



등록특허 10-2706586



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년09월12일
(11) 등록번호 10-2706586
(24) 등록일자 2024년09월10일

- (51) 국제특허분류(Int. Cl.)
G06T 9/40 (2006.01) *H04N 19/119* (2014.01)
H04N 19/13 (2014.01) *H04N 19/159* (2014.01)
H04N 19/176 (2014.01) *H04N 19/186* (2014.01)
H04N 19/593 (2014.01) *H04N 19/61* (2014.01)
H04N 19/96 (2014.01)
- (52) CPC특허분류
G06T 9/40 (2019.05)
H04N 19/119 (2015.01)
- (21) 출원번호 10-2022-7037833
- (22) 출원일자(국제) 2022년01월18일
심사청구일자 2022년10월28일
- (85) 번역문제출일자 2022년10월28일
- (65) 공개번호 10-2022-0160667
- (43) 공개일자 2022년12월06일
- (86) 국제출원번호 PCT/US2022/012738
- (87) 국제공개번호 WO 2022/186907
국제공개일자 2022년09월09일
- (30) 우선권주장
63/157,516 2021년03월05일 미국(US)
17/564,566 2021년12월29일 미국(US)
- (56) 선행기술조사문헌
KR1020170077203 A*
US20180288446 A1*
WO2019210857 A1*

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 9 항

심사관 : 권영학

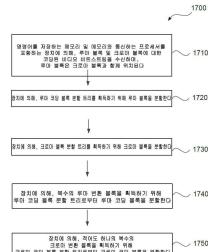
(54) 발명의 명칭 분리된 변환 분할

(57) 요약

비디오 디코딩에서의 변환 블록 분할을 위한 방법, 장치 및 컴퓨터 판독 가능 저장 매체. 이 방법은 장치에 의해 루마 블록 및 크로마 블록에 대한 코딩된 비디오 비트스트림을 수신하는 단계를 포함하고, 루마 블록은 크로마 블록과 함께 위치된다. 이 장치는 명령어를 저장하는 메모리 및 메모리와 통신하는 프로세서를 포함한다.

(뒷면에 계속)

대 표 도 - 도17



이 방법은 또한, 장치에 의해, 루마 코딩 블록 분할 트리를 획득하기 위해 루마 블록을 분할하는 단계; 장치에 의해, 크로마 코딩 블록 분할 트리를 획득하기 위해 크로마 블록을 분할하는 단계; 장치에 의해, 복수의 루마 변환 블록을 획득하기 위해 루마 코딩 블록 분할 트리로부터 루마 코딩 블록을 분할하는 단계; 및 장치에 의해, 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 크로마 코딩 블록 분할 트리로부터 크로마 코딩 블록을 분할하는 단계를 포함한다.

(52) CPC특허분류

HO4N 19/13 (2015.01)

HO4N 19/159 (2015.01)

HO4N 19/176 (2015.01)

HO4N 19/186 (2015.01)

HO4N 19/593 (2015.01)

HO4N 19/61 (2015.01)

HO4N 19/96 (2015.01)

(72) 발명자

페링가세리 크리슈난 마두

미국 94043 캘리포니아주 마운틴 뷰 노스 위스먼
로드 100

리우 산

미국 95129 캘리포니아주 새너제이 네바다 애비뉴
1155

명세서

청구범위

청구항 1

비디오 디코딩에서의 변환 블록 분할을 위한 방법으로서,

명령어를 저장하는 메모리 및 상기 메모리와 통신하는 프로세서를 포함하는 장치에 의해, 루마(luma) 블록 및 크로마(chroma) 블록에 대한 코딩된 비디오 비트스트림을 수신하는 단계(1710) – 상기 루마 블록은 상기 크로마 블록과 함께 위치되고, 상기 코딩된 비디오 비트스트림은 루마 파라미터 및 크로마 파라미터를 포함함 –;

상기 장치에 의해, 루마 코딩 블록 분할 트리를 획득하기 위해 상기 루마 블록을 분할하는 단계(1720);

상기 장치에 의해, 크로마 코딩 블록 분할 트리를 획득하기 위해 상기 크로마 블록을 분할하는 단계(1730);

상기 장치에 의해, 복수의 루마 변환 블록을 획득하기 위해 상기 루마 코딩 블록 분할 트리로부터 루마 코딩 블록을 분할하는 단계(1740); 및

상기 장치에 의해, 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 상기 크로마 코딩 블록 분할 트리로부터 크로마 코딩 블록을 분할하는 단계(1750)

를 포함하고,

상기 복수의 루마 변환 블록을 획득하기 위해 상기 루마 코딩 블록을 분할하는 단계는,

상기 코딩된 비디오 비트스트림으로부터 상기 루마 파라미터를 추출하는 단계 – 상기 루마 파라미터는, 상기 루마 코딩 블록 및 복수의 루마 변환 블록으로부터의 변환 분할 구조를 지시함 – ; 및

복수의 루마 변환 블록을 획득하기 위해 상기 루마 파라미터에 따라 상기 루마 코딩 블록을 분할하는 단계

를 포함하고,

상기 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 상기 크로마 코딩 블록을 분할하는 단계는,

상기 코딩된 비디오 비트스트림으로부터 상기 크로마 파라미터를 추출하는 단계 – 상기 크로마 파라미터는, 상기 크로마 코딩 블록 및 복수의 크로마 변환 블록으로부터의 변환 분할 구조를 지시함 – ; 및

복수의 크로마 변환 블록을 획득하기 위해 상기 크로마 파라미터에 따라 상기 크로마 코딩 블록을 분할하는 단계

를 포함하는,

비디오 디코딩에서의 변환 블록 분할을 위한 방법.

청구항 2

제1항에 있어서,

상기 루마 코딩 블록 분할 트리는 상기 크로마 코딩 블록 분할 트리와 상이한, 비디오 디코딩에서의 변환 블록 분할을 위한 방법.

청구항 3

제1항에 있어서,

상기 루마 코딩 블록 분할 트리는 상기 크로마 코딩 블록 분할 트리와 상이하고,

상기 코딩된 비디오 비트스트림은 루마 파라미터, 제1 크로마 파라미터, 및 제2 크로마 파라미터를 포함하며,

상기 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 상기 크로마 코딩 블록을 분할하는 단계는,

제1 복수의 크로마 변환 블록을 획득하기 위해 상기 제1 크로마 파라미터에 따라 상기 크로마 코딩 블

록의 제1 성분을 분할하는 단계; 및

제2 복수의 크로마 변환 블록을 획득하기 위해 상기 제2 크로마 파라미터에 따라 상기 크로마 코딩 블록의 제2 성분을 분할하는 단계

를 포함하는, 비디오 디코딩에서의 변환 블록 분할을 위한 방법.

청구항 4

제1항에 있어서,

상기 크로마 코딩 블록의 변환 깊이는 0 및 1 중 하나를 포함하거나, 또는

상기 크로마 코딩 블록은 쿼드트리 변환 분할을 포함하거나, 또는 정사각형이 아닌 직사각형 블록인 크로마 코딩 블록은 이진 변환 분할을 지시하는,

비디오 디코딩에서의 변환 블록 분할을 위한 방법.

청구항 5

제1항에 있어서,

상기 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 상기 크로마 코딩 블록을 분할하는 단계는,

상기 크로마 코딩 블록이 최소 코딩 블록 크기보다 큰 조건, 또는

상기 크로마 코딩 블록이 최대 코딩 블록 크기보다 작은 조건, 또는

적어도 하나의 인트라 예측 모드를 사용하는 크로마 코딩 블록 조건, 또는

적어도 하나의 변환 유형을 사용하는 크로마 코딩 블록 조건

중 적어도 하나를 만족시키는 것에 응답하여 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 상기 크로마 코딩 블록을 분할하는 단계

를 포함하는, 비디오 디코딩에서의 변환 블록 분할을 위한 방법.

청구항 6

제5항에 있어서,

상기 최소 코딩 블록 크기는 4 또는 8 또는 16 또는 32 또는 64 중 적어도 하나를 포함하는 최소 코딩 폭에 대응하거나, 또는

상기 최대 코딩 블록 크기는 32 또는 64 또는 128 또는 256 중 적어도 하나를 포함하는 최대 코딩 길이에 대응하거나, 또는

상기 적어도 하나의 인트라 예측 모드는 적어도 하나의 방향성 인트라 예측 모드를 포함하는,

비디오 디코딩에서의 변환 블록 분할을 위한 방법.

청구항 7

제1항에 있어서,

상기 코딩된 비디오 비트스트림은 루마 코딩 파라미터, 루마 변환 파라미터, 크로마 코딩 파라미터, 및 크로마 변환 파라미터를 포함하고,

상기 코딩된 비디오 비트스트림의 엔트로피 디코딩 동안, 상기 크로마 변환 파라미터는 상기 루마 코딩 파라미터, 또는 상기 루마 변환 파라미터, 또는 상기 크로마 코딩 파라미터 중 적어도 하나를 컨텍스트(context)로서 사용하는,

비디오 디코딩에서의 변환 블록 분할을 위한 방법.

청구항 8

비디오 디코딩에서의 변환 블록 분할을 위한 장치로서,

명령어를 저장하는 메모리; 및

상기 메모리와 통신하는 프로세서를

포함하며,

상기 프로세서가 상기 명령어를 실행할 때, 상기 프로세서는 상기 비디오 디코딩에서의 변환 블록 분할을 위한 장치로 하여금 제1항 내지 제7항 중 어느 하나의 항에 따른 방법을 수행하게 하도록 구성되는,

비디오 디코딩에서의 변환 블록 분할을 위한 장치.

청구항 9

명령어를 저장하는 비일시적 컴퓨터 판독 가능 저장 매체로서,

상기 명령어가 프로세서에 의해 실행할 때, 상기 명령어는 상기 프로세서로 하여금 제1항 내지 제7항 중 어느 하나의 항에 따른 방법을 수행하게 하도록 구성되는, 비일시적 컴퓨터 판독 가능 저장 매체.

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2021년 3월 5일에 출원된 미국 가출원 제63/157,516호 및 2021년 12월 29일에 출원된 미국 가출원 제17/564,566호에 기초하여 우선권의 이익을 주장하며, 이들은 모두 그 전체가 참조로서 본 명세서에 포함된다.

[0002] 본 개시는 비디오 코딩 및/또는 디코딩 기술에 관한 것으로, 구체적으로는, 분리된(decoupled) 변환 분할(partitioning)의 개선된 설계 및 시그널링에 관한 것이다.

배경 기술

[0003] 본 명세서에서 제공된 이러한 배경 설명은 본 개시의 맥락을 일반적으로 제시하기 위한 것이다. 현재 명명된 발명자의 작업은 본 배경 섹션에서 설명된 한도 내에서 본 출원을 제출할 당시 선행 기술로 인정되지 않을 수 있는 설명의 측면은 물론 명시적으로나 암시적으로 본 개시에 대한 선행 기술로 인정되지 않는다.

[0004] 비디오 코딩 및 디코딩은 모션 보상과 함께 픽처 간 예측을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처를 포함할 수 있으며, 각각의 픽처는 예를 들어 1920 x 1080 휘도 샘플 및 연관된 전체 또는 부분 샘플링된 색차 샘플의 공간 차원을 갖는다. 일련의 픽처는 예를 들어 초당 60개의 픽처 또는 초당 60개의 프레임의 고정 또는 가변 픽처 속도(다르게는 프레임 속도로서 지칭됨)를 가질 수 있다. 압축되지 않은 비디오는 특정 비트레이트 요구사항을 갖는다. 예를 들어, 1920 x 1080의 픽셀 해상도, 60 프레임/초의 프레임 속도, 색상 채널당 픽셀당 8 비트에서 4:2:0의 크로마 색브샘플링을 사용하는 비디오에는 1.5 Gbit/s에 가까운 대역폭이 필요하다. 이러한 1시간의 비디오는 600GB 이상의 저장 공간을 필요로 한다.

[0005] 비디오 코딩 및 디코딩의 한 가지 목적은 압축을 통해 압축되지 않은 입력 비디오 신호의 중복성을 감소시키는 것일 수 있다. 압축은 앞서 언급된 대역폭 및/또는 저장 공간 요구사항을 경우에 따라 2배 이상으로 감소시키는 데 도움이 될 수 있다. 무손실 압축 및 손실 압축, 및 이들의 조합이 모두 사용될 수 있다. 무손실 압축은 디코딩 프로세스를 통해 압축된 원래 신호에서 원래 신호의 정확한 복사본이 재구성될 수 있는 기술을 지칭한다. 손실 압축은 원래 비디오 정보가 코딩 중에 완전히 유지되지 않고 디코딩 중에 완전히 복원될 수 없는 코딩/디코딩 프로세스를 지칭한다. 손실 압축을 사용하는 경우, 재구성된 신호는 원래 신호와 동일하지 않을 수 있지만, 원래 신호와 재구성된 신호 사이의 왜곡은 약간의 정보 손실에도 불구하고 의도된 애플리케이션에 유용하도록 재구성된 신호를 렌더링할 수 있을 만큼 충분히 작게 만들어진다. 비디오의 경우, 손실 압축은 많은 애플리케이션에서 널리 사용된다. 허용되는 왜곡의 수량은 애플리케이션에 따라 다르다. 예를 들어, 특정 소비자 비디오 스트리밍 애플리케이션의 사용자는 영화 또는 텔레비전 방송 애플리케이션의 사용자보다 더 높은 왜곡을 허용할 수 있다. 특정 코딩 알고리즘에 의해 달성 가능한 압축 비율은 다양한 왜곡 허용 오차를 반영하도록 선택되거나 조정될 수 있다: 더 높은 허용 왜곡은 일반적으로 더 높은 손실과 더 높은 압축 비율을 생성하는 코딩 알고리즘을 허용한다.

[0006] 비디오 인코더 및 디코더는 예를 들어 모션 보상, 푸리에 변환, 양자화, 및 엔트로피 코딩을 포함하는 여러 광범위한 카테고리 및 단계로부터의 기술을 이용할 수 있다.

[0007] 비디오 코덱 기술은 인트라 코딩으로 알려진 기술을 포함할 수 있다. 인트라 코딩에서, 샘플 값은 이전에 재구성된 참조 픽처로부터의 샘플 또는 다른 데이터에 대한 참조 없이 표현된다. 일부 비디오 코덱에서, 픽처는 공간적으로 샘플 블록으로 세분화된다. 샘플의 모든 블록이 인트라 모드에서 코딩되는 경우, 그 픽처는 인트라 픽처로서 지칭될 수 있다. 인트라 픽처 및 독립 디코더 리프레시 픽처와 같은 파생물은 디코더 상태를 재설정하는 데 사용될 수 있으므로, 코딩된 비디오 비트스트림 및 비디오 세션의 제1 픽처 또는 정지 이미지로 사용될 수 있다. 인트라 예측 후의 블록의 샘플은 주파수 도메인으로 변환될 수 있고, 그렇게 생성된 변환 계수는 엔트로피 코딩 전에 양자화될 수 있다. 인트라 예측은 사전 변환 도메인에서 샘플 값을 최소화하는 기술을 나타낸다. 일부 경우에, 변환 후의 DC 값이 더 작고, AC 계수가 더 작을수록 엔트로피 코딩 후 블록을 나타내기 위해 주어진 양자화 단계 크기에서 필요한 비트가 더 적다.

[0008] 예를 들어 MPEG-2 생성 코딩 기술로부터 알려진 것과 같은 전통적인 인트라 코딩은 인트라 예측을 사용하지 않는다. 그러나, 일부 새로운 비디오 압축 기술은, 예를 들어, 주변의 샘플 데이터 및/또는 공간적으로 이웃하는 인코딩 및/또는 디코딩 중에 획득되고 인트라 코딩되거나 또는 디코딩되는 데이터 블록을 디코딩 순서에서 선행하는 메타데이터에 기초하여 블록의 코딩/디코딩을 시도하는 기술을 포함한다. 이러한 기술은 이후 "인트라 예측" 기술이라고 한다. 적어도 일부 경우에, 인트라 예측은 다른 참조 픽처가 아닌 재구성 중인 현재 픽처의 참조 데이터만을 사용한다는 점에 유의한다.

- [0009] 인트라 예측에는 많은 상이한 형태가 있을 수 있다. 주어진 비디오 코딩 기술에서 이러한 기술 중 하나 이상이 사용될 수 있는 경우, 사용 중인 기술은 인트라 예측 모드로서 지칭될 수 있다. 하나 이상의 인트라 예측 모드는 특정 코덱으로 제공될 수 있다. 특정 경우에, 모드는 서브 모드를 가질 수 있고, 및/또는 다양한 파라미터와 연관될 수 있으며, 모드/서브모드 정보 및 비디오 블록에 대한 인트라 코딩 파라미터는 모드 코드워드에 개별적으로 또는 침합적으로 포함되어 코딩될 수 있다. 주어진 모드, 서브모드 및/또는 파라미터 조합에 사용할 코드워드는 인트라 예측을 통해 코딩 효율 이득에 영향을 미칠 수 있으며, 코드워드를 비트스트림으로 변환하는데 사용되는 엔트로피 코딩 기술도 마찬가지이다.
- [0010] 인트라 예측의 특정 모드는 H.264에 도입되었고, H.265에서 개선되었으며, 공동 탐사 모델(Joint Exploration Model, JEM), 다목적 비디오 코딩(versatile video coding, VVC) 및 벤치마크 세트(benchmark set, BMS)와 같은 새로운 코딩 기술에서 더욱 개선되었다. 일반적으로, 인트라 예측의 경우, 예측기 블록은 사용 가능하게 된 이웃 샘플 값을 사용하여 형성될 수 있다. 예를 들어, 특정 방향 및/또는 라인을 따라 이웃 샘플의 특정 세트의 이용 가능한 값이 예측기 블록으로 복사될 수 있다. 사용 방향에 대한 참조는 비트스트림에서 코딩되거나 자체적으로 예측될 수 있다.
- [0011] 도 1a를 참조하면, 우측 하부에 묘사된 것은 H.265의 33개의 가능한 인트라 예측기 방향(H.265에서 규정된 35개 인트라 모드의 알려진 33개 각도 모드에 대응함)에서 규정된 9개의 예측기 방향의 서브 세트이다. 화살표가 수렴하는 지점(101)은 예측 중인 샘플을 나타낸다. 화살표는 101에서 샘플을 예측하는 데 사용되는 이웃 샘플로부터의 방향을 나타낸다. 예를 들어, 화살표(102)는 샘플(101)이 이웃 샘플 또는 수평 방향에서 45도 각도로 우측 상부에 있는 샘플들로부터 예측됨을 지시한다. 마찬가지로, 화살표(103)는 샘플(101)이 이웃 샘플 또는 수평 방향으로부터 22.5도 각도로 샘플(101)의 좌측 하부에 있는 샘플들로부터 예측됨을 지시한다.
- [0012] 여전히 도 1a를 참조하면, 좌측 상단에는 4×4 샘플의 정사각형 블록(104)이 도시되어 있다(대시 형태의 볼드 체 선으로 지시됨). 정사각형 블록(104)은 각각 "S"로 라벨링되며, Y 차원에서의 위치(예를 들어, 행 인덱스) 및 X 차원에서의 위치(예를 들어, 열 인덱스)의 16개의 샘플을 포함한다. 예를 들어, 샘플 S21은 Y 차원에서 두 번째 샘플(상단으로부터)이고 X 차원에서 첫 번째(좌측으로부터) 샘플이다. 유사하게, 샘플 S44는 Y 및 X 차원 모두에서 블록(104)의 네 번째 샘플이다. 블록 크기가 4×4 샘플이므로, S44는 우측 하단에 있다. 유사한 번호 매기기 방식을 따르는 예시적인 참조 샘플이 추가로 표시된다. 참조 샘플은 블록(104)에 대해 R로 라벨링로 되며, Y 위치(예를 들어, 행 인덱스) 및 X 위치(열 인덱스)이다. H.264 및 H.265 모두에서, 재구성 중인 블록에 인접하게 이웃하는 예측 샘플이 사용된다.
- [0013] 블록(104)의 인트라 꾹처 예측은 시그널링된 예측 방향에 따라 이웃 샘플로부터 참조 샘플 값을 복사함으로써 시작될 수 있다. 예를 들어, 코딩된 비디오 비트스트림이 이러한 블록(104)에 대해 화살표(102)의 예측 방향을 지시하는 시그널링을 포함하는 것으로 가정한다 – 즉, 샘플은 예측 샘플 또는 수평 방향으로부터 45도 각도로 우측 상부에 있는 샘플들로부터 예측된다. 이러한 경우에, 샘플 S41, S32, S23, S14는 동일한 참조 샘플 R05로부터 예측된다. 그런 다음, 샘플 S44는 참조 샘플 R08로부터 예측된다.
- [0014] 특정 경우에, 다수의 참조 샘플의 값은 참조 샘플을 계산하기 위해, 특히 방향이 45도로 균등하게 나누어지지 않는 경우에, 예를 들어 보간을 통해 결합될 수 있다.
- [0015] 비디오 코딩 기술이 계속 발전함에 따라 가능한 방향의 개수가 증가하였다. H.264(2003년)에서, 예를 들어, 9개의 상이한 방향이 인트라 예측을 위해 이용 가능하다. H.265(2013년)에서는 33개로 증가했고, 본 개시의 당시, JEM/VVC/BMS는 최대 65개 방향을 지원할 수 있다. 가장 적절한 인트라 예측 방향을 식별하는 것을 돋기 위한 실험 연구가 수행되었으며, 엔트로피 코딩의 특정 기술은 방향에 대한 특정 비트 페널티를 허용하는 소수의 비트에서 이러한 가장 적절한 방향을 인코딩하는 데 사용될 수 있다. 또한, 방향 자체는 때때로 디코딩된 이웃 블록의 인트라 예측에서 사용된 이웃 방향으로부터 예측될 수 있다.
- [0016] 도 1b는 시간이 지남에 따라 개발된 다양한 인코딩 기술에서 증가하는 개수의 예측 방향을 도시하기 위해 JEM에 따른 65개의 인트라 예측 방향을 묘사하는 개략도(180)를 도시한다.
- [0017] 코딩된 비디오 비트스트림의 예측 방향에 대한 인트라 예측 방향을 나타내는 비트의 매핑 방식은 비디오 코딩 기술에 따라 상이할 수 있으며, 예를 들어, 예측 방향의 단순한 직접 매핑에서 인트라 예측 모드, 코드워드, 가장 가능성 있는 모드를 포함하는 복잡한 적응 방식 및 유사한 기술에 이르기까지 다양할 수 있다. 그러나, 모든 경우에, 특정 다른 방향보다 비디오 컨텐츠에서 통계적으로 발생할 가능성이 적은 인트로 예측에 대한 특정 방향이 있을 수 있다. 비디오 압축의 목표는 중복성을 줄이는 것이므로, 잘 설계된 비디오 코딩 기술에서 가능

성이 적은 방향은 가능성이 더 높은 방향보다 더 많은 비트 개수로 표현될 수 있다.

[0018] 인터 픽처 예측, 또는 인터 예측은 모션 보상에 기초할 수 있다. 모션 보상에서, 모션 벡터(이후 MV)에 의해 지시된 방향으로 공간적으로 천이된 후에, 이전에 재구성된 픽처 또는 그 일부(참조 픽처)로부터의 샘플 데이터가 새로 재구성된 픽처 또는 픽처 부분(예를 들어, 블록)의 예측에 사용될 수 있다. 일부 경우에, 참조 픽처는 현재 재구성 중인 픽처와 동일할 수 있다. MV는 X와 Y의 2차원 또는 3차원을 가질 수 있으며, 세 번째 차원은 사용 중인 참조 픽처의 지시이다(시간 차원과 유사).

[0019] 일부 비디오 압축 기술에서, 샘플 데이터의 특정 영역에 적용 가능한 현재 MV는 다른 MV로부터, 예를 들어 재구성 중인 영역에 공간적으로 인접하고 디코딩 순서에서 현재 MV에 선행하는 샘플 데이터의 다른 영역과 관련된 다른 MV로부터 예측될 수 있다. 그렇게 하면, 상관된 MV에서의 중복성을 제거함으로써 MV를 코딩하는 데 필요한 데이터의 전체 양을 상당히 줄일 수 있으므로, 압축 효율을 증가시킬 수 있다. 예를 들어, 카메라에서 발생된 입력 비디오 신호(자연 비디오라고 함)를 코딩하는 경우 단일 MV가 적용 가능한 영역보다 큰 영역이 비디오 시퀀스에서 유사한 방향으로 이동하고, 따라서 일부 경우에 이웃 영역의 MV에서 발생된 유사한 모션 벡터를 사용하여 예측할 수 있는 통계적 가능성이 있기 때문에, MV 예측은 효과적으로 작동할 수 있다. 그 결과, 주어진 영역에 대한 실제 MV가 주변 MV로부터 예측된 MV와 유사하거나 동일하다. 그러한 MV는 차례로 엔트로피 코딩 후에 MV가 이웃 MV(들)로부터 예측되기 보다는 직접 코딩되는 경우에 사용되는 것보다 더 적은 개수의 비트로 표현될 수 있다. 일부 경우에, MV 예측은 원래 신호(즉, 샘플 스트림)에서 발생된 신호(즉, MV)의 무손실 압축의 예가 될 수 있다. 다른 경우에, MV 예측 자체는 예를 들어 주변의 여러 MV로부터 예측기를 계산하는 경우 반올림 오류로 인해 손실될 수 있다.

[0020] 다양한 MV 예측 메커니즘이 H.265/HEVC(ITU-T Rec. H.265, "고효율 비디오 코딩", 2016년 12월)에서 설명된다. H.265가 규정하는 많은 MV 예측 메커니즘 중에서, 아래에서 설명되는 기술은 이후 "공간적 병합(spatial merge)"으로 지칭되는 기술이다.

[0021] 구체적으로, 도 2를 참조하면, 현재 블록(201)은 공간적으로 천이된 동일한 크기의 이전 블록으로부터 예측 가능한 것으로 모션 검색 프로세스 동안 인코더에 의해 발견된 샘플을 포함한다. 해당 MV를 직접 코딩하는 대신, MV는 예를 들어, A0, A1, 및 B0, B1, B2(각각 202 내지 206)으로 표시된 5개의 주변 샘플 중 하나와 연관된 MV를 사용하여 가장 최근의(디코딩 순서에서) 참조 픽처로부터 하나 이상의 참조 픽처와 연관된 메타데이터로부터 유도될 수 있다. H.265에서, MV 예측은 이웃 블록이 사용하고 있는 동일한 참조 픽처의 예측기를 사용한다.

발명의 내용

[0022] 본 개시는 비디오 인코딩 및/또는 디코딩을 위한 방법, 장치, 및 컴퓨터 판독 가능 저장 매체의 다양한 실시예를 설명한다.

[0023] 일 측면에 따르면, 본 개시의 실시예는 비디오 디코딩에서의 변환 블록 분할을 위한 방법을 제공한다. 이 방법은, 장치에 의해 루마 블록 및 크로마 블록에 대한 코딩된 비디오 비트스트림을 수신하는 단계를 포함하고, 루마 블록은 크로마 블록과 함께 위치된다. 이 장치는 명령어를 저장하는 메모리 및 메모리와 통신하는 프로세서를 포함한다. 이 방법은 또한, 장치에 의해, 루마 코딩 블록 분할 트리를 획득하기 위해 루마 블록을 분할하는 단계; 장치에 의해, 크로마 코딩 블록 분할 트리를 획득하기 위해 크로마 블록을 분할하는 단계; 장치에 의해, 복수의 루마 변환 블록을 획득하기 위해 루마 코딩 블록 분할 트리로부터 루마 코딩 블록을 분할하는 단계; 및 장치에 의해, 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 크로마 코딩 블록 분할 트리로부터 크로마 코딩 블록을 분할하는 단계를 포함한다.

[0024] 다른 측면에 따르면, 본 개시의 실시예는 비디오 인코딩 및/또는 디코딩을 위한 장치를 제공한다. 이 장치는 명령어를 저장하는 메모리; 및 메모리와 통신하는 프로세서를 포함한다. 프로세서가 명령어를 실행할 때, 프로세서는 장치로 하여금 비디오 디코딩 및/또는 인코딩을 위한 상기 방법을 수행하도록 구성된다.

[0025] 다른 측면에서, 본 개시의 실시예는 비디오 디코딩 및/또는 인코딩을 위해 컴퓨터에 의해 실행될 때 컴퓨터로 하여금 비디오 디코딩 및/또는 인코딩을 위한 상기 방법을 수행하도록 하는 명령어를 저장하는 비일시적 컴퓨터 판독 가능 매체를 제공한다.

[0026] 상기 및 다른 측면 및 이들의 구현은 도면, 설명 및 청구범위에서 더 상세하게 설명된다.

도면의 간단한 설명

[0027] 개시된 주제의 추가 특징, 특성 및 다양한 이점은 다음의 상세한 설명 및 첨부 도면으로부터 더 명백해질 것이다.

도 1a는 인트라 예측 방향 모드의 예시적인 서브세트의 개략도를 도시한다.

도 1b는 예시적인 인트라 예측 방향의 예시를 도시한다.

도 2는 일 예에서 모션 벡터 예측을 위한 현재 블록 및 그 주변 공간 병합 후보의 개략도를 도시한다.

도 3은 예시적인 실시예에 따른 통신 시스템(300)의 간략화된 블록도의 개략도를 도시한다.

도 4는 예시적인 실시예에 따른 통신 시스템(400)의 간략화된 블록도의 개략도를 도시한다.

도 5는 예시적인 실시예에 따른 비디오 디코더의 간략화된 블록도의 개략도를 도시한다.

도 6은 예시적인 실시예에 따른 비디오 인코더의 간략화된 블록도의 개략도를 도시한다.

도 7은 다른 예시적인 실시예에 따른 비디오 인코더의 블록도를 도시한다.

도 8은 다른 예시적인 실시예에 따른 비디오 디코더의 블록도를 도시한다.

도 9는 본 개시의 예시적인 실시예에 따른 코딩 블록 분할 방식을 도시한다.

도 10은 본 개시의 예시적인 실시예에 따른 코딩 블록 분할의 다른 방식을 도시한다.

도 11은 본 개시의 예시적인 실시예에 따른 코딩 블록 분할의 다른 방식을 도시한다.

도 12는 본 개시의 예시적인 실시예에 따른 코딩 블록 분할의 다른 방식을 도시한다.

도 13은 본 개시의 예시적인 실시예에 따라 코딩 블록을 다중 변환 블록으로 분할하는 방식 및 변환 블록의 코딩 순서를 도시한다.

도 14는 본 개시의 예시적인 실시예에 따라 코딩 블록을 다중 변환 블록으로 분할하는 다른 방식 및 변환 블록의 코딩 순서를 도시한다.

도 15는 본 개시의 예시적인 실시예에 따라 코딩 블록을 다중 변환 블록으로 분할하는 다른 방식을 도시한다.

도 16은 본 개시의 예시적인 실시예에 따른 루마 및 크로마 성분에 대한 코딩 트리 구조의 일 예를 도시한다.

도 17은 본 개시의 예시적인 실시예에 따른 방법의 흐름도를 도시한다.

도 18은 본 개시의 예시적인 실시예에 따른 루마 및 크로마 성분에 대한 분리된 변환 분할의 일 예를 도시한다.

도 19는 본 개시의 예시적인 실시예에 따른 컴퓨터 시스템의 개략도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0028] 본 발명은 이제 본 발명의 일부를 형성하고 실시예의 특정 예를 예시로서 도시한 첨부 도면을 참조하여 이하에서 상세하게 설명될 것이다. 그러나, 본 발명은 다양한 상이한 형태로 구현될 수 있고, 따라서 커버되거나 청구된 주제는 아래에 설명되는 임의의 실시예에 제한되지 않는 것으로 해석되도록 의도된다. 또한, 본 발명은 방법, 장치, 컴포넌트 또는 시스템으로 구현될 수 있다. 따라서, 본 발명의 실시예는 예를 들어 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합의 형태를 취할 수 있다.

[0029] 명세서 및 청구범위 전반에 걸쳐, 용어는 명시적으로 언급된 의미를 넘어 문맥에서 제안되거나 암시된 미묘한 의미를 가질 수 있다. 본 명세서에서 사용된 "일 실시예에서" 또는 "일부 실시예에서"라는 문구는 반드시 동일한 실시예를 지칭하는 것은 아니며, 본 명세서에서 사용된 "다른 실시예에서" 또는 "또 다른 실시예에서"라는 문구는 반드시 상이한 실시예를 지칭하는 것은 아니다. 마찬가지로, 본 명세서에서 사용된 "하나의 구현에서" 또는 "일부 구현에서"라는 문구는 반드시 동일한 구현을 지칭하는 것은 아니며, 본 명세서에서 사용되는 "다른 구현에서" 또는 "또 다른 구현에서"라는 문구는 반드시 상이한 구현을 지칭하는 것은 아니다. 예를 들어, 청구된 주제는 예시적인 실시예/구현의 조합을 전체적으로 또는 부분적으로 포함하는 것으로 의도된다.

[0030] 일반적으로, 용어는 문맥에서의 용법으로부터 적어도 부분적으로 이해될 수 있다. 예를 들어, 본 명세서에 사용되는 "및", "또는", 또는 "및/또는"과 같은 용어는 그러한 용어가 사용되는 문맥에 적어도 부분적으로 의존할 수 있는 다양한 의미를 포함할 수 있다. 일반적으로 A, B 또는 C와 같은 리스트를 연관시키는 데 사용되는 경

우 "또는"은 여기에서 포괄적인 의미로 사용되는 A, B 및 C는 물론 여기에서 배타적인 의미로 사용되는 A, B 또는 C를 의미하도록 의도된다. 또한, 본 명세서에 사용되는 용어 "하나 이상" 또는 "적어도 하나"는 문맥에 따라 적어도 부분적으로는 단수 의미로 임의의 특징, 구조 또는 특성을 설명하는 데 사용될 수 있거나 또는 복수 의미로 특징, 구조 또는 특성의 조합을 설명하는 데 사용될 수 있다. 유사하게, "하나(a)", "하나(an)" 또는 "상기(the)"와 같은 용어는 문맥에 따라 적어도 부분적으로는 단수 용법을 전달하거나 또는 복수 용법을 전달하는 것으로 이해될 수 있다. 또한, "~에 기초한" 또는 "~에 의해 결정된"이라는 용어는 배타적인 인자 세트를 전달하는 것으로 반드시 의도되지 않는 것으로 이해될 수 있으며, 대신에 문맥에 따라 적어도 부분적으로 반드시 명시적으로 설명되지 않은 추가 인자의 존재를 허용할 수 있다.

[0031] 도 3은 본 개시의 실시예에 따른 통신 시스템(300)의 간략화된 블록도를 예시한다. 통신 시스템(300)은 예를 들어 네트워크(350)를 통해 서로 통신할 수 있는 복수의 단말 장치를 포함한다. 예를 들어, 통신 시스템(300)은 네트워크(350)를 통해 상호 연결된 제1 쌍의 단말 장치(310, 320)를 포함한다. 도 3의 예에서, 제1 쌍의 단말 장치(310, 320)는 데이터의 단방향 전송을 수행할 수 있다. 예를 들어, 단말 장치(310)는 네트워크(350)를 통해 다른 단말 장치(320)로의 전송을 위해 비디오 데이터(예를 들어, 단말 장치(310)에 의해 캡처된 비디오 픽처의 스트림)를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림의 형태로 전송될 수 있다. 단말 장치(320)는 네트워크(350)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처를 복원하며, 복원된 비디오 데이터에 따라 비디오 픽처를 디스플레이할 수 있다. 단방향 데이터 전송은 매체 서빙 애플리케이션 등에서 구현될 수 있다.

[0032] 다른 예에서, 통신 시스템(300)은 예를 들어 화상 회의 애플리케이션 동안 구현될 수 있는 코딩된 비디오 데이터의 양방향 전송을 수행하는 제2 쌍의 단말 장치(330, 340)를 포함한다. 데이터의 양방향 전송을 위해, 예에서, 단말 장치(330, 340)의 각각의 단말 장치는 네트워크(350)를 통해 단말 장치(330, 340)의 다른 단말 장치로의 전송을 위해 (예를 들어, 단말 장치에 의해 캡처된 비디오 픽처의 스트림의) 비디오 데이터를 코딩할 수 있다. 단말 장치(330, 340)의 각각의 단말 장치는 또한 단말 장치(330, 340)의 다른 단말 장치에 의해 전송된 코딩된 비디오 데이터를 수신할 수 있고, 비디오 픽처를 복원하기 위해 코딩된 비디오 데이터를 디코딩할 수 있으며 복원된 비디오 데이터에 따라 액세스 가능한 디스플레이 장치에서 비디오 픽처를 디스플레이할 수 있다.

[0033] 도 3의 예에서, 단말 장치(310, 320, 330, 340)는 서버, 개인용 컴퓨터 및 스마트 폰으로 구현될 수 있지만, 본 개시의 기본 원칙의 적용 가능성은 이에 제한되지 않을 수 있다. 본 개시의 실시예는 데스크탑 컴퓨터, 랩탑 컴퓨터, 태블릿 컴퓨터, 매체 플레이어, 웨어러블 컴퓨터, 전용 화상 회의 장비 등에서 구현될 수 있다. 네트워크(350)는 예를 들어 유선 및/또는 무선 통신 네트워크를 포함하여 단말 장치(310, 320, 330, 340) 사이에서 코딩된 비디오 데이터를 운반하는 임의의 개수 또는 유형의 네트워크를 나타낸다. 통신 네트워크(350)는 회선 교환, 패킷 교환, 및/또는 다른 채널 유형에서 데이터를 교환할 수 있다. 대표적인 네트워크는 통신 네트워크, 근거리 통신망, 광역 통신망 및/또는 인터넷을 포함한다. 본 논의의 목적을 위해, 네트워크(350)의 아키텍처 및 토플로지는 여기에서 명시적으로 설명되지 않는 한 본 개시의 작동에 중요하지 않을 수 있다.

[0034] 도 4는 개시된 주제에 대한 애플리케이션에 대한 예로서, 비디오 스트리밍 환경에서 비디오 인코더 및 비디오 디코더의 배치를 도시한다. 개시된 주제는 예를 들어 화상 회의, 디지털 TV 방송, 게이밍 가상 현실, CD, DVD, 메모리 스틱 등을 포함하는 디지털 매체 상에 압축된 비디오 저장 등을 포함하는 다른 비디오 애플리케이션에 동일하게 적용될 수 있다.

[0035] 비디오 스트리밍 시스템은 비디오 소스(401), 예를 들어 압축되지 않은 비디오 픽처 또는 이미지(402)의 스트림을 생성하기 위한 디지털 카메라를 포함할 수 있는 비디오 캡처 서브시스템(413)을 포함할 수 있다. 예에서, 비디오 픽처의 스트림(402)은 비디오 소스(401)의 디지털 카메라에 의해 녹화된 샘플을 포함한다. 인코딩된 비디오 데이터(404)(또는 코딩된 비디오 비트스트림)와 비교되는 경우 높은 데이터 볼륨을 강조하기 위해 굵은 선으로 표시된 비디오 픽처(402)의 스트림은 비디오 소스(401)와 결합된 비디오 인코더(403)를 포함하는 전자 장치(420)에 의해 처리될 수 있다. 비디오 인코더(403)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 측면을 가능하게 하거나 구현하기 위해 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 압축되지 않은 비디오 픽처의 스트림(402)과 비교되는 경우 더 낮은 데이터 볼륨을 강조하기 위해 가는 선으로 묘사된 인코딩된 비디오 데이터(404)(또는 인코딩된 비디오 비트스트림(404))는 향후 사용을 위해 스트리밍 서버(405)에 저장될 수 있거나 또는 다운스트림 비디오 장치(도시되지 않음)에 직접 저장될 수 있다. 도 4의 클라이언트 서브시스템(406, 408)과 같은 하나 이상의 스트리밍 클라이언트 서브시스템은 인코딩된 비디오 데이터(404)의 사본(407, 409)을 검색하기 위해 스트리밍 서버(405)에 액세스할 수 있다. 클라이언트 서브시스템(406)은 예를 들어 전자 장치(430)에 비디오 디코더(410)를 포함할 수 있다. 비디오 디코더(410)는 인코딩된 비디오 데이터

의 인입 사본(407)을 디코딩하고 디스플레이(412)(예를 들어, 디스플레이 스크린) 또는 다른 렌더링 장치(도시되지 않음) 상에서 렌더링될 수 있는 압축되지 않은 비디오 픽처(411)의 인출 스트림을 생성한다. 비디오 디코더(410)는 본 개시에서 설명된 다양한 기능들 중 일부 또는 전부를 수행하도록 구성될 수 있다. 일부 스트리밍 시스템에서, 인코딩된 비디오 데이터(404, 407, 409)(예를 들어, 비디오 비트스트림)은 특정 비디오 코딩/압축 표준에 따라 인코딩될 수 있다. 이러한 표준의 예는 ITU-T 권고 H.265를 포함한다. 일 예에서, 개발 중인 비디오 코딩 표준은 비공식적으로 다목적 비디오 코딩(Versatile Video Coding, VVC)으로 알려져 있다. 개시된 주제는 VVC의 맥락 및 다른 비디오 코딩 표준에서 사용될 수 있다.

[0036] 전자 장치(420, 430)는 다른 컴포넌트(도시되지 않음)를 포함할 수 있음에 유의한다. 예를 들어, 전자 장치(420)는 비디오 디코더(미도시)를 포함할 수 있고, 전자 장치(430)는 비디오 인코더(도시되지 않음)도 포함할 수 있다.

[0037] 도 5는 본 개시의 임의의 실시예에 따른 비디오 디코더(510)의 블록도를 도시한다. 비디오 디코더(510)는 전자 장치(530)에 포함될 수 있다. 전자 장치(530)는 수신기(531)(예를 들어, 수신 회로)를 포함할 수 있다. 비디오 디코더(510)는 도 4의 예에서 비디오 디코더(410) 대신에 사용될 수 있다.

[0038] 수신기(531)는 비디오 디코더(510)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스를 수신할 수 있다. 동일하거나 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스가 디코딩될 수 있으며, 여기서 각각의 코딩된 비디오 시퀀스의 디코딩은 다른 코딩된 비디오 시퀀스와 독립적이다. 각각의 비디오 시퀀스는 다른 비디오 프레임 또는 이미지와 연관될 수 있다. 코딩된 비디오 시퀀스는 인코딩된 비디오 데이터 또는 인코딩된 비디오 데이터를 전송하는 스트리밍 소스를 저장하는 저장 장치에 대한 하드웨어/소프트웨어 링크일 수 있는 채널(501)로부터 수신될 수 있다. 수신기(531)는 인코딩된 비디오 데이터를 코딩된 오디오 데이터 및/또는 보조 데이터 스트리밍과 같은 다른 데이터와 함께 수신할 수 있으며, 이는 각각의 처리 회로(도시되지 않음)로 전달될 수 있다. 수신기(531)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분할할 수 있다. 네트워크 지터(jitter)를 방지하기 위해, 버퍼 메모리(515)는 수신기(531)와 엔트로피 디코더/파서(520)(이하 "파서(520)") 사이에 배치될 수 있다. 특정 애플리케이션에서, 버퍼 메모리(515)는 비디오 디코더(510)의 일부로서 구현될 수 있다. 다른 적용에서, 비디오 디코더(510)(도시되지 않음) 외부에 있거나 비디오 디코더(510)와 분할되어 있을 수 있다. 또 다른 적용으로서, 예를 들어 네트워크 지터를 방지하기 위해 비디오 디코더(510) 외부에 버퍼 메모리(도시되지 않음)가 있을 수 있고, 또한 예를 들어 재생 타이밍을 처리하기 위해 비디오 디코더(510) 내부에 또 다른 추가의 버퍼 메모리(515)가 있을 수 있다. 수신기(531)가 충분한 대역폭과 제어 가능성의 저장/전달 장치 또는 동시성 네트워크로부터 데이터를 수신하는 경우, 버퍼 메모리(515)는 필요하지 않거나 작을 수 있다. 인터넷과 같은 최선의 패킷 네트워크 상에서의 사용을 위해, 충분한 크기의 버퍼 메모리(515)가 필요할 수 있고, 그것의 크기는 비교적 클 수 있다. 이러한 버퍼 메모리는 적응형 크기로 구현될 수 있고, 운영 체제 또는 비디오 디코더(510) 외부의 유사한 요소(도시되지 않음)에서 적어도 부분적으로 구현될 수 있다.

[0039] 비디오 디코더(510)는 코딩된 비디오 시퀀스로부터 심볼(521)을 재구성하기 위해 파서(520)를 포함할 수 있다. 이러한 심볼의 카테고리는 비디오 디코더(510)의 작동을 관리하는 데 사용되는 정보와 전자 장치의 필수 부분이 아니지만 도 5에 도시된 바와 같이 전자 장치(530)에 연결될 수 있는 디스플레이(512)와 같은 렌더링 장치(예를 들어, 디스플레이 스크린)와 같은 렌더링 장치를 제어하기 위한 정보를 포함한다. 렌더링 장치(들)에 대한 제어 정보는 보충 항상 정보(Supplemental Enhancement Information, SEI 메시지) 또는 비디오 사용가능성 정보(Video Usability Information, VUI) 파라미터 세트 조각(도시되지 않음)의 형태일 수 있다. 파서(520)는 파서(520)에 의해 수신된 코딩된 비디오 시퀀스를 파싱/엔트로피 디코딩할 수 있다. 코딩된 비디오 시퀀스의 엔트로피 코딩은 비디오 코딩 기술 또는 표준에 따를 수 있고, 가변 길이 코딩, 허프만 코딩, 상황 감도가 있거나 없는 산술 코딩 등을 포함하는 다양한 원리를 따를 수 있다. 파서(520)는 서브그룹에 대응하는 적어도 하나의 파라미터에 기초하여, 비디오 디코더의 픽셀들의 서브그룹들 중 적어도 하나에 대한 서브그룹 파라미터의 세트를 코딩된 비디오 시퀀스로부터 추출할 수 있다. 서브그룹은 픽처 그룹(Group of Picture, GOP), 픽처, 타일, 슬라이스, 매크로블록, 코딩 유닛(Coding Unit, CU), 블록, 변환 유닛(Transform Unit, TU), 예측 유닛(Prediction Unit, PU) 등을 포함할 수 있다. 파서(520)는 또한 변환 계수(예를 들어, 푸리에 변환 계수), 양자화기 파라미터 값, 모션 벡터 등과 같은 코딩된 비디오 시퀀스 정보로부터 추출할 수 있다.

[0040] 파서(520)는 버퍼 메모리(515)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 작동을 수행하여 심볼(521)을 생성할 수 있다.

[0041] 심볼(521)의 재구성은 코딩된 비디오 픽처의 유형 또는 그 일부(예: 인터 및 인트라 픽처, 인터 및 인트라

블록), 및 기타 인자에 따라 다수의 상이한 처리 유닛 또는 기능 유닛을 포함할 수 있다. 포함되는 유닛 및 유닛이 포함되는 방식은 파서(520)에 의해 코딩된 비디오 시퀀스로부터 파싱되는 서브그룹 제어 정보에 의해 제어될 수 있다. 파서(520)와 아래의 다수의 처리 유닛 또는 기능 유닛 사이의 이러한 서브그룹 제어 정보의 흐름은 간결함을 위해 도시되지 않는다.

[0042] 이미 언급된 기능 블록 외에, 비디오 디코더(510)는 개념적으로 아래에서 설명되는 바와 같이 다수의 기능 유닛으로 세분될 수 있다. 상업적 제약 조건에서 작동하는 실제 구현에서, 이러한 기능 유닛들 중 많은 것이 서로 밀접하게 상호작용하고, 적어도 부분적으로 서로 통합될 수 있다. 그러나, 개시된 주제의 다양한 기능을 명확하게 설명하기 위해 기능 유닛으로의 개념적 세분화가 아래의 개시에서 채택된다.

[0043] 제1 유닛은 스케일러/역변환 유닛(551)을 포함할 수 있다. 스케일러/역변환 유닛(551)은 사용할 역변환 유형을 지시하는 정보, 블록 크기, 양자화 인자/파라미터, 양자화 스케일링 행렬, 파서(520)로부터의 심볼(들)(521)로서의 라이(lie)를 포함하는 제어 정보 뿐만 아니라 양자화된 변환 계수를 수신할 수 있다. 스케일러/역변환 유닛(551)은 애그리게이터(555)에 입력될 수 있는 샘플 값을 포함하는 블록을 출력할 수 있다.

[0044] 일부 경우에, 스케일러/역변환(551)의 출력 샘플은 인트라 코딩된 블록, 즉, 이전에 재구성된 픽처로부터 예측 정보를 사용하지 않지만 현재 픽처의 이전에 재구성된 부분으로부터 예측 정보를 사용할 수 있는 블록에 속할 수 있다. 이러한 예측 정보는 인트라 픽처 예측 유닛(552)에 의해 제공될 수 있다. 일부 경우에, 인트라 픽처 예측 유닛(552)은 이미 재구성된 주변 블록 정보이자 또한 현재 픽처 베퍼(558)에 저장된 주변 블록 정보를 사용하여 재구성 중인 블록과 동일한 크기 및 형상의 블록을 생성할 수 있다. 현재 픽처 베퍼(558)는 예를 들어 부분적으로 재구성된 현재 픽처 및/또는 완전히 재구성된 현재 픽처를 베퍼링한다. 애그리게이터(555)는, 일부 구현에서, 샘플 단위로, 인트라 예측 유닛(552)이 생성한 예측 정보를 스케일러/역변환 유닛(551)에 의해 제공되는 출력 샘플 정보에 추가할 수 있다.

[0045] 다른 경우에, 스케일러/역변환 유닛(551)의 출력 샘플은 인터 코딩되고 잠재적으로 모션 보상된 블록에 속할 수 있다. 그러한 경우에, 모션 보상 예측 유닛(553)은 픽처 간 예측에 사용되는 샘플들을 폐치하기 위해 참조 픽처 메모리(557)에 액세스할 수 있다. 블록에 속하는 심볼(521)에 따라 폐치된 샘플을 모션 보상한 후, 이러한 샘플은 애그리게이터(555)에 의해 스케일러/역변환 유닛(551)의 출력(유닛(551)의 출력은 잔차 샘플 또는 잔차 신호로서 지칭될 수 있음)에 추가될 수 있어서 출력 샘플 정보를 생성할 수 있다. 모션 보상 예측 유닛(553)이 예측 샘플을 폐치하는 참조 픽처 메모리(557) 내의 주소는 예를 들어, X, Y 컴포넌트(шу프트) 및 참조 픽처 성분(시간)을 가질 수 있는 심볼(521)의 형태로 모션 보상 예측 유닛(553)에서 이용가능할 수 있는 모션 벡터에 의해 제어될 수 있다. 모션 보상은 또한 정확한 서브 샘플 모션 벡터가 사용 중일 때 참조 픽처 메모리(557)로부터 폐치된 샘플 값의 보간을 포함할 수 있고, 또한 모션 벡터 예측 메커니즘 등과 연관될 수 있다.

[0046] 애그리게이터(555)의 출력 샘플은 루프 필터 유닛(556)에서 다양한 루프 필터링 기술의 대상이 될 수 있다. 비디오 압축 기술은 코딩된 비디오 시퀀스(또한 코딩된 비디오 비트스트림으로 지칭됨)에 포함된 파라미터에 의해 제어되고 파서(520)로부터의 심볼(521)로서 루프 필터 유닛(556)에 이용가능하게 되는 인루프 필터 기술을 포함할 수 있지만, 이전에 재구성되고 루프 필터링된 샘플 값에 대응할 뿐만 아니라 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서에서) 부분을 디코딩하는 동안 획득된 메타 정보에 대응할 수도 있다. 여러 유형의 루프 필터는 루프 필터 유닛(556)의 일부로서 다양한 순서로 포함될 수 있으며, 이는 아래에서 더 자세히 설명될 것이다.

[0047] 루프 필터 유닛(556)의 출력은 렌더 장치(512)로 출력될 수 있을 뿐만 아니라 미래의 픽처 간 예측에 사용하기 위해 참조 픽처 메모리(557)에 저장될 수 있는 샘플 스트림일 수 있다.

[0048] 완전히 재구성된 특정 코딩된 픽처는 미래 예측을 위한 참조 픽처로서 사용될 수 있다. 예를 들어, 현재 픽처에 대응하는 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 (예를 들어, 파서(520)에 의해) 참조 픽처로서 식별되면, 현재 픽처 베퍼(558)는 참조 픽처 메모리(557)의 일부가 될 수 있고, 신선한 현재 픽처 베퍼는 다음 코딩된 픽처의 재구성을 시작하기 전에 재할당될 수 있다.

[0049] 비디오 디코더(210)는 ITU-T Rec. H.265와 같은 표준에서 채택된 미리 결정된 비디오 압축 기술에 따라 디코딩 작동을 수행할 수 있다. 코딩된 비디오 시퀀스가 비디오 압축 기술 또는 표준의 신택스와 비디오 압축 기술 또는 표준에서 문서화된 프로파일 모두를 준수한다는 의미에서, 코딩된 비디오 시퀀스는 사용 중인 비디오 압축 기술 또는 표준에 의해 지정된 신택스를 따를 수 있다. 구체적으로, 프로파일은 해당 프로파일에서 사용할 수 있는 유일한 도구로서 비디오 압축 기술 또는 표준에서 사용할 수 있는 모든 도구로부터 특정 도구를 선택할 수

있다. 표준 준수를 위해, 코딩된 비디오 시퀀스의 복잡성이 비디오 압축 기술 또는 표준의 수준에 의해 정의된 범위 내에 있을 수 있다.. 일부 경우에, 레벨이 최대 픽처 크기, 최대 프레임 속도, 최대 재구성 샘플 속도(예를 들어, 초당 폐가샘플로 측정됨), 최대 참조 픽처 크기 등을 제한한다. 레벨에 의해 설정된 제한은 경우에 따라 코딩된 비디오 시퀀스에서 시그널링되는 가상 참조 디코더(Hypothetical Reference Decoder, HRD) 사양 및 HRD 버퍼 관리를 위한 메타데이터를 통해 추가로 제한될 수 있다.

[0050] 일부 예시적인 실시예에서, 수신기(510)는 인코딩된 비디오와 함께 추가(중복) 데이터를 수신할 수 있다. 추가 데이터는 코딩된 비디오 시퀀스(들)의 일부로 포함될 수 있다. 추가 데이터는 비디오 디코더(510)에 의해 데이터를 적절하게 디코딩하고 및/또는 원래의 비디오 데이터를 더 정확하게 재구성하기 위해 사용될 수 있다. 추가 데이터는 예를 들어 시간, 공간 또는 신호 잡음 비율(signal noise ration, SNR) 항상 계층, 중복 슬라이스, 중복 픽처, 순방향 오류 정정 코드 등의 형태일 수 있다.

[0051] 도 6은 본 개시의 예시적인 실시예에 따른 비디오 인코더(603)의 블록도를 도시한다. 비디오 인코더(603)는 전자 장치(620)에 포함될 수 있다. 전자 장치(620)는 전송기(640)(예를 들어, 전송 회로)를 더 포함할 수 있다. 비디오 인코더(603)는 도 4의 예에서 비디오 인코더(403) 대신에 사용될 수 있다.

[0052] 비디오 인코더(603)는 비디오 인코더(603)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(601)(도 6의 예에서 전자 장치(620)의 일부가 아님)로부터 비디오 샘플을 수신할 수 있다. 다른 예에서, 비디오 소스(601)는 전자 장치(620)의 일부로서 구현될 수 있다.

[0053] 비디오 소스(601)는 임의의 적절한 비트 깊이(예를 들어, 8 비트, 10 비트, 12 비트, ...), 임의의 색상 공간(예를 들어, BT.601 YCrCB, RGB, XYZ...) 및 임의의 적절한 샘플링 구조(예를 들어, YCrCb 4:2:0, YCrCb 4:4:4)일 수 있는 디지털 비디오 샘플 스트림의 형태로 비디오 인코더(603)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 매체 서빙 시스템에서, 비디오 소스(601)는 미리 준비된 비디오를 저장할 수 있는 저장 장치일 수 있다. 픽처 회의 시스템에서, 비디오 소스(601)는 비디오 시퀀스로서 로컬 이미지 정보를 캡처하는 카메라일 수 있다. 비디오 데이터는 순서대로 볼 때 모션을 부여하는 복수의 개별 픽처 또는 이미지로서 제공될 수 있다. 픽처 자체는 픽셀의 공간 어레이로 구성될 수 있으며, 여기서 각각의 픽셀은 사용 중인 샘플링 구조, 색 공간 등에 따라 하나 이상의 샘플을 포함할 수 있다. 당업자는 픽셀과 샘플 사이의 관계를 쉽게 이해할 수 있다. 아래 설명은 샘플에 중점을 둔다.

[0054] 일부 예시적인 실시예에 따르면, 비디오 인코더(603)는 실시간으로 또는 애플리케이션에 의해 요구되는 임의의 다른 시간 제약 하에서 소스 비디오 시퀀스의 픽처를 코딩된 비디오 시퀀스(643)로 코딩하고 압축할 수 있다. 적절한 코딩 속도를 강화하는 것은 제어기(650)의 기능 중 하나를 구성한다. 일부 실시예에서, 제어기(650)는 후술하는 바와 같이 다른 기능 유닛을 제어하고 다른 기능 유닛에 기능적으로 결합될 수 있다. 결합은 간결함을 위해 표시되지 않는다. 제어기(650)에 의해 설정되는 파라미터는 레이트 제어 관련 파라미터(픽처 스킁(picture skip), 양자화기, 비율 왜곡 최적화(rate-distortion optimization) 기술의 람다 값, ...), 픽처 크기, 픽처 그룹(GOP) 레이아웃, 최대 모션 벡터 검색 범위 등을 포함할 수 있다. 제어기(650)는 특정 시스템 설계에 최적화된 비디오 인코더(603)에 속하는 다른 적절한 기능을 갖도록 구성될 수 있다.

[0055] 일부 예시적인 실시예에서, 비디오 인코더(603)는 코딩 루프에서 작동하도록 구성될 수 있다. 과도하게 단순화된 설명으로서, 예에서, 코딩 루프는 소스 코더(630)(예를 들어, 코딩될 입력 픽처에 기초하여 심볼 스트림과 같은 심볼을 생성하는 역할을 하며, 참조 픽처(들)임) 및 비디오 인코더(603)에 내장된 (로컬) 디코더(633)를 포함할 수 있다. 디코더(633)는 임베디드 디코더(633)가 엔트로피 코딩 없이 소스 코더(630)에 의해 코딩된 비디오 스트림을 처리하더라도 (원격) 디코더가 생성할 것이라는 것과 같은 유사한 방식으로 샘플 데이터를 생성하기 위해 심볼을 재구성한다(심볼과 엔트로피 코딩의 코딩된 비디오 비트스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술에서 무손실이기 때문임). 재구성된 샘플 스트림(샘플 데이터)은 참조 픽처 메모리(634)에 입력될 수 있다. 심볼 스트림의 디코딩이 디코더 위치(로컬 또는 원격)에 관계없이 정확한 비트 결과를 가져오기 때문에, 참조 픽처 메모리(634)내의 컨텐츠는 또한 로컬 인코더와 원격 인코더 사이에서 정확한 비트이다. 다시 말해서, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "보는(see)" 것과 정확히 동일한 샘플 값을 참조 픽처 샘플로 "본다(see)". 참조 픽처 동기화(예를 들어, 채널 오류로 인해 동기화가 유지될 수 없는 경우 결과적인 드리프트)의 이러한 기본 원리는 코딩 품질을 향상시키는 데 사용된다.

[0056] "로컬" 디코더(633)의 작동은 비디오 디코더(510)와 같은 "원격" 디코더의 작동과 동일할 수 있으며, 이는 도 5와 관련하여 위에서 이미 상세하게 설명되었다. 그러나, 또한 도 5를 간단히 참조하면, 심볼이 이용 가능하고 엔트로피 코더(645) 및 파서(520)에 의한 코딩된 비디오 시퀀스에 대한 심볼의 인코딩/디코딩이 무손실일 수 있

기 때문에, 베퍼 메모리(515) 및 파서(520)를 포함하는 비디오 디코더(510)의 엔트로피 디코딩 부분은 인코더의 로컬 디코더(633)에서 완전히 구현되지 않을 수 있다.

[0057] 이 시점에서 행해질 수 있는 관찰은 디코더에서만 존재할 수 있는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한 대응하는 인코더에 실질적으로 동일한 기능 형태로 존재할 필요가 있을 수 있다는 것이다. 이러한 이유로, 개시된 주제는 때때로 인코더의 디코딩 부분과 연합하는 디코더 작동에 초점을 맞출 수 있다. 따라서, 인코더 기술에 대한 설명은 포괄적으로 설명된 디코더 기술의 반대이므로 축약될 수 있다. 특정 영역 또는 측면에서만 인코더의 더 자세한 설명이 필요하며 아래에서 제공된다.

[0058] 작동 중에, 일부 예시적인 구현에서, 소스 코더(630)는 "참조 픽처"로 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 픽처를 참조하여 예측적으로 입력 픽처를 코딩하는 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(632)은 입력 픽처의 픽셀 블록과 입력 픽처에 대한 예측 참조(들)로서 선택될 수 있는 참조 픽처(들)의 픽셀 블록 사이의 색상 채널의 차이(또는 잔차)를 코딩한다.

[0059] 로컬 비디오 디코더(633)는 소스 코더(430)에 의해 생성된 심볼에 기초하여 참조 픽처로 지정될 수 있는 픽처의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(632)의 작동은 유리하게는 순실 프로세스일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 6에 도시되지 않음)에서 디코딩될 수 있는 경우, 재구성된 비디오 시퀀스는 일반적으로 일부 오류가 있는 소스 비디오 시퀀스의 복제일 수 있다. 로컬 비디오 디코더(633)는 참조 픽처에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스를 복제하고 재구성된 참조 픽처로 하여금 참조 픽처 캐시(634)에 저장되도록 할 수 있다. 이러한 방식으로, 비디오 인코더(603)는 원단(원격) 비디오 디코더(전송 오류 없음)에 의해 획득될 재구성된 참조 픽처로서 공통 컨텐츠를 갖는 재구성된 참조 픽처의 복사를 로컬에서 저장할 수 있다.

[0060] 예측기(635)는 코딩 엔진(632)에 대한 예측 검색을 수행할 수 있다. 즉, 코딩될 새로운 픽처에 대해, 예측기(635)는 샘플 데이터(후보 참조 픽셀 블록으로서) 또는 새로운 픽처에 대한 적절한 예측 참조 역할을 할 수 있는, 참조 픽처 모션 벡터, 블록 형상 등과 같은 특정 메타데이터에 대해 참조 픽처 메모리(634)를 검색할 수 있다. 예측기(635)는 적절한 예측 참조를 찾기 위해 샘플 블록 기반으로 작동할 수 있다. 일부 경우에, 예측기(635)에 의해 획득된 검색 결과에 의해 결정된 바와 같이, 입력 픽처는 참조 픽처 메모리(634)에 저장된 다수의 참조 픽처로부터 도출된 예측 참조를 가질 수 있다.

[0061] 제어기(650)는 예를 들어 비디오 데이터를 인코딩하는데 사용되는 파라미터 및 서브그룹 파라미터의 설정을 포함하는 소스 코더(630)의 코딩 작동을 관리할 수 있다.

[0062] 전술한 모든 기능 유닛의 출력은 엔트로피 코더(645)에서 엔트로피 코딩될 수 있다. 엔트로피 코더(645)는 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술에 따라 심볼을 무손실 압축함으로써 다양한 기능 유닛에 의해 생성된 심볼을 코딩된 비디오 시퀀스로 변환한다.

[0063] 전송기(640)는 인코딩된 비디오 데이터를 저장할 수 있는 저장 장치에 대한 하드웨어/소프트웨어 링크일 수 있는 통신 채널(660)을 통한 전송을 준비하기 위해 엔트로피 코더(645)에 의해 생성된 코딩된 비디오 시퀀스(들)를 베퍼링할 수 있다. 전송기(640)는 비디오 코더(603)로부터의 코딩된 비디오 데이터를 전송될 다른 데이터, 예를 들어 코딩된 오디오 데이터 및/또는 보조 데이터 스트림(소스는 도시되지 않음)과 병합할 수 있다.

[0064] 제어기(650)는 비디오 인코더(603)의 작동을 관리할 수 있다. 코딩 동안, 제어기(650)는 각각의 픽처에 적용될 수 있는 코딩 기술에 영향을 미칠 수 있는 특정 코딩된 픽처 유형을 각각의 코딩된 픽처에 할당할 수 있다. 예를 들어, 픽처는 종종 다음의 픽처 유형들 중 하나로서 할당될 수 있다:

[0065] 인트라 픽처(I 픽처)는 예측 소스로서 시퀀스의 다른 픽처를 사용하지 않고 코딩되고 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱은 예를 들어 독립 디코더 리프레시(Independent Decoder Refresh, IDR) 픽처를 포함하는 다양한 유형의 인트라 픽처를 허용한다. 당업자는 I 픽처의 이러한 변형 및 각각의 애플리케이션 및 특징을 알고 있다.

[0066] 예측 픽처(P 픽처)는 각각의 블록의 샘플 값을 예측하기 위해 최대 하나의 모션 벡터 및 참조 인덱스를 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다.

[0067] 양방향 예측 픽처(B 픽처)는 각각의 블록의 샘플 값을 예측하기 위해 최대 2개의 모션 벡터 및 참조 인덱스를 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다. 유사하게, 다수의 예측 픽처는 단일 블록의 재구성을 위해 2개 이상의 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.

- [0068] 소스 픽처는 일반적으로 복수의 샘플 코딩 블록(예를 들어, 각각 4×4 , 8×8 , 4×8 또는 16×16 샘플의 블록)으로 공간적으로 세분되고 블록 기반으로 코딩될 수 있다. 블록은 블록의 각각의 픽처에 적용된 코딩 할당에 의해 결정된 바와 같이 다른(이미 코딩된) 블록을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 픽처의 블록은 비예측적으로 코딩될 수 있거나 또는 그들은 동일한 픽처의 이미 코딩된 블록을 참조하여 예측적으로 코딩될 수 있다(공간 예측 또는 인트라 예측). P 픽처의 픽셀 블록은 하나의 이전에 코딩된 참조 픽처를 참조하여 공간적 예측을 통해 또는 시간적 예측을 통해 예측적으로 코딩될 수 있다. B 픽처의 블록은 하나 또는 2개의 이전에 코딩된 참조 픽처를 참조하여 공간적 예측을 통해 또는 시간적 예측을 통해 예측적으로 코딩될 수 있다. 소스 픽처 또는 중간 처리된 픽처는 다른 목적을 위해 다른 유형의 블록으로 세분화될 수 있다. 코딩 블록 및 다른 유형의 블록의 분할은 아래에서 더 상세히 설명되는 바와 같이 동일한 방식을 따를 수도 있고 따르지 않을 수도 있다.
- [0069] 비디오 인코더(603)는 ITU-T Rec. H265와 같은 미리 결정된 비디오 코딩 기술 또는 표준에 따라 코딩 작동을 수행할 수 있다. 그 작동에서, 비디오 인코더(603)는 입력 비디오 시퀀스에서 시간적 및 공간적 중복성을 이용하는 예측 코딩 작동을 포함하는 다양한 압축 작동을 수행할 수 있다. 이에 따라, 코딩된 비디오 데이터는 사용되는 비디오 코딩 기술 또는 표준에 의해 지정된 신택스를 따를 수 있다.
- [0070] 일부 예시적인 실시예에서, 전송기(640)는 인코딩된 비디오와 함께 추가 데이터를 전송할 수 있다. 소스 코더(630)는 코딩된 비디오 시퀀스의 일부로서 그러한 데이터를 포함할 수 있다. 추가 데이터는 시간/공간/SNR 향상 계층, 중복 픽처 및 슬라이스, SEI 메시지, VUI 파라미터 세트 단편 등과 같은 다른 형태의 중복 데이터를 포함할 수 있다.
- [0071] 비디오는 시간적 시퀀스에서 복수의 소스 픽처(비디오 픽처)로 캡처될 수 있다. 픽처 간 예측(종종 인트라 예측으로 축약됨)은 주어진 픽처에서 공간적 상관을 사용하고, 픽처 간 예측은 픽처 사이의 (시간적 또는 다른) 상관을 사용한다. 예를 들면, 현재 픽처로 지칭되는 인코딩/디코딩 중인 특정 픽처는 블록으로 분할될 수 있다. 비디오에서 이전에 코딩되고 여전히 버퍼링된 참조 픽처의 참조 블록과 유사한 경우, 현재 픽처의 블록은 모션 벡터로 지칭되는 벡터에 의해 코딩될 수 있다. 모션 벡터는 참조 픽처의 참조 블록을 가리키며, 여러 참조 픽처가 사용 중인 경우 참조 픽처를 식별하는 3차원을 가질 수 있다.
- [0072] 일부 예시적인 실시예에서, 양방향 예측 기술은 픽처 간 예측에 사용될 수 있다. 이러한 양방향 예측 기술에 따르면, 디코딩 순서에서 비디오의 현재 픽처에 모두 앞선(그러나, 디스플레이 순서에서 각각 과거 및 미래일 수 있음) 제1 참조 픽처 및 제2 참조 픽처와 같은 두 개의 참조 픽처가 사용된다. 현재 픽처의 블록은 제1 참조 픽처의 제1 참조 블록을 가리키는 제1 모션 벡터 및 제2 참조 픽처의 제2 참조 블록을 가리키는 제2 모션 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 공동으로 예측될 수 있다.
- [0073] 또한, 코딩 효율을 향상시키기 위해 픽처 간 예측에 병합 모드 기술이 사용될 수 있다.
- [0074] 본 개시의 일부 예시적인 실시예에 따르면, 픽처 간 예측 및 픽처 내 예측과 같은 예측은 블록 단위로 수행된다. 예를 들어, 비디오 픽처 시퀀스의 픽처는 압축을 위해 코딩 트리 유닛(coding tree unit)으로 분할되며, 픽처의 CTU는 64×64 픽셀, 32×32 픽셀 또는 16×16 픽셀과 같은 동일한 크기를 가질 수 있다. 일반적으로, CTU는 1개의 루마(luma) CTB와 2개의 크로마(chroma) CTB인 3개의 병렬 코딩 트리 블록(coding tree block, CTB)을 포함할 수 있다. 각각의 CTU는 재귀적으로 쿼드트리를 하나 또는 여러 개의 코딩 유닛(coding unit, CU)으로 분할될 수 있다. 예를 들어, 64×64 픽셀의 CTU는 64×64 픽셀의 하나의 CU, 또는 32×32 픽셀의 4개의 CU로 분할될 수 있다. 32×32 블록 중 하나 이상의 각각은 16×16 픽셀의 4개의 CU로 추가로 분할될 수 있다. 일부 예시적인 실시예에서, 각각의 CU는 인터 예측 유형 또는 인트라 예측 유형과 같은 다양한 예측 유형 중에서 CU에 대한 예측 유형을 결정하기 위해 인코딩 중에 분석될 수 있다. CU는 시간 및/또는 공간 예측 가능성에 따라 하나 이상의 예측 유닛(prediction unit, PU)으로 분할될 수 있다. 일반적으로, 각각의 PU는 루마 예측 블록(PB)과 2개의 크로마 PB를 포함한다. CU를 PU(또는 상이한 색상 채널의 PB)로 분할하는 것은 다양한 공간 패턴에서 수행될 수 있다. 예를 들어, 루마 또는 크로마 PB는 8×8 픽셀, 16×16 픽셀, 8×16 픽셀, 16×8 샘플 등과 같은 샘플에 대한 값(예를 들어, 루마 값)의 행렬을 포함할 수 있다.
- [0075] 도 7은 본 개시의 다른 예시적인 실시예에 따른 비디오 인코더(703)의 도면을 도시한다. 비디오 인코더(703)는 비디오 픽처의 시퀀스에서 현재 비디오 픽처 내의 샘플 값의 처리 블록(예를 들어, 예측 블록)을 수신하고, 처리 블록을 코딩된 비디오 시퀀스의 일부인 코딩된 픽처로 인코딩하도록 구성된다. 예시적인 비디오 인코더(703)는 도 4 예의 비디오 인코더(403) 대신에 사용될 수 있다.

- [0076] 예를 들어, 비디오 인코더(703)는 8 x 8 샘플의 예측 블록 등과 같은 처리 블록에 대한 샘플 값의 행렬을 수신한다. 따라서, 비디오 인코더(703)는 처리 블록이 예를 들어 비율 왜곡 최적화(rate-distortion optimization, RDO)를 사용하여 인트라 모드, 인터 모드 또는 이중 예측 모드를 사용하여 최상으로 코딩되는지 여부를 결정한다. 처리 블록이 인트라 모드에서 코딩되는 것으로 결정되는 경우, 비디오 인코더(703)는 처리 블록을 코딩된 픽처로 인코딩하기 위해 인트라 예측 기술을 사용할 수 있고, 처리 블록이 인터 모드 또는 이중 예측 모드에서 코딩되는 것으로 결정되는 경우, 비디오 인코더(703)는 처리 블록을 코딩된 픽처로 인코딩하기 위해 인터 예측 또는 이중 예측 기술을 각각 사용할 수 있다. 일부 예시적인 실시예에서, 병합 모드는 모션 벡터가 예측기 외부의 코딩된 모션 벡터 성분의 이점 없이 하나 이상의 모션 벡터 예측기로부터 유도되는 픽처 간 예측의 서브 모드로서 사용될 수 있다. 일부 다른 예시적인 실시예에서, 대상 블록에 적용 가능한 모션 벡터 성분이 존재할 수 있다. 따라서, 비디오 인코더(703)는 처리 블록의 예측 모드를 결정하기 위한 모드 결정 모듈과 같은 도 7에 명시적으로 도시되지 않은 컴포넌트를 포함할 수 있다.
- [0077] 도 7의 예에서, 비디오 인코더(703)는 인터 인코더(730), 인트라 인코더(722), 잔차 계산기(723), 스위치(726), 잔차 인코더(724), 일반 제어기(721) 및 도 7의 예시적인 배열에서 도시된 바와 같이 함께 결합된 엔트로피 인코더(725)를 포함한다.
- [0078] 인터 인코더(730)는 현재 블록(예를 들어, 처리 블록)의 샘플을 수신하고, 블록을 참조 픽처(예를 들어, 디스플레이 순서에서 이전 픽처 및 이후 픽처의 블록)의 하나 이상의 참조 블록과 비교하며, 인터 예측 정보(예를 들어, 인터 인코딩 기술에 따른 중복 정보의 설명, 모션 벡터, 병합 모드 정보)를 생성하고, 임의의 적절한 기술을 사용하여 인터 예측 정보에 기초하여 인터 예측 결과(예를 들어, 예측된 블록)를 계산하도록 구성된다. 일부 예에서, 참조 픽처는 도 6의 예시적인 인코더(620)에 내장된 디코딩 유닛(633)(아래에서 더 상세하게 설명되는 바와 같이, 도 7의 잔차 디코더(728)로서 도시됨)을 사용하여 인코딩된 비디오 정보에 기초하여 디코딩된 디코딩된 참조 픽처이다.
- [0079] 인트라 인코더(722)는 현재 블록(예를 들어, 처리 블록)의 샘플을 수신하고, 블록을 동일한 이미 코딩된 블록과 비교하며, 변환 후에 양자화된 계수를 생성하고, 그리고 일부 경우에 또한 인트라 예측 정보(예를 들어, 하나 이상의 인트라 인코딩 기술에 따른 인트라 예측 방향 정보)를 생성하도록 구성된다. 인트라 인코더(722)는 동일한 픽처의 인트라 예측 정보 및 참조 블록에 기초하여 인트라 예측 결과(예를 들어, 예측된 블록)를 계산할 수 있다.
- [0080] 일반 제어기(721)는 일반 제어 데이터를 결정하고 일반 제어 데이터에 기초하여 비디오 인코더(703)의 다른 컴포넌트를 제어하도록 구성될 수 있다. 예에서, 일반 제어기(721)는 블록의 예측 모드를 결정하고, 그 예측 모드에 기초하여 스위치(726)에게 제어 신호를 제공한다. 예를 들어, 예측 모드가 인트라 모드인 경우, 일반 제어기(721)는 잔차 계산기(723)에 의해 사용할 인트라 모드 결과를 선택하기 위해 스위치(726)를 제어하고, 인트라 예측 정보를 선택하고 비트스트림에 인트라 예측 정보를 포함시키기 위해 엔트로피 인코더(725)를 제어하며, 블록에 대한 예측 모드가 인터 모드인 경우, 일반 제어기(721)는 잔차 계산기(723)에 의해 사용할 인터 예측 결과를 선택하기 위해 스위치(726)를 제어하고, 인터 예측 정보를 선택하고 비트스트림에 인터 예측 정보를 포함시키기 위해 엔트로피 인코더(725)를 제어한다.
- [0081] 잔차 계산기(723)는 수신된 블록과 인트라 인코더(722) 또는 인터 인코더(730)로부터 선택된 블록에 대한 예측 결과 사이의 차이(잔차 데이터)를 계산하도록 구성될 수 있다. 잔차 인코더(724)는 변환 계수를 생성하기 위해 잔차 데이터를 인코딩하도록 구성될 수 있다. 예를 들어, 잔차 인코더(724)는 변환 계수를 생성하기 위해 잔차 데이터를 공간 도메인에서 주파수 도메인으로 변환하도록 구성될 수 있다. 그 다음, 변환 계수는 양자화된 변환 계수를 획득하기 위해 양자화 처리된다. 다양한 예시적인 실시예에서, 비디오 인코더(703)는 또한 잔차 디코더(728)를 포함한다. 잔차 디코더(728)는 역변환을 수행하고, 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(722) 및 인터 인코더(730)에 의해 적절하게 사용될 수 있다. 예를 들어, 인터 인코더(730)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기초하여 디코딩된 블록을 생성할 수 있고, 인트라 인코더(722)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기초하여 디코딩된 블록을 생성할 수 있다. 디코딩된 블록은 디코딩된 픽처를 생성하도록 적절하게 처리되고 디코딩된 픽처는 메모리 회로(도시되지 않음)에 버퍼링될 수 있으며 참조 픽처로서 사용될 수 있다.
- [0082] 엔트로피 인코더(725)는 인코딩된 블록을 포함하고 엔트로피 코딩을 수행하기 위해 비트스트림을 포맷하도록 구성될 수 있다. 엔트로피 인코더(725)는 비트스트림에서 다양한 정보를 포함하도록 구성된다. 예를 들어, 엔트로피 인코더(725)는 비트스트림에 일반 제어 데이터, 선택된 예측 정보(예를 들어, 인트라 예측 정보 또는 인터

예측 정보), 잔차 정보 및 다른 적절한 정보를 포함하도록 구성될 수 있다. 인터 모드 또는 이중 예측 모드의 병합 서브모드에서 블록을 코딩하는 경우, 잔차 정보가 없음에 유의한다.

[0083] 도 8은 본 개시의 다른 실시예에 따른 예시적인 비디오 디코더(810)의 도면을 도시한다. 비디오 디코더(810)는 코딩된 비디오 시퀀스의 일부인 코딩된 픽처를 수신하고, 재구성된 픽처를 생성하기 위해 코딩된 픽처를 디코딩 하도록 구성된다. 예에서, 비디오 디코더(810)는 도 4의 예에서 비디오 디코더(410) 대신에 사용될 수 있다.

[0084] 도 8의 예에서, 비디오 디코더(810)는 엔트로피 디코더(871), 인터 디코더(880), 잔차 디코더(873), 재구성 모듈(874) 및 도 8의 예시적인 배열에서 도시된 바와 같이 함께 결합된 인트라 디코더(872)를 포함한다.

[0085] 엔트로피 디코더(871)는 코딩된 픽처를 구성하는 신팩스 요소를 나타내는 특정 심볼을 코딩된 픽처로부터 재구성하도록 구성될 수 있다. 그러한 심볼은 예를 들어, 블록이 코딩되는 모드(예를 들어, 인트라 모드, 인터 모드, 이중 예측 모드, 병합 서브모드 또는 다른 서브모드), 인트라 디코더(872) 또는 인터 디코더(880)에 의한 예측을 위해 사용되는 특정 샘플 또는 메타데이터를 식별할 수 있는 예측 정보(예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 예를 들어 양자화된 변환 계수의 형태의 잔차 정보 등을 포함할 수 있다. 예에서, 예측 모드가 인터 또는 이중 예측 모드인 경우, 인터 예측 정보는 인터 디코더(880)에게 제공되고, 예측 유형이 인트라 예측 유형인 경우, 인트라 예측 정보는 인트라 디코더(872)에게 제공된다. 잔차 정보는 역양자화될 수 있고 잔차 디코더(873)에게 제공된다.

[0086] 인터 디코더(880)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기초하여 인터 예측 결과를 생성하도록 구성될 수 있다.

[0087] 인트라 디코더(872)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기초하여 예측 결과를 생성하도록 구성될 수 있다.

[0088] 잔차 디코더(873)는 역양자화된 변환 계수를 추출하기 위해 역양자화를 수행하고, 잔차를 주파수 도메인에서 공간 도메인으로 변환하기 위해 역양자화된 변환 계수를 처리하도록 구성될 수 있다. 잔차 디코더(873)는 또한 엔트로피 디코더(871)에 의해 제공될 수 있는(양자화기 파라미터(Quantizer Parameter, QP)를 포함하기 위해) 특정 제어 정보를 사용할 수 있다(이것은 단지 낮은 데이터 볼륨 제어 정보일 수 있으므로 데이터 경로가 도시되지 않음).

[0089] 재구성 모듈(874)은 재구성된 비디오의 일부로서 재구성된 픽처의 일부를 형성하는 재구성된 블록을 형성하기 위해 공간 도메인에서 잔차 디코더(873)에 의해 출력된 잔차와 예측 결과(경우에 따라 인터 또는 인트라 예측 모듈에 의해 출력됨)를 결합하도록 구성될 수 있다. 디블록킹(deblocking) 작동과 같은 다른 적절한 작동이 또한 시각적 품질을 향상시키기 위해 수행될 수 있다.

[0090] 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 임의의 적절한 기술을 사용하여 구현될 수 있음에 유의한다. 일부 예시적인 실시예에서, 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 하나 이상의 집적 회로를 사용하여 구현될 수 있다. 다른 실시예에서, 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 소프트웨어 명령을 실행하는 하나 이상의 프로세서를 사용하여 구현될 수 있다.

[0091] 코딩 블록 분할로 전환하고, 일부 예시적인 구현에서, 미리 결정된 패턴이 적용될 수 있다. 도 9에 도시된 바와 같이, 제1 미리 정의된 레벨(예를 들어, 64 x 64 블록 레벨)에서 시작하여 제2 미리 정의된 레벨(예를 들어, 4 x 4 레벨)까지의 예시적인 4 방향 파티션(partition) 트리가 채용될 수 있다. 예를 들어, 기본 블록은 902, 904, 906 및 908에 의해 지시된 4개의 분할 옵션의 대상이 될 수 있으며, R로 지정된 파티션은 도 9에서 지시된 것과 동일한 파티션 트리가 가장 낮은 레벨(예를 들어, 4 x 4 레벨)까지 낮은 스케일로 반복될 수 있다는 점에서 재귀적 파티션에 대해 허용된다. 일부 구현에서, 도 9의 분할 방식에 추가적인 제한이 적용될 수 있다. 도 9에의 구현에서, 직사각형 파티션(예를 들어, 1:2/2:1 직사각형 파티션)이 허용될 수 있지만 재귀적으로는 허용되지 않을 수 있는 반면, 정사각형 파티션은 재귀적으로 허용된다. 재귀를 사용하는 도 9에 따른 분할은, 필요한 경우, 코딩 블록의 최종 세트를 생성한다. 이러한 방식은 하나 이상의 색상 채널에 적용될 수 있다.

[0092] 도 10은 분할 트리를 형성하기 위해 재귀적 분할을 허용하는 미리 정의된 분할 패턴의 다른 예를 도시한다. 도 10에 도시된 바와 같이, 예시적인 10 방향 분할 구조 또는 패턴이 미리 정의될 수 있다. 루트 블록은 미리 정의된 레벨(예를 들어, 128 x 128 레벨, 또는 64 x 64 레벨로부터)에서 시작할 수 있다. 도 10의 예시적인 분할 구조는 다양한 2:1/1:2 및 4:1/1:4 직사각형 파티션을 포함한다. 도 10의 두 번째 행에서 1002, 1004, 1006 및 1008으로 지시된 3개의 서브 파티션을 갖는 파티션 유형은 "T 유형" 파티션으로 지칭될 수 있다. "T 유형"

파티션(1002, 1004, 1006, 1008)은 좌측 T 유형, 상단 T 유형, 우측 T 유형 및 하단 T 유형으로 지정될 수 있다. 일부 구현에서, 도 10의 직사각형 파티션들 중 어느 것도 추가로 세분화되도록 허용되지 않는다. 코딩 트리 깊이는 루트 노드 또는 루트 블록으로부터의 분할 깊이를 지시하기 위해 추가로 정의될 수 있다. 예를 들어, 루트 노드 또는 루트 블록에 대한 코딩 트리 깊이, 예를 들어 128 x 128 블록의 경우, 0으로 설정될 수 있으며, 루트 블록이 도 10에 따라 한 번 더 분할된 후, 코딩 트리 깊이는 1만큼 증가될 수 있다. 일부 구현에서, 1010의 전체 정사각형 파티션만이 도 10의 패턴을 따르는 분할 트리의 다음 레벨로의 재귀적 분할을 허용할 수 있다. 즉, 패턴(1002, 1004, 1006, 1008)을 갖는 정사각형 파티션에 대해 재귀적 분할이 허용되지 않을 수 있다. 재귀를 사용한 도 10에 따른 분할은, 필요한 경우, 코딩 블록의 최종 세트를 생성한다. 이러한 방식은 하나 이상의 색상 채널에 적용될 수 있다.

[0093]

위의 분할 절차 또는 다른 절차 중 임의의 것에 따라 기본 블록을 나누거나 분할한 후, 다시 파티션 또는 코딩 블록의 최종 세트가 획득될 수 있다. 이러한 파티션 각각은 다양한 분할 레벨 중 하나에 있을 수 있다. 각각의 파티션은 코딩 블록(coding block, CB)으로 지정될 수 있다. 위의 다양한 예시적인 분할 구현에 대해, 각각의 결과적인 CB는 임의의 허용된 크기 및 분할 레벨일 수 있다. 그것들은 일부 기본적인 코딩/디코딩 결정이 행해질 수 있고 코딩/디코딩 파라미터가 인코딩된 비디오 비트스트림에서 최적화되고, 결정되며, 시그널링될 수 있는 유닛을 형성할 수 있기 때문에 코딩 블록으로 지정된다. 최종 파티션에서 가장 높은 레벨은 코딩 블록 분할 트리의 깊이를 나타낸다. 코딩 블록은 루마 코딩 블록 또는 크로마 코딩 블록일 수 있다.

[0094]

일부 다른 예시적인 구현에서, 기본 루마 및 크로마 블록을 코딩 단위로 재귀적으로 분할하기 위해 큐드트리 구조가 사용될 수 있다. 이러한 분할 구조는 기본 CTU의 다양한 로컬 특성에 분할을 적응시키기 위해 큐드트리 구조를 사용함으로써 코딩 유닛(coding unit, CU)으로 분할되는 코딩 트리 유닛(coding tree unit, CTU)으로 지정될 수 있다. 그러한 구현에서, 크기가 꽉쳐 경계에 맞을 때까지 블록이 큐드 트리 분할을 유지할 수 있도록 암시적 큐드트리 분할이 꽉쳐 경계에서 수행될 수 있다. CU라는 용어는 루마 및 크로마 코딩 블록(CB)의 유닛을 집합적으로 지칭하는 데 사용된다.

[0095]

일부 구현에서, CB는 추가로 분할될 수 있다. 예를 들어, CB는 코딩 및 디코딩 프로세스 동안 인트라 또는 프레임 간 예측을 위해 다중 예측 블록(prediction block, PB)으로 추가로 분할될 수 있다. 다시 말해서, CB는 개별적인 예측 결정/구성이 행해질 수 있는 상이한 서브 파티션으로 추가로 분할될 수 있다. 병렬로, CB는 비디오 데이터의 변환 또는 역변환이 수행되는 레벨을 기술하기 위해 복수의 변환 블록(transform block, TB)으로 추가로 분할될 수 있다. CB를 PB와 TB로 분할하는 방식은 동일하거나 동일하지 않을 수 있다. 예를 들어, 각각의 분할 방식은 예를 들어 비디오 데이터의 다양한 특성에 기초한 고유한 절차를 사용하여 수행될 수 있다. PB 및 TB 분할 방식은 일부 예시적인 구현에서 독립적일 수 있다. PB 및 TB 분할 방식 및 경계는 일부 다른 예시적인 구현에서 상관될 수 있다. 일부 구현에서, 예를 들어, TB는 PB 파티션 후에 분할될 수 있고, 특히 각각의 PB는 코딩 블록의 분할 후에 결정된 후에, 하나 이상의 TB로 추가로 분할될 수 있다. 예를 들어, 일부 구현에서, PB는 1, 2, 4, 또는 다른 개수의 TB로 분할될 수 있다.

[0096]

일부 구현에서, 기본 블록을 코딩 블록으로 그리고 추가로 예측 블록 및/또는 변환 블록으로 분할하기 위해, 루마 채널 및 크로마 채널은 상이하게 처리될 수 있다. 예를 들어, 일부 구현에서, 예측 블록 및/또는 변환 블록으로의 코딩 블록의 분할은 루마 채널에 대해 허용될 수 있는 반면, 코딩 블록의 예측 블록 및/또는 변환 블록으로의 이러한 분할은 크로마 채널(들)에 대해 허용되지 않을 수 있다. 따라서, 그러한 구현에서, 루마 블록의 변환 및/또는 예측은 코딩 블록 레벨에서만 수행될 수 있다. 다른 예를 들면, 루마 채널 및 크로마 채널(들)에 대한 최소 변환 블록 크기는 상이할 수 있다. 예를 들어, 루마 채널에 대한 코딩 블록은 크로마 채널보다 더 작은 변환 및/또는 예측 블록으로 분할되도록 허용될 수 있다. 또 다른 예를 들면, 코딩 블록을 변환 블록 및/또는 예측 블록으로 분할하는 최대 깊이는 루마 채널과 크로마 채널 사이에 상이할 수 있다. 예를 들어, 루마 채널에 대한 코딩 블록은 크로마 채널(들)보다 더 깊은 변환 및/또는 예측 블록으로 분할되도록 허용될 수 있다. 특정 예를 들면, 루마 코딩 블록은 최대 2 레벨까지 내려가는 재귀적인 파티션에 의해 표현될 수 있는 다중 크기의 변환 블록으로 분할될 수 있으며, 정사각형, 2:1/1:2 및 4:1/1:4과 같은 변환 블록 형상 및 4 x 4에서 64 x 64까지의 변환 블록 크기가 허용될 수 있다. 그러나, 크로마 블록의 경우, 루마 블록에 대해 지정된 가능한 가장 큰 변환 블록만이 허용될 수 있다.

[0097]

코딩 블록을 PB로 분할하기 위한 일부 예시적인 구현에서, PB 분할의 깊이, 형상, 및/또는 다른 특성은 PB가 인트라 코딩되는지 또는 인터 코딩되는지에 의존할 수 있다.

[0098]

코딩 블록(또는 예측 블록)을 변환 블록으로 분할하는 것은 큐드트리 분할 및 미리 정의된 패턴 분할을 포함하

지만 이에 제한되지 않는 다양한 예시적인 방식으로, 재귀적으로 또는 비재귀적으로, 구현될 수 있으며, 코딩 블록 또는 예측 블록의 경계에서 변환 블록에 대한 추가적인 고려 사항을 갖는다. 일반적으로, 결과적인 변환 블록은 상이한 분할 레벨에 있을 수 있고, 동일한 크기가 아닐 수 있으며, 형상이 정사각형일 필요가 없을 수 있다(예를 들어, 그들은 일부 허용된 크기 및 종횡비를 갖는 직사각형일 수 있음).

[0099] 일부 구현에서, 코딩 분할 트리 방식 또는 구조가 사용될 수 있다. 루마 및 크로마 채널에 사용되는 코딩 분할 트리 방식은 동일할 필요는 없다. 즉, 루마 및 크로마 채널은 별도의 코딩 트리 구조를 가질 수 있다. 또한, 루마 및 크로마 채널이 동일하거나 상이한 코딩 분할 트리 구조를 사용하는지 여부와 사용되는 실제 코딩 분할 트리 구조는 코딩되는 슬라이스가 P, B 또는 I 슬라이스인지 여부에 따라 달라질 수 있다. 예를 들어, I 슬라이스의 경우, 크로마 채널 및 루마 채널은 별도의 코딩 분할 트리 구조 또는 코딩 분할 트리 구조 모드를 가질 수 있는 반면, P 또는 B 슬라이스의 경우, 루마 및 크로마 채널은 동일한 코딩 분할 트리 방식을 공유할 수 있다. 별도의 코딩 분할 트리 구조 또는 모드가 적용되는 경우, 루마 채널은 하나의 코딩 분할 트리 구조에 의해 CB로 분할될 수 있고, 크로마 채널은 다른 코딩 분할 트리 구조에 의해 크로마 CB로 분할될 수 있다.

[0100] 코딩 블록 및 변환 블록 분할의 특정 예시적인 구현이 아래에서 설명된다. 이러한 예시적인 구현에서, 기본 코딩 블록은 위에서 설명된 재귀적인 큐드트리 분할을 사용하여 코딩 블록으로 분할될 수 있다. 각각의 레벨에서, 특정 파티션의 추가 큐드트리 분할이 계속되어야 하는지 여부는 로컬 비디오 데이터 특성에 의해 결정될 수 있다. 결과적인 CB는 다양한 크기의 다양한 큐드트리 분할 레벨에 있을 수 있다. 꾹쳐 간(시간) 또는 꾹쳐 내(공간) 예측을 사용하여 꾹쳐 영역을 코딩할지 여부에 대한 결정은 CB 레벨(또는 모든 3색 채널에 대한 CU 레벨)에서 이루어질 수 있다. 각각의 CB는 PB 분할 유형에 따라 1개, 2개, 4개 또는 다른 개수의 PB로 추가로 분할될 수 있다. 하나의 PB 내에서, 동일한 예측 과정이 적용될 수 있으며, 관련 정보는 PB 단위로 디코더로 전송된다. PB 분할 유형에 기초한 예측 프로세스를 적용하여 잔차 블록을 획득한 후, CB는 CB에 대한 코딩 트리와 유사한 다른 큐드트리 구조에 따라 TB로 분할할 수 있다. 이러한 특정 구현에서, CB 또는 TB는 정사각형 형상으로 제한될 수 있지만 이것으로만 제한되지는 않는다. 추가로 이러한 특정 예에서, PB는 인터 예측을 위한 정사각형 또는 직사각형 형상일 수 있고 단지 인트라 예측을 위한 정사각형일 수도 있다. 코딩 블록은 예를 들어 4개의 정사각형 형상의 TB로 추가로 분할될 수 있다. 각각의 TB는 잔차 큐드트리(Residual Quad-Tree, RQT)로 지칭되는 더 작은 TB로 재귀적으로(큐드트리 분할을 사용하여) 추가로 분할될 수 있다.

[0101] 기본 코딩 블록을 CB 및 다른 PB 및/또는 TB로 분할하기 위한 다른 특정 예가 아래에서 설명된다. 예를 들어, 도 10에 도시된 것과 같은 다중 파티션 유닛 유형을 사용하는 것보다, 이진 및 삼진 분할 세그먼테이션 구조를 사용하는 중첩된 다중 유형 트리를 갖는 큐드트리가 사용될 수 있다. CB, PB 및 TB 개념의 분할(즉, CB를 PB 및/또는 TB로 분할, 및 PB를 TB로 분할)은 최대 변환 길이에 비해 크기가 너무 큰 크기를 갖는 CB에 필요한 경우를 제외하고 포기될 수 있으며, 여기서 이러한 CB는 추가 분할이 필요할 수 있다. 이러한 예시적인 분할 방식은, 예측 및 변환이 둘 다 추가 분할 없이 CB 레벨에 대해 수행될 수 있도록, CB 분할 형상에 대해 더 많은 유연성을 지원하도록 설계될 수 있다. 이러한 코딩 트리 구조에서, CB는 정사각형 또는 직사각형 형상 중 하나를 가질 수 있다. 구체적으로, 코딩 트리 블록(coding tree block, CTB)은 먼저 큐드트리 구조에 의해 분할될 수 있다. 그런 다음, 큐드트리 리프(leaf) 노드는 다중 유형 트리 구조로 추가로 분할될 수 있다. 다중 유형 트리 구조의 예는 도 11에 도시되어 있다. 구체적으로, 도 11의 예시적인 다중 유형 트리 구조는 수직 이진 분할(SPLIT_BT_VER)(1102), 수평 이진 분할(SPLIT_BT_HOR)(1104), 수직 삼진 분할(SPLIT_TT_VER)(1106) 및 수평 삼진 분할(SPLIT_TT_HOR)(1108)로 지칭되는 4가지 분할 유형을 포함한다. 그런 다음, CB는 다중 유형 트리의 잎(leave)에 대응한다. 이러한 예시적인 구현에서, CB가 최대 변환 길이에 비해 너무 크지 않는 한, 이러한 분할은 추가 분할 없이 예측 및 변환 처리 모두에 사용된다. 이는 대부분의 경우, CB, PB 및 TB가 중첩된 다중 유형 트리 코딩 블록 구조를 갖는 큐드트리에서 동일한 블록 크기를 갖는다는 것을 의미한다. 지원되는 최대 변환 길이가 CB의 색상 성분의 폭 또는 높이보다 작은 경우에 예외가 발생한다.

[0102] 하나의 CTB에 대한 블록 파티션의 중첩된 다중 유형 트리 코딩 블록 구조를 갖는 큐드트리에 대한 일 예가 도 12에 도시되어 있다. 보다 상세하게, 도 12는 CTB(1200)가 4개의 정사각형 파티션(1202, 1204, 1206, 1208)으로 분할된 큐드트리임을 보여준다. 분할을 위한 도 11의 다중 유형 트리 구조를 추가로 사용하기 위한 결정은 큐드트리 분할 파티션 각각에 대해 행해진다. 도 12의 예에서, 파티션(1204)은 더 이상 분할되지 않는다. 파티션(1202, 1208)은 각각 다른 큐드트리 분할을 채택한다. 파티션(1202)의 경우, 제2 레벨 큐드트리 분할 좌측 상단, 우측 상단, 좌측 하단 및 우측 하단 파티션은 각각 큐드트리의 제3 레벨 분할, 도 11의 1104, 비 분할, 및 도 11의 1108을 채택한다. 파티션(1208)은 또 다른 큐드트리 분할을 채택하고, 제2 레벨 큐드트리 분할, 좌측 상단, 우측 상단, 좌측 하단 및 우측 하단 파티션은 각각 도 11의 1106의 제3 레벨 분할, 비 분할, 비 분할

및 도 11의 1104를 채택한다. 1208의 제3 레벨 좌측 상단 파티션의 2개의 서브 파티션은 1104 및 1108에 따라 추가로 분할된다. 파티션(1206)은 도 12의 1102를 따르는 제2 레벨 분할 패턴을 도 11의 1108 및 1102에 따른 제3 레벨로 추가로 분할되는 2개의 파티션으로 채택한다.

[0103] 위의 특정 예에 대해, 최대 루마 변환 크기는 64×64 일 수 있고 최대 지원 크로마 변환 크기는 예를 들어 32×32 에서 루마와 다를 수 있다. 루마 코딩 블록 또는 크로마 코딩 블록의 폭 또는 높이가 최대 변환 폭 또는 높이보다 더 큰 경우, 루마 코딩 블록 또는 크로마 코딩 블록은 해당 방향에서의 변환 크기 제한을 충족하기 위해 수평 및/또는 수직 방향으로 자동 분할될 수 있다.

[0104] 위의 기본 코딩 블록을 CB로 분할하기 위한 특정 예에서, 코딩 트리 방식은 루마 및 크로마가 별도의 블록 트리 구조를 가질 수 있는 능력을 지원할 수 있다. 예를 들어, P 및 B 슬라이스의 경우, 하나의 CTU의 루마 및 크로마 CTB는 동일한 코딩 트리 구조를 공유할 수 있다. 예를 들어, I 슬라이스의 경우, 루마 및 크로마는 별도의 코딩 블록 트리 구조를 가질 수 있다. 별도의 블록 트리 모드가 적용되는 경우, 루마 CTB는 하나의 코딩 트리 구조에 의해 루마 CB로 분할될 수 있고, 크로마 CTB는 다른 코딩 트리 구조에 의해 크로마 CB로 분할될 수 있다. 이것은 I 슬라이스의 CU가 루마 성분의 코딩 블록 또는 2개의 크로마 성분의 코딩 블록으로 구성될 수 있고, P 또는 B 슬라이스의 CU가 비디오가 단색이 아닌 한 항상 세 가지 색상 성분의 코딩 블록으로 구성된다.

[0105] 코딩 블록 또는 예측 블록을 변환 블록으로 분할하기 위한 예시적인 구현, 및 변환 블록의 코딩 순서가 아래에서 더 상세하게 설명된다. 일부 예시적인 구현에서, 변환 분할은 예를 들어, 4×4 내지 64×64 의 범위를 갖는 변환 블록을 사용하여 다중 형상, 예를 들어 1:1(정사각형), 1:2/2:1, 및 1:4/4:1의 변환 블록을 지원할 수 있다. 일부 구현에서, 코딩 블록이 64×64 보다 작거나 같으면, 변환 블록 분할은 크로마 블록의 경우 변환 블록 크기가 코딩 블록 크기와 동일하도록 루마 성분에만 적용될 수 있다. 그렇지 않고, 코딩 블록 폭 또는 높이가 64보다 크면, 루마 및 크로마 코딩 블록 모두는 각각 암시적으로 $\min(W, 64) \times \min(H, 64)$ 및 $\min(W, 32) \times \min(H, 32)$ 변환 블록의 배수로 분할될 수 있다.

[0106] 일부 예시적인 구현에서, 인트라 및 인터 코딩된 블록 모두에 대해, 코딩 블록은 미리 정의된 개수의 레벨(예를 들어, 2 레벨)까지의 분할 깊이를 갖는 다중 변환 블록으로 추가로 분할될 수 있다. 변환 블록 분할 깊이 및 크기는 관련될 수 있다. 현재 깊이의 변환 크기에서 다음 깊이의 변환 크기로의 매핑 예는 다음의 [표 1]에 나타나 있다.

[0107] [표 1]은 변환 파티션 크기 설정을 나타낸다.

표 1

현재 깊이의 변환 크기	다음 깊이의 변환 크기
TX_4X4	TX_4X4
TX_8X8	TX_4X4
TX_16X16	TX_8X8
TX_32X32	TX_16X16
TX_64X64	TX_32X32
TX_4X8	TX_4X4
TX_8X4	TX_4X4
TX_8X16	TX_8X8
TX_16X8	TX_8X8
TX_16X32	TX_16X16
TX_32X16	TX_16X16
TX_32X64	TX_32X32
TX_64X32	TX_32X32
TX_4X16	TX_4X8
TX_16X4	TX_8X4
TX_8X32	TX_8X16
TX_32X8	TX_16X8
TX_16X64	TX_16X32
TX_64X16	TX_32X16

[0108]

[표 1]의 예시적인 매핑에 기초하여, 1:1 정사각형 블록에 대해, 다음 레벨 변환 분할은 4개의 1:1 정사각형 서브 변환 블록을 생성할 수 있다. 변환 파티션은 예를 들어 4×4 에서 중단될 수 있다. 이와 같이, 4×4 의 현재 깊이에 대한 변환 크기는 다음 깊이에 대한 4×4 의 동일한 크기에 대응한다. [표 1]의 예에서, 1:2/2:1 비 정사각형 블록의 경우, 다음 레벨 변환 분할은 2개의 1:1 정사각 서브 변환 블록을 생성하는 반면, 1:4/4:1 비 정사각형 블록의 경우, 다음 레벨 변환 분할은 2개의 1:2/2:1 서브 변환 블록을 생성할 것이다.

[0109]

일부 예시적인 구현에서, 인트라 코딩된 블록의 루마 성분에 대해, 추가 제한이 적용될 수 있다. 예를 들어, 변환 분할의 각각의 레벨에 대해, 모든 서브 변환 블록은 동일한 크기를 갖도록 제한될 수 있다. 예를 들어, 32×16 코딩 블록의 경우, 레벨 1 변환 분할은 2개의 16×16 서브 변환 블록을 생성하고, 레벨 2 변환 분할은 8개의 8×8 서브 변환 블록을 생성한다. 다시 말해서, 제2 레벨 분할은 변환 유닛을 동일한 크기로 유지하기 위해 모든 제1 레벨 서브 블록에 적용되어야 한다. [표 1]에 따른 인트라 코딩된 정사각형 블록에 대한 변환 블록 분할의 예는 화살표로 도시된 코딩 순서와 함께 도 13에 도시되어 있다. 구체적으로, 1302는 정사각형 코딩 블록을 나타낸다. [표 1]에 따른 4개의 동일한 크기의 변환 블록으로의 제1 레벨 분할은 화살표로 지시된 코딩 순서와 함께 1304에 도시된다. [표 1]에 따른 16개의 동일한 크기의 변환 블록으로의 모든 제1 레벨 동일 크기의 블록의 제2 레벨 분할이 화살표로 지시된 코딩 순서와 함께 1306에 도시된다.

[0111]

일부 예시적인 구현에서, 인터 코딩된 블록의 루마 성분에 대해, 인트라 코딩에 대한 상기 제한이 적용되지 않을 수 있다. 예를 들어, 변환 분할의 제1 레벨 이후, 서브 변환 블록 중 어느 하나는 하나 이상의 레벨과 독립적으로 추가로 분할될 수 있다. 따라서, 결과적인 변환 블록은 동일한 크기일 수도 있고 아닐 수도 있다. 코딩 순서에 따른 인터 코딩된 블록의 변환 블록으로의 예시적인 분할은 도 14에 도시되어 있다. 도 14의 예에서, 인터 코딩된 블록(1402)은 [표 1]에 따라 2개의 레벨에서 변환 블록으로 분할된다. 제1 레벨에서, 인터 코딩된 블록은 동일한 크기의 4개의 변환 블록으로 분할된다. 그런 다음, 4개의 변환 블록 중 하나만(모두가 아님)이 4개의 서브 변환 블록으로 추가로 분할되어 1404로 표시된 것처럼 2개의 상이한 크기를 갖는 총 7개의 변환 블록이 생성된다. 이러한 7개의 변환 블록의 예시적인 코딩 순서는 도 14의 1404에서 화살표로 도시된다.

- [0112] 일부 예시적인 구현에서, 크로마 성분(들)의 경우, 변환 블록에 대한 일부 추가 제한이 적용될 수 있다. 예를 들어, 크로마 성분(들)에 대해 변환 블록 크기는 코딩 블록 크기만큼 줄 수 있지만, 미리 정의된 크기, 예를 들어 8×8 보다 작지 않다.
- [0113] 일부 다른 예시적인 구현에서, 폭(W) 또는 높이(H)가 64보다 큰 코딩 블록에 대해, 루마 및 크로마 코딩 블록 모두는 각각 암시적으로 $\min(W, 64) \times \min(H, 64)$ 및 $\min(W, 32) \times \min(H, 32)$ 변환 유닛의 배수로 분할될 수 있다.
- [0114] 도 15는 코딩 블록 또는 예측 블록을 변환 블록으로 분할하기 위한 다른 대안의 예시적인 방식을 추가로 도시한다. 도 15에 도시된 바와 같이, 재귀적 변환 분할을 사용하는 대신에, 미리 정의된 분할 유형의 세트가 코딩 블록의 변환 유형에 따라 코딩 블록에 적용될 수 있다. 도 15에 도시된 특정 예에서, 6개의 예시적인 분할 유형 중 하나는 코딩 블록을 다양한 개수의 변환 블록으로 분할하는 데 적용될 수 있다. 이러한 방식은 코딩 블록 또는 예측 블록에 적용될 수 있다.
- [0115] 보다 구체적으로, 도 15의 분할 방식은 도 15에 도시된 바와 같이 임의의 주어진 변환 유형에 대해 최대 6개의 파티션 유형을 제공한다. 이러한 방식에서, 모든 코딩 블록 또는 예측 블록은 예를 들어, 비율 왜곡 비용 (rate-distortion cost)에 기초하여 변환 유형이 할당될 수 있다. 일 예에서, 코딩 블록 또는 예측 블록에 할당된 파티션 유형은 코딩 블록 또는 예측 블록의 변환 파티션 유형에 기초하여 결정될 수 있다. 특정 파티션 유형은 도 15에 도시된 4가지 파티션 유형에 의해 나타낸 바와 같이, 변환 블록 분할 크기 및 패턴(또는 파티션 타입)에 대응할 수 있다. 다양한 변환 유형과 다양한 파티션 유형 사이의 대응 관계는 미리 정의될 수 있다. 비율 왜곡 비용에 기초하여 코딩 블록 또는 예측 블록에 할당될 수 있는 변환 유형을 지시하는 대문자로 표시된 레이블과 함께 예시적인 대응이 아래에서 표시된다.
- PARTITION_NONE: 블록 크기와 동일한 변환 크기를 할당한다.
 - PARTITION_SPLIT: 블록 크기의 폭의 1/2 및 블록 크기의 높이의 1/2인 변환 크기를 할당한다.
 - PARTITION_HORZ: 블록 크기와 동일한 폭 및 블록 크기의 높이의 1/2인 변환 크기를 할당한다.
 - PARTITION_VERT: 블록 크기의 폭의 1/2 및 블록 크기와 동일한 높이인 변환 크기를 할당한다.
 - PARTITION_HORZ4: 블록 크기와 동일한 폭 및 블록 크기의 높이의 1/4인 변환 크기를 할당한다.
 - PARTITION_VERT4: 블록 크기의 폭의 1/4 및 블록 크기와 동일한 높이로 변환 크기를 할당한다.
- [0116] 위의 예에서, 도 15에 도시된 바와 같은 파티션 유형은 모두 분할된 변환 블록에 대한 균일한 변환 크기를 포함한다. 이것은 제한이 아니라 단순한 예이다. 일부 다른 구현에서, 혼합된 변환 블록 크기는 특정 파티션 유형(또는 패턴)의 분할된 변환 블록에 대해 사용될 수 있다.
- [0117] 다양한 실시예에서, 반 분리된 분할(semi decoupled partitioning, SDP) 방식이 사용될 수 있으며, 여기서 하나의 슈퍼 블록(super block, SB)의 루마 및 크로마 블록은 동일하거나 상이한 블록 분할 구조를 가질 수 있다. SDP 방식은 또한 반 분할 트리(semi-separate tree, SST) 또는 크로마 성분에 대한 유연한 블록 분할이라고도 한다.
- [0118] 일부 구현에서, SDP 방식은 루마 코딩된 블록의 크기, 또는 루마 블록 분할 트리의 깊이에 의존할 수 있다. 일 예로, 루마 블록의 크기(예를 들어, 면적)가 제1 임계값(T1)보다 크거나 또는 루마 블록의 코딩 트리 분할 깊이가 제2 임계값(T2)보다 작거나 같은 경우, 크로마 블록은 루마 블록과 동일한 코딩 트리 구조를 사용할 수 있다.
- [0119] 그렇지 않으면, 다른 예를 들면, 루마 블록의 크기가 T1보다 작거나 같은 경우, 또는 루마 분할 깊이가 T2보다 큰 경우, 대응하는 크로마 블록은 루마 성분과의 상이한 코딩 블록 분할을 가질 수 있으며, 이는 크로마 성분에 대한 유연한 블록 분할이라고 한다. 일부 구현에서, T1은 양의 정수이지만, 예를 들어 128 또는 256으로 제한되지 않는다. T2는 양의 정수이지만, 예를 들어 1 또는 2로 제한되지 않는다.
- [0120] 도 16은 루마 성분(1610) 및 크로마 성분(1650)에 대한 상이한 코딩 블록 분할 구조의 예를 도시하며, 여기서 T2는 1로 설정된다. 제1 레벨(레벨 = 1)에서, 루마 성분 및 크로마 성분은 동일한 구조를 갖는다. 제2 레벨(레벨 = 2)에서, 루마 성분의 일부 레벨 1 블록과 크로마 성분의 일부 레벨 1 블록은 동일한 구조, 예를 들어 각각 1611 및 1651, 1641 및 1681을 가지며, 루마 성분의 일부 레벨 1 블록과 크로마 성분의 일부 레벨 1 블록

은 상이한 구조, 예를 들어, 각각 1621 및 1661, 1631 및 1671을 갖는다.

[0127] 다양한 실시예에서, 개선된 반 분리된 분할(SDP) 방식이 사용될 수 있으며, 여기서 루마 및 크로마 성분은 슈퍼 블록의 루트 노드로부터 부분 트리 구조를 공유할 수 있고, 루마 및 크로마가 분할 트리 분할을 시작할 때의 조건은 루마의 분할 정보에 좌우된다.

[0128] 반 분리된 분할의 구현과 연관되어 몇 가지 문제/이슈가 있을 수 있다. 예를 들어, 일부 구현에서, 크로마 성분은 변환 분할을 허용하지 않을 수 있다. 즉, 변환 유닛 크기는 코딩 유닛 크기와 동일하다. 루마 성분과 크로마 성분이 상이한 코딩 블록 분할 트리를 사용하는 경우, 크로마 성분은 일반적으로 더 큰 분할 크기를 선호할 수 있고, 및/또는 개별적으로 시그널링되는 경우 크로마 성분에 대한 변환 분할이 유리할 수 있다.

[0129] 본 개시는 위에서 논의된 이슈들/문제들 중 적어도 하나를 해결하는, 비디오 코딩 및/또는 디코딩에서의 변환 블록 분할을 위한 다양한 실시예를 설명한다.

[0130] 다양한 실시예에서, 도 17의 비디오 디코딩에서 블록 분할을 변환하기 위한 방법(1700)을 참조한다. 이 방법(1700)은, 명령어를 저장하는 메모리 및 메모리와 통신하는 프로세서를 포함하는 장치에 의해, 루마 블록 및 크로마 블록에 대한 코딩된 비디오 비트스트림을 수신하는 단계 1710 – 루마 블록은 크로마 블록과 같은 위치에 있음 –; 상기 장치에 의해, 루마 코딩 블록 분할 트리를 획득하기 위해 루마 블록을 분할하는 단계 1720; 상기 장치에 의해, 크로마 블록 분할 트리를 획득하기 위해 크로마 블록을 분할하는 단계 1730; 상기 장치에 의해, 복수의 루마 변환 블록을 획득하기 위해 루마 코딩 블록 분할 트리로부터 루마 코딩 블록을 분할하는 단계 1740; 및/또는 상기 장치에 의해, 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 크로마 코딩 블록 분할 트리로부터 크로마 코딩 블록을 분할하는 단계 1750 중 일부 또는 전부를 포함할 수 있다.

[0131] 일부 구현에서, 장치는 루마 코딩 블록 및 크로마 코딩 블록을 개별적으로 분할할 수 있고, 획득된 복수의 루마 변환 블록은 획득된 적어도 하나의 복수의 크로마 변환 블록 중 어느 하나와 상이하다.

[0132] 일부 다른 구현에서, 장치는 루마 코딩 블록 및 크로마 코딩 블록을 개별적으로 분할할 수 있고, 획득된 복수의 루마 변환 블록은 획득된 적어도 하나의 복수의 크로마 변환 블록 중 하나와 동일하다.

[0133] 본 개시의 다양한 실시예에서, 블록(예를 들어, 코딩 블록, 예측 블록 또는 변환 블록으로 제한되지 않음)의 크기는 블록의 폭 또는 높이를 지칭할 수 있다. 블록의 폭 또는 높이는 픽셀 유닛에서 정수일 수 있다.

[0134] 본 개시의 다양한 실시예에서, 블록(예를 들어, 코딩 블록, 예측 블록, 또는 변환 블록으로 제한되지 않음)의 크기는 블록의 영역 크기를 지칭할 수 있다. 블록의 영역 크기는 픽셀 유닛에서 블록의 폭에 블록의 높이를 곱하여 계산되는 정수일 수 있다.

[0135] 본 개시의 다양한 일부 실시예에서, 블록(예를 들어, 코딩 블록, 예측 블록, 또는 변환 블록으로 제한되지 않음)의 크기는 블록의 폭 또는 높이의 최대값, 블록의 폭 또는 높이의 최소값, 또는 블록의 종횡비를 지칭할 수 있다. 블록의 종횡비는 블록의 폭을 높이로 나눈 값으로 계산될 수 있거나, 또는 블록의 높이를 폭으로 나눈 값으로 계산될 수 있다.

[0136] 본 개시의 다양한 실시예에서, 슈퍼 블록(SB)은 최대 코딩 유닛(largest coding unit, LCU), 예를 들어, 128x128 블록을 지칭할 수 있다. 루마 블록은 크로마 블록과 같은 위치에 있는 루마 블록을 지칭할 수 있다. 반 분리된 분할(SDP)은 반 분리된 트리(semi-decoupled tree, SDT)로서도 지칭될 수 있다. 블록 분할(splitting)은 블록 분할(partitioning)로도 지칭될 수 있다.

[0137] 단계 1710을 참조하면, 장치는 도 5에서의 전자 장치(530)일 수 있거나, 또는 도 8에서의 비디오 디코더(810)일 수 있다. 일부 구현에서, 장치는 도 6에서의 인코더(620)의 디코더(633)일 수 있다. 다른 구현에서, 장치는 도 5에서의 전자 장치(530)의 일부, 도 8에서의 비디오 디코더(810)의 일부, 도 6에서의 인코더(620)의 디코더(633)의 일부일 수 있다. 코딩된 비디오 비트스트림은 도 8에서의 코딩된 비디오 시퀀스, 또는 도 6 또는 7에서의 중간 코딩된 데이터일 수 있다. 코딩된 비디오 비트스트림은 루마 블록(예를 들어, 도 18에서의 1810) 및 크로마 블록(예를 들어, 도 18에서의 1850)을 위한 것일 수 있고, 루마 블록 및 크로마 블록은 각각 함께 위치된다.

[0138] 일부 구현에서, 별개의 변환 유닛 분할 및 시그널링이 상이한 색상 성분, 예를 들어, 루마 성분 및 크로마 성분에 대해 적용될 수 있다. 크로마 성분은 하나보다 많은 색상 성분, 예를 들어 제1 색상 성분(청색) 및 제2 색상 성분(적색)을 포함할 수 있다.

- [0139] 도 18은 코딩 블록 분할이 상이한 경우 별도의 변환 유닛 분할을 적용하는 예를 도시하며, 여기서 실선은 코딩 블록 분할 경계를 지시하고 점선은 변환 유닛 분할 경계를 지시한다. 루마 블록(1810)은 레벨 1에서 1811, 1821, 1831 및 1841을 포함한다. 크로마 블록(1850)은 레벨 1에서 1851, 1861, 1871 및 1881을 포함한다. 도 18은 코딩 블록 분할(실선) 패턴이 예를 들어 1811 및 1851, 1821 및 1861, 1841 및 1881에서 루마와 크로마 성분 사이에 상이한 반면, 변환 블록 분할(점선)은 루마 및 크로마 모두에 대해 활성화되지만 상이한 분할 패턴, 예를 들어 1821 및 1861을 가지는 것을 나타낸다.
- [0140] 단계 1720을 참조하면, 장치는 루마 코딩 블록 분할 트리를 획득하기 위해 루마 블록을 분할할 수 있다. 도 18을 참조하면, 루마 블록(1810)은 1810에서 실선으로 도시된 루마 코딩 블록 분할 트리를 획득하기 위해 분할된다.
- [0141] 단계 1730을 참조하면, 장치는 크로마 코딩 블록 분할 트리를 획득하기 위해 크로마 블록을 분할할 수 있다. 도 18을 참조하면, 크로마 블록(1850)은 1850에서 실선으로 도시된 크로마 코딩 블록 분할 트리를 획득하기 위해 분할된다.
- [0142] 단계 1740을 참조하면, 장치는 복수의 루마 변환 블록을 획득하기 위해 루마 코딩 블록 분할 트리로부터 루마 코딩 블록을 분할할 수 있다. 1810에서 일부 루마 코딩 블록은 1810에서 점선으로 도시된 복수의 루마 변환 블록으로 추가로 분할되도록 분할된다. 일 예로, 루마 코딩 블록(1811의 상반부)은 2개의 루마 변환 블록(1812, 1813)으로 분할되도록 분할된다. 다른 예를 들면, 루마 코딩 블록(1831)은 4개의 루마 변환 블록(1832, 1833, 1834, 1835)으로 분할되도록 분할된다.
- [0143] 단계 1750을 참조하면, 장치는 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 크로마 코딩 블록 분할 트리로부터 크로마 코딩 블록을 분할할 수 있다. 1850에서 일부 루마 코딩 블록은 1850에서 점선으로 도시된 복수의 루마 변환 블록으로 추가로 분할되도록 분할된다. 일 예로, 크로마 코딩 블록(1861의 우측 절반)은 2개의 크로마 변환 블록(1863, 1864)으로 분할되도록 분할된다. 다른 예를 들면, 크로마 코딩 블록(1871)은 4개의 크로마 변환 블록(1872, 1873, 1874, 1875)으로 분할되도록 분할된다.
- [0144] 다양한 실시예에서, 루마 코딩 블록 분할 트리는 크로마 코딩 블록 분할 트리와 상이하고, 및/또는 코딩된 비디오 비트스트림은 루마 파라미터 및 크로마 파라미터를 포함한다. 루마 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수도 있고 루마 코딩 블록 및/또는 복수의 루마 변환 블록으로부터의 변환 분할 구조를 지시할 수 있다. 크로마 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 크로마 코딩 블록 및/또는 복수의 크로마 변환 블록으로부터의 변환 분할 구조를 지시할 수 있다.
- [0145] 단계 1740은 복수의 루마 변환 블록을 획득하기 위해 루마 파라미터에 따라 루마 코딩 블록을 분할하는 단계를 포함할 수 있고, 및/또는 단계 1750은 복수의 크로마 변환 블록을 획득하기 위해 크로마 파라미터에 따라 크로마 코딩 블록을 분할하는 단계를 포함할 수 있다.
- [0146] 일부 구현에서, 루마 및 크로마 색상 채널이 상이한 코딩 블록 분할 트리를 적용하는 경우, 변환 유닛 분할은 크로마 채널에 대해 허용될 수 있고 변환 유닛 분할은 루마 채널과 비교하여 크로마 채널에 대해 별도로 시그널링될 수 있다.
- [0147] 다양한 실시예에서, 루마 코딩 블록 분할 트리는 크로마 코딩 블록 분할 트리와 상이하고, 및/또는 코딩된 비디오 비트스트림은 루마 파라미터, 제1 크로마 파라미터, 및 제2 크로마 파라미터를 포함한다. 루마 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 루마 코딩 블록 및/또는 복수의 루마 변환 블록으로부터의 변환 분할 구조를 지시할 수 있다. 제1 크로마 파라미터는 크로마 성분의 제1 색상(예를 들어, 청색)에 대응하고, 제2 크로마 파라미터는 크로마 성분의 제2 색상(예를 들어, 적색)에 대응한다. 제1 크로마 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 크로마 코딩 블록 및/또는 크로마 성분의 제1 색상에 대한 복수의 크로마 변환 블록으로부터의 변환 분할 구조를 지시할 수 있다. 제2 크로마 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 크로마 코딩 블록 및/또는 크로마 성분의 제2 색상에 대한 복수의 크로마 변환 블록으로부터의 변환 분할 구조를 지시할 수 있다. 제1 및 제2 색상에 대한 변환 분할 구조는 다를 수 있다.
- [0148] 단계 1740은 복수의 루마 변환 블록을 획득하기 위해 루마 파라미터에 따라 루마 코딩 블록을 분할하는 단계를 포함할 수 있다. 단계 1750은 제1 복수의 크로마 변환 블록을 획득하기 위해 제1 크로마 파라미터에 따라 크로마 코딩 블록의 제1 성분을 분할하는 단계, 및/또는 제2 복수의 크로마 변환 블록을 획득하기 위해 제2 크로마 파라미터에 따라 크로마 코딩 블록의 제2 성분을 분할하는 단계를 포함할 수 있다. 크로마 코딩 블록의 제1 성

분은 코딩 블록의 제1 색상 채널(예를 들어, 청색)을 지칭할 수 있고, 및/또는 크로마 코딩 블록의 제2 성분은 코딩 블록의 제2 색상 채널(예를 들어, 적색)을 지칭할 수 있다.

[0149] 일부 구현에서, 루마 및 크로마 색상 채널이 상이한 코딩 블록 분할 트리를 적용하는 경우, 변환 유닛 분할이 크로마에 대해 허용되고, 변환 유닛 분할이 크로마의 상이한 색상 성분, 예를 들어 Cb 및 Cr에 대해 별도로 시그널링될 수 있다.

[0150] 일부 다른 구현에서, 색상 성분이 상이한 코딩 블록 분할 트리를 적용하는 경우, 변환 유닛 분할은 각각의 색상 성분에 대해 허용되고 변환 유닛 분할은 각각의 색상 성분에 대해 별도로 시그널링될 수 있다.

[0151] 다양한 실시예에서, 크로마 코딩 블록의 변환 깊이는 0 또는 1을 포함하고, 및/또는 크로마 코딩 블록은 쿼드트리 변환 분할을 포함하며, 및/또는 정사각형이 아닌 직사각형 블록인 크로마 코딩 블록은 이진 변환 분할을 시시한다.

[0152] 일부 구현에서, 상이한 색상 성분에 대해, 상이한 변환 유닛 분할 패턴 및/또는 깊이가 적용될 수 있다. 일 예에서, 크로마 채널 또는 성분에 대해, 변환 깊이는 0 또는 1로 제한될 수 있다. 다른 예에서, 크로마 채널 또는 성분에 대해, 쿼드트리 변환 분할만이 허용된다. 다른 예에서, 크로마 채널 또는 성분에 대해, 정사각형이 아닌 직사각형 블록의 경우, 이진 변환 분할만이 허용된다. 다른 예에서, 수평 이진 변환 분할 또는 수직 이진 변환 분할이 적용되는지 여부는 코딩 블록의 종횡비에 따라 달라지며, 코딩 블록의 종횡비가 1보다 큰 경우(예를 들어, 수평 차원이 코딩 블록의 수직 차원보다 긴 경우), 수직 이진 변환 분할이 사용될 수 있고, 코딩 블록의 종횡비가 1보다 작은 경우(예를 들어, 수평 차원이 코딩 블록의 수직 차원보다 짧은 경우), 수평 이진 변환 분할이 코딩 블록에 사용될 수 있다.

[0153] 다양한 실시예에서, 단계 1750은, 크로마 코딩 블록이 최소 코딩 블록 크기보다 큰 조건, 또는 크로마 코딩 블록이 최대 코딩 블록 크기보다 작은 조건, 또는 적어도 하나의 인트라 예측 모드를 사용하는 크로마 코딩 블록 조건, 또는 적어도 하나의 변환 유형을 사용하는 크로마 코딩 블록 조건 중 적어도 하나를 만족하는 것에 응답하여 적어도 하나의 복수의 크로마 변환 블록을 획득하기 위해 크로마 코딩 블록을 분할하는 단계를 포함할 수 있다. 일부 구현에서, 최소 코딩 블록 크기는 4, 8, 16, 32, 또는 64 중 적어도 하나를 포함하는 최소 코딩 폭에 대응하고, 최대 코딩 블록 크기는 32, 64, 128, 또는 256 중 적어도 하나를 포함하는 최대 코딩 길이에 대응하며, 및/또는 적어도 하나의 인트라 예측 모드는 적어도 하나의 방향성 인트라 예측 모드를 포함한다.

[0154] 일부 다른 구현에서, 상이한 색상 성분에 대해, 변환 분할을 허용하는 최소 코딩 블록 크기는 상이할 수 있다. 일 예에서, MxN보다 작은 크로마 코딩 블록의 경우, 변환 유닛 분할이 허용되지 않는다. M 및 N의 예시적인 값은 4, 8, 16, 32, 64를 포함하지만 이에 제한되지 않는다.

[0155] 일부 다른 구현에서, 상이한 색상 성분에 대해, 변환 분할을 허용하는 최대 코딩 블록 크기는 상이할 수 있다.

[0156] 일부 다른 구현에서, 크기가 MxN보다 큰 크로마 코딩 블록에 대해, 변환 유닛 분할이 허용되지 않을 수 있다. M 및 N의 예시적인 값은 32, 64, 128, 256을 포함하지만 이에 제한되지는 않는다.

[0157] 일부 다른 구현에서, 인트라 코딩된 블록에 대해, 루마 및 크로마 색상 채널이 상이한 코딩 블록 분할 트리를 적용하는 경우, 크로마 색상 채널에서의 변환 유닛 분할은 특정 인트라 예측 모드에 대해서만 허용될 수 있다. 일 예에서, 크로마 색상 채널의 크로마 코딩된 블록에서의 변환 유닛 분할은 방향성 인트라 예측 모드에 대해서만 허용될 수 있다.

[0158] 일부 다른 구현에서, 크로마에서의 변환 유닛 분할이 허용되는 경우, 특정 변환 유형만이 허용된다.

[0159] 다양한 실시예에서, 코딩된 비디오 비트스트림은 루마 코딩 파라미터, 루마 변환 파라미터, 크로마 코딩 파라미터, 및 크로마 변환 파라미터를 포함하고, 및/또는 코딩된 비디오 비트스트림의 엔트로피 디코딩 동안, 크로마 변환 파라미터는 루마 코딩 파라미터, 루마 변환 파라미터, 또는 크로마 코딩 파라미터 중 적어도 하나를 컨택스트로서 사용한다.

[0160] 루마 코딩 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 루마 성분의 코딩 블록 분할 구조를 지시할 수 있다. 루마 변환 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 루마 코딩 블록 및/또는 복수의 루마 변환 블록으로부터의 변환 분할 구조를 지시할 수 있다. 크로마 코딩 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 크로마 성분의 코딩 블록 분할 구조를 지시할 수 있다. 크로마 변환 파라미터는 코딩된 비디오 비트스트림으로부터 추출될 수 있고 크로마 코딩 블록 및/또는 복수의 크로마 변환

블록으로부터의 변환 분할 구조를 지시할 수 있다.

[0161] 일부 다른 구현에서, 크로마 변환 블록 분할 패턴을 시그널링하기 위해, 루마 코딩 블록 분할 플래그 및/또는 변환 블록 분할 플래그가 크로마 변환 블록 분할 플래그를 엔트로피 코딩하기 위한 컨텍스트로서 사용될 수 있다.

[0162] 다양한 실시예에서, 크로마 성분의 코딩된 블록으로부터의 변환 유닛 분할이 허용되는 경우, 모든 크로마 변환 유닛의 크기는 코딩된 블록 내에서 동일할 수 있다. 일 예에서, 크로마 성분(또는 각각의 크로마 채널)의 변환 깊이를 지시하기 위해 플래그가 코딩된 블록 레벨에서 시그널링될 수 있다. 다른 예에서, 코딩된 블록 내의 루마 변환 유닛의 변환 크기는 동일하지 않을 수 있는 반면, 모든 크로마 변환 유닛의 크기는 코딩된 블록 내에서 동일하다.

[0163] 위에서 설명된 본 기술은 컴퓨터 판독 가능 명령을 사용하여 컴퓨터 소프트웨어로서 구현될 수 있고 하나 이상의 컴퓨터 판독 가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 19는 개시된 주제의 특정 실시예를 구현하기에 적합한 컴퓨터 시스템(2600)을 도시한다.

[0164] 컴퓨터 소프트웨어는 하나 이상의 컴퓨터 중앙 처리 장치(central processing unit, CPU), 그래픽 처리 장치(GPU) 등에 의한 어셈블리, 컴파일, 링크 또는 유사한 메커니즘을 통해 직접 실행될 수 있는 명령을 포함하는 코드를 생성할 수 있는 임의의 적절한 기계 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.

[0165] 명령은 예를 들어 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게임 장치, 사물인터넷 장치 등을 포함하는 다양한 유형의 컴퓨터 또는 그 컴포넌트에서 실행될 수 있다.

[0166] 컴퓨터 시스템(2600)에 대해 도 19에 도시된 컴포넌트는 본질적으로 예시적이며 본 개시의 실시예를 구현하는 컴퓨터 소프트웨어의 사용 또는 기능의 범위에 대한 어떠한 제한도 제안하려는 것이 아니다. 컴포넌트의 구성은 컴퓨터 시스템(2600)의 예시적인 실시예에 도시된 컴포넌트의 임의의 하나 또는 조합과 관련된 임의의 종속성 또는 요구사항을 갖는 것으로 해석되어서는 안된다.

[0167] 컴퓨터 시스템(2600)은 특정 휴면 인터페이스 입력 장치를 포함할 수 있다. 이러한 휴면 인터페이스 입력 장치는 예를 들어, 촉각 입력(예: 키 입력, 스와이프, 데이터 장갑 모션), 오디오 입력(예: 음성, 박수), 시각적 입력(예: 제스처), 후각 입력(도시되지 않음)을 통해 한 명 이상의 인간 사용자에 의한 입력에 응답할 수 있다. 휴면 인터페이스 장치는 또한 오디오(예: 음성, 음악, 주변 소리), 이미지(예: 스캔된 이미지, 정지 이미지 카메라로부터 획득하는 광학 이미지), 비디오(예: 2차원 비디오, 입체 비디오를 포함한 3차원 비디오)와 같이 인간에 의한 의식적 입력과 직접 관련이 없는 특정 매체를 캡처하는 데 사용될 수 있다.

[0168] 입력 휴면 인터페이스 장치는 키보드(2601), 마우스(2602), 트랙 패드(2603), 터치 스크린(2610), 데이터 글로브(도시되지 않음), 조이스틱(2605), 마이크(2606), 스캐너(2607) 및 카메라(2608) 중 하나 이상(각각 도시된 것 중 하나만)을 포함할 수 있다.

[0169] 컴퓨터 시스템(2600)은 또한 특정 휴면 인터페이스 출력 장치를 포함할 수 있다. 이러한 후면 인터페이스 출력 장치는 예를 들어 촉각 출력, 소리, 빛 및 냄새/맛을 통해 한 명 이상의 인간 사용자의 감각을 자극할 수 있다. 이러한 휴면 인터페이스 출력 장치는, 촉각 출력 장치(예를 들어, 터치 스크린(2610), 데이터 글로브(도시되지 않음), 또는 조이스틱(2605)에 의한 촉각 피드백을 포함하지만, 입력 장치로서 기능하지 않는 촉각 피드백 장치 일 수도 있음), 오디오 출력 장치(예: 스피커(2609), 헤드폰(도시되지 않음)), 시각 출력 장치(예: CRT 스크린, LCD 스크린, 플라즈마 스크린, OLED 스크린, 터치 스크린 입력 능력을 갖거나 갖지 않는 각각, 촉각 피드백 능력을 갖거나 또는 갖지 않는 각각을 포함하는 스크린(2610)과 같음 – 그 중 일부는 입체 출력, 가상 현실 안경(도시되지 않음), 홀로그래픽 디스플레이 및 스모크 탱크(smoke tank, 도시되지 않음))와 같은 수단을 통해 2차원 시각적 출력 또는 3차원 이상의 출력을 출력할 수 있음) – 및 프린터(도시되지 않음)를 포함할 수 있다.

[0170] 컴퓨터 시스템(2600)은 또한 인간이 액세스 가능한 저장 장치 및 CD/DVD ROM/RW(2620)를 포함하는 광학 매체 또는 유사 매체(2621), 썬 드라이브(thumb-drive)(2622), 탈착식 하드 드라이브 또는 솔리드 스테이트 드라이브(2623), 테이프 및 플로피 디스크와 같은 레거시 자기 매체(도시되지 않음), 보안 동글과 같은 특수 ROM/ASIC/PLD 기반 장치(도시되지 않음) 등과 같은 이와 연관된 매체를 포함할 수 있다.

[0171] 통상의 기술자는 또한 현재 개시된 주제와 관련하여 사용되는 "컴퓨터 판독 가능 매체"라는 용어가 전송 매체, 반송파 또는 다른 일시적 신호를 포함하지 않는다는 것을 이해해야 한다.

[0172] 컴퓨터 시스템(2600)은 또한 하나 이상의 통신 네트워크(2655)에 대한 인터페이스(2654)를 포함할 수 있다. 네

트워크는 예를 들어 무선, 유선, 광일 수 있다. 네트워크는 또한 로컬, 광역, 대도시, 차량 및 산업, 실시간, 지연 허용 등일 수 있다. 네트워크의 예로는 이더넷과 같은 근거리 통신망, 무선 LAN, GSM, 3G, 4G, 5G, LTE 등을 포함하는 셀룰러 네트워크, 케이블 TV, 위성 TV 및 지상파 방송 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크, CANBus를 포함하는 차량 및 산업용 등을 포함한다. 특정 네트워크에는 일반적으로 특정 범용 데이터 포트 또는 주변 장치 버스(2649)(예를 들어, 컴퓨터 시스템(2600)의 USB 포트와 같음)에 부착된 외부 네트워크 인터페이스 어댑터가 필요하다. 다른 것들은 일반적으로 아래에 설명된 바와 같이 시스템 버스에 대한 부착에 의해 컴퓨터 시스템(2600)의 코어에 통합된다(예를 들어, PC 컴퓨터 시스템에 대한 이더넷 인터페이스 또는 스마트 폰 컴퓨터 시스템에 대한 셀룰러 네트워크 인터페이스). 이러한 네트워크 중 하나를 사용하여, 컴퓨터 시스템(2600)은 다른 엔티티와 통신할 수 있다. 이러한 통신은 단방향, 수신 전용(예를 들어, 방송 TV), 단방향 전송 전용(예를 들어, 특정 CANbus 장치에 대한 CANbus) 또는 양방향, 예를 들어, 로컬 또는 광역 디지털 네트워크를 사용하는 다른 컴퓨터 시스템일 수 있다. 특정 프로토콜 및 프로토콜 스택은 상기한 바와 같이 각각의 네트워크 및 네트워크 인터페이스에서 사용될 수 있다.

[0173] 전술한 휴면 인터페이스 장치, 인간 액세스 가능 저장 장치 및 네트워크 인터페이스는 컴퓨터 시스템(2600)의 코어(2640)에 부착될 수 있다.

[0174] 코어(2640)는 하나 이상의 중앙 처리 장치(CPU)(2641), 그래픽 처리 장치(Graphics Processing Unit, GPU)(2642), FPGA(Field Programmable Gate Area)(2643) 형태의 특수 프로그램 가능 처리 유닛, 특정 태스크에 대한 하드웨어 가속기(2644), 그래픽 어댑터(2650) 등을 포함할 수 있다. 읽기 전용 메모리(Read-only memory, ROM)(2645), 랜덤 액세스 메모리(Random-access memory, 2646), 내부 비 사용자 액세스 가능 하드 드라이브, SSD 및 유사체(2647)와 같은 내부 대용량 저장소와 함께 이러한 장치는 시스템 버스(2648)를 통해 연결될 수 있다. 일부 컴퓨터 시스템에서, 시스템 버스(2648)는 추가 CPU, GPU 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형태로 액세스될 수 있다. 주변 장치는 코어의 시스템 버스(2648)에 직접 부착되거나, 또는 주변 장치 버스(2649)를 통해 부착될 수 있다. 예에서, 스크린(2610)은 그래픽 어댑터(2650)에 연결될 수 있다. 주변 장치 버스의 아키텍처에는 PCI, USB 등이 포함된다.

[0175] CPU(2641), GPU(2642), FPGA(2643) 및 가속기(2644)는 조합하여 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령을 실행할 수 있다. 이 컴퓨터 코드는 ROM(2645) 또는 RAM(2646)에 저장될 수 있다. 과도기 데이터는 RAM(2646)에 저장될 수도 있지만, 영구 데이터는 예를 들어 내부 대용량 저장소(2647)에 저장될 수 있다. 하나 이상의 CPU(2641), GPU(2642), 대용량 저장소(2647), ROM(2645), RAM(2646) 등과 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 모든 메모리 장치에 대한 빠른 저장 및 검색이 가능해질 수 있다.

[0176] 컴퓨터 판독 가능 매체는 다양한 컴퓨터 구현 작동을 수행하기 위한 컴퓨터 코드를 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시의 목적을 위해 특별히 설계되고 구성된 것일 수 있거나, 또는 컴퓨터 소프트웨어 분야의 당업자에게 잘 알려져 있고 이용 가능한 종류일 수 있다.

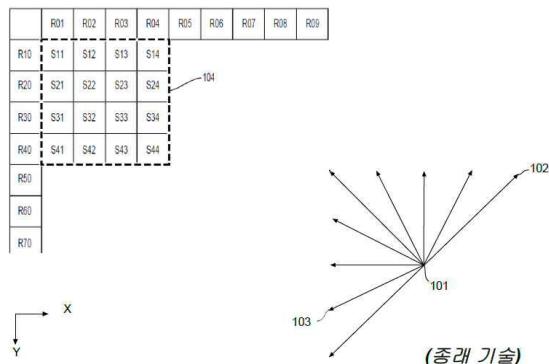
[0177] 비제한적인 예로서, 아키텍처를 갖는 컴퓨터 시스템(2600), 특히 코어(2640)는 하나 이상의 유형의 컴퓨터 판독 가능 매체에 구현된 소프트웨어를 실행하는 프로세서(들)(CPU, GPU, FPGA, 가속기 등을 포함함)의 결과로서 기능을 제공할 수 있다. 이러한 컴퓨터 판독 가능 매체는 위에서 소개한 바와 같이 사용자 액세스 가능 대용량 저장소와 연관된 매체일 수 있으며, 코어 내부 대용량 저장소(2647) 또는 ROM(2645)과 같은 비 일시적 특성을 가진 코어(2640)의 특정 저장소일 수 있다. 본 개시의 다양한 실시예를 구현하는 소프트웨어는 이러한 장치에 저장되고 코어(2640)에 의해 실행될 수 있다. 컴퓨터 판독 가능 매체는 특정 필요에 따라 하나 이상의 메모리 장치 또는 칩을 포함할 수 있다. 소프트웨어는 코어(2640) 및 특히 그 안의 프로세서(CPU, GPU, FPGA 등을 포함함)가 RAM(2646)에 저장된 데이터 구조를 정의하는 것과 소프트웨어에서 정의된 프로세스에 따라 이러한 데이터 구조를 수정하는 것을 포함하여 여기에서 설명된 특정 프로세스 또는 특정 프로세스의 특정 부분을 실행하도록 할 수 있다. 추가로 또는 다르게는, 컴퓨터 시스템은 여기에서 설명된 특정 프로세스나 특정 프로세스의 특정 부분을 실행하기 위해 소프트웨어 대신 또는 소프트웨어와 함께 작동할 수 있는 회로(예를 들어, 가속기(2644)에 고정되거나 다른 방식으로 구현된 로직의 결과로서 기능을 제공할 수 있다. 소프트웨어에 대한 참조는 로직을 포함할 수 있으며, 적절한 경우에 그 반대도 마찬가지이다. 컴퓨터 판독 가능 매체에 대한 참조는 실행을 위한 소프트웨어를 저장하는 회로(집적 회로(integrated circuit, IC)와 같음), 실행을 위한 로직을 구현하는 회로, 또는 적절한 경우 둘 다를 포함할 수 있다. 본 개시는 하드웨어 및 소프트웨어의 임의의 적절한 조합을 포함한다.

[0178] 특정 발명이 예시적인 실시예를 참조하여 설명되었지만, 이러한 설명은 제한을 의미하지 않는다. 본 발명의 예

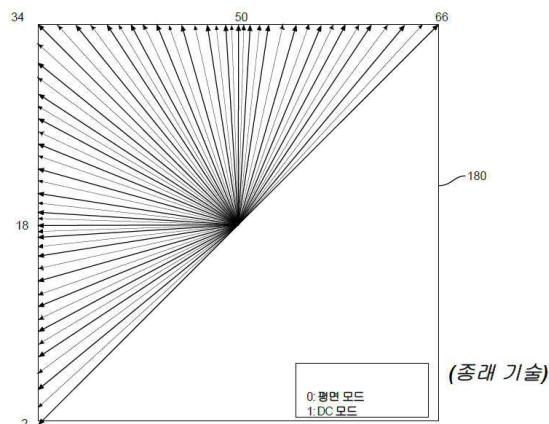
시적인 실시예 및 추가적인 실시예의 다양한 수정은 이러한 설명으로부터 당업자에게 명백할 것이다. 당업자는 본 발명의 사상 및 범위를 벗어남이 없이 본 명세서에 예시되고 설명된 예시적인 실시예에 대해 이들 및 다양한 다른 수정이 이루어질 수 있음을 쉽게 인식할 것이다. 따라서, 첨부된 청구범위는 그러한 수정 및 대안적인 실시예를 포함할 것으로 생각된다. 도면 내의 특정 비율은 과장될 수 있으며 다른 비율은 최소화될 수 있다. 따라서, 본 개시 및 도면은 제한적이기보다는 예시적인 것으로 간주되어야 한다.

도면

도면 1a



도면 1b

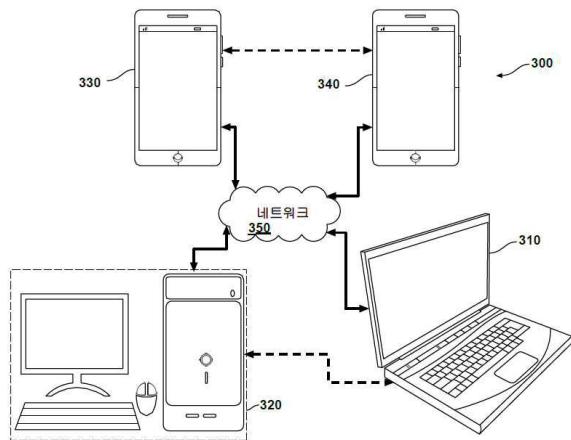


도면2

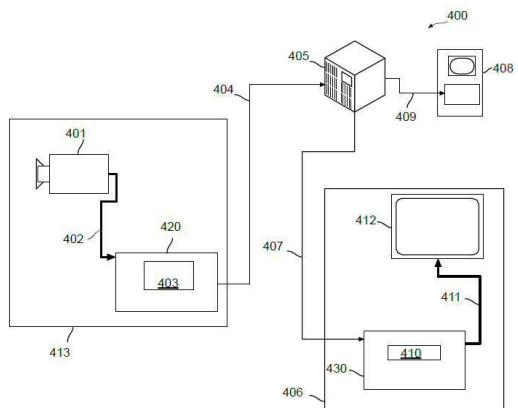


(종래 기술)

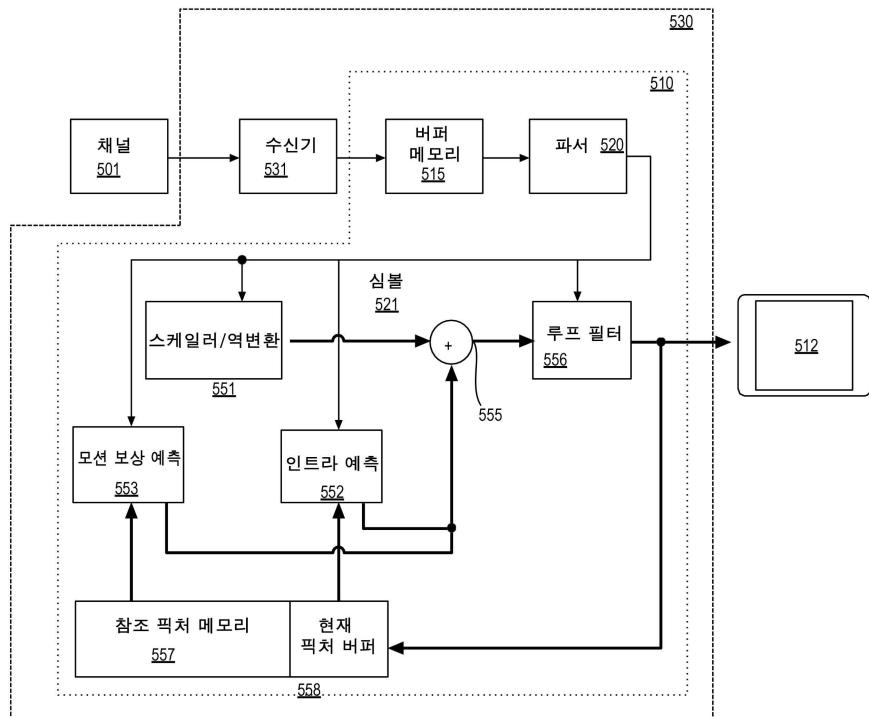
도면3



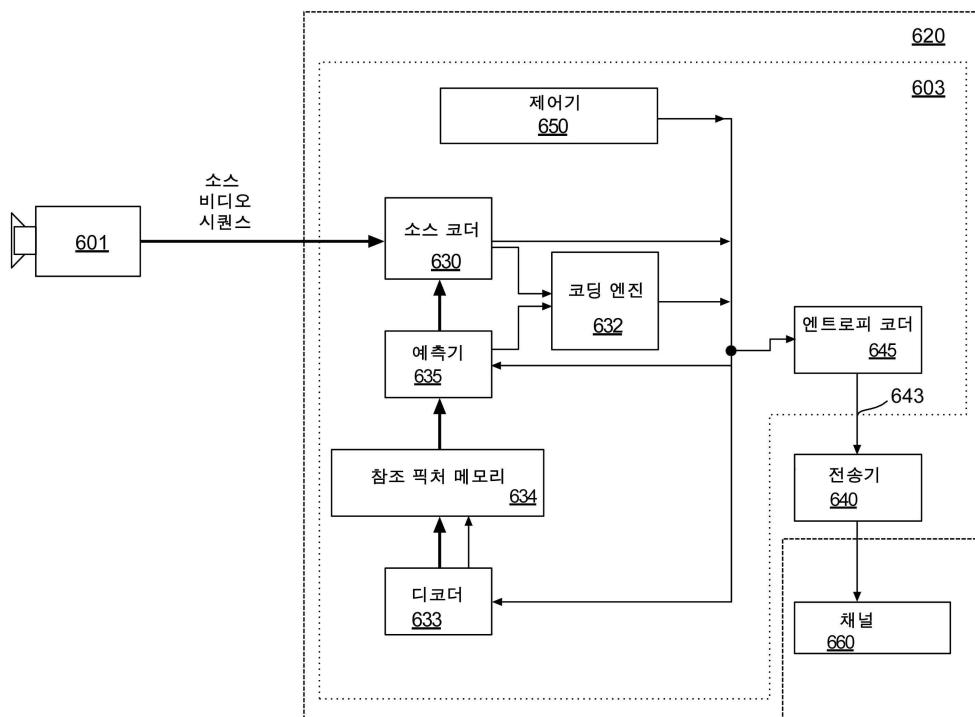
도면4



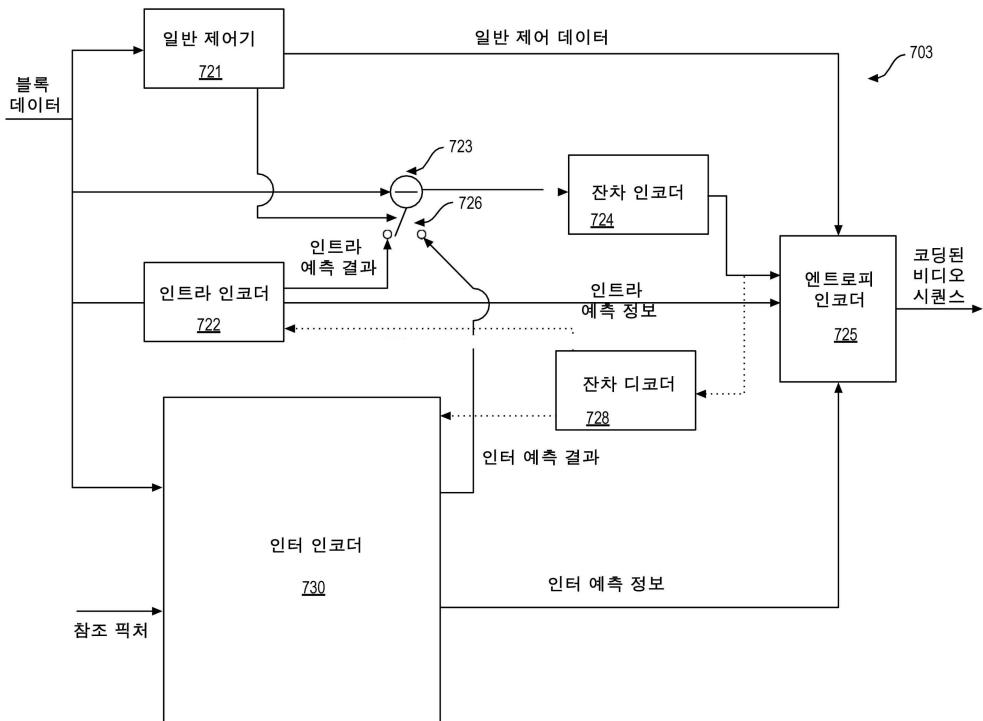
도면5



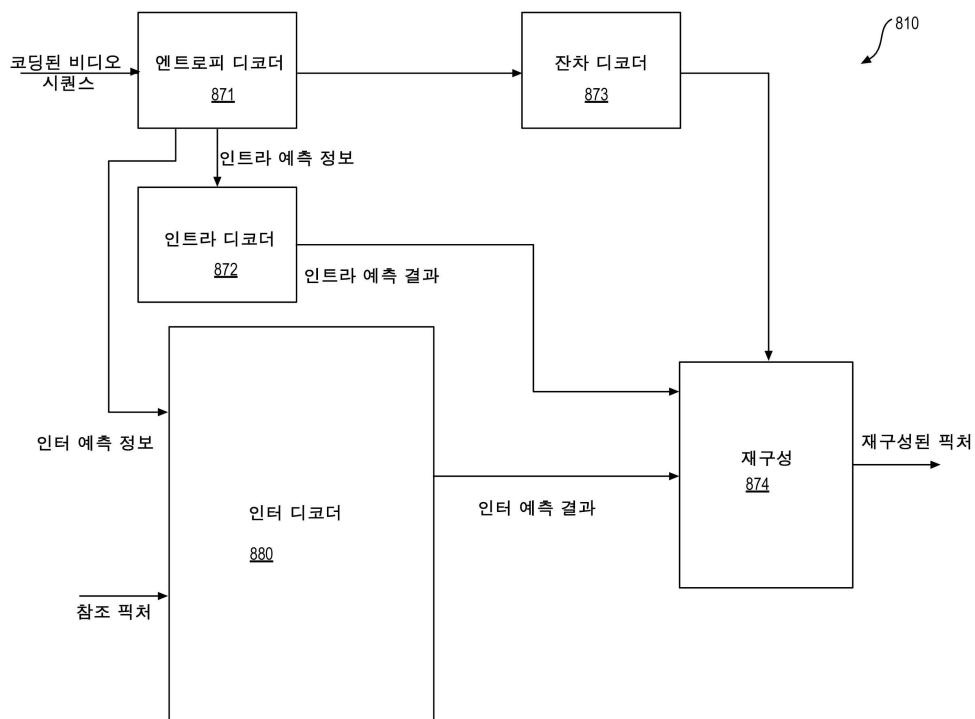
도면6

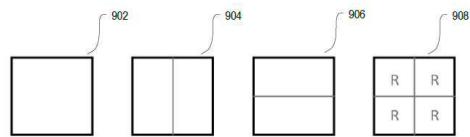
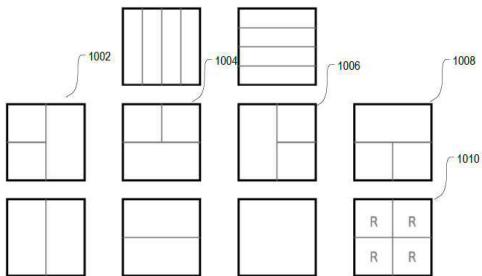
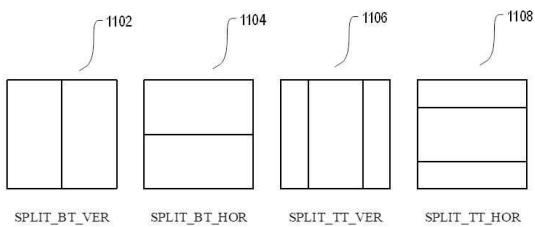
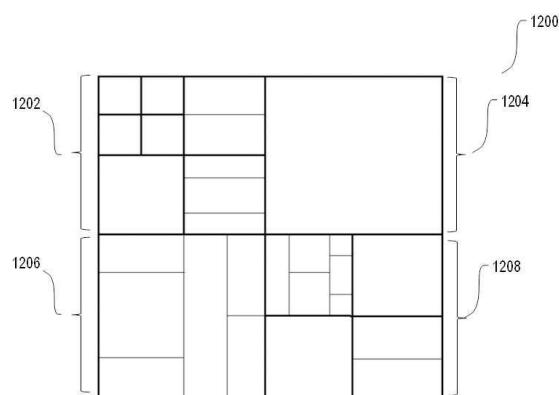


도면7

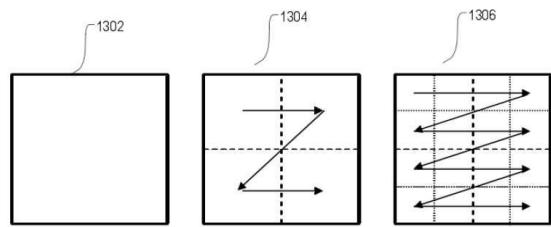


도면8

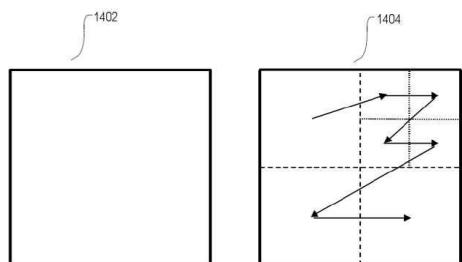


도면9**도면10****도면11****도면12**

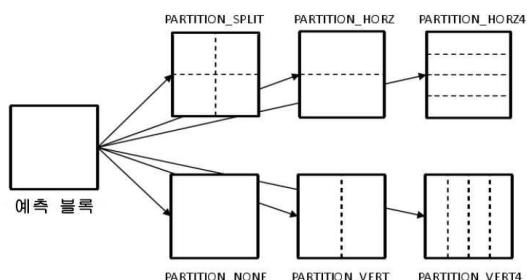
도면13



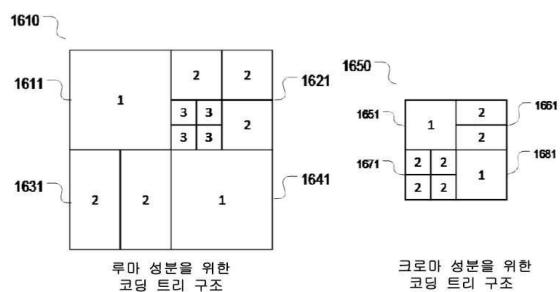
도면14



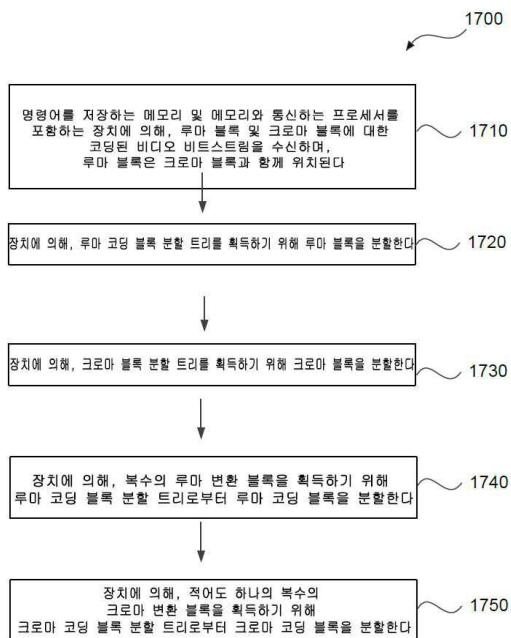
도면15



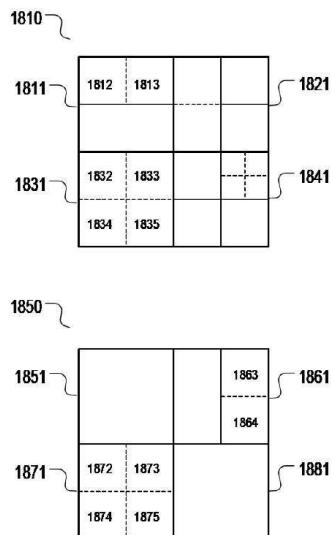
도면16



도면17



도면18



도면19

