(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년08월26일
(11) 등록번호 10-1547743
(24) 등록일자 2015년08월20일

(51) 국제특허분류(Int. Cl.)
H04N 19/00 (2014.01)
(21) 출원번호 10-2013-7020603
(22) 출원일자(국제) 2012년01월03일
심사청구일자 2013년08월02일
(85) 번역문제출일자 2013년08월02일
(65) 공개번호 10-2013-0095324
(43) 공개일자 2013년08월27일
(86) 국제출원번호 PCT/US2012/020108
(87) 국제공개번호 WO 2012/094342
국제공개일자 2012년07월12일
(30) 우선권주장
13/341,368 2011년12월30일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
US20060093034 A1
Sjaberg R et al: "Fine granularity slices",JCTVC-C154,2010.10.2.

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
천 잉
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
천 페이송
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인코리아나

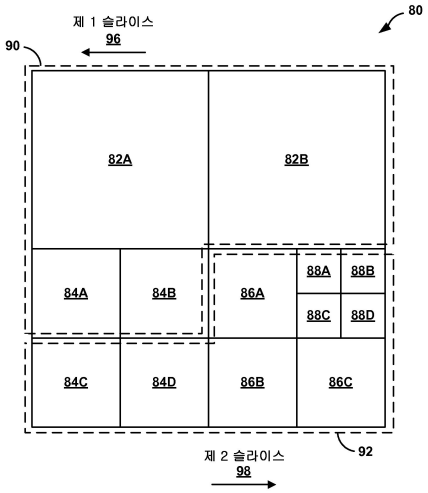
전체 청구항 수 : 총 60 항 심사관 : 김영태

(54) 발명의 명칭 **비디오 코딩에서의 프레임 분할**

(57) 요 약

하나의 예에서, 본 개시물은 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들(LCUs)을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 방법을 설명한다. 이 예에서, 본 방법은 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도(granularity)를 결정하는 단계를 포함한다. 그 방법은 또한 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하는 단계를 포함한다. 그 방법은 또한 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 가지지 않는, 프레임의 독립적으로 디코딩가능한 부분을 디코딩하는 단계를 포함한다.

대 표 도 - 도3b



(30) 우선권주장
61/430,104 2011년01월05일 미국(US)
61/435,098 2011년01월21일 미국(US)
61/454,166 2011년03월18일 미국(US)
61/492,751 2011년06월02일 미국(US)

특허청구의 범위

청구항 1

하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 방법으로서,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 표시하는 비트스트림으로부터의 하나 이상의 신택스 엘리먼트를 디코딩하는 단계;

상기 하나 이상의 신택스 엘리먼트에 기초하여 상기 세분도 (granularity) 를 결정하는 단계;

상기 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하는 단계; 및

상기 독립적으로 디코딩가능한 부분은 상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션이 없는, 상기 비트 스트림으로부터의 상기 프레임의 독립적으로 디코딩가능한 부분을 디코딩하는 단계를 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 세분도를 결정하는 단계는 상기 하나 이상의 최대 코딩 단위의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 코딩 단위 (coding unit; CU) 깊이를 결정하는 단계를 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 CU 깊이를 결정하는 단계는, 화상 파라미터 세트에서의 CU 깊이 값을 디코딩하는 단계를 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 LCU의 상기 제 1 섹션의 어드레스를 결정하는 단계를 더 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 5

제 4 항에 있어서,

상기 LCU의 상기 제 1 섹션의 상기 어드레스를 결정하는 단계는 슬라이스 헤더의 슬라이스 어드레스를 디코딩하는 단계를 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고;

상기 방법은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 디코딩하는 단계; 및

상기 제 1 독립적으로 디코딩가능한 부분을 갖는 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 디코딩하는 단계; 및

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 디코딩하는 단계를 더 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 7

제 6 항에 있어서,

상기 쿼드트리 구조의 제 1 부분을 디코딩하는 단계는,

상기 제 1 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리 (division) 를 표시하는 하나 이상의 분할 플래그들을 디코딩하는 단계; 및

상기 제 2 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 디코딩하는 단계를 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고,

상기 방법은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 디코딩하는 단계;

상기 제 1 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 식별하는 단계; 및

상기 제 1 독립적으로 디코딩가능한 부분과는 별개로, 상기 제 2 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 식별하는 단계를 더 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 독립적으로 디코딩가능한 부분의 말단의 표시를 디코딩하는 단계를 더 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 10

하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 장치로서,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 표시하는 비트스트림으로부터의 하나 이상의 신택스 엘리먼트를 디코딩하고,

상기 하나 이상의 신택스 엘리먼트에 기초하여 상기 세분도를 결정하고;

상기 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하며; 및

상기 독립적으로 디코딩가능한 부분은 상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션이 없는, 상기 비트 스트림으로부터의 상기 프레임의 독립적으로 디코딩가능한 부분을 디코딩하도록 구성된 하나 이상의 프로세서들을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 11

제 10 항에 있어서,

상기 세분도를 결정하는 것은 상기 하나 이상의 LCU 의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 코딩 단위 (CU) 깊이를 결정하는 것을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 12

제 11 항에 있어서,

상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 CU 깊이를 결정하는 것은, 화상 파라미터 세트에서의 CU 깊이 값을 디코딩하는 것을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 13

제 10 항에 있어서,

상기 하나 이상의 프로세서들은 상기 LCU의 상기 제 1 섹션의 어드레스를 결정하도록 더 구성되는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 14

제 13 항에 있어서,

상기 LCU의 상기 제 1 섹션의 상기 어드레스를 결정하는 것은 슬라이스 헤더의 슬라이스 어드레스를 디코딩하는 것을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 15

제 10 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고,

상기 하나 이상의 프로세서들은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 디코딩하고;

상기 제 1 독립적으로 디코딩가능한 부분을 갖는, 상기 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 디코딩하며; 그리고

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 디코딩하도록 더 구성되는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 16

제 15 항에 있어서,

상기 쿼드트리 구조의 제 1 부분을 디코딩하는 것은,

상기 제 1 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 디코딩하는 것; 및

상기 제 2 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 디코딩하는 것을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 17

제 10 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고,

상기 하나 이상의 프로세서들은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 디코딩하고;

상기 제 1 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 식별하며; 그리고

상기 제 1 독립적으로 디코딩가능한 부분과는 별개로, 상기 제 2 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 식별하도록 더 구성되는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 18

제 10 항에 있어서,

상기 하나 이상의 프로세서들은 상기 독립적으로 디코딩가능한 부분의 말단의 표시를 디코딩하도록 더

구성되는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 19

제 10 항에 있어서,

상기 장치는 모바일 디바이스를 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 20

하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 장치로서,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 표시하는 비트스트림으로부터의 하나 이상의 신택스 엘리먼트를 디코딩하는 수단;

상기 하나 이상의 신택스 엘리먼트에 기초하여 상기 세분도를 결정하는 수단;

상기 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하는 수단; 및

상기 독립적으로 디코딩가능한 부분은 상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션이 없는, 상기 비트 스트림으로부터의 상기 프레임의 독립적으로 디코딩가능한 부분을 디코딩하는 수단을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 21

제 20 항에 있어서,

상기 세분도를 결정하는 것은 상기 하나 이상의 LCU 의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 코딩 단위 (CU) 깊이를 결정하는 것을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 22

제 21 항에 있어서,

상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 CU 깊이를 결정하는 것은, 화상 파라미터 세트에서의 CU 깊이 값을 디코딩하는 것을 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 23

제 20 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고,

상기 장치는,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 디코딩하는 수단;

상기 제 1 독립적으로 디코딩가능한 부분을 갖는, 상기 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 디코딩하는 수단; 및

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 디코딩하는 수단을 더 포함하는, 비디오 데이터의 프레임을 디코딩하는 장치.

청구항 24

하나 이상의 프로세서들에 의한 실행 시, 상기 하나 이상의 프로세서들로 하여금, 하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 방법을 수행하게 하는 명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 방법은,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 표시하는 비트스트림으로부터의 하나 이상의 신택스 엘리먼트를 디코딩하는 단계;

상기 하나 이상의 신택스 엘리먼트에 기초하여 상기 세분도 (granularity) 를 결정하는 단계;

상기 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하는 단계; 및

상기 독립적으로 디코딩가능한 부분은 상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션이 없는, 상기 비트 스트림으로부터의 상기 프레임의 독립적으로 디코딩가능한 부분을 디코딩하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 25

제 24 항에 있어서,

상기 세분도를 결정하는 단계는 상기 하나 이상의 LCU 의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 코딩 단위 (CU) 깊이를 결정하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 26

제 25 항에 있어서,

상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 CU 깊이를 결정하는 단계는, 화상 파라미터 세트에서의 CU 깊이 값을 디코딩하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 27

제 24 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고; 상기 방법은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 디코딩하는 단계; 및

상기 제 1 독립적으로 디코딩가능한 부분을 갖는 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 디코딩하는 단계; 및

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 디코딩하는 단계를 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 28

하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 방법으로서,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하는 단계;

상기 LCU의 제 1 섹션 및 상기 LCU의 제 2 섹션을 생성하기 위해 상기 결정된 세분도를 이용하여 LCU를 분할하는 단계;

상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션을 포함하지 않는, 상기 프레임의 독립적으로 디코딩가능한 부분을 생성하는 단계; 및

상기 결정된 세분도의 표시 및 상기 프레임의 상기 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 29

제 28 항에 있어서,

상기 세분도를 결정하는 단계는 상기 하나 이상의 LCU 의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 코딩 단위 (CU) 깊이를 결정하는 단계를 포함하고;

상기 비트스트림을 생성하는 단계는 CU 깊이 값을 포함하는 상기 비트스트림을 생성하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 30

제 29 항에 있어서,

상기 결정된 세분도의 상기 표시를 포함하는 상기 비트스트림을 생성하는 단계는, 화상 파라미터 세트에 상기 CU 깊이 값을 포함하는 비트스트림을 생성하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 31

제 28 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고;

상기 방법은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 생성하는 단계;

상기 제 1 독립적으로 디코딩가능한 부분을 갖는 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 표시하는 단계; 및

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 표시하는 단계를 더 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 32

제 31 항에 있어서,

상기 쿼드트리 구조의 제 1 부분을 표시하는 단계는,

상기 제 1 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 단계; 및

상기 제 2 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 33

제 28 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고;

상기 방법은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 생성하는 단계;

상기 제 1 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 표시하는 단계; 및

상기 제 1 독립적으로 디코딩가능한 부분과는 별개로, 상기 제 2 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 표시하는 단계를 더 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 34

제 28 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 단계는 상기 독립적으로 디코딩가능한 부분의 말단의 표시를 생성하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 35

제 34 항에 있어서,

상기 독립적으로 디코딩가능한 부분의 말단의 표시를 생성하는 단계는 상기 독립적으로 디코딩가능한 부분의 상기 말단을 식별하는 1 비트 플래그를 생성하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 36

제 35 항에 있어서,

상기 1 비트 플래그는 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할되는 상기 세분도보다 작은 세분도를 갖는 코딩 단위들에 대해 생성되지 않는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 37

하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 장치로서,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하고;

상기 LCU의 제 1 섹션 및 상기 LCU의 제 2 섹션을 생성하기 위해 상기 결정된 세분도를 이용하여 LCU를 분할하고;

상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션을 포함하지 않는, 상기 프레임의 독립적으로 디코딩가능한 부분을 생성하며; 그리고

상기 결정된 세분도의 표시 및 상기 프레임의 상기 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하도록 구성된 하나 이상의 프로세서들을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 38

제 37 항에 있어서,

상기 세분도를 결정하는 것은 상기 하나 이상의 LCU 의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 코딩 단위 (CU) 깊이를 결정하는 것을 포함하고;

상기 비트스트림을 생성하는 것은 CU 깊이 값을 포함하는 상기 비트스트림을 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 39

제 38 항에 있어서,

상기 결정된 세분도의 상기 표시를 포함하는 상기 비트스트림을 생성하는 것은, 화상 파라미터 세트에 상기 CU 깊이 값을 포함하는 비트스트림을 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 40

제 37 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고,

상기 하나 이상의 프로세서들은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 생성하고;

상기 제 1 독립적으로 디코딩가능한 부분을 갖는, 상기 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 표시하며; 그리고

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 표시하도록 더 구성되는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 41

제 40 항에 있어서,

상기 쿼드트리 구조의 제 1 부분을 표시하는 것은,

상기 제 1 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 것; 및

상기 제 2 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 42

제 37 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고,

상기 하나 이상의 프로세서들은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 생성하고;

상기 제 1 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 표시하며; 및

상기 제 1 독립적으로 디코딩가능한 부분과는 별개로, 상기 제 2 독립적으로 디코딩가능한 부분에 대한 양자화 파라미터에서의 변화를 표시하도록 더 구성되는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 43

제 37 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 것은 상기 독립적으로 디코딩가능한 부분의 말단의 표시를 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 44

제 43 항에 있어서,

상기 독립적으로 디코딩가능한 부분의 말단의 표시를 생성하는 것은 상기 독립적으로 디코딩가능한 부분의 상기 말단을 식별하는 1 비트 플래그를 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 45

제 44 항에 있어서,

상기 1 비트 플래그는 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할되는 상기 세분도보다 작은 세분도를 갖는 코딩 단위들에 대해 생성되지 않는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 46

제 37 항에 있어서,

상기 장치는 모바일 디바이스를 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 47

하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 장치로서,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하는 수단;

상기 LCU의 제 1 섹션 및 상기 LCU의 제 2 섹션을 생성하기 위해 상기 결정된 세분도를 이용하여 LCU를 분할하

는 수단;

상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션을 포함하지 않는, 상기 프레임의 독립적으로 디코딩가능한 부분을 생성하는 수단; 및

상기 결정된 세분도의 표시 및 상기 프레임의 상기 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 수단을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 48

제 47 항에 있어서,

상기 세분도를 결정하는 것은 상기 하나 이상의 LCU 의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 코딩 단위 (CU) 깊이를 결정하는 것을 포함하며; 그리고

상기 비트스트림을 생성하는 것은 CU 깊이 값을 포함하는 상기 비트스트림을 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 49

제 48 항에 있어서,

상기 결정된 세분도의 상기 표시를 포함하는 상기 비트스트림을 생성하는 것은, 화상 파라미터 세트에 상기 CU 깊이 값을 포함하는 비트스트림을 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 50

제 47 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하고; 상기 장치는,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 생성하는 수단;

상기 제 1 독립적으로 디코딩가능한 부분을 갖는 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 표시하는 수단; 및

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 표시하는 수단 을 더 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 51

제 50 항에 있어서,

상기 쿼드트리 구조의 상기 제 1 부분을 표시하는 것은,

상기 제 1 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 것; 및

상기 제 2 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 것을 포함하는, 비디오 데이터의 프레임을 인코딩하는 장치.

청구항 52

하나 이상의 프로세서들에 의한 실행 시, 상기 하나 이상의 프로세서들로 하여금, 하나 이상의 최대 코딩 단위 (largest coding units; LCUs) 에 비해, 계층적으로 배열된 복수의 작은 코딩 단위들을 포함하는 상기 하나 이상의 LCU를 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 방법을 수행하게 하는 명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 방법은,

상기 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하는 단계;

상기 LCU의 제 1 섹션 및 상기 LCU의 제 2 섹션을 생성하기 위해 상기 결정된 세분도를 이용하여 LCU를 분할하는 단계;

상기 LCU의 상기 제 1 섹션을 포함하고 상기 LCU의 상기 제 2 섹션을 포함하지 않는, 상기 프레임의 독립적으로 디코딩가능한 부분을 생성하는 단계; 및

상기 결정된 세분도의 표시 및 상기 프레임의 상기 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 53

제 52 항에 있어서,

상기 세분도를 결정하는 단계는 상기 하나 이상의 LCU 의 상기 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 CU 깊이를 결정하는 단계를 포함하고;

상기 비트스트림을 생성하는 단계는 코딩 단위 (CU) 깊이 값을 포함하는 상기 비트스트림을 생성하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 54

제 53 항에 있어서,

상기 결정된 세분도의 상기 표시를 포함하는 상기 비트스트림을 생성하는 단계는, 화상 파라미터 세트에 상기 CU 깊이 값을 포함하는 비트스트림을 생성하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 55

제 52 항에 있어서,

상기 프레임의 상기 독립적으로 디코딩가능한 부분은 제 1 독립적으로 디코딩가능한 부분을 포함하며,

상기 방법은,

상기 LCU의 상기 제 2 섹션을 포함하는, 상기 프레임의 제 2 독립적으로 디코딩가능한 부분을 생성하는 단계;

상기 제 1 독립적으로 디코딩가능한 부분을 갖는, 상기 계층적 배열의 작은 코딩 단위들을 식별하는 쿼드트리 구조의 제 1 부분을 표시하는 단계; 및

상기 제 2 독립적으로 디코딩가능한 부분을 갖는, 상기 쿼드트리 구조의 상기 제 1 부분과는 별개인 상기 쿼드트리 구조의 제 2 부분을 표시하는 단계를 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 56

제 55 항에 있어서,

상기 쿼드트리 구조의 제 1 부분을 표시하는 단계는,

상기 제 1 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 단계; 및

상기 제 2 독립적으로 디코딩가능한 부분 내의 코딩 단위 분리를 표시하는 하나 이상의 분할 플래그들을 생성하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 57

제 1항에 있어서,

상기 세분도를 결정하는 단계는 하나 이상의 신택스 엘리먼트에 기초하여 하나의 세분도 보다 많은 세분도들로부터 세분도를 선택하는 단계를 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 58

제 57항에 있어서,

상기 세분도를 선택하는 단계는 미리 결정된 사이즈의 독립적으로 디코딩가능한 부분들을 달성하는 세분도를 선택하는 단계를 포함하는, 비디오 데이터의 프레임을 디코딩하는 방법.

청구항 59

제 28항에 있어서,

상기 세분도를 결정하는 단계는 하나의 세분도 보다 많은 세분도들로부터 세분도를 선택하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

청구항 60

제 59항에 있어서,

상기 세분도를 결정하는 단계는 미리 결정된 사이즈의 독립적으로 디코딩가능한 부분들을 달성하는 세분도를 결정하는 단계를 포함하는, 비디오 데이터의 프레임을 인코딩하는 방법.

명 세 서

기 술 분 야

[0001] 본 출원은 2011년 1월 5일자로 출원된 미국 가출원 제61/430,104호, 2011년 1월 21일로 출원된 미국 가출원 제61/435,098호, 2011년 3월 18일자로 출원된 미국 가출원 제61/454,166호, 및 2011년 6월 2일자로 출원된 미국 가출원 제61/492,751호를 우선권 주장하며, 그것들의 전체 내용들은 참조로 본원에 통합된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 코딩 기법들에 관한 것이고, 더 상세하게는, 비디오 코딩 기법들의 프레임 분할 양태들에 관한 것이다.

배 경 기 술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인휴대 정보단말들 (PDAs), 랩톱 또는 데스크톱 컴퓨터들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 비디오 원격회의 디바이스들 등을 포함하는 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (Advanced Video Coding; AVC) 에 의해 규정된 표준들 및 이러한 표준들의 확장안들에 기재된 것들과 같은 비디오 압축 기법들을 구현하여, 디지털 비디오 정보를 더 효율적으로 송신하고 수신한다. 새로운 비디오 코딩 표준들, 이를테면 MPEG 및 ITU-T 간의 협력체인 JCT-VC (Joint Collaborative Team - Video Coding) 에 의해 개발 중인 고 효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준이 개발 중에 있다. 신흥 (emerging) HEVC 표준은 때때로 H.265라고 지칭되지만, 이러한 지정은 공식적으로 이루어진 것은 아니다.

발명의 내용

과제의 해결 수단

[0005] 본 개시물은 비디오 데이터의 프레임을 때때로 슬라이스들이라고 지칭되는, 프레임의 독립적으로 디코딩가능한 부분들로 분할하는 기법들을 설명한다. 신흥 HEVC 표준에 부합하게, 비디오 데이터의 블록은 코딩 단위 (coding unit; CU) 라고 지칭될 수도 있다. CU는 계층적 쿼드트리 구조에 따라 서브 CU들로 분할될 수도 있다. 예를 들어, 비트스트림 내의 신택스 데이터는 화소들의 수의 측면에서 비디오 데이터의 프레임의 가장 큰 코딩 단위인 최대 코딩 단위 (largest coding unit; LCU) 를 정의할 수도 있다. LCU는 서브 CU들로 분할될 수도 있고, 각각의 서브 CU는 서브 CU들로 더 분할될 수도 있다. 비트스트림에 대한 신택스 데이터는 최대 CU 깊이라고 지칭되는, LCU가 분할될 수도 있는 최대 횟수를 정의할 수도 있다.

[0006] 일반적으로, 비디오 데이터의 프레임을, 신흥 HEVC 표준에서 "슬라이스들"이라고 지칭되는, 그 프레임의 독립적으로 디코딩가능한 부분들로 분할하는 기법들이 설명된다. 이들 슬라이스들의 콘텐츠를 하나 이상의 완전한 코딩 단위들 (CUs), 이를테면 프레임의 하나 이상의 완전한 최대 코딩 단위들 (LCUs) 로 제한하는 대신, 본 개

시물에서 설명된 기법들은 슬라이스들이 LCU의 일 부분을 포함할 수도 있는 방법을 제공할 수도 있다. LCU가 2 개의 섹션들로 분리되는 것을 가능하게 함에 있어서, 그 기법들은 임의의 주어진 프레임을 분할하는 경우에 필요한 슬라이스들의 수를 감소시킬 수도 있다. 슬라이스들의 수를 감소시키면, 압축된 비디오 데이터를 디코딩하는데 이용된 신택스 엘리먼트들을 저장하는 슬라이스 헤더 데이터의 형태의 오버헤드 데이터를 감소시키며, 오버헤드 데이터의 양이 압축된 비디오 데이터의 양에 비하여 감소함에 따라 압축 효율을 개선시킬 수도 있다. 이런 방식으로, 그 기법들은 인코딩된 비디오 데이터의 더 효율적인 저장 및 송신을 촉진할 수도 있다.

[0007] 일 예에서, 본 개시물의 양태들은 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 방법에 관한 것이다. 그 방법은 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도 (granularity) 를 결정하는 단계; 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하는 단계; 및 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션이 없는, 프레임의 독립적으로 디코딩가능한 부분을 디코딩하는 단계를 포함한다.

[0008] 다른 예에서, 본 개시물의 양태들은 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 장치에 관한 것이다. 그 장치는 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 결정하고; 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하며; 그리고 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션이 없는, 프레임의 독립적으로 디코딩가능한 부분을 디코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0009] 다른 예에서, 본 개시물의 양태들은 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 장치에 관한 것이다. 그 장치는 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 결정하는 수단; 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하는 수단; 및 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션이 없는, 프레임의 독립적으로 디코딩가능한 부분을 디코딩하는 수단을 포함한다.

[0010] 다른 예에서, 본 개시물의 양태들은 하나 이상의 프로세서들에 의한 실행 시, 하나 이상의 프로세서들로 하여금, 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩하는 방법을 수행하게 하는 명령들을 저장한 컴퓨터 판독가능 저장 매체에 관한 것이다. 그 방법은 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 결정하는 단계; 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별하는 단계; 및 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션이 없는, 프레임의 독립적으로 디코딩가능한 부분을 디코딩하는 단계를 포함한다.

[0011] 다른 예에서, 본 개시물의 양태들은 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 방법에 관한 것이다. 그 방법은 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하는 단계; LCU의 제 1 섹션 및 LCU의 제 2 섹션을 생성하기 위해 결정된 세분도를 이용하여 LCU를 분할하는 단계; LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 포함하지 않는, 프레임의 독립적으로 디코딩가능한 부분을 생성하는 단계; 및 결정된 세분도의 표시 및 프레임의 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 단계를 포함한다.

[0012] 다른 예에서, 본 개시물의 양태들은 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 장치에 관한 것이다. 그 장치는 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하고; LCU의 제 1 섹션 및 LCU의 제 2 섹션을 생성하기 위해 결정된 세분도를 이용하여 LCU를 분할하고; LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 포함하지 않는, 프레임의 독립적으로 디코딩가능한 부분을 생성하며; 그리고 결정된 세분도의 표시 및 프레임의 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0013] 다른 예에서, 본 개시물의 양태들은 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 장치에 관한 것이다. 그 장치는 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하는 수단; LCU의 제 1 섹션 및 LCU의 제 2 섹션을 생성하기 위해 결정된 세분도를 이용하여 LCU를 분할하는 수단; LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 포함하지 않는, 프레임의 독립적으로 디코딩가능한 부분을 생성하는 수단; 및 결정된 세분도의 표시 및 프레임의 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 수단을 포함한다.

[0014] 다른 예에서, 본 개시물의 양태들은 하나 이상의 프로세서들에 의한 실행 시, 하나 이상의 프로세서들로 하여금, 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩하는 방법을 수행하게 하는 명령들을 저장한 컴퓨터 판독가능 저장 매체에 관한 것이다. 그 방법은 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정하는 단계; LCU의 제 1 섹션 및 LCU의 제 2 섹션을 생성하기 위해 결정된 세분도를 이용하여 LCU를 분할하는 단계; LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 포함하지 않는, 프레임의 독립적으로 디코딩가능한 부분을 생성하는 단계; 및 결정된 세분도의 표시 및 프레임의 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성하는 단계를 포함한다.

[0015] 본 개시물의 하나 이상의 양태들의 상세는 첨부 도면들 및 다음의 설명에서 언급된다. 본 개시물에서 설명되는 기법들의 다른 특징들, 목적들, 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

[0016] 도 1은 본 개시물의 기법들 중 하나 이상을 구현할 수도 있는 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2는 본 개시물의 기법들에 부합하는 코딩된 단위들 (CUs) 의 쿼드트리 파티셔닝을 예시하는 개념도이다.

도 3a는 본 개시물의 기법들에 부합하는 슬라이스들로 CU들의 쿼드트리를 분할하는 것을 예시하는 개념도이다.

도 3b는 본 개시물의 기법들에 부합하는 슬라이스들로 CU들을 분할하는 것을 예시하는 개념도이다.

도 4는 본 개시물의 기법들을 구현할 수도 있는 비디오 인코더를 예시하는 블록도이다.

도 5는 본 개시물의 기법들을 구현할 수도 있는 비디오 디코더를 예시하는 블록도이다.

도 6은 본 개시물에서 설명된 기법들에 부합하는 비디오 데이터를 인코딩하는 방법을 예시하는 흐름도이다.

도 7은 본 개시물에서 설명된 기법들에 부합하는 비디오 데이터를 디코딩하는 방법을 예시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 개시물의 기법들은 일반적으로 비디오 데이터의 프레임을 독립적으로 디코딩가능한 부분들로 분할하는 것을 포함하며, 독립적으로 디코딩가능한 부분들 사이의 경계는 코딩 단위 (CU), 이를테면 HEVC 표준에 의해 지정된 최대 CU (LCU) 내에 위치될 수도 있다. 예를 들어, 본 개시물의 양태들은 비디오 데이터의 프레임을 분할하는 세분도를 결정하는 것, 결정된 세분도를 이용하여 그 프레임을 분할하는 것, 및 CU 깊이를 이용하여 그 세분도를 식별하는 것에 관련될 수도 있다. 본 개시물의 기법들은 또한 프레임을 독립적으로 디코딩가능한 부분들로 분할하는 것과 연관되는 다양한 파라미터들을 생성하는 것 및/또는 디코딩하는 것을 포함할 수도 있다. 예를 들어, 본 개시물의 양태들은 CU 깊이를 이용하여 비디오 데이터의 프레임을 분할하는데 이용되는 세분도를 식별하는 것, 각각의 독립적으로 디코딩가능한 부분에 대해 계층적 쿼드트리 구조의 개별 부분들을 식별하는 것, 및 각각의 독립적으로 디코딩가능한 부분에 대해 양자화 파라미터 (즉, 델타 QP) 에서의 변화들 (즉, 델타들) 을 식별하는 것에 관련될 수도 있다.

[0018] 도 1은 비디오 데이터의 프레임들을 독립적으로 디코딩가능한 부분들로 분할하는 본 개시물에서 설명된 기법들을 활용하도록 구성될 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템 (10) 을 도시하는 블록도이다. 본 개시물의 양태들에 따르면, 비디오 데이터의 프레임의 독립적으로 디코딩가능한 부분들은 제안된 이른바 고효율 비디오 코딩 (HEVC) 표준을 포함한, 다양한 비디오 코딩 표준들과 부합하는 비디오 데이터의 "슬라이스

들"이라고 일반적으로 지칭될 수도 있다. 슬라이스는 독립적으로 디코딩가능한 것으로서 서술될 수도 있는데, 프레임의 슬라이스는 정보에 대해 동일한 프레임의 다른 슬라이스들에 의존하지 않고 그러므로 임의의 다른 슬라이스와는 독립적으로 디코딩될 수도 있기 때문이며, 그래서 "독립적으로 디코딩가능한 부분"으로 칭해질 수도 있다. 슬라이스들이 독립적으로 디코딩가능한 것임을 보장함으로써, 하나의 슬라이스에서의 에러들 또는 누락 데이터는 그 프레임 내의 임의의 다른 슬라이스로 전파되지 않는다. 에러들을 프레임 내의 단일 슬라이스로 고립시키는 것은 그런 에러들을 보상하는 시도를 또한 지원할 수도 있다.

[0019] 도 1의 예에서 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의한 디코딩을 위한 인코딩된 비디오를 생성하는 소스 디바이스 (12) 를 구비한다. 소스 디바이스 (12) 는 인코딩된 비디오를 통신 채널 (16) 을 통해 목적지 디바이스 (14) 로 송신할 수도 있거나 또는 인코딩된 비디오를 저장 매체 (34) 또는 파일 서버 (36) 상에 저장할 수도 있어서, 인코딩된 비디오는 목적지 디바이스 (14) 에 의해 원하는 대로 액세스될 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 스마트폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들 등을 포함하여, 매우 다양한 디바이스들 중 임의의 것을 포함할 수도 있다.

[0020] 많은 경우들에서, 이러한 디바이스들은 무선 통신을 위해 장착될 수도 있다. 그런고로, 통신 채널 (16) 은 무선 채널, 유선 채널, 또는 인코딩된 비디오 데이터의 송신에 적합한 무선 및 유선 채널들의 조합을 포함할 수도 있다. 예를 들면, 통신 채널 (16) 은 임의의 무선 또는 유선 통신 매체, 이를테면 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들, 또는 무선 및 유선 매체들의 임의의 조합을 포함할 수도 있다. 통신 채널 (16) 은 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크의 부분을 형성할 수도 있다. 통신 채널 (16) 은, 유선 또는 무선 매체들의 임의의 적합한 조합을 포함하여, 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 비디오 데이터를 송신하기 위한 임의의 적합한 통신 매체, 또는 상이한 통신 매체들의 컬렉션을 일반적으로 나타낸다. 통신 채널 (16) 은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0021] 비디오 데이터의 프레임들을 슬라이스들로 분할하는 본 개시물에서 설명된 기법들은, 본 개시물의 예들에 따라서, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 중 임의의 것의 지원 하의 비디오 코딩, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단-방향 또는 양-방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0022] 도 1의 예에서 추가로 도시된 바와 같이, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 변조기/복조기 (22) 및 송신기 (24) 를 구비한다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스와 같은 소스를 구비할 수도 있다. 비디오 캡처 디바이스는, 예로서, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 피드 인터페이스, 및/또는 컴퓨터 그래픽 데이터를 생성하는 컴퓨터 그래픽 시스템 중 하나 이상을 소스 비디오로서 포함할 수도 있다. 하나의 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 이른바 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 본 개시물의 기법들은, 그러나, 무선 애플리케이션들 또는 설정들로 제한될 필요는 없고, 비디오 인코딩 및/또는 디코딩 능력들을 포함하는 비-무선 디바이스들에 적용될 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 본원에서 설명되는 기법들을 지원할 수 있는 코딩 디바이스들의 단지 예들이다.

[0023] 캡처된, 사전-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 통신 표준, 이를테면 무선 통신 프로토콜에 따라 모뎀 (22) 에 의해 변조될 수도 있고, 목적지 디바이스 (14) 에 송신기 (24) 를 통해 송신될 수도 있다. 모뎀 (22) 은 갖가지 믹서들, 필터들, 증폭기들 또는 신호 변조를 위해 설계된 다른 컴포넌트들을 구비할 수도 있다. 송신기 (24) 는 증폭기들, 필터들, 및 하나 이상의 안테나들을 포함하여, 데이터를 송신하기 위해 설계된 회로들을 구비할 수도 있다.

[0024] 비디오 인코더 (20) 에 의해 인코딩되는 캡처된, 사전-캡처된, 또는 컴퓨터-생성된 비디오는 또한 나중의 소비

를 위해 저장 매체 (34) 또는 파일 서버 (36) 상에 저장될 수도 있다. 저장 매체 (34) 는 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오를 저장하는 임의의 다른 적합한 디지털 저장 매체들을 포함할 수도 있다. 저장 매체 (34) 상에 저장된 인코딩된 비디오는 그 다음에 디코딩 및 플레이백을 위해 목적지 디바이스 (14) 에 의해 액세스될 수도 있다.

[0025] 파일 서버 (36) 는 인코딩된 비디오를 저장하고 그 인코딩된 비디오를 목적지 디바이스 (14) 에 송신하는 것이 가능한 임의의 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버 (예컨대, 웹사이트 용), FTP 서버, 네트워크 접속 스토리지 (network attached storage; NAS) 디바이스들, 로컬 디스크 드라이브, 또는 인코딩된 비디오 데이터를 저장하고 그것을 목적지 디바이스에 송신하는 것이 가능한 임의의 다른 유형의 디바이스를 포함한다. 파일 서버 (36) 는 목적지 디바이스 (14) 에 의해 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 액세스될 수도 있다. 이는 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 양쪽 모두의 조합을 포함할 수도 있다. 파일 서버 (36) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.

[0026] 본 개시물은 다른 디바이스, 이를테면 비디오 디코더 (30) 에 특정 정보를 "시그널링하는" 비디오 인코더 (20) 를 일반적으로 지칭할 수도 있다. 그러나, 비디오 인코더 (20) 는 특정 신택스 엘리먼트들과 비디오 데이터의 갖가지 인코딩된 부분들을 연관시킴으로써 정보를 시그널링할 수도 있다는 것이 이해되어야 한다. 다시 말하면, 비디오 인코더 (20) 는 특정 신택스 엘리먼트들을 비디오 데이터의 갖가지 인코딩된 부분들의 헤더들에 저장함으로써 데이터를 "시그널링"할 수도 있다. 일부 경우들에서, 그런 신택스 엘리먼트들은 비디오 디코더 (30) 에 의해 수신되고 디코딩되기 전에 인코딩되고 저장 (예컨대, 저장 매체 (34) 또는 파일 서버 (36) 에 저장) 될 수도 있다. 따라서, 용어 "시그널링"은, 인코딩시에 매체에 신택스 엘리먼트들을 저장할 때 (이 매체에 저장된 후의 임의의 시간에 디코딩 디바이스에 의해 취출될 수도 있다) 발생할 수도 있는 것과 같이, 압축된 비디오 데이터를 디코딩하는데 필요한 다른 데이터 또는 신택스의 통신이 실시간 또는 거의 실시간으로 또는 어떤 기간에 걸쳐 일어나든지 간에 그런 통신을 일반적으로 지칭할 수도 있다.

[0027] 목적지 디바이스 (14) 는, 도 1의 예에서, 수신기 (26), 모뎀 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 구비한다. 목적지 디바이스 (14) 의 수신기 (26) 는 채널 (16) 을 통해 정보를 수신하고, 모뎀 (28) 은 비디오 디코더 (30) 를 위한 복조된 비트스트림을 생성하기 위해 그 정보를 복조한다. 채널 (16) 을 통해 통신되는 정보는 비디오 데이터 디코딩에서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성되는 다양한 신택스 정보를 포함할 수도 있다. 그런 신택스는 또한 저장 매체 (34) 또는 파일 서버 (36) 상에 저장되는 인코딩된 비디오 데이터와 함께 포함될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 비디오 데이터를 인코딩 또는 디코딩하는 것이 가능한 개별 인코더-디코더 (CODEC) 의 부분을 형성할 수도 있다.

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

[0029] 비디오 인코더 (20) 와 비디오 디코더 (30) 는 비디오 압축 표준, 이를테면 현재 개발 중인 고 효율 비디오 코딩 (HEVC) 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HM) 을 준수할 수도 있다. 대안으로, 비디오 인코더 (20) 와 비디오 디코더 (30) 는, 다르게는 MPEG-4, 파트 10, 고급 비디오 코딩 (AVC) 이라고 지칭되는 ITU-T H.264 표준과 같은 다른 독점 또는 산업 표준들, 또는 그런 표준들의 확장안들에 따라 동작할 수도 있다. 본 개시물의 기법들은, 그러나, 임의의 특정한 코딩 표준으로 제한되지 않는다. 다른 예들은 MPEG-2 및 ITU-T H.263을 포함한다.

[0030] HEVC 표준은 비디오 데이터의 블록을 코딩 단위 (coding unit; CU) 라고 지칭한다. 일반적으로, CU가 사이즈 차이 (size distinction) 를 가지지 않는다는 점을 제외하면, CU는 H.264에 따라 코딩된 매크로블록과 유사한 목적을 가진다. 따라서, CU는 서브 CU들로 분할될 수도 있다. 일반적으로, 본 개시물에서의 CU에 대한 언급들은 한 화상의 최대 코딩 단위 (LCU) 또는 LCU의 서브 CU를 지칭할 수도 있다. 예를 들어, 비트스트림 내의 신택스 데이터는 화소들의 수의 측면에서 가장 큰 코딩 단위인 LCU를 정의할 수도 있다. LCU는

서브 CU들로 분할될 수도 있고, 각각의 서브 CU는 서브 CU들로 분할될 수도 있다. 비트스트림에 대한 신택스 데이터는 최대 CU 깊이라고 지칭되는, LCU가 분할될 수도 있는 최대 횟수를 정의할 수도 있다. 따라서, 비트스트림은 또한 최소 코딩 단위 (smallest coding unit; SCU) 를 정의할 수도 있다.

[0031] LCU는 계층적 쿼드트리 (quadtree) 데이터 구조에 연관될 수도 있다. 대체로, 쿼드트리 데이터 구조는 CU 당 하나의 노드를 포함하며, 여기서 루트 노드가 LCU에 대응한다. CU가 4 개의 서브 CU들로 분할되면, 그 CU에 대응하는 노드는 4 개의 잎 (leaf) 노드들을 포함하며, 그것들의 각각은 서브 CU들 중 하나에 대응한다.

쿼드트리 데이터 구조의 각각의 노드는 신택스 데이터를 대응하는 CU에 제공할 수도 있다. 예를 들어, 쿼드트리에서의 노드는 그 노드에 대응하는 CU가 서브 CU들로 분할되는지의 여부를 표시하는 분할 플래그를 포함할 수도 있다. CU에 대한 신택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU가 서브 CU들로 분할되는지의 여부에 의존할 수도 있다.

[0032] 분할되지 않은 CU는 하나 이상의 예측 단위들 (PU들) 을 포함할 수도 있다. 일반적으로, PU는 대응하는 CU의 전부 또는 일 부분을 나타내고, PU에 대한 참조 샘플을 취출하기 위한 데이터를 포함한다. 예를 들어, PU가 인트라 모드 인코딩되는 경우, PU는 PU에 대한 인트라 예측 모드를 설명하는 데이터를 포함할 수도 있다.

다른 예로서, PU가 인터 모드 인코딩되는 경우, PU는 PU에 대한 움직임 벡터를 정의하는 데이터를 포함할 수도 있다. 움직임 벡터를 정의하는 데이터는, 예를 들어, 움직임 벡터의 수평 성분, 움직임 벡터의 수직 성분, 움직임 벡터에 대한 분해능 (예컨대, 1/4 화소 정밀도 또는 1/8 화소 정밀도), 움직임 벡터가 가리키는 참조 프레임, 및/또는 움직임 벡터에 대한 참조 목록 (예컨대, 목록 0 또는 목록 1) 을 서술할 수도 있다. PU(들)를 정의하는 CU에 대한 데이터는 또한, 예를 들어, 하나 이상의 PU들로의 CU의 파티셔닝을 서술할 수도 있다. 파티셔닝 모드들은 CU가 코딩되지 않는지, 인트라 예측 모드 인코딩되는지, 또는 인터 예측 모드 인코딩되는지 사이에서 상이할 수도 있다.

[0033] 하나 이상의 PU들을 갖는 CU는 또한 하나 이상의 변환 단위들 (TUs) 을 포함할 수도 있다. PU를 이용한 예측에 뒤이어, 비디오 인코더는 PU에 대응하는 CU의 부분에 대한 잔차 (residual) 값을 계산할 수도 있다. 잔차 값은 변환, 양자화, 및 스캐닝될 수도 있다. TU는 PU의 사이즈로 반드시 제한되지는 않는다. 따라서, TU들은 동일한 CU에 대해 대응하는 PU들보다 더 크거나 더 작을 수도 있다. 일부 예들에서, TU의 최대 사이즈는 대응하는 CU의 사이즈일 수도 있다. 본 개시물은 또한 CU, PU, 또는 TU 중 임의의 것을 지칭하기 위해 용어 "블록"을 이용한다.

[0034] 본 개시물의 양태들이 "최대 코딩 단위 (LCU)"를 제안된 HEVC 표준에서 지정된 바와 같은 것으로 지칭할 수도 있지만, 용어 "최대 코딩 단위"의 범위는 제안된 HEVC 표준으로 제한되지 않는다는 것이 이해되어야 한다. 예를 들어, 용어 최대 코딩 단위는 일반적으로 코딩 단위의 상대 사이즈를 지칭할 수도 있는데 그 코딩 단위가 인코딩된 비디오 데이터의 다른 코딩 단위들에 관련될 수 있어서이다. 다르게 말하면, 최대 코딩 단위는 (예컨대, 그 프레임에서의 다른 코딩 단위들과 비교하여) 하나 이상의 상이한 크기의 코딩 단위들을 갖는 비디오 데이터의 프레임에서의 상대 최대 코딩 단위를 지칭할 수도 있다. 다른 예에서, 용어 최대 코딩 단위는, 연관된 신택스 엘리먼트들 (예컨대, 계층적 쿼드트리 구조를 서술하는 신택스 엘리먼트들 등) 을 가질 수도 있는, 제안된 HEVC 표준에서 지정된 바와 같은 최대 코딩 단위를 지칭할 수도 있다.

[0035] 일반적으로, 인코딩된 비디오 데이터는 예측 데이터 및 잔차 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 인트라-예측 모드 또는 인터-예측 모드 동안에 예측 데이터를 생성할 수도 있다. 인트라-예측은 일반적으로 화상의 블록에서의 화소 값들을, 동일한 화상의 이웃하는 이전에 코딩된 블록에서의 참조 샘플들을 기준으로 예측하는 것을 수반한다. 인터-예측은 일반적으로, 화상의 블록에서의 화소 값들을, 이전에 코딩된 화상의 데이터를 기준으로 예측하는 것을 수반한다.

[0036] 인트라- 또는 인터-예측에 뒤이어, 비디오 인코더 (20) 는 그 블록에 대한 잔차 화소 값들을 계산할 수도 있다.

잔차 값들은 일반적으로 그 블록에 대한 예측된 화소 값 데이터 및 그 블록의 실제 (true) 화소 값 데이터 사이의 차이들에 대응한다. 예를 들어, 잔차 값들은 코딩된 화소들 및 예측 화소들 사이의 차이들을 나타내는 화소 차이 값들을 포함할 수도 있다. 일부 예들에서, 코딩된 화소들은 코딩될 화소들의 블록에 연관될 수도 있고, 예측 화소들은 코딩된 블록을 예측하는데 이용되는 화소들의 하나 이상의 블록과 연관될 수도 있다.

[0037] 블록의 잔차 값을 추가로 압축하기 위해, 잔차 값은 가능한 한 많은 데이터 (또한 "에너지"라 지칭됨) 를 가능한 한 적은 계수들로 압축하는 변환 계수들의 세트로 변환될 수도 있다. 변환 기법들은 이산 코사인 변환 (DCT) 프로세스 또는 개념적으로 유사한 프로세스, 정수 변환들, 웨이브릿 변환들, 또는 다른 유형들의 변환들을 포함할 수도 있다. 그 변환은 화소들의 잔차 값들을 공간적 도메인에서부터 변환 도메인으로

변환시킨다. 변환 계수들은 원본 블록과 보통 동일한 사이즈인 계수들의 2차원 매트릭스에 대응한다. 다르게 말하면, 원본 블록에서의 화소들만큼 많은 변환 계수들이 그대로 존재한다. 그러나, 변환으로 인해, 대다수의 변환 계수들은 0과 동일한 값들을 가질 수도 있다.

[0038] 비디오 인코더 (20) 는 그 다음에 비디오 데이터를 추가로 압축시키기 위해 변환 계수들을 양자화한다. 양자화는 일반적으로, 상대적으로 큰 범위 내의 값들을 상대적으로 작은 범위의 값들로 매핑하며, 따라서 양자화된 변환 계수들을 표현하는데 필요한 데이터의 양을 감소시키는 것을 수반한다. 더 구체적으로는, 양자화는 LCU 레벨에서 정의될 수도 있는 양자화 파라미터 (QP) 에 따라 적용될 수도 있다. 따라서, 양자화의 동일한 레벨은 LCU 내의 CU들의 상이한 PU들에 연관된 TU들에서의 모든 변환 계수들에 적용될 수도 있다. 그러나, QP 자체를 시그널링하기보다는, QP에서의 변화 (즉, 델타) 가 LCU와 함께 시그널링될 수도 있다. 델타 QP는 그 LCU에 대한 양자화 파라미터에서의, 일부 기준 QP, 이를테면 이전에 통신된 LCU의 QP에 관한 변화를 정의한다.

[0039] 양자화를 뒤이어, 비디오 인코더 (20) 는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 포함하는 2차원 매트릭스로부터 1차원 벡터를 생성할 수도 있다. 비디오 인코더 (20) 는 그 다음에 데이터를 한층 더 압축하기 위해 결과적인 어레이를 엔트로피 인코딩할 수도 있다. 일반적으로, 엔트로피 코딩은 양자화된 변환 계수들의 시퀀스 및/또는 다른 신택스 정보를 통합하여 압축하는 하나 이상의 프로세스들을 포함한다. 예를 들어, 신택스 엘리먼트들, 이를테면 델타 QP들, 예측 벡터, 코딩 모드들, 필터들, 오프셋들, 또는 다른 정보는 엔트로피 코딩된 비트스트림에 또한 포함될 수도 있다. 그 다음 스캐닝된 계수들은 임의의 신택스 정보와 함께, 예컨대, 콘텐츠 적응 가변 길이 코딩 (content adaptive variable length coding; CAVLC), 콘텍스트 적응 이진 산술 코딩 (context adaptive binary arithmetic coding; CABAC), 또는 다른 엔트로피 코딩 프로세스를 통해 엔트로피 코딩된다.

[0040] 다시, 본 개시물의 기법들은 비디오 데이터의 프레임을 독립적으로 디코딩가능한 슬라이스들로 분할하는 것을 포함한다. 일부 경우들에서, 비디오 인코더 (20) 는 특정 사이즈로 된 슬라이스들을 형성할 수도 있다. 하나의 이러한 경우는, 이더넷 네트워크 또는 계층 2 (L2) 아키텍처가 이더넷 프로토콜을 활용하는 (여기에서 계층들 및 뒤따르는 숫자는 개방형 시스템간 상호접속 (Open System Interconnection; OSI) 모델의 대응하는 계층을 지칭한다) 임의의 다른 유형의 네트워크를 통해 슬라이스들을 송신하기 위해 준비될 수도 있다. 이 예에서, 비디오 인코더 (20) 는 1500 바이트일 수도 있는 최대 송신 단위 (MTU) 보다 약간만 작은 슬라이스들을 형성할 수도 있다.

[0041] 보통, 비디오 인코더들은 LCU를 추종하여 슬라이스를 분할한다. 다시 말하면, 비디오 인코더들은 슬라이스가 하나 이상의 완전한 LCU들을 포함하게, 슬라이스 세분도를 LCU의 사이즈로 제한하도록 구성될 수도 있다. 슬라이스 세분도를 LCU로 제한하는 것은, 그러나, 특정 사이즈의 슬라이스들을 형성하도록 시도하는 경우에 도전과제들을 제시할 수도 있다. 예를 들어, 이 방식으로 구성된 비디오 인코더들은 상대적으로 큰 LCU들을 갖는 프레임들에서 특정 사이즈의 슬라이스 (예컨대, 소정의 양의 데이터를 포함하는 슬라이스) 를 생성하는 것이 가능하지 않을 수도 있다. 다시 말하면, 상대적으로 큰 LCU들은 소망의 사이즈보다 상당히 아래에 있는 슬라이스로 나타날 수도 있다. 본 개시물은 슬라이스를 생성하는 경우에 비디오 데이터의 블록, 이를테면 LCU가 더 작은 부분들로 나누어질 (예컨대, 분리될) 수도 있는 정도로서 "세분도 (granularity)"를 일반적으로 지칭한다. 그런 세분도는 또한 일반적으로 "슬라이스 세분도"라고 지칭될 수도 있다. 다시 말하면, 세분도 (또는 슬라이스 세분도) 는 상이한 슬라이스들로 나누어질 수도 있는 LCU 내의 서브 CU들의 상대 사이즈를 지칭할 수도 있다. 아래에서 더욱 상세히 설명되는 바와 같이, 세분도는 슬라이스 분할이 일어나는 계층적 CU 깊이에 따라 식별될 수도 있다.

[0042] 예시를 위해 위에서 제공된 1500 바이트 타겟 최대 슬라이스 사이즈의 예를 고려한다. 이 예시에서, 완전 (full) -LCU 슬라이스 세분도로 구성된 비디오 인코더는 500 바이트의 제 1 LCU, 400 바이트의 제 2 LCU 및 900 바이트의 제 3 LCU를 생성할 수도 있다. 비디오 인코더는 제 1 및 제 2 LCU들을 총 900 바이트의 슬라이스 사이즈의 슬라이스에 저장할 수도 있으며, 여기서 제 3 LCU의 추가는 1500 바이트 최대 슬라이스 사이즈를 대략 300 바이트만큼 초과할 수도 있다 (900 바이트 + 900 바이트 - 1500 바이트 = 300 바이트). 따라서, 슬라이스의 최종 LCU는 이 타겟 최대 용량까지 슬라이스를 채우지 않을 수도 있고, 슬라이스의 나머지 용량은 다른 완전 LCU를 수용하기에 충분히 크지 않을 수도 있다. 결과적으로, 슬라이스는 제 1 및 제 2 LCU만을 저장할 수도 있으며 제 3 LCU와 1500 바이트의 타켓 사이즈 마이너스 제 3 LCU의 900 바이트, 또는 900 바이트보다 작은 사이즈를 갖는 잠재적으로 임의의 부가적인 LCU들을 저장하기 위해 다른 슬라이스가 생성된다. 3 개보다는 2 개의 슬라이스들이 요구되기 때문에, 제 2 슬라이스는 슬라이스 헤더들의 형태로 부가적인 오버헤드

를 도입하여, 대역폭 및 저장 비효율성들을 생성한다.

[0043] 본 개시물에서 설명된 기법들에 따라, 비디오 인코더 (20) 는 비디오 데이터의 프레임을 LCU보다 작은 세분도의 슬라이스들로 분할할 수도 있다. 다시 말하면, 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 는 LCU 내에 위치될 수도 있는 경계를 이용하여 비디오 데이터의 프레임을 슬라이스들로 분할할 수도 있다. 일 예에서, 비디오 인코더 (20) 는 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 LCU을 포함하는 복수의 블록 사이즈의 CU들을 갖는 비디오 데이터의 프레임을 독립적으로 디코딩가능한 슬라이스들로 분할할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정할 수도 있다. 비디오 인코더 (20) 는 또한 LCU의 제 1 섹션 및 LCU의 제 2 섹션을 생성하기 위해 결정된 세분도를 이용하여 LCU를 분할할 수도 있다. 비디오 인코더 (20) 는 또한 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 포함하지 않는, 프레임의 독립적으로 디코딩가능한 부분을 생성할 수도 있다. 비디오 인코더 (20) 는 또한 결정된 세분도의 표시 및 프레임의 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성할 수도 있다.

[0044] 비디오 인코더 (20) 는 프레임을 독립적으로 디코딩가능한 슬라이스들로 분할할 세분도를 결정하는 경우에 다양한 파라미터들을 고려할 수도 있다. 예를 들어, 위에서 지적했듯이, 비디오 인코더 (20) 는 소망의 슬라이스 사이즈에 기초하여 프레임을 분할할 세분도를 결정할 수도 있다. 다른 예들에서, 도 4에 관해 더 상세히 설명되는 바와 같이, 비디오 인코더 (20) 는 에러 결과들 대 비디오 데이터를 시그널링하는데 필요한 비트들의 수 (예컨대, 때때로 레이트-왜곡이라고 지칭됨) 를 고려하고 이들 에러 결과들 대 비디오 데이터를 시그널링하는데 필요한 비트들의 수 (또는 그 비교) 에 근거한 세분도의 결정에 기초할 수도 있다.

[0045] 일 예에서, 비디오 인코더 (20) 는 비디오 데이터의 프레임이 LCU보다 작은 세분도에서 슬라이스들로 분할될 것을 결정할 수도 있다. 예시의 목적을 위해 제공된 단지 하나의 예로서, 비디오 데이터의 프레임에 연관된 LCU는 64 화소 바이 64 화소의 사이즈일 수도 있다. 이 예에서, 비디오 인코더 (20) 는 32 화소 바이 32 화소의 CU 세분도를 이용하여 프레임이 슬라이스들로 분할될 것임을 결정할 수도 있다. 다시 말하면, 비디오 인코더 (20) 는 32 화소 바이 32 화소 사이즈 이상인 CU들 사이의 경계를 이용하여 프레임을 슬라이스들로 분할할 수도 있다. 이러한 세분도는, 예를 들어, 특정 슬라이스 사이즈를 달성하기 위하여, 구현될 수도 있다. 일부 예들에서, 세분도는 CU 깊이를 이용하여 표현될 수도 있다. 다시 말하면, 32 화소 바이 32 화소의 세분도에서 슬라이스들로 분할될 64 화소 바이 64 화소 사이즈인 LCU에 대해, 세분도는 1의 CU 깊이에 의해 표현될 수도 있다.

[0046] 다음으로, 비디오 인코더 (20) 는 LCU의 제 1 섹션 및 LCU의 제 2 섹션을 생성하기 위해 결정된 세분도에서 LCU를 분할함으로써 프레임을 슬라이스들로 분할할 수도 있다. 위에서 제공된 예에서, 비디오 인코더 (20) 는 예기되는 (prospective) 슬라이스의 최종 LCU를 제 1 및 제 2 섹션으로 분할할 수도 있다. 다시 말하면, LCU의 제 1 섹션은 LCU에 연관되는, 비디오 데이터의 하나 이상의 32 화소 바이 32 화소 블록들을 포함할 수도 있는 반면, LCU의 제 2 섹션은 LCU에 연관되는 나머지 32 화소 바이 32 화소 블록들을 포함할 수도 있다. 위의 예에서 동일한 사이즈의 화소 블록들을 포함하는 것으로 특정되었지만, 각각의 섹션은 상이한 수의 화소 블록들을 포함할 수도 있다. 예를 들어, 제 1 섹션은 8 화소 바이 8 화소 블록들을 포함할 수도 있는 반면 제 2 섹션은 나머지 3 개의 8 화소 바이 8 화소 블록들을 포함할 수도 있다. 덧붙여서, 위의 예에서 정사각형 화소 블록들인 것으로 설명되었지만, 각각의 섹션은 사각형 화소 블록 또는 임의의 다른 유형의 화소 블록을 포함할 수도 있다.

[0047] 이런 방식으로, 비디오 인코더 (20) 는 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 포함하지 않는, 프레임의 독립적으로 디코딩가능한 부분, 예컨대, 슬라이스를 생성할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 하나 이상의 완전 LCU들, 뿐만 아니라 위에서 식별된 분할 LCU의 제 1 섹션을 포함하는 슬라이스를 생성할 수도 있다. 비디오 인코더 (20) 는 그러므로 특정 사이즈의 슬라이스 (예컨대, 소정 양의 데이터) 를 형성하기 위해 시도하는 경우에 유연성을 제공할 수도 있는, LCU보다 작은 세분도에서 슬라이스를 생성하기 위해 본 개시물에서 설명된 기법들을 구현할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 결정된 세분도를 화상들의 그룹 (예컨대, 하나를 넘는 프레임) 에 적용할 수도 있다.

[0048] 비디오 인코더 (20) 는 또한 결정된 세분도의 표시 및 프레임의 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성할 수도 있다. 다시 말하면, 비디오 인코더 (20) 는 하나 이상의 화상들이 슬라이스들로 분할될 수도 있는 세분도와 그것을 뒤따르는 하나 이상의 화상들을 시그널링할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 프레임이 슬라이스들로 분할될 수도 있는 CU 깊이를 식별함으로써 세분도를 표시할 수도

있다. 그런 예들에서, 비디오 인코더 (20) 는 CU 깊이로서 시그널링될 수도 있는 세분도에 기초하여 하나 이상의 신택스 엘리먼트들을 비트스트림에 포함시킬 수도 있다. 덧붙여서, 비디오 인코더 (20) 는 슬라이스가 시작되는 어드레스 (예컨대, "슬라이스 어드레스") 를 표시할 수도 있다. 슬라이스 어드레스는 슬라이스가 프레임 내에서 시작하는 상대 포지션을 표시할 수도 있다. 슬라이스 어드레스는 슬라이스 세분도 레벨로 제공될 수도 있다. 일부 예들에서, 슬라이스 어드레스는 슬라이스 헤더에 제공될 수도 있다.

[0049] 본 개시물의 양태들에 따르면, 비디오 디코더 (30) 는 비디오 프레임의 독립적으로 디코딩가능한 부분들을 디코딩할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 비디오 프레임의 하나 이상의 독립적으로 디코딩가능한 부분들을 포함하는 비트스트림을 수신하고 그 비트스트림을 디코딩할 수도 있다. 더 구체적으로는, 비디오 디코더 (30) 는, 슬라이스들이 프레임의 LCU 미만인 세분도로 형성되었던, 비디오 데이터의 독립적으로 디코딩가능한 슬라이스들을 디코딩할 수도 있다. 다시 말하면, 예를 들어, 비디오 디코더 (30) 는 LCU 미만의 세분도로 형성되었던 슬라이스를 수신하고 비트스트림에 포함된 데이터를 이용하여 그 슬라이스를 재구성하도록 구성될 수도 있다. 일 예에서, 아래에서 더욱 상세히 설명되는 바와 같이, 비디오 디코더 (30) 는 비트스트림에 포함된 하나 이상의 신택스 엘리먼트들 (예컨대, 슬라이스가 분할되었던 CU 깊이를 식별하는 신택스 엘리먼트, 하나 이상의 분할 플래그들 등) 에 기초하여 세분도를 결정할 수도 있다.

[0050] 슬라이스 세분도는 하나의 화상에 적용될 수도 있거나 또는 다수의 화상들 (예컨대, 화상들의 그룹) 에 적용될 수도 있다. 예를 들어, 슬라이스 세분도는 파라미터 세트, 이를테면 화상 파라미터 세트 (picture parameter set; PPS) 로 시그널링될 수 있다. PPS는 화상들의 시퀀스 내의 하나 이상의 화상들 (예컨대, 비디오 데이터의 하나 이상의 프레임들) 에 적용될 수도 있는 파라미터들을 일반적으로 포함한다. 보통, PPS는 슬라이스를 디코딩하기 전에 (예컨대, 슬라이스 헤더 및 슬라이스 데이터를 디코딩하기 전에) 디코더 (30) 에 전송될 수도 있다. 슬라이스 헤더에서의 신택스 데이터는 어떤 PPS를 참조할 수도 있으며, 그것은 슬라이스를 위해 그 PPS를 "활성화"할 수도 있다. 다시 말하면, 비디오 디코더 (30) 는 슬라이스 헤더 디코딩 시에 PPS에서 시그널링되는 파라미터들을 적용할 수도 있다. 일부 예들에 따르면, 일단 PPS가 특정 슬라이스에 대해 활성화되었다면, 그 PPS는 상이한 화상 파라미터 세트가 (예컨대, 다른 슬라이스 헤더에서 참조되는 것에 의해) 활성화되기까지 유지될 수도 있다.

[0051] 위에서 지적했듯이, 본 개시물의 양태들에 따르면, 슬라이스 세분도는 파라미터 세트, 이를테면 PPS에서 시그널링될 수도 있다. 따라서, 슬라이스는 특정 PPS를 참조함으로써 특정 세분도가 할당될 수도 있다. 다시 말하면, 비디오 디코더 (30) 는 슬라이스에 대한 특정 PPS를 참조할 수도 있는, 그 슬라이스에 연관된 헤더 정보를 디코딩할 수도 있다. 비디오 디코더 (30) 는 그 다음에 슬라이스를 디코딩하는 경우에 PPS에서 식별된 슬라이스 세분도를 그 슬라이스에 적용할 수도 있다. 덧붙여서, 본 개시물의 양태들에 따르면, 비디오 디코더 (30) 는 슬라이스가 시작되는 어드레스 (예컨대, "슬라이스 어드레스") 를 표시하는 정보를 디코딩할 수도 있다. 슬라이스 어드레스는 슬라이스 세분도 레벨에서 슬라이스 헤더에 제공될 수도 있다. 도 1에 도시되진 않았지만, 일부 양태들에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 각각이 오디오 인코더 및 디코더와 통합될 수도 있고, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함하여, 공통 데이터 스트림 또는 개별 데이터 스트림들에서의 오디오 및 비디오 양쪽 모두의 인코딩을 핸들링할 수도 있다. 적용가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜들을 준수할 수도 있다.

[0052] 비디오 인코더 (20) 와 비디오 디코더 (30) 각각은 다양한 적합한 인코더 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 집적회로들 (ASICs), 필드 프로그램가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그것들의 임의의 조합 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행하기 위해, 적합한 비일시적 컴퓨터 판독가능 매체 내에 소프트웨어에 대한 명령을 저장하고 하나 이상의 프로세서들을 이용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC) 의 부분으로서 개별 디바이스 내에 통합될 수도 있다.

[0053] 도 2는 본 개시물의 기법들 및 신흥 HEVC 표준에 부합하는 코딩된 단위들 (CUs) 의 계층적 쿼드트리 파티셔닝을 예시하는 개념도이다. 도 2에 보인 예에서, LCU (CU_0) 는 128 화소 바이 128 화소의 사이즈이다. 다시 말하면, CU_0 는 분리되지 않은 CU 깊이 0에서 128 화소 바이 128 화소의 사이즈 (예컨대, N = 64) 이다. 비디오 인코더 (20) 는 CU_0 를 각각이 서브 CU를 포함하는 4 개의 사분역들로 분할할지, 또는 분할 없이 CU_0 를 인

코딩할지를 결정할 수도 있다. 이 결정은, 예를 들어, CU_0 에 연관된 비디오 데이터의 복잡도에 기초할 수도 있으며, 여기서 더 복잡한 비디오 데이터가 분할의 확률을 증가시킨다.

[0054] CU_0 를 분할하는 결정은 분할 플래그에 의해 표현될 수도 있다. 일반적으로, 분할 플래그는 비트스트림 내에 신택스 엘리먼트로서 포함될 수도 있다. 다시 말하면, CU_0가 분할되지 않으면, 분할 플래그는 0으로 설정될 수도 있다. 반대로, CU_0 가 서브 CU들을 포함하는 사분역들로 분할되면, 그 분할 플래그는 1로 설정될 수도 있다. 도 3a 및 도 3b에 관해 더 상세히 설명되는 바와 같이, 비디오 인코더, 이를테면 비디오 인코더 (20) (도 1) 는, LCU 및 LCU의 서브 CU들의 분할을 표시하는 쿼드트리 데이터 구조를 분할 플래그들을 이용하여 나타낼 수도 있다.

[0055] CU 깊이는 LCU, 이를테면 CU_0가 분할된 횟수를 표시하는데 이용될 수도 있다. 예를 들어, CU_0를 분할한 후 (예컨대, 분할 플래그 = 1), 결과적인 서브 CU들은 1의 깊이를 가진다. CU의 CU 깊이는, LCU 사이즈가 알려져 있다는 것을 전제로, 그 CU의 사이즈의 표시를 또한 제공할 수도 있다. 도 2에 보인 예에서, CU_0는 128 화소 바이 128 화소의 사이즈이다. 따라서, 깊이 1에서의 각각의 CU (도 2의 예에서 CU_1로서 도시됨) 는, 64 화소 바이 64 화소의 사이즈이다.

[0056] 이런 방식으로, CU들은 최대 계층 깊이에 도달되기까지 서브 CU들로 재귀적으로 나누어질 수도 있다. CU는 최대 계층 깊이를 초과하게 분리될 수 없다. 도 2에 보인 예에서, CU_0 는 4의 최대 계층 깊이에 도달되기까지 서브 CU들로 분리될 수도 있다. 4의 CU 깊이 (예컨대, CU_4) 에서, CU들은 8 화소 바이 8 화소의 사이즈이다.

[0057] CU_0 가 도 2의 예에서 128 화소 바이 128 화소의 사이즈이고 4의 최대 계층 깊이를 갖는 것으로 도시되었지만, 그것은 예시의 목적을 위한 하나의 예로서만 제공된다. 다른 예들은 더 크거나 또는 더 작은 그리고 동일하거나 또는 대안적인 최대 계층 깊이를 가지는 LCU들을 포함할 수도 있다.

[0058] 도 3a 및 도 3b는 본 개시물의 기법들에 부합하는, 일 예의 쿼드트리 (50) 및 대응하는 최대 코딩 단위 (80) 를 도시하는 개념도들이다. 쿼드트리 (50) 는 계층적 형태로 배열된 노드들을 포함한다. 각각의 노드는 자식이 없는 잎 노드일 수도 있거나, 또는 4 개의 자식 노드들을 가질 수도 있으며, 그래서 이름 "쿼드트리"를 가질 수도 있다. 도 3a의 일 예에서, 쿼드트리 (50) 는 루트 노드 (52) 를 포함한다. 루트 노드 (52) 는 잎 노드들 (54A 및 54B) (잎 노드들 (54)) 과 노드들 (56A 및 56B) 을 포함하는 4 개의 자식 노드들을 가진다. 노드들 (56) 이 잎 노드들이 아니기 때문에, 노드들 (56) 각각은 4 개의 자식 노드들을 포함한다. 다시 말하면, 도 3a에 도시된 예에서, 노드 (56A) 는 4 개의 자식 잎 노드들 (58A-58D) 을 가지는 반면, 노드 (56B) 는 3 개의 잎 노드들 (60A-60C) (잎 노드들 (60)) 및 노드 (62) 를 가진다. 덧붙여서, 노드 (62) 는 4 개의 잎 노드들 (64A-64D) (잎 노드들 (64)) 을 가진다.

[0059] 쿼드트리 (50) 는 대응하는 최대 코딩 단위 (LCU), 이를테면 이 예에서 LCU (80) 의 특성들을 서술하는 데이터를 포함할 수도 있다. 예를 들어, 쿼드트리 (50) 는, 그것의 구조에 의해, LCU (80) 의 서브 CU들로의 분할을 서술할 수도 있다. LCU (80) 가 2Nx2N의 사이즈를 가진다고 가정한다. 이 예에서, LCU (80) 는, 2 개의 서브 CU들 (82A 및 82B) (서브 CU들 (82)) 이 사이즈 NxN인 4 개의 서브 CU들을 가진다. LCU (80) 의 나머지 2 개의 서브 CU들은 더 작은 서브 CU들로 추가로 분할된다. 다시 말하면, 도 3b에 도시된 예에서, LCU (80) 의 서브 CU들 중 하나는 사이즈 N/2xN/2의 서브 CU들 (84A-84D) 로 분할되는 반면, LCU (80) 의 다른 서브 CU는 사이즈 N/2xN/2의 서브 CU들 (86A-86C) (서브 CU들 (86)) 과 사이즈 N/4xN/4의 서브 CU들 (88A-88D) (서브 CU들 (88)) 로서 식별되는 추가로 분리된 (divided) 서브 CU로 분할된다.

[0060] 도 3a 및 도 3b에 도시된 예에서, 쿼드트리 (50) 의 구조는 LCU (80) 의 분할에 대응한다. 다시 말하면, 루트 노드 (52) 는 LCU (80) 에 대응하고 잎 노드들 (54) 은 서브 CU들 (82) 에 대응한다. 더구나, 잎 노드들 (58) (그것들은 노드 (56A) 의 자식 노드이며, 이는 노드 (56A) 가 잎 노드 (58) 를 가리키는 포인터를 구비함을 통상 의미한다) 은 서브 CU들 (84) 에 대응하며, 잎 노드들 (60) (예컨대, 노드 (56B) 에 속함) 은 서브 CU들 (86) 에 대응하고, 잎 노드들 (64) (예컨대, 노드 (62) 에 속함) 은 서브 CU들 (88) 에 대응한다.

[0061] 도 3a 및 도 3b에 도시된 예에서, LCU (80) (그것은 루트 노드 (52) 에 대응함) 는, 제 1 섹션 (90) 및 제 2 섹션 (92) 으로 분할된다. 본 개시물의 양태들에 따르면, 비디오 인코더, 이를테면 비디오 인코더 (20) 는, LCU (80) 를 제 1 섹션 (90) 및 제 2 섹션 (92) 으로 분할하고, LCU (80) 가 속하는 프레임의 제 1 독립적으로

디코딩가능한 부분에 제 1 섹션 (90) 을 포함시킬 수도 있고, LCU (80) 가 속하는 프레임의 제 2 독립적으로 디코딩가능한 부분에 제 2 섹션 (92) 을 포함할 수도 있다. 다시 말하면, 비디오 인코더 (20) 는 제 1 슬라이스 (예컨대, 화살표 96에 의해 나타내어짐) 가 제 1 섹션 (90) 을 포함하고 제 2 슬라이스 (예컨대, 화살표 98에 의해 나타내어짐) 가 제 2 섹션 (92) 을 포함하도록, LCU (80) 를 포함하는 비디오 데이터의 프레임을 (예컨대, "슬라이스 분할" 화살표 94에 의해 나타내어진 바와 같이) 슬라이스들로 분할할 수도 있다. 예를 들어, 제 1 슬라이스 (96) 는, 슬라이스의 상대적 말단으로서 위치될 수도 있는 LCU (80) 의 제 1 섹션 (90) 외에도, 하나 이상의 완전한 LCU들을 포함할 수도 있다. 비슷하게, 제 2 슬라이스 (98) 는 LCU (80) 의 제 2 섹션 (92) 으로 시작하고 하나 이상의 부가적인 다른 LCU들을 포함할 수도 있다.

[0062] 도 3a 및 도 3b에 관해 도시되고 설명된 방식으로, CU (80) 를 포함하는 비디오 데이터의 프레임을 독립적으로 디코딩가능한 슬라이스들로 분할하기 위해, 슬라이스들이 생성되는 세분도는 이 개시물의 기법들에 따라 LCU (80) 의 사이즈보다 작아야만 한다. 일 예에서, 설명을 위해 LCU (80) 가 64 화소 바이 64 화소의 사이즈 (예컨대, N = 32) 라고 가정한다. 이 예에서, 슬라이스 세분도는 16 화소 바이 16 화소들이다. 예를 들어, 슬라이스 경계에 의해 분리되는 최소 CU들의 사이즈들은 16 화소 바이 16 화소의 사이즈이다.

[0063] 프레임의 LCU, 이를테면 LCU (80) 가 슬라이스들로 분할될 수도 있는 세분도는 그 분할이 일어나는 CU 깊이 값에 따라 식별될 수도 있다. 도 3a의 예에서, 슬라이스 분할 (94) 이 2의 CU 깊이에서 일어난다. 예를 들어, 제 1 슬라이스 (96) 에 포함될 수도 있는 제 1 섹션 (90) 및 제 2 슬라이스 (98) 에 포함될 수도 있는 제 2 섹션 (92) 사이의 경계는 2의 CU 깊이에 위치된 잎 노드들 (58B 및 58C) 사이에 위치된다.

[0064] 도 3b에 도시된 예는 LCU (80) 가 분리되는 세분도를 개념적으로 더 도시한다. 예를 들어, 본 개시물은 슬라이스를 생성하는 경우에 LCU가 나누어지는 정도로서 "세분도"를 일반적으로 지칭할 수도 있다. 도 3b에 도시된 바와 같이, LCU (80) 의 서브 CU들 (84) 은 제 1 섹션 (90) 및 제 2 섹션 (92) 사이의 경계가 위치되게 하는 최소 CU들이다. 다시 말하면, 제 1 섹션 (90) 이 제 2 섹션 (92) 으로부터 분리되게 하는 경계는 서브 CU들 (84A/84B) 및 서브 CU들 (84C/84D) 사이에 위치된다. 따라서, 이 예에서, 슬라이스 (96) 의 최종 CU는 서브 CU (84B) 인 반면, 슬라이스 (98) 의 초기 CU는 서브 CU (84C) 이다.

[0065] LCU (80) 보다 작은 CU 세분도를 이용하여 슬라이스들을 생성하는 것은 특정 사이즈의 슬라이스 (예컨대, 소정 양의 데이터) 를 형성하는 것을 시도하는 경우에 유연성을 제공할 수도 있다. 더구나, 위에서 지적했듯이, 본 개시물의 기법들에 따라 프레임을 슬라이스들로 분할하는 것은 압축된 비디오 데이터를 특정하는데 필요한 슬라이스들의 수를 감소시킬 수도 있다. 압축된 비디오 데이터를 특정하는데 필요한 슬라이스들의 수를 감소시키는 것은 오버헤드 데이터 (예컨대, 슬라이스 헤더들에 연관된 오버헤드) 를 감소시키며, 이에 의해 오버헤드 데이터의 양이 압축된 비디오 데이터의 양에 비하여 감소함에 따라 압축 효율을 개선시킬 수도 있다.

[0066] LCU (80) 를 포함하는 프레임을 독립적으로 디코딩가능한 슬라이스들 (96 및 98) 로 분할하는 경우, 본 개시물의 양태들에 따르면, LCU (80) 에 대한 계층적 쿼드트리 정보는 각각의 독립적으로 디코딩가능한 슬라이스로 분리되고 표현될 수도 있다. 예를 들어, 위에서 언급했듯이, 쿼드트리 (50) 의 노드들에 대한 데이터는 노드에 대응하는 CU가 분할되는지의 여부를 서술할 수도 있다. CU가 분할된다면, 4 개의 부가적인 노드들이 쿼드트리 (50) 에 존재할 수도 있다. 일부 예들에서, 쿼드트리의 노드는 다음의 의사코드와 유사하게 구현될 수도 있다:

```
quadtree_node {
        boolean split_flag(1);
        // signaling data
        if (split_flag) {
                quadtree_node child1;
                quadtree_node child2;
                quadtree_node child3;
                quadtree_node child4;
        }
}
```

[0067]

[0068] split_flag 값은 현재 노드에 대응하는 CU가 분할되는지의 여부를 나타내는 1 비트 값일 수도 있다. CU가 분할되지 않으면, split_flag 값은 '0'일 수도 있는 반면, CU가 분할된다면, split_flag 값은 '1'일 수도 있다. 쿼드트리 (50) 의 예에 관하여, 분할 플래그 값들의 어레이는 10011000001000000일 수도 있다.

[0069] 쿼드트리 정보, 이를테면 LCU (80) 에 연관된 쿼드트리 (50) 는, LCU (80) 를 포함하는 슬라이스의 시작부분에서 통상 제공된다. 그러나, LCU (80) 가 상이한 슬라이스들로 나누어지고 쿼드트리 정보를 포함하는 슬라이스가 손실되거나 또는 손상된다면, 비디오 디코더는 제 2 슬라이스 (98) (예컨대, 쿼드트리 정보없는 슬라이스) 에 포함된 LCU (80) 의 일 부분을 적절히 디코딩하지 못할 수도 있다. 다시 말하면, 비디오 디코더는 LCU (80) 의 나머지가 서브 CU들로 분할되는 방법을 식별하지 못할 수도 있다.

[0070] 본 개시물의 양태들은 상이한 슬라이스들로 분할된 LCU, 이를테면 LCU (80) 에 대한 계층적 쿼드트리 정보를 분리하는 것과, 쿼드트리 정보의 분리된 부분들을 각각의 슬라이스에 제공하는 것을 포함한다. 예를 들어, 비디오 인코더 (20) 는 LCU (80) 의 시작부분에서 분할 플래그들의 형태로 쿼드트리 정보를 통상 제공할 수도 있다. 그러나, LCU (80) 에 대한 쿼드트리 정보가 이런 식으로 제공된다면, 제 1 섹션 (90) 은 분할 플래그들의 모두를 포함할 수도 있지만 제 2 섹션 (92) 은 어떠한 분할 플래그들도 포함하지 않는다. 제 1 슬라이스 (96) (이것은 제 1 섹션 (90) 을 포함함) 가 손실되거나 또는 손상된다면, 제 2 슬라이스 (98) (이것은 제 2 섹션 (92) 을 포함함) 는 적절히 디코딩되지 못할 수도 있다.

[0071] LCU (80) 를 상이한 슬라이스들로 분할하는 경우, 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 는 또한, 제 1 섹션 (90) 에 적용가능한 쿼드트리 정보가 제 1 슬라이스 (96) 에 제공되고 제 2 섹션 (92) 에 적용가능한 쿼드트리 정보가 제 2 슬라이스 (96) 에 제공되도록 연관된 쿼드트리 정보를 분리할 수도 있다. 다시 말하면, LCU (80) 를 제 1 섹션 (90) 및 제 2 섹션 (92) 으로 분할하는 경우, 비디오 인코더 (20) 는 제 1 섹션 (90) 에 연관된 분할 플래그들을 제 2 섹션 (92) 에 연관된 분할 플래그들로부터 분리할 수도 있다. 비디오 인코더 (20) 는 그 다음에 제 1 섹션 (90) 에 대한 분할 플래그들을 제 1 슬라이스 (96) 에 그리고 제 2 섹션 (92) 에 대한 분할 플래그들을 제 2 슬라이스 (98) 에 제공할 수도 있다. 이런 식으로, 제 1 슬라이스 (96) 가 손상되거나 또는 손실된다면, 비디오 디코더는 제 2 슬라이스 (98) 에 포함되는 LCU (80) 의 나머지 부분을 여전히 적절히 디코딩할 수도 있다.

[0072] LCU에 대한 쿼드트리 정보의 부분만을 포함하는 LCU의 섹션을 적절히 디코딩하기 위해, 일부 예들에서, 비디오 디코더 (30) 는 LCU의 다른 섹션에 연관된 쿼드트리 정보를 재구성할 수도 있다. 예를 들어, 제 2 섹션 (92) 수신 시, 비디오 디코더 (30) 는 쿼드트리 (50) 의 누락 부분을 재구성할 수도 있다. 그렇게 하기 위해, 비디오 디코더 (30) 는 수신된 슬라이스의 제 1 CU의 인덱스 값을 식별할 수도 있다. 인덱스 값은 서브 CU가 속하는 사분역을 식별하며, 이에 의해 LCU 내에서의 서브 CU의 상대 포지션의 표시를 제공할 수도 있다.
다시 말하면, 도 3b에 도시된 예에서, 서브 CU (84A) 는 0의 인덱스 값을 가질 수도 있으며, 서브 CU (84B) 는 1의 인덱스 값을 가질 수도 있으며, 서브 CU (84C) 는 2의 인덱스 값을 가질 수도 있고, 서브 CU (84D) 는 3의 인덱스 값을 가질 수도 있다. 그런 인덱스 값들은 슬라이스 헤더 내에 신택스 엘리먼트들로서 제공될 수도 있다.

[0073] 따라서, 제 2 섹션 (92) 수신 시, 비디오 디코더 (30) 는 서브 CU (84C) 의 인덱스 값을 식별할 수도 있다.

비디오 디코더 (30) 는 그 다음에 서브 CU (84C) 가 좌측 아래 사분역에 속한다는 것과 서브 CU (84C) 의 부모 노드가 분할 플래그를 포함해야 한다는 것을 식별하기 위해 인덱스 값을 이용할 수도 있다. 다시 말하면, 서브 CU (84C) 가 인덱스 값을 갖는 서브 CU이기 때문에, 부모 CU는 분할 플래그를 반드시 포함한다.

[0074] 덧붙여서, 비디오 디코더 (30) 는 제 2 섹션 (92) 에 포함된 쿼드트리 (50) 의 노드들의 모두를 추론할 수도 있다. 일 예에서, 비디오 디코더 (30) 는 쿼드트리 (50) 의 수신된 부분을 이용하여 그리고 깊이우선 (depth-first) 쿼드트리 탐색 알고리즘을 이용하여 이러한 정보를 추론할 수도 있다. 깊이우선 탐색 알고리즘에 따르면, 비디오 디코더 (30) 는 확장된 노드가 잎 노드들을 가지지 않을 때까지 쿼드트리 (50) 의 수신된 부분의 제 1 노드를 확장한다. 비디오 디코더 (30) 는 아직 확장되지 않은 가장 최근의 노드로 복귀하기까지 확장된 노드를 탐색한다. 비디오 디코더 (30) 는 쿼드트리 (50) 의 수신된 부분의 모든 노드들이 확장되기까지 이런 식으로 계속한다.

[0075] LCU (80) 를 상이한 슬라이스들로 분할하는 경우, 비디오 인코더 (20) 는 또한 비디오 데이터 디코딩 시에 비디오 디코더 (30) 를 지원하기 위해 다른 정보를 제공할 수도 있다. 예를 들어, 본 개시물의 양태들은 비트스트림 내에 포함된 하나 이상의 신택스 엘리먼트들을 이용하여 슬라이스의 상대적 말단을 식별하는 것을 포함한다. 일 예에서, 비디오 인코더, 이를테면 비디오 인코더 (20) 는, 특정 CU가 슬라이스의 최종 CU (예컨대, 분할 전의 최종 CU) 인지의 여부를 표시하기 위해 1 비트의 슬라이스 말단 플래그를 생성하고 그 슬라이스 말단 플래그를 프레임의 각각의 CU에 제공할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 슬라이스 말단 플래그를 CU가 슬라이스의 상대적 말단에 위치되면 '0'의 값으로 그리고 CU가 슬라이스의 상대적 말단에 위치되면 '1'의 값으로 설정할 수도 있다. 도 3b에 도시된 예에서, 서브 CU (84B) 는 '1'의 슬라이스 말단 플래그를 포함할 것인 반면, 나머지 CU들은 '0'의 슬라이스 말단 플래그를 포함할 것이다.

[0076] 일부 예들에서, 비디오 인코더 (20) 는 프레임을 슬라이스들로 분할하는데 이용된 세분도 이상인 CU들에 대해서만 슬라이스 말단 표시 (예컨대, 슬라이스 말단 플래그) 를 제공할 수도 있다. 도 3b에 도시된 예에서, 비디오 인코더 (20) 는 슬라이스 말단 플래그만을 16 화소 바이 16 화소 세분도 이상인 CU들, 즉, CU들 (82A, 82B, 84A-84D, 및 86A-86C) 에 제공할 수도 있다. 이런 식으로, 비디오 인코더 (20) 는 슬라이스 말단 플래그가 프레임의 모든 CU에 제공되는 접근법을 통해 비트 절약을 달성할 수도 있다.

[0077] 개별 양자화 데이터는 또한 LCU, 이를테면 LCU (80) 가 상이한 슬라이스들로 분할되는 예들에서 각각의 슬라이스에 대해 제공될 수도 있다. 예를 들어, 위에서 지적했듯이, 양자화는 LCU 레벨에서 정의될 수도 있는 양자화 파라미터 (QP) (예컨대, 그것은 델타 QP에 의해 식별될 수도 있음) 에 따라 적용될 수도 있다. 그러나, 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 는 상이한 슬라이스들로 분할된 LCU의 각각의 부분에 대한 델타 QP 값을 표시할 수도 있다. 도 3b에 도시된 예에서, 비디오 인코더 (20) 는 제 1 슬라이스 (96) 및 제 2 슬라이스 (98) 에 각각 포함될 수도 있는 제 1 섹션 (90) 및 제 2 섹션 (92) 에 대해 별개의 델타 QP들을 제공할 수도 있다.

[0078] 도 3a 및 도 3b의 특정 양태들이 설명의 목적을 위해 비디오 인코더 (20) 및 비디오 디코더 (30) 에 관해 설명되었지만, 다른 비디오 코딩 단위들, 이를테면 다른 프로세서들, 프로세싱 유닛들, 인코더/디코더들 (CODECs) 을 포함하는 하드웨어 기반 코딩 유닛들 등이 또한 도 3a 및 도 3b에 관해 설명된 예들 및 기법들을 수행하도록 구성될 수도 있다는 것이 이해되어야 한다.

[0079] 도 4는 비디오 데이터의 프레임을 본 개시물에서 설명되는 독립적으로 디코딩가능한 부분들로 분할하기 위한 기법들 중 임의의 것 또는 모두를 구현할 수도 있는 비디오 인코더 (20) 의 일 예를 도시하는 블록도이다. 대체로, 비디오 인코더 (20) 는 비디오 프레임들 내의 CU들의 인트라- 및 인터-코딩을 수행할 수도 있다. 인트라 코딩은 공간 예측에 의존하여, 주어진 비디오 프레임 내의 비디오에서의 공간적 리던던시를 감소시키거나 또는 제거한다. 인터-코딩은 시간적 예측에 의존하여, 비디오 시퀀스의 현재 프레임 및 이전에 코딩된 프레임들 사이의 시간적 리던던시를 감소시키거나 또는 제거한다. 인트라-모드 (I-모드) 는 여러 가지 공간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있고, 단-방향 예측 (P-모드) 또는 양방향 예측 (B-모드) 과 같은 인터-모드들은 여러 가지 시간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다.

[0080] 도 4에 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩될 비디오 프레임 내의 현재 비디오 블록을 수신한다. 도 4의 예에서, 비디오 인코더 (20) 는 움직임 보상 유닛 (144), 움직임 추정 유닛 (142), 인트라 예측 유닛 (146), 참조 프레임 저장부 (164), 합산기 (150), 변환 유닛 (152), 양자화 유닛 (154), 및 엔트로피 코딩 유닛 (156) 을 구비한다. 도 4에 예시된 변환 유닛 (152) 은 실제 변환을 수행하는 유닛이며, CU의 TU와 혼동하면 안된다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역 양자화 유닛 (158), 역 변환 유

닛 (160), 및 합산기 (162) 를 구비한다. 디블로킹 (deblocking) 필터 (도 4에서 미도시) 가 또한 재구성된 비디오로부터 블록형 (blockiness) 아티팩트들을 제거하기 위해 필터 블록 경계들 내에 포함될 수도 있다. 원한다면, 디블로킹 필터는 통상 합산기 (162) 의 출력을 필터링할 것이다.

[0081] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 그 프레임 또는 슬라이스는 다수의 비디오 블록들, 예컨대, 최대 코딩 단위들 (LCUs) 로 나누어질 수도 있다. 움직임 추정 유닛 (142) 과 움직임 보상 유닛 (144) 은 시간적 압축을 제공하기 위해 하나 이상의 참조 프레임들에서 하나 이상의 블록들에 대해 수신된 비디오 블록의 인터-예측 코딩을 수행한다. 인트라 예측 유닛 (146) 은 공간적 압축을 제공하기 위해 코딩될 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 대해, 수신된 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다.

[0082] 모드 선택 유닛 (140) 은 코딩 모드들, 예컨대 인트라 또는 인터 중 하나를, 에러 결과들 대 각각의 코딩 모드 하에서의 비디오 데이터를 시그널링하는데 필요한 비트들의 수 (예컨대, 때때로 레이트-왜곡이라 지칭됨) 에 기초하여 선택하고, 결과적인 인트라- 또는 인터-코딩된 블록을 잔차 블록 데이터를 생성하기 위해 합산기 (150) 에 그리고 참조 프레임에서 사용하기 위한 인코딩된 블록을 재구성하기 위해 합산기 (162) 에 제공할 수도 있다. 일부 비디오 프레임들은 지정된 I-프레임들일 수도 있으며, 여기서 I-프레임에서의 모든 블록들은 인트라-예측 모드로 인코딩된다. 일부 경우들에서, 인트라 예측 유닛 (146) 은, 예컨대, 움직임 추정 유닛 (142) 에 의해 수행된 움직임 탐색이 결과적으로 블록의 충분한 예측이 되지 않는 경우에 P- 또는 B-프레임에서 블록의 인트라-예측 인코딩을 수행할 수도 있다.

[0083] 코딩 모드들 중 하나를 선택하는 것에 더하여, 일부 예들에 따르면, 비디오 인코더 (20) 는 LCU보다 작을 수도 있는, 비디오 데이터의 프레임을 분할하는 세분도를 결정하는 것과 같은 다른 기능들을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 갖가지 슬라이스 구성들에 대한 (예컨대, 소정의 왜곡을 초과하는 일 없이 압축을 최대화하는 것을 시도하여) 레이트-왜곡을 계산하고 최선의 결과를 산출하는 세분도를 선택할 수도 있다. 비디오 인코더 (20) 는 세분도를 선택하는 경우에 타겟 슬라이스 사이즈를 고려할 수도 있다. 예를 들어, 위에서 지적했듯이, 일부 경우들에서 특정 사이즈로 된 슬라이스들을 형성하는 것이 바람직할 수도 있다. 하나의 그러한 예는 네트워크를 통해 슬라이스를 송신하기 위해 준비될 수도 있다. 비디오 인코더 (20) 는 타겟 사이즈를 엄밀하게 매칭시키기 위한 시도 시 비디오 데이터의 프레임들을 슬라이스들로 분할하는 세분도를 결정할 수도 있다.

[0084] 비디오 인코더 (20) 가 비디오 데이터의 프레임을 분할할 세분도를 결정하는 예들에서, 비디오 인코더 (20) 는 그러한 세분도를 표시할 수도 있다. 다시 말하면, 비디오 인코더 (20) (이를테면 모드 선택 유닛 (140), 엔트로피 코딩 유닛 (156), 또는 비디오 인코더 (20) 의 다른 유닛) 는 비디오 데이터 디코딩 시에 비디오 디코더를 지원하기 위해 세분도의 표시를 제공할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 분할이 일어날 수도 있는 CU 깊이에 따라 세분도를 식별할 수도 있다.

[0085] 설명의 목적을 위해, 비디오 데이터의 프레임이 128 화소 바이 128 화소의 사이즈인 하나 이상의 LCU들을 가진다고 가정한다. 이 예에서, 비디오 인코더 (20) 는, 예를 들어, 타겟 슬라이스 사이즈를 달성하기 위하여 프레임이 32 화소 바이 32 화소의 세분도에서 슬라이스들로 분할될 수도 있다고 결정할 수도 있다. 비디오 인코더 (20) 는 그러한 세분도를 슬라이스 분할이 일어날 수도 있는 계층 깊이에 따라 표시할 수도 있다. 다시 말하면, 도 3a 및 도 3b에 도시된 계층적 쿼드트리 배열에 따르면, 32 화소 바이 32 화소 서브 CU는 2의 CU 깊이를 가진다. 따라서, 이 예에서, 비디오 인코더 (20) 는 슬라이스 분할이 2의 CU 깊이에서 일어날 수도 있다는 것을 표시함으로써 슬라이스 세분도를 시그널링할 수도 있다.

[0086] 일 예에서, 비디오 인코더 (20) 는 비디오 데이터의 프레임이 슬라이스들로 분할될 수도 있는 세분도의 표시를 화상 파라미터 세트 (PPS) 내에 제공할 수도 있다. 예를 들어, 백그라운드로, 비디오 인코더 (20) 는 네트워크를 통한 송신을 위한 압축된 비디오 데이터를 이른바 "네트워크 추상화 계층 단위들" 또는 NAL (network abstraction layer) 단위들로 포맷할 수도 있다. 각각의 NAL 단위는 NAL 단위에 저장된 데이터의 유형을 식별하는 헤더를 포함할 수도 있다. NAL 단위들에 일반적으로 저장되는 2 개의 데이터 유형들이 있다. NAL 단위에 저장된 제 1 데이터 유형은 압축된 비디오 데이터를 포함하는 비디오 코딩 계층 (video coding layer; VCL) 데이터이다. NAL 단위에 저장되는 제 2 데이터 유형은 비-VCL 데이터라고 지칭되며, 그것은 다수의 NAL 단위들에 공통인 헤더 데이터를 정의하는 파라미터 세트들과 같은 부가적인 정보 및 추가 향상 정보 (supplemental enhancement information; SEI) 를 포함한다. 예를 들어, 파라미터 세트들은 시퀀스-레벨 헤더 정보를 (예컨대, 시퀀스 파라미터 세트들 (sequence parameter set; SPS) 내에) 그리고 드물게 변경되는

화상-레벨 헤더 정보를 (예컨대, 화상 파라미터 세트들 (PPS) 내에) 포함할 수도 있다. 파라미터 세트들 내에 포함된 드물게 변경되는 정보는 각각의 시퀀스 또는 화상에 대해 반복될 필요는 없으며, 이에 의해 코딩 효율을 개선시킨다. 더욱이, 파라미터 세트들의 이용은 헤더 정보의 대역 외 송신을 할 수 있게 함으로써, 에러 내성을 위한 중복적인 송신들이 필요 없게 한다.

[0087] 하나의 예에서, 비디오 데이터의 프레임이 슬라이스들로 분할될 수도 있는 세분도의 표시는 아래의 표 1에 따라 표시될 수도 있다:

[0088] 표 1 - pic_parameter_set_rbsp()

pic_parameter_set_rbsp( ) {	C	Descriptor
pic_parameter_set_id	1	ue(v)
seq_parameter_set_id	1	ue(v)
entropy_coding_mode_flag	1	u(1)
num_ref_idx_l0_default_active_minus1	1	ue(v)
num_ref_idx_l1_default_active_minus1	1	ue(v)
pic_init_qp_minus26 /* relative to 26 */	1	se(v)
slice_granu_CU_depth	1	ue(v)
constrained_intra_pred_flag	1	u(1)
for(i=0;i<15; i++){		
numAllowedFilters[i]	1	ue(v)
for(j=0;j<numAllowedFilters;j++){		
filtIdx[i][j]	1	ue(v)
}		
}		
rbsp_trailing_bits( )	1	
}		

[0089]

[0090] 표 1에 도시된 예에서, slice_granu_CU_depth는 비디오 데이터의 프레임을 슬라이스들로 분할하는데 이용된 세분도를 특정할 수도 있다. 예를 들어, slice_granu_CU_depth는 LCU (예컨대, LCU = 깊이 0) 에 비교하여 슬라이스 분할이 일어날 수도 있는 계층 깊이를 식별하는 것에 의해 프레임을 슬라이스들로 분할하는데 이용된 세분도로서 CU 깊이를 특정할 수도 있다. 본 개시물의 양태들에 따르면, 슬라이스는 (예컨대, 연관된 계층적 쿼드트리 구조에서의 모든 CU들을 포함한) 일련의 LCU들과 불완전 LCU를 포함할 수도 있다. 불완전 LCU는 max_coding_unit_width >> slice_granu_CU_depth 바이 max_coding_unit_height >> slice_granu_CU_depth 만큼 작지만, 더 작지는 않은 사이즈를 갖는 하나 이상의 완전한 CU들을 포함할 수도 있다. 예를 들어, 슬라이스는 max_coding_unit_width >> slice_granu_CU_depth 바이 max_coding_unit_height >> slice_granu_CU_depth 미만이고 슬라이스 내에 완전히 포함되는 LCU에 속하지 않는 사이즈를 갖는 CU를 포함할 수 없다. 다시 말하면, 슬라이스 경계는 max_coding_unit_width >> slice_granu_CU_depth 바이 max_coding_unit_height >> slice_granu_CU_depth의 CU 사이즈 이하인 CU 내에서 발생하지 않을 수도 있다.

[0091] 비디오 인코더 (20) 가 비디오 데이터의 프레임을 슬라이스들로 분할하기 위해 LCU보다 작은 세분도를 결정하는 예들에서, 비디오 인코더 (20) 는 상이한 슬라이스들로 분할 중인 LCU에 대한 계층적 쿼드트리 정보를 분리하고 쿼드트리 정보의 분리된 부분들을 각각의 슬라이스에 제공할 수도 있다. 예를 들어, 도 3a 및 도 3b에 관해 위에서 설명된 바와 같이, 비디오 인코더 (20) 는 분할 중인 LCU의 각각의 선택에 연관된 분할 플래그들을 슬라이스들 사이에서 분리할 수도 있다. 비디오 인코더 (20) 는 그 다음에 분할 LCU의 제 1 섹션에 연관된 분할 플래그들을 제 1 슬라이스에 그리고 분할 LCU의 다른 섹션에 연관된 분할 플래그들을 제 2 슬라이스에 제공할 수도 있다. 이런 식으로, 제 1 슬라이스가 손상되거나 또는 손실된다면, 비디오 디코더는 제 2 슬라이스에 포함되는 LCU의 나머지 부분을 여전히 적절히 디코딩할 수도 있다.

[0092] 부가적으로 또는 대안으로, 비디오 인코더 (20) 는 하나 이상의 신택스 엘리먼트들을 이용하여 슬라이스의 상대적 말단을 식별할 수도 있다. 예를 들어, 비디오 인코더 (20) 는, 특정 CU가 슬라이스의 최종 CU (예컨대,

분할 전의 최종 CU) 인지의 여부를 표시하기 위해, 1 비트의 슬라이스 말단 플래그를 생성하고 슬라이스 말단 플래그를 프레임의 각각의 CU에 제공할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 슬라이스 말단 플래그를 CU가 슬라이스의 상대적 말단에 위치되면 '0'의 값으로 그리고 CU가 슬라이스의 상대적 말단에 위치되면 '1'의 값으로 설정할 수도 있다.

[0093] 일부 예들에서, 비디오 인코더 (20) 는 프레임을 슬라이스들로 분할하는데 이용된 세분도 이상인 CU들에 대해서만 슬라이스 말단 표시 (예컨대, 슬라이스 말단 플래그) 를 제공할 수도 있다. 예를 들어, 설명의 목적을 위해 비디오 인코더 (20) 는 비디오 데이터의 프레임을 슬라이스들로 분할할 세분도를 32 화소 바이 32 화소로 결정하며 LCU 사이즈는 64 화소 바이 64 화소라고 가정한다. 이 예에서, 모드 선택 유닛 (140) 은 슬라이스 말단 플래그에만 32 화소 바이 32 화소 이상의 사이즈인 CU들을 제공할 수도 있다.

[0094] 일 예에서, 비디오 인코더 (20) 는 아래에 도시된 표 2에 따라 슬라이스 말단 플래그를 생성할 수도 있다:

[0095] 표 2 - coding_tree(x0, y0, log2CUSize)

coding_tree(x0, y0, log2CUSize) {	**Descriptor**
if(x0 + (1 << log2CUSize) <= $PicWidthInSamples_L$ && y0 + (1 << log2CUSize) <= $PicHeightInSamples_L$ && log2CUSize > Log2MinCUSize && cuAddress(x0 ,y0) >= sliceAddress)	
split_coding_unit_flag[x0][y0]	u(1) | ae(v)

[0096]

if(adaptive_loop_filter_flag && alf_cu_control_flag) {	
cuDepth = Log2MaxCUSize – log2CUSize	
if(cuDepth <= alf_cu_control_max_depth)	
if(cuDepth == alf_cu_control_max_depth ||	
split_coding_unit_flag[x0][y0] == 0)	
AlfCuFlagIdx++	
}	
if(split_coding_unit_flag[x0][y0]) {	
x1 = x0 + ((1 << log2CUSize) >> 1)	
y1 = y0 + ((1 << log2CUSize) >> 1)	
if(cuAddress(x1,y0) > sliceAddress)	
moreDataFlag = coding_tree(x0, y0, log2CUSize – 1)	
if(cuAddress(x0,y1) > sliceAddress && moreDataFlag && x1 < PicWidthInSamples$_L$)	
moreDataFlag = coding_tree(x1, y0, log2CUSize – 1)	
if(cuAddress(x1,y1) > sliceAddress && moreDataFlag && y1 < PicHeightInSamples$_L$) {	
moreDataFlag = coding_tree(x0, y1, log2CUSize – 1)	
if(moreDataFlag && x1 < PicWidthInSamples$_L$ && y1 < PicHeightInSamples$_L$)	
moreDataFlag = coding_tree(x1, y1, log2CUSize – 1)	
} else {	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlag[x0][y0] = alf_cu_flag[AlfCuFlagIdx]	
coding_unit(x0, y0, log2CUSize)	
if(!entropy_coding_mode_flag)	
moreDataFlag = more_rbsp_data( )	
else {	
if(log2CUsize >= (Log2MaxCUSize - slice_granu_CU_depth){	
end_of_slice_flag	ae(v)
moreDataFlag = !end_of_slice_flag	
}	

[0097]

else	
{	
moreDataFlag = 1;	
}	
}	
}	
return moreDataFlag	
}	

[0098]

[0099] 본 개시물의 특정 양태들이 일반적으로 비디오 인코더 (20) 에 관해 설명되었지만, 그런 양태들은 모드 선택 유닛 (140) 또는 비디오 인코더 (20) 의 하나 이상의 다른 유닛들과 같은 비디오 인코더 (20) 의 하나 이상의 유

닛들에 의해 수행될 수도 있다는 것이 이해되어야 한다.

[0100] 움직임 추정 유닛 (142) 과 움직임 보상 유닛 (144) 은 고도로 통합될 수도 있지만 개념상의 목적들을 위해 별개로 예시된다. 움직임 추정은 인터 코딩을 위해 비디오 블록들에 대한 움직임을 추정하는, 움직임 벡터들을 생성하는 프로세스이다. 움직임 벡터는, 예를 들어, 참조 프레임 내의 참조 샘플에 대한 현재 프레임 내의 예측 단위의 변위 (displacement) 를 표시할 수도 있다. 참조 샘플은 차의 절대값 합 (SAD), 차의 제곱합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 화소 차이의 측면에서 코딩된 PU를 포함하는 CU의 부분에 엄밀하게 매칭된다고 생각되는 블록이다. 움직임 보상 유닛 (144) 에 의해 수행되는 움직임 보상은 움직임 추정에 의해 결정된 움직임 벡터에 기초하여 예측 단위에 대한 값들을 페치하는 것 또는 생성하는 것을 수반할 수도 있다. 다시, 움직임 추정 유닛 (142) 과 움직임 보상 유닛 (144) 은 일부 예들에서 기능적으로 통합될 수도 있다.

[0101] 움직임 추정 유닛 (142) 은 예측 단위와 참조 프레임 저장부 (164) 에 저장된 참조 프레임의 참조 샘플들을 비교함으로써 인터-코딩된 프레임의 예측 단위에 대한 움직임 벡터를 계산한다. 일부 예들에서, 비디오 인코더 (20) 는 참조 프레임 저장부 (164) 에 저장된 참조 프레임들의 부 정수 (sub-integer) 화소 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 프레임의 1/4 화소 위치들, 1/8 화소 위치들, 또는 다른 분수 (fractional) 화소 위치들의 값들을 계산할 수도 있다. 그러므로, 움직임 추정 유닛 (142) 은 풀 (full) 화소 위치들 및 분수 화소 위치들에 대한 움직임 검색을 수행하고 분수 화소 정밀도를 갖는 움직임 벡터를 출력할 수도 있다. 움직임 추정 유닛 (142) 은 계산된 움직임 벡터를 엔트로피 코딩 유닛 (156) 및 움직임 보상 유닛 (144) 에 전송한다. 움직임 벡터에 의해 식별된 참조 프레임의 부분은 참조 샘플이라고 지칭될 수도 있다. 움직임 보상 유닛 (144) 은, 예컨대, PU에 대한 움직임 벡터에 의해 식별된 참조 샘플을 취출함으로써 현재 CU의 예측 단위에 대한 예측 값을 계산할 수도 있다.

[0102] 인트라 예측 유닛 (146) 은 움직임 추정 유닛 (142) 및 움직임 보상 유닛 (144) 에 의해 수행된 인터-예측에 대한 대안으로서, 수신된 블록을 코딩하는 인트라-예측을 수행할 수도 있다. 인트라 예측 유닛 (146) 은 블록에 대한 좌에서 우로, 위에서 아래로의 인코딩 순서를 가정하여, 이웃하는 이전에 코딩된 블록, 예컨대, 현재 블록의 위, 오른쪽 위, 왼쪽 위, 또는 왼쪽의 블록에 대해, 수신된 블록을 인코딩할 수도 있다. 인트라 예측 유닛 (146) 은 다양한 상이한 인트라-예측 모드들로 구성될 수도 있다. 예를 들어, 인트라 예측 유닛 (146) 은 인코딩되는 CU의 사이즈에 기초하여, 특정 수의 예측 모드들, 예컨대, 35 개 예측 모드들로 구성될 수도 있다.

[0103] 인트라 예측 유닛 (146) 은, 예를 들어, 갖가지 인트라-예측 모드들에 대한 (예컨대, 소정의 왜곡을 초과하는 일 없이 압축 최대화를 시도하여) 레이트-왜곡을 계산하는 것 및 최선의 결과를 산출하는 모드를 선택하는 것에 의해 이용가능한 인트라-예측 모드들로부터 인트라-예측 모드를 선택할 수도 있다. 인트라-예측 모드들은 공간적으로 이웃하는 화소들의 값들을 조합하고 조합된 값들을 PU를 예측하는데 이용되는 예측 블록에서의 하나 이상의 화소 포지션들에 적용하는 기능들을 포함할 수도 있다. 일단 예측 블록에서의 모든 화소 포지션들에 대한 값들이 계산되었다면, 인트라 예측 유닛 (146) 은 PU 및 예측 블록 사이의 화소 차이들에 기초하여 예측 모드에 대한 에러 값을 계산할 수도 있다. 인트라 예측 유닛 (146) 은 허용가능 에러 값 대 비디오 데이터를 시그널링하는데 필요한 비트들을 산출하는 인트라-예측 모드가 발견되기까지 인트라-예측 모드들을 테스트하는 것을 계속할 수도 있다. 인트라 예측 유닛 (146) 은 그 다음에 PU를 합산기 (150) 에 전송할 수도 있다.

[0104] 비디오 인코더 (20) 는 코딩 중인 원래의 비디오 블록으로부터 움직임 보상 유닛 (144) 또는 움직임 예측 유닛 (146) 에 의해 계산된 예측 데이터를 감산함으로써 잔차 블록을 형성한다. 합산기 (150) 는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 잔차 블록은 값들의 2차원 매트릭스에 대응할 수도 있으며, 여기서 잔차 블록에서의 값들의 수는 그 잔차 블록에 대응하는 PU에서의 화소들의 수와 동일하다. 잔차 블록에서의 값들은 예측 블록에서 그리고 코딩할 원본 블록에서 병치된 (collocated) 화소들 사이의 차이들에 대응할 수도 있다.

[0105] 변환 유닛 (152) 은 변환, 이를테면 이산 코사인 변환 (DCT), 정수 변환, 또는 개념적으로 유사한 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 유닛 (152) 은 다른 변환들, 이를테면 DCT와 개념적으로 유사한, H.264 표준에 의해 정의된 것들을 수행할 수도 있다. 웨이브릿 변환들, 정수 변환들, 서브-밴드 변환들 또는 다른 유형들의 변환들이 또한 이용될 수 있다. 어쨌든, 변환 유닛 (152) 은 잔차 블록에 변환을 적용하여, 잔차 변환 계수들의 블록을 생성한다. 변환 유닛 (152) 은 화소 값 도메인으로부터의 잔차 정보를 변환 도메인, 이를테면 주파수 도메인으로 변환할 수도 있다.

[0106] 양자화 유닛 (154) 은 잔차 변환 계수들을 양자화하여 비트 레이트를 더욱 감소시킨다. 양자화 프로세스는 계수들의 일부 또는 전부에 연관된 비트 깊이를 감소시킬 수도 있다. 양자화 정도는 양자화 파라미터 (QP) 를 조정함으로써 변경될 수도 있다. 일부 예들에서, QP는 LCU 레벨에서 정의될 수도 있다. 따라서, 동일한 레벨의 양자화가 LCU 내의 CU들의 상이한 PU들에 연관된 TU들에서의 모든 변환 계수들에 적용될 수도 있다. 그러나, QP 자체를 시그널링하기 보다는, QP에서의 변화 (즉, 델타) 가 LCU와 함께 시그널링될 수도 있다. 델타 QP는 그 LCU에 대한 양자화 파라미터에서의, 일부 기준 QP, 이를테면 이전에 통신된 LCU의 QP에 관한 변화를 정의한다.

[0107] LCU가 2 개의 슬라이스들 사이에서 나누어지는 예들에서, 본 개시물의 양태들에 따라, 양자화 유닛 (154) 은 분리된 LCU의 각각의 부분에 대해 별개의 QP들 (또는 델타 QP들) 을 정의할 수도 있다. 설명의 목적을 위해, LCU가 2 개의 슬라이스들 사이에서 그 LCU의 제 1 섹션은 제 1 슬라이스에 포함되고 그 LCU의 제 2 섹션은 제 2 슬라이스에 포함되도록 분할된다고 가정한다. 이 예에서, 양자화 유닛 (154) 은 LCU의 제 1 섹션을 위한 제 1 델타 QP와, 제 1 델타 QP와는 별개이며 LCU의 제 2 섹션을 위한 제 2 델타 QP를 정의할 수도 있다. 일부 예들에서, 제 1 슬라이스에 제공된 델타 QP는 제 2 슬라이스에 제공된 델타 QP와는 상이할 수도 있다.

[0108] 일 예에서, 양자화 유닛 (154) 은 아래에 도시된 표 3에 따라 델타 QP 값들의 표시를 제공할 수도 있다:

[0109] 표 3 - coding_unit(x0, y0, currCodingUnitSize)

[0110]

coding_unit(x0, y0, currCodingUnitSize) {	C	Descriptor
if (firstCUFlag || currCodingUnitSize >=MinQPCodingUnitSize) {		
cu_QP_delta;	**2**	**u(1) | e(v)**
firstCUFlag = false;		
}		

if(x0+currCodingUnitSize < $PicWidthInSamples_L$ && y0+currCodingUnitSize < $PicHeightInSamples_L$ &&		
currCodingUnitSize > MinCodingUnitSize)		
split_coding_unit_flag	2	u(1) | ae(v)
if(split_coding_unit_flag) {		
splitCodingUnitSize = currCodingUnitSize >> 1		
x1 = x0 + splitCodingUnitSize		
y1 = y0 + splitCodingUnitSize		
coding_unit(x0, y0, splitCodingUnitSize)	2 | 3 | 4	
if(x1 < $PicWidthInSamples_L$)		
coding_unit(x1, y0, splitCodingUnitSize)	2 | 3 | 4	
if(y1 < $PicHeightInSamples_L$)		
coding_unit(x0, y1, splitCodingUnitSize)	2 | 3 | 4	
if(x1 < $PicWidthInSamples_L$ && y1 < $PicHeightInSamples_L$)		
coding_unit(x1, y1, splitCodingUnitSize)	2 | 3 | 4	
} else {		
prediction_unit(x0, y0, currCodingUnitSize)	2	
if(PredMode != MODE_SKIP || !(PredMode == MODE_INTRA && **planar_flag** == 1))		
if(entropy_coding_mode_flag) {		
transform_unit_tree(x0, y0, currCodingUnitSize, 0)	3 | 4	
transform_unit_coeff(x0, y0, currCodingUnitSize, 0, 0)	3 | 4	
transform_unit_coeff(x0, y0, currCodingUnitSize, 0, 1)	3 | 4	
transform_unit_coeff(x0, y0, currCodingUnitSize, 0, 2)	3 | 4	
} else		
transform_unit_vlc(x0, y0, currCodingUnitSize)	3 | 4	
}		
}		

[0111]

[0112] 표 2의 예에서, cu_QP_delta는 CU 계층에서의 QP_Y의 값을 변경시킬 수 있다. 다시 말하면, 별도의 cu_QP_delta 값이 상이한 슬라이스들로 분할된 LCU의 2 개의 상이한 섹션들에 대해 정의될 수도 있다. 일부 예들에 따르면, cu_QP_delta의 디코딩된 값은 -26 내지 +25의 범위 내에 있을 수도 있다. cu_QP_delta 값이 CU에 대해 제공되지 않으면, 비디오 디코더는 cu_QP_delta 값이 0과 동일하다고 추론할 수도 있다.

[0113] 일부 예들에서, QP_Y 값은 다음의 수학식 (1) 에 따라 도출될 수도 있으며, 여기서 $QP_{Y,PREV}$ 는 현재 슬라이스의 디코딩 순서에서의 이전의 CU의 루마 (luma) 양자화 파라미터 (QP_Y) 이다.

[0114] $$QP_Y = (QP_{Y,PREV} + cu\_qp\_delta + 52) \% 52 \quad (1)$$

[0115] 덧붙여서, 슬라이스의 제 1 CU에 대해, $QP_{Y,PREV}$ 값은 초기에는, 양자화 파라미터가 수정되기까지 슬라이스의 모든 블록들에 대해 이용되는 초기 QP_Y일 수도 있는 $SliceQP_Y$와 동일하게 설정될 수도 있다. 더구나,

firstCUFlag는 각각의 슬라이스의 시작에서 '참'으로 설정될 수도 있다.

[0116] 본 개시물의 일부 양태들에 따르면, 양자화 유닛 (154) 은 QP_Y 값이 할당될 수도 있는 최소 CU 사이즈를 결정할 수도 있다. 예를 들어, 양자화 유닛 (154) 은 MinQPCodingUnitSize 이상인 CU들에 대해서만 QP 값을 설정할 수도 있다. 일부 예들에서, MinQPCodingUnitSize가 MaxCodingUnitSize (예컨대, 최대 지원된 CU (LCU) 의 사이즈) 와 동일한 경우, 양자화 유닛 (154) 은 슬라이스에서의 첫 번째 CU 및 LCU들에 대한 QP 값만을 시그널링할 수도 있다. 다른 예에서, LCU 및/또는 슬라이스의 첫 번째 CU에 대한 델타 QP 값을 시그널링하는 대신, 양자화 유닛 (154) 은 특정 시퀀스 (예컨대, 프레임들의 시퀀스) 에 대해 고정될 수도 있는, 델타 QP가 설정될 수도 있는 최소 QP CU 사이즈를 시그널링할 수도 있다. 예를 들어, 양자화 유닛 (154) 은, 예를 들어, 화상 파라미터 세트 (PPS) 또는 시퀀스 파라미터 세트 (SPS) 와 같은 파라미터 세트에서 최소 QP CU 사이즈를 시그널링할 수도 있다.

[0117] 다른 예에서, 양자화 유닛 (154) 은 CU 깊이에 따라 QP 값이 할당될 수도 있는 최소 CU 사이즈를 식별할 수도 있다. 다시 말하면, 양자화 유닛 (154) 은 MinQPCUDepth 이상인 (예컨대, 쿼드트리 구조 상에서 상대적으로 높은) 위치에 있는 CU들에 대해서만 QP 값을 설정할 수도 있다. 이 예에서, MinQPCodingUnitSize는 MinQPCUDepth 및 MaxCodingUnitSize에 기초하여 도출될 수 있다. 최소 QP 깊이는, 예를 들어, PPS 또는 SPS와 같은 파라미터 세트에서 시그널링될 수도 있다.

[0118] 양자화에 후속하여, 엔트로피 코딩 유닛 (156) 은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 코딩 유닛 (156) 은 콘텐츠 적응 가변 길이 코딩 (CAVLC), 콘텍스트 적응 이진 산술 코딩 (CABAC), 또는 다른 엔트로피 코딩 기법을 수행할 수도 있다. 엔트로피 코딩 유닛 (156) 에 의한 엔트로피 코딩에 후속하여, 인코딩된 비디오는 다른 디바이스로 송신되거나 또는 나중의 송신 또는 취출을 위해 보관될 수도 있다. 콘텍스트 적응 이진 산술 코딩 (CABAC) 의 경우, 콘텍스트는 이웃하는 코딩 단위들에 기초할 수도 있다.

[0119] 일부 경우들에서, 엔트로피 코딩 유닛 (156) 또는 비디오 인코더 (20) 의 다른 유닛은, 엔트로피 코딩 외에도 다른 코딩 기능들을 수행하도록 구성될 수도 있다. 예를 들어, 엔트로피 코딩 유닛 (156) 은 코딩 단위들 및 파티션들에 대한 CBP 값들을 결정하도록 구성될 수도 있다. 또한, 일부 경우들에서, 엔트로피 코딩 유닛 (156) 은 코딩 단위들 또는 그것의 파티션에서의 계수들의 런 길이 코딩을 수행할 수도 있다. 특히, 엔트로피 코딩 유닛 (156) 은 지그재그 스캔 또는 다른 스캔 패턴을 적용하여 코딩 단위 또는 파티션에서의 변환 계수들을 스캔하고 추가 압축을 위해 0들의 런 (run) 들을 인코딩할 수도 있다. 엔트로피 코딩 유닛 (156) 은 또한 인코딩된 비디오 비트스트림에서의 송신을 위해, 적절한 신택스 엘리먼트들을 갖는 헤더 정보를 구성할 수도 있다.

[0120] 엔트로피 코딩 유닛 (156) 이 슬라이스들에 대한 헤더 정보를 구성하는 예들에서, 본 개시물의 양태들에 따르면, 엔트로피 코딩 유닛 (156) 은 퍼베이시브 (pervasive) 슬라이스 파라미터들의 세트를 결정할 수도 있다. 퍼베이시브 슬라이스 파라미터들은, 예를 들어, 둘 이상의 슬라이스들에 공통인 신택스 엘리먼트들을 포함할 수도 있다. 위에서 지적했듯이, 신택스 엘리먼트들은 슬라이스들의 디코딩 시에 디코더를 지원할 수도 있다. 일부 예들에서 퍼베이시브 슬라이스 파라미터들은 본원에서는 "프레임 파라미터 세트" (FPS) 라고 지칭될 수도 있다. 본 개시물의 양태들에 따르면, FPS는 다수의 슬라이스들에 적용될 수도 있다. FPS는 화상 파라미터 세트 (PPS) 를 지칭할 수도 있고 슬라이스 헤더는 FPS를 지칭할 수도 있다.

[0121] 일반적으로, FPS는 전형적인 슬라이스 헤더의 정보의 대부분을 포함할 수도 있다. FPS는, 그러나, 각각의 슬라이스에 대해 반복될 필요는 없다. 일부 예들에 따르면, 엔트로피 코딩 유닛 (156) 은 FPS를 참조하는 헤더 정보를 생성할 수도 있다. 그 헤더 정보는, 예를 들어, FPS를 식별하는 프레임 파라미터 세트 식별자 (ID) 를 포함할 수도 있다. 일부 경우들에서, 엔트로피 코딩 유닛 (156) 은, 복수의 FPS들의 각각이 상이한 프레임 파라미터 세트 식별자와 연관되는, 복수의 FPS들을 정의할 수 있다. 엔트로피 코딩 유닛 (156) 은 그 다음에 복수의 FPS들 중 적절한 하나를 식별하는 슬라이스 헤더 정보를 생성할 수도 있다.

[0122] 일부 경우들에서, 엔트로피 코딩 유닛 (156) 은, 식별된 FPS가 동일한 프레임의 이전에 디코딩된 슬라이스에 연관된 FPS와는 상이하다면 FPS만을 식별할 수도 있다. 엔트로피 코딩 유닛 (156) 은, 이들 경우들에서, FPS 식별자가 설정되었는지의 여부를 식별하는, 각각의 슬라이스 헤더에서의 플래그를 정의할 수도 있다. 이러한 플래그가 설정되지 않으면 (예컨대, 플래그가 '0'의 값을 가지면), 프레임의 이전에 디코딩된 슬라이스로부터의 FPS 식별자는 현재 슬라이스에 대해 재사용될 수도 있다. 이런 식으로 FPS 식별자 플래그를 사용하면, 특히 다수의 FPS들이 정의되는 경우에 슬라이스 헤더에 의해 소비되는 비트들의 양을 추가로 감소시킬 수도 있

다.

[0123] 일 예에서, 엔트로피 코딩 유닛 (156) 은 아래에 도시된 바와 같이, 표 4에 따라 FPS를 생성할 수도 있다:

[0124] 표 4 - fra_parameter_set_header( )

fra_parameter_set_header( ) {	C	Descriptor
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
fra_parameter_set_id	2	ue(v)
frame_num	2	u(v)
if(IdrPicFlag)		
idr_pic_id	2	ue(v)
pic_order_cnt_lsb	2	u(v)
if(slice_type == P || slice_type == B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type == B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
ref_pic_list_modification( )		
if(nal_ref_idc != 0)		
dec_ref_pic_marking( )	2	
if(entropy_coding_mode_flag) {		
pipe_multi_codeword_flag	2	u(1)
if(!pipe_multi_codeword_flag)		
pipe_max_delay_shift_6	2	ue(v)
else		
balanced_cpus	2	u(8)
if(slice_type != I)		
cabac_init_idc	2	ue(v)
}		

[0125]

slice_qp_delta	2	se(v)
alf_param()		
if(slice_type == P || slice_type == B) {		
mc_interpolation_idc	2	ue(v)
mv_competition_flag	2	u(1)
if (mv_competition_flag) {		
mv_competition_temporal_flag	2	u(1)
}		
}		
if (slice_type == B && mv_competition_flag)		
collocated_from_l0_flag	2	u(1)
sifo_param()		
edge_based_prediction_flag	2	u(1)
if(edge_prediction_ipd_flag = = 1)		
threshold_edge	2	u(8)
}		

[0126]

[0127] 위의 표 4의 예에 포함된 신택스 엘리먼트들에 연관된 의미론 (semantics) 은 신흥 HEVC 표준과 동일하지만, 그러나, 그 의미론은 이 FPS 헤더를 참조하는 모든 슬라이스들에 적용가능하다. 다시 말하면, 예를 들어, fra_parameter_set_id는 프레임 파라미터 세트 헤더의 식별자를 표시한다. 따라서, 동일한 헤더 정보를 공유하는 하나 이상의 슬라이스들은 FPS 식별자를 참조할 수도 있다. 2 개의 FPS 헤더들은 그 헤더들이 동일한 fra_parameter_set_id, frame_num, 및 화상 순서 카운트 (picture order count; POC) 를 가진다면 동일하다.

[0128] 일부 예들에 따르면, FPS 헤더는 화상 파라미터 세트 (PPS) 원시 바이트 시퀀스 페이로드 (raw byte sequence payload; RBSP) 에 포함될 수도 있다. 일 예에서, FPS 헤더는 아래에 도시된 표 5에 따라 PPS에 포함될 수도 있다:

[0129] 표 5 - pic_parameter_set_rbsp( )

pic_parameter_set_rbsp( ) {	C	Descriptor
pic_parameter_set_id	1	ue(v)
…		
num_fps_headers	1	ue(v)
for (i =0; i < num_fps_headers; i++)		
fra_parameter_set_header( )		
rbsp_trailing_bits( )	1	
}		

[0130]

[0131] 일부 예들에 따르면, FPS 헤더는 프레임의 하나 이상의 슬라이스들에 포함될 수도 있다. 일 예에서, FPS 헤더는 아래에 도시된 표 6에 따라 프레임의 하나 이상의 슬라이스들에 포함될 수도 있다:

[0132] 표 6 - slice_header( )

slice_header( ) {	C	Descriptor
first_lctb_in_slice	2	ue(v)
fps_present_flag	2	u(1)
if (fps_present_flag)		
fra_parameter_set_header( )		
else		
fra_parameter_set_id	2	ue(v)
end_picture_flag	2	u(1)
…		

[0133]

[0134] 표 6의 예에서, fps_present_flag는 현재 슬라이스에 대한 슬라이스 헤더가 FPS 헤더를 포함하는지의 여부를 표시할 수도 있다. 덧붙여서, fra_parameter_set_id는 현재 슬라이스가 참조하는 FPS 헤더의 식별자를 특정할 수도 있다. 덧붙여서, 표 6에 도시된 예에 따르면, end_picture_flag는 현재 슬라이스가 현재 화상의 마지막 슬라이스인지의 여부를 표시한다.

[0135] 본 개시물의 특정 양태들 (예컨대, 이를테면 헤더 신택스 및/또는 파라미터 세트들을 생성하는 것) 은 엔트로피 코딩 유닛 (156) 에 관해 서술되었지만, 그런 서술은 설명만을 위해 제공되었다는 것이 이해되어야 한다. 다시 말하면, 다른 예들에서, 다양한 다른 코딩 모듈들이 헤더 데이터 및/또는 파라미터 세트들을 생성하기 위해 이용될 수도 있다. 예를 들어, 헤더 데이터 및/또는 파라미터 세트들은 고정 길이 코딩 모듈 (예컨대, 유유인코딩 (uuencoding; UUE) 또는 다른 코딩 방법) 에 의해 생성될 수도 있다.

[0136] 도 4를 여전히 참조하면, 역 양자화 유닛 (158) 및 역 변환 유닛 (160) 은 역 양자화 및 역 변환을 각각 적용하여, 화소 도메인에서의 잔차 블록을 예컨대, 나중에 참조 블록으로서 사용하기 위해 재구성한다. 움직임 보상 유닛 (144) 은 잔차 블록을 참조 프레임 저장부 (164) 의 프레임들 중 하나의 프레임의 예측 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 움직임 보상 유닛 (144) 은 또한 하나 이상의 보간 필터들을 재구성된 잔차 블록에 적용하여 움직임 추정에서 사용하기 위한 부-정수 화소 값들을 계산할 수도 있다. 합산기 (162) 는 재구성된 잔차 블록을 움직임 보상 유닛 (144) 에 의해 생성된 움직임 보상 예측 블록에 가산하여, 참조 프레임 저장부 (164) 에 저장하기 위한 재구성된 비디오 블록을 생성한다. 재구성된 비디오 블록은 움직임 추정 유닛 (142) 및 움직임 보상 유닛 (144) 에 의해 후속 비디오 프레임에서의 블록을 인터-코딩하기 위한 참조 블록으로서 사용될 수도 있다.

[0137] 본 개시물의 기법들은 또한 시퀀스가 이용할 수 있는 가장 미세한 슬라이스 세분도를 제어하는 프로파일 및/또는 하나 이상의 레벨들을 정의하는 것에 관련된다. 예를 들어, 대부분의 비디오 코딩 표준들에서와 같이, H.264/AVC는 에러 없는 비트스트림들에 대한 신택스, 시맨틱스, 및 디코딩 프로세스를 정의하며, 이들 중의 어느 것이라도 특정 프로파일 또는 레벨을 준수한다. H.264/AVC는 인코더를 특정하지 않지만, 인코더에는 생성된 비트스트림들이 디코더에 대한 표준 준수를 보장하는 임무가 주어진다. 비디오 코딩 표준의 측면에서, "프로파일"은 알고리즘들, 특징들 (features), 또는 도구들 및 그것들에 적용되는 제약들의 서브세트에 해당한다. H.264 표준에 의해 정의된 바와 같이, 예를 들어, "프로파일"은 H.264 표준에 의해 특정되는 전체 비트스트림 신택스의 서브세트이다. "레벨"은 예를 들어, 디코더 메모리 및 컴퓨테이션과 같은 디코더 자원 소비의 한계들에 대응하며, 이 한계들은 화상들의 해상도, 비트 레이트, 및 매크로블록 (MB) 프로세싱 레이트에 관련된다. 프로파일은 profile_idc (프로파일 표시자) 값으로 시그널링될 수 있는 반면, 레벨은 level_idc (레벨 표시자) 값으로 시그널링될 수도 있다.

[0138] H.264 표준은, 예를 들어, 주어진 프로파일의 신택스에 의해 부과되는 경계들 내에서, 디코딩된 화상들의 특정된 사이즈와 같이 비트스트림 내의 신택스 엘리먼트들에 의해 취해진 값들에 의존하여 인코더들 및 디코더들의 성능에서의 큰 변화를 요구하는 것이 여전히 가능하다는 것을 인정한다. H.264 표준은 많은 애플리케이션들에서, 특정 프로파일 내에서 신택스의 모든 가정적 사용들을 처리하는 것이 가능한 디코더를 구현하는 것이 실용적이지도 않고 경제적이지도 않다는 것을 추가로 인정한다. 따라서, H.264 표준은 비트스트림에서 신택스 엘리먼트들의 값들에 부과되는 특정된 제약들의 세트로서 "레벨"을 정의한다. 이들 제약들은 값들에 대한 간단한 제한들일 수도 있다. 다르게는, 이들 제약들은 값들의 산술적 조합들 (예컨대, 화상 폭 곱하기 화상 높이 곱하기 초당 디코딩되는 화상들의 수) 에 대한 제약들의 형태를 취할 수도 있다. H.264 표준은 개개의

구현예들이 각각의 지원된 프로파일들에 대해 상이한 레벨을 지원할 수도 있다는 것을 추가로 규정한다.

[0139] 프로파일을 준수하는 디코더, 이를테면 비디오 디코더 (30) 는 보통은 프로파일에서 정의되는 모든 특징들을 지지한다. 예를 들어, 코딩 특징으로서, B-화상 코딩은 H.264/AVC의 베이스라인 프로파일에서 지원되지 않지만 H.264/AVC의 다른 프로파일들에서 지원된다. 레벨을 준수하는 디코더는 레벨에서 정의된 한계들을 넘어서게 자원들을 요구하지 않는 임의의 비트스트림을 디코딩하는 것이 가능해야 한다. 프로파일들 및 레벨들의 정의들은 해석능력 (interpretability) 에 도움이 될 수 있다. 예를 들어, 비디오 송신 동안, 한 쌍의 프로파일 및 레벨 정의들은 전체 송신 세션에 대해 협상되고 합의될 수도 있다. 더 구체적으로는, H.264/AVC에서, 레벨은 예를 들어, 처리될 필요가 있는 매크로블록들의 수, 디코딩된 화상 버퍼 (decoded picture buffer; DPB) 사이즈, 코딩된 화상 버퍼 (coded picture buffer; CPB) 사이즈, 수직 움직임 벡터 범위, 2 개의 연속적인 MB들 당 움직임 벡터들의 최대 수, 및 B-블록이 8x8 화소들 미만의 서브-매크로블록 파티션들을 가질 수 있는지의 여부에 관한 한계들을 정의할 수도 있다. 이런 방식으로, 디코더는 디코더가 비트스트림을 적절히 디코딩하는 것이 가능한지의 여부를 결정할 수도 있다.

[0140] 본 개시물의 양태들은 슬라이스 세분도가 수정될 수도 있는 정도를 제어하기 위한 프로파일을 정의하는 것에 관련된다. 다시 말하면, 비디오 인코더 (20) 는 특정 CU 깊이보다 작은 세분도에서 비디오 데이터의 프레임을 슬라이스들로 분할하는 능력을 디스에이블시키기 위해 프로파일을 활용할 수도 있다. 일부 예들에서, 프로파일은 LCU 깊이보다 낮은 CU 깊이까지 슬라이스 세분도를 지원하지 않을 수도 있다. 그런 예들에서, 코딩된 비디오 시퀀스에서의 슬라이스들은 LCU 정렬될 수도 있다(예컨대, 각각의 슬라이스가 하나 이상의 완전히 형성된 LCU들을 포함할 수도 있다).

[0141] 덧붙여서, 위에서 지적했듯이, 슬라이스 세분도는 시퀀스 레벨에서, 예컨대, 시퀀스 파라미터 세트로 시그널링될 수도 있다. 그런 예들에서, 화상들에 대해 시그널링된 (예컨대, 화상 파라미터 세트로 시그널링된) 슬라이스 세분도는, 일반적으로 시퀀스 파라미터 세트에서 표시된 슬라이스 세분도 이상이다. 예를 들어, 슬라이스 세분도가 8x8이면, 화상 파라미터 세트들의 각각이 상이한 슬라이스 세분도들 (예컨대, 8x8, 16x16 및 32x32) 을 갖는 3 개의 화상 파라미터 세트들은 비트스트림으로 전달될 수 있다. 이 예에서, 특정 시퀀스에서의 슬라이스들은 화상 파라미터 세트들 중 임의의 것을 참조할 수도 있고, 이에 따라 세분도는 8x8, 16x16 또는 32x32일 수도 있다 (예컨대, 4x4 이하가 아닐 수도 있다).

[0142] 본 개시물의 양태들은 또한 하나 이상의 레벨들을 정의하는 것에 관련된다. 예를 들어, 하나 이상의 레벨들은 그 레벨을 준수하는 디코더 구현예가 특정 슬라이스 세분도 레벨을 지원한다는 것을 표시할 것이다. 다시 말하면, 특정 레벨은 32x32의 CU 사이즈에 대응하는 슬라이스 세분도를 가질 수도 있는 반면, 상위 레벨은 16x16의 CU 사이즈에 대응하는 슬라이스 세분도를 가질 수도 있고, 다른 높은 레벨은 상대적으로 작은 슬라이스 세분도 (예컨대, 8x8 화소의 세분도) 를 허용할 수도 있다.

[0143] 표 7에 도시된 바와 같이, 디코더의 상이한 레벨들은 슬라이스 세분도가 될 수 있는 CU 사이즈의 확장 정도에 대해 다른 제약을 가질 수도 있다.

[0144] 표 7 - 프로파일들 및 레벨들

[0145]

레벨 수	최대 매크로블록 프로세싱 레이트 MaxMBPS (MB/s)	최소 압축 비율 MinCR	2 개의 연속적인 MB들 당 움직임 벡터들의 수 MaxMvsPer2Mb	최소 슬라이스 세분도
3.2	216 000	4	16	64x64
4	245 760	4	16	32x32
4.1	245 760	2	16	16x16
4.2	491 520	2	16	8x8
5	589 824	2	16	
5.1	983 040	2	16	

[0146] 도 4의 예에서, 본 개시물의 특정 양태들, 예컨대, 비디오 데이터의 프레임을 LCU보다 작은 세분도에서 슬라이스들로 분할하는 것에 관련된 양태들은, 비디오 인코더 (20) 의 특정 유닛들에 관해 서술되었다. 그러나, 도 4의 예에서 제공된 기능성 유닛들은 설명을 목적으로 제공되었다는 것이 이해되어야 한다. 다시 말하면, 비디오 인코더 (20) 의 특정 유닛들은 설명의 목적을 위해 따로따로 도시되고 설명될 수도 있지만, 예를 들어, 집적회로 또는 다른 프로세싱 유닛 내에서와 같이, 고도로 통합될 수도 있다. 따라서, 비디오 인코더 (20) 의 하나의 유닛에 주어진 기능들은 비디오 인코더 (20) 의 하나 이상의 다른 유닛들에 의해 수행될 수도 있다.

[0147] 이런 방식으로, 비디오 인코더 (20) 는 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 인코딩할 수도 있는 비디오 인코더의 일 예이다. 일 예에 따르면, 비디오 인코더 (20) 는 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할될 세분도를 결정할 수도 있다. 비디오 인코더 (20) 는 LCU의 제 1 섹션 및 LCU의 제 2 섹션을 생성하기 위해, 그리고 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션을 포함하지 않는 프레임의 독립적으로 디코딩가능한 부분을 생성하기 위해 결정된 세분도를 이용하여 LCU를 분할할 수도 있다. 비디오 인코더 (20) 는 또한 결정된 세분도의 표시 및 프레임의 독립적으로 디코딩가능한 부분을 포함하는 비트스트림을 생성할 수도 있다.

[0148] 도 5는 본 개시물에서 설명되는 독립적으로 디코딩가능한 부분들로 분할된 비디오 데이터의 프레임을 디코딩하기 위한 기법들 중 임의의 것 또는 모두를 구현할 수도 있는 비디오 디코더 (30) 의 일 예를 도시하는 블록도이다. 다시 말하면, 예를 들어, 비디오 디코더 (30) 는 임의의 신택스, 파라미터 세트들, 헤더 데이터, 또는 독립적으로 디코딩가능한 부분들로 분할된 비디오 데이터의 프레임을 디코딩하는 것에 연관된 비디오 인코더 (20) 에 관해 설명된 다른 데이터를 디코딩하도록 구성될 수도 있다.

[0149] 도 5의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (170), 움직임 보상 유닛 (172), 인트라 예측 유닛 (174), 역 양자화 유닛 (176), 역 변환 유닛 (178), 참조 프레임 저장부 (182) 및 합산기 (180) 를 구비한다. 위에서 도 4에 관해 언급된 바와 같이, 비디오 디코더 (30) 에 관해 설명된 유닛들은 고도로 통합될 수도 있지만, 설명의 목적을 위해 따로따로 설명될 수도 있다는 것이 이해되어야 한다.

[0150] 비디오 디코더 (30) 에서 수신된 비디오 시퀀스는 인코딩된 이미지 프레임들의 세트, 프레임 슬라이스들의 세트, 공통 코딩된 화상들의 그룹 (GOP들), 또는 인코딩된 LCU들 및 이러한 LCU들을 디코딩하는 방법에 관한 명령들을 제공하는 신택스 정보를 포함하는 매우 다양한 단위들의 비디오 정보를 포함할 수도 있다. 비디오 디코더 (30) 는, 일부 예들에서, 비디오 인코더 (20) (도 4) 에 관해 설명된 인코딩 패스 (pass) 에 일반적으로 역인 디코딩 패스를 수행할 수도 있다. 예를 들어, 엔트로피 디코딩 유닛 (170) 은 도 4의 엔트로피 인코딩 유닛 (156) 에 의해 수행된 인코딩의 역 디코딩 기능을 수행할 수도 있다. 특히, 엔트로피 디코딩 유닛 (170) 은 CAVLC 또는 CABAC 디코딩, 또는 비디오 인코더 (20) 에 의해 이용된 임의의 다른 유형의 엔트로피 디코딩을 수행할 수도 있다.

[0151] 덧붙여서, 본 개시물의 양태들에 따르면, 엔트로피 디코딩 유닛 (170), 또는 비디오 디코더 (30) 의 다른 모듈, 이를테면 파싱 모듈은, 인코딩된 비디오 시퀀스의 프레임(들)을 인코딩하는데 이용된 LCU들의 사이즈들을 결정하기 위한 (예컨대, 수신된 쿼드트리에 의해 제공된 바와 같은) 신택스 정보, 인코딩된 비디오 시퀀스의 프레임의 각각의 CU가 분할되는 방법 (및 비슷하게, 서브 CU들이 분할되는 방법) 을 서술하는 분할 정보, 각각의 분할물이 인코딩되는 방법을 표시하는 모드들 (예컨대, 인트라- 또는 인터-예측이고, 인트라-예측의 경우, 인트라-예측 인코딩 모드), 각각의 인터-인코딩된 PU에 대한 하나 이상의 참조 프레임들 (및/또는 그 참조 프레임들에 대한 식별자들을 포함하는 참조 목록들), 및 인코딩된 비디오 시퀀스를 디코딩하기 위한 다른 정보를 이용할 수도 있다.

[0152] 비디오 데이터의 프레임이 LCU보다 작은 세분도에서 슬라이스들로 분할된 예들에서, 본 개시물의 기법들에 따르면, 비디오 디코더 (30) 는 이러한 세분도를 식별하도록 구성될 수도 있다. 다시 말하면, 예를 들어, 비디오 디코더 (30) 는 비디오 데이터의 프레임이 수신된 또는 시그널링된 세분도 값에 따라 분할된 세분도를 결정할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 에 관해 위에서 설명된 바와 같이, 세분도는 슬라이스 분할이 일어날 수도 있는 CU 깊이에 따라 식별될 수도 있다. CU 깊이 값은 파라미터 세트의 수신된 신택스, 이를테면 화상 파라미터 세트 (PPS) 에 포함될 수도 있다. 예를 들어, 비디오 데이터의 프레임이 슬라이스들로 분할될 수도 있는 세분도의 표시는, 위에서 설명된 바와 같이, 표 1에 따라 표시될 수도 있다.

[0153] 덧붙여서, 비디오 디코더 (30) 는 슬라이스가 시작하는 어드레스 (예컨대, "슬라이스 어드레스") 를 결정할 수도 있다. 슬라이스 어드레스는 슬라이스가 프레임 내에서 시작하는 상대 포지션을 표시할 수도 있다.

슬라이스 어드레스는 슬라이스 세분도 레벨로 제공될 수도 있다. 일부 예들에서, 슬라이스 어드레스는 슬라이스 헤더에 제공될 수도 있다. 특정 예에서, slice_address 신택스 엘리먼트는 슬라이스가 시작하는 어드레스를 슬라이스 세분도 해상도로 특정할 수도 있다. 이 예에서, slice_address는 비트스트림에서의 (Ceil(Log2(NumLCUsInPicture)) + SliceGranularity) 비트들에 의해 표현될 수도 있으며, 여기서 NumLCUsInPicture는 화상 (또는 프레임) 에서의 LCU들의 수이다. 변수 LCUAddress는 (slice_address >> SliceGranularity) 로 설정될 수도 있고 래스터 스캔 순서로 슬라이스 어드레스의 LCU 부분을 표현할 수도 있다. 변수 GranularityAddress는 (slice_address - (LCUAddress << SliceGranularity)) 로 설정될 수도 있고 z-스캔 순서로 표현된 슬라이스 어드레스의 서브 LCU 부분을 표현할 수도 있다. 그러면 변수 SliceAddress는 (LCUAddress << (log2_diff_max_min_coding_block_size << 1)) + (GranularityAddress << ((log2_diff_max_min_coding_block_size << 1) - SliceGranularity) 로 설정될 수도 있고 슬라이스 디코딩은 슬라이스 시작 좌표에서 가능한 최대 코딩 단위로 시작할 수도 있다.

[0154] 덧붙여서, 슬라이스 분할이 일어난 로케이션을 식별하기 위해, 비디오 디코더 (30) 는 슬라이스의 상대적 말단을 식별하는 하나 이상의 신택스 엘리먼트들을 수신하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 디코딩 중인 CU가 슬라이스의 최종 CU (예컨대, 분할 전의 최종 CU) 인지의 여부를 표시하는 프레임의 각각의 CU에 포함된 1 비트의 슬라이스 말단 플래그를 수신하도록 구성될 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 프레임을 슬라이스들로 분할하는데 이용된 세분도 이상인 CU들에 대해서만 슬라이스 말단 표시 (예컨대, 슬라이스 말단 플래그) 를 수신할 수도 있다.

[0155] 덧붙여서, 비디오 디코더 (30) 는 상이한 슬라이스들로 분할된 LCU에 대한 별도의 계층적 쿼드트리 정보를 수신하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 슬라이스들 사이에서 분할된 LCU의 상이한 섹션들에 연관된 별개인 분할 플래그들을 수신할 수도 있다.

[0156] 일부 예들에서, LCU에 대한 쿼드트리 정보의 부분만을 포함하는 LCU의 현재 섹션을 적절히 디코딩하기 위하여, 비디오 디코더 (30) 는 LCU의 이전의 섹션에 연관된 쿼드트리 정보를 재구성할 수도 있다. 예를 들어, 위에서 도 3a 및 도 3b에 관해 설명된 바와 같이, 비디오 디코더 (30) 는 수신된 슬라이스의 제 1 서브 CU의 인덱스 값을 식별할 수도 있다. 그 다음, 비디오 디코더 (30) 는 수신된 서브 CU가 속하는 사분역을 식별하기 위해 인덱스 값을 사용할 수도 있다. 덧붙여서, 비디오 디코더 (30) 는 (예컨대, 위에서 설명된 바와 같이, 깊이-우선 쿼드트리 탐색 알고리즘 및 수신된 분할 플래그들을 이용하여) LCU의 수신된 섹션의 쿼드트리의 노드들의 전부를 추론할 수도 있다.

[0157] 비디오 인코더 (20) (도 4) 에 관해 위에서 지적했듯이, 본 개시물의 양태들은 또한 비디오 데이터의 프레임이 슬라이스들로 분할될 수도 있는 세분도를 제어하기 위한 하나 이상의 프로파일들 및/또는 레벨들을 정의하는 것에 관련된다. 따라서, 일부 예들에서, 비디오 디코더 (30) 는 도 4에 관해 설명된 그런 프로파일들 및/또는 레벨들을 활용하도록 구성될 수도 있다. 더구나, 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 정의된 임의의 프레임 파라미터 세트들 (FPSs) 을 수신하고 활용하도록 구성될 수도 있다.

[0158] 본 개시물의 특정 양태들이 일반적으로 비디오 디코더 (30) 에 관해 설명되었지만, 그런 양태들은 비디오 디코더 (30) 의 하나 이상의 유닛들, 예컨대 엔트로피 디코딩 유닛 (170), 파싱 모듈, 또는 비디오 디코더 (30) 의 하나 이상의 다른 유닛들에 의해 수행될 수도 있다는 것이 이해되어야 한다.

[0159] 움직임 보상 유닛 (172) 은 엔트로피 디코딩 유닛 (170) 으로부터 수신된 움직임 벡터들에 기초하여 예측 데이터를 생성할 수도 있다. 예를 들어, 움직임 보상 유닛 (172) 은, 어쩌면 보간 필터들에 기초한 보간을 수행하여, 움직임 보상된 블록들을 생성한다. 부-화소 정밀도를 갖는 움직임 추정을 위해 사용될 보간 필터들에 대한 식별자들은 신택스 엘리먼트들에 포함될 수도 있다. 움직임 보상 유닛 (172) 은 비디오 블록의 인코딩 동안에 비디오 인코더 (20) 에 의해 사용된 것과 같은 보간 필터들을 사용하여 참조 블록의 부-정수 화소들에 대한 보간된 값들을 계산할 수도 있다. 움직임 보상 유닛 (172) 은 비디오 인코더 (20) 에 의해 사용된 보간 필터들을 수신된 신택스 정보에 따라 결정하고 그 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.

[0160] 인트라 예측 유닛 (174) 은 현재 프레임의 현재 블록에 대한 예측 데이터를 현재 프레임의 이전에 디코딩된 블록들로부터의 데이터 및 시그널링된 인트라 예측 모드에 기초하여 생성할 수도 있다.

[0161] 일부 예들에서, 역 양자화 유닛 (176) 은 수신된 값들을 비디오 인코더 (20) 에 의해 사용된 스캔 미러링을 이용하여 스캐닝할 수도 있다. 이런 방식으로, 비디오 디코더 (30) 는 수신된 계수들의 1차원 어레이로부터 양자화된 변환 계수들의 2차원 매트릭스를 생성할 수도 있다. 역 양자화 유닛 (176) 은 비트스트림으로 제

공되고 엔트로피 디코딩 유닛 (170) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉, 탈 양자화한다.

[0162] 역 양자화 프로세스는, 예컨대, H.264 디코딩 표준에 의해 또는 HEVC에 의해 정의된 바와 같은, 기존의 프로세스를 포함할 수도 있다. 역 양자화 프로세스는 또한 양자화 정도 및, 마찬가지로 적용되어야 할 역 양자화의 정도를 결정하기 위해, CU에 대해 비디오 인코더 (20) 에 의해 계산되고 시그널링된 양자화 파라미터 (QP) 또는 델타 QP의 사용을 포함할 수도 있다.

[0163] LCU가 2 개의 슬라이스들 사이에서 분리되는 예들에서, 본 개시물의 양태들에 따르면, 양자화 유닛 (176) 은 분리된 LCU의 각각의 부분에 대해 별개의 QP들 (또는 델타 QP들) 을 수신할 수도 있다. 설명의 목적을 위해, LCU가 2 개의 슬라이스들 사이에서, 그 LCU의 제 1 섹션은 제 1 슬라이스에 포함되었고 그 LCU의 제 2 섹션은 제 2 슬라이스에 포함되도록 분할되었다고 가정한다. 이 예에서, 역 양자화 유닛 (176) 은 LCU의 제 1 섹션을 위한 제 1 델타 QP와, 제 1 델타 QP와는 별개이며 LCU의 제 2 섹션을 위한 제 2 델타 QP를 수신할 수도 있다. 일부 예들에서, 제 1 슬라이스와 함께 제공된 델타 QP는 제 2 슬라이스와 함께 제공된 델타 QP와는 상이할 수도 있다.

[0164] 역 변환 유닛 (178) 은 역 변환, 예컨대, 역 DCT, 역 정수 변환, 역 회전 변환, 또는 역 방향 변환을 적용한다. 합산기 (180) 는 잔차 블록들과 움직임 보상 유닛 (72) 또는 인트라 예측 유닛 (74) 에 의해 생성된 대응하는 예측 블록들을 조합하여 디코딩된 블록들을 형성한다. 원한다면, 디블로킹 필터가 또한 블록형 아티팩트들을 제거하기 위하여 디코딩된 블록들을 필터링하는데 적용될 수도 있다. 디코딩된 비디오 블록들은 그 다음에 참조 프레임 저장부 (82) 에 저장되며, 이 참조 프레임 저장부는 후속하는 움직임 보상을 위한 참조 블록들을 제공하고 또한 디스플레이 디바이스 (이를테면 도 1의 디스플레이 디바이스 (32)) 상의 프레젠테이션을 위한 디코딩된 비디오를 생성한다.

[0165] 도 5의 예에서, 본 개시물의 특정 양태들, 예컨대, LCU보다 작은 세분도에서 슬라이스들로 분할된 비디오 데이터의 프레임을 수신하고 디코딩하는 것에 관련된 양태들은, 비디오 디코더 (30) 의 특정 유닛들에 관해 설명되었다. 그러나, 도 5의 예에서 제공된 기능성 유닛들은 설명을 목적으로 제공되었다는 것이 이해되어야 한다. 다시 말하면, 비디오 디코더 (30) 의 특정 유닛들은 설명의 목적을 위해 따로따로 도시되고 설명될 수도 있지만, 예를 들어, 집적회로 또는 다른 프로세싱 유닛 내에서와 같이, 고도로 통합될 수도 있다. 따라서, 비디오 디코더 (30) 의 하나의 유닛에 주어진 기능들은 비디오 디코더의 하나 이상의 다른 유닛들에 의해 수행될 수도 있다.

[0166] 따라서, 도 5는 계층적으로 배열된 복수의 상대적으로 더 작은 코딩 단위들을 포함하는 하나 이상의 최대 코딩 단위들 (LCUs) 을 포함하는 복수의 블록 사이즈 코딩 단위들을 포함하는 비디오 데이터의 프레임을 디코딩할 수도 있는 비디오 디코더 (30) 의 일 예를 제공한다. 다시 말하면, 비디오 디코더 (30) 는 프레임의 독립적으로 디코딩가능한 부분들을 형성하는 경우에 계층적으로 배열된 복수의 작은 코딩 단위들이 분할된 세분도를 결정하고, 결정된 세분도를 이용하여 제 1 섹션 및 제 2 섹션으로 분할된 LCU를 식별할 수도 있다. 비디오 디코더 (30) 는 또한 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션이 없는, 프레임의 독립적으로 디코딩가능한 부분을 디코딩할 수도 있다.

[0167] 도 6은 본 개시물에 부합하는 인코딩 기법을 예시하는 흐름도이다. 설명의 목적을 위해 비디오 인코더 (20) (도 4) 의 컴포넌트들에 의해 수행되는 것으로 일반적으로 설명되었지만, 다른 비디오 인코딩 유닛들, 프로세서들, 프로세싱 유닛들, 인코더/디코더들 (CODECs) 과 같은 하드웨어 기반 코딩 유닛들 등이 또한 도 6의 방법을 수행하도록 구성될 수도 있다는 것이 이해되어야 한다.

[0168] 도 6에 도시된 예의 방법 (220) 에서, 비디오 인코더 (20) 는 처음에는, 본 개시물의 기법들에 따르면 LCU보다 작을 수도 있는 슬라이스들로 프레임을 분할할 세분도를 결정한다 (204). 위에서 설명된 바와 같이, 비디오 데이터의 프레임을 슬라이스들로 분할할 세분도를 결정하는 경우, 비디오 인코더 (20) 는, 예를 들어, 갖가지 슬라이스 구성들에 대한 레이트-왜곡을 고려하여 허용가능 비트레이트 범위 내의 비트레이트를 달성하면서도 또한 허용가능 왜곡 범위 내의 왜곡을 제공하는 세분도를 선택할 수도 있다. 허용가능 비트레이트 범위와 허용가능 왜곡 범위는 프로파일, 이를테면 비디오 코딩 표준, 이를테면 제안된 HEVC 표준에서 특정된 프로파일들에 의해 정의될 수도 있다. 부가적으로 또는 대안으로, 비디오 인코더 (20) 는 세분도를 선택하는 경우에 타깃 슬라이스 사이즈를 고려할 수도 있다. 일반적으로, 세분도를 증가시키면 슬라이스들의 사이즈에 관한 더 많은 제어가 허용될 수도 있지만, 또한 슬라이스들의 인코딩 또는 디코딩 시에 활용되는 코딩 유닛 자원들을 증가시킬 수도 있다.

[0169] 비디오 인코더 (20) 가 비디오 데이터의 프레임을 LCU보다 작은 슬라이스들로 분할하기 위한 세분도를 결정한다면, 비디오 인코더 (20) 는 슬라이스들을 만드는 프로세스에서, 결정된 세분도를 이용하여 LCU를 제 1 섹션 및 제 2 섹션으로 분할할 수도 있다 (206). 다시 말하면, 비디오 인코더 (20) 는 LCU에 포함된 슬라이스 경계를 식별할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 LCU를 제 1 섹션과 제 1 섹션과는 별개인 제 2 섹션으로 분할할 수도 있다.

[0170] LCU를 2 개의 섹션들로 분할하는 경우, 비디오 인코더 (20) 는 또한 LCU에 연관된 쿼드트리를 2 개의 대응하는 섹션들로 분리하고, 쿼드트리의 개별 섹션들을 LCU의 2 개의 섹션들에 포함시킬 수도 있다 (208). 예를 들어, 위에서 설명된 바와 같이, 비디오 인코더 (20) 는 LCU의 제 1 섹션에 연관된 분할 플래그들을 LCU의 제 2 섹션에 연관된 분할 플래그들로부터 분리할 수도 있다. LCU의 섹션들을 포함하는 슬라이스들을 인코딩하는 경우, 비디오 인코더 (20) 는 단지, LCU의 제 1 섹션에 연관된 분할 플래그들을 LCU의 제 1 섹션을 포함하는 슬라이스에 그리고 LCU의 섹션에 연관된 분할 플래그들을 LCU의 제 2 섹션을 포함하는 슬라이스에 포함시킬 수도 있다.

[0171] 덧붙여서, 슬라이스 형성 동안에 LCU를 2 개의 섹션들로 분할하는 경우, 비디오 인코더 (20) 는 각각의 섹션에 대해 별도의 양자화 파라미터 (QP) 또는 델타 QP 값들을 생성할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 LCU의 제 1 섹션에 대한 제 1 QP 또는 델타 QP 값과, LCU의 제 2 섹션에 대한 제 2 QP 또는 델타 QP 값을 생성할 수도 있다. 일부 예들에서, 제 1 섹션에 대한 QP 또는 델타 QP 값은 제 2 섹션에 대한 QP 또는 델타 QP 값과는 상이할 수도 있다.

[0172] 비디오 인코더 (20) 그 다음에 LCU의 제 1 섹션을 포함하고 LCU의 제 2 섹션이 없는, LCU를 포함하는 프레임의 독립적으로 디코딩가능한 부분, 예컨대, 슬라이스를 생성할 수도 있다 (212). 예를 들어, 비디오 인코더 (20) 는 비디오 데이터의 프레임의 하나 이상의 완전한 LCU들, 뿐만 아니라 그 프레임의 분리된 LCU의 제 1 섹션을 포함하는 슬라이스를 생성할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 분리된 LCU의 제 1 섹션에 연관된 분할 플래그들 및 델타 QP 값을 포함할 수도 있다.

[0173] 비디오 인코더 (20) 는 또한 비디오 데이터의 프레임을 슬라이스들로 분할하는데 이용된 세분도의 표시를 제공할 수도 있다 (214). 예를 들어, 비디오 인코더 (20) 는 슬라이스 분할이 일어날 수도 있는 CU 깊이 값을 이용하여 세분도의 표시를 제공할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 세분도를 상이하게 표시할 수도 있다. 예를 들어, 다른 방법으로는, 비디오 인코더 (20) 는 슬라이스 분할이 일어날 수도 있는 서브 CU들의 사이즈를 식별함으로써 세분도를 표시할 수도 있다. 부가적으로 또는 대안으로, 위에서 설명된 바와 같이, 비디오 인코더 (20) 는 다양한 다른 정보, 이를테면 슬라이스의 말단 플래그들, 프레임 파라미터들 세트들 (FPSs) 등을 슬라이스에 포함시킬 수도 있다.

[0174] 비디오 인코더 (20) 는 그 다음에 슬라이스에 연관된 비디오 데이터, 뿐만 아니라 그 슬라이스를 디코딩하기 위한 신택스 정보를 포함하는 비트스트림을 생성할 수도 있다 (216). 본 개시물의 양태들에 따르면, 생성된 비트스트림은 디코더에 실시간으로 (예컨대, 화상 회의 시에) 송신될 수도 있거나 또는 디코더에 의한 장래의 사용 (예컨대, 스트리밍, 다운로딩, 디스크 액세스, 카드 액세스, DVD, 블루레이 등) 을 위해 컴퓨터 판독가능 매체 상에 저장될 수도 있다.

[0175] 도 6에 관해 도시되고 설명된 단계들이 단지 하나의 예로서 제공된다는 것이 또한 이해되어야 한다. 다시 말하면, 도 6의 방법의 단계들은 도 6에 도시된 순서로 반드시 수행될 필요는 없고, 더 적은, 부가적인, 또는 대안적 단계들이 수행될 수도 있다. 예를 들어, 다른 예에 따르면, 비디오 인코더 (20) 는 슬라이스를 생성하기 전에 (예컨대, 세분도 (214) 의 표시와 같은) 신택스 엘리먼트들을 생성할 수도 있다.

[0176] 도 7은 본 개시물에 부합하는 디코딩 기법을 예시하는 흐름도이다. 설명의 목적을 위해 비디오 디코더 (30) (도 5) 의 컴포넌트들에 의해 수행되는 것으로 일반적으로 설명되었지만, 다른 비디오 디코딩 유닛들, 프로세서들, 프로세싱 유닛들, 인코더/디코더들 (CODECs) 과 같은 하드웨어 기반 코딩 유닛들 등이 또한 도 7의 방법을 수행하도록 구성될 수도 있다는 것이 이해되어야 한다.

[0177] 도 7에 도시된 예의 방법 (220) 에서, 비디오 디코더 (30) 는 본원에서 슬라이스라고 지칭되는, 비디오 데이터의 프레임의 독립적으로 디코딩가능한 부분을 수신한다 (222). 그 슬라이스를 수신 시, 비디오 디코더 (30) 는 그 슬라이스가 형성되었던, LCU보다 작을 수도 있는 세분도를 결정한다 (224). 예를 들어, 위에서 설명된 바와 같이, 비디오 인코더는, LCU의 제 1 섹션은 수신된 슬라이스에 포함되는 반면 LCU의 제 2 섹션은 다른 슬라이스에 포함되도록 LCU를 2 개의 섹션들로 분할하는 슬라이스를 생성할 수도 있다. 프레임이 슬라이스

들로 분할된 세분도를 결정하기 위해, 비디오 디코더 (30) 는 세분도의 표시를 수신할 수도 있다. 다시 말하면, 비디오 디코더 (30) 는 슬라이스 분할이 일어날 수도 있는 CU 깊이를 식별하는 CU 깊이 값을 수신할 수도 있다.

[0178] 비디오 데이터의 프레임이 LCU보다 작은 세분도에서 슬라이스들로 분리된 예들에서, 비디오 디코더 (30) 는 섹션들로 분할된 수신된 슬라이스의 LCU를 식별할 수도 있다 (226). 비디오 디코더 (30) 는 또한 LCU의 수신된 섹션에 대한 쿼드트리를 결정할 수도 있다 (228). 다시 말하면, 비디오 디코더 (30) 는 LCU의 수신된 섹션에 연관된 분할 플래그들을 식별할 수도 있다. 덧붙여서, 위에서 설명된 바와 같이, 비디오 디코더 (30) 는 수신된 섹션을 적절히 디코딩하기 위하여 분할된 전체 LCU에 연관된 쿼드트리를 재구성할 수도 있다. 비디오 디코더 (30) 는 또한 LCU의 수신된 섹션에 대한 QP 또는 델타 QP 값을 결정할 수도 있다 (230).

[0179] 비디오 데이터 및 연관된 신택스 정보를 이용하여, 비디오 디코더 (30) 는 그 다음에 LCU의 수신된 섹션을 포함하는 슬라이스를 디코딩할 수도 있다 (232). 도 6에 관해 위에서 설명된 바와 같이, 비디오 디코더 (30) 는, 예를 들어, 슬라이스의 말단 플래그들, 프레임 파라미터 세트들 (FPSs) 등을 포함한, 슬라이스를 디코딩하기 위한 다양한 정보를 수신하고 활용할 수도 있다.

[0180] 도 7에 관해 도시되고 설명된 단계들은 단지 하나의 예로서 제공된다는 것이 또한 이해되어야 한다. 다시 말하면, 도 7의 방법의 단계들은 도 7에 도시된 순서로 반드시 수행될 필요는 없고, 더 적은, 부가적인, 또는 대안적 단계들이 수행될 수도 있다.

[0181] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 또는 송신될 수도 있고 하드웨어 기반 처리 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다.

[0182] 이런 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적 (non-transitory) 인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드들 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0183] 비제한적인 예로, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 이를테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다.

[0184] 그러나, 컴퓨터-판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비-일시적 (non-transient), 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (Disk 및 disc) 는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들로써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

[0185] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 집적회로들 (ASICs), 필드 프로그램가능 로직 어레이들 (FPGAs), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서"는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중의 어느 것을 나타낼 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들

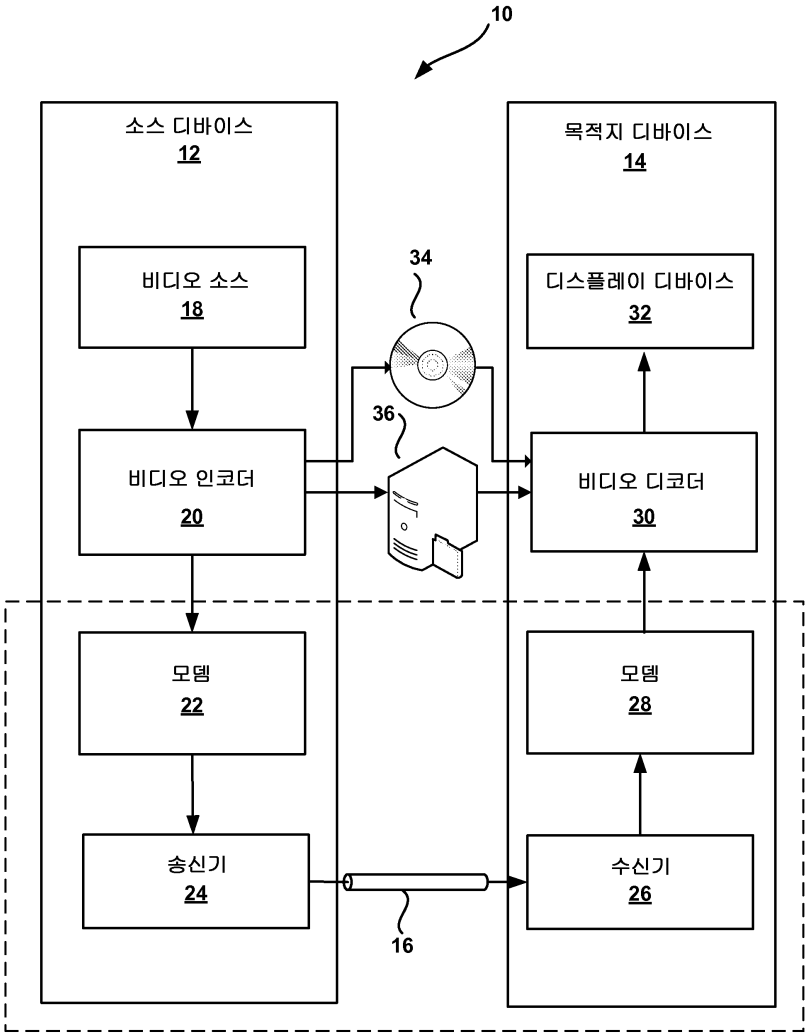
은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0186] 본 개시내용의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 한 세트의 IC들 (예컨대, 칩 셋) 을 포함하여, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 갖가지 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 위에서 설명된 바와 같이, 갖가지 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

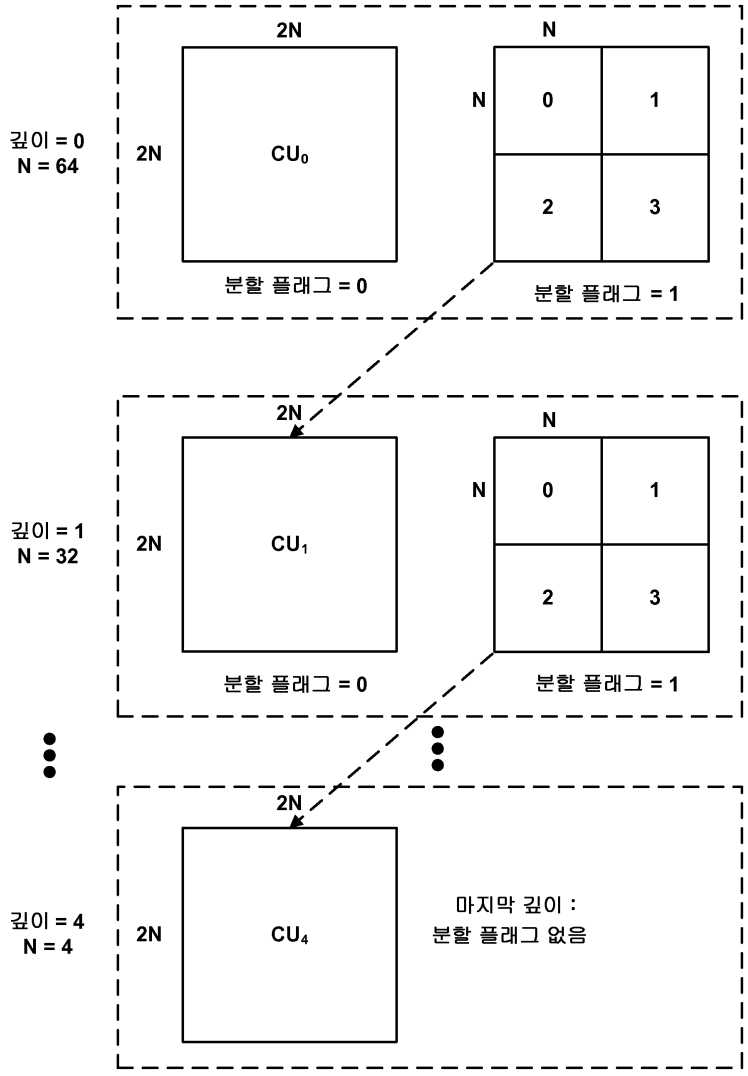
[0187] 이 개시물의 다양한 양태들이 설명되었다. 이들 및 다른 양태들은 다음의 청구항들의 범위 내에 있다.

도면

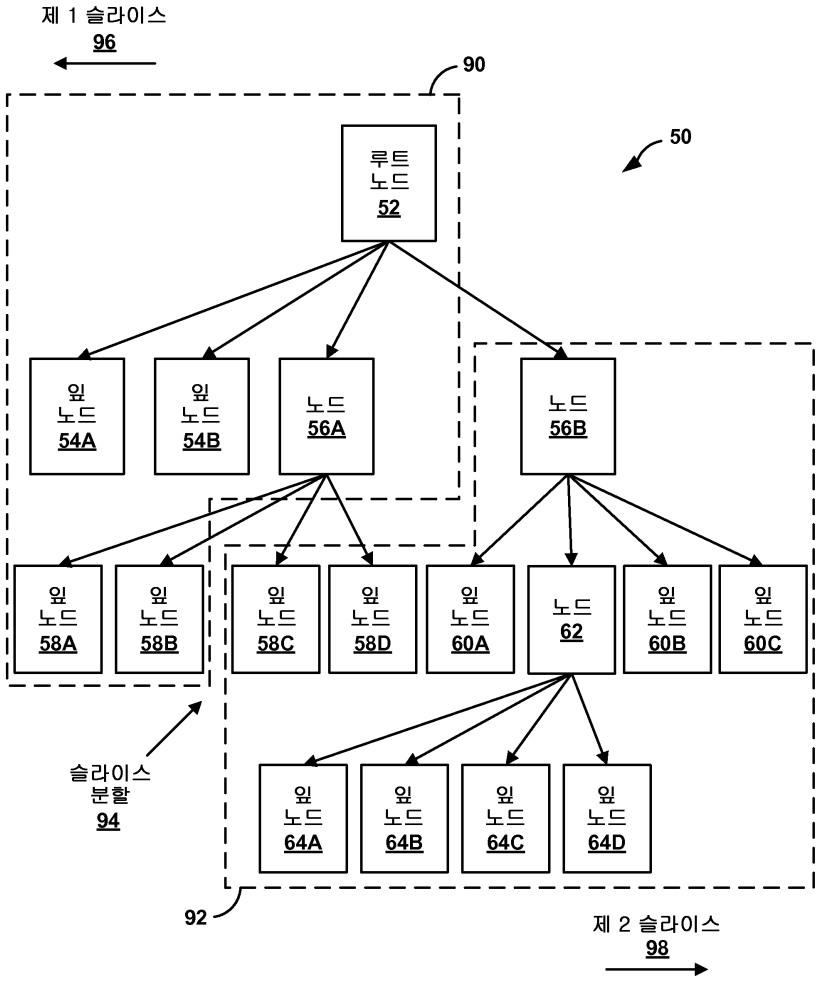
도면1



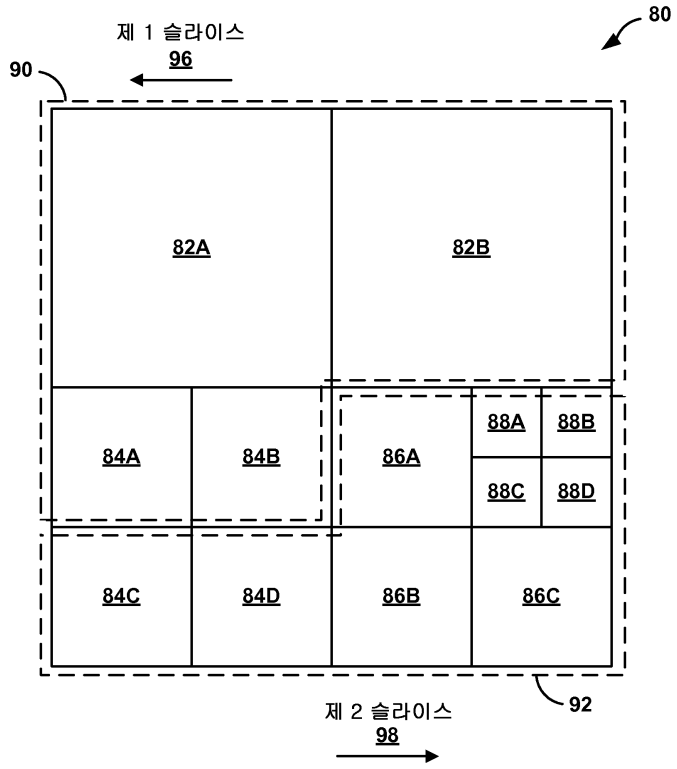
도면2


깊이 = 0
N = 64
2N
2N
CU0
N
N
0
1
2
3
분할 플래그 = 0
분할 플래그 = 1
깊이 = 1
N = 32
2N
2N
CU1
N
N
0
1
2
3
분할 플래그 = 0
분할 플래그 = 1
깊이 = 4
N = 4
2N
2N
CU4
마지막 깊이 :
분할 플래그 없음

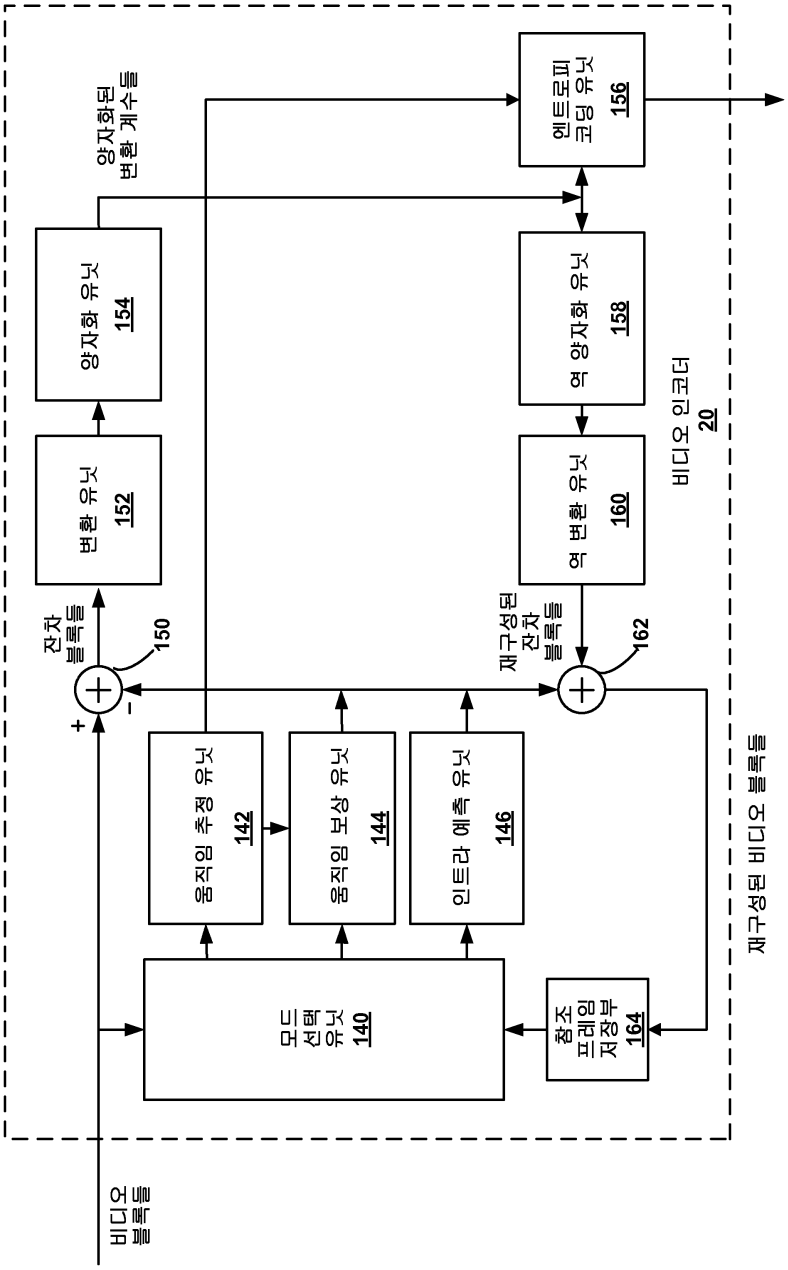
도면3a


제 1 슬라이스
96
90
50
루트
노드
52
잎
노드
54A
잎
노드
54B
노드
56A
노드
56B
잎
노드
58A
잎
노드
58B
잎
노드
58C
잎
노드
58D
잎
노드
60A
노드
62
잎
노드
60B
잎
노드
60C
슬라이스
분할
94
잎
노드
64A
잎
노드
64B
잎
노드
64C
잎
노드
64D
92
제 2 슬라이스
98

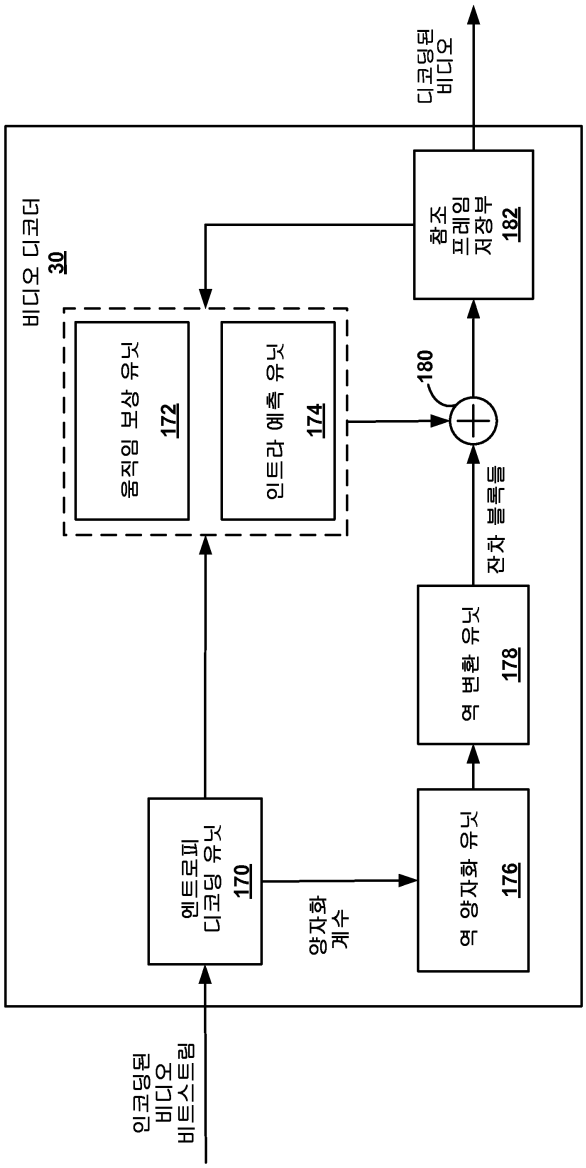
도면3b


80
제 1 슬라이스
96
90
82A
82B
84A
84B
86A
88A
88B
88C
88D
84C
84D
86B
86C
92
제 2 슬라이스
98

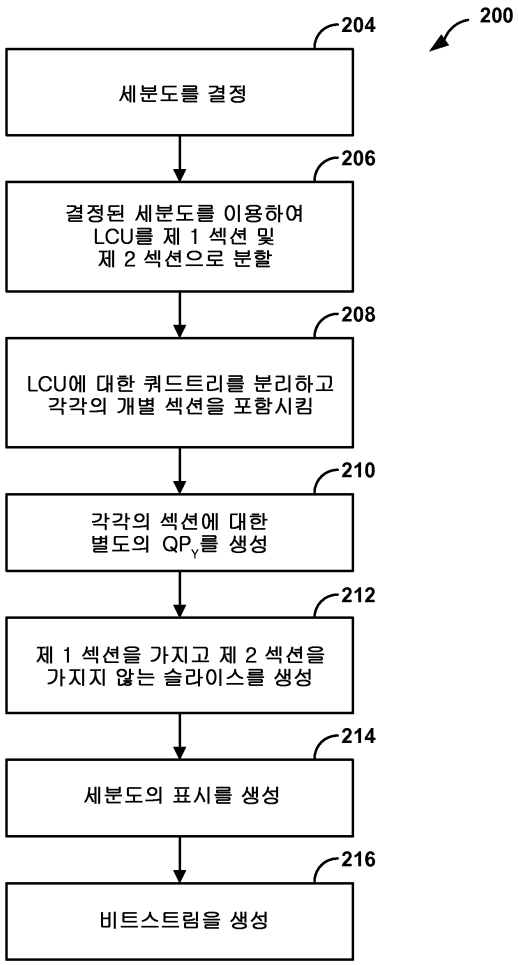
도면4


비디오 블록들
모드 선택 유닛 140
참조 프레임 저장부 164
움직임 추정 유닛 142
움직임 보상 유닛 144
인트라 예측 유닛 146
+
-
잔차 블록들
150
변환 유닛 152
양자화 유닛 154
양자화된 변환 계수들
엔트로피 코딩 유닛 156
역 양자화 유닛 158
역 변환 유닛 160
재구성된 잔차 블록들
162
비디오 인코더 20
재구성된 비디오 블록들

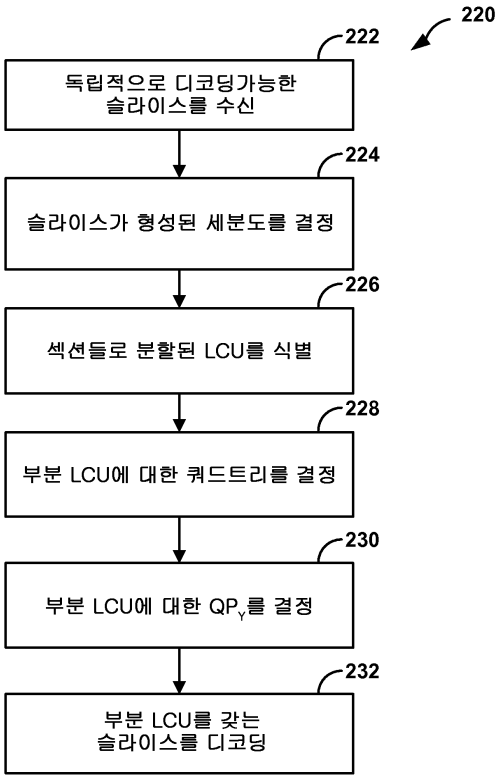
도면5


인코딩된
비디오
비트스트림
비디오 디코더
30
엔트로피
디코딩 유닛
170
양자화
계수
역 양자화 유닛
176
역 변환 유닛
178
잔차 블록들
180
움직임 보상 유닛
172
인트라 예측 유닛
174
참조
프레임
저장부
182
디코딩된
비디오

도면6


200
204
세분도를 결정
206
결정된 세분도를 이용하여
LCU를 제 1 섹션 및
제 2 섹션으로 분할
208
LCU에 대한 쿼드트리를 분리하고
각각의 개별 섹션을 포함시킴
210
각각의 섹션에 대한
별도의 QPY를 생성
212
제 1 섹션을 가지고 제 2 섹션을
가지지 않는 슬라이스를 생성
214
세분도의 표시를 생성
216
비트스트림을 생성

도면7


220
222
독립적으로 디코딩가능한 슬라이스를 수신
224
슬라이스가 형성된 세분도를 결정
226
섹션들로 분할된 LCU를 식별
228
부분 LCU에 대한 쿼드트리를 결정
230
부분 LCU에 대한 QP_Y를 결정
232
부분 LCU를 갖는 슬라이스를 디코딩