



등록특허 10-2305988



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년09월27일
(11) 등록번호 10-2305988
(24) 등록일자 2021년09월17일

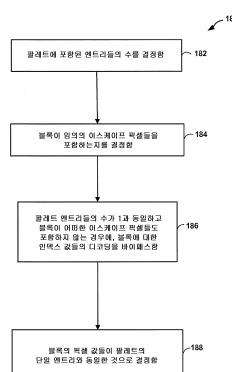
- (51) 국제특허분류(Int. Cl.)
H04N 19/46 (2014.01) *H04N 19/124* (2014.01)
H04N 19/127 (2014.01) *H04N 19/14* (2014.01)
H04N 19/176 (2014.01) *H04N 19/18* (2014.01)
H04N 19/583 (2014.01) *H04N 19/593* (2014.01)
H04N 19/93 (2014.01) *H04N 19/94* (2014.01)
- (52) CPC특허분류
H04N 19/46 (2015.01)
H04N 19/124 (2015.01)
- (21) 출원번호 10-2016-7020836
- (22) 출원일자(국제) 2014년12월31일
 심사청구일자 2019년12월12일
- (85) 번역문제출일자 2016년07월28일
- (65) 공개번호 10-2016-0102071
- (43) 공개일자 2016년08월26일
- (86) 국제출원번호 PCT/US2014/072994
- (87) 국제공개번호 WO 2015/103390
 국제공개일자 2015년07월09일
- (30) 우선권주장
 61/923,163 2014년01월02일 미국(US)
 (뒷면에 계속)
- (56) 선행기술조사문헌
 비특허문헌1(Xun Guo 등, JCTVC-00182, 2013.10.25)*
 (뒷면에 계속)
- 전체 청구항 수 : 총 20 항
- 심사관 : 조우연
- (54) 발명의 명칭 팔레트-기반 비디오 코딩을 위한 컬러 인덱스 코딩

(57) 요 약

팔레트-기반 코딩에서, 비디오 코더는 주어진 블록의 비디오 데이터를 나타내는 컬러들의 테이블로서 소위 “팔레트”를 형성할 수도 있다. 비디오 코더는 비디오 데이터의 현재 블록의 하나 이상의 픽셀 값들에 대한 인덱스 값들을 코딩할 수도 있고, 여기서, 인덱스 값들은 현재 블록의 픽셀 값들을 나타내는 팔레트에서의 엔트리

(뒷면에 계속)

대 표 도 - 도4



들을 나타낸다. 일 방법은 팔레트에서의 엔트리들의 수를 결정하는 단계, 및 비디오 데이터의 블록이 팔레트에서의 어느 엔트리와도 연관되지 않는 임의의 이스케이프 팩셀들을 포함하는지를 결정하는 단계를 포함한다.

그 방법은, 엔트리들의 수가 1이고 블록이 어느 이스케이프 팩셀들도 포함하지 않는다고 결정하는 것에 응답하여, 블록의 팩셀 값들에 대한 인덱스 값들의 디코딩을 바이패싱하는 단계, 및 블록의 팩셀 값들을 팔레트에서의 하나의 엔트리와 동일한 것으로 결정하는 단계를 포함한다.

(52) CPC특허분류

HO4N 19/127 (2015.01)*HO4N 19/14* (2015.01)*HO4N 19/176* (2015.01)*HO4N 19/18* (2015.01)*HO4N 19/583* (2015.01)*HO4N 19/593* (2015.01)*HO4N 19/93* (2015.01)*HO4N 19/94* (2015.01)

(72) 발명자

솔레 로할스 호텔

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

귀리웨이

미국 92122 캘리포니아주 샌디에고 303 피오레 테라스 5230

조쉬 라잔 택스맨

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

추평

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(56) 선행기술조사문헌

JP11088700 A*

JP2010045589 A*

비특허문현2(Sunil Lee 등, JCTVC-N0104, 2013.07.18)*

US20110311151 A1

비특허문현3(Wei Pu 등, JCTVC-P0231, 2014.01.07)

*는 심사관에 의하여 인용된 문현

(30) 우선권주장

61/924,141 2014년01월06일 미국(US)

61/969,759 2014년03월24일 미국(US)

14/586,805 2014년12월30일 미국(US)

명세서

청구범위

청구항 1

인코딩된 비디오 데이터를 디코딩하는 방법으로서,

디코딩될 비디오 데이터의 인코딩된 블록을 표현하는 데이터를 포함하는 인코딩된 비디오 비트스트림을 수신하는 단계로서, 상기 인코딩된 블록은 팔레트-기반 코딩에 따라 인코딩되는, 상기 인코딩된 비디오 비트스트림을 수신하는 단계;

디코딩될 상기 비디오 데이터의 인코딩된 블록의 픽셀 값을 표현하는데 사용된 팔레트에 포함된 엔트리들의 수를 나타내는 제 1 데이터를 획득하기 위해 수신된 상기 인코딩된 비디오 비트스트림을 파싱하는 단계;

상기 비디오 데이터의 인코딩된 블록이 상기 팔레트에 포함된 어느 엔트리에 의해서도 표현되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 나타내는 제 2 데이터를 획득하기 위해 상기 수신된 인코딩된 비디오 비트스트림을 파싱하는 단계;

상기 제 1 데이터로부터, 상기 팔레트에 포함된 엔트리들의 수가 1 과 동일하다고 결정하는 단계;

상기 제 2 데이터로부터, 상기 비디오 데이터의 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다고 결정하는 단계;

상기 팔레트에 포함된 상기 엔트리들의 수가 1 과 동일하고 상기 비디오 데이터의 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다고 결정하는 것에 기초하여, 상기 비디오 데이터의 인코딩된 블록의 픽셀 값들에 대한 인덱스 값을 디코딩을 바이패스하는 단계로서, 상기 인덱스 값을 상기 팔레트의 엔트리들과 연관되는, 상기 인덱스 값을 디코딩을 바이패스하는 단계;

상기 비디오 데이터의 인코딩된 블록의 픽셀 값을 대한 상기 인덱스 값을 디코딩함이 없이, 상기 비디오 데이터의 인코딩된 블록의 모든 픽셀 값을 상기 팔레트에 포함된 하나의 엔트리와 동일하다고 결정하는 단계; 및

상기 비디오 데이터의 인코딩된 블록의 픽셀 값을마다에, 상기 팔레트의 하나의 엔트리에 의해 나타내는 개별적인 값을 할당하는 것에 의해 상기 인코딩된 블록을 복원하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 인코딩된 비디오 비트스트림의 일부로서, 상기 인덱스 값을 상기 비디오 데이터의 블록에 대해 인코딩되는지를 나타내는 1-비트 플래그를 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 1 데이터는 상기 팔레트에서의 상기 엔트리들의 수가 1 과 동일한지를 나타내는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 2 데이터는 상기 비디오 데이터의 인코딩된 블록이 적어도 하나의 이스케이프 픽셀을 포함하는지를 나타내는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 인코딩된 비디오 비트스트림의 일부로서, 상기 팔레트와 연관된 하나 이상의 신택스 엘리먼트들을 수신하는 단계; 및

상기 팔레트와 연관된 상기 하나 이상의 신택스 엘리먼트들을 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

비디오 데이터를 인코딩하는 방법으로서,

팔레트-기반 코딩에 따라 인코딩될 비디오 데이터의 블록의 픽셀 값들을 표현하는데 사용된 팔레트에 포함된 엔트리들의 수를 결정하는 단계;

상기 비디오 데이터의 블록이 상기 팔레트에 포함된 어느 엔트리에 의해서도 표현되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 결정하는 단계;

상기 팔레트에 포함된 상기 엔트리들의 수가 1과 동일하고 상기 비디오 데이터의 블록이 이스케이프 픽셀을 포함하지 않는다고 결정하는 것에 기초하여, 상기 비디오 데이터의 블록의 모든 픽셀 값들이 상기 팔레트에 포함된 하나의 엔트리와 동일하다고 결정하는 단계;

상기 비디오 데이터의 블록의 상기 픽셀 값들에 대한 인덱스 값들의 인코딩을 바이패싱하는 단계로서, 상기 인덱스 값들은 상기 팔레트의 엔트리들과 연관되는, 상기 인덱스 값들의 인코딩을 바이패싱하는 단계;

비디오 데이터의 인코딩된 블록의 픽셀 값들의 픽셀 값마다에, 상기 팔레트의 하나의 엔트리에 의해 나타내는 개별적인 값을 할당하는 것에 의해 상기 비디오 데이터의 인코딩된 블록을 형성하도록 상기 비디오 데이터의 블록을 인코딩하는 단계; 및

인코딩된 비디오 비트스트림으로 상기 비디오 데이터의 인코딩된 블록을 시그널링하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 7

제 6 항에 있어서,

상기 인코딩된 비디오 비트스트림의 일부로서, 상기 인덱스 값들이 상기 비디오 데이터의 블록의 상기 픽셀 값들에 대해 인코딩되는지를 나타내는 1-비트 플래그를 시그널링하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 8

제 6 항에 있어서,

상기 인코딩된 비디오 비트스트림의 일부로서, 상기 팔레트에 포함된 엔트리들의 수가 1 과 동일하다고 나타내는 1-비트 플래그를 시그널링하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 9

제 6 항에 있어서,

상기 인코딩된 비디오 비트스트림의 일부로서, 상기 비디오 데이터의 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다고 나타내는 1-비트 플래그를 시그널링하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 10

제 6 항에 있어서,

상기 팔레트와 연관된 하나 이상의 신택스 엘리먼트들을 인코딩하는 단계; 및

상기 인코딩된 비디오 비트스트림의 일부로서, 상기 팔레트와 연관된 상기 하나 이상의 신택스 엘리먼트들을 시그널링하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 11

비디오 데이터를 디코딩하는 디바이스로서,

디코딩될 비디오 데이터의 인코딩된 블록을 표현하는 데이터를 포함하는 인코딩된 비디오 비트스트림을 저장하도록 구성된 메모리로서, 상기 인코딩된 블록은 팔레트-기반 코딩에 따라 인코딩되는, 상기 메모리; 및

상기 메모리와 통신하는 하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은:

디코딩될 상기 비디오 데이터의 인코딩된 블록의 픽셀 값들을 표현하는데 사용된 팔레트에 포함된 엔트리들의 수를 나타내는 제 1 데이터를 획득하기 위해 저장된 상기 인코딩된 비디오 비트스트림을 파싱하고;

상기 비디오 데이터의 인코딩된 블록이 상기 팔레트에 포함된 어느 엔트리에 의해서도 표현되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 나타내는 제 2 데이터를 획득하기 위해 상기 저장된 인코딩된 비디오 비트스트림을 파싱하고;

상기 제 1 데이터로부터, 상기 팔레트에 포함된 엔트리들의 수가 1 과 동일하다고 결정하고;

상기 제 2 데이터로부터, 상기 비디오 데이터의 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다고 결정하고;

상기 팔레트에 포함된 상기 엔트리들의 수가 1 과 동일하고 상기 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다고 결정하는 것에 기초하여, 상기 인코딩된 블록의 픽셀 값들에 대한 인덱스 값들의 디코딩을 바이패스하는 것으로서, 상기 인덱스 값들은 상기 팔레트의 엔트리들과 연관되는, 상기 인덱스 값들의 디코딩을 바이패스하고;

상기 인코딩된 블록의 픽셀 값들에 대한 상기 인덱스 값들을 디코딩함이 없이, 상기 인코딩된 블록의 모든 픽셀 값들이 상기 팔레트에 포함된 하나의 엔트리와 동일하다고 결정하고; 그리고

상기 인코딩된 블록의 픽셀 값들의 픽셀 값마다에, 상기 팔레트의 하나의 엔트리에 의해 나타내는 개별적인 값을 할당하는 것에 의해 상기 인코딩된 블록을 복원하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

청구항 12

제 11 항에 있어서,

상기 메모리에 저장된 상기 인코딩된 비디오 비트스트림은 상기 인덱스 값들이 상기 인코딩된 블록에 대해 인코딩되는지를 나타내는 1-비트 플래그를 포함하고, 상기 하나 이상의 프로세서들은 또한, 상기 인덱스 값들이 상기 인코딩된 블록에 대해 인코딩되는지를 결정하기 위해 상기 1-비트 플래그의 값을 결정하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

청구항 13

제 12 항에 있어서,

상기 인코딩된 비디오 비트스트림을 수신하도록 구성된 인터페이스 디바이스를 더 포함하는, 비디오 데이터를 디코딩하는 디바이스.

청구항 14

제 11 항에 있어서,

상기 제 1 데이터는 상기 팔레트의 사이즈가 1 과 동일한지를 나타내는 1-비트 플래그를 포함하는, 비디오 데이터를 디코딩하는 디바이스.

청구항 15

제 11 항에 있어서,

상기 제 2 데이터는 상기 인코딩된 블록이 적어도 하나의 이스케이프 픽셀을 포함하는지를 나타내는 1-비트 플

래그를 포함하는, 비디오 데이터를 디코딩하는 디바이스.

청구항 16

명령들로 인코딩된 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 경우, 인코딩된 비디오 데이터를 디코딩하는 디바이스의 하나 이상의 프로세서들로 하여금:

디코딩될 상기 인코딩된 비디오 데이터의 인코딩된 블록을 표현하는 데이터를 포함하는 인코딩된 비디오 비트스트림을 수신하게 하는 것으로서, 상기 인코딩된 블록은 팔레트-기반 코딩에 따라 인코딩되는, 상기 인코딩된 비디오 비트스트림을 수신하게 하고;

디코딩될 상기 비디오 데이터의 인코딩된 블록의 픽셀 값들을 표현하는데 사용된 팔레트에 포함된 엔트리들의 수를 나타내는 제 1 데이터를 획득하기 위해 수신된 상기 인코딩된 비디오 비트스트림을 파싱하게 하고;

상기 비디오 데이터의 인코딩된 블록이 상기 팔레트에 포함된 어느 엔트리에 의해서도 표현되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 나타내는 제 2 데이터를 획득하기 위해 상기 수신된 인코딩된 비디오 비트스트림을 파싱하게 하고;

상기 제 1 데이터로부터, 상기 팔레트에 포함된 엔트리들의 수가 1 과 동일하다고 결정하게 하고;

상기 제 2 데이터로부터, 상기 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다고 결정하게 하고;

상기 팔레트에 포함된 상기 엔트리들의 수가 1 과 동일하고 상기 비디오 데이터의 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다는 결정에 기초하여, 상기 인코딩된 블록의 픽셀 값들에 대한 인덱스 값들의 디코딩을 바이패스하게 하는 것으로서, 상기 인덱스 값들은 상기 팔레트의 엔트리들과 연관되는, 상기 인덱스 값들의 디코딩을 바이패스하게 하고;

상기 인코딩된 블록의 픽셀 값들에 대한 상기 인덱스 값들을 디코딩함이 없이, 상기 인코딩된 블록의 모든 픽셀 값들이 상기 팔레트에 포함된 하나의 엔트리와 동일하다고 결정하게 하고;

상기 비디오 데이터의 인코딩된 블록의 픽셀 값들의 픽셀 값마다에, 상기 팔레트의 하나의 엔트리에 의해 나타내는 개별적인 값을 할당하는 것에 의해 상기 인코딩된 블록을 복원하게 하는, 명령들로 인코딩된 비일시적 컴퓨터 판독가능 저장 매체.

청구항 17

제 16 항에 있어서,

실행될 경우, 하나 이상의 프로세서들로 하여금, 상기 인코딩된 비디오 비트스트림에 포함된 1-비트 플래그의 값을 결정하게 하는 명령들로 추가로 인코딩되고, 상기 1-비트 플래그의 값은 상기 인덱스 값들이 상기 인코딩된 블록에 대해 인코딩되는지를 나타내는, 명령들로 인코딩된 비일시적 컴퓨터 판독가능 저장 매체.

청구항 18

제 16 항에 있어서,

상기 제 1 데이터는 상기 팔레트의 엔트리들의 수가 1 과 동일한지를 나타내는 1-비트 플래그를 포함하는, 명령들로 인코딩된 비일시적 컴퓨터 판독가능 저장 매체.

청구항 19

인코딩된 비디오 데이터를 디코딩하는 장치로서,

디코딩될 상기 인코딩된 비디오 데이터의 인코딩된 블록을 표현하는 데이터를 포함하는 인코딩된 비디오 비트스트림을 수신하기 위한 수단으로서, 상기 인코딩된 블록은 팔레트-기반 코딩에 따라 인코딩되는, 상기 인코딩된 비디오 비트스트림을 수신하기 위한 수단;

디코딩될 상기 인코딩된 비디오 데이터의 인코딩된 블록의 픽셀 값들을 표현하는데 사용된 팔레트에 포함된 엔트리들의 수를 나타내는 제 1 데이터를 획득하기 위해 수신된 상기 인코딩된 비디오 비트스트림을 파싱하기 위한 수단;

상기 인코딩된 비디오 데이터의 인코딩된 블록이 상기 팔레트에 포함된 어느 엔트리에 의해서도 표현되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 나타내는 제 2 데이터를 획득하기 위해 상기 수신된 인코딩된 비디오 비트스트림을 파싱하기 위한 수단;

상기 제 1 데이터로부터, 상기 팔레트에 포함된 엔트리들의 수가 1 과 동일하다고 결정하기 위한 수단;

상기 제 2 데이터로부터, 상기 인코딩된 비디오 데이터의 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다고 결정하기 위한 수단;

상기 팔레트에 포함된 상기 엔트리들의 수가 1 과 동일하고 상기 인코딩된 비디오 데이터의 인코딩된 블록이 이스케이프 픽셀을 포함하지 않는다는 결정에 기초하여, 상기 비디오 데이터의 인코딩된 블록의 픽셀 값들에 대한 인덱스 값들의 디코딩을 바이패스하기 위한 수단으로서, 상기 인덱스 값들은 상기 팔레트의 엔트리들과 연관되는, 상기 인덱스 값들의 디코딩을 바이패스하기 위한 수단;

상기 비디오 데이터의 인코딩된 블록의 픽셀 값들에 대한 상기 인덱스 값들을 디코딩함이 없이, 상기 인코딩된 비디오 데이터의 인코딩된 블록의 모든 픽셀 값들이 상기 팔레트에 포함된 하나의 엔트리와 동일하다고 결정하기 위한 수단; 및

상기 인코딩된 비디오 데이터의 인코딩된 블록의 픽셀 값들의 픽셀 값마다에, 상기 팔레트의 하나의 엔트리에 의해 나타내는 개별적인 값을 할당하는 것에 의해 상기 인코딩된 비디오 데이터의 상기 인코딩된 블록을 복원하기 위한 수단을 포함하는, 인코딩된 비디오 데이터를 디코딩하는 장치.

청구항 20

제 11 항에 있어서,

하나 이상의 집적 회로들; 하나 이상의 마이크로프로세서들; 하나 이상의 디지털 신호 프로세서들 (DSP들); 하나 이상의 필드 프로그래밍가능 게이트 어레이들 (FPGAs); 데스크톱 컴퓨터; 랩톱 컴퓨터; 태블릿 컴퓨터; 폰; 텔레비전; 카메라; 디스플레이 디바이스; 디지털 미디어 플레이어; 비디오 게임 콘솔; 비디오 게임 디바이스; 비디오 스트리밍 디바이스; 또는 무선 통신 디바이스 중 적어도 하나를 더 포함하는, 비디오 데이터를 디코딩하는 디바이스.

발명의 설명

기술 분야

[0001] 본 출원은 2014년 1월 2일자로 출원된 미국 가출원 제61/923,163호, 2014년 1월 6일자로 출원된 미국 가출원 제61/924,141호; 및 2014년 3월 24일자로 출원된 미국 가출원 제61/969,759호의 이익을 주장하고, 이 가출원 각각의 전체 내용은 본 명세서에 참조로 통합된다.

[0002] 본 개시는 비디오 인코딩 및 디코딩에 관한 것이다.

배경 기술

[0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인용 디지털 보조기들 (PDA들), 랩탑 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰들", 비디오 텔레컨퍼런싱 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC)에 의해 정의된 표준들, 현재 개발 중인 고 효율 비디오 코딩 (HEVC) 표준, 및 그러한 표준들의 확장들에서 설명된 기술들과 같은 비디오 압축 기술들을 구현한다. 비디오 디바이스들은 그러한 비디오 압축 기술들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 압축 기술들은 비디오 시퀀스들에 고유한 리던던시를 감소 또는 제거하기 위해 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 수행한다. 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 동일 픽처의 이웃 블록들에서의 레퍼런스 샘플들에 대한 공간 예측을 이용하여

인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일 픽처의 이웃 블록들에서의 레퍼런스 샘플들에 대한 공간 예측, 또는 다른 레퍼런스 픽처들에서의 레퍼런스 샘플들에 대한 시간 예측을 이용할 수도 있다. 픽처들은 프레임들로서 지칭될 수도 있으며, 레퍼런스 픽처들은 레퍼런스 프레임들로서 지칭될 수도 있다.

[0005] 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 발생시킨다. 잔차 데이터는 코딩될 오리지널 블록과 예측 블록 간의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 레퍼런스 샘플들의 블록을 포인팅하는 모션 벡터, 및 코딩된 블록과 예측 블록 간의 차이를 나타내는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가적인 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 계수들을 발생시킬 수도 있으며, 그 후, 이 잔차 계수들은 양자화될 수도 있다. 2차원 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] 멀티뷰 코딩 비트스트림이 예를 들어, 다중의 관점들로부터 뷰들을 인코딩함으로써 생성될 수도 있다. 멀티뷰 코딩 양태들을 사용하는 일부 3차원 (3D) 비디오 표준들이 개발되었다. 예를 들어, 상이한 뷰들이 3D 비디오를 지원하기 위해 좌우안 뷰들을 송신할 수도 있다. 대안으로, 일부 3D 비디오 코딩 프로세스들이 소위 멀티뷰 플러스 깊이 코딩을 적용할 수도 있다. 멀티뷰 플러스 깊이 코딩에서, 3D 비디오 비트스트림은 텍스처 뷰 컴포넌트들 뿐만 아니라 깊이 뷰 컴포넌트들을 포함할 수도 있다. 예를 들어, 각각의 뷰는 하나의 텍스처 뷰 컴포넌트 및 하나의 깊이 뷰 컴포넌트를 포함할 수 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0007] 일반적으로, 본 개시의 기술들은 팔레트-기반 비디오 코딩에 관한 것이다. 팔레트-기반 코딩에서, 비디오 코더 (예를 들어, 비디오 인코더 또는 비디오 디코더)는 특정한 영역 (예를 들어, 주어진 블록)의 비디오 데이터를 나타내는 컬러들 또는 픽셀 값들의 테이블로서 소위 "팔레트"를 형성할 수도 있다. 이러한 방식으로, 비디오 데이터의 현재 블록에 대한 실제 픽셀 값들 또는 그들의 잔차들을 코딩하기 보다는, 비디오 코더는 현재 블록의 픽셀 값들 중 하나 이상에 대한 컬러 또는 팔레트 인덱스를 코딩할 수도 있고, 여기서, 인덱스 값들은 현재 블록의 픽셀 값들을 나타내기 위해 사용되는 팔레트에서의 엔트리들을 나타낸다. 비디오 데이터의 현재 블록에 대한 팔레트 인덱스 값들의 맵이 주어진 스캔 순서 및 런-렝쓰 코딩 기술들을 사용하여 라인 마다 코딩될 수도 있다. 맵의 주어진 라인에서의 인덱스 값들 각각은 명시적으로 코딩될 수도 있고, 주어진 라인의 좌측-모드 (left-mode) 인덱스로부터 예측될 수도 있거나, 주어진 라인 위의 라인에서의 병치된 인덱스로부터 예측될 수도 있다.

[0008] 본 개시의 다양한 기술들은 기존의 팔레트-기반 코딩 기술들을 강화시키는 것에 관한 것이다. 일부 양태들에서, 본 개시는 블록이 특정한 기준을 충족시키는 경우에, 블록에 대한 팔레트 인덱스 값들의 맵의 코딩을 바이패싱하는 기술들에 관한 것이다. 일부 양태들에서, 본 개시는 양자화 파라미터 값들과 팔레트에러 제한 들 사이의 관계를 저장하는 매핑 테이블을 사용하여 주어진 팔레트에 대한 값들의 최대 범위(본 명세서에서 "에러 제한"으로서 또한 지칭함)를 결정하는 것에 관한 것이다. 일부 양태들에서, 본 개시는 대응하는 컬러 채널에서 종래의 계수 코딩에 사용된 양자화 파라미터들에 기초한 대응하는 팔레트에서의 엔트리에 매핑하지 않는 팔레트-코딩된 블록의 픽셀들(본 명세서에서 "이스케이프 픽셀들(escape pixels)"로서 또한 지칭함)에 대한 양자화 파라미터를 정의하는 것에 관한 것이다.

[0009] 일례에서, 본 개시는 비디오 데이터를 디코딩하는 방법에 관한 것이고, 그 방법은 디코딩될 비디오 데이터의 블록의 픽셀 값들을 표현하기 위해 사용된 팔레트에 포함된 엔트리들의 수를 결정하는 단계, 및 비디오 데이터의 블록이 팔레트에서의 엔트리들 중 어느 것과도 연관되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 결정하는 단계를 포함한다. 그 방법은, 팔레트에 포함된 엔트리들의 수가 1과 동일하고 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하지 않는다는 결정에 응답하여, 비디오 데이터의 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들의 디코딩을 바이패싱하는 단계, 및 비디오 데이터의 블록의 픽셀 값들을

팔레트에 포함된 하나의 엔트리와 동일한 것으로 결정하는 단계를 더 포함할 수도 있다.

[0010] 다른 예에서, 본 개시는 비디오 데이터를 인코딩하는 방법에 관한 것이고, 그 방법은 인코딩될 비디오 데이터의 블록의 픽셀 값들을 표현하기 위해 사용된 팔레트에 포함된 엔트리들의 수를 결정하는 단계, 및 비디오 데이터의 블록이 팔레트에서의 엔트리들 중 어느 것과도 연관되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 결정하는 단계를 포함한다. 그 방법은, 팔레트에 포함된 엔트리들의 수가 1과 동일하고 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하지 않는다는 결정에 응답하여, 비디오 데이터의 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들의 인코딩을 바이패싱하는 단계, 및 비디오 데이터의 블록의 픽셀 값들을 팔레트에 포함된 하나의 엔트리와 동일한 것으로 결정함으로써 비디오 데이터의 블록을 인코딩하는 단계를 더 포함할 수도 있다.

[0011] 다른 예에서, 본 개시는 비디오 데이터를 디코딩하는 장치에 관한 것이고, 그 장치는 비디오 데이터를 저장하도록 구성된 메모리, 및 코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트에 포함된 엔트리들의 수를 결정하며, 비디오 데이터의 블록이 팔레트에서의 엔트리들 중 어느 것과도 연관되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 결정하도록 구성된 하나 이상의 프로세서들을 포함한다. 하나 이상의 프로세서들은, 팔레트에 포함된 엔트리들의 수가 1과 동일하고 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하지 않는다는 결정에 응답하여, 비디오 데이터의 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들의 코딩을 바이패싱하며, 비디오 데이터의 블록의 픽셀 값들을 팔레트에 포함된 하나의 엔트리와 동일한 것으로 결정하도록 더 구성될 수 있다.

[0012] 다른 예에서, 본 개시는 비디오 데이터를 코딩하는 장치에 관한 것이고, 그 장치는 코딩될 비디오 데이터의 블록의 픽셀 값들을 표현하기 위해 사용된 팔레트에 포함된 엔트리들의 수를 결정하는 수단, 및 비디오 데이터의 블록이 팔레트에서의 엔트리들 중 어느 것과도 연관되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 결정하는 수단을 포함한다. 그 장치는, 팔레트에 포함된 엔트리들의 수가 1과 동일하고 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하지 않는다는 결정에 응답하여, 비디오 데이터의 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들의 코딩을 바이패싱하는 수단, 및 비디오 데이터의 블록의 픽셀 값들을 팔레트에 포함된 하나의 엔트리와 동일한 것으로 결정하는 수단을 더 포함할 수도 있다.

[0013] 다른 예에서, 본 개시는 명령들로 인코딩된 비일시적 컴퓨터 판독가능 매체에 관한 것이고, 그 명령들은, 실행되는 경우에, 비디오 데이터를 코딩하는 디바이스의 하나 이상의 프로세서들로 하여금, 코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트에 포함된 엔트리들의 수를 결정하게 하며, 비디오 데이터의 블록이 팔레트에서의 엔트리들 중 어느 것과도 연관되지 않는 적어도 하나의 이스케이프 픽셀을 포함하는지를 결정하게 한다. 그 명령들은, 실행되는 경우에, 하나 이상의 프로세서들로 하여금, 팔레트에 포함된 엔트리들의 수가 1과 동일하고 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하지 않는다는 결정에 응답하여, 비디오 데이터의 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들의 코딩을 더 바이패싱하게 하며, 비디오 데이터의 블록의 픽셀 값들을 팔레트에 포함된 하나의 엔트리와 동일한 것으로 더 결정하게 할 수도 있다.

[0014] 일례에서, 본 개시는 비디오 데이터를 디코딩하는 방법에 관한 것이고, 그 방법은 디코딩될 비디오 데이터의 블록의 픽셀 값들을 표현하기 위해 사용된 팔레트를 결정하는 단계, 및 비디오 데이터의 블록에서, 팔레트에서의 하나 이상의 엔트리들 중 어느 것과도 연관되지 않는 하나 이상의 이스케이프 픽셀들을 식별하는 단계를 포함한다. 그 방법은 넌-팔레트 기반 코딩에서 변환 계수 코딩을 위해 사용된 양자화 파라미터 (QP) 값에 기초한 주어진 컬러 채널에 대한 블록의 하나 이상의 이스케이프 픽셀들 모두에 대한 단일 QP 값을 식별하는 단계, 및 식별된 단일 QP 값을 사용하여 하나 이상의 이스케이프 픽셀들 각각을 탈양자화 (dequantize) 하는 단계를 더 포함할 수도 있다. 그 방법은 탈양자화된 이스케이프 픽셀들 및 팔레트에서의 하나 이상의 엔트리들과 연관되는 비디오 데이터의 블록에서의 하나 이상의 픽셀들에 대해 수신된 인덱스 값들에 기초하여 비디오 데이터의 블록의 픽셀 값들을 결정하는 단계를 더 포함할 수도 있다.

[0015] 다른 예에서, 본 개시는 비디오 데이터를 인코딩하는 방법에 관한 것이고, 그 방법은 인코딩될 비디오 데이터의 블록의 픽셀 값들을 표현하기 위해 사용된 팔레트를 결정하는 단계, 및 비디오 데이터의 블록에서, 팔레트에서의 하나 이상의 엔트리들 중 어느 것과도 연관되지 않는 하나 이상의 이스케이프 픽셀들을 식별하는 단계를 포함한다. 그 방법은 넌-팔레트 기반 코딩에서 변환 계수 코딩을 위해 사용된 양자화 파라미터 (QP) 값에 기초한 주어진 컬러 채널에 대한 블록의 하나 이상의 이스케이프 픽셀들 모두에 대한 단일 QP 값을 식별하는 단계, 및 식별된 단일 QP 값을 사용하여 하나 이상의 이스케이프 픽셀들 각각을 양자화하는 단계를 더 포함할

수도 있다. 그 방법은 양자화된 이스케이프 픽셀들 및 팔레트에서의 하나 이상의 엔트리들과 연관되는 비디오 데이터의 블록에서의 하나 이상의 픽셀들에 대한 인덱스 값들을 포함하는 비디오 데이터의 블록의 픽셀 값들을 인코딩하는 단계를 더 포함할 수도 있다.

[0016] 다른 예에서, 본 개시는 비디오 데이터를 코딩하는 장치에 관한 것이고, 그 장치는 비디오 데이터를 저장하도록 구성된 메모리, 및 메모리와 통신하고, 코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트를 결정하도록 구성되며, 비디오 데이터의 블록에서, 팔레트에서의 하나 이상의 엔트리들 중 어느 것과도 연관되지 않는 하나 이상의 이스케이프 픽셀들을 식별하도록 구성된 하나 이상의 프로세서들을 포함한다. 하나 이상의 프로세서들은 년-팔레트 기반 코딩에서 변환 계수 코딩을 위해 사용된 양자화 파라미터 (QP) 값에 기초한 주어진 컬러 채널에 대한 블록의 하나 이상의 이스케이프 픽셀들 모두에 대한 단일 QP 값을 식별하며, 식별된 단일 QP 값을 하나 이상의 이스케이프 픽셀들 각각에 적용하도록 더 구성될 수도 있다. 하나 이상의 프로세서들은 이스케이프 픽셀들 및 하나 이상의 엔트리들과 연관되는 비디오 데이터의 블록에서의 하나 이상의 픽셀들에 대해 수신된 인덱스 값들에 기초하여 비디오 데이터의 블록의 픽셀 값들을 결정하도록 더 구성될 수 있다.

[0017] 다른 예에서, 본 개시는 비디오 데이터를 코딩하는 장치에 관한 것이고, 그 장치는 코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트를 결정하는 수단, 비디오 데이터의 블록에서, 팔레트에서의 하나 이상의 엔트리들 중 어느 것과도 연관되지 않는 하나 이상의 이스케이프 픽셀들을 식별하는 수단, 년 팔레트 기반 코딩에서 변환 계수 코딩을 위해 사용된 양자화 파라미터 (QP) 값에 기초한 주어진 컬러 채널에 대한 블록의 하나 이상의 이스케이프 픽셀들 모두에 대한 단일 QP 값을 식별하는 수단, 식별된 단일 QP 값을 하나 이상의 이스케이프 픽셀들 각각에 적용하는 수단, 및 이스케이프 픽셀들 및 하나 이상의 엔트리들과 연관된 비디오 데이터의 블록에서의 하나 이상의 픽셀들에 대해 수신된 인덱스 값들에 기초하여 비디오 데이터의 블록의 픽셀 값들을 결정하는 수단을 포함한다.

[0018] 다른 예에서, 본 개시는 명령들로 인코딩된 비일시적 컴퓨터 판독가능 매체에 관한 것이고, 그 명령들은, 실행되는 경우에, 비디오 데이터를 코딩하는 디바이스의 하나 이상의 프로세서들로 하여금, 코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트를 결정하게 하고, 비디오 데이터의 블록에서, 팔레트에서의 하나 이상의 엔트리들 중 어느 것과도 연관되지 않는 하나 이상의 이스케이프 픽셀들을 식별하게 하며, 년-팔레트 기반 코딩에서 변환 계수 코딩을 위해 사용된 양자화 파라미터 (QP) 값에 기초하여 주어진 컬러 채널에 대한 블록의 하나 이상의 이스케이프 픽셀들 모두에 대한 단일 QP 값을 식별하게 한다. 그 명령들은, 실행되는 경우에, 하나 이상의 프로세서들로 하여금, 식별된 단일 QP 값을 하나 이상의 이스케이프 픽셀들 각각에 더 적용하게 할 수도 있으며, 이스케이프 픽셀들 및 하나 이상의 엔트리들과 연관되는 비디오 데이터의 블록에서의 하나 이상의 픽셀들에 대해 수신된 인덱스 값들에 기초하여 비디오 데이터의 블록의 픽셀 값들을 더 결정하게 할 수도 있다.

[0019] 본 명세서에 설명한 기술들은 기존의 팔레트-기반 코딩 기술들 및/또는 데이터 압축 기술들 이상의 하나 이상의 잠재적 이점들 및 개선점들을 제공할 수도 있다. 예를 들어, 본 개시의 다양한 기술들은 데이터 정밀도를 유지하면서 컴퓨팅 자원들 및 대역폭 요건들을 유지하도록 비디오 코딩 디바이스들에 의해 구현될 수도 있다.

이에 따라, 본 개시의 다양한 기술들은 기존의 팔레트-기반 코딩 기술들 및 데이터 압축 기술들의 효율 및 정확성을 향상시키도록 비디오 코딩 디바이스들에 의해 구현될 수도 있다.

[0020] 본 개시의 하나 이상의 예들의 상세들은 첨부한 도면들 및 하기의 설명에 기재된다. 다른 특징들, 목적들, 및 이점들은 설명, 도면들, 및 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0021] 도 1은 본 개시에 설명한 기술들을 활용할 수도 있는 예시적인 비디오 코딩 시스템을 예시하는 블록도이다.

도 2는 본 개시에 설명한 기술들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 3은 본 개시에 설명한 기술들을 구현할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 4는 비디오 디코딩 디바이스가 조건들의 특정한 세트에 기초하여, 팔레트-코딩된 블록의 픽셀들에 대한 인덱스 값들의 디코딩을 바이패스하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스를 예시하는 플로우차트이다.

도 5는 비디오 인코딩 디바이스가 조건들의 특정한 세트에 기초하여, 팔레트-코딩된 블록의 픽셀들에 대한 인덱

스 값들의 인코딩을 바이패스하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스를 예시하는 플로우차트이다.

도 6은 비디오 디코딩 디바이스가 비디오 데이터의 팔레트-코딩된 블록의 하나 이상의 이스케이프 픽셀들을 탈양자화하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스를 예시하는 플로우차트이다.

도 7은 비디오 인코딩 디바이스가 비디오 데이터의 팔레트-코딩된 블록의 하나 이상의 이스케이프 픽셀들을 양자화하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스를 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0022]

본 개시는 비디오 코딩 및 압축을 위한 기술들을 포함한다. 특히, 본 개시는 비디오 데이터의 팔레트-기반 코딩을 위한 기술들을 설명한다. 종래의 비디오 코딩에서는, 이미지들은 연속-톤이고 공간적으로 평활한 것으로 가정된다. 이들 가정들에 기초하여, 블록-기반 변환, 필터링 등과 같은 다양한 툴들이 개발되었고, 이러한 툴들은 자연적인 콘텐츠 비디오들에 대해 양호한 성능을 나타내었다.

[0023]

그러나, 원격 데스크탑, 공동 작업 및 무선 디스플레이와 같은 애플리케이션들에서, (예를 들어, 텍스트 또는 컴퓨터 그래픽과 같은) 컴퓨터 생성된 스크린 콘텐츠는 압축되기에에는 우세한 콘텐츠일 수도 있다. 이러한 타입의 콘텐츠는 개별-톤을 갖고, 선명한 라인들 및 높은-콘트라스트 오브젝트 경계들을 특징으로 하는 경향이 있다. 연속-톤 및 평활함의 가정은 스크린 콘텐츠에 더 이상 적용되지 않을 수도 있고, 따라서, 종래의 비디오 코딩 기술들은 스크린 콘텐츠를 포함하는 비디오 데이터를 압축하기 위해 효율적인 방식이 아닐 수도 있다.

[0024]

본 개시는 스크린 생성된 콘텐츠 코딩에 특히 적합할 수도 있는 팔레트-기반 코딩을 설명한다. 예를 들어, 비디오 데이터의 특정한 영역이 상대적으로 작은 수의 컬러들을 갖는다는 것을 가정하면, 비디오 코더 (예를 들어, 비디오 인코더 또는 비디오 디코더)는 특정한 영역의 비디오 데이터를 표현하기 위해 소위 "팔레트"를 형성할 수도 있다. 팔레트는 특정한 영역 (예를 들어, 주어진 블록)의 비디오 데이터를 표현하는 컬러들 또는 픽셀 값들의 테이블로서 표현될 수도 있다. 예를 들어, 팔레트는 주어진 블록에서 가장 우세한 픽셀 값들을 포함할 수도 있다. 일부 경우들에서, 가장 우세한 픽셀 값들은 블록내에서 가장 빈번하게 발생하는 하나 이상의 픽셀 값들을 포함할 수도 있다. 추가로, 일부 경우들에서, 비디오 코더는 픽셀 값이 블록에서의 가장 우세한 픽셀 값들 중 하나로서 포함될지를 결정하기 위해 임계값을 적용할 수도 있다. 팔레트-기반 코딩의 다양한 양태들에 따르면, 비디오 코더는 비디오 데이터의 현재 블록에 대한 실제 픽셀 값들 또는 그 잔차들을 코딩하는 대신에, 현재 블록의 픽셀 값들 중 하나 이상을 나타내는 인덱스 값들을 코딩할 수도 있다. 팔레트-기반 코딩의 문맥에서, 인덱스 값들은 현재 블록의 개별 픽셀 값들을 표현하기 위해 사용되는 팔레트에서의 각각의 엔트리들을 나타낸다.

[0025]

예를 들어, 비디오 인코더는 블록에 대한 팔레트를 결정하고 (예를 들어, 팔레트를 명시적으로 코딩하고, 팔레트를 예측하거나, 이들의 조합을 행하고), 픽셀 값들 중 하나 이상을 표현하기 위해 팔레트에서의 엔트리를 로케이팅하며, 블록의 픽셀 값들을 표현하기 위해 사용된 팔레트에서의 엔트리를 나타내는 인덱스 값들로 블록을 인코딩함으로써 비디오 데이터의 블록을 인코딩할 수도 있다. 일부 예들에서, 비디오 인코더는 팔레트 및/ 또는 인덱스 값들을 인코딩된 비트스트림에서 시그널링할 수도 있다. 결과적으로, 비디오 디코더는 인코딩된 비트스트림으로부터, 블록의 개별 픽셀들에 대한 인덱스 값들 뿐만 아니라 블록에 대한 팔레트를 획득할 수도 있다. 비디오 디코더는 픽셀들의 인덱스 값들을 팔레트의 엔트리들에 연관시켜 블록의 다양한 픽셀 값을 복원할 수도 있다.

[0026]

비디오 데이터의 팔레트-기반 코딩을 상세히 상술하였다. 팔레트-기반 코딩의 기본 아이디어는, 각각의 CU에 대해, 현재 CU에서 가장 우세한 픽셀 값을 포함하는 (그리고 가장 우세한 픽셀 값들로 이루어질 수도 있는) 팔레트가 유도된다는 것이다. 팔레트의 사이즈 및 엘리먼트들이 비디오 인코더로부터 비디오 디코더로 먼저 송신된다. 그 후, CU에서의 픽셀 값들이 특정한 스캐닝 순서에 따라 인코딩된다. CU에서의 각각의 픽셀 위치에 대해, 플래그, 예를 들어, palette_flag가 픽셀 값이 팔레트에 포함되거나 (즉, "런 모드 (run mode)") 또는 포함되지 않는지 (즉, "픽셀 모드")를 나타내기 위해 먼저 송신된다. "런 모드"에서, CU에서의 픽셀 위치와 연관된 팔레트 인덱스가 픽셀 값의 "런"에 후속하여 시그널링된다. palette_flag 및 팔레트 인덱스 어느 것도, 이들이 모두 동일한 픽셀 값을 갖기 때문에 "런"에 의해 커버되는 후속하는 픽셀 위치들에 대해 송신될 필요가 없다. "픽셀 모드"에서, 픽셀 값은 CU에서의 주어진 픽셀 위치에 대해 송신된다.

- [0027] 각각의 CU에 대해, 현재 CU에서 가장 우세한 픽셀 값들로 이루어지는 주요 컬러 테이블이 유도된다. 테이블의 사이즈 및 엘리먼트들이 먼저 송신된다. 주요 컬러 테이블의 사이즈 및/또는 엘리먼트들은 이웃하는 CU들 (예를 들어, 위 및/또는 좌측의 코딩된 CU)에서의 주요 컬러 테이블의 사이즈 및/또는 엘리먼트들을 사용하여 직접 코딩될 수 있거나 예측적으로 코딩될 수 있다.
- [0028] 일부 예들에서, 현재 CU에서의 픽셀들 각각은 하나의 주요 컬러 테이블 인덱스에 매핑된다. 주요 컬러 인덱스들이 존재하지 않는 이들 픽셀들에 대해, 특수 인덱스 ('다른 인덱스'로 칭함) 가 이들 픽셀들에 할당되고, 이들 픽셀들은 '이스케이프 픽셀'로 불린다. 본 개시의 기술들은 주요 컬러 인덱스들의 코딩 방법에 중점을 둔다.
- [0029] '이스케이프 픽셀'은 고정 길이 코딩, 단항 (unary) 코딩 등과 같은 임의의 기준 엔트로피 코딩 방법을 사용하여 코딩될 수 있다. 이스케이프 값들을 인코딩하기 위한 방법은 양자화 파라미터 (QP) 값에 따라 왼쪽-시프트 연산을 사용하는 것이다. 즉, QP 값에 따라 비트들의 수인, 최상위 비트들만을 인코딩한다. 이러한 목적을 위해, 최신 기술에서 사용된 전략은 픽셀 값에 적용될 오른쪽-시프트인 수에 각각의 QP를 매핑하는 테이블을 사용하는 것이다.
- [0030] 주요 컬러 인덱스의 블록은 라인마다 코딩된다. 각각의 라인에 대해, 라인 모드가 '수평', '수직', 및 '노멀(normal)'로부터 선택된다. '수평' 라인 모드가 선택되는 경우에, 현재 라인에서의 인덱스들 모두는 위 라인에서의 가장 왼쪽 인덱스의 가장 왼쪽 인덱스와 동일하다. '수직' 라인 모드가 선택되는 경우에, 전체 라인은 위에 있는 하나의 라인으로부터 카피된다. '노멀' 모드가 선택되는 경우에, 현재 라인내의 인덱스들은 하나씩 코딩된다. 이러한 경우에서 각각의 인덱스에 대해, 인덱스가 위 라인에서의 병치된 인덱스로부터 카피될 수 있는지 ('copy_from_top'), 또는 인덱스의 좌측 이웃 인덱스로부터 카피될 수 있는지 ('copy_from_left'), 또는 가능한 카피가 없는지 ('no_copy')를 나타내기 위해 선택스 엘리먼트가 사용된다. 가능한 카피가 없는 경우에, 픽셀은 직접 코딩된다.
- [0031] 상기 예들은 팔레트-기반 코딩의 일반적인 설명을 제공하도록 의도된다. 다양한 예들에서, 본 개시에서 설명하는 기술들은 팔레트-기반 코딩 모드들에 의해 형성된 인코딩된 비디오 데이터를 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 예측하는 것, 팔레트들을 유도하는 것, 또는 팔레트-기반 코딩 맵들 및 다른 선택스 엘리먼트들로부터 비디오 데이터를 디코딩하는 것 중 하나 이상의 다양한 조합들에 대한 기술들을 포함할 수도 있다. 이러한 기술들은 비디오 코딩 효율을 향상시킬 수도 있고, 예를 들어, 스크린 생성된 콘텐츠를 표현하기 위해 더 적은 비트들을 요구한다.
- [0032] 본 개시는 팔레트-기반 비디오 코딩에 관한 다양한 기술들을 설명한다. 일부 양태들에서, 본 개시는 비디오 블록과 연관된 팔레트가 하나의 엔트리 또는 컬러만을 포함하고, 비디오 블록이 어떠한 이스케이프 픽셀들도 포함하지 않을 때 비디오 블록에 대한 인덱스 값들의 맵을 코딩하는 것을 바이패싱하는 것에 관한 것이다. 일부 양태들에서, 본 개시는 양자화 파라미터 값들과 팔레트 에러 제한들 사이의 관계를 저장하는 매핑 테이블을 사용하여, 대응하는 팔레트 내에 포함될 수도 있는 고정된 픽셀 값으로부터의 최대 편차를 나타내는 팔레트 "에러 제한"을 유도하는 것에 관한 것이다. 본 개시의 일부 양태들은 대응하는 컬러 채널에서 종래의 계수 코딩에 사용된 양자화 파라미터들을 사용하여, 이스케이프 픽셀 (또는 연관된 예측 에러)의 양자화 및 탈양자화를 위한 양자화 파라미터 (QP)를 유도하는 것에 관한 것이다. 추가로, 본 개시는 팔레트-기반 코딩된 영역에 포함되는 픽셀이 이스케이프 픽셀인지를 나타내기 위해 플래그의 사용을 설명한다. 본 개시의 양태들은 오른쪽-시프트 연산을 이용하는 기술들과 같은, 이스케이프 픽셀 값을 양자화하기 위한 기술들을 또한 설명한다.
- [0033] 일부 양태들에서, 본 개시는 이스케이프 픽셀에 대한 양자화 파라미터 값에 기초하여 매핑 함수 (예를 들어, 시프트 함수)에 따라 팔레트-코딩된 블록의 이스케이프 픽셀들을 양자화하는 것에 관한 것이다. 일부 양태들에서, 본 개시는 팔레트-코딩된 블록이 이스케이프 픽셀인지를 나타내고 그리고/또는 결정하기 위해 플래그를 사용하는 것에 관한 것이다.
- [0034] 본 개시의 일부 예들에서, 비디오 데이터의 팔레트-기반 코딩을 위한 기술들은 비디오 데이터의 인터-예측 코딩 또는 인트라-예측 코딩을 위한 기술들과 같은 하나 이상의 다른 코딩 기술들과 사용될 수도 있다. 예를 들어, 더욱 상세히 후술하는 바와 같이, 인코더 또는 디코더, 또는 조합된 인코더-디코더 (코덱) 가 팔레트-기반 코딩 뿐만 아니라 인터- 및 인트라-예측 코딩을 수행하도록 구성될 수도 있다. 일부 예들에서, 팔레트-기반 코딩 기술들은 고효율 비디오 코딩 (HEVC) 의 하나 이상의 코딩 모드들에서 사용을 위해 구성될 수도 있다. 다른 예들에서, 팔레트-기반 코딩 기술들은 다른 기준의 또는 장래의 시스템들 또는 표준과 독립적으로 또는

그 일부로서 사용될 수 있다.

[0035] 고효율 비디오 코딩 (HEVC) 은 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 및 ISO/IEC 모션 광학 전문가 그룹 (MPEG) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 개발되었다. "WD10"의 "HEVC 드래프트 10"으로서 지칭되는 HEVC 표준의 최근의 드래프트는 http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip로부터 입수 가능한, Bross 등의 문헌 JCTVC-L1003v34 "High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Last Call)," ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 12차 회의: 제네바, CH, 14-23 January 2013, 에 설명된다. 최종 HEVC 표준 문헌은 "ITU-T H.265, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video - High efficiency video coding," ITU (International Telecommunication Union) 의 원격통신 표준화 섹터, April 2013, 로서 공개되어 있다.

[0036] HEVC 프레임워크에 관하여, 일례로서, 팔레트-기반 코딩 기술들은 코딩 유닛 (CU) 모드로서 사용되도록 구성될 수도 있다. 다른 예들에서, 팔레트-기반 코딩 기술들은 HEVC의 프레임워크에서 예측 유닛 (PU) 모드로서 사용되도록 구성될 수도 있다. 이에 따라, CU 모드의 문맥에서 설명된 아래의 개시된 프로세스들 모두가 추가적으로 또는 대안적으로 PU에 적용될 수도 있다. 그러나, 이들 HEVC-기반 예들은, 이러한 기술들이 다른 기준의 또는 아직 개발될 시스템들/표준들과 독립적으로 또는 그 일부로서 작용하도록 적용될 수도 있기 때문에, 본 명세서에 설명하는 팔레트-기반 코딩 기술들의 제약 또는 제한으로 고려되어서는 안 된다. 이들 경우들에서, 팔레트 코딩을 위한 단위는 정사각형 블록들, 직사각형 블록들 또는 심지어 비직사각형 형상의 영역들일 수 있다.

[0037] 도 1은 본 개시의 기술들을 활용할 수도 있는 예시적인 비디오 코딩 시스템(10)을 예시하는 블록도이다. 본 명세서에 사용되는 바와 같이, 용어 "비디오 코더"는 비디오 인코더들 및 비디오 디코더들 모두를 일반적으로 지칭한다. 본 개시에서, 용어를 "비디오 코딩" 또는 "코딩"은 비디오 인코딩 또는 비디오 디코딩을 일반적으로 지칭할 수도 있다. 비디오 코딩 시스템 (10)의 비디오 인코더 (20) 및 비디오 디코더 (30)는 본 개시에 설명한 다양한 예들에 따른 팔레트-기반 비디오 코딩을 위한 기술들을 수행하도록 구성될 수도 있는 디바이스들의 예들을 나타낸다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트-기반 코딩 또는 넌-팔레트 기반 코딩을 사용하여, HEVC 코딩에서 CU들 또는 PU들과 같은, 비디오 데이터의 다양한 블록들을 선택적으로 코딩하도록 구성될 수도 있다. 넌-팔레트 기반 코딩 모드들은 HEVC 드래프트 10에 의해 특정된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간 코딩 모드들 또는 인트라-예측 공간 코딩 모드들을 지칭할 수도 있다.

[0038] 도 1에 도시되어 있는 바와 같이, 비디오 코딩 시스템 (10)은 소스 디바이스 (12) 및 목적지 디바이스 (14)를 포함한다. 소스 디바이스 (12)는 인코딩된 비디오 데이터를 생성한다. 추가로, 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.

[0039] 소스 디바이스 (12) 및 목적지 디바이스 (14)는 데스크탑 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 자동차용 컴퓨터들 등을 포함한 광범위한 디바이스들을 포함할 수도 있다.

[0040] 목적지 디바이스 (14)는 채널 (16)을 통해 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신할 수도 있다. 채널 (16)은 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 이동시킬 수 있는 하나 이상의 미디어 또는 디바이스들을 포함할 수도 있다. 일례에서, 채널 (16)은 소스 디바이스 (12)가 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14)로 실시간으로 송신할 수 있게 하는 하나 이상의 통신 매체를 포함할 수도 있다. 이러한 예에서, 소스 디바이스 (12)는 무선 통신 프로토콜과 같은 통신 표준에 따라 인코딩된 비디오 데이터를 변조할 수도 있고, 변조된 비디오 데이터를 목적지 디바이스 (14)에 송신할 수도 있다. 하나 이상의 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신라인들과 같은 무선 및/또는 유선 통신 매체를 포함할 수도 있다. 하나 이상의 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예를 들어, 인터넷) 와 같은 패킷-기반 네트워크의 일부를 형성할

수도 있다. 하나 이상의 통신 매체는 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 통신을 용이하게 하는 라우터들, 스위치들, 기지국들, 또는 다른 장비를 포함할 수도 있다.

[0041] 다른 예에서, 채널 (16)은 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이러한 예에서, 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 저장하는 다른 적합한 디지털 저장 매체와 같은 각종의 국부적으로 액세스된 데이터 저장 매체를 포함할 수도 있다.

[0042] 추가의 예에서, 채널 (16)은 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 이러한 예에서, 목적지 디바이스 (14)는 스트리밍 또는 다운로딩을 통해 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 인코딩된 비디오 데이터를 목적지 디바이스 (14)에 송신할 수 있는 타입의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버들 (예를 들어, 웹사이트), 파일 전송 프로토콜 (FTP) 서버들, 네트워크 접속형 저장 (NAS) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.

[0043] 목적지 디바이스 (14)는 인터넷 커넥션과 같은 표준 데이터 커넥션을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 예시적인 타입의 데이터 커넥션들은 무선 채널들 (예를 들어, Wi-Fi 커넥션들), 유선 커넥션들 (예를 들어, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 양자의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로딩 송신, 또는 양자의 조합일 수도 있다.

[0044] 본 개시의 기술들은 무선 애플리케이션들 또는 설정들에 제한되지 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 각종 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.

[0045] 도 1에 예시된 비디오 코딩 시스템 (10)은 단지 예이며, 본 개시의 기술들은 인코딩 디바이스와 디코딩 디바이스 사이에 어떠한 데이터 통신도 반드시 포함하지 않는 비디오 코딩 설정들 (예를 들어, 비디오 인코딩 또는 비디오 디코딩)에 적용될 수도 있다. 다른 예에서, 데이터는 로컬 메모리로부터 취출되고, 네트워크를 통해 스트리밍된다. 비디오 인코딩 디바이스가 데이터를 인코딩하여 메모리에 저장할 수도 있고, 그리고/또는 비디오 디코딩 디바이스가 메모리로부터 데이터를 취출하여 디코딩할 수도 있다. 다수의 예들에서, 인코딩 및 디코딩은 서로 통신하지 않는 디바이스들에 의해 수행되지만, 단순히 데이터를 인코딩하여 메모리에 저장하고 그리고/또는 메모리로부터 데이터를 취출하여 디코딩한다.

[0046] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 일부 예들에서, 출력 인터페이스 (22)는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 비디오 소스 (18)는 비디오 캡쳐 디바이스, 예를 들어, 비디오 카메라, 이전에 캡쳐된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하기 위한 비디오 퍼드 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 비디오 데이터의 그러한 소스들의 조합을 포함할 수도 있다.

[0047] 비디오 인코더 (20)는 비디오 소스 (18)로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12)는 인코딩된 비디오 데이터를 출력 인터페이스 (22)를 통해 목적지 디바이스 (14)에 직접 송신한다. 다른 예에서, 인코딩된 비디오 데이터는 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14)에 의한 더 나중의 액세스를 위해 저장 매체 또는 파일 서버 상에 또한 저장될 수도 있다.

[0048] 도 1의 예에서, 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 일부 예들에서, 입력 인터페이스 (28)는 수신기 및/또는 모뎀을 포함한다. 입력 인터페이스 (28)는 채널 (16)을 통해 인코딩된 비디오 데이터를 수신할 수도 있다. 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합되거나 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ダイ오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 각종 디스플레이 디바이스들을 포함할 수도 있다.

[0049] 본 개시는 특정한 정보를 비디오 디코더 (30)와 같은 다른 디바이스에 "시그널링"하거나 "송신"하는 비디오 인코더 (20)를 일반적으로 지칭할 수도 있다. 용어 "시그널링" 또는 "송신"은 압축된 비디오 데이터를 디코딩하기 위해 사용된 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 일반적으로 지칭할 수도 있다. 이러한 통신은 실시간으로 또는 거의 실시간으로 발생할 수도 있다. 대안으로, 이러한 통신은 인코딩시에 인코딩된 비트스트림에서 컴퓨터 판독가능 매체에 선택스 엘리먼트들을 저장할 때 발생할 수도 있고, 그 후, 이러한 매체에 저장된 이후의 임의의 시간에 디코딩 디바이스에 의해 취출될 수도 있는 것과 같이, 일정 기간을 통해 발생할 수도 있다. 따라서, 비디오 디코더 (30)가 특정한 정보를 "수신하는 것"으로서 지칭될 수도 있지만, 정보를 수신하는 것은 실시간 또는 거의 실시간으로 반드시 발생하지 않고, 저장 이후 일부 시간에 매체로부터 취출될 수도 있다.

[0050] 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 응용 주문형 집적 회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 개별 로직, 하드웨어, 또는 이들의 임의의 조합들과 같은 임의의 각종 적합한 회로로서 구현될 수도 있다. 기술들이 소프트웨어로 부분적으로 구현되는 경우에, 디바이스가 적합한 비일시적 컴퓨터 판독가능 저장 매체에 소프트웨어용 명령어들을 저장할 수도 있고, 본 개시의 기술들을 수행하기 위해 하나 이상의 프로세서들을 사용하여 하드웨어에서 명령어들을 실행할 수도 있다. (하드웨어, 소프트웨어, 하드웨어와 소프트웨어의 조합 등을 포함하는) 상술한 것 중 임의의 것이 하나 이상의 프로세서들인 것으로 고려될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 각각의 디바이스에서 조합된 인코더/디코더 (코덱)의 일부로서 통합될 수도 있는 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있다.

[0051] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 상기 언급되고 HEVC 드래프트 10에 설명된 HEVC 표준과 같은 비디오 압축 표준에 따라 동작한다. 베이스 HEVC 표준에 부가하여, 스케일러를 비디오 코딩, 멀티뷰 비디오 코딩, 및 HEVC에 대한 3D 코딩 확장들을 생성하기 위한 진행중인 노력이 존재한다. 또한, 예를 들어, 본 개시에 설명한 바와 같은, 팔레트-기반 코딩 모드들은 HEVC 표준의 확장을 위해 제공될 수도 있다.

일부 예들에서, 팔레트-기반 코딩에 대한 본 개시에 설명된 기술들은 ITU-T-H.264/AVC 표준 또는 장래의 표준들과 같은 다른 비디오 코딩 표준들에 따라 동작하도록 구성된 인코더들 및 디코더들에 적용될 수도 있다.

이에 따라, HEVC 코덱에서 코딩 유닛들 (CU들) 또는 예측 유닛들 (PU들)의 코딩을 위한 팔레트-기반 코딩 모드의 애플리케이션이 예시의 목적을 위해 설명된다.

[0052] HEVC 및 다른 비디오 코딩 표준들에서, 비디오 시퀀스는 일련의 블록들을 통상적으로 포함한다. 블록들은 "프레임들"로서 또한 지칭될 수도 있다. 블록은 S_L , S_{Cb} 및 S_{Cr} 로 표기되는 3개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2차원 어레이 (즉, 블록)이다. S_{Cb} 는 Cb 크로미넌스 샘플들의 2차원 어레이이다. S_{Cr} 은 Cr 크로미넌스 샘플들의 2차원 어레이이다. 크로미넌스 샘플들은 "크로마" 샘플들로서 본 명세서에서 또한 지칭될 수도 있다. 다른 경우들에서, 블록은 단색일 수도 있으며 오직 루마 샘플들의 어레이를 포함할 수도 있다.

[0053] 블록의 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 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하기 위해 사용된 선택스 구조들일 수도 있다. 코딩 트리 블록은 샘플들의 $N \times N$ 블록일 수도 있다. CTU는 "트리 블록" 또는 "최대 코딩 유닛 (LCU)"으로서 또한 지칭될 수도 있다. HEVC의 CTU들은 H.264/AVC와 같은 다른 표준들의 매크로블록들과 대체로 유사할 수도 있다. 그러나, CTU는 반드시 특정 사이즈로 한정되지 않으며 하나 이상의 코딩 유닛들 (CU들)을 포함할 수도 있다.

슬라이스가 레스터 스캔에서 연속적으로 순서화된 정수 개의 CTU들을 포함할 수도 있다. 코딩된 슬라이스가 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스의 슬라이스 헤더는 슬라이스에 관한 정보를 제공하는 선택스 엘리먼트들을 포함하는 선택스 구조일 수도 있다. 슬라이스 데이터는 슬라이스의 코딩된 CTU들을 포함할 수도 있다.

[0054] 본 개시는 샘플들의 하나 이상의 블록들의 샘플들을 코딩하기 위해 사용된 하나 이상의 샘플 블록들 및 선택스 구조들을 지칭하기 위해 용어 "비디오 유닛" 또는 "비디오 블록" 또는 "블록"을 사용할 수도 있다. 비디오

유닛들 또는 블록들의 예시적인 타입들은 CTU들, CU들, PU들, 변환 유닛들 (TU들), 매크로블록들, 매크로블록 파티션들 등을 포함할 수도 있다. 일부 문맥에서, PU들의 논의는 매크로블록들 또는 매크로블록 파티션들의 논의와 교환될 수도 있다.

[0055] 코딩된 CTU 를 생성하기 위해, 비디오 인코더 (20) 는 CTU 의 코딩 트리 블록들에 대해 큐드-트리 파티셔닝을 재귀적으로 수행하여, 코딩 트리 블록들을 코딩 블록들, 따라서, 명칭 "코딩 트리 유닛들"로 분할할 수도 있다. 코딩 블록은 샘플들의 $N \times N$ 블록이다. CU 는 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록, 및 크로마 샘플들의 2개의 대응하는 코딩 블록들, 그리고 그 코딩 블록들의 샘플들을 코딩하는데 사용된 선택스 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, 픽처의 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용된 선택스 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.

[0056] 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 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0057] 비디오 인코더 (20) 가 인터 예측을 이용하여 PU 의 예측 블록들을 생성하면, 비디오 인코더 (20) 는 PU 와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 는 단방향 예측 또는 양방향 예측을 이용하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 단방향 예측을 이용하여 PU 에 대한 예측 블록들을 생성할 때, PU 는 단일 모션 벡터 (MV) 를 가질 수도 있다. 비디오 인코더 (20) 가 양방향 예측을 이용하여 PU 에 대한 예측 블록들을 생성할 때, PU 는 2개의 MV들을 가질 수도 있다.

[0058] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 블록들 (예를 들어, 예측 루마, Cb 및 Cr 블록들) 을 생성한 이후에, 비디오 인코더 (20) 는 CU 에 대한 잔차 블록들을 생성할 수도 있다. CU 의 잔차 블록에서의 각각의 샘플은 CU 의 PU 의 예측 블록에서의 샘플과 CU 의 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 생성할 수도 있다. CU 의 잔차 블록에서의 각각의 샘플은 CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 오리지널 루마 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 수도 있다. 또한, 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 CU 에 대한 Cr 잔차 블록을 또한 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.

[0059] 더욱이, 비디오 인코더 (20) 는 큐드 트리 파티셔닝을 이용하여, CU 의 잔차 블록들 (예를 들어, 루마, Cb, 및 Cr 잔차 블록들) 을 하나 이상의 변환 블록들 (예를 들어, 루마, Cb, 및 Cr 변환 블록들) 로 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 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용된 선택스 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-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다.

[0060] 비디오 인코더 (20) 는 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 계수 블록을 생성할 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라량일 수도 있다. 예를 들어, 비디오 인코더 (20) 는 TU 의 루마 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 루마 계수 블록을 생성할 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.

[0061] 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 계수들이 그 변환 계수들을 나타내는데 사용되는 데이터의 양을 가능하게는 감소시키도록 양자화되어 추가 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들에 대해 컨텍스트-적응 바이너리 산술 코딩 (CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩된 신택스 엘리먼트를 비트스트림으로 출력할 수도 있다. 비트스트림은 엔트로피 인코딩되지 않은 신택스 엘리먼트들을 또한 포함할 수도 있다.

[0062] 비디오 인코더 (20) 는 엔트로피 인코딩된 신택스 엘리먼트를 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 원시 바이트 시퀀스 페이로드 (RBSP) 를 캡슐화한다. NAL 유닛 헤더는, NAL 유닛 태입 코드를 나타내는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 명시된 NAL 유닛 태입 코드는 NAL 유닛의 태입을 나타낸다. RBSP 는, NAL 유닛 내에서 캡슐화되는 정수 개수의 바이트들을 포함하는 신택스 구조일 수도 있다. 일부 경우들에 있어서, RBSP 는 제로 비트들을 포함한다.

[0063] 상이한 태입들의 NAL 유닛들은 상이한 태입들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 태입은 픽처 파라미터 세트 (PPS) 에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 2 태입은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있으며, NAL 유닛의 제 3 태입은 보충 인핸스먼트 정보 (SEI) 에 대한 RBSP 를 캡슐화할 수도 있는 등등이다. (파라미터 세트들 및 SEI 메시지들에 대한 RBSP들과 대조적으로) 비디오 코딩 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 비디오 코딩 계층 (VCL) NAL 유닛들로서 지칭될 수도 있다.

[0064] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 또한, 비디오 디코더 (30) 는 비트스트림으로부터 신택스 엘리먼트들을 획득할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 비트스트림을 파싱하여 비트스트림으로부터 신택스 엘리먼트들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 (예를 들어, 디코딩된) 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하기 위한 프로세스는 일반적으로, 비디오 인코더 (20) 에 의해 수행된 프로세스에 역할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 PU들을 MV들을 이용하여 현재 CU 의 PU들에 대한 예측 샘플 블록들 (즉, 예측 블록들) 을 결정할 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 대해 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 복원할 수도 있다. 픽처의 각각의 CU 에 대한 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 픽처를 복원할 수도 있다.

[0065] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다. 예를 들어, 팔레트-기반 코딩에서, 상술한 인트라-예측 또는 인터-예측 코딩 기술들을 수행하기 보다는, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 특정한 영역 (예를 들어, 주어진 블록) 의 비디오 데이터를 나타내는 컬러들 또는 픽셀 값들의 테이블로서 소위 팔레트를 코딩할 수도 있다. 이러한 방식으로, 비디오 데이터의 현재 블록에 대한 실제 픽셀 값들 또는 그들의 잔차들을 코딩하기 보다는, 비디오 코더는 현재 블록의 픽셀 값들 중 하나 이상에 대한 인덱스 값들을 코딩할 수도 있고, 여기서, 인덱스 값들은 현재 블록의 픽셀 값들을 나타내기 위해 사용되는 팔레트에서의 엔트리들을 나타낸다.

[0066] 예를 들어, 비디오 인코더 (20) 는 블록에 대한 팔레트를 결정하고, 블록의 하나 이상의 개별 픽셀들의 값을 나타내는 값을 갖는 팔레트에서의 엔트리를 로케이팅하며, 블록의 하나 이상의 개별 픽셀 값들을 표현하기 위해 사용된 팔레트에서의 엔트리를 나타내는 인덱스 값들로 블록을 인코딩함으로써 비디오 데이터의 블록을 인코딩 할 수도 있다. 부가적으로, 비디오 인코더 (20) 는 인덱스 값들을 인코딩된 비트스트림에서 시그널링할 수도 있다. 결과적으로, 비디오 디코딩 디바이스 (예를 들어, 비디오 디코더 (30)) 는 인코딩된 비트스트림으로부터, 팔레트를 사용하여 블록의 다양한 개별 픽셀들을 결정하기 위해 사용된 인덱스 값들 뿐만 아니라 블록에 대한 팔레트를 획득할 수도 있다. 비디오 디코더 (30) 는 개별 픽셀들의 인덱스 값들을 팔레트의 엔트리들에 매칭하여 블록의 픽셀 값들을 복원할 수도 있다. 개별 픽셀과 연관된 인덱스 값이 블록에 대한 대응하는 팔레트의 임의의 인덱스 값에 매칭하지 않는 경우들에서, 비디오 디코더 (30) 는 이러한 픽셀을 팔레트-기반

코딩을 위한 이스케이프 픽셀로서 식별할 수도 있다.

[0067] 다른 예에서, 비디오 인코더 (20)는 아래의 동작들에 따라 비디오 데이터의 블록을 인코딩할 수도 있다. 비디오 인코더 (20)는 블록의 개별 픽셀들에 대한 예측 잔차 값들을 결정할 수도 있고, 블록에 대한 팔레트를 결정할 수도 있으며, 개별 픽셀들의 예측 잔차 값들 중 하나 이상의 값을 나타내는 값을 갖는 팔레트에서 엔트리 (예를 들어, 인덱스 값)를 로케이팅할 수도 있다. 부가적으로, 비디오 인코더 (20)는 블록의 각각의 개별 픽셀에 대한 대응하는 예측 잔차 값을 나타내는데 사용된 팔레트에서의 엔트리를 나타내는 인덱스 값들로 블록을 인코딩할 수도 있다. 비디오 디코더 (30)는 소스 디바이스 (12)에 의해 시그널링된 인코딩된 비트 스트림으로부터, 블록의 개별 픽셀들에 대응하는 예측 잔차 값들에 대한 인덱스 값을 뿐만 아니라 블록에 대한 팔레트를 획득할 수도 있다. 설명한 바와 같이, 인덱스 값들은 현재 블록과 연관된 팔레트에서의 엔트리들에 대응할 수도 있다. 결과적으로, 비디오 디코더 (30)는 예측 잔차 값들의 인덱스 값을 팔레트의 엔트리들에 연관시켜 블록의 예측 잔차 값을 복원할 수도 있다. 예측 잔차 값들은 (예를 들어, 인트라 또는 인터 예측을 사용하여 획득된) 예측 값들에 부가되어 블록의 픽셀 값을 복원할 수도 있다.

[0068] 더욱 상세히 후술하는 바와 같이, 팔레트-기반 코딩의 기본 아이디어는, 코딩될 비디오 데이터의 주어진 블록에 대해, 비디오 인코더 (20)가 현재 블록에서 가장 우세한 픽셀 값을 포함하는 팔레트를 유도할 수도 있다는 것이다. 예를 들어, 팔레트는 현재 CU에 대해 우세하고 그리고/또는 대표하는 것으로 결정되거나 가정되는 다수의 픽셀 값을 지정할 수도 있다. 비디오 인코더 (20)는 팔레트의 사이즈 및 엘리먼트들을 비디오 디코더 (30)에 먼저 송신할 수도 있다. 부가적으로, 비디오 인코더 (20)는 특정한 스캐닝 순서에 따라 주어진 블록에서 픽셀 값을 인코딩할 수도 있다. 주어진 블록에 포함된 각각의 픽셀에 대해, 비디오 인코더 (20)는 픽셀 값을 팔레트에서의 대응하는 엔트리에 매핑하는 인덱스 값을 시그널링할 수도 있다. 픽셀 값이 팔레트에 포함되지 않은 경우에 (즉, 팔레트-코딩된 블록의 특정한 픽셀 값을 특정하는 팔레트 엔트리가 존재하지 않는 경우에), 이러한 픽셀은 "이스케이프 픽셀"로서 정의된다. 팔레트-기반 코딩에 따르면, 비디오 인코더 (20)는 이스케이프 픽셀을 위해 예약된 인덱스 값을 인코딩하고 시그널링할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 주어진 블록에 포함된 이스케이프 픽셀에 대한 픽셀 값 또는 잔차 값 (또는 그것의 양자화된 버전들)을 또한 인코딩하고 시그널링할 수도 있다.

[0069] 비디오 인코더 (20)에 의해 시그널링된 인코딩된 비디오 비트스트림의 수신시에, 비디오 디코더 (30)는 비디오 인코더 (20)로부터 수신된 정보에 기초하여 팔레트를 먼저 결정할 수도 있다. 그 후, 비디오 디코더 (30)는 주어진 블록에서의 픽셀 위치들과 연관된 수신된 인덱스 값을 팔레트의 엔트리들에 매핑하여, 주어진 블록의 픽셀 값을 복원할 수도 있다. 일부 경우들에서, 비디오 디코더 (30)는 예를 들어, 픽셀이 이스케이프 픽셀들에 대해 예약된 인덱스 값으로 팔레트-코딩되었다는 것을 결정함으로써, 팔레트-코딩된 블록의 픽셀이 이스케이프 픽셀이라는 것을 결정할 수도 있다. 비디오 디코더 (30)가 팔레트-코딩된 블록에서 이스케이프 픽셀을 식별하는 경우들에서, 비디오 디코더 (30)는 주어진 블록에 포함된 이스케이프 픽셀에 대한 픽셀 값 또는 잔차 값 (또는 그것의 양자화된 버전들)을 수신할 수도 있다. 비디오 디코더 (30)는 개별 픽셀 값을 대응하는 팔레트 엔트리들에 매핑하며, 픽셀 값 또는 잔차 값 (또는 그것의 양자화된 버전들)을 사용함으로써 팔레트-코딩된 블록을 복원하여, 팔레트 코딩된 블록에 포함된 임의의 이스케이프 픽셀들을 복원할 수도 있다.

[0070] 팔레트-기반 코딩은 시그널링 오버헤드의 양을 도입할 수도 있다. 예를 들어, 팔레트 자체 뿐만 아니라 팔레트의 사이즈와 같은 팔레트의 특징들을 시그널링하기 위해서는 다수의 비트들이 필요할 수도 있다. 부가적으로, 블록의 픽셀들에 대한 인덱스 값을 시그널링하기 위해서는 다수의 비트들이 필요할 수도 있다. 예를 들어, 기준의 팔레트-기반 코딩 기술들에 따르면, 팔레트가 제한된 사이즈로 이루어지고, (즉, 단지 하나의 엔트리를 포함하는 팔레트) 블록이 어떠한 이스케이프 픽셀들도 포함하지 않는 경우들에서도, 비디오 인코더 (20)는 비디오 블록에 대해 라인마다에 기초하여, 모두가 팔레트에서의 하나의 엔트리를 식별하는 동일한 인덱스 값인 블록의 픽셀 값들에 대한 팔레트 인덱스들을 여전히 시그널링할 수도 있다. 부가적으로, 기준의 팔레트-기반 코딩 기술들은 이스케이프 픽셀을 나타내기 위해 인덱스 값을 시그널링한 후, 이스케이프 픽셀에 대한 픽셀 값 또는 잔차 값 (또는 그것의 양자화된 버전들)을 시그널링하는 것에 관하여 시그널링 오버헤드를 도입한다.

[0071] 본 개시의 기술들은 일부 예들에서, 이러한 정보를 시그널링하는데 필요한 비트들의 수를 감소시킬 수도 있다. 예를 들어, 본 명세서에 설명한 특정한 기술들은 블록이 팔레트-기반 코딩에 관한 조건들의 특정한 세트를 충족하는 경우에 블록의 하나 이상의 픽셀들에 대한 인덱스 값을 맵의 코딩 (예를 들어, 인코딩 및/또는 디코딩)을 바이패스하는 것에 관한 것이다. 다른 예에서, 본 명세서에 설명한 특정한 기술들은 일반적으로, 주

어진 픽셀이 현재 블록에 대한 팔레트에 관하여 이스케이프 픽셀이라는 것을 나타내기 위해 데이터의 적은 비트들 (예를 들어, 5-비트 인덱스 값 대신에 1-비트 플래그) 을 시그널링하는 것에 관한 것이다. 본 개시의 다양한 기술들은 또한, 주어진 블록에 대한 팔레트에 포함될 수도 있는 픽셀 값들의 범위를 결정하는 것에 관한 것이다. 팔레트가 포함할 수도 있는 픽셀 값들의 범위는 본 명세서에서 팔레트의 "에러 제한"으로서 지정되고, 본 개시의 다양한 기술들은 팔레트와 연관된 블록의 양자화 파라미터 (QP) 에 기초하여 팔레트의 에러 제한을 결정하는 것에 관한 것이다.

[0072] 본 개시의 다른 양태들은 양자화된 이스케이프 값들을 유도하는 것에 관한 것이다. 예를 들어, 이들 양태들 중 일부는 이스케이프 픽셀을 양자화하는 하나 이상의 양자화 파라미터들을 정의하는 기술들에 관한 것이다.

본 개시의 또 다른 양태들은 이스케이프 픽셀 값들을 양자화하는데 특정한 함수들 (예를 들어, 오른쪽-시프트 함수) 를 적용하는 것에 관한 것이다. 이러한 방식으로, 본 개시의 다양한 양태들은 픽처 품질 및 정확성을 유지하면서, 비트 오버헤드를 감소시키고 자원 사용을 완화시키는 것과 같은, 잠재적 이점들을 제공한다.

[0073] 상술한 바와 같이, 비디오 인코더 (20) 는 특정한 환경들 하에서 블록의 다양한 개별 픽셀들에 대한 팔레트 인덱스의 인코딩 및 시그널링을 바이패스하기 위해 본 개시의 기술들을 적용할 수도 있다. 본 개시의 양태들에 따르면, 비디오 인코더 (20) 는 비디오 인코더 (20) 가 블록의 픽셀들이 동일한 컬러로 되어 있다는 것을 결정하는 경우에 팔레트-코딩된 블록에 대한 팔레트 인덱스의 인코딩 및 시그널링을 바이패스할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 픽처의 팔레트-코딩된 CU가 "단일 컬러 CU" 이라는 것을 결정할 수도 있고, 단일 컬러 CU 에 대한 팔레트 인덱스의 인코딩 및 시그널링을 바이패스할 수도 있다.

[0074] 더욱 구체적으로, 비디오 인코더 (20) 는 현재 CU 가 2개의 조건들을 충족시키는 경우에 팔레트-코딩된 CU 가 단일 컬러 CU 인지를 결정할 수도 있다. 비디오 인코더 (20) 가 팔레트-코딩된 CU 가 단일 컬러 CU 인지를 결정하는데 사용할 수도 있는 제 1 조건은 대응하는 팔레트의 사이즈가 1 인지이다. 팔레트의 사이즈가 1 인 경우에, 비디오 인코더 (20) 는 제 1 조건이 단일 컬러 CU 인 팔레트-코딩된 CU 에 관하여 충족된다는 것을 결정할 수도 있다. 더욱 구체적으로, 팔레트 사이즈가 1 인 경우에, 비디오 인코더 (20) 는 팔레트가 팔레트-코딩된 CU 의 (넌-이스케이프) 픽셀들에 대응하는 하나의 컬러만을 포함한다는 것을 결정할 수도 있다. 팔레트 사이즈가 1 인 일부 예들에서, 비디오 인코더 (20) 는 팔레트와 연관된 인덱스 값만이 0 이라는 것을 결정할 수도 있다.

[0075] 비디오 인코더 (20) 가 팔레트-코딩된 CU 에 대한 팔레트 사이즈가 1 이라는 것 (즉, 제 1 조건이 충족된다는 것) 을 결정하는 경우에, 비디오 인코더 (20) 는 팔레트-코딩된 CU 가 단일 컬러 CU 이도록 제 2 조건을 충족하는지를 결정할 수도 있다. 비디오 디코더 (30) 가 팔레트-코딩된 CU가 단일 컬러 CU 인지를 결정하는데 사용할 수도 있는 제 2 조건은, 팔레트-코딩된 CU가 어떠한 이스케이프 픽셀들도 포함하지 않는다는 것이다. 팔레트-코딩된 CU가 적어도 하나의 이스케이프 픽셀을 포함하는 경우에, 비디오 인코더 (20) 는, 대응하는 팔레트가 팔레트-코딩된 CU 에 관하여 오직 하나의 컬러만을 나타내더라도, 팔레트-코딩된 CU 가 2개 이상의 컬러들의 픽셀들을 포함한다는 것을 결정할 수도 있다. 예를 들어, 팔레트-코딩된 CU 는 팔레트에 나타난 컬러를 갖는 적어도 하나의 픽셀, 및 상이한 컬러를 갖는 적어도 하나의 이스케이프 픽셀을 포함할 수도 있다.

[0076] 비디오 인코더 (20) 가 팔레트-코딩된 CU가 상술한 조건들 양자를 충족한다는 것을 결정하는 경우에, 비디오 인코더 (20) 는 팔레트-코딩된 CU 가 단일 컬러 CU 이라는 것을 결정할 수도 있다. 더욱 구체적으로, 팔레트-코딩된 CU 가 (1 의 팔레트 사이즈에 의해 도시된) 단일-엔트리 팔레트와 연관되며, 팔레트-코딩된 CU 가 어떠한 이스케이프 픽셀들도 포함하지 않는 경우에, 비디오 인코더 (20) 는 팔레트-코딩된 CU의 모든 개별 픽셀들이 동일한 컬러 (즉, 대응하는 팔레트의 단일 엔트리에 의해 표시된 컬러) 로 되어 있다는 것을 결정할 수도 있다.

다양한 구현들에서, 비디오 인코더 (20) 는 팔레트, 또는 하나 보다 많은 컬러 컴포넌트를 나타내는 조합된 인덱스에 관한 단일 컬러 컴포넌트 기초에 관하여 상술한 단일 컬러 CU 식별 기술들을 적용할 수도 있다.

[0077] 팔레트-코딩된 CU가 오직 하나의 컬러의 픽셀들을 포함한다는 (즉, CU 가 단일 컬러 CU 이라는) 결정에 응답하여, 비디오 인코더 (20) 는 단일 컬러 CU 의 픽셀들에 대한 팔레트 인덱스 값들의 맵을 인코딩하고 시그널링하는 것을 바이패스하기 위해 본 개시의 기술들을 구현할 수도 있다. CU 의 픽셀들에 대한 팔레트 인덱스 값들의 맵의 인코딩 및 시그널링을 바이패스함으로써, 비디오 인코더 (20) 는 컴퓨팅 자원들 및 대역폭을 보존할 수도 있고, 그렇지 않으면, 컴퓨팅 자원들 대역폭은 CU 에 대한 컬러 정보를 인코딩하고 시그널링하는데 소모된다. 단일 컬러 CU 의 각각의 개별 픽셀을 인코딩하고 시그널링하는 대신에, 비디오 인코더 (20) 는 단일 컬러 CU 를 구성하는 픽셀들의 전체 세트에 대한 컬러 정보를 (예를 들어, 비디오 디코더 (30) 예) 더욱 효율적으로 나타내기 위해 본 개시의 기술들을 구현할 수도 있다.

[0078]

본 개시의 일부 예들에 따르면, 비디오 인코더 (20) 는 비디오 인코더 (20) 가 단일 컬러 CU 에 대해 라인마다에 기초하여 팔레트 인덱스 값들의 인코딩 (및 시그널링) 을 바이패스하였는지를 나타내기 위해 플래그를 인코딩하고 시그널링할 수도 있다. CU 의 각각의 픽셀에 대한 개별 인덱스 값들 대신에 전체 CU 에 대한 1-비트 플래그를 인코딩하고 시그널링함으로써, 비디오 인코더 (20) 는 기존의 팔레트-기반 코딩 기술들에 비하여 컴퓨팅 차원들 및 시그널링 대역폭을 보존할 수도 있다. 더욱이, 비디오 인코더 (20) 는, CU 에 대해 비디오 인코더 (20) 에 의해 시그널링된 단일-엔트리 팔레트가 CU 의 모든 개별 픽셀들에 대한 컬러 정보를 포함하기 때문에, 인코딩된 단일 컬러 CU 의 정확성 및 품질을 유지할 수도 있다. 다양한 예들에서, 비디오 인코더 (20) 는 시퀀스 파라미터 세트 (SPS), 픽처 파라미터 세트 (PPS), 또는 슬라이스 헤더에서와 같이, 다양한 방식들로 플래그를 인코딩하고 시그널링할 수도 있다. 다양한 예들에서, 비디오 인코더 (20) 는 또한 CTU 마다에 기초하여, CU 마다에 기초하여, 또는 임의의 블록 사이즈의 블록에 대해 플래그를 인코딩하고 시그널링할 수도 있다.

[0079]

비디오 인코더 (20) 가 팔레트-코딩된 단일 컬러 블록의 개별 픽셀들에 대한 팔레트 인덱스 값의 인코딩 및 시그널링을 바이패스하는 예들에서, 비디오 디코더 (30) 는 단일 컬러 블록을 복원하기 위해 본 개시의 다양한 기술들을 적용할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 비디오 인코더 (20) 에 관하여 상술한 것들에 대해 역인 동작들을 수행하여 팔레트-코딩된 블록이 단일 컬러 블록이라는 것을 결정할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재 블록에 대한 팔레트가 1 의 사이즈를 갖는다는 것을 결정하여, 블록이 제1 조건을 충족하여 단일 컬러 블록으로서 적합하다는 것을 결정할 수도 있다. 다양한 예들에서, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터 인코딩된 비트스트림에서 팔레트를 수신할 수도 있거나, 팔레트를 복원할 수도 있다.

[0080]

부가적으로, 비디오 디코더 (30) 는 블록이 어떠한 이스케이프 픽셀들도 포함하지 않는다는 것을 결정하여, 블록이 제2 조건을 충족하여 단일 컬러 블록으로서 적합하다는 것을 결정할 수도 있다. 블록에 대한 팔레트의 사이즈가 1 이라는 것 (제1 조건) 과, 블록이 어떠한 이스케이프 픽셀들도 포함하지 않는다는 것 (제2 조건) 을 결정하는 것에 기초하여, 비디오 디코더 (30) 는 현재 블록이 단일 컬러 블록이라는 것을 결정하기 위해 본 개시의 기술들을 구현할 수도 있다. 이러한 방식으로, 비디오 디코더 (30) 는, 그렇지 않으면 픽셀마다에 기초하여 팔레트 인덱스를 디코딩함으로써 블록을 복원하도록 요구되는 컴퓨팅 차원들 및 대역폭을 보존하면서, 팔레트-코딩된 블록을 정확하게 복원하기 위해 본 개시의 기술들을 구현할 수도 있다.

[0081]

다른 예들에서, 비디오 디코더 (30) 는 본 개시의 기술들에 따라, 비디오 인코더 (20) 가 팔레트-코딩된 블록에 대한 팔레트 인덱스의 인코딩 및 시그널링을 바이패스하였는지를 나타내는 플래그를, 인코딩된 비디오 비트스트림에서 수신할 수도 있다. 비디오 인코더 (20) 가 팔레트-코딩된 블록에 대한 팔레트 인덱스 값들의 맵의 인코딩 및 시그널링을 바이패스하였다는 것을 나타내는 플래그를 비디오 디코더 (30) 가 수신하는 경우에서, 비디오 디코더 (30) 는 현재 블록이 팔레트-코딩되며, 단일 컬러 블록이라는 것을 결정하기 위해 본 개시의 기술들을 구현할 수도 있다. 더욱 구체적으로, 플래그가 인에이블되는 경우에 (예를 들어, 1 의 값으로 설정되는 경우에), 비디오 디코더 (30) 는 팔레트-코딩된 블록이 단일 컬러 블록이라는 것을 결정할 수도 있다. 결과적으로, 비디오 디코더 (30) 는 블록에 대한 팔레트에서의 단일 엔트리의 컬러 정보에 따라 블록의 각각의 픽셀을 복원하기 위해 본 개시의 기술들을 구현할 수도 있다. 이러한 방식으로, 비디오 디코더 (30) 는, 블록의 각각의 개별 픽셀에 대한 (변화하는 비트깊이의) 개별 인덱스 값들을 사용하기 보다는, 전체 블록에 대한 1-비트 플래그를 사용하여 팔레트-코딩된 블록을 정확하게 복원하기 위해 본 개시의 기술들을 구현할 수도 있다.

[0082]

다른 예에서, 비디오 인코더 (20) 는 팔레트-코딩된 블록에 대한 팔레트에 대한 에러 제한을 유도하기 위해 본 개시의 기술들을 구현할 수도 있다. 본 명세서에서 사용되는 바와 같이, 용어들 "에러 제한" 또는 팔레트 에러 제한"은 팔레트의 엔트리들이 포함할 수 있는 (예를 들어, 컬러 정보에 관한) 값들의 범위를 지정할 수도 있다. 더욱 구체적으로, 팔레트 에러 제한은 상이한 팔레트 엔트리들이 부담하거나, 나타내야 하는 컬러 값에서의 최소 변동을 정의한다. 상술한 바와 같이, 팔레트-기반 코딩에 따라 블록을 인코딩하기 위해, 비디오 인코더 (20) 는 블록 내에서 (픽셀마다에 기초하여) 가장 빈번하게 발생하는 컬러 값들을 포함하도록 대응하는 팔레트를 구성할 수도 있다.

[0083]

팔레트를 구성하는데 있어서, 비디오 인코더 (20) 는 팔레트의 다양한 엔트리들이 서로로부터 최소 변동을 나타내야 한다는 것을 결정할 수도 있다. 더욱 구체적으로, 비디오 인코더 (20) 는 팔레트의 2개의 엔트리들이 충분히 유사하지 않아 2개의 엔트리들이 단일 엔트리로서 그룹화될 수 있도록 팔레트를 구성할 수도 있다. 2개의 가능한 팔레트 엔트리들이 팔레트 에러 제한 내에 있는 경우에, 비디오 인코더 (20) 는 팔레트에서 엔트

리들 양자를 나타내기 위해 2개의 엔트리들 중 하나를 사용할 수도 있다.

[0084] 그러나, 비디오 인코더 (20) 가 (블록에서 일반적으로 발생하는) 2개의 엔트리들이 적어도 팔레트 에러 제한 만큼 상이하다는 것을 결정하는 경우에, 비디오 인코더 (20) 는 팔레트에서 엔트리들 양자를 포함할 수도 있다.

엔트리들이 3개의 컬러 컴포넌트들에 의해 표현되는 예들에서, 비디오 인코더 (20) 는 엔트리들이 컬러 컴포넌트들 중 적어도 하나에 관하여 적어도 팔레트 에러 제한 만큼 상이한 경우에 팔레트에서 엔트리들 양자를 포함할 수도 있다. 예를 들어, 팔레트 에러 제한이 5 의 값으로 설정되는 경우에, 비디오 인코더 (20) 는 아래의 부울 표현: $abs(A1-B1) > 5 \text{ || } abs(A2-B2) > 5 \text{ || } abs(A3-B3) > 5$ (여기서 "abs" 는 컬러 컴포넌트 파라미터들 사이의 차이를 나타냄) 에 기초하여, (엔트리들 양자가 블록에서 일반적으로 충분히 발생한다는 것을 가정하여) 팔레트에서 엔트리들 양자를 포함하는지 결정할 수도 있다.

[0085] 설명한 바와 같이, 비디오 인코더 (20) 는 블록의 일반적으로 발생하는 (또는 비교적 일반적으로 발생하는) 픽셀 값들을 팔레트의 엔트리들로 클러스터링함으로써 팔레트를 구성할 수도 있다. 비디오 인코더 (20) 는 픽셀 값들이 컬러 정보에 관하여 최소 변동을 나타내도록 일반적으로 발생하는 픽셀 값들을 선택할 수도 있다.

결과적으로, 일반적으로 발생하는 픽셀 값들의 선택된 세트 내의 픽셀 값들 사이의 최소 변동은 대응하는 팔레트의 에러 제한을 형성할 수도 있다. 팔레트 에러 제한이 여러 픽셀 값들을 포함할 수도 있지만, 팔레트가 적어도 팔레트 에러 제한만큼 상이한 픽셀 값들의 쌍 모두를 반드시 포함하지 않을 수도 있다는 것이 이해될 것이다. 따라서, 동일한 팔레트 에러 제한이 변화하는 사이즈들의 팔레트들에 적용될 수도 있다. 비디오 인코더 (20) 는 팔레트에 포함될 컬러 값들에 관한 결정들을 하는데 있어서 팔레트 에러 제한을 사용할 수도 있다.

[0086] 비디오 인코더 (20) 는 팔레트에 대한 에러 제한을 정의하기 위해 본 개시의 기술들을 구현할 수도 있다. 본 개시의 다양한 양태들에 따르면, 비디오 인코더 (20) 는 팔레트-코딩된 블록에 대한 양자화 파라미터 (QP) 에 기초하여 팔레트 에러 제한을 결정할 수도 있다. 다양한 예들에서, 비디오 인코더 (20) 는 팔레트 에러 제한이 대응하는 블록에 대한 QP 값에 정비례한다는 것을 결정할 수도 있다. 더욱 구체적으로, 이들 예들에서, 비디오 인코더 (20) 는 더 큰 QP 값으로 양자화되는 블록에 대한 팔레트에 대해 더 큰 에러 제한, 및 더 작은 QP 값으로 양자화되는 블록에 대한 팔레트에 대해 더 작은 에러 제한을 할당할 수도 있다.

[0087] 따라서, 비디오 인코더 (20) 는 더 큰 QP 값들로 양자화되는 블록들에 대한 픽셀 값들 사이에서 더 큰 변동을 요구하는 팔레트들을 정의할 수도 있으며, 더 큰 QP 값들로 양자화되는 블록들에 대한 픽셀 값들 사이에서 더 작은 변동을 요구하는 팔레트들을 정의할 수도 있다. 부가적으로, 비디오 인코더 (20) 는 테이블 (예를 들어, 매핑 테이블 또는 루업 테이블) 을 생성하고 그리고/또는 저장하여, 각각의 QP 값과 대응하는 팔레트 에러 제한 사이의 관계를 반영할 수도 있다. 이러한 방식으로, 비디오 인코더 (20) 는 각각의 QP 값과 대응하는 에러 제한 사이의 관계를 저장하기 위해 테이블을 사용함으로써 계산 효율을 향상시키기 위해 본 개시의 다양한 기술들을 구현할 수도 있다. 더욱 구체적으로, QP 값들과 대응하는 팔레트 에러 제한 사이의 관계를 저장하기 위해 테이블을 사용함으로써, 비디오 인코더 (20) 는 대응하는 팔레트 에러 제한을 유도하기 위해 각각의 팔레트에 대한 함수를 푸는 비교적 계산적으로 비용이 드는 기술들 이상의 향상된 효율을 제공하기 위해 본 명세서에 설명한 기술들을 구현할 수도 있다. 따라서, 비디오 인코더 (20) 는 본 개시의 다양한 양태들에 따라, 대응하는 블록이 양자화되는 QP 값에 기초하여 (팔레트의 에러 제한에 따라) 팔레트를 주문제작하여, 블록의 QP 값에 기초하여 블록에 대한 팔레트의 콘텐츠를 결정할 수도 있다.

[0088] 비디오 인코더 (20) 는 일부 예들에서, 양자화된 이스케이프 픽셀 유도를 위해 본 개시의 다양한 기술들을 구현할 수도 있다. 더욱 구체적으로, 비디오 인코더 (20) 는 이스케이프 픽셀에 대한 QP 의 양자화 값을 정의하기 위해 기술들을 구현할 수도 있다. 예를 들어, 팔레트-기반 코딩 기술들에 따르면, 비디오 인코더 (20) 가 팔레트-코딩된 블록에서 이스케이프 픽셀을 검출하는 경우에, 비디오 인코더 (20) 는, 대응하는 팔레트가 이스케이프 픽셀에 대한 어떠한 엔트리들도 포함하지 않기 때문에, 픽셀 값들, 또는 그것의 예측 에러를 인코딩하고 시그널링할 수도 있다. 부가적으로, 시그널링 대역폭을 보존하기 위해, 비디오 인코더 (20) 는 시그널링 이전에 이스케이프 픽셀의 인코딩된 픽셀 값을 양자화할 수도 있다.

[0089] 기존의 팔레트-기반 코딩 기술들에 따르면, 이스케이프 픽셀을 양자화하기 위해 정의되는 양자 값 (QP 값) 은 없었다. 비디오 인코더 (20) 는 이스케이프 픽셀을 양자화하는 QP 값을 정의하기 위해 본 개시의 기술들을 구현할 수도 있다. 더욱 구체적으로, 비디오 인코더 (20) 는 동일한 컬러 채널 (예를 들어, 루마 (Y), 크로마 (U, Cr), 또는 크로마 (V, Cb)) 내의 종래의 계수 인코딩에 대한 QP 값과 동일한 것으로서 이스케이프 픽셀에 대한 QP 값을 정의할 수도 있다. 일례에서, 비디오 인코더 (20) 는 동일한 컬러 채널 내의 그리고 동일

한 양자화 그룹 내의 종래의 계수 인코딩에 대한 QP 값과 동일한 것으로서 이스케이프 픽셀에 대한 QP 값을 정의할 수도 있다. 따라서, 비디오 인코더 (20)는 주어진 채널 내에서 단일 QP 값에 따라 모든 이스케이프 픽셀들을 양자화할 수도 있다. 그에 따라, 비디오 인코더 (20)가 단일 채널 내에서만 모든 이스케이프 픽셀들에 대한 QP 값을 정의할 수도 있기 때문에, 비디오 인코더 (20)는 상이한 채널들에 관하여 이스케이프 픽셀들을 양자화하는 상이한 QP 값들을 사용할 수도 있다.

[0090] 비디오 디코더 (30)는 본 개시의 다양한 기술들에 따라 이스케이프 픽셀들을 탈양자화하기 위해, 상술한 동작들에 대해 역 동작들을 수행할 수도 있다. 예를 들어, 비디오 디코더 (30)는 비디오 인코더 (20)로부터 인코딩된 비디오 비트스트림에서 수신된 정보에 기초하여, 동일한 QP 값을 사용하여 단일 채널의 모든 이스케이프 픽셀들을 탈양자화할 수도 있다. 더욱 구체적으로, 본 개시의 양태들에 따르면, 비디오 디코더 (30)는 현재 채널을 통해 통신된 블록들에 대한 종래의 변환 계수 탈양자화를 위한 QP 값에 기초하여 결정되는 QP 값을 사용하여 특정한 채널을 통해 통신된 임의의 이스케이프 픽셀들 (또는 그것의 예측 에러들/잔차 값들)을 탈양자화할 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 상이한 채널들 중에서 상이한 종래의 변환 계수 코딩을 위한 QP 값에 기초하여, 상이한 QP 값들을 사용하여 상이한 채널들을 통해 통신된 이스케이프 픽셀들을 탈양자화하기 위해 본 개시의 기술들을 구현할 수도 있다.

[0091] 이러한 방식으로, 비디오 인코더 (20) 및 비디오 디코더 (30)는 특정한 채널을 통해 통신된 모든 이스케이프 픽셀들을 (양자화하고 그리고/또는 탈양자화하기 위해) 단일 QP 값에게 정의하고 적용하도록 본 명세서에 설명한 기술들을 구현할 수도 있다. 따라서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트-기반 코딩을 통해 검출된 이스케이프 픽셀들에 대한 QP 값을 정의하기 위해 본 개시의 양태들을 적용할 수도 있고, 여기서, 기존의 팔레트-기반 코딩 기술들은 이스케이프 픽셀들에 대한 QP 값을 명시적으로 정의하지 않았다.

[0092] 부가적으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팔레트-코딩된 블록에서의 이스케이프 픽셀의 포함을 나타내고 그리고/또는 검출하기 위해 플래그를 사용하는데 본 개시의 다른 기술들을 구현할 수도 있다. 기존의 팔레트-기반 코딩 기술들에 따르면, 이스케이프 픽셀들은 "예약된" 팔레트 인덱스 값을 사용하여 시그널링되고 검출될 수도 있다. 예를 들어, 기존의 팔레트-기반 코딩 기술들에 따르면, 이스케이프 픽셀을 나타내는 예약된 팔레트 인덱스 값은 32 일 수도 있다. 더욱 구체적으로, 32의 팔레트 인덱스 값은 2개의 이스케이프 픽셀들이 상이한 픽셀 값들을 갖는지에 관계없이, 모든 이스케이프 픽셀들에 대해 사용될 수도 있다. 따라서, 기존의 팔레트-기반 코딩 기술들에 따르면, 비디오 코딩 디바이스들은 팔레트-코딩 블록들의 각각의 이스케이프 픽셀들에 대해 (32의) 5-비트 값을 사용할 수도 있다.

[0093] 비디오 인코더 (20)는 팔레트-코딩된 블록에서 이스케이프 픽셀의 표시를 시그널링하는 것에 관하여 픽처 정밀도를 유지하면서, 컴퓨팅 자원들 (예를 들어, 저장부 및 메모리)을 보존하고 대역폭 소모를 감소시키기 위해 본 개시의 기술들을 구현할 수도 있다. 예를 들어, 비디오 인코더 (20)는 팔레트-코딩된 블록에서의 픽셀이 이스케이프 픽셀인지를 나타내기 위해 플래그를 인코딩하고 시그널링할 수도 있다. 본 명세서에 설명한 바와 같이, 플래그는, 인에이블될 때, 연관된 픽셀이 "다른 인덱스"로서 지정된 팔레트 인덱스에 할당된다는 것을 나타낼 수도 있다. 비디오 인코더 (20)는 팔레트에 관하여 이스케이프 픽셀을 나타내기 위해 통상적으로 사용된 32의 팔레트 인덱스 값을 대체하기 위해 플래그의 "다른 인덱스" 상태를 사용할 수도 있다. 따라서, 비디오 인코더 (20)는 팔레트-코딩된 블록의 픽셀이 이스케이프 픽셀이라는 것을 나타내기 위해 5-비트 인덱스 값 대신에 1-비트 플래그를 인코딩하고 시그널링할 수도 있다. 결과적으로, 이스케이프 픽셀이 1-비트 플래그에 의해 나타내질 때, 비디오 인코더 (20)는 인코딩된 비디오 비트스트림에서 이스케이프 픽셀의 픽셀 값 (또는 그것의 잔차 데이터)을 인코딩하고 시그널링할 수도 있다.

[0094] 비디오 디코더 (30)는 팔레트-코딩된 블록의 픽셀이 이스케이프 픽셀이라는 것을 결정하는데 1-비트 플래그를 사용하기 위해 본 개시의 기술들을 또한 구현할 수도 있다. 다양한 예들에서, 비디오 디코더 (30)는 팔레트-코딩된 블록에서 이스케이프 픽셀을 식별하기 위해 1-비트 플래그를 사용하는데 있어서, 비디오 인코더 (20)에 관하여 상술한 인코딩 및 시그널링 동작들에 관하여 역 동작들을 수행할 수도 있다. 예를 들어, 비디오 디코더 (30)는 팔레트-코딩된 블록의 픽셀과 연관된 인에이블된 1-비트 플래그를 수신할 수도 있다. 인에이블된 상태에 있는 1-비트 플래그에 기초하여, 비디오 디코더 (30)는 연관된 픽셀에 대한 컬러 정보가 현재 블록에 대한 팔레트에 포함되지 않는다는 것을 결정할 수도 있다. 다시 말해, 비디오 디코더 (30)는 수신된 1-비트 플래그가 인에이블되면, 연관된 픽셀이 이스케이프 픽셀이라는 것을 결정할 수도 있다. 이러한 방식으로, 비디오 디코더 (30)는 팔레트-코딩된 블록에서 이스케이프 픽셀을 식별하기 위해 1-비트 플래그를 사용하여 팔레트-코딩된 블록을 복원하기 위해 본 개시의 기술들을 구현할 수도 있다. 따라서, 비디오 디코더 (30)는 팔레트-코딩된 블록들에서 이스케이프 픽셀들을 식별하는 것과 관련하여 컴퓨팅 자원들 (예를 들어,

저장부 및/또는 메모리) 및 대역폭 요건들을 보존할 수도 있다. 부가적으로, 이스케이프 팩셀이 1-비트 플래그에 의해 표시될 때, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림에서, 임의의 식별된 이스케이프 팩셀들에 대응하는 컬러 정보 (또는 그것의 잔차 데이터)를 수신할 수도 있으며, 그에 따라 팔레트-코딩된 블록을 복원할 수도 있다.

[0095] 비디오 인코더 (20) 및 비디오 디코더 (30)는 팔레트-기반 코딩에 따라 이스케이프 팩셀들의 팩셀 값들을 양자화하고 탈양자화하기 위해 본 개시의 기술들을 또한 구현할 수도 있다. 예를 들어, 비디오 인코더 (20)는 본 개시의 양태들에 따라 이스케이프 팩셀들의 팩셀 값들을 양자화함으로써 컴퓨팅 자원들 (예를 들어, 메모리 사용, 프로세서 클록 사이클 등)을 보존할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 제산 연산들을 시프트 연산들 (예를 들어, 오른쪽-시프트 연산들)로 대체함으로써 이스케이프 팩셀 값들을 양자화하기 위해 본 명세서에 설명한 기술들을 구현할 수도 있다. 더욱 구체적으로, 비디오 인코더 (20)는 대응하는 이스케이프 팩셀의 QP 값에 기초하여 특정한 오른쪽-시프트 연산을 결정할 수도 있다.

[0096] 예를 들어, 비디오 인코더 (20)는 각각의 이스케이프 팩셀의 QP 값을 오른쪽-시프트의 양에 매핑하여 팩셀 값에 적용하는 테이블을 형성할 수도 있다. 비디오 인코더 (20)는 52개의 엔트리들을 포함하는 테이블을 형성할 수도 있다. 예를 들어, 52-엔트리 매핑 테이블은 주어진 이스케이프 팩셀에 대한 각각의 가능한 QP 값에 대응하는 오른쪽-시프트 양을 제공할 수도 있다. 대안으로, 비디오 인코더 (20)는 테이블에서의 대응하는 QP 값 엔트리에 기초하여, 각각의 팩셀에 대한 오른쪽-시프트 양을 결정하기 위해 매핑 동작을 적용할 수도 있다. 매핑 함수가 팔레트-기반 코딩에 따른 이스케이프 팩셀들에 대한 기준의 양자화 기술들에 따라 사용된 52-엔트리 매핑 테이블에 비교하여, 더욱 계산적으로 효율적일 수도 있으며, 메모리 요건들을 보존할 수도 있다. 본 명세서에 설명한 바와 같이 함수를 풀어서 오른쪽-시프트 값 (피연산자)을 유도함으로써, 비디오 인코더 (20)는 52-엔트리 테이블을 저장하기 위한 비디오 디코더 (30)의 필요성을 제거하여, 비디오 디코더 (30)가 탈양자화 프로세스에 대한 저장 요건들을 감소시키면서 이스케이프 팩셀들을 탈양자화시킬 수 있게 할 수도 있다.

[0097] 다양한 예들에서, 비디오 인코더 (20)는 상술한 매핑 동작에 기초하여 이스케이프 팩셀에 대한 오른쪽-시프트 양을 결정하며, 결정된 오른쪽-시프트 양을 사용하여 이스케이프 팩셀 값에 선형 함수를 적용함으로써 이스케이프 팩셀을 양자화할 수도 있다. 비디오 인코더 (20)가 이스케이프 팩셀을 양자화하기 위해 적용할 수도 있는 선형 함수의 예가 아래와 같다:

[0098] $\text{Right_shift} = a * ((QP + b) \gg c) + d$

[0099] 여기서, a, b, c, 및 d는 모두 정수 파라미터들이다. 부가적으로, ">>" 연산자는 오른쪽-시프트 연산을 표기한다. 상기 식을 적용한 특정한 결과에서, 비디오 인코더 (20)는 이스케이프 팩셀 값에 대한 오른쪽-시프트 양이 3이라는 것을 결정할 수도 있다. 결과적인 오른쪽-시프트 연산은 $\text{Right_shift} = (QP \gg 3)$ 로서 표현될 수도 있다.

[0100] 비디오 디코더 (30)는 양자화된 이스케이프 팩셀 값을 탈양자화하기 위해, 비디오 인코더 (20)에 관하여 상술한 동작들의 역 동작들을 수행하기 위해 본 개시의 기술들을 구현할 수도 있다. 예를 들어, 비디오 디코더 (30)는 대응하는 양자화된 이스케이프 팩셀 값을 탈양자화하는데 있어서 QP 값에 기초하여 (예를 들어, 대응하는 왼쪽-시프트 연산에 대한) 시프트 양을 계산하기 위해 본 개시의 기술들을 구현할 수도 있다. 이러한 방식으로, 비디오 디코더 (30)는 52-엔트리 매핑 테이블을 저장하는 대신에 매핑 함수를 레버리징함으로써 컴퓨팅 자원들을 보존하기 위해 본 개시의 양태들을 또한 적용할 수도 있다.

[0101] 상술한 바와 같이, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 팩쳐 품질 및 데이터 정밀도를 유지하면서, 팔레트-기반 코딩에 관하여 향상된 코딩 효율을 제공하기 위해, 개별적으로든지 또는 임의의 조합으로든지 그리고/또는 순차적으로든지, 본 개시의 다양한 기술들을 구현할 수도 있다. 따라서, 본 명세서에 설명한 기술들은 팔레트-기반 비디오 코딩의 기준의 기술들 이상의 다양한 잠재적 이점들을 제공할 수도 있다. 특정한 예들에서, 상술한 바와 같이, 본 개시의 기술들은 비디오 코딩 디바이스들이 비디오 데이터의 정확성을 유지하면서, 비디오 데이터를 더욱 효율적으로 인코딩하고 그리고/또는 디코딩할 수 있게 하고 대역폭 소모를 감소시킬 수 있게 할 수도 있다.

[0102] 일부 예들에서, 본 명세서에 설명한 비디오 데이터의 팔레트-기반 코딩을 위한 기술들은 인터- 또는 인트라-예측 코딩을 위한 기술들과 같은 하나 이상의 다른 코딩 기술들과 사용될 수도 있다. 예를 들어, 더욱 상세히 후술하는 바와 같이, 인코더 또는 디코더, 또는 조합된 인코더-디코더 (코덱)가 팔레트-기반 코딩 뿐만 아니라

인터- 및 인트라-예측 코딩을 수행하도록 구성될 수도 있다.

[0103] 다양한 예들에서, 본 개시는 주요 컬러 인덱스 코딩 기술들의 상이한 양태들을 설명한다. 설명한 방법들 중 일부 또는 모두를 조합하는 것이 가능할 수도 있다.

[0104] 이제, 인덱스 예측 방향의 코딩의 예가 설명된다. 위에서 언급한 바와 같이, 각각의 인덱스에 대해, 3개의 가능한 예측 방향들: 'copy_from_top', 'copy_from_left', 및 'no_copy' 이 존재한다. 3개의 코드 워드들은 3개의 방향들에 할당되어야 한다. 예를 들어, 코드 워드들은 '0', '01', 및 '10' 일 수 있다. 위 라인에서의 병치된 픽셀 및 좌측 이웃하는 픽셀이 동일한 경우에, 2개의 코드 워드들만이 필요할 수도 있다. 예를 들어, 이러한 경우에, '0'은 '카피 없음'을 나타낼 수 있고, '1'은 상부 또는 좌측으로부터의 카피를 나타낼 수 있다.

[0105] 상술한 바와 같이, 일부 경우들에서, 비디오 블록에 대한 컬러 인덱스 맵의 코딩이 바이패스될 수도 있다. 주요 컬러들의 수가 1과 동일하고 '이스케이프 픽셀'이 존재하지 않으면, 인덱스 블록의 코딩은 바이패스될 수 있다. 이러한 원칙은 각각의 개별 컬러 컴포넌트에 적용될 수 있거나, 하나 보다 많은 컬러 컴포넌트를 포함하는 조합 인덱스에 적용될 수 있다.

[0106] 다른 예에서, 플래그 (또는 다른 타입의 선택스 엘리먼트) 가 인덱스 코딩을 바이패싱하는 이러한 특징이 사용되는지 여부를 나타내기 위해 코딩된 비트스트림에서 시그널링될 수 있다. 예를 들어, 비디오 인코더는 비디오 데이터의 코딩된 표현을 포함하는 비트스트림에서, 인덱스 코딩의 바이패싱이 사용되는지 여부를 나타내기 위해 선택스 엘리먼트 (예를 들어, 플래그) 를 시그널링할 수도 있다. 그에 따라, 비디오 디코더는 비트스트림으로부터, 인덱스 코딩의 바이패싱이 사용되는지 여부를 나타내는 선택스 엘리먼트를 획득할 수도 있다. 플래그는 SPS, PPS, 슬라이스 헤더, 또는 다른 구조, 또는 CTU 마다 또는 CU 마다 또는 임의의 다른 블록 사이즈들로 시그널링될 수도 있다.

[0107] 따라서, 일부 예들에서, 비디오 인코더는 비트스트림에서, 인덱스 블록이 비트스트림에서 시그널링되는지를 나타내는 선택스 엘리먼트를 시그널링할 수도 있다. 일부 예들에서, 비디오 인코더는 SPS, PPS, 또는 비트스트림에서의 슬라이스 헤더에서 선택스 엘리먼트를 시그널링할 수도 있다. 더욱이, 일부 예들에서, 비디오 인코더는 CTU 마다에 기초하여 또는 CU 마다에 기초하여 선택스 엘리먼트를 시그널링할 수도 있다. 일부 이러한 예들에서, 비디오 디코더는 비트스트림으로부터, 인덱스 블록이 비트스트림에서 시그널링되는지를 나타내는 선택스 엘리먼트를 획득할 수도 있다. 플래그는 SPS, PPS, 슬라이스 헤더, 또는 다른 선택스 구조, 또는 CTU 마다 또는 CU 마다 또는 임의의 다른 블록 사이즈들로 시그널링될 수 있다. 따라서, 일부 예들에서, 비디오 디코더는 SPS, PPS, 또는 비트스트림에서의 슬라이스 헤더에서 선택스 엘리먼트를 획득할 수도 있다. 더욱이, 일부 예들에서, 비디오 디코더는 CTU 마다에 기초하여 또는 CU 마다에 기초하여 선택스 엘리먼트를 획득할 수도 있다.

[0108] 이제, 인덱스들의 비트 평면 코딩의 예가 설명된다. 노멀 라인 모드에서, 인덱스가 상부 또는 좌측으로부터 예측될 수 없거나; 수평 모드에서, 라인이 현재 라인으로부터 가장 왼쪽 인덱스로부터 카피되는 경우에, 인덱스 값은 직접 코딩되어야 한다. 이러한 경우에, 인덱스 값은 인덱스 값의 2진수 표현에 따라 빈 (bin) 마다 코딩될 수도 있다. 예를 들어, 라인 i에 위치된 인덱스를 가정하면, 컬럼 j는:

$$C_{i,j} = b_{0ij} + 2b_{1ij} + \dots + 2^N b_{Nij} = [b_{0ij} b_{1ij} \dots b_{Nij}]_2 \text{에 의해 표기되고,}$$

[0110] 여기서, $b_{kij} = 0$ 또는 1이다. 그 후, b_{kij} 는 CABAC 컨텍스트들로서 b_{kij} 의 코딩된 이웃하는 인덱스 값들을 사용하여 코딩될 수 있다. 예를 들어, b_{kij} 는 컨텍스트로서 $b_{k(i-1)j} + b_{k(i-1)j}$ 를 사용할 수 있다. b_{kij} 는 단일 컨텍스트를 사용하여, 또는 어떠한 컨텍스트도 없이, 즉, 코딩을 바이패스하여 또한 코딩될 수도 있다.

[0111] 더 높은 쓰루풋을 가능하게 하기 위해, 인덱스의 빈들 중 일부가 바이패스로 코딩되며 다른 것들은 CABAC 컨텍스트들을 사용한다. 예를 들어, 표현의 최상위 빈만이 컨텍스트를 사용하는 반면에, 다른 것들은 바이패스 모드로 코딩된다.

[0112] 이제, '다른 인덱스'를 나타내기 위한 플래그의 예가 설명된다. 이러한 예에서, 1비트 플래그가 인덱스가 '다른 인덱스'인지 여부를 나타내기 위해 사용될 수 있다. 이러한 플래그는 컨텍스트로서 플래그의 주위의 코딩된 이웃 인덱스들을 갖는 CABAC를 사용하여 코딩될 수 있다.

[0113] 이제, 오른쪽 시프트용 함수를 사용하는 이스케이프 값의 양자화의 예가 설명된다. 각각의 QP로부터 오른쪽

-시프트의 양에 매핑하기 위한 테이블은 52개의 엔트리들을 요구한다. 매핑 함수는 이러한 메모리 요건들을 세이브할 수도 있으며, 오른쪽-시프트를 컴퓨팅하는데 효율적인 방식을 제공할 수도 있다. 예를 들어, 선형 함수:

[0114] $\text{Right_shift} = a * ((QP + b) \gg c) + d$

[0115] 가 적용될 수도 있고,

[0116] 여기서, a, b, c, 및 d 는 정수 파라미터들이다. 이러한 함수의 특정한 예가 아래와 같다:

[0117] $\text{Right_shift} = (QP \gg 3)$

[0118] 이제, 인덱스들의 이진화 및 코딩의 예가 설명된다. 이러한 예에서, 먼저, 플래그가 인덱스가 제로인지 여부를 나타내기 위해 컨텍스트로서 이웃 코딩된 인덱스들을 사용하여 코딩된다. 인덱스가 제로가 아니면, 인덱스는 $C > 0$ 이라는 것을 가정한다. 그 후, C-1 이 바이패스 CAVAC 코딩을 사용하여 이진화되고 코딩된다.

이진화 방법들의 예들이 단항, 절삭형 단항, 지수형 골룸, 또는 고정 또는 적응형 파라미터들을 갖는 골룸-라이스를 포함하지만, 이에 한정되지 않는다.

[0119] 이제, 이스케이프 팩셀들의 표시 플래그의 바이패스를 위한 예시적인 기술이 설명된다. 일례에서, 플래그는 팩셀이 '이스케이프 팩셀'인지 여부 (즉, 주요 컬러 테이블에 존재하는지 여부) 를 나타내기 위해 사용될 수 있다. 이러한 플래그는, 주요 컬러들의 수가 '이스케이프 팩셀' 이 존재하지 않는다는 것을 암시적인 나타내는, 주요 컬러들의 최대 수 보다 적으면 바이패스될 수 있다. 이러한 주요 컬러들의 최대 수는 미리 정의될 수 있거나 적응적으로 조절될 수 있다. 플래그가 바이패스될 때, 플래그를 나타내는 데이터는 비트스트림에 포함되지 않는다.

[0120] 예를 들어, 일부 예들에서, 비디오 인코더는 비트스트림으로부터, 블록에 대한 주요 컬러들의 수가 주요 컬러들의 최대 허용 수 보다 작은 경우에 플래그들을 나타내는 데이터를 생략할 수도 있다. 따라서, 블록에서 팩셀들에 대한 별개의 샘플 값들의 수가 주요 컬러들의 최대 허용 수 보다 작으면, 블록의 팩셀들의 별개 샘플 값들 각각에 대한 주요 컬러 테이블에 엔트리가 존재할 수 있으며, 블록의 팩셀들 중 어느 것도 이스케이프 팩셀이 아니다. 반대로, 블록에서 팩셀에 대한 별개 샘플 값들의 수가 주요 컬러들의 최대 허용 수 보다 크면, 블록의 팩셀들 중 하나 이상이 이스케이프 팩셀이다. 따라서, 블록에서 팩셀에 대한 별개 샘플 값들의 수가 주요 컬러들의 최대 허용 수 보다 크면, 비디오 인코더는 블록의 팩셀들 중 어느 것이 이스케이프 팩셀인지 나타내기 위해 플래그들을 시그널링할 수도 있다.

[0121] 일례에서, 비디오 디코더는 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림으로부터, 블록에서 팩셀들의 별개 샘플 값들의 수가 주요 컬러 테이블에서 컬러들의 최대 허용 수 보다 큰 경우에 블록에서의 팩셀이 이스케이프 팩셀인지를 나타내는 신팩스 엘리먼트를 획득할 수도 있다. 이러한 예에서, 비디오 디코더는 블록에서 팩셀들의 별개 샘플 값들의 수가 주요 컬러 테이블에서 컬러들의 최대 허용 수 보다 작은 경우에는 비트스트림으로부터 신팩스 엘리먼트를 획득하지 못한다. 팩셀이 이스케이프 팩셀이 아닌 경우에, 비디오 디코더는 팩셀에 대한 인덱스에 기초하여, 팩셀에 대한 샘플 값을 특정하는 주요 컬러 테이블에서 엔트리를 결정할 수도 있다.

[0122] 유사한 예에서, 블록에서 팩셀들의 별개 샘플 값들의 수가 주요 컬러 테이블에서 컬러들의 최대 허용 수 보다 크면, 비디오 인코더는 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림에서, 블록의 팩셀이 이스케이프 팩셀인지를 나타내는 신팩스 엘리먼트를 나타내는 데이터를 포함할 수도 있다. 블록에서 팩셀들의 별개 샘플 값들의 수가 주요 컬러 테이블에서 컬러들의 최대 허용 수 보다 작은 경우에는, 비디오 인코더는 비트스트림으로부터 신팩스 엘리먼트를 생략할 수도 있다. 팩셀이 이스케이프 팩셀이 아닌 경우에, 비디오 인코더는 비트스트림에서, 팩셀에 대한 샘플 값을 특정하는 주요 컬러 테이블에서 엔트리를 특정하는 인덱스를 나타내는 데이터를 포함할 수도 있다.

[0123] 다른 예에서, 플래그 (또는 다른 타입의 신팩스 엘리먼트) 가 이스케이프 팩셀들의 표시 플래그를 바이패싱하는 이러한 특징이 사용되는지 여부를 나타내기 위해 코딩된 비트스트림에서 시그널링될 수 있다. 예를 들어, 비디오 인코더는 코딩된 비트스트림에서, 이스케이프 팩셀의 표시 신팩스 엘리먼트 (예를 들어, 표시 플래그) 를 바이패싱하는 것이 사용되는지 여부를 나타내기 위해 신팩스 엘리먼트를 시그널링할 수도 있다. 그에 따라, 비디오 디코더는 비트스트림으로부터, 이스케이프 팩셀의 표시 신팩스 엘리먼트를 바이패싱하는 것이 사용되는지 여부를 나타내는 신팩스 엘리먼트를 획득할 수도 있다. 플래그는 SPS, PPS, 슬라이스 헤더, 또는 다

른 구조, 또는 CTU 마다 또는 CU 마다 또는 임의의 다른 블록 사이즈들로 시그널링될 수 있다.

[0124] 따라서, 일부 예들에서, 비디오 인코더는 비트스트림에서, 비트스트림이 제 1 신택스 엘리먼트 (즉, 픽셀이 이스케이프 픽셀인지를 나타내는 신택스 엘리먼트)를 포함하는지 나타내는 제 2 신택스 엘리먼트를 시그널링할 수도 있다. 더욱이, 일부 예들에서, 비디오 디코더는 비트스트림으로부터, 비트스트림이 제 1 신택스 엘리먼트 (즉, 픽셀이 이스케이프 픽셀인지를 나타내는 신택스 엘리먼트)를 포함하는지 나타내는 제 2 신택스 엘리먼트를 획득할 수도 있다. 일부 예들에서, 이러한 제 2 신택스 엘리먼트는 시퀀스 파라미터 세트, 픽처 파라미터 세트, 또는 슬라이스 헤더에서 시그널링될 수도 있다. 일부 예들에서, 제 2 신택스 엘리먼트는 CTU 마다에 기초하거나 CU 마다에 기초하여 시그널링된다.

[0125] 이제, 양자화된 이스케이프 픽셀 값들 또는 양자화된 이스케이프 예측 에러들의 예시적인 엔트로피 코딩 방법들이 설명된다. 일부 예들에서, 양자화된 이스케이프 픽셀 값들 (예측 에러들)은 고정 길이 코드워드를 사용하여 이진화된다. 코드워드의 제 1 빈에 대해, CABAC 코딩이 컨텍스트 모델링으로 적용된다. 코드워드의 나머지 빈들에 대해, CABAC 바이패스 코딩이 동일한 확률들로 적용된다. 이러한 예에서, 코드워드의 길이는 각각의 루미넌스-크로미넌스 채널 (YUV 또는 RGB)에 대한 QP 값들에 의존한다. 예를 들어, 입력 8-비트 길이 데이터를 가정하면, 스텝 사이즈 4의 양자화 이후에, 양자화된 값은 [0, 63]의 범위 내에 있으며, 따라서, 6-비트 고정 길이 코드워드가 송신될 비트들을 감소시키기 위해, 8-비트 코드워드 대신에 사용될 수도 있다.

[0126] 예를 들어, 비디오 디코더는 비디오 데이터의 픽처의 픽셀이 이스케이프 픽셀인지를 결정할 수도 있다. 픽셀이 이스케이프 픽셀이 아니라는 결정에 응답하여, 비디오 디코더는 픽셀에 대한 인덱스를 결정할 수도 있으며, 픽셀에 대한 인덱스에 기초하여, 픽셀에 대한 샘플 값을 특정하는 팔레트 엔트리를 결정할 수도 있다.

팔레트 엔트리는 샘플 값들을 특정하는 팔레트 엔트리들을 포함하는 팔레트에 있을 수도 있다. 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 디코더는 고정 길이 코드워드의 제 1 빈을 엔트로피 디코딩하기 위해 컨텍스트 모델링으로 CABAC를 사용할 수도 있다. 더욱이, 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 디코더는 고정 길이 코드워드의 제 1 빈에 후속하는 고정 길이 코드워드의 각각의 빈을 엔트로피 디코딩하기 위해 CABAC 바이패스 코딩을 사용할 수도 있다. 더욱이, 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 디코더는 픽셀에 대한 샘플 값을 결정하기 위해 고정 길이 코드워드를 역-이진화할 수도 있다. 이러한 예에서, 고정 길이 코드워드의 길이는 픽처의 각각의 채널 (예를 들어, 루미넌스, 크로미넌스 등의 채널)에 대한 양자화 파라미터 (QP) 값들에 의존한다.

[0127] 유사한 예에서, 비디오 인코더는 비디오 데이터의 픽처의 픽셀이 이스케이프 픽셀인지를 결정할 수도 있다. 픽셀은, 픽셀의 샘플 값이 샘플 값들을 특정하는 팔레트 엔트리들을 포함하는 팔레트에서 팔레트 엔트리에 의해 특정된 샘플 값에 대응할 때 이스케이프 픽셀일 수도 있다. 픽셀이 이스케이프 픽셀이 아니라는 결정에 응답하여, 비디오 인코더는 픽셀에 대한 인덱스를 결정할 수도 있으며, 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림에서, 픽셀에 대한 인덱스를 나타내는 데이터를 포함할 수도 있다. 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 인코더는 고정 길이 코드워드를 생성하기 위해 픽셀의 샘플 값을 이진화할 수도 있다. 더욱이, 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 인코더는 고정 길이 코드워드의 제 1 빈을 엔트로피 인코딩하기 위해 컨텍스트 모델링으로 CABAC를 사용할 수도 있다. 또한, 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 인코더는 고정 길이 코드워드의 제 1 빈에 후속하는 고정 길이 코드워드의 각각의 빈을 엔트로피 인코딩하기 위해 CABAC 바이패스 코딩을 사용할 수도 있다. 일부 예들에서, 고정 길이 코드워드의 길이는 픽처의 각각의 채널 (예를 들어, 루미넌스, 크로미넌스 등의 채널)에 대한 QP 값들에 의존한다.

[0128] 이제, 팔레트 에러 제한 유도의 예시적인 기술이 설명된다. 일부 예들에서, 팔레트 사이즈는 QP 와 관련된다. 예를 들어, 더 큰 팔레트 에러 제한이 더 큰 QP, 따라서, 팔레트 인덱스들의 더 작은 그룹들에 대해 할당될 수도 있고; 더 작은 팔레트 에러 제한이 더 작은 QP, 따라서, 팔레트 인덱스들의 더 많은 그룹들에 대해 할당될 수도 있다. 52개 엔트리들의 매핑 테이블 (룩업 테이블)이 각각의 QP 값과 팔레트 에러 제한 사이의 관계를 저장하기 위해 메모리에서 사용될 수도 있다.

[0129] 예를 들어, 일부 예들에서, 비디오 디코더는 비디오 데이터의 픽처의 픽셀에 대한 인덱스를 결정할 수도 있다. 더욱이, 비디오 디코더는 픽셀에 대한 인덱스에 기초하여, 팔레트에서 엔트리를 결정할 수도 있으며, 결정된 엔트리는 픽셀에 대한 샘플 값을 특정하고, 팔레트의 사이즈는 QP 와 관련된다. 유사하게, 일부 예들에서, 비디오 인코더는 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림에서, 팔레트에서 엔트리의 인덱스를 나

타내는 데이터를 포함할 수도 있고, 여기서, 팔레트의 사이즈는 양자화 파라미터와 관련된다. 일부 이러한 예들에서, 비디오 인코더는 양자화 파라미터에 기초하여, 팔레트 에러 제한을 결정할 수도 있다. 이러한 예들에서, 비디오 인코더는 비트스트림에서, 엔트리에 의해 특정된 샘플 값과 픽셀의 샘플 값 사이의 차이가 팔레트 에러 제한 미만인 경우에만 팔레트에서 엔트리의 인덱스를 나타내는 데이터를 포함할 수도 있다.

[0130] 이제, 양자화된 이스케이프 픽셀 유도의 예시적인 기술이 설명된다. 일부 예들에서, 각각의 채널의 이스케이프 픽셀 (또는 예측 에러)에 대한 양자화 파라미터는 종래의 계수 코딩을 위한 양자화 파라미터와 동일하다.

다시 말해, 이스케이프 픽셀 (예측 에러) 양자화 또는 탈양자화는 상이한 채널들에서 상이할 수도 있다. 비디오 인코더에서, 이스케이프 픽셀의 각각의 채널은 종래의 계수 코딩을 위해 양자화 파라미터를 사용한다.

비디오 디코더에서, 이스케이프 픽셀의 각각의 채널은 종래의 계수 코딩을 위해 수신된 양자화 파라미터를 사용하여 이스케이프 픽셀 값 또는 이스케이프 픽셀 예측 에러를 복원한다.

[0131] 일부 예들에서, 비디오 디코더는 비디오 데이터의 픽처의 픽셀이 이스케이프 픽셀인지를 결정할 수도 있다.

픽셀이 이스케이프 픽셀이 아니라는 결정에 응답하여, 비디오 디코더는 픽셀에 대한 인덱스에 기초하여, 샘플 값들을 특정하는 엔트리들을 포함하는 팔레트에서 엔트리를 결정할 수도 있으며, 결정된 엔트리는 픽셀의 샘플 값을 특정한다. 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 디코더는 비트스트림에서의 하나 이상의 선택스 엘리먼트들에 기초하며 그리고 팔레트에서 엔트리를 결정하지 않고, 픽셀의 샘플 값을 결정할 수도 있다. 더욱이, 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 디코더는 픽셀의 샘플 값을 탈양자화할 수도 있다. 픽셀의 샘플 값은 픽셀의 잔차 샘플 값일 수도 있으며, 비디오 디코더는 픽셀에 대한 예측 샘플 값을 픽셀의 잔차 샘플 값에 부가하여 픽셀에 대한 디코딩된 샘플 값을 결정할 수도 있다. 더욱이, 일부 이러한 예들에서, 픽셀의 샘플 값은 픽셀의 제 1 샘플 값이고, 양자화 파라미터는 제 1 양자화 파라미터이고, 픽셀의 제 1 샘플 및 제 1 양자화 파라미터는 제 1 채널에 대응한다. 이러한 예에서, 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 디코더는 제 2 양자화 파라미터에 기초하여, 픽셀의 제 2 샘플 값을 탈양자화할 수도 있고, 픽셀의 제 2 샘플 값 및 제 2 양자화 파라미터는 제 2 채널에 대응한다.

[0132] 유사한 예에서, 비디오 인코더는 비디오 데이터의 픽처의 픽셀이 이스케이프 픽셀인지를 결정할 수도 있다.

픽셀은, 픽셀의 샘플 값이 샘플 값들을 특정하는 엔트리들을 포함하는 팔레트에서 엔트리에 대응하지 않을 때 이스케이프 픽셀일 수도 있다. 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 인코더는 양자화 파라미터에 기초하여, 픽셀의 샘플 값을 양자화할 수도 있으며, 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림에서, 픽셀의 양자화된 샘플을 나타내는 데이터를 포함할 수도 있다. 픽셀이 이스케이프 픽셀이 아니라는 결정에 응답하여, 비디오 인코더는 픽셀의 샘플 값을 대응하는 팔레트에서 엔트리를 결정할 수도 있으며, 비트스트림에서, 팔레트에서의 결정된 엔트리에 인덱스를 나타내는 데이터를 포함할 수도 있다. 일부 예들에서, 픽셀에 대한 샘플 값은 픽셀의 잔차 샘플 값이다. 더욱이, 일부 예들에서, 픽셀의 샘플 값은 픽셀의 제 1 샘플 값이고, 양자화 파라미터는 제 1 양자화 파라미터이고, 픽셀의 제 1 샘플 값 및 제 1 양자화 파라미터는 제 1 채널에 대응하며, 픽셀이 이스케이프 픽셀이라는 결정에 응답하여, 비디오 인코더는 제 2 양자화 파라미터에 기초하여, 픽셀의 제 2 샘플 값을 양자화할 수도 있고, 픽셀의 제 2 샘플 값 및 제 2 양자화 파라미터는 제 2 채널에 대응한다. 비디오 인코더는 비트스트림에서, 픽셀의 양자화된 제 2 샘플 값을 나타내는 데이터를 포함할 수도 있다.

[0133] 도 2는 본 개시에 설명한 기술들을 구현할 수도 있는 예시적인 비디오 인코더(20)를 예시하는 블록도이다. 도 2는 설명의 목적을 위해 제공되며, 본 개시에 광범위하게 예시되고 설명되는 바와 같이 기술들을 한정하는 것으로 고려되어서는 안 된다. 설명의 목적을 위해, 본 개시는 HEVC 코딩의 문맥에서 비디오 인코더(20)를 설명한다. 그러나, 본 개시의 기술들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0134] 도 2의 예에서, 비디오 인코더(20)는 비디오 데이터 메모리(98), 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 버퍼(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(미도시)을 포함한다. 비디오 인코더(20)는 본 개시에 설명한 팔레트-기반 코딩 기술들의 다양한 양태들을 수행하도록 구성된 팔레트-기반 인코딩 유닛(122)을 또한 포함한다. 다른 예들에서, 비디오 인코더(20)는 더 많거나, 더 적거나, 상이한 기능적 컴포넌트들을 포함할 수도 있다.

[0135] 비디오 데이터 메모리(98)는 비디오 인코더(20)의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수

도 있다. 비디오 데이터 메모리 (98)에 저장된 비디오 데이터는 예를 들어, 비디오 소스 (18)로부터 획득될 수도 있다. 디코딩된 픽처 버퍼 (116)는 예를 들어, 인트라- 또는 인터-코딩 모드들에서, 비디오 인코더 (20)에 의해 비디오 데이터를 인코딩하는데 사용하기 위한 레퍼런스 비디오 데이터를 저장하는 레퍼런스 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (98) 및 디코딩된 픽처 버퍼 (116)는 동기식 동적 랜덤 액세스 메모리 (SDRAM), 자기 저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들을 포함하는 DRAM과 같은 각종 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (98) 및 디코딩된 픽처 버퍼 (116)는 동일한 메모리 디바이스 또는 개별 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (98)는 비디오 인코더 (20)의 다른 컴포넌트들과 온-칩 (on-chip) 일 수도 있거나, 이들 컴포넌트들에 대해 오프-칩 (off-chip) 일 수도 있다.

[0136] 비디오 인코더 (20)는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20)는 비디오 데이터의 픽처의 슬라이스에서 각각의 CTU를 인코딩할 수도 있다. CTU들 각각은 동일하게 사이징된 루마 코딩 트리 블록들 (CTB들) 및 픽처의 대응하는 CTB들과 연관될 수도 있다. CTU를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100)은 쿼드-트리 파티셔닝을 수행하여, CTU의 CTB들을 점진적으로 작아지는 블록들로 분할할 수도 있다. 더 작은 블록이 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100)은 CTU와 연관된 CTB를 4개의 동일하게 사이징된 서브-블록들로 파티셔닝할 수도 있고, 서브-블록들 중 하나 이상을 4개의 동일하게 사이징된 서브-서브-블록들로 파티셔닝할 수도 있는 등등이다.

[0137] 비디오 인코더 (20)는 CTU의 CU들을 인코딩하여 CU들의 인코딩된 표현들 (즉, 코딩된 CU들)을 생성할 수도 있다. CU를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100)은 CU의 하나 이상의 PU들 중에서 CU와 연관된 코딩 블록들을 파티셔닝할 수도 있다. 따라서, 각각의 PU는 루마 예측 블록 및 대응하는 크로마 예측 블록들과 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)는 다양한 사이즈들을 갖는 PU들을 지원할 수도 있다. 위에서 나타낸 바와 같이, CU의 사이즈는 CU의 루마 코딩 블록의 사이즈를 지칭할 수도 있으며, PU의 사이즈는 PU의 루마 예측 블록의 사이즈를 지칭할 수도 있다. 특정한 CU의 사이즈가 $2N \times 2N$ 이라는 것을 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30)는 인트라 예측을 위한 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들, 및 인터 예측을 위한 $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ 등의 대칭 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)는 인터 예측을 위한 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에 대한 비대칭 파티셔닝을 또한 지원할 수도 있다.

[0138] 인터-예측 프로세싱 유닛 (120)은 CU의 각각의 PU에 대해 인터 예측을 수행함으로써 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU의 하나 이상의 예측 샘플 블록들 및 PU에 대한 모션 정보를 포함할 수도 있다. 인터-예측 유닛 (121)은 PU가 I 슬라이스, P 슬라이스, 또는 B 슬라이스에 있는지에 따라 CU의 PU에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들은 인트라 예측된다. 따라서, PU가 I 슬라이스에 있는 경우에, 인터-예측 유닛 (121)은 PU에 대해 인터 예측을 수행하지 않는다. 따라서, I-모드에서 인코딩된 블록들에 대해, 예측 블록은 동일한 프레임 내의 이전에 인코딩된 이웃하는 블록들로부터의 공간 예측을 사용하여 형성된다.

[0139] PU가 P 슬라이스에 있는 경우에, 인터-예측 프로세싱 유닛 (120)의 모션 추정 유닛은 PU에 대한 레퍼런스 영역에 대해 레퍼런스 픽처들의 리스트 (예를 들어, "RefPicList0")에서 레퍼런스 픽처들을 검색할 수도 있다. PU에 대한 레퍼런스 영역은 레퍼런스 픽처 내에서, PU의 샘플 블록들에 가장 근접하게 대응하는 샘플 블록들을 포함하는 영역일 수도 있다. 모션 추정 유닛은 PU에 대한 레퍼런스 영역을 포함하는 레퍼런스 픽처의 RefPicList0에서 위치를 나타내는 레퍼런스 인덱스를 생성할 수도 있다. 또한, 모션 추정 유닛은 PU의 코딩 블록과 레퍼런스 영역과 연관된 레퍼런스 위치 사이의 공간 변위를 나타내는 MV를 생성할 수도 있다. 예를 들어, MV는 현재 디코딩된 픽처에서의 좌표들로부터 오프셋을 레퍼런스 픽처에서의 좌표들에 제공하는 2차원 벡터일 수도 있다. 모션 추정 유닛은 PU의 모션 정보로서 레퍼런스 인덱스 및 MV를 출력할 수도 있다. 인터-예측 프로세싱 유닛 (120)의 모션 보상 유닛은 PU의 모션 벡터에 의해 표시된 레퍼런스 위치에서 실제 또는 보간된 샘플들에 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.

[0140] PU가 B 슬라이스에 있는 경우에, 모션 추정 유닛은 PU에 대해 단방향-예측 또는 양방향-예측을 수행할 수도 있다. PU에 대해 단방향-예측을 수행하기 위해, 모션 추정 유닛은 PU에 대한 레퍼런스 영역에 대해 RefPicList0 또는 제 2 레퍼런스 픽처 리스트 ("RefPicList1")의 레퍼런스 픽처들을 검색할 수도 있다. 모션 추정 유닛은 PU의 모션 정보로서, 레퍼런스 영역을 포함하는 레퍼런스 픽처의 RefPicList0 또는 RefPicList1에서의 위치를 나타내는 레퍼런스 인덱스, PU의 샘플 블록과 레퍼런스 영역과 연관된 레퍼런스 위치 사이의 공간 변위를 나타내는 MV, 및 레퍼런스 픽처가 RefPicList0 또는 RefPicList1에 있는지를 나타내는

하나 이상의 예측 방향 표시자들을 출력할 수도 있다. 인터-예측 프로세싱 유닛 (120)의 모션 보상 유닛은 PU의 모션 벡터에 의해 표시된 레퍼런스 영역에서 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.

[0141] PU에 대해 양방향 인터-예측을 수행하기 위해, 모션 추정 유닛은 PU에 대한 레퍼런스 영역에 대해 RefPicList0에서 레퍼런스 픽처들을 검색할 수도 있으며, PU에 대한 다른 레퍼런스 영역에 대해 RefPicList1에서 레퍼런스 픽처들을 또한 검색할 수도 있다. 모션 추정 유닛은 레퍼런스 영역들을 포함하는 레퍼런스 픽처들의 RefPicList0 및 RefPicList1에서의 위치들을 나타내는 레퍼런스 픽처 인덱스들을 생성할 수도 있다.

또한, 모션 추정 유닛은 레퍼런스 영역들과 연관된 레퍼런스 위치와 PU의 샘플 블록 사이의 공간 변위를 나타내는 MV들을 생성할 수도 있다. PU의 모션 정보는 PU의 레퍼런스 인덱스들 및 MV들을 포함할 수도 있다. 모션 보상 유닛은 PU의 모션 벡터에 의해 표시된 레퍼런스 영역에서 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.

[0142] 본 개시의 다양한 예들에 따르면, 비디오 인코더 (20)는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다.

HEVC 프레임워크에 관하여, 일례로서, 팔레트-기반 코딩 기술들은 CU 모드로서 사용되도록 구성될 수도 있다.

다른 예들에서, 팔레트-기반 코딩 기술들은 HEVC의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 이에 따라, CU 모드의 문맥에서 (본 개시 전반적으로) 본 명세서에 설명한 개시된 프로세스들 모두가 추가적으로 또는 대안적으로 PU 모드에 적용될 수도 있다. 그러나, 이들 HEVC-기반 예들은, 이러한 기술들이 다른 기준의 또는 아직 개발될 시스템들/표준들과 독립적으로 또는 그 일부로서 작용하도록 적용될 수도 있기 때문에, 본 명세서에 설명하는 팔레트-기반 코딩 기술들의 제약 또는 제한으로 고려되어서는 안 된다. 이들 경우들에서, 팔레트 코딩을 위한 단위는 정사각형 블록들, 직사각형 블록들 또는 심지어 비직사각형 형상의 영역들일 수 있다.

[0143] 예를 들어, 팔레트-기반 인코딩 유닛 (122)은 팔레트-기반 인코딩 모드가 예를 들어, CU 또는 PU에 대해 선택될 때 팔레트-기반 인코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛 (122)은 픽셀 값들을 나타내는 엔트리들을 갖는 팔레트를 생성하고, 비디오 데이터의 블록의 적어도 일부 위치들의 픽셀 값들을 나타내기 위해 팔레트에서 픽셀 값들을 선택하며, 비디오 데이터의 블록의 위치들 중 적어도 일부를 선택된 픽셀 값들에 각각 대응하는 팔레트에서의 엔트리들과 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 인코딩 유닛 (122)에 의해 수행되는 것으로 설명되었지만, 이러한 기능들 중 일부 또는 모두는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0144] 팔레트-기반 인코딩 유닛 (122)은 본 명세서에 설명한 다양한 신택스 엘리먼트들 중 임의의 신택스 엘리먼트를 생성하도록 구성될 수도 있다. 이에 따라, 비디오 인코더 (20)는 본 개시에 설명한 바와 같은 팔레트-기반 코드 모드들을 사용하여 비디오 데이터의 블록들을 인코딩하도록 구성될 수도 있다. 비디오 인코더 (20)는 팔레트 코딩 모드를 사용하여 비디오 데이터의 블록을 선택적으로 인코딩할 수도 있거나, 상이한 모드, 예를 들어, HEVC 인터-예측 또는 인트라-예측 코딩 모드를 사용하여 비디오 데이터의 블록을 인코딩할 수도 있다. 비디오 데이터의 블록은 예를 들어, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU일 수도 있다. 비디오 인코더 (20)는 인터-예측 시간 예측 또는 인트라-예측 공간 코딩 모드들로 일부 블록들을 인코딩할 수도 있으며, 팔레트-기반 코딩 모드로 다른 블록들을 인코딩할 수도 있다.

[0145] 인트라-예측 프로세싱 유닛 (126)은 PU에 대해 인트라 예측을 수행함으로써 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU에 대한 예측 샘플 블록들 및 다양한 신택스 엘리먼트들을 포함할 수도 있다. 인트라-예측 프로세싱 유닛 (126)은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서 PU들에 대한 인트라 예측을 수행할 수도 있다.

[0146] PU에 대해 인트라 예측을 수행하기 위해, 인트라-예측 프로세싱 유닛 (126)은 다중 인트라 예측 모드들을 사용하여 PU에 대한 예측 데이터의 다중 세트들을 생성할 수도 있다. PU에 대한 예측 데이터의 세트를 생성하기 위해 일부 인트라 예측 모드들을 사용할 때, 인트라-예측 프로세싱 유닛 (126)은 인트라 예측 모드들과 연관된 방향들에서 PU의 예측 블록들을 가로질러 이웃하는 PU들의 샘플 블록들로부터 샘플들의 값들을 확장시킬 수도 있다. 이웃하는 PU들은, PU들, CU들, 및 CTU들에 대해 좌-우, 상-하 인코딩 순서를 가정하면, PU의 위에, PU의 위에 그리고 오른쪽에, PU의 위에 그리고 왼쪽에, 또는 PU의 왼쪽에 있을 수도 있다. 인트라-예측 프로세싱 유닛 (126)은 다양한 수의 인트라 예측 모드들, 예를 들어, 33개 방향성 인트라 예측 모드들을 사용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 수는 PU와 연관된 영역의 사이즈에 의존할 수도 있다.

- [0147] 예측 프로세싱 유닛 (100)은 PU들에 대해 인터-예측 프로세싱 유닛 (120)에 의해 생성된 예측 데이터 또는 PU들에 대해 인트라-예측 프로세싱 유닛 (126)에 의해 생성된 예측 데이터 중에서 CU의 PU들에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100)은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 CU의 PU들에 대한 예측 데이터를 선택한다. 선택된 예측 데이터의 예측 샘플 블록들은 선택된 예측 샘플 블록들로서 본 명세서에서 지칭될 수도 있다.
- [0148] 잔차 생성 유닛 (102)은 CU의 코딩 블록들 (예를 들어, 루마, Cb 및 Cr 코딩 블록들) 및 CU의 PU들의 선택된 예측 샘플 블록들 (예를 들어, 예측 루마, Cb 및 Cr 블록들)에 기초하여, CU의 잔차 블록들 (예를 들어, 루마, Cb 및 Cr 잔차 블록들)을 생성할 수도 있다. 예를 들어, 잔차 생성 유닛 (102)은, 잔차 블록들에서의 각각의 샘플이 CU의 코딩 블록에서의 샘플과 CU의 PU의 대응하는 선택된 예측 샘플에서의 대응하는 샘플 사이의 차이와 동일한 값을 갖도록, CU의 잔차 블록들을 생성할 수도 있다.
- [0149] 변환 프로세싱 유닛 (104)은 쿼드-트리 파티셔닝을 수행하여, CU와 연관된 잔차 블록들을 CU의 TU들과 연관된 변환 블록들로 파티셔닝할 수도 있다. 따라서, 일부 예들에서, TU는 루마 변환 블록 및 2개의 크로마 변환 블록들과 연관될 수도 있다. CU의 TU들의 루마 및 크로마 변환 블록들의 사이즈들 및 위치들은 CU의 PU들의 예측 블록들의 사이즈들 및 위치들에 기초할 수도 있거나 기초하지 않을 수도 있다. "잔차 쿼드-트리" (RQT)로서 공지된 쿼드-트리 구조가 영역들 각각과 연관된 노드들을 포함할 수도 있다. CU의 TU들은 RQT의 리프 노드들에 대응할 수도 있다.
- [0150] 변환 프로세싱 유닛 (104)은 하나 이상의 변환들을 TU의 변환 블록들에 적용함으로써 CU의 각각의 TU에 대한 변환 계수 블록들을 생성할 수도 있다. 변환 프로세싱 유닛 (104)은 다양한 변환들을 TU와 연관된 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104)은 이산 코사인 변환 (DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 변환 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (104)은 변환들을 변환 블록에 적용하지 않는다. 이러한 예들에서, 변환 블록은 변환 계수 블록으로서 취급될 수도 있다.
- [0151] 양자화 유닛 (106)은 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 변환 계수들 중 일부 또는 모두와 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수가 양자화 동안 m-비트 변환 계수로 내림 (round down) 될 수도 있고, 여기서 n이 m 보다 크다. 양자화 유닛 (106)은 CU와 연관된 양자화 파라미터 (QP) 값에 기초하여 CU의 TU와 연관된 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20)는 CU와 연관된 QP 값을 조절함으로써 CU와 연관된 계수 블록들에 적용된 양자화의 정도를 조절할 수도 있다. 양자화는 정보를 손실을 도입할 수도 있어서, 양자화된 변환 계수들이 원래의 계수들 보다 더 낮은 정밀도를 가질 수도 있다.
- [0152] 역양자화 유닛 (108) 및 역변환 프로세싱 유닛 (110)은 역양자화 및 역변환들을 계수 블록에 적용하여, 계수 블록으로부터 잔차 블록을 각각 복원할 수도 있다. 복원 유닛 (112)은 복원된 잔차 블록을 예측 프로세싱 유닛 (100)에 의해 생성된 하나 이상의 예측 샘플 블록들로부터의 대응하는 샘플들에 부가하여, TU와 연관된 복원된 변환 블록을 생성한다. 이러한 방식으로 CU의 각각의 TU에 대한 변환 블록들을 복원함으로써, 비디오 인코더 (20)는 CU의 코딩 블록들을 복원할 수도 있다.
- [0153] 필터 유닛 (114)은 하나 이상의 디블록킹 동작들을 수행하여, CU와 연관된 코딩 블록들에서 블록킹 아티팩터들을 감소시킬 수도 있다. 디코딩된 픽처 버퍼 (116)는, 필터 유닛 (114)이 복원된 코딩 블록들에 대해 하나 이상의 디블록킹 동작들을 수행한 이후에 복원된 코딩 블록들을 저장할 수도 있다. 인터-예측 프로세싱 유닛 (120)은 복원된 코딩 블록들을 포함하는 레퍼런스 픽처를 사용하여, 다른 픽처들의 PU들에 대한 인터 예측을 수행할 수도 있다. 또한, 인트라-예측 프로세싱 유닛 (126)은 디코딩된 픽처 버퍼 (116)에서 복원된 코딩 블록들을 사용하여, CU와 동일한 픽처에서 다른 PU들에 대해 인트라 예측을 수행할 수도 있다.
- [0154] 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 있으며, 예측 프로세싱 유닛 (100)으로부터 신택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (118)은 데이터에 대해 하나 이상의 엔트로피 인코딩 연산들을 수행하여, 엔트로피-인코딩된 데이터를 생성할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118)은 데이터에 대해 CABAC 연산, 컨텍스트-적응 가변 길이 코딩 (CAVLC) 연산, 가변-투-가변 (V2V) 길이 코딩 연산, 신택스-기반 컨텍스트-적응 바이너리 산술 코딩 (SBAC) 연산, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 연산, 지수-콜롬 인코딩 연산, 또는 다른 타입의 엔트로피 인코딩 연산을 수행할 수도 있다. 비디오 인코더 (20)는 엔트로피 인코딩 유닛 (118)에 의해 생성된

엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들어, 비트스트림은 CU에 대한 RQT를 나타내는 데이터를 포함할 수도 있다.

[0155] 일부 예들에서, 잔차 코딩은 팔레트 코딩과 함께 수행되지 않는다. 그에 따라, 비디오 인코더(20)는 팔레트 코딩 모드를 사용하여 코딩할 때 변환 또는 양자화를 수행하지 않을 수도 있다. 또한, 비디오 인코더(20)는 잔차 데이터로부터 개별적으로, 팔레트 코딩 모드를 사용하여 생성된 데이터를 엔트로피 인코딩할 수도 있다.

[0156] 본 개시의 기술들 중 하나 이상에 따르면, 비디오 인코더(20) 및 구체적으로는, 팔레트-기반 인코딩 유닛(122)은 예측된 비디오 블록들의 팔레트-기반 비디오 코딩을 수행할 수도 있다. 상술한 바와 같이, 비디오 인코더(20)에 의해 생성된 팔레트는 명시적으로 인코딩되어 비디오 디코더(30)에 전송될 수도 있고, 이전의 팔레트 엔트리들로부터 예측될 수도 있고, 이전의 픽셀 값들로부터 예측될 수도 있거나, 이들의 조합일 수도 있다.

[0157] 팔레트-기반 인코딩 유닛(122)은 본 개시의 기술들을 특정한 환경들 하에서 팔레트-기반 블록의 픽셀들에 대한 팔레트 인덱스 값들의 맵의 인코딩을 바이패스하기 위해 적용할 수도 있다. 본 개시의 양태들에 따르면, 팔레트-기반 인코딩 유닛(122)은, 팔레트 기반 인코딩 유닛(122)이 블록들의 모든 픽셀들이 동일한 컬러로 되어 있다는 것을 결정하는 경우에, 팔레트-코딩된 블록에 대한 팔레트 인덱스 값들의 맵의 인코딩을 바이패스 할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛(122)은 픽처의 팔레트-코딩된 CU가 "단일 컬러 CU"이라는 것을 결정할 수도 있고, 단일 컬러 CU에 대한 팔레트 인덱스 값들의 맵의 인코딩 및 시그널링을 바이패스할 수도 있다.

[0158] 더욱 구체적으로, 팔레트-기반 인코딩 유닛(122)은 현재 CU가 2개의 조건들을 충족시키는 경우에 팔레트-코딩된 CU가 단일 컬러 CU 인지를 결정할 수도 있다. 팔레트-기반 인코딩 유닛(122)이 팔레트-코딩된 CU가 단일 컬러 CU 인지를 결정하는데 사용할 수도 있는 제1 조건은, 대응하는 팔레트의 사이즈가 1과 동일한지이다. 팔레트의 사이즈가 1과 동일한 경우에, 팔레트-기반 인코딩 유닛(122)은 제1 조건이 단일 컬러 CU인 팔레트-코딩된 CU에 관하여 충족된다는 것을 결정할 수도 있다. 더욱 구체적으로, 팔레트 사이즈가 1인 경우에, 팔레트-기반 인코딩 유닛(122)은 팔레트가 팔레트-코딩된 CU의 (년-이스케이프) 픽셀들에 대응하는 오직 하나의 컬러만을 포함한다는 것을 결정할 수도 있다. 팔레트 사이즈가 1인 일부 예들에서, 팔레트-기반 인코딩 유닛(122)은 팔레트에 포함된 인덱스 값만이 0이라는 것을 결정할 수도 있다.

[0159] 팔레트-기반 인코딩 유닛(122)이 팔레트-코딩된 CU에 대한 팔레트 사이즈가 1이라는 것(즉, 제1 조건이 충족된다는 것)을 결정하는 경우에, 팔레트-기반 인코딩 유닛(122)은 팔레트-코딩된 CU가 단일 컬러 CU이도록 제2 조건을 충족하는지를 결정할 수도 있다. 팔레트-기반 인코딩 유닛(122)이 팔레트-코딩된 CU가 단일 컬러 CU 인지를 결정하는데 사용할 수도 있는 제2 조건은, 팔레트-코딩된 CU가 어떠한 이스케이프 픽셀들도 포함하지 않는다는 것이다. 팔레트-코딩된 CU가 적어도 하나의 이스케이프 픽셀을 포함하는 경우에, 비디오 인코더(20)는, 대응하는 팔레트가 팔레트-코딩된 CU에 관하여 오직 하나의 컬러만을 나타내더라도, 팔레트-코딩된 CU가 2개 이상의 컬러들의 픽셀들을 포함한다는 것을 결정할 수도 있다. 예를 들어, 팔레트-코딩된 CU는 팔레트에 나타난 컬러를 갖는 적어도 하나의 픽셀, 및 상이한 컬러를 갖는 적어도 하나의 이스케이프 픽셀을 포함할 수도 있다.

[0160] 팔레트-기반 인코딩 유닛(122)이 팔레트-코딩된 CU가 상술한 조건들 양자를 충족한다는 것을 결정하는 경우에, 팔레트-기반 인코딩 유닛(122)은 팔레트-코딩된 CU가 단일 컬러 CU이라는 것을 결정할 수도 있다. 더욱 구체적으로, 팔레트-코딩된 CU가 (1의 팔레트 사이즈에 의해 도시된) 단일-엔트리 팔레트와 연관되며, 팔레트-코딩된 CU가 어떠한 이스케이프 픽셀들도 포함하지 않는 경우에, 팔레트-기반 인코딩 유닛(122)은 팔레트-코딩된 CU의 모든 개별 픽셀들이 동일한 컬러(즉, 대응하는 팔레트의 단일 엔트리에 의해 표시된 컬러)로 되어 있다는 것을 결정할 수도 있다. 다양한 구현들에서, 팔레트-기반 인코딩 유닛(122)은 팔레트, 또는 하나 보다 많은 컬러 컴포넌트를 나타내는 조합된 인덱스에 관한 단일 컬러 컴포넌트 기초에 관하여 상술한 단일 컬러 CU 식별 기술들을 적용할 수도 있다.

[0161] 팔레트-코딩된 CU가 오직 하나의 컬러의 픽셀들을 포함한다는(즉, CU가 단일 컬러 CU이라는) 결정에 응답하여, 팔레트-기반 인코딩 유닛(122)은 단일 컬러 CU의 픽셀들에 대한 팔레트 인덱스 값들의 맵을 인코딩하는 것을 바이패스하기 위해 본 개시의 기술들을 구현할 수도 있다. CU에 대한 팔레트 인덱스 값들의 맵의 인코딩을 바이패스함으로써, 팔레트-기반 인코딩 유닛(122)은 컴퓨팅 자원들 및 대역폭을 보존할 수도 있고, 그렇지 않으면, 컴퓨팅 자원들 대역폭은 CU에 대한 컬러 정보를 인코딩하고 시그널링하는데 소모된다. 단일

컬러 CU 의 픽셀들에 대한 팔레트 인덱스 값들의 맵을 인코딩하는 대신에, 팔레트-기반 인코딩 유닛 (122) 은 단일 컬러 CU 를 구성하는 픽셀들의 전체 세트에 대한 컬러 정보를 더욱 효율적으로 나타내기 위해 본 개시의 기술들을 구현할 수도 있다. 팔레트-기반 인코딩 유닛 (122) 이 검출된 단일 컬러 CU 의 모든 개별 픽셀들에 대한 팔레트 인덱스의 인코딩을 바이패스하는 경우들에서, 비디오 인코더 (20) (또는 그것의 하나 이상의 컴포넌트들) 는 CU 의 픽셀들에 대한 팔레트 인덱스 값들의 맵을 시그널링할 필요가 없을 수도 있어서, 컴퓨팅 자원들을 보존하고 대역폭 소모를 감소시킨다.

[0162] 본 개시의 일부 예들에 따르면, 팔레트-기반 인코딩 유닛 (122) 은 비디오 인코더 (20) 가 단일 컬러 CU 에 대해 라인마다에 기초하여 팔레트 인덱스 값들의 인코딩 (및 시그널링) 을 바이패스하였다는 것을 나타내기 위해 플래그를 인코딩할 수도 있다. CU 의 픽셀들에 대한 팔레트 인덱스 값들의 (임의의 사이즈의) 맵 대신에 전체 CU 에 대한 1-비트 플래그를 인코딩함으로써, 팔레트-기반 인코딩 유닛 (122) 은 비디오 인코더 (20) 로 하여금 기존의 팔레트-기반 코딩 기술들에 비하여 컴퓨팅 자원들 및 시그널링 대역폭을 보존하게 할 수도 있다.

더욱이, 팔레트-기반 인코딩 유닛 (122) 은, 팔레트-기반 인코딩 유닛 (122) 에 의해 인코딩되고 CU 에 대해 비디오 인코더 (20) 에 의해 시그널링된 단일-엔트리 팔레트가 CU 의 모든 픽셀들에 대한 컬러 정보를 포함하기 때문에, 인코딩된 단일 컬러 CU 의 정밀도 및 품질을 유지할 수도 있다. 다양한 예들에서, 팔레트-기반 인코딩 유닛 (122) 은 SPS, PPS, 또는 슬라이스 헤더에서와 같은 다양한 방식들로 플래그를 인코딩할 수도 있다.

다양한 예들에서, 팔레트-기반 인코딩 유닛 (122) 은 또한, CTU 마다에 기초하여, CU 마다에 기초하여, 또는 임의의 블록 사이즈의 블록에 대해 플래그를 인코딩할 수도 있다.

[0163] 다른 예에서, 팔레트-기반 인코딩 유닛 (122) 은 팔레트-코딩된 블록에 대한 팔레트에 대해 에러 제한을 유도하기 위해 본 개시의 기술들을 구현할 수도 있다. 본 명세서에서 사용되는 바와 같이, 용어들 "에러 제한" 또는 "팔레트 에러 제한"은 팔레트의 엔트리들이 포함할 수 있는 (예를 들어, 컬러 정보에 관한) 픽셀 값들의 최소 변동을 지칭할 수도 있다. 더욱 구체적으로, 팔레트 에러 제한은 임의의 2개의 팔레트 엔트리들이 나타내야 하는 컬러 값에서의 최소 변동을 정의한다. 설명한 바와 같이, 팔레트-기반 코딩에 따라 블록을 인코딩하기 위해, 팔레트-기반 인코딩 유닛 (122) 은 블록 내에서 (픽셀 마다에 기초하여) 가장 빈번하게 발생하는 컬러 값들을 포함하도록 대응하는 팔레트를 구성할 수도 있다.

[0164] 다시 말해, 팔레트-기반 인코딩 유닛 (122) 은 팔레트에서 블록의 높은-발생 (또는 비교적 더 높은 발생) 픽셀 값들을 클러스터링함으로써 팔레트를 구성할 수도 있다. 팔레트-기반 인코딩 유닛 (122) 은 픽셀 값들이 컬러 정보에 관하여 적어도 특정한 최소 변동을 나타내도록 높은-발생 픽셀 값들을 선택할 수도 있다. 결과적으로, 높은-발생 픽셀 값들의 선택된 세트 내의 픽셀 값의 최소 변동은 대응하는 팔레트의 에러 제한을 형성할 수도 있다. 팔레트 에러 제한이 여러 픽셀 값들을 포함할 수도 있지만, 팔레트가 적어도 팔레트 에러 제한을 나타내는 픽셀 값들의 쌍 모두를 반드시 포함하지 않을 수도 있다는 것이 이해될 것이다. 따라서, 동일한 팔레트 에러 제한이 변화하는 사이즈들의 팔레트들에 적용될 수도 있다. 팔레트-기반 인코딩 유닛 (122) 은 팔레트에 포함될 컬러 값들에 관한 결정들을 하는데 있어서 팔레트 에러 제한을 사용할 수도 있다.

[0165] 팔레트-기반 인코딩 유닛 (122) 은 팔레트에 대한 에러 제한을 정의하기 위해 본 개시의 기술들을 구현할 수도 있다. 본 개시의 다양한 양태들에 따르면, 팔레트-기반 인코딩 유닛 (122) 은 팔레트-코딩된 블록에 대한 양자화 파라미터 (QP) 에 기초하여 팔레트 에러 제한을 결정할 수도 있다. 다양한 예들에서, 팔레트-기반 인코딩 유닛 (122) 은 팔레트 에러 제한이 대응하는 블록에 대한 QP 값에 정비례한다는 것을 결정할 수도 있다. 더욱 구체적으로, 이들 예들에서, 팔레트-기반 인코딩 유닛 (122) 은 더 큰 QP 값으로 양자화되는 블록에 대한 팔레트에 대해 더 큰 에러 제한, 및 더 작은 QP 값으로 양자화되는 블록에 대한 팔레트에 대해 더 작은 에러 제한을 할당할 수도 있다.

[0166] 부가적으로, 팔레트-기반 인코딩 유닛 (122) 은 팔레트-코딩된 블록의 QP 값을 대응하는 팔레트 에러 제한에 매핑하는 테이블을 저장함으로써 팔레트 에러 제한을 유도하기 위해 본 명세서에 설명한 기술들을 구현할 수도 있다. 결과적으로, 팔레트-기반 인코딩 유닛 (122) 은 대응하는 팔레트를 결정하기 위해 테이블로부터 특정한 팔레트 에러 제한 엔트리에 액세스할 수도 있다. 팔레트 에러 제한을 팔레트-코딩된 블록의 QP 에 매핑하는 테이블을 사용함으로써, 팔레트-기반 인코딩 유닛 (122) 은 기존의 팔레트-기반 코딩 기술들에 비하여 계산적 효율을 향상시킬 수도 있다. 더욱 구체적으로, 본 명세서에 설명한 매핑 테이블-기반 팔레트 에러 제한 유도 기술을 구현함으로써, 팔레트-기반 인코딩 유닛 (122) 은, 그렇지 않으면, 각각의 팔레트에 대한 에러 제한을 유도하기 위해 함수를 실행하는 것에 의존하는 기존의 기술들에 따라 소모되는 자원들을 보존할 수도 있다.

[0167] 따라서, 팔레트-기반 인코딩 유닛 (122)은 더 큰 QP 값들로 양자화되는 블록들에 대한 픽셀 값들 사이에서 더 큰 변동을 요구하는 더 큰 수의 팔레트 인덱스들을 수용할 수 있는 팔레트들을 정의할 수도 있으며, 더 큰 QP 값들로 양자화되는 픽셀 값들 사이에서 더 작은 변동을 요구하는 팔레트들을 정의할 수도 있다. 이러한 방식으로, 팔레트-기반 인코딩 유닛 (122)은 본 개시의 다양한 양태들을 구현하여, 대응하는 블록이 양자화되는 QP 값에 기초하여 (팔레트의 에러 제한에 기초하는) 팔레트를 주문제작하여, 블록의 QP 값에 기초하여 블록에 대한 팔레트의 콘텐츠를 결정할 수도 있다. 부가적으로, 팔레트-기반 인코딩 유닛 (122)은 테이블 (예를 들어, 매핑 테이블 또는 룩업 테이블)을 생성하고 그리고/또는 저장하여, 각각의 QP 값과 대응하는 팔레트에 에러 제한 사이의 관계를 반영할 수도 있다. 이러한 방식으로, 팔레트-기반 인코딩 유닛 (122)은 각각의 팔레트에 대한 에러 제한을 유도하기 위해 함수를 푸는 것에 의존하는 기존의 팔레트-코딩 기술들에서 보다 덜 계산적으로 비용이 드는 방식으로 대응하는 팔레트에 대한 팔레트 에러 제한을 유도하기 위해 본 개시의 기술들을 구현할 수도 있다.

[0168] 팔레트-기반 인코딩 유닛 (122)은 일부 예들에서, 양자화된 이스케이프 픽셀 유도를 위해 본 개시의 다양한 기술들을 구현할 수도 있다. 더욱 구체적으로, 팔레트-기반 인코딩 유닛 (122)은 이스케이프 픽셀에 대한 QP의 양자화 값을 정의하기 위해 기술들을 구현할 수도 있다. 예를 들어, 팔레트-기반 코딩 기술들에 따르면, 팔레트-기반 인코딩 유닛 (122)이 팔레트-코딩된 블록에서 이스케이프 픽셀을 검출하는 경우에, 팔레트-기반 인코딩 유닛 (122)은, 대응하는 팔레트가 이스케이프 픽셀에 대한 어떠한 엔트리들도 포함하지 않기 때문에, 픽셀 값들, 또는 그것의 예측 에러를 인코딩할 수도 있다. 부가적으로, 시그널링 대역폭을 보존하기 위해, 양자화 유닛 (106)은 비디오 인코더 (20)의 다른 컴포넌트들에 의한 시그널링 이전에 이스케이프 픽셀의 인코딩된 픽셀 값을 양자화할 수도 있다.

[0169] 기존의 팔레트-기반 코딩 기술들에 따르면, 이스케이프 픽셀을 양자화하기 위해 정의되는 양자 값 (QP 값)은 없었다. 팔레트-기반 인코딩 유닛 (122)은 이스케이프 픽셀을 양자화하는 QP 값을 정의하기 위해 본 개시의 기술들을 구현할 수도 있다. 더욱 구체적으로, 팔레트-기반 인코딩 유닛 (122)은 동일한 컬러 채널 내의 종래의 계수 인코딩에 대한 QP 값과 동일한 것으로서 이스케이프 픽셀에 대한 QP 값을 정의할 수도 있다. 따라서, 팔레트-기반 인코딩 유닛 (122)은 주어진 컬러 채널 내에서, 단일 QP 값에 따라 모든 이스케이프 픽셀들을 양자화할 수도 있다. 부가적으로, 팔레트-기반 인코딩 유닛 (122)이 단일 컬러 채널 내에서만 모든 이스케이프 픽셀들에 대한 QP 값을 정의할 수도 있기 때문에, 팔레트-기반 인코딩 유닛 (122)은 상이한 컬러 채널들에 관하여 이스케이프 픽셀들을 양자화하는 상이한 QP 값을 사용할 수도 있다.

[0170] 부가적으로, 팔레트-기반 인코딩 유닛 (122)은 팔레트-코딩된 블록에서의 이스케이프 픽셀의 포함을 나타내고 그리고/또는 검출하기 위해 플래그를 사용하는데 본 개시의 다른 기술들을 구현할 수도 있다. 기존의 팔레트-기반 코딩 기술들에 따르면, 이스케이프 픽셀들은 "예약된" 팔레트 인덱스 값을 사용하여 시그널링되고 검출될 수도 있다. 예를 들어, 기존의 팔레트-기반 코딩 기술들에 따르면, 이스케이프 픽셀을 나타내는 예약된 팔레트 인덱스 값은 32 일 수도 있다. 더욱 구체적으로, 32 의 팔레트 인덱스 값은 2개의 이스케이프 픽셀들이 상이한 픽셀 값을 갖는지에 관계없이, 모든 이스케이프 픽셀들에 대해 사용될 수도 있다. 따라서, 기존의 팔레트-기반 코딩 기술들에 따르면, 비디오 코딩 디바이스들은 팔레트-코딩 블록들의 각각의 이스케이프 픽셀들에 대해 (32 의) 5-비트 값을 사용할 수도 있다.

[0171] 팔레트-기반 인코딩 유닛 (122)은 팔레트-코딩된 블록에서 이스케이프 픽셀의 표시를 시그널링하는 것에 관하여 픽처 정밀도를 유지하면서, 컴퓨팅 자원들 (예를 들어, 저장부 및 메모리)을 보존하고 대역폭 소모를 감소시키기 위해 본 개시의 기술들을 구현할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛 (122)은 팔레트-코딩된 블록에서의 픽셀이 이스케이프 픽셀인지를 나타내기 위해 플래그 (예를 들어, 1-비트 데이터 유닛)를 인코딩할 수도 있다. 본 명세서에 설명한 바와 같이, 플래그는, 인에이블될 때, 연관된 픽셀이 "다른 인덱스"로서 지칭된 팔레트 인덱스에 할당된다는 것을 나타낼 수도 있다. 팔레트-기반 인코딩 유닛 (122)은 팔레트에 관하여 이스케이프 픽셀을 나타내기 위해 통상적으로 사용된 32 의 팔레트 인덱스 값을 대체하기 위해 플래그의 "다른 인덱스" 상태를 사용할 수도 있다. 따라서, 팔레트-기반 인코딩 유닛 (122)은 팔레트-코딩된 블록의 픽셀이 이스케이프 픽셀이라는 것을 나타내기 위해 5-비트 인덱스 값 대신에 1-비트 플래그를 인코딩할 수도 있다 (그리고 비디오 인코더 (20)의 다른 컴포넌트들은 1-비트 플래그를 시그널링할 수도 있다). 결과적으로, 비디오 인코더 (20) (또는 그것의 하나 이상의 컴포넌트들)는 인코딩된 비디오 비트스트림에서, 픽셀 값, 또는 그것의 잔차 데이터를 인코딩하고 시그널링할 수도 있다.

[0172] 팔레트-기반 인코딩 유닛 (122) 및 양자화 유닛 (106) 중 하나 또는 양자는 팔레트-기반 코딩에 따라 이스케이프 픽셀들의 픽셀 값들을 양자화하기 위해 본 개시의 기술들을 또한 구현할 수도 있다. 예를 들어, 팔레트-

기반 인코딩 유닛 (122) 및/또는 양자화 유닛 (106)은 본 개시의 양태들에 따라 이스케이프 픽셀들의 픽셀 값들을 양자화함으로써 컴퓨팅 자원들 (예를 들어, 메모리 사용, 프로세서 클록 사이클 등)을 보존할 수도 있다.

일부 예들에서, 양자화 유닛 (106)은 제산 연산들을 시프트 연산들 (예를 들어, 오른쪽-시프트 연산들)로 대체함으로써 이스케이프 픽셀 값들을 양자화하기 위해 본 명세서에 설명한 기술들을 구현할 수도 있다. 더욱 구체적으로, 양자화 유닛 (106)은 대응하는 이스케이프 픽셀의 QP 값에 기초하여 특정한 오른쪽-시프트 연산을 결정할 수도 있다. 예를 들어, 양자화 유닛 (106)은 파라미터 또는 피연산자로서 QP 값을 포함하는 함수를 풀어서 오른쪽-시프트 값을 유도할 수도 있다.

[0173] 예를 들어, 기존의 기술들에 따르면, 팔레트-기반 인코딩 유닛 (122)은 각각의 이스케이프 픽셀의 QP 값을 오른쪽-시프트의 양에 매핑하여 픽셀 값에 적용하는 테이블을 형성할 수도 있다. 팔레트-기반 인코딩 유닛 (122)은 52개의 엔트리들을 포함하는 테이블을 형성할 수도 있다. 예를 들어, 52-엔트리 매핑 테이블은 주어진 이스케이프 픽셀에 대한 각각의 가능한 QP 값에 대응하는 오른쪽-시프트 양을 제공할 수도 있다. 결과적으로, 양자화 유닛 (106)은 테이블에서의 대응하는 QP 값 엔트리에 기초하여, 각각의 픽셀에 대한 오른쪽-시프트 양을 결정하기 위해 매핑 동작을 적용할 수도 있다.

[0174] 다양한 예들에서, 양자화 유닛 (106)은 오른쪽-시프트 양을 결정하기 위해 이스케이프 픽셀 값에 선형 함수를 적용하는 것에 기초하여 이스케이프 픽셀에 대한 오른쪽-시프트 양을 결정함으로써 이스케이프 픽셀을 양자화하기 위해 본 개시의 다양한 기술들을 구현할 수도 있다. 양자화 유닛 (106)이 이스케이프 픽셀을 양자화하는 오른쪽-시프트 양을 유도하기 위해 적용할 수도 있는 선형 함수의 예가 아래와 같다:

[0175] $\text{Right_shift} = a * ((QP + b) \gg c) + d$

[0176] 여기서, a, b, c, 및 d는 모두 정수 파라미터들이다. 부가적으로, ">>" 연산자는 오른쪽-시프트 연산을 표기한다. 상기 식을 적용한 특정한 결과에서, 양자화 유닛 (106)은 이스케이프 픽셀 값에 대한 오른쪽-시프트 양이 3이라는 것을 결정할 수도 있다. 결과적인 오른쪽-시프트 연산은 $\text{Right_shift} = (QP \gg 3)$ 로서 표현될 수도 있다. 함수 (예를 들어, 상술한 선형 함수)를 풀어서, 양자화 유닛 (106)은 디코딩 디바이스 (예를 들어, 비디오 디코더 (30) 및/또는 그것의 다양한 컴포넌트들)가, 그렇지 않으면 52-엔트리 테이블을 저장하기 위해 소모되는 저장 자원들을 보존하면서, 이스케이프 픽셀을 탈양자화할 수 있게 할 수도 있다. 이러한 방식으로, 양자화 유닛 (106)은 비디오 디코더 (30)에 관한 저장 요건들을 완화하면서 이스케이프 픽셀을 양자화하기 위해 본 개시의 기술들을 구현할 수도 있다.

[0177] 본 개시에 설명한 기술들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 예측하는 것, 팔레트들을 유도하는 것, 또는 팔레트-기반 코딩 맵들 및 다른 신택스 엘리먼트들을 송신하는 것 중 하나 이상의 다양한 조합들에 대한 기술들을 또한 포함할 수도 있다.

[0178] 도 3은 본 개시의 기술들을 구현하도록 구성된 예시적인 비디오 디코더 (30)를 예시하는 블록도이다. 도 3은 설명의 목적을 위해 제공되며, 본 개시에 광범위하게 예시되고 설명되는 바와 같이 기술들을 한정하는 것이 아니다. 설명의 목적을 위해, 본 개시는 HEVC 코딩의 문맥에서 비디오 디코더 (30)를 설명한다. 그러나, 본 개시의 기술들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0179] 도 3의 예에서, 비디오 디코더 (30)는 비디오 데이터 메모리 (148), 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 본 개시에 설명한 팔레트-기반 코딩 기술들의 다양한 양태들을 수행하도록 구성된 팔레트-기반 디코딩 유닛 (165)을 또한 포함한다. 다른 예들에서, 비디오 디코더 (30)는 더 많거나, 더 적거나, 상이한 기능적 컴포넌트들을 포함할 수도 있다.

[0180] 비디오 데이터 메모리 (148)는 비디오 디코더 (30)의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (148)에 저장된 비디오 데이터는, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체에 액세스함으로써, 예를 들어, 컴퓨터-판독가능 매체 (16)로부터, 예를 들어, 카메라와 같은 로컬 비디오 소스로부터 획득될 수도 있다. 비디오 데이터 메모리 (148)는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (CPB)를 형성할 수도 있다. 디코딩된 픽처 버퍼 (162)는 예를 들어, 인트라- 또는 인터-코딩 모드들에서, 비디오 디코더 (30)에 의해 비디오 데이터를 디코딩하는데 사용하기 위한 레퍼런스 비디오 데이터를 저장하는 레퍼런스 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (148) 및 디코딩된 픽처

버퍼 (162)는 동기식 동적 랜덤 액세스 메모리 (SDRAM), 자기 저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들을 포함하는 DRAM과 같은 각종 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (148) 및 디코딩된 픽처 버퍼 (162)는 동일한 메모리 디바이스 또는 개별 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (148)는 비디오 디코더 (30)의 다른 컴포넌트들과 온-칩 (on-chip) 일 수도 있거나, 이를 컴포넌트들에 대해 오프-칩 (off-chip) 일 수도 있다.

[0181] 비디오 데이터 메모리 (148), 즉, CPB는 비트스트림의 인코딩된 비디오 데이터 (예를 들어, NAL 유닛들)을 수신하고 저장할 수도 있다. 엔트로피 디코딩 유닛 (150)은 비디오 데이터 메모리 (148)로부터 인코딩된 비디오 데이터 (예를 들어, NAL 유닛들)를 수신할 수도 있으며, NAL 유닛들을 파싱하여 신택스 엘리먼트들을 디코딩할 수도 있다. 엔트로피 디코딩 유닛 (150)은 NAL 유닛들에서 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 획득된 (예를 들어, 추출된) 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0182] 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 것의 일부로서, 엔트로피 디코딩 유닛 (150)은 코딩된 슬라이스 NAL 유닛들로부터 신택스 엘리먼트들을 추출하고 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관한 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 신택스 엘리먼트들은 슬라이스를 포함하는 픽처와 연관된 PPS를 식별하는 신택스 엘리먼트를 포함할 수도 있다.

[0183] 비트스트림으로부터 신택스 엘리먼트들을 디코딩하는 것에 부가하여, 비디오 디코더 (30)는 넌-파티셔닝된 CU에 대한 복원 동작을 수행할 수도 있다. 넌-파티셔닝된 CU에 대한 복원 동작을 수행하기 위해, 비디오 디코더 (30)는 CU의 각각의 TU에 대해 복원 동작을 수행할 수도 있다. CU의 각각의 TU에 대해 복원 동작을 수행함으로써, 비디오 디코더 (30)는 CU의 잔차 블록들을 복원할 수도 있다.

[0184] CU의 TU에 대해 복원 동작을 수행하는 것의 일부로서, 역양자화 유닛 (154)은 TU와 연관된 계수 블록들을 역양자화, 즉, 탈양자화할 수도 있다. 역양자화 유닛 (154)은 양자화의 정도, 그리고 유사하게, 적용할 역양자화 유닛 (154)에 대한 역양자화의 정도를 결정하기 위해 TU의 CU와 연관된 QP를 사용할 수도 있다. 즉, 압축비, 즉, 원래의 시퀀스를 나타내기 위해 사용된 비트들의 수와 압축된 것의 비율은 변환 계수들을 양자화할 때 사용된 QP의 값을 조절함으로써 제어될 수도 있다. 압축비는 이용된 엔트로피 코딩의 방법에 또한 의존할 수도 있다.

[0185] 역양자화 유닛 (154)이 계수 블록을 역양자화한 이후에, 역변환 프로세싱 유닛 (156)은 TU와 연관된 잔차 블록을 생성하기 위해 하나 이상의 역변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (156)은 역 DCT, 역 정수 변환, 역 카루넨 루베 변환 (KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역변환을 계수 블록에 적용할 수도 있다.

[0186] PU가 인트라 예측을 사용하여 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 사용하여, 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.

[0187] 예측 프로세싱 유닛 (152)은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 제 1 레퍼런스 픽처 리스트 (RefPicList0) 및 제 2 레퍼런스 픽처 리스트 (RefPicList1)를 구성할 수도 있다. 더욱이, PU가 인터 예측을 사용하여 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 레퍼런스 영역들을 결정할 수도 있다. 모션 보상 유닛 (164)은, PU에 대한 하나 이상의 레퍼런스 블록들에서의 샘플 블록들에 기초하여, PU에 대한 예측 블록들 (예를 들어, 예측 루마, Cb 및 Cr 블록들)을 생성할 수도 있다.

[0188] 복원 유닛 (158)은 적용가능하면, CU의 TU들과 연관된 변환 블록들 (예를 들어, 루마, Cb 및 Cr 변환 블록들) 및 CU의 PU들의 예측 블록들 (예를 들어, 루마, Cb 및 Cr 블록들), 즉, 인트라-예측 데이터 또는 인터-예측 데이터를 사용하여, CU의 코딩 블록들 (예를 들어, 루마, Cb 및 Cr 코딩 블록들)을 복원할 수도 있다. 예를

들어, 복원 유닛 (158)은 변환 블록들 (예를 들어, 루마, Cb 및 Cr 변환 블록들)의 샘플들을 예측 블록들 (예를 들어, 예측 루마, Cb 및 Cr 블록들)의 대응하는 샘플들에 부가하여, CU의 코딩 블록들들 (예를 들어, 루마, Cb 및 Cr 코딩 블록들)을 복원할 수도 있다.

[0189] 필터 유닛 (160)은 디블록킹 동작을 수행하여, CU의 코딩 블록들 (예를 들어, 루마, Cb 및 Cr 코딩 블록들)과 연관된 블록킹 아티팩트들을 감소시킬 수도 있다. 비디오 디코더 (30)는 CU의 코딩 블록들 (예를 들어, 루마, Cb 및 Cr 코딩 블록들)을 디코딩된 픽처 버퍼 (162)에 저장할 수도 있다. 디코딩된 픽처 버퍼 (162)는 후속 모션 보상, 인트라 예측, 및 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상의 프리젠테이션을 위해 레퍼런스 픽처들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30)는 디코딩된 픽처 버퍼 (162)에서의 블록들 (예를 들어, 루마, Cb 및 Cr 블록들)에 기초하여, 다른 CU들의 PU들에 대한 인트라 예측 또는 인터 예측을 수행할 수도 있다. 이러한 방식으로, 비디오 디코더 (30)는 비트스트림으로부터, 유효 계수 블록의 변환 계수 레벨들을 추출할 수도 있고, 변환 계수 레벨들을 역 양자화할 수도 있고, 변환을 변환 계수 레벨들에 적용하여 변환 블록을 생성할 수도 있고, 변환 블록에 적어도 부분적으로 기초하여 코딩 블록을 생성할 수도 있으며, 디스플레이를 위한 코딩 블록을 출력할 수도 있다.

[0190] 본 개시의 다양한 예들에 따르면, 비디오 디코더 (30)는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165)은 팔레트-기반 디코딩 모드가 예를 들어, CU 또는 PU에 대해 선택될 때 팔레트-기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165)은 픽셀 값들을 나타내는 엔트리들을 갖는 팔레트를 생성하도록 구성될 수도 있다. 더욱이, 이러한 예에서, 팔레트-기반 디코딩 유닛 (165)은 비디오 데이터의 블록의 적어도 일부 위치들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신할 수도 있다. 이러한 예에서, 팔레트-기반 디코딩 유닛 (165)은 정보에 기초하여 팔레트에서 픽셀 값들을 선택할 수도 있다. 부가적으로, 이러한 예에서, 팔레트-기반 디코딩 유닛 (165)은 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 복원할 수도 있다. 다양한 기능들이 팔레트-기반 디코딩 유닛 (165)에 의해 수행되는 것으로 설명되었지만, 이러한 기능들 중 일부 또는 모두는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0191] 팔레트-기반 디코딩 유닛 (165)은 팔레트 코딩 모드 정보를 수신할 수도 있으며, 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용된다는 것을 나타낼 때 상기 동작들을 수행할 수도 있다. 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용되지 않는다는 것을 나타낼 때, 또는 다른 모든 정보가 상이한 모드의 사용을 나타낼 때, 팔레트-기반 디코딩 유닛 (165)은 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용되지 않는다는 것을 나타낼 때, HEVC 인터-예측 또는 인트라-예측 코딩 모드와 같은 넌-팔레트 기반 코딩 모드를 사용하여 비디오 데이터의 블록을 디코딩한다. 비디오 데이터의 블록은 예를 들어, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU 일 수도 있다. 비디오 디코더 (30)는 인터-예측 시간 예측 또는 인트라-예측 공간 코딩 모드들로 일부 블록들을 디코딩할 수도 있으며, 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.

[0192] 본 개시의 기술들 중 하나 이상에 따르면, 비디오 디코더 (30) 및 구체적으로는, 팔레트-기반 디코딩 유닛 (165)은 팔레트-코딩된 비디오 블록들의 팔레트-기반 비디오 디코딩을 수행할 수도 있다. 상술한 바와 같이, 비디오 디코더 (30)에 의해 디코딩된 팔레트는 비디오 인코더 (20)에 의해 명시적으로 인코딩 및 시그널링될 수도 있고, 수신된 팔레트-코딩된 블록에 관하여 비디오 블록 (30)에 의해 복원될 수도 있고, 이전의 팔레트 엔트리들로부터 예측될 수도 있고, 이전의 픽셀 값들로부터 예측될 수도 있거나, 이들의 조합일 수도 있다.

[0193] 상술한 바와 같이, 팔레트-코딩된 비디오 블록이 조건들의 특정한 세트를 충족하면, 비디오 인코더 (20) (및 팔레트-기반 인코딩 유닛 (122)과 같은, 그것의 다양한 컴포넌트들)는 블록의 픽셀들에 대한 팔레트 인덱스 값들의 맵의 인코딩 및 시그널링을 바이패스할 수도 있다. 비디오 인코더 (20)가 팔레트-코딩된 단일 컬러 블록에 대한 팔레트 인덱스 값들의 맵의 인코딩 및 시그널링을 바이패스하는 예들에서, 비디오 디코더 (30) (및 팔레트-기반 디코딩 유닛 (165)과 같은, 그것의 특정 컴포넌트들)는 단일 컬러 블록을 복원하기 위해 본 개시의 다양한 기술들을 적용할 수도 있다. 일부 예들에서, 팔레트-기반 디코딩 유닛 (165)은 비디오 인코더 (20)의 팔레트-기반 인코딩 유닛 (122)에 관하여 상술한 것들에 대해 역인 동작들을 수행하여, 팔레트-코딩된 블록이 단일 컬러 블록이라는 것을 결정할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165)은 현재 블록에 대한 팔레트가 1의 사이즈를 갖는다는 것을 결정하여, 블록이 제 1 조건을 충족하여 단일 컬러 블록으로서 적합하다는 것을 결정할 수도 있다. 다양한 예들에서, 비디오 디코더 (30)는 비디오 인코더 (20)로

부터 인코딩된 비트스트림에서 팔레트를 수신할 수도 있거나, 인코딩된 비디오 비트스트림에 포함된 다양한 다른 데이터를 사용하여 팔레트를 유도할 수도 있다.

[0194] 부가적으로, 비디오 디코더 (30) 는 블록이 어떠한 이스케이프 픽셀들도 포함하지 않는다는 것을 결정하여, 블록이 제 2 조건을 충족하여 단일 컬러 블록으로서 적합하다는 것을 결정할 수도 있다. 블록에 대한 팔레트의 사이즈가 1 이라는 것 (따라서 제 1 조건을 충족함) 과, 블록이 어떠한 이스케이프 픽셀들도 포함하지 않는다는 것 (따라서 제 2 조건을 충족함) 을 결정하는 것에 기초하여, 팔레트-기반 디코딩 유닛 (165) 은 현재 블록이 단일 컬러 블록이라는 것을 결정하기 위해 본 개시의 기술들을 구현할 수도 있다. 결과적으로, 팔레트-기반 디코딩 유닛 (165) 은 대응하는 팔레트의 단일 엔트리에 표시된 컬러 정보에 따라 단일 컬러 블록의 모든 픽셀들을 복원함으로써 단일 컬러 블록을 복원할 수도 있다. 이러한 방식으로, 팔레트-기반 디코딩 유닛 (165) 은, 그렇지 않으면 블록의 모든 픽셀들에 대한 팔레트 인덱스 값들의 맵에 의존함으로써 블록을 복원하도록 요구되는 컴퓨팅 자원들 및 대역폭을 보존하면서, 팔레트-코딩된 블록을 정확하게 복원하기 위해 본 개시의 기술들을 구현할 수도 있다.

[0195] 일부 예들에서, 비디오 디코더 (30) 는 본 개시의 기술들에 따라, 비디오 인코더 (20) 가 팔레트-인코딩된 블록의 하나 이상의 픽셀들에 대한 팔레트 인덱스 값들의 맵의 인코딩 및 시그널링을 바이패스하였는지를 나타내는 플래그를, 인코딩된 비디오 비트스트림에서 수신할 수도 있다. 비디오 인코더 (20) 가 팔레트-인코딩된 블록에 대한 팔레트 인덱스 값들의 맵의 인코딩 및 시그널링을 바이패스하였다는 것을 나타내는 플래그를 비디오 디코더 (30) 가 수신하는 경우들에서, 팔레트-기반 디코딩 유닛 (165) 은 현재 블록이 팔레트-코딩되며, 단일 컬러 블록이라는 것을 결정하기 위해 본 개시의 기술들을 구현할 수도 있다. 더욱 구체적으로, 플래그가 인에이블되는 경우에 (예를 들어, 1 의 값으로 설정되는 경우에), 팔레트-기반 디코딩 유닛 (165) 은 팔레트-코딩된 블록이 단일 컬러 블록이라는 것을 결정할 수도 있다. 결과적으로, 팔레트-기반 디코딩 유닛 (165) 은 블록에 대한 팔레트에서의 단일 엔트리의 컬러 정보에 따라 블록의 모든 픽셀들을 복원하기 위해 본 개시의 기술들을 구현할 수도 있다. 따라서, 팔레트-기반 디코딩 유닛 (165) 은, 블록의 상이한 픽셀들, 또는 픽셀들의 그룹들 (예를 들어, 라인) 에 대한 (변화하는 비트깊이의) 개별 인덱스 값들을 사용하기 보다는, 전체 블록에 대한 1-비트 플래그를 사용하여 팔레트-인코딩된 블록을 정확하게 복원하기 위해 본 개시의 기술들을 구현할 수도 있다. 이러한 방식으로, 팔레트-기반 디코딩 유닛 (165) 은 단일 컬러 팔레트-코딩된 블록들을 복원하는데 있어서 비디오 디코더 (30) 에서의 컴퓨팅 자원 소모를 보존할 수도 있으며, 정밀도 및 픽쳐 품질을 유지하면서 단일 컬러 팔레트 코딩된 블록들을 복원하는데 필요한 데이터를 수신하기 위해 비디오 디코더 (30) 에 의해 요구된 대역폭을 감소시킬 수도 있다.

[0196] 상술한 바와 같이, 비디오 인코더 (20) (및 팔레트-기반 인코딩 유닛 (122) 및/또는 양자화 유닛 (106) 과 같은, 그것의 컴포넌트들) 은 강화된 컴퓨팅 효율로 팔레트-코딩된 블록의 이스케이프 픽셀 값들을 양자화하기 위해 본 개시의 특정한 기술들을 구현할 수도 있다. 비디오 디코더 (30) (및 팔레트-기반 디코딩 유닛 (165) 및/또는 역양자화 유닛 (154) 과 같은, 그것의 다양한 컴포넌트들) 는 본 개시의 다양한 기술들에 따라 이스케이프 픽셀들을 탈양자화하기 위해, 비디오 인코더 (20) 에 관하여 상술한 동작들에 대해 역 동작들을 수행할 수도 있다. 예를 들어, 역양자화 유닛 (154) 은 비디오 인코더 (20) 로부터 인코딩된 비디오 비트스트림에서 수신된 정보에 기초하여, 동일한 QP 값을 사용하여 단일 컬러 채널의 모든 이스케이프 픽셀들을 탈양자화할 수도 있다. 더욱 구체적으로, 본 개시의 양태들에 따르면, 역양자화 유닛 (154) 은 현재 컬러 채널을 통해 통신된 블록들에 대한 종래의 변환 계수 탈양자화를 위한 QP 값에 기초하여 결정되는 QP 값을 사용하여 특정한 컬러 채널을 통해 통신된 임의의 이스케이프 픽셀들 (또는 그것의 예측 예러들/잔차 값들) 을 탈양자화할 수도 있다. 일부 예들에서, 역양자화 유닛 (154) 은 상이한 채널들 중에서 상이한 종래의 변환 계수 코딩을 위해 사용된 QP 값에 기초하여, 상이한 QP 값들을 사용하여 상이한 컬러 채널들을 통해 통신된 이스케이프 픽셀들을 탈양자화하기 위해 본 개시의 기술들을 구현할 수도 있다.

[0197] 이러한 방식으로, 비디오 디코더 (30) 는 특정한 컬러 채널을 통해 통신된 모든 이스케이프 픽셀들을 (탈양자화하기 위해) 단일 QP 값에게 정의하고 적용하도록 본 명세서에 설명한 기술들을 구현할 수도 있다. 따라서, 비디오 디코더 (30) 는 팔레트-기반 코딩을 통해 검출된 이스케이프 픽셀들에 대한 QP 값을 정의하기 위해 본 개시의 양태들을 적용할 수도 있고, 여기서, 기준의 팔레트-기반 코딩 기술들은 이스케이프 픽셀들에 대한 QP 값을 정의하지 않았다.

[0198] 일부 예들에서, 역양자화 유닛 (154) 과 같은 비디오 디코더 (30) 의 컴포넌트들은, 양자화된 이스케이프 픽셀 값을 탈양자화하기 위해, 비디오 인코더 (20) (및/또는 양자화 유닛 (106) 과 같은 그것의 컴포넌트들) 에 관하여 상술한 동작들의 역 동작들을 수행하도록 본 개시의 기술들을 구현할 수도 있다. 예를 들어, 역양자화

유닛 (154)은 대응하는 양자화된 이스케이프 픽셀 값을 탈양자화하는데 있어서 QP 값에 기초하여 (예를 들어, 대응하는 왼쪽-시프트 연산에 대한) 시프트 양을 계산하기 위해 본 개시의 기술들을 구현할 수도 있다. 이러한 방식으로, 역양자화 유닛 (154)은 52-엔트리 매핑 테이블을 저장하는 대신에 함수를 레버리징함으로써, 저장 활용과 같은 컴퓨팅 자원들을 보존하기 위해 본 개시의 양태들을 또한 적용할 수도 있다.

[0199]

도 4는 비디오 디코딩 디바이스가 조건들의 특정한 세트에 기초하여, 팔레트-코딩된 블록의 픽셀들에 대한 인덱스 값들의 디코딩을 바이패스하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스 (180)를 예시하는 플로우차트이다. 프로세스 (180)가 본 개시의 양태들에 따라 각종 디바이스들에 의해 수행될 수도 있지만, 프로세스 (180)는 설명의 용이함을 위해 도 1 및 도 3의 비디오 디코더 (30)에 관하여 본 명세서에 설명된다. 프로세스 (180)는, 비디오 디코더 (30)가 디코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트에 포함된 엔트리들의 수를 결정할 때 시작할 수도 있다 (182). 부가적으로, 비디오 디코더 (30)는 비디오 데이터의 블록이 팔레트의 엔트리들 중 어느 것과도 연관되지 않은 적어도 하나의 이스케이프 픽셀을 포함하는지 여부를 결정할 수도 있다 (184). 예를 들어, 블록의 픽셀의 컬러 정보가 팔레트의 어떠한 엔트리에도 매핑되지 않을 때, 비디오 디코더 (30)는 이러한 픽셀을 이스케이프 픽셀로서 식별할 수도 있다. 다양한 예들에서, 비디오 디코더 (30)는 비디오 인코더 (20)에 의해 시그널링된 플래그를 사용하여 또는 비디오 인코더 (20)에 의해 시그널링된 인덱스 값 (예를 들어, 상술한 "다른 인덱스")에 의해 이스케이프 픽셀을 식별할 수도 있다.

[0200]

결과적으로, 비디오 디코더 (30)는, 팔레트에 포함된 엔트리들이 수가 1과 동일하고 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하지 않는다는 결정에 응답하여, 비디오 데이터의 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들의 디코딩을 바이패스할 수도 있다 (186). 일례로서, 비디오 디코더 (30)는, 인코딩된 비디오 비트스트림의 일부로서, 비디오 데이터의 블록과 연관된 인코딩된 비디오 데이터, 예를 들어, 신팩스 엘리먼트들 및/또는 플래그들을 수신할 수도 있고, 여기서, 블록과 연관된 인코딩된 비디오 데이터는 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들을 포함하지 않는다. 부가적으로, 비디오 디코더 (30)는 비디오 데이터의 블록의 픽셀 값들이 팔레트에 포함된 하나의 엔트리와 동일한 것으로 결정할 수도 있다 (188). 예를 들어, 비디오 디코더 (30)는 팔레트의 단일 엔트리에 의해 표시된 컬러 정보를 블록의 모든 픽셀들에게 할당함으로써 블록을 복원할 수도 있다.

[0201]

일례에서, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림의 일부로서, 인덱스 값들이 비디오 데이터의 블록에 대해 인코딩되는지를 나타내는 플래그를 더 수신할 수도 있다. 일례에서, 팔레트에 포함된 엔트리들의 수를 결정하기 위해, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림의 일부로서, 팔레트에서의 엔트리들의 수가 1과 동일한지를 나타내는 플래그를 수신할 수도 있다. 일례에서, 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하는지 결정하기 위해, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림의 일부로서, 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하는지 나타내는 플래그를 수신할 수도 있다. 일례에서, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림의 일부로서, 팔레트와 연관된 하나 이상의 신팩스 엘리먼트들을 수신할 수도 있다. 이러한 예에서, 비디오 디코더 (30)는 팔레트와 연관된 하나 이상의 신팩스 엘리먼트들을 디코딩할 수도 있다.

[0202]

도 5는 비디오 인코딩 디바이스가 조건들의 특정한 세트에 기초하여, 팔레트-코딩된 블록의 픽셀들에 대한 인덱스 값들의 인코딩을 바이패스하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스 (200)를 예시하는 플로우차트이다. 프로세스 (200)가 본 개시의 양태들에 따라 각종 디바이스들에 의해 수행될 수도 있지만, 프로세스 (200)는 설명의 용이함을 위해 도 1 및 도 2의 비디오 인코더 (20)에 관하여 본 명세서에 설명된다. 프로세스 (200)는, 비디오 인코더 (20)가 인코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트에 포함된 엔트리들의 수를 결정할 때 시작할 수도 있다 (202). 부가적으로, 비디오 인코더 (20)는 비디오 데이터의 블록이 팔레트에서의 엔트리들 중 어느 것과도 연관되지 않은 적어도 하나의 이스케이프 픽셀을 포함하는지 여부를 결정할 수도 있다 (204).

[0203]

결과적으로, 비디오 인코더 (20)는, 팔레트에 포함된 엔트리들의 수가 1과 동일하고 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하지 않는다는 결정에 응답하여, 블록의 픽셀 값들이 팔레트의 하나의 엔트리와 동일하다는 것을 결정할 수도 있으며, 비디오 데이터의 블록의 픽셀 값들에 대한 팔레트와 연관된 인덱스 값들의 인코딩을 바이패스할 수도 있다 (206). 예를 들어, 비디오 인코딩 디바이스 (20)는 블록의 픽셀 값들을 블록에 대응하는 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 인코딩하지 않고 블록에 대한 데이터, 예를 들어, 신팩스 엘리먼트들 및/또는 플래그들을 인코딩할 수도 있다. 부가적으로, 비디오 인코딩 디바이스 (20)는 비디오 데이터의 블록과 연관된 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다

(208).

[0204] 일례에서, 하나 이상의 신팩스 엘리먼트들을 인코딩하기 위해, 비디오 인코더 (20)는 인코딩된 비디오 비트스 트림의 일부로서, 인덱스 값들이 비디오 데이터의 블록의 픽셀 값들에 대해 인코딩되는지 나타내는 플래그를 인코딩할 수도 있다. 일례에서, 하나 이상의 신팩스 엘리먼트들을 인코딩하기 위해, 비디오 인코더 (20)는 인코딩된 비디오 비트스트림의 일부로서, 팔레트의 사이즈가 1과 동일한지를 나타내는 플래그를 인코딩할 수도 있다. 일례에서, 하나 이상의 신팩스 엘리먼트들을 인코딩하기 위해, 비디오 인코더 (20)는 인코딩된 비디오 비트스트림의 일부로서, 비디오 데이터의 블록이 적어도 하나의 이스케이프 픽셀을 포함하는지 나타내는 플래그를 인코딩할 수도 있다. 일례에서, 비디오 인코더 (20)는 팔레트와 연관된 하나 이상의 신팩스 엘리먼트들을 인코딩할 수도 있다. 이러한 예에서, 비디오 인코더 (20)는 인코딩된 비디오 비트스트림의 일부로서, 팔레트와 연관된 하나 이상의 신팩스 엘리먼트들을 시그널링할 수도 있다.

[0205] 도 6은 비디오 디코딩 디바이스가 비디오 데이터의 팔레트-코딩된 블록의 하나 이상의 이스케이프 픽셀들을 탈 양자화하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스 (220)를 예시하는 플로우차트이다.

프로세스 (220)가 본 개시의 양태들에 따라 각종 디바이스들에 의해 수행될 수도 있지만, 프로세스 (220)는 설명의 용이함을 위해 도 1 및 도 3의 비디오 디코더 (30)에 관하여 본 명세서에 설명된다. 프로세스 (220)는, 비디오 디코더 (30)가 디코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트를 결정할 때 시작할 수도 있다 (222). 부가적으로, 비디오 디코더 (30)는, 비디오 데이터의 블록에서, 팔레트에서의 하나 이상의 엔트리들 중 어느 것과도 연관되지 않은 하나 이상의 이스케이프 픽셀들을 식별할 수도 있다 (224).

[0206] 결과적으로, 비디오 디코더 (30)는 년-팔레트 기반 코딩에서 변환 계수 코딩을 위해 사용된 QP 값에 기초하여 주어진 컬러 채널에 대한 블록의 하나 이상의 이스케이프 픽셀들 모두에 대한 단일 양자화 파라미터 (QP) 값을 식별할 수도 있다 (226). 예를 들어, 비디오 디코더 (30)는 단일 QP 값이 블록과 연관된 컬러 채널의 종래 계수 디코딩을 위해 사용된 QP 값과 동일하다는 것을 결정할 수도 있다. 부가적으로, 비디오 디코더 (30)는 식별된 단일 QP 값을 사용하여 하나 이상의 이스케이프 픽셀들 각각을 탈양자화할 수도 있다 (228). 결과적으로, 비디오 디코더 (30)는 탈양자화된 이스케이프 픽셀들 및 팔레트에서의 하나 이상의 엔트리들과 연관된 비디오 데이터의 블록에서 하나 이상의 픽셀들에 대해 수신된 인덱스 값들에 기초하여 비디오 데이터의 블록에 대한 픽셀 값들을 결정할 수도 있다 (230).

[0207] 일례에서, 팔레트의 임의의 2개의 엔트리들은 적어도 팔레트에러 제한 만큼 변한다. 일례에서, 팔레트에러 제한은 블록과 연관된 팔레트 QP 값에 정비례한다. 일례에서, 하나 이상의 이스케이프 픽셀들을 식별하기 위해, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림에서, 하나 이상의 이스케이프 픽셀들 각각과 연관된 1-비트 플래그를 수신할 수도 있으며, 각각의 수신된 1-비트 플래그의 값에 기초하여, 하나 이상의 이스케이프 픽셀들 각각이 팔레트의 엔트리들 중 어느 것과도 연관되지 않는다는 것을 결정할 수도 있다. 일례에서, 하나 이상의 이스케이프 픽셀들 각각이 팔레트의 엔트리들 중 어느 것과도 연관되지 않는다는 것을 결정하기 위해, 비디오 디코더 (30)는 각각의 수신된 1-비트 플래그에 기초하고 그리고 이스케이프 픽셀들과 연관된 미리 정의된 다른 인덱스 값을 디코딩하지 않고 하나 이상의 이스케이프 픽셀들 각각이 팔레트의 엔트리들 중 어느 것과도 연관되지 않는다는 것을 결정할 수도 있다.

[0208] 도 7은 비디오 인코딩 디바이스가 비디오 데이터의 팔레트-코딩된 블록의 하나 이상의 이스케이프 픽셀들을 양자화하기 위해 본 개시의 기술들을 구현할 수도 있는 예시적인 프로세스 (240)를 예시하는 플로우차트이다.

프로세스 (240)가 본 개시의 양태들에 따라 각종 디바이스들에 의해 수행될 수도 있지만, 프로세스 (240)는 설명의 용이함을 위해 도 1 및 도 2의 비디오 인코더 (20)에 관하여 본 명세서에 설명된다. 프로세스 (240)는, 비디오 인코더 (20)가 인코딩될 비디오 데이터의 블록의 픽셀 값들을 나타내기 위해 사용된 팔레트를 결정할 때 시작할 수도 있다 (242). 부가적으로, 비디오 인코더 (20)는, 비디오 데이터의 블록에서, 팔레트에서의 하나 이상의 엔트리들 중 어느 것과도 연관되지 않은 하나 이상의 이스케이프 픽셀들을 식별할 수도 있다 (244).

[0209] 결과적으로, 비디오 인코더 (20)는 블록의 하나 이상의 이스케이프 픽셀들 모두에 대한 단일 양자화 파라미터 (QP) 값을 식별할 수도 있다 (246). 예를 들어, 비디오 인코더 (20)는 단일 QP 값이 블록과 연관된 컬러 채널의 종래 계수 인코딩을 위해 사용된 QP 값과 동일하다는 것을 결정할 수도 있다. 부가적으로, 비디오 인코더 (20)는 식별된 단일 QP 값을 사용하여 하나 이상의 이스케이프 픽셀들 각각을 양자화할 수도 있다 (248).

[0210]

일례에서, 단일 QP 값을 식별하기 위해, 비디오 인코더 (20)는 단일 QP 값이 블록과 연관된 컬러 채널의 종래 계수 인코딩을 위해 사용된 QP 값과 동일하다는 것을 결정할 수도 있다. 일례에서, 비디오 인코더 (20)는 팔레트의 팔레트 에러 제한이 블록과 연관된 QP 값에 정비례한다는 것을 결정할 수도 있고, 여기서, 팔레트의 임의의 2개의 엔트리들이 적어도 팔레트 에러 제한 만큼 변화한다. 일례에서, 팔레트 에러 제한을 결정하기 위해, 비디오 인코더 (20)는 팔레트 에러 제한을 블록과 연관된 QP 값에 매핑하는 테이블을 사용하여 팔레트 에러 제한을 식별할 수도 있다. 일례에서, 비디오 인코더 (20)는 이스케이프 픽셀들과 연관된 미리 정의된 다른 인덱스 값을 인코딩하지 않고 하나 이상의 이스케이프 픽셀들 각각과 연관된 1-비트 플래그를 인코딩할 수도 있으며, 여기서, 각각의 1-비트 플래그의 값은, 하나 이상의 이스케이프 픽셀들 중 각각의 이스케이프 픽셀이 팔레트의 엔트리들 중 어느 것과도 연관되지 않는다는 것을 나타낸다. 일례에서, 식별된 단일 QP 값을 사용하여 하나 이상의 이스케이프 픽셀들 각각을 양자화하기 위해, 비디오 인코더 (20)는 식별된 단일 QP 값에 기초하여 함수를 풀 수도 있다. 하나의 이러한 예에서, 함수를 풀기 위해, 비디오 인코더 (20)는 식별된 단일 QP 값에 기초하여 오른쪽-시프트 연산을 수행할 수도 있다.

[0211]

일부 예들에서, 비디오 데이터의 팔레트-기반 코딩을 위한 기술들은 인터- 또는 인트라-예측 코딩을 위한 기술들과 같은 하나 이상의 다른 코딩 기술들과 사용될 수도 있다. 예를 들어, 더욱 상세히 후술하는 바와 같이, 인코더 또는 디코더, 또는 조합된 인코더-디코더 (코덱) 가 팔레트-기반 코딩 뿐만 아니라 인터- 및 인트라-예측 코딩을 수행하도록 구성될 수도 있다.

[0212]

예에 의존하여, 본 명세서에서 설명된 기술들의 임의의 특정 작동들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있고, 함께 부가되거나 병합되거나 또는 제거될 수도 있음 (예를 들어, 설명된 모든 작동들 또는 이벤트들이 그 기술들의 실시를 위해 필수적인 것은 아님) 이 인식되어야 한다. 더욱이, 특정 예들에 있어서, 작동들 또는 이벤트들은 순차적인 것보다는, 예를 들어, 다중-스레딩된 프로세싱, 인터럽트 프로세싱, 또는 다중의 프로세서들을 통해 동시에 수행될 수도 있다. 또한, 본 개시의 특정한 양태들이 명확성을 위해 단일 모듈 또는 유닛에 의해 수행되는 것으로 설명되었지만, 본 개시의 기술들이 비디오 코더와 연관된 유닛들 또는 모듈들의 조합에 의해 수행될 수도 있다는 것을 이해해야 한다.

[0213]

본 개시의 특정한 양태들이 예시의 목적을 위해 개발중인 HEVC 표준에 관하여 설명되었다. 그러나, 본 개시에 설명된 기술들은 아직 개발되지 않은 다른 표준 또는 전매특허 비디오 코딩 프로세스들을 포함하는 다른 비디오 코딩 프로세스들에 유용할 수도 있다.

[0214]

상술한 기술들은 비디오 인코더 (20) (도 1 및 도 2) 및/또는 비디오 디코더 (30) (도 1 및 도 3)에 의해 수행될 수도 있고, 이 양자는 일반적으로 비디오 코더로서 지칭될 수도 있다. 유사하게, 비디오 코딩은 적용가능할 때, 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.

[0215]

기술들의 다양한 양태들의 특정한 조합들을 상술하였지만, 이들 조합들은 단지 본 개시에 설명된 기술들의 예들을 예시하기 위해 제공된다. 그에 따라, 본 개시의 기술들은 이들 예시적인 조합들에 한정되어서는 안되며, 본 개시에 설명된 기술들의 다양한 양태들의 임의의 가능한 조합을 포함할 수도 있다.

[0216]

하나 이상의 예들에 있어서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상으로 저장 또는 전송되고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비-일시적인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 캐리어파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시에서 설명된 기술들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0217]

한정이 아닌 예로서, 그러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장부, 자기 디스크 저장부 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 이용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 커넥션이 컴퓨터 판독가능 매체로 적절히 명명된다. 예를 들어, 동축 케이블, 광섬유 케이블, 꼬임쌍선, 디지털 가입자 라인 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신된다면, 동축 케이블, 광섬유 케

이블, 꼬임상선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 커넥션들, 캐리어파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않지만 대신 비-일시적인 유형의 저장 매체들로 지향됨을 이해해야 한다. 본 명세서에서 사용된 바와 같은 디스크 (disk) 및 디스크 (disc) 는 컴팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서, 디스크 (disk) 는 통상적으로 데이터를 자기적으로 재생하지만 디스크 (disc) 는 레이저들을 이용하여 데이터를 광학적으로 재생한다. 상기의 조합들이 또한, 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

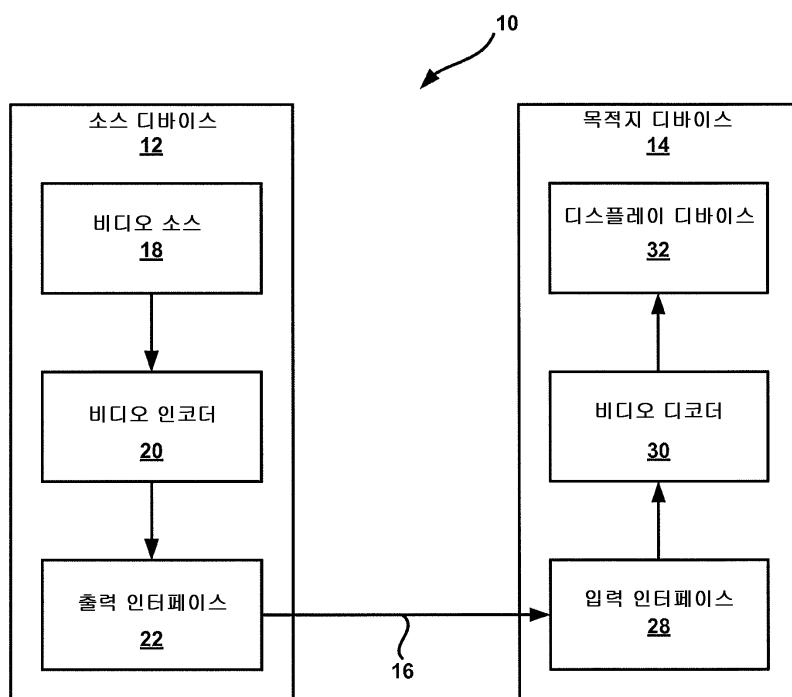
[0218] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적된 또는 별도의 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 이에 따라, 본 명세서에서 사용된 바와 같은 용어 "프로세서" 는 본 명세서에서 설명된 기술들의 구현에 적합한 전술한 구조 또는 임의의 다른 구조 중 임의의 구조를 지칭할 수도 있다. 추가적으로, 일부 양태들에 있어서, 본 명세서에서 설명된 기능은 인코딩 및 디코딩을 위해 구성되고 결합된 코덱에서 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 그 기술들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0219] 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 적합한 소프트웨어 및/또는 펌웨어와 함께 상기 설명된 바와 같은 하나 이상의 프로세서들을 포함하여 코덱 하드웨어 유닛으로 결합되거나 또는 상호운용식 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

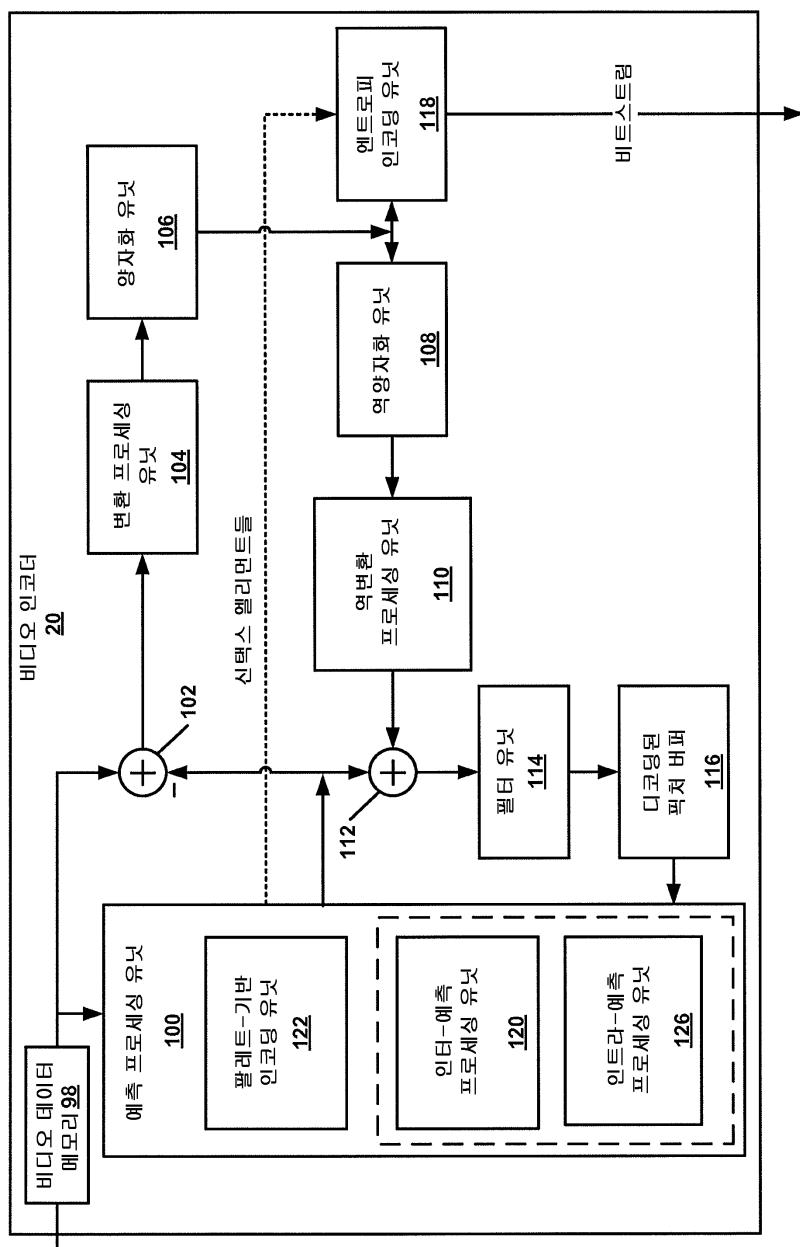
[0220] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

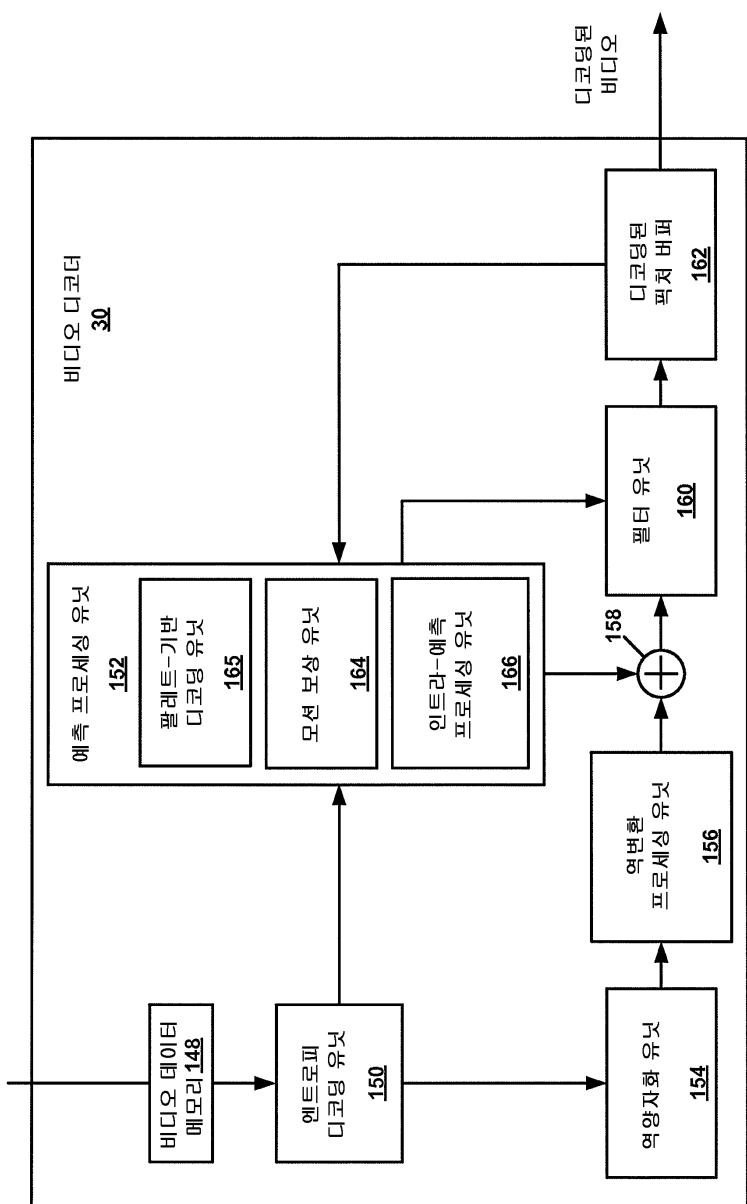
도면1



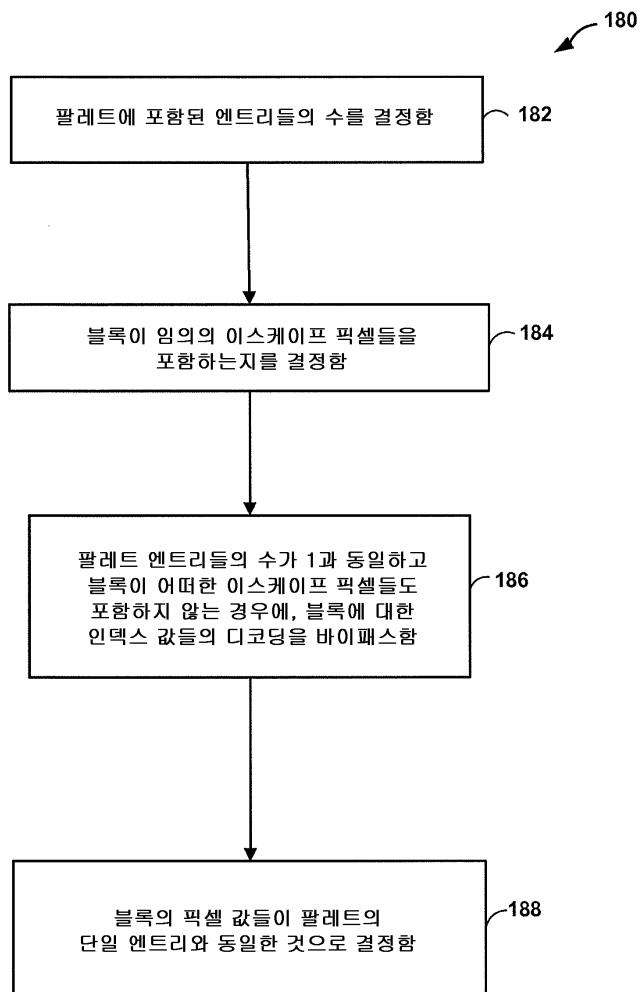
도면2



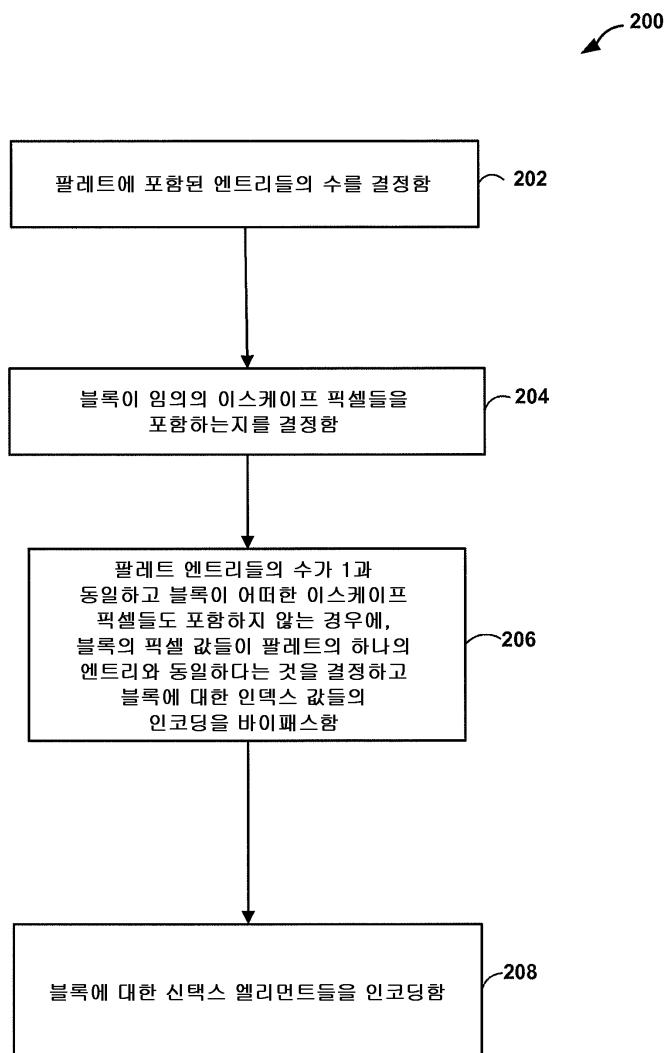
도면3



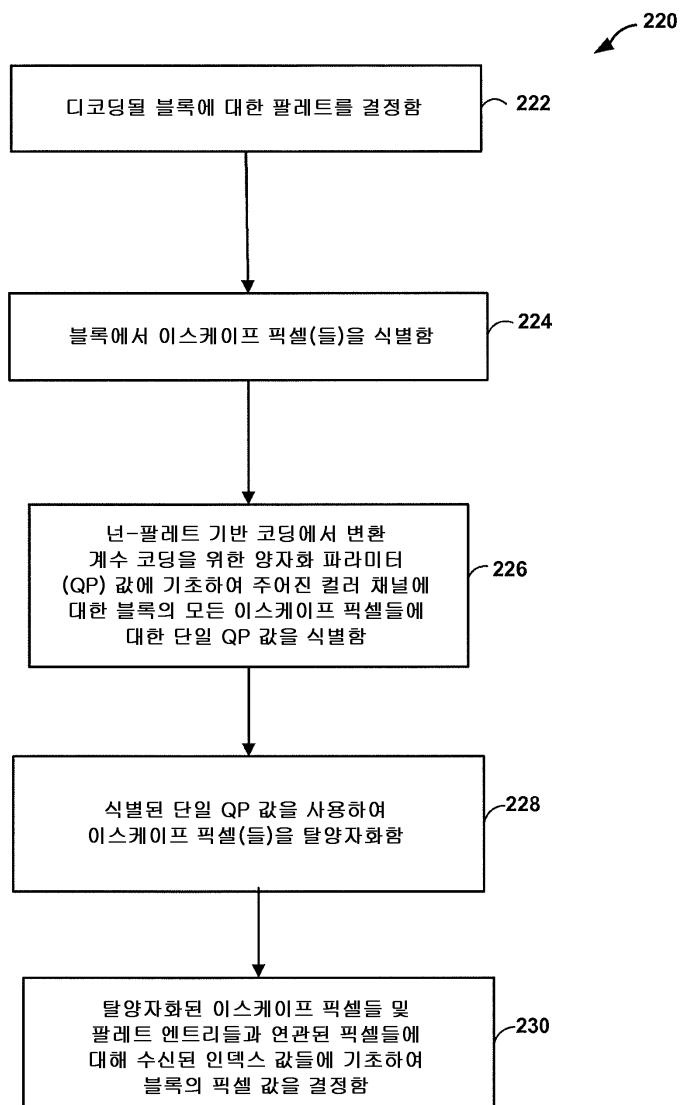
도면4



도면5



도면6



도면7

