



등록특허 10-2205328



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년01월19일
(11) 등록번호 10-2205328
(24) 등록일자 2021년01월14일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/46 (2014.01) *HO4N 19/119* (2014.01)
HO4N 19/50 (2014.01) *HO4N 19/90* (2014.01)
HO4N 19/93 (2014.01)
- (52) CPC특허분류
HO4N 19/46 (2015.01)
HO4N 19/119 (2015.01)
- (21) 출원번호 10-2015-7031433
- (22) 출원일자(국제) 2014년04월04일
심사청구일자 2019년03월19일
- (85) 번역문제출일자 2015년10월30일
- (65) 공개번호 10-2015-0140722
- (43) 공개일자 2015년12월16일
- (86) 국제출원번호 PCT/US2014/033013
- (87) 국제공개번호 WO 2014/165784
국제공개일자 2014년10월09일

(30) 우선권주장
61/809,236 2013년04월05일 미국(US)
(뒷면에 계속)

(56) 선행기술조사문헌
논문1(Zhaotai Pan 등, 2012)*
(뒷면에 계속)

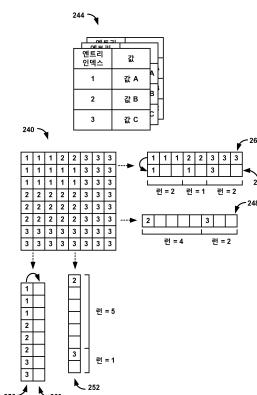
전체 청구항 수 : 총 28 항

심사관 : 조우연

(54) 발명의 명칭 팔레트-기반 비디오 코딩에서 팔레트 인덱스들의 결정

(57) 요 약

이 예에서, 비디오 데이터를 코딩하는 방법은, 비디오 데이터의 블록에서 제 1 픽셀과 연관된 제 1 인덱스 값을 결정하는 단계로서, 여기서 제 1 인덱스 값은 제 1 픽셀의 포지션을 픽셀 값들의 팔레트의 엔트리에 관련시키는, 상기 제 1 인덱스 값을 결정하는 단계, 제 1 인덱스 값에 기초하여, 비디오 데이터의 블록에서 하나 이상의 제 2 픽셀들과 연관된 하나 이상의 제 2 인덱스 값들을 결정하는 단계로서, 여기서 제 2 인덱스 값들은 하나 이상의 제 2 픽셀들의 포지션들을 픽셀 값들의 팔레트의 하나 이상의 엔트리들에 관련시키는, 상기 하나 이상의 제 2 인덱스 값을 결정하는 단계, 및 비디오 데이터의 블록의 제 1 픽셀 및 하나 이상의 제 2 픽셀들을 코딩하는 단계를 포함한다.

대 표 도 - 도5

(52) CPC특허분류

HO4N 19/50 (2015.01)

HO4N 19/90 (2015.01)

HO4N 19/93 (2015.01)

(72) 발명자

솔레 로할스 호텔

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

조쉬 라잔 택스맨

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(56) 선행기술조사문헌

KR101058010 B1*

서적1(JAMES D. MURRAY 등, 1996)*

US20030169932 A1

KR1020010095835 A

표준문서1(Xun Guo 등, 2013)

*는 심사관에 의하여 인용된 문헌

(30) 우선권주장

61/810,649 2013년04월10일 미국(US)

14/244,711 2014년04월03일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

비디오 데이터의 블록의 팔레트 인덱스들의 제 1 라인에 포함된 제 1 팔레트 인덱스에 대하여, 상기 제 1 팔레트 인덱스의 값을 코딩할지 또는 상기 팔레트 인덱스들의 제 1 라인과 상이한 상기 팔레트 인덱스들의 제 2 라인에 포함된 제 2 팔레트 인덱스를 카피할지의 여부를 결정하는 단계로서, 상기 제 1 팔레트 인덱스 및 상기 제 2 팔레트 인덱스는 개별적인 제 1 픽셀 포지션 및 제 2 픽셀 포지션을 상기 비디오 데이터의 블록에 대한 픽셀 값들의 팔레트들의 적어도 하나의 엔트리로 맵핑하는, 상기 코딩할지 또는 카피할지의 여부를 결정하는 단계;

상기 제 1 팔레트 인덱스의 값을 코딩할 것을 결정하는 것에 기초하여, 상기 제 1 팔레트 인덱스의 값을 표시하는 데이터를 코딩하는 단계; 및

상기 제 2 팔레트 인덱스를 카피할 것을 결정하는 것에 기초하여, 상기 제 1 라인의 제 1 팔레트 인덱스가 상기 제 2 라인의 제 2 팔레트 인덱스로부터 카피되는 것을 표시하는 데이터를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 팔레트 인덱스로 코딩하기 위한 스캔 방향에 있어서 상기 제 1 팔레트 인덱스를 뒤따르는 연속하는 인덱스들의 수의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 제 1 팔레트 인덱스의 값을 코딩할 것을 결정하는 것에 기초하여, 상기 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하는 단계는 상기 제 1 팔레트 인덱스와 동일한 연속하는 인덱스들의 수의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하는 단계를 포함하고,

상기 제 2 팔레트 인덱스를 카피할 것을 결정하는 것에 기초하여, 상기 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하는 단계는 상기 제 2 라인으로부터 카피된 연속하는 인덱스들의 수의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 비디오 데이터를 코딩하는 방법은,

상기 블록에서 하나 이상의 개별 픽셀 포지션들의 하나 이상의 픽셀 값들이 상기 픽셀 값들의 팔레트에 포함되지 않는 것을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 제 2 라인을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 6

제 5 항에 있어서,

상기 제 1 라인 및 상기 제 2 라인은 상기 비디오 데이터의 블록의 개별적인 제 1 로우와 제 2 로우 또는 개별적인 제 1 컬럼과 제 2 컬럼을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 7

제 5 항에 있어서,

상기 제 2 라인을 표시하는 하나 이상의 신택스 엘리먼트들을 코딩하는 단계는 상기 비디오 데이터의 블록에서 상기 제 2 라인의 위치를 식별하는 데이터를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 8

제 7 항에 있어서,

상기 제 2 라인의 위치를 식별하는 데이터를 코딩하는 단계는 상기 비디오 데이터의 블록에서 상기 제 2 라인을 식별하는 변위 값을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 9

제 7 항에 있어서,

상기 제 2 라인을 표시하는 하나 이상의 신택스 엘리먼트들을 코딩하는 단계는 1진 또는 절단된 1진 코드를 사용하여 상기 하나 이상의 신택스 엘리먼트들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 10

제 1 항에 있어서,

상기 제 1 팔레트 인덱스 및 상기 제 2 팔레트 인덱스를 포함하는 상기 비디오 데이터의 블록에 대한 인덱스들의 맵을 결정하는 단계; 및

상기 맵의 인덱스들을 코딩하기 위한 스캔 방향을 표시하는 데이터를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 11

제 1 항에 있어서,

상기 제 1 팔레트 인덱스의 값을 코딩할지 또는 상기 제 1 팔레트 인덱스에 대하여 상기 제 2 팔레트 인덱스를 카피할지 여부를 표시하는 하나 이상의 신택스 엘리먼트들을 코딩하는 단계를 더 포함하며,

상기 제 1 팔레트 인덱스의 값을 코딩할지 또는 상기 제 2 팔레트 인덱스를 카피할지 여부를 결정하는 것은 상기 하나 이상의 신택스 엘리먼트들에 기초하여 결정하는 것을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 12

제 1 항에 있어서,

코딩하는 단계는 디코딩하는 단계를 포함하고,

상기 비디오 데이터를 코딩하는 방법은,

상기 제 1 팔레트 인덱스의 픽셀 값을 상기 제 2 팔레트 인덱스의 픽셀 값을, 개별 제 1 팔레트 인덱스 및 제 2 팔레트 인덱스를 상기 팔레트의 적어도 하나의 엔트리와 매칭함으로써, 결정하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 13

제 1 항에 있어서,

코딩하는 단계는 인코딩하는 단계를 포함하고,

상기 인코딩하는 단계는 상기 제 1 팔레트 인덱스 및 상기 제 2 팔레트 인덱스를 나타내는 데이터를 인코딩된 비트스트림으로 인코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 14

비디오 데이터를 코딩하는 장치로서,

상기 비디오 데이터를 저장하는 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

비디오 데이터의 블록의 팔레트 인덱스들의 제 1 라인에 포함된 제 1 팔레트 인덱스에 대하여, 상기 제 1 팔레트 인덱스의 값을 코딩할지 또는 상기 팔레트 인덱스들의 제 2 라인에 포함된 제 2 팔레트 인덱스를 카피할지의 여부를 결정하는 것으로서, 상기 제 1 팔레트 인덱스 및 상기 제 2 팔레트 인덱스는 개별적인 제 1 픽셀 포지션 및 제 2 픽셀 포지션을 상기 비디오 데이터의 블록에 대한 픽셀 값들의 팔레트들의 적어도 하나의 엔트리로 맵핑하는, 상기 코딩할지 또는 카피할지의 여부를 결정하고;

상기 제 1 팔레트 인덱스의 값을 코딩할 것을 결정하는 것에 기초하여, 상기 제 1 팔레트 인덱스의 값을 표시하는 데이터를 코딩하며; 그리고

상기 제 2 팔레트 인덱스를 카피할 것을 결정하는 것에 기초하여, 상기 제 1 라인의 제 1 팔레트 인덱스가 상기 제 2 라인의 제 2 팔레트 인덱스로부터 카피되는 것을 표시하는 데이터를 코딩하도록

구성되는, 비디오 데이터를 코딩하는 장치.

청구항 15

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 추가로, 상기 제 1 팔레트 인덱스로 코딩하기 위한 스캔 방향에 있어서 상기 제 1 팔레트 인덱스를 뒤따르는 연속하는 인덱스들의 수의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 16

제 15 항에 있어서,

상기 제 1 팔레트 인덱스의 값을 코딩할 것을 결정하는 것에 기초하여, 상기 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하기 위해, 상기 하나 이상의 프로세서들은 상기 제 1 팔레트 인덱스와 동일한 연속하는 인덱스들의 수의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩되고,

제 2 인덱스를 카피할 것을 결정하는 것에 기초하여, 상기 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하기 위해, 상기 하나 이상의 프로세서들은 상기 제 2 라인으로부터 카피된 연속하는 인덱스들의 수의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 17

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 추가로,

상기 블록에서 하나 이상의 개별 픽셀 포지션들의 하나 이상의 픽셀 값들이 상기 픽셀 값들의 팔레트에 포함되지 않는 것을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하도록

구성되는, 비디오 데이터를 코딩하는 장치.

청구항 18

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 상기 제 2 라인을 표시하는 하나 이상의 신팩스 엘리먼트들을 코딩하도록 구성

되는, 비디오 데이터를 코딩하는 장치.

청구항 19

제 18 항에 있어서,

상기 제 1 라인 및 상기 제 2 라인은 상기 비디오 데이터 블록의 개별적인 제 1 로우와 제 2 로우 또는 개별적인 제 1 컬럼과 제 2 컬럼을 포함하는, 비디오 데이터를 코딩하는 장치.

청구항 20

제 18 항에 있어서,

상기 제 2 라인을 표시하는 하나 이상의 선택스 엘리먼트들을 코딩하기 위해, 상기 하나 이상의 프로세서들은 상기 비디오 데이터의 블록에서 상기 제 2 라인의 위치를 식별하는 데이터를 코딩하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 21

제 20 항에 있어서,

상기 제 2 라인의 위치를 식별하는 데이터를 코딩하기 위해, 상기 하나 이상의 프로세서들은 상기 비디오 데이터의 블록에서 상기 제 2 라인을 식별하는 범위 값을 코딩하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 22

제 20 항에 있어서,

상기 제 2 라인을 표시하는 하나 이상의 선택스 엘리먼트들을 코딩하기 위해, 상기 하나 이상의 프로세서들은, 1진 또는 절단된 1진 코드를 사용하여 상기 하나 이상의 선택스 엘리먼트들을 코딩하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 23

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 추가로,

제 1 인덱스 및 상기 제 2 팔레트 인덱스를 포함하는 상기 비디오 데이터의 블록에 대한 인덱스들의 맵을 결정하고; 그리고

상기 맵의 인덱스들을 코딩하기 위한 스캔 방향을 표시하는 데이터를 코딩하도록
구성되는, 비디오 데이터를 코딩하는 장치.

청구항 24

제 14 항에 있어서,

상기 하나 이상의 프로세서들은 추가로,

상기 제 1 팔레트 인덱스의 값을 코딩할지 또는 상기 제 1 팔레트 인덱스를 상기 제 2 팔레트 인덱스에 관련하여 코딩할지 여부를 표시하는 하나 이상의 선택스 엘리먼트들을 코딩하도록

구성되고,

제 1 인덱스의 값을 코딩할지 또는 상기 제 2 팔레트 인덱스를 카피할지 여부를 결정하기 위해, 상기 하나 이상의 프로세서들은 상기 하나 이상의 선택스 엘리먼트들에 기초하여 결정하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 25

제 14 항에 있어서,

코딩하기 위해, 상기 하나 이상의 프로세서들은 디코딩하도록 구성되고,

상기 디코딩하기 위해, 상기 하나 이상의 프로세서들은,

상기 제 1 팔레트 인덱스의 픽셀 값을 상기 제 2 팔레트 인덱스의 픽셀 값을, 개별 제 1 팔레트 인덱스 및 제 2 팔레트 인덱스를 상기 팔레트의 적어도 하나의 엔트리와 매칭함으로써, 결정하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 26

제 14 항에 있어서,

코딩하기 위해, 상기 하나 이상의 프로세서들은 인코딩하도록 구성되고,

상기 인코딩하기 위해, 상기 하나 이상의 프로세서들은,

상기 제 1 팔레트 인덱스 및 상기 제 2 팔레트 인덱스를 나타내는 데이터를 인코딩된 비트스트림으로 인코딩하도록 구성되는, 비디오 데이터를 코딩하는 장치.

청구항 27

비디오 데이터를 코딩하는 장치로서,

비디오 데이터의 블록의 팔레트 인덱스들의 제 1 라인에 포함된 제 1 팔레트 인덱스에 대하여, 상기 제 1 팔레트 인덱스의 값을 코딩할지 또는 상기 팔레트 인덱스들의 제 1 라인과 상이한 상기 팔레트 인덱스들의 제 2 라인에 포함된 제 2 팔레트 인덱스들을 카피할지의 여부를 결정하는 수단으로서, 상기 제 1 팔레트 인덱스 및 상기 제 2 팔레트 인덱스는 개별적인 제 1 픽셀 포지션 및 제 2 픽셀 포지션을 상기 비디오 데이터의 블록에 대한 픽셀 값들의 팔레트들의 적어도 하나의 엔트리로 맵핑하는, 상기 코딩할지 또는 카피할지의 여부를 결정하는 수단

상기 제 1 팔레트 인덱스의 값을 코딩할 것을 결정하는 것에 기초하여, 상기 제 1 팔레트 인덱스의 값을 표시하는 데이터를 코딩하는 수단; 및

상기 제 2 팔레트 인덱스를 카피할 것을 결정하는 것에 기초하여, 상기 제 1 라인의 제 1 팔레트 인덱스가 상기 제 2 라인의 제 2 팔레트 인덱스로부터 카피되는 것을 표시하는 데이터를 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하는 장치.

청구항 28

명령들을 저장하는 비-일시적 컴퓨터 판독가능 매체로서,

상기 명령들은, 실행될 경우, 하나 이상의 프로세서들로 하여금,

비디오 데이터의 블록의 팔레트 인덱스들의 제 1 라인에 포함된 제 1 팔레트 인덱스에 대하여, 상기 제 1 팔레트 인덱스의 값을 코딩할지 또는 상기 팔레트 인덱스들의 제 1 라인과 상이한 상기 팔레트 인덱스들의 제 2 라인에 포함된 제 2 팔레트 인덱스들을 카피할지의 여부를 결정하게 하는 것으로서, 상기 제 1 팔레트 인덱스 및 상기 제 2 팔레트 인덱스는 개별적인 제 1 픽셀 포지션 및 제 2 픽셀 포지션을 상기 비디오 데이터의 블록에 대한 픽셀 값들의 팔레트들의 적어도 하나의 엔트리로 맵핑하는, 상기 코딩할지 또는 카피할지의 여부를 결정하게 하고;

상기 제 1 팔레트 인덱스의 값을 코딩할 것을 결정하는 것에 기초하여, 상기 제 1 팔레트 인덱스의 값을 표시하는 데이터를 코딩하게 하고; 그리고

상기 제 2 팔레트 인덱스를 카피할 것을 결정하는 것에 기초하여, 상기 제 1 라인의 제 1 팔레트 인덱스가 상기 제 2 라인의 제 2 팔레트 인덱스로부터 카피되는 것을 표시하는 데이터를 코딩하게 하는, 비-일시적 컴퓨터 판독가능 매체.

청구항 29

삭제

청구항 30

삭제

발명의 설명

기술 분야

- [0001] 본 출원은 2013년 4월 5일자로 출원된 미국 가출원 제 61/809,236 호 및 2013년 4월 10일자로 출원된 미국 가출원 제 61/810,649 호의 이점을 청구하며, 이들의 전체 내용들은 각각 참조에 의해 본원에 통합된다.
- [0002] 본 개시는 비디오 인코딩 및 디코딩에 관한 것이다.

배경 기술

- [0003] 디지털 비디오 성능들은, 디지털 텔레비전, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, PDA들(personal digital assistants), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, 이북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔, 셀룰러 또는 위성 라디오 텔레폰들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding), 현재 개발 하에 있는 HEVC (High Efficiency Video Coding) 표준에 의해 정의된 표준들, 및 이러한 표준들의 확장들에서 설명된 바와 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기술들을 구현함으로써 디지털 비디오 정보를 보다 효율적으로 송신하고, 수신하고, 인코딩하고, 디코딩하고, 및/또는 저장할 수도 있다.

- [0004] 비디오 압축 기술들은 비디오 시퀀스들에 고유한 리던던시를 감소시키거나 제거하기 위해 공간(인트라-픽처) 예측 및/또는 시간(인터-픽처) 예측을 포함할 수도 있다. 블록 기반 비디오 코딩에 대해, 비디오 슬라이스(즉, 비디오 프레임 또는 비디오 프레임의 일부)는 비디오 블록들로 파티셔닝될 수도 있다. 픽처의 인트라-코딩된(I) 슬라이스에서의 비디오 블록들은 동일한 픽처에서 이웃하는 블록들 내의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 픽처의 인터 코딩된(P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽처의 이웃하는 블록들 내의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 픽처들 내의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 픽처들은 프레임들이라고 지칭될 수도 있고, 참조 픽처들은 참조 프레임들로 지칭될 수도 있다.

- [0005] 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 발생한다. 잔차 데이터는 코딩될 오리지널 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 포인팅하는 모션 벡터에 따라 인코딩되고, 잔차 데이터는 코딩된 블록과 예측 블록 간의 차이를 나타낸다. 인트라-코딩된 블록은 인트라-코딩 모드와 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 계수들을 발생시킬 수도 있으며, 그 후, 이 잔차 계수들은 양자화될 수도 있다. 2 차원 어레이로 초기에 배열되는 양자화된 계수들은 계수들의 1 차원 벡터를 생성하기 위해 스캐닝될 수도 있으며, 엔트로피 코딩이 훨씬 더 많은 압축을 달성하도록 적용될 수도 있다.

- [0006] 멀티 뷰 코딩 비트스트림은, 예를 들어, 다중의 관점들로부터의 인코딩 뷰들에 의해 생성될 수도 있다. 멀티 뷰 코딩 양태들을 이용하는 일부 3 차원(3D) 비디오 표준들이 개발되었다. 예를 들어, 상이한 뷰들은 좌안 뷰 및 우안 뷰를 송신하여 3D 비디오를 지원할 수도 있다. 대안적으로, 일부 3D 비디오 코딩 프로세스들은 소위 멀티 뷰 플러스 심도 코딩을 적용할 수도 있다. 멀티 뷰 플러스 심도 코딩에 있어서, 3D 비디오 비트스트림은 텍스처 뷰 컴포넌트들뿐 아니라 심도 뷰 컴포넌트들도 포함할 수도 있다. 예를 들어, 각각의 뷰는 하나의 텍스처 뷰 컴포넌트 및 하나의 심도 뷰 컴포넌트를 포함할 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0007]

본 개시의 기술들은 팔레트-기반 비디오 코딩에 관련된다. 예를 들어, 팔레트 기반 코딩에서, 비디오 코더(비디오 인코더 또는 비디오 디코더)는 특정 영역(예컨대, 소정 블록)의 비디오 데이터를 표현하기 위한 컬러 테이블로서, 소위 "팔레트"를 형성할 수도 있다. 팔레트-기반 코딩은 상대적으로 작은 수의 컬러들을 갖는 비디오 데이터의 영역들을 코딩하는데 특히 유용할 수도 있다. 실제 픽셀 값들(또는 그들의 잔차들)을 코딩하는 것보다, 비디오 코더는 픽셀들의 컬러들을 표현하는 팔레트에서의 엔트리들과 픽셀들을 관련시키는 하나 이상의 픽셀들에 대한 인덱스 값들을 코딩할 수도 있다. 팔레트는 명시적으로 인코딩되어 디코더로 전송되거나, 이전의 팔레트 엔트리들로부터 예측되거나, 또는 이들의 조합이 될 수도 있다. 본 개시에서 설명된 기술들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 팔레트들을 코딩하는 것, 팔레트들을 예측하는 것, 팔레트들을 유도하는 것, 및 팔레트-기반 코딩 맵들과 다른 십екс 앤트리들을 코딩하는 것 중 하나 이상의 다양한 조합들에 대한 기술들을 포함한다.

[0008]

일 예에서, 비디오 데이터를 코딩하는 방법은, 제 1 픽셀 값들을 표시하는 제 1 엔트리들을 갖는 제 1 팔레트를 결정하는 단계, 제 1 팔레트의 제 1 엔트리들에 기초하여, 제 2 팔레트의 제 2 픽셀 값들을 표시하는 하나 이상의 제 2 엔트리들을 결정하는 단계, 및 제 2 팔레트를 사용하여 비디오 데이터의 블록의 픽셀들을 코딩하는 단계를 포함한다.

[0009]

다른 예에서, 비디오 데이터를 코딩하는 장치는, 비디오 데이터를 저장하는 메모리, 및 제 1 픽셀 값들을 표시하는 제 1 엔트리들을 갖는 제 1 팔레트를 결정하고, 제 1 팔레트의 제 1 엔트리들에 기초하여, 제 2 팔레트의 제 2 픽셀 값들을 표시하는 하나 이상의 제 2 엔트리들을 결정하며, 그리고 제 2 팔레트를 사용하여 비디오 데이터의 블록의 픽셀들을 코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0010]

또 다른 예에서, 비디오 데이터를 코딩하는 장치는, 제 1 픽셀 값들을 표시하는 제 1 엔트리들을 갖는 제 1 팔레트를 결정하는 수단, 제 1 팔레트의 제 1 엔트리들에 기초하여, 제 2 팔레트의 제 2 픽셀 값들을 표시하는 하나 이상의 제 2 엔트리들을 결정하는 수단, 및 제 2 팔레트를 사용하여 비디오 데이터의 블록의 픽셀들을 코딩하는 수단을 포함한다.

[0011]

다른 예에서, 비-일시적 컴퓨터 판독가능 매체는, 실행될 경우, 하나 이상의 프로세서들로 하여금, 제 1 픽셀 값들을 표시하는 제 1 엔트리들을 갖는 제 1 팔레트를 결정하게 하고, 제 1 팔레트의 제 1 엔트리들에 기초하여, 제 2 팔레트의 제 2 픽셀 값들을 표시하는 하나 이상의 제 2 엔트리들을 결정하게 하며, 그리고 제 2 팔레트를 사용하여 비디오 데이터의 블록의 픽셀들을 코딩하게 하는 명령들을 저장한다.

[0012]

다른 예에서, 비디오 데이터를 코딩하는 방법은, 비디오 데이터의 블록에서 제 1 픽셀과 연관된 제 1 인덱스 값을 결정하는 단계로서, 여기서 제 1 인덱스 값은 제 1 픽셀의 포지션을 픽셀 값들의 팔레트의 엔트리에 관련시키는, 상기 제 1 인덱스 값을 결정하는 단계, 제 1 인덱스 값에 기초하여, 비디오 데이터의 블록에서 하나 이상의 제 2 픽셀들과 연관된 하나 이상의 제 2 인덱스 값을 결정하는 단계로서, 여기서 제 2 인덱스 값들은 하나 이상의 제 2 픽셀들의 포지션들을 픽셀 값들의 팔레트의 하나 이상의 엔트리들에 관련시키는, 상기 하나 이상의 제 2 인덱스 값을 결정하는 단계, 및 비디오 데이터의 블록의 제 1 픽셀 및 하나 이상의 제 2 픽셀들을 코딩하는 단계를 포함한다.

[0013]

다른 예에서, 비디오 데이터를 코딩하는 장치는 비디오 데이터를 저장하는 메모리를 포함하고, 하나 이상의 프로세서들을 저장하며, 하나 이상의 프로세서들은, 비디오 데이터의 블록에서 제 1 픽셀과 연관된 제 1 인덱스 값을 결정하는 것으로서, 여기서 제 1 인덱스 값은 제 1 픽셀의 포지션을 픽셀 값들의 팔레트의 엔트리에 관련시키는, 상기 제 1 인덱스 값을 결정하고, 제 1 인덱스 값에 기초하여, 비디오 데이터의 블록에서 하나 이상의 제 2 픽셀들과 연관된 하나 이상의 제 2 인덱스 값을 결정하는 것으로서, 여기서 제 2 인덱스 값들은 하나 이상의 제 2 픽셀들의 포지션들을 픽셀 값들의 팔레트의 하나 이상의 엔트리들에 관련시키는, 상기 하나 이상의 제 2 인덱스 값을 결정하며, 그리고 비디오 데이터의 블록의 제 1 픽셀 및 하나 이상의 제 2 픽셀들을 코딩하도록 구성된다.

[0014]

또 다른 예에서, 비디오 데이터를 코딩하는 장치는, 비디오 데이터의 블록에서 제 1 픽셀과 연관된 제 1 인덱스 값을 결정하는 수단으로서, 여기서 제 1 인덱스 값은 제 1 픽셀의 포지션을 픽셀 값들의 팔레트의 엔트리에 관련시키는, 상기 제 1 인덱스 값을 결정하는 수단, 제 1 인덱스 값에 기초하여, 비디오 데이터의 블록에서 하나 이상의 제 2 픽셀들과 연관된 하나 이상의 제 2 인덱스 값을 결정하는 수단으로서, 여기서 제 2 인덱스 값들은 하나 이상의 제 2 픽셀들의 포지션들을 픽셀 값들의 팔레트의 하나 이상의 엔트리들에 관련시키는, 상기 하나 이상의 제 2 인덱스 값을 결정하며, 그리고 비디오 데이터의 블록의 제 1 픽셀 및 하나 이상의 제 2 픽셀들을 코딩하도록 구성된다.

나 이상의 제 2 인덱스 값들을 결정하는 수단, 및 비디오 데이터의 블록의 제 1 픽셀 및 하나 이상의 제 2 픽셀들을 코딩하는 수단을 포함한다.

[0015] 다른 예에서, 비-일시적 컴퓨터 판독가능 매체는, 실행될 경우, 하나 이상의 프로세서들로 하여금, 비디오 데이터의 블록에서 제 1 픽셀과 연관된 제 1 인덱스 값을 결정하게 하는 것으로서, 여기서 제 1 인덱스 값은 제 1 픽셀의 포지션을 픽셀 값들의 팔레트의 엔트리에 관련시키는, 상기 제 1 인덱스 값을 결정하게 하고, 제 1 인덱스 값에 기초하여, 비디오 데이터의 블록에서 하나 이상의 제 2 픽셀들과 연관된 하나 이상의 제 2 인덱스 값을 결정하게 하는 것으로서, 여기서 제 2 인덱스 값들은 하나 이상의 제 2 픽셀들의 포지션들을 픽셀 값들의 팔레트의 하나 이상의 엔트리들에 관련시키는, 상기 하나 이상의 제 2 인덱스 값을 결정하게 하며, 그리고 비디오 데이터의 블록의 제 1 픽셀 및 하나 이상의 제 2 픽셀들을 코딩하게 하는 명령들을 저장한다.

[0016] 본 개시의 하나 이상의 예들의 상세들은 첨부 도면들 및 하기의 설명에 기재된다. 다른 특징들, 목적들, 및 이점들은 그 설명, 도면들, 및 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0017] 도 1 은 본 개시에서 설명된 기술들을 활용할 수도 있는 예시적인 비디오 코딩 시스템을 도시한 블록 다이어그램이다.

도 2 는 본 개시에 설명된 기술들을 구현할 수도 있는 예시적인 비디오 인코더를 도시하는 블록 다이어그램이다.

도 3 은 본 개시에 설명된 기술들을 구현할 수도 있는 예시적인 비디오 디코더를 도시하는 블록 다이어그램이다.

도 4 는 본 개시의 기술들과 부합하는, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 일 예를 도시하는 개념 다이어그램이다.

도 5 는 본 개시의 기술들과 부합하는, 픽셀들의 블록에 대하여 팔레트로의 인덱스들을 결정하는 일 예를 예시하는 개념 다이어그램이다.

도 6 은 본 개시의 기술들과 부합하는, 팔레트 코딩 모드를 사용하여 비디오 데이터를 코딩하는 예시적인 프로세스를 도시하는 플로우차트이다.

도 7 은 본 개시의 기술들과 부합하는, 팔레트-기반 코딩에서 팔레트를 결정하는 예시적인 프로세스를 도시하는 플로우차트이다.

도 8 은 본 개시의 기술들과 부합하는, 팔레트-기반 비디오 코딩에서 비디오 데이터의 블록의 인덱스들을 결정하는 예시적인 프로세스를 도시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 개시는 비디오 코딩 및 압축을 위한 기술들을 포함한다. 특히, 본 개시는 비디오 데이터의 팔레트-기반 코딩을 위한 기술들을 설명한다. 종래의 비디오 코딩에서, 이미지들은 연속하는 톤이고 공간적으로 평활한 것으로 가정된다. 이를 가정하여, 블록-기반 변환, 필터링, 등등과 같은 다양한 툴들이 개발되었으며, 그러한 툴들은 자연적 컨텐츠 비디오에 대하여 우수한 성능을 보여주었다.

[0019] 그러나, 원격 테스크톱, 협업 작업 및 무선 디스플레이와 같은 애플리케이션들에서, (예컨대, 텍스트 또는 컴퓨터 그래픽과 같은) 컴퓨터 생성 스크린 컨텐츠는 압축될 우세한 컨텐츠일 수도 있다. 이러한 타입의 컨텐츠는 불연속 톤 (discrete-tone) 을 가지고, 선명한 라인들과 높은 콘트라스트의 오브젝트 경계들을 특징으로 하는 경향이 있다. 연속 톤 및 평활성의 가정은 더 이상 스크린 컨텐츠에 적용될 수 없으며, 따라서 종래의 비디오 코딩 기술들은 스크린 컨텐츠를 포함하는 비디오 데이터를 압축하는데 있어 효율적인 방식일 수 없다.

[0020] 본 개시는 특히, 스크린 생성 컨텐츠 코딩에 적합할 수도 있는, 팔레트-기반 코딩을 설명한다. 예를 들어, 비디오 데이터의 특정 영역을 가정하는 것은 상대적으로 작은 수의 컬러들을 갖는다. 비디오 코더 (비디오 인코더 또는 비디오 디코더)는 특정 영역 (예컨대, 소정 블록) 의 비디오 데이터를 표현하기 위한 컬러 테이블로서, 소위 "팔레트" 를 형성할 수도 있다. 각각의 픽셀은 픽셀의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 코더는 픽셀 값을 팔레트에서의 적절한 값에 관련시키는 인덱스를 코

딩할 수도 있다.

[0021] 앞의 예에서, 비디오 인코더는 비디오 데이터의 블록을, 그 블록에 대한 팔레트를 결정하고 (예컨대, 팔레트를 명시적으로 코딩하거나, 팔레트를 예측하거나, 또는 이들을 조합하고), 각 픽셀의 값을 표현하도록 팔레트에 엔트리를 로케이팅하고, 픽셀 값을 팔레트에 관련시키는 픽셀들에 대한 인덱스 값들로 블록을 인코딩함으로써, 인코딩할 수도 있다. 비디오 디코더는, 인코딩된 비트스트림으로부터, 블록에 대한 팔레트뿐만 아니라 블록의 픽셀들에 대한 인덱스 값들을 획득할 수도 있다. 비디오 디코더는 블록의 픽셀 값들을 복원하기 위해 픽셀들의 인덱스 값들을 팔레트의 엔트리들에 관련시킬 수도 있다.

[0022] 앞의 예는 팔레트-기반 코딩의 일반적인 설명을 제공하도록 의도된다. 다양한 예들에서, 본 개시에서 설명된 기술들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 예측하는 것, 팔레트들을 유도하는 것, 및 팔레트-기반 코딩 맵들 및 다른 신팩스 엘리먼트들을 송신하는 것 중 하나 이상의 다양한 조합들에 대한 기술들을 포함할 수도 있다. 그러한 기술들은 예컨대, 스크린 생성 컨텐츠를 표현하기 위해 더 적은 비트들을 요구하여, 비디오 코딩 효율을 개선할 수도 있다.

[0023] 비디오 데이터의 팔레트-기반 코딩을 위한 기술들은 인터-예측 코딩 또는 인트라-예측 코딩을 위한 기술들과 같은 하나 이상의 다른 코딩 기술들과 함께 사용될 수도 있다. 예를 들어, 이하 더 상세히 설명되는 것과 같이, 인코더 또는 디코더, 또는 결합된 인코더-디코더 (코덱) 는 인터-예측 코딩 및 인트라-예측 코딩뿐만 아니라 팔레트-기반 코딩을 수행하도록 구성될 수도 있다.

[0024] 일부 예들에서, 팔레트-기반 코딩 기술들은 하나 이상의 비디오 코딩 표준들과 사용하기 위해 구성될 수도 있다. 예를 들어, HEVC (High Efficiency Video Coding) 은 ITU-T VCEG (Video Coding Experts Group) 및 ISO/IEC MPEG (Motion Picture Experts Group) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 개발된 새로운 비디오 코딩 표준이다. 최근의 HEVC 텍스트 명세서 드래프트는, Bross 등의 "High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Consent)", JCVC-L1003_v13, ITU-T SG16 WP 3 및 ISO/IEC JCT 1/SC 29/WG 11 의 JCT-VC 의 제 12 차 회의, 2013년 1월 14 - 23일 ("HEVC Draft 10")에 기술되어 있고, http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003_v13.zip에서 입수가능하다.

[0025] HEVC 프레임워크에 대하여, 일 예로서, 팔레트-기반 코딩 기술들은 코딩 유닛 (CU) 모드로서 사용되도록 구성될 수도 있다. 다른 예들에서, 팔레트-기반 코딩 기술들은 HEVC의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 따라서, CU 모드의 맥락에서 설명되는 이하 개시된 프로세스들 전부는, 부가적으로 또는 대안적으로, PU에 적용할 수도 있다. 그러나, 이를 HEVC-기반 예들은 본원에 설명된 팔레트-기반 코딩 기술들의 제한 또는 한정으로 고려되어서는 안되며, 이는 그러한 기술들이 다른 기존의 또는 아직 개발되지 않은 시스템들/표준들과 독립적으로 또는 이들의 부분으로서 작용하도록 적용될 수도 있기 때문이다. 이를 케이스들에서, 팔레트 코딩을 위한 유닛은 정사각형 블록들, 직사각형 블록들 또는 심지어 직사각형 형상이 아닌 영역들일 수 있다.

[0026] 도 1은 본 개시의 기술들을 활용할 수도 있는 예시적인 비디오 코딩 시스템 (10)을 도시한 블록 다이어그램이다. 본원에서 사용된 바와 같이, 용어 "비디오 코더"는 일반적으로 비디오 인코더들 및 비디오 디코더들 양자를 지칭한다. 본 개시에 있어서, 용어를 "비디오 코딩" 또는 "코딩"은 일반적으로 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다. 비디오 코딩 시스템 (10)의 비디오 인코더 (20) 및 비디오 디코더 (30)는, 본 개시에 설명된 다양한 예들에 따른 팔레트-기반 비디오 코딩을 위한 기술들을 수행하도록 구성된 디바이스들의 예들을 나타낸다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트-기반 코딩 또는 비-팔레트 기반 코딩을 사용하여, HEVC 코딩에서 CU들 및 PU들과 같은 비디오 데이터의 다양한 블록들을 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트 기반 코딩 모드들은 HEVC 드래프트 10에 의해 명시된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간 코딩 모드를 또는 인트라-예측 공간 코딩 모드들을 지칭할 수도 있다.

[0027] 도 1에 도시된 바와 같이, 비디오 코딩 시스템 (10)은 소스 디바이스 (12) 및 목적지 디바이스 (14)를 포함한다. 소스 디바이스 (12)는 인코딩된 비디오 데이터를 생성한다. 이에 따라, 소스 디바이스 (12)는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치로서 지칭될 수도 있다. 목적지 디바이스 (14)는 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 이에 따라, 목적지 디바이스 (14)는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치로서 지칭될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.

- [0028] 소스 디바이스 (12) 및 목적지 디바이스 (14)는 데스크탑 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예를 들어, 램프) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 차량내 컴퓨터들 등을 포함하여 광범위한 디바이스들을 포함할 수도 있다.
- [0029] 목적지 디바이스 (14)는 소스 디바이스 (12)로부터의 인코딩된 비디오 데이터를 채널 (16)을 통해 수신할 수도 있다. 채널 (16)은 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 이동 가능한 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 일 예에 있어서, 채널 (16)은, 소스 디바이스 (12)가 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14)로 실시간으로 송신할 수 있게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에 있어서, 소스 디바이스 (12)는 인코딩된 비디오 데이터를 무선 통신 프로토콜과 같은 통신 표준에 따라 변조할 수도 있고, 변조된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들과 같은 무선 및/또는 유선 통신 매체들을 포함할 수도 있다. 하나 이상의 통신 매체들은 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예를 들어, 인터넷) 와 같은 패킷-기반 네트워크의 부분을 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12)로부터 목적지 디바이스 (14)로의 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.
- [0030] 다른 예에 있어서, 채널 (16)은, 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에 있어서, 목적지 디바이스 (14)는 디스크 액세스 또는 카드 액세스를 통해 저장 매체에 액세스할 수도 있다. 저장 매체는 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 국부적으로 액세스된 데이터 저장 매체들을 포함할 수도 있다.
- [0031] 추가의 예에 있어서, 채널 (16)은, 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에 있어서, 목적지 디바이스 (14)는 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터에 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는, 인코딩된 비디오 데이터를 저장하고, 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신하는 것이 가능한 타입의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버들 (예를 들어, 웹 사이트용), 파일 전송 프로토콜 (FTP) 서버들, 네트워크 부착형 저장 (NAS) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.
- [0032] 목적지 디바이스 (14)는 인코딩된 비디오 데이터에 인터넷 커넥션과 같은 표준 데이터 커넥션을 통해 액세스할 수도 있다. 예시적인 타입들의 데이터 커넥션들은 파일 서버 상에 저장되는 인코딩된 비디오 데이터에 액세스하기에 적합한, 무선 채널들 (예를 들어, Wi-Fi 커넥션들), 유선 커넥션들 (예를 들어, DSL, 케이블 모뎀 등), 또는 이를 양자의 조합들을 포함할 수도 있다. 인코딩된 비디오 데이터의 파일 서버로부터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이를 양자의 조합일 수도 있다.
- [0033] 본 개시의 기술들은 무선 애플리케이션들 또는 설정들에 한정되지 않는다. 그 기술들은, 공중 경유 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예를 들어, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들의 지원으로 비디오 코딩에 적용될 수도 있다. 일부 예들에 있어서, 비디오 코딩 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화와 같은 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.
- [0034] 도 1은 단지 예일 뿐이고, 본 개시의 기술들은, 인코딩 디바이스와 디코딩 디바이스 간의 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들 (예를 들어, 비디오 인코딩 또는 비디오 디코딩)에 적용할 수도 있다. 다른 예들에 있어서, 데이터는 로컬 메모리로부터 추출되거나 네트워크상으로 스트리밍되는 등등이다. 비디오 인코딩 디바이스는 데이터를 인코딩하고 메모리에 저장할 수도 있고, 및/또는 비디오 디코딩 디바이스는 데이터를 메모리로부터 추출하고 디코딩할 수도 있다. 다수의 예들에 있어서, 인코딩 및 디코딩은, 서로 통신하지 않지만 단순히 데이터를 메모리로 인코딩하고 및/또는 데이터를 메모리로부터 추출하고 디코딩하는 디바이스들에 의해 수행된다.
- [0035] 도 1의 예에 있어서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 일부 예들에 있어서, 출력 인터페이스 (22)는 변조기/복조기 (모뎀) 및/또는 송신기를 포함

할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예를 들어, 비디오 카메라, 이전에 캡처된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하기 위한 비디오 퍼드 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽 시스템, 또는 비디오 데이터의 그러한 소스들의 조합을 포함할 수도 있다.

[0036] 비디오 인코더 (20) 는 비디오 소스 (18)로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에 있어서, 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 직접 송신한다. 다른 예들에 있어서, 인코딩된 비디오 데이터는 또한, 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14) 에 의한 추후 액세스를 위해 저장 매체 또는 파일 서버 상에 저장될 수도 있다.

[0037] 도 1 의 예에 있어서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 예들에 있어서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함한다. 입력 인터페이스 (28) 는 인코딩된 비디오 데이터를 채널 (16) 상으로 수신할 수도 있다. 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합될 수도 있거나 또는 그 외부에 있을 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32) 는 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들을 포함할 수도 있다.

[0038] 본 개시는 일반적으로 특정 정보를, 비디오 디코더 (30) 와 같은 다른 디바이스로 "시그널링" 또는 "송신"하는 비디오 인코더 (20) 를 지칭할 수도 있다. 용어 "시그널링" 또는 "송신"은 일반적으로, 압축된 비디오 데이터를 디코딩하는데 사용되는 다른 데이터 및/또는 신택스 엘리먼트들의 통신을 지칭할 수도 있다. 그러한 통신은 실시간 또는 준-실시간으로 발생할 수도 있다. 대안적으로, 그러한 통신은, 인코딩 시 신택스 엘리먼트들을 인코딩된 비트스트림으로 컴퓨터 판독가능 저장 매체에 저장할 경우 (그 후, 이 매체에 저장된 이후 임의의 시간에서 디코딩 디바이스에 의해 취출될 수도 있음) 에 발생할 수도 있는 것과 같이 시간 기간에 걸쳐 발생할 수도 있다. 따라서, 비디오 디코더 (30) 가 특정 정보를 "수신하는" 것으로 지칭될 수도 있지만, 정보의 수신은 반드시 실시간 또는 준-실시간으로 발생할 필요는 없고, 저장 이후 임의의 시간에 매체로부터 취출될 수도 있다.

[0039] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 별도의 로직, 하드웨어, 또는 이들의 임의의 조합들과 같은 다양한 적절한 회로 중 임의의 회로로서 구현될 수도 있다. 기술들이 부분적으로 소프트웨어로 구현된다면, 디바이스는 적절한 비-일시적인 컴퓨터 판독가능 저장 매체에 소프트웨어에 대한 명령들을 저장할 수도 있으며, 본 개시의 기술들을 수행하기 위해 하나 이상의 프로세서들을 사용하여 하드웨어로 명령들을 실행할 수도 있다. (하드웨어, 소프트웨어, 하드웨어와 소프트웨어의 조합 등을 포함한) 임의의 전술한 내용은 하나 이상의 프로세서들인 것으로 고려될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이들 중 어느 것도 결합된 인코더/디코더 (CODEC) 의 일부로서 각각의 디바이스에 통합될 수도 있다.

[0040] 일부 예들에 있어서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 앞서 언급되고 HEVC 드래프트 10 에 설명된 비디오 압축 표준에 따라 동작한다. 기본 HEVC 표준에 추가하여, HEVC 에 대해 스케일러를 비디오 코딩, 멀티 뷰 비디오 코딩, 및 3D 코딩 확장들을 생성하기 위한 노력들이 계속되고 있다. 추가로, 예컨대 본 개시에 설명된 것과 같은 팔레트-기반 코딩 모드들은 HEVC 표준의 확장을 위해 제공될 수도 있다. 일부 예들에 있어서, 팔레트-기반 코딩을 위해 본 개시에 설명된 기술들은 ITU-T-H.264/AVC 표준 또는 미래의 표준들과 같은 다른 비디오 코딩 표준들에 따라 동작하도록 구성된 인코더들 및 디코더들에 적용될 수도 있다. 따라서, HEVC 코덱에서 코딩 유닛들 (CU들) 또는 예측 유닛들 (PU들) 의 코딩을 위한 팔레트-기반 코딩 모드의 적용은 예시의 목적들을 위해 설명된다.

[0041] HEVC 및 다른 비디오 코딩 표준들에 있어서, 비디오 시퀀스는 통상적으로 픽처들의 시리즈를 포함한다. 픽처들은 또한 "프레임들"로서 지칭될 수도 있다. 픽처는 S_L , S_{Cb} , 및 S_{Cr} 로 표기되는 3 개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2 차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 크로미넌스 샘플들 의 2 차원 어레이이다. S_{Cr} 는 Cr 크로미넌스 샘플들의 2 차원 어레이이다. 크로미넌스 샘플들은 또한, 본 명세서에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우들에 있어서, 픽처는 단색일 수도 있으며 오직 루마 샘플들의 어레이만을 포함할 수도 있다.

- [0042] 꾹처의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛들 (CTU들) 의 세트를 생성할 수도 있다. CTU들 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2 개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들일 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로서 지칭될 수도 있다. HEVC 의 CTU들은 H.264/AVC 와 같은 다른 표준들의 매크로블록들과 대체로 유사할 수도 있다. 하지만, CTU 는 반드시 특정 사이즈로 한정되지는 않으며, 하나 이상의 코딩 유닛들 (CU들) 을 포함할 수도 있다. 슬라이스는 레스터 스캔에서 연속적으로 순서화된 정수 개수의 CTU들을 포함할 수도 있다.
- [0043] 코딩된 슬라이스는 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스의 슬라이스 헤더는 슬라이스에 관한 정보를 제공하는 신택스 엘리먼트들을 포함하는 신택스 구조일 수도 있다. 슬라이스 데이터는 슬라이스의 코딩된 CTU들을 포함할 수도 있다.
- [0044] 본 개시물은 용어 "비디오 유닛" 또는 "비디오 블록" 또는 "블록" 을 사용하여, 샘플들의 하나 이상의 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들 및 하나 이상의 샘플 블록들을 지칭할 수도 있다. 비디오 유닛들 또는 블록들의 예시적인 타입들은, CTU들, CU들, PU들, 변환 유닛들 (TU들), 매크로블록들, 매크로블록 파티션들, 등등을 포함할 수도 있다. 일부 맥락들에서, PU들이 논의는 매크로블록 파티션들의 매크로블록들의 논의와 상호교환될 수도 있다.
- [0045] 코딩된 CTU 를 생성하기 위해, 비디오 인코더 (20) 는 CTU 의 코딩 트리 블록들에 쿼드 트리 파티셔닝을 재귀적으로 수행하여, 코딩 트리 블록들을 코딩 블록들, 따라서, 명칭 "코딩 트리 유닛들" 로 분할할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다. CU 는 루마 샘플들의 코딩 블록, 및 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 갖는 꾹처의 크로마 샘플들의 2개의 대응하는 코딩 블록들, 그리고 코딩 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은, 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록일 수도 있다. CU 의 예측 유닛 (PU) 은 루마 샘플들의 예측 블록, 꾹처의 크로마 샘플들의 2 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용된 신택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.
- [0046] 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 이용하여 PU 에 대한 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 인트라 예측을 이용하여 PU 의 예측 블록들을 생성하면, 비디오 인코더 (20) 는 PU 와 연관된 꾹처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.
- [0047] 비디오 인코더 (20) 가 인터 예측을 이용하여 PU 의 예측 블록들을 생성하면, 비디오 인코더 (20) 는 PU 와 연관된 꾹처 이외의 하나 이상의 꾹처들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 는 PU 의 예측 블록들을 생성하기 위해 단방향 예측 또는 양방향 예측을 사용할 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 생성하기 위해 단방향 예측을 사용할 경우, PU 는 단일 MV 를 가질 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 생성하기 위해 양방향 예측을 사용할 경우, PU 는 2 개의 MV들을 가질 수도 있다.
- [0048] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 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 중 하나에서의 루마 샘플과 CU 의 오리지널 루마 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다. 부가적으로, 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 중 하나에서의 Cb 샘플과 CU 의 오리지널 Cb 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 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 중 하나에서의 Cr 샘플과 CU 의 오리지널 Cr 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 수도 있다.
- [0049] 더욱이, 비디오 인코더 (20) 는 쿼드 트리 파티셔닝을 이용하여, CU 의 루마, Cb, 및 Cr 잔차 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록은, 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용된 신택스 구조들일 수도 있다. 따라서, 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-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블

록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다.

[0050] 비디오 인코더 (20) 는 TU 의 루마 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 루마 계수 블록을 생성할 수도 있다. 계수 블록은 변환 계수들의 2 차원 어레이일 수도 있다. 변환 계수는 스칼라량일 수도 있다. 비디오 인코더 (20) 는 TU 의 Cb 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 TU 의 Cr 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 Cr 계수 블록을 생성할 수도 있다.

[0051] 계수 블록 (예를 들어, 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 위해 사용되는 데이터의 양을 가능하면 감소시키기 위해 변환 계수들이 양자화되어 추가적인 압축을 제공하는 프로세스를 지칭한다.

비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 십택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 십택스 엘리먼트들에 대해 컨텍스트-적응형 이진 산술 코딩 (CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩된 십택스 엘리먼트들을 비트스트림으로 출력할 수도 있다.

[0052] 비디오 인코더 (20) 는 엔트로피 인코딩된 십택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 비트스트림은 네트워크 추상화 계층 (NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛들 각각은 NAL 유닛 헤더를 포함하고 RBSP (raw byte sequence payload) 를 캡슐화한다. NAL 유닛 헤더는, NAL 유닛 타입 코드를 나타내는 십택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 명시된 NAL 유닛 타입 코드는 NAL 유닛의 타입을 나타낸다. RBSP 는, NAL 유닛 내에서 캡슐화되는 정수 개수의 바이트들을 포함하는 십택스 구조일 수도 있다. 일부 경우들에 있어서, RBSP 는 제로 비트들을 포함한다.

[0053] 상이한 타입들의 NAL 유닛들은 상이한 타입들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 타입은 픽처 파라미터 세트 (PPS) 에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 2 타입은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있으며, NAL 유닛의 제 3 타입은 SEI 에 대한 RBSP 를 캡슐화할 수도 있는 등 등이다. (파라미터 세트들 및 SEI 메시지들에 대한 RBSP들과 대조적으로) 비디오 코딩 데이터에 대한 RBSP 들을 캡슐화하는 NAL 유닛들은 비디오 코딩 계층 (NCL) NAL 유닛들로서 지칭될 수도 있다.

[0054] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 부가적으로, 비디오 디코더 (30) 는 비트스트림을 파싱하여, 비트스트림으로부터 십택스 엘리먼트들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 십택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하기 위한 프로세스는 일반적으로, 비디오 인코더 (20) 에 의해 수행된 프로세스에 역일 수도 있다.

[0055] 예를 들어, 비디오 디코더 (30) 는 현재 CU 의 PU들에 대한 예측 샘플 블록들을 경정하기 위해 PU들의 MV들을 사용할 수도 있다. 부가적으로, 비디오 디코더 (30) 는 현재 CU 의 TU들과 연관된 변환 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 변환 계수 블록들에 역변환들을 수행하여 현재 CU 의 TU들과 연관된 변환 블록들을 복원할 수도 있다. 비디오 디코더 (30) 는 현재 CU 의 PU들에 대한 예측 샘플 블록들의 샘플들을, 현재 CU 의 TU들의 변환 블록들의 대응하는 샘플들에 부가함으로써, 현재 CU 의 코딩 블록들을 복원 할 수도 있다. 픽처의 각각의 CU 에 대한 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 픽처를 복원 할 수도 있다.

[0056] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다. 예를 들어, 팔레트 기반 코딩에서, 전술된 인트라-예측 코딩 또는 인터-예측 코딩 기술들을 수행하는 것보다, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 특정 영역 (예컨대, 소정 블록) 의 비디오 데이터를 표현하기 위한 컬러 테이블로서, 소위 팔레트를 코딩할 수도 있다. 각각의 픽셀은 픽셀의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 픽셀 값을 팔레트에서의 적절한 값에 관련시키는 인덱스를 코딩할 수도 있다.

[0057] 앞의 예에서, 비디오 인코더 (20) 는 비디오 데이터의 블록을, 그 블록에 대한 팔레트를 결정하고, 각 픽셀의 값을 표현하도록 팔레트에 엔트리를 로케이팅하고, 픽셀 값을 팔레트에 관련시키는 픽셀들에 대한 인덱스 값을 들로 팔레트를 인코딩함으로써, 인코딩할 수도 있다. 비디오 디코더 (30) 는, 인코딩된 비트스트림으로부터, 블록에 대한 팔레트뿐만 아니라 블록의 픽셀들에 대한 인덱스 값을 획득할 수도 있다. 비디오 디코더

(30) 는 블록의 픽셀 값들을 복원하기 위해 픽셀들의 인덱스 값들을 팔레트의 엔트리들에 관련시킬 수도 있다.

[0058] 팔레트 기반 코딩은 특정 양의 시그널링 오버헤드를 가질 수도 있다. 예를 들어 팔레트의 사이즈와 같은 팔레트의 특징들뿐만 아니라 팔레트 자체를 시그널링하기 위해, 다수의 비트들이 요구될 수도 있다. 추가로, 다수의 비트들은 블록의 픽셀들에 대한 인덱스 값들을 시그널링하기 위해 요구될 수도 있다. 본 개시의 기술들은 일부 예들에 있어서, 그러한 정보를 시그널링하는데 필요한 비트들의 수를 감소시킬 수도 있다. 예를 들어, 본 개시에서 설명된 기술들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 예측하는 것, 팔레트들을 유도하는 것, 및 팔레트-기반 코딩 맵들과 다른 신팩스 엘리먼트들을 송신하는 것 중 하나 이상의 다양한 조합들에 대한 기술들을 포함할 수도 있다.

[0059] 본 개시의 양태들은 팔레트 예측에 관한 것이다. 예를 들어, 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 제 1 픽셀 값들을 표시하는 제 1 엔트리들을 갖는 제 1 팔레트를 결정할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 그 후, 제 1 팔레트의 제 1 엔트리들에 기초하여, 제 2 팔레트의 제 2 픽셀 값들을 표시하는 하나 이상의 제 2 엔트리들을 결정할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 또한, 제 2 팔레트를 사용하여 비디오 데이터의 블록의 픽셀들을 코딩할 수도 있다.

[0060] 제 1 엔트리들에 기초하여 제 2 팔레트의 제 2 엔트리들을 결정할 경우, 비디오 인코더 (20)는 제 2 팔레트를 복원하기 위해 비디오 디코더에 의해 사용될 수도 있는 다양한 신팩스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는, 전체 팔레트 (또는 개별 팔레트를 갖는 비디오 데이터의 각각의 컬러 성분, 예컨대 Y, Cb, Cr, 또는 Y, U, V, 또는 R, G, B의 경우에, 팔레트들)가 현재 코딩되고 있는 블록의 하나 이상의 이웃하는 블록들로부터 카피되는 것을 표시하기 위해, 하나 이상의 신팩스 엘리먼트들을 비트스트림에 인코딩할 수도 있다. 현재 블록의 현재 팔레트의 엔트리들이 예측되는 (예컨대, 카피되는) 팔레트는 예측 팔레트로 지칭될 수도 있다. 예측 팔레트는 공간적으로 이웃하는 블록들을 포함하는 하나 이상의 이웃하는 블록들로부터의 팔레트 엔트리들 및/또는 이웃하는 블록들을 블록들의 특정 스캔 순서로 포함할 수도 있다. 예를 들어, 이웃하는 블록들은 현재 코딩되고 있는 블록의 좌측에 (좌측 이웃 블록) 또는 상부에 (상부 이웃 블록) 공간적으로 로케이팅될 수도 있다. 다른 예에 있어서, 비디오 인코더 (20)는 현재 블록의 원인이 되는 이웃 (causal neighbor)의 가장 빈번한 샘플 값을 사용하여 예측 팔레트 엔트리들을 결정할 수도 있다.

또 다른 예에 있어서, 이웃하는 블록들은 블록들을 코딩하는데 사용되는 특정 스캔 순서에 따라 현재 코딩되고 있는 블록을 이웃할 수도 있다. 즉, 이웃하는 블록들은 스캔 순서에 있어서 현재 블록 이전에 코딩된 하나 이상의 블록들일 수도 있다. 비디오 인코더 (20)는 팔레트(들)이 카피되는 이웃하는 블록들의 위치를 표시하기 위해 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다.

[0061] 일부 예들에서, 팔레트 예측은 엔트리-방식으로 수행될 수도 있다. 예를 들어, 비디오 인코더 (20)는, 예측 팔레트의 각 엔트리에 대하여, 그 팔레트 엔트리가 현재 블록에 대한 팔레트에 포함되는지 여부를 나타내기 위해, 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20)가 현재 블록에 대한 팔레트의 엔트리를 예측하지 못한다면, 비디오 인코더 (20)는 예측되지 않은 엔트리들뿐만 아니라 그러한 엔트리들의 수를 명시하기 위해 하나 이상의 추가의 신팩스 엘리먼트들을 인코딩할 수도 있다.

[0062] 일부 예들에서, 전체 팔레트를 예측하기 위한 기술들은 팔레트의 하나 이상의 엔트리들을 예측하기 위한 기술들과 결합될 수도 있다. 예를 들어, 비디오 인코더 (20)는, 현재 팔레트가 예측 팔레트로부터 전체가 카피되는지 여부를 나타내기 위해 하나 이상의 신팩스 엘리먼트들을 비트스트림으로 인코딩할 수도 있다. 그렇지 않다면, 비디오 인코더 (20)는 예측 팔레트에서 각각의 엔트리가 카피되는지 여부를 나타내기 위해 하나 이상의 신팩스 엘리먼트들을 비트스트림으로 인코딩할 수도 있다.

[0063] 다른 예에서, 다수의 엔트리들 및 팔레트 값을 시그널링하는 대신, 비디오 인코더 (20)는, 각각의 팔레트 값을 시그널링한 후에, 시그널링된 팔레트 값이 그 팔레트에 대한 최종 팔레트 엔트리인지 여부를 나타내기 위해 플래그를 시그널링할 수도 있다. 비디오 인코더 (20)는 팔레트가 특정 최대 사이즈에 이미 도달한 경우, 그러한 "팔레트의 종료" 플래그를 시그널링하지 않을 수도 있다.

[0064] 본 개시의 양태들에 따라, 비디오 인코더 (20)는 팔레트 예측이 가능한지 및/또는 활성인지 여부를 나타내기 위해 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 예시의 목적들을 위한 일 예로서, 비디오 인코더 (20)는 각각의 블록 (예컨대, CU 또는 PU)에 대하여, 비디오 인코더 (20)가 개별 블록에 대한 팔레트를 예측하기 위해 팔레트 예측을 사용하는지 여부를 나타내기 위해 pred_palette_flag를 인코딩할 수도 있다. 일부 예들에 있어서, 비디오 인코더는 각각의 컬러 성분에 대한 개별 플래그 (예컨대, 각각의 블록에 대하여 3

개의 플래그들) 을 시그널링할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 블록의 모든 컬러 성분들에 적용가능한 단일 플래그를 시그널링할 수도 있다.

[0065] 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 앞서 식별된 정보를 획득하고, 그 데이터를 사용하여 팔레트를 복원할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 특정 팔레트가 다른 팔레트로부터 예측되는지 여부를 나타내는 데이터뿐만 아니라, 비디오 디코더 (30) 가 적절한 예측 팔레트 엔트리들을 사용하게 하는 정보를 수신할 수도 있다.

[0066] 일부 경우들에서, 부가적으로 또는 대안적으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트를 온 더 플라이 (on-the-fly) 로, 즉 동적으로 구성할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 코딩 동안 비어있는 팔레트에 엔트리들을 추가할 수도 있다. 즉, 비디오 인코더 (20) 는 픽셀 값들이 블록에서의 포지션들을 위해 생성되고 송신될 때, 팔레트에 픽셀 값들을 추가할 수도 있다. 블록에서 상대적으로 이후에 코딩되는 픽셀들은, 픽셀 값을 송신하는 대신, 예컨대 인덱스 값을 갖는, 팔레트의 이전에 부가된 엔트리들을 지칭할 수도 있다. 유사하게, 블록에서의 포지션에 대한 새로운 픽셀 값을 수신하면, 비디오 디코더 (30) 는 비디오 인코더 (20) 와 동일한 프로세스를 따르고, 그 픽셀 값을 팔레트에 포함시킬 수도 있다. 이러한 방식으로, 비디오 디코더 (30) 는 비디오 인코더 (20) 와 동일한 팔레트를 구성한다. 비디오 디코더 (30) 는 팔레트에 이미 포함된 값을 갖는 픽셀들에 대하여, 그 값을 식별하는 인덱스 값을 수신할 수도 있다. 비디오 디코더 (30) 는 블록의 픽셀들을 복원하기 위해, 수신된 정보, 예컨대 팔레트에 대한 픽셀 값을 및 인덱스 값을 사용할 수도 있다.

[0067] 일부 경우들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 고정된 사이즈의 팔레트를 유지할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 가장 최근의 복원된 픽셀 값을 팔레트에 복원된 것으로서 부가할 수도 있다. 팔레트에 부가되는 각각의 엔트리에 대하여, 팔레트에 가장 먼저 부가된 엔트리는 폐기된다. 이는 또한 때때로, 선입 선출 (FIFO) 로 지정된다. 팔레트를 업데이트하는 이러한 프로세스는, 오직 팔레트 모드를 사용하여 코딩된 블록들에만, 또는 코딩 모드와 관계없이 블록들 모두에 적용될 수도 있다.

[0068] 전술된 기술들은 일반적으로 팔레트-기반 코딩을 위해 팔레트를 구성하고 및/또는 송신하는 비디오 인코더 (20) 및 비디오 디코더 (30) 에 관련된다. 본 개시물의 다른 양태들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 픽셀 값을 결정하게 하는 맵을 구성하고 및/또는 송신하는 것에 관련된다. 예를 들어, 본 개시물의 다른 양태들은 특정 픽셀을 팔레트의 엔트리에 관련시키는 인덱스들의 맵을 구성하고 및/또는 송신하는 것에 관련된다.

[0069] 일부 예들에서, 비디오 인코더 (20) 는 블록의 픽셀들이 팔레트에서 대응하는 값을 가지는지 여부를 나타낼 수도 있다. 예시의 목적들을 위한 일 예에서, 맵의 (i, j) 엔트리는 비디오 데이터의 블록에서 (i, j) 픽셀 포지션에 대응하는 것으로 가정한다. 이러한 예에서, 비디오 인코더 (20) 는 블록의 각 픽셀 포지션에 대한 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 (i, j) 위치에서의 픽셀 값이 팔레트에서의 값을 중 하나인 것을 나타내기 위해, (i, j) 엔트리에 대하여 1 과 동일한 플래그를 세팅할 수도 있다. 컬러가 팔레트에 포함될 경우 (즉, 플래그가 1 과 동일할 경우), 비디오 인코더 (20) 는 또한 팔레트에서 컬러를 식별하는 (i, j) 엔트리에 대한 팔레트 인덱스를 나타내는 데이터를 인코딩할 수도 있다. 픽셀의 컬러가 팔레트에 포함되지 않을 경우 (즉, 플래그가 0 과 동일할 경우), 비디오 인코더 (20) 는 또한, 그 픽셀에 대한 샘플 값을 나타내는 데이터를 인코딩할 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 전술된 데이터를 획득하고, 그 데이터를 사용하여 블록에서의 특정 위치에 대한 팔레트 인덱스 및/또는 픽셀 값을 결정할 수도 있다.

[0070] 일부 경우들에 있어서, 소정 포지션에서의 픽셀이 맵핑되는 팔레트 인덱스와 이웃하는 픽셀이 동일한 팔레트 인덱스로 맵핑될 확률 간에 상관이 존재할 수도 있다. 즉, 픽셀이 특정 팔레트 인덱스로 맵핑될 경우, (공간적 위치와 관련하여) 하나 이상의 이웃하는 픽셀들이 동일한 팔레트 인덱스로 맵핑될 확률은 상대적으로 높을 수도 있다.

[0071] 본 개시의 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 비디오 데이터의 블록의 하나 이상의 인덱스들을 비디오 데이터의 동일한 블록의 하나 이상의 인덱스들과 관련하여 결정하고 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 비디오 데이터의 블록에서 제 1 픽셀과 연관된 제 1 인덱스 값을 결정하도록 구성될 수도 있고, 여기서 제 1 인덱스 값은 제 1 픽셀의 값을 팔레트의 엔트리에 관련시킨다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 또한, 제 1 인덱스 값에 기초하

여, 비디오 데이터의 블록에서 하나 이상의 제 2 픽셀과 연관된 하나 이상의 제 2 인덱스 값들을 결정하고, 비디오 데이터의 블록의 제 1 및 하나 이상의 제 2 픽셀들을 코딩하도록 구성될 수도 있다. 따라서, 이러한 예에서, 맵의 인덱스들은 맵의 하나 이상의 다른 인덱스들과 관련하여 코딩될 수도 있다.

[0072] 일부 예들에서, 비디오 인코더 (20)는 동일한 인덱스 값으로 맵핑되는, 소정의 스캔 순서에서 연속하는 픽셀들의 수를 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 유사-값의 인덱스 값들의 스트링은 본원에서 "런"으로 지칭될 수도 있다. 일부 예들에서, 픽셀 값은 팔레트에서 정확히 하나의 인덱스 값과 연관될 수도 있다. 따라서, 일부 경우들에 있어서, 값들의 런은 또한 유사-값의 픽셀 값들을 지칭할 수도 있다. 다른 예들에서, 이하 손실 코딩에 대하여 설명되는 것과 같이, 1 초과의 픽셀 값을 팔레트에서 동일한 인덱스 값에 맵핑할 수도 있다. 그러한 예들에서, 값들의 런은 유사-값의 인덱스 값들을 지칭한다.

[0073] 예시의 목적들을 위한 일 예에 있어서, 소정의 스캔 순서에서 2 개의 연속하는 인덱스들이 상이한 값을 갖는다면, 런은 0 과 동일하다. 소정의 스캔 순서에서 2 개의 연속하는 인덱스들이 동일한 값을 가지지만 스캔 순서에서 제 3 인덱스는 상이한 값을 갖는다면, 런은 1 과 동일하다. 비디오 디코더 (30)는 인코딩된 비트 스트림으로부터 런을 표시하는 신팩스 엘리먼트들을 획득하고, 그 데이터를 사용하여 동일한 인덱스 값을 갖는 연속하는 픽셀 위치들의 수를 결정할 수도 있다.

[0074] 부가적으로 또는 대안적으로, 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30)는 맵의 하나 이상의 엔트리들에 대하여 라인 카피를 수행할 수도 있다. 그 엔트리들은 또한, 맵의 엔트리들과 블록의 픽셀 포지션들 간의 관계로 인해 "포지션들"로 지칭될 수도 있다. 라인 카피는, 일부 예들에서, 스캔 방향에 의존할 수도 있다. 예를 들어, 비디오 인코더 (20) 가 블록에서의 특정 포지션에 대한 픽셀 값 또는 인덱스 맵 값이 (수평 스캔에 대하여) 특정 포지션 상부의 라인에서의 픽셀 또는 인덱스 값, 또는 (수직 스캔에 대하여) 특정 포지션의 좌측의 컬럼과 동일한 것을 나타낼 수도 있다. 비디오 인코더 (20)는 또한, 특정 포지션의 상부 또는 특정 포지션의 좌측의 컬럼의 대응하는 픽셀 값을 또는 인덱스들과 동일한, 스캔 순서에서의 픽셀 값을 또는 인덱스들의 수를, 런으로서 나타낼 수도 있다. 이러한 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 명시된 이웃하는 라인으로부터 및 현재 코딩되고 있는 블록의 라인에 대한 명시된 수의 엔트리들로부터 픽셀 또는 인덱스 값을 카피할 수도 있다.

[0075] 일부 경우들에 있어서, 값들이 카피되는 라인들은, 예컨대 현재 코딩되고 있는 포지션의 라인의 상부 또는 좌측에 바로 인접할 수도 있다. 다른 예들에서, 블록의 다수의 라인들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 버퍼링되어 맵의 다수의 라인들 중 임의의 것이 현재 코딩되고 있는 맵의 라인을 위한 예측 값을 사용될 수도 있다. 예시의 목적들을 위한 일 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 픽셀들의 현재 로우를 코딩하기 전에 픽셀 값들의 인덱스들의 이전 4 개 로우들을 저장하도록 구성될 수도 있다. 이러한 예에서, 예측 로우 (예컨대, 인덱스들 또는 픽셀 값들이 카피되는 로우)는 절단된 1진 (unary) 코드 또는 1진 코드들과 같은 다른 코드들과 함께 비트스트림에 표시될 수도 있다. 절단된 1진 코드에 대하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 최대 로우 계산 (예컨대, row_index-1) 또는 최대 컬럼 계산 (예컨대, column_index-1)에 기초하여 절단된 1진 코드에 대한 최대 값을 결정할 수도 있다. 추가로, 카피되는 예측 로우로부터의 포지션들의 수의 표시는 또한, 비트스트림에 포함될 수도 있다. 일부 경우들에 있어서, 현재 포지션이 예측되고 있는 컬럼 또는 라인이 다른 블록 (예컨대, CU 또는 CTU)에 속한다면, 그러한 예측은 불가능할 수도 있다.

[0076] 본 개시의 양태들에 따르면, 소위 엔트리들의 런들을 코딩하기 위한 기술들은 전술된 라인 카피를 위한 기술들과 함께 사용될 수도 있다. 예를 들어, 비디오 인코더 (20)는 맵에서의 엔트리의 값이 팔레트로부터 획득되는지, 또는 맵에서의 엔트리의 값이 맵에서 이전에 코딩된 라인으로부터 획득되는지 여부를 표시하는 하나 이상의 신팩스 엘리먼트들 (예컨대, 플래그)을 인코딩할 수도 있다. 비디오 인코더 (20)는 또한, 팔레트의 인덱스 값 또는 그 라인 (로우 또는 컬럼)에서의 엔트리의 위치를 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20)는 또한, 동일한 값을 공유하는 연속하는 엔트리들의 수를 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 비디오 디코더 (30)는 인코딩된 비트스트림으로부터 그러한 정보를 획득하고, 그 정보를 사용하여 블록에 대한 픽셀 값을 및 맵을 복원할 수도 있다.

[0077] 전술된 것과 같이, 맵의 인덱스들은 특정 순서로 스캔된다. 본 개시의 양태들에 따르면, 스캔 방향은 수직 이거나, 수평이거나, 대각 방향 (예컨대, 블록에서 대각선으로 45 도 또는 135 도)일 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 블록의 인덱스들을 스캔하기 위한 스캔 방향을 표시하는, 각 블록에 대한 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 부가적으로 또는 대안적으로, 스캔 방향은 예컨대, 블록

사이즈, 컬러 공간, 및/또는 컬러 성분과 같은 소위 부가 정보에 기초하여 시그널링되거나 추론될 수도 있다.

비디오 인코더 (20)는 블록의 각각의 컬러 성분에 대한 스캔들을 명시할 수도 있다. 대안적으로, 명시된 스캔은 블록의 모든 컬러 성분들에 적용할 수도 있다.

[0078] 본 개시의 기술들은 또한, 팔레트-기반 코딩의 다른 양태들을 포함한다. 예를 들어, 본 개시의 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 블록이 팔레트 코딩 모드를 사용하여 코딩되는 것을 표시하도록, 각각의 블록에 대한 하나 이상의 신팩스 엘리먼트들을 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팔레트-기반 코딩 모드가 특정 블록을 코딩하기 위해 사용될 것인지 여부를 표시하도록 팔레트 모드 플래그 (PLT_Mode_flag)를 코딩할 수도 있다. 이러한 예에서, 비디오 인코더 (20)는 현재 인코딩되고 있는 블록 ("현재 블록")이 팔레트 모드를 사용하여 인코딩되고 있는 것을 명시하기 위해 1 과 동일한 PLT_Mode_flag 을 인코딩할 수도 있다. 0 과 동일한 PLT_Mode_flag 의 값은, 현재 블록이 팔레트 모드를 사용하여 인코딩되지 않는 것을 명시한다. 이 경우, 비디오 디코더 (30)는 인코딩된 비트스트림으로부터 PLT_Mode_flag 를 획득하고, 그 블록을 디코딩하기 위해 팔레트-기반 코딩 모드를 적용할 수도 있다. 1 초과의 팔레트-기반 코딩 모드가 사용가능한 (예컨대, 코딩을 위해 1 초과의 팔레트-기반 기술이 사용가능한) 경우들에 있어서, 하나 이상의 신팩스 엘리먼트들은 블록에 대해 복수의 상이한 팔레트 모드들 중 하나를 표시할 수도 있다.

[0079] 일부 경우들에서, 비디오 인코더 (20)는 현재 블록이 팔레트 모드를 사용하여 인코딩되지 않는 것을 명시하기 위해 0 과 동일한 PLT_Mode_flag 를 인코딩할 수도 있다. 그러한 경우들에서, 비디오 인코더 (20)는 다양한 인터-예측, 인트라-예측 또는 다른 코딩 모드들 중 임의의 코딩 모드를 사용하여 블록을 인코딩할 수도 있다. PLT_Mode_flag 가 0 과 동일할 경우, 비디오 인코더 (20)는 개별 블록을 인코딩하기 위해 사용되는 특정 모드를 표시하기 위해 추가의 정보 (예컨대, 신팩스 엘리먼트들) 을 송신할 수도 있다. 일부 예들에서, 이하 설명되는 것과 같이, 모드는 HEVC 코딩 모드일 수도 있다. PLT_Mode_flag 의 사용은 예시의 목적들을 위해 설명된다. 다른 예들에서, 멀티-비트 코드들과 같은 다른 신팩스 엘리먼트들은 팔레트-기반의 코딩 모드가 하나 이상의 블록들을 위해 사용될 것인지 여부를 표시하거나, 또는 복수의 모드들 중 어느 것이 사용될 것인지를 표시하기 위해 사용될 수도 있다.

[0080] 팔레트-기반 코딩 모드가 사용될 경우, 팔레트는 비디오 디코더 (30)에 의한 사용을 위해 인코딩된 비디오 데이터 스트림에서 예컨대, 본원에 설명된 기술들 중 하나 이상을 사용하여 비디오 인코더 (20)에 의해 송신된다. 팔레트는 각각의 블록에 대하여 송신될 수도 있거나, 또는 다수의 블록들 중에서 공유될 수도 있다. 팔레트는 블록에 대하여 우세적인 및/또는 대표적인 다수의 픽셀 값들을 지칭할 수도 있다.

[0081] 본 개시의 양태들에 따르면, 예컨대, 팔레트에 포함된 픽셀 값들의 수에 대하여, 팔레트의 사이즈는 고정될 수도 있거나, 또는 인코딩된 비트스트림에서 하나 이상의 신팩스 엘리먼트들을 사용하여 시그널링될 수도 있다. 이하 더 상세히 설명되는 것과 같이, 픽셀 값은 예컨대, 코딩을 위해 사용된 컬러 공간에 의존하여 다수의 샘플들로 구성될 수도 있다. 예를 들어, 픽셀 값은 루마 및 크로미넌스 샘플들 (예컨대, 루마, U 크로미넌스 및 V 크로미넌스 (YUV) 또는 루마, Cb 크로미넌스, 및 Cr 크로미넌스 (YCbCr) 샘플들) 을 포함할 수도 있다. 다른 예에서, 픽셀 값은 적색, 녹색 및 청색 (RGB) 샘플들을 포함할 수도 있다. 본원에 설명된 것과 같이, 용어 픽셀 값은 일반적으로 픽셀에 기여하는 샘플들 중 하나 이상을 지칭할 수도 있다. 즉, 용어 픽셀 값은 반드시 픽셀에 기여하는 모든 샘플들을 지칭하는 것이 아니라, 픽셀에 기여하는 단일 샘플 값을 설명하는데 사용될 수도 있다.

[0082] 본 개시의 양태들에 따르면, 팔레트는 특정 블록의 각 컬러 성분에 대하여 개별적으로 송신될 수도 있다. 예를 들어, YUV 컬러 공간에서, (Y 값들을 표현하는) Y 성분에 대한 팔레트, (U 값들을 표현하는) U 성분에 대한 다른 팔레트, 및 (V 값들을 표현하는) V 성분에 대한 또 다른 팔레트가 존재할 수도 있다. 다른 예에서, 팔레트는 특정 블록의 모든 성분들을 포함할 수도 있다. 이러한 예에서, 팔레트에서 i 번째 엔트리는 3 개의 값을 (예컨대, Yi, Ui, Vi) 을 포함할 수도 있다. 본 개시물의 양태들에 따르면, 하나 이상의 신팩스 엘리먼트들은 각각의 성분 (예컨대, Y, U, V, 등등)에 대하여 팔레트의 사이즈를 개별적으로 표시할 수도 있다. 다른 예들에서, 모든 성분들에 대하여 단일 사이즈가 사용될 수도 있으며, 따라서 하나 이상의 신팩스 엘리먼트들은 모든 성분들의 사이즈를 표시한다.

[0083] 본 개시의 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더는 순실 방식 또는 무순실 방식으로 팔레트-기반 코딩을 수행할 수도 있다. 즉, 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 블록의 픽셀 값들에 매칭하는 팔레트 엔트리들을 사용하여 (또는 픽셀 값이 팔레트에 포함되지 않은 경우,

실제 픽셀 값들을 전송함으로써 블록에 대한 비디오 데이터를 손실없이 코딩할 수도 있다. 다른 예들에서, 이하 도 5 와 관련하여 더 상세히 설명되는 것과 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 블록의 픽셀 값들에 정확히 매칭하지 않는 팔레트 엔트리들을 사용하여 블록에 대한 비디오 데이터를 코딩할 수도 있다 (손실 코딩).

[0084] 일부 예들에서, 비디오 데이터의 팔레트-기반 코딩을 위한 기술들은 인터-예측 코딩 또는 인트라-예측 코딩을 위한 기술들과 같은 하나 이상의 다른 코딩 기술들과 함께 사용될 수도 있다. 예를 들어, 이하 더 상세히 설명되는 것과 같이, 인코더 또는 디코더, 또는 결합된 인코더-디코더 (코덱) 는 인터-예측 코딩 및 인트라-예측 코딩뿐만 아니라 팔레트-기반 코딩을 수행하도록 구성될 수도 있다.

[0085] 도 2 는 본 개시의 기술들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 도시한 블록 다이어그램이다. 도 2 는 설명의 목적들로 제공되며, 본 개시에서 대략적으로 예시화되고 설명된 바와 같은 기술들의 한정으로서 고려되지 않아야 한다. 설명의 목적들로, 본 개시는 HEVC 코딩의 맥락에서 비디오 인코더 (20) 를 설명한다. 하지만, 본 개시의 기술들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0086] 비디오 인코더 (20) 는 본 개시에 설명된 다양한 예들에 따른 팔레트-기반 비디오 코딩을 위한 기술들을 수행하도록 구성될 수도 있는 디바이스의 일 예를 나타낸다. 예를 들어, 비디오 인코더 (20) 는 팔레트-기반 코딩 또는 비-팔레트 기반 코딩을 사용하여, HEVC 코딩에서 CU들 및 PU들과 같은 비디오 데이터의 다양한 블록들을 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트 기반 코딩 모드들은 HEVC 드래프트 10 에 의해 명시된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간 코딩 모드들 또는 인트라-예측 공간 코딩 모드들을 지칭할 수도 있다. 비디오 인코더 (20) 는 일 예에서, 픽셀 값들을 표시하는 엔트리들을 갖는 팔레트를 생성하고, 비디오 데이터의 블록의 적어도 일부 포지션들의 픽셀 값들을 나타내기 위해 팔레트에서 픽셀 값을 선택하며, 각각 선택된 픽셀 값들에 대응하는 팔레트에서의 엔트리들과 비디오 데이터의 블록의 포지션들 중 적어도 일부를 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 시그널링된 정보는 비디오 데이터를 디코딩하기 위해 비디오 디코더 (30) 에 의해 사용될 수도 있다.

[0087] 도 2 의 예에 있어서, 비디오 인코더 (20) 는 예측 프로세싱 유닛 (100), 잔차 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 프로세싱 유닛 (110), 복원 유닛 (112), 필터 유닛 (114), 디코딩된 픽처 베퍼 (116), 및 엔트로피 인코딩 유닛 (118) 을 포함한다. 예측 프로세싱 유닛 (100) 은 인터 예측 프로세싱 유닛 (120) 및 인트라 예측 프로세싱 유닛 (126) 을 포함한다. 인터 예측 프로세싱 유닛 (120) 은 모션 추정 유닛 및 모션 보상 유닛 (비도시) 을 포함한다. 비디오 인코더 (20) 는 또한, 본 개시에 설명된 팔레트-기반 코딩 기술들의 다양한 양태들을 수행하도록 구성된 팔레트-기반 인코딩 유닛 (122) 을 포함한다. 다른 예들에 있어서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능 컴포넌트들을 포함할 수도 있다.

[0088] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터의 픽처의 슬라이스에서 각각의 CTU 를 인코딩할 수도 있다. CTU들 각각은, 픽처의 동일하게 사이징된 루마 코딩 트리 블록들 (CTB들) 및 대응하는 CTB들과 연관될 수도 있다. CTU 를 인코딩하는 부분으로서, 예측 프로세싱 유닛 (100) 은 큐드 트리 파티셔닝을 수행하여, CTU 의 CTB들을 점진적으로 더 작은 블록들로 분할할 수도 있다. 더 작은 블록은 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 CTU 와 연관된 CTB 를 4 개의 동일하게 사이징된 서브블록들로 파티셔닝할 수도 있고, 그 서브블록들 중 하나 이상을 동일하게 사이징된 서브-서브블록들로 파티셔닝할 수도 있는 등등이다.

[0089] 비디오 인코더 (20) 는 CTU 의 CU들을 인코딩하여 CU들의 인코딩된 표현들 (즉, 코딩된 CU들) 을 생성할 수도 있다. CU 를 인코딩하는 부분으로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU들 중에서 CU 와 연관된 코딩 블록들을 파티셔닝할 수도 있다. 따라서, 각각의 PU 는 루마 예측 블록 및 대응하는 크로마 예측 블록들과 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다양한 사이즈들을 갖는 PU 들을 지원할 수도 있다. 상기 나타낸 바와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있으며, PU 의 사이즈는 PU 의 루마 예측 블록의 사이즈를 지칭할 수도 있다. 특정 CU 의 사이즈가 $2Nx2N$ 이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인트라 예측을 위해 $2Nx2N$ 또는 NxN 의 PU 사이즈들을 지원하고, 인터 예측을 위해 $2Nx2N$, $2NxN$, $Nx2N$, NxN 등의 대칭적인 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 인터 예측을 위해 $2NxN$, $2NxN$, $nLx2N$, 및 $nRx2N$ 의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

[0090] 인터-예측 프로세싱 유닛 (120) 은 CU 의 각각의 PU 에 인터 예측을 수행함으로써 PU 에 대한 예측 데이터를 생

성할 수도 있다. PU에 대한 예측 데이터는, PU의 예측 샘플 블록들 및 PU에 대한 모션 정보를 포함할 수도 있다. 인터-예측 유닛 (121)은, PU가 I 슬라이스인지, P 슬라이스인지 또는 B 슬라이스인지에 의존하여 CU의 PU에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에 있어서, 모든 PU들은 인트라 예측된다. 따라서, PU가 I 슬라이스에 있으면, 인터 예측 유닛 (121)은 PU에 대해 인터 예측을 수행하지 않는다. 따라서, I-모드에서 인코딩된 블록들에 대해, 예측된 블록은 동일한 프레임 내에서 이전에 인코딩된 이웃하는 블록들로부터의 공간 예측을 사용하여 형성된다.

[0091] PU가 P 슬라이스에 있다면, 인터-예측 프로세싱 유닛 (120)의 모션 추정 유닛은 PU에 대한 참조 영역을 위해 참조 픽처들의 리스트 (예를 들어, "RefPicList0")에서 참조 픽처들을 탐색할 수도 있다. PU에 대한 참조 영역은, 참조 픽처 내에서, PU의 샘플 블록들에 가장 근접하게 대응하는 샘플 블록들을 포함하는 영역일 수도 있다. 모션 추정 유닛은, PU에 대한 참조 영역을 포함하는 참조 픽처의 RefPicList0에서의 포지션을 표시하는 참조 인덱스를 생성할 수도 있다. 부가적으로, 모션 추정 유닛은, 참조 영역과 연관된 참조 위치와 PU의 코딩 블록 간의 공간 변위를 표시하는 MV를 생성할 수도 있다. 예를 들어, MV는, 현재 디코딩된 픽처에서의 좌표들로부터 참조 픽처에서의 좌표들까지의 오프셋을 제공하는 2 차원 벡터일 수도 있다. 모션 추정 유닛은 PU의 모션 정보로서 참조 인덱스 및 MV를 출력할 수도 있다. 인터-예측 프로세싱 유닛 (120)의 모션 보상 유닛은 PU의 모션 벡터에 의해 표시된 참조 위치에서의 실제 및 보간된 샘플들에 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.

[0092] PU가 B 슬라이스에 있다면, 모션 추정 유닛은 PU에 대해 단방향 예측 또는 양방향 예측을 수행할 수도 있다. PU에 대해 단방향 예측을 수행하기 위해, 모션 추정 유닛은 PU에 대한 참조 영역을 위해 RefPicList0 또는 제 2 참조 픽처 리스트 ("RefPicList1")의 참조 픽처들을 탐색할 수도 있다. 모션 추정 유닛은, PU의 모션 정보로서, 참조 영역을 포함하는 참조 픽처의 RefPicList0 또는 RefPicList1에서의 포지션을 표시하는 참조 인덱스, PU의 샘플 블록과 참조 영역과 연관된 참조 위치 간의 공간 변위를 표시하는 MV, 및 참조 픽처가 RefPicList0에 있는지 또는 RefPicList1에 있는지를 표시하는 하나 이상의 예측 방향 표시자들을 출력할 수도 있다. 인터-예측 프로세싱 유닛 (120)의 모션 보상 유닛은 PU의 모션 벡터에 의해 표시된 참조 영역에서의 실제 및 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.

[0093] PU에 대해 양방향 인터 예측을 수행하기 위해, 모션 추정 유닛은 PU에 대한 참조 영역을 위해 RefPicList0에서의 참조 픽처들을 탐색할 수도 있고, 또한 PU에 대한 다른 참조 영역을 위해 RefPicList1에서의 참조 픽처들을 탐색할 수도 있다. 모션 추정 유닛은, 참조 영역들을 포함하는 참조 픽처들의 RefPicList0 및 RefPicList1에서의 포지션들을 표시하는 참조 픽처 인덱스들을 생성할 수도 있다. 부가적으로, 모션 추정 유닛은, PU의 샘플 블록과 참조 영역들과 연관된 참조 위치 간의 공간 변위들을 표시하는 MV들을 생성할 수도 있다. PU의 모션 정보는 PU의 MV들 및 참조 인덱스들을 포함할 수도 있다. 모션 보상 유닛은 PU의 모션 벡터에 의해 표시된 참조 영역에서의 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.

[0094] 본 개시의 다양한 예들에 따르면, 비디오 인코더 (20)는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다. HEVC 프레임워크에 대하여, 일 예로서, 팔레트-기반 코딩 기술들은 코딩 유닛 (CU) 모드로서 사용되도록 구성될 수도 있다. 다른 예들에서, 팔레트-기반 코딩 기술들은 HEVC의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 따라서, CU 모드의 맥락에서 본원에 (본 개시 전반에 걸쳐) 설명되는 개시된 프로세스들 전부는, 부가적으로 또는 대안적으로, PU에 적용할 수도 있다. 그러나, 이들 HEVC-기반 예들은 본원에 설명된 팔레트-기반 코딩 기술들의 제한 또는 한정으로 고려되어서는 안되며, 이는 그러한 기술들이 다른 기존의 또는 아직 개발되지 않은 시스템들/표준들과 독립적으로 또는 이들의 부분으로서 작용하도록 적용될 수도 있기 때문이다. 이들 케이스들에서, 팔레트 코딩을 위한 유닛은 정사각형 블록들, 직사각형 블록들 또는 심지어 직사각형 형상이 아닌 영역들일 수 있다.

[0095] 팔레트-기반 인코딩 유닛 (122)은 예컨대, 팔레트-기반 인코딩 모드가 예컨대, CU 또는 PU에 대하여 선택될 경우, 팔레트-기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛 (122)은, 픽셀 값들을 표시하는 엔트리들을 갖는 팔레트를 생성하고, 비디오 데이터의 블록의 적어도 일부 포지션들의 픽셀 값들을 나타내기 위해 팔레트에서 픽셀 값들을 선택하며, 각각 선택된 픽셀 값들에 대응하는 팔레트에서의 엔트리들과 비디오 데이터의 블록의 포지션들 중 적어도 일부를 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 인코딩 유닛 (122)에 의해 수행되고 있는 것으로 설명되지만, 그러한 기능들의 일부 또는 전부는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

- [0096] 팔레트-기반 인코딩 유닛 (122) 은 본원에 설명된 다양한 십екс 엘리먼트들 중 임의의 것을 생성하도록 구성될 수도 있다. 따라서, 비디오 인코더 (20) 는 본 개시에 설명된 것과 같은 팔레트-기반 코드 모드들을 사용하여 비디오 데이터의 블록들을 인코딩하도록 구성될 수도 있다. 비디오 인코더 (20) 는 선택적으로 팔레트 코딩 모드를 사용하여 비디오 데이터의 블록을 인코딩하거나, 또는 예컨대, HEVC 인터-예측 또는 인트라-예측 코딩 모드와 같은 상이한 모드를 사용하여 비디오 데이터의 블록을 인코딩할 수도 있다. 비디오 데이터의 블록은 예컨대, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU 일 수도 있다. 비디오 인코더 (20) 는 인터-예측 시간 예측 또는 인트라-예측 공간 코딩 모드들로 일부 블록들을 인코딩하고, 팔레트-기반 코딩 모드로 다른 블록들을 디코딩할 수도 있다.
- [0097] 인트라-예측 프로세싱 유닛 (126) 은 PU 에 인트라 예측을 수행함으로써 PU 에 대한 예측 데이터를 생성할 수도 있다. PU 에 대한 예측 데이터는, PU 에 대한 예측 샘플 블록들 및 다양한 십екс 엘리먼트들을 포함할 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에 있어서의 PU들에 인트라 예측을 수행할 수도 있다.
- [0098] PU 에 인트라 예측을 수행하기 위해, 인트라-예측 프로세싱 유닛 (126) 은 다중의 인트라 예측 모드들을 이용하여, PU 에 대한 예측 데이터의 다중의 세트들을 생성할 수도 있다. 인트라 예측 모드를 이용하여 PU 에 대한 예측 데이터의 세트를 생성하기 위해, 인트라-예측 프로세싱 유닛 (126) 은 인트라 예측 모드와 연관된 방향으로 PU 의 샘플 블록들에 걸쳐 이웃하는 PU들의 샘플 블록들로부터의 샘플들을 확장할 수도 있다. 이웃하는 PU들은, PU들, CU들, 및 CTU들에 대한 좌측-우측으로, 상부-하부로의 인코딩 순서를 가정할 때, PU 의 상부, 상부 우측으로, 상부 좌측으로, 또는 좌측으로일 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 다양한 개수들의 인트라 예측 모드들, 예를 들어, 33 개의 지향성 인트라 예측 모드들을 이용할 수도 있다. 일부 예들에 있어서, 인트라 예측 모드들의 수는 PU 와 연관된 영역의 사이즈에 의존할 수도 있다.
- [0099] 예측 프로세싱 유닛 (100) 은 PU들에 대한 인터-예측 프로세싱 유닛 (120) 에 의해 생성된 예측 데이터 또는 PU 들에 대한 인트라-예측 프로세싱 유닛 (126) 에 의해 생성된 예측 데이터 중으로부터 CU 의 PU들에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에 있어서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 CU 의 PU들에 대한 예측 데이터를 선택한다. 선택된 예측 데이터의 예측 샘플 블록들은 본원에서 선택된 예측 샘플 블록들로서 지칭될 수도 있다.
- [0100] 잔차 생성 유닛 (102) 은, CU 의 루마, Cb, 및 Cr 코딩 블록 및 CU 의 PU들의 선택된 예측 루마, Cb, 및 Cr 블록들에 기초하여, CU 의 루마, Cb, 및 Cr 잔차 블록들을 생성할 수도 있다. 예를 들어, 잔차 생성 유닛 (102) 은, 잔차 블록들에서의 각각의 샘플이 CU 의 코딩 블록에서의 샘플과 CU 의 PU 의 대응하는 선택된 예측 샘플 블록에서의 대응하는 샘플 간의 차이와 동일한 값을 갖도록, CU 의 잔차 블록들을 생성할 수도 있다.
- [0101] 변환 프로세싱 유닛 (104) 은 퀼드 트리 파티셔닝을 수행하여, CU 와 연관된 잔차 블록들을 CU 의 TU들과 연관된 변환 블록들로 파티셔닝할 수도 있다. 따라서, TU 는 루마 변환 블록 및 2 개의 크로마 변환 블록들과 연관될 수도 있다. CU 의 TU들의 루마 및 크로마 변환 블록들의 사이즈들 및 포지션들은 CU 의 PU들의 예측 블록들의 사이즈들 및 포지션들에 기초할 수도 있거나 기초하지 않을 수도 있다. "잔차 퀼드 트리" (RQT) 로서 공지된 퀼드 트리 구조는 그 영역들 각각과 연관된 노드들을 포함할 수도 있다. CU 의 TU들은 RQT 의 리프 노드들에 대응할 수도 있다.
- [0102] 변환 프로세싱 유닛 (104) 은 TU 의 변환 블록들에 하나 이상의 변환들을 적용함으로써 CU 의 각각의 TU 에 대한 변환 계수 블록들을 생성할 수도 있다. 변환 프로세싱 유닛 (104) 은 TU 와 연관된 변환 블록에 다양한 변환들을 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 이산 코사인 변환 (DCT), 지향성 변환, 또는 개념적으로 유사한 변환을 변환 블록에 적용할 수도 있다. 일부 예들에 있어서, 변환 프로세싱 유닛 (104) 은 변환 블록에 변환들을 적용하지 않는다. 그러한 예들에 있어서, 변환 블록은 변환 계수 블록으로서 처리될 수도 있다.
- [0103] 양자화 유닛 (106) 은 계수 블록에 있어서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 변환 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안 m-비트 변환 계수로 라운드 다운될 수도 있으며, 여기서, n 은 m 보다 크다. 양자화 유닛 (106) 은 CU 와 연관된 양자화 파라미터 (QP) 값에 기초하여 CU 의 TU 와 연관된 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 CU 와 연관된 QP 값을 조정함으로써 CU 와 연관된 계수 블록들에 적용된 양자화도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서, 양자화된 변환 계수들은 오리지널

계수들보다 더 낮은 정확도를 가질 수도 있다.

[0104] 역양자화 유닛 (108) 및 역변환 프로세싱 유닛 (110) 은, 각각, 계수 블록에 역양자화 및 역변환을 적용하여, 그 계수 블록으로부터 잔차 블록을 복원할 수도 있다. 복원 유닛 (112) 은 복원된 잔차 블록을, 예측 프로세싱 유닛 (100) 에 의해 생성된 하나 이상의 예측 샘플 블록들로부터의 대응하는 샘플들에 가산하여, TU 와 연관된 복원된 변환 블록을 생성할 수도 있다. 이러한 방식으로 CU 의 각각의 TU 에 대한 변환 블록들을 복원함으로써, 비디오 인코더 (20) 는 CU 의 코딩 블록들을 복원할 수도 있다.

[0105] 필터 유닛 (114) 은 하나 이상의 디블록킹 (deblocking) 동작들을 수행하여, CU 와 연관된 코딩 블록들에서의 블록킹 아티팩트들을 감소시킬 수도 있다. 디코딩된 픽처 버퍼 (116) 는, 필터 유닛 (114) 이 복원된 코딩 블록들에 대해 하나 이상의 디블록킹 동작들을 수행한 이후 복원된 코딩 블록들을 저장할 수도 있다. 인터-예측 프로세싱 유닛 (120) 은 복원된 코딩 블록들을 포함하는 참조 픽처를 이용하여, 다른 픽처들의 PU들에 인터 예측을 수행할 수도 있다. 부가적으로, 인트라-예측 프로세싱 유닛 (126) 은 디코딩된 픽처 버퍼 (116) 에 있어서의 복원된 코딩 블록들을 이용하여, CU 와 동일한 픽처에 있어서의 다른 PU들에 대해 인트라 예측을 수행할 수도 있다.

[0106] 엔트로피 인코딩 유닛 (118) 은 비디오 인코더 (20) 의 다른 기능 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 양자화 유닛 (106) 으로부터 계수 블록들을 수신할 수도 있고, 예측 프로세싱 유닛 (100) 으로부터 신택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (118) 은 데이터에 하나 이상의 엔트로피 인코딩 동작들을 수행하여 엔트로피 인코딩된 데이터를 생성할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 컨텍스트 적응형 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 신택스 기반 컨텍스트 적응형 바이너리 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수 골롬 (Exponential-Golomb) 인코딩 동작, 또는 다른 타입의 엔트로피 인코딩 동작을 데이터에 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩 유닛 (118) 에 의해 생성된 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들어, 그 비트스트림은 CU 에 대한 RQT 를 나타내는 데이터를 포함할 수도 있다.

[0107] 일부 예들에서, 잔차 코딩은 팔레트 코딩과 함께 수행되지 않는다. 따라서, 비디오 인코더 (20) 는 팔레트 코딩 모드를 사용하여 코딩할 경우, 변환 또는 양자화를 수행하지 않을 수도 있다. 추가로, 비디오 인코더 (20) 는 잔차 데이터와 별개로 팔레트 코딩 모드를 사용하여 생성된 데이터를 엔트로피 인코딩할 수도 있다.

[0108] 도 3 은 본 개시의 기술들을 구현하도록 구성된 예시적인 비디오 디코더 (30) 를 도시한 블록 다이어그램이다. 도 3 은 설명의 목적들로 제공되며, 본 개시에 대략적으로 예시화되고 설명된 바와 같은 기술들에 대한 한정이 아니다. 설명의 목적들로, 본 개시는 HEVC 코딩의 맥락에서 비디오 디코더 (30) 를 설명한다. 하지만, 본 개시의 기술들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0109] 비디오 디코더 (30) 는 본 개시에 설명된 다양한 예들에 따른 팔레트-기반 비디오 코딩을 위한 기술들을 수행하도록 구성될 수도 있는 디바이스의 일 예를 나타낸다. 예를 들어, 비디오 디코더 (30) 는 팔레트-기반 코딩 또는 비-팔레트 기반 코딩을 사용하여, HEVC 코딩에서 CU들 및 PU들과 같은 비디오 데이터의 다양한 블록들을 선택적으로 디코딩하도록 구성될 수도 있다. 비-팔레트 기반 코딩 모드들은 HEVC 드래프트 10 에 의해 명시된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간 코딩 모드들 또는 인트라-예측 공간 코딩 모드들을 지칭할 수도 있다. 비디오 디코더 (30) 는 일 예에서, 픽셀 값들을 표시하는 엔트리들을 갖는 팔레트를 생성하고, 비디오 데이터의 블록의 적어도 일부 포지션들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신하고, 그 정보에 기초하여 팔레트에서 픽셀 값들을 선택하고, 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 복원하도록 구성될 수도 있다.

[0110] 도 3 의 예에 있어서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 복원 유닛 (158), 필터 유닛 (160), 및 디코딩된 픽처 버퍼 (162) 를 포함한다. 예측 프로세싱 유닛 (152) 은 모션 보상 유닛 (164) 및 인트라-예측 프로세싱 유닛 (166) 을 포함한다. 비디오 디코더 (30) 는 또한, 본 개시에 설명된 팔레트-기반 코딩 기술들의 다양한 양태들을 수행하도록 구성된 팔레트-기반 디코딩 유닛 (165) 를 포함한다. 다른 예들에 있어서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능 컴포넌트들을 포함할 수도 있다.

[0111] 코딩된 픽처 버퍼 (CPB) 는 비트스트림의 인코딩된 비디오 데이터 (예를 들어, NAL 유닛들) 를 수신 및 저장할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 CPB 로부터

수신하고, NAL 유닛들을 파싱하여 신택스 엘리먼트들을 디코딩할 수도 있다. 엔트로피 디코딩 유닛 (150)은 NAL 유닛들에 있어서의 엔트로피 인코딩된 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 복원 유닛 (158), 및 필터 유닛 (160)은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0112] 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 부분으로서, 엔트로피 디코딩 유닛 (150)은 코딩된 슬라이스 NAL 유닛들로부터 신택스 엘리먼트들을 추출 및 엔트로피 디코딩할 수도 있다. 디코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관련된 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에 있어서의 신택스 엘리먼트들은, 슬라이스를 포함하는 화상과 연관된 PPS를 식별하는 신택스 엘리먼트를 포함할 수도 있다.

[0113] 비트스트림으로부터 신택스 엘리먼트들을 디코딩하는 것에 부가하여, 비디오 디코더 (30)는 비-파티셔닝된 CU에 복원 동작을 수행할 수도 있다. 비-파티셔닝된 CU에 복원 동작을 수행하기 위해, 비디오 디코더 (30)는 CU의 각각의 TU에 복원 동작을 수행할 수도 있다. CU의 각각의 TU에 대해 복원 동작을 수행함으로써, 비디오 디코더 (30)는 CU의 잔차 블록들을 복원할 수도 있다.

[0114] CU의 각각의 TU에 복원 동작을 수행하는 부분으로서, 역양자화 유닛 (154)은 TU와 연관된 계수 블록들을 역양자화, 즉, 텔양자화할 수도 있다. 역양자화 유닛 (154)은 TU의 CU와 연관된 QP 값을 이용하여, 양자화도 및 적용할 역양자화 유닛 (154)에 대한 역양자화도를 결정할 수도 있다. 즉, 압축 비율, 즉, 오리지널 시퀀스 및 압축된 시퀀스를 나타내는데 사용된 비트들의 수의 비율은, 변환 계수들을 양자화할 경우에 사용된 QP의 값을 조정함으로써 제어될 수도 있다. 압축 비율은 또한, 채용된 엔트로피 코딩의 방법에 의존할 수도 있다.

[0115] 역양자화 유닛 (154)이 계수 블록을 역양자화한 이후, 역변환 프로세싱 유닛 (156)은 TU와 연관된 잔차 블록을 생성하기 위해 계수 블록에 하나 이상의 역변환들을 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (156)은 역 DCT, 역 정수 변환, 역 KLT (Karhunen-Loeve transform), 역 회전 변환, 역 지향성 변환, 또는 다른 역변환을 계수 블록에 적용할 수도 있다.

[0116] PU가 인트라 예측을 이용하여 인코딩되면, 인트라-예측 프로세싱 유닛 (166)은 인트라 예측을 수행하여 PU에 대한 예측 블록들을 생성할 수도 있다. 인트라-예측 프로세싱 유닛 (166)은 인트라 예측 모드를 이용하여, 공간적으로 이웃하는 PU들의 예측 블록들에 기초하여 PU에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다. 인트라-예측 프로세싱 유닛 (166)은 비트스트림으로부터 디코딩된 하나 이상의 신택스 엘리먼트들에 기초하여 PU에 대한 인트라 예측 모드를 결정할 수도 있다.

[0117] 예측 프로세싱 유닛 (152)은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 제 1 참조 꽂쳐 리스트 (RefPicList0) 및 제 2 참조 꽂쳐 리스트 (RefPicList1)를 구성할 수도 있다. 더욱이, PU가 인터 예측을 이용하여 인코딩되면, 엔트로피 디코딩 유닛 (150)은 PU에 대한 모션 정보를 추출할 수도 있다. 모션 보상 유닛 (164)은, PU의 모션 정보에 기초하여, PU에 대한 하나 이상의 참조 영역들을 결정할 수도 있다. 모션 보상 유닛 (164)은, PU에 대한 하나 이상의 참조 블록들에서의 샘플들에 기초하여, PU에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.

[0118] 복원 유닛 (158)은 CU의 TU들과 연관된 루마, Cb, 및 Cr 변환 블록들 및 CU의 PU들의 예측 루마, Cb, 및 Cr 블록들, 즉, 적용 가능한 것과 같이 인트라-예측 데이터 또는 인터-예측 데이터를 이용하여, CU의 루마, Cb, 및 Cr 코딩 블록들을 복원할 수도 있다. 예를 들어, 복원 유닛 (158)은 루마, Cb, 및 Cr 변환 블록들의 샘플들을 예측 루마, Cb, 및 Cr 블록들의 대응하는 샘플들에 가산하여, CU의 루마, Cb, 및 Cr 코딩 블록들을 복원할 수도 있다.

[0119] 필터 유닛 (160)은 디블록킹 동작을 수행하여, CU의 루마, Cb, 및 Cr 코딩 블록들과 연관된 블록킹 아티팩트들을 감소시킬 수도 있다. 비디오 디코더 (30)는 CU의 루마, Cb, 및 Cr 코딩 블록들을 디코딩된 꽂쳐 버퍼 (162)에 저장할 수도 있다. 디코딩된 꽂쳐 버퍼 (162)는 후속 모션 보상, 인트라 예측, 및 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상으로의 프리젠테이션을 위한 참조 꽂쳐들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30)는, 디코딩된 꽂쳐 버퍼 (162)에서의 루마, Cb 및 Cr 블록들에 기초하여, 다른 CU들의 PU들에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다. 이러한 방식으로, 비디오 디코더 (30)는, 비트스트림으로부터, 유효 루마 계수 블록의 변환 계수 레벨들을 추출하고, 변환

계수 레벨들을 역양자화하고, 변환 계수 레벨들에 변환을 적용하여 변환 블록을 생성하고, 변환 블록에 적어도 부분적으로 기초하여, 코딩 블록을 생성하며, 디스플레이를 위해 코딩 블록을 출력할 수도 있다.

[0120] 본 개시의 다양한 예들에 따르면, 비디오 디코더 (30)는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다.

팔레트-기반 디코딩 유닛 (165)은 예컨대, 팔레트-기반 디코딩 모드가 예컨대, CU 또는 PU에 대하여 선택될 경우, 팔레트-기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165)은, 픽셀 값들을 표시하는 엔트리들을 갖는 팔레트를 생성하고, 비디오 데이터의 블록의 적어도 일부 포지션들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신하고, 그 정보에 기초하여 팔레트에서 픽셀 값들을 선택하고, 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 복원하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 디코딩 유닛 (165)에 의해 수행되고 있는 것으로 설명되지만, 그러한 기능들의 일부 또는 전부는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0121] 팔레트-기반 디코딩 유닛 (165)은 팔레트 코딩 모드 정보를 수신할 수도 있고, 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용되는 것을 표시할 경우, 상기 동작들을 수행할 수도 있다. 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용되지 않는 것을 표시할 경우, 또는 다른 모드 정보가 상이한 모드의 사용을 표시할 경우, 팔레트-기반 디코딩 유닛 (165)은 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용되지 않는 것을 표시할 경우, 예컨대, HEVC 인터-예측 또는 인트라-예측 코딩 모드와 같은 비-팔레트 기반의 코딩 모드를 사용하여 비디오 데이터의 블록을 디코딩한다. 비디오 데이터의 블록은 예컨대, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU 일 수도 있다. 비디오 디코더 (30)는 인터-예측 시간 예측 또는 인트라-예측 공간 코딩 모드들로 일부 블록들을 디코딩하고, 팔레트-기반 코딩 모드로 다른 블록들을 디코딩할 수도 있다. 팔레트-기반 코딩 모드는 복수의 상이한 팔레트-기반 코딩 모드들 중 하나를 포함할 수도 있거나, 단일 팔레트-기반 코딩 모드가 존재할 수도 있다.

[0122] 팔레트-기반 디코딩 유닛 (165)에 의해 수신된 팔레트 코딩 모드 정보는 플래그와 같은 팔레트 모드 선택스 엘리먼트를 포함할 수도 있다. 팔레트 모드 선택스 엘리먼트의 제 1 값은 팔레트 코딩 모드가 블록에 적용되는 것을 표시하고, 팔레트 모드 선택스 엘리먼트의 제 2 값은 팔레트 코딩 모드가 비디오 데이터의 블록에 적용되지 않는 것을 표시한다. 팔레트-기반 디코딩 유닛 (165)은 (예컨대, 비디오 인코더 (20)로부터) 예측 유닛 레벨, 코딩 유닛 레벨, 슬라이스 레벨, 또는 픽처 레벨 중 하나 이상에서 팔레트 코딩 모드 정보를 수신할 수도 있거나, 또는 팔레트 코딩 모드가 픽처 파라미터 세트 (PPS), 시퀀스 파라미터 세트 (SPS) 또는 비디오 파라미터 세트 (VPS) 중 적어도 하나에서 가능한지 여부의 표시를 수신할 수도 있다.

[0123] 일부 예들에서, 팔레트-기반 디코딩 유닛 (165)은 코딩 블록의 사이즈, 프레임 타입, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 비디오 데이터의 블록과 연관된 멀티-뷰 코딩에서의 뷰 id 중 하나 이상에 기초하여 팔레트 코딩 모드 정보를 추론할 수도 있다.

[0124] 팔레트-기반 디코딩 유닛 (165)은 또한, 팔레트에서의 엔트리들 중 적어도 일부를 정의하는 정보를 비디오 데이터와 함께 수신하고, 수신된 정보에 적어도 부분적으로 기초하여 팔레트를 생성하도록 구성될 수도 있다. 팔레트의 사이즈는 고정되거나 가변적일 수도 있다. 일부 케이스들에서, 팔레트의 사이즈는 가변적이고, 비디오 데이터와 함께 시그널링된 정보에 기초하여 조정 가능하다. 시그널링된 정보는 팔레트에서의 엔트리가 그 팔레트에서 최종 엔트리인지 여부를 명시할 수도 있다. 또한, 일부 케이스들에서, 팔레트는 최대 사이즈를 가질 수도 있다. 팔레트의 사이즈는 또한, 조건적으로 송신되거나 추론될 수도 있다. 조건들은 CU의 사이즈, 프레임 타입, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 일 수도 있다.

[0125] 팔레트는 블록의 루마 성분 및 크로마 성분에 대한 픽셀 값들을 표시하는 엔트리들을 포함하는 단일 팔레트일 수도 있다. 이러한 경우에, 팔레트에서 각각의 엔트리는 루마 성분 및 2 개의 크로마 성분들에 대한 픽셀 값을 표시하는 트리플 엔트리이다. 대안적으로, 팔레트는 블록의 루마 성분의 픽셀 값을 표시하는 엔트리들을 포함하는 루마 팔레트, 및 블록의 개별 크로마 성분들에 대한 픽셀 값을 표시하는 엔트리들을 포함하는 크로마 팔레트들을 포함한다.

[0126] 일부 예들에서, 팔레트-기반 디코딩 유닛 (165)은 이전에 프로세싱된 데이터에 기초하여 팔레트에서 엔트리들을 예측함으로써 팔레트를 생성할 수도 있다. 이전에 프로세싱된 데이터는 이전에 디코딩된 이웃하는 블록들에 대하여, 팔레트들, 또는 팔레트들로부터의 정보를 포함할 수도 있다. 팔레트-기반 디코딩 유닛 (165)은 팔레트에서의 엔트리들이 예측될 것인지 여부를 표시하는 예측 선택스 엘리먼트를 수신할 수도 있다. 예측 선택스 엘리먼트는 각각, 루마 및 크로마 성분들에 대한 팔레트들에서의 엔트리들이 예측될 것인지 여부를

표시하는 복수의 예측 선택스 엘리먼트들을 포함할 수도 있다.

[0127] 예측 팔레트에 대하여, 예컨대 예측 팔레트는 공간적으로 이웃하는 블록들을 포함하는 하나 이상의 이웃하는 블록들로부터의 팔레트 엔트리들 및/또는 이웃하는 블록들을 블록들의 특정 스캔 순서로 포함할 수도 있다. 일 예에서, 이웃하는 블록들은 현재 코딩되고 있는 블록의 좌측에 (좌측 이웃 블록) 또는 상부에 (상부 이웃 블록) 공간적으로 로케이팅될 수도 있다. 다른 예에 있어서, 팔레트-기반 디코딩 유닛 (165)은 현재 블록의 원인이 되는 이웃의 가장 빈번한 샘플 값들을 사용하여 예측 팔레트 엔트리들을 결정할 수도 있다. 또 다른 예에 있어서, 이웃하는 블록들은 블록들을 코딩하는데 사용되는 특정 스캔 순서에 따라 현재 코딩되고 있는 블록을 이웃할 수도 있다. 즉, 이웃하는 블록들은 스캔 순서에 있어서 현재 블록 이전에 코딩된 하나 이상의 블록들일 수도 있다. 팔레트-기반 디코딩 유닛 (165)은 팔레트(들)이 카피되는 이웃하는 블록들의 위치를 표시하기 위해 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다.

[0128] 따라서, 일 예에 있어서, 팔레트-기반 디코딩 유닛 (165)은 일부 예들에서, 슬라이스 도는 꾹처에서 좌측 이웃 블록 또는 상부 이웃 블록에 대한 팔레트에서의 엔트리들에 기초하여 팔레트에서의 엔트리들 중 적어도 일부를 예측할 수도 있다. 이 경우, 좌측 이웃 블록 또는 상부 이웃 블록에 대한 팔레트에서의 엔트리들에 기초하여 예측된 팔레트에서의 엔트리들은 예측을 위해 좌측 이웃 블록 또는 상부 이웃 블록의 선택을 표시하는 선택스 엘리먼트에 기초하여 팔레트-기반 디코딩 유닛 (165)에 의해 예측될 수도 있다. 선택스 엘리먼트는 예측을 위해 좌측 이웃 블록 또는 상부 이웃 블록의 선택을 표시하는 값을 갖는 플래그일 수도 있다.

[0129] 일부 예들에서, 팔레트-기반 디코딩 유닛 (165)은 팔레트에서의 적어도 일부 선택된 엔트리들이 엔트리 기반으로 예측될 것인지 여부를 표시하는 하나 이상의 예측 선택스 엘리먼트를 수신하고, 그에 따라 엔트리들을 생성할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165)은 예측 팔레트의 각 엔트리에 대하여, 그 팔레트 엔트리가 현재 블록에 대한 팔레트에 포함되는지 여부를 나타내기 위해, 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다. 엔트리가 예측되지 않을 경우, 팔레트-기반 디코딩 유닛 (165)은 예측되지 않은 엔트리들뿐만 아니라 그러한 엔트리들의 수를 명시하기 위해 하나 이상의 추가의 선택스 엘리먼트들을 디코딩할 수도 있다. 따라서, 팔레트-기반 디코딩 유닛 (165)은 엔트리들 중 일부를 예측하고, 추가의 엔트리들의 수를 포함하여 팔레트에서의 다른 엔트리들을 직접 명시하는 정보를 수신할 수도 있다.

[0130] 일부 예들에서, 전체 팔레트를 예측하기 위한 기술들은 팔레트의 하나 이상의 엔트리들을 예측하기 위한 기술들과 결합될 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165)은 현재 팔레트가 예측 팔레트로부터 전체가 카피되는지 여부를 나타내기 위해 하나 이상의 선택스 엘리먼트들을 비트스트림으로 디코딩할 수도 있다. 그렇지 않다면, 팔레트-기반 디코딩 유닛 (165)은 예측 팔레트에서 각각의 엔트리가 카피되는지 여부를 나타내기 위해 하나 이상의 선택스 엘리먼트들을 비트스트림으로 디코딩할 수도 있다.

[0131] 다른 예에서, 다수의 엔트리들 및 팔레트 값들을 수신하는 대신, 팔레트-기반 디코딩 유닛 (165)은, 각각의 팔레트 값 이후에, 시그널링된 팔레트 값이 그 팔레트에 대한 최종 팔레트 엔트리인지 여부를 나타내기 위한 플래그를 수신할 수도 있다. 팔레트-기반 디코딩 유닛 (165)은 팔레트가 특정 최대 사이즈에 이미 도달한 경우, 그러한 "팔레트의 종료" 플래그를 수신하지 않을 수도 있다.

[0132] 팔레트-기반 디코딩 유닛 (165)에 의해 수신되고 비디오 데이터의 블록의 적어도 일부 포지션들을 팔레트에서의 엔트리들과 연관시키는 정보는 블록에서 포지션들의 적어도 일부에 대한 팔레트 인덱스 값들을 포함하는 맵 정보를 포함할 수도 있고, 팔레트 인덱스 값들의 각각은 그 팔레트에서의 엔트리들 중 하나에 대응한다. 맵 정보는 동일한 팔레트 인덱스 값을 갖는 블록에서 연속하는 포지션들의 수를 각각 표시하는 하나 이상의 런 선택스 엘리먼트들을 포함할 수도 있다.

[0133] 일부 예들에서, 팔레트-기반 디코딩 유닛 (165)은 라인 카피를 표시하는 정보를 수신할 수도 있고, 이에 따라 블록에서 포지션들의 라인에 대한 꾹셀 또는 인덱스 값들은 그 블록에서 포지션들의 다른 라인에 대한 꾹셀 또는 인덱스 값들로부터 카피된다. 팔레트-기반 디코딩 유닛 (165)은 이러한 정보를 사용하여 라인 카피를 수행하고, 블록의 다양한 포지션들에 대한 꾹셀 값들 또는 팔레트에서의 엔트리들을 결정할 수도 있다. 포지션들의 라인은 블록의 포지션들의 로우, 로우의 일부, 컬럼, 또는 컬럼의 일부를 포함할 수도 있다.

[0134] 팔레트-기반 디코딩 유닛 (165)은 블록의 하나 이상의 포지션들에 대한 꾹셀 값들을 수신하고, 그 꾹셀 값들을 팔레트에서의 엔트리들에 부가하여 온 더 플레이로 팔레트의 적어도 일부분을 동적으로 생성함으로써 적어도 부분적으로 팔레트를 생성할 수도 있다. 꾹셀 값들을 부가하는 것은, 엔트리들의 초기 세트를 포함하는 초기 팔레트에 또는 엔트리들의 초기 세트를 포함하지 않는 빈 팔레트에 꾹셀 값들을 부가하는 것을 포함할 수도 있

다. 일부 예들에서, 부가하는 것은, 엔트리들의 초기 세트를 포함하는 초기 팔레트에 새로운 엔트리들을 부가하거나 초기 팔레트에서 기존의 엔트리들을 채우기 위해 픽셀 값들을 부가하는 것, 또는 초기 팔레트에서 엔트리들의 픽셀 값들을 대체하거나 변화시키는 것을 포함한다.

[0135] 일부 예들에서, 팔레트-기반 디코딩 유닛 (165)은 팔레트에 대하여 고정된 최대 사이즈를 결정할 수도 있다.

최대 사이즈에 도달하면, 팔레트-기반 디코딩 유닛 (165)은 팔레트의 하나 이상의 엔트리들을 제거할 수도 있다. 일 예에서, 팔레트-기반 디코딩 유닛 (165)은 예컨대, FIFO 큐를 사용하여 팔레트의 가장 오래된 엔트리를 제거할 수도 있다. 다른 예에서, 팔레트-기반 디코딩 유닛 (165)은 최소로 사용된 엔트리를 제거할 수도 있다. 또 다른 예에서, 팔레트-기반 디코딩 유닛 (165)은 제거될 후보 엔트리가 팔레트에 부가된 때와 그 엔트리의 상대적인 사용에 기초하여, 어떤 엔트리를 제거할 것인지에 관한 가중된 결정을 실행할 수도 있다.

[0136] 일부 예들에서, 그 팔레트는 블록에서의 포지션들 중 하나에 대하여 팔레트로부터 선택된 픽셀 값이 그 블록에서의 포지션들의 실제 픽셀 값과 상이한 양자화된 팔레트일 수도 있고, 따라서 디코딩 프로세스는 순실된다.

예를 들어, 동일한 픽셀 값이 상이한 실제 픽셀 값들을 갖는 2 개의 상이한 포지션들에 대하여 팔레트로부터 선택될 수도 있다.

[0137] 도 4 는 본 개시의 기술들과 부합하는, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 일 예를 도시하는 개념 다이어그램이다. 도 4 의 예는 제 1 팔레트들 (184)과 연관된 제 1 코딩 유닛 (CU) (180) 및 제 2 팔레트들 (192)과 연관된 제 2 CU (188)을 갖는 픽처 (178)를 포함한다. 이하 더 상세히 설명되는 것과 같이, 본 개시의 기술들에 따라, 제 2 팔레트들 (192)은 제 1 팔레트들 (184)에 기초한다. 픽처 (178)는 또한, 인트라-예측 코딩 모드로 코딩되는 블록 (196) 및 인터-예측 코딩 모드로 코딩되는 블록 (200)을 포함한다.

[0138] 도 4 의 기술들은 비디오 인코더 (20) (도 1 및 도 2) 및 비디오 디코더 (30) (도 1 및 도 3)의 맥락에서 및 설명의 목적들을 위해 HEVC 비디오 코딩 표준과 관련하여 설명된다. 그러나, 본 개시의 기술들이 이러한 방식으로 한정되는 것은 아니며, 다른 비디오 코딩 프로세스들 및/또는 표준들에서 다른 비디오 코딩 프로세서들 및/또는 디바이스들에 의해 적용될 수도 있음이 이해되어야 한다.

[0139] 일반적으로, 팔레트는 도 4 의 예에서 현재 코딩되고 있는 CU, 즉 CU (188)에 대하여 우세적이고 및/또는 대표적인 다수의 픽셀 값들을 지칭한다. 제 1 팔레트들 (184) 및 제 2 팔레트들 (192)은 다수의 팔레트들을 포함하는 것으로 도시된다. 일부 예들에서, 본 개시의 양태들에 따라, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))는 CU의 각각의 컬러 성분에 대하여 개별적으로 팔레트들을 코딩할 수도 있다.

예를 들면, 비디오 인코더 (20)는 CU의 루마 (Y) 성분에 대한 팔레트, CU의 크로마 (U) 성분에 대한 다른 팔레트, 및 CU의 크로마 (V) 성분에 대한 또 다른 팔레트를 인코딩할 수도 있다. 이러한 예에서, Y 팔레트의 엔트리들은 CU의 픽셀들의 Y 값들을 표현할 수도 있고, U 팔레트의 엔트리들은 CU의 픽셀들의 U 값들을 표현할 수도 있고, V 팔레트의 엔트리들은 CU의 픽셀들의 V 값들을 표현할 수도 있다.

[0140] 다른 예들에서, 비디오 인코더 (20)는 CU의 모든 컬러 성분들에 대하여 단일 팔레트를 인코딩할 수도 있다.

이러한 예에서, 비디오 인코더 (20)는 Y_i , U_i , 및 V_i 를 포함하는 트리플 값인 i 번째 엔트리를 갖는 팔레트를 인코딩할 수도 있다. 이 경우에, 팔레트는 픽셀들의 성분들 각각에 대한 값을 포함한다. 따라서, 다수의 개별 팔레트들을 갖는 팔레트들의 세트로서의 팔레트들 (184 및 192)의 표현은 단지 하나의 예일 뿐이고 한정하는 것으로 의도되는 것은 아니다.

[0141] 도 4 의 예에서, 제 1 팔레트들 (184)은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3을 각각 갖는 3 개의 엔트리들 (202 - 206)을 포함한다. 엔트리들 (202 - 206)은 인덱스 값들을 각각 픽셀 값 A, 픽셀 값 B, 픽셀 값 C을 포함하는 픽셀 값들에 관련시킨다. 본원에 설명된 것과 같이, 제 1 CU (180)의 실제 픽셀 값들을 코딩하는 것보다, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))는 인덱스들 1 내지 3을 사용하여 블록의 픽셀들을 코딩하기 위해 팔레트-기반 코딩을 사용할 수도 있다. 즉, 제 1 CU (180)의 각 픽셀 포지션에 대하여, 비디오 인코더 (20)는 픽셀에 대한 인덱스 값을 인코딩할 수도 있고, 여기서 인덱스 값은 제 1 팔레트들 (184) 중 하나 이상에서의 픽셀 값과 연관된다. 비디오 디코더 (30)는 비트스트림으로부터 인덱스 값을 획득하고, 인덱스 값을 및 제 1 팔레트들 (184) 중 하나 이상을 사용하여 픽셀 값을 복원할 수도 있다. 따라서, 제 1 팔레트들 (184)은 팔레트-기반 디코딩시 비디오 디코더 (30)에 의한 사용을 위해, 비디오 인코더 (20)에 의해 인코딩된 비디오 데이터 비트스트림에서 송신된다. 일반적으로, 하나 이상의 팔레트들은 각각의 CU에 대하여 송신될 수도 있거나, 또는 상이한 CU

들 중에서 공유될 수도 있다.

[0142] 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제 1 팔레트들 (184) 에 기초하여 제 2 팔레트들 (192) 을 결정할 수도 있다. 예를 들면, 비디오 인코더 (20) 는, CU 에 대한 팔레트가 하나 이상의 다른 CU들, 예컨대 (공간적으로 또는 스캔 순서에 기초하여) 이웃하는 CU들 또는 원인이 되는 이웃의 가장 빈번한 샘플들로부터 예측되는지 여부를 표시하기 위해, (일 예로서, 제 2 CU (188) 를 포함하는) 각각의 CU 에 대한 pred_palette_flag 를 인코딩할 수도 있다. 예를 들어, 그러한 플래그의 값이 1 과 동일할 경우, 비디오 디코더 (30) 는 제 2 CU (188) 에 대한 제 2 팔레트들 (192) 이 하나 이상의 이미 디코딩된 팔레트들로부터 예측되고 그리고 제 2 CU (188) 에 대한 어떤 새로운 팔레트들도 pred_palette_flag 를 포함하는 비트스트림에 포함되지 않는 것을 결정할 수도 있다. 그러한 플래그가 0 과 동일할 경우, 비디오 디코더 (30) 는 제 2 CU (188) 에 대한 팔레트 (192) 가 새로운 팔레트로서 비트스트림에 포함되는 것을 결정할 수도 있다. 일부 예들에서, pred_palette_flag 는 CU 의 각각의 상이한 컬러 성분에 대하여 개별적으로 코딩될 수도 있다 (예컨대, YUV 비디오에서의 CU 에 대하여 3 개의 플래그들, 즉 Y 에 대하여 하나의 플래그, U 에 대하여 하나의 플래그, 및 V 에 대하여 하나의 플래그). 다른 예들에서, 단일 pred_palette_flag 는 CU 의 모든 컬러 성분들에 대하여 코딩될 수도 있다.

[0143] 앞의 예에서, pred_palette_flag 는 현재 블록에 대한 팔레트의 엔트리들 중 임의의 것이 예측되는지 여부를 표시하기 위해 CU 마다 시그널링된다. 일부 예들에서, 하나 이상의 선택스 엘리먼트들이 엔트리 기반으로 시그널링될 수도 있다. 즉, 플래그는 엔트리가 현재 팔레트에 존재하는지 여부를 표시하기 위해 팔레트 예측자의 각각의 엔트리에 대하여 시그널링될 수도 있다. 전술된 것과 같이, 팔레트 엔트리가 예측되지 않다면, 팔레트 엔트리는 명시적으로 시그널링될 수도 있다.

[0144] 제 1 팔레트들 (184) 과 관련이 있는 제 2 팔레트들 (192) 을 결정할 때 (예컨대, pred_palette_flag 는 1 과 동일함), 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 예측 팔레트들, 이 예에서는 제 1 팔레트들 (184) 이 결정되는 하나 이상의 블록들을 로케이팅할 수도 있다. 예측 팔레트들은 현재 코딩되고 있는 CU 의 하나 이상의 이웃하는 CU들 (예컨대, (공간적으로 또는 스캔 순서에 기초하여) 이웃하는 CU들 또는 원인이 되는 이웃의 가장 빈번한 샘플들), 즉 제 2 CU (188) 과 연관된다. 하나 이상의 이웃하는 CU들의 팔레트들은 예측자 팔레트와 연관될 수도 있다. 도 4 에 도시된 예와 같은 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 2 CU (188) 에 대한 예측 팔레트를 결정할 때, 좌측의 이웃하는 CU, 즉 제 1 CU (180) 를 로케이팅할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 2 CU (188) 와 관련하여 다른 포지션들에서의 하나 이상의 CU들, 예컨대 상부 CU, 즉 CU (196) 를 로케이팅할 수도 있다.

[0145] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 계층 (hierarchy) 에 기초하여 팔레트 예측을 위한 CU 를 결정할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 예측을 위해 좌측의 이웃하는 CU, 즉 제 1 CU (180) 를 초기에 식별할 수도 있다. 좌측의 이웃하는 CU 가 예측을 위해 사용가능하지 않다면, (예컨대, 좌측의 이웃하는 CU 가 인트라-예측 모드 또는 인트라-예측 모드와 같은 팔레트-기반 코딩 모드 이외의 모드로 코딩되거나, 또는 꾹쳐 또는 슬라이스의 최좌측 에지에 로케이팅된다면), 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 상부의 이웃하는 CU, 즉 CU (196) 를 식별할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 예측에 사용가능한 팔레트를 갖는 CU 를 로케이팅할 때 까지, 미리 결정된 순서의 위치들에 따라 사용가능한 CU 에 대한 탐색을 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.

[0146] 도 4 의 예가 단일 CU, 즉 제 1 CU (180) 로부터의 예측 팔레트들로서 제 1 팔레트들 (184) 을 도시하지만, 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃하는 CU들의 조합으로부터 예측을 위한 팔레트들을 로케이팅할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더는 복수의 이웃하는 CU들 중 하나 또는 조합의 팔레트들에 기초하여 팔레트를 생성하기 위해 하나 이상의 공식들, 함수들, 규칙들 등을 적용할 수도 있다.

[0147] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 예측을 위한 다수의 잠재적인 후보들을 포함하는 후보 리스트를 구성할 수도 있다. 그러한 예들에서, 비디오 인코더 (20) 는 팔레트 예측을 위해 사용되는 현재 CU 가 선택되는 (예컨대, 팔레트를 카피하는) 리스트에서의 후보 CU 를 표시하도록 후보 리스토의 인덱스를 인코딩할 수도 있다. 비디오 디코더 (30) 는 동일한 방식을 후보 리스트를 구성하고, 인덱스를 디코딩하고, 디코딩된 인덱스를 사용하여 현재 CU 와 함께 사용하기 위한 대응하는 CU 의 팔레트를 선택

할 수도 있다.

[0148] 예시의 목적들을 위한 일 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 코딩되고 있는 CU 의 상부에 포지셔닝된 하나의 CU 및 현재 코딩되고 있는 CU 의 좌측에 포지셔닝된 하나의 CU 를 포함하는 후보 리스트를 구성할 수도 있다. 이러한 예에서, 비디오 인코더 (20) 는 후보 선택을 표시하기 위해 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 현재 CU 에 대한 팔레트가 현재 CU 의 좌측에 포지셔닝된 CU 로부터 카피되는 것을 표시하기 위해 0 의 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 현재 CU 에 대한 팔레트가 현재 CU 상부에 포지셔닝된 CU 로부터 카피되는 것을 표시하기 위해 1 의 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 디코더 (30) 는 플래그를 디코딩하고, 팔레트 예측을 위해 적절한 CU 를 선택한다.

[0149] 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 다른 팔레트들에 포함된 샘플 값들이 하나 이상의 이웃하는 CU들에서 발생하는 빈도에 기초하여 현재 코딩되고 있는 CU 에 대한 팔레트를 결정한다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 미리 결정된 수의 CU들의 코딩 동안 가장 빈번하게 사용된 인덱스 값들과 연관된 컬러들을 트래킹할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재 코딩되고 있는 CU 에 대한 팔레트에 가장 빈번하게 사용되는 컬러들을 포함시킬 수도 있다.

[0150] 전술된 것과 같이, 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더는 현재 CU 를 코딩하기 위해 이웃하는 CU 로부터 전체 팔레트를 카피할 수도 있다. 부가적으로 또는 대안적으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 엔트리 방식 기반의 팔레트 예측을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20) 는, 개별 엔트리들이 예측 팔레트(예컨대, 다른 CU 의 팔레트) 에 기초하여 예측되는지 여부를 표시하는 팔레트의 전체 엔트리에 대한 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 이러한 예에서, 비디오 인코더 (20) 는, 소정 엔트리가 예측 팔레트로부터의 예측된 값(예컨대, 이웃하는 CU 와 연관된 팔레트의 대응하는 엔트리) 일 경우, 소정의 엔트리에 대하여 1 과 동일한 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 특정 엔트리가 다른 CU 의 팔레트로부터 예측되지 않는 것을 나타내기 위해 특정 엔트리에 대하여 0 과 동일한 값을 갖는 플래그를 인코딩할 수도 있다. 이러한 예에서, 비디오 인코더 (20) 는 또한, 예측되지 않은 팔레트 엔트리의 값을 표시하는 다른 데이터를 인코딩할 수도 있다.

[0151] 도 4 의 예에서, 제 2 팔레트들 (192) 은 각각, 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 엔트리 인덱스 값 3, 및 엔트리 인덱스 값 4 을 갖는 4 개의 엔트리들 (208 - 214) 을 포함한다. 엔트리들 (208 - 214) 은 인덱스 값들을 각각 픽셀 값 A, 픽셀 값 B, 픽셀 값 C 및 픽셀 값 D 을 포함하는 픽셀 값들에 관련시킨다. 본 개시의 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 전술된 기술들 중 임의의 것을 사용하여, 팔레트 예측의 목적들을 위해 제 1 CU (180) 를 로케이팅하고 제 2 CU (188) 를 코딩하기 위해 제 1 팔레트들 (184) 의의 엔트리들 1 내지 3 을 제 2 팔레트들 (192) 의 엔트리들 1 내지 3 로 카피할 수도 있다. 이러한 방식으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제 1 팔레트들 (184) 에 기초하여 제 2 팔레트들 (192) 을 결정할 수도 있다. 추가로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제 2 팔레트들 (192) 과 함께 포함될 엔트리 4 에 대한 데이터를 코딩할 수도 있다. 그러한 정보는 예측자 팔레트로부터 예측되지 않은 팔레트 엔트리들의 수 및 그 팔레트 엔트리들에 대응하는 픽셀 값들을 포함할 수도 있다.

[0152] 일부 예들에서, 본 개시의 양태들에 따르면, 하나 이상의 신택스 엘리먼트들은 제 2 팔레트들 (192) 과 같은 팔레트들이 (도 4 에 제 1 팔레트들 (184) 로 도시되지만 하나 이상의 블록들로부터의 엔트리들로 구성될 수도 있는) 예측 팔레트로부터 전체적으로 예측되는지의 여부 또는 제 2 팔레트들 (192) 의 특정 엔트리들이 예측되는지의 여부를 표시할 수도 있다. 예를 들어, 초기 신택스 엘리먼트는 엔트리들 전부가 예측되는지 여부를 표시할 수도 있다. 초기 신택스 엘리먼트가 엔트리들 전부가 예측되지 않는 것을 표시한다면 (예컨대, 0 의 값을 갖는 플래그), 하나 이상의 추가의 신택스 엘리먼트들이 제 2 팔레트들 (192) 의 어떤 엔트리들이 예측 팔레트로부터 예측되는지를 표시할 수도 있다.

[0153] 본 개시의 일부 양태들에 따르면, 팔레트 예측과 연관된 특정 정보는 코딩되고 있는 데이터의 하나 이상의 특징들로부터 추론될 수도 있다. 즉, 비디오 인코더 (20) 가 신택스 엘리먼트들을 인코딩하는 것 (및 비디오 디코더 (30) 가 그러한 신택스 엘리먼트들을 디코딩하는 것) 보다, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 코딩되고 있는 데이터의 하나 이상의 특징들에 기초하여 팔레트 예측을 수행할 수도 있다.

[0154] 일 예에서, 예시의 목적들을 위해, 전술된 pred_palette_flag 의 값은 예들로서, 코딩되고 있는 CU 의 사이즈, 프레임 타입, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또

는 멀티-뷰 코딩에서의 뷰 id 중 하나 이상으로 추론될 수도 있다. 즉, 일 예로서 CU 의 사이즈에 대하여, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 전술된 pred_palette_flag 가 미리 결정된 사이즈를 초과하는 임의의 CU들에 대하여 1 과 동일한 것으로 결정할 수도 있다. 이러한 예에서, pred_palette_flag 는 인코딩된 비트스트림에서 시그널링될 필요는 없다.

[0155] pred_palette_flag 와 관련하여 전술되었지만, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 또한 또는 대안적으로, 팔레트가 예측을 위해 사용되는 후보 CU 와 같이 팔레트 예측과 연관된 다른 정보, 또는 코딩되고 있는 데이터의 하나 이상의 특징들에 기초하여 팔레트 예측 후보들을 구성하기 위한 규칙들을 추론할 수도 있다.

[0156] 본 개시물의 다른 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트를 온 더 플라이로 구성할 수도 있다. 예를 들어, 초기에 제 2 CU (188) 를 코딩할 경우, 팔레트들 (192) 에 어떤 입력들도 존재하지 않는다. 비디오 인코더 (20) 및 비디오 디코더 (30) 가 제 2 CU (188) 의 픽셀들에 대한 새로운 값들을 코딩할 때, 각각의 새로운 값은 팔레트들 (192) 에 포함된다. 즉, 예를 들어, 비디오 인코더 (20) 는 픽셀 값들이 생성되고 CU (188) 에서의 포지션들을 위해 시그널링될 때, 픽셀 값들을 팔레트들 (192) 에 부가한다. 비디오 인코더 (20) 가 CU 에서 상대적으로 나중에 인코딩할 때, 비디오 인코더 (20) 는 픽셀 값들을 시그널링하는 것보다 인덱스 값들을 사용하여 팔레트에 이미 포함된 것과 동일한 값을 갖는 픽셀들을 인코딩할 수도 있다. 유사하게, 비디오 디코더 (30) 가 제 2 CU (188) 에서의 포지션에 대하여 (예컨대, 비디오 인코더 (20) 에 의해 시그널링된) 새로운 픽셀 값을 수신할 경우, 비디오 디코더 (30) 는 그 픽셀 값을 팔레트들 (192) 에 포함시킨다. 제 2 CU (188) 에서 상대적으로 나중에 디코딩된 픽셀 포지션들이 제 2 팔레트들 (192) 에 부가된 픽셀 값들을 가질 경우, 비디오 디코더 (30) 는 제 2 CU (188) 의 픽셀 값들의 복원을 위해 제 2 팔레트들 (192) 에서 대응하는 픽셀 값들을 식별하는, 예컨대 인덱스 값들과 같은 정보를 수신할 수도 있다.

[0157] 일부 예들에서, 이하 더 상세히 설명되는 것과 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트들 (184 및 192) 을 최대 팔레트 사이즈로 또는 그 미만으로 유지할 수도 있다. 본 개시물의 양태들에 따르면, 최대 팔레트 사이즈가 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 위해 동일한 프로세스를 수행한다. 팔레트 엔트리들을 제거하기 위한 일 예시적인 프로세스는, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 팔레트의 가장 오래된 엔트리를 제거하는 선입 선출 (FIFO) 기술이다. 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 그 팔레트로부터 최소로 빈번히 사용되는 팔레트 입력을 제거할 수도 있다. 또 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 어떤 엔트리를 제거할 것인지 결정하기 위해 FIFO 및 사용 빈도 프로세스들 양자를 가중할 수도 있다. 즉, 엔트리의 제거는 엔트리가 얼마나 오래된 것인지 및 얼마나 자주 사용되는지에 기초할 수도 있다.

[0158] 일부 양태들에 따르면, 엔트리 (픽셀 값) 가 팔레트로부터 제거되고 그 픽셀 값이 코딩되고 있는 CU 의 이후 포지션에서 다시 발생한다면, 비디오 인코더 (20) 는 팔레트에 엔트리를 포함시키고 인덱스를 인코딩하는 대신, 픽셀 값을 인코딩할 수도 있다. 부가적으로 또는 대안적으로, 비디오 인코더 (20) 는 예컨대, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 CU 에서의 포지션들을 스캔할 때, 제거된 이후의 팔레트 내에 팔레트 엔트리들을 재입력할 수도 있다.

[0159] 일부 예들에서, 팔레트를 온 더 플라이로 유도하는 기술은 팔레트를 결정하기 위한 하나 이상의 다른 기술들과 결합될 수도 있다. 특히, 일 예로서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 (예컨대, 제 1 팔레트들 (184) 로부터 제 2 팔레트들 (192) 을 예측하기 위해 팔레트 예측을 사용하여) 제 2 팔레트들 (192) 을 초기에 코딩할 수도 있고, 제 2 CU (188) 의 픽셀들을 코딩할 때 제 2 팔레트들 (192) 을 업데이트할 수도 있다. 예를 들어, 초기 팔레트를 송신할 시, 비디오 인코더 (20) 는 CU 에서의 추가 위치들의 픽셀 값들이 스캔될 때, 초기 팔레트에 값을 부가하거나 초기 팔레트에서 값을 변경시킬 수도 있다. 유사하게, 초기 팔레트를 수신할 시, 비디오 디코더 (30) 는 CU 에서의 추가 위치들의 픽셀 값들이 스캔될 때, 초기 팔레트에 값을 부가하거나 초기 팔레트에서 값을 변경시킬 수도 있다.

[0160] 비디오 인코더 (20) 는, 일부 예들에서, 현재 CU 가 전체 팔레트의 송신, 또는 온 더 플라이 팔레트 생성, 또는 초기 팔레트의 송신과 초기 팔레트의 온 더 플라이 유도에 의한 업데이트와의 조합을 사용하는지 여부를 시그널링할 수도 있다. 일부 예들에서, 초기 팔레트는 최대 팔레트 사이즈의 전체 팔레트일 수도 있고, 이 경우 초기 팔레트에서의 값을 변경될 수도 있다. 다른 예들에서, 초기 팔레트는 최대 팔레트 사이즈보다 더 작

을 수도 있고, 이 경우 비디오 인코더 (20) 및 비디오 디코더 (30) 는 초기 팔레트에 값들을 부가하고 및/또는 초기 팔레트의 값들을 변경할 수도 있다.

[0161] 본 개시의 양태들에 따르면, 예컨대, 팔레트에 포함된 픽셀 값들의 수에 대하여, 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 과 같은 팔레트의 사이즈는 고정될 수도 있거나, 또는 인코딩된 비트스트림에서 하나 이상의 선택스 엘리먼트들을 사용하여 시그널링될 수도 있다. 예를 들어, 일부 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 사이즈를 코딩하기 위해 1 진 코드들 또는 절단된 1 진 코드들 (예컨대, 팔레트 사이즈의 최대 제한치에서 절단하는 코드들) 을 사용할 수도 있다. 다른 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 사이즈를 코딩하기 위해 지수 골롬 또는 라이스 골롬 (Rice-Golomb) 코드들을 사용할 수도 있다.

[0162] 다른 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트의 각각의 엔트리 이후에 팔레트의 사이즈를 표시하는 데이터를 코딩할 수도 있다. 일 예로서 제 2 팔레트들 (192) 에 대하여, 비디오 인코더 (20) 는 엔트리들 (208 - 214) 의 각각 이후에 정지 플래그를 인코딩할 수도 있다. 이러한 예에서, 1 과 동일한 정지 플래그는, 현재 코딩되고 있는 엔트리가 제 2 팔레트들 (192) 의 최종 엔트리인 것을 명시할 수도 있는 반면, 0 과 동일한 정지 플래그는 제 2 팔레트들 (192) 에 추가의 엔트리들이 존재하는 것을 나타낼 수도 있다. 따라서, 비디오 인코더 (20) 는 엔트리들 (208 - 212) 의 각각 이후에 0 의 값을 갖는 정지 플래그들 및 엔트리 (214) 이후에 1 의 값을 갖는 정지 플래그를 인코딩할 수도 있다. 일부 경우들에서, 정지 플래그는 구성된 팔레트가 최대 팔레트 사이즈 제한치에 도달하면, 비트스트림에 포함되지 않을 수도 있다. 앞의 예들이 팔레트들의 사이즈를 명시적으로 시그널링하기 위한 기술을 개시하지만, 다른 예들에서 팔레트들의 사이즈는 조건적으로 송신되거나 소위 부가 정보 (예컨대, 앞서 언급된 것과 같이, 코딩되고 있는 CU 의 사이즈, 프레임 타입, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 와 같은 특징 정보) 에 기초하여 추론될 수도 있다.

[0163] 본 개시의 기술들은 손실없이, 또는 대안적으로, 약간의 손실들로 (손실 코딩) 데이터를 코딩하는 것을 포함한다. 예를 들어, 손실 코딩에 대하여, 비디오 인코더 (20) 는 팔레트들의 픽셀 값들을 CU 에서의 실제 픽셀 값들에 정확히 매칭하지 않고 CU 의 픽셀들을 코딩할 수도 있다. 본 개시의 기술들이 손실 코딩에 적용될 경우, 일부 제한들이 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 서로 미리 결정된 범위 내에 있을 경우, 팔레트의 엔트리들을 병합 (양자화) 할 수도 있다. 즉, 새로운 팔레트 값의 여러 마진 내에 있는 팔레트 값이 이미 존재한다면, 새로운 팔레트 값은 팔레트에 부가되지 않는다. 다른 예에서, 블록에서의 복수의 상이한 픽셀 값들이 단일 팔레트 엔트리로, 또는 등가의 단일 팔레트 픽셀 값으로 맵핑될 수도 있다.

[0164] 비디오 디코더 (30) 는 특정 팔레트가 손실되지 않는지 손실되는지 여부에 관계없이, 동일한 방식으로 픽셀 값을 디코딩할 수도 있다. 일 예로서, 비디오 디코더 (30) 는 팔레트가 손실되지 않는지 손실되는지 여부에 관계없이, 픽셀 포지션에 대한 팔레트에서의 엔트리를 선택하기 위해 코딩된 블록에서의 소정 픽셀 포지션에 대하여 비디오 인코더 (20) 에 의해 송신된 인덱스 값을 사용할 수도 있다. 이러한 예에서, 팔레트 엔트리의 픽셀 값은 오리지널 픽셀 값과 정확히 매칭하는지, 또는 매칭하지 않는지의 여부에 따라 코딩된 블록에서의 픽셀 값으로서 사용된다.

[0165] 손실 코딩의 일 예에서, 예시의 목적들을 위해, 비디오 인코더 (20) 는 델타 값으로 지정되는 여러 범위를 결정할 수도 있다. 후보 픽셀 값 엔트리 Plt_cand 는 CU 또는 PU 와 같은 코딩될 블록의 포지션에서의 픽셀 값에 대응할 수도 있다. 팔레트의 구성 동안, 비디오 인코더 (20) 는 후보 픽셀 값 엔트리 Plt_cand 와 팔레트에서 기존의 픽셀 값 엔트리들 전부 간의 절대 차이를 결정한다. 후보 픽셀 값 엔트리 Plt_cand 와 팔레트에서 기존의 픽셀 값 엔트리들 간의 절대 차이들 모두가 델타 값보다 크다면, 비디오 인코더 (20) 는 픽셀 값 후보를 팔레트에 엔트리로서 부가할 수도 있다. 후보 픽셀 값 엔트리 Plt_cand 와 팔레트에서 적어도 하나의 기존의 픽셀 값 엔트리 간의 절대 차이가 델타 값과 동일하거나 그보다 작다면, 비디오 인코더 (20) 는 후보 픽셀 값 엔트리 Plt_cand 를 팔레트에 부가하지 않을 수도 있다. 따라서, 픽셀 값 엔트리 Plt_cand 를 코딩할 경우, 비디오 인코더 (20) 는 픽셀 값 엔트리 Plt_cand 에 최인접하는 픽셀 값을 갖는 엔트리를 선택할 수도 있고, 따라서 일부 손실을 시스템에 도입한다. 팔레트가 다수의 성분들 (예컨대, 3 개의 컬러 성분들) 로 구성될 경우, 개별 성분 값들의 절대 차이의 합은 델타 값과의 비교를 위해 사용될 수도 있다. 대안적으로 또는 부가적으로, 각각의 성분 값에 대한 절대 차이는 제 2 델타 값과 비교될 수도 있다.

- [0166] 일부 예들에서, 전술된 팔레트에서의 기존의 픽셀 값 엔트리들은 유사한 엘타 비교 프로세스를 사용하여 부가될 수도 있다. 다른 예들에서, 팔레트에서의 기존의 픽셀 값들은 다른 프로세스들을 사용하여 부가될 수도 있다. 예를 들어, 하나 이상의 초기 픽셀 값 엔트리들이 팔레트를 구성하는 엘타 비교 프로세스를 시작하기 위해 (엘타 비교 없이) 팔레트에 부가될 수도 있다. 전술된 프로세스는 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 루마 및/또는 크로마 팔레트들을 생성하도록 구현될 수도 있다.
- [0167] 팔레트 구성과 관련하여 전술된 기술들은 또한, 픽셀 코딩 동안 비디오 인코더 (20) 및 비디오 디코더 (30)에 의해 사용될 수도 있다. 예를 들어, 픽셀 값의 인코딩시, 비디오 인코더 (20)는 픽셀의 값을 팔레트에서 엔트리들의 픽셀 값들과 비교할 수도 있다. 픽셀의 값과 팔레트에서 엔트리들 중 하나 간의 절대 픽셀 값 차이가 엘타 값과 동일하거나 그보다 작다면, 비디오 인코더 (20)는 픽셀 값을 팔레트의 엔트리로서 인코딩할 수도 있다. 즉, 이러한 예에서, 비디오 인코더 (20)는 픽셀 값이 충분히 작은 (예컨대, 미리 결정된 범위 내의) 절대 차이 대 팔레트 엔트리를 생성할 경우, 팔레트의 엔트리들 중 하나를 사용하여 픽셀 값을 인코딩한다.
- [0168] 일부 예들에서, 비디오 인코더 (20)는 픽셀 값을 인코딩하기 위해 (코딩되고 있는 픽셀 값과 비교하여) 최소 절대 픽셀 값 차이를 산출하는 팔레트 엔트리를 선택할 수도 있다. 일 예로서, 비디오 인코더 (20)는 픽셀 값에 대하여 사용될 팔레트 엔트리, 예컨대 비디오 디코더 (30)에서 코딩된 픽셀 값을 복원하는데 사용될 팔레트 픽셀 값 엔트리를 표시하기 위해 인덱스를 인코딩할 수도 있다. 픽셀의 값과 팔레트에서의 모든 엔트리 간의 절대 픽셀 값 차이가 엘타보다 크다면, 인코더는 픽셀 값을 인코딩하기 위해 팔레트 엔트리들 중 하나를 사용하지 않을 수도 있고, 대신 코딩될 픽셀의 픽셀 값을 비디오 디코더 (30)로 송신할 수도 있다 (그리고 가능하면 그 픽셀 값을 팔레트로의 엔트리로서 부가한다).
- [0169] 다른 예에서, 비디오 인코더 (20)는 픽셀 값을 인코딩하기 위해 팔레트의 엔트리를 선택할 수도 있다. 비디오 인코더 (20)는 선택된 엔트리를 예측 픽셀 값으로서 사용할 수도 있다. 즉, 비디오 인코더 (20)는 실제 픽셀 값과 선택된 엔트리 간의 차이를 나타내는 잔차 값을 결정하고, 잔차를 인코딩할 수도 있다. 비디오 인코더 (20)는 팔레트의 엔트리들에 의해 예측되는 블록에서의 픽셀들에 대한 잔차 값을 생성할 수도 있고, 픽셀들의 블록에 대한 개별 잔차 픽셀 값을 포함하는 잔차 블록을 생성할 수도 있다. 비디오 인코더 (20)는 그 후, (도 2와 관련하여 전술된 것과 같은) 변환 및 양자화를 잔차 블록에 적용할 수도 있다. 이러한 방식으로, 비디오 인코더 (20)는 양자화된 잔차 변환 계수들을 생성할 수도 있다.
- [0170] 비디오 디코더 (30)는 잔차 블록을 재생하기 위해 변환 계수들을 역변환 및 역양자화할 수도 있다. 그 후에, 비디오 디코더 (30)는 픽셀 값에 대한 예측 팔레트 엔트리 값 및 잔차 값을 사용하여 픽셀 값을 복원할 수도 있다. 예를 들어, 비디오 디코더 (30)는 잔차 값과 팔레트 엔트리 값을 결합하여 코딩된 픽셀 값을 복원할 수도 있다.
- [0171] 일부 예들에서, 엘타 값은 상이한 CU 사이즈들, 픽처 사이즈들, 컬러 공간들 또는 상이한 컬러 성분들에 대하여 상이할 수도 있다. 엘타 값은 다양한 코딩 조건들에 기초하여 미리 결정되거나 결정될 수도 있다. 예를 들어, 비디오 인코더 (20)는 PPS, SPS, VPS 및/또는 슬라이스 헤더에서의 신팩스와 같은 하이 레벨 신팩스를 사용하여 엘타 값을 비디오 디코더 (30)로 시그널링할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 동일한 고정된 엘타 값을 사용하도록 미리 구성될 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 (예컨대, 전술된 것과 같은 CU 사이즈, 컬러 공간, 컬러 성분 등등과 같은) 부가 정보에 기초하여 엘타 값을 적응적으로 유도할 수도 있다.
- [0172] 일부 예들에서, 손실 코딩 팔레트 모드는 HEVC 코딩 모드로서 포함될 수도 있다. 예를 들어, 코딩 모드들은 인트라-예측 모드, 인터-예측 모드, 무손실 코딩 팔레트 모드, 및 손실 코딩 팔레트 모드를 포함할 수도 있다. HEVC 코딩에서, 도 2 및 도 3과 관련하여 전술된 것과 같이, 양자화 파라미터 (QP)는 허용된 왜곡을 제어하기 위해 사용된다. 팔레트-기반 코딩에 대한 엘타의 값은 계산될 수도 있거나, 그렇지 않으면 QP의 함수로서 결정될 수도 있다. 예를 들어, 전술된 엘타 값은 $1 << (QP/6)$ 또는 $1 << ((QP+d)/6)$ 일 수도 있고, 여기서 d는 상수이고 "<<"는 비트방식 좌측-쉬프트 연산자를 나타낸다.
- [0173] 본 개시에 설명된 손실 코딩 기술들을 사용하는 팔레트의 생성은 비디오 인코더 (20), 비디오 디코더 (30) 또는 이들 양자에 의해 수행될 수도 있다. 예를 들어, 비디오 인코더 (20)는 전술된 엘타 비교 기술들을 사용하여 CU에 대한 팔레트에서의 엔트리들을 생성하고, 비디오 디코더 (30)에 의한 사용을 위해 팔레트의 구성에 대한 정보를 시그널링할 수도 있다. 즉, 비디오 인코더 (20)는 CU에 대한 팔레트에서의 엔트리들에 대한 픽셀 값을 표시하는 정보를 시그널링한 후, 그러한 팔레트 엔트리들과 연관된 픽셀 값을 사용하여 픽셀 값을

들을 인코딩하도록 구성될 수도 있다. 비디오 디코더 (30)는 그러한 정보를 사용하여 팔레트를 구성한 후, 엔트리들을 사용하여 코딩된 블록의 픽셀 값들을 디코딩할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 코딩된 블록의 하나 이상의 픽셀 부분들에 대한 팔레트 엔트리들을 식별하는 인덱스 값들을 시그널링할 수도 있고, 비디오 디코더 (30)는 인덱스 값들을 사용하여 팔레트로부터 적절한 픽셀 값 엔트리들을 취출할 수도 있다.

[0174] 다른 예들에서, 비디오 디코더 (30)는 전술된 멜타 비교 기술들을 적용함으로써 팔레트를 구성하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30)는 코딩된 블록 내의 포지션들에 대한 픽셀 값들을 수신하고, 그 픽셀 값들과 팔레트에서 기존의 픽셀 값 엔트리들 간의 절대 차이들이 멜타 값보다 큰지 여부를 결정할 수도 있다. 만약 그렇다면, 비디오 디코더 (30)는, 예컨대 비디오 인코더 (20)에 의해 시그널링된 대응하는 인덱스 값들을 사용하는 블록의 다른 픽셀 포지션들에 대한 픽셀 값들의 팔레트-기반 디코딩에서 추후 사용을 위해, 픽셀 값들을 팔레트에 엔트리들로서 부가할 수도 있다. 이 경우에, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트를 생성하기 위해 동일하거나 유사한 프로세스들을 적용한다. 만약 그렇지 않다면, 비디오 디코더 (30)는 픽셀 값들을 팔레트에 부가하지 않을 수도 있다.

[0175] 예시의 목적들을 위한 일 예에서, 비디오 디코더 (30)는 블록에서의 다양한 픽셀 포지션들에 대한 인덱스 값들 또는 픽셀 값들을 수신할 수도 있다. 인덱스 값이 예컨대, 픽셀 포지션에 대해 수신된다면, 비디오 디코더 (30)는 그 인덱스 값을 사용하여 팔레트에서 엔트리를 식별하고, 픽셀 포지션에 대해 팔레트 엔트리의 픽셀 값을 사용할 수도 있다. 픽셀 값이 픽셀 포지션에 대해 수신된다면, 비디오 디코더 (30)는 수신된 픽셀 값을 픽셀 포지션에 대해 사용하고, 또한 멜타 비교를 적용하여 픽셀 값이 팔레트에 부가되어 향후 팔레트 코딩을 위해 사용되어야만 하는지 여부를 결정할 수도 있다.

[0176] 인코더 측에서, 블록에서의 포지션에 대한 픽셀 값이 그 픽셀 값과 멜타 값 이하인 팔레트에서의 기존의 픽셀 값 엔트리와의 절대 차이를 생성한다면, 비디오 인코더 (20)는 그 포지션에 대한 픽셀 값을 복원하는데 사용하기 위해 팔레트에서 엔트리를 식별하기 위한 인덱스 값을 전송할 수도 있다. 블록에서의 포지션에 대한 픽셀 값이 그 픽셀 값과 전부가 멜타 값보다 큰 팔레트에서의 기존의 픽셀 값 엔트리들 간의 절대 차이 값들을 생성한다면, 비디오 인코더 (20)는 픽셀 값을 전송하고, 그 픽셀 값을 팔레트에서 새로운 엔트리로서 부가할 수도 있다. 팔레트를 구성하기 위해, 비디오 디코더 (30)는 예컨대, 전술된 것과 같이, 인코더에 의해 시그널링된 멜타 값들을 사용하거나, 고정된 또는 공지된 멜타 값에 의존하거나, 멜타 값을 추론 또는 유도할 수도 있다.

[0177] 전술된 것과 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 비디오 데이터를 코딩할 때, 인트라-예측 모드, 인터-예측 모드, 무손실 코딩 팔레트 모드, 및 손실 코딩 팔레트 모드를 포함하는 코딩 모드들을 사용할 수도 있다. 본 개시의 일부 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트-기반 코딩이 가능한지 여부를 표시하는 하나 이상의 선택스 엘리먼트들을 코딩할 수도 있다. 예를 들어, 각각의 CU에서, 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 CU (또는 다른 예들에서 PU)에 대하여 사용될 것인지 여부를 표시할 수도 있다. 예를 들어, 이러한 플래그는 CU 레벨에서 인코딩된 비디오 비트스트림에서 시그널링되고, 그 후 인코딩된 비디오 비트스트림을 디코딩할 시 비디오 디코더 (30)에 의해 수신될 수도 있다.

[0178] 이러한 예에서, 1 과 동일한 상기 PLT_Mode_flag 의 값은, 현재 CU 가 팔레트-기반 코딩 모드를 사용하여 인코딩되는 것을 명시할 수도 있다. 이러한 경우에, 비디오 디코더 (30)는 CU 를 디코딩하기 위해 팔레트-기반 코딩 모드를 적용할 수도 있다. 일부 예들에서, 선택스 엘리먼트는 CU 에 대하여 복수의 상이한 팔레트-기반 코딩 모드들 중 하나 (예컨대, 손실 또는 무손실) 를 표시할 수도 있다. 0 과 동일한 상기 PLT_Mode_flag 의 값은, 현재 CU 가 팔레트 모드 이외의 모드를 사용하여 인코딩되는 것을 명시할 수도 있다. 예를 들어, 다양한 인터-예측, 인트라-예측 또는 다른 코딩 모드들 중 임의의 것이 사용될 수도 있다. PLT_Mode_flag 의 값이 0 일 경우, 비디오 인코더 (20)는 또한, 개별 CU 를 인코딩하기 위해 사용된 특정 모드 (예컨대, HEVC 코딩 모드) 를 표시하기 위해 추가의 데이터를 인코딩할 수도 있다. PLT_Mode_flag 의 사용은 예시의 목적들을 위해 설명된다. 그러나, 다른 예들에서, 멀티-비트 코드들과 같은 다른 선택스 엘리먼트들은 팔레트-기반의 코딩 모드가 CU (또는 다른 예들에서 PU) 를 위해 사용될 것인지 여부를 표시하거나, 또는 복수의 모드들 중 어느 것이 코딩을 위해 사용될 것인지를 표시하기 위해 사용될 수도 있다.

[0179] 일부 예들에서, 전술된 플래그 또는 다른 선택스 엘리먼트들은 CU (또는 PU) 레벨보다 높은 레벨에서 송신될 수

도 있다. 예를 들어, 비디오 인코더 (20) 는 슬라이스 레벨에서 그러한 플래그를 시그널링할 수도 있다. 이 경우에, 1 과 동일한 값은 슬라이스에서 CU들 모두가 팔레트 모드를 사용하여 인코딩되는 것을 표시한다. 이러한 예에서, 예컨대 팔레트 모드 또는 다른 모드들에 대하여, 어떤 추가의 모드 정보도 CU 레벨에서 시그널링되지 않는다. 다른 예에서, 비디오 인코더 (20) 는 PPS, SPS 또는 VPS 에서 그러한 플래그를 시그널링 할 수도 있다.

[0180] 본 개시의 일부 양태들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 모드가 특정 슬라이스, 핵처, 시퀀스 등등에 대하여 불가능한지 또는 가능한지 여부를 명시하는 (예컨대, 전술된 플래그와 같은) 하나 이상의 선택스 엘리먼트들을 슬라이스, PPS, SPS, 또는 VPS 레벨들 중 하나에서 코딩할 수도 있는 반면, PLT_Mode_flag 는 팔레트-기반 코딩 모드가 각각의 CU 에 대하여 사용되는지 여부를 표시한다. 이 경우에, 슬라이스, PPS, SPS, 또는 VPS 레벨에서 전송된 플래그 또는 다른 선택스 엘리먼트가 팔레트 코딩 모드가 불가능한 것을 표시한다면, 일부 예들에서, 각각의 CU 에 대하여 PLT_Mode_flag 를 시그널링할 필요가 없을 수도 있다. 대안적으로, 슬라이스, PPS, SPS, 또는 VPS 레벨에서 전송된 플래그 또는 다른 선택스 엘리먼트가 팔레트 코딩 모드가 가능한 것을 표시한다면, PLT_Mode_flag 는 팔레트-기반의 코딩 모드가 각각의 CU 에 대하여 사용될 것인지 여부를 표시하도록 추가로 시그널링될 수도 있다. 다시, 전술된 것과 같이, CU 의 팔레트-기반 코딩을 표시하기 위한 이들 기술의 적용은 부가적으로 또는 대안적으로 PU 의 팔레트-기반 코딩을 표시하는데 사용될 수 있다.

[0181] 일부 예들에서, 전술된 선택스 엘리먼트들은 비트스트림에서 조건적으로 시그널링될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 CU 의 사이즈, 프레임 타입, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 에 기초하여 선택스 엘리먼트들을 오직 개별적으로 인코딩 또는 디코딩할 수도 있다.

[0182] 전술된 예들은 예컨대 비트스트림에서의 하나 이상의 선택스 엘리먼트들을 사용한 명시적인 시그널링에 관한 것 이지만, 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 코딩 모드가 활성이고 및/ 또는 특정 블록을 코딩하기 위해 사용되는지 여부를 암시적으로 결정할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트-기반 코딩이 블록에 대해 사용되는지 여부를, 예컨대 CU 의 사이즈, 프레임 타입, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 에 기초하여 결정할 수도 있다.

[0183] 도 4 의 기술들이 앞서 CU들 (HEVC) 의 맥락에서 설명되었지만, 그 기술들은 예측 유닛들 (PU들) 에 또는 다른 비디오 코딩 프로세스들 및/또는 표준들에서 적용될 수도 있음이 이해되어야만 한다.

[0184] 도 5 는 본 개시의 기술들과 부합하는, 픽셀들의 블록에 대하여 팔레트로의 인덱스들을 결정하는 일 예를 예시 하는 개념 다이어그램이다. 예를 들어, 도 5 는 인덱스 값들과 연관된 픽셀들의 개별 포지션들을 팔레트들 (244) 의 엔트리에 관련시키는 인덱스 값들 (값들 1, 2, 및 3) 의 맵 (240) 을 포함한다. 팔레트들 (244) 은 앞서 도 4 와 관련하여 설명된 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 과 유사한 방식으로 결정될 수도 있다.

[0185] 다시, 도 5 의 기술들은 비디오 인코더 (20) (도 1 및 도 2) 및 비디오 디코더 (30) (도 1 및 도 3) 의 맥락에서 및 설명의 목적들을 위해 HEVC 비디오 코딩 표준과 관련하여 설명된다. 그러나, 본 개시의 기술들이 이러한 방식으로 한정되는 것은 아니며, 다른 비디오 코딩 프로세스들 및/또는 표준들에서 다른 비디오 코딩 프로세서들 및/또는 디바이스들에 의해 적용될 수도 있음이 이해되어야 한다.

[0186] 맵 (240) 은 각각의 픽셀 포지션에 대한 인덱스 값을 포함하는 것으로 도 5 의 예에서 도시되지만, 다른 예들에서, 모든 픽셀 포지션들이 픽셀 값을 팔레트들 (244) 의 엔트리에 관련시키는 인덱스 값과 연관되지 않을 수도 있음이 이해되어야만 한다. 즉, 전술된 것과 같이, 일부 예들에서, 픽셀 값이 팔레트들 (244) 에 포함되지 않을 경우 맵 (240) 에서의 포지션에 대한 실제 픽셀 값 (또는 그 양자화 버전) 의 표시를, 비디오 인코더 (20) 는 인코딩할 수도 있다 (그리고 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 획득할 수도 있다).

[0187] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 어떤 픽셀 포지션들이 인덱스 값들과 연관되는지를 표시하는 추가의 맵을 코딩하도록 구성될 수도 있다. 예를 들어, 맵에서의 (i, j) 엔트리는 CU 의 (i, j) 포지션에 대응하는 것으로 가정한다. 비디오 인코더 (20) 는 맵의 각각의 엔트리 (즉, 각각의 픽셀 포지션) 에 대하여 그 엔트리가 연관된 인덱스 값을 표시하는 하나 이상의 선택스 값들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 의 (i, j) 위치에서의 픽셀 값이 팔레트들 (244) 에서의 값들 중 하나인

것을 나타내기 위해, 1의 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20)는 그러한 일 예에서, 그 픽셀 값을 팔레트에 표시하고 비디오 디코더가 픽셀 값을 복원하게 하기 위해 (도 5의 예에서 값들 1 내지 3으로 도시된) 팔레트 인덱스를 인코딩할 수도 있다. 팔레트들 (244)이 단일 엔트리 및 연관된 픽셀 값을 포함하는 경우들에서, 비디오 인코더 (20)는 인덱스 값의 시그널링을 스kip할 수도 있다. 비디오 인코더 (20)는 CU의 (i, j) 위치에서의 픽셀 값이 팔레트들 (244)에서의 값들 중 하나가 아닌 것을 나타내기 위해, 0의 값을 갖도록 플래그를 인코딩할 수도 있다. 이러한 예에서, 비디오 인코더 (20)는 또한, 픽셀 값을 복원할 시 비디오 디코더 (30)에 의한 사용을 위해 픽셀 값의 표시를 인코딩할 수도 있다. 일부 경우들에서, 픽셀 값은 순실 방식으로 코딩될 수도 있다.

[0188] CU의 일 포지션에서 픽셀의 값은 CU의 다른 포지션들에서 하나 이상의 다른 픽셀들의 값들의 표시를 제공할 수도 있다. 예를 들어, CU의 이웃하는 픽셀 포지션들이 동일한 픽셀 값을 가지거나, (1초파의 픽셀 값이 단일 인덱스 값으로 맵핑될 수도 있는 순실 코딩의 경우에) 동일한 인덱스 값으로 맵핑될 수도 있는 확률은 상대적으로 높을 수도 있다.

[0189] 따라서, 본 개시의 양태들에 따르면, 비디오 인코더 (20)는 동일한 픽셀 값 또는 인덱스 값을 가지는, 소정 스캔 순서에서 연속하는 픽셀들 또는 인덱스 값들의 수를 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 전술된 것과 같이, 유사 값의 픽셀 또는 인덱스 값들의 스트링은 본원에서 런으로 지칭될 수도 있다. 예시의 목적들을 위한 일 예에 있어서, 소정의 스캔 순서에서 2개의 연속하는 픽셀들 또는 인덱스들이 상이한 값들을 갖는다면, 런은 0과 동일하다. 소정의 스캔 순서에서 2개의 연속하는 픽셀들 또는 인덱스들이 동일한 값을 가지지만 스캔 순서에서 제3 픽셀 또는 인덱스는 상이한 값을 갖는다면, 런은 1과 동일하다. 동일한 값을 갖는 3개의 연속하는 인덱스들 또는 픽셀들에 대하여, 런은 2, 등등이다. 비디오 디코더 (30)는 인코딩된 비트스트림으로부터 런을 표시하는 신팩스 엘리먼트들을 획득하고, 그 데이터를 사용하여 동일한 픽셀 또는 인덱스 값을 갖는 연속하는 위치들의 수를 결정할 수도 있다.

[0190] 예시의 목적들을 위한 일 예에서, 맵 (240)의 라인 (248)을 고려한다. 수평의 좌-우 스캔 방향을 가정할 때, 라인 (248)은 "2"의 5개의 인덱스 값들 및 "3"의 3개의 인덱스 값들을 포함한다. 본 개시의 양태들에 따르면, 비디오 인코더 (20)는 스캔 방향에서 라인 (248)의 제1포지션에 대하여 2의 인덱스 값을 인코딩할 수도 있다. 추가로, 비디오 인코더 (20)는 시그널링된 인덱스 값과 동일한 인덱스 값을 가지는 스캔 방향에서 연속하는 값들의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 라인 (248)의 예에서, 비디오 인코더 (20)는 4의 런을 시그널링할 수도 있으며, 이에 따라 스캔 방향에서 뒤따르는 4개의 포지션들의 인덱스 값들이 시그널링된 인덱스 값과 동일한 인덱스 값을 공유하는 것을 표시한다. 비디오 인코더 (20)는 라인 (248)에서 다음 상이한 인덱스 값에 대하여 동일한 프로세스를 수행할 수도 있다. 즉, 비디오 인코더 (20)는 3의 인덱스 값 및 2의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 비디오 디코더 (30)는 인덱스 값 및 동일한 인덱스 값을 갖는 스캔 방향에서 연속하는 인덱스들의 수 (런)를 표시하는 신팩스 엘리먼트들을 획득할 수도 있다.

[0191] 전술된 것과 같이, 맵의 인덱스들은 특정 순서로 스캔된다. 본 개시의 양태들에 따르면, 스캔 방향은 수직 이거나, 수평이거나, 대각 방향 (예컨대, 블록에서 대각선으로 45도 또는 135도)일 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 블록의 인덱스들을 스캔하기 위한 스캔 방향을 표시하는, 각 블록에 대한 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 부가적으로 또는 대안적으로, 스캔 방향은 예컨대, 블록 사이즈, 컬러 공간, 및/또는 컬러 성분과 같은 소위 부가 정보에 기초하여 시그널링되거나 추론될 수도 있다. 비디오 인코더 (20)는 블록의 각각의 컬러 성분에 대한 스캔들을 명시할 수도 있다. 대안적으로, 명시된 스캔은 블록의 모든 컬러 성분들에 적용할 수도 있다.

[0192] 예를 들어, 컬럼 기반 스캔에 대하여, 맵 (240)의 컬럼 (252)을 고려한다. 수직의 상-하 스캔 방향을 가정할 때, 컬럼 (252)은 "2"의 6개의 인덱스 값들 및 "3"의 2개의 인덱스 값들을 포함한다. 본 개시의 양태들에 따르면, 비디오 인코더 (20)는 (컬럼 (252)의 상대적으로 상부에서) 스캔 방향에서 라인 (252)의 제1포지션에 대하여 2의 인덱스 값을 인코딩할 수도 있다. 추가로, 비디오 인코더 (20)는 5의 런을 시그널링할 수도 있으며, 이에 따라 스캔 방향에서 뒤따르는 5개의 포지션들의 인덱스 값들이 시그널링된 인덱스 값과 동일한 인덱스 값을 공유하는 것을 표시한다. 비디오 인코더 (20)는 또한 스캔 방향에서 다음 포지션에 대하여 3의 인덱스 값 및 1의 런을 표시하는 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 비디오 디코더 (30)는 인덱스 값 및 동일한 인덱스 값을 갖는 스캔 방향에서 연속하는 인덱스들의 수 (런)를 표시하는 신팩스 엘리먼트들을 획득할 수도 있다.

[0193] 본 개시의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 부가적으로 또는 대안적으로, 맵 (240) 의 하나 이상의 엔트리들에 대하여 라인 카피를 수행할 수도 있다. 라인 카피는, 일부 예들에서, 스캔 방향에 의존할 수도 있다. 예를 들어, 비디오 인코더 (20) 가 맵에서의 특정 엔트리에 대한 픽셀 또는 인덱스 값이 (수평 스캔에 대하여) 특정 엔트리 상부의 라인에서의 픽셀 또는 인덱스 값, 또는 (수직 스캔에 대하여) 특정 엔트리 좌측의 컬럼과 동일한 것을 나타낼 수도 있다. 비디오 인코더 (20) 는 또한, 특정 엔트리의 상부의 라인 또는 좌측의 컬럼에서의 인덱스들과 동일한, 스캔 순서에서의 픽셀 또는 인덱스 값들의 수를, 런으로서 나타낼 수도 있다. 이러한 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 명시된 이웃하는 라인으로부터 및 현재 코딩되고 있는 맵의 라인에 대한 명시된 수의 엔트리들로부터 픽셀 또는 인덱스 값들을 카피할 수도 있다.

[0194] 예시의 목적들을 위한 일 예에서, 맵 (240) 의 컬럼들 (256 및 260) 을 고려한다. 수직의 상-하 스캔 방향을 가정할 때, 컬럼 (256) 은 "1" 의 3 개의 인덱스 값들 및 "2" 의 3 개의 인덱스 값들을 포함한다. 컬럼 (260) 은 스캔 방향에서 동일한 순서를 갖는 동일한 인덱스 값들을 포함한다. 본 개시의 양태들에 따라, 비디오 인코더 (20) 는 전체 컬럼 (260) 이 컬럼 (256) 으로부터 카피되는 것을 표시하는 하나 이상의 선택스 엘리먼트들을 컬럼 (260) 에 대해 인코딩할 수도 있다. 하나 이상의 선택스 엘리먼트들은 맵 (240) 의 상대적 인 상부에서 컬럼 (260) 의 제 1 엔트리와 연관될 수도 있다. 비디오 디코더 (30) 는 라인 카피를 표시하는 선택스 엘리먼트들을 획득하고, 컬럼 (260) 을 디코딩할 때, 컬럼 (260) 에 대하여 컬럼 (256) 의 인덱스 값들을 카피할 수도 있다.

[0195] 본 개시의 양태들에 따르면, 소위 엔트리들의 런들을 코딩하기 위한 기술들은 전술된 라인 카피를 위한 기술들과 함께 사용될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 맵에서의 엔트리의 값이 팔레트로부터 획득되는지, 또는 맵에서의 엔트리의 값이 맵 (240) 에서 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 그 라인 (로우 또는 컬럼) 에서의 엔트리의 위치를 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 동일한 값을 공유하는 엔트리들의 수를 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 그러한 정보를 획득하고, 그 정보를 사용하여 블록에 대한 픽셀 값들 및 맵을 복원할 수도 있다.

[0196] 예시의 목적들을 위한 일 예에서, 맵 (240) 의 컬럼들 (264 및 268) 을 고려한다. 수평의 좌-우 스캔 방향을 가정할 때, 로우 (264) 는 "1" 의 3 개의 인덱스 값들, "2" 의 2 개의 인덱스 값들 및 "3" 의 3 개의 인덱스 값들을 포함한다. 로우 (268) 는 "1" 의 5 개의 인덱스 값들 및 "3" 의 3 개의 인덱스 값들을 포함한다. 이러한 예에서, 비디오 인코더 (20) 는 로우 (268) 에 대하여 데이터를 인코딩할 때 런을 뒤따르는 로우 (264) 의 특정 엔트리들을 식별할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 로우 (268) 의 제 1 포지션 (로우 (268) 의 최좌측 포지션) 이 로우 (264) 의 제 1 포지션과 동일한 것을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 로우 (268) 에서 스캔 방향에서 2 개의 연속하는 엔트리들의 다음 런이 로우 (264) 의 제 1 포지션과 동일한 것을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다.

[0197] 일부 예들에서, 비디오 인코더 (20) 는 또한, 현재 픽셀 또는 인덱스 값을 다른 로우 (또는 컬럼) 에서의 포지션과 관련하여 코딩할 것인지 또는 런 선택스 엘리먼트를 사용하여 현재 픽셀 또는 인덱스 값을 코딩할 것인지 여부를 결정할 수도 있다. 예를 들어, 로우 (264) 의 제 1 포지션 및 (전술된) 2 개 엔트리들의 런을 표시하는 하나 이상의 선택스 엘리먼트를 인코딩한 후에, 비디오 인코더 (20) 는 라인 (268) 에서 (좌측으로부터 우측으로) 제 4 및 제 5 포지션들에 대하여, 제 4 포지션에 대하여 1 의 값을 표시하는 하나 이상의 선택스 엘리먼트들 및 1 의 런을 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 따라서, 비디오 인코더 (20) 는 다른 라인 (또는 컬럼) 을 참조하지 않고 이들 2 개의 포지션들을 인코딩한다. 그 후, 비디오 인코더 (20) 는 로우 (268) 에서 3 의 인덱스 값을 갖는 제 1 포지션을 (예컨대, 상부 로우 (264)로부터의 카피, 및 동일한 인덱스 값을 갖는 스캔 순서에서 연속하는 포지션들의 런을 표시하는) 상부 로우 (264) 와 관련하여 코딩할 수도 있다. 따라서, 본 개시의 양태들에 따르면, 비디오 인코더 (20) 는 예컨대, 런을 사용하여 라인 (또는 컬럼) 의 픽셀 또는 인덱스 값들을 그 라인 (또는 컬럼) 의 다른 값들과 관련하여 코딩하거나, 라인 (또는 컬럼) 의 픽셀 또는 인덱스 값들을 다른 라인 (또는 컬럼) 의 값들과 관련하여 코딩하거나, 이들의 조합 사이에서 선택할 수도 있다. 비디오 인코더 (20) 는 일부 예들에서, 선택을 실행하기 위해 레이트/왜곡 최적화를 수행할 수도 있다.

[0198] 비디오 디코더 (30) 는 전술된 선택스 엘리먼트들을 수신하고 로우 (268) 를 복원할 수도 있다. 예를 들어,

비디오 디코더 (30)는 현재 코딩되고 있는 맵 (240)의 포지션에 대한 연관된 인덱스 값을 카피하기 위한 이웃하는 로우에서의 특정 위치를 표시하는 데이터를 획득할 수도 있다. 비디오 디코더 (30)는 또한, 동일한 인덱스 값을 갖는 스캔 순서에서 연속하는 포지션들의 수를 표시하는 데이터를 획득할 수도 있다.

[0199] 일부 경우들에서, 엔트리들이 카피되는 라인은 (도 5의 예들에 도시된 것과 같이) 현재 코딩되고 있는 라인의 엔트리에 직접 인접할 수도 있다. 그러나, 다른 예들에서, 다수의 라인들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 베퍼링되어 맵의 다수의 라인들 중 임의의 것이 현재 코딩되고 있는 맵의 라인을 위한 예측 엔트리들로서 사용될 수도 있다. 따라서, 일부 예들에서, 엔트리에 대한 픽셀 값은 현재 로우 바로 위의 로우 (또는 좌측의 걸림) 또는 현재 로우 위의 2 이상의 로우들 (또는 좌측의 걸림)에서의 엔트리의 픽셀 값들과 동일하게 시그널링될 수도 있다.

[0200] 예시의 목적들을 위한 일 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 엔트리들의 현재 로우를 코딩하기 전에 엔트리들의 이전 n 개 로우들을 저장하도록 구성될 수도 있다. 이러한 예에서, 비디오 인코더 (20)는 절단된 1진 코드 또는 다른 코드들을 갖는 비트스트림에 예측 로우 (엔트리들이 카피되는 로우)를 표시할 수도 있다. 다른 예에서, 비디오 인코더 (20)는 현재 라인을 코딩하기 위한 참조로서 사용되는 맵 (240)의 예측 라인과 현재 라인 간의 변위 값을 인코딩할 수도 있다 (디코더 (30)는 디코딩할 수도 있다).

즉, 비디오 인코더 (20)는 인덱스 값이 카피되는 특정 라인 (또는 걸림)의 표시를 인코딩할 수도 있다. 일부 예들에서, 변위 값은 변위 벡터일 수도 있다. 즉, c[0], c[1], ..., 는 맵 (240)의 현재 라인의 인덱스들을 표시한다고 하고, u[0], u[1], u[2], ..., 는 상부 이웃하는 라인과 같은 맵 (240)의 예측 라인의 인덱스들을 표시한다고 하자. 이러한 예에서, 변위 벡터가 d라면, c[i]에 대한 인덱스 값은 u[i+d]로부터 예측될 수도 있다. d의 값은 1진, 절단된 1진, 지수 골룸 또는 골룸 라이스 코드들을 사용하여 코딩될 수도 있다.

[0201] 도 6은 본 개시의 기술들과 부합하는, 팔레트 코딩 모드를 사용하여 비디오 데이터를 코딩하는 예시적인 프로세스를 도시하는 플로우차트이다. 도 6의 방법은 비디오 인코더 (20) (도 1 및 도 2) 또는 비디오 디코더 (30) (도 1 및 도 3)와 같은 비디오 코더와 관련하여 설명된다. 그러나, 다른 비디오 코딩 디바이스들이 유사한 방법을 수행하도록 구성될 수도 있는 것이 이해되어야만 한다. 또한, 그 방법의 특정 단계들은 상이한 순서로 또는 병렬로 수행될 수도 있다. 유사하게, 다양한 예들에서 특정 단계들은 생략될 수도 있고, 다른 단계들이 부가될 수도 있다.

[0202] 비디오 인코더 및/또는 비디오 디코더 (30)와 같은 비디오 코더는 또한, 비디오 데이터의 현재 블록을 코딩하기 위한 모드가 팔레트-기반 코딩 모드인지 여부를 초기에 결정할 수도 있다 (280). 전술된 것과 같이, 하나 이상의 선택스 엘리먼트들은 비트스트림에 포함될 수도 있고 현재 블록의 코딩 모드를 표시한다 (예컨대, PLT_Mode_flag 선택스 엘리먼트). 다른 예들에서, 비디오 코더는 전술된 것과 같은 소위 부가 정보에 기초하여 결정을 실행할 수도 있다.

[0203] 임의의 경우, 팔레트-기반 코딩 모드가 현재 코딩되고 있는 블록에 대하여 사용되지 않는다면 (단계 (280)의 "아니오" 브랜치), 비디오 코더는 팔레트-기반의 코딩 모드 이외의 모드를 사용하여 비디오 데이터의 블록을 코딩할 수도 있다 (282). 예를 들어, 비디오 코더는 인트라-모드, 인터-모드, 또는 다른 코딩 모드와 같은 비-팔레트 기반 모드를 사용하여 비디오 데이터의 블록을 코딩할 수도 있다.

[0204] 팔레트-기반의 코딩 모드가 현재 코딩되고 있는 블록에 대하여 사용된다면 (단계 (280)의 "예" 브랜치), 비디오 코더는 현재 블록을 코딩하기 위한 팔레트를 결정할 수도 있다 (284). 이하 도 7 및 도 8과 관련하여 설명되는 것과 같이, 일부 예들에서, 비디오 코더는 비디오 데이터의 하나 이상의 다른, 이전에 코딩된 블록들과 연관된 팔레트에 기초하여 현재 블록에 대한 팔레트를 결정할 수도 있다.

[0205] 비디오 코더는 또한, 현재 코딩되고 있는 블록에 대한 인덱스 값을 결정할 수도 있다 (286). 예를 들어, 이하 도 9와 관련하여 더 상세히 설명되는 것과 같이, 비디오 코더는 블록의 어떤 픽셀 포지션들이 픽셀의 포지션을 연관된 픽셀 값을 갖는 결정된 팔레트의 엔트리에 관련시키는 인덱스 값을 사용하여 코딩되는지를 표시하는 맵을 결정할 수도 있다. 일부 경우들에서, 비디오 코더는 하나 이상의 인덱스 값을 블록의 다른 인덱스 값들과 관련하여 결정할 수도 있다. 예를 들어, 비디오 코더는 인덱스 값을 런 및/또는 인덱스 값을 그 블록의 다른 라인 또는 걸림에 로케이팅된 인덱스 값에 기초하여 결정할 수도 있다.

[0206] 그 후, 비디오 코더는 결정된 팔레트 및 인덱스 값을 사용하여 블록의 비디오 데이터를 코딩한다 (280). 예를 들어, 비디오 인코더 (20)와 관련하여, 비디오 인코더 (20)는 인코딩된 비트스트림에서 팔레트뿐만 아니라

라 인덱스 값들을 표시하는 데이터를 인코딩할 수도 있다. 비디오 인코더 (20)는 또한, 결정된 팔레트에서 대응하는 픽셀을 가지지 않는 임의의 픽셀 포지션들에 대하여, 그러한 포지션들에서의 실제 픽셀 값들 또는 그들의 양자화된 버전들을 인코딩할 수도 있다. 비디오 디코더 (30)는 인코딩된 비트스트림으로부터 팔레트, 인덱스 값들, 및 픽셀 값들을 파싱하고 디코딩할 수도 있고, 그 데이터를 사용하여 비디오 데이터의 블록을 복원할 수도 있다.

[0207] 도 7 은 본 개시의 기술들과 부합하는, 팔레트-기반 코딩에서 팔레트를 결정하는 예시적인 프로세스를 도시하는 플로우차트이다. 도 7 의 방법은 비디오 인코더 (20) (도 1 및 도 2) 또는 비디오 디코더 (30) (도 1 및 도 3) 와 같은 비디오 코더와 관련하여 설명된다. 그러나, 다른 비디오 코딩 디바이스들이 유사한 방법을 수행하도록 구성될 수도 있는 것이 이해되어야만 한다. 또한, 그 방법의 특정 단계들은 상이한 순서로 또는 병렬로 수행될 수도 있다. 유사하게, 다양한 예들에서 특정 단계들은 생략될 수도 있고, 다른 단계들이 부가될 수도 있다. 일부 예들에서, 도 7 의 기술들은 도 6 의 단계 (284) 동안 수행될 수도 있다.

[0208] 도 7 의 예에서, 비디오 코더는 팔레트의 사이즈를 결정할 수도 있다 (300). 일부 예들에서, 팔레트의 사이즈는 고정될 수도 있다. 다른 예들에서, 팔레트의 사이즈는 코딩 동안 (예컨대, 팔레트로부터 엔트리들을 부가하거나 제거함으로써) 동적으로 조정될 수도 있다. 비디오 코더는 팔레트의 사이즈를 표시하는 하나 이상의 선택스 엘리먼트들을 코딩할 수도 있다.

[0209] 비디오 코더는 현재 블록을 코딩하기 위한 팔레트가 하나 이상의 다른, 이전의 코딩된 팔레트들로부터 예측되는지 여부를 결정할 수도 있다 (302). 팔레트가 예측된다면 (단계 (302) 의 예 브랜치), 비디오 코더는 예측 팔레트를 결정할 수도 있다 (304). 예를 들어, 비디오 인코더 (20)는 현재 블록에 대한 팔레트가 하나 이상의 이전에 인코딩된 팔레트들에 기초하는 것을 표시하는 데이터뿐만 아니라 예측 팔레트와 연관된 블록의 위치를 표시하는 (또는 예측 팔레트 자체를 식별하는) 데이터를 인코딩할 수도 있다. 유사하게, 비디오 디코더 (30)는 인코딩된 비트스트림으로부터 그러한 데이터를 획득할 수도 있다. 도 4 와 관련하여 전술된 것과 같이, 예측 팔레트는 현재 코딩되고 있는 블록의 이웃하는 블록과 연관될 수도 있다.

[0210] 비디오 코더는 결정된 예측 팔레트의 하나 이상의 엔트리들에 기초하여 현재 블록을 코딩하기 위한 팔레트의 하나 이상의 엔트리들을 결정할 수도 있다 (306). 일부 예들에서, 비디오 코더는 현재 블록을 코딩하기 위해 또 다른 블록으로부터 전체 팔레트를 카피할 수도 있다. 다른 예들에서, 비디오 코더는 또 다른 팔레트로부터 엔트리들을 선택적으로 카피할 수도 있다. 팔레트 예측은 일부 예들에서, 팔레트 엔트리들이 하나 이상의 이전에 코딩된 블록들에서 사용되는 빈도에 기초할 수도 있다.

[0211] 비디오 코더는 또한, 팔레트의 하나 이상의 예측되지 않은 엔트리들을 결정할 수도 있다 (308). 예를 들어, 일부 경우들에서, 팔레트의 오직 일 부분만이 다른 이전에 코딩된 팔레트들로부터 예측될 수도 있다. 다른 경우들에서, 팔레트는 결코 예측 코딩되지 않을 수도 있다. 그러한 경우들에서, 비디오 코더는 본원에 설명된 팔레트 예측 기술들을 수행하지 않고 (또는 수행한 후에) 팔레트 엔트리들을 결정할 수도 있다.

[0212] 도 8 은 본 개시의 기술들과 부합하는, 비디오 데이터의 블록의 인덱스들을 결정하는 예시적인 프로세스를 도시하는 플로우차트이다. 도 8 의 방법은 비디오 인코더 (20) (도 1 및 도 2) 또는 비디오 디코더 (30) (도 1 및 도 3) 와 같은 비디오 코더와 관련하여 설명된다. 그러나, 다른 비디오 코딩 디바이스들이 유사한 방법을 수행하도록 구성될 수도 있는 것이 이해되어야만 한다. 또한, 그 방법의 특정 단계들은 상이한 순서로 또는 병렬로 수행될 수도 있다. 유사하게, 다양한 예들에서 특정 단계들은 생략될 수도 있고, 다른 단계들이 부가될 수도 있다. 일부 예들에서, 도 8 의 기술들은 도 6 의 단계 (286) 동안 수행될 수도 있다.

[0213] 비디오 코더는 예컨대, 팔레트 기반의 코딩에서 인덱스 값들의 맵을 스캔하기 위한 스캔 방향을 초기에 결정할 수도 있다 (320). 일부 예들에서, 도 5 의 예와 관련하여 전술된 것과 같이, 비디오 코더는 특정 스캔 방향을 사용하도록 미리 구성될 수도 있다. 다른 예들에서, 비디오 코더는 스캔 방향을 표시하는 하나 이상의 선택스 엘리먼트들을 코딩하고, 그 선택스 엘리먼트들에 기초하여 스캔 방향을 결정할 수도 있다. 또 다른 예들에서, 비디오 코더는 소위 부가 정보에 기초하여 스캔 방향을 결정할 수도 있다.

[0214] 비디오 코더는 또한 팔레트의 연관된 인덱스 값들을 갖는, 현재 코딩되고 있는 비디오 데이터의 블록의 포지션들의 맵을 결정할 수도 있다. 예를 들어, 현재 코딩되고 있는 블록의 하나 이상의 픽셀 값들은 그 블록으로부터의 팔레트에 표현되지 않을 수도 있다. 그러한 예들에서, 비디오 코더는 (팔레트에서 픽셀 값을 식별하는 인덱스 값을 코딩하는 것보다) 픽셀 값들을 직접 코딩할 수도 있다. 맵은 어떤 픽셀 포지션들이 인덱스 값과 연관되는지 및 어떤 픽셀 포지션들이 인덱스 값과 연관되지 않는지를 표시할 수도 있다.

- [0215] 비디오 코더는 하나 이상의 제 1 인덱스 값들을 결정한다 (324). 블록에서의 소정의 픽셀에 대하여, 비디오 코더는 소정의 픽셀의 픽셀 값을 그 블록에 대한 팔레트에서의 엔트리에 관련시키는 인덱스 값을 결정할 수도 있다.
- [0216] 비디오 코더는 또한 제 1 인덱스 값들에 기초하여 하나 이상의 제 2 인덱스 값들을 결정한다 (326). 예를 들어, 앞서 도 5 와 관련하여 설명된 것과 같이, 비디오 코더는 하나 이상의 인덱스 값들을 다른 이미 코딩된 인덱스 값들과 관련하여 결정할 수도 있다. 일부 예들에서, 비디오 코더는 연속하는 동일 값의 인덱스 값들의 런에 기초하여 제 2 인덱스 값들을 결정할 수도 있다. 다른 예들에서, 비디오 코더는 현재 코딩되고 있는 픽셀들의 라인과 상이한 라인에 포함된 하나 이상의 제 1 인덱스 값들에 기초하여 제 2 인덱스 값들을 결정할 수도 있다. 다른 예들에서, 비디오 코더는 제 2 인덱스 값들을 결정하기 위한 기술들의 조합을 적용할 수도 있다.
- [0217] 예에 의존하여, 본 명세서에서 설명된 기술들 중 임의의 기술의 특정 동작들 또는 이벤트들은 상이한 순서로 수행될 수 있고, 전체적으로 부가되거나, 병합되거나, 또는 제거될 수도 있음 (예를 들어, 설명된 모든 동작들 또는 이벤트들이 그 기술들의 실시를 위해 필수적인 것은 아님) 이 인식되어야 한다. 더욱이, 특정 예들에 있어서, 동작들 또는 이벤트들은 순차적인 것보다는, 예를 들어, 멀티-스레드 프로세싱, 인터럽트 프로세싱 또는 다중의 프로세서들을 통해 동시에 수행될 수도 있다. 추가로, 본 개시물의 특정 양태들은 명확함을 목적으로 단일 모듈 또는 유닛에 의해 수행되는 것으로 설명되지만, 본 개시물의 기술들은 비디오 코더와 연관된 유닛들 또는 모듈들의 조합에 의해 수행될 수도 있음이 이해되어야 한다.
- [0218] 본 개시의 특정 양태들은 예시의 목적들로 HEVC 표준을 개발하는 것과 관련하여 설명되었다. 하지만, 본 개시에서 설명된 기술들은 다른 표준 또는 아직 개발되지 않은 전매특허의 비디오 코딩 프로세스들을 포함하는 다른 비디오 코딩 프로세스들에 대해 유용할 수도 있다.
- [0219] 전술된 기술들은 양자가 일반적으로 비디오 코더로 지칭될 수도 있는 비디오 인코더 (20) (도 1 및 도 2) 및/또는 비디오 디코더 (30) (도 1 및 도 3) 에 의해 수행될 수도 있다. 유사하게, 비디오 코딩은 적용 가능한 경우, 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.
- [0220] 기술들의 다양한 양태들의 특정 조합들이 전술되었지만, 이를 조합들은 단지 본 개시에 설명된 기술들의 예들을 예시하기 위해 제공된다. 따라서, 본 개시의 기술들은 이들 예시적인 조합들에 제한되어야 하는 것이 아니고, 본 개시에 설명된 기술들의 다양한 양태들의 임의의 상상할 수 있는 조합을 함축할 수도 있다.
- [0221] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 상기 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 또는 컴퓨터 판독가능 매체를 통해 송신되며 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예를 들어, 통신 프로토콜에 따라, 일 장소에서 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체는 일반적으로 (1) 비-일시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는, 본 개시에서 설명된 기술들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위한 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.
- [0222] 비-제한적인 예로서, 이러한 컴퓨터 판독가능 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 스토리지 디바이스들, 플래시 메모리, 또는 요구되는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하기 위해 사용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속물이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들면, 명령들이 동축 케이블, 광섬유 케이블, 연선, 디지털가입자 회선 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 반송파들, 신호들, 또는 다른 일시적 매체들을 포함하는 것이 아니라, 대신에 비-일시적인, 유형의 저장 매체들과 관련되는 것이 이해되어야만 한다. 본원에서 이용된 디스크 (disk) 와 디스크 (disc) 는, 컴팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크, 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 통상 자기적으로 데이터를 재생하는 반면, 디스크 (disc) 들은 레이저들을 이용하여 광학적으로 데이터를

재생한다. 위의 조합들도 컴퓨터 관독가능 매체들의 범위 내에 포함되어야 한다.

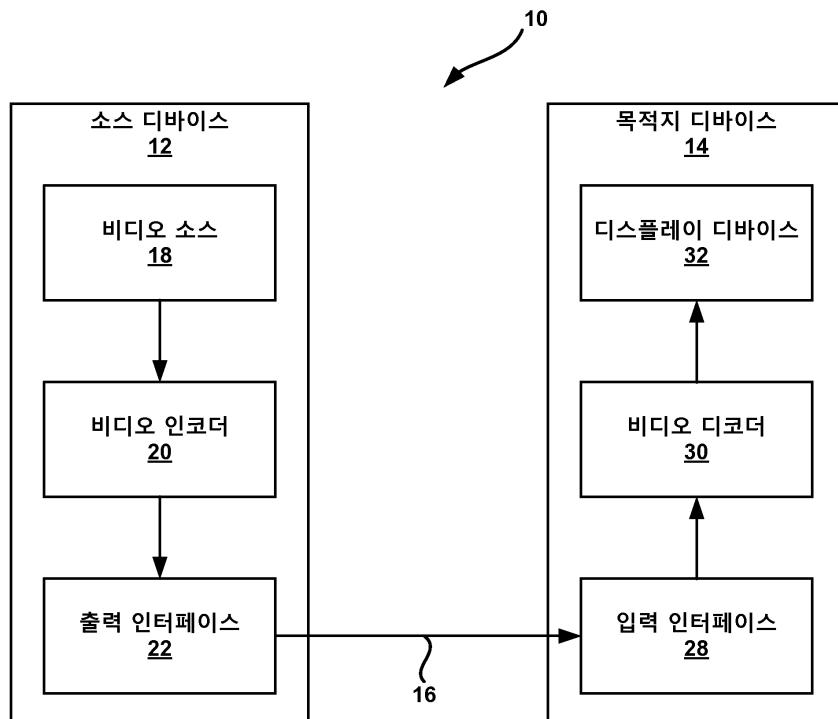
[0223] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적된 또는 별도의 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본 명세서에서 사용되는 바와 같은 용어 "프로세서"는 본 명세서에서 설명된 기술들의 구현에 적절한 전술한 구조 또는 임의의 다른 구조 중 임의의 구조를 지칭할 수도 있다. 부가적으로, 일부 양태들에 있어서, 본 명세서에서 설명된 기능은 인코딩 및 디코딩을 위해 구성되고 결합된 코텍에서 통합된 전용 하드웨어 모듈 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 그 기술들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0224] 본 개시의 기술들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩 세트)를 포함하여 매우 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들 또는 유닛들이 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시에서 설명되지만, 반드시 상이한 하드웨어 유닛들에 의한 실현을 요구하지는 않는다. 오히려, 상기 설명된 바와 같이, 다양한 유닛들은 적절한 소프트웨어 및/또는 펌웨어와 함께 상기 설명된 바와 같은 하나 이상의 프로세서들을 포함하여 코텍 하드웨어 유닛으로 결합되거나 상호작용하는 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

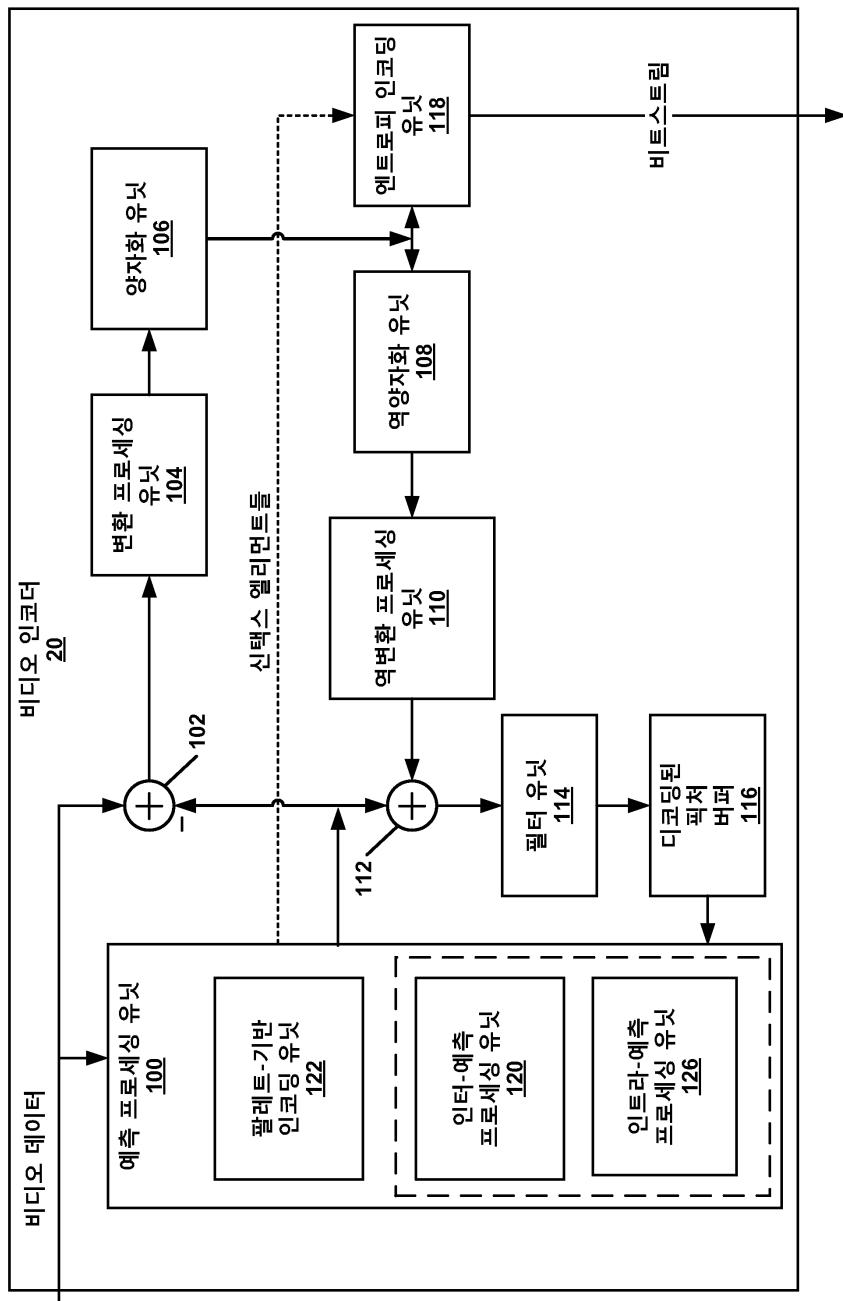
[0225] 다양한 예들이 설명되었다. 이들 및 다른 예들은 하기의 특허청구범위 내에 있다.

도면

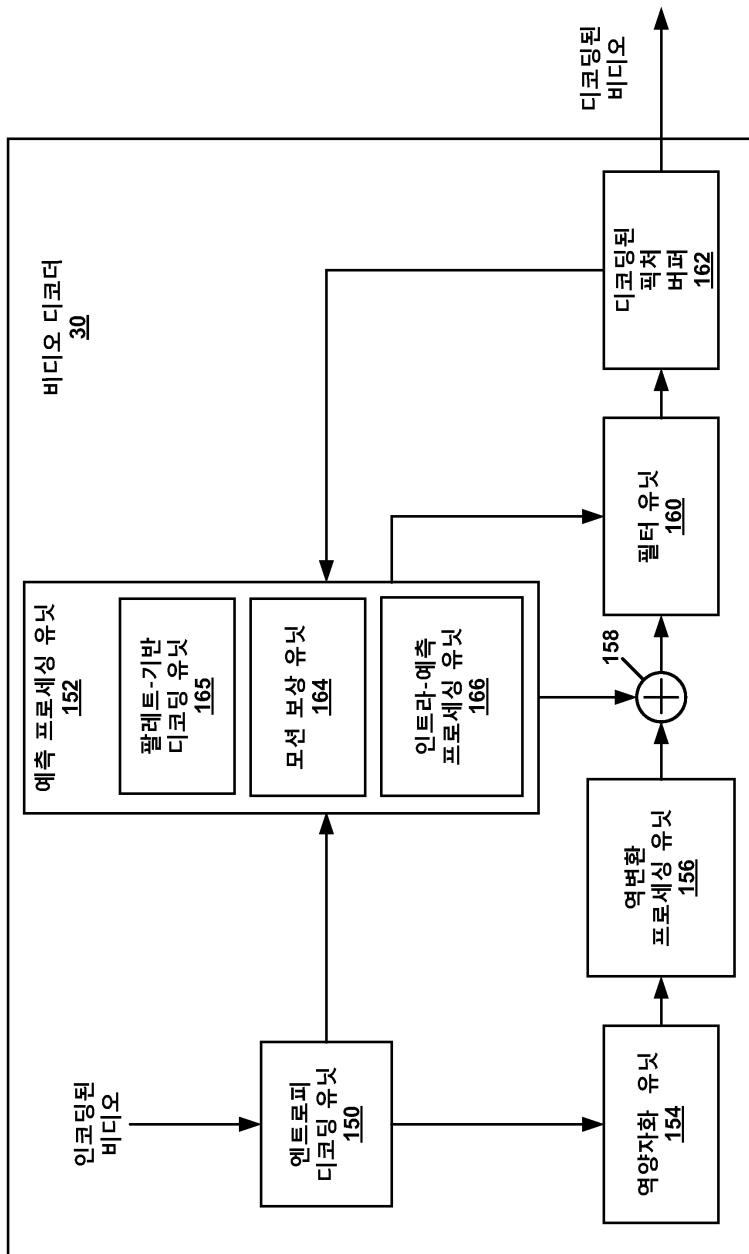
도면1



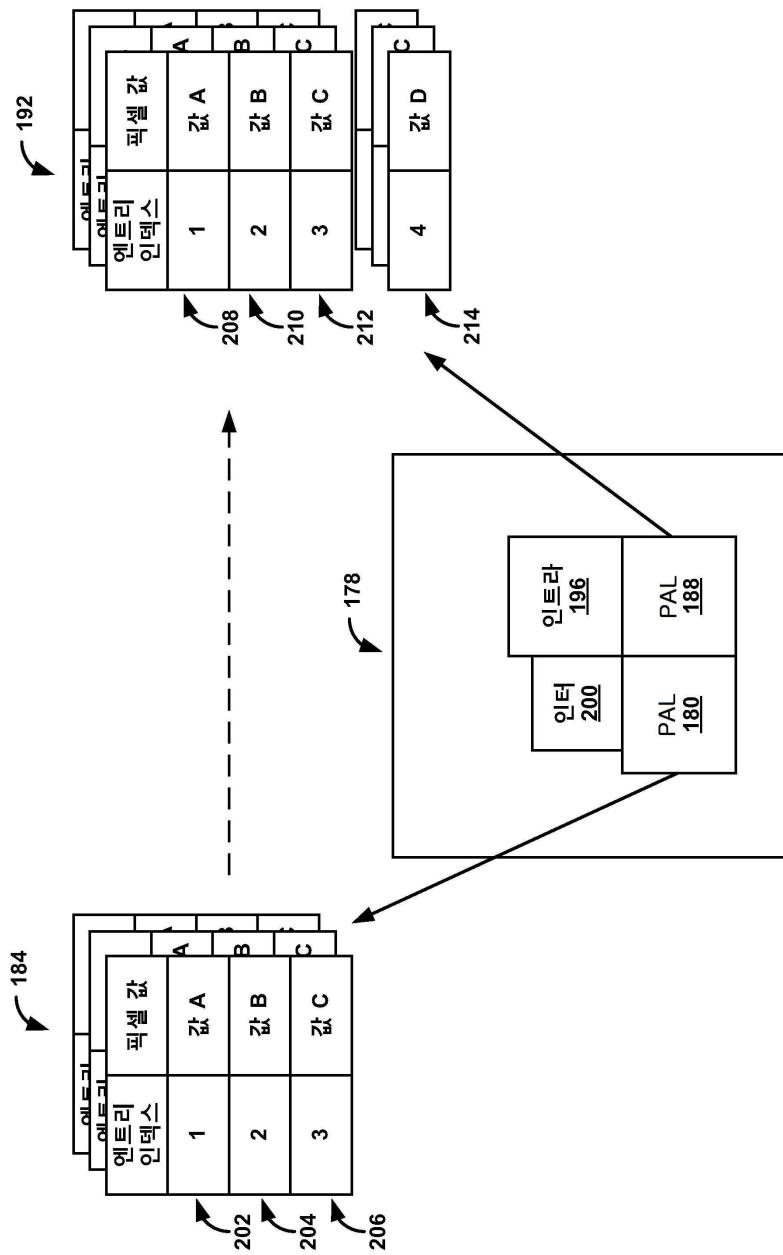
도면2



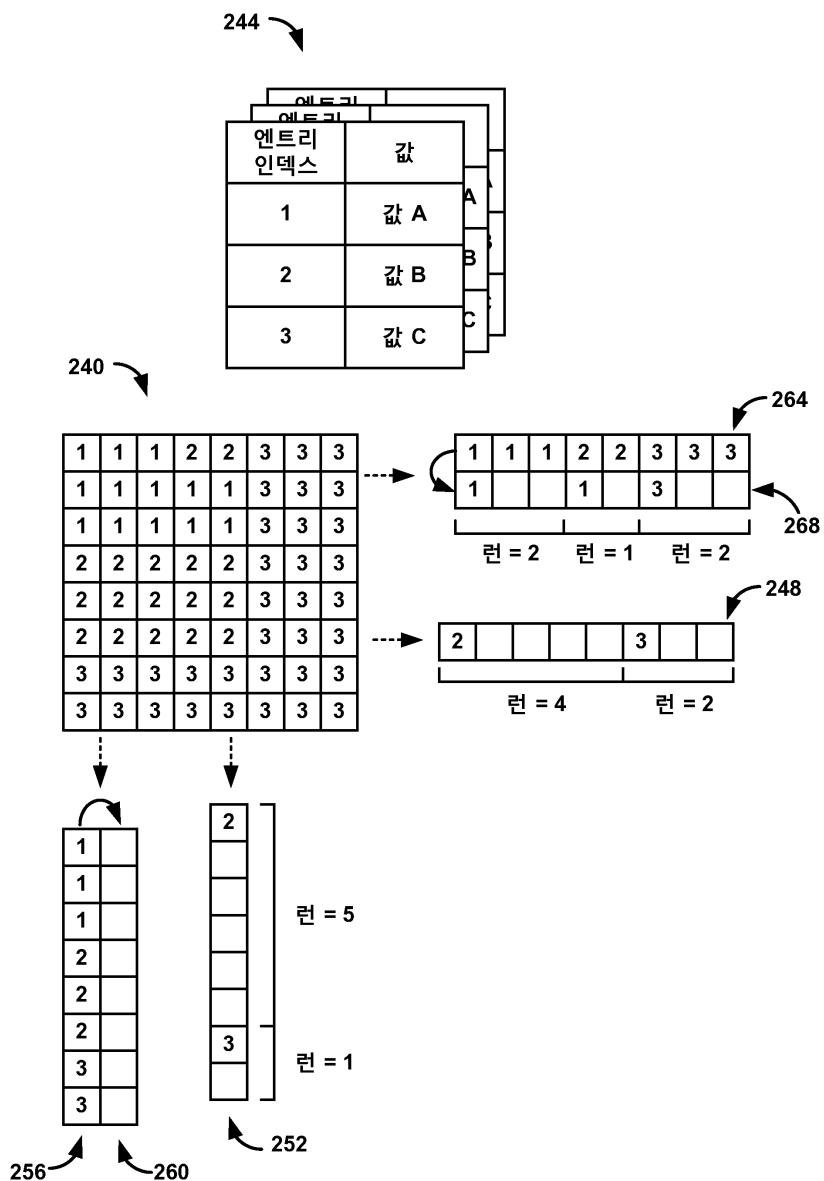
도면3



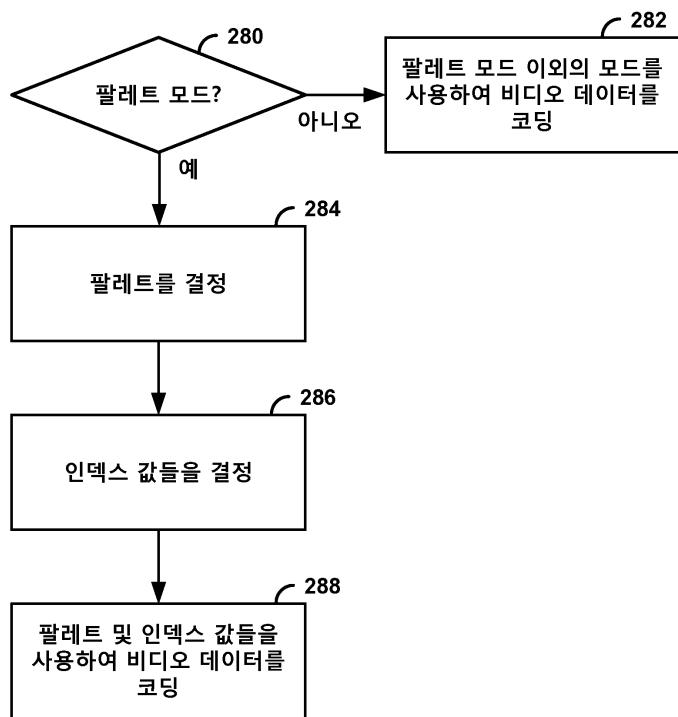
도면4



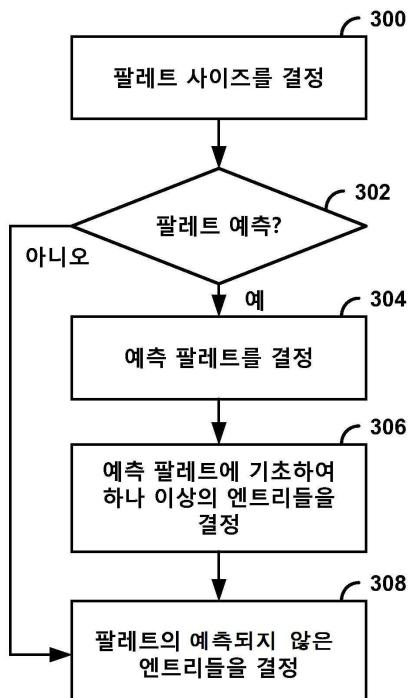
도면5



도면6



도면7



도면8

