



등록특허 10-2624245



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2024년01월11일  
(11) 등록번호 10-2624245  
(24) 등록일자 2024년01월09일

- (51) 국제특허분류(Int. Cl.)  
*HO4N 19/44* (2014.01) *HO4N 19/176* (2014.01)  
*HO4N 19/186* (2014.01) *HO4N 19/463* (2014.01)  
*HO4N 19/50* (2014.01) *HO4N 19/593* (2014.01)  
*HO4N 19/70* (2014.01) *HO4N 19/93* (2014.01)
- (52) CPC특허분류  
*HO4N 19/44* (2015.01)  
*HO4N 19/105* (2015.01)
- (21) 출원번호 10-2018-7010294
- (22) 출원일자(국제) 2016년09월14일  
심사청구일자 2021년08월26일
- (85) 번역문제출일자 2018년04월11일
- (65) 공개번호 10-2018-0053702
- (43) 공개일자 2018년05월23일
- (86) 국제출원번호 PCT/US2016/051745
- (87) 국제공개번호 WO 2017/048849  
국제공개일자 2017년03월23일
- (30) 우선권주장  
62/218,497 2015년09월14일 미국(US)  
(뒷면에 계속)
- (56) 선행기술조사문현
- (73) 특허권자  
웰컴 인코포레이티드  
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자  
세레진 바딤  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775  
조쉬 라잔 래스맨  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775  
카르체비츠 마르타  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- (74) 대리인  
특허법인코리아나

(뒷면에 계속)

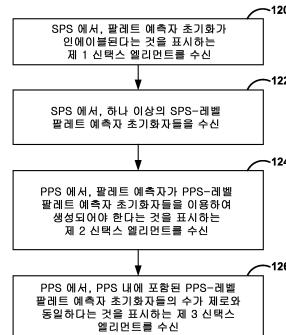
전체 청구항 수 : 총 44 항

심사관 : 이남숙

(54) 발명의 명칭 비디오 코딩을 위한 팔레트 예측자 초기화 및 병합

**(57) 요 약**

비디오 데이터를 디코딩하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리, 및 꾹쳐 파라미터 세트 (PPS)에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 신팩스 엘리먼트를 수신하고; PPS에서, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 신팩스 엘리먼트를 수신하고; 그리고 제 1 신팩스 엘리먼트 및 제 2 신팩스 엘리먼트에 기초하여 비디오 데이터의 블록을 디코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.

**대 표 도**

(52) CPC특허분류

*HO4N 19/176* (2015.01)

*HO4N 19/186* (2015.01)

*HO4N 19/463* (2015.01)

*HO4N 19/50* (2015.01)

*HO4N 19/593* (2015.01)

*HO4N 19/70* (2015.01)

*HO4N 19/93* (2015.01)

(56) 선행기술조사문헌

Jing Ye, et al., On palette predictor initialization for Screen Content Coding, JCTVC-U0084, 2015.06.10.\*

C. Gisquet, et al., Non-CE1: On palette prediction for slices, JCTVC-T0048, 2015.02.15.

Rajan Joshi, et al., High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 4, JCTVC-U1005\_r1 2015.09.05.

Vadim Seregin, et al., SPS and PPS palette predictor initialization, JCTVC-V0042, 2015.10.06.

\*는 심사관에 의하여 인용된 문헌

(30) 우선권주장

62/237,952 2015년10월06일 미국(US)

15/263,686 2016년09월13일 미국(US)

## 명세서

### 청구범위

#### 청구항 1

비디오 데이터를 디코딩하는 방법으로서,

픽처 파라미터 세트 (PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 선택스 엘리먼트를 수신하는 단계;

상기 제 1 선택스 엘리먼트가 상기 팔레트 예측자가 상기 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 것에 응답하여, 상기 PPS 에서, 상기 PPS 내에 포함된 상기 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 선택스 엘리먼트를 수신하는 단계;

상기 제 2 선택스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 응답하여, 하나 이상의 팔레트 테이블 엔트리 값들을 수신하는 단계;

상기 하나 이상의 팔레트 테이블 엔트리 값들을 포함하는 팔레트 테이블을 생성하는 단계; 및

상기 팔레트 테이블에 기초하여 비디오 데이터의 블록을 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 2

제 1 항에 있어서,

시퀀스 파라미터 세트 (SPS) 에서, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 제 3 선택스 엘리먼트를 수신하는 단계; 및

상기 SPS 에서, 하나 이상의 SPS-레벨 팔레트 예측자 초기화자들을 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 3

제 2 항에 있어서,

상기 SPS 에서, 상기 SPS 내에 포함된 SPS-레벨 팔레트 예측자 초기화자들의 수를 표시하는 제 4 선택스 엘리먼트를 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 4

제 2 항에 있어서,

상기 제 1 선택스 엘리먼트는 상기 팔레트 예측자가 SPS-레벨 팔레트 예측자 엔트리들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 5

제 2 항에 있어서,

상기 제 3 선택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 6

제 1 항에 있어서,

상기 제 1 선택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 7

삭제

## 청구항 8

제 1 항에 있어서,

상기 비디오 데이터의 블록의 샘플에 대하여, 인덱스를 수신하는 단계;

상기 샘플에 대한 상기 인덱스에 대응하는 상기 하나 이상의 팔레트 테이블 엔트리 값들 중의 하나를 이용하여 상기 비디오 데이터의 블록을 복원하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 9

제 1 항에 있어서,

상기 제 2 선택스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 기초하여, (1) 팔레트 예측자가 단색 엔트리들을 포함하는지 여부의 표시, (2) 루마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시, 또는 (3) 크로마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시 중의 하나 이상이 상기 PPS 내에 존재하지 않는 것으로 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 10

비디오 데이터를 인코딩하는 방법으로서,

인코딩된 비디오 데이터의 비트스트림에서의 픽처 파라미터 세트 (PPS) 선택스 구조에서의 포함을 위하여, 팔레트 예측자가 PPS-레벨 팔레트 예측자 엔트리들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 선택스 엘리먼트를 생성하는 단계;

상기 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되는 것에 응답하여, 상기 PPS 선택스 구조에서의 포함을 위하여, 상기 PPS 선택스 구조 내에 포함된 PPS-레벨 팔레트 예측자 엔트리들의 수가 제로와 동일하다는 것을 표시하는 제 2 선택스 엘리먼트를 생성하는 단계;

상기 PPS 선택스 구조 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것에 응답하여, 상기 PPS 선택스 구조에서의 포함을 위하여, 하나 이상의 팔레트 테이블 엔트리 값들을 생성하는 단계; 및

상기 제 1 선택스 엘리먼트, 상기 제 2 선택스 엘리먼트, 및 상기 하나 이상의 팔레트 테이블 엔트리 값들을 포함하는 인코딩된 비디오 데이터의 비트스트림을 출력하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

## 청구항 11

제 10 항에 있어서,

인코딩된 비디오 데이터의 비트스트림에서의 시퀀스 파라미터 세트 (SPS) 선택스 구조에서의 포함을 위하여, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 제 3 선택스 엘리먼트를 생성하는 단계; 및

상기 SPS 선택스 구조에서의 포함을 위하여, 하나 이상의 SPS-레벨 팔레트 예측자 엔트리들을 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

## 청구항 12

제 11 항에 있어서,

상기 제 1 선택스 엘리먼트는 상기 팔레트 예측자가 상기 SPS-레벨 팔레트 예측자 엔트리들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시하는, 비디오 데이터를 인코딩하는 방법.

## 청구항 13

제 11 항에 있어서,

상기 SPS 선택스 구조에서의 포함을 위하여, 상기 SPS 내에 포함된 SPS-레벨 팔레트 예측자 엔트리들의 수를 표시하는 제 4 선택스 엘리먼트를 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

## 청구항 14

제 11 항에 있어서,

상기 제 3 신택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 인코딩하는 방법.

### 청구항 15

제 10 항에 있어서,

상기 제 1 신택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 인코딩하는 방법.

### 청구항 16

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

꼭처 파라미터 세트 (PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 신택스 엘리먼트를 수신하고;

상기 제 1 신택스 엘리먼트가 상기 팔레트 예측자가 상기 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 것에 응답하여, 상기 PPS 에서, 상기 PPS 내에 포함된 상기 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 신택스 엘리먼트를 수신하고;

상기 제 2 신택스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 응답하여, 하나 이상의 팔레트 테이블 엔트리 값을 수신하고;

상기 하나 이상의 팔레트 테이블 엔트리 값을 포함하는 팔레트 테이블을 생성하며; 그리고

상기 팔레트 테이블에 기초하여 비디오 데이터의 블록을 디코딩하도록

구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 17

제 16 항에 있어서,

상기 하나 이상의 프로세서들은,

시퀀스 파라미터 세트 (SPS) 에서, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 제 3 신택스 엘리먼트를 수신하고; 그리고

상기 SPS 에서, 하나 이상의 SPS-레벨 팔레트 예측자 초기화자들을 수신하도록

추가로 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 18

제 17 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 SPS 에서, 상기 SPS 내에 포함된 SPS-레벨 팔레트 예측자 초기화자들의 수를 표시하는 제 4 신택스 엘리먼트를 수신하도록

추가로 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 19

제 17 항에 있어서,

상기 제 1 신택스 엘리먼트는 상기 팔레트 예측자가 SPS-레벨 팔레트 예측자 엔트리들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시하는, 비디오 데이터를 디코딩하기 위한 디바이스.

## 청구항 20

제 17 항에 있어서,

상기 제 3 신택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

## 청구항 21

제 16 항에 있어서,

상기 제 1 신택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

## 청구항 22

삭제

## 청구항 23

제 17 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 블록의 샘플에 대하여, 인덱스를 수신하고; 그리고

상기 샘플에 대한 상기 인덱스에 대응하는 상기 하나 이상의 팔레트 테이블 엔트리 값들 중의 하나를 이용하여  
상기 비디오 데이터의 블록을 복원하도록

추가로 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

## 청구항 24

제 16 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 제 2 신택스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 기초하여, (1) 팔레트 예측자가 단색 엔트리들을 포함하는지 여부의 표시, (2) 루마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시, 또는 (3) 크로마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시 중의 하나 이상이 상기 PPS 내에 존재하지 않는 것으로 결정하도록

추가로 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

## 청구항 25

비디오 데이터를 인코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

인코딩된 비디오 데이터의 비트스트림에서의 뷔처 파라미터 세트 (PPS) 신택스 구조에서의 포함을 위하여, 팔레트 예측자가 PPS-레벨 팔레트 예측자 엔트리들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 신택스 엘리먼트를 생성하고;

상기 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되는 것에 응답하여, 상기 PPS 신택스 구조에서의 포함을 위하여, 상기 PPS 신택스 구조 내에 포함된 PPS-레벨 팔레트 예측자 엔트리들의 수가 제로와 동일하다는 것을 표시하는 제 2 신택스 엘리먼트를 생성하고;

상기 PPS 신택스 구조 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것에 응답하여, 상기 PPS 신택스 구조에서의 포함을 위하여, 하나 이상의 팔레트 테이블 엔트리 값들을 생성하며; 그리고

상기 제 1 신택스 엘리먼트, 상기 제 2 신택스 엘리먼트, 및 상기 하나 이상의 팔레트 테이블 엔트리

값들을 포함하는 인코딩된 비디오 데이터의 비트스트림을 출력하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

#### 청구항 26

제 25 항에 있어서,

상기 하나 이상의 프로세서들은,

인코딩된 비디오 데이터의 비트스트림에서의 시퀀스 파라미터 세트 (SPS) 신택스 구조에서의 포함을 위하여, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 제 3 신택스 엘리먼트를 생성하고; 그리고 상기 SPS 신택스 구조에서의 포함을 위하여, 하나 이상의 SPS-레벨 팔레트 예측자 엔트리들을 생성하도록 추가로 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

#### 청구항 27

제 26 항에 있어서,

상기 제 1 신택스 엘리먼트는 상기 팔레트 예측자가 상기 SPS-레벨 팔레트 예측자 엔트리들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시하는, 비디오 데이터를 인코딩하기 위한 디바이스.

#### 청구항 28

제 26 항에 있어서,

상기 SPS 신택스 구조에서의 포함을 위하여, 상기 SPS 내에 포함된 SPS-레벨 팔레트 예측자 엔트리들의 수를 표시하는 제 4 신택스 엘리먼트를 생성하는 것을 더 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

#### 청구항 29

제 26 항에 있어서,

상기 제 3 신택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

#### 청구항 30

제 25 항에 있어서,

상기 제 1 신택스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

#### 청구항 31

비디오 데이터를 디코딩하기 위한 장치로서,

픽처 파라미터 세트 (PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 신택스 엘리먼트를 수신하는 수단;

상기 제 1 신택스 엘리먼트가 상기 팔레트 예측자가 상기 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 것에 응답하여, 상기 PPS 에서, 상기 PPS 내에 포함된 상기 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 신택스 엘리먼트를 수신하는 수단;

상기 제 2 신택스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 응답하여, 하나 이상의 팔레트 테이블 엔트리 값들을 수신하는 수단;

상기 하나 이상의 팔레트 테이블 엔트리 값들을 포함하는 팔레트 테이블을 생성하는 수단; 및

상기 제 1 신택스 엘리먼트 및 상기 제 2 신택스 엘리먼트에 기초하여 비디오 데이터의 블록을 디코딩하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

#### 청구항 32

제 31 항에 있어서,

시퀀스 파라미터 세트 (SPS) 에서, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 제 3 신팩스 엘리먼트를 수신하는 수단; 및

상기 SPS 에서, 하나 이상의 SPS-레벨 팔레트 예측자 초기화자들을 수신하는 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 33

제 32 항에 있어서,

상기 SPS 에서, 상기 SPS 내에 포함된 SPS-레벨 팔레트 예측자 초기화자들의 수를 표시하는 제 4 신팩스 엘리먼트를 수신하는 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 34

제 32 항에 있어서,

상기 제 1 신팩스 엘리먼트는 상기 팔레트 예측자가 SPS-레벨 팔레트 예측자 엔트리들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 35

제 32 항에 있어서,

상기 제 3 신팩스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 36

제 31 항에 있어서,

상기 제 1 신팩스 엘리먼트는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 37

삭제

### 청구항 38

제 31 항에 있어서,

상기 비디오 데이터의 블록의 샘플에 대하여, 인덱스를 수신하는 수단; 및

상기 샘플에 대한 상기 인덱스에 대응하는 상기 하나 이상의 팔레트 테이블 엔트리 값들 중의 하나를 이용하여 상기 비디오 데이터의 블록을 복원하는 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 39

제 31 항에 있어서,

상기 제 2 신팩스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 기초하여, (1) 팔레트 예측자가 단색 엔트리들을 포함하는지 여부의 표시, (2) 루마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시, 또는 (3) 크로마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시 중의 하나 이상이 상기 PPS 내에 존재하지 않는 것으로 결정하는 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 40

명령들을 저장하는 컴퓨터 관독가능 저장 매체로서,

상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 경우, 상기 하나 이상의 프로세서들로 하여금,

픽처 파라미터 세트 (PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 신팩스 엘리먼트를 수신하게 하고;

상기 제 1 선택스 엘리먼트가 상기 팔레트 예측자가 상기 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 것에 응답하여, 상기 PPS 에서, 상기 PPS 내에 포함된 상기 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 선택스 엘리먼트를 수신하게 하고;

상기 제 2 선택스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 응답하여, 하나 이상의 팔레트 테이블 엔트리 값들을 수신하게 하고;

상기 하나 이상의 팔레트 테이블 엔트리 값들을 포함하는 팔레트 테이블을 생성하게 하며; 그리고

상기 제 1 선택스 엘리먼트 및 상기 제 2 선택스 엘리먼트에 기초하여 비디오 데이터의 블록을 디코딩하게 하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 41

제 40 항에 있어서,

실행될 경우, 상기 하나 이상의 프로세서들로 하여금,

시퀀스 파라미터 세트 (SPS) 에서, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 제 3 선택스 엘리먼트를 수신하게 하고; 그리고

상기 SPS 에서, 하나 이상의 SPS-레벨 팔레트 예측자 초기화자들을 수신하게 하는

명령들을 추가로 저장하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 42

제 41 항에 있어서,

실행될 경우, 상기 하나 이상의 프로세서들로 하여금,

상기 SPS 에서, 상기 SPS 내에 포함된 SPS-레벨 팔레트 예측자 초기화자들의 수를 표시하는 제 4 선택스 엘리먼트를 수신하게 하는 명령들을 추가로 저장하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 43

제 41 항에 있어서,

상기 제 1 선택스 엘리먼트는 상기 팔레트 예측자가 SPS-레벨 팔레트 예측자 엔트리들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 44

제 41 항에 있어서,

상기 제 3 선택스 엘리먼트는 1-비트 플래그를 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 45

제 40 항에 있어서,

상기 제 1 선택스 엘리먼트는 1-비트 플래그를 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 46

삭제

#### 청구항 47

제 40 항에 있어서,

상기 비디오 데이터의 블록의 샘플에 대하여, 인덱스를 수신하고; 그리고

상기 샘플에 대한 상기 인덱스에 대응하는 상기 하나 이상의 팔레트 테이블 엔트리 값들 중의 하나를 이용하여 상기 비디오 데이터의 블록을 복원하는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

## 청구항 48

제 40 항에 있어서,

상기 제 2 신택스 엘리먼트가 상기 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 상기 수가 제로와 동일하다는 것을 표시하는 것에 기초하여, (1) 팔레트 예측자가 단색 엔트리들을 포함하는지 여부의 표시, (2) 루마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시, 또는 (3) 크로마 팔레트 예측자 엔트리들에 대한 비트 심도의 표시 중의 하나 이상이 상기 PPS 내에 존재하지 않는 것으로 결정하는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

## 발명의 설명

### 기술 분야

[0001] 이 개시물은 2015년 9월 14일자로 출원된 미국 특허 출원 제 62/218,497호 및 2015년 10월 6일자로 출원된 미국 특허 출원 제 62/237,952호의 이익을 주장하고, 그 각각의 전체 내용은 참조로 본원에 편입된다.

[0002] 이 개시물은 비디오 인코딩 및 비디오 디코딩에 관한 것이다.

### 배경 기술

[0003] 디지털 비디오 기능들은, 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템 (digital direct broadcast system)들, 무선 브로드캐스트 시스템들, 개인 정보 단말 (personal digital assistant; PDA)들, 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더 (e-book reader)들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들 내로 편입될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, 진보된 비디오 코딩 (Advanced Video Coding; AVC), 최근 완결된 ITU-T H.265에 의해 정의된 표준들, 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, 및 이러한 표준들의 확장들에서 설명된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신하고, 수신하고, 인코딩하고, 디코딩하고, 및/또는 저장할 수도 있다.

[0004] 비디오 압축 기법들은 비디오 시퀀스들에 내재된 중복성을 감소시키거나 제거하기 위하여 공간적 (인트라-픽처 (intra-picture)) 예측 및/또는 시간적 (인터-픽처 inter-picture)) 예측을 수행한다. 블록-기반 비디오 코딩을 위하여, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분)는 비디오 블록들로 파티셔닝 (partitioning) 될 수도 있으며, 이 비디오 블록들은 트리블록 (treeblock)들, 코딩 유닛 (coding unit; CU)들 및/또는 코딩 노드들로서 또한 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처에서의 이웃하는 블록들 내의 참조 샘플들에 대한 공간적 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽처에서의 이웃하는 블록들 내의 참조 샘플들에 대한 공간적 예측, 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들로서 지칭될 수도 있고, 참조 픽처들은 참조 프레임들로서 지칭될 수도 있다.

[0005] 공간적 또는 시간적 예측은 코딩되어야 할 블록에 대한 예측 블록으로 귀착된다. 잔차 데이터 (residual data)는 코딩되어야 할 원래의 블록과 예측 블록 사이의 픽셀 차이들을 표현한다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 지시하는 모션 벡터와, 코딩된 블록과 예측 블록과의 사이의 차이를 표시하는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위하여, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어 잔차 변환 계수들로 귀착될 수도 있고, 그 다음으로, 이들은 양자화될 수도 있다. 초기에 2 차원 어레이로 배치된 양자화된 변환 계수들은 변환 계수들의 1 차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩은 훨씬 더 많은 압축을 달성하기 위하여 적용될 수도 있다.

### 발명의 내용

[0006] 이 개시물은 비디오 코딩 (예컨대, 비디오 인코딩 및 비디오 디코딩) 및 압축을 위한 기법들을 설명한다.

특히, 이 개시물은 비디오 데이터의 팔레트-기반 코딩을 위한 기법들을 설명하고, 더 구체적으로, 이 개시물은 팔레트 예측자 (palette predictor) 들을 초기화하기 위한 기법들을 설명한다.

[0007] 하나의 예에서, 비디오 데이터를 디코딩하는 방법은 꽉쳐 파라미터 세트 (picture parameter set; PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자 (PPS-level palette predictor initializer) 들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 십екс 엘리먼트를 수신하는 단계; PPS 에서, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로 (zero) 와 동일하다는 것을 표시하는 제 2 십екс 엘리먼트를 수신하는 단계; 및 제 1 십екс 엘리먼트 및 제 2 십екс 엘리먼트에 기초하여 비디오 데이터의 블록을 디코딩하는 단계를 포함한다.

[0008] 또 다른 예에서, 비디오 데이터를 인코딩하는 방법은 인코딩된 비디오 데이터의 비트스트림에서의 꽉쳐 파라미터 세트 (PPS) 십екс 구조에서의 포함을 위하여, 팔레트 예측자가 PPS-레벨 팔레트 예측자 엔트리들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 십екс 엘리먼트를 생성하는 단계; PPS 십екс 구조에서의 포함을 위하여, PPS 내에 포함된 PPS-레벨 팔레트 예측자 엔트리들의 수가 제로와 동일하다는 것을 표시하는 제 2 십екс 엘리먼트를 생성하는 단계; 및 제 1 십екс 엘리먼트 및 제 2 십екс 엘리먼트를 포함하는 인코딩된 비디오 데이터의 비트스트림을 출력하는 단계를 포함한다.

[0009] 또 다른 예에 따르면, 비디오 데이터를 디코딩하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리, 및 꽉쳐 파라미터 세트 (PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 십екс 엘리먼트를 수신하고; PPS 에서, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 십екс 엘리먼트를 수신하고; 그리고 제 1 십екс 엘리먼트 및 제 2 십екс 엘리먼트에 기초하여 비디오 데이터의 블록을 디코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0010] 또 다른 예에서, 비디오 데이터를 인코딩하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리, 및 인코딩된 비디오 데이터의 비트스트림에서의 꽉쳐 파라미터 세트 (PPS) 십екс 구조에서의 포함을 위하여, 팔레트 예측자가 PPS-레벨 팔레트 예측자 엔트리들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 십екс 엘리먼트를 생성하고; PPS 십екс 구조에서의 포함을 위하여, PPS 내에 포함된 PPS-레벨 팔레트 예측자 엔트리들의 수가 제로와 동일하다는 것을 표시하는 제 2 십екс 엘리먼트를 생성하고; 그리고 제 1 십екс 엘리먼트 및 제 2 십екс 엘리먼트를 포함하는 인코딩된 비디오 데이터의 비트스트림을 출력하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0011] 또 다른 예에서, 비디오 데이터를 디코딩하기 위한 장치는 꽉쳐 파라미터 세트 (PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 십екс 엘리먼트를 수신하기 위한 수단; PPS 에서, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 십екс 엘리먼트를 수신하는 수단; 및 제 1 십екс 엘리먼트 및 제 2 십екс 엘리먼트에 기초하여 비디오 데이터의 블록을 디코딩하는 수단을 포함한다.

[0012] 또 다른 예에서, 비디오 데이터를 인코딩하기 위한 장치는 인코딩된 비디오 데이터의 비트스트림에서의 꽉쳐 파라미터 세트 (PPS) 십екс 구조에서의 포함을 위하여, 팔레트 예측자가 PPS-레벨 팔레트 예측자 엔트리들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 십екс 엘리먼트를 생성하기 위한 수단; PPS 십екс 구조에서의 포함을 위하여, PPS 내에 포함된 PPS-레벨 팔레트 예측자 엔트리들의 수가 제로와 동일하다는 것을 표시하는 제 2 십екс 엘리먼트를 생성하기 위한 수단; 및 제 1 십екс 엘리먼트 및 제 2 십екс 엘리먼트를 포함하는 인코딩된 비디오 데이터의 비트스트림을 출력하기 위한 수단을 포함한다.

[0013] 또 다른 예에서, 컴퓨터 관독가능 저장 매체는 명령들을 저장하고, 상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 경우, 하나 이상의 프로세서들로 하여금, 꽉쳐 파라미터 세트 (PPS) 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 1 십екс 엘리먼트를 수신하게 하고; PPS 에서, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 2 십екс 엘리먼트를 수신하게 하고; 그리고 제 1 십екс 엘리먼트 및 제 2 십екс 엘리먼트에 기초하여 비디오 데이터의 블록을 디코딩하게 한다.

[0014] 또 다른 예에서, 컴퓨터 관독가능 저장 매체는 명령들을 저장하고, 상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 경우, 하나 이상의 프로세서들로 하여금, 인코딩된 비디오 데이터의 비트스트림에서의 꽉쳐 파라미터 세트 (PPS) 십екс 구조에서의 포함을 위하여, 팔레트 예측자가 PPS-레벨 팔레트 예측자 엔트리들을 이용하

여 생성되어야 한다는 것을 표시하는 제 1 신택스 엘리먼트를 생성하게 하고; PPS 신택스 구조에서의 포함을 위하여, PPS 내에 포함된 PPS-레벨 팔레트 예측자 엔트리들의 수가 제로와 동일하다는 것을 표시하는 제 2 신택스 엘리먼트를 생성하게 하고; 그리고 제 1 신택스 엘리먼트 및 제 2 신택스 엘리먼트를 포함하는 인코딩된 비디오 데이터의 비트스트림을 출력하게 한다.

[0015] 개시물의 하나 이상의 예들의 세부 사항들은 동반된 도면들 및 이하의 설명에서 기재된다. 다른 특징들, 목적들, 및 장점들은 설명, 도면들, 및 청구항들로부터 명백할 것이다.

### 도면의 간단한 설명

[0016] 도 1 은 이 개시물에서 설명된 기법들을 사용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 이 개시물의 기법들과 부합하는, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 예를 예시하는 개념도이다.

도 3 은 이 개시물의 기법들과 부합하는, 비디오 블록에 대한 팔레트로의 인덱스들을 결정하는 예들을 예시하는 개념도이다.

도 4 는 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 예시하는 블록도이다.

도 5 는 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 예시하는 블록도이다.

도 6 은 이 개시물의 하나 이상의 기법들에 따라, 비디오 인코더의 일 예의 동작을 예시하는 플로우차트이다.

도 7 은 이 개시물의 하나 이상의 기법들에 따라, 비디오 디코더의 일 예의 동작을 예시하는 플로우차트이다.

도 8 은 이 개시물의 하나 이상의 기법들에 따라, 비디오 디코더의 일 예의 동작을 예시하는 플로우차트이다.

### 발명을 실시하기 위한 구체적인 내용

[0017] 이 개시물은 비디오 코딩 (예컨대, 비디오 인코딩 및 비디오 디코딩) 및 압축을 위한 기법들을 설명한다. 특히, 이 개시물은 비디오 데이터의 팔레트-기반 코딩을 위한 기법들을 설명한다. 팔레트-기반 비디오 코딩에서, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더)는 픽셀들의 블록에 대한 팔레트 테이블 (또한, 간단하게 "팔레트"로서 지칭됨)을 유도하고, 여기서, 팔레트 테이블에서의 각각의 엔트리는 팔레트 테이블로의 인덱스들에 의해 식별되는 컬러 값들을 포함한다.

[0018] 팔레트 모드에서 블록을 코딩하는 것의 일부로서, 비디오 코더는 블록에 대하여 이용되어야 할 팔레트 테이블을 먼저 결정한다. 그 다음으로, 블록의 각각의 픽셀 (또는 샘플)에 대한 팔레트 인덱스들은 팔레트로부터의 어느 엔트리가 픽셀 (샘플)을 예측하거나 복원하기 위하여 이용되어야 하는지를 표시하기 위하여 코딩될 수도 있다. 이 개시물은 일반적으로, 팔레트를 생성하기 위한, 즉, 팔레트의 팔레트 엔트리들을 결정하기 위한 기법들에 관한 것이다. 더 구체적으로, 이 개시물은 팔레트 예측자를 초기화하기 위한 기법들에 관한 것이다. 팔레트 예측자를 초기화하는 것은 일반적으로, 비디오 블록들의 픽처, 슬라이스, 타일, 또는 다른 이러한 그룹의 제 1 블록에 대한 팔레트 예측자를 생성하는 프로세스를 지칭한다. 이하에서 더 상세하게 설명되는 바와 같이, 후속 블록들에 대한 팔레트 예측자는 전형적으로, 이전에 이용된 팔레트 예측자를 업데이트함으로써 생성된다. 그러나, 제 1 블록에 대하여, 이전에 이용된 팔레트 예측자는 이용가능하지 않고, 그러므로, 팔레트 예측자는 초기화될 필요가 있다.

[0019] 팔레트 엔트리들을 시그널링하기 위하여 필요한 비트들을 감소시키기 위하여, 비디오 디코더는 새로운 팔레트 엔트리들을 결정하기 위한 팔레트 예측자를 사용할 수도 있다. 팔레트 예측자는 예를 들어, 이전에 이용된 팔레트로부터의 엔트리들을 포함한다. 일부 구현예들에서, 팔레트 예측자는 가장 최근에 이용된 팔레트와 동일하게 설정될 수도 있고, 가장 최근에 이용된 팔레트의 모든 엔트리들을 포함하는 반면, 다른 구현예들에서는, 팔레트 예측자가 가장 최근에 이용된 팔레트로부터의 전부보다 더 적은 엔트리들을 포함할 수도 있다. 팔레트 예측자는 또한, 단지 가장 최근에 이용된 팔레트 외에, 다른 이전에 이용된 팔레트들로부터의 엔트리들을 포함할 수도 있다. 팔레트 예측자는 블록들을 코딩하기 위하여 이용된 팔레트들과 동일한 크기일 수도 있거나, 블록들을 코딩하기 위하여 이용된 팔레트들보다 더 크거나 더 작을 수도 있다.

[0020] 팔레트 예측자를 사용하여 비디오 데이터의 블록에 대한 팔레트를 생성하기 위하여, 비디오 디코더는 예를 들어, 팔레트 예측자의 연관된 엔트리가 팔레트 내에 포함된다는 것을 표시하는 플래그에 대한 제 1 값, 및 팔

레트 예측자의 연관된 엔트리가 팔레트 내에 포함되지 않는다는 것을 표시하는 플래그에 대한 제 2 값과 함께, 팔레트 예측자의 각각의 엔트리에 대한 1-비트 플래그를 수신할 수도 있다. 팔레트 예측자가 비디오 데이터의 블록에 대하여 이용된 팔레트보다 더 큰 사례들에서, 비디오 디코더는 일단 팔레트에 대한 최대 크기가 도달되면, 플래그들을 수신하는 것을 정지시킬 수도 있다.

[0021] 팔레트의 일부 엔트리들은 팔레트 예측자를 이용하여 결정되는 대신에, 직접적으로 시그널링될 수도 있다. 이러한 엔트리들에 대하여, 비디오 디코더는 예를 들어, 엔트리와 연관된 루마 (luma) 및 2 개의 크로마 (chroma) 컴포넌트들에 대한 컬러 값들을 표시하는 3 개의 별도의  $m$ -비트 값들을 수신할 수도 있고, 여기서,  $m$ 은 비디오 데이터의 비트 심도를 표현한다. 팔레트 예측자로부터 유도된 그 팔레트 엔트리들은 직접적으로 시그널링된 엔트리들에 대하여 필요한 다수의  $m$ -비트 값들과 비교하여, 1-비트 플래그를 오직 요구하므로, 팔레트 예측자를 이용하여 일부 또는 모든 팔레트 엔트리들을 시그널링하는 것은 새로운 팔레트의 엔트리들을 시그널링하기 위하여 필요한 비트들의 수를 감소시킬 수도 있고, 그 결과, 팔레트 모드 코딩의 전체적인 코딩 효율을 개선시킬 수도 있다.

[0022] 위에서 설명된 바와 같이, 많은 사례들에서, 하나의 블록에 대한 팔레트 예측자는 하나 이상의 이미 코딩된 블록들을 코딩하기 위하여 이용된 팔레트에 기초하여 결정된다. 그러나, 슬라이스 또는 타일에서의 제 1 코딩 트리 유닛을 코딩할 때와 같은 일부 코딩 시나리오들에서는, 이전에 코딩된 블록의 팔레트가 이용불가능할 수도 있고, 이에 따라, 팔레트 예측자는 이전에 이용된 팔레트들의 엔트리들을 이용하여 생성될 수 없다. 이러한 사례에서는, 팔레트 예측자 초기화자들이 시퀀스 파라미터 세트 (sequence parameter set; SPS) 및/또는 픽처 파라미터 세트 (PPS)에서 시그널링될 수도 있다. 팔레트 예측자 초기화자들은 이전에 이용된 팔레트가 이용가능하지 않을 때에 팔레트 예측자를 생성하기 위하여 이용될 수도 있는 값들이다.

[0023] PPS 는 일반적으로, 각각의 슬라이스 세그먼트 헤더에서 발견된 신팩스 엘리먼트에 의해 결정된 바와 같은 제로 이상의 전체 코딩된 픽처들에 적용하는 신팩스 엘리먼트들을 포함하는 신팩스 구조를 지칭한다. SPS 는 일반적으로, 각각의 슬라이스 세그먼트 헤더에서 발견된 신팩스 엘리먼트에 의해 참조된 PPS에서 발견된 신팩스 엘리먼트의 내용에 의해 결정된 바와 같은 제로 이상의 전체 CVS 들에 적용하는 신팩스 엘리먼트들을 포함하는 신팩스 구조를 지칭한다. 이에 따라, SPS 는 일반적으로, PPS 보다 더 높은 레벨의 신팩스 구조인 것으로 고려되어, 이는 SPS 내에 포함된 신팩스 엘리먼트들이 PPS 내에 포함된 신팩스 엘리먼트들과 비교하여, 일반적으로 덜 빈번하게 변경되고 비디오의 더 큰 부분에 적용한다는 것을 의미한다.

[0024] 팔레트 모드 코딩의 현존하는 구현예들에서, 팔레트 예측자 초기화가 SPS-레벨에서 인에이블될 경우, PPS-레벨 신팩스는 비디오 디코더가 SPS-레벨 팔레트 예측자 엔트리들 (또한, SPS-레벨 팔레트 예측자 초기화자들로서 본원에서 지칭됨)을 이용하여 팔레트 예측자를 생성하는 것과, PPS-레벨 팔레트 예측자 엔트리들 (또한, PPS-레벨 팔레트 예측자 초기화자들로서 본원에서 지칭됨)을 이용하여 팔레트 예측자를 생성하는 것 사이에서 선택하는 것을 오직 가능하게 한다. 팔레트 모드 코딩의 현존하는 구현예들은 팔레트 예측자 초기화가 일단 SPS 레벨에서 인에이블되면, PPS 레벨에서 디스에이블되는 것을 허용하지 않는다. 이 개시물은 팔레트 예측자의 크기를 제로로 설정함으로써 PPS 레벨에서 팔레트 예측자 초기화를 디스에이블하기 위한 기법들을 도입한다.

[0025] 또한, 팔레트 모드 코딩의 현존하는 구현예들에서, 비디오 디코더는 SPS 팔레트 예측자 초기화자들 또는 PPS 팔레트 예측자 초기화자들의 어느 하나에 기초하여 팔레트 예측자를 초기화한다. 이 개시물의 기법들에 따르면, 비디오 디코더는 SPS 팔레트 예측자 초기화자들 및 PPS 팔레트 예측자 초기화자들의 양자에 기초하여 팔레트 예측자를 초기화할 수도 있다. 예를 들어, 팔레트 예측자는 SPS 팔레트 예측자 엔트리들과, 그 다음으로, PPS 팔레트 예측자 엔트리들, 또는 그 반대로 포함함으로써 초기화될 수도 있다. 다른 예의 구현 예들에서, 비디오 디코더는 SPS 팔레트 예측자 초기화자들의 전부가 아니라, 오직 일부를 이용할 수도 있다. 예를 들어, 팔레트 예측자를 초기화할 때, 재이용 플래그는 SPS 팔레트 예측자 엔트리들 중의 어느 엔트리들이 팔레트 예측자 초기화를 위하여 재이용되어야 하는지를 표시하기 위하여 엔트리마다 시그널링될 수도 있다.

[0026] 비디오 코더는 슬라이스 또는 타일에서의 제 1 블록 (예컨대, HEVC에서의 코딩 유닛 또는 코딩 트리 유닛, 또는 상이한 코딩 표준에서의 상이한 타입의 블록)에 대한 팔레트 예측자를 결정, 예컨대, 초기화할 수도 있다. 그 다음으로, 블록에 대한 팔레트 테이블은 팔레트 예측자로부터 결정될 수도 있다. 팔레트 예측자가 제 1 블록에 대하여 초기화된 후, 제 1 블록에 대한 팔레트 예측자는 제 2 블록에 대한 새로운 팔레트 예측자를 결정하기 위하여 업데이트될 수도 있다.

[0027] 도 1은 이 개시물에서 설명된 기법들을 사용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템 (10)을 예시하는 블록도이다. 도 1에서 도시된 바와 같이, 시스템 (10)은 목적지 디바이스 (14)에 의해 더 이후의

시간에 디코딩되어야 할 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12)를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는, 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 (set-top) 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 광범위한 디바이스들 중의 임의의 것을 포함할 수도 있다. 일부 경우들에는, 소스 디바이스 (12) 및 목적지 디바이스 (14)가 무선 통신을 위하여 구비될 수도 있다.

[0028] 목적지 디바이스 (14)는 링크 (16)를 통해 디코딩되어야 할 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16)는 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 링크 (16)는 소스 디바이스 (12)가 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14)로 직접 송신하는 것을 가능하게 하기 위한 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조될 수도 있고, 목적지 디바이스 (14)로 송신될 수도 있다. 통신 매체는 라디오 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12)로부터 목적지 디바이스 (14)로의 통신을 가능하게 하기 위해 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0029] 대안적으로, 인코딩된 데이터는 출력 인터페이스 (22)로부터 저장 디바이스 (26)로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (26)로부터 액세스될 수도 있다. 저장 디바이스 (26)는 하드 드라이브, 블루-레이 (Blu-ray) 디스크들, DVD 들, CD-ROM 들, 플래시 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적당한 디지털 저장 매체들과 같은, 다양한 분산되거나 국소적으로 액세스된 데이터 저장 매체들 중의 임의의 것을 포함할 수도 있다. 추가의 예에서, 저장 디바이스 (26)는 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오를 유지할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14)는 스트리밍 또는 다운로드를 통해 저장 디바이스 (26)로부터 저장된 비디오 데이터를 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있으며 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 일 예의 파일 서버들은 (예컨대, 웹사이트를 위한) 웹 서버, FTP 서버, 네트워크 연결 저장 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다.

목적지 디바이스 (14)는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터를 액세스할 수도 있다. 이것은 파일 서버 상에 저장된 인코딩된 비디오 데이터를 액세스하기에 적당한 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 저장 디바이스 (26)로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양자의 조합일 수도 있다.

[0030] 이 개시물의 기법들은 무선 애플리케이션들 또는 세팅들로 반드시 제한되지는 않는다. 기법들은 오버-디-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상에서의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들 중의 임의의 것의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 및/또는 화상 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위하여 일방향 (one-way) 또는 양방향 (two-way) 비디오 송신을 지원하도록 구성될 수도 있다.

[0031] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 일부 경우들에는, 출력 인터페이스 (22)가 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12)에서, 비디오 소스 (18)는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 비디오 컨텐츠 제공자로부터 비디오를 수신하기 위한 비디오 공급 인터페이스, 및/또는 소스 비디오로서 컴퓨터 그래픽 데이터를 생성하기 위한 컴퓨터 그래픽 시스템과 같은 소스, 또는 이러한 소스들의 조합을 포함할 수도 있다. 하나의 예로서, 비디오 소스 (18)가 비디오 카메라일 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14)는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 이 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

- [0032] 캡처된 (captured), 프리-캡처된 (pre-captured), 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20)에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12)의 출력 인터페이스 (22)를 통해 목적지 디바이스 (14)로 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안적으로), 디코딩 및/또는 재생을 위하여, 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 더 이후의 액세스를 위해 저장 디바이스 (26) 상으로 저장될 수도 있다.
- [0033] 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 일부 경우들에는, 입력 인터페이스 (28)가 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14)의 입력 인터페이스 (28)는 링크 (16)를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16)를 통해 통신되거나 저장 디바이스 (26) 상에서 제공된 인코딩된 비디오 데이터는 비디오 데이터를 디코딩함에 있어서, 비디오 디코더 (30)와 같은 비디오 디코더에 의한 이용을 위하여 비디오 인코더 (20)에 의해 생성된 다양한 신택스 엘리먼트들을 포함할 수도 있다. 이러한 신택스 엘리먼트들은 통신 매체 상에서 송신되거나, 저장 매체 상에 저장되거나, 파일 서버에 저장된 인코딩된 비디오 데이터와 함께 포함될 수도 있다.
- [0034] 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합될 수도 있거나, 목적지 디바이스 (14)의 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14)는 통합된 디스플레이 디바이스를 포함할 수도 있고, 또한, 외부의 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14)는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (liquid crystal display; LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode; OLED) 디스플레이, 또는 또 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중의 임의의 것을 포함할 수도 있다.
- [0035] 비디오 인코더 (20) 및 비디오 디코더 (30)는 최근-완결된 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 압축 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HEVC Test Model; HM)을 준수할 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30)는 MPEG-4, Part 10, 진보된 비디오 코딩 (AVC), 또는 이러한 표준들의 확장들로서 대안적으로 지정된 ITU-T H.264 표준과 같은 다른 독점적 또는 산업 표준들에 따라 동작할 수도 있다. 그러나, 이 개시물의 기법들은 임의의 특정한 코딩 표준으로 제한되지는 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263을 포함한다. 비디오 코딩 표준들은 그 스케일러블 비디오 코딩 (Scalable Video Coding; SVC) 및 멀티뷰 비디오 코딩 (Multiview Video Coding; MVC) 확장들을 포함하는, ITU-T H.261, ISO/IEC MPEG-1 비주얼 (Visual), ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼, ITU-T H.264 (또한, ISO/IEC MPEG-4 AVC로서 알려짐), 및 ITU-T H.265/HEVC를 포함한다.
- [0036] 최근, 모션을 갖는 텍스트 및 그래픽과 같은 스크린-컨텐츠 자료에 대한 새로운 코딩 툴들의 연구가 요청되었고, 스크린 컨텐츠에 대한 코딩 효율을 개선시키는 기술들이 제안되었다. 이 활동은 현재 진행 중이고, H.265/HEVC 스크린 컨텐츠 코딩 (screen content coding; SCC) 확장으로서 완료될 것으로 예상된다. JCTVC-U1005, ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11의 비디오 코딩에 관한 공동 협력팀 (Joint Collaborative Team on Video Coding; JCT-VC), 21 차 회의: Warsaw, PL, 19 June - 26 June 2015는 HEVC SCC 작업 초안 4 (WD4) JCTVC-U1005를 기재하고, [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/21\\_Warsaw/wg11/JCTVC-U1005-v2.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/21_Warsaw/wg11/JCTVC-U1005-v2.zip)에서 입수가능하고, 그 전체 내용은 참조로 본원에 편입된다.
- [0037] 이 개시물의 기법들은 설명의 용이함을 위하여 HEVC 용어를 사용할 수도 있다. 그러나, 이 개시물의 기법들은 HEVC로 제한된다는 것이 가정되지 않아야 하고, 실제로, 이 개시물의 기법들은 HEVC 및 그 확장들에 대한 계승자 표준들, 및 다른 미래의 표준들 또는 프로세스들에서 구현될 수도 있다는 것이 명시적으로 고려된다.
- [0038] 도 1에서 도시되지 않았지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통의 데이터 스트림 또는 별도의 데이터 스트림들에서 오디오 및 비디오 양자의 인코딩을 처리하기 위한 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능한 경우, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol; UDP)과 같은 다른 프로토콜들을 준수할 수도 있다.
- [0039] 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (digital signal processor; DSP)들, 주문형 집적 회로 (application specific integrated circuit; ASIC)들, 필드 프로그래밍 가능한 게이트 어레이 (field programmable gate array; FPGA)들, 개별 로직,

소프트웨어, 하드웨어, 펌웨어 또는 그 임의의 조합들과 같은 다양한 적당한 인코더 회로부 중의 임의의 것으로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현될 때, 디바이스는 소프트웨어를 위한 명령들을 적당한 비-일시적 컴퓨터-관독가능 매체 내에 저장할 수도 있고, 이 개시물의 기법들을 수행하기 위하여 하나 이상의 프로세서들을 이용하여 명령들을 하드웨어로 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 인코더들 또는 디코더들 중의 어느 하나는 조합된 인코더/디코더 (combined encoder/decoder; CODEC)의 일부로서 개개의 디바이스 내에 통합될 수도 있다.

[0040] 위에서 도입된 바와 같이, JCT-VC는 HEVC 표준의 개발을 최근에 완결하였다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM)로서 지정된 비디오 코딩 디바이스의 진화형 모델에 기초하였다. HM은 예컨대, ITU-T H.264/AVC에 따라 혼존하는 디바이스들에 관하여 비디오 코딩 디바이스들의 몇몇 추가적인 기능들을 추정한다. 예를 들어, H.264가 9개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM은 35개만큼 많은 인트라-예측 인코딩 모드들을 제공할 수도 있다.

[0041] HEVC 및 다른 비디오 코딩 사양들에서, 비디오 시퀀스는 전형적으로 일련의 픽처들을 포함한다. 픽처들은 또한, "프레임들"로서 지정될 수도 있다. 픽처는  $S_L$ ,  $S_{Cb}$  및  $S_{Cr}$ 로 나타낸 3개의 샘플 어레이들을 포함할 수도 있다.  $S_L$ 은 루마 샘플 (luma sample)들의 2차원 어레이 (즉, 블록)이다.  $S_{Cb}$ 는 Cb 크로미넌스 샘플 (chrominance sample)들의 2차원 어레이이다.  $S_{Cr}$ 은 Cr 크로미넌스 샘플들의 2차원 어레이이다. 크로미넌스 샘플들은 또한, 본원에서 "크로마" 샘플들로서 지정될 수도 있다. 다른 사례들에서, 픽처는 단색일 수도 있고 루마 샘플들의 어레이를 오직 포함할 수도 있다.

[0042] 픽처의 인코딩된 표현을 생성하기 위하여, 비디오 인코더 (20)는 코딩 트리 유닛 (coding tree unit; CTU)들의 세트를 생성할 수도 있다. CTU들의 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하기 위해 이용된 신팩스 구조들을 포함할 수도 있다. 단색 픽처들, 또는 3개의 별도의 컬러 평면들을 가지는 픽처들에서는, CTU가 단일 코딩 트리 블록과, 코딩 트리 블록의 샘플들을 코딩하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU는 또한, "트리 블록" 또는 "최대 코딩 유닛 (largest coding unit)" (LCU)으로서 지정될 수도 있다. HEVC의 CTU들은 H.264/AVC와 같은 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU는 반드시 특정한 크기로 제한되지는 않고, 하나 이상의 코딩 유닛 (CU)들을 포함할 수도 있다. 슬라이스는 레스터 스캔 순서 (raster scan order)로 연속으로 순서화된 정수 (integer number)의 CTU들을 포함할 수도 있다.

[0043] 코딩된 CTU를 생성하기 위하여, 비디오 인코더 (20)는 코딩 트리 블록들을 코딩 블록들, 이 때문에, 명칭 "코딩 트리 유닛들"로 분할하기 위하여, CTU의 코딩 트리 블록들에 대해 쿼드-트리 파티셔닝 (quad-tree partitioning)을 재귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 NxN 블록일 수도 있다. CU는 루마 샘플들의 코딩 블록과, 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 가지는 픽처의 크로마 샘플들의 2개의 대응하는 코딩 블록들과, 코딩 블록들의 샘플들을 코딩하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 단색 픽처들, 또는 3개의 별도의 컬러 평면들을 가지는 픽처들에서는, CU가 단일 코딩 블록과, 코딩 블록의 샘플들을 코딩하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다.

[0044] 비디오 인코더 (20)는 CU의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록이다. CU의 예측 유닛 (PU)은 루마 샘플들의 예측 블록, 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록들을 예측하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 단색 픽처들, 또는 3개의 별도의 컬러 평면들을 가지는 픽처들에서는, PU가 단일 예측 블록과, 예측 블록을 예측하기 위하여 이용된 신팩스 구조들을 포함할 수도 있다. 비디오 인코더 (20)는 CU의 각각의 PU의 루마, Cb, 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.

[0045] 비디오 인코더 (20)는 PU에 대한 예측 블록들을 생성하기 위하여 인트라 예측 또는 인터 예측을 이용할 수도 있다. 비디오 인코더 (20)가 PU의 예측 블록들을 생성하기 위하여 인트라 예측을 이용할 경우, 비디오 인코더 (20)는 PU와 연관된 픽처의 디코딩된 샘플들에 기초하여 PU의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20)가 PU의 예측 블록들을 생성하기 위하여 인터 예측을 이용할 경우, 비디오 인코더 (20)는 PU와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU의 예측 블록들을 생성할 수도 있다.

있다.

[0046] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU 들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성한 후, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에서의 각각의 샘플은 CU 의 예측 루마 블록들 중의 하나에서의 루마 샘플과, CU 의 원래의 루마 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 표시한다. 게다가, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 생성할 수도 있다.

CU 의 Cb 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cb 블록들 중의 하나에서의 Cb 샘플과, CU 의 원래의 Cb 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 표시할 수도 있다. 비디오 인코더 (20) 는 또한, CU 에 대한 Cr 잔차 블록을 생성할 수도 있다. CU 의 Cr 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cr 블록들 중의 하나에서의 Cr 샘플과, CU 의 원래의 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 표시할 수도 있다.

[0047] 또한, 비디오 인코더 (20) 는 CU 의 루마, Cb, 및 Cr 잔차 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해하기 위하여 퀼드-트리 파티셔닝을 이용할 수도 있다. 변환 블록은 동일한 변환에 적용되는 샘플들의 직사각형 (예컨대, 정사각형 또는 비-정사각형) 블록이다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하기 위하여 이용된 십екс 구조들을 포함할 수도 있다. 이에 따라, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록 (sub-block) 일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다. 단색 픽처들, 또는 3 개의 별도의 컬러 평면들을 가지는 픽처들에서는, TU 가 단일 변환 블록과, 변환 블록의 샘플들을 변환하기 위하여 이용된 십екс 구조들을 포함할 수도 있다.

[0048] 비디오 인코더 (20) 는 TU 에 대한 루마 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용할 수도 있다. 계수 블록은 변환 계수들의 2 차원 어레이일 수도 있다. 변환 계수는 스칼라량 (scalar quantity) 일 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cb 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용할 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cr 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용할 수도 있다.

[0049] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록, 또는 Cr 계수 블록) 을 생성한 후, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로, 변환 계수들을 표현하기 위하여 이용된 데이터의 양을 아마도 감소시키기 위하여 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 십екс 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 십екс 엘리먼트들에 대해 컨텍스트-적응 2 진 산술 코딩 (Context-Adaptive Binary Arithmetic Coding; CABAC) 을 수행할 수도 있다.

[0050] 비디오 인코더 (20) 는 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 NAL 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은 에뮬레이션 방지 비트들이 필요에 따라 산재된 RBSP 의 형태로, NAL 유닛에서의 데이터의 타입의 표시 및 그 데이터를 포함하는 바이트들을 포함하는 십екс 구조이다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하고 RBSP 를 캡슐화 (encapsulate) 한다. NAL 유닛 헤더는 NAL 유닛 타입 코드를 표시하는 십екс 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 타입 코드는 NAL 유닛의 타입을 표시한다. RBSP 는 NAL 유닛 내에서 캡슐화되는 정수 개수의 바이트들을 포함하는 십екс 구조일 수도 있다. 일부 사례들에서, RBSP 는 제로 비트들을 포함한다.

[0051] 상이한 타입들의 NAL 유닛들은 상이한 타입들의 RBSP 들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 타입은 PPS 에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 2 타입은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 3 타입은 SEI 메시지들에 대한 RBSP 를 캡슐화할 수도 있는 등과 같다. (파라미터 세트들 및 SEI 메시지들에 대한 RBSP들과 대조적으로) 비디오 코딩 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 VCL NAL 유닛들로서 지칭될 수도 있다.

[0052] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 게다가, 비디오 디코더 (30) 는 비트스트림으로부터 십екс 엘리먼트들을 획득하기 위하여 비트스트림을 파싱 (parse) 할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 십екс 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하기 위한 프로세스는 비디오 인코더 (20) 에 의해 수행된 프로세스와 일반적으로 상반적일 수도 있다. 게다가, 비디오 디코더 (30) 는 현재

의 CU 의 TU 들과 연관된 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 TU 들과 연관된 변환 블록들을 복원하기 위하여 계수 블록들에 대해 역변환들을 수행할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 PU 들에 대한 예측 블록들의 샘플들을 현재의 CU 의 TU 들의 변환 블록들의 대응하는 샘플들에 추가함으로써, 현재의 CU 의 코딩 블록들을 복원할 수도 있다. 꾹처의 각각의 CU 에 대한 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 꾹처를 복원할 수도 있다.

[0053] 도 2 는 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 예를 예시하는 개념도이다. 도 2 의 예는 팔레트들의 제 1 세트 (즉, 제 1 팔레트들 (184)) 과 연관되는 제 1 블록 (180) 및 팔레트들의 제 2 세트 (즉, 제 2 팔레트들 (192)) 와 연관되는 제 2 블록 (188) 을 가지는 꾹처 (178) 를 포함한다. 이하에서 더 상세하게 설명된 바와 같이, 제 2 팔레트들 (192) 은 제 1 팔레트들 (184) 에 기초하여 결정될 수도 있다. 꾹처 (178) 는 또한, 인트라-예측 코딩 모드로 코딩된 블록 (196), 및 인터-예측 코딩 모드로 코딩되는 블록 (200) 을 포함한다.

[0054] 도 2 의 기법들은 비디오 인코더 (20) (도 1 및 도 2) 및 비디오 디코더 (30) (도 1 및 도 3) 의 문맥에서 설명된다. 팔레트-기반 코딩 기법들은 예를 들어, 비디오 데이터의 블록에 대한 코딩 모드로서 이용될 수도 있다. 도 2 에 대하여 설명된 다양한 블록들은 예를 들어, HEVC 의 문맥에서 이해된 바와 같이 CTU 들, CU 들, PU 들, 또는 TU 들에 대응할 수도 있지만, 도 2 에서 설명된 블록들은 임의의 특정한 코딩 표준의 블록 구조로 제한되지는 않고, HEVC 와는 상이한 블록 포맷들을 사용하는 미래의 코딩 표준들과 호환가능할 수도 있다.

[0055] 일반적으로, 팔레트는 현재 코딩되고 있는 블록 (예컨대, 도 2 의 예에서의 블록 (188)) 에 대하여 지배적 및/ 또는 대표적인 다수의 꾹셀 값들을 지칭한다. 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 은 다수의 팔레트들을 포함하는 것으로서 도시된다. 일부 예들에서, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 블록의 각각의 컬러 컴포넌트에 대하여 별도로 팔레트들을 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 블록의 루마 (Y) 컴포넌트에 대한 팔레트, 블록의 크로마 (U) 컴포넌트에 대한 또 다른 팔레트, 및 블록의 크로마 (V) 컴포넌트에 대한 또 다른 팔레트를 인코딩할 수도 있다. 이 예에서, Y 팔레트의 엔트리들은 블록의 꾹셀들의 Y 값들을 표현할 수도 있고, U 팔레트의 엔트리들은 블록의 꾹셀들의 U 값들을 표현할 수도 있고, V 팔레트의 엔트리들은 블록의 꾹셀들의 V 값들을 표현할 수도 있다.

[0056] 다른 예들에서, 비디오 인코더 (20) 는 블록의 모든 컬러 컴포넌트들에 대한 단일 팔레트를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는  $Y_i$ ,  $U_i$ , 및  $V_i$  를 포함하는, 트리플 값인  $i$  번째 엔트리를 가지는 팔레트를 인코딩할 수도 있다. 이 경우, 팔레트는 꾹셀들의 컴포넌트들의 각각에 대한 값들을 포함한다. 따라서, 다수의 개별적인 팔레트들을 가지는 팔레트들의 세트로서의 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 의 표현은 단지 하나의 예이고 제한하는 것으로 의도된 것은 아니다.

[0057] 도 2 의 예에서, 제 1 팔레트들 (184) 의 각각은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3 을 각각 가지는 3 개의 엔트리들 (202 내지 206) 을 포함한다. 엔트리들 (202 내지 206) 은 인덱스 값들을 꾹셀 값 A, 꾹셀 값 B, 및 꾹셀 값 C 를 각각 포함하는 꾹셀 값들에 관련시킨다. 제 1 팔레트들 (184) 의 각각은 실제적으로 인덱스들 및 열 헤더 (column header) 들을 포함하지 않지만, 꾹셀 값들 A, B 및 C 를 포함하기만 하고, 인덱스들은 팔레트에서 엔트리들을 식별하기 위하여 이용되는 것이 주목되어야 한다.

[0058] 본원에서 설명된 바와 같이, 제 1 블록 (180) 의 실제적인 꾹셀 값들을 직접적으로 코딩하는 것이 아니라, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 인덱스들 1 내지 3 을 이용하여 블록의 꾹셀들을 코딩하기 위하여 팔레트-기반 코딩을 이용할 수도 있다. 즉, 제 1 블록 (180) 의 각각의 꾹셀 위치에 대하여, 비디오 인코더 (20) 는 꾹셀에 대한 인덱스 값을 인코딩할 수도 있고, 여기서, 인덱스 값은 제 1 팔레트들 (184) 중의 하나 이상에서의 꾹셀 값과 연관된다. 비디오 디코더 (30) 는 비트스트림으로부터 인덱스 값을 획득할 수도 있고, 인덱스 값들 및 제 1 팔레트들 (184) 중의 하나 이상을 이용하여 꾹셀 값들을 복원할 수도 있다. 다시 말해서, 블록에 대한 각각의 개개의 인덱스 값에 대하여, 비디오 디코더 (30) 는 제 1 팔레트들 (184) 중의 하나에서의 엔트리를 결정할 수도 있다. 비디오 디코더 (30) 는 블록에서의 개개의 인덱스 값을 팔레트에서의 결정된 엔트리에 의해 특정된 꾹셀 값으로 대체할 수도 있다. 비디오 인코더 (20) 는 팔레트-기반 디코딩에서의 비디오 디코더 (30) 에 의한 이용을 위한 인코딩된 비디오 데이터 비트스트림 내의 제 1 팔레트들 (184) 을 송신할 수도 있다. 일반적으로, 하나 이상의 팔레트들은 각각의 블록에 대하여 송신될 수도 있거나, 상이한 블록들 사이에서 공유될 수도 있다.

[0059] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 하나 이상의 팔레트 예측자들에 기초하여 제 2 팔레트들 (192) 을 결정할 수도 있다. 팔레트 예측자들은 예를 들어, 제 1 팔레트들 (184)

의 엔트리들의 일부 또는 전부를 포함할 수도 있고, 아마도, 다른 팔레트들로부터의 엔트리들을 마찬가지로 포함할 수도 있다. 일부 예들에서, 팔레트 예측자들은 선입선출 (first-in-first-out) 방법론을 이용하여 생성될 수도 있고, 여기서, 제 1 팔레트들 (184)의 엔트리들을 팔레트 예측자들에 추가할 시에, 팔레트 예측자들에서의 현재 가장 오래된 엔트리들은 최대 크기에서 또는 최대 크기 미만에서 팔레트 예측자들을 유지하기 위하여 삭제된다. 다른 예들에서, 팔레트 예측자는 상이한 기법들을 이용하여 업데이트 및/또는 유지될 수도 있다. 이 개시물의 기법들은 일반적으로, 팔레트 예측자를 초기화하는 것에 관한 것이고, 팔레트 예측자들을 업데이트하고 유지하기 위한 수많은 상이한 기법들과 함께 이용될 수도 있다.

[0060] 이 개시물의 기법들은 일반적으로 팔레트 예측자에 기초하여 팔레트를 생성하는 것에 관한 것이고, 더 구체적으로, 팔레트 예측자들의 초기화를 제어하기 위한 기법들에 관한 것이지만, 이 개시물의 기법들은 팔레트들을 생성하기 위한 다른 기법들과 함께 이용될 수도 있다. 하나의 예로서, 비디오 인코더 (20)는 블록에 대한 팔레트가 (공간적으로 또는 스캔 순서에 기초하여) 이웃하는 블록 또는 인과적 이웃의 가장 빈번한 샘플들과 같은 하나 이상의 다른 블록들과 연관된 하나 이상의 팔레트들로부터 예측되는지 여부를 표시하기 위하여, (예로서, 제 2 블록 (188)을 포함하는) 각각의 블록에 대한 pred\_palette\_flag를 인코딩할 수도 있다. 예를 들어, 이러한 플래그의 값이 1과 동일할 때, 비디오 디코더 (30)는, 제 2 블록 (188)에 대한 제 2 팔레트들 (192)이 하나 이상의 이미 디코딩된 팔레트들로부터 예측되고, 그러므로, 제 2 블록 (188)에 대한 새로운 팔레트들이 pred\_palette\_flag를 포함하는 비트스트림 내에 포함되지 않는 것으로 결정할 수도 있다. 이러한 플래그가 제로와 동일할 때, 비디오 디코더 (30)는 제 2 블록 (188)에 대한 제 2 팔레트들 (192)이 새로운 팔레트로서 비트스트림 내에 포함되는 것으로 결정할 수도 있다. 일부 예들에서, pred\_palette\_flag는 블록의 각각의 상이한 컬러 컴포넌트에 대하여 별도로 코딩될 수도 있다 (예컨대, YUV 비디오에서의 블록에 대하여 3개의 플래그들, Y에 대하여 하나, U에 대하여 하나, 및 V에 대하여 하나). 다른 예들에서, 단일 pred\_palette\_flag는 블록의 모든 컬러 컴포넌트들에 대하여 코딩될 수도 있다.

[0061] 상기 예에서, pred\_palette\_flag는 현재의 블록에 대한 팔레트의 엔트리들 중의 임의의 것이 예측되는지 여부를 표시하기 위하여 블록마다 시그널링된다. 이것은 제 2 팔레트들 (192)이 제 1 팔레트들 (184)과 동일하며 추가적인 정보가 시그널링되지 않는다는 것을 의미한다. 다른 예들에서, 하나 이상의 신팩스 엘리먼트들은 엔트리마다에 기초하여 시그널링될 수도 있다. 즉, 플래그는 그 엔트리가 현재의 팔레트에서 존재하는지 여부를 표시하기 위하여 팔레트 예측자의 각각의 엔트리에 대하여 시그널링될 수도 있다. 위에서 언급된 바와 같이, 팔레트 엔트리가 예측되지 않을 경우, 팔레트 엔트리는 명시적으로 시그널링될 수도 있다. 다른 예들에서는, 이 2개의 방법들이 조합될 수 있다. 예를 들어, 먼저, pred\_palette\_flag가 시그널링된다. 플래그가 0일 경우, 엔트리마다의 예측 플래그가 시그널링될 수도 있다. 게다가, 새로운 엔트리들의 수 및 그 명시적 값들이 시그널링될 수도 있다.

[0062] 제 1 팔레트들 (184)에 관하여 제 2 팔레트들 (192)을 결정할 때 (예컨대, pred\_palette\_flag는 1과 동일함), 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 예측 팔레트들, 이 예에서는 제 1 팔레트들 (184)이 이로부터 결정되는 하나 이상의 블록들을 위치시킬 수도 있다. 예측 팔레트들은 (예컨대, (공간적으로 또는 스캔 순서에 기초하여) 이웃하는 블록들 또는 인과적 이웃의 가장 빈번한 샘플들과 같은) 현재 코딩되고 있는 블록, 즉, 제 2 블록 (188)의 하나 이상의 이웃하는 블록들과 연관될 수도 있다. 하나 이상의 이웃하는 블록들의 팔레트들은 예측자 팔레트와 연관될 수도 있다. 도 2에서 예시된 예와 같은 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 제 2 블록 (188)에 대한 예측 팔레트를 결정할 때, 좌측의 이웃하는 블록, 제 1 블록 (180)을 위치시킬 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 상부 블록, 블록 (196)과 같은, 제 2 블록 (188)에 관한 다른 위치들에서 하나 이상의 블록들을 위치시킬 수도 있다. 또 다른 예에서, 팔레트 모드를 이용하였던 스캔 순서에서의 최후 블록에 대한 팔레트는 예측 팔레트로서 이용될 수도 있다.

[0063] 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 계층구조 (hierarchy)에 기초하여 팔레트 예측을 위한 블록을 결정할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팔레트 예측을 위한 좌측의 이웃하는 블록, 제 1 블록 (180)을 초기에 식별할 수도 있다. 좌측의 이웃하는 블록이 예측을 위하여 이용가능하지 않을 경우 (예컨대, 좌측의 이웃하는 블록이 인트라-예측 모드 또는 인트라-예측 모드와 같은 팔레트-기반 코딩 모드 이외의 모드로 코딩되거나, 꽉쳐 또는 슬라이스의 가장 좌측 예지에서 위치됨), 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 상부의 이웃하는 블록, 블록 (196)을 식별할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팔레트 예측을 위하여 이용가능한 팔레트를 가지는 블록을 위치시킬 때까지, 로케이션들의 미리 결정된 순서에 따라 이용가능한 블록에 대한 검색을 계속할 수도 있다.

일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 다수의 블록들 및/또는 이웃하는 블록의 복원된 샘플들에 기초하여 예측 팔레트를 결정할 수도 있다.

[0064] 도 2 의 예는 단일 블록 (즉, 제 1 블록 (180))으로부터의 예측 팔레트들로서 제 1 팔레트들 (184)을 예시하지만, 다른 예들에서는, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)가 이웃하는 블록들의 조합으로부터의 예측을 위한 팔레트들을 위치시킬 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더는 (공간적으로 또는 스캔 순서에서의) 복수의 이웃하는 블록들 중의 하나 또는 조합의 팔레트들에 기초하여 예측 팔레트를 생성하기 위하여 하나 이상의 공식들, 함수들, 규칙들 등을 적용할 수도 있다.

[0065] 또 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팔레트 예측을 위한 다수의 잠재적인 후보들을 포함하는 후보 리스트를 구성할 수도 있다. 이러한 예들에서, 비디오 인코더 (20)는 팔레트 예측을 위하여 이용된 현재의 블록이 그로부터 선택되는 (예컨대, 팔레트를 복사하는) 리스트에서의 후보 블록을 표시하기 위하여 후보 리스트로의 인덱스를 인코딩할 수도 있다. 비디오 디코더 (30)는 동일한 방식으로 후보 리스트를 구성할 수도 있고, 인덱스를 디코딩할 수도 있고, 현재의 블록과 함께 이용하기 위한 대응하는 블록의 팔레트를 선택하기 위하여 디코딩된 인덱스를 이용할 수도 있다. 또 다른 예에서, 리스트에서의 표시된 후보 블록의 팔레트는 현재의 블록에 대한 현재의 팔레트의 엔트리마다의 예측을 위한 예측 팔레트로서 이용될 수도 있다.

[0066] 예시의 목적들을 위한 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 현재 코딩되고 있는 블록의 상부에 위치되는 하나의 블록과, 현재 코딩되고 있는 블록의 좌측에 위치되는 하나의 블록을 포함하는 후보 리스트를 구성할 수도 있다. 이 예에서, 비디오 인코더 (20)는 후보 선택을 표시하기 위하여 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 현재의 블록에 대한 팔레트가 현재의 블록의 좌측에 위치된 블록으로부터 복사된다는 것을 표시하기 위하여 0의 값을 가지는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20)는 현재의 블록에 대한 팔레트가 현재의 블록 상부에 위치된 블록으로부터 복사된다는 것을 표시하기 위하여 1의 값을 가지는 플래그를 인코딩할 수도 있다. 비디오 디코더 (30)는 플래그를 디코딩하고, 팔레트 예측을 위하여 적절한 블록을 선택한다. 또 다른 예에서, 플래그는 상부 또는 좌측의 이웃하는 블록의 팔레트가 예측 팔레트로서 이용되는지 여부를 표시할 수도 있다. 그 다음으로, 예측 팔레트에서의 각각의 엔트리에 대하여, 그 엔트리가 현재의 블록에 대한 팔레트에서 이용되는지 여부가 표시될 수도 있다.

[0067] 또 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 하나 이상의 다른 팔레트들 내에 포함된 샘플 값들이 하나 이상의 이웃하는 블록들에서 발생하는 빈도에 기초하여 현재 코딩되고 있는 블록에 대한 팔레트를 결정한다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 미리 결정된 수의 블록들의 코딩 동안에 가장 빈번하게 이용된 인덱스 값들과 연관된 컬러들을 추적할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 현재 코딩되고 있는 블록에 대한 팔레트에서 가장 빈번하게 이용된 컬러들을 포함할 수도 있다.

[0068] 위에서 언급된 바와 같이, 일부 예들에서는, 비디오 인코더 (20) 및/또는 비디오 디코더가 현재의 블록을 코딩하기 위하여 이웃하는 블록으로부터 전체 팔레트를 복사할 수도 있다. 추가적으로 또는 대안적으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 엔트리별 기반의 팔레트 예측을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20)는, 각각의 엔트리들이 예측 팔레트 (예컨대, 또 다른 블록의 팔레트)에 기초하여 예측되는지 여부를 표시하는 팔레트의 개개의 엔트리에 대한 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다.

이 예에서, 비디오 인코더 (20)는, 엔트리가 예측 팔레트로부터의 예측된 값 (예컨대, 이웃하는 블록과 연관된 팔레트의 대응하는 엔트리) 일 때, 소정의 엔트리에 대하여 1의 값을 가지는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20)는 특정한 엔트리가 또 다른 블록의 팔레트로부터 예측되지 않는다는 것을 표시하기 위하여 특정한 엔트리에 대하여 0의 값을 가지는 플래그를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20)는 또한, 비-예측된 팔레트 엔트리의 값을 표시하는 추가적인 데이터를 인코딩할 수도 있다.

[0069] 이 개시물은 현재의 블록에 대한 팔레트를 예측하기 위한 몇몇 대안적인 기법들을 설명한다. 하나의 예에서, 하나 이상의 이전에 코딩된 이웃하는 블록들로부터의 팔레트 엔트리들을 포함하는 예측 팔레트는 엔트리들의 수, N 을 포함한다. 이 경우, 비디오 인코더 (20)는 먼저, 예측 팔레트와 동일한 크기, 즉, 크기 N 을 가지는 2진 벡터 V 를 비디오 디코더 (30)로 송신한다. 2진 벡터에서의 각각의 엔트리는 예측 팔레트에서의 대응하는 엔트리가 재이용되거나 현재의 블록에 대한 팔레트로 복사될 것인지 여부를 표시한다. 예를 들어,  $V(i) = 1$  은, 이웃하는 블록에 대한 예측 팔레트에서의 i 번째 엔트리가 재이용될 것이고, 현재의 블록에

서 상이한 인덱스를 가질 수도 있는, 현재의 블록에 대한 팔레트로 복사될 것이라는 것을 의미한다.

[0070] 게다가, 비디오 인코더 (20) 는, 얼마나 많은 새로운 팔레트 엔트리들이 현재의 블록에 대한 팔레트 내에 포함되는지를 표시하는 수  $M$  을 송신할 수도 있고, 그 다음으로, 새로운 팔레트 엔트리들의 각각에 대한 픽셀 값은 비디오 디코더 (30) 로 송신한다. 이 예에서, 현재의 블록에 대한 팔레트의 최종 크기는  $M + S$  와 동일한 것으로서 유도될 수도 있고, 여기서,  $S$  는 현재의 블록에 대한 팔레트에서 재이용될 수도 있거나 이것으로 복사될 수도 있는 예측 팔레트에서의 엔트리들의 수이다 (즉,  $V(i) = 1$ ). 현재의 블록에 대한 팔레트를 생성하기 위하여, 비디오 디코더 (30) 는 송신된 새로운 팔레트 엔트리들과, 예측 팔레트로부터 재이용되는 복사된 팔레트 엔트리들을 병합할 수도 있다. 일부 경우들에는, 병합이 픽셀 값들에 기초할 수도 있어서, 현재의 블록에 대한 팔레트에서의 엔트리들은 팔레트 인덱스와 함께 증가 (또는 감소) 할 수도 있다. 다른 경우들에는, 병합이 엔트리들의 2 개의 세트들, 즉, 새로운 팔레트 엔트리들 및 복사된 팔레트 엔트리들의 연결일 수도 있다.

[0071] 또 다른 예에서, 비디오 인코더 (20) 는 먼저, 현재의 블록에 대한 팔레트의 크기,  $N$  의 표시를 비디오 디코더 (30) 로 송신한다. 그 다음으로, 비디오 인코더 (20) 는 현재의 블록에 대한 팔레트와 동일한 크기, 즉, 크기  $N$  을 가지는 벡터  $V$  를 비디오 디코더 (30) 로 송신한다. 벡터에서의 각각의 엔트리는 현재의 블록에 대한 팔레트에서의 대응하는 엔트리가 비디오 인코더 (20) 에 의해 명시적으로 송신되는지, 또는 예측 팔레트로부터 복사되는지 여부를 표시한다. 예를 들어,  $V(i) = 1$  은 비디오 인코더 (20) 가 팔레트에서의  $i$  번째 엔트리를 비디오 디코더 (30) 로 송신하는 것을 의미하고,  $V(i) = 0$  은 팔레트에서의  $i$  번째 엔트리가 예측 팔레트로부터 복사되는 것을 의미한다. 예측 팔레트로부터 복사되는 엔트리들에 대하여 (즉,  $V(i) = 0$ ), 비디오 인코더 (20) 는 예측 팔레트에서의 어느 엔트리가 현재의 블록에 대한 팔레트에서 이용되는지를 시그널링하기 위하여 상이한 방법들을 이용할 수도 있다. 일부 경우들에는, 비디오 인코더 (20) 가 예측 팔레트로부터 현재의 블록에 대한 팔레트로 복사되어야 할 엔트리의 팔레트 인덱스를 시그널링할 수도 있다. 다른 경우들에는, 비디오 인코더 (20) 가 현재의 블록에 대한 팔레트에서의 인덱스 및 예측 팔레트에서의 인덱스 사이의 차이인 인덱스 오프셋을 시그널링할 수도 있다.

[0072] 상기 2 개의 예들에서, 현재의 블록에 대한 팔레트의 예측을 위하여 이용된 예측 팔레트를 생성하기 위하여 이용된 하나 이상의 이전에 코딩된 이웃하는 블록들은 현재의 블록에 대하여 상부-이웃하는 (즉, 상부) 블록 또는 좌측-이웃하는 블록일 수도 있다. 일부 예들에서, 이웃하는 블록들의 후보 리스트가 구성될 수도 있고, 비디오 인코더 (20) 는 어느 후보 이웃하는 블록들 및 연관된 팔레트들이 현재의 블록에 대한 팔레트 예측을 위하여 이용되는지를 표시하기 위하여 인덱스를 송신한다. 어떤 블록들, 예컨대, 슬라이스의 시작부 또는 다른 슬라이스 경계들에서 위치되는 블록들, 또는 비디오 데이터의 슬라이스 또는 픽처에서의 가장 좌측 블록들에 대하여, 팔레트 예측은 디스에이블될 수도 있다.

[0073] 추가적인 예에서, 비디오 인코더 (20) 는 현재의 블록에 대한 팔레트 내에 포함된 엔트리들의 수의 표시를 비디오 디코더 (30) 로 송신한다. 그 다음으로, 팔레트 엔트리들의 각각에 대하여, 비디오 인코더 (20) 는 팔레트 엔트리가 비디오 인코더 (20) 에 의해 명시적으로 송신되는지 여부, 또는 그것이 이전에 복원된 픽셀로부터 유도되는지 여부를 표시하기 위하여, 플래그 또는 다른 십렉스 엘리먼트를 송신한다. 예를 들어, 1 과 동일하게 설정된 1-비트 플래그는 비디오 인코더 (20) 가 팔레트 엔트리를 명시적으로 전송하는 것을 의미할 수도 있고, 0 과 동일하게 설정된 1-비트 플래그는 팔레트 엔트리가 이전에 복원된 픽셀로부터 유도되는 것을 의미할 수도 있다. 이전에 복원된 픽셀로부터 유도되는 팔레트 엔트리들의 각각에 대하여, 비디오 인코더 (20) 는 팔레트 엔트리에 대응하는 현재의 블록 또는 이웃하는 블록에서의 복원된 픽셀의 픽셀 로케이션에 관한 또 다른 표시를 송신한다. 일부 경우들에는, 복원된 픽셀 로케이션 표시가 현재의 블록의 상부-좌측 위치에 대한 벡터 (displacement vector) 일 수도 있다. 다른 경우들에는, 복원된 픽셀 로케이션 표시가 현재의 블록에 대한 팔레트 엔트리를 특정하기 위해 이용될 수 있는 복원된 픽셀들의 리스트로의 인덱스일 수도 있다. 예를 들어, 이 리스트는 HEVC 에서의 정상적인 인트라 예측을 위해 이용될 수도 있는 모든 참조 픽셀들을 포함할 수도 있다.

[0074] 도 2 의 예에서, 제 2 팔레트들 (192) 은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 엔트리 인덱스 값 3, 및 엔트리 인덱스 4 를 각각 가지는 4 개의 엔트리들 (208 내지 214) 을 포함한다. 엔트리들 (208 내지 214) 은 인덱스 값들을, 픽셀 값 A, 픽셀 값 B, 픽셀 값 C, 및 픽셀 값 D 를 각각 포함하는 픽셀 값들에 관련시킨다. 이 개시물의 하나 이상의 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 예측의 목적들을 위하여 제 1 블록 (180) 을 위치시키고 제 2 블록 (188) 을 코딩하기 위하여 제 1 팔레트들 (184) 의 엔트리들 1 내지 3 을 제 2 팔레트들 (192) 의 엔트리들 1 내지 3 으로 복사하기 위하여 상기 설명된 기법들

중의 임의의 것을 이용할 수도 있다. 이러한 방법으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 제 1 팔레트들 (184)에 기초하여 제 2 팔레트들 (192)을 결정할 수도 있다. 게다가, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 제 2 팔레트들 (192)과 함께 포함되어야 할 엔트리 4에 대한 데이터를 코딩할 수도 있다. 이러한 정보는 예측 팔레트로부터 예측되지 않은 팔레트 엔트리들의 수 및 그 팔레트 엔트리들에 대응하는 픽셀 값들을 포함할 수도 있다.

[0075] 일부 예들에서, 이 개시물의 양태들에 따르면, 하나 이상의 신팩스 엘리먼트들은 제 2 팔레트들 (192)과 같은 팔레트들이 (도 2에서 제 1 팔레트들 (184)로서 도시되지만, 하나 이상의 블록들로부터의 엔트리들로 구성될 수도 있는) 예측 팔레트로부터 전체적으로 예측되는지 여부 또는 제 2 팔레트들 (192)의 특정한 엔트리들이 예측되는지 여부를 표시할 수도 있다. 예를 들어, 초기 신팩스 엘리먼트는 엔트리들의 전부가 예측되는지 여부를 표시할 수도 있다. 초기 신팩스 엘리먼트가 엔트리들의 전부가 예측되는지 않는다는 것을 표시할 경우 (예컨대, 0의 값을 가지는 플래그), 하나 이상의 추가적인 신팩스 엘리먼트들은 제 2 팔레트들 (192)의 어느 엔트리들이 예측 팔레트로부터 예측되는지를 표시할 수도 있다.

[0076] 이 개시물의 일부 양태들에 따르면, 팔레트 예측과 연관된 어떤 정보는 코딩되고 있는 데이터의 하나 이상의 특성들로부터 추론될 수도 있다. 즉, 비디오 인코더 (20)가 신팩스 엘리먼트들을 인코딩하는 것 (그리고 비디오 디코더 (30)가 이러한 신팩스 엘리먼트들을 디코딩하는 것)이 아니라, 비디오 인코더 (20) 및 비디오 디코더 (30)는 코딩되고 있는 데이터의 하나 이상의 특성들에 기초하여 팔레트 예측을 수행할 수도 있다.

[0077] 예에서, 예시의 목적들을 위하여, 위에서 설명된 pred\_palette\_flag의 값은 예들로서, 코딩되고 있는 블록의 크기, 프레임 타입, 컬러 공간, 컬러 컴포넌트, 프레임 크기, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 중의 하나 이상으로부터 추론될 수도 있다. 즉, 예로서의 블록의 크기에 대하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 위에서 설명된 pred\_palette\_flag가 미리 결정된 크기를 초과하거나 미리 결정된 크기보다 더 작은 임의의 블록들에 대하여 1과 동일한 것으로 결정할 수도 있다. 이 예에서, pred\_palette\_flag는 인코딩된 비트스트림에서 시그널링될 필요가 없다.

[0078] pred\_palette\_flag에 대하여 위에서 설명되었지만, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 또한, 또는 대안적으로, 코딩되고 있는 데이터의 하나 이상의 특성들에 기초하여, 팔레트가 예측을 위해 이용되는 후보 블록과 같은, 팔레트 예측과 연관된 다른 정보, 또는 팔레트 예측 후보들을 구성하기 위한 규칙들을 추론할 수도 있다.

[0079] 이 개시물의 다른 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팔레트를 온-더-플라이 (on-the-fly)로 구성할 수도 있다. 예를 들어, 초기에 제 2 블록 (188)을 코딩할 때, 제 2 팔레트들 (192) 내에는 어떤 엔트리들도 없다. 비디오 인코더 (20) 및 비디오 디코더 (30)가 제 2 블록 (188)의 픽셀들에 대한 새로운 값을 코딩할 때, 각각의 새로운 값은 제 2 팔레트들 (192) 내에 포함된다. 즉, 예를 들어, 비디오 인코더 (20)는 픽셀 값을 생성되고 블록 (188)에서의 위치들에 대해 시그널링될 때에 픽셀 값을 제 2 팔레트들 (192)에 추가한다. 비디오 인코더 (20)가 블록에서 상대적으로 더 이후에 픽셀들을 인코딩할 때, 비디오 인코더 (20)는 픽셀 값을 시그널링하는 것이 아니라, 인덱스 값을 이용하여 팔레트 내에 이미 포함된 것들과 동일한 값을 가지는 픽셀들을 인코딩할 수도 있다. 유사하게, 비디오 디코더 (30)가 제 2 블록 (188)에서의 위치에 대한 (예컨대, 비디오 인코더 (20)에 의해 시그널링된) 새로운 픽셀 값을 수신할 때, 비디오 디코더 (30)는 픽셀 값을 제 2 팔레트들 (192) 내에 포함한다. 제 2 블록 (188)에서 상대적으로 더 이후에 디코딩된 픽셀 위치들이 제 2 팔레트들 (192)에 추가되었던 픽셀 값을 가질 때, 비디오 디코더 (30)는 제 2 블록 (188)의 픽셀 값을 복원을 위하여 제 2 팔레트들 (192)에서 대응하는 픽셀 값을 삭제하는, 예컨대, 인덱스 값을 같은 정보를 수신할 수도 있다.

[0080] 일부 예들에서, 이하에서 더 상세하게 설명된 바와 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 제 1 팔레트들 (184) 및 제 2 팔레트들 (192)을 최대 팔레트 크기에서 또는 그 미만으로 유지할 수도 있다.

이 개시물의 양태들에 따르면, 최대 팔레트 크기가 도달될 경우, 예컨대, 제 2 팔레트들 (192)이 온-더-플라이로 동적으로 구성될 때, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 제 2 팔레트들 (192)의 엔트리를 제거하기 위하여 동일한 프로세스를 수행한다. 팔레트 엔트리들을 제거하기 위한 하나의 예의 프로세스는, 비디오 인코더 (20) 및 비디오 디코더 (30)가 팔레트의 가장 오래된 엔트리를 제거하는 선입선출 (first-in-first-out; FIFO) 기법이다. 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트로부터 최소로 빈번하게 이용된 팔레트 엔트리를 제거할 수도 있다. 또 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 어느 엔트리를 제거할 것인지를 결정하기 위하여 FIFO 및 이용 빈도 프로세스들의 양자를 가

중화할 수도 있다. 즉, 엔트리의 제거는 엔트리가 얼마나 오래된 것인지와, 그것이 얼마나 빈번하게 이용되는지에 기초할 수도 있다.

[0081] 일부 양태들에 따르면, 엔트리 (픽셀 값) 가 팔레트로부터 제거되고 픽셀 값이 코딩되고 있는 블록의 더 이후의 위치에서 다시 발생할 경우, 비디오 인코더 (20) 는 팔레트 내에 엔트리를 포함하고 인덱스를 인코딩하는 대신에, 픽셀 값을 인코딩할 수도 있다. 추가적으로 또는 대안적으로, 비디오 인코더 (20) 는 예컨대, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 블록에서의 위치들을 스캔할 때, 제거된 이후의 팔레트 내로 팔레트 엔트리들을 재입력할 수도 있다.

[0082] 일부 예들에서, 팔레트를 온-더-플라이로 유도하기 위한 기법들은 팔레트를 결정하기 위한 하나 이상의 다른 기법들과 조합될 수도 있다. 특히, 예로서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 (예컨대, 제 1 팔레트들 (184) 로부터 제 2 팔레트들 (192) 을 예측하기 위한 팔레트 예측을 이용하여) 제 2 팔레트들 (192) 을 초기에 코딩할 수도 있고, 제 2 블록 (188) 의 픽셀들을 코딩할 때에 제 2 팔레트들 (192) 을 업데이트할 수도 있다. 예를 들어, 초기 팔레트를 송신할 시에, 비디오 인코더 (20) 는 블록에서의 추가적인 로케이션들의 픽셀 값들이 스캔될 때, 값들을 초기 팔레트에 추가할 수도 있거나, 초기 팔레트에서의 값들을 변경시킬 수도 있다. 마찬가지로, 초기 팔레트를 수신할 시에, 비디오 디코더 (30) 는 블록에서의 추가적인 로케이션들의 픽셀 값들이 스캔될 때, 값들을 초기 팔레트에 추가 (즉, 포함) 할 수도 있거나 초기 팔레트에서의 값들을 변경시킬 수도 있다.

[0083] 비디오 인코더 (20) 는, 일부 예들에서, 현재의 블록이 전체 팔레트의 송신, 또는 온-더-플라이 팔레트 생성, 또는 초기 팔레트의 송신과, 온-더-플라이 유도에 의한 초기 팔레트의 업데이트와의 조합을 이용하는지 여부를 시그널링할 수도 있다. 일부 예들에서, 초기 팔레트는 최대 팔레트 크기인 전체 팔레트일 수도 있고, 이 경우, 초기 팔레트에서의 값들이 변경될 수도 있다. 다른 예들에서, 초기 팔레트는 최대 팔레트 크기보다 더 작을 수도 있고, 이 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 값들을 초기 팔레트에 추가할 수도 있고 및/또는 초기 팔레트의 값들을 변경시킬 수도 있다.

[0084] 이 개시물의 하나 이상의 양태들에 따르면, 예컨대, 팔레트 내에 포함된 픽셀 값들의 수의 측면에서의, 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 과 같은 팔레트들의 크기는 고정될 수도 있거나, 인코딩된 비트스트림에서의 하나 이상의 신팩스 엘리먼트들을 이용하여 시그널링될 수도 있다. 예를 들어, 일부 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 크기를 코딩하기 위하여 1진 코드들 또는 절단된 1진 코드들 (예컨대, 팔레트 크기의 최대 제한에서 절단하는 코드들) 을 이용할 수도 있다. 다른 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 크기를 코딩하기 위하여 지수-골롬 (Exponential-Golomb) 또는 라이스-골롬 (Rice-Golomb) 코드들을 이용할 수도 있다.

[0085] 또 다른 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트의 각각의 엔트리 이후에 팔레트의 크기를 표시하는 데이터를 코딩할 수도 있다. 예로서의 제 2 팔레트들 (192) 에 대하여, 비디오 인코더 (20) 는 엔트리들 (208 내지 214) 의 각각 이후에 정지 플래그 (stop flag) 를 인코딩할 수도 있다. 이 예에서, 1 과 동일한 정지 플래그는, 현재 코딩되고 있는 엔트리가 제 2 팔레트들 (192) 의 최종 엔트리인 것으로 특정할 수도 있는 반면, 제로와 동일한 정지 플래그는 제 2 팔레트들 (192) 내에 추가적인 엔트리들이 있다는 것을 표시할 수도 있다. 따라서, 비디오 인코더 (20) 는 엔트리들 (208 내지 212) 의 각각 이후에 제로의 값을 가지는 정지 플래그들과, 엔트리 (214) 이후에 1 의 값을 가지는 정지 플래그를 인코딩할 수도 있다.

일부 사례들에서, 정지 플래그는 구성된 팔레트가 최대 팔레트 크기 제한에 도달할 시에 비트스트림 내에 포함되지 않을 수도 있다. 상기 예들이 팔레트들의 크기를 명시적으로 시그널링하기 위한 기법들을 개시하지만, 다른 예들에서는, 팔레트들의 크기가 또한, 조건적으로 송신될 수도 있거나, 소위 부가 정보 (예컨대, 위에서 언급된 바와 같이, 코딩되고 있는 블록의 크기, 프레임 타입, 컬러 공간, 컬러 컴포넌트, 프레임 크기, 프레임 레이트, 스케일러를 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 와 같은 특성 정보) 에 기초하여 추론될 수도 있다.

[0086] 이 개시물의 기법들은 손실없이, 또는 대안적으로, 약간의 손실들로 (손실있는 코딩) 데이터를 코딩하는 것을 포함한다. 예를 들어, 손실있는 코딩에 대하여, 비디오 인코더 (20) 는 팔레트들의 픽셀 값들을 블록에서의 실제적인 픽셀 값들과 정확하게 일치시키지 않으면서 블록의 픽셀들을 코딩할 수도 있다. 이 개시물의 기법들이 손실있는 코딩에 적용될 때, 일부 한정들이 팔레트에 적용될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 과 같은 팔레트들을 양자화할 수도 있다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 엔트리들의 픽셀 값들이 서로의 미리 결정된

범위 내에 있을 때에 팔레트의 엔트리들을 병합 또는 조합 (즉, 양자화) 할 수도 있다. 다시 말해서, 새로운 팔레트 값의 여러 마진 내에 있는 팔레트 값이 이미 있을 경우에는, 새로운 팔레트 값이 팔레트에 추가되지 않는다. 다른 예에서, 블록에서의 복수의 상이한 픽셀 값들이 단일 팔레트 엔트리로, 또는 등가적으로, 단일 팔레트 픽셀 값으로 맵핑될 수도 있다.

[0087] 비디오 디코더 (30)는 특정한 팔레트가 손실없는지 또는 손실있는지 여부에 관계없이, 동일한 방식으로 픽셀 값들을 디코딩할 수도 있다. 하나의 예로서, 비디오 디코더 (30)는 팔레트가 손실없는지 또는 손실있는지 여부에 관계없이, 픽셀 위치에 대한 팔레트에서의 엔트리를 선택하기 위하여 코딩된 블록에서의 소정의 픽셀 위치에 대하여 비디오 인코더 (20)에 의해 송신된 인덱스 값을 이용할 수도 있다. 이 예에서, 팔레트 엔트리의 픽셀 값은 원래의 픽셀 값과 정확하게 일치하는지 여부에 관계 없이, 코딩된 블록에서의 픽셀 값으로서 이용된다.

[0088] 손실있는 코딩의 예에서, 예시의 목적들을 위하여, 비디오 인코더 (20)는 델타 값 (delta value)으로서 지정된 에러 경계를 결정할 수도 있다. 후보 픽셀 값 엔트리 Plt\_cand는 CU 또는 PU와 같은, 코딩되어야 할 블록의 위치에서의 픽셀 값에 대응할 수도 있다. 팔레트의 구성 동안에, 비디오 인코더 (20)는 후보 픽셀 값 엔트리 Plt\_cand와, 팔레트에서의 기존의 픽셀 값 엔트리들의 전부와의 사이의 절대 차이를 결정한다. 후보 픽셀 값 엔트리 Plt\_cand와, 팔레트에서의 기존의 픽셀 값 엔트리들과의 사이의 절대 차이들의 전부가 델타 값보다 더 클 경우, 비디오 인코더 (20)는 픽셀 값 후보를 팔레트에 엔트리로서 추가할 수도 있다. 픽셀 값 엔트리 Plt\_cand와, 팔레트에서의 적어도 하나의 기존의 픽셀 값 엔트리와의 사이의 절대 차이가 델타 값과 동일하거나 그보다 더 작을 경우, 비디오 인코더 (20)는 후보 픽셀 값 엔트리 Plt\_cand를 팔레트에 추가하지 않을 수도 있다. 이에 따라, 픽셀 값 엔트리 Plt\_cand를 코딩할 때, 비디오 인코더 (20)는 픽셀 값 엔트리 Plt\_cand에 가장 근접한 픽셀 값을 갖는 엔트리를 선택할 수도 있음으로써, 약간의 손실을 시스템 내로 도입할 수도 있다. 팔레트가 다수의 컴포넌트들 (예컨대, 3 개의 컬러 컴포넌트들)로 구성될 때, 개별적인 컴포넌트 값들의 절대 차이의 합은 델타 값에 대한 비교를 위하여 이용될 수도 있다. 대안적으로 또는 추가적으로, 각각의 컴포넌트 값에 대한 절대 차이는 제 2 델타 값에 대하여 비교될 수도 있다.

[0089] 일부 예들에서, 위에서 언급된 팔레트에서의 기존의 픽셀 값 엔트리들은 유사한 델타 비교 프로세스를 이용하여 추가되었을 수도 있다. 다른 예들에서, 팔레트에서의 기존의 픽셀 값들은 다른 프로세스들을 이용하여 추가되었을 수도 있다. 예를 들어, 하나 이상의 초기 픽셀 값 엔트리들은 팔레트를 구성하는 델타 비교 프로세스를 시작하기 위하여 (델타 비교 없이) 팔레트에 추가될 수도 있다. 위에서 설명된 프로세스는 루마 및/또는 크로마 팔레트들을 생성하기 위하여 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 구현될 수도 있다.

[0090] 팔레트 구성에 대하여 위에서 설명된 기법들은 또한, 픽셀 코딩 동안에 비디오 인코더 (20) 및 비디오 디코더 (30)에 의해 이용될 수도 있다. 예를 들어, 픽셀 값의 인코딩할 때, 비디오 인코더 (20)는 픽셀의 값을 팔레트에서의 엔트리들의 픽셀 값들과 비교할 수도 있다. 픽셀의 값과 팔레트에서의 엔트리들 중의 하나와의 사이의 절대 픽셀 값 차이가 델타 값과 동일하거나 그보다 더 작을 경우, 비디오 인코더 (20)는 픽셀 값을 팔레트의 엔트리로서 인코딩할 수도 있다. 즉, 이 예에서, 비디오 인코더 (20)는 픽셀 값이 충분히 작은 (예컨대, 미리 결정된 범위 내의) 절대 차이 대 팔레트 엔트리를 생성할 때에 팔레트의 엔트리들 중의 하나를 이용하여 픽셀 값을 인코딩한다.

[0091] 일부 예들에서, 비디오 인코더 (20)는 픽셀 값을 인코딩하기 위하여 (코딩되고 있는 픽셀 값과 비교하여) 최소 절대 픽셀 값 차이를 산출하는 팔레트 엔트리를 선택할 수도 있다. 예로서, 비디오 인코더 (20)는 픽셀 값에 대하여 이용될 팔레트 엔트리, 예컨대, 비디오 디코더 (30)에서 코딩된 픽셀 값을 복원하기 위하여 이용될 팔레트 픽셀 값 엔트리를 표시하기 위하여 인덱스를 인코딩할 수도 있다. 픽셀의 값과, 팔레트에서의 엔트리들의 전부와의 사이의 절대 픽셀 값 차이가 델타보다 더 클 경우, 인코더는 픽셀 값을 인코딩하기 위하여 팔레트 엔트리들 중의 하나를 이용하지 않을 수도 있고, 그 대신에, (아마도 양자화 후의) 픽셀의 픽셀 값을 비디오 디코더 (30)로 송신할 수도 있다 (그리고 아마도 픽셀 값을 팔레트로의 엔트리로서 추가함).

[0092] 다른 예에서, 비디오 인코더 (20)는 픽셀 값을 인코딩하기 위한 팔레트의 엔트리를 선택할 수도 있다. 비디오 인코더 (20)는 선택된 엔트리를 예측 픽셀 값으로서 이용할 수도 있다. 즉, 비디오 인코더 (20)는 실제적인 픽셀 값과, 선택된 엔트리와의 사이의 차이를 표현하는 잔차 값을 결정할 수도 있고, 잔차를 인코딩할 수도 있다. 비디오 인코더 (20)는 팔레트의 엔트리들에 의해 예측되는 블록에서의 픽셀들에 대한 잔차 값을 생성할 수도 있고, 픽셀들의 블록에 대한 개개의 잔차 픽셀 값을 포함하는 잔차 블록을 생성할 수도 있

다. 비디오 인코더 (20)는 이후에 (도 2에 대하여 위에서 언급된 바와 같은) 변환 및 양자화를 잔차 블록에 적용할 수도 있다. 이러한 방식으로, 비디오 인코더 (20)는 양자화된 잔차 변환 계수들을 생성할 수도 있다. 또 다른 예에서, 잔차는 손실없이 (변환 및 양자화 없이) 또는 변환 없이 코딩될 수도 있다.

[0093] 비디오 디코더 (30)는 잔차 블록을 재생하기 위해 변환 계수들을 역변환 및 역양자화할 수도 있다. 그 다음으로, 비디오 디코더 (30)는 픽셀 값에 대한 예측 팔레트 엔트리 값 및 잔차 값을 이용하여 픽셀 값을 복원할 수도 있다. 예를 들어, 비디오 디코더 (30)는 코딩된 픽셀 값을 복원하기 위하여 잔차 값을 팔레트 엔트리 값과 조합할 수도 있다.

[0094] 일부 예들에서, 렐타 값은 상이한 CU 크기들, 픽처 크기들, 컬러 공간들 또는 상이한 컬러 컴포넌트들에 대하여 상이할 수도 있다. 렐타 값은 다양한 코딩 조건들에 기초하여 미리 결정되거나 결정될 수도 있다. 예를 들어, 비디오 인코더 (20)는 PPS, SPS, VPS 및/또는 슬라이스 헤더에서의 신택스와 같은 하이 레벨 신택스를 이용하여 렐타 값을 비디오 디코더 (30)로 시그널링할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 동일한 고정된 렐타 값을 이용하도록 미리 구성될 수도 있다. 또 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 (예컨대, 위에서 언급된 바와 같은 블록 크기, 컬러 공간, 컬러 컴포넌트 등과 같은) 부가 정보에 기초하여 렐타 값을 적응적으로 유도할 수도 있다.

[0095] 일부 예들에서, 손실있는 코딩 팔레트 모드는 HEVC 코딩 모드로서 포함될 수도 있다. 예를 들어, 코딩 모드들은 인트라-예측 모드, 인터-예측 모드, 손실없는 코딩 팔레트 모드, 및 손실있는 코딩 팔레트 모드를 포함할 수도 있다. HEVC 코딩에서, 도 2 및 도 3에 대하여 위에서 언급된 바와 같이, 양자화 파라미터 (QP)는 허용된 왜곡을 제어하기 위하여 이용된다. 팔레트-기반 코딩에 대한 렐타의 값은 계산될 수도 있거나, 이와 다르게, QP의 함수로서 결정될 수도 있다. 예를 들어, 상기 설명된 렐타 값은  $1 \ll (QP/6)$  또는  $1 \ll ((QP+d)/6)$  일 수도 있으며, 여기서, d는 상수이고 " $\ll$ "는 비트별 좌측-시프트 연산자 (bitwise left-shift operator)를 표현한다.

[0096] 이 개시물에서 설명된 손실있는 코딩 기법들을 이용한 팔레트의 생성은 비디오 인코더 (20), 비디오 디코더 (30), 또는 양자에 의해 수행될 수도 있다. 예를 들어, 비디오 인코더 (20)는 위에서 설명된 렐타 비교 기법들을 이용하여 블록에 대한 팔레트에서의 엔트리들을 생성할 수도 있고, 비디오 디코더 (30)에 의한 이용을 위하여 팔레트의 구성에 대한 정보를 시그널링할 수도 있다. 즉, 비디오 인코더 (20)는 블록에 대한 팔레트에서의 엔트리들에 대한 픽셀 값들을 표시하는 정보를 시그널링하고, 그 다음으로, 이러한 팔레트 엔트리들과 연관된 픽셀 값들을 이용하여 픽셀 값들을 인코딩하도록 구성될 수도 있다. 비디오 디코더 (30)는 이러한 정보를 이용하여 팔레트를 구성할 수도 있고, 그 다음으로, 코딩된 블록의 픽셀 값들을 디코딩하기 위하여 엔트리들을 이용할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 코딩된 블록의 하나 이상의 픽셀 위치들에 대한 팔레트 엔트리들을 식별하는 인덱스 값들을 시그널링할 수도 있고, 비디오 디코더 (30)는 팔레트로부터 적절한 픽셀 값 엔트리들을 취출 (retrieve) 하기 위하여 인덱스 값들을 이용할 수도 있다.

[0097] 다른 예들에서, 비디오 디코더 (30)는 위에서 설명된 렐타 비교 기법들을 적용함으로써 팔레트를 구성하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30)는 코딩된 블록 내의 위치들에 대한 픽셀 값들을 수신할 수도 있고, 픽셀 값들과 팔레트에서의 기존의 픽셀 값 엔트리들과의 사이의 절대 차이들이 렐타 값보다 더 큰지 여부를 결정할 수도 있다. 그러한 경우, 비디오 디코더 (30)는, 예컨대, 비디오 인코더 (20)에 의해 시그널링된 대응하는 인덱스 값들을 이용한 블록의 다른 픽셀 위치들에 대한 픽셀 값들의 팔레트-기반 디코딩에서의 더 이후의 이용을 위하여, 픽셀 값들을 팔레트에서의 엔트리들로서 추가할 수도 있다. 이 경우, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트를 생성하기 위하여 동일하거나 유사한 프로세스들을 적용한다. 그렇지 않을 경우, 비디오 디코더 (30)는 픽셀 값들을 팔레트에 추가하지 않을 수도 있다.

[0098] 예시의 목적들을 위한 예에서, 비디오 디코더 (30)는 블록에서의 다양한 픽셀 위치들에 대한 인덱스 값들 또는 픽셀 값들을 수신할 수도 있다. 인덱스 값이 픽셀 위치에 대해 수신될 경우, 예를 들어, 비디오 디코더 (30)는 팔레트에서 엔트리를 식별하기 위하여 인덱스 값을 이용할 수도 있고, 픽셀 위치에 대한 팔레트 엔트리의 픽셀 값을 이용할 수도 있다. 픽셀 값이 픽셀 위치에 대해 수신될 경우, 비디오 디코더 (30)는 픽셀 위치에 대한 수신된 픽셀 값을 이용할 수도 있고, 또한, 픽셀 값이 팔레트에 추가되어야 하고 그 다음으로, 팔레트 코딩을 위하여 더 이후에 이용되어야 하는지 여부를 결정하기 위하여, 렐타 비교를 적용할 수도 있다.

[0099] 인코더 측에서는, 블록에서의 위치에 대한 픽셀 값이 픽셀 값과, 렐타 값이 하인 팔레트에서의 기존의 픽셀 값 엔트리와의 사이의 절대 차이를 생성할 경우, 비디오 인코더 (20)는 그 위치에 대한 픽셀 값을 복원함에 있어서 이용하기 위하여 팔레트에서 엔트리를 식별하기 위한 인덱스 값을 전송할 수도 있다. 블록에서의 위치에

대한 픽셀 값이 픽셀 값과, 레타 값보다 모두 더 큰 팔레트에서의 기존의 픽셀 값 엔트리들과의 사이의 절대 차이 값들을 생성할 경우, 비디오 인코더 (20)는 픽셀 값을 전송할 수도 있고, 픽셀 값을 팔레트에서의 새로운 엔트리로서 추가할 수도 있다. 팔레트를 구성하기 위하여, 비디오 디코더 (30)는 예컨대, 위에서 설명된 바와 같이, 인코더에 의해 시그널링된 레타 값들을 이용할 수도 있거나, 고정된 또는 알려진 레타 값에 의존할 수도 있거나, 레타 값을 추론 또는 유도할 수도 있다.

[0100] 위에서 언급된 바와 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 비디오 데이터를 코딩할 때, 인트라-예측 모드, 인터-예측 모드, 손실없는 코딩 팔레트 모드, 및 손실있는 코딩 팔레트 모드를 포함하는 코딩 모드들을 이용할 수도 있다. 이 개시물의 일부 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트-기반 코딩이 인에이블되는지 여부를 표시하는 하나 이상의 선택스 엘리먼트들을 코딩할 수도 있다. 예를 들어, 각각의 블록에서, 비디오 인코더 (20)는 플래그 PLT\_Mode\_flag 와 같은 선택스 엘리먼트를 인코딩할 수도 있다. PLT\_Mode\_flag 또는 다른 선택스 엘리먼트는 팔레트-기반 코딩 모드가 소정의 블록 (예컨대, CU 또는 PU)에 대하여 이용되어야 하는지 여부를 표시할 수도 있다. 예를 들어, 이 플래그는 블록 레벨 (예컨대, CU 레벨)에서 인코딩된 비디오 비트스트림에서 시그널링될 수도 있고, 그 다음으로, 인코딩된 비디오 비트스트림을 디코딩할 시에 비디오 디코더 (30)에 의해 수신될 수도 있다.

[0101] 이 예에서, 1 과 동일한 이 PLT\_Mode\_flag 의 값은 현재의 블록이 팔레트-기반 코딩 모드를 이용하여 인코딩된다는 것을 특정할 수도 있다. 이 경우, 비디오 디코더 (30)는 블록을 디코딩하기 위하여 팔레트-기반 코딩 모드를 적용할 수도 있다. 일부 예들에서, 선택스 엘리먼트는 블록에 대한 복수의 상이한 팔레트-기반 코딩 모드들 중의 하나 (예컨대, 손실있음 또는 손실없음)를 표시할 수도 있다. 0 과 동일한 이 PLT\_Mode\_flag 의 값은 현재의 블록이 팔레트 모드 이외의 모드를 이용하여 인코딩된다는 것을 특정할 수도 있다. 예를 들어, 다양한 인터-예측, 인트라-예측, 또는 다른 코딩 모드들 중의 임의의 것이 이용될 수도 있다. PLT\_Mode\_flag 의 값이 0 일 때, 비디오 인코더 (20)는 또한, 개개의 블록을 인코딩하기 위하여 이용된 특정 모드 (예컨대, HEVC 코딩 모드)를 표시하기 위하여 추가적인 데이터를 인코딩할 수도 있다. PLT\_Mode\_flag 의 이용은 예의 목적들을 위하여 설명된다. 그러나, 다른 예들에서, 멀티-비트 코드들과 같은 다른 선택스 엘리먼트들은 팔레트-기반 코딩 모드가 블록 (예컨대, CU 또는 PU)에 대하여 이용되어야 하는지 여부를 표시하거나, 또는 복수의 모드들 중의 어느 것이 코딩을 위하여 이용되어야 하는지를 표시하기 위하여 이용될 수도 있다.

[0102] 일부 예들에서, 상기 설명된 플래그 또는 다른 선택스 엘리먼트들은 블록 레벨보다 더 높은 레벨에서 송신될 수도 있다. 예를 들어, 비디오 인코더 (20)는 슬라이스 레벨에서 이러한 플래그를 시그널링할 수도 있다. 이 경우, 1 과 동일한 값은 슬라이스에서의 블록들의 전부가 팔레트 모드를 이용하여 인코딩된다는 것을 표시한다. 이 예에서, 예컨대, 팔레트 모드 또는 다른 모드들에 대한 추가적인 모드 정보는 블록 레벨에서 시그널링되지 않는다. 또 다른 예에서, 비디오 인코더 (20)는 PPS, SPS, 또는 VPS 에서의 이러한 플래그를 시그널링할 수도 있다.

[0103] 이 개시물의 일부 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팔레트 모드가 특정한 슬라이스, 픽처, 시퀀스 등에 대하여 인에이블 또는 디스에이블되는지 여부를 특정하는 (예컨대, 상기 설명된 플래그와 같은) 하나 이상의 선택스 엘리먼트들을 슬라이스, PPS, SPS, 또는 VPS 레벨들 중의 하나에서 코딩할 수도 있는 반면, PLT\_Mode\_flag 는 팔레트-기반 코딩 모드가 각각의 블록에 대하여 이용되는지 여부를 표시한다. 이 경우, 슬라이스, PPS, SPS 또는 VPS 레벨에서 전송된 플래그 또는 다른 선택스 엘리먼트가 팔레트 코딩 모드가 디스에이블된다는 것을 표시할 경우, 일부 예들에서는, 각각의 블록에 대하여 PLT\_Mode\_flag 를 시그널링하기 위한 필요성이 없을 수도 있다. 대안적으로, 슬라이스, PPS, SPS, 또는 VPS 레벨에서 전송된 플래그 또는 다른 선택스 엘리먼트가 팔레트 코딩 모드가 인에이블된다는 것을 표시할 경우, PLT\_Mode\_flag 는 팔레트-기반 코딩 모드가 각각의 블록에 대하여 이용되어야 하는지 여부를 표시하기 위하여 추가로 시그널링될 수도 있다.

[0104] 일부 예들에서, 상기 설명된 선택스 엘리먼트들은 비트스트림에서 조건적으로 시그널링될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 블록의 크기, 프레임 타입, 컬러 공간, 컬러 컴포넌트, 프레임 크기, 프레임 레이트, 스케일러를 비디오 코딩에서의 계층 id, 및/또는 멀티-뷰 코딩에서의 뷰 id 에 기초하여 선택스 엘리먼트들을 각각 인코딩하거나 디코딩하기만 할 수도 있다.

[0105] 위에서 설명된 예들은 예컨대, 비트스트림에서의 하나 이상의 선택스 엘리먼트들에 의한 명시적 시그널링에 관한 것이지만, 다른 예들에서는, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 팔레트 코딩 모드가 활성화되는지 여부를 표시하기 위하여 추가로 시그널링될 수도 있다.

고 및/또는 특정한 블록을 코딩하기 위하여 이용되는지 여부를 묵시적으로 결정할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예를 들어, 블록의 크기, 프레임 타입, 컬러 공간, 컬러 컴포넌트, 프레임 크기, 프레임 레이트, 스케일러를 비디오 코딩에서의 계층 id, 및/또는 멀티-뷰 코딩에서의 뷰 id 에 기초하여 팔레트-기반 코딩이 블록에 대하여 이용되는지 여부를 결정할 수도 있다.

[0106] 도 3 은 이 개시물의 기법들과 부합하는, 비디오 블록에 대한 팔레트로의 인덱스들을 결정하는 예들을 예시하는 개념도이다. 예를 들어, 도 3 은 인덱스 값들과 연관된 픽셀들의 개개의 위치들을 팔레트들 (244) 의 엔트리에 관련시키는 인덱스 값들 (값들 1, 2, 및 3) 의 맵 (240) 을 포함한다. 팔레트들 (244) 은 도 2 에 대하여 위에서 설명된 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 과 유사한 방식으로 결정될 수도 있다.

[0107] 다시, 도 3 의 기법들은 비디오 인코더 (20) (도 1 및 도 4) 및 비디오 디코더 (30) (도 1 및 도 5) 의 문맥에서, 그리고 설명의 목적들을 위한 HEVC 비디오 코딩 표준에 대하여 설명된다. 그러나, 이 개시물의 기법들은 이러한 방법으로 제한되는 것은 아니고, 다른 비디오 코딩 프로세서들 및/또는 다른 비디오 코딩 프로세스들 및/또는 표준들에서의 디바이스들에 의해 적용될 수도 있다는 것이 이해되어야 한다.

[0108] 맵 (240) 은 각각의 픽셀 위치에 대한 인덱스 값을 포함하는 것으로서 도 3 의 예에서 예시되지만, 다른 예들에서는, 모든 픽셀 위치들이 블록의 픽셀 값을 특정하는 팔레트들 (244) 의 엔트리를 표시하는 인덱스 값과 연관되지는 않을 수도 있다는 것이 이해되어야 한다. 즉, 위에서 언급된 바와 같이, 일부 예들에서, 비디오 인코더 (20) 는 픽셀 값이 팔레트들 (244) 내에 포함되지 않을 경우, 맵 (240) 에서의 위치에 대한 실제적인 픽셀 값 (또는 그 양자화된 버전) 의 표시를 인코딩할 수도 있다 (그리고 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 획득할 수도 있음).

[0109] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 어느 픽셀 위치들이 인덱스 값들과 연관되는지를 표시하는 추가적인 맵을 코딩하도록 구성될 수도 있다. 예를 들어, 맵에서의 (i, j) 엔트리는 블록의 (i, j) 위치에 대응하는 것으로 가정한다. 비디오 인코더 (20) 는 엔트리가 연관된 인덱스 값을 가지는지 여부를 표시하는, 맵의 각각의 엔트리 (즉, 각각의 픽셀 위치) 에 대한 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 블록에서의 (i, j) 로케이션에서의 픽셀 값이 팔레트들 (244) 에서의 값들 중의 하나라는 것을 표시하기 위하여 1 의 값을 가지는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 이러한 예에서, 또한, 그 픽셀 값을 팔레트에서 표시하고 비디오 디코더가 픽셀 값을 복원하는 것을 허용하기 위하여 (도 3 의 예에서 값들 1 내지 3 으로서 도시된) 팔레트 인덱스를 인코딩할 수도 있다. 팔레트들 (244) 이 단일 엔트리 및 연관된 픽셀 값을 포함하는 사례들에서, 비디오 인코더 (20) 는 인덱스 값의 시그널링을 스kip (skip) 할 수도 있다. 비디오 인코더 (20) 는 블록에서의 (i, j) 로케이션에서의 픽셀 값이 팔레트들 (244) 에서의 값들 중의 하나가 아니라는 것을 표시하기 위하여 0 의 값을 가지도록 플래그를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 또한, 픽셀 값을 복원함에 있어서 비디오 디코더 (30) 에 의한 이용을 위하여 픽셀 값의 표시를 인코딩할 수도 있다. 일부 사례들에서, 픽셀 값은 손실 방식으로 코딩될 수도 있다.

[0110] 블록의 하나의 위치에서의 픽셀의 값은 블록의 다른 위치들에서의 하나 이상의 다른 픽셀들의 값들의 표시를 제공할 수도 있다. 예를 들어, (하나를 초과하는 픽셀 값이 단일 인덱스 값으로 맵핑될 수도 있는 손실있는 코딩의 경우에) 블록의 이웃하는 픽셀 위치들이 동일한 픽셀 값을 가지거나 동일한 인덱스 값으로 맵핑될 수도 있는 상대적으로 높은 확률이 있을 수도 있다.

[0111] 따라서, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 는 동일한 픽셀 값 또는 인덱스 값을 가지는, 소정의 스캔 순서에서의 연속적인 픽셀들 또는 인덱스 값들의 수를 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 위에서 언급된 바와 같이, 유사-값의 픽셀 또는 인덱스 값들의 스트링은 런 (run) 으로서 본원에서 지칭될 수도 있다. 예시의 목적들을 위한 예에서, 소정의 스캔 순서에서의 2 개의 연속 픽셀들 또는 인덱스들이 상이한 값들을 가질 경우, 런은 제로와 동일하다. 소정의 스캔 순서에서의 2 개의 연속 픽셀들 또는 인덱스들이 동일한 값을 가지지만 스캔 순서에서의 제 3 픽셀 또는 인덱스가 상이한 값을 가질 경우, 런은 1 과 동일하다. 동일한 값을 갖는 3 개의 연속 인덱스들 또는 픽셀들에 대하여, 런은 2 이고, 이하 등등과 같다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 런을 표시하는 신팩스 엘리먼트들을 획득할 수도 있고, 동일한 픽셀 또는 인덱스 값을 가지는 연속 로케이션들의 수를 결정하기 위하여 데이터를 이용할 수도 있다.

[0112] 일부 예들에서, 현재의 블록에 대한 팔레트 내에 있는 픽셀 값들을 가지는 현재의 블록에서의 모든 픽셀 로케이션들은 팔레트 인덱스와, 그 다음으로, 연속적인 픽셀 로케이션들에서의 픽셀 값의 "런" 으로 인코딩된다.

팔레트에서 오직 하나의 엔트리가 있을 경우에는, 팔레트 인덱스 또는 "런"의 송신이 현재의 블록에 대해 스킵될 수도 있다. 현재의 블록에서의 픽셀 로케이션들 중의 하나에서의 픽셀 값이 팔레트에서의 픽셀 값과 정확한 일치를 가지지 않을 경우에는, 비디오 인코더 (20)가 가장 근접한 픽셀 값을 가지는 팔레트 엔트리들 중의 하나를 선택할 수도 있고, 원래의 픽셀 값 및 팔레트 내에 포함된 예측 픽셀 값 사이의 예측 에러 또는 잔차 값을 계산할 수도 있다. 비디오 인코더 (20)는 픽셀 로케이션에 대한 잔차 값을 인코딩하고 비디오 디코더로 송신한다. 그 다음으로, 비디오 디코더 (30)는 대응하는 수신된 팔레트 인덱스에 기초하여 픽셀 로케이션에서의 픽셀 값을 유도할 수도 있고, 그 다음으로, 유도된 픽셀 값 및 잔차 값을 현재의 블록에서의 픽셀 로케이션에서 원래의 픽셀 값을 예측하기 위하여 이용된다. 하나의 예에서, 잔차 값은, 잔차 값을 변환하고, 변환 계수들을 양자화하고, 양자화된 변환 계수들을 엔트로피 인코딩하기 위하여 RQT를 적용하는 것과 같은, HEVC 초안 10에 의해 특정된 HEVC 방법을 이용하여 인코딩된다. 일부 경우에는, 상기 예가 손실있는 코딩으로서 지칭될 수도 있다.

[0113] 예시의 목적들을 위한 예에서는, 맵 (240)의 라인 (248)을 고려한다. 수평의 좌-우 스캔 방향을 가정하면, 라인 (248)은 "2"의 5개의 인덱스 값을 및 "3"의 3개의 인덱스 값을 포함한다. 이 개시물의 양태들에 따르면, 비디오 인코더 (20)는 스캔 방향에서 라인 (248)의 제1 위치에 대한 2의 인덱스 값을 인코딩할 수도 있다. 게다가, 비디오 인코더 (20)는 시그널링된 인덱스 값을 동일한 인덱스 값을 가지는 스캔 방향에서 연속적인 값들의 런을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 라인 (248)의 예에서, 비디오 인코더 (20)는 4의 런을 시그널링할 수도 있음으로써, 스캔 방향에서의 뒤따르는 4개의 위치들의 인덱스 값을 시그널링된 인덱스 값을 동일한 인덱스 값을 공유한다는 것을 표시할 수도 있다.

비디오 인코더 (20)는 라인 (248)에서 다음의 상이한 인덱스 값을 대하여 동일한 프로세스를 수행할 수도 있다. 즉, 비디오 인코더 (20)는 3의 인덱스 값을 및 2의 런을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 디코더 (30)는 인덱스 값을, 동일한 인덱스 값을 가지는 스캔 방향에서의 연속적인 인덱스들의 수 (런)를 표시하는 선택스 엘리먼트들을 획득할 수도 있다.

[0114] 위에서 언급된 바와 같이, 맵의 인덱스들은 특정한 순서로 스캔된다. 이 개시물의 양태들에 따르면, 스캔 방향은 수직, 수평, 또는 대각 (예컨대, 블록에서 대각선으로 45도 또는 135도)일 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 블록의 인덱스들을 스캔하기 위한 스캔 방향을 표시하는, 각각의 블록에 대한 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 추가적으로 또는 대안적으로, 스캔 방향은 예를 들어, 블록 크기, 컬러 공간, 및/또는 컬러 컴포넌트와 같은 소위 부가 정보에 기초하여 시그널링되거나 추론될 수도 있다. 비디오 인코더 (20)는 블록의 각각의 컬러 컴포넌트에 대한 스캔들을 특정할 수도 있다. 대안적으로, 특정된 스캔은 블록의 모든 컬러 컴포넌트들에 적용할 수도 있다.

[0115] 예를 들어, 열 기반 스캔에 대하여, 맵 (240)의 열 (252)을 고려한다. 수직의 상-하 스캔 방향을 가정하면, 열 (252)은 "1"의 하나의 인덱스 값, "2"의 5개의 인덱스 값을, 및 "3"의 2개의 인덱스 값을 포함한다. 이 개시물의 양태들에 따르면, 비디오 인코더 (20)는 (열 (252)의 상대적인 상부에서) 스캔 방향에서의 열 (252)의 제1 위치에 대한 1의 인덱스 값을 인코딩할 수도 있다. 게다가, 비디오 인코더 (20)는 제로의 런을 시그널링할 수도 있음으로써, 스캔 방향에서의 뒤따르는 위치의 인덱스 값이 상이하다는 것을 표시할 수도 있다. 그 다음으로, 비디오 인코더 (20)는 스캔 방향에서의 다음의 위치에 대한 2의 인덱스 값을, 4의 런, 즉, 스캔 방향에서의 뒤따르는 4개의 위치들의 인덱스 값을 시그널링된 인덱스 값을 동일한 인덱스 값을 공유한다는 것을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 그 다음으로, 비디오 인코더 (20)는 스캔 방향에서의 다음의 상이한 인덱스 값을 대한 3의 인덱스 값을, 1의 런을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 디코더 (30)는 인덱스 값을, 동일한 인덱스 값을 가지는 스캔 방향에서의 연속적인 인덱스들의 수 (런)를 표시하는 선택스 엘리먼트들을 획득할 수도 있다.

[0116] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30)는 추가적으로 또는 대안적으로, 맵 (240)의 하나 이상의 엔트리들에 대한 라인 복사를 수행할 수도 있다. 라인 복사는 일부 예들에서, 스캔 방향에 종속될 수도 있다. 예를 들어, 비디오 인코더 (20)는 맵에서의 특정한 엔트리에 대한 픽셀 또는 인덱스 값이 (수평 스캔에 대하여) 특정한 엔트리 상부의 라인, 또는 (수직 스캔에 대하여) 특정한 엔트리 좌측의 열에서의 픽셀 또는 인덱스 값과 동일하다는 것을 표시할 수도 있다. 비디오 인코더 (20)는 또한, 특정한 엔트리의 상부의 라인 또는 특정한 엔트리의 좌측의 열에서의 엔트리와 동일한, 스캔 순서에서의 픽셀 또는 인덱스 값들의 수를 런으로서 표시할 수도 있다. 이 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 특정된 이웃하는 라인으로부터, 그리고 현재 코딩되고 있는 맵의 라인에 대한 특정된 수의 엔트리들로부터

픽셀 또는 인덱스 값들을 복사할 수도 있다.

[0117] 예시의 목적들을 위한 예에서는, 맵 (240)의 열들 (256 및 260)을 고려한다. 수직의 상하 스캔 방향을 가정하면, 열 (256)은 "1"의 3개의 인덱스 값들, "2"의 3개의 인덱스 값들, 및 "3"의 2개의 인덱스 값들을 포함한다. 열 (260)은 스캔 방향에서 동일한 순서를 가지는 동일한 인덱스 값들을 포함한다. 이 개시물의 양태들에 따르면, 비디오 인코더 (20)는 전체 열 (260)이 열 (256)로부터 복사된다는 것을 표시하는, 열 (260)에 대한 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 하나 이상의 선택스 엘리먼트들은 맵 (240)의 상대적인 상부에서 열 (260)의 제 1 엔트리와 연관될 수도 있다. 비디오 디코더 (30)는 라인 복사를 표시하는 선택스 엘리먼트들을 획득할 수도 있고, 열 (260)을 디코딩할 때에 열 (260)에 대한 열 (256)의 인덱스 값들을 복사할 수도 있다.

[0118] 이 개시물의 양태들에 따르면, 소위 엔트리들의 런들을 코딩하기 위한 기법들은 위에서 설명된 라인 복사를 위한 기법들과 함께 이용될 수도 있다. 예를 들어, 비디오 인코더 (20)는 맵에서의 엔트리의 값이 팔레트로부터 획득되는지, 또는 맵에서의 엔트리의 값이 맵 (240)에서의 이전에 코딩된 라인으로부터 획득되는지 여부를 표시하는 하나 이상의 선택스 엘리먼트들 (예컨대, 플래그)을 인코딩할 수도 있다. 비디오 인코더 (20)는 또한, 팔레트의 인덱스 값 또는 라인 (행 또는 열)에서의 엔트리의 로케이션을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20)는 또한, 동일한 값을 공유하는 연속적인 엔트리들의 수를 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 디코더 (30)는 인코딩된 비트스트림으로부터 이러한 정보를 획득할 수도 있고, 맵 및 블록에 대한 픽셀 값들을 복원하기 위하여 그 정보를 이용할 수도 있다.

[0119] 예시의 목적들을 위한 예에서, 맵 (240)의 행 (row) 들 (264 및 268)을 고려한다. 수평의 좌-우 스캔 방향을 가정하면, 행 (264)은 "1"의 5개의 인덱스 값들 및 "3"의 3개의 인덱스 값들을 포함한다. 행 (268)은 "1"의 3개의 인덱스 값들, "2"의 2개의 인덱스 값들, 및 "3"의 3개의 인덱스 값들을 포함한다. 이 예에서, 비디오 인코더 (20)는 행 (268)에 대한 데이터를 인코딩할 때, 행 (264)의 특정한 엔트리들과, 그 다음으로, 런을 식별할 수도 있다. 예를 들어, 비디오 인코더 (20)는 행 (268)의 제 1 위치 (행 (268)의 가장 좌측 위치)가 행 (264)의 제 1 위치와 동일하다는 것을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20)는 또한, 행 (268)에서의 스캔 방향에서 2개의 연속적인 엔트리들의 다음 런이 행 (264)의 제 1 위치와 동일하다는 것을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다.

[0120] 일부 예들에서, 비디오 인코더 (20)는 또한, 또 다른 행 (또는 열)에서의 위치에 관하여 현재의 픽셀 또는 인덱스 값을 코딩할 것인지, 또는 런 선택스 엘리먼트를 이용하여 현재의 픽셀 또는 인덱스 값을 코딩할 것인지 여부를 결정할 수도 있다. 예를 들어, 행 (264)의 제 1 위치 및 (위에서 언급된) 2개 엔트리들의 런을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩한 후, 비디오 인코더 (20)는 (좌측으로부터 우측으로) 라인 (268)에서의 제 4 및 제 5 위치들에 대하여, 제 4 위치에 대한 2의 값을 표시하는 하나 이상의 선택스 엘리먼트들과, 1의 런을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 이 때문에, 비디오 인코더 (20)는 또 다른 라인 (또는 열)을 참조하지 않으면서 이 2개의 위치들을 인코딩한다. 그 다음으로, 비디오 인코더 (20)는 (예컨대, 상부 행 (264)으로부터의 복사, 및 동일한 인덱스 값을 가지는 스캔 순서에서의 연속적인 위치들의 런을 표시하는) 상부 행 (264)에 관하여 행 (268)에서의 3의 인덱스 값을 가지는 제 1 위치를 코딩할 수도 있다. 이 때문에, 이 개시물의 양태들에 따르면, 비디오 인코더 (20)는 예컨대, 런을 이용하여 라인 (또는 열)의 다른 값들에 관하여 라인 (또는 열)의 픽셀 또는 인덱스 값을 코딩하거나, 또 다른 라인 (또는 열)의 값들에 관하여 라인 (또는 열)의 픽셀 또는 인덱스 값을 코딩하거나, 또는 이들의 조합 사이에서 선택할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 선택을 행하기 위하여 레이트/왜곡 최적화를 수행할 수도 있다.

[0121] 비디오 디코더 (30)는 위에서 설명된 선택스 엘리먼트들을 수신할 수도 있고, 행 (268)을 복원할 수도 있다. 예를 들어, 비디오 디코더 (30)는 현재 코딩되고 있는 맵 (240)의 위치에 대한 연관된 인덱스 값을 그로부터 복사하기 위한 이웃하는 행에서의 특정한 로케이션을 표시하는 데이터를 획득할 수도 있다. 비디오 디코더 (30)는 또한, 동일한 인덱스 값을 가지는 스캔 순서에서 연속적인 위치들의 수를 표시하는 데이터를 획득할 수도 있다.

[0122] 일부 사례들에서, 엔트리들이 그로부터 복사되는 라인은 (도 3의 예들에서 예시된 바와 같이) 현재 코딩되고 있는 라인의 엔트리에 직접적으로 인접할 수도 있다. 그러나, 다른 예들에서, 다수의 라인들은 비디오 인코

더 (20) 및/또는 비디오 디코더 (30)에 의해 베퍼링될 수도 있어서, 맵의 다수의 라인들 중의 임의의 것은 현재 코딩되고 있는 맵의 라인에 대한 예측 엔트리들로서 이용될 수도 있다. 이 때문에, 일부 예들에서, 엔트리에 대한 픽셀 값은 현재의 행 바로 상부의 행 (또는 그 좌측의 열) 또는 현재의 행 상부의 2 개 이상의 행들 (또는 그 좌측의 열)에서의 엔트리의 픽셀 값과 동일하도록 시그널링될 수도 있다.

[0123]

예시의 목적들을 위한 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 엔트리들의 현재의 행을 코딩하기 이전에 엔트리들의 이전의  $n$  개의 행들을 저장하도록 구성될 수도 있다. 이 예에서, 비디오 인코더 (20)는 절단된 1진 코드 또는 다른 코드들을 갖는 비트스트림에서 예측 행 (엔트리들이 그로부터 복사되는 행)을 표시할 수도 있다. 또 다른 예에서, 비디오 인코더 (20)는 현재의 라인과, 현재의 라인을 코딩하기 위한 참조로서 이용된 맵 (240)의 예측 라인과의 사이의 변위 값을 인코딩할 수도 있다 (그리고 디코더 (30)는 디코딩할 수도 있음). 즉, 비디오 인코더 (20)는 인덱스 값이 그로부터 복사되는 특정한 라인 (또는 열)의 표시를 인코딩할 수도 있다. 일부 예들에서, 변위 값은 변위 벡터일 수도 있다. 즉,  $c[0], c[1], \dots$ , 는 맵 (240)의 현재의 라인의 인덱스들을 나타낸다고 하고,  $u[0], u[1], u[2], \dots$ , 는 상부의 이웃하는 라인과 같은 맵 (240)의 예측 라인의 인덱스들을 나타낸다고 한다. 이 예에서, 변위 벡터가  $d$  인 것으로 주어지면,  $c[i]$ 에 대한 인덱스 값은  $d$  가 네거티브 값들을 취하는 것을 회피하기 위하여  $u[i+d]$  또는  $u[i-d]$ 로부터 예측될 수도 있다.  $d$ 의 값은 1진, 절단된 1진, 지수 골룸, 또는 골룸-라이스 코드들을 이용하여 코딩될 수도 있다.

[0124]

또 다른 예로서, 비디오 인코더 (20)는 현재 코딩되고 있는 맵의 라인으로 복사하기 위하여 이웃하는 라인 및 이웃하는 라인의 엔트리들의 수 또는 부분을 표시하는, "상부 라인 좌측 절반으로부터 복사" 또는 "상부 라인 우측 절반으로부터 복사"와 같은 명령을 시그널링할 수도 있다. 추가적인 예로서, 인덱스 값들의 맵은 코딩하기 전에 재순서화될 수도 있다. 예를 들어, 인덱스 값들의 맵은 코딩 효율을 개선시키기 위하여 90, 180, 또는 270 도만큼 회전될 수도 있거나, 상부면 하부로 또는 좌측면 우측으로 플립 (flip) 될 수도 있다.

[0125]

다른 예들에서, 비디오 인코더 (20)는 맵 (240)의 유사-값의 인덱스 값들의 런들을 비디오 디코더 (30)로 송신하지 않을 수도 있다. 이 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 런들의 값들을 목시적으로 유도할 수도 있다. 하나의 예에서, 런의 값은 일정한 값, 예컨대, 4, 8, 16 등일 수도 있다. 또 다른 예에서, 런의 값은 예를 들어, 블록 크기, 양자화 파라미터 (QP), 프레임 타입, 컬러 컴포넌트, 컬러 포맷 (예컨대, 4:4:4, 4:2:2, 또는 4:2:0), 컬러 공간 (예컨대, YUV 또는 RGB), 스캔 방향 및/또는 현재의 블록에 대한 다른 타입들의 특성 정보와 같은, 코딩되고 있는 비디오 데이터의 현재의 블록에 대한 부가 정보에 종속적일 수도 있다. 런의 값이 블록 크기에 종속될 경우에는, 런은 현재의 블록의 폭, 현재의 블록의 높이, 현재의 블록의 절반-폭 (또는 절반-높이), 현재의 블록의 폭 및/또는 높이의 분수, 또는 현재의 블록의 폭 및/또는 높이의 배수와 동일할 수도 있다. 또 다른 예에서, 비디오 인코더 (20)는 VPS, SPS, PPS, 및/또는 슬라이스 헤더에서의 신팩스와 같은 하이 레벨 신팩스를 이용하여 런의 값을 비디오 디코더 (30)로 시그널링할 수도 있다.

[0126]

추가적으로 또는 대안적으로, 비디오 인코더 (20)는 맵 (240)을 비디오 디코더 (30)로 송신할 필요조차 없을 수도 있다. 그 대신에, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 맵 (240) 내에 포함된 인덱스 값들의 각각의 런의 시작 위치 또는 로케이션을 목시적으로 유도할 수도 있다. 하나의 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 적용된 비디오 코딩 표준은 런이 어떤 로케이션들에서 오직 시작할 수 있는 것으로 결정할 수도 있다. 예를 들어, 런은 각각의 행의 시작부, 또는 코딩되고 있는 현재의 블록의 매  $N$  행들의 시작부에서 오직 시작할 수도 있다. 시작 로케이션은 상이한 스캔 방향들에 대하여 상이할 수도 있다. 예를 들어, 수직 스캔이 이용될 경우, 런은 열의 시작부, 또는 현재의 블록의 매  $N$  개의 열들의 시작부에서 오직 시작할 수도 있다.

[0127]

또 다른 예에서, 시작 로케이션은 예를 들어, 블록 크기, QP, 프레임 타입, 컬러 컴포넌트, 컬러 포맷 (예컨대, 4:4:4, 4:2:2, 또는 4:2:0), 컬러 공간 (예컨대, YUV 또는 RGB), 스캔 방향, 및/또는 현재의 블록에 대한 다른 타입들의 특성 정보와 같은, 현재의 블록에 대한 부가 정보에 따라 유도될 수도 있다. 런의 시작 로케이션이 블록 크기에 종속될 경우, 시작 로케이션은 각각의 행 및/또는 각각의 열의 중간-포인트, 또는 각각의 행 및/또는 열의 분수 (예컨대,  $1/n, 2/n, \dots, (n-1)/n$ ) 일 수도 있다. 또 다른 예에서, 비디오 인코더 (20)는 PPS, SPS, VPS, 및/또는 슬라이스 헤더에서의 신팩스와 같은 하이 레벨 신팩스를 이용하여 시작 위치를 비디오 디코더 (30)로 시그널링할 수도 있다.

[0128]

일부 예들에서는, 위에서 각각 설명된 목시적 시작 위치 유도 및 목시적 런 유도가 조합될 수도 있다. 예를

들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 맵의 유사-값의 인덱스 값들의 런이 2 개의 이웃하는 시작 위치들 사이의 거리와 동일한 것으로 결정할 수도 있다. 시작 위치가 현재의 블록의 매 행의 시작부 (즉, 제 1 위치) 인 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 런의 길이가 현재의 블록의 전체 행의 길이와 동일한 것으로 결정할 수도 있다.

[0129] HEVC SCC 확장에 대한 다양한 제안들은 현재의 블록에 대한 팔레트를 유도하기 위하여 팔레트 예측자를 이용하는 팔레트 모드를 포함한다. 비디오 인코더 (20) 및 비디오 디코더 (30)는 현재 코딩되고 있는 블록에 대한 팔레트 테이블을 결정하기 이전에, 팔레트 예측자를 업데이트할 수도 있다. 이전에 이용된 팔레트 예측자가 이용가능하지 않은 제 1 블록에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30)는 PPS 및 SPS에서 각각 시그널링된 팔레트 예측자 엔트리들로부터의 이 초기화를 인에이블하기 위하여, 2 개의 플래그들: pps\_palette\_predictor\_initializer\_present\_flag 및 sps\_palette\_predictor\_initializer\_present\_flag 의 이용으로, PPS 레벨에서, 또는 SPS 레벨에서 중의 어느 하나에서 팔레트 예측자를 초기화할 수도 있다. PPS 및 SPS 팔레트 예측자의 크기는 신팩스 엘리먼트들 pps\_num\_palette\_predictor\_initializer\_minus1 및 sps\_num\_palette\_predictor\_initializer\_minus1에 의해 각각 정의된다. 일부 제안된 시그널링 메커니즘들 하에서, PPS 또는 SPS에서 시그널링된 팔레트 예측자의 최소 크기는 1 이다.

[0130] 팔레트 예측자가 초기화될 수도 있는 하나의 예의 순서는 WD4 섹션 9.3.2.3에서 설명되어 있다.

[0131] pps\_palette\_predictor\_initializer\_present\_flag 가 1 과 동일할 경우, 팔레트 예측자는 PPS 예측자에 기초하여 초기화되고,

[0132] 그렇지 않고, (pps\_palette\_predictor\_initializer\_present\_flag 가 0 과 동일함) 일 경우에는, sps\_palette\_predictor\_initializer\_present\_flag 가 1 과 동일할 경우, 팔레트 예측자는 SPS 예측자로부터 초기화되고,

[0133] 그렇지 않을 경우 (양자의 플래그들이 0 과 동일할 때), 팔레트 예측자 크기는 0 으로 설정된다.

[0134] 위에서 설명된 초기화 규칙으로부터 알 수 있는 바와 같이, sps\_palette\_predictor\_initializer\_present\_flag 가 인에이블될 경우, 꾹처마다에 기초하여, 팔레트 예측자 크기를 0 으로 설정하기 위한 것인 팔레트 예측자를 디스에이블하기 위한 방법이 없다. 팔레트 예측자 크기를 0 으로 설정하기 위한 능력은, SPS에서 시그널링된 팔레트 예측자가 어떤 꾹처에서의 팔레트 엔트리들을 예측하기 위하여 관련되지 않고 그것들을 PPS에서 전송하는 것이 많은 레이트를 소비할 수도 있을 경우에 바람직할 수도 있다. 또 다른 잠재적인 문제점은, 양자의 PPS 및 SPS 팔레트 예측자들이 존재할 때, PPS 및 SPS 팔레트 예측자들이 많은 공통적인 팔레트 엔트리들을 가질 수도 있으므로, 현재의 시그널링은 중복적일 수도 있다는 것이다.

[0135] 이 개시물은 팔레트 예측자 초기화 제어에 관련된 다양한 기법들을 도입한다. 위에서 도입된 첫 번째 문제점을 잠재적으로 해결하기 위하여, 이 개시물은 팔레트 예측자 초기화를 제어하기 위하여 이용될 수 있는 새로운 신팩스 엘리먼트 palette\_predictor\_initialization\_flag 를 추가하기 위한 기법들을 설명한다. 이 플래그의 값이 1 과 동일할 때, 팔레트 예측자 초기화가 이용된다. 그렇지 않을 경우, 팔레트 예측자 크기는 0 과 동일하게 설정된다. 이 플래그는 VPS, SPS, PPS, 슬라이스 헤더 또는 그 확장들, 또는 다른 곳과 같은 임의의 파라미터 세트에서 시그널링될 수 있다. 하나의 예에서, 플래그는 PPS에서 시그널링될 수도 있다. 이러한 플래그를 이용하는 잠재적인 장점은, SPS 팔레트 예측자 초기화자들이 시그널링될 때, 플래그가 팔레트 예측자를 디스에이블 (팔레트 예측자 크기를 0 으로 설정) 하기 위한 능력을 제공한다는 것이다.

[0136] 2 개의 팔레트 예측자들 (예컨대, PPS에서의 하나, 및 SPS에서의 또 다른 하나) 이 있을 수 있으므로, palette\_predictor\_initialization\_flag 가 하나의 파라미터 세트에서 오직 시그널링될 때, 비트스트림 적합성 규칙은 파라미터 세트 파싱 종속성을 회피하기 위하여, 그리고 palette\_predictor\_initialization\_flag 가 1 과 동일하게 설정되고 팔레트 예측자들이 임의의 파라미터 세트에서 시그널링되지 않을 경우에 대하여 도입될 수 있다. 비트스트림 적합성 한정은 다음과 같이 표현될 수도 있다: 적합한 비트스트림은 palette\_predictor\_initialization\_flag 가 1 과 동일할 때에 1 과 동일한 pps\_palette\_predictor\_initializer\_present\_flag 및 sps\_palette\_predictor\_initializer\_present\_flag 중의 적어도 하나를 포함할 것이다. 이 규칙은, 그것이 현재의 신팩스 엘리먼트들이 일부 다른 신팩스 엘리먼트들에 의해 수정되거나 대체될 수 있지만, 유사한 역할들을 할 때, 유사한 문제들을 해결하기 위하여 또한 적용 가능할 수도 있으므로, 현존하는 그리고 제안된 신팩스 엘리먼트들로 한정되도록 의도된 것은 아니다.

[0137] 대안적인 해결책에서, 비디오 인코더 (20)는 PPS에서 0 과 동일한 팔레트 예측자 초기화자를 시그널링하도록

허용될 수도 있다. 이 경우, 접미사 "minus1"은 선택스 엘리먼트 `pps_num_palette_predictor_initializer_minus1`로부터 제거될 수도 있다. 게다가, 시맨틱들은 팔레트 예측자 초기화자 크기가 제로일 수 있다는 것을 반영하기 위하여 변경될 수도 있다. 이에 따라, 현존하는 초기화 규칙은 PPS에서 제로 팔레트 예측자를 가지는 것을 허용하기 위하여 변경될 수 있다. 이러한 경우, PPS에서의 팔레트 예측자 크기를 0과 동일하게 설정하는 것은 예를 들어, 0과 동일한 `pps_num_palette_predictor_initializer`를 갖는 1과 동일한 `pps_palette_predictor_initializer_present_flag`를 시그널링함으로써, SPS 팔레트 예측자 초기화자가 시그널링될 때에도 팔레트 예측자 초기화를 스위치 오프 (switch off) 하기 위하여 이용될 수 있다. 즉, 비디오 디코더 (30)가 PPS에서, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 `pps_num_palette_predictor_initializer`에 대한 값을 수신할 경우, 비디오 디코더 (30)는 블록이 연관되는 SPS가 팔레트 예측자 초기화가 인에이블되게 하더라도, 팔레트 예측자 초기화가 PPS와 연관된 비디오 블록들에 대하여 디스에이블되는 것으로 결정하도록 구성될 수도 있다.

[0138] 추가적으로, 새롭게 도입된 선택스 엘리먼트, `pps_num_palette_predictor_initializer` 값은, 그 선택스 엘리먼트들이 초기화자 크기가 0과 동일할 때에 필요하지 않으므로, `monochrome_palette_flag`, `luma_bit_depth_entry_minus8`, 및 `chroma_bit_depth_entry_minus8`과 같은 팔레트 예측자 초기화자 엔트리들에 특정적인 선택스 엘리먼트들에 대한 새로운 게이팅 플래그 (gating flag)로서 이용될 수 있다. 이에 따라, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 선택스 엘리먼트 `pps_num_palette_predictor_initializer`에 기초하여, 비디오 디코더 (30)는 선택스 엘리먼트들 `monochrome_palette_flag`, `luma_bit_depth_entry_minus8`, 및 `chroma_bit_depth_entry_minus8`가 PPS에서 존재하지 않는 것으로 결정할 수도 있다.

[0139] 예를 들어, PPS 시그널링은 다음과 같이 수정될 수 있고, 여기서, 변경들은 밑줄 및 중간줄로 이하에서 도시되어 있다:

|   |       |
|---|-------|
| <code>pps_palette_predictor_initializer_present_flag</code>                         | u(1)  |
| <code>if( pps_palette_predictor_initializer_present_flag ) {</code>                 |       |
| <u><code>pps_num_palette_predictor_initializer</code></u>                           | ue(v) |
| <code>if( pps_num_palette_predictor_initializer &gt; 0 ) {</code>                   |       |
| <u><code>monochrome_palette_flag</code></u>   | u(1)  |
| <u><code>luma_bit_depth_entry_minus8</code></u>                                     | ue(v) |
| <code>if( !monochrome_palette_flag )</code>   |       |
| <u><code>chroma_bit_depth_entry_minus8</code></u>                                   | ue(v) |
| <u><code>pps_num_palette_predictor_initializer_minus1</code></u>                    | ue(v) |
| <code>numComps = monochrome_palette_flag ? 1 : 3</code>                             |       |
| <code>for( comp = 0; comp &lt; numComps; comp++ )</code>                            |       |
| <code>for( i = 0; i &lt; pps_num_palette_predictor_initializer_minus1; i++ )</code> |       |
| <u><code>pps_palette_predictor_initializer</code></u> <code>[ comp ][ i ]</code>    | u(v)  |
| <code>}</code>  |       |
| <code>}</code>  |       |

[0140]

[0141] 선택스 엘리먼트 `pps_num_palette_predictor_initializer`는 꽉쳐 팔레트 예측자 초기화자에서의 엔트리들의 수를 특정한다. 비디오 디코더 (30)가 `pps_num_palette_predictor_initializer`에 대한 값이 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 것으로 결정할 경우, 비디오 디코더 (30)는 팔레트 예측자 초기화가 PPS에 대하여 디스에이블되는 것으로 결정할 수도 있다.

[0142]

`pps_num_palette_predictor_initializer`의 값이 0 내지 `PaletteMaxPredictorSize` 까지의 범위에 있다는 것은 비트스트림 적합성의 요건일 수도 있다.

[0143]

선택스 엘리먼트 `pps_palette_predictor_initializer``[ comp ][ i ]`는 아래의 `PredictorPaletteEntries`를 초기화하기 위하여 이용되는 `i`-번째 팔레트 엔트리의 `comp`-번째 컴포넌트의 값을 특정한다. 0 내지 `pps_num_palette_predictor_initializer` - 1 까지의 범위에서의 `i`의 값들에 대하여, `pps_palette_predictor_initializer``[ 0 ][ i ]`의 값은 0 내지  $(1 << \text{BitDepthEntryY}) - 1$  까지의 범위에 있을 것이고, `pps_palette_predictor_initializer``[ 1 ][ i ]` 및 `pps_palette_predictor_initializer``[ 2 ][ i ]`의 값들은 0 내지  $(1 << \text{BitDepthEntryC}) - 1$  까지의 범위에 있을 것이다.

[0144]

초기화 규칙은 이에 따라 변경될 수 있다:

[0145]

이 프로세스의 출력들은 초기화된 팔레트 예측자 변수들 `PredictorPaletteSize` 및 `PredictorPaletteEntries` 이

다.

[0146] 변수 numComps 는 다음과 같이 유도된다:

[0147] numComps = ( ChromaArrayType == 0 ) ? 1 : 3

[0148] - pps\_palette\_predictor\_initializer\_present\_flag 가 1 과 동일할 경우, 다음이 적용된다:

[0149] - PredictorPaletteSize 는 pps\_num\_palette\_predictor\_initializer 와 동일하게 설정됨

[0150] - 아래 PredictorPaletteEntries 는 다음과 같이 유도된다:

[0151] 

```
for( comp = 0; comp < numComps; comp++ )
```

[0152] 

```
for( i = 0; i < PredictorPaletteSize; i++ )
```

[0153] PredictorPaletteEntries[ comp ][ i ] = pps\_palette\_predictor\_initializers[ comp ][ i ]

[0154] - 그렇지 않을 경우에는 (pps\_palette\_predictor\_initializer\_present\_flag 가 0 과 동일함), sps\_palette\_predictor\_initializer\_present\_flag 가 1 과 동일할 경우, 다음이 적용된다:

[0155] - PredictorPaletteSize 는 sps\_num\_palette\_predictor\_initializer\_minus1 플러스 1 과 동일하게 설정된다.

[0156] - 아래 PredictorPaletteEntries 는 다음과 같이 유도된다:

[0157] 

```
for( comp = 0; comp < numComps; comp++ )
```

[0158] 

```
for( i = 0; i < PredictorPaletteSize; i++ )
```

[0159] PredictorPaletteEntries[ comp ][ i ] = sps\_palette\_predictor\_initializers[ comp ][ i ]

[0160] - 그렇지 않을 경우에는 (pps\_palette\_predictor\_initializer\_present\_flag 가 0 과 동일하고 sps\_palette\_predictor\_initializer\_present\_flag 가 0 과 동일함), PredictorPaletteSize 가 0 과 동일하게 설정되고, PredictorPaletteSize 가 0 과 동일하게 설정된다.

[0161] 이 개시물은 또한, 팔레트 예측자 초기화 예측과 관련된 기법들을 설명한다. 가능한 중복적인 엔트리들로 위에서 도입된 두 번째 문제점을 잠재적으로 해결하기 위하여, 이 개시물은 현존하는 기법들 대신에, 팔레트 예측자 초기화를 위하여 양자의 SPS 및 PPS 팔레트 예측자 초기화자들을 이용하기 위한 기법들을 설명하고, 여기서, 오직 하나의 예측자가 한 번에 이용될 수 있다. 이에 따라, 팔레트 예측자를 초기화할 때, 비디오 디코더 (30)는 SPS-레벨 팔레트 예측자 초기화자들 또는 PPS-레벨 팔레트 예측자 초기화자들 중의 오직 하나 대신에, 양자의 SPS-레벨 팔레트 예측자 초기화자들 및 PPS-레벨 팔레트 예측자 초기화자들을 팔레트 예측자 내에 포함할 수도 있다.

[0162] 하나의 예에서, 팔레트 예측자는 SPS 팔레트 예측자 엔트리들과, 그 다음으로, PPS 팔레트 예측자 엔트리들을, 또는 그 반대 순서로 포함함으로써 초기화될 수 있다. 더 일반적인 접근법에서는, 전체 SPS 팔레트 예측자 를 이용하는 대신에, 재이용된 플래그들이 어느 엔트리들이 팔레트 예측자 초기화를 위하여 이용될 것인지를 표시하기 위하여 엔트리마다 시그널링될 수 있다.

[0163] 양자의 SPS 및 PPS 초기화자들이 이용될 때에는, 조합된 초기화자의 크기가 최대 팔레트 예측자 크기보다 더 클 경우, 일부 엔트리들이 폐기되어야 하고 이 여분의 시그널링과 연관된 비트들이 낭비될 수 있으므로, SPS 및 PPS 초기화자들의 모두 합한 크기가 최대 팔레트 예측자 크기 이하인 것과 같은 추가적인 제약이 조합된 초기화자의 총 시그널링된 크기에 대해 부과될 수 있다. 또 다른 더 간단한 대안에서, PPS 및 SPS 초기화자 크기는 최대 팔레트 예측자 크기의 절반으로 한도설정될 수 있다.

[0164] 하나의 예의 구현예에서는, 선택스 엘리먼트 pps\_palette\_predictor\_initializer\_present\_flag 가 1 과 동일할 때, SPS 팔레트 예측자에서의 각각의 엔트리에 대하여, 하나의 플래그는 그 엔트리가 팔레트 예측자를 초기화하기 위하여 이용되는지 여부를 시그널링하기 위하여 전송된다. 시그널링 방법은 PalettePredictorEntryReuseFlags 를 시그널링하기 위하여 WD4 의 섹션 7.3.8.13 에서 이용된 것과 동일할 수도 있다. 그 후, 추가적인 팔레트 예측자 초기화자 엔트리들의 수가 시그널링된다. 이것은 값 0 을 취할 수도 있다. 이것은 실제적인 PPS 팔레트 예측자 엔트리들에 선행한다.

- [0165] SPS 와 PPS 사이의 종속성을 회피하기 위하여, 신택스 엘리먼트 `sps_palette_predictor_initializer_present_flag` 가 1 과 동일할 때에는, `sps_palette_initializer_reuse` 플래그가 전송될 수도 있다. 이 플래그의 값이 1 과 동일할 때에는, SPS 팔레트 예측자 엔트리들의 수가 시그널링된다. 이것은 각각의 SPS 팔레트 예측자 엔트리에 대한 재이용 플래그, 추가적인 팔레트 예측자 엔트리들의 수, 및 최종적으로, 실제적인 PPS 팔레트 예측자 엔트리들의 시그널링에 선행한다. 시그널링된 SPS 팔레트 예측자 엔트리들의 수는 `sps_num_palette_predictor_initializer_minus1 + 1` 이하일 수도 있다. 예를 들어, `sps_num_palette_predictor_initializer_minus1` 이 10 과 동일하고 PPS 에서 시그널링된 SPS 팔레트 예측자 엔트리들의 수가 6 과 동일할 경우, 재이용 플래그들은 최초 6 개의 SPS 팔레트 예측자 엔트리들에 대해 오직 전송되고, 최후 4 개의 SPS 팔레트 예측자 엔트리들은 현재의 PPS 를 지시하는 픽처에 대한 팔레트 예측자 초기화에서 이용되지 않을 수도 있다.
- [0166] 최후의 예에 후속하여, 팔레트 예측자 초기화 재이용 플래그들은 어느 엔트리들이 PPS 또는 SPS, 또는 양자로부터의 적어도 하나의 팔레트 예측자 초기화자로 구성될 수 있는 디폴트 팔레트 예측자 초기화자로부터 재이용되는지를 표시하기 위하여 슬라이스 헤더에서 시그널링될 수도 있다.
- [0167] 예를 들어, SPS 팔레트 예측자 초기화자는 공통적으로 이용된 팔레트 엔트리들을 포함할 수도 있고, PPS 팔레트 예측자 초기화자는, 픽처에 특정적이고 SPS 엔트리들과는 상이한 엔트리들을 오직 가질 수도 있다. 양자의 PPS 및 SPS 초기화자들로부터의 팔레트 예측자를 초기화하는 것은, 이러한 공통적인 엔트리들이 각각의 PPS 팔레트 예측자 초기화자로 반복적으로 시그널링되는 WD4 설계와는 반대로, 모든 시그널링된 PPS 들에 걸쳐 동일할 수도 있는 PPS 에서의 초기화자 엔트리들의 추가적인 시그널링을 회피할 수도 있다.
- [0168] 유사한 방식으로, 팔레트 예측자 초기화자는 더 이전에 전송된 PPS 들 및/또는 SPS 들로부터 조합될 수 있다. 예를 들어, 위에서 설명된 것과 같은 SPS 대신에, 더 이전에 전송된 PPS로부터의 다른 팔레트 예측자 엔트리들이 팔레트 예측자 초기화자에 추가될 수 있다. 하나의 예에서, 팔레트 예측자 초기화자는 SPS 초기화자로부터 하나 이상의 엔트리들을 승계하는 대신에 또는 이에 추가하여, 이전에-시그널링된 PPS 초기화자들로부터 하나 이상의 엔트리들을 승계할 수도 있다. 예로서 이전에-시그널링된 PPS 초기화자들에 대하여 위에서 설명되었지만, 이 개시물의 기법들은 또한, 하나 이상의 이전에-시그널링된 SPS 초기화자들로부터의 하나 이상의 엔트리들의 승계를 가능하게 할 수도 있다는 것이 인식될 것이다. 예를 들어, 엔트리들은 이전에-시그널링된 SPS 초기화자들로부터, 이전에-시그널링된 PPS 초기화자들로부터, 또는 양자의 이전에-시그널링된 SPS 초기화자들 및 이전에-시그널링된 PPS 초기화자들의 조합으로부터 승계될 수도 있다.
- [0169] 또 다른 예에서, 팔레트 예측자 초기화자는 단지 업데이트될 수 있다. 예를 들어, 그것은 초기에 최초 시그널링된 SPS, PPS, 또는 양자의 초기화자들로 구성될 수도 있고, 이 팔레트 예측자는 새롭게 시그널링된 PPS 또는 SPS 의 엔트리들을 팔레트 예측자 초기화자로 삽입함으로써 업데이트될 수도 있다. 엔트리들은 종료부에서, 또는 기본적으로 임의의 로케이션에서, 팔레트 예측자 초기화자의 시작부로 삽입될 수 있다. 삽입하면서 예측자 크기를 초과하는 엔트리들은 무시될 수 있다. 예를 들어, 이러한 업데이트들이 초기화자로 하여금, 예측자 크기 제한을 초과하게 할 경우들에는, 크기 제한을 충족시킨 후에 프로세싱되는 업데이트된 엔트리들이 무시될 수 있거나, 폐기될 수 있거나, 제외될 수 있다.
- [0170] 설명된 접근법의 하나의 특정한 예에서, 팔레트 예측자 초기화자는 SPS 또는 PPS, 또는 양자의 세트들로부터의 새롭게 시그널링된 엔트리들과, 새로운 엔트리들을 뒤따르는 이전의 팔레트 예측자 초기화자의 과거의 엔트리들을 포함할 수도 있다.
- [0171] 이 개시물은 또한, 스케일링 리스트 업데이트에 관련된 기법들을 설명한다. 조합된 팔레트 예측자 초기화자의 유사한 사상은 스케일링 리스트 또는 변환을 위한 양자화 행렬 (quantization matrix) 에 적용될 수 있다. 시그널링된 SPS 및 PPS 스케일링 리스트들이 있을 수 있고, PPS 또는 SPS 리스트의 어느 하나를 오직 이용하는 대신에, 특정한 픽처에 대하여 이용된 스케일링 리스트는 PPS 에서 시그널링된 리스트들에 의해 업데이트되거나 첨부된 SPS 스케일링 리스트로 구성될 수 있다. 예를 들어, 4x4, 8x8, 16x16, 및 32x32 변환 크기들에 대한 스케일링 리스트들이 SPS 에서 시그널링된다는 것을 고려한다. PPS 는 어떤 스케일링 리스트들, 예를 들어, 8x8 및 16x16 변환 크기들과 연관된 것들을 오직 시그널링할 수도 있다. 그 다음으로, 픽처에 대하여 이용된 최종적인 스케일링 리스트는 다음과 같을 수 있다: SPS로부터의 4x4, PPS로부터의 8x8, PPS로부터의 16x16, 및 SPS로부터의 32x32.
- [0172] 또 다른 예에서, SPS 스케일링 리스트들은 예를 들어, 변환 심도 또는 크기들이 PPS 에서 정의될 때, 완전한 세트를 포함하지 않을 수도 있다. 이 경우, SPS 는 PPS 와 반드시 중첩되지는 않는, 기본적인 또는 공통적인

스케일링 리스트들을 오직 가질 수도 있고, 픽처에 대한 특정 스케일링 리스트들은 PPS에서 시그널링될 수도 있고, SPS 및 PPS 스케일링 리스트들의 더 이후의 조합된 세트는 픽처에 대하여 이용된다.

[0173] 도 4 는 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20)를 예시하는 블록도이다. 비디오 인코더 (20)는 비디오 슬라이스들 내의 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 소정의 비디오 프레임 또는 픽처 내의 비디오에서의 공간적 중복성을 감소시키거나 제거하기 위하여 공간적 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내의 비디오에 있어서의 시간적 중복성을 감소시키거나 제거하기 위하여 시간적 예측에 의존한다. 인트라-모드 (I 모드)는 몇몇 공간 기반 압축 모드들 중의 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향-예측 (B 모드)과 같은 인터-모드들은 몇몇 시간 기반 압축 모드들 중의 임의의 것을 지칭할 수도 있다.

[0174] 도 4 의 예에서, 비디오 인코더 (20)는 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 엔트로피 인코딩 유닛 (56)을 포함한다. 예측 프로세싱 유닛 (41)은 팔레트-기반 인코딩 유닛 (42), 인터 예측 프로세싱 유닛 (44), 및 인트라 예측 유닛 (46)을 포함한다. 비디오 블록 복원을 위하여, 비디오 인코더 (20)는 또한, 역양자화 유닛 (58), 역변환 프로세싱 유닛 (60), 합산기 (62), 필터 유닛 (64), 및 디코딩된 픽처 버퍼 (DPB; 66)를 포함한다.

[0175] 도 4에서 도시된 바와 같이, 비디오 인코더 (20)는 비디오 데이터를 수신하고, 수신된 비디오 데이터를 비디오 데이터 메모리 (33) 내에 저장한다. 비디오 데이터 메모리 (33)는 비디오 인코더 (20)의 컴포넌트들에 의해 인코딩되어야 할 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (33) 내에 저장된 비디오 데이터는 예를 들어, 비디오 소스 (18)로부터 획득될 수도 있다. DPB (66)는 예컨대, 인트라-코딩 또는 인터-코딩 모드들에서 비디오 인코더 (20)에 의해 비디오 데이터를 인코딩함에 있어서의 이용을 위한 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66)는 동기식 DRAM (synchronous DRAM; SDRAM)을 포함하는 동적 랜덤 액세스 메모리 (dynamic random access memory; DRAM), 자기저항성 RAM (magnetoresistive RAM; MRAM), 저항성 RAM (resistive RAM; RRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중의 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (33)는 비디오 인코더 (20)의 다른 컴포넌트들과 온-칩 (on-chip) 일 수도 있거나, 그 컴포넌트들에 관하여 오프-칩 (off-chip) 일 수도 있다.

[0176] 파티셔닝 유닛 (35)은 비디오 데이터 메모리 (33)로부터 비디오 데이터를 취출하고, 비디오 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한, 슬라이스 (slice)들, 타일 (tile)들, 또는 다른 더 큰 유닛들로의 파티셔닝뿐만 아니라, 예컨대, LCU들 및 CU들의 큐드트리 구조에 따른 비디오 블록 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20)는 인코딩되어야 할 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 일반적으로 예시한다. 슬라이스는 다수의 비디오 블록들로 (그리고 아마도 타일들로서 지정된 비디오 블록들의 세트들로) 분할될 수도 있다. 예측 프로세싱 유닛 (41)은 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨)에 기초하여, 현재의 비디오 블록에 대하여, 복수의 인트라 코딩 모드들 중의 하나 또는 복수의 인터 코딩 모드들 중의 하나와 같은 복수의 가능한 코딩 모드들 중의 하나를 선택할 수도 있다. 예측 프로세싱 유닛 (41)은 결과적인 인트라-코딩된 또는 인터-코딩된 블록을, 잔차 블록 데이터를 생성하기 위하여 합산기 (50)에, 그리고 참조 픽처로서의 이용을 위한 인코딩된 블록을 복원하기 위하여 합산기 (62)에 제공할 수도 있다.

[0177] 예측 프로세싱 유닛 (41) 내의 인트라 예측 유닛 (46)은 공간적 압축을 제공하기 위하여, 코딩되어야 할 현재의 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 관한 현재의 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다. 예측 프로세싱 유닛 (41) 내의 인터 예측 프로세싱 유닛 (44)은 시간적 압축을 제공하기 위하여, 하나 이상의 참조 픽처들에서의 하나 이상의 예측 블록들에 관한 현재의 비디오 블록의 인터-예측 코딩을 수행한다.

[0178] 인터 예측 프로세싱 유닛 (44)은 비디오 시퀀스에 대한 미리 결정된 패턴에 따라 비디오 슬라이스에 대한 인터-예측 모드를 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들 또는 B 슬라이스들로서 지정할 수도 있다. 인터 예측 프로세싱 유닛 (44)에 의해 수행된 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 예를 들어, 모션 벡터는 참조 픽처 내의 예측 블록에 관하여 현재의 비디오 프레임 또는 픽처 내의 비디오 블록의 PU의 변위를 표시할

수도 있다.

[0179]

예측 블록은, 절대차의 합 (sum of absolute difference; SAD), 제곱차의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는 픽셀 차이의 측면에서, 코딩되어야 할 비디오 블록의 PU 와 근접하게 일치시키기 위하여 구해지는 블록이다. 일부 예들에서, 비디오 인코더 (20) 는 DPB (66) 내에 저장된 참조 픽처들의 정수-미만 (sub-integer) 픽셀 위치들에 대한 값을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 픽처의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값을 보간할 수도 있다. 그러므로, 인터 예측 프로세싱 유닛 (44) 은 전체 픽셀 위치들 및 분수 픽셀 위치들에 관하여 모션 검색을 수행할 수도 있고, 분수 픽셀 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0180]

인터 예측 프로세싱 유닛 (44) 은 PU 의 위치를 참조 픽처의 예측 블록의 위치와 비교함으로써, 인터-코딩된 슬라이스에서의 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 참조 픽처는 제 1 참조 픽처 리스트 (List 0) 또는 제 2 참조 픽처 리스트 (List 1) 로부터 선택될 수도 있고, 이들의 각각은 DPB (66) 내에 저장된 하나 이상의 참조 픽처들을 식별한다. 인터 예측 프로세싱 유닛 (44) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 으로 전송한다.

[0181]

인터 예측 프로세싱 유닛 (44) 에 의해 수행된 모션 보상은 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페치 (fetch) 하거나 생성하여, 서브-픽셀 정밀도 (sub-pixel precision) 로의 보간들을 아마도 수행하는 것을 수반할 수도 있다. 현재의 비디오 블록의 PU 에 대한 모션 벡터를 결정할 시에, 인터 예측 프로세싱 유닛 (44) 은 모션 벡터가 참조 픽처 리스트들 중의 하나에서 지시하는 예측 블록을 위치시킬 수도 있다. 비디오 인코더 (20) 는 코딩되고 있는 현재의 비디오 블록의 픽셀 값을로부터 예측 블록의 픽셀 값을 감산하여 픽셀 차이 값을 형성함으로써 잔차 비디오 블록을 형성한다. 픽셀 차이 값을 블록에 대한 잔차 데이터를 형성하고, 루마 및 크로마 차이 컴포넌트들의 양자를 포함할 수도 있다. 합산기 (50) 는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 표현한다. 인터 예측 프로세싱 유닛 (44) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 이용을 위한 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0182]

예측 프로세싱 유닛 (41) 은 또한, 이 개시물에서 설명된 팔레트-기반 코딩 기법들의 다양한 양태들을 수행하도록 구성된 팔레트-기반 인코딩 유닛 (42) 을 포함한다. 팔레트-기반 인코딩 유닛 (42) 은 예를 들어, 팔레트-기반 인코딩 모드가 예컨대, CU 또는 PU 와 같은 블록에 대하여 선택될 때에 팔레트-기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛 (42) 은, 픽셀 값을 표시하는 엔트리들을 가지는 팔레트를 생성하고, 비디오 데이터의 블록의 적어도 일부의 위치들의 픽셀 값을 나타내기 위하여 팔레트에서 픽셀 값을 선택하고, 비디오 데이터의 블록의 위치들 중의 적어도 일부를, 선택된 픽셀 값을 각각 대응하는 팔레트에서의 엔트리들과 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 인코딩 유닛 (42) 에 의해 수행되는 것으로서 설명되지만, 이러한 기능들의 일부 또는 전부는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0183]

팔레트-기반 인코딩 유닛 (42) 은 본원에서 설명된 다양한 신택스 엘리먼트들 중의 임의의 것을 생성하도록 구성될 수도 있다. 따라서, 비디오 인코더 (20) 는 이 개시물에서 설명된 바와 같은 팔레트-기반 코드 모드들을 이용하여 비디오 데이터의 블록들을 인코딩하도록 구성될 수도 있다. 비디오 인코더 (20) 는 선택적으로, 팔레트 코딩 모드를 이용하여 비디오 데이터의 블록을 인코딩할 수도 있거나, 또는 예컨대, HEVC 인터-예측 또는 인트라-예측 코딩 모드와 같은 상이한 모드를 이용하여 비디오 데이터의 블록을 인코딩할 수도 있다. 비디오 데이터의 블록은 예를 들어, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU 일 수도 있다. 비디오 인코더 (20) 는 인터-예측 시간적 예측 또는 인트라-예측 공간적 코딩 모드들로 일부 블록들을 인코딩 할 수도 있고, 팔레트-기반 코딩 모드로 다른 블록들을 디코딩할 수도 있다.

[0184]

예측 프로세싱 유닛 (41) 이 인트라 예측, 인터 예측, 또는 팔레트 모드 코딩의 어느 하나를 통해 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 인코더 (20) 는 현재의 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 그러나, 일부 코딩 모드들에 대하여, 비디오 인코더 (20) 는 잔차 블록을 인코딩하지 않을 수도 있다. 생성될 때, 잔차 블록에서의 잔차 비디오 데이터는 하나 이상의 TU 들 내에 포함될 수도 있고 변환 프로세싱 유닛 (52) 에 적용될 수도 있다. 변환 프로세싱 유닛 (52) 은 이산 코사인 변환 (discrete cosine transform; DCT) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 잔차 비디오 데이터를 잔차 변환 계수들로 변환한다. 변환 프로세싱 유닛 (52) 은 잔차 비디오 데이터를 픽셀 도메인으로부터, 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다.

- [0185] 변환 프로세싱 유닛 (52)은 결과적인 변환 계수들을 양자화 유닛 (54)으로 전송할 수도 있다. 양자화 유닛 (54)은 비트 레이트를 추가로 감소시키기 위하여 변환 계수들을 양자화한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화도는 양자화 파라미터를 조절함으로써 수정될 수도 있다. 일부 예들에서, 그 다음으로, 양자화 유닛 (54)은 양자화된 변환 계수들을 포함하는 행렬의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56)은 스캔을 수행할 수도 있다.
- [0186] 양자화에 후속하여, 엔트로피 인코딩 유닛 (56)은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56)은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 2 진 산술 코딩 (CABAC), 신팩스-기반 컨텍스트-적응 2 진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56)에 의한 엔트로피 인코딩에 후속하여, 인코딩된 비트스트림은 비디오 디코더 (30)로 송신될 수도 있거나, 비디오 디코더 (30)에 의한 더 이후의 송신 또는 취출을 위하여 아카이빙될 수도 있다. 엔트로피 인코딩 유닛 (56)은 또한, 코딩되고 있는 현재의 비디오 슬라이스에 대한 모션 벡터들, 및 위에서 논의된 다양한 팔레트 모드 신팩스 엘리먼트들을 포함하는 다른 신팩스 엘리먼트들을 엔트로피 인코딩할 수도 있다.
- [0187] 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60)은 참조 픽처의 참조 블록으로서의 더 이후의 이용을 위해 픽셀 도메인에서 잔차 블록을 복원하기 위하여, 역양자화 및 역변환을 각각 적용한다. 인터 예측 프로세싱 유닛 (44)은 잔차 블록을 참조 픽처 리스트들 중의 하나 내의 참조 픽처들 중의 하나의 참조 픽처의 예측 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 인터 예측 프로세싱 유닛 (44)은 또한, 모션 추정에서의 이용을 위한 정수-미만 픽셀 값들을 계산하기 위하여 하나 이상의 보간 필터들을 복원된 잔차 블록에 적용할 수도 있다. 합산기 (62)는 복원된 블록을 생성하기 위하여, 복원된 잔차 블록을 인터 예측 프로세싱 유닛 (44)에 의해 생성된 모션 보상된 예측 블록에 가산한다.
- [0188] 필터 유닛 (64)은 복원된 블록 (예컨대, 합산기 (62)의 출력)을 필터링하고, 필터링된 복원된 블록을 참조 블록에서의 이용들을 위하여 DPB (66) 내에 저장한다. 참조 블록은 후속 비디오 프레임 또는 픽처에서 블록을 인터-예측하기 위하여, 인터 예측 프로세싱 유닛 (44)에 의해 참조 블록으로서 이용될 수도 있다. 필터 유닛 (64)은 디블록킹 필터링, 샘플 적응적 오프셋 (sample adaptive offset; SAO) 필터링, 적응적 루프 필터링 (adaptive loop filtering; ALF), 또는 다른 타입들의 루프 필터링 중의 하나 이상을 적용할 수도 있다. 필터 유닛 (64)은 복원된 비디오로부터 블록화 아티팩트 (blockiness artifact)들을 제거하기 위하여 디블록킹 필터링을 필터 블록 경계들에 적용할 수도 있고, 전체적인 코딩 품질을 개선시키기 위하여 다른 타입들의 필터링을 적용할 수도 있다. (루프 내의 또는 루프 이후의) 추가적인 루프 필터들이 또한 이용될 수도 있다.
- [0189] 도 5는 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 디코더 (30)를 예시하는 블록도이다. 도 5의 예에서, 비디오 디코더 (30)는 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역양자화 유닛 (86), 역변환 프로세싱 유닛 (88), 합산기 (90), 및 DPB (94)를 포함한다. 예측 프로세싱 유닛 (81)은 팔레트-기반 디코딩 유닛 (82), 모션 보상 유닛 (motion compensation unit; MCU) (83), 및 인트라 예측 유닛 (84)을 포함한다. 일부 예들에서, 비디오 디코더 (30)는 도 4로부터의 비디오 인코더 (20)에 대하여 설명된 인코딩 패스와 일반적으로 상반적인 디코딩 패스를 수행할 수도 있다.
- [0190] 디코딩 프로세스 동안, 비디오 디코더 (30)는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신팩스 엘리먼트들을 표현하는 인코딩된 비디오 비트스트림을 비디오 인코더 (20)로부터 수신한다. 비디오 디코더 (30)는 수신된 인코딩된 비디오 비트스트림을 비디오 데이터 메모리 (78) 내에 저장한다. 비디오 데이터 메모리 (78)는 비디오 디코더 (30)의 컴포넌트들에 의해 디코딩되어야 할, 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (78) 내에 저장된 비디오 데이터를 예를 들어, 링크 (16)를 통해, 저장 디바이스 (26)로부터, 또는 카메라와 같은 로컬 비디오 소스로부터, 또는 물리적 데이터 저장 매체들을 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (78)는 인코딩된 비디오 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (coded picture buffer; CPB)를 형성할 수도 있다. DPB (94)는 예컨대, 인트라-코딩 또는 인터-코딩 모드들에서 비디오 디코더 (30)에 의해 비디오 데이터를 디코딩함에 있어서의 이용을 위한 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94)는 DRAM, SDRAM, MRAM, RRAM, 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중의 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (78)는 비디오 디코더 (30)의 다른 컴포넌트들과 함께 온-칩일 수

도 있거나, 그 컴포넌트들에 관하여 오프-칩일 수도 있다.

[0191] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 양자화된 계수들, 모션 벡터들, 및 다른 십екс 엘리먼트들을 생성하기 위하여 비디오 데이터 메모리 (78) 내에 저장된 비디오 데이터를 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들, 및 위에서 논의된 다양한 팔레트 모드 십екс 엘리먼트들을 포함하는 다른 십екс 엘리먼트들을 예측 프로세싱 유닛 (81) 으로 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 십екс 엘리먼트들을 수신할 수도 있다.

[0192] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81) 의 인트라 예측 유닛 (84) 은 시그널링된 인트라 예측 모드와, 현재의 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B 또는 P) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81) 의 모션 보상 유닛 (83) 은 모션 벡터들과, 엔트로피 디코딩 유닛 (80) 으로부터 수신된 다른 십екс 엘리먼트들에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 픽처 리스트들 중의 하나 내의 참조 픽처들 중의 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는 DPB (94) 내에 저장된 참조 픽처들에 기초한 디폴트 구성 기법들을 이용하여, 참조 프레임 리스트들, List 0 및 List 1 을 구성할 수도 있다.

[0193] 모션 보상 유닛 (83) 은 모션 벡터들 및 다른 십екс 엘리먼트들을 파싱함으로써 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 디코딩되고 있는 현재의 비디오 블록에 대한 예측 블록들을 생성하기 위하여 예측 정보를 이용한다. 예를 들어, 모션 보상 유닛 (83) 은 비디오 슬라이스의 비디오 블록들을 코딩하기 위해 이용된 예측 모드 (예컨대, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 타입 (예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 픽처 리스트들 중의 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재의 비디오 슬라이스에서의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위하여, 수신된 십екс 엘리먼트들의 일부를 이용한다.

[0194] 모션 보상 유닛 (83) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (83) 은 참조 블록들의 정수-미만 픽셀들에 대한 보간된 값들을 계산하기 위하여, 비디오 블록들의 인코딩 동안에 비디오 인코더 (20) 에 의해 이용된 바와 같은 보간 필터들을 이용할 수도 있다. 이 경우, 모션 보상 유닛 (83) 은 수신된 십екс 엘리먼트들로부터 비디오 인코더 (20) 에 의해 이용된 보간 필터들을 결정할 수도 있고, 예측 블록들을 생성하기 위하여 보간 필터들을 이용할 수도 있다.

[0195] 역양자화 유닛 (86) 은, 비트스트림에서 제공되며 엔트로피 디코딩 유닛 (80) 에 의해 디코딩된 양자화된 변환 계수들을 역양자화, 즉, 텔양자화 (de-quantize) 한다. 역양자화 프로세스는 적용되어야 할 양자화도 및, 마찬가지로, 역양자화도를 결정하기 위하여 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 이용을 포함할 수도 있다. 역변환 프로세싱 유닛 (88) 은 픽셀 도메인 (pixel domain) 에서 잔차 블록들을 생성하기 위하여, 역변환, 예컨대, 역 DCT, 역정수 변환 (inverse integer transform), 또는 개념적으로 유사한 역변환 프로세스를 변환 계수들에 적용한다.

[0196] 이 개시물의 다양한 예들에 따르면, 비디오 디코더 (30) 는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다. 팔레트-기반 디코딩 유닛 (82) 은 예를 들어, 팔레트-기반 디코딩 모드가 CU 또는 PU 와 같은 블록에 대하여 선택될 때에 팔레트-기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (82) 은 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 생성하도록 구성될 수도 있다. 또한, 이 예에서, 팔레트-기반 디코딩 유닛 (82) 은 비디오 데이터의 블록의 적어도 일부의 위치들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신할 수도 있다. 이 예에서, 팔레트-기반 디코딩 유닛 (82) 은 정보에 기초하여 팔레트에서의 픽셀 값들을 선택할 수도 있다. 추가적으로, 이 예에서, 팔레트-기반 디코딩 유닛 (82) 은 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 복원할 수도 있다. 다양한 기능들이 팔레트-기반 디코딩 유닛 (82) 에 의해 수행되는 것으로서 설명되지만, 이러한 기능들의 일부 또는 전부는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0197] 팔레트-기반 디코딩 유닛 (82) 은 팔레트 코딩 모드 정보를 수신할 수도 있고, 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용된다는 것을 표시할 때에 상기 동작들을 수행할 수도 있다. 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용되지 않는다는 것을 표시할 때, 또는 다른 모드 정보가 상이한 모드의 이용을 표시할 때, 팔레트-기반 디코딩 유닛 (82) 은 예컨대, HEVC 인터-예측 또는 인트라-예측 코딩 모드와 같은

비-팔레트 기반 코딩 모드를 이용하여 비디오 데이터의 블록을 디코딩한다. 비디오 데이터의 블록은 예를 들어, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU 일 수도 있다. 비디오 디코더 (30)는 인터-예측 시간적 예측 또는 인트라-예측 공간적 코딩 모드들로 일부의 블록들을 디코딩할 수도 있고, 팔레트-기반 코딩 모드로 다른 블록들을 디코딩할 수도 있다. 팔레트-기반 코딩 모드는 복수의 상이한 팔레트-기반 코딩 모드들 중의 하나를 포함할 수도 있거나, 단일 팔레트-기반 코딩 모드가 있을 수도 있다.

[0198] 예측 프로세싱 유닛이 예를 들어, 인트라 또는 인터 예측을 이용하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30)는 역변환 프로세싱 유닛 (88)으로부터의 잔차 블록들을 모션 보상 유닛 (83)에 의해 생성된 대응하는 예측 블록들과 합산함으로써 복원된 비디오 블록을 형성한다. 합산기 (90)는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 표현한다. 필터 유닛 (92)은 예를 들어, 디블록킹 필터링, SAO 필터링, ALF 필터링, 또는 다른 타입들의 필터링 중의 하나 이상을 이용하여 복원된 비디오 블록을 필터링한다. (코딩 루프 내에 또는 코딩 루프 이후 중의 어느 하나에서의) 다른 루프 필터들은 또한, 픽셀 천이 (pixel transition)들을 평탄화하거나, 또는 이와 다르게 비디오 품질을 개선시키기 위하여 이용될 수도 있다. 그 다음으로, 소정의 프레임 또는 픽처에서의 디코딩된 비디오 블록들은, 후속 모션 보상을 위하여 이용된 참조 픽처들을 저장하는 DPB (94) 내에 저장된다. DPB (94)는 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에서의 제시를 위하여 디코딩된 비디오를 저장하는 추가적인 메모리의 일부일 수도 있거나, 이로부터 별도일 수도 있다.

[0199] 비디오 디코더 (30)는 예를 들어, 신팩스 엘리먼트를 수신하고 신팩스 엘리먼트에 따라 팔레트 예측자 초기화 프로세스를 수행하도록 구성될 수도 있다. 신팩스 엘리먼트의 제 1 값은 예를 들어, 팔레트 예측자 초기화가 이용된다는 것을 표시할 수도 있고, 신팩스 엘리먼트의 제 2 값은 팔레트 예측자 크기가 0과 동일하게 설정된다는 것을 표시한다. 신팩스 엘리먼트는 예를 들어, SPS 또는 PPS에서 시그널링될 수도 있다.

[0200] 추가적으로 또는 대안적으로, 비디오 디코더 (30)는 PPS에서, 제 1 팔레트 예측자 초기화자 신팩스 엘리먼트를 수신하고, SPS에서, 제 2 팔레트 예측자 초기화자 신팩스 엘리먼트를 수신하도록 구성될 수도 있다. 제 1 팔레트 예측자 초기화자 신팩스 엘리먼트 및 제 2 팔레트 예측자 초기화자 신팩스 엘리먼트에 기초하여, 비디오 디코더 (30)는 팔레트 예측자를 생성할 수도 있다. 팔레트 예측자를 생성하기 위하여, 비디오 디코더 (30)는 팔레트 예측자 내에, SPS에서 식별된 팔레트 엔트리들과, 그 다음으로, PPS에서 식별된 팔레트 엔트리들, 또는 그 반대 순서로 포함할 수도 있다. 팔레트 예측자를 생성하기 위하여, 비디오 디코더 (30)는 추가적으로 또는 대안적으로, 팔레트 예측자의 엔트리마다 하나 이상의 재이용 플래그들을 수신할 수도 있다.

[0201] 추가적으로 또는 대안적으로, 비디오 디코더 (30)는 PPS에서, 제 1 스케일링 리스트를 수신하고, SPS에서, 제 2 스케일링 리스트를 수신하도록 구성될 수도 있다. 제 1 스케일링 리스트 및 제 2 스케일링 리스트에 기초하여, 비디오 디코더 (30)는 픽처에 대한 스케일링 리스트를 생성할 수도 있다.

[0202] 추가적으로 또는 대안적으로, 비디오 디코더 (30)는 제 1 파라미터 세트에서, 팔레트 예측자 초기화자 엔트리들의 제 1 세트를 수신하도록 구성될 수도 있다. 제 1 파라미터 세트를 수신하는 것에 후속하여, 비디오 디코더 (30)는 제 2 파라미터 세트에서, 팔레트 예측자 초기화자 엔트리들의 제 2 세트를 수신할 수도 있다. 비디오 디코더 (30)는 팔레트 예측자 초기화자 엔트리들의 제 1 세트의 적어도 부분을 이용하여 팔레트 예측자 초기화를 수행할 수도 있다. 팔레트 예측자 초기화를 수행하기 위하여, 비디오 디코더 (30)는 예를 들어, 팔레트 예측자 초기화자 엔트리들의 제 1 세트의 부분에 추가하여, 팔레트 예측자 초기화자 엔트리들의 제 2 세트의 적어도 부분을 이용할 수도 있다. 일부 예들에서, 제 1 파라미터 세트는 PPS 일 수도 있고, 제 2 파라미터 세트는 SPS 일 수도 있다.

[0203] 추가적으로 또는 대안적으로, 비디오 디코더 (30)는 또한, 팔레트 예측자 초기화자 엔트리들의 제 1 세트의 부분을 이용하여 팔레트 예측자 초기화자를 형성하고, 업데이트된 팔레트 예측자 초기화자를 형성하기 위하여 팔레트 예측자 초기화자 엔트리들의 제 2 세트의 부분을 팔레트 예측자 초기화자에 추가하도록 구성될 수도 있다.

업데이트된 팔레트 예측자 초기화자를 형성하기 위하여 팔레트 예측자 초기화자 엔트리들의 제 2 세트의 부분을 팔레트 예측자 초기화자에 추가하기 위해서는, 비디오 디코더 (30)가 업데이트된 팔레트 예측자 초기화자를 형성하기 위하여 팔레트 예측자 초기화자의 시작부에서 팔레트 예측자 초기화자 엔트리들의 제 2 세트의 부분을 추가하도록 구성될 수도 있다.

[0204] 업데이트된 팔레트 예측자 초기화자를 형성하기 위하여 팔레트 예측자 초기화자 엔트리들의 제 2 세트의 부분을 팔레트 예측자 초기화자에 추가하기 위해서는, 비디오 디코더 (30)가 업데이트된 팔레트 예측자 초기화자를 형성하기 위하여 팔레트 예측자 초기화자의 종료부에서 팔레트 예측자 초기화자 엔트리들의 제 2 세트의 부분을

추가하도록 구성될 수도 있다. 비디오 디코더 (30)는 또한, 팔레트 예측자 엔트리들의 제 2 세트의 부분의 전체를 추가하는 것이 초기화자 크기 제한을 초과하는 것으로 결정하고, 결정에 기초하여, 업데이트된 팔레트 예측자 초기화자로부터 팔레트 예측자 엔트리들의 제 2 세트의 부분의 하나 이상의 엔트리들을 제외하도록 구성될 수도 있다. 업데이트된 팔레트 예측자 초기화자는 팔레트 예측자 엔트리들의 제 1 세트의 부분의 서브-부분 및 팔레트 예측자 엔트리들의 제 2 세트의 부분의 서브-부분의 양자를 포함할 수도 있다.

[0205] 도 6 은 이 개시물의 기법들에 따라, 비디오 인코더의 일 예의 동작을 예시하는 플로우차트이다. 도 6 의 플로우차트는 예로서 제공된다. 다른 예들에서, 플로우차트들은 더 많거나, 더 적거나, 또는 상이한 단계들을 포함할 수도 있다. 도 6 에서 설명된 동작은 위에서 설명된 비디오 인코더 (20) 와 같은 비디오 인코더에 의해 수행될 수도 있지만, 도 6 의 동작은 임의의 특정한 타입의 비디오 인코더로 제한되지는 않는다.

[0206] 도 6 의 예에서, 비디오 인코더는 비디오 데이터를 인코딩한다. 비디오 인코더는 인코딩된 비디오 데이터의 비트스트림에서의 SPS 싱택스 구조에서의 포함을 위하여, 제 1 싱택스 엘리먼트를 생성한다 (100). 제 1 싱택스 엘리먼트는 예를 들어, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 sps\_palette\_predictor\_initializer\_present\_flag 과 같은 1-비트 플래그일 수도 있다. 비디오 인코더는 SPS 싱택스 구조에서의 포함을 위하여, 하나 이상의 SPS-레벨 팔레트 예측자들을 생성한다 (102). 비디오 인코더는 인코딩된 비디오 데이터의 비트스트림에서의 PPS 싱택스 구조에서의 포함을 위하여, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 2 싱택스 엘리먼트를 생성한다 (104). 제 2 싱택스 엘리먼트는 예를 들어, 위에서 설명된 pps\_palette\_predictor\_initializer\_present\_flag 와 같은 1-비트 플래그일 수도 있다. 제 2 싱택스 엘리먼트는 팔레트 예측자가 SPS-레벨 팔레트 예측자 초기화자들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시할 수도 있다. 비디오 인코더는 PPS 싱택스 구조에서의 포함을 위하여, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 3 싱택스 엘리먼트를 생성한다 (106). 제 3 싱택스 엘리먼트는 예를 들어, 위에서 설명된 pps\_num\_palette\_predictor\_initializer 싱택스 엘리먼트일 수도 있다. 비디오 인코더는 또한, SPS 싱택스 구조에서의 포함을 위하여, SPS 내에 포함된 SPS-레벨 팔레트 예측자 초기화자들의 수를 표시하는 제 4 싱택스 엘리먼트를 생성할 수도 있다. 제 4 싱택스 엘리먼트는 예를 들어, 위에서 설명된 싱택스 엘리먼트 pps\_num\_palette\_predictor\_initializer\_minus1 일 수도 있다.

[0207] 도 7 은 이 개시물의 기법들에 따라, 비디오 디코더의 일 예의 동작을 예시하는 플로우차트이다. 도 7 의 플로우차트는 예로서 제공된다. 다른 예들에서, 플로우차트들은 더 많거나, 더 적거나, 또는 상이한 단계들을 포함할 수도 있다. 도 7 에서 설명된 동작은 위에서 설명된 비디오 디코더 (30) 와 같은 비디오 디코더에 의해 수행될 수도 있지만, 도 7 의 동작은 임의의 특정한 타입의 비디오 디코더로 제한되지는 않는다.

[0208] 도 7 의 예에서, 비디오 디코더는 비디오 데이터를 디코딩한다. 비디오 디코더는 SPS 에서, 팔레트 예측자 초기화가 인에이블된다는 것을 표시하는 제 1 싱택스 엘리먼트를 수신한다 (120). 제 1 싱택스 엘리먼트는 예를 들어, 위에서 설명된 싱택스 엘리먼트 sps\_palette\_predictor\_initializer\_present\_flag 와 같은 1-비트 플래그일 수도 있다. 비디오 디코더는 SPS 에서, 하나 이상의 SPS-레벨 팔레트 예측자 초기화자들을 수신한다 (122). 비디오 디코더는 PPS 에서, 팔레트 예측자가 PPS-레벨 팔레트 예측자 초기화자들을 이용하여 생성되어야 한다는 것을 표시하는 제 2 싱택스 엘리먼트를 수신한다 (124). 제 2 싱택스 엘리먼트는 예를 들어, 위에서 설명된 싱택스 엘리먼트 pps\_palette\_predictor\_initializer\_present\_flag 와 같은 1-비트 플래그일 수도 있다. 제 2 싱택스 엘리먼트는 팔레트 예측자가 SPS-레벨 팔레트 예측자 초기화자들 중의 임의의 것을 이용하지 않으면서 생성되어야 한다는 것을 추가로 표시할 수도 있다. 비디오 디코더는 PPS 에서, PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시하는 제 3 싱택스 엘리먼트를 수신한다 (126). 제 3 싱택스 엘리먼트는 예를 들어, 위에서 설명된 싱택스 엘리먼트 pps\_num\_palette\_predictor\_initializer 일 수도 있다.

[0209] 제 3 싱택스 엘리먼트가 PPS 내에 포함된 PPS-레벨 팔레트 예측자 초기화자들의 수가 제로와 동일하다는 것을 표시할 때에 팔레트 테이블을 생성하기 위하여, 비디오 디코더는 하나 이상의 팔레트 테이블 엔트리 값들을 수신할 수도 있고, 팔레트 테이블 내에 하나 이상의 수신된 팔레트 테이블 엔트리 값들을 포함함으로써 팔레트 테이블을 생성할 수도 있다. 이에 따라, 팔레트 테이블에 대한 엔트리들의 일부 또는 전부를 결정하기 위하여 팔레트 예측자를 이용하여 팔레트를 생성하는 대신에, 비디오 디코더는 팔레트 테이블에 대한 모든 팔레트 엔트리들을 직접적으로 수신한다. 비디오 데이터의 블록의 샘플에 대하여, 비디오 디코더는 인덱스를 수신하고,

샘플에 대한 인덱스에 대응하는 팔레트 테이블 엔트리를 이용하여 비디오 데이터의 블록을 복원한다.

[0210] 도 8 은 이 개시물의 기법들에 따라, 비디오 디코더의 일 예의 동작을 예시하는 플로우차트이다. 도 8 의 플로우차트는 예로서 제공된다. 다른 예들에서, 플로우차트들은 더 많거나, 더 적거나, 또는 상이한 단계들을 포함할 수도 있다. 도 8 에서 설명된 동작은 위에서 설명된 비디오 디코더 (30) 와 같은 비디오 디코더에 의해 수행될 수도 있지만, 도 8 의 동작은 임의의 특정한 타입의 비디오 디코더로 제한되지는 않는다.

[0211] 도 8 의 예에서, 비디오 디코더는 비디오 데이터를 디코딩한다. 비디오 디코더는 SPS 에서 제 1 팔레트 예측자 초기화 데이터를 수신한다 (140). 제 1 팔레트 예측자 초기화 데이터에 기초하여, 비디오 디코더는 SPS 팔레트 예측자 초기화자들을 결정한다 (142). 비디오 디코더는 PPS 에서 제 2 팔레트 예측자 초기화 데이터를 수신한다 (144). 제 2 팔레트 예측자 초기화 데이터에 기초하여, 비디오 디코더는 PPS 팔레트 예측자 초기화자들을 결정한다 (146). 비디오 디코더는 SPS 팔레트 예측자 초기화자들 및 PPS 팔레트 예측자 초기화자들에 기초하여 팔레트 예측자를 생성한다 (148). 팔레트 예측자를 생성하기 위하여, 비디오 디코더는 예를 들어, 팔레트 예측자 내에 적어도 하나의 PPS 팔레트 예측자 엔트리를 포함할 수도 있고, 팔레트 예측자 내에 적어도 하나의 SPS 팔레트 예측자 엔트리를 포함할 수도 있다.

[0212] 비디오 디코더는 또한, 팔레트 예측자와 비디오 데이터의 블록에 대하여 이용되어야 할 팔레트와의 사이의 차이들을 표시하는 잔차 데이터를 수신할 수도 있다. 이에 따라, 팔레트 예측자 및 잔차 데이터에 기초하여, 비디오 디코더는 비디오 데이터의 블록에 대하여 이용되어야 할 팔레트를 결정할 수 있다.

[0213] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현될 경우, 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 컴퓨터-판독가능 매체를 통해 송신될 수도 있고, 하드웨어-기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터-판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터-판독가능 저장 매체들, 또는 예컨대, 통신 프로토콜에 따라 하나의 장소로부터 또 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이 방식으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비-일시적인 유형의 컴퓨터-판독가능 저장 매체들, 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 이 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독가능 매체를 포함할 수도 있다.

[0214] 제한이 아닌 예로서, 이러한 컴퓨터-판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 희망하는 프로그램 코드를 저장하기 위하여 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속은 컴퓨터-판독가능 매체로 적절하게 칭해진다. 예를 들어, 동축 케이블, 광섬유 케이블, 트위스트 페어 (twisted pair), 디지털 가입자 회선 (digital subscriber line; DSL), 또는 적외선, 라디오 (radio), 및 마이크로파 (microwave) 와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신될 경우, 동축 케이블, 광섬유 케이블, 트위스트 페어, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 반송파들, 신호들, 또는 다른 일시적 매체 (transient medium) 들을 포함하는 것이 아니라, 그 대신에, 비-일시적인, 유형의 저장 매체들에 관한 것이라는 것이 이해되어야 한다. 본원에서 이용된 바와 같은 디스크 (disk) 및 디스크 (disc) 는 컴팩트 디스크 (compact disc; CD), 레이저 디스크 (laser disc), 광학 디스크 (optical disc), 디지털 다기능 디스크 (digital versatile disc; DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (Blu-ray disc) 를 포함하고, 여기서 디스크 (disk) 들은 통상 데이터를 자기적으로 재생하는 반면, 디스크 (disc) 들은 데이터를 레이저들로 광학적으로 재생한다. 상기의 조합들은 컴퓨터-판독가능 매체들의 범위 내에 또한 포함되어야 한다.

[0215] 명령들은 하나 이상의 DSP 들, 범용 마이크로프로세서들, ASIC 들, FPGA 들, 또는 다른 동등한 통합된 또는 별도의 로직 회로부와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 이용된 바와 같은 용어 "프로세서" 는 상기한 구조, 또는 본원에서 설명된 기법들의 구현을 위하여 적당한 임의의 다른 구조 중의 임의의 것을 지칭할 수도 있다. 게다가, 일부 양태들에서는, 본원에서 설명된 기능성이 인코딩 및 디코딩을 위해 구성되거나 조합된 코덱 내에 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에서 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0216]

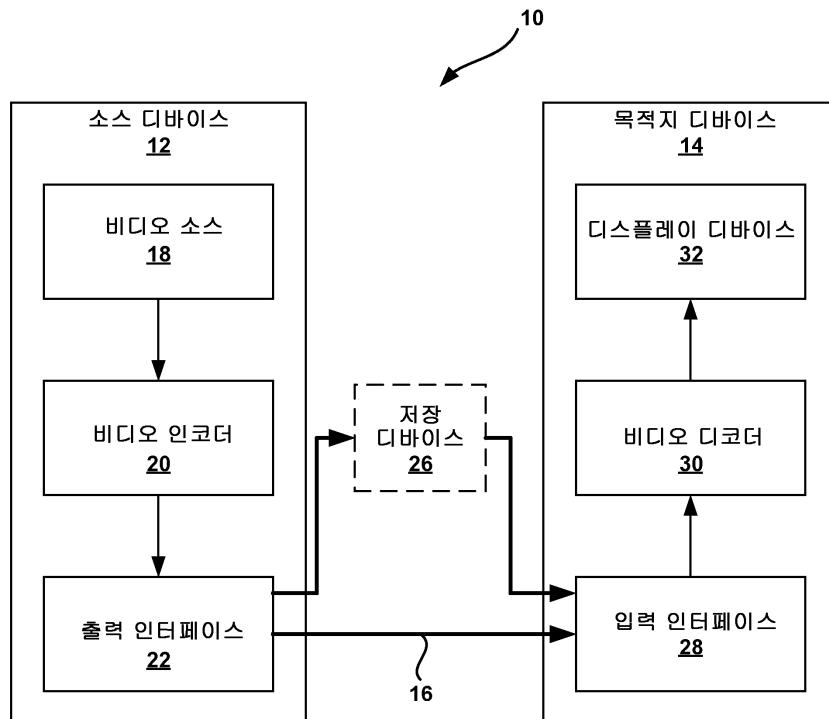
이 개시물의 기법들은 무선 핸드셋, 접적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩셋)를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위하여 이 개시물에서 설명되어 있지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛 내에 조합될 수도 있거나, 적당한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는 상호동작하는 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

[0217]

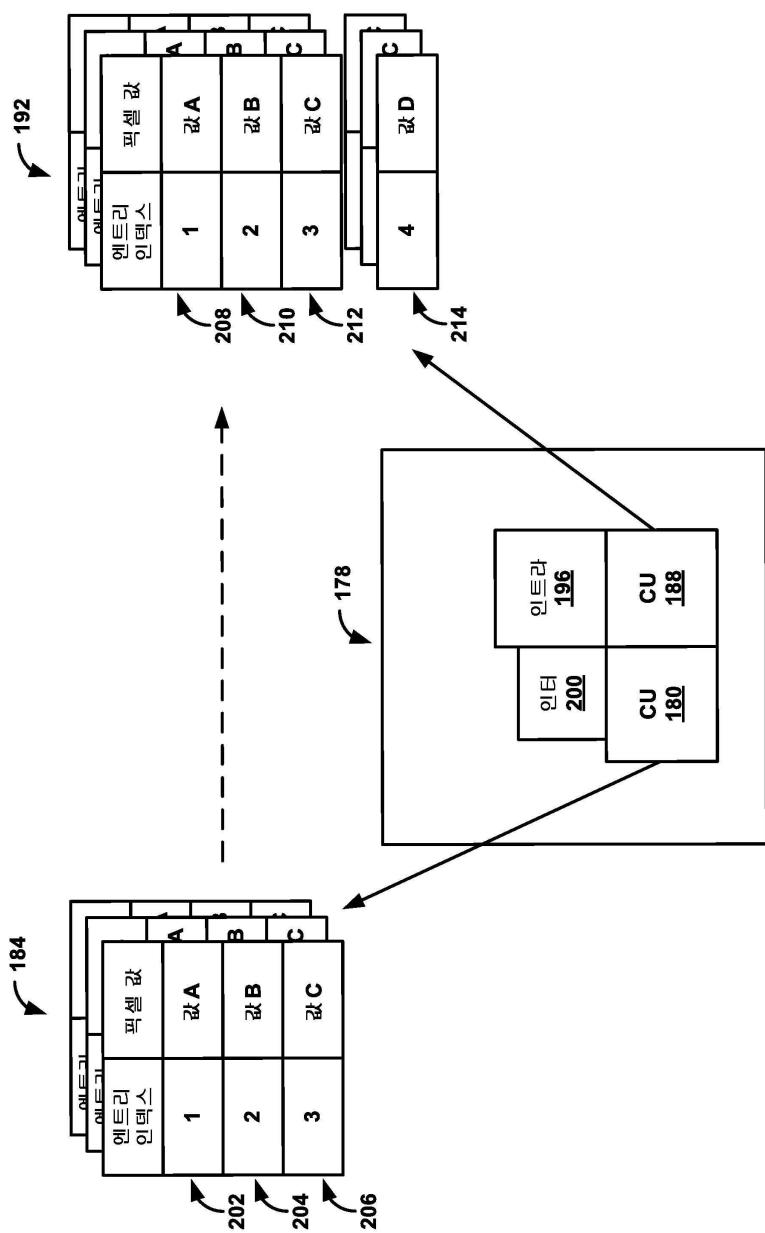
다양한 예들이 설명되었다. 이러한 그리고 다른 예들은 다음의 청구항들의 범위 내에 있다.

## 도면

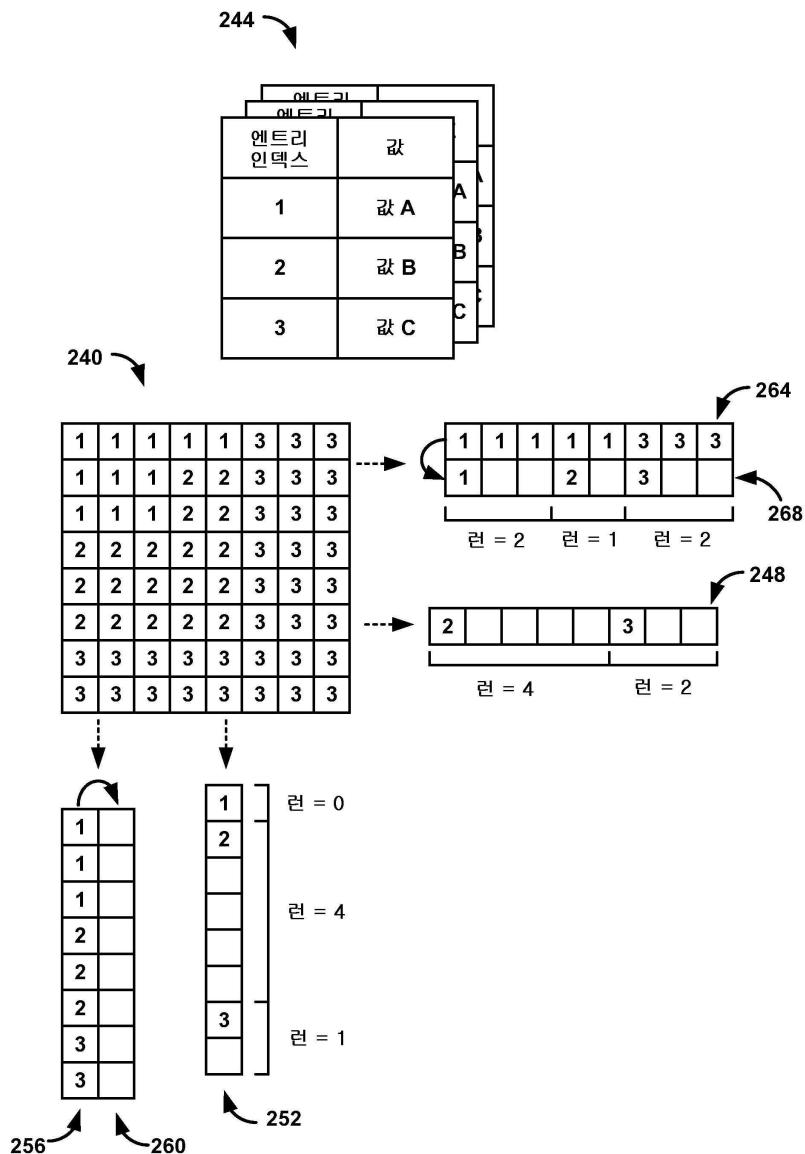
### 도면1



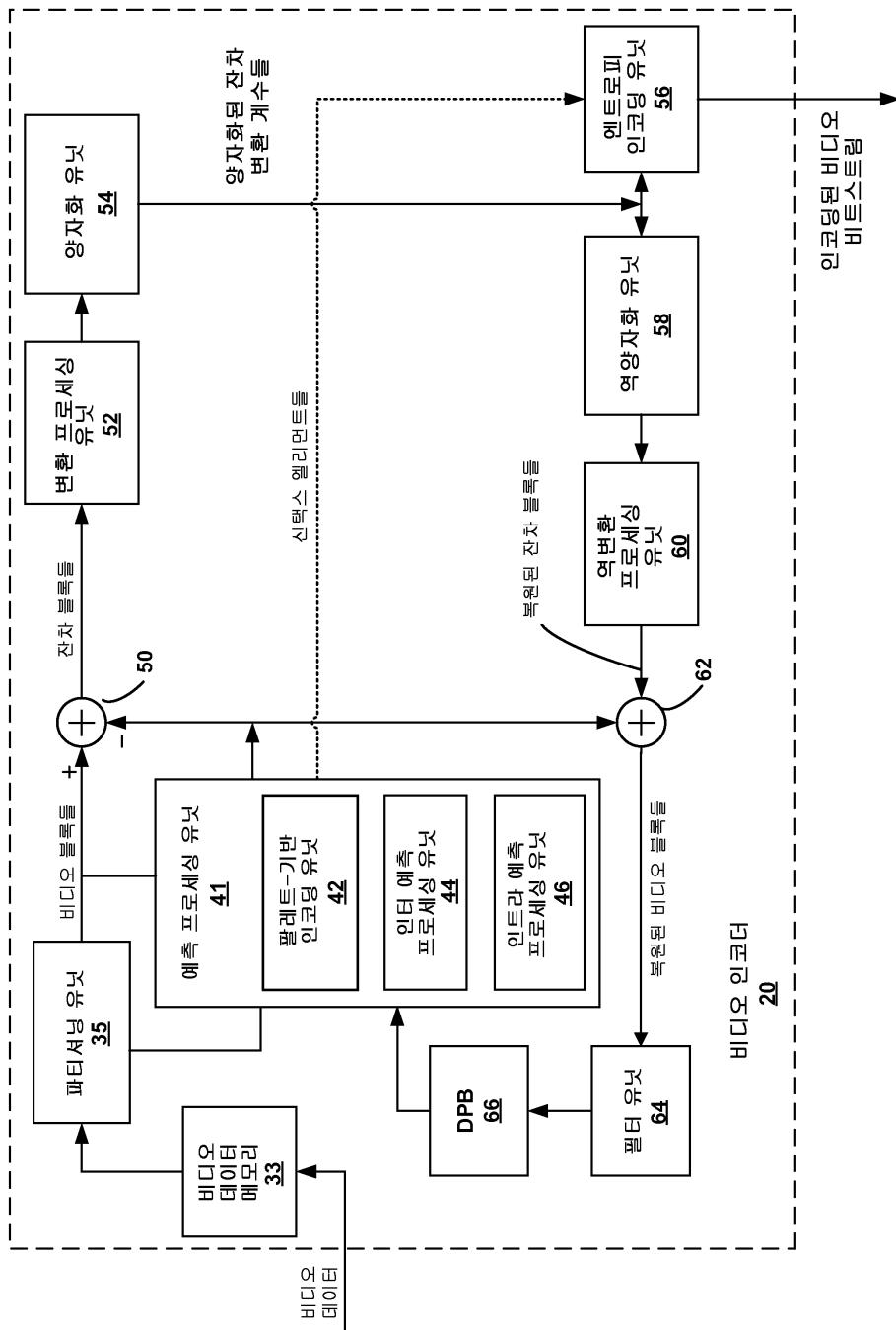
## 도면2



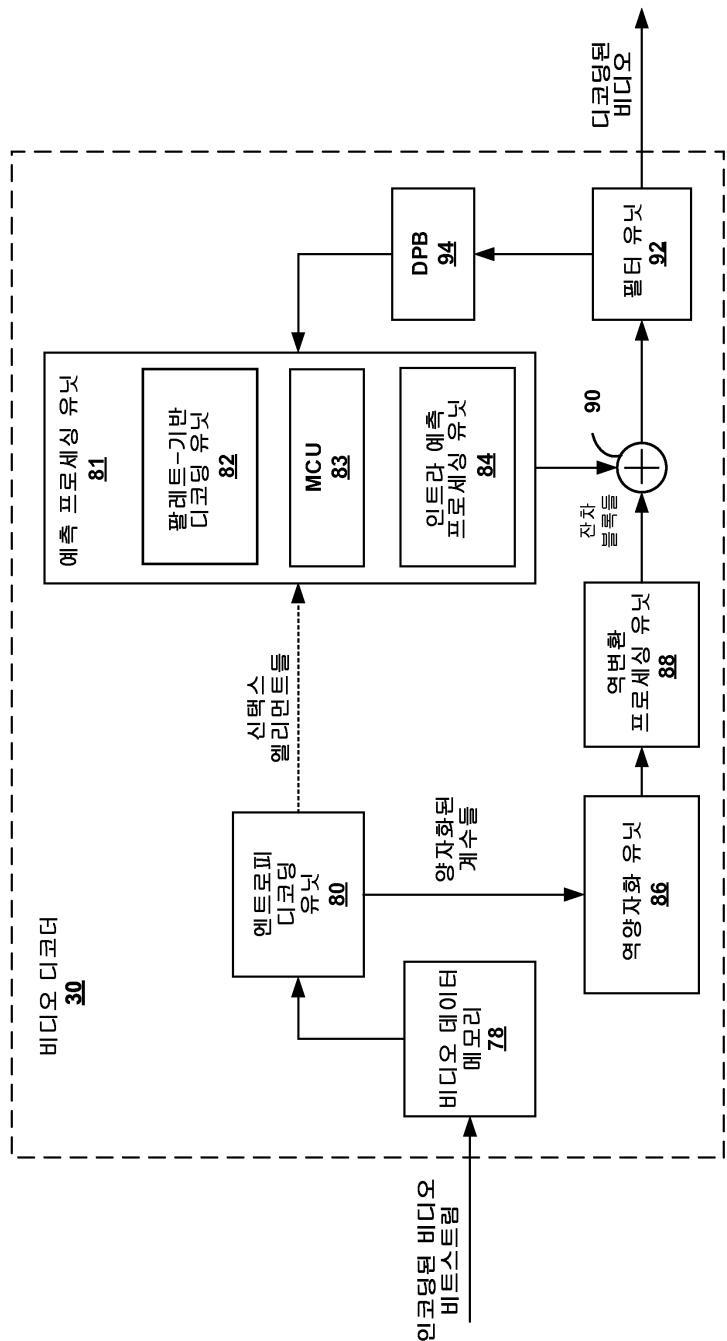
### 도면3



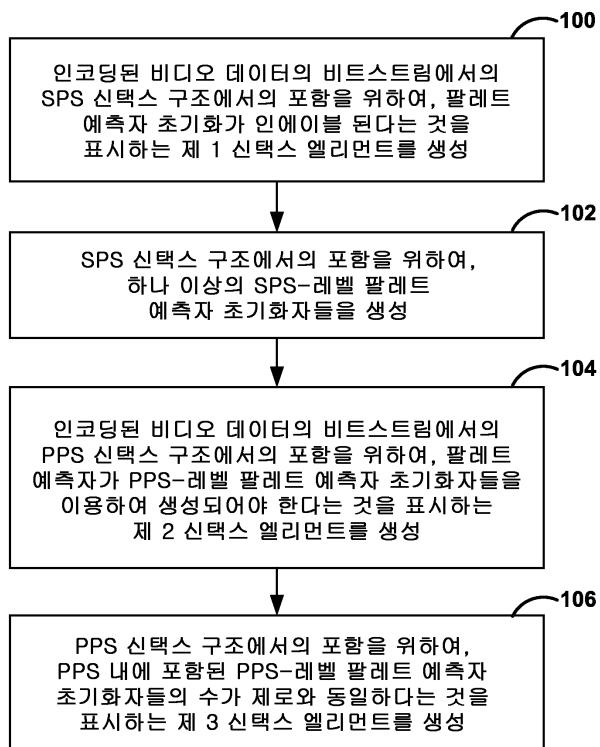
도면4



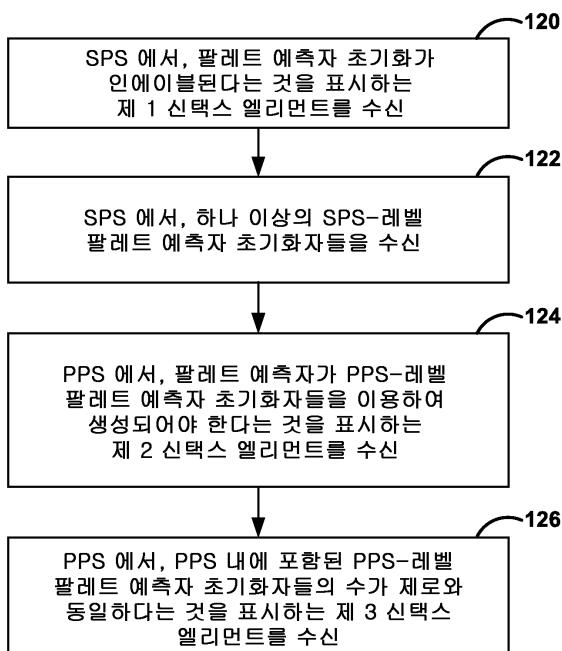
도면5



## 도면6



## 도면7



## 도면8

