



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년01월10일
(11) 등록번호 10-1937548
(24) 등록일자 2019년01월04일

- (51) 국제특허분류(Int. Cl.)
H04N 19/186 (2014.01) H04N 19/124 (2014.01)
H04N 19/129 (2014.01) H04N 19/182 (2014.01)
H04N 19/44 (2014.01) H04N 19/70 (2014.01)
- (52) CPC특허분류
H04N 19/186 (2015.01)
H04N 19/124 (2015.01)
- (21) 출원번호 10-2016-7028027
(22) 출원일자(국제) 2015년03월13일
심사청구일자 2017년12월27일
(85) 번역문제출일자 2016년10월07일
(65) 공개번호 10-2016-0132925
(43) 공개일자 2016년11월21일
(86) 국제출원번호 PCT/US2015/020483
(87) 국제공개번호 WO 2015/138927
국제공개일자 2015년09월17일
- (30) 우선권주장
61/953,667 2014년03월14일 미국(US)
14/656,087 2015년03월12일 미국(US)
- (56) 선행기술조사문헌
GUO L ET AL: "Pal ette Mode for Screen Content Coding", MPEG MEETING, no. m28786 (2013.04.20.)
- (73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
푸 웨이
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
저우 평
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
- (74) 대리인
특허법인코리아나

전체 청구항 수 : 총 34 항

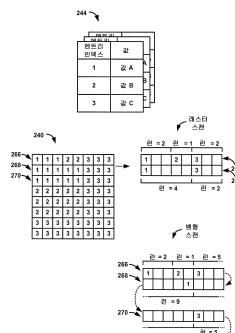
심사관 : 장석환

(54) 발명의 명칭 팔레트 기반 비디오 코딩

(57) 요약

일 예에서, 비디오 데이터를 코딩하는 방법이, 비디오 코더에 의해 그리고 비디오 데이터 블록에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 단계로서, 비디오 데이터 블록의 제 1 라인은 비디오 데이터 블록의 에지에 인접하게 위치된 화소를 포함하고, 비디오 데이터 블록의 제 2 라인은 블록의 에지에 인접하게 그리고 제 1 라인의 화소에 인접하게 위치된 화소를 포함하는, 상기 팔레트를 결정하는 단계를 포함한다. 이 예에서, 그 방법은, 스캔 순서로, 블록의 화소들을 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩하는 단계로서, 제 2 라인의 화소는 스캔 순서에서 제 1 라인의 화소를 바로 뒤따르는, 상기 스캔 순서로 인덱스 값들을 코딩하는 단계를 또한 포함한다.

대표도 - 도5



(52) CPC특허분류

H04N 19/129 (2015.01)

H04N 19/182 (2015.01)

H04N 19/44 (2015.01)

H04N 19/70 (2015.01)

(72) 발명자

조쉬 라잔 랙스맨

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

솔레 로할스 호엘

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

비디오 코더에 의해 그리고 비디오 데이터의 블록에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 단계로서, 상기 비디오 데이터의 블록의 제 1 라인은 상기 비디오 데이터의 블록의 에지에 인접하게 위치한 화소를 포함하고, 상기 비디오 데이터의 블록의 제 2 라인은 상기 블록의 상기 에지에 인접하게 그리고 상기 제 1 라인의 상기 화소에 바로 인접하게 위치한 화소를 포함하는, 상기 팔레트를 결정하는 단계; 및

상기 블록의 화소들을 상기 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 뱀형 스캔 순서로 코딩하는 단계로서, 상기 뱀형 스캔 순서에서 상기 제 2 라인의 상기 화소는 상기 제 1 라인의 상기 화소를 바로 뒤따르는, 상기 인덱스 값들을 뱀형 스캔 순서로 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 라인은 제 1 행이고 상기 제 2 라인은 제 2 행이며,

상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하는 단계는 상기 제 2 행에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 상기 제 1 행에서의 상기 화소에 대한 인덱스 값을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 1 라인은 제 1 열이고 상기 제 2 라인은 제 2 열이며,

상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하는 단계는 상기 제 2 열에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 상기 제 1 열에서의 상기 화소에 대한 인덱스 값을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 에지는 제 1 에지이고,

상기 제 2 라인의 상기 화소는 상기 제 2 라인의 제 1 화소이며,

상기 제 2 라인은 상기 제 1 에지에 평행한 상기 비디오 데이터의 블록의 제 2 에지에 인접하게 위치한 제 2 화소를 포함하고,

상기 인덱스 값들을 코딩하는 단계는,

상기 뱀형 스캔 순서에서의 연속적인 화소들의 런이 동일한 인덱스 값을 갖는 것을 나타내는 하나 이상의 신텍스 엘리먼트들을 코딩하는 단계를 포함하며,

상기 연속적인 화소들의 런은 상기 제 1 라인의 상기 화소와 상기 제 2 라인의 상기 제 1 화소를 포함하지만 상기 제 2 라인의 상기 제 2 화소를 포함하지 않는, 비디오 데이터를 코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 제 2 라인의 상기 화소가 상기 뱀형 스캔 순서에서 상기 제 1 라인의 상기 화소 바로 뒤에 스캔됨을 나타내는 선택스 엘리먼트를 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 팔레트에서의 각각의 엔트리는 상기 복수의 각각의 컬러 값들 중 각각의 컬러 값에 대응하는 고유 인덱스 값을 각각 포함하며,

상기 방법은,

상기 복수의 각각의 컬러 값들 중 각각의 컬러 값들에 대응하지 않는 이스케이프 화소들에 대한 값들을 코딩하는 단계를 더 포함하며,

상기 인덱스 값들 및 상기 이스케이프 화소들을 코딩하는 것은 절단된 이진수 코드를 사용하여 상기 인덱스 값들 및 상기 이스케이프 화소들 중 일방 또는 양방을 코딩하는 것을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 비디오 코더는 비디오 인코더이며,

상기 인덱스 값들을 코딩하는 단계는 상기 인덱스 값들을 인코딩하는 단계를 포함하며,

상기 방법은, 상기 팔레트와 상기 인덱스 값들을 표현하는 하나 이상의 선택스 엘리먼트들을 포함하는 비트스트림을 생성하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 8

제 7 항에 있어서,

상기 비디오 데이터의 블록에 대해, 각각의 화소 값과 상기 각각의 화소 값에 유사한 화소 값들을 갖는 화소들의 각각의 양을 각각 포함하는 복수의 엔트리들을 포함하는 히스토그램을 결정하는 단계;

상기 히스토그램에 기초하여, 메이저 컬러 화소들 또는 이스케이프 화소들로서 상기 비디오 데이터의 블록의 하나 이상의 화소들을 분류하는 단계로서, 이스케이프 화소들로서 분류된 화소들은 메이저 컬러 화소들로서 분류된 화소들보다는 더 낮은 양의 화소들을 나타내는 상기 히스토그램에서의 엔트리들에 대응하는, 상기 하나 이상의 화소들을 분류하는 단계; 및

상기 히스토그램에 기초하여 상기 팔레트를 결정하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 9

제 8 항에 있어서,

상기 히스토그램의 상기 복수의 엔트리들 중 각각의 엔트리는 양자화되는 경우 동일한 화소 값을 갖는 화소들의 양을 나타내는, 비디오 데이터를 코딩하는 방법.

청구항 10

제 8 항에 있어서,

상기 팔레트를 결정하는 단계는,

상기 비디오 데이터의 블록에 대한 상기 팔레트의 제 1 엔트리를 결정하는 단계로서, 상기 제 1 엔트리는 상기 히스토그램의 제 1 엔트리에 대응하는, 상기 팔레트의 제 1 엔트리를 결정하는 단계;

상기 히스토그램의 제 2 엔트리에 포함된 화소 값과 상기 팔레트에 포함된 하나 이상의 화소 값들 간의 차이에 대응하는 거리 값을 결정하는 단계; 및

상기 거리 값이 임계값을 충족시킨다는 결정에 응답하여, 상기 히스토그램의 상기 제 2 엔트리에 포함된 상기

화소 값을 상기 팔레트의 제 2 엔트리 속으로 복사하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 11

제 10 항에 있어서,

상기 임계값은 상기 비디오 데이터의 블록의 양자화 파라미터 (QP) 에 기초하는, 비디오 데이터를 코딩하는 방법.

청구항 12

제 1 항에 있어서,

상기 비디오 코더는 비디오 디코더이며,

상기 인덱스 값들을 코딩하는 단계는 상기 인덱스 값들을 디코딩하는 단계를 포함하며,

상기 방법은, 상기 팔레트와 상기 인덱스 값들을 표현하는 하나 이상의 신택스 엘리먼트들을 포함하는 비트스트림을 수신하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 13

비디오 데이터를 코딩하는 장치로서,

비디오 데이터의 블록을 저장하도록 구성되는 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 블록에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 것으로서, 상기 비디오 데이터의 블록의 제 1 라인은 상기 비디오 데이터의 블록의 에지에 인접하게 위치한 화소를 포함하고, 상기 비디오 데이터의 블록의 제 2 라인은 상기 블록의 상기 에지에 인접하게 그리고 상기 제 1 라인의 상기 화소에 바로 인접하게 위치한 화소를 포함하는, 상기 팔레트를 결정하는 것을 행하도록; 그리고

상기 블록의 화소들을 상기 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 뱀형 스캔 순서로 코딩하는 것으로서, 상기 뱀형 스캔 순서에서 상기 제 2 라인의 상기 화소는 상기 제 1 라인의 상기 화소를 바로 뒤따르는, 상기 인덱스 값들을 뱀형 스캔 순서로 코딩하는 것을 행하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 14

제 13 항에 있어서,

상기 제 1 라인은 제 1 행이고 상기 제 2 라인은 제 2 행이며,

상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하기 위해, 상기 하나 이상의 프로세서들은, 상기 제 2 행에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 상기 제 1 행에서의 상기 화소에 대한 인덱스 값을 코딩하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 15

제 13 항에 있어서,

상기 제 1 라인은 제 1 열이고 상기 제 2 라인은 제 2 열이며,

상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하기 위해, 상기 하나 이상의 프로세서들은, 상기 제 2 열에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 상기 제 1 열에서의 상기 화소에 대한 인덱스 값을 코딩하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 16

제 13 항에 있어서,

상기 예지는 제 1 예지이며,

상기 제 2 라인의 상기 화소는 상기 제 2 라인의 제 1 화소이며,

상기 제 2 라인은 상기 제 1 예지에 평행한 상기 비디오 데이터의 블록의 제 2 예지에 인접하게 위치한 제 2 화소를 포함하고,

상기 인덱스 값들을 코딩하기 위해, 상기 하나 이상의 프로세서들은,

상기 뱀형 스캔 순서에서의 연속적인 화소들의 런 - 상기 연속적인 화소들의 런은 상기 제 1 라인의 상기 화소와 상기 제 2 라인의 상기 제 1 화소를 포함하지만 상기 제 2 라인의 상기 제 2 화소를 포함하지 않음 - 이 동일한 인덱스 값을 갖는 것을 나타내는 하나 이상의 신택스 엘리먼트들을 코딩하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 17

제 13 항에 있어서,

상기 하나 이상의 프로세서들은 또한, 상기 제 2 라인의 상기 화소가 상기 뱀형 스캔 순서에서 상기 제 1 라인의 상기 화소 바로 뒤에 스캔됨을 나타내는 신택스 엘리먼트를 코딩하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 18

제 13 항에 있어서,

상기 팔레트에서의 각각의 엔트리는 상기 복수의 각각의 컬러 값들 중 각각의 컬러 값에 대응하는 고유 인덱스 값을 각각 포함하며,

상기 하나 이상의 프로세서들은 또한,

상기 복수의 각각의 컬러 값들 중 각각의 컬러 값들에 대응하지 않는 이스케이프 화소들에 대한 값들을 코딩하도록 구성되고,

상기 하나 이상의 프로세서들은, 절단된 이진수 코드를 사용하여 상기 인덱스 값들 및 상기 이스케이프 화소들 중 일방 또는 양방을 코딩하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 19

제 13 항에 있어서,

비디오 인코더를 더 포함하며,

상기 인덱스 값들을 코딩하기 위해, 상기 하나 이상의 프로세서들은 상기 인덱스 값들을 인코딩하도록 구성되고,

상기 하나 이상의 프로세서들은 또한, 상기 팔레트와 상기 인덱스 값들을 표현하는 하나 이상의 신택스 엘리먼트들을 포함하는 비트스트림을 생성하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 20

제 19 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 비디오 데이터의 블록에 대해, 각각의 화소 값과 상기 각각의 화소 값에 유사한 화소 값들을 갖는 화소들의 각각의 양을 각각 포함하는 복수의 엔트리들을 포함하는 히스토그램을 결정하도록;

상기 히스토그램에 기초하여, 메이저 컬러 화소들 또는 이스케이프 화소들 - 이스케이프 화소들로서 분류된 화소들은 메이저 컬러 화소들로서 분류된 화소들보다는 더 낮은 양의 화소들을 나타내는 상기 히스토그램에서의 엔트리들에 대응함 - 로서 상기 비디오 데이터의 블록의 하나 이상의 화소들을 분류하도록; 그리고

상기 히스토그램에 기초하여 상기 팔레트를 결정하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 21

제 20 항에 있어서,

상기 히스토그램의 상기 복수의 엔트리들 중 각각의 엔트리는 양자화되는 경우 동일한 화소 값을 갖는 화소들의 양을 나타내는, 비디오 데이터 코딩 장치.

청구항 22

제 20 항에 있어서,

상기 팔레트를 결정하기 위해, 상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 블록에 대한 상기 팔레트의 제 1 엔트리 - 상기 제 1 엔트리는 상기 히스토그램의 제 1 엔트리에 대응함 - 를 결정하도록;

상기 히스토그램의 제 2 엔트리에 포함된 화소 값과 상기 팔레트에 포함된 하나 이상의 화소 값들 간의 차이에 대응하는 거리 값을 결정하도록; 그리고

상기 거리 값이 임계값을 충족시킨다는 결정에 응답하여, 상기 히스토그램의 상기 제 2 엔트리에 포함된 상기 화소 값을 상기 팔레트의 제 2 엔트리 속으로 복사하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 23

제 22 항에 있어서,

상기 임계값은 상기 비디오 데이터의 블록의 양자화 파라미터 (QP) 에 기초하는, 비디오 데이터 코딩 장치.

청구항 24

제 13 항에 있어서,

비디오 디코더를 더 포함하며,

상기 인덱스 값들을 코딩하기 위해, 상기 하나 이상의 프로세서들은 상기 인덱스 값들을 디코딩하도록 구성되고,

상기 하나 이상의 프로세서들은 또한, 상기 팔레트와 상기 인덱스 값들을 표현하는 하나 이상의 신택스 엘리먼트들을 포함하는 비트스트림을 수신하도록 구성되는, 비디오 데이터 코딩 장치.

청구항 25

비디오 데이터를 코딩하는 장치로서,

비디오 코더에 의해 그리고 비디오 데이터의 블록에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 수단으로서, 상기 비디오 데이터의 블록의 제 1 라인은 상기 비디오 데이터의 블록의 에지에 인접하게 위치한 화소를 포함하고, 상기 비디오 데이터의 블록의 제 2 라인은 상기 블록의 상기 에지에 인접하게 그리고 상기 제 1 라인의 상기 화소에 바로 인접하게 위치한 화소를 포함하는, 상기 팔레트를 결정하는 수단; 및

상기 블록의 화소들을 상기 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 뱀형 스캔 순서로 코딩하는 수단으로서, 상기 뱀형 스캔 순서에서 상기 제 2 라인의 상기 화소는 상기 제 1 라인의 상기 화소를 바로 뒤따르는, 상기 인덱스 값들을 뱀형 스캔 순서로 코딩하는 수단을 포함하는, 비디오 데이터 코딩 장치.

청구항 26

제 25 항에 있어서,

상기 제 1 라인은 제 1 행이고 상기 제 2 라인은 제 2 행이며,

상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하는 수단은 상기 제 2 행에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 상기 제 1 행에서의 상기 화소에 대한 인덱스 값을 코딩하는 수단을 포함하는, 비디오 데이터 코딩 장치.

청구항 27

제 25 항에 있어서,

상기 제 1 라인은 제 1 열이고 상기 제 2 라인은 제 2 열이며,

상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하는 수단은 상기 제 2 열에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 상기 제 1 열에서의 상기 화소에 대한 인덱스 값을 코딩하는 수단을 포함하는, 비디오 데이터 코딩 장치.

청구항 28

제 25 항에 있어서,

상기 에지는 제 1 에지이며,

상기 제 2 라인의 상기 화소는 상기 제 2 라인의 제 1 화소이며,

상기 제 2 라인은 상기 제 1 에지에 평행한 상기 비디오 데이터의 블록의 제 2 에지에 인접하게 위치한 제 2 화소를 포함하고,

상기 인덱스 값들을 코딩하는 수단은,

상기 뱀형 스캔 순서에서의 연속적인 화소들의 런이 동일한 인덱스 값을 갖는 것을 나타내는 하나 이상의 선택스 엘리먼트들을 코딩하는 수단으로서, 상기 연속적인 화소들의 런은 상기 제 1 라인의 상기 화소와 상기 제 2 라인의 상기 제 1 화소를 포함하지만 상기 제 2 라인의 상기 제 2 화소를 포함하지 않는, 상기 하나 이상의 선택스 엘리먼트들을 코딩하는 수단을 포함하는, 비디오 데이터 코딩 장치.

청구항 29

제 25 항에 있어서,

상기 제 2 라인의 상기 화소가 상기 뱀형 스캔 순서에서 상기 제 1 라인의 상기 화소 바로 뒤에 스캔됨을 나타내는 선택스 엘리먼트를 코딩하는 수단을 더 포함하는, 비디오 데이터 코딩 장치.

청구항 30

명령들을 저장하고 있는 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행되는 경우, 비디오 코더의 하나 이상의 프로세서들로 하여금,

비디오 데이터의 블록에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 것으로서, 상기 비디오 데이터의 블록의 제 1 라인은 상기 비디오 데이터의 블록의 에지에 인접하게 위치한 화소를 포함하고, 상기 비디오 데이터의 블록의 제 2 라인은 상기 블록의 상기 에지에 인접하게 그리고 상기 제 1 라인의 상기 화소에 바로 인접하게 위치한 화소를 포함하는, 상기 팔레트를 결정하는 것을 행하게 하며; 그리고

상기 블록의 화소들을 상기 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 뱀형 스캔 순서로 코딩하는 것으로서, 상기 뱀형 스캔 순서에서 상기 제 2 라인의 상기 화소는 상기 제 1 라인의 상기 화소를 바로 뒤따르는, 상기 인덱스 값들을 뱀형 스캔 순서로 코딩하는 것을 행하게 하는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 31

제 30 항에 있어서,

상기 제 1 라인은 제 1 행이고 상기 제 2 라인은 제 2 행이며,

상기 하나 이상의 프로세서들로 하여금 상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하게 하는 상기 명령들은, 상기 하나 이상의 프로세서들로 하여금, 상기 제 1 행에서의 상기 화소에 대한 인덱스 값을 상기 제 2 행에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 코딩하게 하는 명령들을 포함하는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 32

제 30 항에 있어서,

상기 제 1 라인은 제 1 열이고 상기 제 2 라인은 제 2 열이며,

상기 하나 이상의 프로세서들로 하여금 상기 인덱스 값들을 상기 뱀형 스캔 순서로 코딩하게 하는 상기 명령들은, 상기 하나 이상의 프로세서들로 하여금, 상기 제 1 열에서의 상기 화소에 대한 인덱스 값을 상기 제 2 열에서의 상기 화소에 대한 인덱스 값을 코딩하기 바로 전에 코딩하게 하는 명령들을 포함하는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 33

제 30 항에 있어서,

상기 예지는 제 1 예지이며,

상기 제 2 라인의 상기 화소는 상기 제 2 라인의 제 1 화소이며,

상기 제 2 라인은 상기 제 1 예지에 평행한 상기 비디오 데이터의 블록의 제 2 예지에 인접하게 위치한 제 2 화소를 포함하고,

상기 하나 이상의 프로세서들로 하여금 상기 인덱스 값들을 코딩하게 하는 명령들은, 상기 하나 이상의 프로세서들로 하여금,

상기 뱀형 스캔 순서에서의 연속적인 화소들의 런 - 상기 연속적인 화소들의 런은 상기 제 1 라인의 상기 화소와 상기 제 2 라인의 상기 제 1 화소를 포함하지만 상기 제 2 라인의 상기 제 2 화소를 포함하지 않음 - 이 동일한 인덱스 값을 갖는 것을 나타내는 하나 이상의 신택스 엘리먼트들을 코딩하게 하는 명령들을 포함하는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 34

제 30 항에 있어서,

상기 하나 이상의 프로세서들로 하여금, 상기 제 2 라인의 상기 화소가 상기 뱀형 스캔 순서에서 상기 제 1 라인의 상기 화소 바로 뒤에 스캔됨을 나타내는 신택스 엘리먼트를 코딩하게 하는 명령들을 더 포함하는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2014년 3월 14일자로 출원된 미국 가출원 제61/953,667호를 우선권 주장하며, 그 전체 내용은 참조로 본원에 통합된다.

[0002] 본 개시물은 비디오 인코딩 및 디코딩에 관한 것이다.

배경 기술

[0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말기들 (PDA들), 랩톱 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, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (Advanced Video Coding, AVC) 에 의해 규정된 표준들, 현재 개발중인 고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장본들에 기재된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현하는 것에 의해 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 압축 기법들은 공간적 (픽처 내) 예측 및/또는 시간적 (픽처 간) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거할 수도 있다. 블록 기반 비디오 코딩의 경우, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분) 가 비디오 블록들로 구획화될 수도 있다. 픽처의 인트라 코딩식 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측을 이용하여 인코딩된다. 픽처의 인터 코딩식 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 픽처들에서의 참조 샘플들에 관한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들이라고 지칭될 수도 있고, 참조 픽처들은 참조 프레임들이라고 지칭될 수도 있다.

[0005] 공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록이 생겨나게 한다. 잔차 데이터는 코딩될 원래의 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터 코딩식 블록이 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라 인코딩되고, 잔차 데이터는 코딩된 블록 및 예측 블록 사이의 차이를 나타낸다. 인트라 코딩식 블록이 인트라 코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 결과적으로 잔차 계수들이 생겨나며, 그 계수들은 그 다음에 양자화될 수도 있다. 처음에는 2차원 어레이로 배열된 양자화된 계수들은, 계수들의 1차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩이 더 많은 압축을 달성하기 위해 적용될 수도 있다.

발명의 내용

해결하려는 과제

[0006] 본 개시물의 기법들은 팔레트 (palette) 기반 비디오 코딩에 관련된다. 예를 들어, 팔레트 기반 코딩에서, 비디오 코더 (비디오 인코더 또는 비디오 디코더) 가 특정 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 표현하기 위한 컬러들의 테이블로서 이른바 "팔레트"를 형성할 수도 있다. 팔레트 기반 코딩은 비교적 적은 수의 컬러들을 갖는 비디오 데이터의 영역들을 코딩하는데 특히 유용할 수도 있다. 실제 화소 값들 (또는 그것들의 잔차들) 을 코딩하는 것이 아니라, 비디오 코더는 화소들을 그 화소들의 컬러들을 표현하는 팔레트에서의 엔트리들과 관련시키는 하나 이상의 화소들에 대한 인덱스 값들을 코딩할 수도 있다. 팔레트가 디코더로 명시적으로 인코딩되고 전송되거나, 이전의 팔레트 엔트리들로부터 예측되거나, 또는 그것들의 조합으로 처리될 수도 있다. 본 개시물에서 설명되는 기법들은 팔레트 기반 코딩 모드들을 시그널링하는 것, 팔레트들을

코딩하는 것, 팔레트들을 예측하는 것, 팔레트들을 도출하는 것, 그리고 팔레트 기반 코딩 맵들 및 다른 선택스 엘리먼트들을 코딩하는 것 중 하나 이상의 다양한 조합들을 위한 기법들을 포함할 수도 있다.

과제의 해결 수단

- [0007] 하나의 예에서, 비디오 데이터를 코딩하는 방법이, 비디오 코더에 의해 그리고 비디오 데이터 블록에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 단계로서, 비디오 데이터 블록의 제 1 라인은 비디오 데이터 블록의 에지에 인접하게 위치한 화소를 포함하고, 비디오 데이터 블록의 제 2 라인은 블록의 에지에 인접하게 그리고 제 1 라인의 화소에 인접하게 위치한 화소를 포함하는, 상기 팔레트를 결정하는 단계를 포함한다. 이 예에서, 그 방법은, 스캔 순서로, 블록의 화소들을 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩하는 단계로서, 제 2 라인의 화소는 스캔 순서에서 제 1 라인의 화소를 바로 뒤따르는, 상기 스캔 순서로 인덱스 값들을 코딩하는 단계를 또한 포함한다.
- [0008] 다른 예에서, 비디오 데이터를 코딩하는 장치가 비디오 데이터 블록을 저장하도록 구성되는 메모리와, 하나 이상의 프로세서들을 구비한다. 이 예에서, 하나 이상의 프로세서들은, 비디오 데이터 블록 - 상기 비디오 데이터 블록의 제 1 라인은 상기 비디오 데이터 블록의 에지에 인접하게 위치한 화소를 포함하고, 상기 비디오 데이터 블록의 제 2 라인은 상기 블록의 상기 에지에 인접하게 그리고 상기 제 1 라인의 상기 화소에 인접하게 위치한 화소를 포함함 - 에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하도록; 및 상기 제 2 라인의 상기 화소가 스캔 순서에서 상기 제 1 라인의 상기 화소를 바로 뒤따르는 스캔 순서로, 상기 블록의 화소들을 상기 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩하도록 구성된다.
- [0009] 다른 예에서, 비디오 데이터를 코딩하는 장치가, 비디오 코더에 의해 그리고 비디오 데이터 블록 - 상기 비디오 데이터 블록의 제 1 라인은 상기 비디오 데이터 블록의 에지에 인접하게 위치한 화소를 포함하고, 상기 비디오 데이터 블록의 제 2 라인은 상기 블록의 상기 에지에 인접하게 그리고 상기 제 1 라인의 상기 화소에 인접하게 위치한 화소를 포함함 - 에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 수단; 및 상기 제 2 라인의 상기 화소가 스캔 순서에서 상기 제 1 라인의 상기 화소를 바로 뒤따르는 스캔 순서로, 상기 블록의 화소들을 상기 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩하는 수단을 포함한다.
- [0010] 다른 예에서, 컴퓨터 판독가능 매체는, 실행되는 경우, 비디오 코더의 하나 이상의 프로세서들로 하여금, 비디오 데이터 블록 - 상기 비디오 데이터 블록의 제 1 라인은 상기 비디오 데이터 블록의 에지에 인접하게 위치한 화소를 포함하고, 상기 비디오 데이터 블록의 제 2 라인은 상기 블록의 상기 에지에 인접하게 그리고 상기 제 1 라인의 상기 화소에 인접하게 위치한 화소를 포함함 - 에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하게 하며; 그리고 제 2 라인의 상기 화소가 스캔 순서에서 제 1 라인의 상기 화소를 바로 뒤따르는 스캔 순서로, 블록의 화소들을 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩하게 하는 명령들을 저장하고 있다.
- [0011] 본 개시물의 하나 이상의 예들의 상세는 첨부 도면들 및 다음의 설명에서 언급된다. 다른 특징들, 목적들, 및 장점들은 상세한 설명, 도면들, 및 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

- [0012] 도 1은 본 개시물에서 설명되는 기법들을 이용할 수도 있는 일 예의 비디오 코딩 시스템을 도시하는 블록도이다.
- 도 2는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 도시하는 블록도이다.
- 도 3은 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 도시하는 블록도이다.
- 도 4는 본 개시물의 기법들과 일치하게, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 일 예를 도시하는 개념도이다.
- 도 5는 본 개시물의 기법들과 일치하게, 화소들의 블록을 위한 팔레트로의 인덱스들을 결정하는 일 예를 도시하는 개념도이다.
- 도 6a와 도 6b는 본 개시물의 기법들에 일치하게, 예의 스캔 순서들을 도시하는 개념도들이다.
- 도 7a 내지 도 7c는 본 개시물의 기법들에 일치하게, 팔레트 기반 비디오 코딩의 예의 모드들을 도시하는 개념

도들이다.

도 8a 내지 도 8c는 본 개시물의 기법들에 일치하게, 팔레트 기반 비디오 코딩의 예의 모드들을 도시하는 개념도들이다.

도 9는 본 개시물의 기법들에 일치하게, 팔레트 코딩 모드를 사용하여 비디오 데이터를 코딩하는 일 예의 프로세스를 도시하는 흐름도이다.

도 10은 본 개시물의 기법들과 일치하게, 비디오 데이터의 블록을 위한 팔레트를 결정하는 일 예의 프로세스를 도시하는 개념도이다.

발명을 실시하기 위한 구체적인 내용

- [0013] 본 개시물은 비디오 코딩과 압축을 위한 기법들을 포함한다. 대체로, 본 개시물은 비디오 데이터의 팔레트 기반 코딩을 위한 기법들을 설명한다. 전통적인 비디오 코딩에서, 이미지들이 연속 톤이고 공간적으로 매끄러운 것으로 가정된다. 이들 가정들에 기초하여, 블록 기반 변환, 필터링 등과 같은 다양한 도구들이 개발되었고, 이러한 도구들은 자연 콘텐츠 비디오들에 대해 양호한 성능을 보여 주었다.
- [0014] 그러나, 원격 데스크톱, 협력 작업 및 무선 디스플레이 같은 애플리케이션들에서, 컴퓨터 생성된 스크린 콘텐츠(예컨대, 이를테면 텍스트 또는 컴퓨터 그래픽)가 압축될 지배적인 콘텐츠일 수도 있다. 이 유형의 콘텐츠는 이산 톤 및 피쳐 예리한 라인들과, 높은 콘트라스트 물체 경계들을 갖는 경향이 있다. 연속-톤 및 평활도의 가정은 스크린 콘텐츠에 더 이상 적용할 수도 없고, 따라서 전통적인 비디오 코딩 기법들은 스크린 콘텐츠를 포함하는 비디오 데이터를 압축하기 위한 효율적인 방도들이 아닐 수도 있다.
- [0015] 본 개시물은 팔레트 기반 코딩을 설명하는데, 이 코딩은 스크린 생성된 콘텐츠 코딩에 특히 적합할 수도 있다. 예를 들어, 비디오 데이터의 특정 영역이 비교적 작은 수의 컬러들을 갖는다고 가정한다. 비디오 코더(비디오 인코더 또는 비디오 디코더)가 특정 영역(예컨대, 주어진 블록)의 비디오 데이터를 표현하기 위한 컬러들의 테이블로서 이른바 "팔레트"를 코딩할 수도 있다. 각각의 화소는 화소의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 코더는 화소 값을 팔레트에서의 적절한 엔트리에 매핑하는 인덱스를 코딩할 수도 있다.
- [0016] 위의 예에서, 비디오 인코더가, 블록에 대한 팔레트를 결정하며(예컨대, 팔레트를 명시적으로 코딩하며, 그것을 예측하며, 또는 그것들의 조합을 행하며), 각각의 화소의 값을 표현하기 위해 팔레트에서의 엔트리의 위치를 찾아내고, 화소 값을 팔레트에 매핑하는 화소들에 대한 인덱스 값들로 팔레트를 인코딩함으로써 비디오 데이터 블록을 인코딩할 수도 있다. 비디오 디코더가, 인코딩된 비트스트림으로부터, 블록에 대한 팔레트, 뿐만 아니라 블록의 화소들에 대한 인덱스 값들을 획득할 수도 있다. 비디오 디코더는 블록의 화소 값들을 복원하기 위해 화소들의 인덱스 값들을 팔레트의 엔트리들에 매핑할 수도 있다.
- [0017] 위의 예는 팔레트 기반 코딩의 일반적인 설명을 제공하기 위한 것이다. 다양한 예들에서, 본 개시물에서 설명되는 기법들은 팔레트 기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 예측하는 것, 팔레트들을 도출하는 것, 그리고 팔레트 기반 코딩 맵들 및 다른 신택스 엘리먼트들을 송신하는 것 중 하나 이상의 다양한 조합들을 위한 기법들을 포함할 수도 있다. 이러한 기법들은 비디오 코딩 효율을 개선하여, 예컨대, 스크린 생성된 콘텐츠를 표현하기 위해 더 적은 비트들을 요구할 수도 있다.
- [0018] 비디오 데이터의 팔레트 기반 코딩을 위한 기법들은 하나 이상의 다른 코딩 기법들, 이를테면 인터 예측 또는 인트라 예측 코딩을 위한 기법들과 함께 사용될 수도 있다. 예를 들어, 아래에서 더욱 상세히 설명되는 바와 같이, 인코더 또는 디코더, 또는 결합된 인코더-디코더(코덱)가, 인터 예측 및 인트라 예측 코딩, 뿐만 아니라 팔레트 기반 코딩을 수행하도록 구성될 수도 있다.
- [0019] 일부 예들에서, 팔레트 기반 코딩 기법들은 하나 이상의 비디오 코딩 표준들과 함께 사용하기 위해 구성될 수도 있다. 예를 들어, 고 효율 비디오 코딩(HEVC)은, ITU-T 비디오 코딩 전문가 그룹(Video Coding Experts Group, VCEG) 및 ISO/IEC 동 화상 전문가 그룹(Motion Picture Experts Group, MPEG)의 JCT-VC(Joint Collaboration Team on Video Coding)에 의해 개발된 새로운 비디오 코딩 표준이다. 최근의 HEVC 텍스트 규격이 www.itu.int/rec/T-REC-H.265-201304-I에서 입수 가능한 『Bross et al., "High Efficiency Video Coding (HEVC) Text Specification", ('HEVC Version 1')』에서 설명되어 있다.
- [0020] HEVC 프레임워크에 관해, 일 예로서, 팔레트 기반 코딩 기법들이 코딩 유닛(coding unit, CU) 모드로서 사용되

도록 구성될 수도 있다. 다른 예들에서, 팔레트 기반 코딩 기법들은 HEVC의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 따라서, CU 모드의 맥락에서 설명되는 다음의 개시된 프로세스들의 모두가, 부가적으로 또는 대안으로, PU에 적용될 수도 있다. 그러나, 이들 HEVC 기반 예들은 본원에서 설명되는 팔레트 기반 코딩 기법들의 제약 또는 제한으로서 간주되지 않아야 하는데, 이러한 기법들이 독립적으로 또는 다른 기존의 또는 아직 개발되지 않은 시스템들/표준들의 부분으로서 작동하기 위해 적용될 수도 있어서이다. 이들 경우들에서, 팔레트 코딩을 위한 유닛은 정사각형 블록들, 직사각형 블록들 또는 심지어 비-직사각형 형상의 지역들일 수 있다.

[0021] 본 개시물의 양태들은 각각의 컬러 값에 각각 대응하는 블록의 화소들을 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩하는 것에 관련된다. 비디오 코더가 맵의 인덱스 값들을 특정 순서로 코딩할 수도 있는데, 그 특정 순서는 스캔 순서 또는 스캔 방향이라고 지칭될 수도 있다. 예를 들면, 아래의 도 6a의 예에 관해 더 상세히 설명되는 바와 같이, 맵의 인덱스들은 래스터 스캔 순서라고 지칭될 수도 있는 것에서 상단에서 하단으로, 좌측에서 우측으로 스캔될 수도 있다. 래스터 스캔 순서에서, 현재 라인의 첫 번째 인덱스는 이전 라인의 마지막 인덱스 뒤에 바로 스캔될 수도 있다.

[0022] 그러나, 일부 예들에서, 래스터 스캔 순서로 인덱스들을 스캔하는 것은 효율적이지 않을 수도 있다. 예를 들면, 비디오 데이터 블록의 제 1 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 비디오 데이터 블록의 제 2 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 제 1 라인의 마지막 화소가 제 2 라인의 마지막 화소에 인접하고, 제 1 에지 및 제 2 에지가 평행한 경우, 제 1 라인에서의 마지막 화소는 제 2 라인에서의 마지막 화소와 동일하지만 제 2 라인에서의 제 1 화소와는 상이하다. 이 상황 (즉, 제 1 라인에서의 마지막 화소가 제 2 라인에서의 마지막 화소와 동일 또는 유사하지만 제 2 라인에서의 제 1 화소와는 상이한 경우) 은 컴퓨터 생성된 스크린 콘텐츠에서 비디오 콘텐츠의 다른 유형들보다 더 빈번하게 발생할 수도 있다. 다르게 말하면, 스크린 콘텐츠에 대해, 블록의 동일한 에지에 위치한 화소들이 블록의 대향 에지들에 위치한 화소들과는 동일 또는 유사한 값들을 가질 더 높은 확률이 있을 수도 있다. 래스터 스캐닝이 이러한 상관을 이용하지 못할 수도 있다.

[0023] 본 개시물의 하나 이상의 기법들에 따라, 비디오 코더가 맵을 인코딩하는 경우 맵형 스캔 순서를 이용할 수도 있다. 예를 들면, 위의 예를 계속하여 그리고 도 6b의 예에 관해 아래에서 더욱 상세히 설명되는 바와 같이, 비디오 인코더 (20) 는 제 2 라인의 마지막 화소가 스캔 순서에서 제 1 라인의 마지막 화소를 바로 뒤따라 스캔되도록 맵을 코딩할 수도 있다. 이런 식으로, 비디오 인코더 (20) 는 런 길이 코딩의 효율을 개선할 수도 있다.

[0024] 도 1은 본 개시물의 기법들을 이용할 수도 있는 일 예의 비디오 코딩 시스템 (10) 을 도시하는 블록도이다. 본원에서 사용되는 바와 같이, "비디오 코더"라는 용어는 비디오 인코더들 및 비디오 디코더들 양쪽 모두를 일반적으로 지칭한다. 본 개시물에서, "비디오 코딩" 또는 "코딩"이란 용어들은 비디오 인코딩 또는 비디오 디코딩을 일반적으로 지칭할 수도 있다. 비디오 코딩 시스템 (10) 의 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물에서 설명되는 다양한 예들에 따라 팔레트 기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스들의 예들을 나타낸다. 예를 들어, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 팔레트 기반 코딩 또는 비-팔레트 기반 코딩 중 어느 하나를 사용하여 비디오 데이터의 다양한 블록들, 이를테면 HEVC 코딩에서의 CU들 또는 PU들을, 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트 기반 코딩 모드들은 다양한 인터 예측 시간적 코딩 모드들 또는 인터 예측 공간적 코딩 모드들, 이를테면 HEVC 버전 1에 의해 특정된 다양한 코딩 모드들을 지칭할 수도 있다.

[0025] 도 1에 도시된 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 와 목적지 디바이스 (14) 를 구비한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 생성한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치라고 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치라고 지칭될 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.

[0026] 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예컨대, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 차량내 컴퓨터들 등

을 포함한 다양한 범위의 디바이스들을 포함할 수도 있다.

[0027] 목적지 디바이스 (14) 는 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 채널 (16) 을 통해 수신할 수도 있다. 채널 (16) 은 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 이동시킬 수 있는 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 하나의 예에서, 채널 (16) 은 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접 실시간으로 송신하는 것을 가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 통신 표준, 이를테면 무선 통신 프로토콜에 따라 변조할 수도 있고, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 이를테면 라디오 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 의 일부를 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.

[0028] 다른 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 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 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 국부적으로 액세스되는 데이터 저장 매체들을 포함할 수도 있다.

[0029] 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. 이 예에서, 목적지 디바이스 (14) 는 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터를 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버들 (예컨대, 웹사이트용), 파일 전송 프로토콜 (file transfer protocol, FTP) 서버들, 네트워크 부속 스토리지 (network attached storage; NAS) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.

[0030] 목적지 디바이스 (14) 는 표준 데이터 접속, 이를테면 인터넷 접속을 통해, 인코딩된 비디오 데이터에 액세스할 수도 있다. 예의 유형들의 데이터 접속들은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽 모두의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.

[0031] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 제한되지 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 중 임의의 것의 지원 하의 비디오 코딩, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0032] 도 1은 단지 일 예이고 본 개시물의 기법들은 인코딩 및 디코딩 디바이스들 간에 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들 (예컨대, 비디오 인코딩 또는 비디오 디코딩) 에 적용될 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 추출되며, 네트워크를 통해 스트리밍되는 등등이 된다. 비디오 인코딩 디바이스가 데이터를 인코딩하고 메모리에 저장할 수도 있으며, 그리고/또는 비디오 디코딩 디바이스가 메모리로부터 데이터를 추출하고 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은, 서로 통신하지 않지만 단순히 데이터를 메모리에 인코딩하고 및/또는 메모리로부터 데이터를 추출하고 디코딩하는 디바이스들에 의해 수행된다.

[0033] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 구비한다. 일부 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 구비할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오 데이터를 포함한 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하는 비디오 피드 인터페이스, 및/또는 비디오 데이터를 생성하는 컴퓨터 그래픽 시스템, 또는 비디오 데이터의 이러한 소스들의 조합을 포함할

수도 있다.

- [0034] 비디오 인코더 (20) 는 비디오 소스 (18) 로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12) 는 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 직접 송신할 수도 있다. 다른 예들에서, 인코딩된 비디오 데이터는 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14) 에 의한 나중의 액세스를 위해 저장 매체 또는 파일 서버 상에 또한 저장될 수도 있다.
- [0035] 도 1의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 구비한다. 일부 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 구비한다. 입력 인터페이스 (28) 는 채널 (16) 을 통해 인코딩된 비디오 데이터를 수신할 수도 있다. 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 또는 그것 외부에 있을 수도 있다. 대체로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32) 는 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들을 포함할 수도 있다.
- [0036] 본 개시물은 다른 디바이스, 이를테면 비디오 디코더 (30) 에 특정 정보를 "시그널링하는" 또는 "송신하는" 비디오 인코더 (20) 에 일반적으로 관련이 있을 수도 있다. "시그널링"또는 "송신"이란 용어는 압축된 비디오 데이터를 디코딩하는데 사용되는 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 일반적으로 지칭할 수도 있다. 이러한 통신은 실시간 또는 거의 실시간으로 일어날 수도 있다. 대안으로, 이러한 통신은, 인코딩 시에 선택스 엘리먼트들을 인코딩된 비트스트림으로 컴퓨터 판독가능 저장 매체에 저장하고 그 선택스 엘리먼트들이 이 매체에 저장된 후의 임의의 시간에 디코딩 디바이스에 의해 추출될 수도 있는 경우에 일어날 바와 같이 어떤 기간 (span of time) 에 걸쳐 일어날 수도 있다. 따라서, 비디오 디코더 (30) 가 특정 정보를 "수신하는" 것으로서 지칭될 수도 있지만, 정보의 수신은 실시간 또는 거의 실시간으로 반드시 일어날 필요는 없고 저장 후의 언젠가에 매체로부터 추출될 수도 있다.
- [0037] 비디오 인코더 (20) 와 비디오 디코더 (30) 각각은 다양한 적합한 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 개별 로직, 하드웨어, 또는 그것들의 임의의 조합 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 부분적으로 소프트웨어로 구현되면, 디바이스가 적합한 비일시적 (non-transitory) 컴퓨터 판독가능 저장 매체 내에 소프트웨어에 대한 명령을 저장할 수도 있고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행하여 본 개시물의 기법들을 수행할 수도 있다. 전술한 바 (하드웨어, 소프트웨어, 하드웨어 및 소프트웨어의 조합 등을 포함) 중 임의의 것은 하나 이상의 프로세서들이라고 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC) 의 일부로서 개별 디바이스 내에 통합될 수도 있다.
- [0038] 일부 예들에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 비디오 압축 표준, 이를테면 위에서 언급되고 HEVC 버전 1에서 설명된 HEVC 표준에 따라 동작한다. 기본 HEVC 표준 외에도, HEVC에 대한 스케일러블 비디오 코딩, 멀티뷰 비디오 코딩, 및 3D 코딩 확장본들을 생성하기 위한 진행중인 노력들이 있다. 덧붙여서, 본 개시물에서 설명되는 바와 같은, 예컨대, 팔레트 기반 코딩 모드들은, HEVC 표준의 확장본에 제공될 수도 있다. 일부 예들에서, 팔레트 기반 코딩에 대한 본 개시물에서 설명된 기법들은 다른 비디오 코딩 표준들, 이를테면 ITU-T-H.264/AVC 표준 또는 장래의 표준들에 따라 동작하도록 구성되는 인코더들 및 디코더들에 적용될 수도 있다. 따라서, HEVC 코덱에서의 코딩 유닛들 (CU들) 또는 예측 유닛들 (prediction units, PU들) 의 코딩을 위한 팔레트 기반 코딩 모드의 애플리케이션이 예의 목적들을 위해 설명된다.
- [0039] HEVC 및 다른 비디오 코딩 표준들에서, 비디오 시퀀스가 일련의 픽처들을 통상 포함한다. 픽처들은 "프레임들"이라고 또한 지칭될 수도 있다. 픽처가, S_L , S_{Cb} 및 S_{Cr} 로 표시되는 3 개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 색차 샘플들의 2차원 어레이이다. S_{Cr} 은 Cr 색차 샘플들의 2차원 어레이이다. 색차 샘플들은 본원에서 "크로마" 샘플들이라고 또한 지칭될 수도 있다. 다른 경우들에서, 픽처가 모노크롬일 수도 있고 루마 샘플들의 어레이만을 포함할 수도 있다.
- [0040] 픽처의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛 (coding tree unit, CTU) 들의 세트를 생성할 수도 있다. CTU들의 각각은, 루마 샘플들의 코딩 트리 블록 (coding tree block), 크로마 샘플들의 2 개의 대응 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 선택스 구조들일 수도 있다. 코딩 트리 블록이 샘플들의 $N \times N$ 블록일 수도 있다. CTU가 "트리 블록" 또는 "최대 코딩 유닛

(largest coding unit; LCU)"이고 또한 지칭될 수도 있다. HEVC의 CTU들은 다른 표준들, 이를테면 H.264/AVC의 매크로블록들과 대체로 유사할 수도 있다. 그러나, CTU가 특정 사이즈로 반드시 제한되는 것은 아니고 하나 이상의 코딩 유닛들(CU들)을 포함할 수도 있다. 슬라이스가 래스터 스캔으로 연속하여 순서화된 정수 수의 CTU들을 포함할 수도 있다.

[0041] 코딩된 슬라이스가 슬라이스 헤더와 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.

[0042] 본 개시물은 하나 이상의 샘플 블록들 및 그 하나 이상의 샘플 블록들을 코딩하는데 사용된 선택스 구조들을 지칭하기 위해 "비디오 유닛" 또는 "비디오 블록" 또는 "블록"이란 용어를 사용할 수도 있다. 비디오 유닛들 또는 블록들의 예의 유형들은 CTU들, CU들, PU들, 변환 유닛들(TU들), 매크로블록들, 매크로블록 구획들을 포함할 수도 있다. 일부 맥락들에서, PU들의 논의는 매크로블록 구획들의 매크로블록들의 논의와 교환될 수도 있다.

[0043] 코딩된 CTU를 생성하기 위해, 비디오 인코더(20)는 CTU의 코딩 트리 블록들에 대해 쿼드트리 구획화를 재귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들로 나눌 수도 있으며, 그래서 그 이름이 "코딩 트리 유닛들"이다. 코딩 블록이 샘플들의 NxN 블록이다. CU가, 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 대응하는 두 개의 코딩 블록들과, 그 코딩 블록들의 샘플들을 코딩하는데 사용된 선택스 구조들일 수도 있다. 비디오 인코더(20)는 CU의 코딩 블록을 하나 이상의 예측 블록들로 구획화할 수도 있다. 예측 블록이 동일한 예측이 적용되는 샘플들의 직사각형(즉, 정사각형 또는 정사각형 아닌) 블록일 수도 있다. CU의 예측 유닛(PU)이 픽처의 루마 샘플들의 예측 블록, 크로마 샘플들의 두 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용된 선택스 구조들일 수도 있다. 비디오 인코더(20)는 CU의 각각의 PU의 루마, Cb 및 Cr 예측 블록들에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다.

[0044] 비디오 인코더(20)는 PU에 대한 예측 블록들을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수도 있다. 비디오 인코더(20)가 PU의 예측 블록들을 생성하기 위해 인트라 예측을 사용하면, 비디오 인코더(20)는 그 PU에 연관된 픽처의 디코딩된 샘플들에 기초하여 그 PU의 예측 블록들을 생성할 수도 있다.

[0045] 비디오 인코더(20)가 PU의 예측 블록들을 생성하기 위해 인터 예측을 사용하면, 비디오 인코더(20)는 그 PU에 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 그 PU의 예측 블록들을 생성할 수도 있다. 비디오 인코더(20)는 PU의 예측 블록들을 생성하기 위해 단-예측(uni-prediction) 또는 양-예측(bi-prediction)을 사용할 수도 있다. 비디오 인코더(20)가 단예측을 사용하여 PU에 대한 예측 블록들을 생성하는 경우, PU는 단일 MV를 가질 수도 있다. 비디오 인코더(20)가 양예측을 사용하여 PU에 대한 예측 블록들을 생성하는 경우, PU는 두 개의 MV들을 가질 수도 있다.

[0046] 비디오 인코더(20)가 CU의 하나 이상의 PU들에 대한 예측 루마, Cb 및 Cr 블록들을 생성한 후, 비디오 인코더(20)는 그 CU에 대한 루마 잔차 블록을 생성할 수도 있다. CU의 루마 잔차 블록에서의 각각의 샘플은 CU의 예측 루마 블록들 중 하나의 예측 루마 블록에서의 루마 샘플 및 CU의 원래의 루마 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낸다. 덧붙여서, 비디오 인코더(20)는 CU에 대한 Cb 잔차 블록을 생성할 수도 있다. CU의 Cb 잔차 블록에서의 각각의 샘플은 CU의 예측 Cb 블록들 중 하나의 예측 Cb 블록에서의 Cb 샘플과 CU의 원래의 Cb 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다. 비디오 인코더(20)는 CU에 대한 Cr 잔차 블록을 또한 생성할 수도 있다. CU의 Cr 잔차 블록에서의 각각의 샘플은 CU의 예측 Cr 블록들 중 하나의 예측 Cr 블록에서의 Cr 샘플 및 CU의 원래의 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다.

[0047] 더욱이, 비디오 인코더(20)는 쿼드트리 구획화를 사용하여 CU의 루마, Cb 및 Cr 잔차 블록들을 하나 이상의 루마, Cb 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록이 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU의 변환 유닛(PU)이 루마 샘플들의 변환 블록, 크로마 샘플들의 대응하는 2 개의 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용된 선택스 구조들일 수도 있다. 따라서, CU의 각각의 TU는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록에 연관될 수도 있다. TU에 연관된 루마 변환 블록은 CU의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU의 Cr 잔차 블록의 서브-블록일 수도 있다.

- [0048] 비디오 인코더 (20) 는 하나 이상의 변환들을 TU의 루마 변환 블록에 적용하여 그 TU에 대한 루마 계수 블록을 생성할 수도 있다. 계수 블록이 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수가 스칼라 양일 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU의 Cb 변환 블록에 적용하여 TU에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU의 Cr 변환 블록에 적용하여 TU에 대한 Cr 계수 블록을 생성할 수도 있다.
- [0049] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후, 비디오 인코더 (20) 는 그 계수 블록을 양자화할 수도 있다. 양자화는 변환 계수들이 그 변환 계수들을 표현하는데 사용된 데이터의 양을 가능한 한 줄이도록 양자화되어서, 추가의 압축을 제공하는 프로세스를 일반적으로 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들에 대해 컨텍스트 적응 이진 산술 코딩 (Context-Adaptive Binary Arithmetic Coding; CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩된 선택스 엘리먼트들을 비트스트림으로 출력할 수도 있다.
- [0050] 비디오 인코더 (20) 는 엔트로피 인코딩된 선택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 그 비트스트림은 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 그 비트스트림은 네트워크 추상화 계층 (network abstraction layer, NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하고 원시 바이트 시퀀스 페이로드 (raw byte sequence payload, RBSP) 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 선택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 유형 코드는 NAL 유닛의 유형을 나타낸다. RBSP가 NAL 유닛 내에 캡슐화되는 정수 수의 바이트들을 포함하는 선택스 구조일 수도 있다. 일부 경우들에서, RBSP가 영 비트들을 포함한다.
- [0051] 상이한 유형들의 NAL 유닛들이 상이한 유형들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, 제 1 유형의 NAL 유닛이 픽처 파라미터 세트 (picture parameter set, PPS) 에 대한 RBSP를 캡슐화할 수도 있으며, 제 2 유형의 NAL 유닛이 코딩된 슬라이스에 대한 RBSP를 캡슐화할 수도 있으며, 제 3 유형의 NAL 유닛이 SEI에 대한 RBSP를 캡슐화할 수도 있다는 등등이다. 비디오 코딩 데이터에 대한 RBSP들 (파라미터 세트들 및 SEI 메시지들에 대한 RBSP과는 대조적임) 을 캡슐화하는 NAL 유닛들은, 비디오 코딩 계층 (video coding layer, VCL) NAL 유닛들이라고 지칭될 수도 있다.
- [0052] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 덧붙여서, 비디오 디코더 (30) 는 비트스트림으로부터 선택스 엘리먼트들을 디코딩하기 위해 그 비트스트림을 파싱할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 선택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하는 프로세스는 비디오 인코더 (20) 에 의해 수행된 프로세스에 일반적으로 역일 수도 있다.
- [0053] 예를 들면, 비디오 디코더 (30) 는 현재 CU의 PU들에 대한 예측 샘플 블록들을 결정하기 위해 그 PU들의 MV들을 사용할 수도 있다. 덧붙여서, 비디오 디코더 (30) 는 현재 CU의 TU들에 연관된 변환 계수 블록들을 역 양자화할 수도 있다. 비디오 디코더 (30) 는 현재 CU의 TU들에 연관된 변환 블록들을 복원하기 위해 변환 계수 블록들에 대해 역 변환들을 수행할 수도 있다. 비디오 디코더 (30) 는 현재 CU의 PU들에 대한 예측 샘플 블록들의 샘플들을 현재 CU의 TU들의 변환 블록들의 대응하는 샘플들에 가산함으로써 현재 CU의 코딩 블록들을 복원할 수도 있다. 픽처의 각각의 CU에 대한 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 그 픽처를 복원할 수도 있다.
- [0054] 일부 예들에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 팔레트 기반 코딩을 수행하도록 구성될 수도 있다. 예를 들어, 팔레트 기반 코딩에서는, 위에서 설명된 인트라 예측 또는 인터 예측 코딩 기법들을 수행하는 것이 아니라, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 특정 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 표현하기 위한 컬러들의 테이블로서 이른바 팔레트를 코딩할 수도 있다. 각각의 화소는 화소의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 화소 값을 팔레트에서의 적절한 값에 관련시키는 인덱스를 코딩할 수도 있다.
- [0055] 위의 예에서, 비디오 인코더 (20) 는 블록에 대한 팔레트를 결정하며, 각각의 화소의 값을 표현하기 위해 팔레트에서 엔트리의 위치를 찾아내고, 화소 값을 팔레트에 관련시키는 화소들에 대한 인덱스 값들로 팔레트를 인코딩함으로써 비디오 데이터 블록을 인코딩할 수도 있다. 비디오 디코더 (30) 는, 인코딩된 비트스트림으로부터

터, 블록에 대한 팔레트, 뿐만 아니라 그 블록의 화소들에 대한 인덱스 값들을 획득할 수도 있다. 비디오 디코더 (30) 는 블록의 화소 값들을 복원하기 위해 화소들의 인덱스 값들을 팔레트의 엔트리들에 매핑할 수도 있다.

[0056] 본 개시물의 양태들은 팔레트 도출을 지향한다. 하나의 예로서, 비디오 인코더 (20) 는 현재 블록에서의 화소들의 히스토그램을 도출함으로써 현재 블록에 대한 팔레트를 도출할 수도 있다. 일부 예들에서, 히스토그램은 $H = \{(v_i, f_i), i = \{0, 1, 2, \dots, M\}\}$ 으로서 표현될 수도 있으며 여기서 $M + 1$ 은 현재 블록에서의 상이한 화소 값들의 수이며, v_i 는 화소 값이며, f_i 는 v_i 의 발생 수 (즉, 현재 블록에서의 얼마나 많은 화소들이 화소 값 v_i 를 갖는지) 이다. 그런 예들에서, 히스토그램은 화소 값이 현재 블록에서 발생하는 횟수를 일반적으로 나타낸다.

[0057] 비디오 인코더 (20) 는 히스토그램을 도출하는 경우 하나 이상의 변수들을 초기화할 수도 있다. 하나의 예로서, 비디오 인코더 (20) 는 팔레트 인덱스 (idx) 를 0으로 초기화 (즉, $idx=0$ 으로 설정) 할 수도 있다. 다른 예로서, 비디오 인코더 (20) 는 팔레트 (P) 를 엠프티가 되게 초기화 (즉, $P = \emptyset$, $j = 0$ 으로 설정.) 할 수도 있다.

[0058] 비디오 인코더 (20) 는 히스토그램을, 예컨대, 내림 차순으로 정렬할 수도 있어서, 더 많은 발생들을 갖는 화소들이 값들의 리스트의 앞 근처에 배치된다. 예를 들면, 비디오 인코더 (20) 는 f_i 의 내림 차순에 따라 H 를 정렬할 수도 있고 순서화된 리스트는 $H_0 = \{(u_i, f_i), i = \{0, 1, 2, \dots, M\}, f_i \geq f_{i+1}\}$ 로서 표현될 수도 있다. 이 예에서, 순서화된 리스트는 가장 빈번하게 발생하는 화소 값들을 리스트의 앞 (상단) 에 그리고 최소로 빈번하게 발생하는 화소 값들을 리스트의 뒤 (하단) 에 포함한다.

[0059] 비디오 인코더 (20) 는 히스토그램으로부터의 하나 이상의 엔트리들을 팔레트에 복사할 수도 있다. 하나의 예로서, 비디오 인코더 (20) 는 최대 빈도수를 갖는 히스토그램에서의 엔트리를 팔레트에 삽입할 수도 있다. 예를 들면, 비디오 인코더 (20) 는 (j, u_j) 를 팔레트 P 에 삽입 (즉, $P = P \cup \{idx, u_j\}$) 할 수도 있다. 일부 예들에서, 엔트리를 팔레트에 삽입한 후, 비디오 인코더 (20) 는 팔레트 속으로의 삽입을 위한 두 번째로 큰 빈도수를 갖는 히스토그램에서의 엔트리를 평가할 수도 있다. 예를 들면, 비디오 인코더 (20) 는 $idx = idx + 1$, $j = j + 1$ 로 설정할 수도 있다.

[0060] 비디오 인코더 (20) 는 두 번째로 큰 빈도수를 갖는 엔트리 (즉, u_{j+1}) 가 팔레트에서의 임의의 화소 (즉, x) 의 인근 내에 있는지의 여부 (즉, $Distance(u_{j+1}, x) < Thresh$) 를 결정할 수도 있다. 예를 들면, 비디오 인코더 (20)는 그 엔트리의 값이 팔레트에서의 임의의 화소의 값의 임계 거리 내에 있는지의 여부를 결정함으로써 그 엔트리가 팔레트에서의 임의의 화소의 인근 내에 있는지의 여부를 결정할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 거리 함수를 유연하게 선택할 수도 있다. 하나의 예로서, 비디오 인코더 (20) 는 세 개의 컬러 성분들 (예컨대, 휘도, 청색 색상 색차, 및 적색 색상 (hue) 색차의 각각), 또는 하나의 컬러 성분 (예컨대, 휘도, 청색 색상 색차, 또는 적색 색상 색차 중 하나) 의 차들의 절대값 합 (sum of absolute differences, SAD) 또는 제곱 예측 오차들의 합 (sum of squared errors of prediction, SSE) 으로서 거리 함수를 선택할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 임계 값 (Thresh) 을 유연하게 선택할 수도 있다. 하나의 예로서, 비디오 인코더 (20) 는 임계 값을 현재 블록의 양자화 파라미터 (quantization parameter, QP) 에 의존하도록 선택할 수도 있다. 다른 예로서, 비디오 인코더 (20) 는 임계 값을 idx 의 값 또는 j 의 값에 의존하도록 선택할 수도 있다.

[0061] 두 번째로 큰 빈도수를 갖는 엔트리 (즉, u_{j+1}) 가 팔레트에서의 임의의 화소의 인근 내에 있다고 비디오 인코더 (20) 가 결정하면, 비디오 인코더 (20) 는 그 엔트리를 히스토그램에 삽입하지 않을 수도 있다. 두 번째로 큰 빈도수를 갖는 엔트리 (즉, u_{j+1}) 가 팔레트에서의 임의의 화소의 인근 내에 있지 않다고 비디오 인코더 (20) 가 결정하면, 비디오 인코더 (20) 는 그 엔트리를 히스토그램에 삽입할 수도 있다.

[0062] 비디오 인코더 (20) 는 하나 이상의 조건들이 충족되기까지 엔트리들을 팔레트에 계속 삽입할 수도 있다. 일부 예의 조건들은 $idx = M$ 인 경우, $j = M$ 인 경우, 또는 팔레트의 사이즈가 미리 정의된 값보다 더 큰 경우이다.

[0063] 팔레트 기반 코딩이 특정한 양의 시그널링 오버헤드를 가질 수도 있다. 예를 들어, 다수의 비트들이 팔레트

의 특성들, 이를테면 팔레트의 사이즈, 뿐만 아니라 팔레트 자체를 시그널링하기 위해 필요할 수도 있다. 덧붙여서, 다수의 비트들이 블록의 화소들에 대한 인덱스 값들을 시그널링하기 위해 필요할 수도 있다. 본 개시물의 기법들은, 일부 예들에서, 이러한 정보를 시그널링하는데 필요한 비트들의 수를 감소시킬 수도 있다.

예를 들어, 본 개시물에서 설명되는 기법들은 팔레트 기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 예측하는 것, 팔레트들을 도출하는 것, 그리고 팔레트 기반 코딩 맵들 및 다른 선택스 엘리먼트들을 송신하는 것 중 하나 이상의 다양한 조합들을 위한 기법들을 포함할 수도 있다.

[0064] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트를 다른 팔레트를 사용하여 예측할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 1 화소 값들을 나타내는 제 1 엔트리들을 갖는 제 1 팔레트를 결정할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 그 다음에, 제 1 팔레트의 제 1 엔트리들에 기초하여, 제 2 팔레트의 제 2 화소 값들을 나타내는 하나 이상의 제 2 엔트리들을 결정할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 2 팔레트를 사용하여 비디오 데이터 블록의 화소들을 또한 코딩할 수도 있다.

[0065] 제 1 팔레트에서의 엔트리들에 기초하여 제 2 팔레트의 엔트리들을 결정하는 경우, 비디오 인코더 (20) 는 다양한 선택스 엘리먼트들을 인코딩할 수도 있는데, 그 다양한 선택스 엘리먼트들은 비디오 디코더에 의해 제 2 팔레트를 복원하는데 사용될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 전체 팔레트 (또는 별도의 팔레트를 갖는 비디오 데이터의 각각의 컬러 성분, 예컨대, Y, Cb, Cr, 또는 Y, U, V, 또는 R, G, B의 경우의 팔레트들) 가 현재 코딩되고 있는 블록의 하나 이상의 이웃 블록들로부터 복사됨을 나타내기 위해 비트스트림에서 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 현재 블록의 현재 팔레트의 엔트리들이 예측되는 (예컨대, 복사되는) 팔레트는 예측 팔레트라고 지칭될 수도 있다. 예측 팔레트는 공간적으로 이웃하는 블록들 및/또는 블록들의 특정 스캔 순서에서의 이웃하는 블록들을 포함하는 하나 이상의 이웃 블록들로부터의 팔레트 엔트리들을 포함할 수도 있다. 예를 들어, 이웃 블록들은 현재 코딩되고 있는 블록의 좌측 (좌측 이웃 블록) 또는 상측 (상부 이웃 블록) 에 공간적으로 위치될 수도 있다. 다른 예에서, 비디오 인코더 (20) 는 현재 블록의 인과적 이웃 (causal neighbor) 에서의 가장 빈번한 샘플 값들을 사용하여 예측 팔레트 엔트리들을 결정할 수도 있다. 다른 예에서, 이웃 블록들은 블록들을 코딩하는데 사용되는 특정 스캔 순서에 따라 현재 코딩되고 있는 블록에 이웃할 수도 있다. 다시 말하면, 이웃 블록들은 스캔 순서에서 현재 블록보다 먼저 코딩된 하나 이상의 블록들일 수도 있다. 비디오 인코더 (20) 는 팔레트(들)가 복사되는 이웃 블록들의 로케이션을 나타내는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다.

[0066] 일부 예들에서, 팔레트 예측이 엔트리-방식으로 수행될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 예측 팔레트의 각각의 엔트리에 대해, 팔레트 엔트리가 현재 블록에 대한 팔레트에 포함되는지의 여부를 나타내기 위해 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 가 현재 블록에 대한 팔레트의 엔트리를 예측하지 않는다면, 비디오 인코더 (20) 는 비-예측된 엔트리들, 뿐만 아니라 이러한 엔트리들의 수를 특정하기 위해 하나 이상의 추가적인 선택스 엘리먼트들을 인코딩할 수도 있다.

[0067] 위에서 설명된 선택스 엘리먼트들은 팔레트 예측 벡터라고 지칭될 수도 있다. 예를 들어, 위에서 지적했듯이, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 이웃 블록들로부터의 하나 이상의 팔레트들 (통칭하여 참조 팔레트이라고 지칭됨) 에 기초하여 현재 블록에 대한 팔레트를 예측할 수도 있다. 참조 팔레트를 생성하는 경우, 선입선출 (first-in first-out, FIFO) 가 최신 팔레트를 대기행렬 (queue) 의 앞에 추가함으로써 사용될 수도 있다. 대기행렬이 미리 정의된 임계값을 초과하면, 가장 오래된 엘리먼트들이 팝 아웃 (pop out) 될 수도 있다. 새로운 엘리먼트들을 대기행렬의 앞으로 푸시한 후, 전지작업 (pruning) 프로세스가 대기행렬의 시작부분에서부터 카운트하여 중복된 엘리먼트들을 제거하도록 적용될 수도 있다. 구체적으로는, 일부 예들에서, 참조 팔레트에서의 화소 값들이 현재 팔레트에 대해 재사용되는지의 여부를 나타내기 위해 0-1 벡터를 비디오 인코더 (20) 는 인코딩할 수도 있고 (고 비디오 디코더 (30) 는 디코딩할 수도 있다). 일 예로서, 아래의 표 1의 예에서 도시된 바와 같이, 참조 팔레트가 여섯 개의 아이템들 (예컨대, 여섯 개의 인덱스 값들 및 각각의 화소 값들) 을 포함할 수도 있다.

표 1

인덱스	화소 값
0	v_0
1	v_1

2	v_2
3	v_3
4	v_4
5	v_5

표 1

예시의 목적을 위한 일 예에서, 비디오 인코더 (20) 는 v_0 , v_2 , v_3 , v_4 , 및 v_5 가 현재 팔레트에서 재사용되는 반면 v_1 는 재사용되지 않음을 나타내는 벡터 (1, 0, 1, 1, 1, 1) 을 시그널링할 수도 있다. v_0 , v_2 , v_3 , v_4 , 및 v_5 를 재사용하는 것 외에도, 비디오 인코더 (20) 는 5와 6에 의한 인덱스들을 이용하여 현재 팔레트에 두 개의 새로운 아이тем들을 추가할 수도 있다. 이 예에 대한 현재 팔레트는 아래의 표 2에서 도시되어 있다.

표 2

Pred 플래그	인덱스	화소 값
1	0	v_0
0		
1	1	v_2
1	2	v_3
1	3	v_4
1	4	v_5
	5	u_0
	6	u_1

표 2

팔레트 예측 0-1 벡터를 코딩하기 위해, 벡터에서의 각각의 아이тем에 대해, 비디오 인코더 (20) 는 그것의 값을 표현하기 위해 하나의 비트를 코딩할 수도 있고 그 비트는 CABAC 바이패스를 사용하여 코딩될 수도 있다. CABAC에 관해, 일 예로서, 비디오 코더 (비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 비디오 데이터의 블록에 연관된 심볼들을 코딩하기 위해 확률 모델 (또한 콘텍스트 모델이라고도 지칭됨) 을 선택할 수도 있다.

예를 들어, 인코더에서, 타겟 심볼이 확률 모델을 사용함으로써 코딩될 수도 있다. 디코더에서, 타겟 심볼이 확률 모델을 사용함으로써 파싱될 수도 있다. 일부 경우들에서, 빈들은 콘텍스트 적응 및 비-콘텍스트 적응 코딩의 조합을 사용하여 코딩될 수도 있다. 예를 들어, 비디오 코더가 하나 이상의 빈들에 대한 정규 산술 코딩 콘텍스트 적응 프로세스를 바이패스, 또는 생략하기 위해 바이패스 모드를 사용하면서 다른 빈들에 대해 콘텍스트 적응 코딩을 사용할 수도 있다. 그런 예들에서, 비디오 코더는 빈들을 바이패스 코딩하기 위해 고정 확률 모델을 사용할 수도 있다. 다시 말하면, 바이패스 코딩된 빈들은 상황 또는 확률 업데이트들을 포함하지 않는다. 대체로, 빈들을 콘텍스트 코딩하는 것은 콘텍스트 코딩 모드를 사용하여 빈들을 코딩하는 것을 지칭할 수도 있다. 비슷하게, 빈들을 바이패스 코딩하는 것은 바이패스 코딩 모드를 사용하여 빈들을 코딩하는 것을 지칭할 수도 있다.

그러나, 팔레트들 간의 높은 상관으로 인해, 팔레트 예측 벡터에서의 영 및 일은 동등하게 분포되지 않을 수도 있으며, 이는 코딩 효율에 영향을 미칠 수도 있는데, 예컨대, 0-1 벡터가 팔레트들 간의 높은 상관을 이용하지 못할 수도 있기 때문이다. 덧붙여, 예측될 수 없는 팔레트 아이тем들의 수 (예컨대, 새로운 팔레트 엔트리들 (위의 표 2의 예에서의 u_0 및 u_1) 의 수) 는 각각의 비트가 CABAC 바이패스를 사용하여 코딩된 단항 코드 (unary code) 를 사용하여 2치화되고 시그널링될 수도 있다. 팔레트 예측 벡터에서의 문제들과 마찬가지로, 동일한 확률을 갖도록 단항 코드에서의 각각의 비트를 모델화하는 것은 차선일 수도 있는, 예컨대, 실제로, 단항 코드에서의 비트들은 상이한 확률들을 가질 수도 있기 때문이다.

본 개시물의 하나 이상의 기법들에 따라, 비디오 인코더 (20) 는, 예컨대, 이러한 빈들의 바이패스 코딩에 의존

하는 기법들에 비하여 코딩 효율을 개선하기 위해 콘텍스트(들)와 함께 CABAC를 사용하여, 팔레트 예측 벡터 및/또는 단항 코딩된 나머지 팔레트 사이즈를 코딩할 수도 있다. 일부 예들에서, 팔레트 예측 벡터에서의 아이템들은 단일 CABAC 콘텍스트를 공유할 수 있다. 이런 식으로, 복잡도는 감소될 수도 있다. 일부 예들에서, 단항 코드는 단일 콘텍스트를 사용할 수 있다. 또한 이런 식으로, 복잡도는 감소될 수도 있다.

[0076] 일부 예들에서, 비디오 인코더 (20) 는 고정 길이 코딩을 사용하여 인덱스 값들을 코딩할 수도 있다. 그런 예들에서, 비디오 인코더 (20) 는 고정 길이 코드에서의 각각의 비트를 CABAC 바이패스를 사용하여 코딩할 수도 있다. 그러나, 일부 경우들에서, 팔레트에서의 인덱스들의 수는 2의 거듭제곱이 아닐 수도 있다. 그런 경우들에서, 고정 길이 코드의 코드워드들은 완전히 이용되지 않을 수도 있다. 예를 들면, 팔레트 사이즈가 5이면, 인덱스들은 0, 1, 2, 3, 및 4가 될 것이다. 고정 길이 코드로 각각의 인덱스를 코딩하기 위해, 비디오 인코더 (20) 는 세 개의 비트들을 사용해야 할 것이다. 그러나, 세 개의 비트들을 사용하는 것에 의해, 비디오 인코더 (20) 는 세 개의 코드워드들 ($8-5 = 3$) 을 낭비할 수도 있으며, 이는 코딩 효율에 영향을 미칠 수 있다. 일부 경우들에서, 비디오 인코더 (20) 가 이스케이프 화소들을 코딩하는 경우 유사한 문제가 발생할 수도 있다. 본 개시물의 하나 이상의 기법들에 따라, 고정 길이 코딩을 사용하는 것과는 대조적으로, 비디오 인코더 (20) 는 인덱스 값들 및/또는 이스케이프 화소들을 코딩하기 위해 절단형 (truncated) 이진수 코드를 이용할 수도 있다.

[0077] 본 개시물의 다른 양태들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 가 화소 값들을 결정하는 것을 허용하는 맵을 구축하는 것 및/또는 송신하는 것에 관련된다. 예를 들어, 본 개시물의 다른 양태들은 특정 화소를 팔레트의 엔트리에 관련시키는 인덱스들의 맵을 구축하는 것 및/또는 송신하는 것에 관련된다.

[0078] 일부 예들에서, 비디오 인코더 (20) 는 블록의 화소들이 팔레트에서의 대응하는 값을 갖는지의 여부를 나타낼 수도 있다. 예시의 목적을 위한 일 예에서, 맵의 (i, j) 엔트리가 비디오 데이터 블록에서의 (i, j) 화소 포지션에 대응한다고 가정한다. 이 예에서, 비디오 인코더 (20) 는 블록의 각각의 화소 포지션에 대한 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 (i, j) 로케이션에서의 화소 값이 팔레트에서의 값들 중 하나임을 나타내기 위해 (i, j) 엔트리에 대해 1과 동일한 플래그를 설정할 수도 있다. 컬러가 팔레트에 포함되는 (즉, 플래그가 1과 동일한) 경우 비디오 인코더 (20) 는 팔레트에서의 컬러를 식별하는 (i, j) 엔트리에 대한 팔레트 인덱스를 나타내는 데이터를 또한 인코딩할 수도 있다. 화소의 컬러가 팔레트에 포함되지 않는 (즉, 플래그가 0과 동일한) 경우 비디오 인코더 (20) 는 그 화소에 대한 샘플 값을 나타내는 데이터를 또한 인코딩할 수도 있는데, 그 화소는 이스케이프 화소라고 지칭될 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 위에서 설명된 데이터를 획득하고 그 데이터를 사용하여 블록에서의 특정 로케이션에 대한 팔레트 인덱스 및/또는 화소 값을 결정할 수도 있다.

[0079] 일부 경우들에서는, 주어진 포지션에 있는 화소가 매핑되는 팔레트 인덱스 및 동일한 팔레트 인덱스에 매핑되어 있는 이웃 화소의 확률 간에 상관성이 있을 수도 있다. 다시 말하면, 화소가 특정 팔레트 인덱스에 매핑되는 경우, (공간적 로케이션 측면에서의) 하나 이상의 이웃 화소들이 동일한 팔레트 인덱스에 매핑될 확률은 비교적 높을 수도 있다.

[0080] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 비디오 데이터 블록의 하나 이상의 인덱스들을 비디오 데이터의 동일한 블록의 하나 이상의 인덱스들을 기준으로 결정 및 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 비디오 데이터 블록에서의 제 1 화소에 연관된 제 1 인덱스 값을 결정하도록 구성될 수도 있으며, 여기서 제 1 인덱스 값은 제 1 화소의 값을 팔레트의 엔트리에 연관시킨다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 1 인덱스 값에 기초하여, 비디오 데이터 블록에서의 하나 이상의 제 2 화소들에 연관된 하나 이상의 제 2 인덱스 값들을 결정하도록, 그리고 비디오 데이터 블록의 제 1 화소 및 하나 이상의 제 2 화소들을 코딩하도록 또한 구성될 수도 있다. 따라서, 이 예에서, 맵의 인덱스들은 맵의 하나 이상의 다른 인덱스들을 기준으로 코딩될 수도 있다.

[0081] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 맵의 인덱스들을 그 맵의 하나 이상의 다른 인덱스들을 기준으로 코딩하기 위해 여러 방법들을 사용할 수도 있다. 예를 들면,

[0082] http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=8765에서 입수 가능한 『Guo et al., "RCE4: Summary report of HEVC Range Extensions Core Experiments 4 (RCE4) on palette coding for screen content," Document JCTVC-P0035, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 16th Meeting: San Jose, US, 9-17 Jan. 2014』 (이하, "JCTVC-P0035") 는, 특히 스크린 콘텐츠를 코딩하는 경우 상당한 BD 레이트 감소를 성취하는 것으로 보고되었던 팔레트

모드 코딩의 두 가지 방법들을 설명한다.

- [0083] 맵의 인덱스들을 맵의 하나 이상의 다른 인덱스들을 기준으로 코딩하기 위해 JCTVC-P0035에 의해 설명된 제 1 방법은 세 개의 코딩 모드들을 제공한다. 각각의 화소 라인에 대해, 플래그가 코딩 모드를 나타내기 위해 시그널링될 수도 있다. 제 1 방법은 세 가지 모드들, 즉, 수평 모드, 수직 모드, 및 정상 모드를 가질 수도 있다. 덧붙여, 화소가 이스케이프 화소로서 분류된다면, 화소 값은 송신될 수도 있다. 수평 모드, 수직 모드, 및 정상 모드는 아래에서 도 7a 내지 도 7c를 참조하여 더 상세히 설명된다.
- [0084] 맵의 인덱스들을 맵의 하나 이상의 다른 인덱스들을 기준으로 코딩하기 위해 JCTVC-P0035에 의해 설명된 제 2 방법은 좌측 CU의 팔레트에 기초하여 현재 팔레트를 코딩하는 엔트리-방식 예측 스킴의 사용을 제공한다. 제 2 방법은 CU에서의 화소들이 래스터 스캔 순서에서 세 가지 모드들, 즉 "상측 복사 모드 (Copy above mode)", "런 모드", 및 "화소 모드"를 사용하여 인코딩된다는 것을 제공한다. "상측 복사 모드", "런 모드", 및 "화소 모드"는 아래에서 도 8a 내지 도 8c를 참조하여 더 상세히 설명된다.
- [0085] 본질적으로, JCTVC-P0035에 의해 설명된 위의 두 가지 방법들의 양쪽 모두는 이전에 복원된 화소들을 복사하는 유사한 구조를 공유한다. 런 길이가 코딩과 유사하게, 후보 시작 포지션 및 런-길이 (run-length) 가 코딩되어야 한다. 예를 들어, 제 1 방법에서, 네 개까지의 후보 시작 복사 포지션들이 있다. 수직 모드 및 수평 모드의 경우, 런-길이는 라인의 폭이 되도록 암시적으로 배정되며, 정상 모드의 경우, 런-길이는 1로 암시적으로 배정된다. 제 1 방법과는 달리, 제 2 방법에서의 런-길이 값은, 예컨대, 비트스트림에서 명시적으로 코딩된다. 일부 예들에서, 제 2 방법은 제 1 방법보다는 더 나은 코딩 효율을, 예컨대 그것의 더욱 유연한 구조로 인해 가질 수도 있다.
- [0086] 일부 예들에서, 비디오 인코더 (20) 는 동일한 인덱스 값에 매핑되는 주어진 스캔 순서에서의 다수의 연속적인 화소들을 나타내는 하나 이상의 인덱스 엘리먼트들을 인코딩할 수도 있다. 유사 값으로 된 인덱스 값들의 스트링은 "런"이라고 지칭될 수도 있고 JCTVC-P0035에 의해 설명된 "런 모드"와 유사한 모드를 사용하여 코딩될 수도 있다. 일부 예들에서, 화소 값이 팔레트에서의 정확히 하나의 인덱스 값과 연관될 수도 있다. 따라서, 일부 경우들에서는, 값들의 런이 유사 값으로 된 화소 값들의 스트링이라고 또한 지칭될 수도 있다. 다른 예들에서, 아래의 손실 코딩에 관해 설명되는 바와 같이, 하나를 초과하는 화소 값이 팔레트에서의 동일한 인덱스 값에 매핑될 수도 있다. 그런 예들에서, 값들의 런이 유사 값으로 된 인덱스 값들을 지칭한다.
- [0087] 예시의 목적을 위한 일 예에서, 주어진 스캔 순서에서의 두 개의 연속적인 인덱스들이 상이한 값들을 갖는다면, 런은 0과 동일하다. 주어진 스캔 순서에서의 두 개의 연속적인 인덱스들이 동일한 값을 갖지만 스캔 순서에서의 제 3 인덱스가 상이한 값을 갖는다면, 런은 1과 동일하다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 런을 나타내는 인덱스 엘리먼트들을 획득하고 그 데이터를 사용하여 동일한 인덱스 값을 갖는 연속적인 화소 로케이션들의 수를 결정할 수도 있다.
- [0088] 부가적으로 또는 대안으로, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 맵의 하나 이상의 엔트리들에 대한 라인 복사를 수행할 수도 있다. 그런 경우들에서, 인덱스들은 JCTVC-P0035에 의해 설명된 "상측 복사 모드"와 유사한 모드를 사용하여 코딩될 수도 있다. 엔트리들은 맵의 엔트리들과 블록의 화소 포지션들 간의 관계로 인해 "포지션들"이라고 또한 지칭될 수도 있다. 라인 복사는, 일부 예들에서, 스캔 방향에 의존할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 블록에서의 특정 포지션에 대한 화소 값 또는 인덱스 맵 값이 (수평 스캔의 경우) 특정 포지션 상측의 라인에서의 화소 또는 인덱스 값과 또는 (수직 스캔의 경우) 특정 포지션 좌측의 열과 동일하다는 것을 나타낼 수도 있다. 비디오 인코더 (20) 는, 런으로서, 특정 포지션 좌측의 열 또는 상측의 대응하는 화소 값들 또는 인덱스들과 동일한 스캔 순서에서의 화소 값들 또는 인덱스들의 수를 또한 나타낼 수도 있다. 이 예에서, 비디오 인코더 (20) 및 또는 비디오 디코더 (30) 는 현재 코딩되고 있는 블록의 라인에 대해 특정 이웃 라인으로부터 그리고 특정 수의 엔트리들로부터 화소 또는 인덱스를 복사할 수도 있다.
- [0089] 일부 경우들에서, 값들이 복사되는 라인은 현재 코딩되고 있는 포지션의 라인에 바로 인접할, 예컨대 그 라인의 좌측 또는 상측에 있을 수도 있다. 다른 예들에서, 블록의 다수의 라인들이 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 에 의해 버퍼링될 수도 있어서, 맵의 다수의 라인들 중 임의의 라인이 현재 코딩되고 있는 맵의 라인에 대한 예측 값들로서 사용될 수도 있다. 예시의 목적을 위한 일 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재 행의 화소들을 코딩하기 전에 이전의 네 개의 행들의 인덱스들 또는 화소 값들을 저장하도록 구성될 수도 있다. 이 예에서, 예측 행 (인덱스들 또는 화소 값들이 복사되는 행) 은 절단된 단항 코드 또는 단항 코드들과 같은 다른 코드들을 이용하여 비트스트림에서 나타내어질 수도 있다. 절

단된 단항 코드에 관해, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 최대 행 계산 (예컨대, row_index-1) 또는 최대 열 계산 (예컨대, column_index-1) 에 기초하여 절단된 단항 코드에 대한 최대 값을 결정할 수도 있다. 덧붙여서, 복사되는 예측 행으로부터의 포지션들의 수의 표시가 비트스트림에 또한 포함될 수도 있다. 일부 경우들에서, 현재 포지션이 예측되고 있는 라인 또는 열이 다른 블록 (예컨대, CU 또는 CTU) 에 속한다면 이러한 예측은 가능하지 않게 될 수도 있거나 또는 디폴트 인덱스 값 (예컨대, 0) 이 사용될 수도 있다.

[0090] 일부 예들에서, 이른바 엔트리들의 런들을 코딩하는 기법들은 위에서 설명된 라인 복사를 위한 기법들과 연계하여 사용될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 맵에서의 엔트리의 값이 팔레트로부터 획득되는지 또는 맵에서의 엔트리의 값이 그 맵에서의 이전에 코딩된 라인으로부터 획득되는지 (예컨대, 복사 모드 또는 화소 모드) 를 나타내는 하나 이상의 신택스 엘리먼트들 (예컨대, 플래그) 을 인코딩할 수도 있다. 비디오 인코더 (20) 는 팔레트의 인덱스 값 또는 라인 (행 또는 열) 에서의 엔트리의 로케이션을 나타내는 하나 이상의 신택스 엘리먼트들을 또한 인코딩할 수도 있다. 비디오 인코더 (20) 는 동일한 값을 공유하는 연속적인 엔트리들의 수를 나타내는 하나 이상의 신택스 엘리먼트들을 또한 인코딩할 수도 있다 (예컨대, 런 모드). 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 이러한 정보를 획득하고 그 정보를 사용하여 블록에 대한 맵 및 화소 값들을 복원할 수도 있다.

[0091] 위에서 지적했듯이, 맵의 인덱스들은 특정 순서로 스캔되는데, 그 특정 순서는 스캐닝 순서 또는 스캐닝 방향이라고 지칭될 수도 있다. 그러므로, 특정 스캐닝 순서로 인덱스들을 스캔하는 것은 인덱스들이 비트스트림에 포함되는 (또는 그 비트스트림으로부터 파싱되는) 방식을 일반적으로 지칭한다. 일부 예들에서, 스캔 순서는 수직, 수평, 또는 대각으로 (예컨대, 블록에서 45 도 또는 135 도 대각으로) 될 수도 있다. 예를 들면, 아래의 도 6a의 예에 관해 더 상세히 예시되고 설명되는 바와 같이, 맵의 인덱스들은 래스터 스캔 순서라고 지칭될 수도 있는 것에서 상측에서 하측으로, 좌측에서 우측으로 스캔될 수도 있다. 래스터 스캔 순서에서, 현재 라인의 첫 번째 인덱스는 이전 라인의 마지막 인덱스 뒤에 바로 스캔될 수도 있다.

[0092] 그러나, 일부 예들에서, 래스터 스캔 순서로 인덱스들을 스캔하는 것은 효율적이지 않을 수도 있다. 예를 들면, 비디오 데이터 블록의 제 1 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 비디오 데이터 블록의 제 2 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 제 1 라인의 마지막 화소가 제 2 라인의 마지막 화소에 인접하고, 제 1 에지 및 제 2 에지가 평행한 경우, 제 1 라인에서의 마지막 화소는 제 2 라인에서의 마지막 화소와 동일하지만, 제 2 라인에서의 제 1 화소와는 상이하다. 이 상황 (즉, 제 1 라인에서의 마지막 화소가 제 2 라인에서의 마지막 화소와 동일 또는 유사하지만 제 2 라인에서의 제 1 화소와는 상이한 경우) 은 컴퓨터 생성된 스크린 콘텐츠에서 비디오 콘텐츠의 다른 유형들보다 더 빈번하게 발생할 수도 있다. 다르게 말하면, 스크린 콘텐츠에 대해, 블록의 동일한 에지에 위치한 화소들이 블록의 대향 에지들에 위치한 화소들과는 동일 또는 유사한 값들을 가질 더 높은 확률이 있을 수도 있다. 래스터 스캐닝이 이러한 상관을 이용하지 못할 수도 있다.

[0093] 본 개시물의 하나 이상의 기법들에 따라, 비디오 인코더 (20) 가 맵을 인코딩하는 경우 뱀형 스캔 순서를 이용할 수도 있다. 예를 들면, 위의 예를 계속하여 그리고 도 6b의 예에 관해 아래에서 더욱 상세히 설명되는 바와 같이, 비디오 인코더 (20) 는 제 2 라인의 마지막 화소가 스캔 순서에서 제 1 라인의 마지막 화소 직후에 스캔되도록 맵을 인코딩할 수도 있다. 이런 식으로, 비디오 인코더 (20) 는 런 길이 코딩의 효율을 개선할 수도 있다.

[0094] 비디오 디코더 (30) 는 맵을 디코딩하는 경우 뱀형 스캔 순서를 또한 이용할 수도 있다. 예를 들면, 비디오 디코더 (30) 는 스캔 순서에서의 제 1 라인의 마지막 화소 직후에 제 2 라인의 마지막 화소가 스캔되도록 맵을 디코딩할 수도 있다. 이런 식으로, 비디오 디코더 (30) 는 런 길이 코딩의 효율을 개선할 수도 있다.

[0095] 일부 예들에서, 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 수직 스캐닝, 피아노 형, 및 지그재그를 비제한적으로 포함할 수도 있는 하나 이상의 다른 스캐닝 순서들을 사용할 수도 있다. 피아노 형 스캔 순서가, 각각의 건반이 상단에서는 (검은 건반들 간에) 두 개 화소 폭이고 하단에서는 적어도 두 개 화소 폭인 피아노의 흰 건반들의 에지들을 뒤따름으로써 표현될 수도 있다.

[0096] 일부 예들에서, 비디오 인코더 (20) 는 블록의 인덱스들을 스캔하기 위한 스캔 방향을 나타내는, 각각의 블록에 대한 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 부가적으로 또는 대안으로, 스캔 방향은 예를 들어, 블록 사이즈, 컬러 공간, 및/또는 컬러 성분과 같은 이른바 사이드 정보에 기초하여 시그널링 또는 유추

될 수도 있다. 비디오 인코더 (20) 는 블록의 각각의 컬러 성분에 대한 스캔들을 특정할 수도 있다. 대안으로, 특정 스캔이 블록의 모든 컬러 성분들에 적용될 수도 있다.

[0097] 팔레트 기반 코딩 모드가 사용되는 경우, 팔레트가 비디오 인코더 (20) 에 의해 비디오 디코더 (30) 에 의한 사용을 위한 인코딩된 비디오 데이터 비트스트림에서, 예컨대 본원에서 설명되는 기법들 중 하나 이상을 사용하여 송신된다. 팔레트가 각각의 블록에 대해 송신될 수도 있거나 또는 다수의 블록들 사이에서 공유될 수도 있다. 팔레트는 블록에 대해 지배적인 및/또는 그 블록을 대표하는 화소 값들의 수를 지칭할 수도 있다.

[0098] 일부 예들에서, 예컨대 팔레트에 포함되는 화소 값들의 수의 측면에서, 팔레트의 사이즈는, 인코딩된 비트스트림에서의 하나 이상의 선택스 엘리먼트들을 사용하여 정해질 수도 있거나 또는 시그널링될 수도 있다. 아래에서 더욱 상세히 설명되는 바와 같이, 화소 값이, 예컨대 코딩을 위해 사용되는 컬러 공간에 의존하여 다수의 샘플들로 이루어질 수도 있다. 예를 들어, 화소 값이 루마 및 색차 샘플들 (예컨대, 루마, U 색차 및 V 색차 (YUV) 또는 루마, Cb 색차, 및 Cr 색차 (YCbCr) 샘플들) 을 포함할 수도 있다. 다른 예에서, 화소 값이 적색, 녹색, 및 청색 (RGB) 샘플들을 포함할 수도 있다. 본원에서 설명된 바와 같이, 화소 값이란 용어는 화소에 기여하는 하나 이상의 샘플들을 일반적으로 참조할 수도 있다. 다시 말하면, 화소 값이란 용어는 화소에 기여하는 모든 샘플들을 반드시 참조하는 것은 아니고, 화소에 기여하는 단일 샘플 값을 설명하는데 사용될 수도 있다.

[0099] 일부 예들에서, 비디오 데이터의 팔레트 기반 코딩을 위한 기법들은 하나 이상의 다른 코딩 기법들, 이를테면 인터 예측 또는 인트라 예측 코딩을 위한 기법들과 함께 사용될 수도 있다. 예를 들어, 아래에서 더욱 상세히 설명되는 바와 같이, 인코더 또는 디코더, 또는 결합된 인코더-디코더 (코덱) 가, 인터 예측 및 인트라 예측 코딩, 뿐만 아니라 팔레트 기반 코딩을 수행하도록 구성될 수도 있다.

[0100] 도 2는 본 개시물의 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20) 를 도시하는 블록도이다. 도 2는 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들의 제한으로서 고려되지 않아야 한다. 설명의 목적으로, 본 개시물은 HEVC 코딩의 맥락에서 비디오 인코더 (20) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용 가능할 수도 있다.

[0101] 비디오 인코더 (20) 는 본 개시물에서 설명되는 다양한 예들에 따라 팔레트 기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스의 일 예를 나타낸다. 예를 들어, 비디오 인코더 (20) 는 팔레트 기반 코딩 또는 비-팔레트 기반 코딩 중 어느 하나를 사용하여 비디오 데이터의 다양한 블록들, 이를테면 HEVC 코딩에서의 CU들 또는 PU들을, 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트 기반 코딩 모드들은 다양한 인터 예측 시간적 코딩 모드들 또는 인터 예측 공간적 코딩 모드들, 이를테면 HEVC 초안 10에 의해 특정된 다양한 코딩 모드들을 지칭할 수도 있다. 비디오 인코더 (20) 는, 하나의 예에서, 화소 값들을 나타내는 엔트리들을 갖는 팔레트를 생성하도록, 비디오 데이터 블록의 적어도 일부 포지션들의 화소 값들을 표현하기 위해 팔레트에서의 화소 값들을 선택하도록, 그리고 비디오 데이터 블록의 포지션들 중 적어도 일부의 포지션들과 선택된 화소 값들에 각각 대응하는 팔레트에서의 엔트리들을 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 시그널링된 정보는 비디오 데이터를 디코딩하기 위해 비디오 디코더 (30) 에 의해 사용될 수도 있다.

[0102] 도 2의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 부 (100), 잔차 생성 부 (102), 변환 프로세싱 부 (104), 양자화 부 (106), 역 양자화 부 (108), 역 변환 프로세싱 부 (110), 복원 부 (112), 필터 부 (114), 디코딩된 픽처 버퍼 (116), 및 엔트로피 인코딩 부 (118) 를 포함한다. 예측 프로세싱 부 (100) 는 인터 예측 프로세싱 부 (120) 와 인트라 예측 프로세싱 부 (126) 를 구비한다. 인터 예측 프로세싱 부 (120) 는 모션 추정 부와 모션 보상 부 (미도시) 를 구비한다. 비디오 인코더 (20) 는 본 개시물에서 설명되는 팔레트 기반 코딩 기법들의 다양한 양태들을 수행하도록 구성되는 팔레트 기반 인코딩 부 (122) 를 구비한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0103] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터의 픽처의 슬라이스에서의 각각의 CTU를 인코딩할 수도 있다. CTU들의 각각은 픽처의 동일 사이즈로 된 루마 코딩 트리 블록 (CTB) 들 및 대응하는 CTB들에 연관될 수도 있다. CTU를 인코딩하는 부분으로서, 예측 프로세싱 부 (100) 는 쿼드트리 구획화를 수행하여 CTU의 CTB들을 점차적으로 더 작은 블록들로 분할할 수도 있다. 더 작은 블록은 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 부 (100) 는 CTU에 연관된 CTB를 네 개의 동일 사이즈로 된 서브-블록들로 구획화하며, 그 서브-블록들 중 하나 이상을 네 개의 동일 사이즈로 된 서브 서브-블록들로 구획화하는 등등을 수행할 수도 있다.

- [0104] 비디오 인코더 (20) 는 CTU의 CU들을 인코딩하여 CU들의 인코딩된 표현들 (즉, 코딩된 CU들) 을 생성할 수도 있다. CU를 인코딩하는 부분으로서, 예측 프로세싱 부 (100) 는 CU의 하나 이상의 PU들 중에서 CU에 연관된 코딩 블록들을 구획화할 수도 있다. 따라서, 각각의 PU는 루마 예측 블록 및 대응하는 크로마 예측 블록들에 연관될 수도 있다. 비디오 인코더 (20) 와 비디오 디코더 (30) 는 다양한 사이즈들을 갖는 PU들을 지원할 수도 있다. 위에서 나타난 바와 같이, CU의 사이즈는 CU의 루마 코딩 블록의 사이즈를 지칭할 수도 있고 PU의 사이즈는 PU의 루마 예측 블록의 사이즈를 지칭할 수도 있다. 특정 CU의 사이즈가 $2N \times 2N$ 이라고 가정하면, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 인트라 예측을 위한 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들과, 인터 예측을 위한 $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, 또는 유사한 것의 대칭적 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 와 비디오 디코더 (30) 는 인터 예측을 위해 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에 대한 비대칭 구획화를 또한 지원할 수도 있다.
- [0105] 인터 예측 프로세싱 부 (120) 는 CU의 각각의 PU에 대해 인터 예측을 수행함으로써 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU의 예측 샘플 블록들 및 그 PU에 대한 모션 정보를 포함할 수도 있다. 인터 예측 프로세싱 부 (120) 는 PU가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 중 어느 것에 있는지에 의존하여 CU의 PU에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들이 인트라 예측된다. 그러므로, PU가 I 슬라이스에 있으면, 인터 예측 프로세싱 부 (120) 는 PU에 대해 인터 예측을 수행하지 않는다. 따라서, I-모드에서 인코딩된 블록들에 대해, 예측된 블록은 동일한 프레임 내의 이전에 인코딩된 이웃 블록들로부터 공간적 예측을 사용하여 형성된다.
- [0106] PU가 P 슬라이스에 있다면, 인터 예측 프로세싱 부 (120) 의 모션 추정 부는 PU에 대한 참조 지역을 참조 픽처들의 리스트 (예컨대, "RefPicList0") 에서의 참조 픽처들에서 검색할 수도 있다. PU에 대한 참조 지역은, 참조 픽처 내의, PU의 샘플 블록들에 가장 밀접하게 대응하는 샘플 블록들을 포함하는 지역일 수도 있다. 모션 추정 부는 PU에 대한 참조 지역을 포함하는 참조 픽처의 RefPicList0에서의 포지션을 나타내는 참조 인덱스를 생성할 수도 있다. 덧붙여서, 모션 추정 부는 PU의 코딩 블록 및 참조 지역에 연관된 참조 로케이션 사이의 공간적 변위를 나타내는 MV를 생성할 수도 있다. 예를 들면, MV는 현재 코딩된 픽처에서의 좌표들로부터 참조 픽처에서의 좌표들로의 오프셋을 제공하는 2차원 벡터일 수도 있다. 모션 추정 부는 참조 인덱스 및 MV를 PU의 모션 정보로서 출력할 수도 있다. 인터 예측 프로세싱 부 (120) 의 모션 보상 부는 PU의 모션 벡터에 의해 가리켜진 참조 로케이션에 있는 실제 또는 보간된 샘플들에 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.
- [0107] PU가 B 슬라이스에 있다면, 모션 추정 부는 PU에 대해 단-예측 또는 양-예측을 수행할 수도 있다. PU에 대한 단-예측을 수행하기 위해, 모션 추정 부는 PU에 대한 참조 지역을 RefPicList0 또는 제 2 참조 픽처 리스트 ("RefPicList1") 의 참조 픽처들에서 검색할 수도 있다. 모션 추정 부 (122) 는, PU의 모션 정보로서, 참조 지역을 포함하는 참조 픽처의 RefPicList0 또는 RefPicList1에서의 포지션을 나타내는 참조 인덱스, PU의 샘플 블록 및 참조 지역에 연관된 참조 로케이션 사이의 공간적 변위를 나타내는 MV, 및 참조 픽처가 RefPicList0에 있는지 또는 RefPicList1에 있는지를 나타내는 하나 이상의 예측 방향 표시자들을 출력할 수도 있다. 인터 예측 프로세싱 부 (120) 의 모션 보상 부는 PU의 모션 벡터에 의해 가리켜진 참조 지역에 있는 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.
- [0108] PU에 대한 양방향 인터 예측을 수행하기 위해, 모션 추정 부는 그 PU에 대한 참조 지역을 RefPicList0에서의 참조 픽처들에서 검색할 수도 있고, 또한 그 PU에 대한 다른 참조 지역을 RefPicList1에서의 참조 픽처들에서 검색할 수도 있다. 모션 추정 부는 참조 지역들을 포함하는 참조 픽처들의 RefPicList0 및 RefPicList1에서의 포지션들을 나타내는 참조 픽처 인덱스들을 생성할 수도 있다. 덧붙여서, 모션 추정 부는 참조 지역들에 연관된 참조 로케이션 및 PU의 샘플 블록 사이의 공간적 변위들을 나타내는 MV들을 생성할 수도 있다. PU의 모션 정보는 PU의 MV들 및 참조 인덱스들을 포함할 수도 있다. 모션 보상 부는 PU의 모션 벡터에 의해 나타내어진 참조 지역에 있는 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 샘플 블록들을 생성할 수도 있다.
- [0109] 본 개시물의 다양한 예들에 따라, 비디오 인코더 (20) 는 팔레트 기반 코딩을 수행하도록 구성될 수도 있다. HEVC 프레임워크에 관해, 일 예로서, 팔레트 기반 코딩 기법들이 코딩 유닛 (CU) 모드로서 사용되도록 구성될 수도 있다. 다른 예들에서, 팔레트 기반 코딩 기법들은 HEVC의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 따라서, CU 모드의 맥락에서 (본 개시물을 통해) 본원에서 설명되는 개시된 프로세스들의 모두는 PU에 부가적으로 또는 대안으로 적용될 수도 있다. 그러나, 이들 HEVC 기반 예들은 본원에서 설명되는 팔레트 기반 코딩 기법들의 제약 또는 제한으로서 간주되지 않아야 하는데, 이러한 기법들이 독립적으로 또

는 다른 기존의 또는 아직 개발되지 않은 시스템들/표준들의 부분으로서 작동하기 위해 적용될 수도 있어서이다. 이들 경우들에서, 팔레트 코딩을 위한 유닛은 정사각형 블록들, 직사각형 블록들 또는 심지어 비-직사각형 형상의 지역들일 수 있다.

[0110] 팔레트 기반 인코딩 부 (122) 는, 팔레트 기반 인코딩 모드가, 예컨대, CU 또는 PU에 대해 선택되는 경우, 예를 들어 팔레트 기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트 기반 인코딩 부 (122) 는, 화소 값들을 나타내는 엔트리들을 갖는 팔레트를 생성하도록, 비디오 데이터 블록의 적어도 일부 포지션들의 화소 값들을 표현하기 위해 팔레트에서의 화소 값들을 선택하도록, 그리고 비디오 데이터 블록의 포지션들 중 적어도 일부의 포지션들과 선택된 화소 값들에 각각 대응하는 팔레트에서의 엔트리들을 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 비록 다양한 기능들이 팔레트 기반 인코딩 부 (122) 에 의해 수행되는 것으로서 설명되지만, 이러한 기능들의 일부 또는 전부는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0111] 일부 예들에서, 팔레트 기반 인코딩 부 (122) 는 블록의 적어도 일부 포지션들과 팔레트에서의 엔트리들을 특정 순서로 연관시키는 정보에 의해 나타내어진 팔레트 인덱스 값들을 인코딩할 수도 있다. 예를 들면, 팔레트 인덱스 값들을 인코딩하기 위해 팔레트 기반 인코딩 부 (122) 에 의해 사용되는 스캔 순서는 팔레트 인덱스 값들을 디코딩하기 위해 팔레트 기반 디코딩 부, 이를테면 비디오 디코더 (30) 의 팔레트 기반 디코딩 부 (165) 에 의해 사용되는 동일한 스캔 순서일 수도 있다. 일부 예들에서, 팔레트 기반 인코딩 부 (122) 는 팔레트 인덱스 값들을 코딩하기 위해 사용되는 스캔 순서를 특정하기 위해 하나 이상의 신덱스 엘리먼트들을 인코딩할 수도 있다.

[0112] 위에서 논의된 바와 같이, 맵의 인덱스들은 래스터 스캔 순서라고 지칭될 수도 있는 것에서 상단에서 하단으로, 좌측에서우측으로 스캔될 수도 있다. 래스터 스캔 순서에서, 현재 라인의 첫 번째 인덱스는 이전 라인의 마지막 인덱스 뒤에 바로 스캔될 수도 있다.

[0113] 그러나, 일부 예들에서, 래스터 스캔 순서로 인덱스들을 스캔하는 것은 효율적이지 않을 수도 있다. 예를 들면, 비디오 데이터 블록의 제 1 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 비디오 데이터 블록의 제 2 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 제 1 라인의 마지막 화소가 제 2 라인의 마지막 화소에 인접하고, 제 1 에지 및 제 2 에지가 평행한 경우, 제 1 라인에서의 마지막 화소는 제 2 라인에서의 마지막 화소와 동일하지만, 제 2 라인에서의 제 1 화소와는 상이하다. 이 상황 (즉, 제 1 라인에서의 마지막 화소가 제 2 라인에서의 마지막 화소와 동일 또는 유사하지만 제 2 라인에서의 제 1 화소와는 상이한 경우) 은 컴퓨터 생성된 스크린 콘텐츠에서 비디오 콘텐츠의 다른 유형들보다 더 빈번하게 발생할 수도 있다. 다르게 말하면, 스크린 콘텐츠에 대해, 블록의 동일한 에지에 위치한 화소들이 블록의 대향 에지들에 위치한 화소들과는 동일 또는 유사한 값들을 가질 더 높은 확률이 있을 수도 있다. 래스터 스캐닝이 이러한 상관을 이용하지 못할 수도 있다.

[0114] 본 개시물의 하나 이상의 기법들에 따라, 팔레트 기반 인코딩 부 (122) 는 맵을 인코딩하는 경우 뱀형 스캔 순서를 이용할 수도 있다. 예를 들면, 팔레트 기반 인코딩 부 (122) 는 스캔 순서에서 제 1 라인의 마지막 화소 직후에 제 2 라인의 마지막 화소가 스캔되도록 맵을 인코딩할 수도 있다. 이런 식으로, 팔레트 기반 인코딩 부 (122) 는 런 길이 코딩의 효율을 개선할 수도 있다.

[0115] 팔레트 기반 인코딩 부 (122) 는 본원에서 설명되는 다양한 신덱스 엘리먼트들 중 임의의 것을 생성하도록 구성될 수도 있다. 따라서, 비디오 인코더 (20) 는 본 개시물에서 설명되는 바와 같은 팔레트 기반 코드 모드들을 사용하여 비디오 데이터의 블록들을 인코딩하도록 구성될 수도 있다. 비디오 인코더 (20) 는 선택적으로, 팔레트 코딩 모드를 사용하여 비디오 데이터 블록을 인코딩하거나, 또는 상이한 모드, 예컨대, 이러한 HEVC 인터 예측 또는 인트라 예측 코딩 모드를 사용하여 비디오 데이터 블록을 인코딩할 수도 있다. 비디오 데이터 블록은, 예를 들어, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU일 수도 있다. 비디오 인코더 (20) 는 일부 블록들을 인터 예측 시간적 예측 또는 인터 예측 공간적 코딩 모드들로 인코딩하고 다른 블록들을 팔레트 기반 코딩 모드로 디코딩할 수도 있다.

[0116] 인트라 예측 프로세싱 부 (126) 는 PU에 대해 인트라 예측을 수행함으로써 그 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU에 대한 예측 샘플 블록들과 다양한 신덱스 엘리먼트들을 포함할 수도 있다. 인트라 예측 프로세싱 부 (126) 는 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU들에 대해 인트라 예측을 수행할 수도 있다.

- [0117] PU에 대해 인트라 예측을 수행하기 위해, 인트라 예측 프로세싱 부 (126) 는 PU에 대한 예측 데이터의 다수의 세트들을 생성하기 위해 다수의 인트라 예측 모드들을 사용할 수도 있다. 인트라 예측 모드를 사용하여 PU에 대한 예측 데이터의 세트를 생성하기 위해, 인트라 예측 프로세싱 부 (126) 는 이웃하는 PU들의 샘플 블록들로부터 인트라 예측 모드에 연관된 방향에서 PU의 샘플 블록들을 가로질러 샘플들을 확장할 수도 있다. PU들, CU들, 및 CTU들에 대한 좌측에서 우측으로, 상단에서 하단으로의 인코딩 순서를 가정하면, 이웃하는 PU들은 PU의 상측, 우상측, 좌상측, 또는 좌측에 있을 수도 있다. 인트라 예측 프로세싱 부 (126) 는 다양한 수들의 인트라 예측 모드들, 예컨대, 33 개의 방향성 인트라 예측 모드들을 사용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 수는 PU에 연관된 지역의 사이즈에 의존할 수도 있다.
- [0118] 예측 프로세싱 부 (100) 는 CU의 PU들에 대한 예측 데이터를, 그 PU들에 대해 인트라 예측 프로세싱 부 (120) 에 의해 생성된 예측 데이터 또는 그 PU들에 대해 인트라 예측 프로세싱 부 (126) 에 의해 생성된 예측 데이터 중에서 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 부 (100) 는 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 CU의 PU들에 대한 예측 데이터를 선택한다. 선택된 예측 데이터의 예측 샘플 블록들은 본원에서 선택된 예측 샘플 블록들이라고 지칭될 수도 있다.
- [0119] 잔차 생성 부 (102) 는, CU의 루마, Cb 및 Cr 코딩 블록과 CU의 PU들의 선택된 예측 루마, Cb 및 Cr 블록들에 기초하여, CU의 루마, Cb 및 Cr 잔차 블록들을 생성할 수도 있다. 예를 들면, 잔차 생성 부 (102) 는 CU의 잔차 블록들에서의 각각의 샘플이 CU의 코딩 블록에서의 샘플 및 CU의 PU의 대응하는 선택된 예측 샘플 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 가지도록 CU의 잔차 블록들을 생성할 수도 있다.
- [0120] 변환 프로세싱 부 (104) 는 쿼드트리 구획화를 수행하여 CU에 연관된 잔차 블록들을 CU의 TU들에 연관된 변환 블록들로 구획화할 수도 있다. 따라서, TU가 루마 변환 블록 및 두 개의 크로마 변환 블록들에 연관될 수도 있다. CU의 TU들의 루마 및 크로마 변환 블록들의 사이즈들 및 포지션들은 그 CU의 PU들의 예측 블록들의 사이즈들 및 포지션들에 기초할 수도 있거나 또는 기초하지 않을 수도 있다. "잔차 쿼드 트리 (residual quad-tree)" (RQT) 로서 알려진 쿼드트리 구조가 지역들의 각각에 연관된 노드들을 포함할 수도 있다. CU의 TU들은 RQT의 리프 노드들에 대응할 수도 있다.
- [0121] 변환 프로세싱 부 (104) 는 CU의 각각의 TU에 대한 변환 계수 블록들을, 하나 이상의 변환들을 그 TU의 변환 블록들에 적용함으로써 생성할 수도 있다. 변환 프로세싱 부 (104) 는 다양한 변환들을 TU에 연관된 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 부 (104) 는 이산 코사인 변환 (discrete cosine transform, DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 변환 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 부 (104) 는 변환들을 변환 블록에 적용하지 않는다. 그런 예들에서, 변환 블록은 변환 계수 블록으로서 다루어질 수도 있다.
- [0122] 양자화 부 (106) 는 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 변환 계수들의 일부 또는 전부에 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n -비트 변환 계수가 양자화 동안에 m -비트 변환 계수로 버림될 (rounded down) 수도 있으며, 여기서 n 은 m 보다 크다. 양자화 부 (106) 는 CU에 연관된 양자화 파라미터 (QP) 값에 기초하여 그 CU의 TU에 연관된 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 CU에 연관된 QP 값을 조정함으로써 그 CU에 연관된 계수 블록들에 적용되는 양자화 정도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있고, 따라서 양자화된 변환 계수들은 원래의 것들보다 더 낮은 정밀도를 가질 수도 있다.
- [0123] 역 양자화 부 (108) 와 역 변환 프로세싱 부 (110) 는 계수 블록으로부터 잔차 블록을 복원하기 위해 역 양자화 및 역 변환들을 계수 블록에 각각 적용할 수도 있다. 복원 부 (112) 는 TU에 연관된 복원된 변환 블록을 생성하기 위해 복원된 잔차 블록을 예측 프로세싱 부 (100) 에 의해 생성된 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 가산할 수도 있다. CU의 각각의 TU에 대한 변환 블록들을 이런 식으로 복원함으로써, 비디오 인코더 (20) 는 CU의 코딩 블록들을 복원할 수도 있다.
- [0124] 필터 부 (114) 는 하나 이상의 블록화제거 (deblocking) 동작들을 수행하여 CU에 연관된 코딩 블록들에서의 블록화 아티팩트들을 감소시킬 수도 있다. 디코딩된 픽처 버퍼 (116) 는, 필터 부 (114) 가 복원된 코딩 블록들에 대해 하나 이상의 블록화제거 동작들을 수행한 후에 복원된 코딩 블록들을 저장할 수도 있다. 인트라 예측 프로세싱 부 (120) 는 다른 픽처들의 PU들에 대해 인트라 예측을 수행하기 위해 복원된 코딩 블록들을 포함하는 참조 픽처를 사용할 수도 있다. 덧붙여서, 인트라 예측 프로세싱 부 (126) 는 CU와 동일한 픽처에서의 다른 PU들에 대해 인트라 예측을 수행하기 위해 디코딩된 픽처 버퍼 (116) 에서의 복원된 코딩 블록들을 사용할

수도 있다.

- [0125] 엔트로피 인코딩 부 (118) 는 비디오 인코더 (20) 의 다른 기능성 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 부 (118) 는 양자화 부 (106) 로부터 계수 블록들을 수신할 수도 있고 예측 프로세싱 부 (100) 로부터 신택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 부 (118) 는 그 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여 엔트로피 인코딩된 데이터를 생성할 수도 있다. 예를 들어, 엔트로피 인코딩 부 (118) 는 그 데이터에 대해 컨텍스트 적응 가변 길이 코딩 (context-adaptive variable length coding, CAVLC) 동작, CABAC 동작, 가변 대 가변 (variable-to-variable, V2V) 길이 코딩 동작, 신택스 기반 컨텍스트 적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding, SBAC) 동작, 확률 간격 구획화 엔트로피 (Probability Interval Partitioning Entropy, PIPE) 코딩 동작, 지수-골롬 (Exponential-Golomb) 인코딩 동작, 또는 다른 유형의 엔트로피 인코딩 동작을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩 부 (118) 에 의해 생성된 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들면, 그 비트스트림은 CU에 대한 RQT를 표현하는 데이터를 포함할 수도 있다.
- [0126] 일부 예들에서, 잔차 코딩이 팔레트 코딩과 함께 수행되지 않는다. 따라서, 비디오 인코더 (20) 는 팔레트 코딩 모드를 사용하여 코딩하는 경우 변환 또는 양자화를 수행하지 않을 수도 있다. 덧붙여서, 비디오 인코더 (20) 는 잔차 데이터와는 별도로 팔레트 코딩 모드를 사용하여 생성된 데이터를 엔트로피 인코딩할 수도 있다.
- [0127] 도 3은 본 개시물의 기법들을 구현하도록 구성되는 일 예의 비디오 디코더 (30) 를 도시하는 블록도이다. 도 3은 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들로 제한하고 있지는 않다. 설명의 목적으로, 본 개시물은 HEVC 코딩의 맥락에서 비디오 디코더 (30) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용 가능할 수도 있다.
- [0128] 비디오 디코더 (30) 는 본 개시물에서 설명되는 다양한 예들에 따라 팔레트 기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스의 일 예를 나타낸다. 예를 들어, 비디오 디코더 (30) 는 팔레트 기반 코딩 또는 비-팔레트 기반 코딩 중 어느 하나를 사용하여 비디오 데이터의 다양한 블록들, 이를테면 HEVC 코딩에서의 CU들 또는 PU들을, 선택적으로 디코딩하도록 구성될 수도 있다. 비-팔레트 기반 코딩 모드들은 다양한 인터 예측 시간적 코딩 모드들 또는 인터 예측 공간적 코딩 모드들, 이를테면 HEVC 초안 10에 의해 특정된 다양한 코딩 모드들을 지칭할 수도 있다. 비디오 디코더 (30) 는, 하나의 예에서, 화소 값들을 나타내는 엔트리들을 갖는 팔레트를 생성하도록, 비디오 데이터 블록의 적어도 일부 포지션들과 팔레트에서의 엔트리들을 연관시키는 정보를 수신하도록, 그 정보에 기초하여 팔레트에서의 화소 값들을 선택하도록, 그리고 선택된 화소 값들에 기초하여 블록의 화소 값들을 복원하도록 구성될 수도 있다.
- [0129] 도 3의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 부 (150), 예측 프로세싱 부 (152), 역 양자화 부 (154), 역 변환 프로세싱 부 (156), 복원 부 (158), 필터 부 (160), 및 디코딩된 픽처 버퍼 (162) 를 구비한다. 예측 프로세싱 부 (152) 은 모션 보상 부 (164) 와 인트라 예측 프로세싱 부 (166) 를 구비한다. 비디오 디코더 (30) 는 본 개시물에서 설명되는 팔레트 기반 코딩 기법들의 다양한 양태들을 수행하도록 구성되는 팔레트 기반 인코딩 부 (165) 를 구비한다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.
- [0130] 코딩된 픽처 버퍼 (coded picture buffer, CPB) 가 비트스트림의 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신 및 저장할 수도 있다. 엔트로피 디코딩 부 (150) 는 CPB로부터 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신하고 신택스 엘리먼트들을 디코딩하기 위해 그 NAL 유닛들을 파싱할 수도 있다. 엔트로피 디코딩 부 (150) 는 NAL 유닛들에서의 엔트로피 인코딩된 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 부 (152), 역 양자화 부 (154), 역 변환 프로세싱 부 (156), 복원 부 (158), 및 필터 부 (160) 는 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.
- [0131] 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 부분으로서, 엔트로피 디코딩 부 (150) 는 코딩된 슬라이스 NAL 유닛들로부터 신택스 엘리먼트들을 추출하고 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들의 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관계된 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 신택스 엘리먼트들은 슬라이스를 포함하는 픽처에 연관된 PPS를 식별하는 신택스 엘리먼트를 포함할 수도 있

다.

- [0132] 비트스트림으로부터 선택스 엘리먼트들을 디코딩하는 것 외에도, 비디오 디코더 (30) 는 비구획화된 CU에 대해 복원 동작을 수행할 수도 있다. 비구획화된 CU에 대해 복원 동작을 수행하기 위해, 비디오 디코더 (30) 는 CU의 각각의 TU에 대해 복원 동작을 수행할 수도 있다. CU의 각각의 TU에 대한 복원 동작을 수행함으로써, 비디오 디코더 (30) 는 CU의 잔차 블록들을 복원할 수도 있다.
- [0133] CU의 TU에 대해 복원 동작을 수행하는 부분으로서, 역 양자화 부 (154) 는 TU에 연관된 계수 블록들을 역 양자화, 즉, 탈양자화 (de-quantization) 할 수도 있다. 역 양자화 부 (154) 는 TU의 CU에 연관된 QP 값을 사용하여 양자화 정도와, 비슷하게, 역 양자화 부 (154) 에 대해 적용할 역 양자화 정도를 결정할 수도 있다. 다시 말하면, 압축 비율, 즉, 원래의 시퀀스 및 압축된 시퀀스를 표현하는데 사용된 비트들의 수의 비율은, 변환 계수들을 양자화하는 경우에 사용된 QP의 값을 조정함으로써 제어될 수도 있다. 압축 비율은 채용된 엔트로피 코딩하는 방법에 또한 의존할 수도 있다.
- [0134] 역 양자화 부 (154) 가 계수 블록을 역 양자화한 후, 역 변환 프로세싱 부 (156) 는 TU에 연관된 잔차 블록을 생성하기 위하여 하나 이상의 역 변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역 변환 프로세싱 부 (156) 는 역 DCT, 역 정수 변환, 역 카루넨-로베 변환 (Karhunen-Loeve transform, KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 계수 블록에 적용할 수도 있다.
- [0135] PU가 인트라 예측을 사용하여 인코딩되면, 인트라 예측 프로세싱 부 (166) 는 PU에 대한 예측 블록들을 생성하기 위해 인트라 예측을 수행할 수도 있다. 인트라 예측 프로세싱 부 (166) 는 인트라 예측 모드를 사용하여 공간적으로 이웃하는 PU들의 예측 블록들에 기초하여 PU에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다. 인트라 예측 프로세싱 부 (166) 는 비트스트림으로부터 디코딩된 하나 이상의 선택스 엘리먼트들에 기초하여 PU에 대한 인트라 예측 모드를 결정할 수도 있다.
- [0136] 예측 프로세싱 부 (152) 는 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 제 1 참조 픽처 리스트 (RefPicList0) 및 제 2 참조 픽처 리스트 (RefPicList1) 를 구축할 수도 있다. 더욱이, PU가 인터 예측을 사용하여 인코딩되면, 엔트로피 디코딩 부 (150) 는 그 PU에 대한 모션 정보를 추출할 수도 있다. 모션 보상 부 (164) 는, PU의 모션 정보에 기초하여, PU에 대한 하나 이상의 참조 지역들을 결정할 수도 있다. 모션 보상 부 (164) 는, PU에 대한 하나 이상의 참조 블록들에서의 샘플 블록들에 기초하여, PU에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다.
- [0137] 복원 부 (158) 는, 해당되는 경우, CU의 TU들에 연관된 루마, Cb 및 Cr 변환 블록들과 그 CU의 PU들의 예측 루마, Cb 및 Cr 블록들, 즉 인트라 예측 데이터 또는 인터 예측 데이터 중 어느 하나를 사용하여 그 CU의 루마, Cb 및 Cr 코딩 블록들을 복원할 수도 있다. 예를 들어, 복원 부 (158) 는 루마, Cb 및 Cr 변환 블록들의 샘플들을 예측 루마, Cb 및 Cr 블록들의 대응하는 샘플들에 가산하여 CU의 루마, Cb 및 Cr 코딩 블록들을 복원할 수도 있다.
- [0138] 필터 부 (160) 는 블록화제거 동작을 수행하여 CU의 루마, Cb 및 Cr 코딩 블록들에 연관된 블록화 아티팩트들을 감소시킬 수도 있다. 비디오 디코더 (30) 는 CU의 루마, Cb 및 Cr 코딩 블록들을 디코딩된 픽처 버퍼 (162) 에 저장할 수도 있다. 디코딩된 픽처 버퍼 (162) 는 후속하는 모션 보상, 인트라 예측, 및 디스플레이 디바이스, 이를테면 도 1의 디스플레이 디바이스 (32) 상의 프레젠테이션을 위해 참조 픽처들을 제공할 수도 있다. 예를 들면, 비디오 디코더 (30) 는, 디코딩된 픽처 버퍼 (162) 에서의 루마, Cb 및 Cr 블록들에 기초하여, 다른 CU들의 PU들에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다. 이런 식으로, 비디오 디코더 (30) 는, 비트스트림으로부터, 중대한 루마 계수 블록의 변환 계수 레벨들을 추출하며, 그 변환 계수 레벨들을 역 양자화하며, 그 변환 계수 레벨들에 변환을 적용하여 변환 블록을 생성하며, 그 변환 블록에 적어도 부분적으로 기초하여, 코딩 블록을 생성하고, 그 코딩 블록을 디스플레이를 위해 출력할 수도 있다.
- [0139] 본 개시물의 다양한 예들에 따라, 비디오 디코더 (30) 는 팔레트 기반 코딩을 수행하도록 구성될 수도 있다. 팔레트 기반 디코딩 부 (165) 는, 팔레트 기반 디코딩 모드가, 예컨대, CU 또는 PU에 대해 선택되는 경우, 예를 들어 팔레트 기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트 기반 디코딩 부 (165) 는, 화소 값들을 나타내는 엔트리들을 갖는 팔레트를 생성하도록, 비디오 데이터 블록의 적어도 일부 포지션들과 팔레트에서의 엔트리들을 연관시키는 정보를 수신하도록, 그 정보에 기초하여 팔레트에서의 화소 값들을 선택하도록, 그리고 선택된 화소 값들에 기초하여 블록의 화소 값들을 복원하도록 구성될 수도 있다. 비록 다양한 기능들이 팔레트 기반 디코딩 부 (165) 에 의해 수행되는 것으로서 설명되지만, 이러한 기능들의 일부 또는 전부는 다른

프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

- [0140] 팔레트 기반 디코딩 부 (165) 는 팔레트 코딩 모드 정보를 수신하고, 팔레트 코딩 모드가 블록에 적용됨을 팔레트 코딩 모드 정보가 나타내는 경우 위의 동작들을 수행할 수도 있다. 팔레트 코딩 모드가 블록에 적용되지 않음을 팔레트 코딩 모드 정보가 나타내는 경우, 또는 다른 모드 정보가 상이한 모드의 사용을 나타내는 경우, 팔레트 코딩 모드가 블록에 적용되지 않음을 팔레트 코딩 모드 정보가 나타낼 때, 팔레트 기반 디코딩 부 (165) 는 비-팔레트 기반 코딩 모드, 예컨대, 이러한 HEVC 인터 예측 또는 인트라 예측 코딩 모드를 사용하여 비디오 데이터 블록을 디코딩한다. 비디오 데이터 블록은, 예를 들어, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU일 수도 있다. 비디오 디코더 (30) 는 일부 블록들을 인터 예측 시간적 예측 또는 인터 예측 공간적 코딩 모드들로 디코딩하고 다른 블록들을 팔레트 기반 코딩 모드로 디코딩할 수도 있다. 팔레트 기반 코딩 모드는 복수의 상이한 팔레트 기반 코딩 모드들 중 하나를 포함할 수도 있거나, 또는 단일 팔레트 기반 코딩 모드가 있을 수도 있다.
- [0141] 팔레트 기반 디코딩 부 (165) 에 의해 수신된 팔레트 코딩 모드 정보는 팔레트 모드 선택스 엘리먼트, 이를테면 플래그를 포함할 수도 있다. 팔레트 모드 선택스 엘리먼트의 제 1 값이 팔레트 코딩 모드가 블록에 적용됨을 나타내고 팔레트 모드 선택스 엘리먼트의 제 2 값이 팔레트 코딩 모드가 비디오 데이터 블록에 적용되지 않음을 나타낸다. 팔레트 기반 디코딩 부 (165) 는 예측 유닛 레벨, 코딩 유닛 레벨, 슬라이스 레벨, 또는 픽처 레벨 중 하나 이상에서 (예컨대, 비디오 인코더 (20) 로부터) 팔레트 코딩 모드 정보를 수신할 수도 있거나, 또는 팔레트 코딩 모드가 픽처 파라미터 세트 (PPS), 시퀀스 파라미터 세트 (sequence parameter set, SPS) 또는 비디오 파라미터 세트 (video parameter set, VPS) 중 적어도 하나에서 가능하게 되는지의 여부의 표시를 수신할 수도 있다.
- [0142] 일부 예들에서, 팔레트 기반 디코딩 부 (165) 는 비디오 데이터 블록에 연관되는 코딩 블록의 사이즈, 프레임 유형, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 중 하나 이상에 기초하여 팔레트 코딩 모드 정보를 유추할 수도 있다.
- [0143] 팔레트 기반 디코딩 부 (165) 는 비디오 데이터와 함께 팔레트에서의 엔트리들 중 적어도 일부의 엔트리들을 정의하는 정보를 수신하고, 수신된 정보에 적어도 부분적으로 기초하여 팔레트를 생성하도록 또한 구성될 수도 있다. 팔레트의 사이즈는 고정되거나 또는 가변될 수도 있다. 일부 경우들에서, 팔레트의 사이즈는 가변적이고 비디오 데이터와 함께 시그널링된 정보에 기초하여 조정 가능하다. 시그널링된 정보는 팔레트에서의 엔트리가 팔레트에서의 마지막 엔트리인지의 여부를 특정할 수도 있다. 또한, 일부 경우들에서, 팔레트는 최대 사이즈를 가질 수도 있다. 팔레트의 사이즈 또는 팔레트의 최대 사이즈는 또한 조건부로 송신 또는 유추될 수도 있다. 조건들은 CU의 사이즈, 프레임 유형, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id일 수도 있다.
- [0144] 팔레트는 블록의 루마 성분 및 크로마 성분들에 대한 화소 값들을 나타내는 엔트리들을 포함하는 단일 팔레트일 수도 있다. 이 경우, 팔레트에서의 각각의 엔트리는 루마 성분 및 두 개의 크로마 성분들에 대한 화소 값들을 나타내는 3중 (triple) 엔트리이다. 대안으로, 팔레트는 블록의 루마 성분의 화소 값들을 나타내는 엔트리들을 포함하는 루마 팔레트와, 블록의 각각의 크로마 성분들에 대한 화소 값들을 나타내는 엔트리들을 포함하는 크로마 팔레트들을 포함한다.
- [0145] 일부 예들에서, 팔레트 기반 디코딩 부 (165) 는 이전에 프로세싱된 데이터에 기초하여 팔레트에서의 엔트리들을 예측함으로써 팔레트를 생성할 수도 있다. 이전에 프로세싱된 데이터는 이전에 디코딩된 이웃 블록들에 대한 팔레트들, 또는 팔레트들로부터의 정보를 포함할 수도 있다. 팔레트 기반 디코딩 부 (165) 는 팔레트에서의 엔트리들이 예측될지의 여부를 나타내는 예측 선택스 엘리먼트를 수신할 수도 있다. 예측 선택스 엘리먼트는 루마 및 크로마 성분들에 대한 팔레트들에서의 엔트리들이 예측될지의 여부를 각각 나타내는 복수의 예측 선택스 엘리먼트들을 포함할 수도 있다.
- [0146] 예측 팔레트에 관해, 예를 들어, 예측 팔레트가 공간적으로 이웃하는 블록들 및/또는 블록들의 특정 스캔 순서에서의 이웃하는 블록들을 포함하는 하나 이상의 이웃 블록들로부터의 팔레트 엔트리들을 포함할 수도 있다. 일 예에서, 이웃 블록들은 현재 코딩되고 있는 블록의 좌측 (좌측 이웃 블록) 또는 상측 (상부 이웃 블록) 에 공간적으로 위치될 수도 있다. 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 현재 블록의 인과적 이웃에서의 가장 빈번한 샘플 값들을 사용하여 예측 팔레트 엔트리들을 결정할 수도 있다. 다른 예에서, 이웃 블록들은 블록들을 코딩하는데 사용되는 특정 스캔 순서에 따라 현재 코딩되고 있는 블록에 이웃할 수도 있다. 다시 말하면, 이웃 블록들은 스캔 순서에서 현재 블록보다 먼저 코딩된 하나 이상의 블록들일 수도 있다.

팔레트 기반 디코딩 부 (165) 는 팔레트(들)가 복사되는 이웃 블록들의 로케이션을 나타내는 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다.

[0147] 따라서, 일 예에서, 팔레트 기반 디코딩 부 (165) 는, 일부 예들에서, 슬라이스 또는 픽처에서의 좌측 이웃 블록 또는 상단 이웃 블록에 대한 팔레트에서의 엔트리들에 기초하여 팔레트에서의 엔트리들의 적어도 일부를 예측할 수도 있다. 이 경우, 좌측 이웃 블록 또는 상단 이웃 블록에 대한 어느 하나의 팔레트에서의 엔트리들에 기초하여 예측되는 팔레트에서의 엔트리들은 예측을 위한 좌측 이웃 블록 또는 상단 이웃 블록의 선택을 나타내는 선택스 엘리먼트에 기초하여 팔레트 기반 디코딩 부 (165) 에 의해 예측될 수도 있다. 선택스 엘리먼트는 예측을 위한 좌측 이웃 블록 또는 상단 이웃 블록의 선택을 나타내는 값을 갖는 플래그일 수도 있다.

[0148] 일부 예들에서, 팔레트 기반 디코딩 부 (165) 는 팔레트에서의 적어도 일부 선택된 엔트리들이 엔트리 단위 기반으로 예측될 것인지의 여부를 나타내는 하나 이상의 예측 선택스 엘리먼트들을 수신하고, 그에 따라 엔트리들을 생성할 수도 있다. 예를 들어, 팔레트 기반 디코딩 부 (165) 는 예측 팔레트의 각각의 엔트리에 대해, 팔레트 엔트리가 현재 블록에 대한 팔레트에 포함되는지의 여부를 나타내는 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다. 엔트리가 예측되지 않는다면, 팔레트 기반 디코딩 부 (165) 는 비-예측된 엔트리들, 뿐만 아니라 이러한 엔트리들의 수를 특정하는 하나 이상의 추가적인 선택스 엘리먼트들을 디코딩할 수도 있다. 따라서, 팔레트 기반 디코딩 부 (165) 는 엔트리들의 일부를 예측하고 추가적인 엔트리들의 수를 포함하는 팔레트에서의 다른 엔트리들을 직접적으로 특정하는 정보를 수신할 수도 있다.

[0149] 일부 예들에서, 전체 팔레트를 예측하기 위한 기법들은 팔레트의 하나 이상의 엔트리들을 예측하기 위한 기법들과 조합될 수도 있다. 예를 들어, 팔레트 기반 디코딩 부 (165) 는 현재 팔레트가 예측 팔레트로부터 완전히 복사되는지의 여부를 나타내는 비트스트림에서의 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다. 이것이 그 경우가 아니면, 팔레트 기반 디코딩 부 (165) 는 예측 팔레트에서의 각각의 엔트리가 복사되는지의 여부를 나타내는 비트스트림에서의 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다.

[0150] 다른 예에서, 엔트리들의 수와 팔레트 값을 수신하는 대신, 팔레트 기반 디코딩 부 (165) 는, 각각의 팔레트 값 뒤에, 시그널링된 팔레트 값이 팔레트에 대한 최종 팔레트 엔트리인지의 여부를 나타내는 플래그를 수신할 수도 있다. 팔레트 기반 디코딩 부 (165) 는 팔레트가 특정한 최대 사이즈에 이미 도달하였다면, 이러한 "팔레트의 말단" 플래그를 수신하지 못할 수도 있다.

[0151] 비디오 데이터 블록의 적어도 일부 포지션들과 팔레트에서의 엔트리들을 연관시키는, 팔레트 기반 디코딩 부 (165) 에 의해 수신된 정보는, 블록에서의 포지션들의 적어도 일부에 대한 팔레트 인덱스 값들을 나타내는 맵 정보를 포함할 수도 있으며, 팔레트 인덱스 값들의 각각은 팔레트에서의 엔트리들 중 하나의 엔트리에 대응한다. 맵 정보는 동일한 팔레트 인덱스 값을 갖는 블록에서의 연속적인 포지션들의 수를 각각이 나타내는 하나 이상의 런 선택스 엘리먼트들을 포함할 수도 있다.

[0152] 일부 예들에서, 팔레트 기반 디코딩 부 (165) 는 라인 복사를 나타내는 정보를 수신할 수도 있어서, 블록에서의 포지션들의 라인에 대한 화소 또는 인덱스 값들은 블록에서의 다른 라인의 포지션들에 대한 화소 또는 인덱스 값들로부터 복사된다. 팔레트 기반 디코딩 부 (165) 는 블록의 다양한 포지션들에 대한 팔레트에서의 화소 값들 또는 엔트리들을 결정하기 위해 이 정보를 사용하여 라인 복사를 수행할 수도 있다. 포지션들의 라인은 블록의 포지션들의 행, 행의 부분, 열 또는 열의 부분을 포함할 수도 있다.

[0153] 일부 예들에서, 팔레트 기반 디코딩 부 (165) 는 블록의 적어도 일부 포지션들과 팔레트에서의 엔트리들을 특정 순서로 연관시키는 정보에 의해 나타내어진 팔레트 인덱스 값들을 디코딩할 수도 있다. 예를 들면, 팔레트 인덱스 값들을 디코딩하기 위해 팔레트 기반 디코딩 부 (165) 에 의해 사용되는 스캔 순서는 팔레트 인덱스 값들을 인코딩하기 위해 팔레트 기반 인코딩 부, 이를테면 비디오 인코더 (20) 의 팔레트 기반 인코딩 부 (122) 에 의해 사용되는 동일한 스캔 순서일 수도 있다. 일부 예들에서, 팔레트 기반 디코딩 부 (165) 는 팔레트 인덱스 값들을 코딩하기 위해 사용되는 스캔 순서를 특정하는 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다.

[0154] 예를 들면, 위에서 논의된 바와 같이, 맵의 인덱스들은 래스터 스캔 순서라고 지칭될 수도 있는 것에서 상단에서 하단으로, 좌측에서 우측으로 스캔될 수도 있다. 래스터 스캔 순서에서, 현재 라인의 첫 번째 인덱스는 이전 라인의 마지막 인덱스 뒤에 바로 스캔될 수도 있다.

[0155] 그러나, 일부 예들에서, 래스터 스캔 순서로 인덱스들을 스캔하는 것은 효율적이지 않을 수도 있다. 예를 들면, 비디오 데이터 블록의 제 1 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이

터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 비디오 데이터 블록의 제 2 라인이 비디오 데이터 블록의 제 1 에지에 인접한 제 1 화소 및 비디오 데이터 블록의 제 2 에지에 인접한 마지막 화소를 포함하며, 제 1 라인의 마지막 화소가 제 2 라인의 마지막 화소에 인접하고, 제 1 에지 및 제 2 에지가 평행한 경우, 제 1 라인에서의 마지막 화소는 제 2 라인에서의 마지막 화소와 동일하지만, 제 2 라인에서의 제 1 화소와는 상이하다.

이 상황 (즉, 제 1 라인에서의 마지막 화소가 제 2 라인에서의 마지막 화소와 동일 또는 유사하지만 제 2 라인에서의 제 1 화소와는 상이한 경우) 은 컴퓨터 생성된 스크린 콘텐츠에서 비디오 콘텐츠의 다른 유형들보다 더 빈번하게 발생할 수도 있다. 다르게 말하면, 스크린 콘텐츠에 대해, 블록의 동일한 에지에 위치한 화소들이 블록의 대향 에지들에 위치한 화소들과는 동일 또는 유사한 값들을 가질 더 높은 확률이 있을 수도 있다.

래스터 스캐닝이 이러한 상관을 이용하지 못할 수도 있다.

[0156] 본 개시물의 하나 이상의 기법들에 따라, 팔레트 기반 디코딩 부 (165) 는 맵을 디코딩하는 경우 뱀형 스캔 순서를 이용할 수도 있다. 예를 들면, 팔레트 기반 디코딩 부 (165) 는 스캔 순서에서 제 1 라인의 마지막 화소 직후에 제 2 라인의 마지막 화소가 스캔되도록 맵을 디코딩할 수도 있다. 이런 식으로, 팔레트 기반 디코딩 부 (165) 는 런 길이 코딩의 효율을 개선할 수도 있다.

[0157] 팔레트 기반 디코딩 부 (165) 는 블록의 하나 이상의 포지션들에 대한 화소 값들을 수신하고, 그 화소 값들을 팔레트에서의 엔트리들에 추가하여 팔레트의 적어도 부분을 즉시 동적으로 생성함으로써 부분적으로 팔레트를 생성할 수도 있다. 화소 값들을 추가하는 것은 화소 값들을 엔트리들의 초기 세트를 포함하는 초기 팔레트에, 또는 엔트리들의 초기 세트를 포함하지 않는 엠프티 팔레트에 추가하는 것을 포함할 수도 있다. 일부 예들에서, 추가하는 것은, 새로운 엔트리들을 엔트리들의 초기 세트를 포함하는 초기 팔레트에 추가하거나 또는 초기 팔레트에서 기존의 엔트리들을 채우기 위해 화소 값들을 추가하는 것, 또는 초기 팔레트에서 엔트리들의 화소 값들을 교체 또는 변경하는 것을 포함한다.

[0158] 일부 예들에서, 팔레트 기반 디코딩 부 (165) 는 팔레트에 대한 고정된 최대 사이즈를 결정할 수도 있다. 최대 사이즈에 도달 시, 팔레트 기반 디코딩 부 (165) 는 팔레트의 하나 이상의 엔트리들을 제거할 수도 있다. 하나의 예에서, 팔레트 기반 디코딩 부 (165) 는 팔레트의 가장 오래된 엔트리를, 예컨대, FIFO 큐를 사용하여 제거할 수도 있다. 새로운 엔트리들이 대기행렬에 추가된 후, 전지작업 프로세스가 대기행렬에서 중복된 엔트리들을 제거하기 위해 적용될 수도 있다. 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 최소로 사용된 엔트리를 제거할 수도 있다. 또 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 제거할 후보 엔트리가 팔레트에 추가된 때 및 그 엔트리의 상대적 사용량에 기초하여 어떤 엔트리를 제거할지에 관한 가중된 결정을 할 수도 있다.

[0159] 일부 예들에서, 팔레트는 블록에서의 포지션들 중 하나의 포지션에 대해 팔레트로부터 선택된 화소 값이 블록에서의 포지션의 실제 화소 값과는 상이한 양자화된 팔레트일 수도 있어서, 디코딩 프로세스는 손실적이다. 예를 들어, 동일한 화소 값은 상이한 실제 화소 값들을 갖는 두 개의 상이한 포지션들에 대해 팔레트로부터 선택될 수도 있다.

[0160] 도 4는 본 개시물의 기법들과 일치하게, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 일 예를 도시하는 개념도이다. 도 4의 예는 제 1 팔레트들 (184) 에 연관되는 제 1 코딩 유닛 (CU) (180) 과 제 2 팔레트들 (192) 에 연관되는 제 2 CU (188) 를 갖는 픽처 (178) 를 포함한다. 아래에서 더욱 상세히 그리고 본 개시물의 기법들에 따라 설명되는 바와 같이, 제 2 팔레트들 (192) 은 제 1 팔레트들 (184) 에 기초한다. 픽처 (178) 는 인트라 예측 코딩 모드로 코딩된 블록 (196) 과 인터 예측 코딩 모드로 코딩되어 있는 블록 (200) 을 또한 포함한다.

[0161] 도 4의 기법들은 설명의 목적으로 비디오 인코더 (20) (도 1 및 도 2) 와 비디오 디코더 (30) (도 1 및 도 3) 의 맥락에서 그리고 HEVC 비디오 코딩 표준에 관해 설명된다. 그러나, 본 개시물의 기법들은 이런 식으로 제한되지 않고, 다른 비디오 코딩 프로세스들 및/또는 표준들에서 다른 비디오 코딩 프로세서들 및/또는 디바이스들에 의해 적용될 수도 있다는 것이 이해되어야 한다.

[0162] 대체로, 팔레트가 현재 코딩되고 있는 CU, 즉 도 4의 예에서의 CU (188) 에 대해 우세하며 그리고/또는 대표하는 다수의 화소 값들에 관련된다. 제 1 팔레트들 (184) 과 제 2 팔레트들 (192) 은 다수의 팔레트들을 포함하는 것으로서 도시되어 있다. 일부 예들에서, 비디오 코더 (이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 CU의 각각의 컬러 성분에 대해 팔레트들을 따로따로 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU의 루마 (Y) 성분에 대한 팔레트, CU의 크로마 (U) 성분에 대한 다른 팔레트, 및 CU의 크로마 (V) 성분에 대한 또 다른 팔레트를 인코딩할 수도 있다. 이 예에서, Y 팔레트의 엔트리들은 CU의 화소들의

Y 값들을 표현할 수도 있으며, U 팔레트의 엔트리들은 CU의 화소들의 U 값들을 표현할 수도 있고, V 팔레트의 엔트리들은 CU의 화소들의 V 값들을 표현할 수도 있다. 다른 예에서, 비디오 인코더 (20) 는 CU의 루마 (Y) 성분에 대한 팔레트, CU의 두 개의 성분들 (U, V) 에 대한 다른 팔레트를 인코딩할 수도 있다. 이 예에서, Y 팔레트의 엔트리들은 CU의 화소들의 Y 값들을 표현할 수도 있고, U-V 팔레트의 엔트리들은 CU의 화소들의 U-V 값 쌍들을 표현할 수도 있다.

[0163] 다른 예들에서, 비디오 인코더 (20) 는 CU의 모든 컬러 성분들에 대해 단일 팔레트를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 Y_i , U_i , 및 V_i 를 포함하는 3중 값인 i 번째 엔트리를 갖는 팔레트를 인코딩할 수도 있다. 이 경우, 팔레트는 화소들의 성분들의 각각에 대한 값들을 포함한다. 따라서, 다수의 개개의 팔레트들을 갖는 팔레트들의 세트로서의 팔레트들 (184 및 192) 의 표현은 단지 하나의 예이고 제한하려는 의도는 아니다.

[0164] 도 4의 예에서, 제 1 팔레트들 (184) 은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3을 각각 갖는 세 개의 엔트리들 (202~206) 을 포함한다. 엔트리들 (202~206) 은 인덱스 값들을 화소 값 A, 화소 값 B, 및 화소 값 C를 각각 포함하는 화소 값들에 관련시킨다. 본원에서 설명된 바와 같이, 제 1 CU (180) 의 실제 화소 값들을 코딩하는 것이 아니라, 비디오 코더 (이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 인덱스들 (1~3) 을 사용하여 블록의 화소들을 코딩하기 위해 팔레트 기반 코딩을 사용할 수도 있다. 다시 말하면, 제 1 CU (180) 의 각각의 화소 포지션에 대해, 비디오 인코더 (20) 는 화소에 대한 인덱스 값을 인코딩할 수도 있으며, 여기서 인덱스 값은 제 1 팔레트들 (184) 중 하나 이상의 제 1 팔레트들에서의 화소 값에 연관된다. 비디오 디코더 (30) 는 비트스트림으로부터 인덱스 값들을 획득하고 그 인덱스 값들 및 하나 이상의 제 1 팔레트들 (184) 을 사용하여 화소 값들을 복원할 수도 있다. 따라서, 제 1 팔레트들 (184) 는 팔레트 기반 디코딩에서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 인코더 (20) 에 의해 인코딩된 비디오 데이터 비트스트림에서 송신된다. 대체로, 하나 이상의 팔레트들이 각각의 CU에 대해 송신될 수도 있거나 또는 상이한 CU들 중에서 공유될 수도 있다.

[0165] 비디오 인코더 (20) 와 비디오 디코더 (30) 는 제 1 팔레트들 (184) 에 기초하여 제 2 팔레트들 (192) 을 결정할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU에 대한 팔레트가 하나 이상의 다른 CU들, 이를테면 이웃하는 CU들 (공간적으로 또는 스캔 순서에 기초함) 또는 인과적 이웃의 가장 빈번한 샘플들에 연관되는 하나 이상의 팔레트들로부터 예측되는지의 여부를 나타내기 위해 각각의 CU (일 예로서, 제 2 CU (188) 를 포함함) 에 대해 `pred_palette_flag`를 인코딩할 수도 있다. 예를 들어, 이러한 플래그의 값이 1과 동일한 경우, 비디오 디코더 (30) 는 제 2 CU (188) 를 위한 제 2 팔레트들 (192) 이 하나 이상의 이미 디코딩된 팔레트들로부터 예측되고 그러므로 제 2 CU (188) 를 위한 새로운 팔레트들이 `pred_palette_flag`를 포함하는 비트스트림에 포함되어 있지 않다고 결정할 수도 있다. 이러한 플래그가 0과 동일한 경우, 비디오 디코더 (30) 는 제 2 CU (188) 를 위한 팔레트 (192) 가 비트스트림에 새로운 팔레트로서 포함되어 있다고 결정할 수도 있다. 일부 예들에서, `pred_palette_flag`는 CU의 각각의 상이한 컬러 성분에 대해 따로따로 (예컨대, YUV 비디오에서의 CU에 대해 세 개의 플래그들, 즉, Y를 위한 플래그, U를 위한 플래그, 및 V를 위한 플래그로) 코딩될 수도 있다. 다른 예들에서, 단일 `pred_palette_flag`가 CU의 모든 컬러 성분들을 위해 코딩될 수도 있다.

[0166] 위의 예에서, `pred_palette_flag`는 현재 블록에 대한 팔레트의 엔트리들 중 임의의 엔트리가 예측되는지의 여부를 나타내기 위해 CU마다 시그널링된다. 일부 예들에서, 하나 이상의 선택스 엘리먼트들은 엔트리마다 기반으로 시그널링될 수도 있다. 다시 말하면 플래그가 팔레트 예측자 (predictor) 의 각각의 엔트리에 대해 그 엔트리가 현재 팔레트에 존재하는지의 여부를 나타내기 위해 시그널링될 수도 있다. 위에서 언급했듯이, 팔레트 엔트리가 예측되지 않는다면, 팔레트 엔트리는 명시적으로 시그널링될 수도 있다.

[0167] 제 1 팔레트들 (184) 을 기준으로 제 2 팔레트들 (192) 을 결정하는 (예컨대, `pred_palette_flag`가 1과 동일한) 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 예측 팔레트들, 즉 이 예에서 제 1 팔레트들 (184) 이 결정되는 하나 이상의 블록들의 위치를 찾아낼 수도 있다. 예측 팔레트들은 현재 코딩되고 있는 CU, 즉, 제 2 CU (188) 의 하나 이상의 이웃 CU들 (예컨대, 이를테면 이웃 CU들 (공간적으로 또는 스캔 순서에 기초함) 또는 인과적 이웃의 가장 빈번한 샘플들) 에 연관될 수도 있다. 하나 이상의 이웃 CU들의 팔레트들은 예측자 팔레트와 연관될 수도 있다. 일부 예들, 이를테면 도 4에 도시된 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, 제 2 CU (188) 를 위한 예측 팔레트를 결정하는 경우, 좌측 이웃 CU, 즉, 제 1 CU (180) 의 위치를 찾아낼 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 2 CU (188) 를 기준으로 다른 포지션들에서의 하나 이상의 CU들, 이를테면 상부 CU, 즉 CU (196) 의 위치를

찾아낼 수도 있다.

- [0168] 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 계층구조에 기초하여 팔레트 예측을 위한 CU를 결정할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 예측을 위해 좌측 이웃 CU, 즉 제 1 CU (180) 를 초기에 식별할 수도 있다. 좌측 이웃 CU가 예측을 위해 이용 가능하지 않다 (예컨대, 좌측 이웃 CU가 팔레트 기반 코딩 모드와는 다른 모드, 이를테면 인터 예측 모드 또는 인트라 예측 모드로 코딩되거나, 또는 픽처 또는 슬라이스의 최좌측 에지에 위치된다) 면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 상부 이웃 CU, 즉, CU (196) 를 식별할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 예측을 위해 이용 가능한 팔레트를 갖는 CU의 위치를 찾아낼 때까지 로케이션들의 미리 결정된 순서에 따라 이용 가능한 CU를 계속 검색할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 다수의 블록들 및/또는 이웃 블록의 복원된 샘플들에 기초하여 예측 팔레트를 결정할 수도 있다.
- [0169] 도 4의 예가 제 1 팔레트들 (184) 을 단일 CU, 즉, 제 1 CU (180) 로부터의 예측 팔레트들로서 예시하지만, 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 이웃 CU들의 조합으로부터 예측을 위한 팔레트들의 위치를 찾아낼 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더는 하나 이상의 공식들, 함수들, 규칙들 등을 적용하여 복수의 이웃 CU들 중 하나 또는 조합의 팔레트들에 기초하여 팔레트를 생성할 수도 있다.
- [0170] 또 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 팔레트 예측을 위한 다수의 잠재적 후보들을 포함하는 후보 리스트를 구축할 수도 있다. 전지작업 프로세스가 리스트에서의 중복된 후보들을 제거하기 위해 비디오 인코더 (20) 및 비디오 디코더 (30) 둘 다에서 적용될 수도 있다. 그런 예들에서, 비디오 인코더 (20) 는 팔레트 예측을 위해 사용되는 현재 CU가 선택되는 (예컨대, 팔레트를 복사하는) 리스트에서의 후보 CU를 나타내는 후보 리스트에 대한 인덱스를 인코딩할 수도 있다. 비디오 디코더 (30) 는 동일한 방식으로 후보 리스트를 구축하며, 그 인덱스를 디코딩하고, 디코딩된 인덱스를 사용하여 현재 CU와 함께 사용하기 위한 대응하는 CU의 팔레트를 선택할 수도 있다.
- [0171] 예시의 목적을 위한 일 예에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 현재 코딩되고 있는 CU 상측에 위치한 하나의 CU와 현재 코딩되고 있는 CU의 좌측에 위치되는 하나의 CU를 포함하는 후보 리스트를 구축할 수도 있다. 일부 경우들에서, 비디오 인코더 (20) 는 후보 선택을 나타내는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 현재 CU에 대한 팔레트가 현재 CU의 좌측에 위치한 CU로부터 복사됨을 나타내기 위해 0의 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 현재 CU에 대한 팔레트가 현재 CU 상측에 위치한 CU로부터 복사됨을 나타내기 위해 1의 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 디코더 (30) 는 그 플래그를 디코딩하고 팔레트 예측을 위한 적절한 CU를 선택한다.
- [0172] 또 다른 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 하나 이상의 다른 팔레트들에 포함된 샘플 값들이 하나 이상의 이웃 CU들에서 발생하는 빈도수에 기초하여 현재 코딩되고 있는 CU에 대한 팔레트를 결정한다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 미리 결정된 수의 CU들의 코딩 동안 가장 빈번하게 사용되는 인덱스 값들에 연관된 컬러들을 추적할 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재 코딩되고 있는 CU를 위한 팔레트에 가장 빈번하게 사용되는 컬러들을 포함시킬 수도 있다.
- [0173] 일부 예들에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 엔트리 방식 기반 팔레트 예측을 수행할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 각각의 예측 팔레트 엔트리들이 현재 팔레트에서 재사용되는지의 여부 (예컨대, 다른 CU의 팔레트에서의 화소 값들이 현재 팔레트에 의해 재사용되는지의 여부) 를 나타내는 예측 팔레트의 각각의 엔트리에 대한 하나 이상의 선택스 엘리먼트들, 이를테면 하나 이상의 플래그들을 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 주어진 엔트리가 예측 팔레트로부터의 예측된 값 (예컨대, 이웃하는 CU에 연관된 팔레트의 대응하는 엔트리) 인 경우 그 엔트리에 대해 1과 동일한 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 특정 엔트리가 다른 CU의 팔레트로부터 예측되지 않는다는 것을 나타내기 위해 특정 엔트리에 대해 0과 동일한 값을 갖는 플래그를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 비-예측된 팔레트 엔트리의 값을 나타내는 추가적인 데이터를 또한 인코딩할 수도 있다.
- [0174] 도 4의 예에서, 제 2 팔레트들 (192) 은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 엔트리 인덱스 값 3, 및 엔

트리 인덱스 4를 각각 갖는 네 개의 엔트리들 (208~214) 을 포함한다. 엔트리들 (208~214) 은 그 인덱스 값들을 화소 값 A, 화소 값 B, 화소 값 C, 및 화소 값 D를 각각 포함하는 화소 값들에 관련시킨다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 위에서 설명된 기법들 중 임의의 것을 사용하여 팔레트 예측 목적을 위한 제 1 CU들 (180) 의 위치를 찾아내고 제 2 CU (188) 를 코딩하기 위해 제 1 팔레트들 (184) 의 엔트리들 (1~3) 을 제 2 팔레트들 (192) 의 엔트리들 (1~3) 로 복사할 수도 있다. 이런 식으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 1 팔레트들 (184) 에 기초하여 제 2 팔레트들 (192) 을 결정할 수도 있다.

덧붙여서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 제 2 팔레트들 (192) 과 함께 포함될 엔트리 4에 대한 데이터를 코딩할 수도 있다. 이러한 정보는 예측자 팔레트로부터 예측되지 않는 팔레트 엔트리들의 수와 그들 팔레트 엔트리들에 대응하는 화소 값들을 포함할 수도 있다.

[0175] 일부 예들에서, 본 개시물의 양태들에 따르면, 하나 이상의 선택스 엘리먼트들은 팔레트들, 이를테면 제 2 팔레트들 (192) 이 예측 팔레트 (도 4에서 제 1 팔레트들 (184) 로서 도시되지만 하나 이상의 블록들로부터의 엔트리들로 이루어질 수도 있음)로부터 완전히 예측되는지의 여부 또는 제 2 팔레트들 (192) 의 특정 엔트리들이 예측되는지의 여부를 나타낼 수도 있다. 예를 들어, 초기 선택스 엘리먼트가 엔트리들의 모두가 예측되는지의 여부를 나타낼 수도 있다. 엔트리들의 모두가 예측되는 것은 아니라는 것 (예컨대, 0의 값을 갖는 플래그) 을 초기 선택스 엘리먼트가 나타내면, 하나 이상의 추가적인 선택스 엘리먼트들은 제 2 팔레트들 (192) 의 어떤 엔트리들이 예측 팔레트로부터 예측되는지를 나타낼 수도 있다.

[0176] 본 개시물의 일부 양태들에 따르면, 팔레트 예측에 연관된 특정 정보가 코딩되고 있는 데이터의 하나 이상의 특성들로부터 유추될 수도 있다. 다시 말하면, 비디오 인코더 (20) 가 선택스 엘리먼트들을 인코딩하 (고 비디오 디코더 (30) 가 이러한 선택스 엘리먼트들을 디코딩하) 는 것이 아니라 비디오 인코더 (20) 와 비디오 디코더 (30) 는 코딩되고 있는 데이터의 하나 이상의 특성들에 기초하여 팔레트 예측을 수행할 수도 있다.

[0177] 도 5는 본 개시물의 기법들과 일치하게, 화소들의 블록을 위한 팔레트로의 인덱스들을 결정하는 일 예를 도시하는 개념도이다. 예를 들어, 도 5는 인덱스 값들에 연관된 화소들의 각각의 포지션들을 팔레트들 (244) 의 엔트리에 관련시키는 인덱스 값들 (값 1, 2, 및 3) 의 맵 (240) 을 포함한다. 팔레트들 (244) 은 도 4에 관해 위에서 설명된 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 과 유사한 방식으로 결정될 수도 있다.

[0178] 다시, 도 5의 기법들은 설명의 목적으로 비디오 인코더 (20) (도 1 및 도 2) 와 비디오 디코더 (30) (도 1 및 도 3) 의 맥락에서 그리고 HEVC 비디오 코딩 표준에 관해 설명된다. 그러나, 본 개시물의 기법들은 이런 식으로 제한되지 않고, 다른 비디오 코딩 프로세스들 및/또는 표준들에서 다른 비디오 코딩 프로세서들 및/또는 디바이스들에 의해 적용될 수도 있다는 것이 이해되어야 한다.

[0179] 맵 (240) 이 도 5의 예에서 각각의 화소 포지션에 대한 인덱스 값을 포함하는 것으로 도시되지만, 다른 예들에서, 모든 화소 포지션들이 화소 값을 팔레트들 (244) 의 엔트리에 관련시키는 인덱스 값에 연관되는 것은 아닐 수도 있다는 것이 이해되어야 한다. 다시 말하면, 위에서 지적했듯이, 일부 예들에서, 화소 값이 팔레트들 (244) 에 포함되지 않는다면, 맵 (240) 에서의 포지션에 대한 실제 화소 값 (또는 그것의 양자화된 버전) 의 표시를 비디오 인코더 (20) 는 인코딩할 수도 있 (고 비디오 디코더 (30) 는 인코딩된 비트스트림으로 획득할 수도 있) 다.

[0180] 일부 예들에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 어떤 화소 포지션들이 인덱스 값들에 연관되는지를 나타내는 추가적인 맵을 코딩하도록 구성될 수도 있다. 예를 들어, 맵에서의 (i, j) 엔트리가 CU의 (i, j) 포지션에 대응한다고 가정한다. 비디오 인코더 (20) 는 맵의 각각의 엔트리 (즉, 각각의 화소 포지션) 에 대해 그 엔트리가 연관된 인덱스 값을 갖는지의 여부를 나타내는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU에서의 (i, j) 로케이션에 있는 화소 값이 팔레트들 (244) 에서의 값들 중 하나임을 나타내는 1의 값을 갖는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는, 이러한 일 예에서, 팔레트에서 그 화소 값을 나타내기 위해 그리고 비디오 디코더가 그 화소 값을 복원하는 것을 허용하기 위해 팔레트 인덱스 (도 5의 예에서 값 1~3으로서 도시됨) 를 또한 인코딩할 수도 있다. 팔레트들 (244) 이 단일 엔트리 및 연관된 화소 값을 포함하는 경우들에서, 비디오 인코더 (20) 는 인덱스 값의 시그널링을 스킵할 수도 있다. 비디오 인코더 (20) 는 CU에서의 (i, j) 로케이션에 있는 화소 값이 팔레트들 (244) 에서의 값들 중 하나가 아님을 나타내기 위해 0의 값을 갖도록 플래그를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 화소 값을 복원함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 화소 값의 표시를 또한 인코딩할 수도 있다. 일부 경우들에서, 화소 값은 손실 방식으로 코딩될 수도 있다.

[0181] CU의 하나의 포지션에서의 화소의 값은 CU의 다른 포지션들에서의 하나 이상의 다른 화소들의 값들의 표시를 제

공할 수도 있다. 예를 들어, CU의 이웃하는 화소 포지션들이 동일한 화소 값을 가질 또는 동일한 인덱스 값에 매핑될 수도 있을 비교적 높은 확률이 있을 수도 있다 (손실 코딩의 경우, 하나를 초과하는 화소 값이 단일 인덱스 값에 매핑될 수도 있다).

- [0182] 따라서, 비디오 인코더 (20) 는 동일한 화소 값 또는 인덱스 값을 갖는 주어진 스캔 순서에서의 다수의 연속적인 화소들 또는 인덱스 값들을 나타내는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 위에서 언급했듯이, 유사 값으로 된 화소 또는 인덱스 값들의 문자열 (string) 이 본원에서 런이라고 지칭될 수도 있다. 예시의 목적을 위한 일 예에서, 주어진 스캔 순서에서의 두 개의 연속적인 화소들 또는 인덱스들이 상이한 값들을 갖는다면, 런은 0과 동일하다. 주어진 스캔 순서에서의 두 개의 연속적인 화소들 또는 인덱스들이 동일한 값을 갖지만 스캔 순서에서의 세 번째 화소 또는 인덱스가 상이한 값을 갖는다면, 런은 1과 동일하다. 동일한 값을 갖는 세 개의 연속적인 인덱스들 또는 화소들에 대해, 런은 2이고 등등이다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 런을 나타내는 신택스 엘리먼트들을 획득하고 그 데이터를 사용하여 동일한 화소 또는 인덱스 값을 갖는 연속적인 로케이션들의 수를 결정할 수도 있다.

- [0183] 런에 포함될 수도 있는 인덱스들의 수는 스캔 순서에 의해 영향을 받을 수도 있다. 예를 들어, 맵 (240) 의 라인들 (266, 268, 및 270) 의 래스터 스캔을 고려한다. 수평 좌우 스캔 방향 (이들테면 도 6a에 예시된 바와 같은 래스터 스캐닝 순서) 을 가정하면, 행 266은 "1"의 세 개의 인덱스 값들, "2"의 두 개의 인덱스 값들, 및 "3"의 세 개의 인덱스 값들을 포함한다. 행 268은 "1"의 5개의 인덱스 값들과 "3"의 세 개의 인덱스 값들을 포함한다. 이 예에서, 행 266에 대해, 비디오 인코더 (20) 는 행 266의 첫 번째 값 (그 행의 최좌측 값) 이 2의 런을 갖는 1이며, 1의 런을 갖는 2의 인덱스 값이 그 뒤를 따르며, 2의 런을 갖는 3의 인덱스 값이 그 뒤를 따름을 나타내는 신택스 엘리먼트들을 인코딩할 수도 있다. 래스터 스캔을 따라, 비디오 인코더 (20) 는 그러면 최좌측 값으로 행 268의 코딩을 시작할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 행 268의 첫 번째 값이 4의 런을 갖는 1이며, 2의 런을 갖는 3의 인덱스 값이 그 뒤를 따름을 나타내는 신택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 는 라인 270에 대해서도 동일한 방식으로 진행할 수도 있다.

- [0184] 그런고로, 래스터 스캔 순서에서, 현재 라인의 첫 번째 인덱스는 이전 라인의 마지막 인덱스 뒤에 바로 스캔될 수도 있다. 그러나, 일부 예들에서, 래스터 스캔 순서로 인덱스들을 스캔하는 것이 바람직하지 않을 수도 있다. 예를 들면, 비디오 데이터 블록의 제 1 라인 (예컨대, 행 266) 이 비디오 데이터 블록의 제 1 예지에 인접한 제 1 화소 (예컨대, 행 266의 최좌측 화소, 이는 1의 인덱스 값을 가짐) 와 비디오 데이터 블록의 제 2 예지에 인접한 마지막 화소 (예컨대, 행 (266) 의 최우측 화소, 이는 3의 인덱스 값을 가짐) 를 포함하며, 비디오 데이터 블록의 제 2 라인 (예컨대, 행 268) 이 비디오 데이터 블록의 제 1 예지에 인접한 제 1 화소 (예컨대, 행 268의 맨 왼쪽 화소, 이는 1의 인덱스 값을 가짐) 와 비디오 데이터 블록의 제 2 예지에 인접한 마지막 화소 (예컨대, 행 268의 최우측 화소, 이는 3의 인덱스 값을 가짐) 를 포함하며, 제 1 라인의 마지막 화소는 제 2 라인의 마지막 화소에 인접하고, 제 1 예지와 제 2 예지는 평행하며, 제 1 라인에서의 마지막 화소는 제 2 라인에서의 마지막 화소와는 동일한 인덱스 값을 갖지만 제 2 라인에서의 제 1 화소와는 상이한 인덱스 값을 가지는 래스터 스캔 순서로 인덱스들을 스캔하는 것이 바람직하지 않을 수도 있다. 이 상황 (즉, 제 1 라인에서의 마지막 화소의 인덱스 값이 제 2 라인에서의 마지막 화소와 동일하지만 제 2 라인에서의 제 1 화소와는 상이한 경우) 은 컴퓨터 생성된 스크린 콘텐츠에서 비디오 콘텐츠의 다른 유형들보다 더 빈번하게 발생할 수도 있다.

- [0185] 본 개시물의 하나 이상의 기법들에 따라, 비디오 인코더 (20) 가 맵의 인덱스들을 인코딩하는 경우 뱀형 스캔 순서를 이용할 수도 있다. 예를 들면, 비디오 인코더 (20) 는 제 1 라인의 마지막 화소 직후에 제 2 라인의 마지막 화소를 스캔할 수도 있다. 이런 식으로, 비디오 인코더 (20) 는 런 길이 코딩의 효율을 개선할 수도 있다.

- [0186] 예를 들어, 본 개시물의 하나 이상의 기법들에 따라, 래스터 스캔 순서를 사용하는 것과는 대조적으로, 비디오 인코더 (20) 는 뱀형 스캔 순서를 사용하여 맵 (240) 의 값들을 코딩할 수도 있다. 예시의 목적을 위한 일 예에서, 맵 (240) 의 행들 (266, 268, 및 270) 을 고려한다. 뱀형 스캔 순서 (이들테면 도 6b에 예시된 바와 같은 뱀형 스캐닝 순서) 를 사용하여, 비디오 인코더 (20) 는 행 266의 좌측 포지션에서 시작하여, 행 266의 최우측 포지션으로 진행하며, 행 268의 최좌측 포지션을 향해 아래로 이동하며, 행 268의 최좌측 포지션으로 진행하고, 행 270의 최좌측 포지션을 향해 아래로 이동하면서 맵 (240) 의 값들을 코딩할 수도 있다. 예를 들면, 비디오 인코더 (20) 는 행 266의 제 1 포지션이 1이라는 것과 스캔 방향에서의 두 개의 연속적인 엔트리들의 다음 런이 행 266의 제 1 포지션과 동일하다는 것을 나타내는 하나 이상의 신택스 엘리먼트들을 인코딩할 수

도 있다. 비디오 인코더 (20) 는 행 266의 다음 포지션 (즉, 좌우로 제 4 포지션) 이 2라는 것과 스캔 방향에서의 다음의 연속적인 엔트리들이 행 266의 제 4 포지션과는 동일하다는 것을 나타내는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 는 행 266의 다음 포지션 (즉, 제 6 포지션) 이 3이란 것과 스캔 방향에서 다음 런의 다섯 개의 연속적인 엔트리들이 행 266의 제 6 포지션과는 동일하다는 것을 나타내는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 는 행 268의 스캔 방향에서의 다음 포지션 (즉, 우좌로 행 268의 제 4 포지션) 이 1이라는 것과 스캔 방향에서의 다음 런의 아홉 개의 연속적인 엔트리들이 행 268의 제 4 포지션과는 동일하다는 것을 나타내는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다.

[0187] 이런 식으로, 뱀형 스캔 순서를 사용함으로써, 비디오 인코더 (20) 는 더 긴 길이의 런들을 인코딩할 수도 있으며, 이는 코딩 효율을 개선할 수도 있다. 예를 들어, 래스터 스캔을 사용하여, 행 266의 최종 런 (인덱스 값 3의 경우임) 이 2와 동일하다. 뱀형 스캔을 사용하면, 그러나, 행 (266) 의 최종 런은 행 268 속으로 연장하고 5와 동일하다.

[0188] 비디오 디코더 (30) 는 위에서 설명된 선택스 엘리먼트들을 수신하고 행들 (266, 268, 및 270) 을 복원할 수도 있다. 예를 들어, 비디오 디코더 (30) 는, 인코딩된 비트스트림으로부터, 현재 코딩되고 있는 맵 (240) 의 포지션에 대한 인덱스 값을 나타내는 데이터를 획득할 수도 있다. 비디오 디코더 (30) 는 동일한 인덱스 값을 갖는 스캔 순서에서의 연속적인 포지션들의 수를 데이터를 또한 획득할 수도 있다.

[0189] 도 6a와 도 6b는 본 개시물의 기법들에 일치하게, 예의 스캔 순서들을 도시하는 개념도들이다. 도 6a에 예시된 바와 같이, 블록 600A가 제 1 화소 (606A) 와 마지막 화소 (608A) 를 포함하는 제 1 라인 (602A), 제 1 화소 (610A) 와 마지막 화소 (612A) 를 포함하는 제 2 라인 (604A), 및 스캔 순서 (614A) 를 포함할 수도 있다. 도 6b에 예시된 바와 같이, 블록 600A가 제 1 화소 (606B) 와 마지막 화소 (608B) 를 포함하는 제 1 라인 (602B), 제 1 화소 (610B) 와 마지막 화소 (612B) 를 포함하는 제 2 라인 (604B), 및 스캔 순서 (614B) 를 포함할 수도 있다. 도 6a에 예시된 바와 같이, 제 1 화소 (606A) 와 제 1 화소 (610A) 는 제 1 에지 (616A) 에 인접할 수도 있으며, 마지막 화소 (608A) 와 마지막 화소 (612A) 는 제 2 에지 (618A) 에 인접할 수도 있고, 제 1 에지 (616A) 는 제 2 에지 (618A) 에 인접할 수도 있다. 마찬가지로, 제 1 화소 (606B) 와 제 1 화소 (610B) 는 제 1 에지 (616B) 에 인접할 수도 있으며, 마지막 화소 (608B) 와 마지막 화소 (612B) 는 제 2 에지 (618B) 에 인접할 수도 있고, 제 1 에지 (616B) 는 제 2 에지 (618B) 에 인접할 수도 있다.

[0190] 도 6a의 스캔 순서 (614A) 는 래스터 스캔 순서라고 지칭될 수도 있는데, 제 2 라인 (604A) 에서의 첫 번째 값 (610A) 이 이전 라인 (602A) 에서의 마지막 값 (608A) 직후에 스캔되기 때문이다. 도 6b의 스캔 순서 (614B) 는 제 2 라인 (604B) 에서의 첫 번째 값 (610B) 이 이전 라인 (602B) 에서의 마지막 값 (608B) 직후에 스캔되는 뱀형 스캔 순서라고 지칭될 수도 있다.

[0191] 본 개시물의 하나 이상의 기법들에 따라, 래스터 스캔 순서 (예컨대, 스캔 순서 (614A)) 를 사용하여 화소들을 스캔하는과는 대조적으로, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 팔레트 모드 비디오 코딩을 수행하는 경우 뱀형 스캔 순서 (예컨대, 스캔 순서 (614B)) 를 사용하여 화소들을 스캔할 수도 있다.

[0192] 도 7a 내지 도 7c는 본 개시물의 기법들에 일치하게, 팔레트 기반 비디오 코딩의 예의 모드들을 도시하는 개념도들이다. 위에서 논의된 바와 같이, 맵의 인덱스들을 그 맵의 하나 이상의 다른 인덱스들을 기준으로 코딩하는 JCTVC-P0035에 의해 설명된 제 1 방법은, 세 개의 코딩 모드들, 즉, 수평 모드, 수직 모드, 및 정상 모드를 제공한다.

[0193] 인덱스들을 수평 모드를 사용하여 코딩하는 경우, 현재 라인의 인덱스 값들의 모두가 동일할 수도 있다 (즉, 전체 라인들이 동일한 컬러 인덱스를 공유할 수도 있다). 이처럼, 수평 모드에서, 비디오 인코더가 전체 행에 대해 단일 인덱스를 인코딩할 수도 있다. 도 7a에 의해 도시된 바와 같이, 비디오 디코더가 후보 시작 포지션 1 (702) 에 있는 인덱스 (즉, 인덱스 값 0을 가짐) 를 현재 라인 (700) 전체에 걸쳐 (즉, 현재 라인 (700) 의 모든 화소들에 대한 인덱스 값들이 0으로서 디코딩되도록) 복사할 수도 있다.

[0194] 인덱스들을 수직 모드를 사용하여 코딩하는 경우, 현재 라인의 인덱스들은 상부 라인의 인덱스들과 동일할 수도 있다 (즉, 전체 라인은 상측 라인과는 동일할 수도 있다). 이 경우, 비디오 디코더는 상부 라인으로부터의 인덱스들을 현재 라인에 복사할 수도 있다. 도 7b에 의해 예시된 바와 같이, 비디오 디코더가 상부 라인 (706) 의 인덱스들을 현재 라인 (704) 에 (즉, 현재 라인 (704) 의 인덱스들이 상부 라인 (706) 의 인덱스들과

는 동일하도록 복사할 수도 있다. 예를 들면, 비디오 디코더는 후보 시작 포지션 2 (710)로부터의 인덱스 값을 포지션 (712)에 복사할 수도 있다.

[0195] 인덱스들을 정상 모드를 사용하여 코딩하는 경우, 비디오 코더가 각각의 화소 포지션에 대한 인덱스가 각각의 화소 포지션 좌측의 화소 포지션에 대한 인덱스와 동일한지 또는 각각의 화소 포지션 상측의 화소 포지션에 대한 인덱스와 동일한지를 나타내는 제 각각의 화소 포지션에 대한 플래그를 코딩할 수도 있다. 각각의 화소 포지션에 대한 인덱스가 상기 각각의 화소 포지션 좌측의 화소 포지션에 대한 인덱스 또는 상기 각각의 화소 포지션 상측의 화소 포지션에 대한 인덱스와 동일하지 않다면, 비디오 코더는 상기 각각의 화소 포지션에 대한 인덱스를 코딩할 수도 있다. 도 7c에 의해 예시된 바와 같이, 비디오 코더가, 현재 포지션 (714)에 후보 시작 포지션 3 (716)의 컬러 인덱스 (즉, 현재 포지션 (714) 상측의 화소 포지션의 인덱스)가 복사되어야 하는지 또는 후보 시작 포지션 4 (718)의 컬러 인덱스 (즉, 현재 포지션 (714) 좌측의 화소 포지션의 인덱스)가 복사되어야 하는지를 나타내는 현재 포지션 (714)에 대한 플래그를 코딩할 수도 있다.

[0196] 도 8a 내지 도 8c는 본 개시물의 기법들에 일치하게, 팔레트 기반 비디오 코딩의 예의 모드들을 도시하는 개념도들이다. 위에서 논의된 바와 같이, 맵의 인덱스들을 그 맵의 하나 이상의 다른 인덱스들을 기준으로 코딩하는 JCTVC-P0035에 의해 설명된 제 2 방법은, 세 개의 코딩 모드들, 즉, "상측 복사 모드", "런 모드", 및 "화소 모드"를 제공한다.

[0197] "상측 복사 모드"에서, 비디오 코더가 뒤따르는 N 개의 팔레트 인덱스들이 그것들 상측 이웃들과는 각각 동일하다는 것을 나타내는 값 "copy_run" (N)을 코딩할 수도 있다. 도 8a에 의해 예시된 바와 같이, 비디오 코더가, 현재 라인 (802)의 다음의 다섯 개의 팔레트 인덱스들이 그것들의 상부 라인 (804)으로부터의 상측 이웃들 (즉, 후보 시작 포지션 1 (806)에 의해 나타내어진 다섯 개의 포지션들)과 각각 동일하다는 것을 나타내는 5의 copy_run 값을 코딩할 수도 있다. 다른 예로서, 도 8a에서, 비디오 코더가, 현재 라인 (802)이하의 다섯 (5)개 팔레트 인덱스들이 그것들의 상부 라인 (804)으로부터의 상측 이웃들과 동일하다는 것을 나타내는 사 (4)의 copy_run 값을 코딩할 수도 있다.

[0198] "런 모드"에서, 비디오 코더가, 다음 M 개의 팔레트 인덱스들이 시그널링된 팔레트 인덱스와 동일함을 나타내는 "palette_run" (M)이 뒤따르는 팔레트 인덱스를 코딩할 수도 있다. 도 8b에 예시된 바와 같이, 비디오 코더가, 후보 시작 포지션 2 (808)에 대한 팔레트 인덱스와 뒤따르는 다섯 개 포지션들의 인덱스 값들이 후보 시작 포지션 2 (808)에 대한 인덱스 값과 동일함을 나타내는 5의 값을 갖는 palette_run을 코딩할 수도 있다.

[0199] "화소 모드"에서, 비디오 코더가 예측 플래그를 코딩할 수도 있다. 비디오 코더는 현재 화소의 값이 예측 잔차와 예측자로서의 복원된 상단 이웃 화소를 사용하여 코딩됨을 나타내는 1로서 플래그를 코딩할 수도 있다. 비디오 코더는 화소 값이 예측 없이 송신됨을 나타내는 0로서 플래그를 코딩할 수도 있다. 도 8c에 예시된 바와 같이, 예측 플래그가 1이면, 비디오 코더가 상부 라인 (812)에서의 후보 시작 포지션 3 (810)에 대한 화소 값을 현재 라인 (814)에서의 대응하는 포지션에 복사할 수도 있다.

[0200] 도 9는 본 개시물의 기법들에 일치하게, 팔레트 코딩 모드를 사용하여 비디오 데이터를 코딩하는 일 예의 프로세스를 도시하는 흐름도이다. 도 9의 방법은 비디오 코더, 이를테면 비디오 인코더 (20) (도 1 및 도 2) 또는 비디오 디코더 (30) (도 1 및 도 3)에 대해 설명된다. 그러나, 다른 비디오 디코딩 디바이스들이 유사한 방법을 수행하도록 구성될 수도 있다는 것이 이해되어야 한다. 더구나, 그 방법에서의 특정 단계들은 상이한 순서로 또는 병렬로 수행될 수도 있다. 비슷하게, 특정 단계들은 생략될 수도 있고, 다른 단계들이 다양한 예들에서 추가될 수도 있다.

[0201] 비디오 코더, 이를테면 비디오 인코더 및/또는 비디오 디코더 (30)가, 비디오 데이터 블록에 대해, 복수의 각각의 화소 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정할 수도 있다 (902). 일부 예들에서, 비디오 데이터 블록의 제 1 라인 (예컨대, 도 6b의 제 1 라인 (602B))이 비디오 데이터 블록의 제 1 예지 (예컨대, 도 6b의 제 1 예지 (616B))에 인접한 제 1 화소 (예컨대, 도 6b의 제 1 화소 (606B))와 비디오 데이터 블록의 제 2 예지 (예컨대, 도 6b의 제 2 예지 (618B))에 인접한 마지막 화소 (예컨대, 도 6b의 마지막 화소 (608B))를 포함할 수도 있으며, 비디오 데이터 블록의 제 2 라인 (예컨대, 도 6b의 제 2 라인 (604B))이 비디오 데이터 블록의 제 1 예지에 인접한 제 1 화소 (예컨대, 도 6b의 제 1 화소 (610B))와 비디오 데이터 블록의 제 2 예지에 인접한 마지막 화소 (예컨대, 도 6b의 마지막 화소 (612B))를 포함할 수도 있다. 일부 예들에서, 제 1 라인의 마지막 화소는 제 2 라인의 마지막 화소에 인접할 수도 있다. 일부 예들에서, 제 1 예지와 제 2 예지와 평행할 수도 있다.

- [0202] 비디오 코더는 제 2 라인의 화소가 스캔 순서에서 제 1 라인의 화소를 바로 뒤따르는 스캔 순서로, 블록의 화소들을 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩할 수도 있다 (904). 예를 들면, 비디오 인코더 (20) 는 뱀형 스캔 순서 (예컨대, 도 6b에 예시된 바와 같음) 를 사용하여 인덱스 값들에 연관된 비디오 데이터 블록의 화소들의 각각의 포지션들을 팔레트들의 엔트리에 관련시키는 인덱스 값들의 맵 (예컨대, 도 5의 맵 (240)) 을 표현하는 신택스 엘리먼트들을 인코딩할 수도 있다.
- [0203] 일부 예들, 이를테면 도 6b의 예에서, 제 1 라인은 제 1 행일 수도 있고 제 2 라인은 제 2 행일 수도 있다. 일부 예들에서, (예컨대, 도 6b의 예가 90 도 회전되었다면) 제 1 라인은 제 1 열일 수도 있고 제 2 라인은 제 2 열일 수도 있다.
- [0204] 일부 예들에서, 비디오 코더는 제 2 라인의 마지막 화소가 스캔 순서에서 제 1 라인의 마지막 화소 직후에 있다는 것을 나타내는 신택스 엘리먼트를 코딩할 수도 있다. 예를 들면, 비디오 코더는 뱀형 스캔 순서를 나타내는 신택스 엘리먼트를 코딩할 수도 있다. 추가적으로 또는 대안으로, 비디오 코더는, 예를 들어, 블록 사이즈, 컬러 공간, 및/또는 컬러 성분과 같은 이른바 사이드 정보에 기초하여 스캔 순서를 시그널링 또는 유추할 수도 있다.
- [0205] 일부 예들에서, 팔레트에서의 각각의 엔트리는 고유 인덱스 값을 각각 포함할 수도 있다. 그런 예들에서, 비디오 코더는 인덱스 값들 및/또는 하나 이상의 이스케이프 화소들을 코딩할 수도 있다. 일부 예들에서, 비디오 코더는 절단된 이진수 코드를 사용하여 인덱스 값 및 이스케이프 화소들 중 하나 또는 양쪽 모두를 코딩할 수도 있다.
- [0206] 도 9의 방법이 비디오 인코더에 의해 수행되는 것과 같은 일부 예들에서, 비디오 인코더는 팔레트를 생성할 수도 있다. 예를 들면, 비디오 인코더는 도 10의 기법들에 따라 팔레트를 생성할 수도 있다.
- [0207] 도 10은 본 개시물의 기법들과 일치하게, 비디오 데이터의 블록을 위한 팔레트를 결정하는 일 예의 프로세스를 도시하는 개념도이다. 도 10의 기법들은 비디오 인코더, 이를테면 도 1 및 도 2에 예시된 비디오 인코더 (20) 에 의해 수행될 수도 있다. 예시의 목적을 위해, 도 10의 기법들은 도 1 및 도 2의 비디오 인코더 (20) 의 맥락 내에서 설명되지만, 비디오 인코더 (20) 의 구성과는 상이한 구성들을 갖는 비디오 코더들이 도 10의 기법들을 수행할 수도 있다.
- [0208] 비디오 인코더, 이를테면 비디오 인코더 (20) 는, 비디오 데이터 블록에 대한 팔레트를 생성할 수도 있다. 예를 들면, 비디오 인코더 (20) 는 각각의 화소 값과 상기 각각의 화소 값에 유사한 화소 값들을 갖는 화소들의 각각의 양을 각각 포함하는 복수의 엔트리들을 갖는 히스토그램을 도출할 수도 있다. 하나의 예로서, 비디오 인코더 (20) 의 팔레트 기반 인코딩 부 (PBEU) (122) 는 비디오 데이터 블록에 대한 히스토그램 (H) 을 결정할 수도 있다 (1002). 히스토그램 (H) 은 $H = \{(v_i, f_i), i = \{0, 1, 2, \dots, M\}\}$ 에 의해 표현될 수도 있으며 여기서 $M + 1$ 은 현재 블록에서의 상이한 화소 값들의 수이며, v_i 는 화소 값이며, f_i 는 v_i 의 발생들의 수 (즉, 현재 블록에서의 얼마나 많은 화소들이 화소 값 (v_i) 을 가지는지) 이다. 일부 예들에서, 히스토그램의 복수의 엔트리들 중 각각의 엔트리는 동일한 화소 값을 갖는 화소들의 양을 나타낸다. 일부 예들에서, 히스토그램의 복수의 엔트리들 중 각각의 엔트리는 양자화되는 경우 동일한 화소 값을 갖는 화소들의 양을 나타낸다.
- [0209] 일부 예들에서, 비디오 인코더는, 히스토그램에 기초하여, 비디오 데이터 블록의 하나 이상의 화소들을 메이저 (major) 컬러 화소들 또는 이스케이프 (escape) 화소들 (이것들은 메이저 컬러 화소들로서 분류된 화소들보다는 더 낮은 양의 화소들을 나타내는 히스토그램에서의 엔트리들에 대응할 수도 있음) 로서 분류할 수도 있다. 일부 예들에서, 비디오 코더는 이스케이프 화소들로서 분류된 화소들에 대해 실제 값을 코딩할 수도 있다 (즉, 인덱스 값들을 코딩하는 것과는 대조적임). 이스케이프 화소들에 대한 실제 값들은 양자화될 수도 있거나 또는 양자화되지 않을 수도 있다.
- [0210] PBEU (122) 는 하나 이상의 변수들을 초기화할 수도 있다 (1004). 예를 들면, PBEU (122) 는 팔레트 (P) 를 임프티로, 팔레트 (P) 의 인덱스 (idx) 를 0으로, 그리고/또는 히스토그램 $H(j)$ 의 인덱스를 영으로 초기화할 수도 있다.
- [0211] PBEU (122) 는 히스토그램 (H) 을 정렬할 수도 있다 (1006). 예를 들면, PBEU (122) 는 f_i 의 내림 차순에 따라 히스토그램 (H) 을 정렬할 수도 있어서, 더 많은 발생들을 갖는 화소들은 앞쪽에 가깝게 배치된다. 정렬된 히스토그램은 $H_0 = \{(u_i, f_i), i = \{0, 1, 2, \dots, M\}, f_i \geq f_{i+1}\}$ 에 의해 표현될 수도 있다.

- [0212] PBEU (122) 는 엔트리를 팔레트 (P) 에 삽입할 수도 있다 (1008). 예를 들면, PBEU (122) 는 정렬된 히스토그램 (H_0) 으로부터의 엔트리 (j, u_j) 를 팔레트 (P) 속으로 복사할 수도 있다.
- [0213] PBEU (122) 는 팔레트 인덱스 (1012) 및 히스토그램 인덱스 (1016) 를 검증시킬 수도 있다. PBEU (122) 는 정렬된 히스토그램 (H_0) 에서 다음 엔트리에 대응하는 화소 값이 팔레트에 이미 포함된 화소 값들 중 임의의 화소 값의 인근 이내에 있는지의 여부를 결정할 수도 있다 (1020). 예를 들면, PBEU (122) 는 정렬된 히스토그램 (H_0) 에서의 다음 엔트리에 대응하는 화소 값 및 팔레트에 이미 포함된 하나 이상의 화소 값들 간의 차이에 대응하는 거리 값을 결정할 수도 있다. 결정된 차이 값이 임계값을 충족시킨다면, PBEU (122) 는 다음 엔트리에 대응하는 화소 값을 포함하는 엔트리를 팔레트 (P) 에 삽입할 수도 있다 (1020의 "아니오" 분기, 1008). 결정된 차이 값이 그 임계값을 충족시키지 않는다면, PBEU (122) 는 히스토그램 인덱스를 검증시킬 수도 있다 (1020의 "예" 분기, 1016).
- [0214] 일부 예들에서, PBEU (122) 는 하나 이상의 조건들이 하나 이상의 각각의 임계값들을 충족시킨다면 팔레트를 결정하는 것을 중단할 수도 있다. 하나의 예로서, PBEU (122) 는 팔레트 (P) 의 사이즈가 임계 사이즈보다 더 큰 경우 팔레트를 결정하는 것을 중단할 수도 있다 (1010의 "예" 분기). 다른 예로서, PBEU (122) 는 팔레트 인덱스 (idx) 의 값이 임계값 (M) 과 동일한 경우 팔레트를 결정하는 것을 중단할 수도 있다 (1014의 "예" 분기). 다른 예로서, PBEU (122) 는 히스토그램 인덱스 (j) 의 값이 임계값 (M) 과 동일한 경우 팔레트를 결정하는 것을 중단할 수도 있다 (1018의 "예" 분기). 일부 예들에서, M은 현재 블록에서의 상이한 화소 값들의 수에 대응할 수도 있다.
- [0215] 다음의 번호 매겨진 실시예들은 본 개시물의 하나 이상의 양태들을 예시할 수도 있다:
- [0216] 실시예 1. 비디오 데이터를 코딩하는 방법으로서, 비디오 코더에 의해 그리고 비디오 데이터 블록에 대해, 복수의 각각의 컬러 값들을 나타내는 복수의 엔트리들을 갖는 팔레트를 결정하는 단계로서, 비디오 데이터 블록의 제 1 라인은 비디오 데이터 블록의 에지에 인접하게 위치한 화소를 포함하고, 비디오 데이터 블록의 제 2 라인은 블록의 에지에 인접하게 그리고 제 1 라인의 화소에 인접하게 위치한 화소를 포함하는, 상기 팔레트를 결정하는 단계; 및 제 2 라인의 화소가 스캔 순서에서 제 1 라인의 화소를 바로 뒤따르는 스캔 순서로, 블록의 화소들을 팔레트에서의 엔트리들에 매핑하는 인덱스 값들을 코딩하는 단계를 포함하는, 비디오 데이터 코딩 방법.
- [0217] 실시예 2. 실시예 1의 방법에 있어서, 제 1 라인은 제 1 행이고 제 2 라인은 제 2 행이고, 스캔 순서로 인덱스 값들을 코딩하는 단계는 제 2 행에서의 화소에 대한 인덱스 값을 코딩하기 바로 전에 제 1 행에서의 화소에 대한 인덱스 값을 코딩하는 단계를 포함하는, 비디오 데이터 코딩 방법.
- [0218] 실시예 3. 실시예 1에 있어서, 제 1 라인은 제 1 열이고 제 2 라인은 제 2 열이고, 스캔 순서로 인덱스 값들을 코딩하는 단계는 제 2 열에서의 화소에 대한 인덱스 값을 코딩하기 바로 전에 제 1 열에서의 화소에 대한 인덱스 값을 코딩하는 단계를 포함하는, 비디오 데이터 코딩 방법.
- [0219] 실시예 4. 실시예 1 내지 실시예 3 중 어느 하나의 실시예에 있어서, 에지는 제 1 에지이며, 제 2 라인의 화소는 제 2 라인의 제 1 화소이며, 제 2 라인은 제 1 에지에 평행한 비디오 데이터 블록의 제 2 에지에 인접하게 위치한 제 2 화소를 포함하고, 인덱스 값들을 코딩하는 단계는, 동일한 인덱스 값을 갖는 스캔 순서에서의 연속적인 화소들의 런을 나타내는 하나 이상의 신택스 엘리먼트들을 코딩하는 단계로서, 연속적인 화소들의 런은 제 1 라인의 화소와 제 2 라인의 제 1 화소를 포함하지만 제 2 라인의 제 2 화소를 포함하지 않는, 상기 하나 이상의 신택스 엘리먼트들을 코딩하는 단계를 포함하는, 비디오 데이터 코딩 방법.
- [0220] 실시예 5. 실시예 1 내지 실시예 4 중 어느 하나의 실시예에 있어서, 제 2 라인의 화소가 스캔 순서에서 제 1 라인의 화소 바로 뒤에 스캔됨을 나타내는 신택스 엘리먼트를 코딩하는 단계를 더 포함하는, 비디오 데이터 코딩 방법.
- [0221] 실시예 6. 실시예 1 내지 실시예 5 중 어느 하나의 실시예에 있어서, 팔레트에서의 각각의 엔트리는 복수의 각각의 컬러 값들 중 각각의 컬러 값에 대응하는 고유 인덱스 값을 각각 포함하며, 그 방법은, 복수의 각각의 컬러 값들 중 각각의 컬러 값들에 대응하지 않는 이스케이프 화소들에 대한 값들을 코딩하는 단계를 더 포함하며, 인덱스 값들 및 이스케이프 화소들을 코딩하는 것은 절단된 이진수 코드를 사용하여 상기 인덱스 값들 및 상기 이스케이프 화소들 중 하나 또는 양쪽 모두를 코딩하는 것을 포함하는, 비디오 데이터 코딩 방법.
- [0222] 실시예 7. 실시예 1 내지 실시예 6 중 어느 하나의 실시예에 있어서, 비디오 코더는 비디오 인코더이며, 인덱스 값들을 코딩하는 단계는 인덱스 값들을 인코딩하는 단계를 포함하며, 그 방법은, 팔레트와 인덱스 값들을

표현하는 하나 이상의 선택스 엘리먼트들을 포함하는 비트스트림을 생성하는 단계를 더 포함하는, 비디오 데이터 코딩 방법.

- [0223] 실시예 8. 실시예 1 내지 실시예 7 중 어느 하나의 실시예에 있어서, 비디오 데이터 블록에 대해, 각각의 화소 값과 각각의 화소 값에 유사한 화소 값들을 갖는 화소들의 각각의 양을 각각 포함하는 복수의 엔트리들을 포함하는 히스토그램을 결정하는 단계; 그 히스토그램에 기초하여, 메이저 컬러 화소들 또는 이스케이프 화소들로서 비디오 데이터 블록의 하나 이상의 화소들을 분류하는 단계로서, 이스케이프 화소들로서 분류된 화소들은 메이저 컬러 화소들로서 분류된 화소들보다는 더 낮은 양의 화소들을 나타내는 히스토그램에서의 엔트리들에 대응하는, 상기 하나 이상의 화소들을 분류하는 단계; 및 히스토그램에 기초하여 팔레트를 결정하는 단계를 더 포함하는, 비디오 데이터 코딩 방법.
- [0224] 실시예 9. 실시예 1 내지 실시예 8 중 어느 하나의 실시예에 있어서, 히스토그램의 복수의 엔트리들 중 각각의 엔트리는 양자화되는 경우 동일한 화소 값을 갖는 화소들의 양을 나타내는, 비디오 데이터 코딩 방법.
- [0225] 실시예 10. 실시예 1 내지 실시예 9 중 어느 하나의 실시예에 있어서, 팔레트를 결정하는 단계는, 비디오 데이터 블록에 대한 팔레트의 제 1 엔트리를 결정하는 단계로서, 제 1 엔트리는 히스토그램의 제 1 엔트리에 대응하는, 상기 팔레트의 제 1 엔트리를 결정하는 단계; 히스토그램의 제 2 엔트리에 포함된 화소 값과 팔레트에 포함된 하나 이상의 화소 값들 간의 차이에 대응하는 거리 값을 결정하는 단계; 및 거리 값이 임계값을 충족시킨다는 결정에 응답하여, 히스토그램의 제 2 엔트리에 포함된 화소 값을 팔레트의 제 2 엔트리 속으로 복사하는 단계를 포함하는, 비디오 데이터 코딩 방법.
- [0226] 실시예 11. 실시예 1 내지 실시예 10 중 어느 하나의 실시예에 있어서, 임계값은 비디오 데이터 블록의 양자화 파라미터 (QP) 에 기초하는, 비디오 데이터 코딩 방법.
- [0227] 실시예 12. 실시예 1 내지 실시예 11 중 어느 하나의 실시예에 있어서, 비디오 코더는 비디오 디코더이며, 인덱스 값들을 코딩하는 단계는 인덱스 값들을 디코딩하는 단계를 포함하며, 그 방법은, 팔레트와 인덱스 값들을 표현하는 하나 이상의 선택스 엘리먼트들을 포함하는 비트스트림을 수신하는 단계를 더 포함하는, 비디오 데이터 코딩 방법.
- [0228] 실시예 13. 비디오 데이터를 코딩하는 장치로서, 비디오 데이터 블록을 저장하도록 구성되는 메모리; 및 실시예 1 내지 실시예 12의 임의의 조합의 방법을 수행하도록 구성되는 하나 이상의 프로세서들을 포함하는, 비디오 데이터 코딩 장치.
- [0229] 실시예 14. 비디오 데이터를 코딩하는 장치로서, 실시예 1 내지 실시예 12의 임의의 조합의 방법을 수행하는 수단을 포함하는, 비디오 데이터 코딩 장치.
- [0230] 실시예 15. 명령들을 저장하고 있는 컴퓨터 판독가능 매체로서, 상기 명령들은, 실행되는 경우, 비디오 코더의 하나 이상의 프로세서들로 하여금, 실시예 1 내지 실시예 12의 임의의 조합의 방법을 수행하게 하는, 컴퓨터 판독가능 매체.
- [0231] 예에 의거하여, 상이한 시퀀스로 수행될 수도 있는 본원에서 설명된 기법들 중 임의의 기법의 특정 액트들 또는 이벤트들이 부가되거나, 병합되거나, 또는 다 함께 제외될 수도 있다 (예컨대, 모든 설명된 액트들 또는 이벤트들이 그 기법들의 실용화에 필요한 것은 아니다) 는 것이 이해되어야 한다. 더구나, 특정 예들에서, 액트들 또는 이벤트들은 순차적으로라기 보다는, 예컨대, 다중 스레드식 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통하여 동시에 수행될 수도 있다. 덧붙여서, 본 개시물의 특정한 양태들이 명료함을 위해 단일 모듈 또는 유닛에 의해 수행되는 것으로 설명되어 있지만, 본 개시물의 기법들은 비디오 코더에 연관된 유닛들 또는 모듈들의 조합에 의해 수행될 수도 있다는 것이 이해되어야 한다.
- [0232] 본 개시물의 특정 양태들은 예시의 목적으로 HEVC 표준에 대해 설명되었다. 그러나, 본 개시물에서 설명된 기법들은, 아직 개발되지 않은 다른 표준 또는 사유의 비디오 코딩 프로세스들을 포함하는 다른 비디오 코딩 프로세스들에 대해 유용할 수도 있다.
- [0233] 위에서 설명된 기법들은 비디오 인코더 (20) (도 1 및 도 2) 및/또는 비디오 디코더 (30) (도 1 및 도 3) 에 의해 수행될 수도 있으며, 그것들의 양쪽 모두는 비디오 코더라고 일반적으로 지칭될 수도 있다. 비슷하게, 비디오 코딩은 해당되는 경우 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.
- [0234] 그 기법들의 다양한 양태들의 특정 조합들이 위에서 설명되었지만, 이들 조합들은 본 개시물에서 설명된 기법들의 예들을 도시하기 위해서만 제공된다. 따라서, 본 개시물의 기법들은 이들 예의 조합들로 제한되지 않아

야 하고 본 개시물에서 설명된 기법들의 다양한 양태들의 임의의 상상 가능한 조합을 포괄할 수도 있다.

[0235] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 판독가능 저장 매체들을 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 비일시적인 유형의 컴퓨터 판독가능 저장 매체들에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0236] 비제한적인 예로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비일시적, 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc) 는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들으로써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

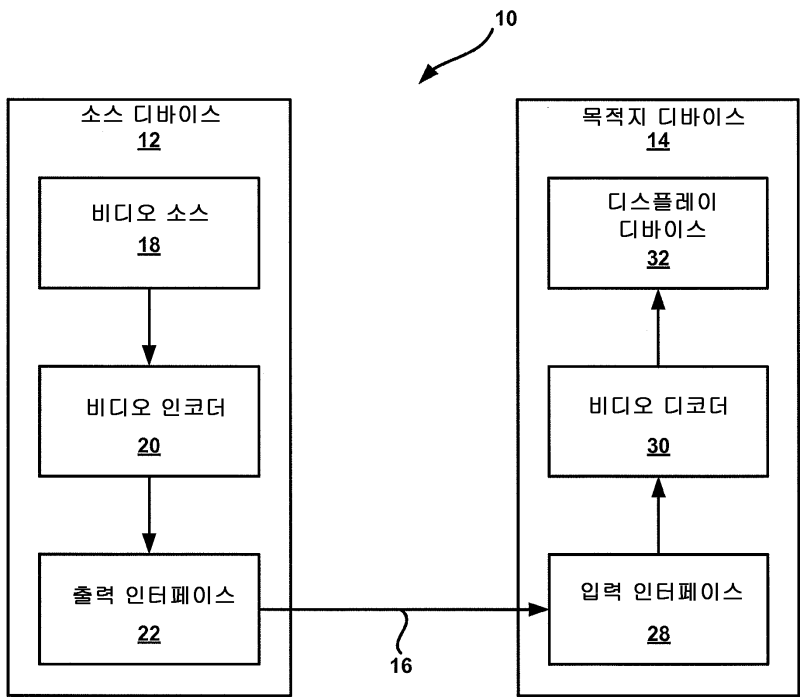
[0237] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 "프로세서"라는 용어는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 나타낼 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0238] 본 개시물의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋) 를 포함하는 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들이 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

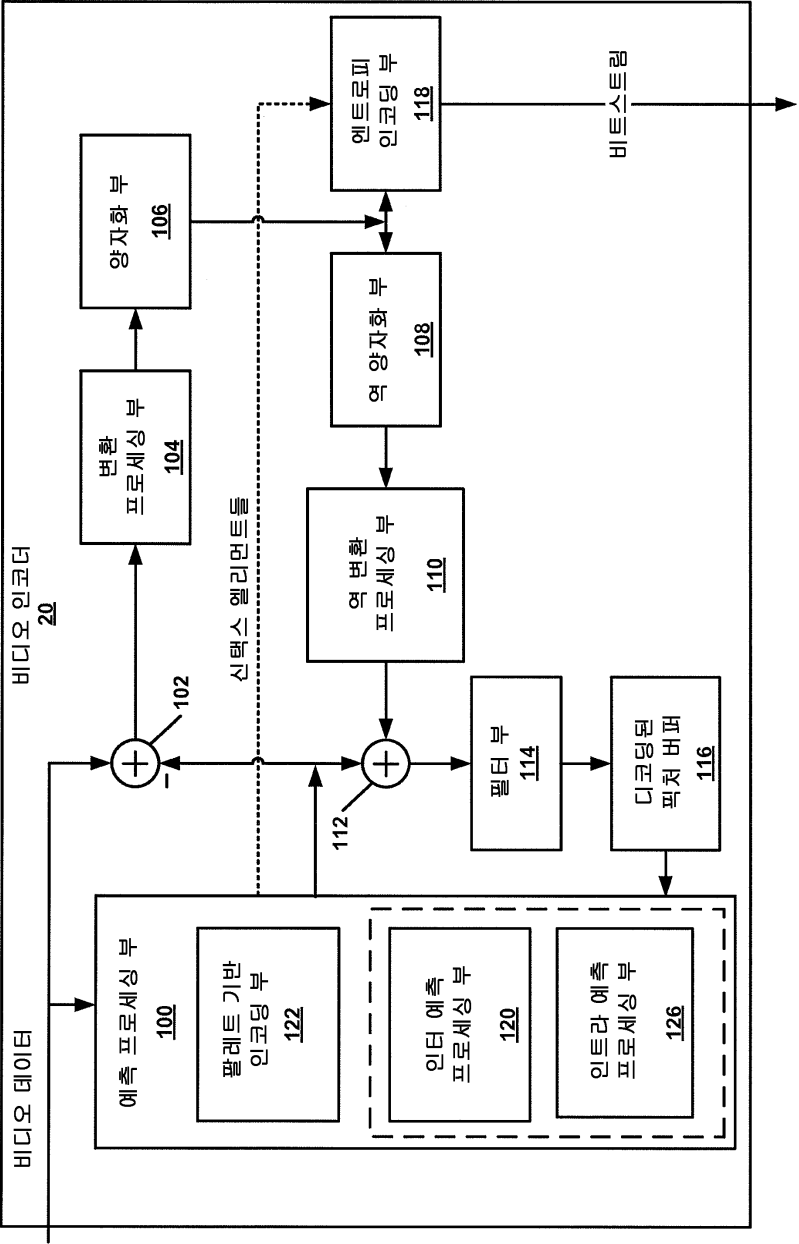
[0239] 다양한 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

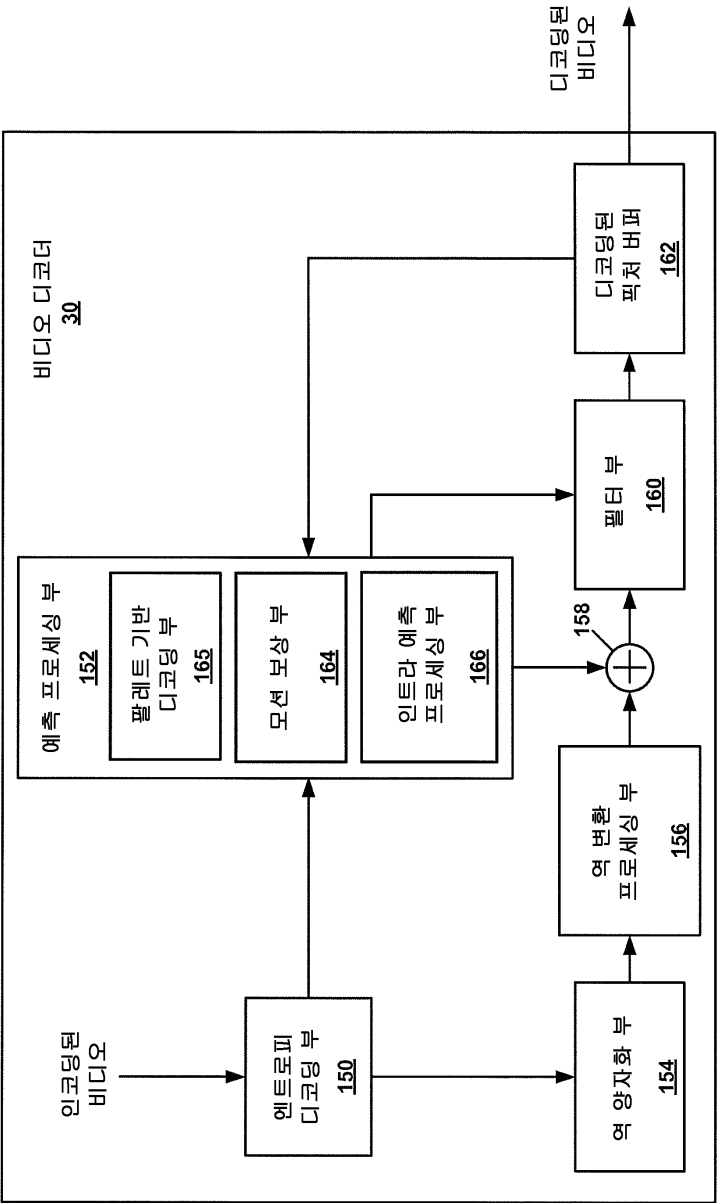
도면1



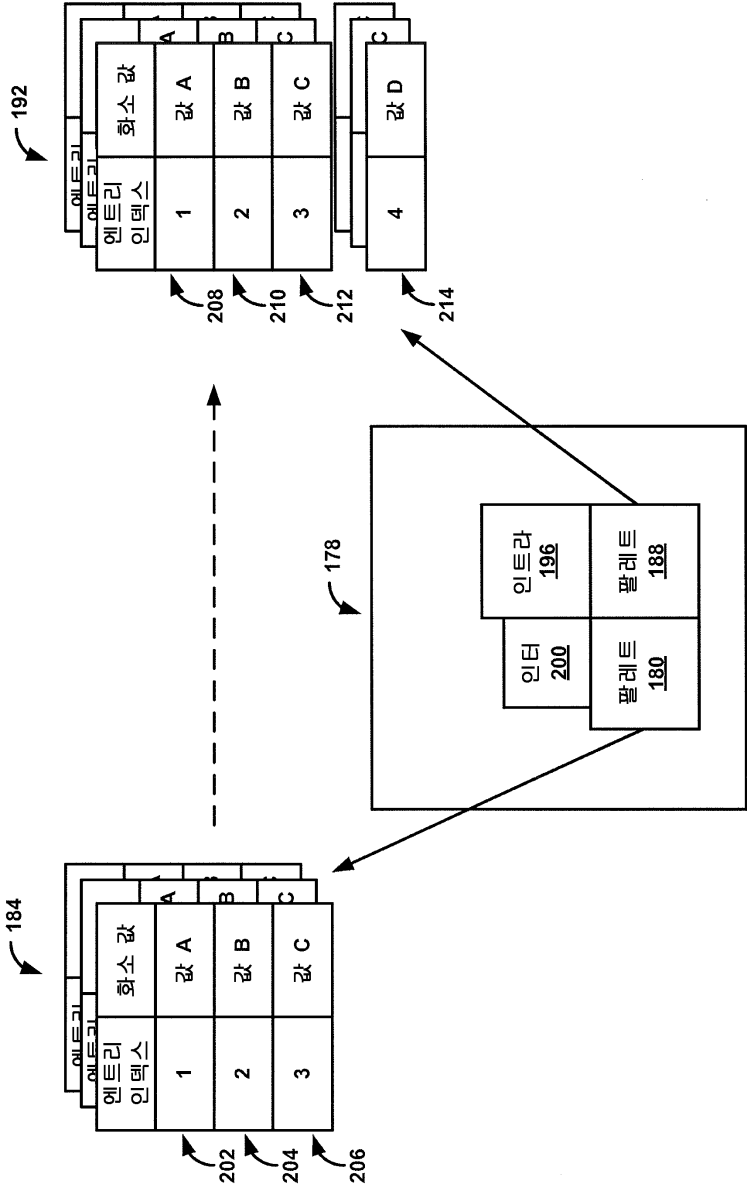
도면2



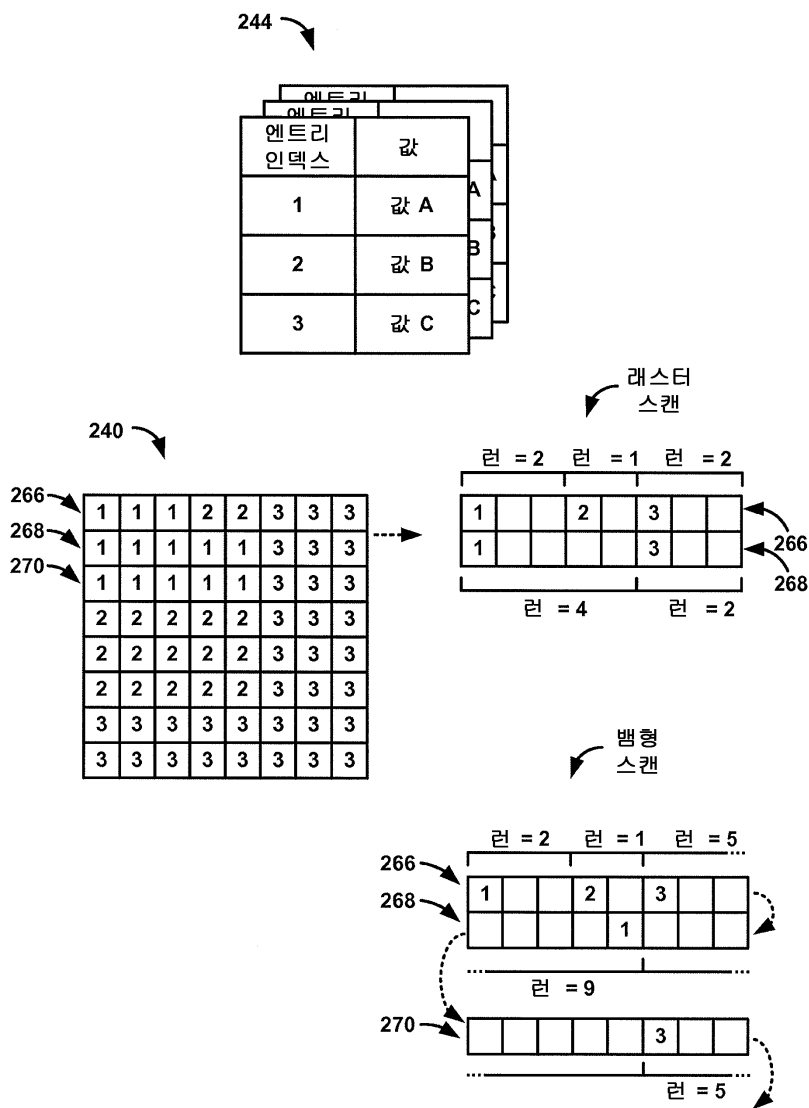
도면3



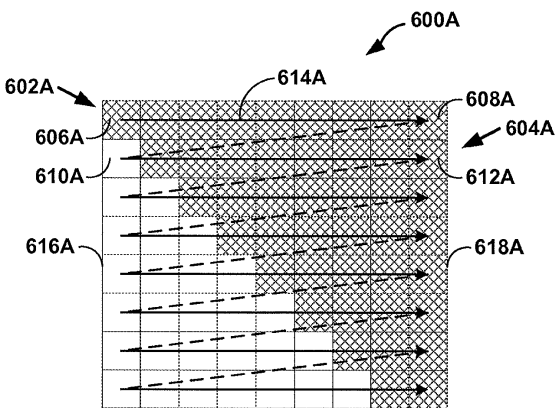
도면4



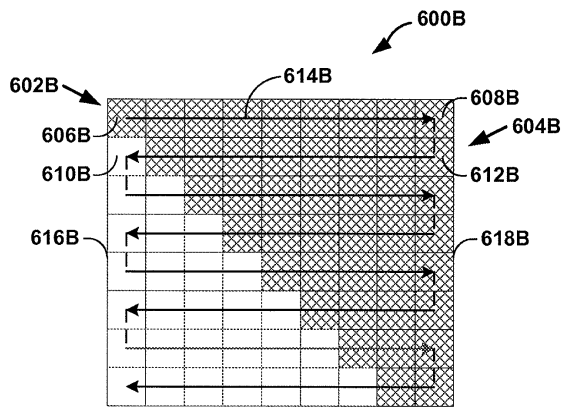
도면5



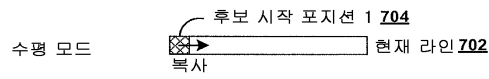
도면6a



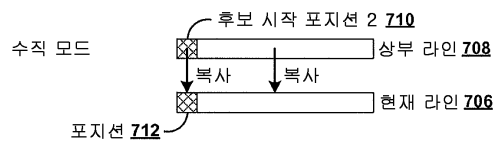
도면6b



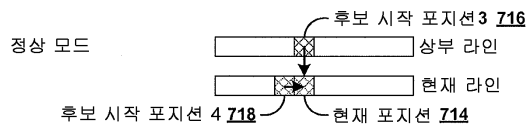
도면7a



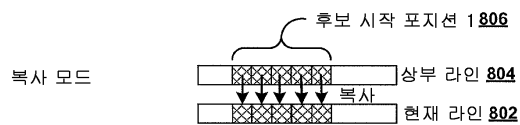
도면7b



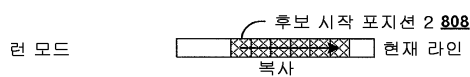
도면7c



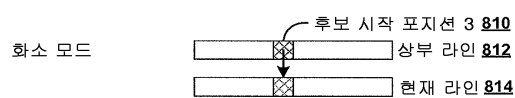
도면8a



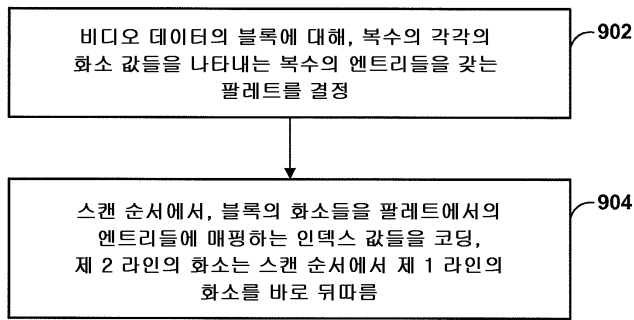
도면8b



도면8c



도면9



도면10

