된 화상을 생성할 수 있다. 인코더는 "모션 추정" 동작을 수행하여 참조 화상의 범위("탐색 창(search window)"으로 지칭됨)에서 정합 영역(matching region)을 탐색할 수 있다. 참조 화상에서의 탐색 창의 위치는 현재 화상에서의 원본 BPU의 위치에 기초하여 결정될 수 있다. 예를 들어, 탐색 창은 참조 화상 내 좌표가 현재 화상에서의 원본 BPU와 동일한 위치에 센터링(center)될 수 있고, 사전 결정된 거리만큼 확장될 수 있다. 인코더가 (예를 들어, 화소-재귀적 알고리즘(pel-recursive algorithm), 블록-정합 알고리즘 등을 사용함으로써) 탐색 창 내 원본 BPU와 유사한 영역을 식별하면, 인코더는 이러한 영역을 정합 영역으로 결정할 수 있다. 정합 영역은 원본 BPU와 상이한 치수(예를 들어, 더 작거나, 동일하거나, 더 크거나, 다른 형상)를 가질 수 있다. 참조 화상과 현재 화상은 (예를 들어, 도 1에 도시된 바와 같이) 타임라인에서 시간적으로 분리되어 있기 때문에, 정합 영역이 시간의 흐름에 따라 원본 BPU의 위치로 "이동"하는 것으로 간주될 수 있다. 인코더는 이러한 모션의 방향 및 거리를 "모션 벡터(motion vector)"로서 기록할 수 있다. (예를 들어, 도 1의 화상(106)과 같이) 다수의 참조 화상이 사용될 경우, 인코더는 각각의 참조 화상에 대해 정합 영역을 탐색하고 연관된 모션 벡터를 결정할 수 있다. 일부 실시예에서, 인코더는 각각의 정합 참조 화상의 정합 영역의 픽셀 값에 가중치를 할당할 수 있다.

[0047] 모션 추정은 병진(translation), 회전(rotation), 줌(zoom) 등과 같은 다양한 유형의 모션을 식별하는 데 사용될 수 있다. 인트라 예측의 경우, 예측 데이터(206)는 예를 들어, 정합 영역의 위치(예를 들어, 좌표), 정합 영역과 연관된 모션 벡터, 참조 화상의 개수, 참조 화상과 연관된 가중치 등을 포함할 수 있다.

[0048] 예측된 BPU(208)를 생성하기 위해, 인코더는 "모션 보상(motion compensation)" 동작을 수행할 수 있다. 모션 보상은 예측 데이터(206)(예를 들어, 모션 벡터) 및 예측 참조(224)에 기초하여 예측된 BPU(208)를 재구성하기 위해 사용될 수 있다. 예를 들어, 인코더는 모션 벡터에 따라 참조 화상의 정합 영역을 이동시킬 수 있고, 인코더는 거기서 현재 화상의 원본 BPU를 예측할 수 있다. (예를 들어, 도 1의 화상(106)과 같이) 다수의 참조 화상이 사용되는 경우, 인코더는 각각의 모션 벡터에 따라 참조 화상의 정합 영역을 이동시켜 정합 영역들의 픽셀 값의 평균을 낼 수 있다. 일부 실시예에서, 인코더가 각각의 정합 참조 화상의 정합 영역의 픽셀 값에 가중치를 할당했을 경우, 인코더는 이동된 정합 영역의 픽셀 값들의 가중치 합을 가산할 수 있다.

[0049] 일부 실시예에서, 인트라 예측은 단일 방향 또는 양방향일 수 있다. 단일 방향 인트라 예측은 현재 화상에 대해 동일한 시간 방향에 있는 하나 이상의 참조 화상을 사용할 수 있다. 예를 들어, 도 1에 도시된 화상(104)은 참조 화상(즉, 화상(102))이 화상(104)보다 선행한다는 점에서 단일 인트라-예측된 화상이다. 양방향 인트라 예측은 현재 화상에 대해 양쪽 시간 방향에 있는 하나 이상의 참조 화상을 사용할 수 있다. 예를 들어, 도 1에 도시된 화상(106)은 참조 화상들(즉, 화상들(104 및 108))이 화상(104)에 대해 양쪽 시간 방향에 있다는 점에서 양방향 인트라-예측된 화상이다.

[0050] 여전히 프로세스(200B)의 순방향 경로를 참조하면, 공간 예측(2042) 및 시간 예측 단계(2044) 이후에, 모드 결정 단계(230)에서 인코더는 프로세스(200B)의 현재 반복에 대한 예측 모드(예를 들어, 인트라 예측 또는 인트라 예측 중 하나)를 선택할 수 있다. 예를 들어, 인코더는 율-왜곡(rate-distortion) 최적화 기법을 수행할 수 있는데, 율-왜곡 최적화 기법에서 인코더는 후보 예측 모드의 비트율과 후보 예측 모드 하에서 재구성된 참조 화상의 왜곡에 따른 비용 함수의 값을 최소화하기 위해 예측 모드를 선택할 수 있다. 선택된 예측 모드에 따라, 인코더는 대응하는 예측된 BPU(208) 및 예측된 데이터(206)를 생성할 수 있다.

[0051] 프로세스(200B)의 재구성 경로에서, 순방향 경로에서 인트라 예측 모드가 선택된 경우, 예측 참조(224)(예를 들

어, 현재 화상에서 인코딩 및 재구성되었던 현재 BPU를 생성한 이후에 인코더는 추후 사용(예를 들어, 현재 화상의 다음 BPU의 외삽)을 위해 예측 참조(224)를 공간 예측 단계(2042)로 직접적으로 공급할 수 있다. 순방향 경로에서 인터 예측 모드가 선택된 경우, 예측 참조(224)(예를 들어, 모든 BPU가 인코딩 및 재구성된 현재 화상)을 생성한 이후에 인코더는 루프 필터 단계(232)에 예측 참조(224)를 공급할 수 있고, 여기서 인코더는 인터 예측에 의해 도입되는 왜곡(예를 들어, 블로킹 아티팩트)을 감소 또는 제거하기 위해 예측 참조(224)에 루프 필터를 적용할 수 있다. 인코더는 루프 필터 단계(232)에서, 예를 들어, 디블로킹(deblocking), 샘플 적응적 오프셋, 적응적 루프 필터 등과 같은 다양한 루프 필터 기법을 적용할 수 있다. 루프-필터된 참조 화상은 추후 사용(예를 들어, 비디오 시퀀스(202)의 미래 화상에 대한 인터-예측 참조 화상으로서 사용)을 위해 버퍼(234)(또는 “디코딩된 화상 버퍼”)에 저장될 수 있다. 인코더는 시간 예측 단계(2044)에서 사용될 하나 이상의 참조 화상을 버퍼(234)에 저장할 수 있다. 일부 실시예에서, 인코더는 바이너리 코딩 단계(226)에서, 양자화된 변환 계수(216), 예측 데이터(206) 및 다른 정보와 함께 루프 필터의 파라미터(예를 들어, 루프 필터 강도)를 인코딩할 수 있다.

[0052] 도 3a는 본 개시의 일부 실시예에 따른 예시적인 디코딩 프로세스(300A)의 개략도를 나타낸다. 프로세스(300A)는 도 2a에 도시된 압축 프로세스(200A)에 대응하는 압축 해제 프로세스일 수 있다. 일부 실시예에서, 프로세스(300A)는 프로세스(200A)의 재구성 경로와 유사할 수 있다. 디코더는 프로세스(300A)에 따라 비디오 비트스트림(228)을 비디오 스트림(304)으로 디코딩할 수 있다. 비디오 스트림(304)은 비디오 시퀀스(202)와 매우 유사할 수 있다. 그러나 압축 및 압축 해제 프로세스(예를 들어, 도 2a 및 도 2b에 도시된 양자화 단계(214))에서의 정보 손실 때문에, 일반적으로, 비디오 스트림(304)은 비디오 시퀀스(202)와 동일하지 않다. 도 2a 및 도 2b에 도시된 프로세스(200A) 및 프로세스(200B)와 유사하게, 디코더는 비디오 비트스트림(228)에서 인코딩된 각각의 화상에 대해 기본 처리 유닛(BPU) 수준에서 프로세스(300A)를 수행할 수 있다. 예를 들어, 디코더는 반복적인 방식으로 프로세스(300A)를 수행할 수 있어, 프로세스(300A)의 일 반복에서 하나의 기본 처리 유닛(a basic processing unit)을 디코딩할 수 있다. 일부 실시예에서, 디코더는 비디오 비트스트림(228)에서 인코딩된 각각의 화상의 영역(예를 들어, 영역들(114 내지 118))에 대해 병렬적으로 프로세스(300A)를 수행할 수 있다.

[0053] 도 3a에서, 디코더는 인코딩된 화상의 기본 처리 유닛(“인코딩된 BPU”로서 지칭됨)과 연관된 비디오 비트스트림(228)의 부분을 바이너리 디코딩 단계(302)에 공급할 수 있다. 바이너리 디코딩 단계(302)에서 디코더는 상기 부분을 예측 데이터(206) 및 양자화된 변환 계수(216)로 디코딩할 수 있다. 디코더는 재구성된 잔차 BPU(222)를 생성하기 위해 양자화된 변환 계수(216)를 역양자화 단계(218) 및 역변환 단계(220)에 공급할 수 있다. 디코더는 예측된 BPU(208)를 생성하기 위해 예측 데이터(206)를 예측 단계(204)에 공급할 수 있다. 디코더는 재구성된 잔차 BPU(222)를 예측된 BPU(208)에 더하여 예측된 참조(224)를 생성할 수 있다. 일부 실시예에서, 예측된 참조(224)는 버퍼(예를 들어, 컴퓨터 메모리 내 디코딩된 화상 버퍼)에 저장될 수 있다. 디코더는 예측된 참조(224)를 프로세스(300A)의 다음 반복에서의 예측 동작을 수행하기 위한 예측 단계(204)에 공급할 수 있다.

[0054] 디코더는 인코딩된 화상의 각각의 인코딩된 BPU를 디코딩하고, 인코딩된 화상의 다음 인코딩된 BPU를 인코딩하기 위한 예측된 참조(224)를 생성하기 위해, 프로세스(300A)를 반복적으로 수행할 수 있다. 인코딩된 화상의 모든 인코딩된 BPU를 디코딩한 다음, 디코더는 디스플레이를 위한 비디오 스트림(304)으로 화상을 출력하고, 비디오 비트스트림(228) 내 다음 인코딩된 화상을 디코딩하도록 진행할 수 있다.

[0055] 바이너리 디코딩 단계(302)에서, 디코더는 인코더에 의해 사용되는 바이너리 코딩 기법(예를 들어, 엔트로피 부호화, 가변 길이 부호화, 산술 부호화, 허프만 부호화, 컨텍스트-적응적 이진 산술 부호화, 또는 임의의 다른 무손실 압축 알고리즘)의 역방향 동작을 수행할 수 있다. 일부 실시예에서, 예측 데이터(206) 및 양자화된 변환 계수(216) 이외에도, 바이너리 디코딩 단계(302)에서 디코더는 예컨대 예측 모드, 예측 동작의 파라미터, 변환 유형, 양자화 프로세스의 파라미터(예를 들어, 양자화 파라미터), 인코더 제어 파라미터(예를 들어, 비트율 제어 파라미터) 등과 같은 다른 정보를 디코딩할 수 있다. 일부 실시예에서, 비디오 비트스트림(228)이 패킷으로 네트워크를 통해 전송되는 경우, 디코더는 바이너리 디코딩 단계(302)로 비디오 비트스트림(228)을 공급하기 전에 이를 디패킷화(depaketize)할 수 있다.

[0056] 도 3b는 본 개시의 일부 실시예에 따른 다른 예시적인 디코딩 프로세스(300B)의 개략도를 나타낸다. 프로세스(300B)는 프로세스(300A)로부터 수정될 수 있다. 예를 들어, 프로세스(300B)는 하이브리드 비디오 코딩 표준(예를 들어, H.26x 시리즈)에 따른 디코더에 의해 사용될 수 있다. 프로세스(300A)와 비교하면, 프로세스(300B)는 추가적으로 예측 단계(204)를 공간 예측 단계(2042)와 시간 예측 단계(2044)로 분할하고, 루프 필터 단계(232)와 버퍼(234)를 추가적으로 포함한다.

- [0057] 프로세스(300B)에서, 디코딩되고 있는 인코딩된 화상(“현재 화상”으로 지칭됨)의 인코딩된 기본 처리 유닛(“현재 BPU”로 지칭됨)에 대해, 바이너리 디코딩 단계(302)로부터 디코더에 의해 디코딩되는 예측 데이터(206)는, 현재 BPU를 인코딩하기 위해 인코더에 의해 어떤 예측 모드가 사용되었는지에 따라 다양한 유형의 데이터를 포함할 수 있다. 예를 들어, 현재 BPU를 인코딩하기 위해 인코더에 의해 인트라 예측이 사용되었을 경우, 예측 데이터(206)는 인트라 예측을 나타내는 예측 모드 표시자(indicator)(예를 들어, 플래그 값), 인트라 예측 동작의 파라미터 등을 포함할 수 있다. 인트라 예측 동작의 파라미터들은 예를 들어, 참조로서 사용되는 하나 이상의 이웃 BPU의 위치(예를 들어, 좌표), 이웃 BPU의 크기, 외삽의 파라미터, 원본 BPU에 대한 이웃 BPU의 방향 등을 포함할 수 있다. 또 다른 예시로서, 현재 BPU를 인코딩하기 위해 인코더에 의해 인터 예측이 사용되었을 경우, 예측 데이터(206)는 인터 예측을 나타내는 예측 모드 표시자(예를 들어, 플래그 값), 인터 예측 동작의 파라미터 등을 포함할 수 있다. 인터 예측 동작의 파라미터는 예를 들어, 현재 BPU와 연관된 참조 화상의 개수, 참조 화상에 각각 연관되는 가중치, 각각의 참조 화상에서의 하나 이상의 정합 영역의 위치(예를 들어, 좌표), 정합 영역과 각각 연관된 하나 이상의 모션 벡터 등을 포함할 수 있다.
- [0058] 예측 모드 표시자에 기초하여, 디코더는 공간 예측 단계(2042)에서 공간 예측(예를 들어, 인트라 예측)을 수행할지 여부, 또는 시간 예측 단계(2044)에서 시간 예측(예를 들어, 인터 예측)을 수행할지 여부를 결정할 수 있다. 이러한 공간 예측 또는 시간 예측 수행에 대한 상세 내용은 도 2b에서 설명되며, 이하에서 반복되지는 않을 것이다. 디코더는 이러한 공간 예측 또는 시간 예측을 수행한 다음 예측된 BPU(208)를 생성할 수 있다. 도 3a에서 설명한 바와 같이, 디코더는 예측된 BPU(208)와 재구성된 잔차 BPU(222)를 더하여 예측 참조(224)를 생성할 수 있다.
- [0059] 프로세스(300B)에서, 디코더는 프로세스(300B)의 다음 반복에서 예측 동작을 수행하기 위해, 예측된 참조(224)를 공간 예측 단계(2042) 또는 시간 예측 단계(2044)에 공급할 수 있다. 예를 들어, 현재 BPU가 공간 예측 단계(2042)에서 인트라 예측을 이용하여 디코딩된 경우, 디코더는 예측 참조(224)(예를 들어, 디코딩된 현재 BPU)를 생성한 다음에 예측 참조(224)를 추후 사용(예를 들어, 현재 화상의 다음 BPU의 외삽)을 위해 공간 예측 단계(2042)로 직접 공급할 수 있다. 현재 BPU가 시간 예측 단계(2044)에서 인터 예측을 이용하여 디코딩된 경우, 디코더는 예측 참조(224)(예를 들어, 모든 BPU가 디코딩된 참조 화상)를 생성한 후에 왜곡(예를 들어, 블로킹 아티팩트)을 감소 또는 제거하기 위해 예측 참조(224)를 루프 필터 단계(232)로 공급할 수 있다. 디코더는 도 2b에 설명된 방식으로 예측 참조(224)에 루프 필터를 적용할 수 있다. 루프-필터링된 참조 화상은 추후 사용을 위해(예를 들어, 비디오 비트스트림(228)의 미래의 인코딩된 화상을 위한 인터-예측 참조 화상으로서 사용하기 위해) 버퍼(234)(예를 들어, 컴퓨터 메모리 내 디코딩된 화상 버퍼)에 저장될 수 있다. 디코더는 시간 예측 단계(2044)에서 사용되도록 하나 이상의 참조 화상을 버퍼(234)에 저장할 수 있다. 일부 실시예에서, 예측 데이터(206)의 예측 모드 표시자가 현재 BPU를 인코딩하는 데에 인터 예측이 사용되었음을 표시할 때, 예측 데이터는 추가적으로 루프 필터의 파라미터(예를 들어, 루프 필터 강도)를 포함할 수 있다.
- [0060] 도 4는 본 개시의 일부 실시예에 따른, 비디오를 인코딩 또는 디코딩하기 위한 예시적인 장치(400)의 블록도이다. 도 4에 도시된 바와 같이, 장치(400)는 프로세서(402)를 포함할 수 있다. 프로세서(402)가 본 명세서에서 설명되는 명령어를 실행할 때, 장치(400)는 비디오 인코딩 또는 디코딩에 특화된 머신이 될 수 있다. 프로세서(402)는 정보를 조작 또는 처리할 수 있는 임의의 유형의 회로망(circuitry)일 수 있다. 예를 들어, 프로세서(402)는 중앙 처리 유닛(또는 “CPU”), 그래픽 처리 유닛(또는 “GPU”), 신경망 처리 유닛(“NPU”), 마이크로 컨트롤러 유닛(“MCU”), 광 프로세서, 프로그램 가능 논리 컨트롤러, 마이크로 컨트롤러, 마이크로 프로세서, 디지털 신호 프로세서, 지적 자산(IP) 코어, 프로그램 가능 논리 어레이(PLA), 프로그램 가능 어레이 논리(PAL), 일반 어레이 논리(GAL), 복합 프로그램 가능 논리 다바이스(CPLD), 필드 프로그램 가능 게이트 어레이(FPGA), 시스템 온 칩(SoC), 주문형 집적 회로(ASIC) 등의 임의의 수의 임의의 조합을 포함할 수 있다. 일부 실시예에서, 프로세서(402)는 단일 논리 컴포넌트로서 그룹화된 프로세서들의 세트일 수도 있다. 예를 들어, 도 4에 도시된 바와 같이, 프로세서(402)는 프로세서(402a), 프로세서(402b), 및 프로세서(402n)를 포함한 다수의 프로세서를 포함할 수 있다.
- [0061] 장치(400)는 또한, 데이터(예를 들어, 명령어의 세트, 컴퓨터 코드, 중간 데이터 등)를 저장하도록 구성되는 메모리(404)를 포함할 수 있다. 예를 들어, 도 4에 도시된 바와 같이, 저장되는 데이터는 프로그램 명령어(예를 들어, 프로세스(200A, 200B, 300A, 또는 300B) 내 단계들을 구현하기 위한 프로그램 명령어) 및 처리를 위한 데이터(예를 들어, 비디오 시퀀스(202), 비디오 비트스트림(228), 또는 비디오 스트림(304))를 포함할 수 있다. 프로세서(402)는 프로그램 명령어 및 처리를 위한 데이터에(예를 들어, 버스(410)를 통해) 접근할 수 있고, 프로그램 명령어를 실행하여 처리를 위한 데이터에 동작 또는 조작을 수행할 수 있다. 메모리(404)는 고속 랜덤-

액세스 저장 디바이스 또는 비-휘발성 저장 디바이스를 포함할 수 있다. 일부 실시예에서, 메모리(404)는 랜덤-액세스 메모리(RAM), 읽기-전용 메모리(ROM), 광 디스크, 자기 디스크, 하드 드라이브, SSD(solid-state drive), 플래시 드라이브, SD(security digital) 카드, 메모리 스틱, CF(compact flash) 카드 등의 임의의 수의 임의의 조합을 포함할 수 있다. 메모리(404)는 또한, 단일 논리 컴포넌트로서 그룹화된 메모리의 그룹(도 4에는 도시되어 있지 않음)일 수도 있다.

[0062] 버스(410)는 내부 버스(예를 들어, CPU-메모리 버스), 외부 버스(예를 들어, USB(universal serial bus) 포트, PCIe(peripheral component interconnect express) 포트) 등과 같이 장치(400) 내 컴포넌트들 간에 데이터를 전달하는 통신 디바이스일 수 있다.

[0063] 모호함을 유발하지 않도록 설명을 용이하게 하기 위해, 본 개시에서 프로세서(402) 및 다른 데이터 처리 회로는 집합적으로 “데이터 처리 회로”로 지칭된다. 데이터 처리 회로는 그 전체가 하드웨어로서, 또는 소프트웨어, 하드웨어 또는 펌웨어의 조합으로서 구현될 수 있다. 게다가, 데이터 처리 회로는 단일의 독립 모듈일 수도 있고, 전체적으로 또는 부분적으로 장치(400)의 임의의 다른 컴포넌트 내에 조합될 수도 있다.

[0064] 장치(400)는 네트워크(예를 들어, 인터넷, 인트라넷, LAN(local area network), 모바일 통신 네트워크 등)와의 유선 또는 무선 통신을 제공하기 위해 네트워크 인터페이스(406)를 추가적으로 포함할 수 있다. 일부 실시예에서, 네트워크 인터페이스(406)는 NIC(network interface controller), RF(radio frequency) 모듈, 트랜스폰더, 트랜시버, 모뎀, 라우터, 게이트웨이, 유선 네트워크 어댑터, 무선 네트워크 어댑터, 블루투스 어댑터, 적외선 어댑터, 근거리 통신(“NFC”) 어댑터, 셀룰러 네트워크 칩 등의 임의의 수의 임의의 조합을 포함할 수 있다.

[0065] 일부 실시예에서, 선택적으로, 장치(400)는 하나 이상의 주변 디바이스로서의 접속을 제공하기 위해 주변 인터페이스(408)를 추가적으로 포함할 수 있다. 도 4에 도시된 바와 같이, 주변 디바이스는 커서 제어 디바이스(예를 들어, 마우스, 터치 패드, 또는 터치 스크린), 키보드, 디스플레이(예를 들어, 음극선관 디스플레이, 액정 디스플레이, 또는 발광 다이오드 디스플레이), 비디오 입력 디바이스(예를 들어, 카메라 또는 비디오 아카이브에 커플링된 입력 인터페이스) 등을 포함할 수 있지만, 이에 국한되는 것은 아니다.

[0066] 비디오 코덱(예를 들어, 프로세스(200A, 200B, 300A, 또는 300B)를 수행하는 코덱)은 장치(400) 내 임의의 소프트웨어 또는 하드웨어 모듈의 임의의 조합으로서 구현될 수 있음을 이해해야 한다. 예를 들어, 프로세스(200A, 200B, 300A, 또는 300B)의 일부 또는 전체 단계는 메모리(404) 내에 로딩될 수 있는 프로그램 명령어와 같이, 장치(400)의 하나 이상의 소프트웨어 모듈로서 구현될 수 있다. 또 다른 예시로, 프로세스(200A, 200B, 300A, 또는 300B)의 일부 또는 전체 단계는 특수화된 데이터 처리 회로(예를 들어, FPGA, ASIC, NPU 등)과 같이, 장치(400)의 하나 이상의 하드웨어 모듈로서 구현될 수 있다.

[0067] 도 5는 본 개시의 일부 실시예에 따른, 예시적인 LMCS(luma mapping with chroma scaling) 프로세스(500)의 개략도를 나타낸다. 예를 들어, 프로세스(500)는 하이브리드 비디오 코딩 표준(예를 들어, H.26x 시리즈)에 따른 디코더에 의해 사용될 수 있다. LMCS는 도 2b의 루프 필터(232) 이전에 적용되는 신규한 처리 블록이다. LMCS는 리셰이퍼(reshaper)로 지칭될 수도 있다.

[0068] LMCS 프로세스(500)는 적응형 구분적 선형 모델(adaptive piecewise linear model)에 기초한 루마 성분 값의 인-루핑(in-looping) 매핑 및 크로마 성분의 루마-의존적 크로마 잔차 스케일링을 포함할 수 있다.

[0069] 도 5에 도시된 바와 같이, 적응형 구분적 선형 모델에 기초한 루마 성분 값의 인-루핑 매핑은 순방향 매핑 단계(518) 및 역방향 매핑 단계(508)를 포함할 수 있다. 크로마 성분의 루마-의존적 크로마 잔차 스케일링은 크로마 스케일링(520)을 포함할 수 있다.

[0070] 매핑 이전 또는 역방향 매핑 이후의 샘플 값은 원본 도메인 내 샘플로서 지칭될 수 있으며, 매핑 이후 그리고 역방향 매핑 이전의 샘플 값은 매핑된 도메인 내 샘플로서 지칭될 수 있다. LMCS가 활성화(enable)될 때, 프로세스(500)의 일부 단계들은 원본 도메인 대신 매핑된 도메인에서 수행될 수 있다. 순방향 매핑 단계(518)와 역방향 매핑 단계(508)는 SPS 플래그를 이용하여 시퀀스 수준에서 활성화/비활성화될 수 있는 것으로 이해된다.

[0071] 도 5에 도시된 바와 같이, $Q^{-1}T^{-1}$ 단계(504), 재구성(506), 및 인트라 예측(508)은 매핑된 도메인에서 수행된다. 예를 들어, $Q^{-1}T^{-1}$ 단계(504)는 역양자화 및 역변환을 포함할 수 있고, 재구성(506)은 루마 예측 및 루마 잔차의 가산을 포함할 수 있으며, 인트라 예측(508)은 루마 인트라 예측을 포함할 수 있다.

[0072] 루프 필터(510), 모션 보상 단계(516 및 530), 인트라 예측 단계(528), 재구성 단계(522), 및 디코딩된 화상 버

퍼(DPB: decoded picture buffer)(512 및 526)는 원본(즉, 매핑되지 않은) 도메인에서 수행된다. 일부 실시예에서, 루프 필터(510)는 더블로킹, 적응형 루프 필터(ALF: adaptive loop filter), 및 샘플 적응형 오프셋(SAO: sample adaptive offset)을 포함할 수 있고, 재구성 단계(522)는 크로마 잔차와 크로마 예측의 가산을 포함할 수 있으며, DPB(512 및 526)는 디코딩된 화상을 참조 화상으로서 저장할 수 있다.

[0073] 일부 실시예에서, 구분적 선형 모델로의 루마 매핑이 적용될 수 있다.

[0074] 루마 성분의 인-루프 매핑(in-loop mapping)은 압축 효율을 향상시키기 위해 동적 범위에 걸쳐 부호어를 재분배함으로써 입력 비디오의 신호 통계를 조정할 수 있다. 루마 매핑은 순방향 매핑 (“FwdMap”) 함수 및 대응하는 역방향 매핑 (“InvMap”) 함수에 의해 수행될 수 있다. “FwdMap” 함수는 16개의 동일한 조각(piece)을 갖는 구분적 선형 모델을 이용하여 시그널링된다(signaled). “InvMap” 함수는 시그널링될 필요가 없고, 그 대신 “FwdMap” 함수로부터 유도된다.

[0075] 구분적 선형 모델의 시그널링이 도 6의 표 1 및 도 7의 표 2에 도시되어 있다. 도 6의 표 1은 타일 그룹 헤더 선택스 구조를 나타낸다. 도 6에 도시되어 있는 바와 같이, 리셰이퍼 모델 파라미터 존재 플래그(reshaper model parameter presence flag)는 루마 매핑 모델이 현재 타일 그룹에 존재하는지를 표시하기 위해 시그널링된다. 루마 매핑 모델이 현재 타일 그룹에 존재할 경우, 대응하는 구분적 선형 모델 파라미터가 도 7의 표 2에 도시되어 있는 선택스 요소를 이용하여 tile_group_reshaper_model()에서 시그널링될 수 있다. 구분적 선형 모델은 입력 신호의 동적 범위를 16개의 동일 조각으로 구획화한다. 16개의 동일 조각의 각각에 대해, 조각의 선형 매핑 파라미터는 그 조각에 할당되는 부호어의 수를 사용하여 표현된다. 10-비트 입력을 예로 들어 보자. 16개 조각의 각각은 디폴트에 의해 조각에 할당되는 64개의 부호어를 가질 수 있다. 시그널링된 부호어의 수는, 스케일링 인자를 계산하고 이에 따라 조각에 대해 매핑 함수를 조정하는 데에 사용될 수 있다. 도 7의 표 2는 또한, 부호어의 개수가 시그널링되는 최소 인덱스 “reshaper_model_min_bin_idx” 및 최대 인덱스 “reshaper_model_max_bin_idx”를 포함하여 정의한다. 조각 인덱스가 reshaper_model_min_bin_idx보다 작거나 reshaper_model_max_bin_idx보다 클 경우, 그 조각에 대한 부호어의 개수가 시그널링되지 않고, 0으로 추론된다(즉, 그 조각에는 부호어가 할당되지 않고, 매핑/스케일링이 그 조각에는 적용되지 않는다).

[0076] tile_group_reshaper_model()이 시그널링된 후에, 타일 그룹 헤더 수준에서는 또 다른 리셰이퍼 활성화 플래그 “tile_group_reshaper_enable_flag”가 도 8에 도시된 것과 같은 LMCS 프로세스가 현재 타일 그룹에 적용되는지를 표시하기 위해 시그널링된다. 현재 타일 그룹에 대해 리셰이퍼가 활성화된 경우 그리고 현재 타일 그룹이 듀얼 트리 파티션을 사용하지 않는 경우, 크로마 스케일링이 현재 타일 그룹에 대해 활성화되는지 여부를 표시하기 위해 추가적인 크로마 스케일링 활성화 플래그가 시그널링된다. 듀얼 트리 파티션은 크로마 분리 트리(chroma separate tree)로 지칭될 수도 있다.

[0077] 구분적 선형 모델은 다음과 같이, 도 7의 표 2에 있는 시그널링된 선택스 요소에 기초하여 구성될 수 있다. “FwdMap” 구분적 선형 모델의 각 i번째 조각(i=0, 1, ..., 15)은 2개의 입력 피벗 지점(InputPivot[]) 및 2개의 출력(매핑된) 피벗 지점(MappedPivot[])에 의해 정의된다. InputPivot[]과 MappedPivot[]은 다음과 같이 시그널링된 선택스에 기초하여 연산된다(보편성을 잃지 않으면서, 입력 비디오의 비트 심도가 10-bit인 것으로 가정한다).

```

1) OrgCW = 64
2) For i = 0:16, InputPivot[ i ] = i * OrgCW
3) For i = reshaper_model_min_bin_idx : reshaper_model_max_bin_idx, SignaledCW[ i ] = OrgCW + (1 - 2*reshape_model_bin_delta_sign_CW[ i ]) * reshape_model_bin_delta_abs_CW[ i ];
4) For i = 0:16, MappedPivot[i] is calculated as follows:
   MappedPivot[ 0 ] = 0;
   for( i = 0; i < 16; i++)
       MappedPivot[ i + 1 ] = MappedPivot[ i ] + SignaledCW[ i ]

```

[0078]

[0079] 역방향 매핑 함수 “InvMap”는 또한 InputPivot[] 및 Mapped Pivot[]에 의해 정의될 수 있다. “FwdMap”와는 달리 “InvMap” 구분적 선형 모델의 경우, 각각의 조각의 2개의 입력 피벗 지점은 MappedPivot[]에 의해 정의될 수 있고, 2개의 출력 피벗 지점은 “FwdMap”의 정반대인 InputPivot[]에 의해 정의될 수 있다. 이러한 방식으로, “FwdMap”의 입력은 동일 조각으로 구획화되지만, “InvMap”의 입력은 동일 조각으로 구획화됨이 보장되지는 않는다.

[0080] 도 5에 도시된 바와 같이, 인터-코딩된 블록의 경우, 모션 보상된 예측이 매핑된 도메인에서 수행될 수 있다. 환언하면, 모션-보상 예측(516) 이후에, DPB 내 참조 신호에 기초하여 Y_{pred} 가 계산되고, 원본 도메인 내 루마 예측 블록을 매핑된 도메인에 매핑하기 위해 “FwdMap” 함수(518)가 적용될 수 있다($Y'_{pred}=FwdMap(Y_{pred})$). 인트라-코딩된 블록의 경우, 인트라 예측에서 사용되는 참조 샘플은 이미 매핑된 도메인에 있기 때문에 “FwdMap” 함수가 적용되지 않는다. 블록이 재구성(506)된 후에, Y_r 이 계산될 수 있다. 매핑된 도메인 내 재구성된 루마 값을 다시 원본 도메인 내 재구성된 루마 값으로 변환하기 위해 “InvMap” 함수(508)가 적용될 수 있다($\hat{Y}_i=InvMap(Y_r)$). “InvMap” 함수(508)는 인트라- 및 인터-코딩된 루마 블록 모두에 적용될 수 있다.

[0081] 루마 매핑 프로세스(순방향 또는 역방향 매핑)는 룩-업-테이블(LUP) 또는 온-더-플라이 계산(on-the-fly computation)을 이용하여 구현될 수 있다. LUT가 사용되는 경우, 표 “FwdMapLUT[]” 및 “InvMapLUT[]”가 타일 그룹 수준에서의 사용을 위해 사전-계산 및 사전-저장될 수 있고, 순방향 및 역방향 매핑은 단순히 각각, $FwdMap(Y_{pred})=FwdMapLUT[Y_{pred}]$ 및 $InvMap(Y_r)=InvMapLUT[Y_r]$ 로 구현될 수 있다. 대안적으로, 온-더-플라이 계산이 사용될 수 있다. 순방향 매핑 함수 “FwdMap”를 예로 들어 보자. 루마 샘플이 속하는 조각을 결정하기 위해, 샘플 값이 6비트(10-비트 비디오를 가정할 때 16개의 동일 조각에 해당됨)만큼 우측 시프트(shift)되어, 조각 인덱스를 획득할 수 있다. 그러면, 그 조각에 대한 선형 모델 파라미터가 검색(retrieve)되어 매핑된 루마 값을 연산하기 위해 온-더-플라이에 적용된다. FwdMap 함수는 다음과 같이 평가된다.

$$Y'_{pred} = FwdMap(Y_{pred}) = ((b2 - b1)/(a2 - a1)) * (Y_{pred} - a1) + b1$$

[0082] 여기서, “i”는 조각 인덱스, a1은 InputPivot[i], a2는 InputPivot[i+1], b1은 MappedPivot[i], 그리고 b2는 MappedPivot[i+1]이다.

[0084] “InvMap” 함수는, 매핑된 도메인 내 조각들이 동일 크기임이 보장되지 않기 때문에 샘플 값이 속하는 조각을 알아낼 때 단순 우측 비트-시프트 대신 조건부 확인이 적용되어야 한다는 것을 제외하고 유사한 방식으로 즉시(on-the-fly) 계산될 수 있다.

[0085] 일부 실시예에서, 루마-의존적 크로마 잔차 스케일링이 수행될 수 있다.

[0086] 크로마 잔차 스케일링은 루마 신호와 대응하는 크로마 신호 사이의 상호작용을 보상하도록 설계된다. 크로마 잔차 스케일링이 활성화되는지 여부 또한 타일 그룹 수준에서 시그널링된다. 도 6의 표 1에 도시된 바와 같이, 루마 매핑이 활성화될 경우 그리고 현재 타일 그룹에 듀얼 트리 파티션이 적용되지 않을 경우, 루마-의존적 크로마 잔차 스케일링이 활성화되는지 여부를 표시하기 위해 추가적인 플래그(예를 들어, tile_group_reshaper_chroma_residual_scale_flag)가 시그널링된다. 루마 매핑이 사용되지 않거나 듀얼 트리 파티션이 현재 타일 그룹에 사용되는 경우, 루마-의존적 크로마 잔차 스케일링은 자동적으로 비활성화된다. 또한, 루마-의존적 크로마 잔차 스케일링은 면적이 4 이하인 크로마 블록에 대해 비활성화될 수 있다.

[0087] 크로마 잔차 스케일링은 대응하는 루마 예측 블록(인트라- 및 인터 코딩된 블록 모두)의 평균 값에 의존한다. 루마 예측 블록의 평균으로, $avgY'$ 는 다음과 같이 계산될 수 있다.

$$avgY' = \frac{\sum_{i=0}^{width-1} \sum_{j=0}^{height-1} predSamples[i][j]}{width * height}$$

[0088] $C_{ScaleInv}$ 의 값은 다음과 같은 단계로 계산된다.

[0090] 1) InvMap 함수에 기초하여 $avgY'$ 가 속하는 구분적 선형 모델의 인덱스(Y_{idx})를 찾는다.

[0091] 2) $C_{ScaleInv}=cScaleInv[Y_{idx}]$, 이때 $cScaleInv[]$ 는 사전-계산된 16-조각의 LUT이다.

[0092] VTM4의 현재 LMCS 방법에서, i의 범위가 0 내지 15일 때 사전-계산된 LUT($cScaleInv[i]$)는 64-엔트리 정적 LUT($CrhomaResidualScaleLut$) 및 SignaledCW[i] 값에 기초하여 다음과 같이 유도된다.

[0093] $ChromaResidualScaleLut[64] = \{16384, 16384, 16384, 16384, 16384, 16384, 16384, 16384, 8192, 8192, 8192, 8192, 5461, 5461, 5461, 5461, 4096, 4096, 4096, 4096, 3277, 3277, 3277, 3277, 2731, 2731, 2731, 2731, 2341, 2341, 2341, 2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638, 1638, 1638, 1489, 1489, 1489, 1489, 1365,$

1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170, 1170, 1170, 1170, 1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024};

[0094] shiftC = 11

[0095] - (SignaledCW [i] == 0)의 경우,

[0096] cScaleInv [i] = (1 << shiftC)

[0097] - 그렇지 않을 경우,

[0098] cScaleInv [i] = ChromaResidualScaleLut[(SignaledCW [i] >> 1)-1]

[0099] 정적 테이블 CrhomaResidualScaleLUT[]는 64개 엔트리를 포함하고, SignaledCW[]는 (입력을 10-비트로 가정할 때) [0, 128] 범위에 있다. 따라서, 크로마 스케일링 인자 LUT(cScaleInv[])를 구성하기 위해 나누기 2(예를 들어, 1만큼의 우측 시프트)가 사용된다. 크로마 스케일링 인자 LUT(cScaleInv[])는 복수의 크로마 스케일링 인자를 포함할 수 있다. LUT(cScaleInv[])는 타일 그룹 수준에서 구성된다.

[0100] 현재 블록이 인트라, CIIP, 또는 인트라 블록 카피(IBC, 현재 화상 참조 또는 CPR로도 알려져 있음)모드를 사용하여 코딩되는 경우, $avgY'$ 는 인트라-, CIIP-, 또는 IBC-예측된 루마 값의 평균으로 계산된다. 그렇지 않을 경우, $avgY'$ 는 순방향 매핑된 인터 예측된 루마 값(즉, 도 5에 도시된 Y'_{pred})의 평균으로 계산된다. 샘플 기반(basis)으로 수행되는 루마 매핑과는 다르게, $C_{ScaleInv}$ 는 전체 크로마 블록에 대해 일정한 값이다. $C_{ScaleInv}$ 을 가지고, 디코더 측에서 크로마 잔차 스케일링이 다음과 같이 적용된다.

[0101] 디코더 측: $\hat{C}_{Res} = \hat{C}_{ResScale} * C_{ScaleInv}$

[0102] 이때, $\hat{C}_{ResScale}$ 는 현재 블록의 재구성된 크로마 잔차이다. 인코더 측에서, 순방향 크로마 잔차 스케일링은 (변환되고 양자화되기 전에) 다음과 같이 수행된다.

[0103] 인코더 측: $C_{ResScale} = C_{Res} * C_{Scale} = C_{Res} / C_{ScaleInv}$

[0104] 일부 실시예에서, 듀얼 트리 파티션이 수행될 수 있다.

[0105] VVC 드래프트 4에서, 코딩 트리 방식은 루마 및 크로마가 별도의 블록 트리 파티션을 갖도록 지원한다. 이는, 듀얼 트리 파티션으로도 불린다. 듀얼 트리 파티션 시그널링은 도 8의 표 3 및 도 9의 표 4에 도시되어 있다. SPS에서 시그널링되는 시퀀스 수준 제어 플래그인 “qtbtt_dual_tree_intra_flag”가 켜질 때 그리고 현재 타일 그룹이 인트라 코딩될 때, 블록 파티션 정보가 먼저 루마, 그리고 다음으로 크로마에 대해 별도로 시그널링될 수 있다. 인터 코딩된 타일 그룹(P 및 B타일 그룹)의 경우, 듀얼 트리 파티션은 허용되지 않는다. 별도의 블록 트리 모드가 적용될 때, 도 9의 표 4에 도시된 바와 같이, 하나의 코딩 트리 구조에 의해 루마 코딩 트리 블록(CTB)이 CU들로 구획되고, 또 다른 코딩 트리 구조에 의해 크로마 CTB들이 크로마 CU들로 구획된다.

[0106] 루마 및 크로마가 상이한 파티션을 갖는 것이 허용되면, 상이한 색상 성분 사이에 의존성을 갖는 코딩 툴에 대해 문제가 발생할 수 있다. 예를 들어, LMCS의 경우, 현재 블록에 적용될 스케일링 인자를 찾기 위해 대응하는 루마 블록의 평균 값이 사용된다. 듀얼 트리가 사용되면, 전체 CTU의 레이턴시(latency)가 발생될 수 있다. 예를 들어, CTU의 루마 블록이 수직으로 한 번 분할되고, CTU의 크로마 블록이 수평으로 한 번 분할되면, CTU의 두 루마 블록은 (크로마 스케일링 인자를 계산하는 데에 필요한 평균 값이 계산될 수 있도록) CTU의 제1 크로마 블록이 디코딩되기 전에 디코딩된다. VVC에서, CTU는 루마 샘플 단위로 128x128만큼 클 수 있다. 이렇게 큰 레이턴시는 하드웨어 디코더 파이프라인 설계에서 큰 문제가 될 수 있다. 따라서, VVC 드래프트 4는 듀얼 트리 파티션 및 루마-의존적 크로마 스케일링의 결합을 금지할 수 있다. 듀얼 트리 파티션이 현재 타일 그룹에 대해 활성화될 때, 크로마 스케일링은 꺼지도록 강요될 수 있다. LMCS의 루마 매핑 부분은 루마 성분에만 동작하고 크로스 색상 성분 의존성 문제는 가지지 않기 때문에, 듀얼 트리의 경우에는 여전히 허용됨에 유의한다. 더 우수한 코딩 효율을 달성하기 위해 색상 성분 간 의존성에 의존하는 코딩 툴의 또 다른 예시는 크로스 성분 선형 모델(CCLM)으로 불린다.

[0107] 따라서, 타일 그룹 수준 크로마 스케일링 인자 LUT(cScaleInv[])의 유도는 용이하게 확장할 수 없다. 유도 프로세스는 현재 64 엔트리를 갖는 일정한 크로마 LUT(ChromaResidualScaleLut)에 의존한다. 16개의 조각을 갖는

10-비트 비디오의 경우, 2로 나누는 추가적인 단계가 적용되어야 한다. 조각의 개수가 변하면, 예를 들어 16개의 조각 대신 8개의 조각이 사용되는 경우, 유도 프로세스는 2로 나누는 대신 4로 나누는 것이 적용되도록 변경되어야 한다. 이러한 추가적인 단계는 정확성 손실을 야기할 뿐 아니라, 불편하고 불필요하다.

[0108] 또한, 크로마 스케일링 인자를 획득하기 위해 사용되는 현재 크로마 블록의 조각 인덱스(Y_{idx})를 계산하기 위해, 전체 루마 블록의 평균 값이 사용될 수 있다. 이는 바람직하지 않으며, 대부분의 경우 불필요하다. 최대 CTU 크기가 128x128인 경우를 고려해보자. 이 경우, 평균 루마 값은 16384개(128x128)의 루마 샘플에 기초하여 계산되는데 이는 복잡하다. 또한, 128x128의 루마 블록 파티션이 인코더에 의해 선택되는 경우, 그 블록이 균일한(homogeneous) 콘텐츠를 포함하고 있을 가능성이 크다. 따라서, 루마 평균을 계산하는 데에는 블록 내 루마 샘플의 서브셋으로 충분할 수 있다.

[0109] 듀얼 트리 파티션에서, 크로마 스케일링은 하드웨어 디코더에 대한 잠재적 파이프라인 문제를 방지하기 위해 꺼질 수 있다. 하지만, 적용될 크로마 스케일링 인자를 표시하기 위해, 대응하는 루마 샘플을 이용하여 유도하는 대신, 명시적인 시그널링이 사용되는 경우에는 이러한 의존성이 방지될 수 있다. 인트라 코딩된 타일 그룹에서 크로마 스케일링을 활성화하는 것은 코딩 효율을 더욱 향상시킬 수 있다.

[0110] 구분적 선형 파라미터의 시그널링은 더욱 개선될 수 있다. 현재, 델타 부호어 값이 16개 조각의 각각에 대해 시그널링된다. 16개 조각에 대해 제한된 수의 상이한 부호어만이 사용되는 것이 종종 관찰되었다. 따라서, 스케일링 오버헤드가 더욱 감소될 수 있다.

[0111] 본 개시의 실시예들은 크로마 스케일링 LUT을 제거함으로써 비디오 콘텐츠를 처리하는 방법을 제공한다.

[0112] 전술한 바와 같이, 64개 엔트리를 갖는 크로마 LUT의 확장은 어려울 수 있으며, 다른 구분적 선형 모델(예를 들어, 8조각, 4조각, 64조각 등)이 사용되는 경우 문제가 될 수 있다. 이는, 크로마 스케일링 인자가 동일한 코딩 효율 달성을 위해 그와 대응하는 조각의 루마 스케일링 인자와 동일하게 설정될 수 있기 때문에 불필요하다. 본 개시의 일부 실시예에서, 크로마 스케일링 인자("chroma_scaling")는 현재 크로마 블록의 조각 인덱스(" Y_{idx} ")에 기초하여 다음과 같이 결정될 수 있다.

[0113] - $Y_{idx} > \text{reshaper_model_max_bin_idx}$, $Y_{idx} < \text{reshaper_model_min_bin_idx}$ 인 경우, 또는 $\text{SignaledCW}[Y_{idx}] = 0$ 인 경우, chroma_scaling을 디폴트로 설정한다. 즉, chroma_scaling=1.0이다.

[0114] - 그렇지 않을 경우, chroma_scaling을 $\text{SignaledCW}[Y_{idx}] / \text{OrgCW}$ 로 설정한다.

[0115] chroma_scaling=1.0일 경우, 스케일링이 적용되지 않는다.

[0116] 위에서 결정된 크로마 스케일링 인자는 프랙셔널 정밀도(fractional precision)를 가질 수 있다. 하드웨어/소프트웨어 플랫폼에 대한 의존성을 피하기 위해 고정 소수점 근사(fixed point approximation)가 적용될 수 있는 것으로 이해된다. 또한, 디코더 측에서 역방향 크로마 스케일링이 수행될 수 있다. 따라서, 나눗셈은 우측 시프트에 이어서 곱셈을 이용한 고정 소수점 산술에 의해 구현될 수 있다. 고정 소수점 정밀도에서 역방향 크로마 스케일링 인자("inverse_chroma_scaling[]")는 다음과 같이 고정 소수점 근사 내 비트의 개수("CSCALE_FP_PREC")에 기초하여 결정될 수 있다.

[0117]
$$\text{inverse_chroma_scaling}[Y_{idx}] = ((1 \ll (\text{luma_bit_depth} - \log_2(\text{TOTAL_NUMBER_PIECES}) + \text{CSCALE_FP_PREC})) + (\text{SignaledCW}[Y_{idx}] \gg 1)) / \text{SignaledCW}[Y_{idx}];$$

[0118] 이때, luma_bit_depth는 루마 비트 심도이고, TOTAL_NUMBER_PIECES는 VVC 드래프트 4에서 16으로 설정되어 있는 구분적 선형 모델 내 조각들의 총 개수이다. "inverse_chroma_scaling[]"의 값은 타일 그룹 당 한 번씩만 계산하면 되고, 위에서 나눗셈은 정수 나눗셈 연산이다.

[0119] 크로마 스케일링 및 역방향 스케일링 인자를 결정하기 위해 추가적인 양자화가 적용될 수 있다. 예를 들어, "SignaledCW"의 모든 짝수($2x_m$)에 대해 역방향 크로마 스케일링 인자가 계산될 수 있고, "SignaledCW"의 홀수($2x_m+1$) 값은 이웃하는 짝수 값 스케일링 인자의 크로마 스케일링 인자를 재사용한다. 환언하면, 다음 방식이 사용될 수 있다.

[0120] for($i = \text{reshaper_model_min_bin_idx}$; $i \leq \text{reshaper_model_max_bin_idx}$; $i++$)

[0121] {

- [0122] $\text{tempCW} = \text{SignaledCW}[i] \gg 1 \} \ll 1;$
- [0123] $\text{inverse_chroma_scaling}[i] = ((1 \ll (\text{luma_bit_depth} - \log_2(\text{TOTAL_NUMBER_PIECES}) + \text{CSCALE_FP_PREC})) + (\text{tempCW} \gg 1)) / \text{tempCW};$
- [0124] }
- [0125] 크로마 스케일링 인자의 양자화는 더 일반화될 수 있다. 예를 들어, 역방향 크로마 스케일링 인자(“ $\text{inverse_chroma_scaling}[]$ ”)는 동일한 크로마 스케일링 인자를 공유하는 다른 모든 이웃하는 값과 함께, “ SignaledCW ”의 모든 n 번째 값에 대해 계산될 수 있다. 예를 들어, “ n ”은 4로 설정될 수 있다. 따라서, 모든 4개의 이웃하는 부호어 값은 동일한 역방향 크로마 스케일링 인자 값을 공유할 수 있다. 일부 실시예에서, “ n ”의 값은 시프팅이 나눗셈 계산에 사용될 수 있도록 2의 거듭제곱(power)일 수 있다. $\log_2(n)$ 의 값은 LOG2_n 으로 표시하며, 위의 식 “ $\text{tempCW} = \text{SignaledCW}[i] \gg 1 \} \ll 1$ ”는 다음과 같이 수정될 수 있다.
- [0126] $\text{tempCW} = \text{SignaledCW}[i] \gg \text{LOG2_n} \} \ll \text{LOG2_n}.$
- [0127] 일부 실시예에서, LOG2_n 의 값은 구분적 선형 모델에서 사용되는 조각의 개수의 함수일 수 있다. 적은 수의 조각이 사용될 경우, 더 큰 LOG2_n 을 사용하는 것이 더 유리할 수 있다. 예를 들어, $\text{TOTAL_NUMBER_PIECES}$ 의 값이 16 이하일 경우, LOG2_n 은 $1 + (4 - \log_2(\text{TOTAL_NUMBER_PIECES}))$ 로 설정될 수 있다. $\text{TOTAL_NUMBER_PIECES}$ 의 값이 16보다 클 경우, LOG2_n 은 0으로 설정될 수 있다.
- [0128] 본 개시의 실시예는 루마 예측 블록의 평균을 단순화함으로써 비디오 콘텐츠를 처리하는 방법을 제공한다.
- [0129] 전술한 바와 같이, 현재 크로마 블록의 조각 인덱스(“ I_{idx} ”)를 결정하기 위해 대응하는 루마 블록의 평균 값이 사용될 수 있다. 하지만, 블록 크기가 큰 경우, 평균화 프로세스는 많은 수의 루마 샘플을 포함할 수 있다. 최악의 경우, 평균화 프로세스에 128x128개의 루마 샘플이 포함될 수 있다.
- [0130] 본 개시의 실시예들은 최악의 경우를 줄이기 위해 $N \times N$ 개의 루마 샘플(N 은 2의 거듭제곱)만을 이용하도록 단순화된 평균화 프로세스를 제공한다.
- [0131] 일부 실시예에서, 2-차원 루마 블록의 두 치수 모두가 사전 설정된 임계값(M) 이하가 아닐 경우(환언하면, 2개의 치수 중 적어도 하나가 M 보다 크면), 그 치수에서 M 개 위치만 사용하도록 “다운샘플링(downsampling)”이 적용될 수 있다. 보편성을 잃지 않으면서, 수평 방향 치수를 예시로 들어보자. 폭이 M 보다 클 경우, x 의 위치에 있는 샘플만이 평균화에 사용되며, $x = i \times (\text{width} \gg \log_2(M))$, $i = 0, \dots, M-1$ 이다.
- [0132] 도 10은 16x8 루마 블록의 평균을 계산하기 위해 제안된 단순화를 적용하는 것의 일 예시를 나타낸다. 이 예시에서, M 은 4로 설정되고, 블록 내 16개의 루마 샘플(빛금 처진 샘플)만이 평균화에 사용된다. 사전 설정된 임계값(M)은 4로 국한되지 않으며, M 은 2의 거듭제곱인 임의의 값으로 설정될 수 있는 것으로 이해된다. 예를 들어, 사전 설정된 임계값(M)은 1, 2, 4, 8 등일 수 있다.
- [0133] 일부 실시예에서, 루마 블록의 수평 방향 치수 및 수직 방향 치수는 상이한 사전 설정된 임계값(M)을 가질 수 있다. 환언하면, 평균화 동작의 최악의 경우는 $M_1 \times M_2$ 개 샘플을 사용할 수 있다.
- [0134] 일부 실시예에서, 샘플의 개수는 치수를 고려하지 않고 평균화 프로세스에서 제한될 수 있다. 예를 들어, 최대 16개의 샘플이 사용될 수 있는데, 이는 1x16, 16x1, 2x8, 8x2, 또는 4x4의 형태의 수평 또는 수직 방향 치수로 분배될 수 있으며, 현재 블록의 형상에 맞는 형태로 선택될 수 있다. 예를 들어, 블록이 좁고 높은 경우에는 2x8 샘플의 매트릭스가 사용될 수 있고, 블록이 넓고 짧은 경우에는 8x2 샘플의 매트릭스가 사용될 수 있으며, 블록이 정사각형인 경우에는 4x4 샘플의 매트릭스가 사용될 수 있다.
- [0135] 크기가 큰 블록이 선택되면, 블록 내 콘텐츠가 더 균일한 경향이 있는 것으로 이해된다. 따라서, 상기 단순화가 평균 값과 전체 루마 블록의 실제 평균 사이에 차이를 발생시킬 수는 있지만, 이러한 차이는 작을 수 있다.
- [0136] 게다가, 디코더-측 모션 벡터 정제(DMVR: decoder-side motion vector refinement)는 모션 보상이 적용되기 전에 디코더로 하여금 모션 탐색을 수행하여 모션 벡터를 유도하도록 요구한다. 따라서, DMVR 모드는 VVC 표준에서, 특히 디코더에 대해 복잡할 수 있다. VVC 표준에서 양방향 광학 흐름(BDOF: bi-directional optical flow) 모드는 상황을 더욱 복잡하게 할 수 있는데, 이는 BDOF가 루마 예측 블록을 획득하기 위해서 DMVR 이후에 적용될 필요가 있는 추가적인 순차적 프로세스이기 때문이다. 크로마 스케일링은 대응하는 루마 예측 블록의 평균 값을 요구하기 때문에, 평균 값이 계산될 수 있기 전에 DMVR 및 BDOF가 적용될 수 있다.

- [0137] 이러한 레이턴시 문제를 해결하기 위해, 본 개시의 일부 실시예에서는 DMVR 및 BDOF 이전에 평균 루마 값을 계산하기 위해 루마 예측 블록이 사용되며, 크로마 스케일링 인자를 획득하기 위해 평균 루마 값이 사용된다. 이는, 크로마 스케일링이 DMVR 및 BDOF 프로세스와 병렬로 적용될 수 있도록 하며, 이로 인해 레이턴시를 상당히 감소시킬 수 있다.
- [0138] 본 개시와 일관되게, 레이턴시 감소의 변형이 고려될 수 있다. 일부 실시예에서, 이러한 레이턴시 감소가 평균 루마 값을 계산하기 위해 루마 예측 블록의 일부만을 사용하는 전술한 단순화된 평균화 프로세스와 결합될 수도 있다. 일부 실시예에서, DMVR 프로세스 이후에 그리고 BDOF 프로세스 이전에, 평균 루마 값을 계산하기 위해 루마 예측 블록이 사용될 수 있다. 그 다음, 크로마 스케일링 인자를 획득하기 위해 평균 루마 값이 사용된다. 이러한 설계는 크로마 스케일링 인자 결정의 정확도를 유지하면서 BDOF 프로세스와 병렬로 크로마 스케일링이 적용될 수 있도록 한다. DMVR 프로세스는 모션 벡터를 정제할 수 있기 때문에, DMVR 프로세스 이후에 정제된 모션 벡터와 함께 예측 샘플을 이용하는 것은, DMVR 프로세스 이전에 모션 벡터와 함께 예측 샘플을 이용하는 것보다 더 정확할 수 있다.
- [0139] 또한, VVC 표준에서, CU 선택스 구조(“coding_unit()”)는 현재 CU에 임의의 비-제로(non-zero) 잔차 계수가 있는지 여부를 표시하기 위해 선택스 요소(“cu_cbf”)를 포함한다. TU 수준에서, TU 선택스 구조(“transform_unit()”)는 현재 TU에 임의의 비-제로 크로마(Cb 또는 Cr) 잔차 계수가 있는지 여부를 표시하기 위해 선택스 요소(“tu_cbf_cb” 및 “tu_cbf_cr”)를 포함한다. 종래에는, VVC 드래프트 4에서 크로마 스케일링이 타일 그룹 수준에서 활성화되는 경우 대응하는 루마 블록의 평균화가 항상 호출된다.
- [0140] 본 개시의 실시예들은 루마 평균화 프로세스를 우회함으로써 비디오 콘텐츠를 처리하는 방법을 추가적으로 제공한다. 개시된 실시예와 일관되게, 크로마 스케일링 프로세스가 잔차 크로마 계수에 적용되기 때문에, 비-제로 크로마 계수가 없을 경우, 루마 평균화 프로세스는 우회될 수 있다. 이는 다음과 같은 조건에 기초하여 결정될 수 있다.
- [0141] 조건 1: cu_cbf가 0과 동일함.
- [0142] 조건 2: tu_cbf_cr 및 tu_cbf_cb가 모두 0과 동일함.
- [0143] 전술한 바와 같이, “cu_cbf”는 현재 CU에 임의의 비-제로 잔차 계수가 있는지 여부를 표시할 수 있으며, “tu_cbf_cb” 및 “tu_cbf_cr”은 현재 TU에 임의의 비-제로 크로마(Cb 또는 Cr) 잔차 계수가 있는지 여부를 표시할 수 있다. 조건 1 또는 조건 2가 충족되면, 루마 평균화 프로세스는 우회될 수 있다.
- [0144] 일부 실시예에서, 평균 값을 유도하는 데에 예측 블록의 NxN 샘플만이 사용되며, 이는 평균화 프로세스를 단순화한다. 예를 들어, N이 1과 동일할 때, 예측 블록의 좌측 상부 샘플만이 사용된다. 하지만, 이러한 예측 블록을 이용한 단순화된 평균화 처리는 여전히 예측 블록이 생성되는 것을 요구하며, 이로 인해 레이턴시를 발생시킨다.
- [0145] 일부 실시예에서, 크로마 스케일링 인자를 생성하기 위해 참조 루마 샘플이 직접 사용될 수 있다. 이는, 디코더로 하여금 루마 예측 프로세스와 병렬로 스케일링 인자를 유도할 수 있도록 하며, 이로 인해 레이턴시가 감소된다. 이하에서는, 참조 루마 샘플을 이용한 인트라 예측 및 인터 예측을 개별적으로 설명할 것이다.
- [0146] 예시적인 인트라 예측에서, 동일한 화상에서 디코딩된 이웃하는 샘플들이 예측 블록을 생성하기 위해 참조 샘플로서 사용될 수 있다. 이러한 참조 샘플들은 예를 들어, 현재 블록의 상부, 현재 블록의 좌측, 또는 현재 블록의 좌측 상부에 있는 샘플을 포함할 수 있다. 이러한 참조 샘플들의 평균은 크로마 스케일링 인자를 유도하기 위해 사용될 수 있다. 일부 실시예에서, 이러한 참조 샘플들 중 일부의 평균이 사용될 수 있다. 예를 들어, 현재 블록의 좌측 상단 위치에 가장 근접한 K개의 참조 샘플(예를 들어, K=3)만 평균화된다.
- [0147] 예시적인 인터 예측에서, 예측 블록을 생성하기 위해 시간 참조 화상으로부터의 참조 샘플들이 사용될 수 있다. 이러한 참조 샘플들은 참조 화상 인덱스 및 모션 벡터에 의해 식별된다. 모션 벡터가 프랙셔널 정밀도를 갖는 경우 내삽법이 적용될 수 있다. 참조 샘플의 평균을 결정하는 데에 사용되는 참조 샘플들은 내삽법 이전 또는 이후의 참조 샘플을 포함할 수 있다. 내삽법 이전의 참조 샘플은 정수 정밀도로 떨어지는(clipped) 모션 벡터를 포함할 수 있다. 개시된 실시예들과 일관되게, 모든 참조 샘플들이 평균을 계산하는 데에 사용될 수 있다. 대안적으로, 평균을 계산하는 데에 참조 샘플들 중 일부(예를 들어, 현재 블록의 좌측 상부 위치에 대응하는 참조 샘플들)만이 사용될 수 있다.
- [0148] 도 5에 도시된 바와 같이, 원본 도메인에서 인터 예측이 수행되는 반면, 리샘플링된 도메인에서는 인트라 예측

(예를 들어, 인트라 예측(514 또는 528))이 수행될 수 있다. 따라서, 인트라 예측의 경우, 예측 블록에 순방향 매핑이 적용될 수 있고, 순방향 매핑 이후의 루마 예측 블록이 평균을 계산하는 데에 사용된다. 레이턴시를 감소시키기 위해, 순방향 매핑 이전의 예측 블록을 이용하여 평균이 계산될 수 있다. 예를 들어, 순방향 매핑 이전의 블록, 순방향 매핑 이전의 블록의 NxN 부분, 또는 순방향 매핑 이전의 블록의 좌측 상부 샘플이 사용될 수 있다.

[0149] 본 개시의 실시예들은 듀얼-트리 파티션에 대한 크로마 스케일링으로 비디오 콘텐츠를 처리하는 방법을 추가적으로 제공한다.

[0150] 루마에 대한 의존성이 하드웨어 설계 복잡성을 야기할 수 있기 때문에, 듀얼-트리 파티션을 활성화하는 크로마 스케일링이 인트라-코딩된 타일 그룹에 대해 꺼질 수 있다. 하지만, 이러한 제한은 코딩 효율 손실을 야기할 수 있다. $avgY'$ 를 계산하기 위해 대응하는 루마 블록의 샘플 값들을 평균화하고, 조각 인덱스(Y_{idx})를 결정하고, 크로마 스케일링 인자($inverse_chroma_scaling[Y_{idx}]$)를 획득하는 대신, 듀얼 트리 파티션의 경우에서의 루마 의존성을 방지하기 위해 크로마 스케일링 인자가 비트스트림에서 명시적으로 시그널링될 수 있다.

[0151] 크로마 스케일링 인덱스는 상이한 수준에서 시그널링될 수 있다. 예를 들어, 크로마 스케일링 인덱스는 도 11의 표 5에 도시된 바와 같이, 크로마 예측 모드와 함께 코딩 유닛(CU) 수준에서 시그널링될 수 있다. 현재 크로마 블록에 대한 크로마 스케일링 인자를 결정하기 위해 선택 요소($lmcs_scaling_factor_idx$)가 사용될 수 있다. $lmcs_scaling_factor_idx$ 가 존재하지 않을 경우, 현재 크로마 블록에 대한 크로마 스케일링 인자는 부동 소수점 정밀도(floating point precision)에서 1.0과 동일하도록, 또는 고정 소수점 정밀도(fixed point precision)에서 동등하게($1 < CSCALE_FP_PREC$) 추론될 수 있다. $lmcs_chroma_scaling_idx$ 에 대한 허용 값의 범위는 타일 그룹 수준에서 결정되며, 이에 대해서는 추후에 논의할 것이다.

[0152] $lmcs_chroma_scaling_idx$ 의 가능한 값에 따라, 시그널링 비용이 높아질 수 있으며, 특히 작은 블록의 경우 더 그러하다. 따라서, 본 개시의 일부 실시예에서, 도 11의 표 5의 시그널링 조건은 블록 크기 조건을 추가적으로 포함할 수 있다. 예를 들어, 이러한 (이텔릭체 및 회색 음영 처리로 강조되어 있는) 선택 요소($lmcs_chroma_scaling_idx$)는 현재 블록이 주어진 크로마 샘플 개수보다 많을 경우 또는 현재 블록이 주어진 폭(W)보다 넓은 폭 또는 주어진 높이(H)보다 큰 높이를 가질 경우에만 시그널링될 수 있다. 더 작은 블록의 경우, $lmcs_chroma_scaling_idx$ 가 시그널링되지 않으면, 그 크로마 스케일링 인자는 디코더 측에서 결정될 수 있다. 일부 실시예에서, 크로마 스케일링 인자는 부동 소수점 정밀도에서 1.0으로 설정될 수 있다. 일부 실시예에서, 디폴트 $lmcs_chroma_scaling_idx$ 값이 타일 그룹 헤더 수준에서 추가될 수 있다(도 6의 **Error! Reference source not found.** 1 참조). 시그널링된 $lmcs_chroma_scaling_idx$ 를 갖지 않는 작은 블록들은 이러한 타일 그룹 수준 디폴트 인덱스를 사용하여 대응하는 크로마 스케일링 인자를 유도할 수 있다. 일부 실시예에서, 작은 블록의 크로마 스케일링 인자는 명시적으로 시그널링된 스케일링 인자를 갖는 이웃들(예를 들어, 상부 또는 좌측 이웃들)로부터 이어받을 수 있다.

[0153] 이러한 선택 요소($lmcs_chroma_scaling_idx$)는 CU 수준에서 시그널링하는 것 이외에, CTU 수준에서도 시그널링될 수 있다. 하지만, VVC에서 최대 CTU 크기가 128x128로 주어졌을 때, CTU 수준에서 동일한 스케일링을 하는 것은 너무 거칠(coarse) 수 있다. 따라서, 본 개시의 일부 실시예에서, 이러한 선택 요소($lmcs_chroma_scaling_idx$)는 고정된 입상도(granularity)를 이용하여 시그널링될 수 있다. 예를 들어, CTU에서 각각의 16x16 영역에 대해 하나의 $lmcs_chroma_scaling_idx$ 가 시그널링되고 그 16x16 영역 내 모든 샘플들에 적용된다.

[0154] 현재 타일 그룹에 대한 $lmcs_chroma_scaling_idx$ 의 범위는 현재 타일 그룹에서 허용되는 크로마 스케일링 인자 값의 개수에 의존한다. 현재 타일 그룹에서 허용되는 크로마 스케일링 인자 값의 개수는 전술한 바와 같이 64-엔트리 크로마 LUT에 기초하여 결정될 수 있다. 대안적으로, 현재 타일 그룹에서 허용되는 크로마 스케일링 인자 값의 개수는 전술된 크로마 스케일링 인자 계산을 이용하여 결정될 수 있다.

[0155] 예를 들어, “양자화” 방법에서, $LOG2_n$ 의 값이 2로(즉, “n”은 4로) 설정될 수 있으며, 현재 타일 그룹의 구분적 선형 모델 내 각 조각의 부호어 할당은 다음과 같이 설정될 수 있다. {0, 65, 66, 64, 67, 62, 62, 64, 64, 64, 67, 64, 64, 62, 61, 0}. 그러면, 전체 타일 그룹에 대해 단 2개의 가능한 스케일링 인자 값이 존재하게 되는데, 이는 64 내지 67의 임의의 부호어 값이 동일한 스케일링 인자 값(프랙셔널 정밀도에서 1.0)을 가질 수 있고, 60 내지 63의 임의의 부호어 값이 동일한 스케일링 인자 값(프랙셔널 정밀도에서 60/64=0.9375)을 가질 수 있기 때문이다. 할당된 임의의 부호어를 갖지 않는 2개의 말단 조각의 경우, 크로마 스케일링 인자는 디

폴트로서 1.0으로 설정된다. 따라서, 이러한 예시에서, 현재 타일 그룹 내 블록에 대해 “lmcs_chroma_scaling_idx”를 시그널링하는 데에 1-비트면 충분하다.

- [0156] 구분적 선형 모델을 이용하여 가능한 크로마 스케일링 인자 값의 개수를 결정하는 것 이외에, 인코더는 타일 그룹 헤더에서 크로마 스케일링 인자 값들의 세트를 시그널링할 수 있다. 그러면, 블록 수준에서, 크로마 스케일링 인자 값들의 세트 및 블록에 대한 “lmcs_chroma_scaling_idx” 값을 이용하여 블록에 대한 크로마 스케일링 인자 값이 결정될 수 있다.
- [0157] “lmcs_chroma_scaling_idx”를 코딩하기 위해 CABAC 코딩이 적용될 수 있다. 블록의 CABAC 컨텍스트는 블록의 이웃하는 블록들의 “lmcs_chroma_scaling_idx”에 의존할 수 있다. 예를 들어, CABAC 컨텍스트를 형성하기 위해 좌측의 블록 또는 상부에 있는 블록이 사용될 수 있다. 이러한 “lmcs_chroma_scaling_idx”의 선택스 요소의 이진화 측면에서, VVC 드래프트 4 내 ref_idx_10 및 ref_idx_11 선택스 요소에 적용되는 동일한 트렁케이트된(truncated) 라이스 이진법(Rice binarization)이 “lmcs_chroma_scaling_idx”를 이진화하는 데에 사용될 수 있다.
- [0158] “chroma_scaling_idx”를 시그널링하는 것의 이점은, 인코더가 윌 왜곡 비용 측면에서 최고의 “lmcs_chroma_scaling_idx”를 선택할 수 있다는 점이다. 윌 왜곡 최적화를 이용한 “lmcs_chroma_scaling_idx”의 선택은 코딩 효율을 향상시킬 수 있어, 시그널링 비용 상승을 오프셋 하는 데에 도움이 될 수 있다.
- [0159] 본 개시의 실시예들은 LMCS 구분적 선형 모델의 시그널링으로 비디오 콘텐츠를 처리하는 방법을 추가적으로 제공한다.
- [0160] LMCS 방법이 16개 조각을 갖는 구분적 선형 모델을 이용함에도 불구하고, 타일 그룹 내 “SignaledCW[i]”의 고유 값의 개수는 16개보다 훨씬 적은 경향이 있다. 예를 들어, 16개 조각 중 일부는 디폴트 개수의 부호어(“OrgCW”)를 사용할 수 있고, 16개 조각의 일부는 서로와 동일한 개수의 부호어를 가질 수 있다. 따라서, LMCS 구분적 선형 모델을 시그널링하는 대안적인 방법은, 다수의 고유 부호어(“listUniqueCW[]”)를 시그널링하는 단계와, 현재 조각에 대한 “listUniqueCW[]”의 요소를 표시하기 위해 조각의 각각에 대한 인덱스를 전송하는 단계를 포함할 수 있다.
- [0161] 수정된 선택스 표가 도 12의 Error! Reference source not found에 제공되어 있다. 도 12의 표 6에서, 새로운 또는 수정된 선택스는 이탤릭체 및 회색 음영 처리로 강조되어 있다.
- [0162] 개시된 시그널링 방법의 의미(semantics)는 다음과 같으며, 변화에는 밑줄이 쳐져 있다.
- [0163] **resaper_model_min_bin_idx**는 리셰이퍼 구성 프로세스에서 사용될 최소 빈(또는 조각) 인덱스를 지정한다. **reshape_model_min_bin_idx**의 값은 0부터 **MaxBinIdx**를 포함한 범위 내에 있을 것이다. **MaxBinIdx**의 값은 15와 동일할 것이다.
- [0164] **resaper_model_delta_max_bin_idx**는 최대 허용된 빈(또는 조각) 인덱스 **MaxBinIdx** 마이너스 리셰이퍼 구성 프로세스에서 사용될 최대 빈 인덱스를 지정한다. **reshape_model_max_bin_idx**의 값은 **MaxBinIdx-reshape_model_delta_max_bin_idx**와 동일하게 설정된다.
- [0165] **reshape_model_bin_delta_abs_cw_prec_minus1 plus1**은 선택스 **reshape_model_bin_delta_abs_CW[i]**의 표현에 사용되는 비트의 개수를 지정한다.
- [0166] **resaper_model_bin_num_unique_cw_minus1 plus1**은 부호어 어레이 **listUniqueCW**의 크기를 지정한다.
- [0167] **resaper_model_bin_delta_abs_CW[i]**는 i번째 빈에 대한 델타 부호어 절대값을 지정한다.
- [0168] **resaper_model_bin_delta_sign_CW_flag[i]**는 다음과 같이, **reshape_model_bin_delta_abs_CW[i]**의 부호(sign)을 지정한다.
- [0169] - **reshape_model_bin_delta_sign_CW_flag[i]**가 0과 동일하면, 대응하는 가변 **RspDeltaCW[i]**는 양의 값이다.
- [0170] - 그렇지 않을 경우(**reshape_model_bin_delta_sign_CW_flag[i]**가 0이 아닐 경우), 대응하는 가변 **RspDeltaCW[i]**는 음의 값이다.
- [0171] **reshape_model_bin_delta_sign_CW_flag[i]**가 존재하지 않을 때, 이는 0과 동일한 것으로 추론된다.
- [0172] 가변 $RspDeltaCW[i]$ 는 $RspDeltaCW[i]=(1-$

$2 \times \text{reshape_model_bin_delta_sign_CW}[i] \times \text{reshape_model_bin_delta_abs_CW}[i]$ 로써 유도된다.

[0173] 가변 $\text{listUniqueCW}[0]$ 은 OrgCW 와 동일하게 설정된다. $i=1 \cdots \text{reshaper_model_bin_num_unique_cw_minus1}$ 일 때 ($i=1 \cdots \text{reshaper_model_bin_num_unique_cw_minus1}$, inclusive), 가변 $\text{listUniqueCW}[i]$ 은 다음과 같이 유도된다.

[0174] - 가변 OrgCW 가 $(1 \leq \text{BitDepth}_Y)/(\text{MaxBinIdx}+1)$ 와 동일하게 설정된다.

[0175] - $\text{listUniqueCW}[i] = \text{OrgCW} + \text{RspDeltaCW}[i-1]$

[0176] $\text{reshaper_model_bin_cw_idx}[i]$ 는 $\text{RspCW}[i]$ 를 유도하는 데에 사용되는 어레이 $\text{listUniqueCW}[]$ 의 인덱스를 지정한다. $\text{reshaper_model_bin_cw_idx}[i]$ 의 값은 0 내지 $(\text{reshaper_model_bin_num_unique_cw_minus1}+1)$ 을 포함한 범위 내에 있을 것이다.

[0177] $\text{RspCW}[i]$ 는 다음과 같이 유도된다.

[0178] - $\text{reshaper_model_min_bin_idx} \leq i \leq \text{reshaper_model_max_bin_idx}$ 일 경우, $\text{RspCW}[i] = \text{listUniqueCW}[\text{reshaper_model_bin_cw_idx}[i]]$ 이다.

[0179] - 그렇지 않을 경우, $\text{RspCW}[i] = 0$ 이다.

[0180] BitDepth_Y 의 값이 10과 같을 경우, $\text{RspCW}[i]$ 의 값은 32 내지 $2 \times \text{OrgCW} - 1$ 의 범위에 있을 수 있다.

[0181] 본 개시의 실시예들은 블록 수준에서 조건부 크로마 스케일링으로 비디오 콘텐츠를 처리하는 방법을 추가적으로 제공한다.

[0182] 도 6의 표 1에 도시된 바와 같이, 크로마 스케일링이 적용되는지 여부는 타일 그룹 수준에서 시그널링되는 “ $\text{tile_group_reshaper_chroma_residual_scale_flag}$ ”에 의해 결정될 수 있다.

[0183] 하지만, 블록 수준에서 크로마 스케일링을 적용할지 여부를 결정하는 것이 유용할 수 있다. 예를 들어, 일부 개시된 실시예에서, 크로마 스케일링이 현재 블록에 적용되는지를 표시하기 위해 CU 수준 플래그가 시그널링될 수 있다. CU 수준 플래그의 존재는 타일 그룹 수준 플래그(“ $\text{tile_group_reshaper_chroma_residual_scale_flag}$ ”)로 조건화될 수 있다. 즉, CU 수준 플래그는 크로마 스케일링이 타일 그룹 수준에서 허용될 때에만 시그널링될 수 있다. 인코더가 크로마 스케일링이 현재 블록에 대해 유리한지 여부에 기초하여 크로마 스케일링을 사용할지 여부를 선택하도록 허용될 때, 이는 상당한 시그널링 오버헤드를 야기할 수 있다.

[0184] 개시된 실시예들과 일관되게, 상기 시그널링 오버헤드를 방지하기 위해, 크로마 스케일링이 블록에 적용될지 여부가 블록의 예측 모드로 조건화될 수 있다. 예를 들어, 블록이 인터 예측되는 경우, 예측 신호는 양호한 경향이 있는데, 특히 참조 화상이 시간적 거리상 더 가까운 경우에 더 그러하다. 따라서, 잔차(residual)가 매우 작을 것으로 예측되기 때문에, 크로마 스케일링이 우회될 수 있다. 예를 들어, 더 높은 시간 수준에 있는 화상들은 시간적 거리상 가까운 참조 화상을 가지는 경향이 있다. 또한, 인근 참조 화상을 사용하는 화상들 내 블록의 경우, 크로마 스케일링이 비활성화될 수 있다. 이러한 조건이 충족되었는지 여부를 결정하기 위해 현재 화상과 블록의 참조 화상 사이의 화상 오더 카운트(POC: picture order count) 차이가 사용될 수 있다.

[0185] 일부 실시예에서, 모든 인터 코딩된 블록에 대해 크로마 스케일링이 비활성화될 수 있다. 일부 실시예에서, 크로마 스케일링은 VVC 표준에서 정의되는 결합된 인트라/인터 예측(CIIP) 모드에 대해 비활성화될 수 있다.

[0186] VVC 표준에서, CU 선택 구조(“ $\text{coding_unit}()$ ”)는 현재 CU에 임의의 비-제로 잔차 계수가 존재하는지 여부를 표시하기 위해 선택 요소(“ cu_cbf ”)를 포함한다. TU 수준에서, TU 선택 구조(“ $\text{transform_unit}()$ ”)는 현재 TU에 임의의 비-제로 크로마(Cb 또는 Cr) 잔차 계수가 존재하는지를 표시하기 위해 선택 요소들(“ tu_cbf_cb ” 및 “ tu_cbf_cr ”)을 포함한다. 크로마 스케일링 프로세스는 이러한 플래그들로 조건화될 수 있다. 전술한 바와 같이, 대응하는 루마 크로마 스케일링 프로세스의 평균화는 비-제로 잔차 계수가 존재하지 않는 경우에 호출될 수 있다. 평균화의 호출로, 크로마 스케일링 프로세스는 우회될 수 있다.

[0187] 도 13은 비디오 콘텐츠를 처리하는 컴퓨터-구현되는 방법(1300)의 흐름도를 나타낸다. 일부 실시예에서, 방법(1300)은 코덱(예를 들어, 도 2a 내지 도 2b의 인코더 또는 도 3a 내지 도 3b의 디코더)에 의해 수행될 수 있다. 예를 들어, 코덱은 비디오 시퀀스를 인코딩 또는 트랜스코딩하는 장치(예를 들어, 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 컴포넌트로서 구현될 수 있다. 일부 실시예에서, 비디오 시퀀스는 비압축된 비디오 시퀀스(예를 들어, 비디오 시퀀스(202)) 또는 디코딩되는 압축된 비디오 시퀀스(예를 들어, 비디오 스트림

(304))일 수 있다. 일부 실시예에서, 비디오 시퀀스는 장치의 프로세서(예를 들어, 프로세서(402))와 연관된 모니터링 디바이스(예를 들어, 도 4의 비디오 입력 디바이스)에 의해 캡처될 수 있는 모니터링 비디오 시퀀스일 수 있다. 비디오 시퀀스는 다수의 화상을 포함할 수 있다. 장치는 화상 수준에서 방법(1300)을 수행할 수 있다. 예를 들어, 장치는 방법(1300)에서 한 번에 하나의 화상을 처리할 수 있다. 또 다른 예시로서, 장치는 방법(1300)에서 한 번에 복수의 화상을 처리할 수 있다. 방법(1300)은 이하와 같은 단계를 포함할 수 있다.

[0188] 단계(1302)에서, 화상과 연관된 크로마 블록과 루마 블록이 수신될 수 있다. 화상은 크로마 성분 및 루마 성분과 연관될 수 있는 것으로 이해된다. 따라서, 화상은 크로마 샘플을 포함하는 크로마 블록과 루마 샘플을 포함하는 루마 블록과 연관될 수 있다.

[0189] 단계(1304)에서, 루마 블록과 연관된 루마 스케일링 정보가 결정될 수 있다. 일부 실시예에서, 루마 스케일링 정보는 화상의 데이터 스트림에서 시그널링되는 선택스 요소들 또는 화상의 데이터 스트림에서 시그널링되는 선택스 요소들에 기초하여 유도되는 변수들일 수 있다. 예를 들어, 루마 스케일링 정보는 위의 식에 기재된 “ $\text{reshape_model_bin_delta_sign_CW}[i]$ ” 및 $\text{reshape_model_bin_delta_abs_CW}[i]$ ” 및/또는 위의 식에 기재된 “ $\text{SignaledCW}[i]$ ” 등을 포함할 수 있다. 일부 실시예에서, 루마 스케일링 정보는 루마 블록에 기초하여 결정되는 변수를 포함할 수 있다. 예를 들어, 루마 블록에 인접한 루마 샘플들(예컨대 루마 블록의 상부에 있는 행 및 루마 블록의 좌측에 있는 열에서의 루마 샘플들)의 평균 값을 계산함으로써 평균 루마 값이 결정될 수 있다.

[0190] 단계(1306)에서, 루마 스케일링 정보에 기초하여 크로마 스케일링 인자가 결정될 수 있다.

[0191] 일부 실시예에서, 루마 블록의 루마 스케일링 인자는 루마 스케일링 정보에 기초하여 결정될 수 있다. 예를 들어, 위의 식 “ $\text{inverse_chromas_scaling}[i] = ((1 \ll (\text{luma_bit_depth} - \log_2(\text{TOTAL_NUMBER_PIECES}) + \text{CSCALE_FP_PREC})) + (\text{tempCW} \gg 1)) / \text{tempCW}$ ”에 따르면, 루마 스케일링 인자는 루마 스케일링 정보(예를 들어, “tempCW”)에 기초하여 결정될 수 있다. 그러면, 크로마 스케일링 인자는 루마 스케일링 인자의 값에 기초하여 더 결정될 수 있다. 예를 들어, 크로마 스케일링 인자는 루마 스케일링 인자의 값과 동일하게 설정될 수 있다. 크로마 스케일링 인자로서 설정되기 전에 루마 스케일링 인자의 값에 추가적인 계산이 적용될 수 있는 것으로 이해된다. 또 다른 예시로서, 크로마 스케일링 인자는 “ $\text{SignaledCW}[Y_{idx}] / \text{OrgCW}$ ”와 동일하게 설정될 수 있는데, 이때 현재 크로마 블록의 조각 인덱스(“ Y_{idx} ”)는 루마 블록과 연관된 평균 루마 값에 기초하여 결정될 수 있다.

[0192] 단계(1308)에서 크로마 블록은 크로마 스케일링 인자를 이용하여 처리될 수 있다. 예를 들어, 크로마 블록의 스케일링된 잔차를 생성하기 위해 크로마 블록의 잔차들이 크로마 스케일링 인자를 이용하여 처리될 수 있다. 크로마 블록은 Cb 크로마 성분 또는 Cr 크로마 성분일 수 있다.

[0193] 일부 실시예에서, 조건이 충족될 경우에 크로마 블록이 처리될 수 있다. 예를 들어, 조건은 비-제로 잔차를 갖지 않는 화상과 연관된 타깃 코딩 유닛; 또는 비-제로 크로마 잔차를 갖지 않는 화상과 연관된 타깃 변환 유닛을 포함할 수 있다. 비-제로 잔차를 갖지 않는 타깃 코딩 유닛은 타깃 코딩 유닛의 제1 코딩된 블록 플래그의 값에 기초하여 결정될 수 있다. 그리고 비-제로 크로마 잔차를 갖지 않는 타깃 변환 유닛은 제1 성분에 대한 제2 코딩된 블록 플래그 및 타깃 변환 유닛의 제2 성분에 대한 제3 코딩된 블록 플래그의 값들에 기초하여 결정될 수 있다. 예를 들어, 제1 성분은 Cb 성분일 수 있고, 제2 성분은 Cr 성분일 수 있다.

[0194] 방법(1300)의 각 단계는 독립적인 방법으로서 실행될 수 있는 것으로 이해된다. 예를 들어, 단계(1308)에서 설명되는 크로마 스케일링 인자를 결정하는 방법은 독립적인 방법으로서 실행될 수 있다.

[0195] 도 14는 비디오 콘텐츠를 처리하는 컴퓨터-구현되는 방법(1400)의 흐름도를 나타낸다. 일부 실시예에서, 방법(1300)은 코덱(예를 들어, 도 2a 내지 도 2b의 인코더 또는 도 3a 내지 도 3b의 디코더)에 의해 수행될 수 있다. 예를 들어, 코덱은 비디오 시퀀스를 인코딩 또는 트랜스코딩하는 장치(예를 들어, 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 컴포넌트로서 구현될 수 있다. 일부 실시예에서, 비디오 시퀀스는 비압축된 비디오 시퀀스(예를 들어, 비디오 시퀀스(202)) 또는 디코딩되는 압축된 비디오 시퀀스(예를 들어, 비디오 스트림(304))일 수 있다. 일부 실시예에서, 비디오 시퀀스는 장치의 프로세서(예를 들어, 프로세서(402))와 연관된 모니터링 디바이스(예를 들어, 도 4의 비디오 입력 디바이스)에 의해 캡처될 수 있는 모니터링 비디오 시퀀스일 수 있다. 비디오 시퀀스는 다수의 화상을 포함할 수 있다. 장치는 화상 수준에서 방법(1400)을 수행할 수 있다. 예를 들어, 장치는 방법(1400)에서 한 번에 하나의 화상을 처리할 수 있다. 또 다른 예시로서, 장치는 방법(1400)에서 한 번에 복수의 화상을 처리할 수 있다. 방법(1400)은 이하와 같은 단계를 포함할 수 있다.

- [0196] 단계(1402)에서, 화상과 연관된 크로마 블록 및 루마 블록이 수신될 수 있다. 화상은 크로마 성분 및 루마 성분과 연관될 수 있는 것으로 이해된다. 따라서, 화상은 크로마 샘플을 포함하는 크로마 블록과 루마 샘플을 포함하는 루마 블록과 연관될 수 있다. 일부 실시예에서, 루마 블록은 $N \times M$ 개의 루마 샘플을 포함할 수 있다. N 은 루마 블록의 폭일 수 있고, M 은 루마 블록의 높이일 수 있다. 전술한 바와 같이, 루마 블록의 루마 샘플들은 타깃 크로마 블록의 조각 인덱스를 결정하기 위해 사용될 수 있다. 따라서, 비디오 시퀀스의 화상과 연관된 루마 블록이 수신될 수 있다. N 및 M 은 동일한 값을 가질 수 있는 것으로 이해된다.
- [0197] 단계(1404)에서, N 및 M 중 적어도 하나가 임계값보다 큰 것에 응답하여, $N \times M$ 루마 샘플의 서브세트가 선택될 수 있다. 조각 인덱스 결정을 신속화하기 위해, 특정 조건이 충족될 때, 루마 블록은 “다운샘플링(downsampled)” 될 수 있다. 환언하면, 루마 블록 내 루마 샘플들의 서브세트가 조각 인덱스를 결정하는 데에 사용될 수 있다. 일부 실시예에서, 특정 조건이란 N 및 M 중 적어도 하나가 임계값보다 큰 것이다. 일부 실시예에서, 임계값은 N 및 M 중 적어도 하나에 기초할 수 있다. 임계값은 2의 거듭제곱일 수 있다. 예를 들어, 임계값은 4, 8, 16 등일 수 있다. 일 예시로서 4를 상정할 때, N 또는 M 이 4보다 크면 루마 샘플들의 서브세트가 선택될 수 있다. 도 10의 예시에서, 루마 블록의 폭 및 높이는 모두 임계값인 4보다 크므로, 4×4 샘플의 서브셋이 선택된다. 2×8 , 1×16 등의 서브세트가 처리를 위해 선택될 수도 있는 것으로 이해된다.
- [0198] 단계(1406)에서, $N \times M$ 루마 샘플의 서브세트의 평균 값이 결정될 수 있다.
- [0199] 일부 실시예에서, 평균 값을 결정하는 단계는, 제2 조건이 충족되는지를 결정하는 것; 및 제2 조건이 충족된다면 결정에 응답하여, $N \times M$ 루마 샘플의 서브세트의 평균 값을 결정하는 것을 추가적으로 포함할 수 있다. 예를 들어, 제2 조건은 비-제로 잔차 계수를 갖지 않는 화상과 연관된 타깃 코딩 유닛; 또는 비-제로 크로마 잔차 계수를 갖지 않는 타깃 코딩 유닛을 포함할 수 있다.
- [0200] 단계(1408)에서, 평균값에 기초하여 크로마 스케일링 인자가 결정될 수 있다. 일부 실시예에서, 크로마 스케일링 인자를 결정하기 위해 평균값에 기초하여 크로마 블록의 조각 인덱스가 결정되고, 크로마 블록의 조각 인덱스가 제1 조건을 만족하는지 여부가 결정되며, 그 다음 크로마 블록의 조각 인덱스가 제1 조건을 만족함에 응답하여, 크로마 스케일링 인자가 디폴트 값으로 설정될 수 있다. 디폴트 값은 크로마 스케일링이 적용되지 않음을 표시할 수 있다. 예를 들어, 디폴트 값은 프랙셔널 정밀도로 1.0일 수 있다. 디폴트 값에 고정 소수점 근사가 적용될 수 있는 것으로 이해된다. 크로마 블록의 조각 인덱스가 제1 조건을 만족하지 못함에 응답하여, 크로마 스케일링 인자는 평균 값에 기초하여 결정될 수 있다. 보다 구체적으로, 크로마 스케일링 인자는 $\text{SignaledCW}[Y_{\text{idx}}]/\text{OrgCW}$ 로 설정될 수 있으며, 타깃 크로마 블록의 조각 인덱스(“ Y_{idx} ”)가 대응하는 루마 블록의 평균값에 기초하여 결정될 수 있다.
- [0201] 일부 실시예에서, 제1 조건은 크로마 블록의 조각 인덱스가 시그널링되는 부호어의 최대 인덱스보다 큰 것 또는 시그널링되는 부호어의 최소 인덱스보다 작은 것을 포함할 수 있다. 시그널링되는 부호어의 최대 인덱스와 최소 인덱스는 다음과 같이 결정될 수 있다.
- [0202] 부호어는 입력 신호(예를 들어, 루마 샘플)에 기초한 구분적 선형 모델(예를 들어, LMCS)을 이용하여 생성될 수 있다. 전술한 바와 같이, 입력 신호들의 동적 범위가 여러 조각(예를 들어, 16개 조각)으로 구획화될 수 있고, 입력 신호들의 각 조각은 출력으로서 부호어의 빈을 생성하는 데에 사용될 수 있다. 따라서, 부호어의 각 빈은 입력 신호의 조각에 대응하는 빈 인덱스를 가질 수 있다. 이러한 예시에서, 빈 인덱스의 범위는 0 내지 15일 수 있다. 일부 실시예에서, 출력(예를 들어, 부호어)의 값은 최소값(예를 들어, 0)과 최대값(예를 들어, 255) 사이이고, 최소값과 최대값 사이의 값을 갖는 복수의 부호어가 시그널링될 수 있다. 그리고 시그널링되는 복수의 부호어의 빈 인덱스가 결정될 수 있다. 시그널링되는 복수의 부호어의 빈 인덱스 중에, 시그널링되는 복수의 부호어의 빈들의 최대 빈 인덱스와 최소 빈 인덱스가 추가적으로 결정될 수 있다.
- [0203] 크로마 스케일링 인자 이외에, 방법(1400)은 시그널링되는 복수의 부호어의 빈들에 기초하여 루마 스케일링 인자를 추가적으로 결정할 수 있다. 루마 스케일링 인자는 역방향 크로마 스케일링 인자로서 사용될 수 있다. 루마 스케일링 인자를 결정하는 식은 전술되어 있으며, 그 설명은 여기서는 생략한다. 일부 실시예에서, 복수의 수의 이웃하는 시그널링된 부호어는 루마 스케일링 인자를 공유한다. 예를 들어, 2개의 또는 4개의 이웃하는 시그널링된 부호어는 동일한 스케일링 인자를 공유할 수 있으며, 이는 루마 스케일링 인자를 결정하는 부담을 줄일 수 있다.
- [0204] 단계(1410)에서, 크로마 블록은 크로마 스케일링 인자를 이용하여 처리될 수 있다. 도 5를 참조하여 전술한 바와 같이, 복수의 크로마 스케일링 인자가 타일 그룹 수준에서 크로마 스케일링 인자 LUT를 구성할 수 있고, 디

코더 측에서 타깃 블록의 재구성된 크로마 잔차에 적용될 수 있다. 유사하게, 크로마 스케일링 인자들은 인코더 측에서 적용될 수도 있다.

- [0205] 방법(1400)의 각 단계는 독립적인 방법으로서 실행될 수 있는 것으로 이해된다. 예를 들어, 단계(1308)에서 설명된 크로마 스케일링 인자를 결정하는 방법은 독립적인 방법으로서 실행될 수 있다.
- [0206] 도 15는 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법(1500)의 흐름도를 나타낸다. 일부 실시예에서, 방법(1500)은 코덱(예를 들어, 도 2a 내지 도 2b의 인코더 또는 도 3a 내지 도 3b의 디코더)에 의해 수행될 수 있다. 예를 들어, 코덱은 비디오 시퀀스를 인코딩 또는 트랜스코딩하는 장치(예를 들어, 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 컴포넌트로써 구현될 수 있다. 일부 실시예에서, 비디오 시퀀스는 비압축된 비디오 시퀀스(예를 들어, 비디오 시퀀스(202)) 또는 디코딩되는 압축된 비디오 시퀀스(예를 들어, 비디오 스트림(304))일 수 있다. 일부 실시예에서, 비디오 시퀀스는 장치의 프로세서(예를 들어, 프로세서(402))와 연관된 모니터링 디바이스(예를 들어, 도 4의 비디오 입력 디바이스)에 의해 캡처될 수 있는 모니터링 비디오 시퀀스일 수 있다. 비디오 시퀀스는 다수의 화상을 포함할 수 있다. 장치는 화상 수준에서 방법(1500)을 수행할 수 있다. 예를 들어, 장치는 방법(1500)에서 한 번에 하나의 화상을 처리할 수 있다. 또 다른 예시로서, 장치는 방법(1500)에서 한 번에 복수의 화상을 처리할 수 있다. 방법(1500)은 이하와 같은 단계를 포함할 수 있다.
- [0207] 단계(1502)에서, 수신된 비디오 데이터에 크로마 스케일링 인덱스가 존재하는지 여부가 결정될 수 있다.
- [0208] 단계(1504)에서, 크로마 스케일링 인덱스가 수신된 비디오 데이터에 존재하지 않는다는 결정에 응답하여, 수신된 비디오 데이터에 크로마 스케일링이 적용되지 않는다고 결정될 수 있다.
- [0209] 단계(1506)에서, 크로마 스케일링이 수신된 비디오 데이터에 존재한다는 결정에 응답하여, 크로마 스케일링 인덱스에 기초하여 크로마 스케일링 인자가 결정될 수 있다.
- [0210] 도 16은 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법(1600)의 흐름도를 나타낸다. 일부 실시예에서, 방법(1600)은 코덱(예를 들어, 도 2a 내지 도 2b의 인코더 또는 도 3a 내지 도 3b의 디코더)에 의해 수행될 수 있다. 예를 들어, 코덱은 비디오 시퀀스를 인코딩 또는 트랜스코딩하는 장치(예를 들어, 장치(400))의 하나 이상의 소프트웨어 또는 하드웨어 컴포넌트로써 구현될 수 있다. 일부 실시예에서, 비디오 시퀀스는 비압축된 비디오 시퀀스(예를 들어, 비디오 시퀀스(202)) 또는 디코딩되는 압축된 비디오 시퀀스(예를 들어, 비디오 스트림(304))일 수 있다. 일부 실시예에서, 비디오 시퀀스는 장치의 프로세서(예를 들어, 프로세서(402))와 연관된 모니터링 디바이스(예를 들어, 도 4의 비디오 입력 디바이스)에 의해 캡처될 수 있는 모니터링 비디오 시퀀스일 수 있다. 비디오 시퀀스는 다수의 화상을 포함할 수 있다. 장치는 화상의 수준에서 방법(1600)을 수행할 수 있다. 예를 들어, 장치는 방법(1600)에서 한 번에 하나의 화상을 처리할 수 있다. 또 다른 예시로서, 장치는 방법(1600)에서 한 번에 복수의 화상을 처리할 수 있다. 방법(1600)은 이하의 단계들을 포함할 수 있다.
- [0211] 단계(1602)에서, 입력 비디오 신호의 동적 범위에 대해 사용되는 복수의 고유한 부호어가 수신될 수 있다.
- [0212] 단계(1604)에서, 인덱스가 수신될 수 있다.
- [0213] 단계(1606)에서, 복수의 고유한 부호어 중 적어도 하나가 인덱스에 기초하여 선택될 수 있다.
- [0214] 단계(1608)에서, 선택된 적어도 하나의 부호어에 기초하여 크로마 스케일링 인자가 결정될 수 있다.
- [0215] 일부 실시예에서, 명령어를 포함하는 비-일시적인 컴퓨터-판독 가능 저장 매체 또한 제공되며, 명령어는 전술한 방법을 수행하기 위한 디바이스(예컨대 개시되어 있는 인코더 또는 디코더)에 의해 실행될 수 있다. 비-일시적인 매체의 보편적인 형태는 예를 들어, 플로피 디스크, 플래시블 디스크, 하드 디스크, SSD(solid-state disk), 자성 테이프, 또는 임의의 다른 자성 데이터 저장 매체, CD-ROM, 임의의 다른 광학 데이터 저장 매체, 홀 패턴을 갖는 임의의 물리적 매체, RAM, PROM, EPROM, FLASH-EPROM 또는 임의의 다른 플래시 메모리, NVRAM, 캐쉬, 레지스터, 임의의 다른 메모리 칩 또는 카트리지, 및 그 네트워크화된 버전을 포함한다. 디바이스는 하나 이상의 프로세서(CPU), 입력/출력 인터페이스, 네트워크 인터페이스, 및/또는 메모리를 포함할 수 있다.
- [0216] 전술된 실시예들은 하드웨어, 또는 소프트웨어(프로그램 코드), 또는 하드웨어와 소프트웨어의 조합에 의해 구현될 수 있는 것으로 이해된다. 소프트웨어에 의해 구현되는 경우, 이는 전술된 컴퓨터-판독 가능 매체에 저장될 수 있다. 프로세서에 의해 실행될 때, 소프트웨어는 개시된 방법을 수행할 수 있다. 본 개시에 설명되는 연산 유닛들과 다른 기능적 유닛들은 하드웨어, 또는 소프트웨어, 또는 하드웨어와 소프트웨어의 조합에 의해 구현될 수 있다. 통상의 기술자는 또한, 전술된 모듈/유닛 중 다수의 것이 하나의 모듈/유닛으로 조합될 수 있고,

전술된 각각의 모듈/유닛이 복수의 서브-모듈/서브-유닛으로 더 세분화될 수 있음을 이해할 것이다.

- [0217] 실시예들은 다음의 조항을 이용하여 추가적으로 설명될 수 있다.
- [0218] 1. 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법으로서,
- [0219] 화상과 연관된 크롬 블록과 루마 블록을 수신하는 단계;
- [0220] 루마 블록과 연관된 루마 스케일링 정보를 결정하는 단계;
- [0221] 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하는 단계; 및
- [0222] 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계를 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.
- [0223] 2. 제1항에 따른 방법에 있어서,
- [0224] 루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하는 단계는,
- [0225] 루마 스케일링 정보에 기초하여 루마 블록의 루마 스케일링 인자를 결정하는 단계;
- [0226] 루마 스케일링 인자의 값에 기초하여 크로마 스케일링 인자를 결정하는 단계를 더 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.
- [0227] 3. 제2항에 따른 방법에 있어서,
- [0228] 루마 스케일링 인자의 값에 기초하여 크로마 스케일링 인자를 결정하는 단계는,
- [0229] 크로마 스케일링 인자를 루마 스케일링 인자의 값과 동일하게 설정하는 단계를 더 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.
- [0230] 4. 제1항 내지 제3항 중 어느 한 항에 따른 방법에 있어서,
- [0231] 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계는,
- [0232] 제1 조건이 충족되는지 결정하는 단계; 및
- [0233] 제1 조건이 충족된다는 결정에 응답하여, 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 것; 또는
- [0234] 제1 조건이 충족되지 않는다는 결정에 응답하여, 크로마 스케일링 인자를 이용한 크로마 블록의 처리를 우회하는 것
- [0235] 중 하나를 수행하는 단계를 더 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.
- [0236] 5. 제4항에 따른 방법에 있어서,
- [0237] 제1 조건은,
- [0238] 비-제로 잔차를 가지지 않는 화상과 연관된 타깃 코딩 유닛; 또는
- [0239] 비-제로 크로마 잔차를 가지지 않는 화상과 연관된 타깃 변환 유닛을 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.
- [0240] 6. 제5항에 따른 방법에 있어서,
- [0241] 비-제로 잔차를 가지지 않는 타깃 코딩 유닛은 타깃 코딩 유닛의 제1 코딩된 블록 플래그의 값에 기초하여 결정되고,
- [0242] 비-제로 크로마 잔차를 가지지 않는 타깃 변환 유닛은 제1 크로마 성분에 대한 제2 코딩된 블록 플래그 및 타깃 변환 유닛의 제2 루마 크로마 성분에 대한 제3 코딩된 블록 플래그에 기초하여 결정되는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.
- [0243] 7. 제6항에 따른 방법에 있어서,
- [0244] 제1 코딩된 블록 플래그의 값은 0이고; 및
- [0245] 제2 코딩된 블록 플래그 및 제3 코딩된 블록 플래그의 값이 0인, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되

는 방법.

8. 제1항 내지 제7항 중 어느 한 항에 따른 방법에 있어서,

크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계는,

크로마 스케일링 인자를 이용하여 크로마 블록의 잔차를 처리하는 단계를 포함하는, 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현되는 방법.

9. 비디오 콘텐츠 처리 장치로서,

명령어의 세트를 저장하는 메모리; 및

메모리에 커플링되는 프로세서를 포함하고, 프로세서는 장치로 하여금,

화상과 연관된 크롬 블록과 루마 블록을 수신하고;

루마 블록과 연관된 루마 스케일링 정보를 결정하고,

루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하고, 및

크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 것을 수행하게 하는 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

10. 제9항에 따른 장치에 있어서,

루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정함에 있어서, 프로세서는 장치로 하여금,

루마 스케일링 정보에 기초하여 루마 블록의 루마 스케일링 인자를 결정하고;

루마 스케일링 인자의 값에 기초하여 크로마 스케일링 인자를 결정하는 것을 더 수행하게 하는 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

11. 제10항에 따른 장치에 있어서,

루마 스케일링 인자의 값에 기초하여 크로마 스케일링 인자를 결정함에 있어서, 프로세서는 장치로 하여금,

루마 스케일링 인자의 값과 동일하게 크로마 스케일링 인자를 설정하는 것을 더 수행하게 하는 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

12. 제9항 내지 제11항 중 어느 한 항에 따른 장치에 있어서,

크로마 스케일링 인자를 이용하여 크로마 블록을 처리함에 있어서, 프로세서는 장치로 하여금,

제1 조건이 충족되었는지 결정하고; 및

제2 조건이 충족된다는 결정에 응답하여, 크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 것; 또는

제2 조건이 충족되지 않는다는 결정에 응답하여, 크로마 스케일링 인자를 이용한 크로마 블록의 처리를 우회하는 것

중 하나를 수행하는 것을 더 수행하게 하는 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

13. 제12항에 따른 장치에 있어서,

제1 조건은,

비-제로 잔차를 가지지 않는 화상과 연관된 타깃 코딩 유닛; 또는

비-제로 크로마 잔차를 가지지 않는 화상과 연관된 타깃 변환 유닛을 포함하는, 비디오 콘텐츠 처리 장치.

14. 제13항에 따른 장치에 있어서,

비-제로 잔차를 가지지 않는 타깃 코딩 유닛은 타깃 코딩 유닛의 제1 코딩된 블록 플래그의 값에 기초하여 결정되고, 및

비-제로 크로마 잔차를 가지지 않는 타깃 변환 유닛은 제1 크로마 성분에 대한 제2 코딩된 블록 플래그 및 타깃 변환 유닛의 제2 크로마 성분에 대한 제3 코딩된 블록 플래그의 값에 기초하여 결정되는, 비디오 콘텐츠 처리

장치.

15. 제14항에 따른 장치에 있어서,

제1 코딩된 블록 플래그의 값은 0이고,

제2 코딩된 블록 플래그 및 제3 코딩된 블록 플래그의 값은 0인, 비디오 콘텐츠 처리 장치.

16. 제9항 내지 제15항 중 어느 한 항에 따른 장치에 있어서,

크로마 스케일링 인자를 이용하여 크로마 블록을 처리함에 있어서, 프로세서는 장치로 하여금,

크로마 스케일링 인자를 이용하여 크로마 블록의 잔차를 처리하는 것을 더 수행하게 하는 명령어의 세트를 실행하도록 구성되는, 비디오 콘텐츠 처리 장치.

17. 명령어의 세트를 저장하는 비-일시적 컴퓨터-판독 가능 저장 매체로서, 명령어의 세트는 디바이스가 비디오 콘텐츠를 처리하기 위한 방법을 수행하도록 디바이스의 하나 이상의 프로세서에 의해 실행될 수 있으며, 방법은,

화상과 연관된 크로마 블록과 루마 블록을 수신하는 단계;

루마 블록과 연관된 루마 스케일링 정보를 결정하는 단계;

루마 스케일링 정보에 기초하여 크로마 스케일링 인자를 결정하는 단계; 및

크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계를 포함하는, 비-일시적 컴퓨터-판독 가능 저장 매체.

18. 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현된 방법으로서,

화상과 연관된 크로마 블록과 루마 블록을 수신하는 단계 - 루마 블록은 $N \times M$ 루마 샘플을 포함함 - ;

N 과 M 중 적어도 하나가 임계값보다 큰 것에 응답하여, $N \times M$ 루마 샘플의 서브세트를 선택하는 단계;

$N \times M$ 루마 샘플의 서브세트의 평균 값을 결정하는 단계;

평균 값에 기초하여 크로마 스케일링 인자를 결정하는 단계; 및

크로마 스케일링 인자를 이용하여 크로마 블록을 처리하는 단계를 포함하는, 컴퓨터-구현된 방법.

19. 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현된 방법으로서,

수신되는 비디오 데이터에 크로마 스케일링 인덱스가 존재하는지 여부를 결정하는 단계;

수신되는 비디오 데이터에 크로마 스케일링 인덱스가 존재하지 않는다는 결정에 응답하여, 수신되는 비디오 데이터에 크로마 스케일링이 적용되지 않음을 결정하는 단계; 및

수신되는 비디오 데이터에 크로마 스케일링이 존재한다는 결정에 응답하여, 크로마 스케일링 인덱스에 기초하여 크로마 스케일링 인자를 결정하는 단계를 포함하는, 컴퓨터-구현된 방법.

20. 비디오 콘텐츠를 처리하기 위한 컴퓨터-구현된 방법으로서,

입력 비디오 신호의 동적 범위에 대해 사용되는 복수의 고유 부호어(codeword)를 수신하는 단계;

인덱스를 수신하는 단계;

인덱스에 기초하여, 복수의 고유 부호어 중 적어도 하나를 선택하는 단계; 및

선택된 적어도 하나의 부호어에 기초하여 크로마 스케일링 인자를 결정하는 단계를 포함하는, 컴퓨터-구현된 방법.

컴퓨터-판독 가능한 프로그램 코드를 이용함으로써 상기 방법을 구현하는 것 이외에도, 상기 방법은 논리 게이트, 스위치, ASIC, 프로그램 가능한 논리 제어기, 및 내장형 마이크로 제어기의 형태로 구현될 수도 있다. 따라서, 이러한 제어기는 하드웨어 컴포넌트로서 고려될 수 있고, 제어기 내에 포함되어 다양한 기능을 구현하도록 구성되는 장치는 하드웨어 컴포넌트 내측 구조로서 고려될 수도 있다. 또는, 다양한 기능을 구현하도록 구성되는 장치는 방법을 구현하도록 구성되는 소프트웨어 모듈과 하드웨어 컴포넌트 내측 구조 모두로서 고려될 수도

있다.

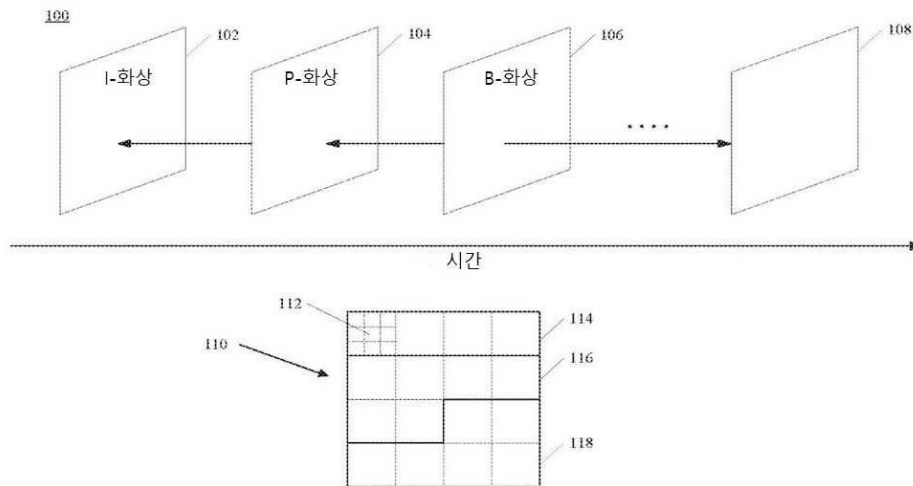
[0303] 본 개시는 컴퓨터, 예를 들어 프로그램 모듈에 의해 실행되는 컴퓨터-실행 가능한 명령어의 일반적인 맥락에서 설명될 수 있다. 일반적으로, 프로그램 모듈은 루틴, 프로그램, 객체(object), 어셈블리, 데이터 구조, 클래스, 또는 특정 작업을 수행하거나 또는 특정한 추상 데이터 유형을 구현하기 위해 사용되는 이와 유사한 것을 포함할 수 있다. 본 개시의 실시예들은 분산된 연산 환경에서 구현될 수도 있다. 분산된 연산 환경에서는 통신 네트워크를 통해 연결되는 원격 처리 디바이스들을 이용함으로써 작업이 실행된다. 분산된 연산 환경에서, 프로그램 모듈은 저장 디바이스를 포함하는 로컬 및 원격 컴퓨터 저장 매체에 위치될 수 있다.

[0304] 본 명세서에서, “제1” 및 “제2”와 같은 상대적인 용어는 엔티티(entity) 또는 동작을 다른 엔티티 또는 동작으로부터 구별하기 위해서만 사용되는 것으로, 이들 엔티티들 또는 동작들 간의 어떤 실제적인 관계 또는 시퀀스를 요구 또는 암시하는 것은 아니다. 또한, “포함하다(comprising, containing, including)”, “구비하다(having)”, 및 다른 유사한 형태의 단어들은 의미적으로 동등한 것으로 의도되며, 이러한 단어들 중 임의의 것 앞에 오는 항목(들)이 이러한 항목(들)의 완전한 목록을 의미하는 것이 아니고, 나열된 항목(들)에 국한되는 것은 아님을 의미한다는 점에서, 개방형(open-ended)인 것으로 의도된다.

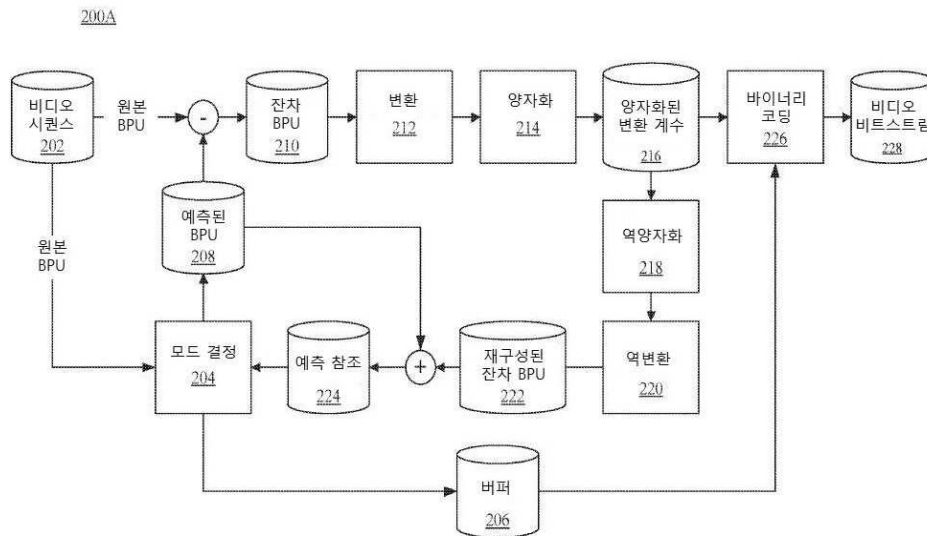
[0305] 전술한 설명에서, 구현에 별로 변할 수 있는 다양한 특정 디테일을 참조하여 실시예들이 설명되었다. 설명된 실시예들에 대한 특정한 적응 및 수정이 이루어질 수 있다. 다른 실시예들은 본 명세서에 개시되는 상세한 설명 및 본 개시의 실시를 고려함으로써, 통상의 기술자에게 명백해질 수 있다. 상세한 설명과 예시들은 이어지는 청구항에 의해 나타나는 본 개시의 진정한 범위 및 사상을 갖는 예시로서만 고려되도록 의도된다. 도면에 도시된 단계들의 시퀀스는 예시적인 목적만을 위한 것으로, 단계들의 임의의 특정 시퀀스로 제한되도록 의도한 것은 아니다. 이와 같이, 통상의 기술자는 이러한 단계들이 동일한 방법을 구현하면서도 상이한 순서로 수행될 수 있음을 이해할 것이다.

도면

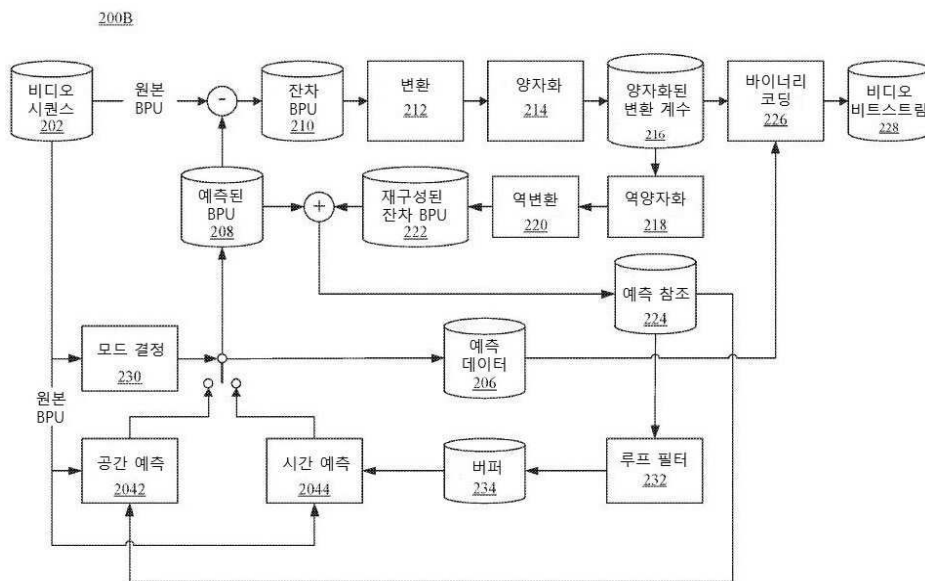
도면1



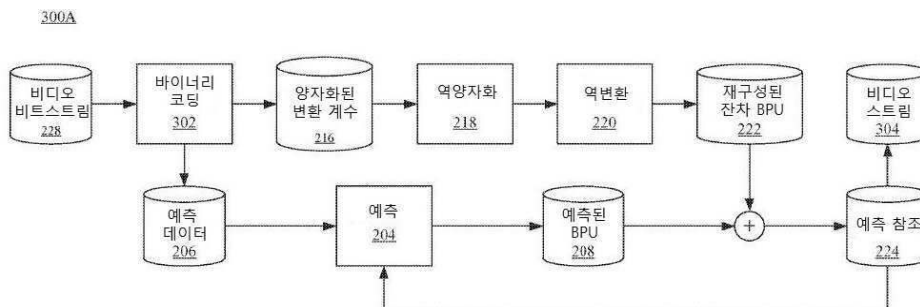
도면2a



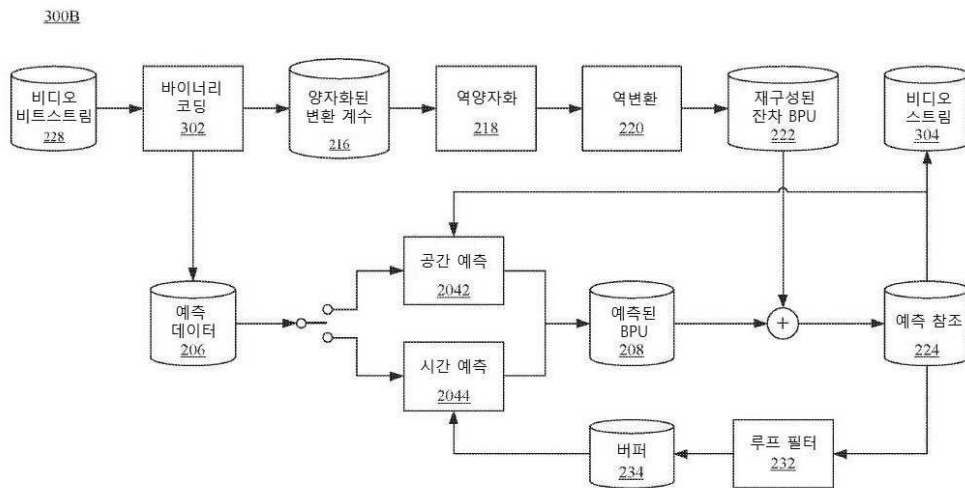
도면2b



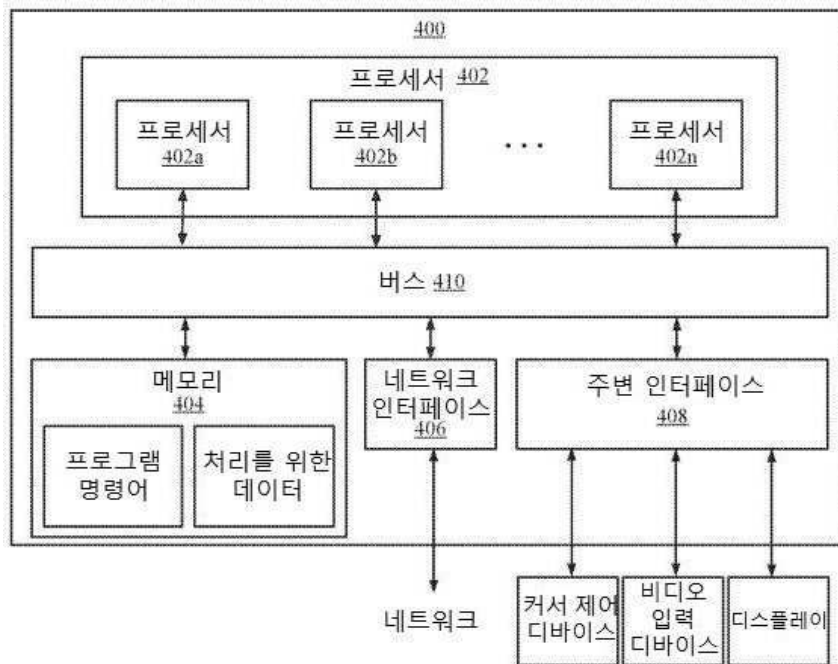
도면3a



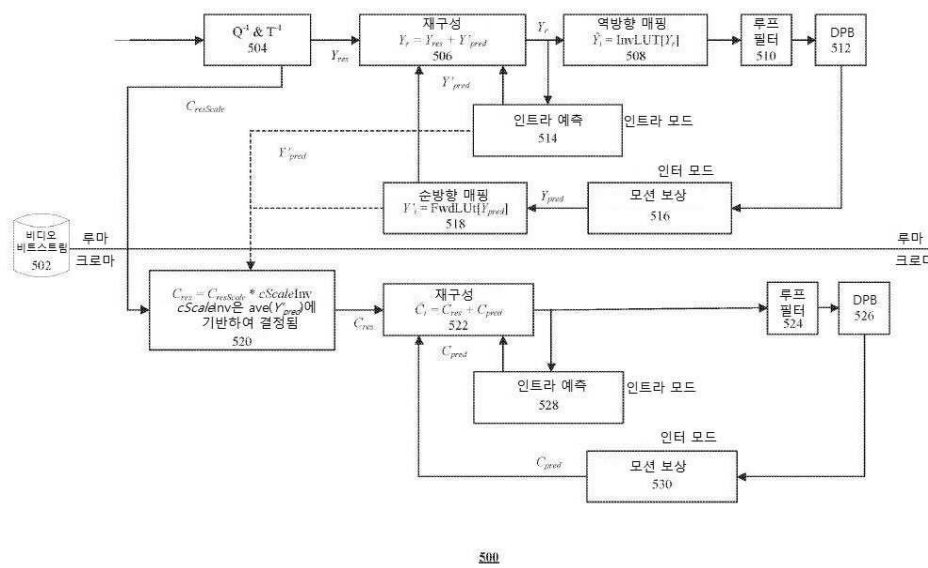
도면3b



도면4



도면5



도면6

표 1

tile_group_header() {	Descriptor
...	
if (sps_resaper_enabled_flag) {	
tile_group_resaper_model_present_flag	u(1)
if (tile_group_resaper_model_present_flag)	
tile_group_resaper_model ()	
tile_group_resaper_enable_flag	u(1)
if (tile_group_resaper_enable_flag && !(qtbtt_dual_tree_intra_flag && tile_group_type == 1))	
tile_group_resaper_chroma_residual_scale_flag	u(1)
}	
byte_alignment()	
}	

도 6: LMCS 구분적 선형 모델에 대한 타일 그룹 레벨 신텍스 표

도면7

표 2

tile_group_resaper_model() {	Descriptor
resaper_model_min_bin_idx	ue(v)
resaper_model_delta_max_bin_idx	ue(v)
resaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
for (i = resaper_model_min_bin_idx; i <= resaper_model_max_bin_idx; i++) {	
reshape_model_bin_delta_abs_CW[i]	u(v)
if (reshape_model_bin_delta_abs_CW[i] > 0)	
resaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
}	

도 7: LMCS 구분적 선형 모델에 대한 타일 그룹 레벨 신텍스 표

도면8

표 3

coding_tree_unit() {	Descriptor
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
if(tile_group_sao_luma_flag tile_group_sao_chroma_flag)	
sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)	
if(tile_group_alf_enabled_flag){	
alf_ctb_flag[0 xCtb >> Log2CtbSize yCtb >> Log2CtbSize]	ac(v)
if(alf_chroma_idc == 1 alf_chroma_idc == 3)	
alf_ctb_flag[1 xCtb >> Log2CtbSize yCtb >> Log2CtbSize]	ac(v)
if(alf_chroma_idc == 2 alf_chroma_idc == 3)	
alf_ctb_flag[2 xCtb >> Log2CtbSize yCtb >> Log2CtbSize]	ac(v)
}	
if((tile_group_type == 1 CurrPicIsOnlyRef) && qtbt_dual_tree_intra_flag)	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbLog2SizeY, 0)	
else	
coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0, SINGLE_TREE)	
}	

도 8: 코딩 트리 유닛 신텍스 구조

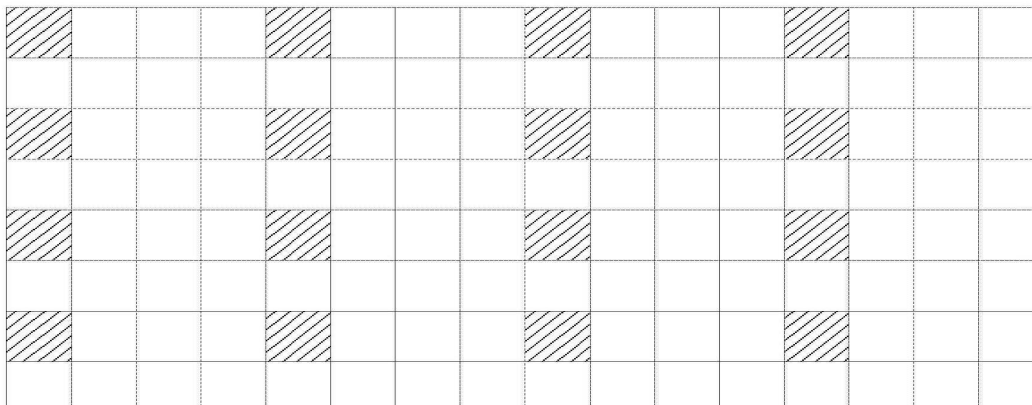
도면9

표 4

dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth) {	Descriptor
if(log2CbSize > 6) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
dual_tree_implicit_qt_split(x0, y0, log2CbSize - 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1, cqtDepth + 1)	
if(y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1)	
} else {	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA)	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA)	
}	
}	

도 9: 듀얼 트리 파티션 신텍스 구조

도면10



도면11

표 5

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
....	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
{	
intra_chroma_pred_mode[x0][y0]	ae(v)
if (tile_group_resaper_chroma_residual_scale_flag)	
lmcs_chroma_scaling_idx[x0][y0]	ae(v)
}	
}	
....	

도 11: 코딩 트리 유닛 선택스 구조

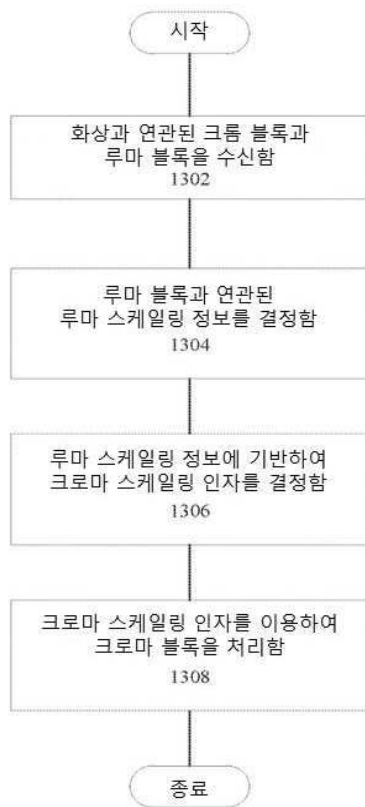
도면12

표 6

tile_group_resaper_model () {	Descriptor
resaper_model_min_bin_idx	ue(v)
resaper_model_delta_max_bin_idx	ue(v)
resaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
resaper_model_bin_num_unique_cw_minus1	ue(v)
for (i = 0; i <= resaper_model_bin_num_unique_cw_minus1; i++) {	
reshape_model_bin_delta_abs_CW [i]	u(v)
if (resaper_model_bin_delta_abs_CW [i] > 0)	
resaper_model_bin_delta_sign_CW_flag [i]	u(1)
}	
for (i = resaper_model_min_bin_idx; i <= resaper_model_max_bin_idx; i++)	
resaper_model_bin_cw_idx [i]	u(v)
}	

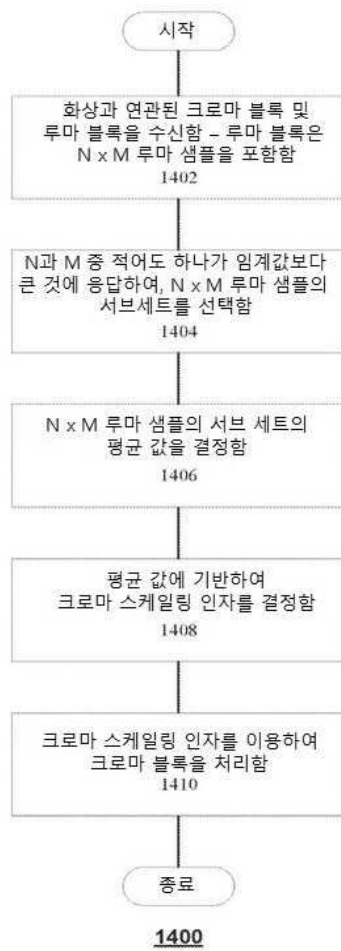
도 12: 타일 그룹 레벨에서의 LMCS 구분적 선형 모델의 수정된 시그널링

도면13

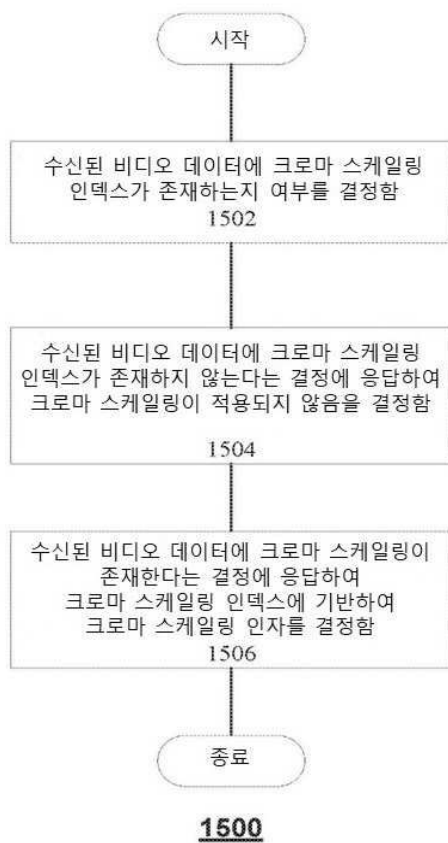


1300

도면14



도면15



도면16

