



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년03월14일
(11) 등록번호 10-1958536
(24) 등록일자 2019년03월08일

- (51) 국제특허분류(Int. Cl.)
H04N 19/93 (2014.01) H04N 19/115 (2014.01)
H04N 19/129 (2014.01) H04N 19/13 (2014.01)
H04N 19/176 (2014.01) H04N 19/186 (2014.01)
H04N 19/463 (2014.01) H04N 19/70 (2014.01)
H04N 19/91 (2014.01)
(52) CPC특허분류
H04N 19/93 (2015.01)
H04N 19/115 (2015.01)
(21) 출원번호 10-2016-7035729
(22) 출원일자(국제) 2015년05월22일
심사청구일자 2018년07월20일
(85) 번역문제출일자 2016년12월20일
(65) 공개번호 10-2017-0007464
(43) 공개일자 2017년01월18일
(86) 국제출원번호 PCT/US2015/032256
(87) 국제공개번호 WO 2015/179810
국제공개일자 2015년11월26일

- (73) 특허권자
켈컴 인코퍼레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
조쉬 라잔 렉스맨
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
세레진 바딤
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
(뒷면에 계속)
특허법인코리어나

(30) 우선권주장
62/002,054 2014년05월22일 미국(US)
(뒷면에 계속)

(56) 선행기술조사문헌
Jianqing Zhu, et al., 'AHG10: Modified copy above mode for palette based coding', (JCTVC-Q094), JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 17th Meeting: Valencia, ES, 29. March 2014
(뒷면에 계속)

전체 청구항 수 : 총 12 항

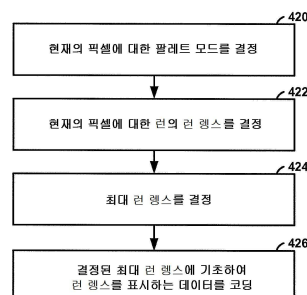
심사관 : 조우연

(54) 발명의 명칭 팔레트-기반 비디오 코딩에서 런들을 코딩

(57) 요약

일 예에서, 비디오 데이터를 코딩하는 방법은, 픽셀의 값을 픽셀을 코딩하기 위하여 이용된 컬러들의 팔레트에서의 컬러 값에 관련시키는 팔레트 인덱스와 연관된 픽셀에 대하여, 픽셀의 팔레트 인덱스로 코딩되는 팔레트 인덱스들의 런의 런 령스를 결정하는 단계를 포함하고, 그 방법은 또한, 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔
(뒷면에 계속)

대표도 - 도11



레트 인덱스들의 최대 런을 위한 최대 런 랭스를 결정하는 단계, 및 결정된 최대 런 랭스에 기초하여 런 랭스를 표시하는 데이터를 코딩하는 단계를 포함한다.

(52) CPC특허분류

H04N 19/129 (2015.01)
H04N 19/13 (2015.01)
H04N 19/176 (2015.01)
H04N 19/186 (2015.01)
H04N 19/463 (2015.01)
H04N 19/70 (2015.01)
H04N 19/91 (2015.01)

(72) 발명자

푸 웨이

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

솔레 로할스 호엘

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

라파카 크리쉬나칸트

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(56) 선행기술조사문헌

Wei Pu, et al., 'AHG10: Suggested Software for Palette Coding based on RExt6.0', (JCTVC-Q0174), JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 17th Meeting: Valencia, ES, 18. March 2014

DÖne Bugdayci, 'AHG10: AHG10: Improvements on palette coding', (JCTVC-Q0071), JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 17th Meeting: Valencia, ES, 28. March 2014

Jianle Chen, et al., 'Description of screen content coding technology proposal by Qualcomm', (JCTVC-Q031), JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 17th Meeting: Valencia, ES, 28. March 20*

Guillaume Laroche, et al., 'AHG10: Run coding for palette mode', (JCTVC-Q0066), JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 17th Meeting: Valencia, ES, 17. March 2014

*는 심사관에 의하여 인용된 문헌

(30) 우선권주장

62/010,313	2014년06월10일	미국(US)
62/015,240	2014년06월20일	미국(US)
62/031,766	2014년07월31일	미국(US)
62/040,978	2014년08월22일	미국(US)
62/114,533	2015년02월10일	미국(US)
62/115,099	2015년02월11일	미국(US)
14/719,222	2015년05월21일	미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

팔레트 인덱스와 연관된 비디오 데이터의 블록의 픽셀로서, 상기 팔레트 인덱스는 상기 픽셀의 값을 상기 픽셀을 코딩하기 위하여 이용된 컬러들의 팔레트에서의 컬러 값에 관련시키는, 상기 비디오 데이터의 블록의 상기 픽셀에 대하여, 상기 픽셀의 상기 팔레트 인덱스로 코딩되는 팔레트 인덱스들의 런의 런 렱스를 결정하는 단계를 포함하고,

상기 비디오 데이터의 블록에서의 픽셀들의 수를 결정하는 단계;

상기 블록의 픽셀들의 팔레트 인덱스들을 스캔하는데 이용된 스캔 순서에 기초하여 상기 비디오 데이터의 블록에서의 상기 픽셀의 위치를 결정하는 단계;

상기 블록의 픽셀들의 수 마이너스 결정된 상기 픽셀의 위치 마이너스 1 에 기초하여, 상기 픽셀의 상기 팔레트 인덱스로 코딩될 수 있는 상기 블록의 픽셀들의 팔레트 인덱스들의 최대 런을 위한 최대 런 렱스를 결정하는 단계; 및

결정된 상기 최대 런 렱스에 기초하여 상기 런 렱스를 표시하는 데이터를 코딩하는 단계

를 특징으로 하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 런 렱스를 표시하는 데이터를 코딩하는 단계는 k번째 차수의 절단된 지수-골롬 (TEGk) 코드를 이용하여 상기 데이터를 코딩하는 단계를 포함하고,

상기 TEGk 코드는 1 진 프리픽스 및 2 진 서픽스를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 TEGk 코드를 이용하여 상기 데이터를 코딩하는 단계는, 상기 1 진 프리픽스의 트레일링 2 진 1 의 값이

$\left\lfloor \log_2 \left(\frac{x_{max}}{2^k} + 1 \right) \right\rfloor == l(x)$ 일 경우에 절단되도록, $l(x) = \left\lfloor \log_2 \left(\frac{x}{2^k} + 1 \right) \right\rfloor$ 에 대응하는 절단된 1 진

코드에 기초하여 상기 1 진 프리픽스를 결정하는 단계를 포함하고, 상기 x 는 팔레트 인덱스들의 런의 런 렱스의 값이고, 상기 x_{max} 는 상기 최대 런 렱스인, 비디오 데이터를 코딩하는 방법.

청구항 4

제 3 항에 있어서,

절단되는 상기 1 진 프리픽스에 기초하여, 상기 2 진 서픽스를 $x - 2^k(2^{l(x)} - 1)$ 의 절단된 2 진 표현으로서 결정하는 단계를 더 포함하며,

상기 절단된 2 진 표현의 최대 값은 $x_{max} - 2^k(2^{l(x)} - 1)$ 인, 비디오 데이터를 코딩하는 방법.

청구항 5

제 3 항에 있어서,

절단되지 않은 상기 1 진 프리픽스에 기초하여, 상기 2 진 서픽스를 $k + l(x)$ 비트들을 이용한 $x - 2^k(2^{l(x)} - 1)$ 의 2 진 표현으로서 결정하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 6

제 2 항에 있어서,

k번째 차수의 절단된 지수-골롭 코드를 이용하여 상기 데이터를 코딩하는 단계는, 제 2 차수의 절단된 지수-골롭 코드를 이용하여 상기 데이터를 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 7

제 2 항에 있어서,

상기 런 렱스를 표시하는 데이터를 코딩하는 단계는, 상기 런 렱스에 대하여 0 초과와 플래그를 코딩하는 단계, 상기 런 렱스에 대하여 1 초과와 플래그를 코딩하는 단계, 상기 런 렱스에 대하여 2 초과와 플래그를 코딩하는 단계, 및 상기 TEGk 코드를 이용하여 상기 런 렱스에 대하여 나머지 빈들을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 8

제 2 항에 있어서,

상기 런 렱스를 표시하는 데이터를 코딩하는 단계는, 상기 런 렱스에 대하여 0 초과와 플래그를 코딩하는 단계 및 상기 최대 런 렱스 마이너스 1 과 동일하게 설정된 TEGk 코드에 대한 최대 값을 갖는 TEGk 코드를 이용하여 상기 런 렱스 마이너스 1 을 표시하는 데이터를 코딩하는 단계를 포함하며,

상기 k 는 상기 TEGk 코드가 TEG0 코드를 포함하도록 0 과 동일한, 비디오 데이터를 코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 런 렱스를 표시하는 데이터를 코딩하는 단계는 상기 데이터를 인코딩하는 단계를 포함하고,

상기 방법은,

상기 컬러들의 팔레트를 표시하는 데이터를 인코딩하는 단계; 및

상기 팔레트 인덱스와 연관된 상기 픽셀에 대하여, 상기 픽셀을 인코딩하는데 이용된 팔레트 모드를 표시하는 데이터를 인코딩하는 단계

를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 10

제 1 항에 있어서,

상기 런 렱스를 표시하는 데이터를 코딩하는 단계는 상기 데이터를 디코딩하는 단계를 포함하고,

상기 방법은,

인코딩된 비트스트림으로부터, 상기 컬러들의 팔레트를 표시하는 데이터를 획득하는 단계;

상기 팔레트 인덱스와 연관된 상기 픽셀에 대하여, 상기 픽셀을 인코딩하는데 이용된 팔레트 모드를 표시하는 데이터를 획득하는 단계; 및

상기 팔레트 모드 및 상기 컬러들의 팔레트를 이용하여 상기 픽셀의 값을 결정하는 단계

를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 11

비디오 데이터를 프로세싱하기 위한 장치로서,

팔레트 인덱스와 연관된 비디오 데이터의 블록의 픽셀로서, 상기 팔레트 인덱스는 상기 픽셀의 값을 상기 픽셀을 코딩하기 위하여 이용된 컬러들의 팔레트에서의 컬러 값에 관련시키는, 상기 비디오 데이터의 블록의 상기 픽셀에 대하여, 상기 픽셀의 상기 팔레트 인덱스로 코딩되는 팔레트 인덱스들의 런의 런 렱스를 결정하는 수단을 포함하고,

상기 비디오 데이터의 블록에서의 픽셀들의 수를 결정하는 수단;

상기 블록의 픽셀들의 팔레트 인덱스들을 스캔하는데 이용된 스캔 순서에 기초하여 상기 비디오 데이터의 블록에서의 상기 픽셀의 위치를 결정하는 수단;

상기 블록의 픽셀들의 수 마이너스 결정된 상기 픽셀의 위치 마이너스 1 에 기초하여, 상기 픽셀의 상기 팔레트 인덱스로 코딩될 수 있는 상기 블록의 픽셀들의 팔레트 인덱스들의 최대 런을 위한 최대 런 렱스를 결정하는 수단; 및

결정된 상기 최대 런 렱스에 기초하여 상기 런 렱스를 표시하는 데이터를 코딩하는 수단

을 특징으로 하는, 비디오 데이터를 프로세싱하기 위한 장치.

청구항 12

명령들을 저장한 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 명령들은, 실행될 경우, 하나 이상의 프로세서들로 하여금, 제 1 행 내지 제 10 행 중 어느 한 행에 기재된 방법을 수행하게 하는, 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

발명의 설명

기술 분야

[0001] 이 출원은 2014 년 5 월 22 일자로 출원된 미국 가출원 제 62/002,054 호, 2014 년 6 월 10 일자로 출원된 미국 가출원 제 62/010,313 호, 2014 년 6 월 20 일자로 출원된 미국 가출원 제 62/015,240 호, 2014 년 7 월 31 일자로 출원된 미국 가출원 제 62/031,766 호, 2014 년 8 월 22 일자로 출원된 미국 가출원 제 62/040,978 호, 2015 년 2 월 10 일자로 출원된 미국 가출원 제 62/114,533 호, 및 2015 년 2 월 11 일자로 출원된 미국 가출원 제 62/114,533 호의 이익을 주장하고, 이들 각각의 전체 내용들은 본원에 참조로 편입된다.

[0002] 이 개시물은 비디오 인코딩 및 디코딩에 관한 것이다.

배경 기술

[0003] 디지털 비디오 기능들은, 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말 (personal digital assistant; PDA) 들, 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 및 기타 등등을 포함하는 광범위한 디바이스들 내에 편입될 수 있다.

디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, 진보된 비디오 코딩 (Advanced Video Coding; AVC) 에 의해 정의된 표준들, 현재 개발 중에 있는 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준, 및 이러한 표준들의 확장들에서 설명된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신하고, 수신하고, 인코딩하고, 디코딩하고, 및/또는 저장할 수도 있다.

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

[0005] 공간적 또는 시간적 예측은 코딩되어야 할 블록에 대한 예측 블록으로 귀착된다. 잔차 데이터 (residual data) 는 코딩되어야 할 원래의 블록과 예측 블록과의 사이의 픽셀 차이들을 표현한다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 지시하는 모션 벡터 (motion vector) 에 따라 인코딩되고, 잔차 데이터는 코딩된 블록 및 예측 블록 사이의 차이를 표시한다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔

차 데이터에 따라 인코딩된다. 추가의 압축을 위하여, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어 잔차 계수들로 귀착될 수도 있고, 이 잔차 계수들은 그 후에 양자화될 수도 있다. 초기에 2 차원 어레이로 배열된 양자화된 계수들은 계수들의 1 차원 벡터를 생성하기 위하여 스캐닝될 수도 있고, 훨씬 더 많은 압축을 달성하기 위하여 엔트로피 코딩 (entropy coding) 이 적용될 수도 있다.

발명의 내용

- [0006] 이 개시물의 기법들은 팔레트-기반 비디오 코딩에 관한 것이다. 예를 들어, 팔레트 기반 코딩에서, 비디오 코더 (비디오 인코더 또는 비디오 디코더) 는 특정한 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 표현하기 위한 컬러들의 테이블로서 "팔레트 (palette)" 를 형성할 수도 있다. 팔레트-기반 코딩은 상대적으로 작은 수의 컬러들을 가지는 비디오 데이터의 영역들을 코딩하기 위하여 특히 유용할 수도 있다. 실제적인 픽셀 값들 (또는 그들의 잔차들) 을 코딩하는 것이 아니라, 비디오 코더는 픽셀들을, 픽셀들의 컬러들을 표현하는 팔레트에서의 엔트리 (entry) 들과 관련시키는 픽셀들 중의 하나 이상에 대한 팔레트 인덱스들을 코딩할 수도 있다. 이 개시물에서 설명된 기법들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 유도하는 것, 및 팔레트-기반 코딩 맵들 및 다른 신택스 엘리먼트들을 송신하는 것 중의 하나 이상의 다양한 조합들을 위한 기법들을 포함할 수도 있다.
- [0007] 예에서, 비디오 데이터를 프로세싱하는 방법은, 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 개개의 샘플이 비디오 데이터의 블록을 코딩하기 위한 컬러들의 팔레트 내에 포함되지 않은 적어도 하나의 개개의 샘플의 컬러 값에 기초하여 코딩되는지 여부를 표시하는 블록-레벨 신택스 엘리먼트의 값을 결정하는 단계, 및 값에 기초하여 비디오 데이터의 블록을 코딩하는 단계를 포함한다.
- [0008] 또 다른 예에서, 비디오 데이터를 프로세싱하기 위한 디바이스는, 비디오 데이터의 샘플들의 블록을 저장하도록 구성된 메모리, 및 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 개개의 샘플이 비디오 데이터의 블록을 코딩하기 위한 컬러들의 팔레트 내에 포함되지 않은 적어도 하나의 개개의 샘플의 컬러 값에 기초하여 코딩되는지 여부를 표시하는 블록-레벨 신택스 엘리먼트의 값을 결정하고, 그리고 값에 기초하여 비디오 데이터의 블록을 코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.
- [0009] 또 다른 예에서, 비디오 데이터를 프로세싱하기 위한 장치는, 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 개개의 샘플이 비디오 데이터의 블록을 코딩하기 위한 컬러들의 팔레트 내에 포함되지 않은 적어도 하나의 개개의 샘플의 컬러 값에 기초하여 코딩되는지 여부를 표시하는 블록-레벨 신택스 엘리먼트의 값을 결정하기 위한 수단, 및 값에 기초하여 비디오 데이터의 블록을 코딩하기 위한 수단을 포함한다.
- [0010] 또 다른 예에서, 비-일시적 컴퓨터-판독가능 매체는 명령들을 저장하고, 상기 명령들은, 실행될 경우, 하나 이상의 프로세서들로 하여금, 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 개개의 샘플이 비디오 데이터의 블록을 코딩하기 위한 컬러들의 팔레트 내에 포함되지 않은 적어도 하나의 개개의 샘플의 컬러 값에 기초하여 코딩되는지 여부를 표시하는 블록-레벨 신택스 엘리먼트의 값을 결정하게 하고, 그리고 값에 기초하여 비디오 데이터의 블록을 코딩하게 한다.
- [0011] 또 다른 예에서, 비디오 데이터를 프로세싱하는 방법은, 비디오 데이터의 블록을 코딩하기 위한 컬러 값들의 팔레트의 최대 팔레트 크기를 표시하는 데이터, 또는 컬러 값들의 팔레트를 결정하기 위한 팔레트 예측자 (palette predictor) 의 최대 팔레트 예측자 크기를 표시하는 데이터 중의 적어도 하나를 코딩하는 단계, 및 데이터에 따라 비디오 데이터의 블록을 코딩하는 단계를 포함한다.
- [0012] 또 다른 예에서, 비디오 데이터를 프로세싱하기 위한 디바이스는, 비디오 데이터의 블록을 저장하도록 구성된 메모리, 및 비디오 데이터의 블록을 코딩하기 위한 컬러 값들의 팔레트의 최대 팔레트 크기를 표시하는 데이터, 또는 컬러 값들의 팔레트를 결정하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 표시하는 데이터 중의 적어도 하나를 코딩하고, 그리고 비트스트림으로부터 코딩된 데이터에 따라 비디오 데이터의 블록을 코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.
- [0013] 또 다른 예에서, 비디오 데이터를 프로세싱하기 위한 장치는, 비디오 데이터의 블록을 코딩하기 위한 컬러 값들의 팔레트의 최대 팔레트 크기를 표시하는 데이터, 또는 컬러 값들의 팔레트를 결정하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 표시하는 데이터 중의 적어도 하나를 코딩하기 위한 수단, 및 데이터에 따라 비디오 데이터의 블록을 코딩하기 위한 수단을 포함한다.
- [0014] 또 다른 예에서, 비-일시적 컴퓨터-판독가능 매체는 명령들을 저장하고, 상기 명령들은, 실행될 경우, 하나 이

상의 프로세서들로 하여금, 비디오 데이터의 블록을 코딩하기 위한 컬러 값들의 팔레트의 최대 팔레트 크기를 표시하는 데이터, 또는 컬러 값들의 팔레트를 결정하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 표시하는 데이터 중의 적어도 하나를 코딩하게 하고, 그리고 데이터에 따라 비디오 데이터의 블록을 코딩하게 한다.

[0015] 또 다른 예에서, 비디오 데이터를 코딩하는 방법은, 픽셀의 값을, 픽셀을 코딩하기 위하여 이용된 컬러들의 팔레트에서의 컬러 값에 관련시키는 팔레트 인덱스와 연관된 픽셀에 대하여, 픽셀의 팔레트 인덱스로 코딩되는 팔레트 인덱스들의 런의 런 령스를 결정하는 단계, 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정하는 단계, 및 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 코딩하는 단계를 포함한다.

[0016] 또 다른 예에서, 비디오 데이터를 코딩하기 위한 디바이스는, 픽셀의 값을, 픽셀을 코딩하기 위하여 이용된 컬러들의 팔레트에서의 컬러 값에 관련시키는 팔레트 인덱스와 연관된 비디오 데이터의 픽셀을 저장하도록 구성된 메모리, 및 픽셀에 대하여, 픽셀의 팔레트 인덱스로 코딩되는 팔레트 인덱스들의 런의 런 령스를 결정하고, 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정하고, 그리고 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0017] 또 다른 예에서, 비디오 데이터를 프로세싱하기 위한 장치는, 픽셀의 값을, 픽셀을 코딩하기 위하여 이용된 컬러들의 팔레트에서의 컬러 값에 관련시키는 팔레트 인덱스와 연관된 픽셀에 대하여, 픽셀의 팔레트 인덱스로 코딩되는 팔레트 인덱스들의 런의 런 령스를 결정하기 위한 수단, 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정하기 위한 수단, 및 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 코딩하기 위한 수단을 포함한다.

[0018] 또 다른 예에서, 비-일시적 컴퓨터-판독가능 매체는 명령들을 저장하고, 상기 명령들은, 실행될 경우, 하나 이상의 프로세서들로 하여금, 픽셀의 값을, 픽셀을 코딩하기 위하여 이용된 컬러들의 팔레트에서의 컬러 값에 관련시키는 팔레트 인덱스와 연관된 픽셀에 대하여, 픽셀의 팔레트 인덱스로 코딩되는 팔레트 인덱스들의 런의 런 령스를 결정하게 하고, 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정하게 하고, 그리고 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 코딩하게 한다.

[0019] 개시물의 하나 이상의 예들의 세부 사항들은 동반된 도면들 및 이하의 설명에서 기재된다. 다른 특징들, 목적들, 및 장점들은 설명, 도면들, 및 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0020] 도 1 은 이 개시물에서 설명된 기법들을 사용할 수도 있는 일 예의 비디오 코딩 시스템을 예시하는 블록도이다.

도 2 는 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 예시하는 블록도이다.

도 3 은 이 개시물에서 설명된 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 예시하는 블록도이다.

도 4 는 이 개시물의 기법들과 부합하는, 팔레트-기반 비디오 코딩을 위한 팔레트 엔트리들을 결정하는 예를 예시하는 개념도이다.

도 5 는 이 개시물의 기법들과 부합하는, 픽셀들의 블록에 대한 팔레트로의 팔레트 인덱스들을 결정하는 예를 예시하는 개념도이다.

도 6 은 이 개시물의 기법들과 부합하는, 픽셀들의 블록에 대한 최대 런 령스를 결정하는 예를 예시하는 개념도이다.

도 7 은 이 개시물의 기법들과 부합하는, 블록의 임의의 샘플이 탈출 샘플들로서 인코딩되는지 여부를 표시하는 하나 이상의 블록-레벨 신택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 인코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다.

도 8 은 이 개시물의 기법들과 부합하는, 블록의 임의의 샘플이 탈출 샘플로서 디코딩되는지 여부를 표시하는 하나 이상의 블록-레벨 신택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 디코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다.

도 9 는 이 개시물의 기법들과 부합하는, 최대 팔레트 크기 및 최대 팔레트 예측자 크기를 표시하는 하나 이상의 신택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 인코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다.

우차트이다.

도 10 은 이 개시물의 기법들과 부합하는, 최대 팔레트 크기 및 최대 팔레트 예측자 크기를 표시하는 하나 이상의 선택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 인코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다.

도 11 은 이 개시물의 기법들과 부합하는, 최대 잠재적 런 렱스에 기초하여 픽셀들의 런의 런 렱스를 표시하는 데이터를 코딩 (인코딩 또는 디코딩) 하기 위한 일 예의 프로세스를 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

- [0021] 이 개시물의 양태들은 비디오 코딩 및 압축을 위한 기법들에 관한 것이다. 특히, 이 개시물은 비디오 데이터의 팔레트-기반 코딩을 위한 기법들을 설명한다. 전통적인 비디오 코딩에서는, 이미지들이 연속-톤 (continuous-tone) 이고 공간적으로 평활 (smooth) 한 것으로 가정된다. 이 가정들에 기초하여, 블록-기반 변환, 필터링 등과 같은 다양한 툴들이 개발되었고, 이러한 툴들은 자연적 콘텐츠 비디오에 대하여 양호한 성능을 보여주었다.
- [0022] 그러나, 원격 데스크톱, 협력 작업, 및 무선 디스플레이와 같은 애플리케이션들에서는, 컴퓨터 생성된 스크린 콘텐츠가 압축되어야 할 지배적인 콘텐츠일 수도 있다. 이 타입의 콘텐츠는 이산-톤 (discrete-tone) 을 가지며, 예리한 라인들 및 높은 콘트라스트의 객체 경계들을 특징으로 하는 경향이 있다. 연속-톤 및 평활화의 가정은 더 이상 적용되지 않을 수도 있고, 이에 따라, 전통적인 비디오 코딩 기법들은 콘텐츠를 압축하기 위한 비효율적인 방법들일 수도 있다.
- [0023] 이 개시물은 특히, 스크린 생성된 콘텐츠 코딩 (예컨대, 스크린 콘텐츠 코딩 (screen content coding; SCC) 을 위하여 적당할 수도 있는 팔레트-기반 코딩을 설명한다. 비디오 데이터의 팔레트-기반 코딩을 위한 기법들은 인터-예측 또는 인트라-예측 코딩을 위한 기법들과 같은 하나 이상의 다른 코딩 기법들과 함께 이용될 수도 있다. 예를 들어, 이하에서 더욱 상세하게 설명된 바와 같이, 인코더 또는 디코더, 또는 조합된 인코더-디코더 (코덱) 는 인터-예측 및 인트라-예측 코딩뿐만 아니라, 팔레트-기반 코딩을 수행하도록 구성될 수도 있다.
- [0024] 일부 예들에서, 팔레트-기반 코딩 기법들은 하나 이상의 비디오 코딩 표준들과 함께 이용하기 위하여 구성될 수도 있다. 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 은 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 및 ISO/IEC 모션 픽처 전문가 그룹 (Motion Picture Experts Group; MPEG) 의 비디오 코딩에 관한 합동 협력팀 (Joint Collaboration Team on Video Coding; JCT-VC) 에 의해 개발되고 있는 새로운 비디오 코딩 표준이다. 최근의 HEVC 텍스트 사양 초안은, Bross 등, "High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Consent)", JCVCL1003_v13, ITU-T SG16 WP 3 및 ISO/IEC JCT 1/SC 29/WG 11 의 JCT-VC 의 제 12 차 회의, 2013 년 1 월 14 - 23 일 ("HEVC 초안 10") 에서 설명되어 있다.
- [0025] HEVC 프레임워크에 대하여, 예로서, 팔레트-기반 코딩 기법들은 코딩 유닛 (coding unit; CU) 모드로서 이용되도록 구성될 수도 있다. 다른 예들에서, 팔레트-기반 코딩 기법들은 HEVC 의 프레임워크에서 PU 모드로서 이용되도록 구성될 수도 있다. 따라서, CU 모드의 맥락에서 설명된 다음의 개시된 프로세스들의 전부는 추가적으로 또는 대안적으로, PU 에 적용할 수도 있다. 그러나, 이 HEVC-기반 예들은 본원에서 설명된 팔레트-기반 코딩 기법들의 한정 또는 제한으로 간주되지 않아야 하는데, 이것은 이러한 기법들이 독립적으로, 또는 다른 기존의 또는 아직 개발되어야 할 시스템들/표준들의 일부로서 작동하도록 적용될 수도 있기 때문이다. 이 경우들에는, 팔레트 코딩을 위한 유닛이 정사각형 블록들, 직사각형 블록들, 또는 심지어 비-직사각형 형상의 영역들일 수 있다.
- [0026] 팔레트-기반 코딩에서는, 비디오 데이터의 특정한 영역이 상대적으로 작은 수의 컬러들을 가지는 것으로 가정될 수도 있다. 비디오 코더 (비디오 인코더 또는 비디오 디코더) 는 특정한 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 표현하기 위한 컬러들의 테이블로서, 소위 "팔레트" 를 코딩할 수도 있다. 각각의 픽셀은 픽셀의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 코더는 픽셀 값을 팔레트에서의 적절한 값에 관련시키는 인덱스를 코딩할 수도 있다.
- [0027] 상기 예에서, 비디오 인코더는, 블록에 대한 팔레트를 결정함으로써, 각각의 픽셀의 값을 표현하기 위하여 팔레트에서 엔트리를 위치시킴으로써, 그리고 픽셀 값을 팔레트에 관련시키는 픽셀들에 대한 팔레트 인덱스들 (또는 팔레트 인덱스 값들로서 지칭됨) 로 팔레트를 인코딩함으로써, 비디오 데이터의 블록을 인코딩할 수도 있다.

비디오 디코더는 인코딩된 비트스트림으로부터, 블록에 대한 팔레트뿐만 아니라, 블록의 픽셀들에 대한 팔레트 인덱스들을 획득할 수도 있다. 비디오 디코더는 블록의 픽셀 값들을 재구성하기 위하여 픽셀들의 팔레트 인덱스들을 팔레트의 엔트리들에 관련시킬 수도 있다. 픽셀들 (및/또는 픽셀 값을 표시하는 관련된 팔레트 인덱스들) 은 샘플들로서 일반적으로 지칭될 수도 있다.

[0028] 블록에서의 샘플들은 수평 래스터 스캔 순서 (horizontal raster scanning order) 를 이용하여 프로세싱 (예컨대, 스캐닝됨) 된다. 예를 들어, 비디오 인코더는 수평 래스터 스캔 순서를 이용하여 팔레트 인덱스들을 스캐닝함으로써, 팔레트 인덱스들의 2 차원 블록을 1 차원 어레이로 변환할 수도 있다. 마찬가지로, 비디오 디코더는 수평 래스터 스캔 순서를 이용하여 팔레트 인덱스들의 블록을 재구성할 수도 있다. 따라서, 이 개시물은 이전의 샘플을, 스캔 순서에서 블록에서 현재 코딩되고 있는 샘플을 선행하는 샘플로서 지칭할 수도 있다. 수직 래스터 스캔 순서와 같은, 수평 래스터 스캔 이외의 스캔들이 또한 적용가능할 수도 있다는 것을 인식해야 한다. 상기 예는 팔레트-기반 코딩의 일반적인 설명을 제공하도록 의도된 것이다.

[0029] 팔레트는 전형적으로, 인덱스에 의해 변호부여되고 컬러 컴포넌트 (예를 들어, RGG, YUV 등) 값들 또는 강도 (intensity) 들을 표현하는 엔트리들을 포함한다. 비디오 인코더 및 비디오 디코더의 양자는 팔레트 엔트리들의 수, 각각의 팔레트 엔트리에 대한 컬러 컴포넌트 값들, 및 현재의 블록에 대한 팔레트 엔트리들의 정확한 순서를 결정한다. 이 개시물에서는, 각각의 팔레트 엔트리가 샘플의 모든 컬러 컴포넌트들에 대한 값들을 특징하는 것으로 가정된다. 그러나, 이 개시물의 개념들은 각각의 컬러 컴포넌트에 대하여 별도의 팔레트를 이용하는 것에 적용가능하다.

[0030] 일부 예들에서, 팔레트는 이전에 코딩된 블록들로부터의 정보를 이용하여 구성될 수도 있다. 즉, 팔레트는 이전의 블록 (들) 을 코딩하기 위하여 이용된 팔레트 (들) 로부터 예측되는 예측된 팔레트 엔트리들을 포함할 수도 있다. 예를 들어, 표준 제출 문서 Wei Pu 등, "AHG10: Suggested Software for Palette Coding based on RExt6.0," JCTVC-Q0094, Valencia, ES, 2014 년 3 월 27 일 - 4 월 4 일 (이하, JCTVC-Q0094) 에서 설명된 바와 같이, 팔레트는 예측자 팔레트로부터 복사되는 엔트리들을 포함할 수도 있다. 예측자 팔레트는 팔레트 모드 또는 다른 재구성된 샘플들을 이용하여 이전에 코딩된 블록들로부터의 팔레트 엔트리들을 포함할 수도 있다. 예측자 팔레트에서의 각각의 엔트리에 대하여, 2 진 플래그는 플래그와 연관된 엔트리가 현재의 팔레트로 복사되는지 여부를 표시 (예컨대, 플래그 = 1 에 의해 표시됨) 하기 위하여 복사될 수도 있다. 2 진 플래그들의 스트링 (string) 은 2 진 팔레트 예측 벡터로서 지칭될 수도 있다. 현재의 블록을 코딩하기 위한 팔레트는 또한, (예컨대, 팔레트 예측 벡터와는 별도로) 명시적으로 코딩될 수도 있는 새로운 팔레트 엔트리들의 수를 포함할 수도 있다. 새로운 엔트리들의 수의 표시는 또한 코딩될 수도 있다. 예측된 엔트리들 및 새로운 엔트리들의 합은 블록에 대한 총 팔레트 크기를 표시할 수도 있다.

[0031] JCTVC-Q0094 에서 제안된 바와 같이, 팔레트-기반 코딩 모드로 코딩된 블록에서의 각각의 샘플은 이하에서 기재된 바와 같이, 3 개의 팔레트 모드들 중의 하나를 이용하여 코딩될 수도 있다:

[0032] • 탈출 모드 : 이 모드에서는, 샘플 값이 팔레트 엔트리로서 팔레트 내로 포함되지 않고, 양자화된 샘플 값은 모든 컬러 컴포넌트들에 대하여 명시적으로 시그널링된다. 그것은 새로운 팔레트 엔트리들의 시그널링과 유사하지만, 새로운 팔레트 엔트리들에 대하여, 컬러 컴포넌트 값들은 양자화되지 않는다.

[0033] • CopyFromTop 모드 (또한, CopyAbove 모드로서 지칭됨) : 이 모드에서는, 현재의 샘플에 대한 팔레트 엔트리 인덱스가 블록에서의 바로 상부에 위치한 샘플로부터 복사된다.

[0034] • 값 모드 (또한, 인덱스 모드로서 지칭됨) : 이 모드에서는, 팔레트 엔트리 인덱스의 값이 명시적으로 시그널링된다.

[0035] 본원에서 설명된 바와 같이, 팔레트 엔트리 인덱스는 팔레트 인덱스 또는 간단하게 인덱스로서 지칭될 수도 있다. 이 용어들은 이 개시물의 기법들을 설명하기 위하여 상호 교환가능하게 이용될 수 있다. 게다가, 이하에서 더욱 상세하게 설명된 바와 같이, 팔레트 인덱스는 하나 이상의 연관된 컬러 또는 강도 값들을 가질 수도 있다. 예를 들어, 팔레트 인덱스는 픽셀의 단일 컬러 또는 강도 컴포넌트 (예컨대, RGB 데이터의 적색 컴포넌트, YUV 데이터의 Y 컴포넌트 등) 와 연관된 단일 연관된 컬러 또는 강도 값을 가질 수도 있다. 또 다른 예에서, 팔레트 인덱스는 다수의 연관된 컬러 또는 강도 값들을 가질 수도 있다. 일부 사례들에서, 팔레트-기반 코딩은 단색 비디오 (monochrome video) 를 코딩하기 위하여 적용될 수도 있다. 따라서, "컬러 값" 은 픽셀 값을 생성하기 위하여 이용된 임의의 컬러 또는 비-컬러 (non-color) 컴포넌트를 일반적으로 지칭할 수도 있다.

- [0036] CopyFromTop 및 값 모드들에 대하여, (또한, 간단하게 런으로 지칭될 수도 있는) 런 값이 또한 시그널링될 수도 있다. 런 값은 함께 코딩되는 팔레트-코딩된 블록에서의 특정한 스캔 순서에서의 연속적인 샘플들 (예컨대, 샘플들의 런) 의 수를 표시할 수도 있다. 일부 사례들에서는, 런의 각각의 샘플이 팔레트로의 연관된 인덱스를 가지기 때문에, 샘플들의 런은 또한, 팔레트 인덱스들의 런으로서 지칭될 수도 있다.
- [0037] 런 값은 동일한 팔레트-코딩 모드를 이용하여 코딩되는 팔레트 인덱스들의 런을 표시할 수도 있다. 예를 들어, 값 모드에 대하여, 비디오 코더 (비디오 인코더 또는 비디오 디코더) 는 팔레트 인덱스 (또한, 팔레트 인덱스 값 또는 간단하게 인덱스 값으로서 지칭됨) 와, 동일한 팔레트 인덱스를 가지며 팔레트 인덱스로 코딩되고 있는 스캔 순서에서의 연속적인 샘플들의 수를 표시하는 런 값을 코딩할 수도 있다. CopyFromTop 모드에 대하여, 비디오 코더는 현재의 샘플 값에 대한 인덱스가 상부-이웃 샘플 (예컨대, 블록에서 현재 코딩되고 있는 샘플의 상부에 위치되는 샘플) 의 인덱스에 기초하여 복사된다는 표시와, 상부-이웃 샘플로부터 팔레트 인덱스를 또한 복사하며 팔레트 인덱스로 코딩되고 있는 스캔 순서에서의 연속적인 샘플들의 수를 표시하는 런 값을 코딩할 수도 있다. 따라서, 상기 예들에서, 팔레트 인덱스들의 런은 동일한 값을 가지는 팔레트 인덱스들의 런, 또는 상부-이웃 팔레트 인덱스들로부터 복사되는 팔레트 인덱스들의 런을 지칭한다.
- [0038] 이 때문에, 런은 주어진 모드에 대하여, 동일한 모드에 속하는 후속 샘플들의 수를 특정할 수도 있다. 일부 사례들에서, 인덱스 및 런 값을 시그널링하는 것은 런 랭스 코딩과 유사할 수도 있다. 예시의 목적들을 위한 예에서, 블록의 연속적인 팔레트 인덱스들의 스트링은 0, 2, 2, 2, 2, 5 일 수도 있다 (예컨대, 여기서, 각각의 인덱스는 블록에서의 샘플에 대응함). 이 예에서, 비디오 코더는 값 모드를 이용하여 제 2 샘플 (예컨대, 2 의 제 1 팔레트 인덱스) 을 코딩할 수도 있다. 2 와 동일한 인덱스를 코딩한 후, 비디오 코더는, 3 개의 후속 샘플들이 2 의 동일한 팔레트 인덱스를 또한 가진다는 것을 표시하는 3 의 런을 코딩할 수도 있다. 유사한 방식으로, CopyFromTop 모드를 이용하여 인덱스를 코딩한 후에 4 개의 팔레트 인덱스들의 런을 코딩하는 것은, 총 5 개의 팔레트 인덱스들이 현재 코딩되고 있는 샘플 위치의 상부의 행 (row) 에서의 대응하는 팔레트 인덱스들로부터 복사된다는 것을 표시할 수도 있다.
- [0039] 이 개시물에서 설명된 기법들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 유도하는 것, 및 팔레트-기반 코딩 맵들 및 다른 신택스 엘리먼트들을 송신하는 것 중의 하나 이상의 다양한 조합들을 위한 기법들을 포함할 수도 있다. 일부 예들에서, 이 개시물의 기법들은 JCTVC-Q0094 (뿐만 아니라, 기고문 JCTVC-Q0094 로 업로딩되었던 팔레트 모드를 구현하는 참조 소프트웨어) 에서 존재하는 팔레트 모드들, 팔레트 인덱스들, 런들, 및 팔레트 크기들의 시그널링과 연관된 잠재적인 중복성들을 해결하기 위하여 이용될 수도 있다. 따라서, 이하에서 더욱 상세하게 설명된 바와 같이, 이 개시물의 기법들은 일부 사례들에서, 팔레트 모드를 이용하여 비디오 데이터를 코딩할 때에 효율을 개선시킬 수도 있고 비트레이트 (bitrate) 를 개선시킬 수도 있다.
- [0040] 이 개시물의 어떤 양태들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 그리고 특히, 탈출 샘플 (escape sample) 들을 시그널링하는 것과 연관된 기법들에 관한 것이다. 예를 들어, 탈출 샘플들 (또한, 탈출 픽셀들로서 지칭됨) 은 블록을 코딩하기 위한 팔레트에서 표현된 대응하는 컬러를 가지지 않는 블록의 샘플들 (또는 픽셀들) 일 수도 있다. 따라서, 탈출 샘플들은 팔레트로부터의 컬러 엔트리 (또는 픽셀 값) 를 이용하여 재구성되지 않을 수도 있다. 그 대신에, 탈출 샘플들에 대한 컬러 값들은 팔레트의 컬러 값들과는 별도로 비트스트림에서 시그널링된다.
- [0041] 이하에서 더욱 상세하게 설명된 바와 같이, 비디오 코더 (예컨대, 비디오 인코더 및 비디오 디코더) 는 예컨대, 위에서 "탈출 모드" 로서 지칭된 프로세스를 이용하여, 팔레트-코딩된 블록의 샘플이 블록에 대한 팔레트 내에 포함되지 않은 샘플의 컬러에 기초하여 코딩되는지 여부를 표시하는 샘플-당 (per-sample) 데이터를 코딩할 수도 있다. 하나의 예에서, 비디오 코더는 샘플이 예컨대, 탈출 모드를 이용하여 탈출 샘플로서 코딩되는지 여부를 표시하는 각각의 샘플에 대한 플래그를 코딩할 수도 있다 (본원에서는, 묵시적 탈출 시그널링으로서 지칭됨). 또 다른 예에서, 비디오 코더는 샘플이 예컨대, 탈출 모드를 이용하여 탈출 샘플로서 코딩된다는 것을 표시하는 샘플에 대한 (이하에서 설명된 바와 같이, 추가적인 팔레트 인덱스와 같은) 다른 신택스를 코딩할 수도 있다 (본원에서는, 명시적 탈출 시그널링으로서 지칭됨).
- [0042] 이 개시물의 양태들에 따르면, 팔레트-코딩된 블록에 대하여, 하나 이상의 신택스 엘리먼트들은 블록-레벨 (예컨대, CU 레벨 또는 LCU 레벨) 에서, 블록의 임의의 샘플이 예컨대, 탈출 샘플로서 코딩된, 팔레트 내에 포함되지 않은 샘플의 컬러 값에 기초하여 코딩되는지 여부를 표시할 수도 있다. 하나 이상의 신택스 엘리먼트들은 블록-레벨 탈출 신택스로서 지칭될 수도 있다. 예를 들어, 블록-레벨 신택스는 CU 또는 LCU 와 같은 비

디오 데이터의 블록을 코딩하면서 코딩되거나 결정되는 신택스를 지칭할 수도 있다. 블록-레벨 신택스는 헤더 내에, 또는 블록과 연관되는 다른 데이터 (예컨대, 블록의 특성을 설명하는, 블록 이전에 또는 블록에 후속하여 코딩되는 데이터) 와 함께 포함될 수도 있다. 대조적으로, 블록-레벨 신택스가 아닌 다른 신택스는 슬라이스 헤더 내에, 또는 비디오 데이터의 개별적인 픽셀들과 함께 포함될 수도 있다.

[0043] 하나의 예에서, 비디오 코더는 블록의 임의의 샘플이 팔레트 내에 포함되지 않은 컬러 값에 기초하여 코딩되는지 여부를 표시하는 (블록-레벨 탈출 플래그로서 지칭될 수도 있는) 플래그를 코딩하고 및/또는 결정하도록 구성될 수도 있다. 예를 들어, 제로 (zero) 의 플래그 값은 블록의 샘플들 중의 어느 것도 탈출 모드를 이용하여 코딩되지 않는다는 것을 표시할 수도 있다. 즉, 블록의 모든 샘플들의 값은 블록을 코딩하기 위한 팔레트 내에 포함되는 컬러 값에 기초하여 결정될 수도 있다. 1 의 플래그 값은 블록의 적어도 하나의 샘플이 탈출 모드를 이용하여 코딩된다는 것을 표시할 수도 있다. 즉, 적어도 하나의 샘플의 값은 블록을 코딩하기 위한 팔레트 내에 포함되지 않고, 별도로 시그널링될 수도 있다. 이 때문에, 플래그는 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 샘플이 블록을 코딩하기 위한 팔레트 내에 포함되지 않은 컬러 값을 가지는지 여부를 표시할 수도 있다.

[0044] 이하에서 더욱 상세하게 설명된 바와 같이, 블록-레벨 탈출 신택스는 일부 사례들에서, 비트 절감으로 귀착될 수도 있다. 예를 들어, 전체 블록의 임의의 샘플들이 탈출 샘플로서 코딩되는지 여부를 결정함으로써, 비디오 코더는 탈출 샘플들과 연관된 어떤 신택스 엘리먼트들의 코딩을 스킵 (skip) 할 수 있을 수도 있다. 즉, 신택스가 샘플들이 탈출 샘플들로서 코딩되지 않는다는 것을 표시하는 사례들에서는, 비디오 코더가 (예컨대, 위에서 언급된 샘플-당 신택스와 같은) 블록에 대한 탈출 샘플들과 연관된 임의의 다른 신택스를 코딩하지 않을 수도 있다. 이하에서 더욱 상세하게 설명된 바와 같이, 비디오 코더는 또한, 신택스가 블록의 적어도 하나의 샘플이 코딩되고 있는 블록에 대한 팔레트의 크기에 기초하여 탈출 샘플로서 코딩된다는 것을 표시할 때 어떤 신택스의 코딩을 스킵할 수도 있다. 따라서, 이 개시물의 기법들은 팔레트-기반 코딩을 이용하여 비디오 데이터를 코딩할 때에 비트레이트 및 코딩 효율을 개선시킬 수도 있다.

[0045] 이 개시물의 다른 양태들은 팔레트-모드에 대한 최대 팔레트 파라미터들을 코딩하는 것에 관한 것이다. 예를 들어, 팔레트에 대한 최대 팔레트 크기는 전형적으로, 비디오 인코더 및 비디오 디코더의 양자에서 정의되는 정적 값 (static value) 일 수도 있다. 마찬가지로, (이하에서 더욱 상세하게 설명된 바와 같이, 팔레트들을 예측하기 위하여 이용된) 팔레트 예측자의 최대 크기는 또한, 비디오 인코더 및 비디오 디코더의 양자에서 정의되는 정적 값일 수도 있다. 이 때문에, 이 최대 팔레트 파라미터들은 코딩되고 있는 비디오 데이터의 특정한 특성들에 관계 없이, 변경되지 않을 수도 있다.

[0046] 이 개시물의 양태들에 따르면, 비디오 코더는 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하도록 구성될 수도 있다. 예를 들어, 이 개시물의 양태들에 따르면, 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터는 시퀀스 파라미터 세트 (sequence parameter set; SPS) 와 같은 파라미터 세트 내에 포함될 수도 있다. 따라서, 비디오 코더는 비디오 데이터의 블록을 코딩하기 위한 컬러 값들의 팔레트의 최대 팔레트 크기를 표시하는 데이터, 또는 컬러 값들의 팔레트를 결정하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 표시하는 데이터 중의 적어도 하나를 코딩할 수도 있다.

[0047] 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하는 것은 코딩 효율을 개선시킬 수도 있는 신축성 (flexibility) 을 제공할 수도 있다. 예를 들어, 기법들은 비디오 코더가 코딩되고 있는 비디오 데이터의 특성들에 기초하여 (예컨대, 데이터의 비트-심도 (bit-depth), 블록 크기, 데이터와 연관된 프로파일 또는 레벨 등에 기초하여) 상이한 크기들의 팔레트들 및 팔레트 예측자들을 이용하도록 할 수도 있다. 따라서, 최대 팔레트 파라미터들은 코딩되고 있는 비디오 데이터에 맞추어질 수도 있어서, 상대적으로 더 큰 최대 팔레트 파라미터들이 이러한 파라미터들로부터 이득을 얻을 수도 있는 블록들에 대하여 정의될 수도 있다. 게다가, 상대적으로 더 작은 최대 팔레트 파라미터들은 상대적으로 더 큰 파라미터들로부터 이득을 얻을 가능성이 더 적은 블록들에 대한 팔레트들을 구성하는 것과 연관된 복잡도를 감소시키도록 정의될 수도 있다.

[0048] 이 개시물의 다른 양태들은 팔레트-기반 비디오 코딩을 위한 다양한 신택스 엘리먼트들을 코딩하는 기법들에 관한 것이다. 예를 들어, 이 개시물의 기법들은 코딩되고 있는 신택스의 최대 잠재적인 값을 고려하는 코드들을 이용하여, 팔레트 인덱스들의 런 값 (또한, 런 력스 값으로서 지칭됨), 팔레트 예측 벡터, 또는 다른 팔레트 관련된 신택스와 같은, 팔레트 코딩을 위한 신택스를 코딩하는 것을 포함한다. 일부 사례들에서, 이 개시물의 양태들에 따르면, 신택스는 이하에서 더욱 상세하게 설명된 바와 같이, 지수 골롬 (Exponential Golomb) 코드의 형태를 이용하여 코딩될 수도 있다. 기법들은 일부 사례들에서, 팔레트 관련된 신택스를 표현하기 위하여

필요하게 된 비트들의 수를 감소시킬 수도 있다.

- [0049] 도 1 은 이 개시물의 기법들을 사용할 수도 있는 일 예의 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 이용된 바와 같이, 용어 "비디오 코더" 는 일반적으로 비디오 인코더들 및 비디오 디코더들의 양자를 지칭한다. 이 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 일반적으로 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다. 비디오 코딩 시스템 (10) 의 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 이 개시물에서 설명된 다양한 예들에 따라 팔레트-기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스들의 예들을 표현한다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트-기반 코딩 또는 비-팔레트-기반 (non-palette-based) 코딩의 어느 하나를 이용하여, HEVC 코딩에서의 CU 들 및 PU 들과 같은 비디오 데이터의 다양한 블록들을 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트-기반 코딩 모드들은 HEVC 초안 10 에 의해 특정된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간적 코딩 모드들 또는 인트라-예측 공간적 코딩 모드들을 지칭할 수도 있다.
- [0050] 도 1 에서 도시된 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 및 목적지 디바이스 (14) 를 포함한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 생성한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치로서 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치로서 지칭될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.
- [0051] 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크톱 컴퓨터들, 이동 컴퓨팅 디바이스들, 노트북 (예컨대, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 (set-top) 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 차량내 컴퓨터들 등을 포함하는 광범위한 디바이스들을 포함할 수도 있다.
- [0052] 목적지 디바이스 (14) 는 채널 (16) 을 통해 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신할 수도 있다. 채널 (16) 은 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 하나의 예에서, 채널 (16) 은 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 직접 송신하는 것을 가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 무선 통신 프로토콜과 같은 통신 표준에 따라 인코딩된 비디오 데이터를 변조시킬 수도 있고, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 라디오 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 무선 및/또는 유선 통신 매체들을 포함할 수도 있다. 하나 이상의 통신 매체들은 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 와 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 가능하게 하는 다른 장비를 포함할 수도 있다.
- [0053] 또 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 예컨대, 디스크 액세스 또는 카드 액세스를 통해 저장 매체를 액세스할 수도 있다. 저장 매체는 블루-레이 (Blu-ray) 디스크들, DVD 들, CD-ROM 들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적당한 디지털 저장 매체들과 같은, 다양한 국소적으로 액세스된 데이터 저장 매체들을 포함할 수도 있다.
- [0054] 추가의 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 또 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 파일 서버 또는 다른 중간 저장 디바이스에서 저장되는 인코딩된 비디오 데이터를 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있으며 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 타입의 서버일 수도 있다. 일 예의 파일 서버들은 (예컨대, 웹사이트를 위한) 웹 서버들, 파일 전송 프로토콜 (file transfer protocol; FTP) 서버들, 네트워크 연결 저장 (network attached storage; NAS) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.
- [0055] 목적지 디바이스 (14) 는 인터넷 접속과 같은 표준 데이터 접속을 통해 인코딩된 비디오 데이터를 액세스할 수도 있다. 데이터 접속들의 일 예의 타입들은, 파일 서버 상에 저장된 인코딩된 비디오 데이터를 액세스하기에 적당한 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, DSL, 케이블 모뎀 등), 또는 양자의 조

합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양자의 조합일 수도 있다.

[0056] 이 개시물의 기법들은 무선 애플리케이션들 또는 설정들로 제한되지는 않는다. 기법들은 오버-디-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상에서의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 및/또는 영상 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위하여 일방향 (one-way) 또는 양방향 (two-way) 비디오 송신을 지원하도록 구성될 수도 있다.

[0057] 도 1 에서 예시된 비디오 코딩 시스템 (10) 은 단지 예이고, 이 개시물의 기법들은 인코딩 및 디코딩 디바이스들 사이의 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들 (예컨대, 비디오 인코딩 또는 비디오 디코딩) 에 적용할 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 취출 (retrieve) 되거나, 네트워크 상에서 스트리밍되는 등등과 같다. 비디오 인코딩 디바이스는 데이터를 인코딩하고 이를 메모리에 저장할 수도 있고, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 취출하고 이를 디코딩할 수도 있다. 다수의 예들에서, 인코딩 및 디코딩은, 서로 통신하지 않지만, 간단하게 데이터를 메모리로 인코딩하고 및/또는 메모리로부터 데이터를 취출하여 데이터를 디코딩하는 디바이스들에 의해 수행된다.

[0058] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 일부 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전-캡처된 비디오 데이터를 포함하는 비디오 아카이브 (video archive), 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하기 위한 비디오 공급 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽 시스템, 또는 비디오 데이터의 이러한 소스들의 조합을 포함할 수도 있다.

[0059] 비디오 인코더 (20) 는 비디오 소스 (18) 로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12) 는 출력 인터페이스 (22) 를 통해 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접적으로 송신한다. 다른 예들에서, 인코딩된 비디오 데이터는 또한, 디코딩 및/또는 재생을 위한 목적지 디바이스 (14) 에 의한 더 이후의 액세스를 위하여 저장 매체 또는 파일 서버 상으로 저장될 수도 있다.

[0060] 도 1 의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함한다. 입력 인터페이스 (28) 는 채널 (16) 상에서 인코딩된 비디오 데이터를 수신할 수도 있다. 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합될 수도 있거나, 목적지 디바이스 (14) 의 외부에 있을 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32) 는 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들을 포함할 수도 있다.

[0061] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (digital signal processor; DSP) 들, 애플리케이션 특정 집적 회로 (application-specific integrated circuit; ASIC) 들, 필드-프로그램밍가능한 게이트 어레이 (field-programmable gate array; FPGA) 들, 개별 로직, 하드웨어, 또는 그 임의의 조합들과 같은 다양한 적당한 회로부 중의 임의의 것으로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어로 구현될 경우, 디바이스는 소프트웨어를 위한 명령들을 적당한 비-일시적 (non-transitory) 컴퓨터-판독가능 저장 매체에 저장할 수도 있고, 이 개시물의 기법들을 수행하기 위하여 하나 이상의 프로세서들을 이용하여 명령들을 하드웨어로 실행할 수도 있다. 상기한 것 (하드웨어, 소프트웨어, 하드웨어 및 소프트웨어의 조합 등을 포함함) 중의 임의의 것은 하나 이상의 프로세서들인 것으로 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 인코더들 또는 디코더들 중의 어느 하나는 조합된 인코더/디코더 (CODEC) 의 일부로서 각각의 디바이스 내에 통합될 수도 있다.

[0062] 이 개시물은 일반적으로, 비디오 인코더 (20) 가 어떤 정보를 비디오 디코더 (30) 와 같은 또 다른 디바이스에 "시그널링" 또는 "송신" 하는 것을 지칭할 수도 있다. 용어 "시그널링" 또는 "송신"은 일반적으로, 압축된 비디오 데이터를 디코딩하기 위하여 이용된 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 이러한 통신은 실시간 또는 실시간에 근접하게 발생할 수도 있다. 대안적으로, 이러한 통신은,

인코딩 시에 인코딩된 비트스트림으로, 선택스 엘리먼트들을 컴퓨터-판독가능 저장 매체에 저장하고, 다음으로, 이러한 선택스 엘리먼트들이 이 매체에 저장된 후의 임의의 시간에 디코딩 디바이스에 의해 추출될 수도 있을 때에 발생할 수도 있는 것과 같이, 시간의 기간에 걸쳐 발생할 수도 있다.

[0063] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 위에서 언급되고 HEVC 초안 10 에서 설명된 HEVC 표준과 같은 비디오 압축 표준에 따라 동작한다. 기본 HEVC 표준에 추가하여, HEVC 를 위한 스케일러블 비디오 코딩, 멀티뷰 비디오 코딩, 및 3D 코딩 확장들을 생성하기 위한 노력들이 계속되고 있다. 게다가, 예컨대, 이 개시물에서 설명된 것과 같은 팔레트-기반 코딩 모드들은 HEVC 표준의 확장을 위하여 제공될 수도 있다. 일부 예들에서, 팔레트-기반 코딩을 위하여 이 개시물에서 설명된 기법들은 ITU-T-H.264/AVC 표준 또는 미래의 표준들과 같은 다른 비디오 코딩 표준들에 따라 동작하도록 구성된 인코더들 및 디코더들에 적용될 수도 있다. 따라서, HEVC 코덱에서 코딩 유닛 (CU) 들 또는 예측 유닛 (prediction unit; PU) 들의 코딩을 위한 팔레트-기반 코딩 모드의 적용은 예시의 목적들을 위하여 설명된다.

[0064] HEVC 및 다른 비디오 코딩 표준들에서는, 비디오 시퀀스가 전형적으로 일련의 픽처들을 포함한다. 픽처들은 또한, "프레임들" 로서 지칭될 수도 있다. 픽처는 S_L , S_{Cb} 및 S_{Cr} 로 나타낸 3 개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플 (luma sample) 들의 2 차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 크로미넌스 샘플 (chrominance sample) 들의 2 차원 어레이이다. S_{Cr} 은 Cr 크로미넌스 샘플 (chrominance sample) 들의 2 차원 어레이이다. 크로미넌스 샘플들은 또한, 본원에서 "크로마 (chroma)" 샘플들로서 지칭될 수도 있다. 다른 사례들에서, 픽처는 단색일 수도 있고 루마 샘플들의 어레이를 오직 포함할 수도 있다.

[0065] 픽처의 인코딩된 표현을 생성하기 위하여, 비디오 인코더 (20) 는 코딩 트리 유닛 (coding tree unit; CTU) 들의 세트를 생성할 수도 있다. CTU 들의 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2 개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하기 위하여 이용된 선택스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 $N \times N$ 블록일 수도 있다. CTU 는 또한, "트리 블록" 또는 "최대 코딩 유닛 (largest coding unit)" (LCU) 으로서 지칭될 수도 있다. HEVC 의 CTU 들은 H.264/AVC 와 같은 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 반드시 특정한 크기로 제한되는 것은 아니고, 하나 이상의 코딩 유닛 (CU) 들을 포함할 수도 있다. 슬라이스는 래스터 스캔 (raster scan) 에서 연속적으로 순서화된 정수 개수 (integer number) 의 CTU들을 포함할 수도 있다.

[0066] 코딩된 CTU 를 생성하기 위하여, 비디오 인코더 (20) 는 코딩 트리 블록들을 코딩 블록들, 이 때문에, 명칭 "코딩 트리 유닛들" 로 분할하기 위하여, CTU 의 코딩 트리 블록들에 대해 쿼드-트리 파티셔닝 (quad-tree partitioning) 을 재귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 $N \times N$ 블록이다. CU 는 루마 샘플들의 코딩 블록과, 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 가지는 픽처의 크로마 샘플들의 2 개의 대응하는 코딩 블록들, 그리고 코딩 블록들의 샘플들을 코딩하기 위하여 이용된 선택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록일 수도 있다. CU 의 예측 유닛 (PU) 은 루마 샘플들의 예측 블록, 픽처의 크로마 샘플들의 2 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하기 위하여 이용된 선택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb, 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.

[0067] 비디오 인코더 (20) 는 PU 에 대한 예측 블록들을 생성하기 위하여 인트라 예측 또는 인터 예측을 이용할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위하여 인트라 예측을 이용할 경우, 비디오 인코더 (20) 는 PU 와 연관된 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0068] 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위하여 인터 예측을 이용할 경우, 비디오 인코더 (20) 는 PU 와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 는 PU 의 예측 블록들을 생성하기 위하여 단방향-예측 (uni-prediction) 또는 양방향-예측 (bi-prediction) 을 이용할 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 생성하기 위하여 단방향-예측을 이용할 때, PU 는 단일 모션 벡터 (MV) 를 가질 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 생성하기 위하여 양방향-예측을 이용할 때, PU 는 2 개의 MV 들을 가질 수도 있다.

[0069] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성한 후, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록 (luma residual block) 을 생성할 수도 있다. CU 의 루마 잔차 블

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

[0070] 또한, 비디오 인코더 (20) 는 CU 의 루마, Cb 및 Cr 잔차 블록들을 하나 이상의 루마, Cb 및 Cr 변환 블록들로 분해하기 위하여 쿼트-트리 파티셔닝을 이용할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하기 위하여 이용된 선택스 구조들일 수도 있다. 이에 따라, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록 (sub-block) 일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다.

[0071] 비디오 인코더 (20) 는 TU 에 대한 루마 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용할 수도 있다. 계수 블록은 변환 계수들의 2 차원 어레이일 수도 있다. 변환 계수는 스칼라량 (scalar quantity) 일 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cb 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용할 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cr 계수 블록을 생성하기 위하여 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용할 수도 있다.

[0072] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록, 또는 Cr 계수 블록) 을 생성한 후, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로, 변환 계수들을 표현하기 위해 이용된 데이터의 양을 아마도 감소시키기 위하여 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 표시하는 선택스 엘리먼트들에 대해 컨텍스트-적응 2 진 산술 코딩 (Context-Adaptive Binary Arithmetic Coding; CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피-인코딩된 선택스 엘리먼트들을 비트스트림에서 출력할 수도 있다.

[0073] 비디오 인코더 (20) 는 엔트로피-인코딩된 선택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 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 캡슐화 (encapsulate) 한다. NAL 유닛 헤더는 NAL 유닛 타입 코드를 표시하는 선택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 타입 코드는 NAL 유닛의 타입을 표시한다. Rbsp 는 NAL 유닛 내에서 캡슐화되는 정수 개수의 바이트들을 포함하는 선택스 구조일 수도 있다. 일부 사례들에서, Rbsp 는 제로 비트들을 포함한다.

[0074] 상이한 타입들의 NAL 유닛들은 상이한 타입들의 Rbsp들을 캡슐화할 수도 있다. 예를 들어, 제 1 타입의 NAL 유닛은 픽처 파라미터 세트 (PPS) 에 대한 Rbsp 를 캡슐화할 수도 있고, 제 2 타입의 NAL 유닛은 코딩된 슬라이스에 대한 Rbsp 를 캡슐화할 수도 있으며, 제 3 타입의 NAL 유닛은 SEI 에 대한 Rbsp 를 캡슐화할 수도 있는 등등과 같다. (파라미터 세트들 및 SEI 메시지들에 대한 Rbsp들과 대조적으로) 비디오 코딩 데이터에 대한 Rbsp들을 캡슐화하는 NAL 유닛들은 비디오 코딩 계층 (video coding layer; VCL) NAL 유닛들로서 지칭될 수도 있다.

[0075] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 게다가, 비디오 디코더 (30) 는 비트스트림으로부터 선택스 엘리먼트들을 디코딩하기 위하여 비트스트림을 파싱 (parse) 할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 선택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 재구성할 수도 있다. 비디오 데이터를 재구성하기 위한 프로세스는 일반적으로, 비디오 인코더 (20) 에 의해 수행된 프로세스와 상반적일 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재의 CU 의 PU들에 대한 예측 블록들을 결정하기 위하여 PU 들의 MV 들을 이용할 수도 있다. 게다가, 비디오 디코더 (30) 는 현재의 CU 의 TU 들과 연관된 변환 계수 블록들을 역양자화할 수도 있다. 비

디오 디코더 (30) 는 현재의 CU 의 TU 들과 연관된 변환 블록들을 재구성하기 위하여 변환 계수 블록들에 대해 역변환들을 수행할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 PU 들에 대한 예측 블록들의 샘플들을 현재의 CU 의 TU 들의 변환 블록들의 대응하는 샘플들에 추가함으로써, 현재의 CU 의 코딩 블록들을 재구성할 수도 있다. 픽처의 각각의 CU 에 대한 코딩 블록들을 재구성함으로써, 비디오 디코더 (30) 는 픽처를 재구성할 수도 있다.

[0076] 일부 예들에서, 비디오 인코더 (20) 및 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) 및 비디오 디코더 (30) 는 특정한 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 표현하기 위한 컬러들의 테이블로서, 소위 팔레트를 코딩할 수도 있다. 각각의 픽셀은 픽셀의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 픽셀 값을 팔레트에서의 적절한 값에 관련시키는 인덱스를 코딩할 수도 있다.

[0077] 상기 예에서, 비디오 인코더 (20) 는 블록에 대한 팔레트를 결정함으로써, 각각의 픽셀의 값을 표현하기 위하여 팔레트에서 엔트리를 위치시킴으로써, 그리고 픽셀 값을 팔레트에 관련시키는 픽셀들에 대한 팔레트 인덱스들로 팔레트를 인코딩함으로써, 비디오 데이터의 블록을 인코딩할 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터, 블록에 대한 팔레트뿐만 아니라 블록의 픽셀들에 대한 팔레트 인덱스들을 획득할 수도 있다. 비디오 디코더 (30) 는 블록의 픽셀 값들을 재구성하기 위하여, 픽셀들의 팔레트 인덱스들을 팔레트의 엔트리들에 관련시킬 수도 있다.

[0078] 위에서 언급된 바와 같이, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트의 팔레트 인덱스들을 코딩하기 위하여 다수의 상이한 팔레트 코딩 모드들을 이용할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록의 팔레트 인덱스들을 코딩하기 위하여 탈출 모드, CopyFromTop 모드 (또한, CopyAbove 모드로서 지칭됨), 또는 값 모드 (또한, 인덱스 모드로서 지칭됨) 를 이용할 수도 있다. 일반적으로, "탈출 모드" 를 이용하여 샘플을 코딩하는 것은 일반적으로, 블록을 코딩하기 위한 팔레트에서 표현된 대응하는 컬러를 가지지 않는 블록의 샘플을 코딩하는 것을 지칭할 수도 있다. 위에서 언급된 바와 같이, 이러한 샘플들은 탈출 샘플들 또는 탈출 픽셀들로서 지칭될 수도 있다.

[0079] 또 다른 예의 팔레트 코딩 모드는 Yu-Wen Huang 등, "Description of Screen Content Core Experiment 3 (SCCE3): Palette Mode", JCTVC-Q1123, Valencia, ES, 27 2014 년 3 월 27 일 - 2014 년 4 월 4 일 (이하, Q1123) 에서 설명된 바와 같이, 제 3 스크린 콘텐츠 코딩 코어 실험, 서브테스트 B.6 에서 설명되어 있고, 또 다른 모드는 2014 년 3 월 26 일자로 Canon 에 의해 배포된 소프트웨어 내에 도입되었다. 이 모드에 대한 매크로는 "CANON_NEW_RUN_LAST_TRANSITION" 이었고, 본원에서 트랜지션 런 모드 (Transition Run mode) 로서 지칭될 수도 있다. 트랜지션 런은 비디오 인코더 (20) 또는 비디오 디코더 (30) 가 인덱스 값과, 그 다음으로, 동일한 팔레트 인덱스를 가지는 후속 샘플들의 수를 특징하는 런을 코딩할 수도 있다는 점에서 값 모드와 유사할 수도 있다.

[0080] 값 모드와 트랜지션 런 모드 사이의 차이는 트랜지션 런 모드의 팔레트 인덱스가 비트스트림에서 시그널링되지 않는다는 것이다. 오히려, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 인덱스를 추론할 수도 있다. 본원에서 설명된 바와 같이, 값을 추론하는 것은 비트스트림에서 코딩되는 값을 표현하는 전용 인덱스를 참조하지 않는 값의 결정을 지칭할 수도 있다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비트스트림에서의 값에 대한 전용 인덱스 엘리먼트를 코딩하지 않으면서, 값을 추론할 수도 있다. 추론된 인덱스는 트랜지션 인덱스로서 지칭될 수도 있다.

[0081] 일부 예들에서, 팔레트 모드들을 시그널링하는 2 개의 방법들이 있을 수도 있다. 팔레트 모드들을 시그널링하기 위한 제 1 기법은 명시적 탈출 시그널링으로서 지칭될 수도 있다. 예를 들어, JCTVC-Q0094 에서, 매크로 "PLT_REMOVE_ESCAPE_FLAG" 가 제로일 경우, 비디오 인코더 (20) 는 블록에서 코딩되고 있는 샘플이 탈출 모드에서 코딩되는지 여부를 표시하기 위하여, 블록의 각각의 샘플에 대한 탈출 플래그를 명시적으로 인코딩할 수도 있다. 샘플이 탈출 모드로 코딩되지 않을 경우, 비디오 인코더 (20) 는 모드가 CopyFromTop 또는 값인지 여부를 표시하기 위한 추가적인 데이터를 인코딩할 수도 있다. 일부 사례들에서, 추가적인 데이터는 SPoint 플래그로서 본원에서 지칭된 플래그일 수도 있다 (예컨대, 제로의 SPoint 플래그 값은 CopyFromTop 모드를 표시할 수도 있고, 1 의 SPoint 플래그 값은 값 모드를 표시할 수도 있고, 그 반대도 마찬가지임).

[0082] 이 때문에, 명시적 탈출 시그널링으로, SPoint 플래그는 표시된 모드와 연관된 픽셀 값들의 런에 대한 특정한 런 타입을 표시하기 위하여 이용될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 현재 코딩되고 있는 인덱

스 및 런에서 코딩되고 있는 후속 팔레트 인덱스들의 런이 CopyFromTop 모드 또는 값 모드를 이용하여 코딩되는 지 여부를 표시하기 위하여, SPoint 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 후속 런 샘플들에 대한 탈출 플래그 (예컨대, "PLT_REMOVE_ESCAPE_FLAG") 및 SPoint 플래그 (필요할 때) 를 인코딩하지 않는다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 런 내에 포함된 샘플들에 대한 탈출 플래그 및 SPoint 플래그의 값들을 추론할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비트 스트림에서 이러한 값들을 표현하는 전용 신택스를 참조하지 않으면서, 런 내에 포함된 샘플들에 대한 탈출 플래그 및 SPoint 플래그의 값을 결정할 수도 있다.

[0083] 팔레트 모드들을 시그널링하기 위한 제 2 기법은 묵시적 탈출 시그널링으로서 지칭될 수도 있다. 예를 들어, JCTVC-Q0094 로부터의 매크로 "PLT_REMOVE_ESCAPE_FLAG" 가 1 일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 특수한 인덱스를, 임의의 팔레트 엔트리에 대응하지 않는 팔레트에 수용하기 위하여, 팔레트의 팔레트 엔트리들의 수를 1 만큼 증가시키도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 주어진 블록에 대한 증가된 팔레트에서의 최후 팔레트 인덱스로서, 추가적인 인덱스를 포함할 수도 있다. 추가적인 인덱스는 탈출 모드의 표시로서 이용될 수도 있다.

[0084] 묵시적 탈출 시그널링을 수행할 때, 비디오 인코더 (20) 는 블록의 특정한 샘플 값에 대하여, 추가적인 샘플이 탈출 샘플 (예컨대, 블록을 코딩하기 위한 팔레트에서 표현된 컬러 값을 가지지 않는 샘플) 로서 코딩된다는 것을 표시하기 위하여 추가적인 인덱스를 표현하는 데이터를 인코딩될 수도 있다. 비디오 인코더 (20) 는 또한, 탈출 샘플의 컬러 값 (들) 을 인코딩할 수도 있다. 따라서, 묵시적 탈출 시그널링의 경우, 명시적 신택스를 이용하여 시그널링되어야 할 오직 2 개의 가능한 모드들 (예컨대, CopyFromTop 모드 또는 값 모드 (또한, 인덱스 모드로서 지칭됨)) 이 있다. 예를 들어, 오직 SPoint flag 가 모드들 사이를 구별하기 위하여 시그널링될 수도 있다. 샘플이 값 모드에서 코딩되고 값 모드에 대한 인덱스가 탈출 인덱스 (예컨대, 팔레트에 대한 상기 언급된 추가적인 인덱스) 와 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 탈출 샘플로서 코딩되어야 할 샘플을 추론할 수도 있다. 이 경우, 런은 시그널링되지 않는다. 트랜지션 런 모드를 갖는 묵시적 탈출 시그널링을 이용할 때, SPoint 플래그는 값들 0 (예컨대, 값 모드), 1 (예컨대, CopyFromTop 모드), 또는 2 (예컨대, 트랜지션 런 모드) 를 취할 수도 있다.

[0085] 이 개시물에서 설명된 기법들은 팔레트-기반 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 유도하는 것, 및 팔레트-기반 코딩 맵 및 다른 신택스 엘리먼트들을 송신하는 것 중의 하나 이상의 다양한 조합들을 위한 기법들을 포함할 수도 있다. 일부 예들에서, 이 개시물의 기법들은 JCTVC-Q0094 (뿐만 아니라, 기고문 JCTVC-Q0094 로 업로딩되었던 팔레트 모드를 구현하는 참조 소프트웨어) 에서 존재하는 팔레트 모드들, 팔레트 인덱스들, 런들, 및 팔레트 크기들의 시그널링과 연관된 잠재적인 중복성들을 해결하기 위하여 이용될 수도 있다.

[0086] JCTVC-Q0094 에서 설명된 기법들과 연관된 소프트웨어에서는, 어떤 시그널링 중복성들이 이미 고려되었고 제거되었다. 예를 들어, JCTVC-Q0094 에서는, 팔레트 모드로 코딩된 블록이 전형적으로 현재의 블록을 예측하기 위하여 상부-이웃 블록으로부터의 재구성된 샘플들을 이용할 수 없으므로, SPoint 플래그는 블록의 최초 행에서의 샘플들에 대하여 시그널링되지 않는다. 상부-이웃 블록은 일반적으로 블록과 이웃하고 블록 상부에 위치되는 블록을 지칭할 수도 있다. 유사하게, 현재 코딩되고 있는 샘플을 선행하는 샘플에 대한 모드가 CopyFromTop 일 경우, 현재의 픽셀에 대한 모드는 CopyFromTop 일 수 없다.

[0087] 그러나, 이 개시물은 전적으로 또는 선택적으로 제거될 수 있는 다른 시그널링 중복성들 및/또는 비효율성들을 인식한다. 이하에서 더욱 상세하게 설명된 바와 같이, 기법들은 왜곡을 실질적으로 초래하지 않으면서, 비디오 코딩 비트레이트를 개선시킨다. 하나의 예로서, 현재의 샘플의 바로 상부의 샘플이 탈출 샘플일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CopyFromTop 모드를 이용하여 현재의 샘플을 코딩하지 않도록 구성될 수도 있다. 이 경우, 비디오 인코더 (20) 는 샘플에 대한 SPoint 를 시그널링하지 않을 수도 있고, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 필요할 경우, SPoint 플래그를 값 모드와 동일한 것으로 추론할 수도 있다.

[0088] 또 다른 예에서, 이 개시물의 기법들에 따르면, 이전의 샘플뿐만 아니라, 블록에서의 현재의 샘플의 바로 상부의 샘플도 탈출 샘플들이 아니고, 이전 및 상부 샘플들이 동일한 팔레트 인덱스를 가질 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CopyFromTop 모드를 이용하여 현재의 샘플을 코딩하지 않도록 구성된다. 이것은 CopyFromTop 모드에 대하여, 현재의 샘플의 인덱스가 이전의 샘플과 동일할 것이기 때문이다. 이전의 샘플에 대한 모드가 값 모드이었을 경우, 값 모드와 연관된 런은 현재의 샘플을 편입하기 위하여 1 만큼 확장될

것이다. 다른 한편으로, 이전의 샘플에 대한 모드가 CopyFromTop 이었을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 위에서 언급된 바와 같이, CopyFromTop 모드를 이용하여 현재의 샘플을 코딩하지 않도록 구성될 수도 있다. 이에 따라, 이 경우, 비디오 인코더 (20) 는 현재의 샘플에 대한 SPoint 플래그를 시그널링하지 않을 수도 있고, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 필요할 경우, SPoint 플래그를 값 모드와 동일한 것으로 추론할 수도 있다.

[0089] 또 다른 예에서, 이 개시물의 기법들에 따르면, 이전의 런이 코딩되고 있는 블록의 폭 마이너스 (minus) 1 이상일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CopyFromTop 모드를 이용하여 현재의 샘플을 코딩하지 않도록 구성될 수도 있다. CopyFromTop 모드는 위에서 설명된 바와 같이, CopyFromTop 모드를 후행하지 않을 수도 있으므로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 이전의 샘플과 연관된 모드가 CopyFromTop 모드를 이용하여 코딩될 경우, 현재의 샘플로부터의 모드가 CopyFromTop 모드를 이용하여 코딩되지 않을 수도 있다는 것을 추론할 수도 있다. 이전의 런이 값 모드를 이용하여 코딩되었고, 이전의 런이 블록의 폭 마이너스 1 이상이었을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 이전의 샘플 및 현재의 샘플의 바로 상부의 샘플에 대한 팔레트 인덱스들이 (위에서 설명된 예와 유사한 방식으로) 동일한 것으로 결정하도록 구성될 수도 있다. 이 경우, 현재의 샘플이 동일한 인덱스를 가지지 않을 경우, CopyFromTop 모드를 불가능하게 한다. 이에 따라, 이 경우, 비디오 인코더 (20) 는 현재의 샘플에 대한 SPoint 플래그를 시그널링하지 않을 수도 있고, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 필요할 경우, SPoint 플래그를 값 모드와 동일한 것으로 추론할 수도 있다.

[0090] 또 다른 예에서, 이 개시물의 양태들에 따르면, 팔레트 크기가 명시적 탈출 시그널링을 이용할 때에 코딩되는 블록에 대한 것일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 2013 년 7 월 12 일자로 출원된 미국 가출원 제 61/845,824 호, 2013 년 11 월 1 일자로 출원된 미국 가출원 제 61/899,048 호, 또는 2013 년 12 월 6 일자로 출원된 미국 가출원 제 61/913,040 호에서 설명된 팔레트 인덱스들과 같은 어떤 팔레트 인덱스들을 코딩하지 않도록 구성될 수도 있다. 게다가, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 필요할 경우에 SPoint 플래그를 값 모드와 동일한 것으로 추론할 수도 있으므로, 비디오 인코더 (20) 는 SPoint 플래그를 코딩하지 않도록 구성될 수도 있다. 이것은 현재의 샘플이 탈출 샘플 (예컨대, 블록을 코딩하기 위한 팔레트에서 표현된 컬러 값을 가지지 않는 샘플) 로서 코딩되지 않을 경우, 현재의 샘플에 대한 팔레트 인덱스가 이미 알려져 있고, (오직 하나의 가능한 팔레트 인덱스로서) 채로와 동일한 것으로 유도된다. 이 경우, 오직 런이 시그널링된다. 양자의 모드들은 동일한 결과를 제공하므로, CopyFromTop 및 값 모드들 사이를 구별하는 것은 필요하지 않다. 유사하게, 묵시적 탈출 시그널링에 대하여, 팔레트 크기가 2 일 때, 비디오 인코더 (20) 는 값 및 탈출 모드들 사이를 구별하기 위하여 팔레트 인덱스들을 시그널링할 수도 있지만, SPoint 플래그의 시그널링은 상기와 동일한 이유들로 필요하지 않다.

[0091] 이 개시물의 기법들은 또한, 값, CopyFromTop, 및 트랜지션 런 모드들을 이용할 때에 중복성들을 제거하기 위하여 이용될 수도 있다. 이 때문에, 기법들은 왜곡을 실질적으로 초래하지 않으면서, 비디오 코딩 비트 레이트 효율을 개선시킬 수도 있다. 예시의 목적들을 위한 예에서, 현재의 샘플은 값 모드에서 코딩되고, 트랜지션 런 모드는 이용을 위하여 이용가능하지 않다 (예컨대, 오직 CopyFromAbove 및 값 모드들이 이용가능함). 이 예에서, 이전의 샘플에 대한 모드가 값일 때, 현재의 샘플의 인덱스는 이전의 샘플의 그것과 동일할 수 없고, 이와 다를 경우, 현재의 샘플은 이전의 값 모드 내로 포함되고, 값 모드에 대한 런은 1 만큼 증분 (increment) 된다. 유사하게, 이전의 샘플에 대한 모드가 CopyFromTop 일 때, 코딩되어야 할 현재의 샘플의 인덱스는 상부의 것과 동일할 수 없고, 이와 다를 경우, 현재의 샘플은 CopyFromTop 모드로 코딩되고, 아마도, CopyFromTop 모드에 대한 런은 1 만큼 증분될 것이다.

[0092] 상기 설명된 관계를 유념하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인덱스가 이전의 샘플 (예컨대, 이전의 샘플이 값 모드에 있을 경우) 또는 상부 샘플 (예컨대, 이전의 샘플이 CopyFromTop 모드에 있을 경우) 에 대한 인덱스보다 더 클 때에 현재의 샘플에 대한 인덱스를 1 만큼 감소시킬 수도 있다. 이 프로세스는 C. Gisquet 등, "AHG10: Palette Index Coding" JCTVC-Q0064, Valencia, ES, 2014 년 3 월 27 일 - 2014 년 4 월 4 일 (이하, JCTVC-Q0064) 에서 설명되어 있다. 또한, 최대 가능한 팔레트 인덱스들의 수는 이전의 조건이 참 (true) (현재의 인덱스가 이전의 좌측 또는 상부 팔레트 인덱스들보다 더 큼) 인지 여부에 관계 없이, 1 만큼 감소될 수도 있다. 예를 들어, 인덱스를 코딩하기 위하여 (예컨대, 절단된 2 진 코드 (truncated binary code) 와 같은) 가변 길이 코드를 이용할 때, 팔레트 엔트리들의 수는 1 만큼 감소될 수도 있다.

[0093] 이 개시물의 양태들에 따르면, 위에서 설명된 프로세스에 대해 대안적으로 또는 추가적으로, 값 모드에서의 인덱스 조절 프로세스는 트랜지션 런 모드를 이용할 때에 추가로 수정될 수도 있다. 예를 들어, 이 개시물의

양태들에 따르면, 샘플이 블록에서의 최초 샘플이 아니고 이전의 샘플이 탈출 샘플로서 코딩되지 않을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이하에서 설명된 인덱스 조절 프로세스를 수행할 수도 있다.

[0094] 이 개시물의 양태들에 따르면, 이전의 샘플이 값 또는 트랜지션 런 모드에서 코딩될 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 엔트리들의 수를 1 만큼 감소시키도록 구성될 수도 있다. 게다가, 인덱스 값이 이전의 샘플의 인덱스 값 (인덱스 또는 트랜지션 인덱스) 이상일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재의 인덱스 값을 1 만큼 감소시키도록 구성될 수도 있다. 이 개시물은 이 감분된 (decremented) 값을 조절된 인덱스 값으로서 지칭할 수도 있다. 다음으로, 이전의 샘플에 대한 인덱스 값이 트랜지션 인덱스와 동일하지 않고, 팔레트 엔트리들의 수가 1 보다 더 클 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 가변 "업데이트" 를 1 로 설정하고, 이와 다를 경우, "업데이트" 를 제로로 설정하도록 구성될 수도 있다. 업데이트가 1 과 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 엔트리들의 수를 1 만큼 추가로 감소시킬 수도 있다. 이 개시물은 팔레트 엔트리들의 감분된 수를 조절된 팔레트 크기로서 지칭할 수도 있다.

[0095] 게다가, 트랜지션 인덱스가 이전의 샘플에 대한 인덱스 값 이상일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 트랜지션 인덱스를 1 만큼 감소시키도록 구성될 수도 있다. 이 개시물은 감분된 트랜지션 인덱스를 조절된 트랜지션 인덱스 값으로서 지칭할 수도 있다. 업데이트가 1 과 동일하고 조절된 인덱스 값이 조절된 트랜지션 인덱스 값보다 더 클 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 조절된 인덱스 값을 1 만큼 추가로 감소시키도록 구성될 수도 있다. 추가적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 조절된 팔레트 크기가 1 보다 더 클 경우에 최후 인덱스 조절을 오직 수행하도록 구성될 수도 있다. 이것은 조절된 인덱스 값이 조절된 팔레트 값이 1 보다 더 클 경우에 오직 시그널링될 수도 있기 때문이다.

[0096] 조절된 팔레트 크기가 1 보다 더 클 경우, 비디오 인코더 (20) 는 팔레트 인덱스들의 최대 가능한 수가 조절된 팔레트 크기와 동일할 수도 있다는 것을 참작하면서, 조절된 인덱스 값의 표시를 인코딩할 수도 있다. 이 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본원에서 설명된 절단된 2 진 코딩과 같은, 코딩을 위한 절단된 2 진화 (truncated binarization) 를 이용하도록 구성될 수도 있다.

[0097] 일부 예들에서, 이 개시물의 양태들에 따르면, 위에서 설명된 프로세스와 유사한 프로세스는 픽셀 값과, 현재의 샘플의 바로 상부의 픽셀을 위하여 이용된 모드를 검사함으로써 수행될 수도 있다. 즉, 현재의 픽셀의 좌측에 위치한 픽셀에 대하여 위에서 설명된 프로세스는 상부 이웃 픽셀들 대신에 수행될 수도 있고, 여기서, 좌측 샘플 값 및 위에서 설명된 모드는 상부 픽셀 값 및 모드로 대체된다.

[0098] 예를 들어, 샘플이 최초 행에 있지 않고, 이전의 샘플이 CopyFromTop 모드에서 코딩되고, 상부 샘플이 탈출 샘플로서 코딩되지 않을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 엔트리들의 수를 1 만큼 감소시키도록 구성될 수도 있다. 게다가, 현재의 인덱스 값이 바로 상부의 샘플의 인덱스 값 이상일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재의 인덱스 값을 1 만큼 감소시키도록 구성될 수도 있다. 또한, 이 감분된 인덱스 값은 조절된 인덱스 값으로서 지칭될 수도 있다. 다음으로, 바로 상부의 샘플에 대한 인덱스 값이 트랜지션 인덱스와 동일하지 않고, 팔레트 엔트리들의 수가 1 보다 더 클 경우, 비디오 인코더 (20) 는 가변 업데이트를 1 로 설정할 수도 있고, 이와 다를 경우, 업데이트를 제로로 설정할 수도 있다.

[0099] 업데이트가 1 과 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 엔트리들의 수를 1 만큼 추가로 감소시키도록 구성될 수도 있고, 이것은 조절된 팔레트 크기로서 지칭될 수도 있다. 게다가, 트랜지션 인덱스가 상부 샘플에 대한 인덱스 값 이상일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 트랜지션 인덱스를 1 만큼 감소시키도록 구성될 수도 있고, 이것은 조절된 트랜지션 인덱스 값으로서 지칭될 수도 있다. 업데이트가 제로와 동일하고 조절된 인덱스 값이 조절된 트랜지션 인덱스 값보다 더 클 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 조절된 인덱스 값을 1 만큼 감소시키도록 구성될 수도 있다. 추가적으로, 조절된 인덱스 값은 전형적으로, 조절된 팔레트 크기가 1 보다 더 클 경우에 오직 시그널링되므로, 최후 인덱스 조절은 조절된 팔레트 크기가 1 보다 더 클 경우에만 수행될 수도 있다.

[0100] 조절된 팔레트 크기가 1 보다 더 클 경우, 비디오 인코더 (20) 는 조절된 인덱스 값의 표시를 인코딩하도록 구성될 수도 있고, 일부 예들에서, 조절된 팔레트 크기와 동일한 팔레트 인덱스들의 최대 가능한 수를 참작할 수도 있다. 이 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본원에서 설명된 절단된 2 진 코딩과 같은, 절단된 2 진화를 이용하도록 구성될 수도 있다.

[0101] 트랜지션 런 모드와 관련된 팔레트 인덱스 시그널링에서의 중복성 제거는 위에서 설명되어 있다. 그러나,

이 기법들은 이하에서, 그리고 2014 년 5 월 23 일자로 출원된 미국 가출원 제 62/002,717 호 및 2014 년 6 월 9 일자로 출원된 미국 가출원 제 62/009,772 호에서 설명된 바와 같이, 제한된 런 방법과 조합될 수도 있다.

이 경우, 어떤 팔레트 인덱스 값 위에서, 런은 항상 제로와 동일하고, 이 때문에, 그 팔레트 인덱스들에 대하여, 비디오 인코더 (20) 는 런 값의 표시를 인코딩하지 않도록 구성될 수도 있다. 오히려, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 런 값을 제로와 동일한 것으로 유도하도록 구성될 수도 있다. 이 조합에 대하여, 트랜지션 런 모드에 대해 위에서 설명된 기법들은 변경되지 않고 남아 있다. 즉, 예를 들어, 위에서 설명된 중복성 제거 기법들은 또한, 제한된 런 모드와 함께 이용될 수도 있다.

[0102] 추가적으로 또는 대안적으로, 이 개시물의 기법들은 또한, 표준 제출 문서 Guillaume Laroche 등, "AHG10: Run Coding for Palette Mode", JCTVC-Q0066, Valencia, ES, 2014 년 3 월 27 일 - 4 월 4 일 (이하, JCTVC-Q0066) 에서 제안된 바와 같이, 제한된 런 기법과 조합될 수도 있다. 이 예에서는, 제한 인덱스가 또한 특정된다. 그러나, 상기 설명된 제한된 런 기법과의 하나의 차이는 제한 인덱스보다 더 큰 팔레트 인덱스들이 또한, 1 이상인 런들을 가질 수도 있다는 것이다. 그러나, 비디오 인코더 (20) 는 런들을 시그널링하지 않을 수도 있다. 이 제 2 제한된 런 기법을 구현할 때, 이 개시물의 중복성 제거 기법들은, 이전의 픽셀의 인덱스 값이 제한 인덱스 값 이하이거나, 상부 픽셀의 인덱스 값이 제한 인덱스 값 이하일 경우에 오직 적용될 수도 있다.

[0103] 상기 설명된 기법들은 일반적으로 (비디오 인코더 (20) 와 같은) 비디오 인코더에 대하여 설명된다. 인코더 측과 동일한 조건들을 이용하는 (예를 들어, 비디오 디코더 (30) 에 의해 구현된 바와 같은) 디코더 측 상에서는, 비디오 디코더 (30) 가 또한, 팔레트 엔트리들의 수 및 트랜지션 인덱스를 조절할 수도 있다. 다음으로, 비디오 디코더 (30) 는 팔레트 엔트리들의 조절된 수를 이용하여 인덱스를 디코딩할 수도 있다. 디코딩된 인덱스는 인코더 측 상에서와 동일한 조건들을 이용하여 (감분 대신에) 증분될 수도 있다.

[0104] 설명된 어떤 기법들은 CopyFromTop 모드가 가능하지 않은 사례들에서 중복성을 감소시키고, 이 때문에, SPoint 플래그의 시그널링은 비디오 인코더 (20) 및 비디오 디코더 (30) 가 값 모드를 추론할 수도 있도록 수정될 수도 있다. 이 개시물의 양태들에 따르면, 중복성 감소 기법들은 트랜지션 런 모드가 또한, 이용되고 있는 경우로 확장될 수도 있다. 이 경우, CopyFromTop 모드는 다음의 조건들 중의 임의의 것이 참일 경우에 가능하지 않다:

[0105] 1. 샘플이 최초 행에 있다.

[0106] 2. 이전의 샘플에 대한 모드는 CopyFromTop 이다.

[0107] 3. 상부의 픽셀은 탈출 모드에서 코딩되고, 샘플은 최초 행에 있지 않고, 이전의 샘플은 CopyFromTop 모드에서 복사되지 않는다.

[0108] 4. 상부의 샘플 및 이전의 샘플은 동일한 인덱스를 가지고, 이전의 샘플은 탈출 모드에서 코딩되지 않는다.

[0109] 이 개시물의 기법들은 또한, 탈출 팔레트 모드에 대한 탈출 플래그를 명시적으로 시그널링하기 위한 대안을 제공한다. 예를 들어, 명시적 탈출 시그널링으로 SPoint 플래그 전에 탈출 플래그를 시그널링하는 대신에, 이 개시물의 양태들에 따르면, 플래그들의 순서는 그 플래그들의 시맨틱 (semantics) 을 또한 변경하면서 교체될 수도 있다. 이 경우, 비디오 인코더 (20) 는 SPoint 플래그를 먼저 비트스트림에서 시그널링할 수도 있다.

이 예에서, 1 과 동일한 SPoint 플래그는 값 모드를 표시할 수도 있는 반면, 제로와 동일한 SPoint 플래그는 현재의 샘플에 대한 팔레트 모드가 CopyFromTop 또는 탈출 중의 어느 하나라는 것을 표시할 수도 있다. 게다가, SPoint 플래그가 1 과 동일할 때, 비디오 인코더 (20) 는 CopyFromTop 모드와 탈출 모드 사이를 구별하기 위하여 탈출 플래그를 시그널링할 수도 있다.

[0110] 상기 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 상기 설명된 플래그들 중의 적어도 하나 또는 상기 설명된 플래그들 (예컨대, SPoint 플래그 또는 탈출 플래그) 의 양자를 코딩하기 위하여 CABAC 를 이용하도록 구성될 수도 있다. 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 컨텍스트 코딩된 빈 (context coded bin) 들의 수를 감소시키기 위하여 CABAC 우회 모드를 이용하여 이러한 플래그들을 코딩하도록 구성될 수도 있다.

[0111] 위에서 설명된 바와 같이, CopyFromTop 모드는 어떤 조건들 하에서 가능하지 않을 수도 있다. 이러한 경우들에는, (예컨대, 플래그들을 교체하는 것과 같은) 대안적인 시그널링 방법을 이용할 때, 비디오 인코더 (20) 가 탈출 플래그를 시그널링하지 않으면서 SPoint 플래그를 오직 시그널링하도록 구성될 수도 있다. 이 경우, SPoint 플래그는 상이한 시맨틱을 가질 수도 있다. 예를 들어, 1 과 동일한 SPoint 플래그는 모드가

값 모드라는 것을 여전히 표시할 수도 있지만, 제로와 동일한 SPoint 플래그는 탈출 모드를 표시할 수도 있다.

SPoint 플래그가 CABAC 를 이용하여 컨텍스트-코딩될 경우, 추가적인 별도의 컨텍스트는 CopyFromTop 모드가 불가능한 경우들에 있어서 SPoint 플래그 값을 코딩하기 위하여 이용될 수도 있다. 위에서 설명된 바와 같이, 팔레트 크기가 1 이고 탈출 모드가 이용되고 있는 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 대안적인 시그널링 방법을 이용할 때에 SPoint 플래그의 코딩을 스킵하도록 구성될 수도 있다.

[0112] 이 개시물의 기법들은 또한, 탈출 팔레트 모드에 대한 탈출 플래그를 시그널링하기 위한 (예컨대, JCTVC-Q0094에 관련된) 또 다른 대안적인 시그널링 기법을 제공한다. 예를 들어, JCTVC-Q0094 에서는, 어떤 시그널링 중복성들이 고려되었고 참조 소프트웨어에서 제거되었다. 하나의 예로서, 현재의 샘플을 코딩할 때, 이전의 샘플에 대한 팔레트 모드가 CopyFromTop 일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CopyFromTop 모드를 이용하여 현재의 픽셀을 코딩하지 않을 수도 있다. 유사하게, 이전의 샘플에 대한 모드가 팔레트 인덱스 "X" 를 갖는 값 모드일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 동일한 팔레트 인덱스 "X" 를 갖는 값 모드를 이용하여 현재의 픽셀을 코딩하지 않을 수도 있다. 파싱 단계에서 (예컨대, 비디오 디코더 (30) 에서 인코딩된 비트스트림으로부터 선택스 엘리먼트들을 파싱할 때), 비디오 디코더 (30) 는 비트스트림을 적당하게 판독하기 위하여 어느 선택스 엘리먼트들이 허용되는지를 결정하기 위하여 상기 언급된 조건들을 검사한다. 이 검사 프로세스는 다수의 이러한 조건들이 검사되어야 할 경우에 부담스러워질 수도 있다.

[0113] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 탈출 모드를 묵시적으로 시그널링하기 위하여 상기 언급된 중복성들을 "제이용" 하도록 구성될 수도 있다. 예를 들어, 블록의 현재의 샘플을 코딩할 때, 이전의 샘플이 CopyFromTop 모드를 이용하여 코딩되고 현재의 픽셀에 대한 모드가 또한 CopyFromTop 으로서 시그널링될 경우, 비디오 디코더 (30) 는 현재의 블록에 대한 모드가 탈출 모드인 것을 추론할 수도 있다. 즉, 비디오 인코더 (20) 는 탈출 모드를 시그널링하기 위하여 CopyFromTop 모드로 코딩되고 있는 행에서의 2 개의 샘플들의 중복성을 이용할 수도 있다. 유사하게, 현재 코딩되고 있는 샘플에 대한 이전의 샘플에 대한 모드가 팔레트 인덱스 "X" 를 갖는 값 모드이고, 시그널링된 모드가 동일한 팔레트 인덱스 "X" 를 갖는 값 모드일 경우, 비디오 디코더 (30) 는 현재의 블록에 대한 모드를 탈출 모드인 것을 추론할 수도 있다. 유사하게, 위에서 설명된 다른 중복성들이 또한 이러한 방법으로 활용될 수도 있다.

[0114] 위에서 설명된 예들에서, 중복성들에 기초하여 탈출 모드를 시그널링하는 것은 비디오 인코더 (20) 가 탈출 모드를 시그널링할 수도 있는 가능한 상황들의 전부를 포함하지는 않는다. 따라서, 이 기법들은 탈출 모드를 시그널링하기 위한 상보적 방법으로서 이용될 수도 있다. 다른 예들에서, 기법들은 비트스트림에 대해 부과될 수도 있어서, 탈출 모드는 이 제약된 상황들에서 오직 시그널링될 수도 있다.

[0115] 이 개시물의 기법들은 또한, 샘플이 제로와 동일한 팔레트 크기에 대한 탈출 샘플인 것을 시그널링하는 것에 관한 것이다. 예를 들어, 이 개시물의 양태들에 따르면, 현재 코딩되고 있는 블록과 연관된 팔레트의 팔레트 크기가 명시적 탈출 시그널링을 이용할 때에 제로와 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록에서의 샘플들의 전부가 탈출 샘플들로서 코딩된다는 것을 추론하도록 구성될 수도 있다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 비트스트림에서 탈출 모드를 표현하는 전용 선택스를 인코딩하거나 디코딩하지 않으면서, 블록에서의 모든 샘플들이 탈출 샘플들 (예컨대, 블록을 코딩하기 위한 팔레트에서 표현된 컬러 값을 가지지 않는 샘플들) 로서 코딩되는 것으로 결정하도록 구성될 수도 있다. 마찬가지로, 현재 코딩되고 있는 블록과 연관된 팔레트의 팔레트 크기가 묵시적 탈출 시그널링을 이용할 때에 1 과 동일할 경우 (예컨대, 위에서 설명된 바와 같이, 오직 팔레트의 인덱스가 탈출 모드를 시그널링하기 위하여 이용된 추가적인 인덱스임), 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록에서의 샘플들의 전부가 탈출 샘플들로서 코딩된다는 것을 추론하도록 구성될 수도 있다.

[0116] (예컨대, 명시적 및 묵시적 탈출 시그널링의 양자에 대한) 상기 설명된 예들의 양자에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록의 나머지에 대한 어떤 팔레트-기반 선택스의 코딩을 스킵할 수도 있다. 예를 들어, 명시적 탈출 시그널링에 대하여, 비디오 인코더 (20) 는 블록의 샘플들에 대한 탈출 플래그를 시그널링하지 않을 수도 있다. 게다가, 명시적 및 묵시적 탈출 시그널링의 양자에 대하여, 비디오 인코더 (20) 는 (묵시적 및 명시적 탈출 시그널링에 대하여) SPoint 플래그를 시그널링하지 않을 수도 있다. 즉, 블록에 대한 모든 샘플들이 탈출 샘플들인 것으로 추론될 수도 있기 때문에, 비디오 인코더 (20) 는 CopyFromTop 및 값 모드들 사이를 구별하기 위하여 Spoint 플래그를 시그널링할 필요가 없다. 비디오 디코더 (30) 는 마찬가지로 이러한 선택스를 디코딩하는 것을 스킵할 수도 있고, 이것은 비트레이트 및 코딩 효율을 개선시킬 수도

있다.

[0117] 대안적인 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 크기를 규범적 방식으로 적어도 하나인 것으로 한정할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 (팔레트 크기 - 1) 이 시그널링되도록 팔레트 크기의 시그널링을 수정하도록 구성될 수도 있다. 예를 들어, 팔레트 예측자가 팔레트 예측자의 각각의 엔트리에 대하여, (예컨대, 도 4 의 예에 대하여 더욱 상세하게 설명된 바와 같이) 이용될 때, 비디오 인코더 (20) 는 개개의 팔레트 예측자 엔트리들이 현재의 블록의 팔레트 내에 포함되는지 여부를 표시하기 위하여 1 비트 플래그를 인코딩할 수도 있다. 이 엔트리들은 예측된 팔레트 엔트리들로서 지칭되고, 팔레트 예측 2 진 벡터 (예컨대, 1 비트 플래그들의 스트링) 에 의해 표시된다. 비디오 인코더 (20) 는 또한, 예측된 엔트리들을 후행하는 새로운 팔레트 엔트리들의 수를 시그널링할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 예측된 엔트리들 이전에 새로운 팔레트 엔트리들의 수를 시그널링할 수도 있다. 여하튼, 예측된 팔레트 엔트리들의 수가 제로일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 새로운 팔레트 엔트리들의 수를 코딩하는 대신에, (새로운 팔레트 엔트리들의 수 - 1) 을 표시하는 데이터를 코딩하도록 구성될 수도 있다.

[0118] 또 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 크기가 0 와 동일하지 않도록 팔레트 모드를 한정하도록 구성될 수도 있다. 예를 들어, 이 한정은 비트스트림 제약으로서 달성될 수 있고, 즉, 비트스트림은 제로와 동일한 팔레트 크기를 갖는 팔레트 코딩된 블록을 포함하지 않을 수 있다.

[0119] 이 개시물의 기법들은 또한 팔레트 크기를 시그널링하는 것에 관한 것이다. 예를 들어, 현재의 블록 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30) 에 의해 현재 코딩되고 있는 CU) 에 대한 팔레트 크기는 (예컨대, 2014 년 4 월 3 일자로 출원된 미국 출원 제 14/244,688 호, 및 2014 년 4 월 3 일자로 출원된 미국 출원 제 14/244,711 호에서 예를 들어, 개시된 바와 같이) 명시적으로 시그널링될 수도 있다. 이러한 예들에서, 팔레트 크기는 (예컨대, 비트스트림에서 명시적으로 시그널링된 바와 같이) (예컨대, 팔레트 예측자를 이용하여 결정된) 예측된 팔레트 엔트리들 및 새로운 팔레트 엔트리들의 양자를 포함한다.

[0120] 이 개시물의 양태들에 따르면, 팔레트 크기가 시그널링될 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 예측된 엔트리들의 수 및 팔레트 크기로부터 블록에 대한 새로운 팔레트 엔트리들의 수를 유도하도록 구성될 수도 있으므로 (예컨대, 팔레트 크기 - 예측된 엔트리들의 수 = 새로운 엔트리들의 수), 새로운 엔트리들의 수를 시그널링하기 위한 필요성이 없을 수도 있다. 게다가, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 팔레트 크기가 시그널링되고 그 시그널링된 크기 수가 현재의 블록에 대한 팔레트를 구성할 때에 도달될 때, 이전의 팔레트들의 엔트리들의 예측을 종결시키도록 구성될 수도 있다.

[0121] 일부 예들에서, 팔레트 크기는 이전의 팔레트들로부터 예측될 수도 있고, 비디오 인코더 (20) 는 오직 차이를 시그널링하도록 구성될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 지수 골롬 (exponential Golomb), 절단된 1 진 (truncated unary), 또는 고정 길이 코드를 이용하여 블록에 대한 팔레트 크기와 예측된 팔레트 크기 사이의 차이를 코딩하도록 구성될 수도 있다. 일부 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예측을, 코딩되고 있는 블록 크기에 종속 (예컨대, 기초함) 하게 하도록 구성될 수도 있다. 예를 들어, 8x8 블록에 대하여, 팔레트 크기는 팔레트 모드를 이용하여 코딩된 최신 8x8 블록 (예컨대, 현재의 블록 이전에 스캔 순서에서 가장 최근에 코딩된 8x8) 과 연관된 팔레트로부터 예측될 수도 있다. 마찬가지로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이전에 코딩된 16x16 블록으로부터의 팔레트에 기초하여 16x16 블록에 대한 팔레트 크기를 예측하도록 구성될 수도 있고, 유사한 관계는 다른 크기들의 블록들로 확장될 수도 있다. 대안적으로, 또 다른 예에서, 팔레트 크기는 현재의 블록 크기 이하보다 더 작거나 동일한 크기로 코딩된 최신 블록으로부터 예측될 수도 있다.

[0122] 이 개시물의 기법들은 또한, 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 시그널링하는 것에 관한 것이다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 SPS 로부터 이러한 데이터를 코딩하도록 구성될 수도 있다. 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하는 것은 신축성을 제공할 수도 있어서, 예컨대, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 상이한 프로파일들, 레벨들, 비트-심도들, 블록 크기들 등에 대한 상이한 크기들의 팔레트들 및 팔레트 예측자들을 이용하도록 할 수도 있다. 비디오 코딩 표준의 맥락에서, 프로파일은 알고리즘들, 특징들, 또는 툴들과, 이들에 적용되는 제약들의 서브세트 (subset) 에 대응할 수도 있다. 예를 들어, 프로파일은 특정한 것에 의해 특정되는 전체 비트스트림 선택스의 서브세트일 수도 있다. 레벨은 픽처들의 해상도, 비트레이트, 및 블록 프로세싱 레이트에 관련될 수도

있는, 예를 들어, 디코더 메모리 및 연산과 같은 디코더 자원 소비의 제한들에 대응할 수도 있다. 프로파일은 `profile_idc` (프로파일 표시자) 값으로 시그널링될 수도 있는 반면, 레벨은 `level_idc` (레벨 표시자) 값으로 시그널링될 수도 있다.

[0123] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예를 들어, 새로운 팔레트 엔트리들의 수를 시그널링할 시에, 팔레트 모드와 연관된 엘리먼트들 및 플래그들을 결정하기 위하여 최대 팔레트 크기에 관한 정보를 이용하도록 구성될 수도 있다. 예로서, 최대 가능한 팔레트 크기는, 비디오 인코더 (20) 에 의해 인코딩될 수도 있고 비디오 디코더 (30) 에 의해 디코딩될 수도 있는 `MAX_PLT_SIZE` 에 의해 나타내어질 수도 있다. 유사하게, 팔레트 예측자 벡터의 최대 가능한 크기는, 비디오 인코더 (20) 에 의해 인코딩될 수도 있고 비디오 디코더 (30) 에 의해 디코딩될 수도 있는 `MAX_PLT_PREDICTOR_SIZE` 에 의해 나타내어질 수도 있다.

[0124] 또 다른 예로서, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 (예컨대, 현재의 블록을 코딩하기 위한 팔레트로 복사되고 있는 팔레트 예측자로부터의 엔트리들의 수를 표현할 수도 있는) 팔레트 예측 2 진 벡터에서의 "1 들" 의 수를 표시하는 데이터를 코딩할 수도 있다. 일부 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예측된 팔레트 엔트리들의 수를 표시하기 위하여 선택스 엘리먼트 `numPredPalette` 를 코딩하도록 구성될 수도 있다. `numPredPalette` 의 값이 `MAX_PLT_SIZE` (즉, 최대 팔레트 크기) 의 값과 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 새로운 팔레트 엔트리들의 수의 코딩을 스킵하도록 구성될 수도 있다. 이와 다르게, `numPredPalette` 의 값이 `MAX_PLT_SIZE` 의 값보다 더 작을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 새로운 엔트리들의 수를 표시하는 데이터를 코딩하기 위하여, 새로운 팔레트 엔트리들의 수에 대한 최대 가능한 값인 (`MAX_PLT_SIZE - numPredPalette`) 에 기초하여 절단된 2 진화를 이용할 수도 있다.

[0125] 일반적으로, 절단된 2 진화는 고유의 디코딩가능성을 유지하면서 파라미터의 2 진화 방법에서 이용된 일부 코드워드들의 길이를 감소시킴으로써, (예컨대, 새로운 팔레트 엔트리들의 수와 같은) 시그널링되고 있는 특정한 파라미터의 최대 가능한 값에 대한 정보를 이용하는 임의의 기법을 포함할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 (예컨대, 새로운 팔레트 엔트리들의 수와 같은) 주어진 파라미터의 최대 값을 이용하여 절단된 2 진 코드를 구성하도록 구성될 수도 있다. 절단된 2 진 코딩을 위한 일 예의 기법들은 http://en.wikipedia.org/wiki/Truncated_binary_encoding 에서 설명되어 있다.

[0126] 유사하게, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 새로운 팔레트 엔트리들의 수의 최대 가능한 값에 기초하여, 새로운 팔레트 엔트리들의 수를 시그널링하고 디코딩하기 위하여 절단된 1 진 또는 지수 골롬 또는 골롬-라이스 (Golomb-Rice) 코드들을 이용할 수도 있다. 예를 들어, (`MAX_PLT_SIZE - numPredPalette`) = 3 일 경우, 비디오 인코더 (20) 는 (예컨대, 규칙적인 1 진 코드를 이용할 때에 시그널링되는 바와 같이) 0001 대신에 3 개를 000 로서 시그널링하기 위하여 절단된 1 진 코드를 이용할 수도 있다. 절단된 지수 골롬 또는 골롬-라이스 코드들의 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 값을 포함하는 간격에 대한 프리픽스 (prefix) 의 길이를 1 만큼 감소시키도록 구성될 수도 있다. 이에 따라, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 프리픽스를 000 ... 001 로부터 000 ... 000 로 변경하도록 구성될 수도 있다. 유사하게, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 값에 따라 그 간격에 대한 2 진화 방법에서 서픽스 (suffix) 비트들의 수를 감소시키도록 구성될 수도 있다.

[0127] 큰 블록들 (및/또는 큰 CU 들) 에 대하여, 팔레트 크기는 최대 팔레트 크기인 경향이 있다. 그러므로, 일부 경우들에는, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 통상적인 방법의 반대로 (`MAX_PLT_SIZE - numPredPalette`) 의 2 진화를 맵핑하도록 구성될 수도 있고, 즉, 더 짧은 코드워드 길이들은 (`MAX_PLT_SIZE - numPredPalette`) 의 더 큰 값들로 배정되고, 더 긴 코드워드 길이들은 (`MAX_PLT_SIZE - numPredPalette`) 의 더 작은 값들로 배정된다. 일부 예들에서, 1 진/절단된 1 진 코드들을 시그널링하기 위하여, 또는 골롬-라이스 또는 지수 골롬 또는 연결된 골롬-라이스 및 지수 골롬 계열의 코드들의 프리픽스로서, 0 과, 그 다음의 1 을 이용하는 대신에, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 1 과, 그 다음의 0 을 이용하도록 구성될 수도 있다.

[0128] 또한, 다른 변형들이 가능하다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이러한 코드들에서의 최초 비트를, 새로운 엔트리들의 수가 제로 또는 비-제로 (non-zero) 인지 여부를 표시하기 위한 플래그로서 해독하도록 구성될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 나머지 비트들을 새로운 팔레트 엔트리들의 수 마이너스 1 로서 해독하도록 구성될 수도 있다. 예시의 목적들을 위한 예에서, 새로

은 팔레트 엔트리들의 최대 값은 8 일 수도 있고, 새로운 팔레트 엔트리들의 수는 3 일 수도 있다. 절단된 1 진 코드를 이용하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 2 진화를 0001 인 것으로 결정하도록 구성될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 가 최초 비트를 플래그 (예컨대, 0: 하나 이상의 새로운 팔레트 엔트리들, 1: 제로의 새로운 팔레트 엔트리들) 로서 해독하도록 구성될 경우, 비트들의 나머지 (001) 는 2 개의 새로운 팔레트 엔트리들이 있다는 것을 표시한다. 절단된 코드들을 이용할 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 값을 1 만큼 아래로 조절하도록 구성될 수도 있다.

[0129] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 상기 설명된 플래그를 반대로 해독하도록 구성될 수도 있다. 이 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 1 의 플래그 값을 하나 이상의 새로운 팔레트 엔트리들로서, 그리고 0 의 플래그 값을 제로의 새로운 팔레트 엔트리들로서 해독하도록 구성될 수도 있다. 이러한 경우, 3 개의 새로운 팔레트 엔트리들을 8 의 최대 값으로 시그널링하기 위한 비트들은 1001 이다.

[0130] 다른 예들에서, 상기 설명된 플래그의 개념은 지수 골롬, 골롬-라이스 등과 같은 다른 코드들로 확장될 수도 있다. 예를 들어, 새로운 팔레트 엔트리들에 대한 최대 값이 비-제로일 때, 비디오 인코더 (20) 는 비-제로의 새로운 엔트리들이 있는지 여부를 표시하는 플래그를 시그널링하도록 구성될 수도 있다. 플래그가 비-제로의 새로운 엔트리들이 있다는 것을 표시할 경우, 새로운 엔트리들의 수 마이너스 1 은 지수 골롬, 골롬-라이스, 지수 골롬 및 골롬-라이스의 연결, 또는 유사한 코드들 또는 그 절단된 버전들을 이용하여 시그널링될 수도 있다. 절단된 버전들이 이용될 때, 최대 값은 1 만큼 아래로 조절될 수도 있다.

[0131] 일부 예들에서, 플래그는 CABAC 를 이용하여 컨텍스트-코딩될 수도 있는 반면, (예컨대, 새로운 팔레트 엔트들 마이너스 1 에 대한) 빈들의 나머지는 우회 코딩될 수도 있다. 대안적으로, 플래그 뿐만 아니라 (새로운 팔레트 엔트리들 마이너스 1 에 대한) 빈들의 나머지는 모두 우회 코딩될 수도 있다. 일부 사례들에서, 새로운 팔레트 엔트리들 마이너스 1 에 대한 코드로부터의 프리픽스 빈들의 고정된 수는 CABAC 를 이용하여 컨텍스트-코딩될 수도 있고, 빈들의 나머지는 우회 코딩될 수도 있다.

[0132] 이 개시물의 양태들에 따르면, 위에서 언급된 바와 같이, 선택스 엘리먼트 MAX_PLT_SIZE 는 SPS 와 같은 파라미터 세트에서 시그널링될 수도 있다. 다른 예들에서, 선택스 엘리먼트 MAX_PLT_SIZE 는 VPS, 픽처 파라미터 세트 (picture parameter set; PPS), 슬라이스 헤더에서, (예컨대, LCU 또는 CU 에 대해 시그널링된 선택스로) 블록 레벨에서, 또는 다른 곳에서 시그널링될 수도 있다. 일부 예들에서, 이 개시물의 양태들에 따르면, 상이한 최대 팔레트 크기들이 상이한 블록 크기들에 대하여 특정될 수도 있다. 다른 예들에서, 최대 팔레트 크기는 코딩되고 있는 비디오 데이터의 프로파일 또는 비트-심도에 종속될 수도 있다. 예를 들어, 더 큰 입력 비트-심도 (또는 프로파일 비트-심도) 에 대하여, 선택스 엘리먼트 MAX_PLT_SIZE 는 상대적으로 더 큰 최대 팔레트 크기를 특정하기 위하여 이용될 수도 있다. 또 다른 예들에서, 최대 팔레트 크기는 추가적으로 또는 대안적으로, 코딩되고 있는 비디오 데이터의 크로마 포맷에 종속될 수도 있다. 예를 들어, 선택스 엘리먼트 MAX_PLT_SIZE 는, 궁극적으로, 4:4:4 크로마 서브-샘플링 포맷된 입력들보다 더 작은 크기들을 가질 수도 있는, 4:2:0 크로마 서브-샘플링 포맷들에 대한 것보다 단색 입력들에 대하여 상대적으로 더 작은 최대 팔레트 크기를 특정하기 위하여 이용될 수도 있다.

[0133] 이 개시물의 양태들에 따르면, 위에서 설명된 방식으로 선택스 엘리먼트 MAX_PLT_SIZE 를 시그널링하는 대신에, 제로와 동일한 MAX_PLT_SIZE 선택스 엘리먼트가 팔레트를 완전히 디스에이블하는 것으로 인해 무효할 수도 있으므로, (MAX_PLT_SIZE-1) 을 표시하는 데이터가 시그널링될 수도 있다.

[0134] 또 다른 예에서, 팔레트 모드를 인에이블/디스에이블하기 위하여 VPS, SPS, PPS, 또는 슬라이스 레벨에서 별도의 플래그를 시그널링하는 대신에, 비디오 인코더 (20) 는 MAX_PLT_SIZE 선택스 엘리먼트를 오직 시그널링하도록 구성될 수도 있다. 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 0 의 MAX_PLT_SIZE 선택스 엘리먼트를 팔레트 모드를 디스에이블하는 것으로서 해독하도록 구성될 수도 있다. 즉, 팔레트 크기 선택스 엘리먼트 (예컨대, MAX_PLT_SIZE 선택스 엘리먼트) 를 수신할 시에, 비디오 디코더 (30) 는 선택스 엘리먼트에 기초하여 팔레트 모드가 디스에이블된 것으로 결정할 수도 있다. MAX_PLT_SIZE 선택스 엘리먼트 또는 (MAX_PLT_SIZE - 1) 은 고정 길이 코드들 (MAX_PLT_SIZE 에 대한 규범적 제한을 가정함) 또는 골롬-라이스 또는 지수 골롬 코드들을 이용하여 시그널링될 수도 있다.

[0135] 위에서 언급된 바와 같이, 이 개시물의 기법들은 또한, 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하는 것을 포함한다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 예측자 크기를 표시하는 VPS, SPS, PPS, 슬라이스 헤더에서, 블록 레벨에서, 또는 다른 곳에서

MAX_PLT_PREDICTOR_SIZE 선택스 엘리먼트를 코딩하도록 구성될 수도 있다. 일부 예들에서, MAX_PLT_PREDICTOR_SIZE 선택스 엘리먼트를 시그널링하는 대신에, (MAX_PLT_PREDICTOR_SIZE-1) 이 시그널링될 수도 있다. 또 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 예측자 크기를 표시하는 다른 데이터를 코딩하도록 구성될 수도 있다.

[0136] 본원에서 설명된 특정한 예들에서, MAX_PLT_PREDICTOR_SIZE 선택스 엘리먼트 또는 (MAX_PLT_PREDICTOR_SIZE - 1) 은 고정 길이 코드들 (예컨대, MAX_PLT_PREDICTOR_SIZE 에 대한 규범적 제한을 가정함) 또는 곱셈 라이스 또는 지수 곱셈 코드들을 이용하여 시그널링될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MAX_PLT_PREDICTOR_SIZE 선택스 엘리먼트에 의해 표시된 크기가 (예컨대, MAX_PLT_SIZE 선택스 엘리먼트에 의해 표시된 바와 같이) 최대 팔레트 크기 이상인 것으로 가정 (예컨대, 자동으로 결정함) 하도록 구성될 수도 있다. 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 고정 길이 코드들 또는 곱셈-라이스 또는 지수 곱셈 코드들을 이용하여 (MAX_PLT_PREDICTOR_SIZE ? MAX_PLT_SIZE) 를 코딩하도록 구성될 수도 있다. 따라서, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 예측자 크기와 최대 팔레트 크기 사이의 델타 (예컨대, 차이) 를 표시하는 데이터를 코딩하도록 구성될 수도 있다.

[0137] 최대 팔레트 크기 및 최대 팔레트 예측자 크기가 SPS 레벨에서 시그널링되는 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 절단된 1 진 코드를 이용하여 새로운 엔트리들의 수를 표시하는 데이터를 코딩하도록 구성될 수도 있다. 새로운 엔트리들의 수 플러스 (plus) 팔레트 예측자로부터 예측된 엔트리들의 수는 모두 함께 SPS 에서 시그널링된 최대 팔레트 크기를 초과하지 않을 수도 있다. 그러나, SPS 에서 시그널링된 최대 팔레트 크기가 상대적으로 클 경우, 새로운 엔트리들의 수는 31 을 초과할 수도 있다. 이 사례에서, 절단된 1 진 코드는 32-비트 길이를 초과하고, 이것은 소프트웨어 및 하드웨어 구현예들을 위하여 바람직하지 않을 수도 있다.

[0138] 이것을 해결하기 위하여, 이 개시물의 양태들에 따르면, 하나의 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 새로운 엔트리들의 수를 시그널링하기 위한 코드의 길이가 32 를 초과하지 않도록 새로운 엔트리들의 수를 한정하도록 구성될 수도 있다. 예를 들어, 1 진 또는 절단된 1 진 코드가 새로운 엔트리들의 수를 시그널링하기 위하여 이용될 경우, 새로운 엔트리들의 수는 31 로 한정될 수도 있다. 32 의 길이 한정은 단지 하나의 예 (예컨대, 다른 길이 한정들이 대안적으로 이용될 수도 있음) 라는 것을 이해해야 한다.

[0139] HEVC 스크린 콘텐츠 코딩 확장들 텍스트 사양 초안 2 (Rajan Joshi 등, "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 2" JCTVC-S1005, Sapporo, JP, 2014 년 6 월 30 일 - 2014 년 7 월 9 일 (이하, JCTVC-S1005)) 에서는, 절단된 1 진 코드가 SPS 에서 시그널링된 최대 팔레트 크기 (palette_max_size) 마이너스 팔레트 예측자로부터 예측되는 팔레트 엔트리들의 수와 동일한 최대 값을 갖는 새로운 팔레트 엔트리들의 수를 시그널링하기 위하여 이용된다. 제안된 한정을 이용하여, 최대 값은 32 와, SPS 에서 시그널링된 최대 팔레트 크기 (palette_max_size) 및 팔레트 예측자로부터 예측된 팔레트 엔트리들의 수 사이의 차이 중의 더 작은 것이 되도록 수정될 수도 있다. 최대 값에 대한 이러한 수정이 수행되고 JCTVC-S1005 의 절단된 1 진 코딩이 이용될 경우, 새로운 팔레트 엔트리들의 최대 수는 코드의 길이가 32 비트들을 초과하지 않으면서, (31 대신에) 32 일 수도 있다.

[0140] 일부 예들에서, 절단된 1 진 코딩 대신에, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 지수 곱셈 또는 그 절단된 버전과 같은 또 다른 코드를 이용하도록 구성될 경우, 최대 허용가능한 새로운 팔레트 엔트리들은 길이가 32 를 초과하지 않도록 적절하게 수정될 수도 있다. 32 의 코드워드 길이를 가지는 다수의 값들이 있을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이러한 값의 최고치를, 새로운 팔레트 엔트리들의 수에 대한 최대 허용가능한 값인 것으로 선택하도록 구성될 수도 있다.

[0141] 최대 수의 새로운 팔레트 엔트리들에 대해 본원에서 설명된 한정은 규범적인 한정으로 행해질 수도 있다. 예를 들어, 비디오 인코더 (20) 는 제약을 갖는 비트스트림을 생성하도록 구성될 수도 있고, 비디오 디코더 (30) 는 준수하는 비트스트림에서 제약에 의존하도록 구성될 수도 있다.

[0142] 이 개시물의 양태들에 따르면, 하나의 예에서, palette_num_signaled_entries 선택스 엘리먼트의 시맨틱은 다음과 같이 JCTVC-S1005 에 관련하여 변경될 수도 있다: 선택스 엘리먼트 palette_num_signaled_entries 는 명시적으로 시그널링되는 현재의 팔레트에서의 엔트리들의 수를 특정한다. 선택스 엘리먼트 palette_num_signaled_entries 의 값은 0 내지 31 까지의 범위에 있을 것이다. 선택스 엘리먼트 palette_num_signaled_entries 가 존재하지 않을 때, 그것은 0 과 동일한 것으로 추론된다.

- [0143] 게다가, 변수 CurrentPaletteSize 의 값은 현재의 팔레트의 크기를 특정하고, 다음과 같이 유도된다:
- [0144] $\text{palette_share_flag} [\text{ x0 }] [\text{ y0 }]$ 가 1 과 동일할 경우,
- [0145] $\text{CurrentPaletteSize} = \text{PreviousPaletteSize} \quad (7-71)$
- [0146] 이와 다를 경우 ($\text{palette_share_flag} [\text{ x0 }] [\text{ y0 }]$ 가 0 과 동일함)
- [0147] $\text{CurrentPaletteSize} = \text{paletteNumPredictedEntries} + \text{palette_num_signaled_entries} \quad (7-72)$
- [0148] 상기 예에서, CurrentPaletteSize 의 값은 0 내지 palette_max_size 까지의 범위에 있을 것이다.
- [0149] 이 개시물의 양태들에 따르면, 최대 값이 상기 설명된 방식으로 수정될 경우, 선택스 엘리먼트 palette_num_signaled_entries 의 값은 값이 0 내지 32 까지의 범위에 있도록 수정될 수도 있다.
- [0150] 또 다른 예에서, 최대 팔레트 크기는 SPS 에서 시그널링될 수도 있고 32 로 제한될 수도 있다. 크기를 제한하는 것은 palette_max_size 선택스 엘리먼트가 최대 허용된 팔레트 크기를 특정하도록, palette_max_size 선택스 엘리먼트에서 palette_max_size 선택스 엘리먼트에 대한 상한을 강제함으로써 달성될 수도 있다. palette_max_size 선택스 엘리먼트의 값은 0 내지 31 까지의 범위에 있을 것이다. 존재하지 않을 때, palette_max_size 선택스 엘리먼트의 값은 0 인 것으로 추론된다. 일부 예들에서, 31 대신에, 값은 32 로 한정될 수도 있다.
- [0151] 또 다른 예에서, palette_max_size 선택스 엘리먼트의 최대 값은, 새로운 팔레트 엔트리들의 수가 palette_max_size 와 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 32 비트들을 초과하지 않는 코드를 이용하여 새로운 팔레트 엔트리들의 수를 코딩하도록 구성될 수도 있도록 한정될 수도 있다. 또 다른 예에서, 새로운 팔레트 엔트리들의 수의 최대 값은 최대 값을 코딩하기 위하여 이용된 코드에 관계 없이, 항상 31 로 제한될 수도 있다. 또 다른 예에서, 최대 값은 32 로 제한될 수도 있다.
- [0152] 이 개시물의 기법들은 또한, (예컨대, CU 또는 LCU 에 대한) 블록-레벨 탈출 시그널링에 관한 것이다. 예를 들어, 이 개시물의 양태들에 따르면, 하나 이상의 선택스 엘리먼트들은 블록-레벨 (예컨대, CU 레벨) 에서, 블록의 샘플들 중의 임의의 것이 탈출 샘플 (예컨대, 블록을 코딩하기 위한 팔레트에서 표현된 컬러 값을 가지지 않는 샘플) 로서 코딩되는지 여부를 표시할 수도 있다. 위에서 언급된 바와 같이, 하나 이상의 선택스 엘리먼트들은 블록-레벨 탈출 선택스로서 지칭될 수도 있다. 또한, 블록-레벨 선택스는, 슬라이스 헤더 내에, 또는 비디오 데이터의 개별적인 픽셀들과 함께 포함될 수도 있는 선택스가 아니라, CU 또는 LCU 와 같은, 비디오 데이터의 블록으로 코딩되거나 결정되는 선택스를 지칭할 수도 있다.
- [0153] 팔레트 코딩을 이용하여 코딩된 샘플들의 블록에서의 적어도 하나의 샘플이 탈출 샘플로서 코딩되는 사례들에서, 이 개시물의 기법들은 이러한 모드의 존재를 시그널링하기 위하여 이용될 수도 있다. 예시의 목적들을 위한 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 코딩되고 있는 블록의 샘플들 중의 임의의 것이 탈출 샘플로서 코딩되는지 여부를 표시하는 (블록-레벨 탈출 플래그로서 지칭될 수도 있는) 플래그를 코딩하도록 구성될 수도 있다. 일부 사례들에서, 제로의 플래그 값은 블록의 샘플들 또는 픽셀들 중의 어느 것도 탈출 샘플들로서 코딩되지 않는다는 것을 표시할 수도 있다. 1 의 플래그 값은 블록의 적어도 하나의 샘플 또는 픽셀이 탈출 샘플로서 코딩된다는 것을 표시할 수도 있다. 이 때문에, 블록-레벨 탈출 선택스는 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 샘플이 블록에 대한 컬러 값들의 팔레트에 대한 인덱스를 이용하지 않으면서 코딩되는지, 예컨대, 탈출 모드를 이용하여 코딩되는지 여부를 표시할 수도 있다.
- [0154] 이 개시물의 양태들에 따르면, 상기 설명된 선택스는 블록-레벨 표시 없이 탈출 샘플들을 시그널링하는 기법들에 비해 비트 절감을 달성할 수도 있다. 예를 들어, 선택스가 블록의 샘플들이 탈출 샘플들로서 코딩되지 않는다는 것을 표시하는 사례들에서 (예컨대, 상기 설명된 플래그가 제로임), 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록에 대한 탈출 샘플들과 연관된 임의의 다른 선택스를 코딩하지 않을 수도 있다. 예를 들어, 본원에서 설명된 명시적 탈출 시그널링에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 샘플-레벨 탈출 모드 플래그들의 코딩을 스킵할 수도 있다. 묵시적 탈출 시그널링에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 탈출 샘플을 표시하는 팔레트에 대한 추가적인 인덱스의 코딩을 스킵할 수도 있다. 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CopyFromTop 모드와 값 모드 사이를 구별하기 위하여 SPoint 플래그를 오직 코딩할 수도 있다.
- [0155] 일부 예들에서, 상기 설명된 플래그는 현재 코딩되고 있는 블록 또는 CU 에 대한 팔레트 엔트리들 전에 시그널

링될 수도 있다. 다른 예들에서, 상기 설명된 플래그는 현재 코딩되고 있는 블록 또는 CU 에 대한 팔레트 엔트리들 후에 시그널링될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 상기 설명된 플래그를 컨텍스트 코딩하도록 구성될 수도 있다. 이러한 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록 또는 CU 크기 및/또는 현재의 블록 또는 CU 에 대한 팔레트 크기에 기초하여 컨텍스트들을 결정할 수도 있다.

[0156] 일부 사례들에서, 탈출 샘플들의 사용은 블록 크기에 의해 변동될 수도 있다. 예를 들어, 탈출 샘플들의 이용은 상대적으로 작은 블록들에서 덜 일반적일 수도 있다. 이러한 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록이 임의의 탈출 샘플들을 포함하지 않는 것으로 결정하고, 상기 설명된 플래그의 코딩을 스킵함으로써 비트 절감을 달성하도록 구성될 수도 있다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 8x8 블록들에 대하여 블록-레벨 플래그를 코딩하지 않을 수도 있고, 여기서, 탈출 샘플들은 더 큰 블록 크기들에서보다 이용될 가능성이 훨씬 더 적다. 유사하게, 큰 블록 크기들 (예컨대, 64x64 픽셀들 이상의 블록들) 에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 탈출 샘플들로서 코딩된 샘플들이 항상 있는 것으로 결정하도록 구성될 수도 있다. 이러한 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록에 대한 블록-레벨 탈출 플래그가 1 과 동일하다는 것 (예컨대, 적어도 하나의 샘플이 탈출 샘플임) 을 추론할 수도 있고, 블록-레벨 탈출 플래그의 코딩을 스킵할 수도 있다 (예컨대, 블록-레벨 탈출 플래그의 표시가 비트스트림 내에 포함되지 않음).

[0157] 이 개시물의 기법들은 또한, 블록의 임의의 샘플들이 탈출 샘플들로서 코딩되는지 여부에 기초하여 샘플들의 블록을 코딩하는 것에 관한 것이다. 예를 들어, 위에서 언급된 바와 같이, 이 개시물의 기법들은 임의의 샘플들이 팔레트-코딩된 블록의 탈출 샘플들로서 코딩되는지 여부를 표시하기 위하여 이용될 수도 있다. 블록이 탈출 샘플들을 포함하지 않는 사례들에서, 그리고 팔레트의 크기가 1 일 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록의 모든 샘플들이 동일한 인덱스 값 (예컨대, 오직 팔레트의 엔트리) 을 가지는 것으로 자동으로 결정하도록 구성될 수도 있다. 그러므로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 블록의 팔레트 인덱스들을 결정하기 위하여 이용된 다른 모든 다른 데이터의 코딩을 스킵할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 SPoint 플래그, 인덱스 시그널링, 및 팔레트 인덱스들의 런들과 연관된 데이터의 코딩을 스킵할 수도 있다.

[0158] 예시의 목적을 위한 예에서, 비디오 디코더 (30) 는 탈출 샘플들로서 코딩되는 현재의 블록에서의 샘플들이 없다는 것을 표시하는 블록-레벨 탈출 플래그 (예컨대, 플래그는 제로와 동일함) 를 디코딩할 수도 있다. 비디오 디코더 (30) 는 또한, 블록에 대한 팔레트가 단일 엔트리를 가진다는 것을 표시하는 데이터 (예컨대, 팔레트 크기가 1 이라는 것을 표시하는 데이터) 를 디코딩할 수도 있거나, 단일 엔트리를 가지는 팔레트를 디코딩할 수도 있다. 이 예에서는, 양자의 조건들을 참으로 평가하는 것 (예컨대, 샘플들은 탈출 샘플들이고 팔레트 크기는 1 임) 에 기초하여, 비디오 디코더 (30) 는 블록의 팔레트 인덱스들의 전부가 팔레트 내에 포함된 단일 엔트리와 동일한 것으로 자동으로 결정할 수도 있다. 비디오 디코더 (30) 는 또한, (예컨대, SPoint 플래그들, 팔레트 인덱스들, 및 런 정보와 같은) 블록의 팔레트 인덱스들을 결정하기 위하여 이용된 다른 데이터의 디코딩을 스킵할 수도 있다.

[0159] 또 다른 예에서, 이 개시물의 양태들에 따르면, 팔레트 크기가 1 일 때, 팔레트 인덱스 제로로 코딩된 샘플들의 런들은 탈출 샘플들에 의해 종결될 수도 있다. 즉, 팔레트 인덱스들의 런은 탈출 샘플로서 코딩되고 있는 위치에 의해 차단될 수도 있다. 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 SPoint 플래그의 코딩을 스킵하도록 구성될 수도 있다. 게다가, 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 인덱스들에 대한 모드가 값 모드뿐만 아니라 값 모드의 인덱스인 것을 추론하도록 구성될 수도 있다 (예컨대, 팔레트에서의 오직 하나의 엔트리로, 값 모드에 대한 인덱스를 시그널링하는 것이 필요하지 않을 수도 있음). 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 런에 바로 후행하는 샘플이 탈출 샘플로서 코딩된다는 것을 추론할 수도 있고, 탈출 관련된 인덱스의 코딩을 스킵할 수도 있다.

[0160] 이 개시물의 기법들은 또한, 팔레트 코딩에서 팔레트 인덱스들의 런의 런 값을 표시하는 데이터를 코딩하는 것에 관한 것이다. 예를 들어, 위에서 언급된 바와 같이, 런 값은 함께 코딩되는 팔레트-코딩된 블록에서 특정한 스캔 순서에서의 연속적인 샘플들 (예컨대, 샘플들의 런) 의 수를 표시할 수도 있다. 일부 사례들에서는, 런의 각각의 샘플이 팔레트로의 연관된 인덱스를 가지기 때문에, 샘플들의 런은 또한, 팔레트 인덱스들의 런으로서 지칭될 수도 있다.

[0161] 런 값은 동일한 팔레트-코딩 모드를 이용하여 코딩되는 팔레트 인덱스들의 런을 표시할 수도 있다. 예를 들

어, 값 모드에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인덱스 값과, 동일한 인덱스 값을 가지며 인덱스 값으로 코딩되고 있는 스캔 순서에서의 연속적인 샘플들의 수를 표시하는 런 값을 코딩할 수도 있다.

CopyFromTop 모드에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재의 샘플 값에 대한 인덱스가 상부-이웃 샘플 (예컨대, 블록에서 현재 코딩되고 있는 샘플의 상부에 위치되는 샘플) 의 인덱스에 기초하여 복사된다는 표시와, 상부-이웃 샘플로부터 인덱스 값을 또한 복사하며 인덱스 값으로 코딩되고 있는 스캔 순서에서의 연속적인 샘플들의 수를 표시하는 런 값을 코딩할 수도 있다.

[0162] 예를 들어, 이 개시물의 양태들에 따르면, 비디오 데이터의 블록에서 팔레트 인덱스들의 런을 표시하는 데이터는 블록에 대한 최대 가능한 런 값에 기초하여 코딩될 수도 있다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 픽셀의 값을 팔레트에서의 컬러 값에 관련시키는 팔레트 인덱스와 연관된 픽셀에 대하여, 픽셀의 팔레트 인덱스로 코딩되고 있는 팔레트 인덱스들의 런의 런 령스를 결정할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정할 수도 있다. 다음으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 코딩할 수도 있다.

[0163] 예시의 목적들을 위한 예에서, 비디오 데이터의 블록에서의 샘플들의 총 수는 N 일 수도 있고, 샘플들의 각각은 0 부터 (N-1) 까지 인덱싱될 수도 있다. 위치 j 를 갖는 샘플에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 가능한 런 값을 (N-j-1) 로서 결정할 수도 있다. 런 값은 현재의 샘플과 동일한 팔레트 코딩 모드 (예컨대, 값 모드 또는 CopyFromTop 모드) 로 코딩되고 있는 후속 샘플들의 수를 표시한다는 것에 주목해야 한다. 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 가능한 런 값을 참작하여, 절단된 2 진화를 이용하여 런 값을 표시하는 데이터를 코딩하도록 구성될 수도 있다. 일반적으로, 절단된 2 진화는 고유의 디코딩가능성을 유지하면서 파라미터의 2 진화 방법에서 이용된 일부 코드워드들의 길이를 감소시킴으로써, (예컨대, 새로운 팔레트 엔트리들의 수와 같은) 시그널링되고 있는 특정한 파라미터의 최대 가능한 값에 대한 정보를 이용하는 임의의 기법을 포함할 수도 있다. 예를 들어, 런의 최대 가능한 값에 기초한 절단된 2 진 코드가 이용될 수도 있다. 유사하게, 절단된 1 진 또는 지수 곱셈 또는 곱셈-라이트 코드들은 런의 최대 가능한 값에 기초하여, 런 값을 코딩하고 디코딩하기 위하여 이용될 수도 있다. 일부 예들에서, 절단된 2 진화는 지수 곱셈 및 곱셈-라이트 코드들의 조합일 수도 있다.

[0164] 예를 들어, k 번째 차수 지수-곱셈 (EGk) 코드 워드는 2 개의 부분들, 프리픽스 (prefix) 및 서픽스 (suffix) 로 구성된다. 주어진 무부호 정수 x 에 대하여, EGk 코드 워드의 프리픽스 부분은 다음의 값에 대응하는 1 진 코드로 구성된다:

[0165]
$$l(x) = \left\lfloor \log_2 \left(\frac{x}{2^k} + 1 \right) \right\rfloor$$

[0166] 서픽스 부분은 $k + l(x)$ 비트들을 이용한 $x - 2^k(2^{l(x)} - 1)$ 의 2 진 표현으로서 연산된다.

[0167] 예로서, 이하의 표 1 은 EGO 에 대한 몇몇 코드 워드들을 포함한다.

[0168] 표 1 - EGO 예

값 x	코드 워드 (프리픽스-서픽스)	코드 워드 길이
0	1	1
1	01-0	3
2	01-1	3
3	001-00	5
4	001-01	5
5	001-10	5
6	001-11	5

[0169]

[0170] 2014 년 6 월 20 일자로 출원된 미국 가출원 제 62/019,223 호에서는, 런 값이 제 2 차수 지수-곱셈 코드를 이

용하여 코딩된다.

[0171] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 절단된 지수-골롬 코드를 이용하여 런 값을 표시하는 데이터를 코딩하도록 구성될 수도 있다. 예를 들어, k 번째 차수 절단된 지수-골롬 (TEGk) 코드 워드는 또한, 2 개의 부분들, 프리픽스 및 서픽스로 구성된다. 프리픽스는 1 진 프리픽스일 수도 있고, 서픽스는 2 진 서픽스일 수도 있다. 예를 들어, 주어진 무부호 정수 x 및 그 최대 가능한 런 값 X_{max} (예컨대, 최대 런 령스) 에 대하여, EGk 코드 워드의 프리픽스 부분은 다음의 값에 대응하는 절단된 1 진 코드로 구성된다:

$$l(x) = \left\lceil \log_2 \left(\frac{x}{2^k} + 1 \right) \right\rceil$$

[0172]

[0173] 구체적으로, 1 진 코드의 "트레일링 1" 은 다음의 경우에 회피될 수 있다:

$$\left\lceil \log_2 \left(\frac{X_{max}}{2^k} + 1 \right) \right\rceil = l(x)$$

[0174]

[0175] 프리픽스가 절단될 경우, 즉, $\left\lceil \log_2 \left(\frac{x}{2^k} + 1 \right) \right\rceil = l(x)$, TEGk 의 서픽스 부분은 $k + l(x)$ 또는 $k + l(x) - 1$ 비트들을 이용한 $x - 2^k(2^{l(x)} - 1)$ 의 절단된 2 진 표현으로서 연산된다. 절단된 2 진 코드의 입력에 대한 최대 심볼 값은 $X_{max} - 2^k(2^{l(x)} - 1)$ 이다.

[0176] 프리픽스가 절단되지 않을 경우, TEGk 의 서픽스 부분은 EGk, 즉, $k + l(x)$ 비트들을 이용한 $x - 2^k(2^{l(x)} - 1)$ 의 2 진 표현과 동일하다. 예로서, 이하의 표 1 은 TEG0 에 대한 몇몇 코드 워드들을 포함한다.

[0177] 표 2 - TEG0 예들 (X=5)

값 x	코드 워드 (프리픽스-서픽스)	코드 워드 길이
0	1	1
1	01-0	3
2	01-1	3
3	00-0	3
4	00-01	4
5	00-10	4

[0178]

[0179] 표 2 의 예는 그 프리픽스를 다수의 제로들과, 그 다음으로 트레일링 1 (예컨대, 00...1_) 인 것으로서 예시하지만, 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다수의 1 들과, 그 다음으로 트레일링 제로 (예컨대, 11...0) 를 코딩할 수도 있다는 것을 이해해야 한다.

[0180] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 위에서 설명된 TEGk 코드를 이용하여 런 값을 코딩할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록 (또는 CU) 에서의 현재의 픽셀 위치에 대하여, 수학적 (X_{max} = 현재의 CU 에서의 픽셀들의 수 - 스캔 순서에서의 현재의 위치 - 1) 에 기초하여 최대 런 값 X_{max} 를 결정할 수도 있다.

[0181] 또 다른 예에서, 런 값이 절단된 1 진 프리픽스를 이용하여 먼저 코딩될 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예컨대, 절단된 값에 기초하여 이에 따라 최대 런 값을 조절하도록 구성될 수도 있다. 예를 들어, 일부 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 런 값을 일련의 3 개의 플래그들: 제로보다 더 크고, 1 보다 더 크고, 2 보다 더 큰 것으로서 코딩하도록 구성될 수도 있다. 이 예에서, 시그널링

된 런이 2 보다 더 클 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 지수 곱셈 및 곱셈-라이스 코드들의 조합 또는 위에서 설명된 TEGk 코드와 같은 또 다른 2 진화 방법을 잠재적으로 갖는 나머지 값 (예컨대, 런 값 -3) 을 코딩할 수도 있다.

[0182] 그러나, $(N-j-1)$ 이 0 과 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 런을 코딩하지 않는다. 마찬가지로, $(N-j-1)$ 이 1 과 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제로보다 더 큰 플래그를 오직 코딩할 수도 있다. 마찬가지로, $(N-j-1)$ 이 2 와 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제로보다 더 크고 1 보다 더 큰 플래그들을 오직 코딩할 수도 있다. 마찬가지로, $(N-j-1)$ 이 3 과 동일할 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제로보다 더 큰 플래그, 1 보다 더 큰 플래그, 및 2 보다 더 큰 플래그를 오직 코딩할 수도 있다. $(N-j-1)$ 이 제로보다 더 큰 플래그, 1 보다 더 큰 플래그, 및 2 보다 더 큰 플래그에 추가하여, 3 보다 더 클 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 $(N-j-4)$ 의 최대 값에 이르는 나머지 값을 코딩할 수도 있다. 유사한 방법으로, 설명된 프로세스는 3 이외의 다수의 플래그들, 예를 들어, 수 M 보다 더 큰 시그널링된 값을 표시하는 플래그들을 이용하기 위하여 확장될 수도 있고, 여기서, M 은 제로로부터 시작하는 비-음수 (non-negative) 값일 수도 있다.

[0183] 이 개시물의 양태들에 따르면, 상기 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 (현재의 CU 에서의 픽셀들의 수 - 스캔 순서에서의 현재의 위치 - 4) 와 동일한 최대 런 값을 갖는, 위에서 설명된 TEGk 코드를 이용하여 나머지 런 령스를 코딩할 수도 있다. 또 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 런 값이 제로보다 더 큰지 여부를 표시하는 플래그 및 나머지 값을 런 령스 마이너스 1 로서 코딩하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제로부터 더 큰 플래그를 코딩할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 최대 런 령스 마이너스 1 과 동일한 TEGk 코드 세트에 대한 최대 값을 갖는 TEGk 코드를 이용하여 런 령스 마이너스 1 을 표시하는 데이터를 코딩할 수도 있다. 하나의 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 k 를 제로와 동일하게 설정할 수도 있어서, TEGk 코드는 TEG0 코드이다.

[0184] 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 코딩을 위한 선택스 엘리먼트들을 코딩하기 위한 상기 설명된 TEG 코드의 임의의 차수를 이용할 수도 있다. 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 k 를 2 와 동일하게 설정할 수도 있어서, TEGk 코드는 TEG2 코드이다.

[0185] 상기 예들은 팔레트 코딩에서의 런 값을 코딩하는 것에 대하여 설명되지만, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 코딩을 위한 다른 선택스를 코딩하기 위하여 (TEGk 코드와 같은) 코드들을 이용하도록 구성될 수도 있다. 예를 들어, 이하에서 더욱 상세하게 설명된 바와 같이, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 2 진 팔레트 예측 벡터, CopyAbove 런 령스, 또는 다른 값들을 코딩하기 위한 상기 설명된 코드들을 이용할 수도 있다.

[0186] Joshi 등, "Non-SCCE3: Contexts for coding index runs", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 의 비디오 코딩에 관한 합동 협력팀 (JCT-VC), 18 차 회의, Sapporo, JP, 2014 년 6 월 30 - 2014 년 7 월 9 일, JCTVC-R0174 (이하, JCTVC-R0174) 에서는, 저자들이 런 령스 코드워드들의 컨텍스트들이 (예컨대, CopyFromTop 모드와 유사한 방식으로 동작할 수도 있는) CopyLeft 모드가 이용될 경우에 인덱스에 종속되도록 하는 것을 제안하였다. 그러나, 이 개시물의 예에 따르면, 현재의 런 모드가 CopyFromAbove 일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 코딩되고 있는 픽셀의 상부에 위치되는 픽셀의 인덱스 값에 기초하여 런을 CABAC 코딩하기 위한 컨텍스트들을 결정할 수도 있다. 이 예에서, 상부-이웃 픽셀은 현재의 CU 의 외부에 있고, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 대응하는 인덱스가 미리 정의된 상수 k 와 동일한 것으로 결정할 수도 있다. 일부 예들에서, 상수 k 는 제로와 동일할 수도 있다.

[0187] 일부 예들에서, 현재의 픽셀을 코딩하기 위한 팔레트 모드가 CopyFromAbove mode 일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 상부-이웃 픽셀이 제로와 동일한 인덱스를 가지는지 여부에 기초하여 런 령스 코드워드의 최초 빈 (first bin) 을 코딩하기 위하여 2 개의 후보 CABAC 컨텍스트들 중의 하나를 선택할 수도 있다. 또 다른 예로서, 현재의 픽셀을 코딩하기 위한 팔레트 모드가 CopyPrevious 모드일 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인덱스가 제로, 1, 2 와 동일하거나, 2 보다 더 큰지 여부에 기초하여 런 령스 코드워드의 최초 빈을 코딩하기 위하여 4 개의 후보 CABAC 컨텍스트들 중의 하나를 선택할 수도 있다.

[0188] 도 2 는 이 개시물의 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20) 를 예시하는 블록도이다. 도 2 는 설명의 목적들을 위하여 제공되고, 이 개시물에서 대략적으로 예시되고 설명된 바와 같은 기법들의 제한으로 고려되지 않아야 한다. 설명의 목적들을 위하여, 이 개시물은 HEVC 코딩의 맥락에서 비디오 인코더 (20)

를 설명한다. 그러나, 이 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0189] 비디오 인코더 (20) 는 이 개시물에서 설명된 다양한 예들에 따라 팔레트-기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스의 예를 표현한다. 예를 들어, 비디오 인코더 (20) 는 팔레트-기반 코딩 또는 비-팔레트-기반 코딩의 어느 하나를 이용하여, HEVC 코딩에서의 CU 들 및 PU 들과 같은 비디오 데이터의 다양한 블록들을 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트-기반 코딩 모드들은 HEVC 초안 10 에 의해 특정된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간적 코딩 모드들 또는 인트라-예측 공간적 코딩 모드들을 지칭할 수도 있다. 비디오 인코더 (20) 는 하나의 예에서, 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 생성하고, 비디오 데이터의 블록에서의 적어도 일부 픽셀 로케이션들의 픽셀 값들을 표현하기 위하여 팔레트에서 픽셀 값들을 선택하고, 비디오 데이터의 블록에서의 픽셀 로케이션들의 적어도 일부를, 팔레트에서의 선택된 픽셀 값들에 각각 대응하는 팔레트에서의 엔트리들과 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 시그널링된 정보는 비디오 데이터를 디코딩하기 위하여 비디오 디코더 (30) 에 의해 이용될 수도 있다.

[0190] 도 2 의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 유닛 (100), 비디오 데이터 메모리 (101), 잔차 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 프로세싱 유닛 (110), 재구성 유닛 (112), 필터 유닛 (114), 디코딩된 픽처 버퍼 (116), 및 엔트로피 인코딩 유닛 (118) 을 포함한다. 예측 프로세싱 유닛 (100) 은 인터-예측 프로세싱 유닛 (120) 및 인트라-예측 프로세싱 유닛 (126) 을 포함한다. 인터-예측 프로세싱 유닛 (120) 은 모션 추정 유닛 및 모션 보상 유닛 (도시되지 않음) 을 포함한다. 비디오 인코더 (20) 는 또한, 이 개시물에서 설명된 팔레트-기반 코딩 기법들의 다양한 양태들을 수행하도록 구성된 팔레트-기반 인코딩 유닛 (122) 을 포함한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능적 컴포넌트들을 포함할 수도 있다.

[0191] 비디오 데이터 메모리 (101) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩되어야 할 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (101) 내에 저장된 비디오 데이터는 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. 디코딩된 픽처 버퍼 (116) 는 예컨대, 인트라-코딩 또는 인터-코딩 모드들에서 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩함에 있어서 이용하기 위한 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (101) 및 디코딩된 픽처 버퍼 (116) 는 동기식 DRAM (synchronous DRAM; SDRAM), 자기저항 RAM (magnetoresistive RAM; MRAM), 저항 RAM (resistive RAM; RRAM), 또는 다른 타입들의 메모리 디바이스들을 포함하는 동적 랜덤 액세스 메모리 (dynamic random access memory; DRAM) 와 같은 다양한 메모리 디바이스들 중의 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (101) 및 디코딩된 픽처 버퍼 (116) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (101) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 함께 온-칩 (on-chip) 일 수도 있거나, 그 컴포넌트들에 관련하여 오프-칩 (off-chip) 일 수도 있다.

[0192] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터의 픽처의 슬라이스에서의 각각의 CTU 를 인코딩할 수도 있다. CTU 들의 각각은 동일한 크기의 루마 코딩 트리 블록 (coding tree block; CTB) 들 및 픽처의 대응하는 CTB 들과 연관될 수도 있다. CTU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 CTU 의 CTB 들을 점진적으로 더 작은 블록들로 분할하기 위하여 쿼드-트리 파티셔닝 (quad-tree partitioning) 을 수행할 수도 있다. 더 작은 블록은 CU 들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 CTU 와 연관된 CTB 를 4 개의 동일한 크기의 서브-블록들로 파티셔닝할 수도 있고, 서브-블록들 중의 하나 이상을 4 개의 동일한 크기의 서브-서브-블록 (sub-sub-block) 들로 파티셔닝할 수도 있는 등등과 같다.

[0193] 비디오 인코더 (20) 는 CU 들의 인코딩된 표현들 (즉, 코딩된 CU 들) 을 생성하기 위하여 CTU 의 CU 들을 인코딩할 수도 있다. CU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU 들 중에서 CU 와 연관된 코딩 블록들을 파티셔닝할 수도 있다. 이에 따라, 각각의 PU 는 루마 예측 블록 및 대응하는 크로마 예측 블록들과 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다양한 크기들을 가지는 PU 들을 지원할 수도 있다. 위에서 표시된 바와 같이, CU 의 크기는 CU 의 루마 코딩 블록의 크기를 지칭할 수도 있고, PU 의 크기는 PU 의 루마 예측 블록의 크기를 지칭할 수도 있다. 특정한 CU 의 크기가 $2N \times 2N$ 인 것으로 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인트라 예측을 위한 $2N \times 2N$ 또는 $N \times N$ 의 PU 크기들과, 인터 예측을 위한 $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, 또는 유사한 것의 대칭적인 PU 크기들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 인터 예측을 위한 $2N \times nU$, $2N \times nD$,

nLx2N, 및 nRx2N 의 PU 크기들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

- [0194] 인터-예측 프로세싱 유닛 (120) 은 CU 의 각각의 PU 에 대해 인터 예측을 수행함으로써 PU 에 대한 예측 데이터를 생성할 수도 있다. PU 에 대한 예측 데이터는 PU 의 예측 블록들 및 PU 에 대한 모션 정보를 포함할 수도 있다. 인터-예측 유닛 (121) 은 PU 가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 내에 있는지 여부에 따라 CU 의 PU 에 대한 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들은 인트라 예측된다. 이 때문에, PU 가 I 슬라이스 내에 있을 경우, 인터-예측 유닛 (121) 은 PU 에 대해 인터 예측을 수행하지 않는다. 이에 따라, I-모드에서 인코딩된 블록들에 대하여, 예측된 블록은 동일한 프레임 내에서 이전에 인코딩된 이웃하는 블록들로부터의 공간적 예측을 이용하여 형성된다.
- [0195] PU 가 P 슬라이스 내에 있을 경우, 인터-예측 프로세싱 유닛 (120) 의 모션 추정 유닛은 PU 에 대한 참조 영역을 위하여 참조 픽처들의 리스트 (예컨대, "RefPicList0") 에서 참조 픽처들을 검색할 수도 있다. PU 에 대한 참조 영역은, 참조 픽처 내에서, PU 의 샘플 블록들에 가장 근접하게 대응하는 샘플 블록들을 포함하는 영역일 수도 있다. 모션 추정 유닛은 PU 에 대한 참조 영역을 포함하는 참조 픽처의 RefPicList0 에서의 위치를 표시하는 참조 인덱스를 생성할 수도 있다. 게다가, 모션 추정 유닛은 PU 의 코딩 블록과, 참조 영역과 연관된 참조 로케이션과의 사이의 공간적 변위를 표시하는 MV 를 생성할 수도 있다. 예를 들어, MV 는 현재의 디코딩된 픽처에서의 좌표들로부터 참조 픽처에서의 좌표들까지의 오프셋을 제공하는 2 차원 벡터일 수도 있다. 모션 추정 유닛은 참조 인덱스 및 MV 를 PU 의 모션 정보로서 출력할 수도 있다. 인터-예측 프로세싱 유닛 (120) 의 모션 보상 유닛은 PU 의 모션 벡터에 의해 표시된 참조 로케이션에서의 실제적인 또는 보간된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.
- [0196] PU 가 B 슬라이스 내에 있을 경우, 모션 추정 유닛은 PU 에 대한 단방향-예측 또는 양방향-예측을 수행할 수도 있다. PU 에 대한 단방향-예측을 수행하기 위하여, 모션 추정 유닛은 PU 에 대한 참조 영역을 위하여 RefPicList0 또는 제 2 참조 픽처 리스트 ("RefPicList1") 의 참조 픽처들을 검색할 수도 있다. 모션 추정 유닛은 PU 의 모션 정보로서, 참조 영역을 포함하는 참조 픽처의 RefPicList0 또는 RefPicList1 에서의 위치를 표시하는 참조 인덱스, PU 의 예측 블록과, 참조 영역과 연관된 참조 로케이션과의 사이의 공간적 변위를 표시하는 MV, 및 참조 픽처가 RefPicList0 또는 RefPicList1 내에 있는지 여부를 표시하는 하나 이상의 예측 방향 표시자들을 출력할 수도 있다. 인터-예측 프로세싱 유닛 (120) 의 모션 보상 유닛은 PU 의 모션 벡터에 의해 표시된 참조 영역에서의 실제적인 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU 의 예측 블록들을 생성할 수도 있다.
- [0197] PU 에 대한 양방향 인터 예측을 수행하기 위하여, 모션 추정 유닛은 PU 에 대한 참조 영역을 위하여 RefPicList0 에서의 참조 픽처들을 검색할 수도 있고, 또한, PU 에 대한 또 다른 참조 영역을 위하여 RefPicList1 에서의 참조 픽처들을 검색할 수도 있다. 모션 추정 유닛은 참조 영역들을 포함하는 참조 픽처들의 RefPicList0 및 RefPicList1 에서의 위치들을 표시하는 참조 픽처 인덱스들을 생성할 수도 있다. 게다가, 모션 추정 유닛은 참조 영역들과 연관된 참조 로케이션과, PU 의 샘플 블록과의 사이의 공간적 변위들을 표시하는 MV 들을 생성할 수도 있다. PU 의 모션 정보는 참조 인덱스들 및 PU 의 MV들을 포함할 수도 있다. 모션 보상 유닛은 PU 의 모션 벡터들에 의해 표시된 참조 영역에서의 실제적인 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU 의 예측 블록들을 생성할 수도 있다.
- [0198] 이 개시물의 다양한 예들에 따르면, 비디오 인코더 (20) 는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다. HEVC 프레임워크에 대하여, 예로서, 팔레트-기반 코딩 기법들은 코딩 유닛 (CU) 모드로서 이용되도록 구성될 수도 있다. 다른 예들에서, 팔레트-기반 코딩 기법들은 HEVC 의 프레임워크에서 PU 모드로서 이용되도록 구성될 수도 있다. 따라서, CU 모드의 맥락에서 본원에서 (이 개시물의 전반에 걸쳐) 설명된 개시된 프로세스들의 전부는 부가적으로 또는 대안적으로, PU 에 적용할 수도 있다. 그러나, 이 HEVC-기반 예들은 본원에서 설명된 팔레트-기반 코딩 기법들의 한정 또는 제한으로 간주되지 않아야 하는데, 이것은 이러한 기법들이 독립적으로, 또는 다른 기준의 또는 아직 개발되어야 할 시스템들/표준들의 일부로서 작동하도록 적용될 수도 있기 때문이다. 이 경우들에는, 팔레트 코딩을 위한 유닛이 정사각형 블록들, 직사각형 블록들, 또는 심지어 비-직사각형 형태의 영역들일 수 있다.
- [0199] 팔레트-기반 인코딩 유닛 (122) 은 예를 들어, 팔레트-기반 인코딩 모드가 예컨대, CU 또는 PU 에 대하여 선택될 때에 팔레트-기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛 (122) 은, 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 생성하고, 비디오 데이터의 블록의 적어도 일부의 위치들의 픽셀 값들을 표현하기 위하여 팔레트에서 픽셀 값들을 선택하고, 비디오 데이터의 블록의 위치들 중의 적어도

일부를, 선택된 픽셀 값들에 각각 대응하는 팔레트에서의 엔트리들과 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 인코딩 유닛 (122) 에 의해 수행되는 것으로서 설명되지만, 이러한 기능들의 일부 또는 전부는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0200] 이 개시물의 양태들에 따르면, 팔레트-기반 인코딩 유닛 (122) 은 본원에서 설명된 팔레트 코딩을 위한 기법들의 임의의 조합을 수행하도록 구성될 수도 있다. 예를 들어, 이 개시물의 양태들에 따르면, 팔레트-기반 인코딩 유닛 (122) 은 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 개개의 샘플이 제 1 팔레트 모드로 코딩되는지 여부를 표시하는 신택스 엘리먼트의 값을 결정할 수도 있고, 여기서, 제 1 팔레트 모드는 블록에 대한 컬러 값들의 팔레트에 대한 인덱스를 이용하지 않으면서, 블록의 개개의 샘플을 코딩하는 것을 포함한다. 예를 들어, 팔레트-기반 인코딩 유닛 (122) 은 블록의 임의의 샘플이 탈출 샘플로서 (블록을 코딩하기 위한 팔레트에서 표현된 컬러 값을 가지지 않는 샘플로서) 인코딩되는지 여부를 표시하는 블록-레벨 신택스 엘리먼트의 값을 결정할 수도 있다. 일부 예들에서, 팔레트-기반 인코딩 유닛 (122) 은 블록의 임의의 샘플이 탈출 샘플들로서 인코딩되는지 여부를 표시하는 블록에 대한 탈출 플래그를 결정할 수도 있다.

[0201] 추가적으로 또는 대안적으로, 팔레트-기반 인코딩 유닛 (122) 은 비디오 데이터의 블록을 코딩하기 위한 컬러 값들의 팔레트의 최대 팔레트 크기를 표시하는 데이터, 또는 컬러 값들의 팔레트를 결정하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 표시하는 데이터 중의 적어도 하나를 결정할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛 (122) 은 SPS 와 같은 파라미터 세트에서 이러한 데이터를 포함할 수도 있다.

[0202] 추가적으로 또는 대안적으로, 팔레트-기반 인코딩 유닛 (122) 은 픽셀의 값을, 픽셀을 코딩하기 위한 컬러들의 팔레트에서 컬러 값에 관련시키는 팔레트 인덱스와 연관된 픽셀에 대하여, 픽셀의 팔레트 인덱스로 코딩되고 있는 팔레트 인덱스들의 런의 런 령스를 결정할 수도 있다. 즉, 팔레트-기반 인코딩 유닛 (122) 은 특정한 샘플들에 대한 인덱스 값이 다른 후속 팔레트 인덱스들의 런으로 인코딩되고 있는 것으로 결정할 수도 있다. 팔레트-기반 인코딩 유닛 (122) 은 또한, 픽셀의 팔레트 인덱스로 인코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정할 수도 있다. 엔트로피 인코딩 유닛 (118) 에 대하여 이하에서 더욱 상세하게 설명된 바와 같이, 비디오 인코더 (20) 는 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 인코딩할 수도 있다.

[0203] 인트라-예측 프로세싱 유닛 (126) 은 PU 에 대해 인트라 예측을 수행함으로써 PU 에 대한 예측 데이터를 생성할 수도 있다. PU 에 대한 예측 데이터는 PU 에 대한 예측 블록들 및 다양한 신택스 엘리먼트들을 포함할 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU 들에 대해 인트라 예측을 수행할 수도 있다.

[0204] PU 에 대해 인트라 예측을 수행하기 위하여, 인트라-예측 프로세싱 유닛 (126) 은 PU 에 대한 예측 데이터의 다수의 세트들을 생성하기 위하여 다수의 인트라 예측 모드들을 이용할 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 PU 에 대한 예측 블록을 생성하기 위하여 이웃하는 PU 들의 샘플 블록들로부터의 샘플들을 이용할 수도 있다. PU 들, CU 들, 및 CTU 들에 대하여 좌측에서 우측, 상부에서 하부의 인코딩 순서를 가정하면, 이웃하는 PU 들은 PU 의 상부, 상부 및 우측, 상부 및 좌측, 또는 좌측에 있을 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 다양한 수들의 인트라 예측 모드들, 예컨대, 33 개의 방향성 인트라 예측 모드들을 이용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 수는 PU 와 연관된 영역의 크기에 종속될 수도 있다.

[0205] 예측 프로세싱 유닛 (100) 은 PU 들에 대한 인트라-예측 프로세싱 유닛 (120) 에 의해 생성된 예측 데이터, 또는 PU 들에 대한 인트라-예측 프로세싱 유닛 (126) 에 의해 생성된 예측 데이터 중으로부터 CU 의 PU 들에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭 (metric) 들에 기초하여 CU 의 PU 들에 대한 예측 데이터를 선택한다. 선택된 예측 데이터의 예측 블록들은 선택된 예측 블록들로서 본원에서 지칭될 수도 있다.

[0206] 잔차 생성 유닛 (102) 은, CU 의 루마, Cb, 및 Cr 코딩 블록 및 CU 의 PU들의 선택된 예측 루마, Cb, 및 Cr 블록들에 기초하여, CU 의 루마, Cb, 및 Cr 잔차 블록들을 생성할 수도 있다. 예를 들어, 잔차 생성 유닛 (102) 은, 잔차 블록들에서의 각각의 샘플이 CU 의 코딩 블록에서의 샘플과, CU 의 PU 의 대응하는 선택된 예측 블록에서의 대응하는 샘플과의 사이의 차이와 동일한 값을 가지도록, CU 의 잔차 블록들을 생성할 수도 있다.

[0207] 변환 프로세싱 유닛 (104) 은 CU 와 연관된 잔차 블록들을 CU 의 TU 들과 연관된 변환 블록들로 파티셔닝하기 위하여 퀴드-트리 파티셔닝을 수행할 수도 있다. 이에 따라, TU 는 루마 변환 블록 및 2 개의 크로마 변환

블록들과 연관될 수도 있다. CU의 TU들의 루마 및 크로마 변환 블록들의 크기들 및 포지션들은 CU의 PU들의 예측 블록들의 크기들 및 위치들에 기초할 수도 있거나 기초하지 않을 수도 있다. "잔차 쿼드 트리"(RQT)로서 알려진 쿼드-트리 구조는 영역들의 각각과 연관된 노드들을 포함할 수도 있다. CU의 TU들은 RQT의 리프 노드(leaf node)들에 대응할 수도 있다.

[0208] 변환 프로세싱 유닛(104)은 하나 이상의 변환들을 TU의 변환 블록들에 적용함으로써 CU의 각각의 TU에 대한 변환 계수 블록들을 생성할 수도 있다. 변환 프로세싱 유닛(104)은 다양한 변환들을 TU와 연관된 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛(104)은 이산 코사인 변환(discrete cosine transform; DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 변환 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛(104)은 변환들을 변환 블록에 적용하지 않는다. 이러한 예들에서는, 변환 블록이 변환 계수 블록으로서 프로세싱될 수도 있다.

[0209] 양자화 유닛(106)은 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 변환 계수들의 일부 또는 전부와 연관된 비트-심도(bit-depth)를 감소시킬 수도 있다. 예를 들어, n -비트 변환 계수는 양자화 동안에 m -비트 변환 계수로 버림(round down)될 수도 있고, 여기서, n 은 m 보다 더 크다. 양자화 유닛(106)은 CU와 연관된 양자화 파라미터(quantization parameter; QP)값에 기초하여 CU의 TU와 연관된 계수 블록을 양자화할 수도 있다. 비디오 인코더(20)는 CU와 연관된 QP값을 조절함으로써, CU와 연관된 계수 블록들에 적용된 양자화도를 조절할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있고, 이에 따라, 양자화된 변환 계수들은 원래의 것들보다 더 낮은 정밀도를 가질 수도 있다.

[0210] 역양자화 유닛(108) 및 역변환 프로세싱 유닛(110)은 계수 블록으로부터 잔차 블록을 재구성하기 위하여, 역양자화 및 역변환들을 계수 블록에 각각 적용할 수도 있다. 재구성 유닛(112)은 TU와 연관된 재구성된 변환 블록을 생성하기 위하여, 재구성된 잔차 블록을, 예측 프로세싱 유닛(100)에 의해 생성된 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 추가할 수도 있다. CU의 각각의 TU에 대한 변환 블록들을 이러한 방법으로 재구성함으로써, 비디오 인코더(20)는 CU의 코딩 블록들을 재구성할 수도 있다.

[0211] 필터 유닛(114)은 CU와 연관된 코딩 블록들에서의 블록킹 아티팩트(blocking artifact)들을 감소시키기 위하여 하나 이상의 디블록킹 동작(deblocking operation)들을 수행할 수도 있다. 필터 유닛(114)이 재구성된 코딩 블록들에 대해 하나 이상의 디블록킹 동작들을 수행한 후, 디코딩된 픽처 버퍼(116)는 재구성된 코딩 블록들을 저장할 수도 있다. 인터-예측 프로세싱 유닛(120)은 다른 픽처들의 PU들에 대해 인터 예측을 수행하기 위하여, 재구성된 코딩 블록들을 포함하는 참조 픽처를 이용할 수도 있다. 게다가, 인트라-예측 프로세싱 유닛(126)은 CU와 동일한 픽처에서의 다른 PU들에 대해 인트라 예측을 수행하기 위하여, 디코딩된 픽처 버퍼(116)에서의 재구성된 코딩 블록들을 이용할 수도 있다.

[0212] 엔트로피 인코딩 유닛(118)은 비디오 인코더(20)의 다른 기능적 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛(118)은 양자화 유닛(106)으로부터 계수 블록들을 수신할 수도 있고, 예측 프로세싱 유닛(100)으로부터 선택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛(118)은 엔트로피-인코딩된 데이터를 생성하기 위하여, 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛(118)은 데이터에 대하여, 컨텍스트-적응 가변 길이 코딩(context-adaptive variable length coding; CAVLC) 동작, CABAC 동작, 가변-대-가변(variable-to-variable; V2V) 길이 코딩 동작, 선택스-기반 컨텍스트-적응 2진 산술 코딩(syntax-based context-adaptive binary arithmetic coding; SBAC) 동작, 확률 간격 파티셔닝 엔트로피(Probability Interval Partitioning Entropy; PIPE) 코딩 동작, 지수-골롬(Exponential-Golomb) 인코딩 동작, 또는 또 다른 타입의 엔트로피 인코딩 동작을 수행할 수도 있다. 비디오 인코더(20)는 엔트로피 인코딩 유닛(118)에 의해 생성된 엔트로피-인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들어, 비트스트림은 CU에 대한 RQT를 표현하는 데이터를 포함할 수도 있다.

[0213] 이 개시물의 양태들에 따르면, 엔트로피 인코딩 유닛(118)은 도 1의 예에 대하여 위에서 설명된 바와 같이, TEGk 코드를 이용하여 팔레트 데이터를 코딩하도록 구성될 수도 있다. 특히, 이 개시물의 양태들에 따르면, 엔트로피 인코딩 유닛(118)은 결정된 최대 런 렱스에 기초하여 팔레트 인덱스들의 런을 위한 런 렱스를 표시하는 데이터를 인코딩할 수도 있다. 일부 예들에서, 엔트로피 인코딩 유닛(118)은 TEG2 코드를 이용하여 런 렱스를 인코딩할 수도 있다.

[0214] 도 3은 이 개시물의 기법들을 구현하도록 구성되는 일 예의 비디오 디코더(30)를 예시하는 블록도이다. 도 3은 설명의 목적들을 위하여 제공되고, 이 개시물에서 대략 예시되고 설명된 바와 같은 기법들에 대해 제한

하지 않는다. 설명의 목적들을 위하여, 이 개시물은 HEVC 코딩의 맥락에서 비디오 디코더 (30) 를 설명한다. 그러나, 이 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0215] 비디오 인코더 (20) 는 이 개시물에서 설명된 다양한 예들에 따라 팔레트-기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스의 예를 표현한다. 예를 들어, 비디오 인코더 (20) 는 팔레트-기반 코딩 또는 비-팔레트-기반 코딩의 어느 하나를 이용하여, HEVC 코딩에서의 CU 들 및 PU 들과 같은 비디오 데이터의 다양한 블록들을 선택적으로 디코딩하도록 구성될 수도 있다. 비-팔레트-기반 코딩 모드들은 HEVC 초안 10 에 의해 특정된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간적 코딩 모드들 또는 인트라-예측 공간적 코딩 모드들을 지칭할 수도 있다. 비디오 디코더 (30) 는 하나의 예에서, 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 생성하고, 비디오 데이터의 블록에서의 적어도 일부 픽셀 로케이션들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신하고, 정보에 기초하여 팔레트에서 픽셀 값들을 선택하고, 팔레트에서의 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 재구성하도록 구성될 수도 있다.

[0216] 도 3 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (150), 비디오 데이터 메모리 (151), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 필터 유닛 (160), 및 디코딩된 픽처 버퍼 (162) 를 포함한다. 예측 프로세싱 유닛 (152) 은 모션 보상 유닛 (164) 및 인트라-예측 프로세싱 유닛 (166) 을 포함한다. 비디오 디코더 (30) 는 또한, 이 개시물에서 설명된 팔레트-기반 코딩 기법들의 다양한 양태들을 수행하도록 구성된 팔레트-기반 디코딩 유닛 (165) 을 포함한다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능적 컴포넌트들을 포함할 수도 있다.

[0217] 비디오 데이터 메모리 (151) 는 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩되어야 할, 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (151) 내에 저장된 비디오 데이터는 예를 들어, 채널 (16) 로부터, 예컨대, 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체들을 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (151) 는 인코딩된 비디오 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (coded picture buffer; CPB) 를 형성할 수도 있다. 디코딩된 픽처 버퍼 (162) 는 예컨대, 인트라-코딩 또는 인터-코딩 모드들에서 비디오 디코더 (30) 에 의해 비디오 데이터를 디코딩함에 있어서 사용하기 위한 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (151) 및 디코딩된 픽처 버퍼 (162) 는 동기식 DRAM (SDRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들을 포함하는 동적 랜덤 액세스 메모리 (DRAM) 와 같은 다양한 메모리 디바이스들 중의 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (151) 및 디코딩된 픽처 버퍼 (162) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (151) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 함께 온-칩일 수도 있거나, 그러한 컴포넌트들에 관련하여 오프-칩일 수도 있다.

[0218] 코딩된 픽처 버퍼 (CPB) 는 비트스트림의 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신할 수도 있고 저장할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 CPB 로부터 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신할 수도 있고, 신텍스 엘리먼트들을 디코딩하기 위하여 NAL 유닛들을 파싱할 수도 있다. 엔트로피 디코딩 유닛 (160) 은 NAL 유닛들에서의 엔트로피-인코딩된 신텍스 엘리먼트들을 엔트로피 디코딩할 수도 있다.

[0219] 이 개시물의 양태들에 따르면, 엔트로피 디코딩 유닛 (150) 은 도 1 의 예에 대하여 위에서 설명된 바와 같이, TEGk 코드를 이용하여 팔레트 데이터를 디코딩하도록 구성될 수도 있다. 특히, 이 개시물의 양태들에 따르면, 엔트로피 디코딩 유닛 (150) 은 결정된 최대 런 렱스에 기초하여 팔레트 인덱스들의 런 (예컨대, 동일한 값을 가지는 인덱스들의 런, 또는 상부-이웃 인덱스들로부터 복사되는 인덱스들의 런) 을 위한 런 렱스를 표시하는 데이터를 디코딩할 수도 있다. 일부 예들에서, 엔트로피 디코딩 유닛 (150) 은 TEG2 코드를 이용하여 런 렱스를 디코딩할 수도 있다.

[0220] 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 및 필터 유닛 (160) 은 비트스트림으로부터 추출된 신텍스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다. 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 것의 일부로서, 엔트로피 디코딩 유닛 (150) 은 코딩된 슬라이스 NAL 유닛들로부터 신텍스 엘리먼트들을 추출할 수도 있고 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들의 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 속하는 신텍스 엘리먼트들을 포함할 수도 있다.

슬라이스 헤더에서의 선택스 엘리먼트들은 슬라이스를 포함하는 픽처와 연관된 PPS 를 식별하는 선택스 엘리먼트를 포함할 수도 있다.

- [0221] 비트스트림으로부터 선택스 엘리먼트들을 디코딩하는 것에 추가하여, 비디오 디코더 (30) 는 비-파티셔닝 (non-partitioned) CU 에 대해 재구성 동작을 수행할 수도 있다. 비-파티셔닝 CU 에 대해 재구성 동작을 수행하기 위하여, 비디오 디코더 (30) 는 CU 의 각각의 TU 에 대해 재구성 동작을 수행할 수도 있다. CU 의 각각의 TU 에 대해 재구성 동작을 수행함으로써, 비디오 디코더 (30) 는 CU 의 잔차 블록들을 재구성할 수도 있다.
- [0222] CU 의 TU 에 대해 재구성 동작을 수행하는 것의 일부로서, 역양자화 유닛 (154) 은 TU 와 연관된 계수 블록들을 역양자화, 즉, 탈-양자화 (de-quantize) 할 수도 있다. 역양자화 유닛 (154) 은 역양자화 유닛 (154) 이 적용하기 위한 양자화도 및 마찬가지로, 역양자화도를 결정하기 위하여 TU 의 CU 와 연관된 QP 값을 이용할 수도 있다. 즉, 압축 비율, 즉, 원래의 시퀀스 및 압축된 것을 표현하기 위하여 이용된 비트들의 수의 비율은 변환 계수들을 양자화할 때에 이용된 QP 의 값을 조절함으로써 제어될 수도 있다. 압축 비율은 또한, 채용된 엔트로피 코딩의 방법에 종속될 수도 있다.
- [0223] 역양자화 유닛 (154) 이 계수 블록을 역양자화한 후, 역변환 프로세싱 유닛 (156) 은 TU 와 연관된 잔차 블록을 생성하기 위하여 하나 이상의 역변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (156) 은 역 DCT, 역정수 변환, 역 카루넨-루베 변환 (Karhunen-Loeve transform; KLT), 역회전 변환, 역방향성 변환, 또는 또 다른 역변환을 계수 블록에 적용할 수도 있다.
- [0224] PU 가 인트라 예측을 이용하여 인코딩될 경우, 인트라-예측 프로세싱 유닛 (166) 은 PU 에 대한 예측 블록들을 생성하기 위하여 인트라 예측을 수행할 수도 있다. 인트라-예측 프로세싱 유닛 (166) 은 공간적으로 이웃하는 PU 들의 예측 블록들에 기초하여 PU 에 대한 예측 루마, Cb, 및 Cr 블록들을 생성하기 위하여 인트라-예측 모드를 이용할 수도 있다. 인트라-예측 프로세싱 유닛 (166) 은 비트스트림으로부터 디코딩된 하나 이상의 선택스 엘리먼트들에 기초하여 PU 에 대한 인트라 예측 모드를 결정할 수도 있다.
- [0225] 예측 프로세싱 유닛 (152) 은 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 제 1 참조 픽처 리스트 (RefPicList0) 및 제 2 참조 픽처 리스트 (RefPicList1) 를 구성할 수도 있다. 또한, PU 가 인터 예측을 이용하여 인코딩될 경우, 엔트로피 디코딩 유닛 (150) 은 PU 에 대한 모션 정보를 추출할 수도 있다. 모션 보상 유닛 (164) 은 PU 의 모션 정보에 기초하여, PU 에 대한 하나 이상의 참조 영역들을 결정할 수도 있다. 모션 보상 유닛 (164) 은 PU 에 대한 하나 이상의 참조 블록들에서의 샘플들 블록들에 기초하여, PU 에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.
- [0226] 재구성 유닛 (158) 은 CU 의 루마, Cb, 및 Cr 코딩 블록들을 재구성하기 위하여, CU 의 TU 들과 연관된 루마, Cb, 및 Cr 변환 블록들 및 CU 의 PU 들의 예측 루마, Cb, 및 Cr 블록들, 즉, 적용 가능한 바와 같이, 인트라-예측 데이터 또는 인터-예측 데이터의 어느 하나를 이용할 수도 있다. 예를 들어, 재구성 유닛 (158) 은 CU 의 루마, Cb, 및 Cr 코딩 블록들을 재구성하기 위하여, 루마, Cb, 및 Cr 변환 블록들의 샘플들을 예측 루마, Cb, 및 Cr 블록들의 대응하는 샘플들에 추가할 수도 있다.
- [0227] 필터 유닛 (160) 은 CU 의 루마, Cb, 및 Cr 코딩 블록들과 연관된 블록킹 아티팩트들을 감소시키기 위하여 디블록킹 동작을 수행할 수도 있다. 비디오 디코더 (30) 는 CU 의 루마, Cb, 및 Cr 코딩 블록들을 디코딩된 픽처 버퍼 (162) 내에 저장할 수도 있다. 디코딩된 픽처 버퍼 (162) 는 후속 모션 보상, 인트라 예측, 및 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에서의 제시를 위하여 참조 픽처들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 디코딩된 픽처 버퍼 (162) 에서의 루마, Cb, 및 Cr 블록들에 기초하여, 다른 CU 들의 PU 들에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다.
- [0228] 이 개시물의 다양한 예들에 따르면, 비디오 디코더 (30) 는 팔레트-기반 코딩을 수행하도록 구성될 수도 있다. 팔레트-기반 디코딩 유닛 (165) 은 예를 들어, 팔레트-기반 디코딩 모드가 예컨대, CU 또는 PU 에 대하여 선택될 때에 팔레트-기반 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165) 은 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 생성하고, 비디오 데이터의 블록에서의 적어도 일부 픽셀 로케이션들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신하고, 정보에 기초하여 팔레트에서 픽셀 값들을 선택하고, 팔레트에서의 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 재구성하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 디코딩 유닛 (165) 에 의해 수행되는 것으로서 설명되지만, 이러한 기능들의 일부 또는 전부는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.
- [0229] 팔레트-기반 디코딩 유닛 (165) 은 팔레트 코딩 모드 정보를 수신할 수도 있고, 팔레트 코딩 모드 정보가 팔레

트 코딩 모드가 블록에 적용되는 것을 표시할 때에 상기 동작들을 수행할 수도 있다. 팔레트 코딩 모드 정보가 팔레트 코딩 모드가 블록에 적용되지 않는다는 것을 표시할 때, 또는 다른 모드 정보가 상이한 모드의 이용을 표시할 때, 비디오 디코더 (30) 는 예컨대, HEVC 인터-예측 또는 인트라-예측 코딩 모드와 같은 비-팔레트-기반 코딩 모드를 이용하여 비디오 데이터의 블록을 디코딩할 수도 있다. 비디오 데이터의 블록은 예를 들어, HEVC 코딩 프로세스에 따라 생성된 CU 또는 PU 일 수도 있다.

[0230] 이 개시물의 양태들에 따르면, 팔레트-기반 디코딩 유닛 (165) 은 본원에서 설명된 팔레트 코딩을 위한 기법들의 임의의 조합을 수행하도록 구성될 수도 있다. 예를 들어, 이 개시물의 양태들에 따르면, 팔레트-기반 디코딩 유닛 (165) 은 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 개개의 샘플이 제 1 팔레트 모드로 코딩되는지 여부를 표시하는 신택스 엘리먼트의 값을 결정할 수도 있고, 여기서, 제 1 팔레트 모드는 블록에 대한 컬러 값들의 팔레트에 대한 인덱스를 이용하지 않으면서, 블록의 개개의 샘플을 코딩하는 것을 포함한다. 예를 들어, 팔레트-기반 디코딩 유닛 (165) 은 블록의 임의의 샘플이 탈출 샘플 (예컨대, 팔레트로부터의 컬러 엔트리를 이용하여 재구성되지 않을 수도 있는 샘플) 로서 디코딩되어야 하는지 여부를 표시하는 블록-레벨 신택스 엘리먼트의 값을 결정할 수도 있다. 일부 예들에서, 팔레트-기반 디코딩 유닛 (165) 은 블록의 임의의 샘플들이 탈출 샘플로서 디코딩되어야 하는지 여부를 표시하는 블록에 대한 1 비트 탈출 플래그를 결정할 수도 있다.

[0231] 추가적으로 또는 대안적으로, 팔레트-기반 디코딩 유닛 (165) 은 비디오 데이터의 블록을 코딩하기 위한 컬러 값들의 팔레트의 최대 팔레트 크기를 표시하는 데이터, 또는 컬러 값들의 팔레트를 결정하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 표시하는 데이터 중의 적어도 하나를 결정할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165) 은 SPS 와 같은 파라미터 세트로부터의 이러한 데이터를 디코딩할 수도 있다.

[0232] 추가적으로 또는 대안적으로, 팔레트-기반 디코딩 유닛 (165) 은 픽셀의 값을, 픽셀을 코딩하기 위한 컬러들의 팔레트에서의 컬러 값에 관련시키는 팔레트 인덱스와 연관된 픽셀에 대하여, 픽셀의 팔레트 인덱스와 함께 디코딩되는 팔레트 인덱스들의 런 (예컨대, 동일한 값을 가지는 인덱스들의 런 또는 상부-이웃 인덱스들로부터 복사되는 인덱스들의 런) 의 런 령스를 결정할 수도 있다. 즉, 팔레트-기반 디코딩 유닛 (165) 은 특정한 샘플에 대한 인덱스 값이 다른 후속 팔레트 인덱스들의 런으로 디코딩되는 것으로 결정할 수도 있다. 팔레트-기반 디코딩 유닛 (165) 은 또한, 픽셀의 팔레트 인덱스로 디코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정할 수도 있다. 엔트로피 디코딩 유닛 (150) 에 대하여 위에서 언급된 바와 같이, 비디오 디코더 (30) 는 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 디코딩할 수도 있다.

[0233] 도 4 는 이 개시물의 기법들과 부합하는, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 예를 예시하는 개념도이다. 도 4 의 예는 제 1 팔레트들 (184) 과 연관되는 제 1 코딩 유닛 (CU) (180) 및 제 2 팔레트들 (192) 과 연관되는 제 2 CU (188) 를 가지는 픽처 (178) 를 포함한다. 이하에서 더욱 상세하게 설명된 바와 같이, 그리고 이 개시물의 기법들에 따르면, 제 2 팔레트들 (192) 은 제 1 팔레트들 (184) 에 기초한다. 픽처 (178) 는 또한, 인트라-예측 코딩 모드로 코딩된 블록 (196) 과, 인터-예측 코딩 모드로 코딩되는 블록 (200) 을 포함한다.

[0234] 도 4 의 기법들은 비디오 인코더 (20) (도 1 및 도 2) 및 비디오 디코더 (30) (도 1 및 도 3) 의 맥락에서, 그리고 설명의 목적들을 위한 HEVC 비디오 코딩 표준에 대하여 설명된다. 그러나, 이 개시물의 기법들은 이러한 방법으로 제한되는 것은 아니고, 다른 비디오 코딩 프로세스들 및/또는 표준들에서 다른 비디오 코딩 프로세서들 및/또는 디바이스들에 의해 적용될 수도 있다는 것을 이해해야 한다.

[0235] 일반적으로, 팔레트는 현재 코딩되고 있는 CU, 도 4 의 예에서의 CU (188) 에 대하여 지배적이고 및/또는 대표적인 다수의 픽셀 값들을 지칭한다. 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 은 다수의 팔레트들을 포함하는 것으로서 도시된다. 일부 예들에서, 이 개시물의 양태들에 따르면, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 CU 의 각각의 컬러 컴포넌트에 대하여 별도로 팔레트들을 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 의 루마 (Y) 컴포넌트에 대한 팔레트, CU 의 크로마 (U) 컴포넌트에 대한 또 다른 팔레트, 및 CU 의 크로마 (V) 컴포넌트에 대한 또 다른 팔레트를 인코딩할 수도 있다. 이 예에서, Y 팔레트의 엔트리들은 CU 의 픽셀들의 Y 값들을 표현할 수도 있고, U 팔레트의 엔트리들은 CU 의 픽셀들의 U 값들을 표현할 수도 있고, V 팔레트의 엔트리들은 CU 의 픽셀들의 V 값들을 표현할 수도 있다.

[0236] 다른 예들에서, 비디오 인코더 (20) 는 CU 의 모든 컬러 컴포넌트들에 대하여 단일 팔레트를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 Y_i , U_i , 및 V_i 를 포함하는, 트리플 값인 i 번째 엔트리를 가지는 팔레트를 인코딩할 수도 있다. 이 경우, 팔레트는 픽셀들의 컴포넌트들의 각각에 대한 값들을 포함한다.

따라서, 다수의 개별적인 팔레트들을 가지는 팔레트들의 세트로서의 제 1 팔레트들 (184 및 192) 의 표현은 단지 하나의 예이고 제한하는 것으로 의도된 것은 아니다.

[0237] 도 4 의 예에서, 제 1 팔레트들 (184) 은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3 을 각각 가지는 3 개의 엔트리들 (202 내지 206) 을 포함한다. 엔트리들 (202 내지 206) 은 팔레트 인덱스들을, 픽셀 값 A, 픽셀 값 B, 및 픽셀 값 C 를 각각 포함하는 픽셀 값들에 관련시킨다. 본원에서 설명된 바와 같이, 제 1 CU (180) 의 실제적인 픽셀 값들을 코딩하는 것이 아니라, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 팔레트 인덱스들 1 내지 3 을 이용하여 블록의 픽셀들을 코딩하기 위하여 팔레트-기반 코딩을 이용할 수도 있다. 즉, 제 1 CU (180) 의 각각의 픽셀 위치에 대하여, 비디오 인코더 (20) 는 픽셀에 대한 인덱스 값을 인코딩할 수도 있고, 여기서, 인덱스 값은 제 1 팔레트들 (184) 중의 하나 이상에서의 픽셀 값과 연관된다. 비디오 디코더 (30) 는 비트스트림으로부터 팔레트 인덱스들을 획득할 수도 있고, 팔레트 인덱스들 및 제 1 팔레트들 (184) 중의 하나 이상을 이용하여 픽셀 값들을 재구성할 수도 있다. 이에 따라, 제 1 팔레트들 (184) 은 팔레트-기반 디코딩 시에 비디오 디코더 (30) 에 의한 이용을 위하여, 인코딩된 비디오 데이터 비트스트림에서 비디오 인코더 (20) 에 의해 송신된다.

[0238] 이 개시물의 양태들에 따르면, 최대 팔레트 크기는 제 1 팔레트들 (184) 에 대하여 시그널링될 수도 있다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예컨대, 제 1 팔레트들 (184) 내에 포함될 수도 있는 엔트리들의 수의 측면에서, 최대 팔레트 크기를 표시하는 데이터를 코딩할 수도 있다. 일부 예들에서, 최대 팔레트 크기 (예컨대, MAX_PLT_SIZE) 를 표시하는 하나 이상의 신택스 엘리먼트들은 CU (180) 에 대하여 활성인 SPS 내에 포함될 수도 있다. 다른 예들에서, 최대 팔레트 크기를 표시하는 하나 이상의 신택스 엘리먼트들은 VPS 또는 PPS 와 같은 또 다른 파라미터 세트 내에, 또는 슬라이스 헤더 데이터 또는 LCU 또는 CU 와 연관된 데이터와 같은 헤더 데이터 내에 포함될 수도 있다.

[0239] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 크기를 표시하는 하나 이상의 신택스 엘리먼트들을 이용하여, 코딩되고 있는 비디오 데이터의 특정한 프로파일, 레벨, 또는 비트-심도에 기초할 수도 있는 최대 팔레트 크기를 변동시킬 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 크기를 표시하는 하나 이상의 신택스 엘리먼트들을 이용하여, CU (180) 와 같은, 코딩되고 있는 블록의 크기에 기초할 수도 있는 최대 팔레트 크기를 변동시킬 수도 있다.

[0240] 예시의 목적들을 위한 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CU (180) 에 대한 제 1 팔레트들 (184) 을 구성할 때에 최대 팔레트 크기를 표시하는 데이터를 이용할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 데이터에 의해 표시된 최대 팔레트 크기에 도달할 때까지, 엔트리들을 제 1 팔레트들 (184) 에 추가하는 것을 계속할 수도 있다. 다음으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 구성된 제 1 팔레트들 (184) 을 이용하여 CU (180) 를 코딩할 수도 있다.

[0241] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제 1 팔레트들 (184) 에 기초하여 제 2 팔레트들 (192) 을 결정할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 예측 팔레트들, 이 예에서는 제 1 팔레트들 (184) 이 그로부터 결정되는 하나 이상의 블록들을 위치시킬 수도 있다. 예측의 목적들을 위하여 이용되는 엔트리들의 조합은 예측자 팔레트로서 지칭될 수도 있다.

[0242] 도 4 의 예에서, 제 2 팔레트들 (192) 은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3 을 각각 가지는 3 개의 엔트리들 (208 내지 212) 을 포함한다. 엔트리들 (208 내지 212) 은 팔레트 인덱스들을, 픽셀 값 A, 픽셀 값 B, 및 픽셀 값 C 를 각각 포함하는 픽셀 값들에 관련시킨다. 이 예에서, 비디오 인코더 (20) 는 (예측자 팔레트가 다수의 블록들의 엔트리들을 포함할 수도 있지만, 예측자 팔레트를 표현하는) 제 1 팔레트들 (184) 의 어느 엔트리들이 제 2 팔레트들 (192) 내에 포함되는지를 표시하는 하나 이상의 신택스 엘리먼트들을 코딩할 수도 있다.

[0243] 도 4 의 예에서, 하나 이상의 신택스 엘리먼트들은 벡터 (216) 로서 예시된다. 벡터 (216) 는 다수의 연관된 빈들 (또는 비트들) 을 가지고, 각각의 빈은 그 빈과 연관된 팔레트 예측자가 현재의 팔레트의 엔트리를 예측하기 위하여 이용되는지 여부를 표시한다. 예를 들어, 벡터 (216) 는 제 1 팔레트들 (184) 의 첫 번째 2 개의 엔트리들 (202 및 204) 이 제 2 팔레트들 (192) 내에 포함 (벡터 (216) 에서의 "1" 의 값) 되는 반면, 제 1 팔레트들 (184) 의 세 번째 엔트리가 제 2 팔레트들 (192) 내에 포함되지 않는다는 것 (벡터 (216) 에서의 "0" 의 값) 을 표시한다. 도 4 의 예에서, 벡터는 불리언 벡터 (Boolean vector) 이다. 벡터는 팔레트 예측 벡터로서 지칭될 수도 있다.

- [0244] 일부 예들에서, 위에서 언급된 바와 같이, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 예측을 수행할 때에 (또한, 팔레트 예측자 테이블 또는 팔레트 예측자 리스트로서 지칭될 수도 있는) 팔레트 예측자를 결정할 수도 있다. 팔레트 예측자는 현재의 블록을 코딩하기 위한 팔레트의 하나 이상의 엔트리들을 예측하기 위하여 이용되는 하나 이상의 이웃하는 블록들의 팔레트들로부터의 엔트리들을 포함할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 리스트를 동일한 방식으로 구성할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 예측자의 어느 엔트리들이 현재의 블록을 코딩하기 위한 팔레트로 복사되어야 하는지를 표시하기 위하여 (백터 (216) 와 같은) 데이터를 코딩할 수도 있다.
- [0245] 이에 따라, 일부 예들에서, 이전에 디코딩된 팔레트 엔트리들은 팔레트 예측자로서의 이용을 위하여 리스트에서 저장된다. 이 리스트는 현재의 팔레트 모드 CU 에서 팔레트 엔트리들을 예측하기 위하여 이용될 수도 있다. 2 진 예측 백터는 리스트에서의 어느 엔트리들이 현재의 팔레트에서 재이용되는지를 표시하기 위하여 비트스트림에서 시그널링될 수도 있다. 2014 년 6 월 27 일자로 출원된 미국 가출원 제 62/018,461 호에서는, 런 렱스 코딩은 2 진 팔레트 예측자를 압축하기 위하여 이용된다. 예에서, 런 렱스 값은 0 번째 차수 지수-골롬 코드를 이용하여 코딩된다.
- [0246] 이 개시물의 양태들에 따르면, 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) (예컨대, 엔트로피 인코딩 유닛 (118) 및 엔트로피 디코딩 유닛 (150)) 는 도 1 의 예에 대하여 위에서 설명된 바와 같이, k 번째 차수 절단된 지수-골롬 (TEGk) 코드를 이용하여 블록의 팔레트에 대한 2 진 팔레트 예측 백터를 코딩 (예컨대, 각각 인코딩 및 디코딩) 하도록 구성될 수도 있다.
- [0247] 일부 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 표준 제출 문서 Seregin 등, "Non-SCCE3: Run-Length Coding for Palette Predictor", JCTVC-R0228, Sapporo, JP, 2014 년 6 월 30 일 - 2014 년 7 월 9 일 (이하, JCTVC-R0228) 에서 설명된 기법들과 함께, TEGk 코드를 이용하여 2 진 팔레트 예측 백터를 코딩하도록 구성될 수도 있다. JCTVC-R0228 에서, 런 렱스 코딩은 다음의 조건들 및 단계들로 2 진 백터에서의 제로 엘리먼트들을 코딩하기 위하여 이용된다:
- [0248] · 1 과 동일한 런 렱스 값은 예측의 종료를 표시한다
 - [0249] · 예측의 종료는 2 진 백터에서의 최후의 1 에 대하여 시그널링되지 않는다
 - [0250] · 선행하는 제로 엘리먼트들의 수는 2 진 백터에서 매 1 에 대하여 코딩된다
 - [0251] · 제로 엘리먼트들의 수가 0 보다 더 클 경우, 수 플러스 1 은 1 의 탈출 값으로 인해 시그널링된다
 - [0252] · 런 렱스 값은 0-차수 지수 골롬 코드를 이용하여 코딩된다
- [0253] 예시의 목적들을 위한 예에서, 2 진 팔레트 예측 백터는 {1100100010000} 와 동일할 수도 있어서, 이것은 팔레트 예측자의 (4 개의 1 들에 의해 표시된) 4 개의 엔트리들이 현재의 블록을 코딩하기 위한 팔레트로 복사된다는 것을 표시할 수도 있다. 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 백터를 0-0-3-4-1 로서 코딩할 수도 있다.
- [0254] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 팔레트 예측자 리스트에서의 팔레트 엔트리들의 수 마이너스 스캔 순서에서의 현재의 위치 마이너스 1 과 동일할 수도 있는, 백터에 대한 최대 런 값 X 를 이용하여 2 진 팔레트 예측 백터를 코딩할 수도 있다. 하나의 예에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 런 값을 코딩하기 위한 TEG0 코드를 이용한다.
- [0255] 이 개시물의 기법들은 또한, 최대 팔레트 예측자 크기를 시그널링하는 것에 관한 것이다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 예컨대, 백터 (216) 내에 포함될 수 있는 비트들의 수의 측면에서, 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 하나 이상의 다른 값들에 관련하여 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩할 수도 있다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 예측자 크기와 최대 팔레트 크기 사이의 델타 (예컨대, 차이) 와 같은 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하도록 구성될 수도 있다. 일부 사례들에서, 비디오 인코더 및 비디오 디코더 (30) 는 고정 길이 코드, 골롬-라이스 코드, 또는 지수 골롬 코드 중의 적어도 하나를 이용하여 델타를 코딩할 수도 있다.
- [0256] 일부 예들에서, 최대 팔레트 예측자 크기를 표시하는 선택스 엘리먼트들은 SPS 내에 포함될 수도 있다. 다른 예들에서, 최대 팔레트 예측자 크기를 표시하는 선택스 엘리먼트들은 VPS 또는 PPS 와 같은 또 다른 파라미

터 세트 내에, 또는 슬라이스 헤더 데이터 또는 LCU 또는 CU 와 연관된 데이터와 같은 헤더 데이터 내에 포함될 수도 있다.

- [0257] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 예측자 크기를 표시하는 선택스 엘리먼트들을 이용하여, 코딩되고 있는 비디오 데이터의 특정한 프로파일, 레벨, 또는 비트-심도에 기초할 수도 있는 최대 팔레트 예측자 크기를 변동시킬 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 팔레트 예측자 크기를 표시하는 선택스 엘리먼트들을 이용하여, 코딩되고 있는 블록의 크기에 기초할 수도 있는 최대 팔레트 예측자 크기를 변동시킬 수도 있다.
- [0258] 예시의 목적들을 위한 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CU (188) 를 코딩하기 위한 제 2 팔레트들 (192) 을 구성할 때에 최대 팔레트 예측자 크기에 관한 데이터를 이용할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 데이터에 의해 표시된 바와 같은 최대 팔레트 예측자 크기에 도달할 때까지, 엔트리들을 예측자 팔레트에 (예컨대, 그리고 비트들을 벡터 (216) 에) 추가하는 것을 계속할 수도 있다. 다음으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 CU (188) 에 대한 제 2 팔레트들 (192) 을 구성하기 위하여 벡터 (216) 를 이용할 수도 있다.
- [0259] 도 5 는 이 개시물의 기법들과 부합하는, 픽셀들의 블록에 대한 팔레트로의 팔레트 인덱스들을 결정하는 예를 예시하는 개념도이다. 예를 들어, 도 5 는 팔레트 인덱스들과 연관된 픽셀들의 개개의 위치들을 팔레트들 (244) 의 엔트리에 관련시키는 팔레트 인덱스들의 맵 (240) 을 포함한다. 예를 들어, 인덱스 1 은 값 A 와 연관되고, 인덱스 2 는 값 B 와 연관되고, 인덱스 3 은 값 C 와 연관된다. 게다가, 탈출 샘플들이 명시적 탈출 시그널링을 이용하여 표시될 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 인덱스 4 와 연관된 맵 (240) 의 샘플들이 탈출 샘플들이라는 것을 표시할 수도 있는, 인덱스 4 로서 도 5 에서 예시된 추가적인 인덱스를 팔레트들 (244) 에 추가할 수도 있다. 이 경우, 비디오 인코더 (20) 는 픽셀 값이 팔레트들 (244) 내에 포함되지 않을 경우에, 맵 (240) 에서의 위치에 대한 실제적인 픽셀 값 (또는 그 양자화된 버전) 의 표시를 인코딩할 수도 있다 (그리고 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 이를 획득할 수도 있다).
- [0260] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 어느 픽셀 위치들이 팔레트 인덱스들과 연관되는지를 표시하는 추가적인 맵을 코딩하도록 구성될 수도 있다. 예를 들어, 맵에서의 (i, j) 엔트리는 CU 의 (i, j) 위치에 대응할 수도 있는 것으로 가정한다. 비디오 인코더 (20) 는 엔트리가 연관된 인덱스 값을 가지는지 여부를 표시하는, 맵의 각각의 엔트리 (즉, 각각의 픽셀 위치) 에 대한 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 에서의 (i, j) 로케이션에서의 픽셀 값이 팔레트들 (244) 에서의 값들 중의 하나인 것을 표시하기 위하여 1 의 값을 가지는 플래그를 인코딩할 수도 있다.
- [0261] 비디오 인코더 (20) 는 이러한 예에서, 또한, 그 픽셀 값을 팔레트에서 표시하고 비디오 디코더가 픽셀 값을 재구성하도록 하기 위하여 (도 5 의 예에서 값들 1 내지 3 으로서 도시된) 팔레트 인덱스를 인코딩할 수도 있다.
- [0262] 팔레트들 (244) 이 단일 엔트리 및 연관된 픽셀 값을 포함하는 사례들에서, 비디오 인코더 (20) 는 인덱스 값의 시그널링을 스킵할 수도 있다. 비디오 인코더 (20) 는 CU 에서의 (i, j) 로케이션에서의 픽셀 값이 팔레트들 (244) 에서의 값들 중의 하나가 아니라는 것을 표시하기 위하여 제로의 값을 가지는 플래그를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 또한, 픽셀 값을 재구성함에 있어서 비디오 디코더 (30) 에 의한 이용을 위하여 픽셀 값의 표시를 인코딩할 수도 있다. 일부 사례들에서, 픽셀 값은 손실 방식으로 코딩될 수도 있다.
- [0263] CU 의 하나의 위치에서의 픽셀의 값은 CU 의 다른 위치들에서의 하나 이상의 다른 픽셀들의 값들의 표시를 제공할 수도 있다. 예를 들어, CU 의 이웃하는 픽셀 위치들이 동일한 픽셀 값을 가질 것이거나, (하나를 초과하는 픽셀 값이 단일 인덱스 값으로 맵핑될 수도 있는 손실 코딩의 경우에) 동일한 인덱스 값으로 맵핑될 수도 있다는 상대적으로 높은 확률이 있을 수도 있다.
- [0264] 따라서, 비디오 인코더 (20) 는 함께 코딩되는 주어진 스캔 순서에서의 연속적인 픽셀들 또는 팔레트 인덱스들의 수를 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 위에서 언급된 바와 같이, 팔레트 인덱스들 (또는 팔레트 인덱스들에 의해 표시된 픽셀 값들) 의 스트림은 런 (run) 으로서 본원에서 지칭될 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 런을 표시하는 선택스 엘리먼트들을 획득할 수도 있고, 동일한 픽셀 또는 인덱스 값을 가지는 연속적인 로케이션들의 수를 결정하기 위하여 데이터를 이용할 수도 있다.
- [0265] 위에서 언급된 바와 같이, 런들은 CopyFromTop 또는 값 모드와 함께 이용될 수도 있다. 예시의 목적들을 위

한 예에서, 맵 (240) 의 행 (row) 들 (264 및 268) 을 고려한다. 수평의 좌측에서 우측 스캔 방향을 가정하면, 행 (264) 은 "1" 의 3 개의 팔레트 인덱스들, "2" 의 2 개의 팔레트 인덱스들, 및 "3" 의 3 개의 팔레트 인덱스들을 포함한다. 행 (268) 은 "1" 의 5 개의 팔레트 인덱스들, "3" 의 2 개의 팔레트 인덱스들, 및 탈출 샘플로서 지칭될 수도 있는, (샘플-레벨 탈출 플래그가 명시적 탈출 시그널링을 위하여 이용될 수도 있지만, 인덱스 4 에 의해 표현된) 팔레트들 (244) 내에 포함되지 않는 하나의 샘플을 포함한다.

[0266] 이 예에서, 비디오 인코더 (20) 는 행 (268) 에 대한 데이터를 인코딩하기 위하여 CopyFromTop 모드를 이용할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 행 (268) 의 제 1 위치 (행 (268) 의 가장 좌측 위치) 가 행 (264) 의 제 1 위치와 동일하다는 것을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 행 (268) 에서의 스캔 방향에서 2 개의 연속적인 엔트리들의 다음 런이 행 (264) 의 제 1 위치와 동일하다는 것을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다.

[0267] (위에서 언급된) 행 (264) 의 제 1 위치 및 2 개의 엔트리들의 런을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩한 후, 비디오 인코더 (20) 는 값 모드를 이용하여, (좌측으로부터 우측으로) 행 (268) 에서의 제 4 및 제 5 위치들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 제 4 위치에 대한 1 의 값을 표시하는 하나 이상의 신택스 엘리먼트들과, 1 의 런 (예컨대, 값 모드) 을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 이 때문에, 비디오 인코더 (20) 는 또 다른 라인에 대한 참조 없이 이 2 개의 위치들을 인코딩한다.

[0268] 다음으로, 비디오 인코더 (20) 는 상부 행 (264) 에 관련하여 CopyFromTop 모드를 이용하여, 행 (268) 에서 3 의 인덱스 값을 가지는 제 1 위치를 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CopyFromTop 모드 및 1 의 런을 시그널링할 수도 있다. 따라서, 비디오 인코더 (20) 는 예컨대, 런을 이용하여, 라인의 다른 값들에 관련하여 라인의 픽셀 값들 또는 팔레트 인덱스들을 코딩하는 것, 또 다른 라인 (또는 열 (column)) 의 값들에 관련하여 라인의 픽셀 값들 또는 팔레트 인덱스들을 코딩하는 것, 또는 그 조합 사이에서 선택할 수도 있다. 비디오 인코더 (20) 는 일부 예들에서, 선택을 행하기 위하여 레이트/왜곡 최적화를 수행할 수도 있다.

[0269] 다음으로, 비디오 인코더 (20) 는, 제 1 팔레트들 (244) 내에 포함되지 않는 (좌측으로부터 우측으로) 행 (268) 의 최종 샘플에 대한 탈출 샘플을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 행 (268) 의 최종 위치를 탈출 샘플로서 인코딩할 수도 있다. 즉, 비디오 인코더 (20) 는 행 (268) 의 최종 위치가 탈출 샘플 (예컨대, 인덱스 4) 이라는 표시뿐만 아니라, 샘플 값의 표시를 인코딩할 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 상기 설명된 신택스를 획득할 수도 있고, 이러한 신택스를 이용하여 행 (268) 을 재구성할 수도 있다.

[0270] 위에서 언급된 바와 같이, 탈출 샘플을 코딩하기 위한 2 개의 기법들이 있을 수도 있다. 예를 들어, 명시적 탈출 시그널링으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 맵 (240) 의 각각의 샘플 위치에 대한 명시적 샘플-당 탈출 모드 플래그를 코딩할 수도 있다. (행 (268) 의 최종 샘플과 같은) 특정한 샘플이 탈출 샘플로서 코딩될 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 특정한 샘플에 대한 컬러 값을 표시하는 데이터를 코딩할 수도 있다. 샘플이 탈출 샘플로서 코딩되지 않을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 SPoint 플래그와 같은, 모드가 CopyFromTop 또는 값인지 여부를 표시하기 위한 추가적인 데이터를 코딩할 수도 있다.

[0271] 묵시적 탈출 시그널링으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 추가적인 인덱스를 팔레트들 (244) 에 추가할 수도 있다 (엔트리 인덱스 4). 비디오 인코더 (20) 및 비디오 디코더 (30) 는 샘플이 탈출 샘플, 예컨대, 인덱스 4 로서 코딩된다는 것을 표시하기 위하여 팔레트들 (244) 로의 추가적인 인덱스를 이용할 수도 있다. 그러나, 추가적인 인덱스는 연관된 컬러 값을 가지지 않는다. 오히려, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 추가적인 인덱스와 연관되는 각각의 샘플에 대한 컬러 값들을 코딩한다. 샘플이 탈출 샘플로서 코딩되지 않을 경우, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 SPoint 플래그와 같은, 모드가 CopyFromTop 또는 값인지 여부를 표시하기 위한 데이터를 코딩할 수도 있다.

[0272] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 데이터의 블록의 모든 샘플들에 대하여, 블록의 적어도 하나의 샘플이 블록에 대한 컬러들의 팔레트 내에 포함되지 않은 컬러 값에 기초하여 코딩되는지 여부를 표시하는 하나 이상의 블록-레벨 신택스 엘리먼트들을 코딩하도록 구성될 수도 있다. 도 5 의 예에 대하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 맵 (240) 의 적어도 하나의 샘플이 탈출 샘플, 즉, 행 (268) 의 최종 샘플로서 코딩된다는 것을 표시하는, 맵 (240) 과 연관된 하나 이상의 신택스

엘리먼트들을 코딩할 수도 있다.

- [0273] 예에서, 하나 이상의 선택스 엘리먼트들은 블록-레벨 탈출 플래그 (이하에서, 간단하게 "탈출 플래그" 로서 지칭됨) 일 수도 있다. 예를 들어, 비디오 인코더 (20) 는 맵 (240) 이 탈출 샘플로서 코딩된 샘플을 포함한다는 것을 표시하기 위하여 1 의 값을 가지는 탈출 플래그를 인코딩할 수도 있다. 마찬가지로, 비디오 디코더 (30) 는, 맵 (240) 이 탈출 샘플로서 코딩된 샘플을 포함한다는 것을 표시하는, 1 의 값을 가지는 탈출 플래그를 디코딩할 수도 있다. 따라서, 탈출 플래그에 따라, 비디오 인코더 (20) 는 맵 (240) 을 인코딩할 수도 있고, 비디오 디코더 (30) 는 맵을 디코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인덱스 4 를, 탈출 샘플들로서 코딩된 샘플들을 표현하기 위하여 이용될 수도 있는 제 1 팔레트들 (244) 에 추가할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 맵 (240) 의 코딩 동안에 이 추가적인 인덱스를 이용할 수도 있다.
- [0274] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 탈출 플래그와, 특정한 블록을 코딩하기 위하여 이용되고 있는 팔레트의 크기에 기초하여, 어떤 선택스의 코딩을 스킵하도록 구성될 수도 있다. 즉, 도 5 의 예는 3 개의 엔트리들을 가지는 제 1 팔레트들 (244) 을 예시하지만, 일부 사례들에서, 비디오 데이터의 블록을 코딩하기 위한 팔레트는 단일 엔트리를 포함할 수도 있다. 이러한 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 단일 엔트리를 가지는 팔레트와, 블록의 샘플들이 탈출 샘플들로서 코딩되지 않는다는 것을 표시하는 탈출 플래그에 기초하여, 어떤 탈출-관련된 선택스의 코딩을 스킵하도록 구성될 수도 있다.
- [0275] 예를 들어, 탈출 플래그가 블록의 샘플들이 탈출 샘플들이 아니라는 것을 표시하는 사례들에서와 같이, 그리고 팔레트의 크기가 1 일 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록의 모든 샘플들이 동일한 인덱스 값 (예컨대, 오직 팔레트의 엔트리) 을 가진다는 것을 추론하도록 구성될 수도 있다. 그러므로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 블록의 팔레트 인덱스들을 결정하기 위하여 이용된 다른 모든 다른 데이터의 코딩을 스킵할 수도 있다.
- [0276] 상기 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록-레벨 탈출 선택스를 명시적으로 코딩할 수도 있다 (예컨대, 비디오 인코더 (20) 는 비트스트림에서 탈출 플래그를 인코딩할 수도 있고, 비디오 디코더 (30) 는 비트스트림으로부터 이러한 플래그를 디코딩할 수도 있음). 그러나, 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 맵 (240) 을 코딩하기 위하여 이용된 팔레트들 (244) 의 크기에 기초하여 블록-레벨 탈출 선택스 엘리먼트의 값을 추론 (예컨대, 상기 언급된 선택스 엘리먼트를 인코딩하거나 디코딩하지 않으면서 결정) 할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록-레벨 탈출 플래그의 값을 결정하기 위하여, 팔레트 크기에 관한 예비적 결정을 행할 수도 있다. 블록-레벨 탈출 선택스 엘리먼트는 팔레트 크기가 제로보다 더 클 때에 비트스트림에서 오직 코딩될 수도 있다.
- [0277] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 데이터의 블록을 코딩하기 위한 팔레트의 크기를 초기에 결정할 수도 있다. 팔레트의 크기가 제로인 것에 기초하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 탈출 플래그가 1 과 동일하다는 것과, 샘플들을 코딩하기 위하여 이용가능한 다른 팔레트 엔트리들이 없으므로, 블록의 모든 샘플들이 탈출 샘플들로서 코딩된다는 것으로 결정할 수도 있다.
- [0278] 예를 들어, 팔레트 크기가 제로인 사례들에서와 같이, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록의 모든 샘플들이 동일한 인덱스 값 (예컨대, 탈출 샘플들을 표시하는 것과 연관된 팔레트의 추가적인 엔트리) 을 가지는 것으로 자동으로 결정하도록 구성될 수도 있다. 그러므로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 탈출 플래그뿐만 아니라, 블록의 팔레트 인덱스들을 결정하기 위하여 이용된 모든 다른 데이터의 코딩을 스킵할 수도 있다.
- [0279] 도 6 은 이 개시물의 기법들과 부합하는, 래스터 스캔 순서를 가정하여, CopyFromAbove 모드에 대한 최대 런 령스를 결정하는 예를 예시하는 개념도이다. 위에서 언급된 바와 같이, 이 개시물의 기법들은 코딩되고 있는 선택스의 최대 잠재적인 값을 참조하는 코드를 이용하여 팔레트 코딩을 위한 선택스를 코딩하는 것을 포함한다.
- [0280] 예시의 목적들을 위한 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 인덱스들의 런 (예컨대, 동일한 값을 가지는 인덱스들의 런, 또는 상부-이웃 인덱스들로부터 복사되는 인덱스들의 런) 의 런 령스를 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재의 팔레트-코딩된 샘플에 대하여, 현재의 샘플과 함께 코딩되고 있는 팔레트 인덱스들의 런의 런 령스를 결정할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔레트 인덱스들의 최대

런을 위한 최대 런 령스를 결정할 수도 있다. 다음으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 결정된 최대 런 령스에 기초하여 런 령스를 표시하는 데이터를 코딩할 수도 있다.

[0281] 일부 사례들에서, 이 개시물의 양태들에 따르면, 선택스는 본원에서 설명된 TEGk 코드와 같은 지수 골롬 코드의 형태를 이용하여 코딩될 수도 있다. TEGk 코드를 이용하기 위하여, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 런 령스를 현재의 CU 에서의 픽셀들의 수 마이너스 스캔 순서에서의 현재의 위치 마이너스 1 로서 결정할 수도 있다.

[0282] 일부 팔레트 코딩 기법들에서, CopyFromAbove 모드 (여기서, 비디오 코더가 현재의 픽셀 상부의 픽셀의 인덱스를 복사함) 와 연관된 픽셀들의 런들은 임의의 탈출 픽셀들을 포함하도록 허용되지 않는다. 즉, 비디오 코더는 현재의 픽셀의 상부-이웃 픽셀이 탈출 샘플로서 코딩된 픽셀일 경우에 CopyFromAbove 런을 정지시켜야 한다. 이 때문에, 최대 CopyFromAbove 런 령스는 현재의 픽셀 위치와, 스캔 순서에서 탈출되는 상부-이웃 픽셀을 가지는 위치 사이의 거리에 의해 경계가 정해진다.

[0283] 예시의 목적들을 위한 예에서, 스캔 순서에서의 CopyFromAbove 런의 시작 위치는 A 이고, 위치 $A+L$ ($L>0$) (또는 일부 예들에서, $A+L$ ($L>1$)) 에서의 픽셀에 대한 상부-이웃 픽셀은 탈출 샘플로서 코딩되고, 위치 $A+I$ ($I<L$) 에서의 픽셀에 대한 상부-이웃 픽셀은 탈출 샘플로서 코딩되지 않는다. 이러한 픽셀 L 이 존재하지 않을 경우, 비디오 인코더 (20) 는 스캔 순서에서 블록에서의 최후 픽셀 후의 위치에 L 을 배정할 수도 있다. 이 개시물의 기법들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최대 코딩된 런 령스가 $L-1$ 보다 더 길지 않다는 한정으로 CopyFromAbove 모드에 대한 런 령스를 코딩하기 위하여 TEGk 를 이용할 수도 있다. 대안적으로, 0 보다 더 크고, 1 보다 더 크고, 2 보다 더 큰 1 진 프리픽스들이 런 령스를 코딩할 때에 이용될 경우, 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 TEGk 를 이용하여 코딩되어야 할 인덱스 맵의 런의 최대 런 령스를 $L-4$ 로 설정할 수도 있다.

[0284] 비디오 디코더 (30) 또는 비디오 인코더 (20) 가 위치에서의 픽셀이 탈출 픽셀에 대응하는지 또는 아닌지 여부를 결정할 수 없는 사례들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 픽셀이 탈출 샘플로서 코딩되지 않는 것처럼 픽셀을 프로세싱할 수도 있고, 즉, 최대 코딩된 런 령스가 정지되지 않는다. 도 6 의 예에서, 파선들 (280) 에 의해 둘러싸인 픽셀들의 어느 것도 탈출 샘플로서 코딩되지 않을 경우, 최대 가능한 런 령스는 35 (즉, 음영처리되지 않은 픽셀 위치들의 수) 이다. 파선들 (280) 내의 픽셀들 중의 하나 이상이 탈출 샘플로서 코딩될 경우, 탈출 픽셀 ("X" 를 갖는 픽셀 위치) 로서 표기된 픽셀이 스캔 순서에서 파선들 (280) 내의 제 1 탈출 픽셀인 것으로 가정하면, 최대 가능한 코딩된 복사 상부 런 령스는 5 이다.

[0285] 일부 예들에서, 비디오 디코더 (30) 는 파선들 (280) 내의 픽셀들에 대한 런 모드 (픽셀들이 코딩되는 팔레트 모드) 를 오직 결정할 수도 있다. 이 때문에, 최악의 경우, 비디오 디코더 (30) 는 BlockWidth-1 픽셀들에 대한 결정을 행한다. 일부 예들에서, 비디오 디코더 (30) 는 런 모드가 검사되는 픽셀들의 수의 최대치에 관한 어떤 한정들을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 픽셀들이 현재의 픽셀과 동일한 행 내에 있을 경우에 파선들 (280) 내의 픽셀들을 오직 검사할 수도 있다. 비디오 디코더 (30) 는 파선들 (280) 내의 모든 다른 픽셀들이 탈출 샘플들로서 코딩되지 않는다는 것을 추론할 수도 있다. 도 6 에서의 예는 래스터 스캔 순서를 가정한다. 그러나, 기법들은 수직, 수평 횡단, 및 수직 횡단과 같은 다른 스캔 순서들에 적용될 수도 있다.

[0286] 도 7 은 이 개시물의 기법들과 부합하는, 블록의 임의의 샘플들이 탈출 샘플들로서 인코딩되는지 여부를 표시하는 하나 이상의 블록-레벨 선택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 인코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다. 도 7 의 프로세스는 예시의 목적들을 위하여 비디오 인코더 (20) 에 의해 수행되는 것으로서 일반적으로 설명되지만, 다양한 다른 프로세서들은 또한, 도 7 에서 도시된 프로세스를 수행할 수도 있다.

[0287] 도 7 의 예에서, 비디오 인코더 (20) 는 비디오 데이터의 블록을 인코딩하기 위한 팔레트를 결정한다 (300). 일부 예들에서, 비디오 인코더 (20) 는 예컨대, 팔레트 예측자를 이용하여, 하나 이상의 이전에 인코딩된 블록들의 팔레트-기반 팔레트들을 결정할 수도 있다. 비디오 인코더 (20) 는 또한, 팔레트의 크기를 결정할 수도 있다 (302). 예를 들어, 비디오 인코더 (20) 는 결정된 팔레트에서의 엔트리들의 수를 결정할 수도 있다.

[0288] 비디오 인코더 (20) 는 팔레트 크기가 제로인지 여부를 결정할 수도 있다 (304). 팔레트 크기가 제로와 동일한 것 (단계 (304) 의 예 브랜치) 에 기초하여, 비디오 인코더 (20) 는 블록의 적어도 하나의 샘플이 탈출 샘플

플이라는 것을 표시하는 그 블록-레벨 탈출 선택스를 결정할 수도 있다 (306). 하나의 예에서, 비디오 인코더 (20) 는 블록-레벨 탈출 플래그가 1 과 동일한 것으로 결정할 수도 있다. 다음으로, 비디오 인코더 (20) 는 블록에 대한 다른 선택스 엘리먼트들을 인코딩하지 않으면서, 블록의 모든 샘플들을 탈출 샘플들로서 인코딩한다 (308). 예를 들어, 블록-레벨 탈출 선택스 엘리먼트가 팔레트 크기가 제로보다 더 클 때에 비트스트림에서 오직 인코딩될 수도 있으므로, 비디오 인코더 (20) 는 블록-레벨 탈출 플래그의 표시를 인코딩하지 않을 수도 있다. 게다가, 비디오 인코더 (20) 는 팔레트 모드 (예컨대, 값 또는 CopyFromTop) 를 표시하는 선택스, 런들을 결정하는 것과 연관된 선택스, 팔레트 인덱스들과 연관된 선택스, 및 임의의 다른 관련된 선택스와 같은, 블록의 팔레트 인덱스들에 대한 다른 데이터를 인코딩하지 않을 수도 있다.

[0289] 팔레트 크기가 제로가 아닐 경우 (단계 (304) 의 아니오 브랜치), 비디오 인코더 (20) 는 블록의 임의의 샘플들이 탈출 샘플들로서 인코딩되는지 여부를 결정할 수도 있다 (310). 블록의 적어도 하나의 샘플이 탈출 샘플로서 인코딩되는 것으로 결정하는 것 (단계 (310) 의 예 브랜치) 에 기초하여, 비디오 인코더 (20) 는 블록의 적어도 하나의 샘플이 탈출 샘플로서 인코딩된다는 것을 표시하는 블록-레벨 탈출 선택스를 결정할 수도 있고, 블록을 갖는 비트스트림에서의 블록-레벨 선택스의 표시를 인코딩할 수도 있다 (312). 예를 들어, 비디오 인코더 (20) 는 블록-레벨 탈출 플래그를 1 과 동일하게 설정할 수도 있고, 비트스트림에서의 탈출 플래그의 표시를 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 블록의 적어도 하나의 픽셀을 탈출 샘플로서 인코딩하는 것을 포함하는 팔레트 코딩 모드들로 블록을 인코딩한다 (314).

[0290] 샘플들이 탈출 샘플들로서 인코딩되지 않는다는 것 (단계 (310) 의 아니오 브랜치) 에 기초하여, 비디오 인코더 (20) 는 블록에 대한 팔레트의 팔레트 크기가 1 과 동일한지 여부를 결정할 수도 있다 (316). 팔레트 크기가 1 과 동일하지 않는 것으로 결정하는 것 (단계 (316) 의 아니오 브랜치) 에 기초하여, 비디오 인코더 (20) 는 블록의 샘플들이 탈출 샘플들로서 인코딩되지 않는다는 것을 표시하는 블록-레벨 탈출 선택스를 결정할 수도 있고, 비트스트림에서의 블록-레벨 탈출 선택스의 표시를 인코딩할 수도 있다 (318). 예를 들어, 비디오 인코더 (20) 는 블록-레벨 탈출 플래그를 제로와 동일하게 설정할 수도 있고, 비트스트림에서의 탈출 플래그의 표시를 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 팔레트 코딩 모드들을 이용하지만, 임의의 탈출 샘플들을 인코딩하지 않으면서, 블록을 인코딩한다 (320). 예를 들어, 비디오 인코더 (20) 는 CopyFromTop 또는 값 모드들을 이용하여 블록의 팔레트 인덱스들을 인코딩할 수도 있고, 이러한 모드들의 이용과 연관된 선택스, 예컨대, 모드들, 팔레트 인덱스들, 런들 등을 표시하는 선택스를 인코딩할 수도 있다.

[0291] 팔레트 크기가 1 과 동일한 것으로 결정하는 것 (단계 (316) 의 예 브랜치) 에 기초하여, 비디오 인코더 (20) 는 블록의 샘플들이 탈출 샘플들로서 코딩되지 않는다는 것을 표시하는 블록-레벨 탈출 선택스를 결정할 수도 있고, 비트스트림에서의 블록-레벨 탈출 선택스의 표시를 인코딩할 수도 있다 (322). 예를 들어, 비디오 인코더 (20) 는 블록-레벨 탈출 플래그를 제로와 동일하게 설정할 수도 있고, 비트스트림에서의 탈출 플래그의 표시를 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 다른 선택스를 인코딩하지 않으면서, 블록의 모든 샘플들이 동일한 인덱스 값을 가진다는 표시를 인코딩한다 (324). 예를 들어, 비디오 인코더 (20) 는 팔레트 모드들의 이용과 연관된 선택스, 예컨대, 모드들, 팔레트 인덱스들, 런들 등을 표시하는 선택스의 인코딩을 스킵할 수도 있다.

[0292] 도 8 은 이 개시물의 기법들과 부합하는, 블록의 임의의 샘플들이 탈출 샘플들로서 디코딩되는지 여부를 표시하는 하나 이상의 블록-레벨 선택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 디코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다. 도 8 의 프로세스는 예시의 목적들을 위하여 비디오 디코더 (30) 에 의해 수행되는 것으로서 일반적으로 설명되지만, 다양한 다른 프로세서들은 또한, 도 8 에서 도시된 프로세스를 수행할 수도 있다.

[0293] 도 8 의 예에서, 비디오 디코더 (30) 는 비디오 데이터의 블록을 디코딩하기 위한 팔레트를 결정한다 (340). 일부 예들에서, 비디오 디코더 (30) 는 예컨대, 디코딩되고 있는 비트스트림에서 시그널링된 팔레트 예측자를 이용하여, 하나 이상의 이전에 인코딩된 블록들의 팔레트-기반 팔레트들을 결정할 수도 있다. 비디오 디코더 (30) 는 또한, 팔레트의 크기를 결정할 수도 있다 (342). 예를 들어, 비디오 디코더 (30) 는 결정된 팔레트에서의 엔트리들의 수를 결정할 수도 있다.

[0294] 비디오 디코더 (30) 는 팔레트 크기가 제로인지 여부를 결정할 수도 있다 (344). 팔레트 크기가 제로와 동일한 것 (단계 (344) 의 예 브랜치) 에 기초하여, 비디오 디코더 (30) 는 블록의 적어도 하나의 샘플이 탈출 샘플로서 인코딩된다는 것을 표시하는 블록-레벨 탈출 선택스를 결정할 수도 있다 (346). 하나의 예에서, 비디오 디코더 (30) 는 블록-레벨 탈출 플래그가 1 과 동일한 것으로 결정할 수도 있다. 예를 들어, 블록-레

벨 탈출 선택스 엘리먼트가 팔레트 크기가 제로보다 더 클 때에 비트스트림에서 오직 코딩될 수도 있으므로, 비디오 디코더 (30) 는 비트스트림으로부터 플래그를 디코딩하지 않으면서, 블록-레벨 탈출 플래그가 1 과 동일하다는 것을 추론할 수도 있다. 다음으로, 비디오 디코더 (30) 는 탈출 샘플과 연관된 컬러를 이용하여, 그리고 (예컨대, 탈출 샘플과 연관된 컬러 값 (들) 이외의) 블록에 대한 다른 선택스 엘리먼트들을 디코딩하지 않으면서, 블록의 모든 샘플들을 디코딩한다 (348). 예를 들어, 위에서 언급된 바와 같이, 비디오 디코더 (30) 는 블록-레벨 탈출 플래그의 표시를 디코딩하지 않을 수도 있다. 게다가, 비디오 디코더 (30) 는 팔레트 모드 (예컨대, 값 또는 CopyFromTop) 를 표시하는 선택스, 인덱스 값들을 표시하는 선택스, 런들을 결정하는 것과 연관된 선택스, 및 임의의 다른 관련된 선택스와 같은, 블록의 팔레트 인덱스들에 대한 다른 데이터를 디코딩하지 않을 수도 있다.

[0295] 팔레트 크기가 제로가 아닐 경우 (단계 (344) 의 아니오 브랜치), 비디오 디코더 (30) 는 비트스트림으로부터 블록-레벨 탈출 선택스를 디코딩할 수도 있고, 블록-레벨 선택스의 값을 결정할 수도 있다 (350). 예를 들어, 비디오 디코더 (30) 는 블록-레벨 탈출 플래그를 디코딩할 수도 있고, 값이 제로 또는 1 과 동일한지 여부를 결정할 수도 있다.

[0296] 다음으로, 비디오 디코더 (30) 는 예컨대, 디코딩된 선택스에 기초하여, 블록의 임의의 샘플들이 탈출 샘플로서 코딩되는지 여부를 결정할 수도 있다 (352). 블록의 적어도 하나의 샘플이 탈출 샘플로서 인코딩되는 것으로 결정하는 것 (단계 (352) 의 예 브랜치) 에 기초하여, 비디오 디코더 (30) 는 블록의 적어도 하나의 샘플을 탈출 샘플로서 디코딩하는 것을 포함하는 팔레트 코딩 모드들로 블록을 디코딩할 수도 있다 (354). 비디오 디코더 (30) 는 또한, 탈출 샘플들에 대응하는 적어도 하나의 컬러 값을 디코딩할 수도 있다.

[0297] 샘플들이 탈출 샘플들로서 인코딩되지 않는다는 것 (단계 (352) 의 아니오 브랜치) 에 기초하여, 비디오 디코더 (30) 는 블록에 대한 팔레트의 팔레트 크기가 1 과 동일한지 여부를 결정할 수도 있다 (356). 팔레트 크기가 1 과 동일하지 않은 것으로 결정하는 것 (단계 (356) 의 아니오 브랜치) 에 기초하여, 비디오 디코더 (30) 는 팔레트 코딩 모드들을 이용하지만, 임의의 탈출 샘플들을 디코딩하지 않으면서, 블록을 디코딩할 수도 있다 (358). 예를 들어, 비디오 디코더 (30) 는 CopyFromTop 또는 값 모드들을 이용하여 블록의 팔레트 인덱스들을 디코딩할 수도 있고, 이러한 모드들의 이용과 연관된 선택스, 예컨대, 모드들, 팔레트 인덱스들, 런들 등을 표시하는 선택스를 디코딩할 수도 있다.

[0298] 팔레트 크기가 1 과 동일한 것으로 결정하는 것 (단계 (356) 의 예 브랜치) 에 기초하여, 비디오 디코더 (30) 는 팔레트의 팔레트 엔트리 (예컨대, 팔레트에서 존재하는 오직 엔트리) 를 이용하여, 그리고 다른 선택스를 디코딩하지 않으면서, 블록을 디코딩할 수도 있다 (360). 예를 들어, 비디오 디코더 (30) 는 팔레트 모드들의 이용과 연관된 선택스, 예컨대, 모드들, 팔레트 인덱스들, 런들 등을 표시하는 선택스의 디코딩을 스킵할 수도 있다.

[0299] 도 9 는 이 개시물의 기법들과 부합하는, 최대 팔레트 크기 및 최대 팔레트 예측자 크기를 표시하는 하나 이상의 선택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 인코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다. 도 9 의 프로세스는 예시의 목적들을 위하여 비디오 인코더 (20) 에 의해 수행되는 것으로서 일반적으로 설명되지만, 다양한 다른 프로세서들은 또한, 도 9 에서 도시된 프로세스를 수행할 수도 있다.

[0300] 도 9 의 예에서, 비디오 인코더 (20) 는 팔레트 모드에서 비디오 데이터의 현재의 블록을 인코딩하기 위한 팔레트의 최대 크기를 결정할 수도 있다 (380). 예를 들어, 비디오 인코더 (20) 는 코딩되고 있는 비디오 데이터의 특성에 기초하여 최대 팔레트 크기를 결정하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 데이터의 비트-심도 (예컨대, 입력 비트-심도 또는 프로파일 비트-심도), 블록의 블록 크기, 비디오 데이터와 연관된 프로파일 또는 레벨 등에 기초하여 최대 팔레트 크기를 결정할 수도 있다.

[0301] 비디오 인코더 (20) 는 또한, 현재의 블록의 팔레트를 생성하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 결정할 수도 있다 (382). 예를 들어, 비디오 인코더 (20) 는 코딩되고 있는 비디오 데이터의 특성에 기초하여 최대 팔레트 예측자 크기를 결정하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 데이터의 비트-심도 (예컨대, 입력 비트-심도 또는 프로파일 비트-심도), 블록의 블록 크기, 비디오 데이터와 연관된 프로파일 또는 레벨 등에 기초하여 최대 팔레트 예측자 크기를 결정할 수도 있다.

[0302] 비디오 인코더 (20) 는 또한, 현재의 블록을 포함하는 비트스트림에서 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 인코딩한다 (386). 일부 예들에서, 비디오 인코더 (20) 는 하나 이상의 다른 값들에 관련하여 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 인코딩할 수도

있다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 는 최대 팔레트 예측자 크기와 최대 팔레트 크기 사이의 델타 (예컨대, 차이) 와 같은 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하도록 구성될 수도 있다.

[0303] 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 는 SPS 에서 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 하나 이상의 선택스 엘리먼트들을 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 또 다른 파라미터 세트 (예컨대, PPS) 에서, 현재의 블록을 포함하는 슬라이스의 슬라이스 헤더에서, 또는 비트스트림에서의 다른 곳에서의 이러한 선택스를 인코딩할 수도 있다.

[0304] 비디오 인코더 (20) 는 또한, 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터에 따라 현재의 블록을 인코딩한다 (388). 예를 들어, 비디오 인코더 (20) 는 최대 팔레트 크기에 의해 제한되는 팔레트, 및/또는 최대 팔레트 예측자 크기에 의해 제한되는 팔레트 예측자를 결정할 수도 있다.

[0305] 도 10 은 이 개시물의 기법들과 부합하는, 최대 팔레트 크기 및 최대 팔레트 예측자 크기를 표시하는 하나 이상의 선택스 엘리먼트들에 기초하여 비디오 데이터의 블록을 인코딩하기 위한 일 예의 프로세스를 예시하는 플로우차트이다. 도 10 의 프로세스는 예시의 목적들을 위하여 비디오 디코더 (30) 에 의해 수행되는 것으로서 일반적으로 설명되지만, 다양한 다른 프로세서들은 또한, 도 10 에서 도시된 프로세스를 수행할 수도 있다.

[0306] 도 10 의 예에서, 비디오 디코더 (30) 는 팔레트 모드에서 디코딩되고 있는 현재의 블록을 포함하는 비트스트림으로부터 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 디코딩한다 (400). 일부 예들에서, 비디오 인코더 (20) 는 하나 이상의 다른 값들에 관련하여 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터를 인코딩할 수도 있다. 예를 들어, 이 개시물의 양태들에 따르면, 비디오 인코더 (20) 는 최대 팔레트 예측자 크기와 최대 팔레트 크기 사이의 델타 (예컨대, 차이) 와 같은 최대 팔레트 예측자 크기를 표시하는 데이터를 코딩하도록 구성될 수도 있다.

[0307] 이 개시물의 양태들에 따르면, 비디오 디코더 (30) 는 SPS 로부터 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다. 다른 예들에서, 비디오 디코더 (30) 는 또 다른 파라미터 세트 (예컨대, PPS) 로부터, 현재의 블록을 포함하는 슬라이스의 슬라이스 헤더로부터, 또는 비트스트림에서의 다른 곳에서 이러한 선택스를 디코딩할 수도 있다.

[0308] 비디오 디코더 (30) 는 디코딩된 데이터에 기초하여 현재의 블록을 디코딩하기 위한 팔레트의 최대 크기를 결정할 수도 있다 (402). 비디오 디코더 (30) 는 또한, 데이터에 기초하여 현재의 블록에 대한 팔레트를 생성하기 위한 팔레트 예측자의 최대 팔레트 예측자 크기를 결정할 수도 있다 (404). 비디오 디코더 (30) 는 또한, 최대 팔레트 크기 및/또는 최대 팔레트 예측자 크기를 표시하는 데이터에 따라 현재의 블록을 디코딩한다 (408). 예를 들어, 비디오 디코더 (30) 는 최대 팔레트 크기에 의해 제한되는 팔레트, 및/또는 최대 팔레트 예측자 크기에 의해 제한되는 팔레트 예측자를 결정할 수도 있다.

[0309] 도 11 은 이 개시물의 기법들과 부합하는, 최대 잠재적 런 령스에 기초하여 픽셀들의 런의 런 령스를 표시하는 데이터를 코딩 (인코딩 또는 디코딩) 하기 위한 일 예의 프로세스를 예시하는 플로우차트이다. 도 11 의 프로세스는 예시의 목적들을 위하여, 비디오 인코더 (20) 또는 비디오 디코더 (30) 과 같은 비디오 코더에 의해 수행되는 것으로서 일반적으로 설명되지만, 다양한 다른 프로세서들은 또한, 도 11 에서 도시된 프로세스를 수행할 수도 있다.

[0310] 도 11 의 예에서, 비디오 코더는 현재의 픽셀을 코딩하기 위한 팔레트 모드를 결정할 수도 있다 (420). 예를 들어, 비디오 코더는 현재의 픽셀이 CopyFromTop 모드, 값 모드, 또는 또 다른 팔레트-기반 코딩 모드를 이용하여 코딩되는지 여부를 결정할 수도 있다. 비디오 코더는 또한, 현재의 픽셀에 대한 런의 런 령스를 결정한다 (422). 예를 들어, 비디오 코더는 현재의 픽셀의 팔레트 인덱스로 코딩되고 있는 팔레트 인덱스들의 수를 결정한다.

[0311] 비디오 코더는 또한, 런에 대한 최대 런 령스를 결정한다 (424). 예를 들어, 비디오 코더는 현재의 픽셀의 팔레트 인덱스로 코딩될 수 있는 팔레트 인덱스들의 최대 런을 위한 최대 런 령스를 결정할 수도 있다. 예에서, 비디오 코더는 픽셀을 포함하는 비디오 데이터의 블록에서의 픽셀들의 수를 결정할 수도 있다. 비디오 코더는 또한, 팔레트 인덱스들을 스캐닝하기 위하여 이용된 스캔 순서에 기초하여 블록에서의 현재의 픽셀의 위치를 결정할 수도 있다. 다음으로, 비디오 코더는 최대 런 령스를 블록에서의 픽셀들의 수 마이너스 현재의 픽셀의 픽셀 위치 마이너스 1 로서 결정한다.

[0312] 비디오 코더는 또한, 결정된 최대 런 령스에 기초하여 런의 런 령스를 표시하는 데이터를 코딩한다 (426).

예를 들어, 이 개시물의 양태들에 따르면, 비디오 코더는 TEGk 코드를 이용하여 런 령스를 표시하는 데이터를 코딩할 수도 있다.

[0313] 예에 따라서는, 본원에서 설명된 기법들 중의 임의의 것의 어떤 액트(act)들 또는 이벤트(event)들이 상이한 시퀀스에서 수행될 수 있거나, 추가될 수도 있거나, 병합될 수도 있거나, 또는 모두 배제(예를 들어, 모든 설명된 액트들 또는 이벤트들이 기법들의 실시를 위해 필요한 것은 아님)될 수도 있다는 것을 인식해야 한다.

또한, 어떤 예들에서는, 액트들 또는 이벤트들이 순차적인 것이 아니라, 예컨대, 멀티-스레딩된(multi-threaded) 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해 동시에 수행될 수도 있다. 게다가, 이 개시물의 어떤 양태들은 명료함의 목적들을 위하여 단일 모듈 또는 유닛에 의해 수행되는 것으로 설명되지만, 이 개시물의 기법들은 비디오 코더와 연관된 유닛들 또는 모듈들의 조합에 의해 수행될 수도 있는 것을 이해해야 한다.

[0314] 이 개시물의 어떤 양태들은 예시의 목적들을 위하여 개발 중인 HEVC 표준에 대하여 설명되었다. 그러나, 이 개시물에서 설명된 기법들은 아직 개발되지 않은 다른 표준 또는 전용 비디오 코딩 프로세스들을 포함하는 다른 비디오 코딩 프로세스들에 대해 유용할 수도 있다.

[0315] 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 있고, 이들 양자는 일반적으로 비디오 코더로서 지칭될 수도 있다. 마찬가지로, 비디오 코딩은 적용가능한 바와 같이, 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.

[0316] 기법들의 다양한 양태들의 특정 조합들이 위에서 설명되지만, 이 조합들은 이 개시물에서 설명된 기법들의 예들을 단지 예시하기 위하여 제공된다. 따라서, 이 개시물의 기법들은 이 예의 조합들에 제한되지 않아야 하고, 이 개시물에서 설명된 기법들의 다양한 양태들의 임의의 상상가능한 조합을 망라할 수도 있다.

[0317] 하나 이상의 예들에서는, 설명된 기능들이 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현될 경우, 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 컴퓨터-판독가능 매체를 통해 송신될 수도 있고, 하드웨어-기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터-판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터-판독가능 저장 매체들, 또는 예컨대, 통신 프로토콜에 따라 하나의 장소로부터 또 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비-일시적 유형의 컴퓨터-판독가능 저장 매체들, 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 이 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독가능 매체를 포함할 수도 있다.

[0318] 제한이 아닌 예로서, 이러한 컴퓨터-판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태로 회망하는 프로그램 코드를 저장하기 위하여 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속은 컴퓨터-판독가능 매체로 적절하게 칭해진다. 예를 들어, 동축 케이블, 광섬유 케이블, 트위스트 페어(twisted pair), 디지털 가입자 회선(digital subscriber line; DSL), 또는 적외선, 라디오(radio), 및 마이크로파(microwave)와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신될 경우, 동축 케이블, 광섬유 케이블, 트위스트 페어, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체들 및 데이터 저장 매체들은 접속들, 반송파들, 신호들, 또는 다른 일시적 매체(transient medium)들을 포함하는 것이 아니라, 그 대신에, 비-일시적인, 유형의 저장 매체들에 관한 것이라는 것을 이해해야 한다. 본원에서 이용된 바와 같은 디스크(disk) 및 디스크(disc)는 콤팩트 디스크(compact disc; CD), 레이저 디스크(laser disc), 광학 디스크(optical disc), 디지털 다기능 디스크(digital versatile disc; DVD), 플로피 디스크(floppy disk) 및 블루레이 디스크(Blu-ray disc)를 포함하고, 여기서 디스크(disk)들은 통상 데이터를 자기적으로 재생하는 반면, 디스크(disc)들은 데이터를 레이저들로 광학적으로 재생한다. 상기의 조합들은 컴퓨터-판독가능 매체들의 범위 내에 또한 포함되어야 한다.

[0319] 명령들은 하나 이상의 디지털 신호 프로세서들(DSP들), 범용 마이크로프로세서들, 애플리케이션 특정 집적 회로들(ASIC들), 필드 프로그래밍가능 로직 어레이들(FPGA들), 또는 다른 등가의 집적된 또는 별도의 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 이용된 바와 같은 용어 "프로

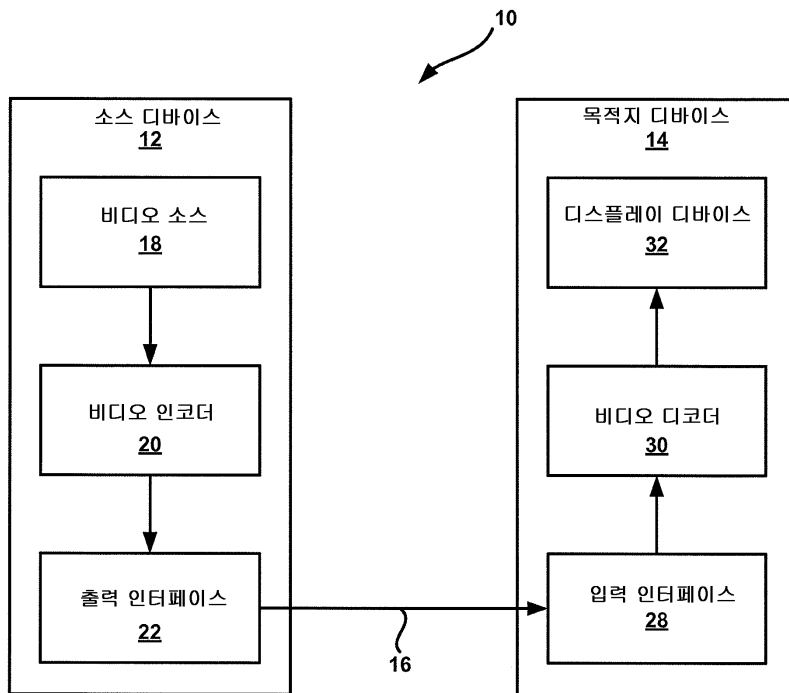
세서" 는 상기한 구조, 또는 본원에서 설명된 기법들의 구현을 위해 적당한 임의의 다른 구조 중의 임의의 것을 지칭할 수도 있다. 게다가, 일부의 양태들에서는, 본원에서 설명된 기능성이 인코딩 및 디코딩을 위해 구성되거나 조합된 코덱 내에 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에서 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0320] 이 개시물의 기법들은 무선 핸드셋 (wireless handset), 집적 회로 (integrated circuit; IC) 또는 IC 들의 세트 (예를 들어, 칩셋) 를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위하여 이 개시물에서 설명되어 있지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛 내에 조합될 수도 있거나, 적당한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는 상호동작하는 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

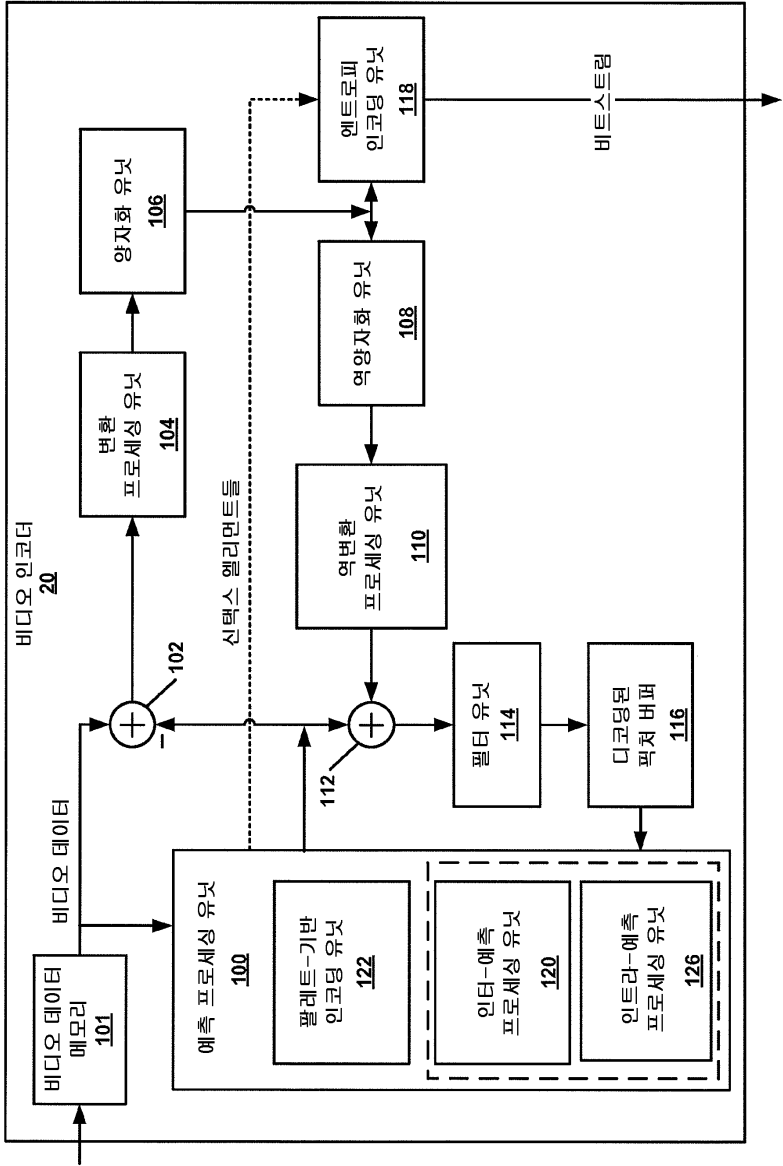
[0321] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

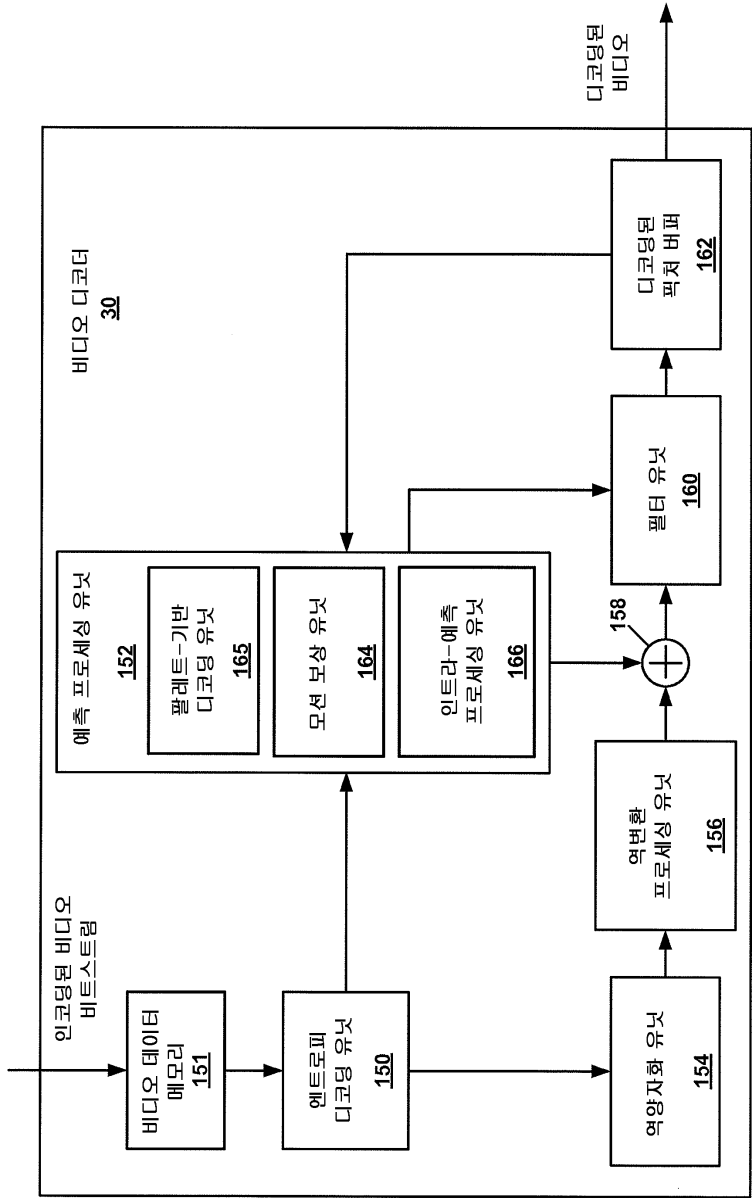
도면1



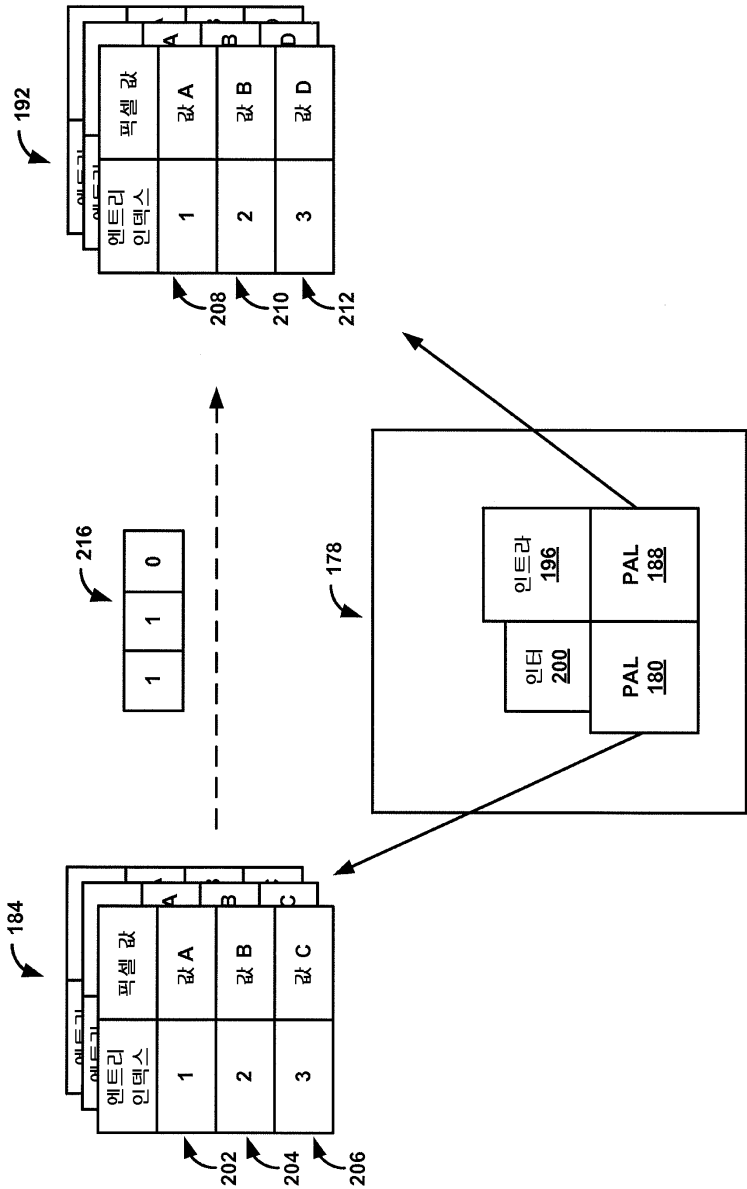
도면2



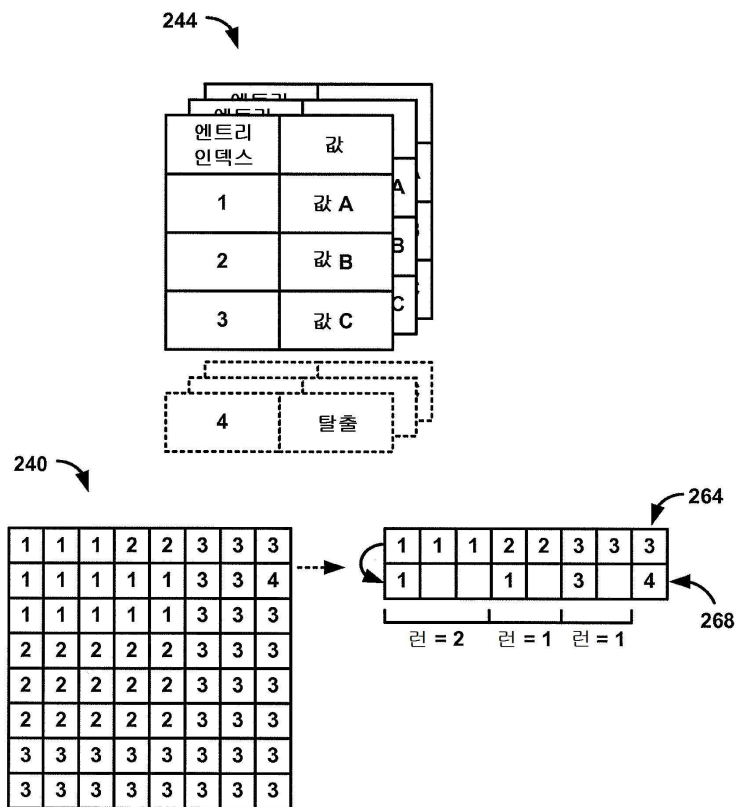
도면3



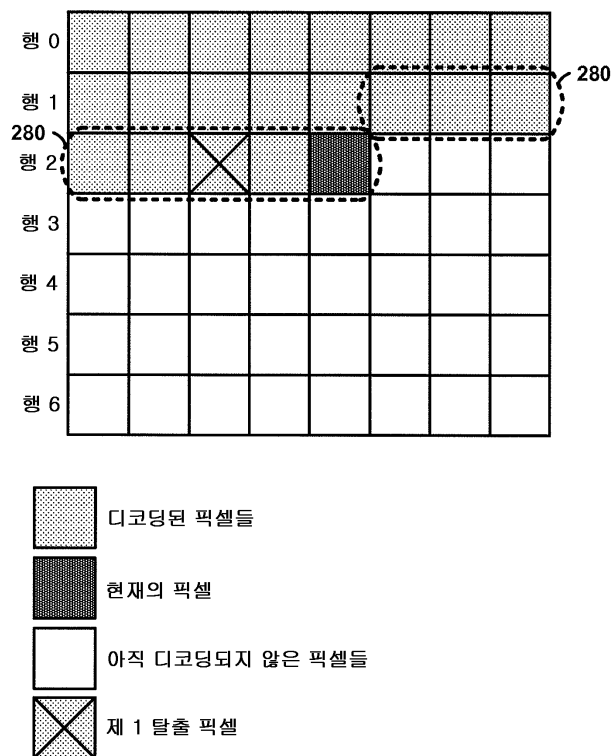
도면4



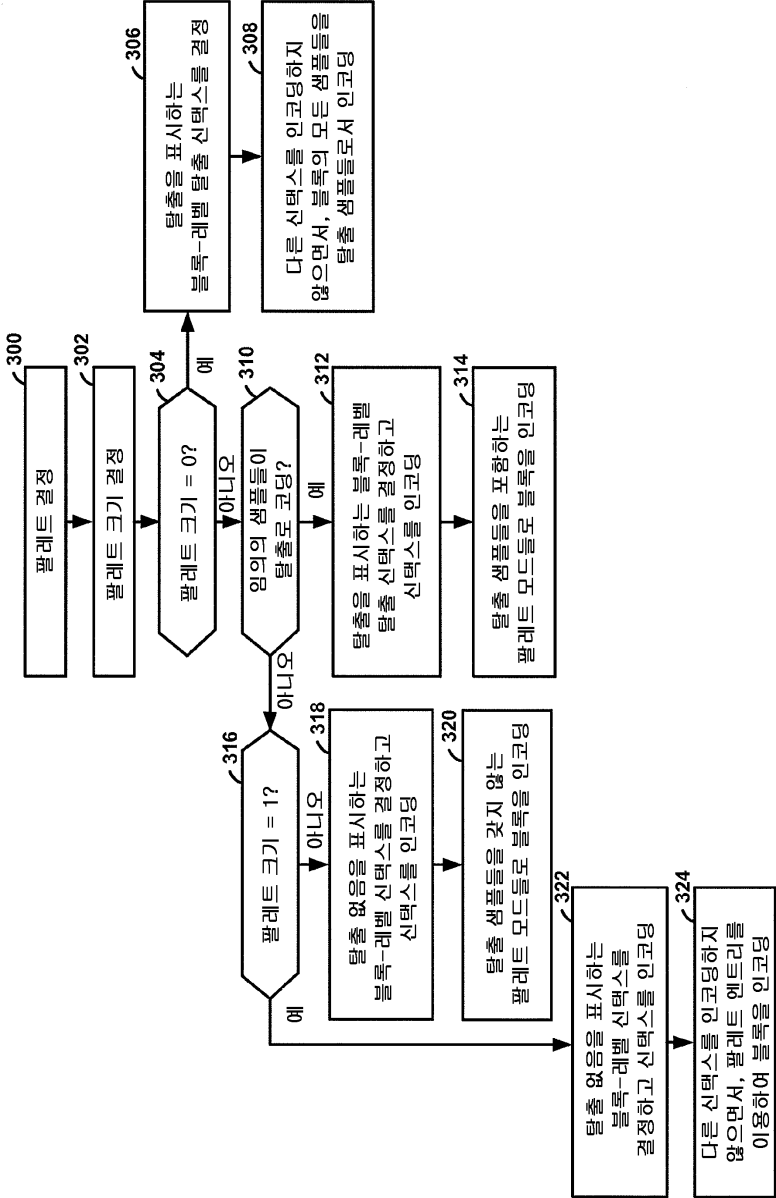
도면5



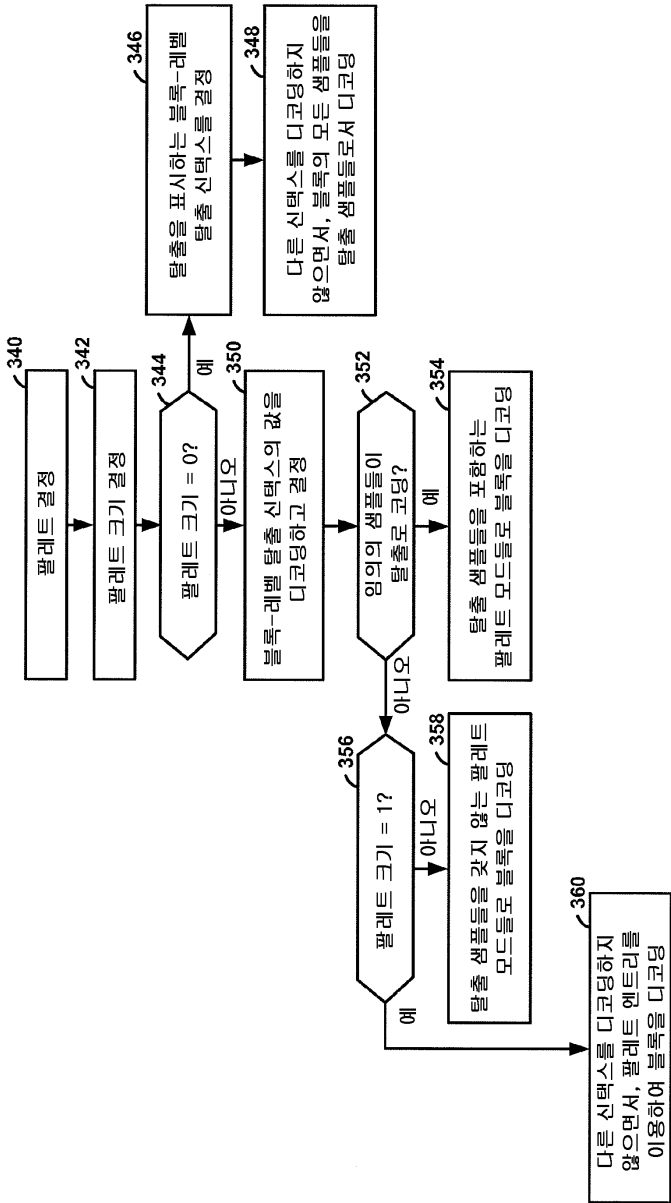
도면6



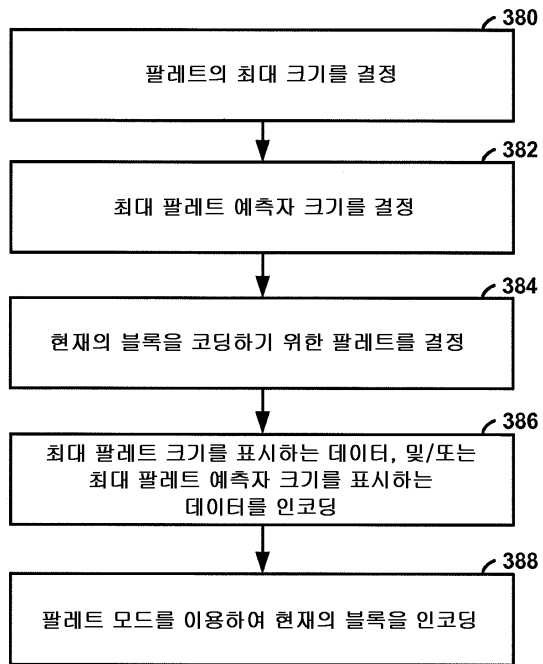
도면7



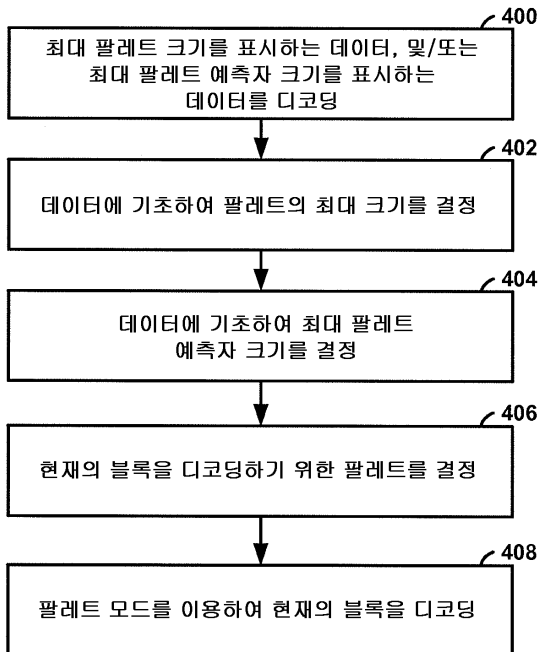
도면8



도면9



도면10



도면11

