(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0002534
(43) 공개일자 2023년01월05일

(51) 국제특허분류(Int. Cl.)
H04N 19/109 (2014.01) *H04N 19/105* (2014.01)
H04N 19/119 (2014.01) *H04N 19/159* (2014.01)
H04N 19/176 (2014.01) *H04N 19/60* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/109 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2022-7037873
(22) 출원일자(국제) 2021년04월13일
심사청구일자 없음
(85) 번역문제출일자 2022년10월28일
(86) 국제출원번호 PCT/US2021/027072
(87) 국제공개번호 WO 2021/211576
국제공개일자 2021년10월21일
(30) 우선권주장
63/009,370 2020년04월13일 미국(US)
17/229,210 2021년04월13일 미국(US)

(71) 출원인
오피 솔루션즈, 엘엘씨
미국 01002 매사추세츠주 암허스트 미들 스트리트 368
(72) 발명자
칼바, 하리
미국 33431 플로리다 보카 라톤 글래이즈 로드 777
푸르트, 보리보예
미국 33431 플로리다 보카 라톤 글래이즈 로드 777
애드직, 벨리보르
미국 01002 매사추세츠 애머스트 미들 스트리트 368
(74) 대리인
특허법인 남앤남

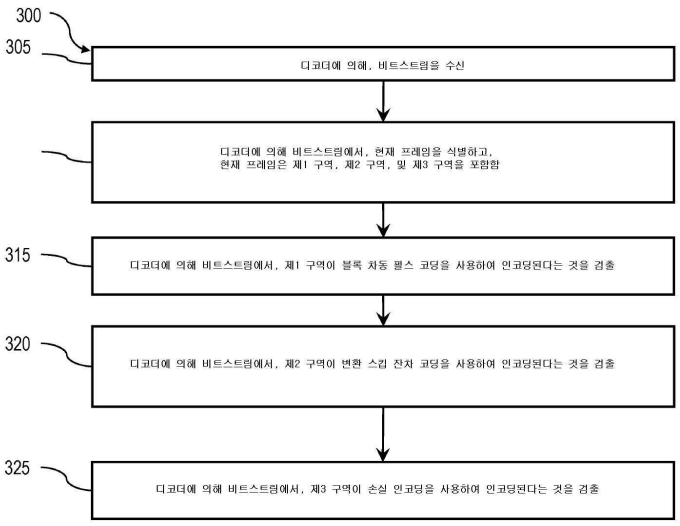
전체 청구항 수 : 총 48 항

(54) 발명의 명칭 **결합된 무손실 및 손실 코딩을 위한 방법들 및 시스템들**

(57) 요 약

디코더는 비트스트림을 수신하고, 비트스트림에서 현재 프레임을 식별하고 – 현재 프레임은 제1 구역 및 제3 구역을 포함함 –, 비트스트림에서, 제1 구역이 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하고, 현재 프레임을 디코딩 – 현재 프레임을 디코딩하는 것은 무손실 인코딩 프로토콜에 대응하는 무손실 디코딩 프로토콜을 사용하여 제1 구역을 디코딩하는 것을 더 포함함 –하도록 구성된 회로를 포함한다.

대 표 도 - 도3



(52) CPC특허분류
H04N 19/119 (2015.01)
H04N 19/159 (2015.01)
H04N 19/176 (2015.01)
H04N 19/60 (2015.01)
H04N 19/70 (2015.01)

명 세 서

청구범위

청구항 1

디코더로서,

코딩된 비디오 비트스트림(bitstream)을 수신하고;

상기 비트스트림에서, 현재 프레임을 식별하고 - 상기 현재 프레임은 제1 구역, 제2 구역 및 제3 구역을 포함함 -,

상기 비트스트림에서, 상기 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩되는 것을 검출하고,

상기 비트스트림에서, 상기 제2 구역이 변환 스킵 잔차 코딩을 사용하여 인코딩되는 것을 검출하고, 그리고

상기 비트스트림에서, 상기 제3 구역이 손실 인코딩을 사용하여 인코딩되는 것을 검출하도록 구성된 회로를 포함하며,

상기 손실 인코딩은 인터-예측 및 인트라-예측 중 적어도 하나를 포함하는, 디코더.

청구항 2

제1 항에 있어서,

상기 비트스트림은 상기 제1 구역에 대응하는 서브-픽처 헤더(sub-picture header)를 더 포함하고;

상기 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩되는 것을 검출하는 것은 적어도 제1 구역이 상기 서브-픽처 헤더에서 블록 차동 펄스 코드 변조를 사용하여 인코딩된다는 표시를 검출하는 것을 더 포함하는, 디코더.

청구항 3

제2 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 디코더.

청구항 4

제2 항에 있어서,

상기 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함되는, 디코더.

청구항 5

제1 항에 있어서,

상기 비트스트림은 상기 제2 구역에 대응하는 서브-픽처 헤더를 더 포함하고;

상기 디코더는 제2 사용 변환 스킵 잔차 코딩이 상기 서브-픽처 헤더에서 변환 스킵 잔차 코딩을 사용하여 인코딩된다는 표시를 검출하도록 추가로 구성되는, 디코더.

청구항 6

제5 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 디코더.

청구항 7

제1 항에 있어서,

상기 디코더는,

제1 프로세서 스레드(thread)를 사용하여 상기 제1 구역을 디코딩하고; 그리고

제2 프로세서 스레드를 사용하여 상기 제2 구역 요소를 디코딩하도록 추가로 구성되는, 디코더.

청구항 8

제1 항에 있어서,

상기 현재 프레임을 디코딩하도록 추가로 구성되고, 상기 현재 프레임을 디코딩하는 것은 블록 차동 펄스 코드 변조에 대응하는 디코딩 프로토콜(decoding protocol)을 사용하여 상기 제1 구역을 디코딩하는 것을 더 포함하는, 디코더.

청구항 9

제1 항에 있어서,

상기 현재 프레임을 디코딩하도록 추가로 구성되고, 상기 현재 프레임을 디코딩하는 것은 변환 스킵 잔차 코딩을 사용하는 것에 대응하는 디코딩 프로토콜을 사용하여 상기 제2 구역을 디코딩하는 것을 더 포함하는, 디코더.

청구항 10

제1 항에 있어서,

상기 현재 프레임을 디코딩하도록 추가로 구성되고, 상기 현재 프레임을 디코딩하는 것은 상기 손실 인코딩 프로토콜에 대응하는 디코딩 프로토콜을 사용하여 상기 제3 구역을 디코딩하는 것을 더 포함하는, 디코더.

청구항 11

결합된 무손실 및 손실 코딩의 방법으로서,

디코더에 의해, 코딩된 비디오 비트스트림을 수신하는 단계;

상기 디코더에 의해 상기 비트스트림에서, 현재 프레임을 식별하는 단계 - 상기 현재 프레임은 제1 구역, 제2 구역, 및 제3 구역을 포함함 -;

상기 디코더에 의해 상기 비트스트림에서, 상기 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩되는 것을 검출하는 단계;

상기 디코더에 의해 상기 비트스트림에서, 상기 제2 구역이 변환 스킵 잔차 코딩을 사용하여 인코딩되는 것을 검출하는 단계; 및

상기 디코더에 의해 상기 비트스트림에서, 상기 제3 구역이 손실 인코딩을 사용하여 인코딩되는 것을 검출하는 단계 - 상기 손실 인코딩은 인터-예측 및 인트라-예측 중 적어도 하나를 포함함 -를 포함하는, 방법.

청구항 12

제11 항에 있어서,

상기 비트스트림은 상기 제1 구역에 대응하는 서브-픽처 헤더를 더 포함하고;

상기 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩되는 것을 검출하는 단계는 적어도 상기 제1 구역이 상기 서브-픽처 헤더에서 블록 차동 펄스 코드 변조를 사용하여 인코딩된다는 표시를 검출하는 단계를 더 포함하는, 방법.

청구항 13

제12 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 방법.

청구항 14

제12 항에 있어서,

상기 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함되는, 방법.

청구항 15

제11 항에 있어서,

상기 비트스트림은 상기 제2 구역에 대응하는 서브-픽처 헤더를 더 포함하고;

상기 디코더는 제2 사용 변환 스킵 잔차 코딩이 상기 서브-픽처 헤더에서 변환 스킵 잔차 코딩을 사용하여 인코딩된다는 표시를 검출하도록 추가로 구성되는, 방법.

청구항 16

제16 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 방법.

청구항 17

제11 항에 있어서,

제1 프로세서 스레드를 사용하여 상기 제1 구역을 디코딩하는 단계; 및

제2 프로세서 스레드를 사용하여 상기 제3 구역 요소를 디코딩하는 단계를 더 포함하는, 방법.

청구항 18

제11 항에 있어서,

상기 현재 프레임을 디코딩하는 단계를 더 포함하고, 상기 현재 프레임을 디코딩하는 단계는 블록 차동 펄스 코드 변조에 대응하는 디코딩 프로토콜을 사용하여 상기 제1 구역을 디코딩하는 단계를 더 포함하는, 방법.

청구항 19

제11 항에 있어서,

상기 현재 프레임을 디코딩하는 단계를 더 포함하고, 상기 현재 프레임을 디코딩하는 단계는 변환 스킵 잔차 코딩을 사용하는 것에 대응하는 디코딩 프로토콜을 사용하여 상기 제2 구역을 디코딩하는 단계를 더 포함하는, 방법.

청구항 20

제11 항에 있어서,

상기 현재 프레임을 디코딩하는 단계를 더 포함하고, 상기 현재 프레임을 디코딩하는 단계는 상기 손실 인코딩 프로토콜에 대응하는 디코딩 프로토콜을 사용하여 상기 제3 구역을 디코딩하는 단계를 더 포함하는, 방법.

청구항 21

디코더로서,

코딩된 비트스트림을 수신하고;

상기 비트스트림에서, 현재 프레임을 식별하고 – 상기 현재 프레임은 제1 서브-픽처 및 제2 서브-픽처를 포함함 –,

상기 비트스트림에서, 상기 제1 서브-픽처가 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하고;

상기 현재 프레임을 디코딩 – 상기 현재 프레임을 디코딩하는 것은 상기 무손실 인코딩 프로토콜에 대응하는 무손실 디코딩 프로토콜을 사용하여 상기 제1 서브-픽처를 디코딩함 –하도록 구성된 회로를 포함하는, 디코더.

청구항 22

제21 항에 있어서,

상기 비트스트림은 상기 제1 서브-픽처에 대응하는 서브-픽처 헤더를 더 포함하고;

검출하는 것은 적어도 상기 제1 서브-픽처가 상기 서브-픽처 헤더에서 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 것을 더 포함하는,

디코더.

청구항 23

제22 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 디코더.

청구항 24

제22 항에 있어서,

상기 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함되는, 디코더.

청구항 25

제21 항에 있어서,

상기 디코더는:

상기 제2 서브-픽처가 손실 인코딩 프로토콜에 따라 인코딩된다는 것을 검출하고;

상기 무손실 인코딩 프로토콜에 대응하는 손실 디코딩 프로토콜에 따라 상기 제2 서브-픽처를 디코딩하도록 추가로 구성되는, 디코더.

청구항 26

제25 항에 있어서,

상기 비트스트림은 상기 제2 서브-픽처에 대응하는 서브-픽처 헤더를 더 포함하고;

상기 디코더는 상기 제2 서브-픽처가 상기 서브-픽처 헤더에서 손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하도록 추가로 구성되는, 디코더.

청구항 27

제26 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 디코더.

청구항 28

제26 항에 있어서,

상기 서브-픽처 헤더는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함되는, 디코더.

청구항 29

제21 항에 있어서,

상기 디코더는

제1 프로세서 스레드를 사용하여 상기 제1 서브-픽처를 디코딩하고;

제2 프로세서 스레드를 사용하여 상기 제2 서브-픽처 요소를 디코딩하도록 추가로 구성되는, 디코더.

청구항 30

제21 항에 있어서,

상기 비트스트림을 수신하고 상기 비트스트림을 양자화된 계수들로 디코딩하도록 구성된 엔트로피 디코더 프로세서;

역 이산 코사인을 수행하는 것을 포함하여 상기 양자화된 계수들을 프로세싱하도록 구성된 역양자화 및 역변환 프로세서;

디블로킹 필터(deblocking filter);

프레임 버퍼; 및

인트라 예측 프로세서를 더 포함하는, 디코더.

청구항 31

제21 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 쿼드트리(quadtree) 플러스 이진 결정 트리의 일부를 형성하는, 디코더.

청구항 32

제21 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 코딩 트리 단위를 포함하는, 디코더.

청구항 33

제21 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 코딩 단위를 포함하는, 디코더.

청구항 34

제21 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 예측 단위를 포함하는, 디코더.

청구항 35

결합된 무손실 및 손실 코딩의 방법으로서,

디코더에서, 비트스트림을 수신하는 단계;

상기 디코더에 의해 상기 비트스트림에서, 현재 프레임을 식별하는 단계 - 상기 현재 프레임은 제1 서브-픽처 및 제2 서브-픽처를 포함함 -,

상기 디코더에 의해 상기 비트스트림에서, 상기 제1 서브-픽처가 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 단계; 및

상기 디코더에 의해, 상기 현재 프레임을 디코딩하는 단계 - 상기 현재 프레임을 디코딩하는 단계는 상기 무손실 인코딩 프로토콜에 대응하는 무손실 디코딩 프로토콜을 사용하여 상기 제1 서브-픽처를 디코딩하는 단계를 더 포함함 -를 포함하는, 방법.

청구항 36

제35 항에 있어서,

상기 비트스트림은 상기 제1 서브-픽처에 대응하는 서브-픽처 헤더를 더 포함하고;

검출하는 단계는 적어도 상기 제1 서브-픽처가 상기 서브-픽처 헤더에서 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 단계를 더 포함하는, 방법.

청구항 37

제36 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 방법.

청구항 38

제36 항에 있어서,

상기 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함되는, 방법.

청구항 39

제39 항에 있어서,

상기 제2 서브-픽처가 손실 인코딩 프로토콜에 따라 인코딩된다는 것을 검출하는 단계; 및

상기 무손실 인코딩 프로토콜에 대응하는 손실 디코딩 프로토콜에 따라 상기 제2 서브-픽처를 디코딩하는 단계를 더 포함하는, 방법.

청구항 40

제39 항에 있어서,

상기 비트스트림은 상기 제2 서브-픽처에 대응하는 서브-픽처 헤더를 더 포함하고;

검출은 상기 제2 서브-픽처가 상기 서브-픽처 헤더에서 손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 더 포함하는, 방법.

청구항 41

제40 항에 있어서,

상기 서브-픽처 헤더는 상기 현재 프레임에 대응하는 데이터에 명시적으로 포함되는, 방법.

청구항 42

제40 항에 있어서,

상기 서브-픽처 헤더는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함되는, 방법.

청구항 43

제35 항에 있어서,

상기 디코더는

제1 프로세서 스레드를 사용하여 상기 제1 서브-픽처를 디코딩하고;

제2 프로세서 스레드를 사용하여 상기 제2 서브-픽처 요소를 디코딩하도록 추가로 구성되는, 방법.

청구항 44

제35 항에 있어서,

상기 디코더는:

상기 비트스트림을 수신하고 상기 비트스트림을 양자화된 계수들로 디코딩하도록 구성된 엔트로피 디코더 프로세서;

역 이산 코사인을 수행하는 것을 포함하여 상기 양자화된 계수들을 프로세싱하도록 구성된 역양자화 및 역변환 프로세서;

디블로킹 필터;

프레임 버퍼; 및

인트라 예측 프로세서를 더 포함하는, 방법.

청구항 45

제35 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 쿼드트리 플러스 이진 결정 트리의 일부를 형성하는, 방법.

청구항 46

제35 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 코딩 트리 단위를 포함하는, 방법.

청구항 47

제35 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 코딩 단위를 포함하는, 방법.

청구항 48

제35 항에 있어서,

상기 제1 서브-픽처 및 상기 제2 서브-픽처 중 적어도 하나는 예측 단위를 포함하는, 방법.

발명의 설명

기 술 분 야

[0001] 본 출원은 그 전체가 본원에 참조로 포함되는 2021년 4월 13일에 출원되고 발명의 명칭이 "METHODS AND SYSTEMS FOR COMBINED LOSSLESS AND LOSSY CODING"인 미국 비가출원 일련 번호 제17/229,210호의 우선권의 이익을 주장하고, 그 전체가 본원에 참조로 포함되는 2020년 4월 13일에 출원되고 발명의 명칭이 "METHODS AND SYSTEMS FOR COMBINED LOSSLESS AND LOSSY CODING"인 미국 가특허출원 일련번호 제63/009,370호의 우선권의 이익을 주장한다.

[0002] 본 발명은 일반적으로 비디오 압축 분야에 관한 것이다. 특히, 본 발명은 결합된 무손실 및 손실 코딩을 위한 방법들 및 시스템들에 관한 것이다.

배 경 기 술

[0003] 비디오 코덱(video codec)은 디지털 비디오를 압축하거나 압축해제하는 전자 회로 또는 소프트웨어를 포함할 수 있다. 압축되지 않은 비디오를 압축 포맷으로 또는 그 반대로 변환할 수 있다. 비디오 압축의 맥락에서, 비디오를 압축하는(및/또는 그 일부 기능을 수행하는) 디바이스는 일반적으로 인코더라고 칭해질 수 있고, 비디오를 압축해제하는(및/또는 그 일부 기능을 수행하는) 디바이스는 디코더라고 칭해질 수 있다.

[0004] 압축된 데이터의 포맷은 표준 비디오 압축 사양을 따를 수 있다. 압축된 비디오가 원본 비디오에 존재하는 일부 정보가 결여된다는 점에서 압축은 손실일 수 있다. 그 결과 원본 비디오를 정확하게 재구성하기 위한 정보가 충분하지 않기 때문에 압축해제된 비디오가 원래의 압축되지 않은 비디오보다 품질이 낮을 수 있다는 것이 포함될 수 있다.

[0005] 비디오 품질, 비디오를 나타내는 데 사용되는 데이터의 양(예를 들어, 비트 레이트에 의해 결정됨), 인코딩 및 디코딩 알고리즘의 복잡성, 데이터 손실들 및 오류들에 대한 민감도, 편집 용이성, 랜덤 액세스, 종단간 지연(예를 들어, 레이턴시) 등 사이에 복잡한 관계들이 있을 수 있다.

[0006] 모션 보상은 비디오에서 카메라 및/또는 객체들의 모션을 고려하여 이전 및/또는 미래 프레임들과 같은 참조 프레임이 주어진 비디오 프레임 또는 그 일부를 예측하는 접근법을 포함할 수 있다. 이는 예컨대, MPEG(Motion Picture Experts Group)의 AVC(advanced video coding) 표준(또한 H.264라고 지칭됨)을 사용한 인코딩 및 디코딩과 같이 비디오 압축을 위한 비디오 데이터의 인코딩 및 디코딩에 사용될 수 있다. 모션 보상은 참조 픽처(picture)를 현재 픽처로 변환하는 관점에서 픽처를 설명할 수 있다. 참조 픽처는 현재 픽처와 비교하면 시간

상 이전일 수 있고, 현재 픽처와 비교하면 미래에서 올 수 있다. 이전에 송신 및/또는 저장된 이미지들로부터 이미지들을 정확하게 합성할 수 있다면, 압축 효율이 개선될 수 있다.

발명의 내용

[0007] 일 양태에서, 디코더는 코딩된 비디오 비트스트림을 수신하고, 비트스트림에서, 현재 프레임을 식별하고 – 현재 프레임은 제1 구역, 제2 구역 및 제3 구역을 포함함 –, 비트스트림에서, 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩된 제1 구역을 검출하고, 비트스트림에서, 제2 구역이 변환 스킵 잔차 코딩을 사용하여 인코딩되는 것을 검출하고, 비트스트림에서, 제3 구역이 손실 인코딩을 사용하여 인코딩되는 것을 검출 – 손실 인코딩은 인터-예측 및 인트라-예측 중 적어도 하나를 포함함 –하도록 구성된 회로를 포함한다.

[0008] 다른 양태에서, 결합된 무손실 및 손실 코딩의 방법은 디코더에 의해, 코딩된 비디오 비트스트림을 수신하는 단계, 디코더에 의해 비트스트림에서, 현재 프레임을 식별하는 단계 – 현재 프레임은 제1 구역, 제2 구역, 및 제3 구역을 포함함 –, 디코더에 의해 비트스트림에서, 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩되는 것을 검출하는 단계, 디코더에 의해 비트스트림에서, 제2 구역이 변환 스킵 잔차 코딩을 사용하여 인코딩되는 것을 검출하는 단계, 및 디코더에 의해 비트스트림에서, 제3 구역이 손실 인코딩을 사용하여 인코딩되는 것을 검출하는 단계 – 손실 인코딩은 인터-예측 및 인트라-예측 중 적어도 하나를 포함함 –를 포함한다.

[0009] 다른 양태에서, 디코더는 코딩된 비트스트림을 수신하고, 비트스트림에서, 현재 프레임을 식별하고 – 현재 프레임은 제1 서브-픽처 및 제2 서브-픽처를 포함함 –, 비트스트림에서 제1 서브-픽처가 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하고, 현재 프레임을 디코딩 – 현재 프레임을 디코딩하는 것은 무손실 인코딩 프로토콜에 대응하는 무손실 디코딩 프로토콜을 사용하여 제1 서브-픽처를 디코딩하는 것을 더 포함함 –하도록 구성된 회로를 포함한다.

[0010] 다른 양태에서, 결합된 무손실 및 손실 코딩의 방법은 디코더에서, 코딩된 비트스트림을 수신하는 단계, 디코더에 의해 비트스트림에서, 현재 프레임을 식별하는 단계 – 현재 프레임은 제1 서브-픽처 및 제2 서브-픽처를 포함함 –, 디코더에 의해 비트스트림에서, 제1 서브-픽처가 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 단계, 및 디코더에 의해, 현재 프레임을 디코딩하는 단계 – 현재 프레임을 디코딩하는 단계는 무손실 인코딩 프로토콜에 대응하는 무손실 디코딩 프로토콜을 사용하여 제1 서브-픽처를 디코딩하는 단계를 더 포함함 –를 포함한다.

[0011] 본 발명의 비제한적 실시예들의 이들 및 다른 양태들 및 특징들은 첨부 도면들과 함께 본 발명의 특정 비제한적 실시예들의 하기 설명을 검토할 때 통상의 기술자들에게 명백해질 것이다.

도면의 간단한 설명

[0012] 본 발명을 예시할 목적으로, 도면들은 본 발명의 하나 이상의 실시예들의 양태들을 도시한다. 그러나, 본 발명이 도면들에 도시된 정확한 배열들 및 수단으로 제한되지 않는다는 것이 이해되어야 한다.

도 1은 복수의 서브-픽처들을 갖는 프레임의 예시적인 실시예를 예시하는 블록도이다.

도 2는 2 개의 서브-픽처들을 갖는 프레임의 예시적인 실시예의 예시이다.

도 3은 현재 주제의 일부 구현들에 따라 비디오를 디코딩하기 위한 예시적인 프로세스를 예시하는 프로세스 흐름도이다.

도 4는 현재 주제의 일부 구현들에 따라 비트 스트림을 디코딩할 수 있는 예시적인 디코더를 예시하는 시스템 블록도이다.

도 5는 현재 주제의 일부 구현들에 따라 비디오를 인코딩하는 예시적인 프로세스를 예시하는 프로세스 흐름도이다.

도 6은 본원에 개시된 방법론들 중 임의의 하나 이상 및 이의 임의의 하나 이상의 부분을 구현하는 데 사용될 수 있는 컴퓨팅 시스템의 블록도이다.

도면들은 반드시 축척에 맞춰진 것은 아니며 가상 라인들, 도식적 표현들 및 단편적인 뷰들에 예시될 수 있다. 소정 사례들에서, 실시예들의 이해를 위해 필요하지 않거나 다른 세부사항들을 표현하기 어렵게 하는 세부사항들은 생략될 수 있다.

발명을 실시하기 위한 구체적인 내용

[0013] 기존의 비디오 코딩 시스템들에서, 비디오 시퀀스는 GOP(Groups-of-pictures)로 나뉜다. 각 GOP는 시간 및 공간 예측의 의미에서 자체-포함된다. 일반적으로, 그룹의 제1 픽처는 후속 픽처들의 참조 픽처로 사용된다. 픽처들 간의 시간적 및 공간적 관계들은 예측 코딩을 사용하여 매우 효율적인 압축을 허용한다.

[0014] 과거의 코딩 시스템들은 일반적으로 인코딩된 프레임의 일부 정보가 인코딩 프로세스 동안 생략되고 디코딩 동안 복구되지 않는 손실 코딩을 사용하여 동작했다. 이러한 손실 프로세스들은 예를 들어 제한 없이 인코더에서 디코더로 비트 스트림으로 송신되는 데이터의 양을 감소시킴으로써 더 높은 정도의 효율성을 달성하기 위해 디코딩된 프레임들 및/또는 비디오 픽처들에서 소정 정도의 세부사항 및/또는 해상도, 프레임 또는 픽처 그룹을 인코딩 및/또는 디코딩하는 데 사용되는 프로세싱 시간 및/또는 메모리 리소스들 등을 희생시킬 수 있다.

[0015] 위의 프로세스에 대한 대안적인 접근법은 무손실 인코딩을 포함할 수 있고, 여기서 프레임은 정보의 손실이 없거나 무시할 수 있게 인코딩되고 디코딩되고; 이것은 출력 프레임 및/또는 비디오 픽처에서 더 큰 해상도 및/또는 다른 세부사항을 초래할 수 있다. 그러나, 무손실 인코딩 및 디코딩은 때때로 아래에서 더 상세히 언급되는 바와 같이 소정 종류의 이미지 프로세싱에 더 효율적일 수 있지만, 무손실 인코딩은 또한 메모리 리소스들 및 프로세싱 시간들의 측면에서 매우 비쌀 수 있다. 이는 픽처 또는 이미지 크기가 최대 8K x 4K(7680x4320)일 수 있는 초고화질(UHD) 비디오 코딩에서 특히 명백하고; 큰 픽처 크기는 칩 및/또는 모듈 설계에 큰 난제를 제기할 수 있다. 그 이유 중 하나는 UHD가 모션 추정 및 보상을 위해 참조 블록들을 버퍼링하기 위한 온-칩 또는 다른 프로세싱 메모리와 모션 추정에서 더 큰 검색 범위를 요구하기 때문이다. UHD 프로세싱은 관련된 더 큰 픽처 크기들로 인해 손실 인코딩 및 디코딩에 대한 난제들을 제시할 수 있다.

[0016] 본원에 개시된 실시예들은 결합된 무손실 및 손실 비디오 압축 코딩을 사용하여 보다 효율적인 시그널링, 디코딩 및 인코딩을 가능하게 한다. 실시예에서, 픽처는 먼저 품질 및 계산 요건들에 기반하여 서브-픽처들로 나뉘어질 수 있다. 인코더는 픽처들 및/또는 GOP의 인코딩 및/또는 디코딩을 수행하는 CPU 또는 다른 디바이스, 회로 또는 구성요소에 프로세싱 코어들(또는 하드웨어 스레드들)이 있는 만큼 많은 서브-픽처들을 생성할 수 있다. 각각의 서브-픽처가 독립적으로 코딩될 수 있기 때문에, 이러한 형태의 태스크 분할은 모든 이용가능한 컴퓨팅 리소스들을 효과적으로 사용함으로써 효율적인 인코딩 및/또는 디코딩을 허용할 수 있다. 또한, 무손실 인코딩은 예를 들어 전체 프레임의 소정 서브-픽처에 대해 변환 및 양자화를 사용하는 손실 코딩보다 더 나은 압축을 제공할 수 있고; 결과적으로, 결합된 무손실 및 손실 코딩은 무손실 코딩 단독보다 우수한 성능을 초래할 수 있다.

[0017] 이제 도 1을 참조하면, 현재 프레임이 복수의 서브-픽처로 나뉘어진 예시적인 실시예가 예시된다. 서브-픽처들은 현재 프레임보다 작은 현재 프레임의 임의의 부분을 포함할 수 있고; 현재 프레임의 서브-픽처들은 결합되어 현재 프레임 모두를 커버할 수 있다. 비록 도 1이 2 개 또는 4 개의 서브-픽처들로 나뉘어진 예시적인 현재 프레임들을 예시하지만, 본 개시내용의 전체를 본 통상의 기술자들은 해상도, 효율성, 또는 임의의 다른 고려사항을 위해 적절하게 임의의 수의 서브-픽처들이 사용될 수 있음을 인식할 것이다.

[0018] 여전히 도 1을 참조하면, 서브-픽처는 정사각형 및/또는 직사각형 형상, 정사각형 및/또는 직사각형 형상들을 갖는 2 개 이상의 블록들의 조합에 의해 정의된 형상 등을 제한 없이 포함하는 임의의 적합한 형상을 가질 수 있다. 각각의 블록은 블록의 하나 이상의 부분들 및/또는 피처(feature)들의 좌표들을 사용하여 식별 및/또는 시그널링될 수 있고, 여기서 좌표들은 프레임 및/또는 픽처의 하나 이상의 모서리들 및/또는 측면에서 측정된 프레임 및/또는 픽처에 걸친 픽셀들의 수를 나타낼 수 있다. 예를 들어, 제한 없이, 블록은 직사각형 블록의 식별을 위한 2 개의 x 좌표들 및 2 개의 y 좌표들과 같은 꼭지점들의 좌표들을 사용하여 식별될 수 있다. 서브-픽처 및/또는 그 일부는 제한 없이 선형 방정식들 또는 수학적으로 등가 식들, 이를테면 라인-세그먼트 종점들을 사용하거나, 제한 없이 지수 또는 다른 곡선들을 사용하여 정의된 하나 이상의 곡선 에지들을 사용하거나 등을 사용한 기하구조 분할을 포함하는 점들, 라인들, 및/또는 형상들의 임의의 적합한 기하구조 설명을 사용하여 대안적으로 또는 추가로 식별될 수 있다.

[0019] 계속해서 도 1을 참조하면, 서브-픽처들은 서로 별도로 코딩될 수 있다. 예를 들어, 제한 없이, 복수의 서브-픽처들의 제1 구역은 제1 프로세서 스레드를 사용하여 인코딩 및/또는 디코딩될 수 있고 제3 구역 요소는 제2 프로세서 스레드를 사용하여 디코딩될 수 있다. 본원에서 사용된 "프로세서 스레드"는 본 개시내용의 전체를 검토할 때 통상의 기술자들에게 발생할 수 있는 다중스레드 병렬 프로세스의 스레드를 실행할 수 있는 임의의 프로세서 코어 및/또는 다른 하드웨어 요소를 포함할 수 있다. 각 서브-픽처가 독립적으로 코딩되는 실시예에

서, 이러한 형태의 태스크 분할은 이용가능한 모든 컴퓨팅 리소스들을 효과적으로 사용함으로써 효율적인 인코딩을 허용할 수 있다.

[0020] 여전히 도 1을 참조하면, 무손실 코딩은 소스 비디오가 임의의 손실 없이 보존되는 것이 위에서 설명된 하나 이상의 이유로 바람직한 픽처의 블록들의 서브세트에 선택적으로 적용될 수 있다. 비제한적인 예로서, 무손실 코딩을 위한 픽처의 서브세트의 선택은 코딩 효율의 이유로 행해질 수 있다. 그러한 경우들에서, 손실 및 무손실 모드들에서 CTU를 코딩하는 레이트-왜곡(RD: rate-distortion) 비용을 평가한 후에 무손실 코딩 모드 판정이 이루어질 수 있다. 소정 사용 사례들에서, 비디오의 부분들은 애플리케이션들에 의해 지시된 이유들 때문에 무손실 모드로 인코딩되도록 사용자에 의해 선택될 수 있다. 비제한적인 예는 소스 품질 유지가 사용자에게 바람직한 프레임의 일부인 상황들을 포함할 수 있다. 이러한 사용자 선택들이 이루어지면, 전체 구역은 RD 분석을 수행하지 않고 무손실 코딩을 사용하는 것으로 표시될 수 있다.

[0021] 대안적으로 또는 추가적으로, 그리고 도 1을 추가로 참조하여, 서브-픽처는 인코더 및/또는 다른 하드웨어 및/또는 소프트웨어 구성요소 및/또는 프로세스에 의해 더 많은 양의 모션이 검출 및/또는 존재하는 픽처의 영역, 구역 및/또는 세분으로 식별될 수 있고; 그러한 구역들은 서브-픽처들이 중요한 것으로 간주될 수 있고 무손실 코딩을 사용하여 코딩될 수 있는 반면, 모션이 거의 또는 전혀 없는 서브-픽처들은 손실 코딩을 사용하여 코딩될 수 있다. 하나의 예가 도 2에 도시되어 있고, 여기서 픽처(200)는 2 개의 서브-픽처들: 모션을 갖는 제1 구역(204) 및 모션이 없는 제3 구역(208)으로 나뉜다. 위에서 언급된 바와 같이, 일부 경우들에서, 무손실 코딩은 변환 및 양자화를 사용하는 손실 코딩보다 더 나은 압축을 제공할 수 있다. 대안 또는 추가 예에서, 픽처(200)는 제1 무손실 프로토콜을 사용하여 인코딩된 제1 구역(204), 제2 무손실 프로토콜을 사용하는 제2 구역(도시되지 않음), 및 손실 프로토콜을 사용하는 제3 구역으로 나뉠 수 있다.

[0022] 다시 도 1을 참조하면, 픽처는 서브-픽처들, 슬라이스들, 타일들로 나뉠 수 있다. 블록(CTU)들은 인트라 또는 인터 코딩 모드에서 코딩될 수 있는 코딩 단위들일 수 있다. 서브-픽처는 단일 CTU 및/또는 복수의 CTU들을 포함할 수 있다. 실시예에서, CTU들의 서브세트 내의 각각의 CTU는 무손실 코딩이 CTU에서 사용되는지 여부를 시그널링할 수 있고; 대안적으로 또는 추가적으로, 제한 없이 연속적으로 위치된 CTU들 세트와 같은 CTU 세트는 함께 시그널링될 수 있다. 무손실 및/또는 손실 코딩은 비트스트림에 제공되는 하나 이상의 헤더들에서 시그널링될 수 있다. 예를 들어, 제한 없이, CTU들은 CTU 헤더에서 무손실 및/또는 손실 코딩 모드를 시그널링함으로써 무손실 코딩 모드에서 코딩될 수 있다. 블록들의 서브세트(CTU)에 대한 무손실 코딩의 선택적인 사용은 대안적으로 또는 추가적으로 더 높은-레벨 신택스 단위에서 시그널링될 수 있다. 예를 들어, 타일, 슬라이스 및/또는 서브-픽처 헤더는 해당 신택스 단위의 모든 CTU들에 대한 무손실 코딩 모드들의 사용을 시그널링할 수 있다. 서브-픽처 헤더는 명시적으로 존재하거나 이전에 시그널링된 픽처 헤더와 같은 다른 헤더의 식별자와 같은 메커니즘을 사용하여 참조로 포함될 수 있다.

[0023] 비제한적인 예로서, 계속해서 도 1을 참조하면, 서브-픽처 헤더, CTU 헤더, 및/또는 다른 헤더 내의 데이터 및/또는 로직은, 제한 없이, 무손실 모드 시그널링이 인에이블되는지, 또는 다시 말해서 무손실 및/또는 손실 모드가 관련 CTU, 서브-픽처 등에 대해 사용되는지 여부를 나타내는 신호를 인코더 및/또는 디코더가 시그널링 및/또는 수신하는지 여부를 나타내는 제1 비트를 포함할 수 있다. 서브-픽처 헤더, CTU 헤더, 및/또는 다른 헤더 내의 데이터 및/또는 로직은 제한 없이, 무손실 및/또는 손실 모드를 나타내는 제2 비트를 포함할 수 있고, 여기서 무손실 모드는 관련 CTU, 서브-픽처 등이 전술한 바와 같이 무손실 인코딩 및 디코딩 프로토콜을 사용하여 인코딩 및 디코딩되는 모드이다. 다음은 이용될 수 있는 로직 및 데이터의 비제한적이고 예시적인 예이다.

[0024] Sub_picture_header{

[0025] ...

[0026] ......

[0027] ......

[0028] ...lossless_mode_signaling [1-bit]

[0029] if(lossless_mode_signaling){

[0030] lossless_mode [1 bit];

[0031] }

[0032] ...

[0033] ......

[0034] ......

[0035] }

[0036] 손실 또는 무손실 모드는 PPS, SPS, 블록, 서브-블록 또는 다른 헤더와 같은 헤더에서 lossless_coding_contraint_flag 등을 사용하여 대안적으로 또는 추가적으로 시그널링될 수 있다.

[0037] 여전히 도 1을 참조하면, 본 개시내용에서 설명된 프로세스들을 수행하도록 구성된 인코더 및/또는 디코더는, 예를 들어 무손실 인코딩 프로토콜에 대응하는 식별자 및/또는 비트를 사용하여 사용된 무손실 인코딩 프로토콜을 시그널링 및/또는 검출하도록 구성될 수 있다. 대안적으로 또는 추가적으로, 인코더 및/또는 디코더는, 예를 들어 주어진 표준, 릴리스(release), 또는 균일 표준을 채택하는 다른 접근법과 일치하는 특정 무손실 인코딩 및 디코딩 프로토콜을 동작하도록 구성될 수 있다. 2 개 이상의 표준 프로토콜들이 있을 수 있고, 그 선택은 2 개 이상의 잠재적 선택들을 인코딩하기에 충분한 수의 비트들을 사용하여 비트스트림으로 시그널링될 수 있다.

[0038] 계속해서 도 1을 참조하면, 무손실 코딩 프로토콜은 이미지들, 비디오들, 프레임들, 픽처들, 서브-픽처들 등의 무손실 인코딩을 위한 임의의 프로토콜을 포함할 수 있다. 비제한적인 예로서, 인코더 및/또는 디코더는 무손실 코딩을 달성할 수 있고, 이는 변환 코딩 스테이지를 우회하고 잔차를 직접 인코딩하는 것이다. 본 개시내용에서 "스킵 잔차 코딩 변환"으로 지칭될 수 있는 이 접근법은 예를 들어 일부 형태들의 블록-기반 하이브리드 비디오 코딩에서 수행되는 바와 같은 이산 코사인 변환(DCT)들의 패밀리로부터의 변환을 적용함으로써 이하에서 더 상세히 설명되는 바와 같이 잔차의 변환을 공간 도메인에서 주파수 도메인으로 스킵함으로써 달성될 수 있다. 무손실 인코딩 및 디코딩은 무손실 코딩을 위한 정규 및 TS 잔차 코딩(RRC, TSRC), 및 무손실 및 손실 동작 모드들에 대한 RRC 및 TSRC에 대한 수정들에 관한 JVET-Q00069의 핵심 실험 CE3-1, 무손실 코딩을 위한 블록 차동 펄스-코드 변조(BDPCM) 및 고급 기법들을 인에이블링(enable)하는 것에 관한 JVET-Q0080의 핵심 실험 CE3-2, 및 상이한 RRC/TSRC 기법들과 BDPCM의 조합 등을 제한 없이 포함하는 하나 이상의 대안적인 프로세스들 및/또는 프로토콜들에 따라 수행될 수 있다.

[0039] 도 1을 추가로 참조하면, 본 개시내용에서 설명된 인코더는 TS 잔차 코딩을 사용하여 하나 이상의 필드들을 인코딩하도록 구성될 수 있고, 여기서 하나 이상의 필드들은 제한 없이 임의의 픽처, 서브-픽처, 코딩 단위, 코딩 트리 단위, 트리 단위, 블록, 슬라이스, 타일 및/또는 이들의 임의의 조합을 포함할 수 있다. 본 개시내용에서 설명된 디코더는 TS 잔차 코딩에 따라 및/또는 TS 잔차 코딩을 사용하여 하나 이상의 필드들을 디코딩하도록 구성될 수 있다. 변환 스킵 모드에서, 필드의 잔차들은 제한 없이 4개의 픽셀들 x 4개의 픽셀들의 크기와 같이 주어진 크기의 비-오버랩 서브블록들 또는 다른 세분들의 단위들로 코딩될 수 있다. 최종 중요 스캔 포지션을 코딩하는 대신, 변환될 필드의 각 스캔 포지션의 양자화 인덱스가 코딩될 수 있고; 최종 서브블록 및/또는 세분 포지션은 이전 세분들의 레벨들에 기반하여 추론될 수 있다. TS 잔차 코딩은 역방향 방식이 아닌 순방향 방식으로 대각 스캔을 수행할 수 있다. 순방향 스캐닝 순서는 변환 블록 내의 스캔 서브블록들 뿐만 아니라 서브블록 및/또는 세분 내의 포지션들에 적용될 수 있고; 실시예에서, 최종 (x, y) 포지션의 시그널링이 없을 수 있다. 비제한적인 예로서, coded_sub_block_flag는 모든 이전 플래그들이 0일 때 최종 서브블록을 제외한 모든 서브블록에 대해 코딩될 수 있다. sig_coeff_flag 컨텍스트 모델링은 감소된 템플릿을 사용할 수 있다. sig_coeff_flag의 컨텍스트 모델은 상단 및 좌측 이웃 값을 따를 수 있고; abs_level_gt1 플래그의 컨텍스트 모델은 또한 좌측 및 상단 sig_coeff_flag 값을 따를 수 있다.

[0040] 여전히 도 1을 참조하면, 비제한적인 예로서, TS 잔차 코딩 프로세스의 제1 스캔 패스 동안, 중요 플래그(sig_coeff_flag), 부호 플래그(coeff_sign_flag), 1보다 큰 절대 레벨 플래그(abs_level_gtx_flag[0]), 및 패리티 (par_level_flag)가 코딩될 수 있다. 주어진 스캔 포지션에 대해, sig_coeff_flag가 1과 같으면, coeff_sign_flag 다음, abs_level_gtx_flag[0](절대 레벨이 1보다 큰지 여부를 지정)가 코딩될 수 있다. abs_level_gtx_flag[0]이 1과 같으면, par_level_flag는 절대 레벨의 패리티를 지정하기 위해 추가로 코딩된다. 제2 또는 후속 스캔 패스 동안, 절대 레벨이 1보다 큰 각 스캔 포지션에 대해, i = 1...4에 대해 최대 4 개의 abs_level_gtx_flag[i]가 코딩되어 주어진 포지션의 절대 레벨이 각각 3, 5, 7 또는 9보다 더 큰지를 나타낼 수 있다. 제3 또는 최종 "나머지" 스캔 패스 동안, 절대 레벨 abs_remainder로 저장될 수 있는 나머지는 바이패스 모드에서 코딩될 수 있다. 절대 레벨의 나머지는 고정된 라이스 파라미터 값(fixed rice

parameter value) 1을 사용하여 이진화될 수 있다.

[0041] 추가로 도 1을 참조하면, 제1 스캔 패스 및 제2 또는 "x보다 큰" 스캔 패스의 빈(bin)들은 제한 없이 TU와 같은 필드의 컨텍스트 코딩된 빈들의 최대 수가 소진될 때까지 컨텍스트 코딩될 수 있다. 잔차 블록에서 컨텍스트 코딩된 빈들의 최대 수는 비제한적인 예에서 샘플 포지션당 평균 1.75*block_width*block_height, 또는 동등하게 1.75 컨텍스트 코딩된 빈들로 제한될 수 있다. 전술된 나머지 스캔 패스와 같은 마지막 스캔 패스의 빈들은 바이패스 코딩될 수 있다. 제한 없이 RemCcbs와 같은 변수는 블록 또는 다른 필드에 대한 컨텍스트-코딩된 빈들의 최대 수로 먼저 설정될 수 있고 컨텍스트-코딩된 빈이 코딩될 때마다 1씩 감소될 수 있다. 비제한적인 예에서, RemCcbs가 4보다 크거나 같으면, sig_coeff_flag, coeff_sign_flag, abs_level_gt1_flag 및 par_level_flag를 포함할 수 있는 제1 코딩 패스의 신택스 요소들은 컨텍스트-코딩된 빈들을 사용하여 코딩될 수 있다. 일부 실시예들에서, 제1 패스를 코딩하는 동안 RemCcbs가 4보다 작아지면, 제1 패스에서 아직 코딩되지 않은 나머지 계수들은 나머지 스캔 패스 및/또는 제3 패스에서 코딩될 수 있다.

[0042] 여전히 도 1을 참조하면, 제1 패스 코딩이 완료된 후, RemCcbs가 4보다 크거나 같으면, abs_level_gt3_flag, abs_level_gt5_flag, abs_level_gt7_flag 및 abs_level_gt9_flag를 포함할 수 있는 제2 코딩 패스의 신택스 요소들은 컨텍스트 코딩된 빈들을 사용하여 코딩될 수 있다. 제2 패스를 코딩하는 동안 RemCcbs가 4보다 작아지면, 제2 패스에서 아직 코딩되지 않은 나머지 계수들은 나머지 및/또는 제3 스캔 패스에서 코딩될 수 있다. 일부 실시예들에서, TS 잔차 코딩을 사용하여 코딩된 블록은 BDPCM 코딩을 사용하여 코딩되지 않을 수 있다. BDPCM 모드에서 코딩되지 않은 블록에 대해, 컨텍스트 코딩된 빈들의 최대 수에 도달할 때까지 스킵 잔차 코딩을 변환하기 위해 레벨 매핑 메커니즘이 적용될 수 있다. 레벨 매핑은 시그널링 비용을 감소시키기 위해 현재 계수 레벨을 예측하도록 상단 및 좌측 이웃 계수 레벨들을 사용할 수 있다. 주어진 잔차 포지션에 대해, absCoeff는 매핑 전 절대 계수 레벨로 나타낼 수 있고, absCoeffMod는 매핑 후 계수 레벨로 나타낼 수 있다. 비제한적인 예로서, X0가 좌측 이웃 포지션의 절대 계수 레벨을 나타내고 X1이 위 이웃 포지션의 절대 계수 레벨을 나타내는 경우, 레벨 매핑은 다음과 같이 수행될 수 있다:

$$pred = \max(X_o, X_1)\,; if\ (absCoeff == pred) absCoeffMod = 1; else\ absCoeffMod$$

[0043] $$= (absCoeffMod < pred)?\, absCoeff + 1 : absCoeff$$

[0044] 이어서 absCoeffMod 값은 위에서 설명한 바와 같이 코딩될 수 있다. 모든 컨텍스트 코딩된 빈들이 소진된 후, 현재 블록 및/또는 필드 및/또는 세분의 나머지 모든 스캔 포지션들에 대해 레벨 매핑이 디스에이블될 수 있다. coded_subblock_flag가 1과 같으면 전술된 바와 같은 3 개의 스캔 패스들은 각각의 서브블록 및/또는 다른 세분에 대해 수행될 수 있고, 이는 서브블록에 적어도 하나의 0이 아닌 양자화된 잔차가 있음을 나타낼 수 있다.

[0045] 일부 실시예들에서, 여전히 도 1을 참조하면, 큰 블록에 대해 변환 스킵 모드를 사용하는 경우, 임의의 값들을 0으로 만들지 않고 전체 블록이 사용될 수 있다. 또한, 변환 스킵 모드에서 변환 시프트가 제거될 수 있다. TS 잔차 코딩에서 신호의 통계적 특성은 변환 계수들의 통계 특성과 다를 수 있다. 변환 스킵 모드에 대한 잔차 코딩은 최대 루마 및/또는 크로마 블록 크기를 지정할 수 있고; 비제한적인 예로서, 설정들은 변환 스킵 모드가 MaxTsSize에 의한 MaxTsSize까지의 크기의 루마 블록들에 사용되도록 허용할 수 있고, 여기서 MaxTsSize의 값은 PPS에서 시그널링될 수 있고 제한 없이 32와 같은 글로벌 최대 가능 값을 가질 수 있다. CU가 변환 스킵 모드에서 코딩될 때, 그의 예측 잔차는 변환 스킵 잔차 코딩 프로세스를 사용하여 양자화되고 코딩될 수 있다.

[0046] 계속해서 도 1을 참조하면, 본 개시내용에서 설명된 인코더는 BDPCM을 사용하여 하나 이상의 필드들을 인코딩하도록 구성될 수 있고, 여기서 하나 이상의 필드들은 제한 없이 임의의 픽처, 서브-픽처, 코딩 단위, 코딩 트리 단위, 트리 단위, 블록, 슬라이스, 타일 및/또는 이들의 임의의 조합을 포함할 수 있다. 본 개시내용에서 설명된 디코더는 BDPCM에 따라 및/또는 BDPCM을 사용하여 하나 이상의 필드들을 디코딩하도록 구성될 수 있다. BDPCM은 픽셀 레벨에서 전체 재구성을 유지할 수 있다. 비제한적인 예로서, BDPCM을 사용한 각 픽셀의 예측 프로세스는 4 개의 주요 단계들을 포함할 수 있고, 이는 블록 내 참조들을 사용하여 각 픽셀을 예측하고, 이어서 블록의 나머지의 후속 픽셀들에 대한 블록 내 참조로 사용되게 재구성할 수 있다: (1) 블록 내 픽셀 예측, (2) 잔차 계산, (3) 잔차 양자화, (4) 픽셀 재구성.

[0047] 여전히 도 1을 참조하면, 블록 내 픽셀 예측은 복수의 참조 픽셀들을 사용하여 각 픽셀을 예측할 수 있고; 비제한적인 예로서, 복수의 참조 픽셀들은 예측될 픽셀(p)의 좌측에 있는 픽셀(α), p 위의 픽셀(β) 및 p의 위쪽 및 좌측에 있는 픽셀(γ)을 포함할 수 있다. p의 예측은 다음과 같이 제한 없이 공식화될 수 있다:

$$p = \begin{cases} min(\alpha,\beta), \text{if } \gamma \le max(\alpha,\beta) \\ max(\alpha,\beta), \text{if } \gamma \ge min(\alpha,\beta) \\ \alpha + \beta - \gamma, \text{Otherwise} \end{cases}$$

[0048]

[0049] 여전히 도 1을 참조하면, 예측 값이 일단 계산되면, 그 잔차가 계산될 수 있다. 이 스테이지의 잔차가 디코더 측에서 무손실이고 액세스가능하지 않기 때문에, $\tilde{r}$ 로서 나타낼 수 있고 예측(p)에서 원래 픽셀 값(o)의 감산으로서 계산될 수 있다:

$$\tilde{r} = o - p$$

[0050]

[0051] 추가로 도 1을 참조하면, 픽셀-레벨 독립성은 잔차 변환을 스킵하고 공간 도메인 양자화를 통합함으로써 달성될 수 있다. 이것은 선형 양자화기(Q)에 의해 수행되어 다음과 같이 양자화된 잔차 값(r)을 계산할 수 있다:

$$r = Q(\tilde{r})$$

[0052]

[0053] 양자화 파라미터(QP)에 의해 부과된 정확한 레이트-왜곡 비율을 수용하기 위해, BDPCM은 예를 들어 제한 없이 위에서 설명된 바와 같이 전송-스킵 모드 방법에서 사용되는 공간 도메인 정규화를 채택할 수 있다. 양자화된 잔차 값(r)은 인코더에 의해 송신될 수 있다.

[0054] 여전히 도 1을 참조하면, BDPCM의 다른 상태는 이전 단계들로부터의 p 및 r을 사용한 픽셀 재구성을 포함할 수 있고, 이는 예를 들어 제한 없이 다음과 같이 디코더에서 또는 디코더에 의해 수행될 수 있다:

$$c = p + r$$

[0055]

[0056] 일단 재구성되면, 현재 픽셀은 동일한 블록 내의 다른 픽셀들에 대한 블록 내 참조로 사용될 수 있다.

[0057] BDPCM 알고리즘의 예측 방식은 원래의 픽셀 값이 예측과 거리가 멀 때 상대적으로 큰 잔차가 있는 경우에 사용될 수 있다. 스크린 콘텐츠에서, 이것은 블록 내 참조들이 배경 계층에 속하는 반면, 현재 픽셀이 전경 계층에 속하거나, 그 반대의 경우에 발생할 수 있다. "계층 전이" 상황이라고 지칭될 수 있는 이러한 상황에서, 참조들에서 이용가능한 정보는 정확한 예측을 위해 적절하지 않을 수 있다. 시퀀스 레벨에서, BDPCM 인에이블 플래그는 SPS에서 시그널링될 수 있고; 이 플래그는, 예를 들어 제한 없이 전술된 바와 같이 변환 스킵 모드가 SPS에서 인에이블되는 경우에만 제한 없이 시그널링될 수 있다. BDPCM이 인에이블될 때, 플래그는, CU 크기가 루마 샘플들의 관점에서 MaxTsSize x MaxTsSize보다 작거나 같고 CU가 인트라 코딩되는 경우 CU 레벨에서 송신될 수 있고, 여기서 MaxTsSize는 변환 스킵 모드가 허용되는 최대 블록 크기이다. 이 플래그는 일반 인트라 코딩 또는 BDPCM이 사용되는지 여부를 나타낼 수 있다. BDPCM이 사용되는 경우, 예측이 수평인지 수직인지를 나타내기 위해 BDPCM 예측 방향 플래그가 송신될 수 있다. 이어서, 필터링되지 않은 참조 샘플들과 함께 규칙적인 수평 또는 수직 인트라 예측 프로세스를 사용하여 블록이 예측될 수 있다.

[0058] 이제 도 3을 참조하면, 결합된 무손실 및 손실 코딩의 방법(300)의 예시적인 실시예가 예시된다. 단계(305)에서, 디코더는 비트스트림을 수신한다. 단계(310)에서, 디코더는 비트스트림에서 현재 프레임을 식별한다. 현재 프레임은 제1 구역, 제2 구역 및 제3 구역을 포함할 수 있고, 이들 중 임의의 구역은 전술된 바와 같은 임의의 구역을 포함할 수 있고; 구역들은 프레임 헤더 정보를 사용하여 플래그 지정될 수 있고/있거나 좌표들, 기하 구조 정보, 각 구역에 포함된 블록들 및/또는 CTU들의 식별 등을 사용하여 묘사되거나 달리 설명될 수 있다. 실시예에서, 디코더는 현재 프레임에서 제1 구역, 제2 구역 및 제3 구역 중 2 개의 구역들만을 식별할 수 있는 반면, 나머지 구역은 현재 프레임의 나머지 타일들, 슬라이스들, 블록들, CTU들 등으로 식별될 수 있다. 3 개 초과의 구역들이 있을 수 있고; 방법(300)은 임의의 추가 구역들과 관련하여 수행되는 본 개시내용에 설명된 임의의 프로세싱 단계를 포함할 수 있다.

[0059] 단계(315)에서, 그리고 계속해서 도 3을 참조하여, 디코더는 비트스트림에서, 제1 구역이 블록 차동 펄스 코드 변조에 따라 인코딩된다는 표시를 검출하고; 이것은 도 1-도 2를 참조하여 전술된 바와 같이 제한 없이 수행될 수 있다. 검출은 예를 들어 전술된 바와 같이 블록 차동 펄스 코드 변조가 인에이블되는 검출을 포함 및/또는 선행될 수 있다. 실시예에서, 비트스트림은 제1 구역에 대응하는 서브-픽처 헤더를 포함할 수 있다. 검출은 적어도 제1 구역이 서브-픽처 및/또는 구역-특정 헤더에서 블록 차동 펄스 코드 변조에 따라 인코딩된다는 표시를 검출하는 것을 포함할 수 있다. 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함될 수

있다. 예를 들어, 제한 없이, sps_bdpcm_enabled_flag는 SPS 및/또는 bdpcm가 시퀀스에 대해 인에이블되는 경우 다른 헤더에서 1로 설정될 수 있다. 1과 동일한 sps_bdpcm_enabled_flag는 intra_bdpcm_luma_flag 및/또는 intra_bdpcm_chroma_flag가 코딩 단위, 및/또는 인트라 코딩 단위들 및/또는 다른 필드들에 대한 다른 필드-특정 신택스에 존재할 수 있음을 지정할 수 있다. 0과 동일한 sps_bdpcm_enabled_flag는 intra_bdpcm_luma_flag 및/또는 intra_bdpcm_chroma_flag가 코딩 단위, 및/또는 인트라 코딩 단위들 및/또는 다른 필드들에 대한 다른 필드-특정 신택스에 존재하지 않는다는 것을 지정할 수 있다. 존재하지 않는 경우, sps_bdpcm_enabled_flag의 값은 0과 동일한 것으로 추론될 수 있다. 실시예에서, 1과 동일한 gci_no_bdpcm_constraint_flag는 제한 없이 OlsInScope 파라미터에 의해 정의될 수 있는 주어진 세트의 모든 픽처들에 대한 sps_bdpcm_enabled_flag가 0과 동일해야 함을 지정할 수 있다. 0과 동일한 gci_no_bdpcm_constraint_flag는 이러한 제약을 부과하지 않을 수 있다. 추가의 비제한적인 예로서, 1과 동일한 intra_bdpcm_luma_flag는 BDPCM이 현재 루마 코딩 블록 및/또는 위치(x0, y0)의 다른 필드에 적용될 수 있음을 지정할 수 있고 즉, 변환이 스킵되고; 루마 인트라 예측 모드는 intra_bdpcm_luma_dir_flag에 의해 지정될 수 있다. 예를 들어 제한 없이, 0과 동일한 intra_bdpcm_luma_flag는 BDPCM이 위치(x0, y0)에서 현재 루마 코딩 블록 및/또는 다른 필드에 적용되지 않음을 지정할 수 있다. intra_bdpcm_luma_flag가 존재하지 않으면, 이는 0과 같은 것으로 추론될 수 있다. 변수(BdpcmFlag[x][y][cIdx])는 x = x0..x0 + cbWidth 1, y = y0..y0 + cbHeight 1 및 cIdx = 0에 대해 intra_bdpcm_luma_flag와 동일하게 설정할 수 있다. 0과 동일한 intra_bdpcm_luma_dir_flag는 BDPCM 예측 방향이 수평임을 지정한다. 1과 동일한 intra_bdpcm_luma_dir_flag는 BDPCM 예측 방향이 수직임을 지정할 수 있다. 변수(BdpcmDir[x][y][cIdx])는 x = x0..x0 + cbWidth 1, y = y0..y0 + cbHeight 1 및 cIdx = 0에 대해 intra_bdpcm_luma_dir_flag와 동일하게 설정될 수 있다. 서브-픽처 및/또는 구역은 현재 프레임의 제 3 서브-픽처 및/또는 다른 요소에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함될 수 있다.

[0060] 단계(320)에서, 그리고 계속해서 도 3을 참조하면, 디코더는 비트스트림에서, 제2 구역이 전송 스킵 잔차 코딩에 따라 인코딩된다는 표시를 검출하고; 이것은 제한 없이 도 1-도 2를 참조하여 전술된 바와 같이 수행될 수 있다. 검출은 예를 들어 전술된 바와 같이 전송 스킵 잔차 코딩이 인에이블되는 검출을 포함 및/또는 이 검출에 선행할 수 있다. 실시예에서, 비트스트림은 제1 구역에 대응하는 서브-픽처 헤더를 포함할 수 있다. 검출은 적어도 제1 구역이 서브-픽처 헤더에서 전송 스킵 잔차 코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 것을 포함할 수 있고; 이것은 변환 스킵 인에이블 플래그를 포함할 수 있다. 예를 들어, 제한 없이, 1과 동일한 sh_ts_residual_coding_disabled_flag는 residual_coding 신택스 구조가 현재 슬라이스 및/또는 다른 필드에 대한 변환 스킵 블록의 잔차 샘플들을 파싱하는 데 사용될 수 있음을 지정할 수 있다. 0과 동일한 sh_ts_residual_coding_disabled_flag는 residual_ts_coding 신택스 구조가 현재 슬라이스에 대한 변환 스킵 블록의 잔차 샘플들을 파싱하는데 사용될 수 있음을 지정할 수 있다. sh_ts_residual_coding_disabled_flag가 존재하지 않는 경우, 이는 0과 동일하다고 추론될 수 있다. transform_skip_flag[x0][y0][cIdx]는 변환이 연관된 변환 블록에 적용될 수 있는지 여부를 지정할 수 있다. 어레이 인덱스들(x0, y0)은 픽처의 좌측-상단 루마 샘플에 대한 고려된 변환 블록의 좌측-상단 루마 샘플의 위치(x0, y0)를 지정할 수 있다. 어레이 인덱스(cIdx)는 색상 성분에 대한 표시자를 지정할 수 있다; 예를 들어, Y의 경우 0, Cb의 경우 1, Cr의 경우 2와 같을 수 있다. 1과 동일한 transform_skip_flag[x0][y0][cIdx]는 연관된 변환 블록에 어떠한 변환도 적용될 수 없음을 지정할 수 있다. 0과 동일한 transform_skip_flag[x0][y0][cIdx]는 변환이 연관된 변환 블록에 적용되는지 여부의 판정이 다른 신택스 요소들에 따름을 지정할 수 있다. 변환 스킵 모드는 대안적으로 또는 추가적으로 암시적으로 시그널링될 수 있다. 예를 들어, transform_skip_flag[x0][y0][cIdx]가 존재하지 않으면, 다음과 같이 추론될 수 있다: BdpcmFlag[x0][y0][cIdx]가 1인 경우, transform_skip_flag[x0][y0][cIdx]는 1과 동일한 것으로 추론될 수 있고; 그렇지 않으면, BdpcmFlag[x0][y0][cIdx]가 0 인 경우, transform_skip_flag[x0][y0][cIdx]가 0과 동일한 것으로 추론될 수 있다. 현재 프레임에 대응하는 데이터에는 서브-픽처 및/또는 구역-특정 헤더가 명시적으로 포함될 수 있다. 서브 픽처 및/또는 구역은 현재 프레임의 제 3 서브-픽처 및/또는 다른 요소에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함될 수 있다.

[0061] 단계(325)에서, 그리고 계속해서 도 3을 참조하면, 디코더는 제3 구역이 손실 인코딩 프로토콜에 따라 인코딩되는 것을 검출하고, 무손실 인코딩 프로토콜에 대응하는 손실 디코딩 프로토콜에 따라 제3 구역을 디코딩하고; 이것은 DCT를 포함하는 프로세스들 및 아래에서 설명되는 다른 프로세스들을 포함하여 본원에 설명된 임의의 손실 디코딩 프로세스에 따라 수행될 수 있다. 비트스트림은 제3 구역에 대응하는 서브-픽처 및/또는 구역-특정 헤더를 포함할 수 있고, 검출은 서브-픽처 헤더에서 제3 구역이 손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 포함할 수 있다. 서브-픽처 및/또는 구역-특정 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함될 수 있다. 서브-픽처 및/또는 구역-특정 헤더는 예를 들어 전술된 바와 같이, 제3 서브-픽처에 대응하는 서

브-픽처 및/또는 구역-특정 헤더의 식별자를 참조하여 포함될 수 있다. 실시예에서, 디코더는 위에서 정의된 바와 같이 제1 프로세서 스레드를 사용하여 제1 구역을 디코딩하고, 제2 프로세서 스레드를 사용하여 제3 구역 요소를 디코딩하도록 구성될 수 있다.

[0062] 계속해서 도 3을 참조하면, 디코더는 현재 프레임을 디코딩할 수 있다. 현재 프레임을 디코딩하는 것은 BDPCM 인코딩 프로토콜에 대응하는 BDPCM 디코딩 프로토콜을 사용하여 제1 구역을 디코딩하는 것을 포함할 수 있다. 현재 프레임을 디코딩하는 것은 전송-스킵 잔차 인코딩 프로토콜에 대응하는 전송-스킵 잔차 디코딩 프로토콜을 사용하여 제2 구역을 디코딩하는 것을 포함할 수 있다. 현재 프레임을 디코딩하는 것은 손실 인코딩 프로토콜에 대응하는 손실 디코딩 프로토콜을 사용하여 제3 구역을 디코딩하는 것을 포함할 수 있다.

[0063] 여전히 도 3을 참조하면, 디코더는 비트 스트림을 수신하고 비트스트림을 양자화된 계수들로 디코딩하도록 구성된 엔트로피 디코더 프로세서, 역 이산 코사인을 수행하는 것을 포함하여 양자화된 계수들을 프로세싱하도록 구성된 역양자화 및 역변환 프로세서, 디블록킹 필터, 프레임 버퍼, 및 인트라 예측 프로세서를 포함할 수 있다. 제1 구역, 제2 구역 및 제3 구역 중 적어도 하나는 쿼드트리 플러스 이진 결정 트리의 일부를 형성할 수 있다. 제1 구역, 제2 구역 및 제3 구역 중 적어도 하나는 코딩 트리 단위를 포함한다. 일부 구현들에서, 제1 구역, 제2 구역, 및 제3 구역 중 적어도 하나는 코딩 트리 단위(CTU), 코딩 단위(CU), 또는 예측 단위(PU)를 포함할 수 있다.

[0064] 도 4는 결합된 손실 및 무손실 코딩 프로토콜들을 사용하여 비트스트림(428)을 디코딩할 수 있는 예시적인 디코더(400)를 예시하는 시스템 블록도이다. 디코더(400)는 엔트로피 디코더 프로세서(404), 역양자화 및 역변환 프로세서(408), 디블로킹 필터(412), 프레임 버퍼(416), 모션 보상 프로세서(420) 및/또는 인트라 예측 프로세서(424)를 포함할 수 있다.

[0065] 동작 중에, 여전히 도 4를 참조하면, 비트 스트림(428)은 *?*디코더(400)에 의해 수신될 수 있고 엔트로피 디코더 프로세서(404)에 입력될 수 있으며, 엔트로피 디코더 프로세서(404)는 비트 스트림의 부분들을 양자화된 계수들로 엔트로피 디코딩할 수 있다. 양자화된 계수들은 역양자화 및 역변환 프로세서(408)에 제공될 수 있고, 이는 역양자화 및 역변환을 수행하여 잔차 신호를 생성할 수 있고, 잔차 신호는 프로세싱 모드에 따라 모션 보상 프로세서(420) 또는 인트라 예측 프로세서(424)의 출력에 추가될 수 있다. 모션 보상 프로세서(420) 및 인트라 예측 프로세서(424)의 출력은 이전에 디코딩된 블록에 기반한 블록 예측을 포함할 수 있다. 예측 및 잔차의 합은 디블록킹 필터(412)에 의해 프로세싱되고 프레임 버퍼(416)에 저장될 수 있다.

[0066] 계속해서 도 4를 참조하면, 디코더(400)는 임의의 순서 및 임의의 반복도로 본 개시내용에 설명된 임의의 실시예에서 임의의 방법, 방법 단계 또는 방법 단계들의 시퀀스를 수행하도록 설계 및/또는 구성될 수 있다. 예를 들어, 디코더(400)는 원하는 또는 명령된 결과가 달성될 때까지 단일 단계 또는 시퀀스를 반복적으로 수행하도록 구성될 수 있고; 단계 또는 단계들의 시퀀스의 반복은 이전 반복들의 출력들을 후속 반복들에 대한 입력들로 사용하여 반복적으로 및/또는 재귀적으로 수행되어, 입력들 및/또는 반복들의 출력들을 집계하여 집계 결과를 생성하고, 하나 이상의 변수들, 이를테면 글로벌 변수들을 감소 또는 감분시키고/시키거나, 더 큰 프로세싱 태스크를 반복적으로 처리되는 더 작은 프로세싱 태스크들의 세트로 나눌 수 있다. 디코더(400)는 2 개 이상의 병렬 스레드들, 프로세서 코어들 등을 사용하여 2 회 이상 단계를 동시에 및/또는 실질적으로 동시에 수행하는 것과 같이, 본 개시내용에 설명된 임의의 단계 또는 단계들의 시퀀스를 병렬로 수행할 수 있고; 병렬 스레드들 및/또는 프로세스들 간의 태스크들의 나눔은 반복들 간의 태스크들의 나눔에 적합한 임의의 프로토콜에 따라 수행될 수 있다. 통상의 기술자들은 본 개시내용 전체를 검토할 때, 단계들, 단계들의 시퀀스들, 프로세싱 태스크들 및/또는 데이터가 반복, 재귀, 및/또는 병렬 프로세싱을 사용하여 세분, 공유, 또는 달리 처리될 수 있는 다양한 방식들을 인식할 것이다.

[0067] 도 5는 모션 벡터 후보 리스트에 단일 글로벌 모션 벡터 후보를 추가하는 것을 포함하는 모션 벡터 후보 리스트를 구성할 수 있는 비디오 인코더(500)의 예시적인 실시예를 예시하는 시스템 블록도이다. 예시적인 비디오 인코더(500) 는 입력 비디오(504)를 수신할 수 있고, 입력 비디오(504)는 트리-구조화 매크로 블록 분할 방식(예를 들어, 쿼드 트리 + 이진 트리)과 같은 프로세싱 방식에 따라 초기에 세그먼트화 및/또는 나뉘어질 수 있다. 트리-구조화 매크로 블록 분할 방식의 예는 픽처 프레임을 코딩 트리 단위(CTU)들이라고 칭해지는 큰 블록 요소들로 분할하는 것을 포함할 수 있다. 일부 구현들에서, 각각의 CTU는 코딩 단위(CU)들이라고 칭해지는 다수의 서브-블록들로 1회 이상 더 분할될 수 있다. 이 분할의 최종 결과는 예측 단위(PU)들이라고 칭해질 수 있는 서브-블록들의 그룹을 포함할 수 있다. 변환 단위(TU)들이 또한 활용될 수 있다.

[0068] 여전히 도 5를 참조하면, 예시적인 비디오 인코더(500)는 인트라 예측 프로세서(512), 모션 벡터 후보 리스트에

단일 글로벌 모션 벡터 후보를 추가하는 것을 포함하는 모션 벡터 후보 리스트를 구성할 수 있는 모션 추정/보상 프로세서(512)(또한 인터 예측 프로세서로 지칭됨), 변환/양자화 프로세서(516), 역양자화/역변환 프로세서(520), 인루프 필터(524), 디코딩된 픽처 버퍼(528), 및/또는 엔트로피 코딩 프로세서(532)를 포함할 수 있다. 비트 스트림 파라미터들은 출력 비트 스트림(536)에 포함시키기 위해 엔트로피 코딩 프로세서(532)에 입력될 수 있다.

[0069] 동작 중에, 계속해서 도 5를 참조하면, 입력 비디오(504)의 프레임의 각 블록에 대해, 인트라 픽처 예측을 통해 블록을 프로세싱할지 또는 모션 추정/보상을 사용하여 프로세싱할지 여부가 결정될 수 있다. 블록은 인트라 예측 프로세서(508) 또는 모션 추정/보상 프로세서(512)에 제공될 수 있다. 블록이 인트라 예측을 통해 프로세싱되어야 하는 경우, 인트라 예측 프로세서(508)는 예측자를 출력하기 위한 프로세싱를 수행할 수 있다. 블록이 모션 추정/보상을 통해 프로세싱되어야 하는 경우, 모션 추정/보상 프로세서(512)는 적용 가능한 경우, 단일 글로벌 모션 벡터 후보를 모션 벡터 후보 리스트에 추가하는 것을 포함하는 모션 벡터 후보 리스트를 구성하는 것을 포함하는 프로세싱을 수행할 수 있다.

[0070] 여전히 도 5를 참조하면, 입력 비디오에서 예측자를 감산함으로써 잔차가 형성될 수 있다. 잔차는 변환/양자화 프로세서(516)에 의해 수신될 수 있고, 변환/양자화 프로세서(516)는 양자화될 수 있는 계수들을 생성하기 위해 변환 프로세싱(예를 들어, 이산 코사인 변환(DCT))을 수행할 수 있다. 양자화된 계수들 및 임의의 연관된 시그널링 정보는 엔트로피 인코딩 및 출력 비트 스트림(536)에 포함하기 위해 엔트로피 코딩 프로세서(532)에 제공될 수 있다. 엔트로피 인코딩 프로세서(532)는 현재 블록의 인코딩과 관련된 시그널링 정보의 인코딩을 지원할 수 있다. 또한, 양자화된 계수들은 역양자화/역변환 프로세서(520)에 제공될 수 있고, 역양자화/역변환 프로세서(520)는 예측자와 결합되고 루프 필터(524)에 의해 프로세싱될 수 있는 픽셀들을 재생할 수 있고, 루프 필터(524)의 출력은 모션 벡터 후보 리스트에 단일 글로벌 모션 벡터 후보를 추가하는 것을 포함하는 모션 벡터 후보 리스트를 구성할 수 있는 모션 추정/보상 프로세서(512)에 의해 사용을 위해 디코딩된 픽처 버퍼(528)에 저장될 수 있다.

[0071] 추가로 도 5를 참조하면, 위에서 몇 가지 변형들이 상세하게 설명되었지만, 다른 수정들 또는 추가들이 가능하다. 예를 들어, 일부 구현들에서, 현재 블록들은 임의의 대칭 블록들(8x8, 16x16, 32x32, 64x64, 128 x 128 등) 및 임의의 비대칭 블록(8x4, 16x8 등)을 포함할 수 있다.

[0072] 일부 구현들에서, 여전히 도 5를 참조하면, 쿼드트리 플러스 이진 결정 트리(QTBT)가 구현될 수 있다. QTBT에서 코딩 트리 단위 레벨에서, QTBT의 분할 파라미터들은 임의의 오버헤드를 송신하지 않고 로컬 특성에 적응하도록 동적으로 유도될 수 있다. 후속하여, 코딩 단위 레벨에서, 공동-분류기 결정 트리 구조는 불필요한 반복들을 제거하고 잘못된 예측의 위험을 제어할 수 있다. 일부 구현들에서, LTR 프레임 블록 업데이트 모드는 QTBT의 모든 리프 노드에서 이용가능한 추가 옵션으로 이용할 수 있다.

[0073] 일부 구현들에서, 여전히 도 5를 참조하면, 추가적인 신택스 요소들은 비트스트림의 상이한 계층 레벨들에서 시그널링될 수 있다. 예를 들어, 플래그는 SPS(Sequence Parameter Set)에 코딩된 인에이블 플래그를 포함함으로써 전체 시퀀스에 대해 인에이블될 수 있다. 또한, CTU 플래그는 코딩 트리 단위(CTU) 레벨에서 코딩될 수 있다.

[0074] 계속해서 도 5를 참조하면, 비일시적 컴퓨터 프로그램 제품들(즉, 물리적으로 구현된 컴퓨터 프로그램 제품들)은 하나 이상의 컴퓨팅 시스템들의 하나 이상의 데이터 프로세서들에 의해 실행될 때, 적어도 하나의 데이터 프로세서가 제한 없이 위에서 설명된 임의의 동작들(400) 및/또는 인코더(500)가 수행하도록 구성될 수 있는 것들을 포함하여, 본 개시내용에서 설명된 동작들 및/또는 이의 단계들을 수행하게 하는 명령들을 저장할 수 있다. 유사하게, 하나 이상의 데이터 프로세서들 및 하나 이상의 데이터 프로세서들에 결합된 메모리를 포함할 수 있는 컴퓨터 시스템이 또한 설명된다. 메모리는 적어도 하나의 프로세서가 본원에 설명된 동작들 중 하나 이상을 수행하게 하는 명령들을 일시적으로 또는 영구적으로 저장할 수 있다. 또한, 방법들은 단일 컴퓨팅 시스템 내에서 또는 2 개 이상의 컴퓨팅 시스템에 분산된 하나 이상의 데이터 프로세서들에 의해 구현될 수 있다. 그러한 컴퓨팅 시스템들은 네트워크(예를 들어, 인터넷, 무선 광역 네트워크, 근거리 통신망, 광역 네트워크, 유선 네트워크 등)를 통한 연결을 포함하는 하나 이상의 연결들을 통해, 다중 컴퓨팅 시스템 등 중 하나 이상 간의 직접 연결을 통해 등으로 연결될 수 있고 데이터 및/또는 커맨드들 또는 다른 명령들 등을 교환할 수 있다.

[0075] 계속해서 도 5를 참조하면, 인코더(500), 디코더(400) 및/또는 이들의 회로는 본 개시내용에 설명된 임의의 실시예에서 임의의 방법, 방법 단계 또는 방법 단계들의 시퀀스를 임의의 순서 및 임의의 반복도로 수행하도록 설계 및/또는 구성될 수 있다. 예를 들어, 인코더(500), 디코더(400), 및/또는 이들의 회로는 원하는 또는 명령

된 결과가 달성될 때까지 단일 단계 또는 시퀀스를 반복적으로 수행하도록 구성될 수 있고; 단계 또는 단계들의 시퀀스의 반복은 이전 반복들의 출력들을 후속 반복들에 대한 입력들로 사용하여 반복적으로 및/또는 재귀적으로 수행되어, 입력들 및/또는 반복들의 출력들을 집계하여 집계 결과를 생성하고, 하나 이상의 변수들, 이를테면 글로벌 변수들을 감소 또는 감분시키고/시키거나, 더 큰 프로세싱 태스크를 반복적으로 처리되는 더 작은 프로세싱 태스크들의 세트로 나눌 수 있다. 인코더(500), 디코더(400), 및/또는 이들의 회로는 2 개 이상의 병렬 스레드들, 프로세서 코어들 등을 사용하여 2 회 이상 단계를 동시에 및/또는 실질적으로 동시에 수행하는 것과 같이, 본 개시내용에 설명된 임의의 단계 또는 단계들의 시퀀스를 병렬로 수행할 수 있고; 병렬 스레드들 및/또는 프로세스들 간의 태스크들의 나눔은 반복들 간의 태스크들의 나눔에 적합한 임의의 프로토콜에 따라 수행될 수 있다. 통상의 기술자들은 본 개시내용 전체를 검토할 때, 단계들, 단계들의 시퀀스들, 프로세싱 태스크들 및/또는 데이터가 반복, 재귀, 및/또는 병렬 프로세싱을 사용하여 세분, 공유, 또는 달리 처리될 수 있는 다양한 방식들을 인식할 것이다.

[0076] 본원에 개시된 실시예들은 디코더를 포함하고, 디코더는 코딩된 비디오 비트스트림을 수신하고, 비트스트림에서, 현재 프레임을 식별하고 – 현재 프레임은 제1 구역, 제2 구역 및 제3 구역을 포함함 –, 비트스트림에서, 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩되는 것을 검출하고, 비트스트림에서, 제2 구역이 변환 스킵 잔차 코딩을 사용하여 인코딩되는 것을 검출하고, 비트스트림에서, 제3 구역이 손실 인코딩을 사용하여 인코딩되는 것을 검출 – 손실 인코딩은 인터-예측 및 인트라-예측 중 적어도 하나를 포함함 –하도록 구성된 회로를 포함한다.

[0077] 일부 실시예들에서, 비트스트림은 제1 구역에 대응하는 서브-픽처 헤더를 더 포함하고, 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩된다는 것을 검출하는 것은 적어도 제1 구역이 서브-픽처 헤더에서 블록 차동 펄스 코드 변조를 사용하여 인코딩된다는 표시를 검출하는 것을 더 포함한다. 일부 실시예들에서, 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함된다. 일부 실시예들에서, 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함된다.

[0078] 일부 실시예들에서, 비트스트림은 제2 구역에 대응하는 서브-픽처 헤더를 더 포함하고, 디코더는 제2 사용 변환 스킵 잔차 코딩이 서브-픽처 헤더에서 변환 스킵 잔차 코딩을 사용하여 인코딩된다는 표시를 검출하도록 추가로 구성된다. 일부 실시예들에서, 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함된다. 일부 실시예들에서, 디코더는 제1 프로세서 스레드를 사용하여 제1 구역을 디코딩하고 제2 프로세서 스레드를 사용하여 제2 구역 요소를 디코딩하도록 추가로 구성된다. 디코더는 현재 프레임을 디코딩하도록 추가로 구성될 수 있고, 현재 프레임을 디코딩하는 것은 블록 차동 펄스 코드 변조에 대응하는 디코딩 프로토콜을 사용하여 제1 구역을 디코딩하는 것을 더 포함한다. 디코더는 현재 프레임을 디코딩하도록 추가로 구성될 수 있고, 현재 프레임을 디코딩하는 것은 변환 스킵 잔차 코딩을 사용하는 것에 대응하는 디코딩 프로토콜을 사용하여 제2 구역을 디코딩하는 것을 더 포함한다. 디코더는 현재 프레임을 디코딩하도록 추가로 구성될 수 있고, 현재 프레임을 디코딩하는 것은 손실 인코딩 프로토콜에 대응하는 디코딩 프로토콜을 사용하여 제3 구역을 디코딩하는 것을 더 포함한다.

[0079] 일부 실시예들에서, 결합된 무손실 및 손실 코딩의 방법은 디코더에 의해, 코딩된 비디오 비트스트림을 수신하는 단계, 디코더에 의해 비트스트림에서, 현재 프레임을 식별하는 단계 – 현재 프레임은 제1 구역, 제2 구역, 및 제3 구역을 포함함 –, 디코더에 의해 비트스트림에서, 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩되는 것을 검출하는 단계, 디코더에 의해 비트스트림에서, 제2 구역이 변환 스킵 잔차 코딩을 사용하여 인코딩되는 것을 검출하는 단계, 및 디코더에 의해 비트스트림에서, 제3 구역이 손실 인코딩을 사용하여 인코딩되는 것을 검출하는 단계 – 손실 인코딩은 인터-예측 및 인트라-예측 중 적어도 하나를 포함함 –를 포함한다.

[0080] 일부 실시예들에서, 비트스트림은 제1 구역에 대응하는 서브-픽처 헤더를 더 포함하고, 제1 구역이 블록 차동 펄스 코드 변조를 사용하여 인코딩된다는 것을 검출하는 것은 적어도 제1 구역이 서브-픽처 헤더에서 블록 차동 펄스 코드 변조를 사용하여 인코딩된다는 표시를 검출하는 것을 더 포함한다. 일부 실시예들에서, 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함된다. 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함된다. 비트스트림은 제2 구역에 대응하는 서브-픽처 헤더를 포함할 수 있고 디코더는 제2 사용 변환 스킵 잔차 코딩이 서브-픽처 헤더에서 변환 스킵 잔차 코딩을 사용하여 인코딩된다는 표시를 검출하도록 추가로 구성된다. 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함될 수 있다.

[0081] 방법은 제1 프로세서 스레드를 사용하여 제1 구역을 디코딩하는 단계 및 제2 프로세서 스레드를 사용하여 제3

구역 요소를 디코딩하는 단계를 포함할 수 있다. 방법은 현재 프레임을 디코딩하는 단계를 포함할 수 있고, 현재 프레임을 디코딩하는 단계는 블록 차동 펄스 코드 변조에 대응하는 디코딩 프로토콜을 사용하여 제1 구역을 디코딩하는 단계를 더 포함한다. 방법은 현재 프레임을 디코딩하는 단계를 포함할 수 있고, 현재 프레임을 디코딩하는 단계는 변환 스킵 잔차 코딩을 사용하는 것에 대응하는 디코딩 프로토콜을 사용하여 제2 구역을 디코딩하는 단계를 더 포함한다. 방법은 현재 프레임을 디코딩하는 단계를 포함할 수 있고, 현재 프레임을 디코딩하는 단계는 손실 인코딩 프로토콜에 대응하는 디코딩 프로토콜을 사용하여 포함하는 3 구역을 디코딩하는 단계를 더 포함한다.

[0082] 일부 실시예들에서, 디코더는 코딩된 비트스트림을 수신하고, 비트스트림에서, 현재 프레임을 식별하고 – 현재 프레임은 제1 서브-픽처 및 제2 서브-픽처를 포함함 –, 비트스트림에서 제1 서브-픽처가 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하고, 현재 프레임을 디코딩 – 현재 프레임을 디코딩하는 것은 무손실 인코딩 프로토콜에 대응하는 무손실 디코딩 프로토콜을 사용하여 제1 서브-픽처를 디코딩하는 것을 더 포함함 – 하도록 구성된 회로를 포함한다.

[0083] 일부 실시예들에서, 비트스트림은 제 1 서브-픽처에 대응하는 서브-픽처 헤더를 포함할 수 있고 검출하는 것은 적어도 제 1 서브-픽처가 서브-픽처 헤더에서 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 것을 포함할 수 있다. 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함될 수 있다. 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함된다. 디코더는 제2 서브-픽처가 손실 인코딩 프로토콜에 따라 인코딩되었음을 검출하고 무손실 인코딩 프로토콜에 대응하는 손실 디코딩 프로토콜에 따라 제2 서브-픽처를 디코딩하도록 추가로 구성될 수 있다. 비트스트림은 제2 서브-픽처에 대응하는 서브-픽처 헤더를 포함할 수 있고 디코더는 제2 서브-픽처가 서브-픽처 헤더에서 손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하도록 추가로 구성될 수 있다. 서브-픽처 헤더는 현재 프레임에 해당하는 데이터에 명시적으로 포함될 수 있다. 서브-픽처 헤더는 제 3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함될 수 있다. 디코더는 제 1 프로세서 스레드를 사용하여 제 1 서브-픽처를 디코딩하고 제 2 프로세서 스레드를 사용하여 제 2 서브-픽처 요소를 디코딩하도록 추가로 구성될 수 있다. 디코더는 비트 스트림을 수신하고 비트스트림을 양자화된 계수들로 디코딩하도록 구성된 엔트로피 디코더 프로세서, 역 이산 코사인을 수행하는 것을 포함하여 양자화된 계수들을 프로세싱하도록 구성된 역양자화 및 역변환 프로세서, 디블록킹 필터, 프레임 버퍼, 및 인트라 예측 프로세서를 포함할 수 있다. 제1 서브-픽처 및 제2 서브-픽처 중 적어도 하나는 쿼드트리 플러스 이진 결정 트리의 일부를 형성할 수 있다. 제1 서브-픽처 및 제2 서브-픽처 중 적어도 하나는 코딩 트리 단위를 포함할 수 있다. 제1 서브-픽처 및 제2 서브-픽처 중 적어도 하나는 코딩 트리 단위를 포함할 수 있다. 제1 서브-픽처 및 제2 서브-픽처 중 적어도 하나는 예측 단위를 포함할 수 있다.

[0084] 일부 실시예들에서, 결합된 무손실 및 손실 코딩의 방법은 디코더에서, 코딩된 비트스트림을 수신하는 단계, 디코더에 의해 비트스트림에서, 현재 프레임을 식별하는 단계 – 현재 프레임은 제1 서브-픽처 및 제2 서브-픽처를 포함함 –, 디코더에 의해 비트스트림에서, 제1 서브-픽처가 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 단계, 및 디코더에 의해, 현재 프레임을 디코딩하는 단계 – 현재 프레임을 디코딩하는 단계는 무손실 인코딩 프로토콜에 대응하는 무손실 디코딩 프로토콜을 사용하여 제1 서브-픽처를 디코딩하는 단계를 더 포함함 –를 포함한다.

[0085] 일부 실시예들에서, 비트스트림은 제 1 서브-픽처에 대응하는 서브-픽처 헤더를 더 포함할 수 있고 검출하는 것은 적어도 제 1 서브-픽처가 서브-픽처 헤더에서 무손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 것을 포함할 수 있다. 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함될 수 있다. 서브-픽처는 제3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함된다. 방법은 제2 서브-픽처가 손실 인코딩 프로토콜에 따라 인코딩되는 것을 검출하는 단계 및 무손실 인코딩 프로토콜에 대응하는 손실 디코딩 프로토콜에 따라 제2 서브-픽처를 디코딩하는 단계를 포함할 수 있다. 일부 실시예들에서, 비트스트림은 제 2 서브-픽처에 대응하는 서브-픽처 헤더를 포함할 수 있고 검출하는 것은 제 2 서브-픽처가 서브-픽처 헤더에서 손실 인코딩 프로토콜에 따라 인코딩된다는 표시를 검출하는 것을 포함할 수 있다. 서브-픽처 헤더는 현재 프레임에 대응하는 데이터에 명시적으로 포함될 수 있다. 서브-픽처 헤더는 제 3 서브-픽처에 대응하는 서브-픽처 헤더의 식별자를 참조하여 포함될 수 있다. 디코더는 제 1 프로세서 스레드를 사용하여 제 1 서브-픽처를 디코딩하고 제 2 프로세서 스레드를 사용하여 제 2 서브-픽처 요소를 디코딩하도록 추가로 구성될 수 있다.

[0086] 일부 실시예들에서, 디코더는 비트 스트림을 수신하고 비트스트림을 양자화된 계수들로 디코딩하도록 구성된 엔트로피 디코더 프로세서, 역 이산 코사인을 수행하는 것을 포함하여 양자화된 계수들을 프로세싱하도록 구성된 역양자화 및 역변환 프로세서, 디블록킹 필터, 프레임 버퍼, 및 인트라 예측 프로세서를 포함할 수 있다. 제1

서브-픽처 및 제2 서브-픽처 중 적어도 하나는 쿼드트리 플러스 이진 결정 트리의 일부를 형성할 수 있다. 제1 서브-픽처 및 제2 서브-픽처 중 적어도 하나는 코딩 트리 단위를 포함할 수 있다. 제1 서브-픽처 및 제2 서브-픽처 중 적어도 하나는 코딩 트리 단위를 포함할 수 있다. 제1 서브-픽처 및 제2 서브-픽처 중 적어도 하나는 예측 단위를 포함할 수 있다.

[0087] 본원에 설명된 양태들 및 실시예들 중 임의의 하나 이상이 컴퓨터 기술 분야의 통상의 기술자들에게 명백할 바와 같이, 본 명세서의 지침들에 따라 프로그래밍된 하나 이상의 기계들(예를 들어, 전자 문서에 대한 사용자 컴퓨팅 디바이스로서 활용되는 하나 이상의 컴퓨팅 디바이스들, 하나 이상의 문서 서버와 같은 서버 디바이스들 등)을 사용하여 편리하게 구현될 수 있음이 유의되어야 한다. 적절한 소프트웨어 코딩은 소프트웨어 기술 분야의 통상의 기술자들에게 명백할 바와 같이, 본 개시내용의 교시들에 기반하여 숙련된 프로그래머들에 의해 용이하게 준비될 수 있다. 소프트웨어 및/또는 소프트웨어 모듈들을 이용하여 위에서 논의된 양태들 및 구현들은 또한 소프트웨어 및/또는 소프트웨어 모듈의 기계 실행가능 명령들의 구현을 지원하기 위한 적절한 하드웨어를 포함할 수 있다.

[0088] 그러한 소프트웨어는 기계-판독가능 저장 매체를 이용하는 컴퓨터 프로그램 제품일 수 있다. 기계-판독가능 저장 매체는 기계(예를 들어, 컴퓨팅 디바이스)에 의한 실행을 위한 명령들의 시퀀스를 저장 및/또는 인코딩할 수 있고 기계가 본원에 설명된 방법론들 및/또는 실시예들 중 어느 하나를 수행하게 하는 임의의 매체일 수 있다. 기계-판독가능 저장 매체의 예들은 자기 디스크, 광 디스크(예를 들어, CD, CD-R, DVD, DVD-R 등), 광자기 디스크, 판독- 전용 메모리 "ROM" 디바이스, 랜덤-액세스 메모리 "RAM" 디바이스, 자기 카드, 광학 카드, 솔리드-스테이트 메모리 디바이스, EPROM, EEPROM 및 이들의 임의의 조합을 포함하지만, 이에 제한되지 않는다. 본원에서 사용되는 기계-판독가능 매체는 단일 매체뿐만 아니라 물리적으로 분리된 매체의 집합, 이를테면, 예를 들어 컴퓨터 메모리와 결합된 콤팩트 디스크 또는 하나 이상의 하드 디스크 드라이브의 집합을 포함하도록 의도된다. 본원에서 사용되는 바와 같이, 기계-판독가능 저장 매체는 일시적인 신호 송신 형태들을 포함하지 않는다.

[0089] 그러한 소프트웨어는 또한 반송파와 같은 데이터 반송파 상에서 데이터 신호로서 운반되는 정보(예를 들어, 데이터)를 포함할 수 있다. 예를 들어, 기계-실행가능 정보는 신호가 기계(예를 들어, 컴퓨팅 디바이스)에 의한 실행을 위한 명령의 시퀀스, 또는 이의 일부 및 기계로 하여금 본원에 설명된 방법론들 및/또는 실시예들 중 어느 하나를 실행하게 하는 임의의 관련 정보(예를 들어, 데이터 구조들 및 데이터)를 인코딩하는 데이터 반송파에 구현된 데이터-운반 신호로서 포함될 수 있다.

[0090] 컴퓨팅 디바이스의 예들은 전자 책 읽기 디바이스, 컴퓨터 워크스테이션, 단말 컴퓨터, 서버 컴퓨터, 핸드헬드 디바이스(예를 들어, 태블릿 컴퓨터, 스마트폰 등), 웹 기기, 네트워크 라우터, 네트워크 스위치, 네트워크 브리지, 해당 기계가 취해야 할 동작을 지정하는 일련의 명령들을 실행할 수 있는 임의의 기계 및 이들의 임의의 조합들을 포함하지만, 이에 제한되지 않는다. 일 예에서, 컴퓨팅 디바이스는 키오스크를 포함 및/또는 키오스크에 포함될 수 있다.

[0091] 도 6은 컴퓨터 시스템(600)의 예시적인 형태의 컴퓨팅 디바이스의 일 실시예의 개략도를 도시하고, 컴퓨터 시스템(600)에, 제어 시스템이 본 개시내용의 양태들 및/또는 방법론들 중 임의의 하나 이상을 수행하게 하기 위한 명령들의 세트가 실행될 수 있다. 또한, 디바이스들 중 하나 이상이 본 개시내용의 양태들 및/또는 방법론들 중 임의의 하나 이상을 수행하게 하기 위해 특별히 구성된 명령들의 세트를 구현하기 위해 다수의 컴퓨팅 디바이스가 활용될 수 있음이 고려된다. 컴퓨터 시스템(600)은 버스(612)를 통해 서로 통신하고 다른 구성요소들과 통신하는 프로세서(604) 및 메모리(608)를 포함한다. 버스(612)는 다양한 버스 아키텍처들 중 임의의 것을 사용하여 메모리 버스, 메모리 제어기, 주변 버스, 로컬 버스, 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는 임의의 여러 유형들의 버스 구조들을 포함할 수 있다.

[0092] 프로세서(604)는 제한 없이 상태 머신으로 조절되고 메모리 및/또는 센서들로부터의 연산 입력들에 의해 지시받을 수 있는 산술 및 논리 유닛(ALU)과 같은 산술 및 논리 연산들을 수행하기 위한 논리 회로를 통합하는 프로세서와 같은 임의의 적합한 프로세서를 포함할 수 있고; 프로세서(604)는 비제한적인 예로서 폰 노이만 및/또는 하버드 아키텍처에 따라 구성될 수 있다. 프로세서(604)는 제한 없이 마이크로제어기, 마이크로프로세서, 디지털 신호 프로세서(DSP), FPGA(Field Programmable Gate Array), CPLD(Complex Programmable Logic Device), GPU(Graphical Processing Unit), 범용 GPU, TPU(Tensor Processing Unit), 아날로그 또는 혼합 신호 프로세서, TPM(Trusted Platform Module), FPU(floating-point unit) 및/또는 SoC(system on a chip)를 포함할 수 있고/있거나 이에 통합될 수 있다.

[0093] 메모리(608)는 랜덤-액세스 메모리 구성요소, 판독 전용 구성요소, 및 이들의 임의의 조합들을 포함하지만 이에

제한되지 않는 다양한 구성요소들(예를 들어, 기계-판독가능 매체)를 포함할 수 있다. 일 예에서, 시작 동안과 같이 컴퓨터 시스템(600) 내의 요소들 간의 정보 전송을 돕는 기본 루틴들을 포함하는 기본 입/출력 시스템(616)(BIOS)은 메모리(608)에 저장될 수 있다. 메모리(608)는 또한 본 개시내용의 양태들 및/또는 방법론들 중 임의의 하나 이상을 구현하는 (예를 들어, 하나 이상의 기계-판독가능 매체에 저장된) 명령들(예를 들어, 소프트웨어)(620)을 포함할 수 있다. 다른 예에서, 메모리(608)는 운영 체제, 하나 이상의 애플리케이션 프로그램들, 다른 프로그램 모듈들, 프로그램 데이터, 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는 임의의 수의 프로그램 모듈들을 더 포함할 수 있다.

[0094] 컴퓨터 시스템(600)은 또한 저장 디바이스(624)를 포함할 수 있다. 저장 디바이스(예를 들어, 저장 디바이스(624))의 예들은 하드 디스크 드라이브, 자기 디스크 드라이브, 광학 매체와 결합된 광학 디스크 드라이브, 솔리드-스테이트 메모리 디바이스, 및 임의의 이들의 조합들을 포함하지만, 이에 제한되지 않는다. 저장 디바이스(624)는 적절한 인터페이스(도시되지 않음)에 의해 버스(612)에 연결될 수 있다. 예시적인 인터페이스들은 SCSI, ATA(Advanced Technology Attachment), 직렬 ATA, USB(Universal Serial Bus), IEEE 1394(FIREWIRE) 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는다. 일 예에서, 저장 디바이스(624)(또는 이들의 하나 이상의 구성요소)는 (예를 들어, 외부 포트 커넥터(도시되지 않음)를 통해) 컴퓨터 시스템(600)과 제거 가능하게 인터페이스될 수 있다. 특히, 저장 디바이스(624) 및 연관된 기계-판독가능 매체(628)는 기계-판독가능 명령들, 데이터 구조들, 프로그램 모듈들, 및/또는 컴퓨터 시스템(600)을 위한 다른 데이터의 비휘발성 및/또는 휘발성 저장을 제공할 수 있다. 일 예에서, 소프트웨어(620)는 기계-판독가능 매체(628) 내에 완전히 또는 부분적으로 상주할 수 있다. 다른 예에서, 소프트웨어(620)는 프로세서(604) 내에 완전히 또는 부분적으로 상주할 수 있다.

[0095] 컴퓨터 시스템(600)은 또한 입력 디바이스(632)를 포함할 수 있다. 일 예에서, 컴퓨터 시스템(600)의 사용자는 입력 디바이스(632)를 통해 커맨드들 및/또는 다른 정보를 컴퓨터 시스템(600)에 입력할 수 있다. 입력 디바이스(632)의 예들은 문자숫자식 입력 디바이스(예를 들어, 키보드), 포인팅 디바이스, 조이스틱, 게임패드, 오디오 입력 디바이스(예를 들어, 마이크로폰, 음성 응답 시스템 등), 커서 제어 디바이스(예를 들어, 마우스), 터치패드, 광학 스캐너, 비디오 캡처 디바이스(예를 들어, 스틸 카메라, 비디오 카메라), 터치스크린, 및 이들의 임의의 조합들을 포함하지만, 이에 제한되지 않는다. 입력 디바이스(632)는 직렬 인터페이스, 병렬 인터페이스, 게임 포트, USB 인터페이스, FIREWIRE 인터페이스, 버스(612)에 대한 직접 인터페이스를 포함하지만 이에 제한되지 않는 다양한 인터페이스들(도시되지 않음), 및 이들의 임의의 조합들 중 임의의 것을 통해 버스(612)에 인터페이스될 수 있다. 입력 디바이스(632)는 아래에서 더 논의되는 디스플레이(636)의 일부이거나 디스플레이(636)와 별개일 수 있는 터치 스크린 인터페이스를 포함할 수 있다. 입력 디바이스(632)는 전술된 바와 같이 그래픽 인터페이스에서 하나 이상의 그래픽 표현들을 선택하기 위한 사용자 선택 디바이스로서 활용될 수 있다.

[0096] 사용자는 또한 저장 디바이스(624)(예를 들어, 이동식 디스크 드라이브, 플래시 드라이브 등) 및/또는 네트워크 인터페이스 디바이스(640)를 통해 컴퓨터 시스템(600)에 커맨드들 및/또는 다른 정보를 입력할 수 있다. 네트워크 인터페이스 디바이스(640)와 같은 네트워크 인터페이스 디바이스는 컴퓨터 시스템(600)을 네트워크(644) 및 이에 연결된 하나 이상의 원격 디바이스들(648)과 같은 다양한 네트워크들 중 하나 이상에 연결하기 위해 활용될 수 있다. 네트워크 인터페이스 디바이스의 예들은 네트워크 인터페이스 카드(예를 들어, 모바일 네트워크 인터페이스 카드, LAN 카드), 모뎀, 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는다. 네트워크의 예들은 광역 네트워크(예를 들어, 인터넷, 기업 네트워크), 근거리 통신망(예를 들어, 사무실, 건물, 캠퍼스 또는 다른 비교적 소규모 지리적 공간), 전화 네트워크, 전화/음성 제공자와 연관된 데이터 네트워크(예를 들어, 모바일 통신 제공자 데이터 및/또는 음성 네트워크), 2 개의 컴퓨팅 디바이스들 간의 직접 연결, 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는다. 네트워크(644)와 같은 네트워크는 유선 및/또는 무선 통신 모드를 이용할 수 있다. 일반적으로, 모든 네트워크 토폴로지가 사용될 수 있다. 정보(예를 들어, 데이터, 소프트웨어(620) 등)는 네트워크 인터페이스 디바이스(640)를 통해 컴퓨터 시스템(600)으로 및/또는 컴퓨터 시스템(600)으로부터 통신될 수 있다.

[0097] 컴퓨터 시스템(600)은 디스플레이 디바이스(636)와 같은 디스플레이 디바이스에 디스플레이가능 이미지를 통신하기 위한 비디오 디스플레이 어댑터(652)를 더 포함할 수 있다. 디스플레이 디바이스의 예들은 액정 디스플레이(LCD), 음극선관(CRT), 플라즈마 디스플레이, 발광 다이오드(LED) 디스플레이, 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는다. 디스플레이 어댑터(652) 및 디스플레이 디바이스(636)는 프로세서(604)와 조합하여 이용되어 본 개시내용의 양태들의 그래픽 표현들을 제공할 수 있다. 디스플레이 디바이스에 추가하여,

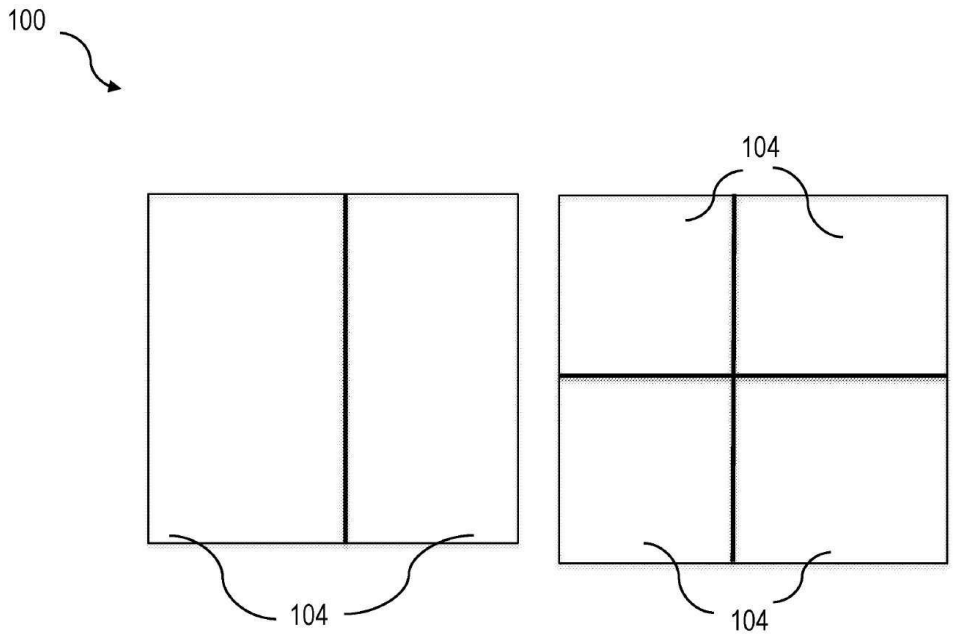
컴퓨터 시스템(600)은 오디오 스피커, 프린터, 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는 하나 이상의 다른 주변 출력 디바이스들을 포함할 수 있다. 이러한 주변 출력 디바이스들은 주변 인터페이스(656)를 통해 버스(612)에 연결될 수 있다. 주변 인터페이스의 예들은 직렬 포트, USB 연결, FIREWIRE 연결, 병렬 연결, 및 이들의 임의의 조합들을 포함하지만 이에 제한되지 않는다.

[0098] 이상은 본 발명의 예시적인 실시예들의 상세한 설명이었다. 본 발명의 사상 및 범위를 벗어나지 않고 다양한 수정들 및 추가들이 이루어질 수 있다. 전술된 다양한 실시예들 각각의 특징들은 연관된 새로운 실시예들에서 다수의 특징 조합들을 제공하기 위해 적절하게 다른 설명된 실시예들의 특징들과 결합될 수 있다. 또한, 전술한 내용이 다수의 개별 실시예들 설명하지만, 본원에 설명된 것은 단지 본 발명의 원리들의 적용을 예시한다. 추가로, 본원의 특정 방법들이 특정 순서로 수행되는 것으로 예시 및/또는 설명될 수 있지만, 순서는 본 개시내용에 따른 방법들, 시스템 및 소프트웨어를 달성하기 위해 통상의 기술 내에서 매우 가변적이다. 따라서, 이 설명은 단지 예로서 취해져야 하고, 본 발명의 범위를 달리 제한하지 않아야 한다.

[0099] 예시적인 실시예들은 위에 개시되었고 첨부 도면에 예시되었다. 통상의 기술자가 본 발명의 사상 및 범위를 벗어나지 않고 본원에 구체적으로 개시된 것에 다양한 변경들, 생략들 및 추가들이 이루어질 수 있음을 이해할 것이다.

도면

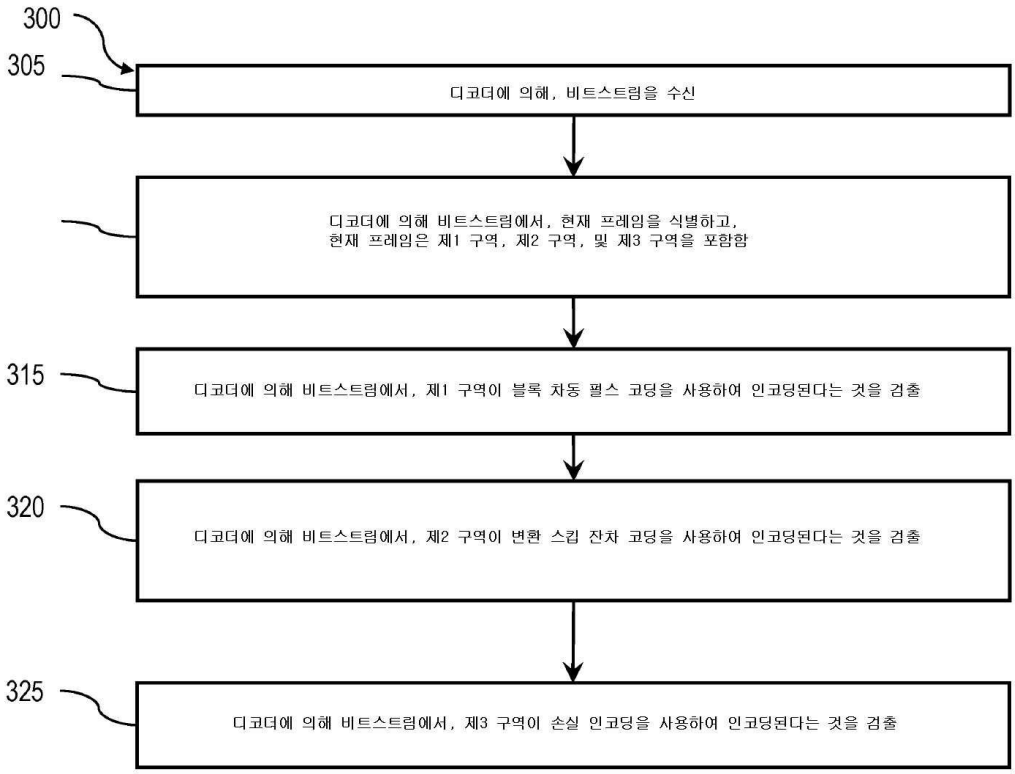
도면1



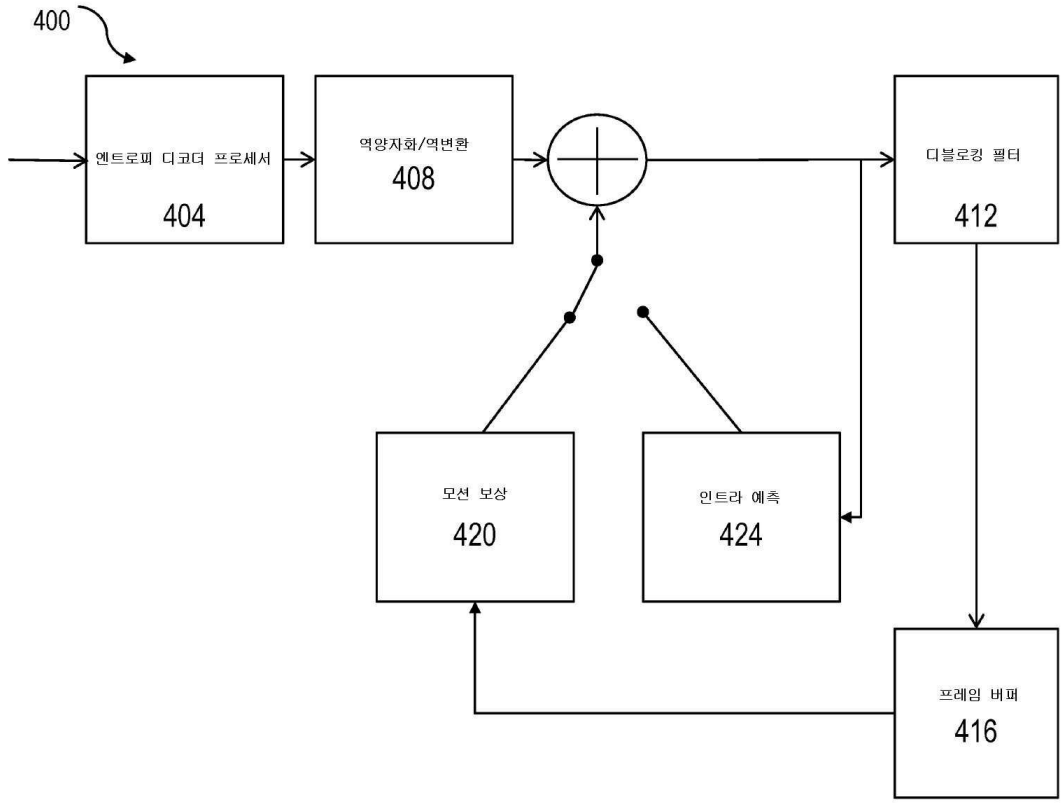
도면2


200

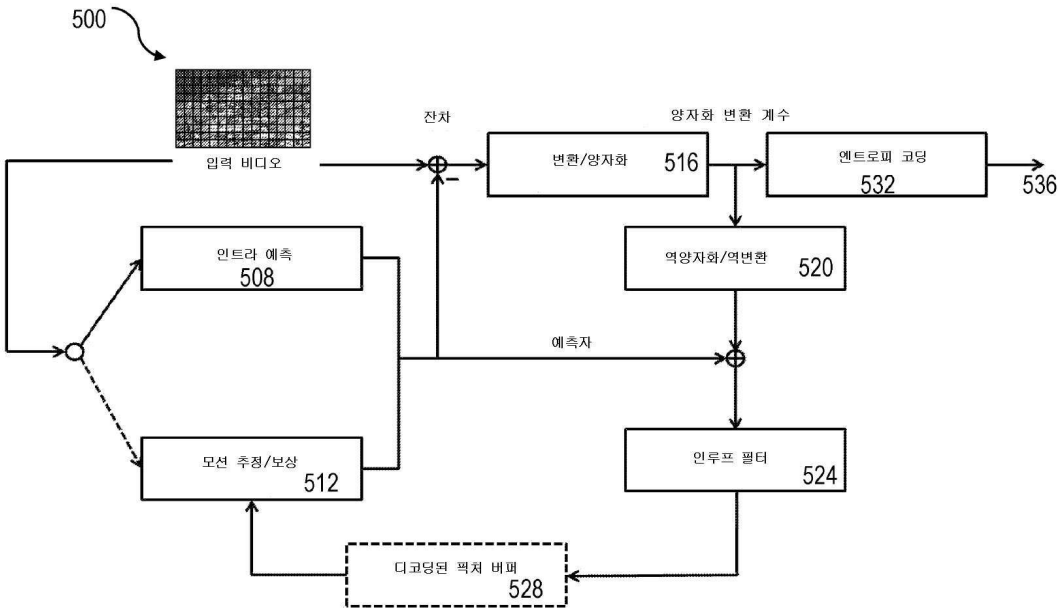
도면3


300
305
디코더에 의해, 비트스트림을 수신
디코더에 의해 비트스트림에서, 현재 프레임을 식별하고,
현재 프레임은 제1 구역, 제2 구역, 및 제3 구역을 포함함
315
디코더에 의해 비트스트림에서, 제1 구역이 블록 차동 펄스 코딩을 사용하여 인코딩된다는 것을 검출
320
디코더에 의해 비트스트림에서, 제2 구역이 변환 스킵 잔차 코딩을 사용하여 인코딩된다는 것을 검출
325
디코더에 의해 비트스트림에서, 제3 구역이 손실 인코딩을 사용하여 인코딩된다는 것을 검출

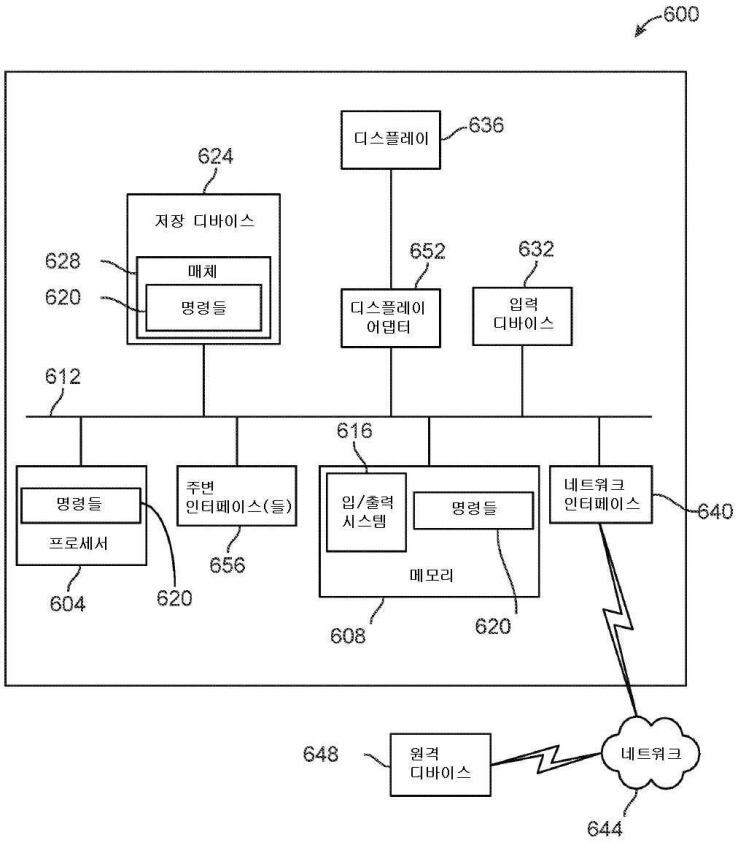
도면4


400
엔트로피 디코더 프로세서
404
역양자화/역변환
408
디블로킹 필터
412
모션 보상
420
인트라 예측
424
프레임 버퍼
416

도면5


500
입력 비디오
잔차
양자화 변환 계수
변환/양자화
516
엔트로피 코딩
532
536
인트라 예측
508
역양자화/역변환
520
예측자
모션 추정/보상
512
인루프 필터
524
디코딩된 픽처 버퍼
528

도면6


600
디스플레이
636
624
저장 디바이스
628
매체
620
명령들
652
632
디스플레이 어댑터
입력 디바이스
612
616
명령들
프로세서
604
620
주변 인터페이스(들)
656
입/출력 시스템
명령들
메모리
608
620
네트워크 인터페이스
640
648
원격 디바이스
네트워크
644