



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2025년07월11일

(11) 등록번호 10-2833527

(24) 등록일자 2025년07월09일

(51) 국제특허분류(Int. Cl.)

H04N 19/157 (2014.01) H04N 19/124 (2014.01)

H04N 19/176 (2014.01) H04N 19/182 (2014.01)

H04N 19/186 (2014.01) H04N 19/196 (2014.01)

H04N 19/463 (2014.01) H04N 19/593 (2014.01)

(52) CPC특허분류

H04N 19/157 (2015.01)

H04N 19/124 (2015.01)

(21) 출원번호 10-2018-7010757

(22) 출원일자(국제) 2016년09월16일

심사청구일자 2021년08월27일

(85) 번역문제출일자 2018년04월16일

(65) 공개번호 10-2018-0056687

(43) 공개일자 2018년05월29일

(86) 국제출원번호 PCT/US2016/052186

(87) 국제공개번호 WO 2017/049119

국제공개일자 2017년03월23일

(30) 우선권주장

62/220,675 2015년09월18일 미국(US)

15/266,623 2016년09월15일 미국(US)

(56) 선행기술조사문헌

Xiaoyu Xiu, et al. On escape color coding for palette coding mode, JCT-VC of ITU-T and ISO/IEC. JCTVC-T0118_r1, 2015-02-06, pp. 1-11

k. Shamrman, et al. AHG18: Worst-case Escape Code Length Mitigation, JCT-VC of ITU-T and ISO/IEC. JCTVC-Q0073, 2014-03-27, pp. 1-6

(73) 특허권자

퀄컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

세레진 바딤

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

조쉬 라잔 랙스맨

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(뒷면에 계속)

(74) 대리인

특허법인코리어나

전체 청구항 수 : 총 46 항

심사관 : 전용옥

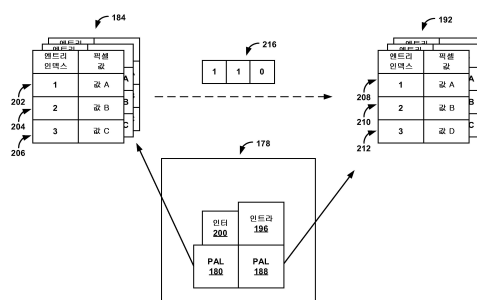
(54) 발명의 명칭 팔레트 모드 비디오 코딩에서의 이스케이프 픽셀 시그널링된 값들의 제한

(57) 요약

비디오 데이터를 저장하도록 구성된 메모리; 및 하나 이상의 프로세서들을 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치. 하나 이상의 프로세서들은 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 것으로서, 비트스트림은 디코딩될 비디오

(뒷면에 계속)

대표도



오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 비트스트림을 수신하고; 그리고 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 비디오 데이터의 블록에서의 샘플을 복원하도록 구성된다.

(52) CPC특허분류

H04N 19/176 (2015.01)

H04N 19/182 (2015.01)

H04N 19/186 (2015.01)

H04N 19/198 (2015.01)

H04N 19/463 (2015.01)

H04N 19/593 (2015.01)

(72) 발명자

라파카 크리스나칸트

미국 92128 캘리포니아주 샌디에고 란초 카멜 드라이브 10508

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

시에 청-테

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 단계로서, 상기 비트스트림은 디코딩될 상기 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 비트스트림을 수신하는 단계;

컴포넌트 비트심도 (n) 를 수신하는 단계로서, 상기 최대 값은 $(1 \ll (n + 1)) - 1$ 에 따라서 계산되는, 상기 컴포넌트 비트심도 (n) 를 수신하는 단계; 및

상기 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 상기 비디오 데이터의 블록에서의 샘플을 복원하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 팔레트 모드에서 사용되는 상기 이스케이프 값들은, 상기 비트스트림으로 시그널링된 이스케이프 값들을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 적어도 하나의 이스케이프 값은, 상기 비디오 데이터의 상기 블록에서의 샘플들을 나타내는데 사용되는 팔레트에서의 임의의 엔트리와 연관되지 않는, 비디오 데이터를 디코딩하는 방법.

청구항 4

삭제

청구항 5

제 1 항에 있어서,

상기 컴포넌트 비트심도는, 칼라 성분 입력 비트심도인, 비디오 데이터를 디코딩하는 방법.

청구항 6

삭제

청구항 7

제 1 항에 있어서,

시퀀스 파라미터 세트에서 시그널링된 상기 컴포넌트 비트심도를 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

제 1 항에 있어서,

상기 컴포넌트 비트심도는, 디코더에 의해 지원되는 최대 비트심도인, 비디오 데이터를 디코딩하는 방법.

청구항 12

제 1 항에 있어서,

상기 최대 값은, 양자화 파라미터 값에 더욱 기초하는, 비디오 데이터를 디코딩하는 방법.

청구항 13

삭제

청구항 14

삭제

청구항 15

제 1 항에 있어서,

상기 최대 값은, 변환 계수들에 또한 부과된 제한들을 상기 이스케이프 값들에 부과함으로써 결정되는, 비디오 데이터를 디코딩하는 방법.

청구항 16

제 1 항에 있어서,

상기 최대 값은, 정밀도 정보에 더욱 기초하는, 비디오 데이터를 디코딩하는 방법.

청구항 17

제 16 항에 있어서,

상기 정밀도 정보는, `extended_precision_processing_flag` 를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 18

제 16 항에 있어서,

`Exp-Golomb` 2진화 프로세스를 호출함으로써, 적어도 상기 정밀도 정보에 기초하여, 상기 이스케이프 값들의 2진화를 유도하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 19

비디오 데이터를 인코딩하는 방법으로서,

적어도 컴포넌트 비트심도에 기초하여, 상기 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들에 대한 최대 값을 결정하는 단계로서, 상기 최대 값은 $(1 \ll (n + 1)) - 1$ 에 따라서 계산되고, 여기서, n 은 컴포넌트 비트심도이며, 상기 이스케이프 값들은 인코딩될 상기 비디오 데이터의 블록에서의 샘플들을 나타내는, 상기 최대 값을 결정하는 단계;

하나 이상의 이스케이프 값들을 상기 최대 값보다 크지 않게 제한하는 단계;

인코딩된 비디오 데이터 비트스트림에서 상기 하나 이상의 이스케이프 값들을 인코딩하는 단계; 및

상기 컴포넌트 비트심도를 시그널링하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 20

제 19 항에 있어서,

상기 팔레트 모드에서 사용되는 상기 이스케이프 값들은, 상기 비트스트림으로 시그널링된 이스케이프 값들을 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 21

제 19 항에 있어서,

상기 하나 이상의 이스케이프 값들은, 상기 비디오 데이터의 상기 블록에서의 샘플들을 나타내는데 사용되는 팔레트에서의 임의의 엔트리와 연관되지 않는, 비디오 데이터를 인코딩하는 방법.

청구항 22

삭제

청구항 23

제 19 항에 있어서,

상기 컴포넌트 비트심도는, 칼라 성분 입력 비트심도인, 비디오 데이터를 인코딩하는 방법.

청구항 24

삭제

청구항 25

제 19 항에 있어서,

상기 컴포넌트 비트심도를 시퀀스 파라미터 세트에서 시그널링하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

제 19 항에 있어서,

상기 컴포넌트 비트심도는, 인코더에 의해 지원되는 최대 비트심도인, 비디오 데이터를 인코딩하는 방법.

청구항 30

제 19 항에 있어서,

상기 최대 값을 결정하는 단계는,

양자화 파라미터 값에 더욱 기초하여, 상기 이스케이프 값들에 대한 최대 값을 결정하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 31

삭제

청구항 32

삭제

청구항 33

제 19 항에 있어서,

상기 최대 값을 결정하는 단계는,

변환 계수들에 또한 부과된 제한들을 상기 이스케이프들 값들에 부과하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 34

제 19 항에 있어서,

상기 최대 값을 결정하는 단계는,

정밀도 정보에 더욱 기초하여, 상기 이스케이프 값들에 대한 상기 최대 값을 결정하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 35

제 34 항에 있어서,

상기 정밀도 정보는, `extended_precision_processing_flag` 를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 36

제 34 항에 있어서,

Exp-Golomb 2진화 프로세스를 호출함으로써, 적어도 상기 정밀도 정보에 기초하여, 상기 이스케이프 값들의 2진화를 유도하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

청구항 45

삭제

청구항 46

삭제

청구항 47

비디오 데이터를 디코딩하도록 구성된 장치로서,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 것으로서, 상기 비트스트림은 디코딩될 상기 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 비트스트림을 수신하고,

컴포넌트 비트심도 (n) 를 수신하는 것으로서, 상기 최대 값은 $(1 \ll (n + 1)) - 1$ 에 따라서 계산되는, 상기 컴포넌트 비트심도 (n) 를 수신하고; 그리고

상기 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 상기 비디오 데이터의 블록에서의 샘플을 복원하도록 구성되는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 48

제 47 항에 있어서,

상기 팔레트 모드에서 사용되는 상기 이스케이프 값들은, 상기 비트스트림으로 시그널링된 이스케이프 값들을 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 49

제 47 항에 있어서,

상기 적어도 하나의 이스케이프 값은, 상기 비디오 데이터의 상기 블록에서의 샘플들을 나타내는데 사용되는 팔레트에서의 임의의 엔트리와 연관되지 않는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 50

삭제

청구항 51

제 47 항에 있어서,

상기 컴포넌트 비트심도는, 칼라 성분 입력 비트심도인, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 52

삭제

청구항 53

제 47 항에 있어서,

상기 하나 이상의 프로세서들은,

시퀀스 파라미터 세트에서 시그널링된 상기 컴포넌트 비트심도를 수신하도록 더 구성되는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 54

삭제

청구항 55

삭제

청구항 56

삭제

청구항 57

제 47 항에 있어서,

상기 컴포넌트 비트심도는, 디코더에 의해 지원되는 최대 비트심도인, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 58

제 47 항에 있어서,

상기 최대 값은, 양자화 파라미터 값에 더욱 기초하는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 59

삭제

청구항 60

삭제

청구항 61

제 47 항에 있어서,

상기 최대 값은, 변환 계수들에 또한 부과된 제한들을 상기 이스케이프 값들에 부과함으로써 결정되는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 62

제 47 항에 있어서,

상기 최대 값은, 정밀도 정보에 더욱 기초하는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 63

제 62 항에 있어서,

상기 정밀도 정보는, `extended_precision_processing_flag` 를 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 64

제 62 항에 있어서,

상기 하나 이상의 프로세서들은,

Exp-Golomb 2진화 프로세스를 호출함으로써, 적어도 상기 정밀도 정보에 기초하여, 상기 이스케이프 값들의 2진

화를 유도하도록 더 구성되는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 65

비디오 데이터를 인코딩하도록 구성된 장치로서,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

적어도 컴포넌트 비트심도에 기초하여, 상기 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들에 대한 최대 값을 결정하는 것으로서, 상기 최대 값은 $(1 \ll (n + 1)) - 1$ 에 따라서 계산되고, 여기서, n 은 컴포넌트 비트심도이며, 상기 이스케이프 값들은 인코딩될 상기 비디오 데이터의 블록에서의 샘플들을 나타내는, 상기 최대 값을 결정하고,

하나 이상의 이스케이프 값들을 상기 최대 값보다 크지 않게 제한하고,

인코딩된 비디오 데이터 비트스트림에서 상기 하나 이상의 이스케이프 값들을 인코딩하고, 그리고

상기 컴포넌트 비트심도를 시그널링하도록 구성되는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 66

제 65 항에 있어서,

상기 팔레트 모드에서 사용되는 상기 이스케이프 값들은, 상기 비트스트림으로 시그널링된 이스케이프 값들을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 67

제 65 항에 있어서,

상기 하나 이상의 이스케이프 값들은 상기 비디오 데이터의 상기 블록에서의 샘플들을 나타내는데 사용되는 팔레트에서의 임의의 엔트리와 연관되지 않는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 68

삭제

청구항 69

제 65 항에 있어서,

상기 컴포넌트 비트심도는, 칼라 성분 입력 비트심도인, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 70

삭제

청구항 71

제 65 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 컴포넌트 비트심도를 시퀀스 파라미터 세트에서 시그널링하도록 더 구성되는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 72

삭제

청구항 73

삭제

청구항 74

삭제

청구항 75

제 65 항에 있어서,

상기 컴포넌트 비트심도는 인코더에 의해 지원되는 최대 비트심도인, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 76

제 65 항에 있어서,

상기 최대 값을 결정하는 것은,

양자화 파라미터 값에 더욱 기초하여, 상기 이스케이프 값들에 대한 최대 값을 결정하는 것을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 77

삭제

청구항 78

삭제

청구항 79

제 65 항에 있어서,

상기 최대 값을 결정하는 것은,

변환 계수들에 또한 부과된 제한들을 상기 이스케이프들 값들에 부과하는 것을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 80

제 65 항에 있어서,

상기 최대 값을 결정하는 것은,

정밀도 정보에 더욱 기초하여, 상기 이스케이프 값들에 대한 상기 최대 값을 결정하는 것을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 81

제 80 항에 있어서,

상기 정밀도 정보는, `extended_precision_processing_flag` 를 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 82

제 80 항에 있어서,

상기 하나 이상의 프로세서들은,

Exp-Golomb 2진화 프로세스를 호출함으로써, 적어도 상기 정밀도 정보에 기초하여, 상기 이스케이프 값들의 2진화를 유도하도록 더 구성되는, 비디오 데이터를 인코딩하도록 구성된 장치.

청구항 83

삭제

청구항 84

삭제

청구항 85

삭제

청구항 86

삭제

청구항 87

삭제

청구항 88

삭제

청구항 89

삭제

청구항 90

삭제

청구항 91

삭제

청구항 92

삭제

청구항 93

비디오 데이터를 디코딩하도록 구성된 장치로서,

상기 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 수단으로서, 상기 비트스트림은 디코딩될 상기 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 비트스트림을 수신하는 수단;

컴포넌트 비트심도 (n) 를 수신하는 수단으로서, 상기 최대 값은 $(1 \ll (n + 1)) - 1$ 에 따라서 계산되는, 상기 컴포넌트 비트심도 (n) 를 수신하는 수단; 및

상기 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 상기 비디오 데이터의 블록에서의 샘플을 복원하는 수단을 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

청구항 94

삭제

청구항 95

명령들을 저장하고 있는 비밀시성 컴퓨터-판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금,

비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하게 하는 것으로서, 상기 비트스트림은 디코딩될 상기 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 비트스트림을 수신하게 하고;

컴포넌트 비트심도 (n) 를 수신하게 하는 것으로서, 상기 최대 값은 $(1 \ll (n + 1)) - 1$ 에 따라서 계산되는, 상기 컴포넌트 비트심도 (n) 를 수신하게 하고;

상기 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 상기 비디오 데이터의 블록에서의 샘플을 복원하게 하는, 비일시성 컴퓨터-판독가능 저장 매체.

청구항 96

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2015년 9월 18일에 출원된 미국 가특허출원 번호 제 62/220,675호의 이익을 주장하며, 이는 본원에 전체적으로 참고로 포함된다.

기술 분야

[0003] 본 개시물은 콘텐츠를 인코딩하고 디코딩하는 것, 좀더 구체적으로는, 팔레트-기반의 코딩 모드에 따라서 콘텐츠를 인코딩하고 디코딩하는 것에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은, 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDAs), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트폰들", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), ITU-T H.265, HEVC (High Efficiency Video Coding), 및 이런 표준들의 확장판들에 의해 정의된 표준들에서 설명되는 것들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이런 비디오 압축 기법들을 구현함으로써, 디지털 비디오 정보를 좀더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거하기 위해 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 수행한다. 블록-기반 비디오 코딩에 있어, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 일부) 는 비디오 블록들로 파티셔닝될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 초래한다. 잔차 데이터는 코딩될 원래 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라서 인코딩되며, 잔차 데이터는 코딩된 블록과 예측 블록 사이의 차이를 나타낸다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라서 인코딩된다. 추가적인 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 그 결과 잔차 계수들이 되고, 그후 양자화될 수도 있다.

[0006] 이미지 또는 비디오와 같은, 콘텐츠는 팔레트 모드를 이용하여 인코딩되고 디코딩될 수도 있다. 일반적으로, 팔레트 모드는 픽처에서의 픽셀들을 코딩하는 또 다른 기법이다. 팔레트 모드는 픽셀들의 콘텐츠를 표현하기 위해 칼라 값들의 팔레트의 사용을 수반한다. 콘텐츠는 콘텐츠 픽셀들이 팔레트에서의 칼라 값들에 대응하는 값들로 표현되도록 인코딩될 수도 있다. 인덱스 맵은 픽셀들을 팔레트에서의 칼라 값들에

맵핑하도록 디코딩될 수도 있으며, 이에 따라서, 칼라 값들을 획득하여 콘텐츠를 복원할 수도 있다.

발명의 내용

과제의 해결 수단

- [0007] 본 개시물의 기법들은 팔레트-기반 비디오 코딩에 관한 것이다. 예를 들어, 팔레트-기반의 코딩에서, 비디오 코더 (비디오 인코더 또는 비디오 디코더) 는 특정의 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 나타내는 칼라들의 테이블로서 "팔레트" 를 형성할 수도 있다. 팔레트-기반의 코딩은 상대적으로 적은 수의 칼라들을 가지는 비디오 데이터의 코딩 영역들에 특히 유용할 수도 있다. 실제 픽셀 값들 (또는, 그들의 잔차들) 을 코딩하는 대신, 비디오 코더는 픽셀들을 픽셀들의 칼라들을 나타내는 팔레트에서의 엔트리들과 관련시키는, 픽셀들 중 하나 이상에 대한 팔레트 인덱스들을 코딩할 수도 있다. 본 개시물에서 설명하는 기법들은 팔레트-기반의 비디오 코딩 모드들을 시그널링하는 것, 팔레트들을 송신하는 것, 팔레트들을 유도하는 것, 및 팔레트-기반의 코딩 맵들 및 다른 신덱스 엘리먼트들을 송신하는 것 중 하나 이상의 다양한 조합들에 대한 기법들을 포함할 수도 있다.
- [0008] 일 예에서, 비디오 데이터를 디코딩하는 방법은 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 단계로서, 비트스트림은 디코딩될 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 수신하는 단계; 및 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 비디오 데이터의 블록에서의 샘플을 복원하는 단계를 포함한다.
- [0009] 다른 예에서, 비디오 데이터를 인코딩하는 방법은 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들에 대한 최대 값을 결정하는 단계로서, 이스케이프 값들은 인코딩될 비디오 데이터의 블록에서의 샘플들을 나타내는, 상기 결정하는 단계; 하나 이상의 이스케이프 값들을 최대 값보다 크지 않게 제한하는 단계; 및 인코딩된 비디오 데이터 비트스트림에서 하나 이상의 이스케이프 값들을 인코딩하는 단계를 포함한다.
- [0010] 다른 예에서, 비디오 데이터를 디코딩하는 방법은 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는 비트스트림을 수신하는 단계로서, 비트스트림은 적어도 하나의 이스케이프 값이 양자화 파라미터 하한 이상인 양자화 파라미터에 따라서 양자화되게 보장하는 제한을 준수하도록 제약되는, 상기 수신하는 단계; 제한에 기초하여 비트스트림으로부터 적어도 하나의 이스케이프 값을 결정하는 단계; 및 적어도 하나의 이스케이프 값을 이용하여 비디오 데이터의 샘플을 복원하는 단계를 포함한다.
- [0011] 다른 예에서, 비디오 데이터를 인코딩하는 방법은 비디오 데이터에 대한 양자화 파라미터 값을 결정하는 단계; 양자화 파라미터 하한을 결정하는 단계; 양자화 파라미터 값이 양자화 파라미터 하한 미만이면, 양자화 파라미터 하한에 따라서, 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들을 양자화하는 단계로서, 이스케이프 값들은 인코딩될 비디오 데이터의 블록에서의 샘플들을 나타내는, 상기 양자화하는 단계; 그리고 인코딩된 비디오 데이터 비트스트림에서 이스케이프 값들을 인코딩하는 단계를 포함한다.
- [0012] 다른 예에서, 비디오 데이터를 디코딩하도록 구성된 장치로서, 상기 장치는 비디오 데이터를 저장하도록 구성된 메모리; 및 하나 이상의 프로세서들을 포함하며, 상기 하나 이상의 프로세서들은, 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 것으로서, 비트스트림은 디코딩될 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 수신하고; 그리고 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 비디오 데이터의 블록에서의 샘플을 복원하도록 구성된다.
- [0013] 다른 예에서, 비디오 데이터를 인코딩하도록 구성된 장치로서, 상기 장치는 비디오 데이터를 저장하도록 구성된 메모리; 및 하나 이상의 프로세서들을 포함하며, 상기 하나 이상의 프로세서들은, 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들에 대한 최대 값을 결정하는 것으로서, 이스케이프 값들은 인코딩될 비디오 데이터의 블록에서의 샘플들을 나타내는, 상기 결정하고; 하나 이상의 이스케이프 값들을 최대 값보다 크지 않게 제한하고; 그리고 인코딩된 비디오 데이터 비트스트림에서 하나 이상의 이스케이프 값들을 인코딩하도록 구성된다.
- [0014] 다른 예에서, 비디오 데이터를 디코딩하도록 구성된 장치로서, 상기 장치는 비디오 데이터를 저장하도록 구성된 메모리; 및 하나 이상의 프로세서들을 포함하며, 상기 하나 이상의 프로세서들은, 비디오 데이터를 디코딩하기

위한 팔레트 모드에서 사용되는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는 비트스트림을 수신하는 것으로서, 비트스트림은 적어도 하나의 이스케이프 값이 양자화 파라미터 하한 이상인 양자화 파라미터에 따라서 양자화되게 보장하는 제한을 준수하도록 제약되는, 상기 수신하고; 제한에 기초하여 비트스트림으로부터 적어도 하나의 이스케이프 값을 결정하고; 그리고 적어도 하나의 이스케이프 값을 이용하여 비디오 데이터의 샘플을 복원하도록 구성된다.

[0015] 다른 예에서, 비디오 데이터를 인코딩하도록 구성된 장치로서, 상기 장치는 비디오 데이터를 저장하도록 구성된 메모리; 및 하나 이상의 프로세서들을 포함하며, 상기 하나 이상의 프로세서들은, 비디오 데이터에 대한 양자화 파라미터 값을 결정하고; 양자화 파라미터 하한을 결정하고; 양자화 파라미터 값이 양자화 파라미터 하한 미만이면 양자화 파라미터 하한에 따라서, 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들을 양자화하는 것으로서, 이스케이프 값들은 인코딩될 비디오 데이터의 블록에서의 샘플들을 나타내는, 상기 양자화하고; 그리고 인코딩된 비디오 데이터 비트스트림에서 이스케이프 값들을 인코딩하도록 구성된다.

[0016] 다른 예에서, 비디오 데이터를 디코딩하도록 구성된 장치로서, 상기 장치는 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 수단으로서, 비트스트림은 디코딩될 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 수신하는 수단; 및 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 비디오 데이터의 블록에서의 샘플을 복원하는 수단을 포함한다.

[0017] 다른 예에서, 비디오 데이터를 디코딩하도록 구성된 장치로서, 상기 장치는 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는 비트스트림을 수신하는 수단으로서, 비트스트림은 적어도 하나의 이스케이프 값이 양자화 파라미터 하한 이상인 양자화 파라미터에 따라서 양자화되게 보장하는 제한을 준수하도록 제약되는, 상기 수신하는 수단; 제한에 기초하여 비트스트림으로부터 적어도 하나의 이스케이프 값을 결정하는 수단; 및 적어도 하나의 이스케이프 값을 이용하여 비디오 데이터의 샘플을 복원하는 수단을 포함한다.

[0018] 다른 예에서, 명령들을 저장하고 있는 비일시성 컴퓨터-판독가능 매체로서, 상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신하는 것으로서, 비트스트림은 디코딩될 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는, 상기 수신하고; 그리고 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 비디오 데이터의 블록에서의 샘플을 복원하게 한다.

[0019] 다른 예에서, 명령들을 저장하고 있는 비일시성 컴퓨터-판독가능 매체로서, 상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는 비트스트림을 수신하는 것으로서, 비트스트림은 적어도 하나의 이스케이프 값이 양자화 파라미터 하한 이상인 양자화 파라미터에 따라서 양자화되게 보장하는 제한을 준수하도록 제약되는, 상기 수신하고; 제한에 기초하여 비트스트림으로부터 적어도 하나의 이스케이프 값을 결정하고; 그리고 적어도 하나의 이스케이프 값을 이용하여 비디오 데이터의 샘플을 복원하게 한다.

[0020] 본 개시물의 하나 이상의 예들의 세부 사항들은 첨부도면 및 아래의 설명에서 개시된다. 다른 특징들, 목적들, 및 이점들은 상세한 설명, 도면들, 및 청구범위로부터 명백할 것이다.

도면의 간단한 설명

[0021] 도 1은 본 개시물에서 설명하는 기법들을 활용할 수도 있는 예시적인 비디오 코딩 시스템을 예시하는 블록도이다.

도 2는 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 3은 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 4는 본 개시물의 기법들에 따른, 팔레트-기반의 비디오 코딩을 위한 팔레트 엔트리들을 결정하는 일 예를 예시하는 개념도이다.

도 5는 본 개시물의 기법들에 따른, 픽셀들의 블록에 대한 팔레트로의 인덱스들을 결정하는 일 예를 예시하는 개념도이다.

도 6 은 본 개시물의 하나 이상의 양태들에 따른, 이스케이프 픽셀 값들이 제한되는 예시적인 인코딩 프로세스를 예시하는 플로우 차트이다.

도 7 은 본 개시물의 하나 이상의 양태들에 따른, 이스케이프 픽셀 값들이 제한되는 예시적인 디코딩 프로세스를 예시하는 플로우 차트이다.

도 8 은 본 개시물의 하나 이상의 양태들에 따른, 하한이 양자화 파라미터에 적용되는 예시적인 인코딩 프로세스를 예시하는 플로우 차트이다.

도 9 는 본 개시물의 하나 이상의 양태들에 따른, 하한이 양자화 파라미터에 적용되는 예시적인 디코딩 프로세스를 예시하는 플로우 차트이다.

발명을 실시하기 위한 구체적인 내용

- [0022] 본 개시물은 비디오 코딩 및 압축을 위한 기법들을 설명한다. 특히, 본 개시물은 비디오 데이터의 팔레트-기반 코딩을 위한 기법들을 설명한다. 일부 예들에서, 기법들의 양태들은 또한 개개의 픽처들과 같은 다른 콘텐츠의 코딩에 이용될 수도 있다. 전통적인 비디오 코딩에서, 이미지들은 연속적인-톤이고 공간적으로 매끄러운 자연 이미지들로 가정된다. 이들 가정들에 기초하여, 블록-기반 변환, 필터링, 등과 같은 다양한 툴들이 개발되었으며, 이러한 툴들은 자연 콘텐츠 비디오들에 대해 우수한 성능을 보였다.
- [0023] 그러나, 원격 데스크탑, 협력 작업, 및 무선 디스플레이와 같은, 애플리케이션들에서는, 컴퓨터 발생된 스크린 콘텐츠가 압축될 지배적인 콘텐츠일 수도 있다. 이 유형의 콘텐츠는 이산-톤을 가지는 경향이 있으며, 날카로운 라인들 및 높은 콘트라스트 오브젝트 경계들을 특징으로 한다. 연속적인-톤 및 평활도 (smoothness) 의 가정이 더 이상 적용되지 않을 수도 있으며, 따라서, 전통적인 비디오 코딩 기법들은 콘텐츠를 압축하는데는 비효율적인 방법들일 수도 있다.
- [0024] 일부 예들에서, 팔레트-기반의 코딩 기법들은 하나 이상의 비디오 코딩 표준들과 함께 사용하도록 구성될 수도 있다. 예를 들어, ITU-T H.265, 고-효율 비디오 코딩 (HEVC) 은 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 과 ISO/IEC 동화상 전문가 그룹 (MPEG) 의 비디오 코딩에 관한 합동 작업팀 (JCT-VC) 에 의해 개발되고 있는 새로운 비디오 코딩 표준이다. HEVC 표준은 2015년 4월, 국제 전기통신 연합, ITU-T H.265, Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services--Coding of moving video, High Efficiency Video Coding (이하, "HEVC") 로서 공개되어 있다.
- [0025] HEVC 프레임워크에 있어서, 일 예로서, 팔레트-기반의 코딩 기법들은 코딩 유닛 (CU) 모드로서 사용되도록 구성될 수도 있다. 다른 예들에서, 팔레트-기반의 코딩 기법들은 HEVC 의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 따라서, CU 코딩의 상황에서 설명된 아래에 개시되는 프로세스들 모두가 추가적으로 또는 대안적으로, PU 코딩에 적용될 수도 있다. 그러나, 이들 HEVC-기반의 예들은, 이러한 기법들이 독립적으로 또는 다른 기준 또는 앞으로 개발될 시스템들/표준들의 일부분으로서 동작하도록 적용될 수도 있기 때문에, 본원에서 설명하는 팔레트-기반의 코딩 기법들의 제한 또는 한정으로서, 간주되지 않아야 한다. 이들 경우들에서, 팔레트 코딩을 위한 유닛은 정사각형 블록들, 직사각형의 블록들 또는 심지어 비-직사각형 형태의 영역들일 수 있다.
- [0026] 팔레트-기반의 코딩에서, 비디오 데이터의 특성의 영역은 상대적으로 적은 개수의 칼라들을 갖는 것으로 가정될 수도 있다. 비디오 코더 (비디오 인코더 및 비디오 디코더) 는 특성의 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 표시하기 위해 칼라들의 테이블로서 소위 "팔레트" 를 코딩할 수도 있다. 블록에서의 각각의 픽셀은 픽셀의 칼라 값을 나타내는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 코더는 블록에서의 픽셀의 칼라 값을 팔레트에서의 적합한 칼라 값에 관련시키는 픽셀에 대한 인덱스를 코딩할 수도 있다.
- [0027] 상기 예에서, 비디오 인코더는 블록에 대한 팔레트를 결정하고, 각각의 픽셀의 값을 표현하기 위해 팔레트에 엔트리를 로케이트하고, 그리고 픽셀 값을 팔레트에 관련시키는 픽셀들에 대한 팔레트 인덱스들 (또한, 팔레트 인덱스 값들로서 지칭됨) 로 팔레트를 인코딩함으로써, 비디오 데이터의 블록을 인코딩할 수도 있다. 비디오 디코더는 인코딩된 비트스트림으로부터, 블록에 대한 팔레트 뿐만 아니라, 블록의 픽셀들에 대한 팔레트 인덱스들을 획득할 수도 있다. 비디오 디코더는 픽셀들의 팔레트 인덱스들을 팔레트의 엔트리들에 관련시켜, 블록의 픽셀 값들을 복원할 수도 있다. 픽셀들 (및/또는 픽셀 값을 포함하는 관련된 팔레트 인덱스들) 은 일반적으로 샘플들로서 지칭될 수도 있다.

- [0028] 블록에서의 샘플들은 어떤 스캐닝 순서를 이용하여 프로세싱 (예컨대, 스캔) 된다고 가정된다. 예를 들어, 비디오 인코더는 스캐닝 순서를 이용하여 팔레트 인덱스들을 스캐닝함으로써 팔레트 인덱스들의 2차원 블록을 1차원 어레이로 변환할 수도 있다. 이와 유사하게, 비디오 디코더는 스캐닝 순서를 이용하여 팔레트 인덱스들의 블록을 복원할 수도 있다. 따라서, 본 개시물은 이전 샘플을, 스캐닝 순서에서 현재 코딩중인 샘플에 선행하는 샘플로서 지칭한다. 예를 들어, 스캐닝 순서는 수평 스캔 순서, 수직 스캐닝 순서, 가로/세로 (뱀모양) 스캐닝 순서, 순방향 또는 역방향 스캐닝 순서 등일 수도 있다. 언급된 것들 이외의 스캐닝들이 또한 적용가능할 수도 있는 것으로 이해되어야 한다. 상기 예는 팔레트-기반의 코딩의 일반적인 설명을 제공하기 위한 것이다.
- [0029] 팔레트는 일반적으로 인덱스로 넘버링되며 칼라 성분 (예를 들어, RGB, YUV, 또는 기타 등등) 값들 또는 강도들을 나타내는 엔트리들을 포함한다. 비디오 인코더 및 비디오 디코더 양자는 팔레트 엔트리들의 개수, 각각의 팔레트 엔트리에 대한 칼라 성분 값들 및 현재의 블록에 대한 팔레트 엔트리들의 정확한 순서정렬을 결정한다. 본 개시물에서는, 각각의 팔레트 엔트리가 샘플의 모든 칼라 성분들에 대한 값들을 규정한다고 가정된다. 그러나, 본 개시물의 컨셉들은 각각의 칼라 성분에 대해 별개의 팔레트를 사용하는 것에 적용가능하다.
- [0030] 일부 예들에서, 팔레트는 이전에 코딩된 블록들로부터의 정보를 이용하여 구성될 수도 있다. 즉, 팔레트는 이전 블록(들)을 코딩하는데 사용되는 팔레트(들)로부터 예측된 예측된 팔레트 엔트리들을 포함할 수도 있다. 예를 들어, 2014년 3월 27일 - 4월 4일, 스페인, 발렌시아, Wei Pu 등, "AHG10: Suggested Software for Palette Coding based on RExt6.0" 표준 제안 문서, JCTVC-Q0094 (이하, JCTVC-Q0094) 에서 설명되어 있는 바와 같이, 팔레트는 예측자 팔레트로부터 복사되는 엔트리들을 포함할 수도 있다. 예측자 팔레트는 팔레트 모드를 이용하여 이전에 코딩된 블록들 또는 다른 복원된 샘플들로부터의 팔레트 엔트리들을 포함할 수도 있다. 예측자 팔레트에서의 각각의 엔트리에 대해, 그 플래그와 연관된 엔트리가 현재의 팔레트에 복사되는지 여부를 표시하기 위해 2진 플래그가 코딩될 수도 있다 (예컨대, 플래그 = 1 로 표시됨). 2진 플래그들의 스트림은 2진 팔레트 예측 벡터로서 지칭될 수도 있다. 현재의 블록을 코딩하기 위한 팔레트는 또한 (예컨대, 팔레트 예측 벡터와는 별개로) 명시적으로 코딩될 수도 있는 새로운 팔레트 엔트리들의 수를 포함할 수도 있다. 새로운 엔트리들의 개수의 표시가 또한 코딩될 수도 있다. 예측된 엔트리들과 새로운 엔트리들의 총합은 블록에 대한 전체 팔레트 사이즈를 표시할 수도 있다.
- [0031] 일부 예들에서, 팔레트-기반의 코딩 모드로 코딩된 블록에서의 각각의 샘플은 아래에 개시된 바와 같은 3개의 팔레트 모드들 중 하나를 이용하여 코딩될 수도 있다:
- [0032] 이스케이프 모드: 이 모드에서, 샘플 값은 팔레트에 팔레트 엔트리로서 포함되지 않을 수도 있으며, 양자화된 샘플 값은 인코더에 의해 모든 칼라 성분들에 대해 명시적으로 비트스트림으로 시그널링된다. 이스케이프 모드는 새로운 팔레트 엔트리들의 시그널링과 유사하지만, 새로운 팔레트 엔트리들에 대해서는, 칼라 성분 값들이 양자화되지 않는다.
- [0033] 상측 복사 (Copy Top) 모드 (또한, 상부복사 모드로 지칭됨): 이 모드에서는, 현재의 샘플에 대한 팔레트 엔트리 인덱스가 블록에서 직상부에 로케이트된 샘플로부터 복사된다. 단지 복사의 길이 (런 길이) 를 규정하는 신택스 엘리먼트만이 상측 복사 모드에 대해 인코더에 의해 비트스트림으로 시그널링된다.
- [0034] 인덱스 모드 (또한, 값 모드 또는 좌측 모드로서 지칭됨): 이 모드에서, 팔레트 엔트리 인덱스의 값은 인코더에 의해, 예를 들어, 트렁케이트된 2진 코드, 그리고 뒤이어서, 스캐닝 순서에서 얼마나 많은 다음 픽셀들이 시그널링된 것 (런 길이) 과 동일한 인덱스를 공유하는지를 규정하는 다른 신택스 엘리먼트를 이용하여, 비트스트림으로 명시적으로 시그널링된다.
- [0035] 앞에서 설명한 바와 같이, 팔레트 엔트리 인덱스는 팔레트 인덱스 또는 간단히 인덱스로서 지칭될 수도 있다. 이들 용어들은 본 개시물의 기법들을 기술하기 위해 상호교환가능하게 사용될 수 있다. 게다가, 아래에서 더 자세히 설명하는 바와 같이, 팔레트 인덱스는 하나 이상의 연관된 칼라 또는 강도 값들을 가질 수도 있다. 예를 들어, 팔레트 인덱스는 픽셀의 단일 칼라 또는 강도 성분과 연관된 단일 연관된 칼라 또는 강도 값 (예컨대, RGB 데이터의 적색 성분, YUV 데이터의 Y 성분, 또는 기타 등등) 을 가질 수도 있다. 다른 예에서, 팔레트 인덱스는 다수의 연관된 칼라 또는 강도 값들을 가질 수도 있다. 일부의 경우, 팔레트-기반의 코딩이 단색 비디오를 코딩하기 위해 적용될 수도 있다. 따라서, "칼라 값" 은 일반적으로 (단색의 경우에) 픽셀 값을 발생시키는데 사용되는 임의의 칼라 또는 비-칼라 성분을 지칭할 수도 있다.
- [0036] 상측 복사 및 인덱스 모드들에 있어서, (간단히 런으로서 또한 지칭될 수도 있는) 런 값이 또한 시그널링될 수

도 있다. 런 값은 함께 코딩되는 팔레트-코딩된 블록에서 특정의 스캔 순서에서 연속된 샘플들의 수 (예컨대, 샘플들의 런) 를 표시할 수도 있다. 일부의 경우, 샘플들의 런은 또한 런의 각각의 샘플이 팔레트에 대한 연관된 인덱스를 가지기 때문에, 팔레트 인덱스들의 런으로서 지칭될 수도 있다.

[0037] 런 값은 동일한 팔레트-코딩 모드를 이용하여 코딩되는 팔레트 인덱스들의 런을 표시할 수도 있다. 예를 들어, 인덱스 모드에 대해, 비디오 코더 (비디오 인코더 또는 비디오 디코더) 는, 동일한 인덱스 값을 가지며 팔레트 인덱스로 코딩중인, 스캔 순서에서 연속된 샘플들의 수를 표시하는 런 값 및 팔레트 인덱스 (또한, 팔레트 인덱스 값 또는 간단히 인덱스 값으로 지칭됨) 를 코딩할 수도 있다. 상부복사 모드에 대해, 비디오 코더는, 현재의 샘플 값에 대한 인덱스가 상부-이웃하는 샘플 (예컨대, 블록에서 현재 코딩중인 샘플 위에 위치되는 샘플) 의 인덱스, 및 상부-이웃하는 샘플들로부터 팔레트 인덱스를 또한 복사하는, 스캔 순서에서 연속된 샘플들의 수를 표시하는 런 값에 기초하여 복사된다는 표시를 코딩할 수도 있다. 따라서, 상기 예들에서, 팔레트 인덱스들의 런은 동일한 값을 가지는 팔레트 인덱스들의 런, 또는 상부-이웃하는 팔레트 인덱스들로부터 복사되는 팔레트 인덱스들의 런을 지칭한다.

[0038] 따라서, 런은 주어진 모드에 대해, 동일한 모드에 속하는 후속 샘플들의 개수를 규정할 수도 있다. 일부의 경우, 인덱스 및 런 값을 시그널링하는 것은 런-길이 코딩과 유사할 수도 있다. 일 예에서, 예시의 목적을 위해, 블록의 연속된 팔레트 인덱스들의 스트링은 (예컨대, 각각의 인덱스가 블록에서의 샘플에 대응하는 경우) 0, 2, 2, 2, 2, 5 일 수도 있다. 이 예에서, 비디오 코더는 인덱스 모드를 이용하여 제 2 샘플 (예컨대, "2" 의 제 1 팔레트 인덱스) 을 코딩할 수도 있다. 2 와 동일한 인덱스를 코딩한 후, 비디오 코더는 3개의 후속 샘플들이 또한 2 의 동일한 팔레트 인덱스를 가진다는 것을 표시하는 3 의 런을 코딩할 수도 있다. 유사한 방법으로, 상부 복사 모드를 이용하여 인덱스를 코딩한 후 4개의 팔레트 인덱스들의 런을 코딩하는 것은 총 5개의 팔레트 인덱스들이 현재 코딩중인 샘플 위치의 상부에 있는 로우에서의 대응하는 팔레트 인덱스들로부터 복사된다는 것을 표시할 수도 있다.

[0039] 이스케이프 샘플들 (또한, 이스케이프 픽셀들로서 지칭됨) 은 블록을 코딩하기 위한 팔레트에 표현된 대응하는 칼라를 가지지 않는 블록의 샘플들 (또는, 픽셀들) 일 수도 있다. 따라서, 이스케이프 샘플들은 팔레트로부터의 칼라 엔트리 (또는, 픽셀 값) 를 이용하여 복원되지 않을 수도 있다. 대신, 이스케이프 샘플들에 대한 칼라 값들은 팔레트의 칼라 값들과는 별개로 비트스트림으로 시그널링된다.

[0040] 비디오 코더 (예컨대, 비디오 인코더 및 비디오 디코더) 는 팔레트-코딩된 블록의 샘플이 블록에 대한 팔레트에 포함되지 않은 샘플의 칼라에 기초하여, 예컨대, 위에서 "이스케이프 모드" 로서 지칭되는 프로세스를 이용하여 코딩되는지 여부를 표시하는 샘플 당 데이터를 코딩할 수도 있다. 일부 예들에서, 비디오 코더는 임의의 팔레트 엔트리에 대응하지 않는 팔레트에 대한 특수 인덱스를 수용하기 위해, 팔레트의 팔레트 엔트리들의 개수를 1 만큼 증가시키도록 구성될 수도 있다. 비디오 코더는 추가적인 인덱스를 주어진 블록에 대한 증가된 팔레트에서의 최종 팔레트 인덱스로서 포함시킬 수도 있다. 추가적인 인덱스는 이스케이프 모드의 표시로서 사용될 수도 있다.

[0041] 위에서 설명된 예에서, 비디오 코더는 블록의 특정의 샘플 값에 대해, 추가적인 샘플이 이스케이프 샘플 (예컨대, 블록을 코딩하기 위한 팔레트에 표현된 칼라 값을 가지지 않는 샘플) 로서 코딩된다는 것을 표시하기 위해 추가적인 인덱스를 나타내는 데이터를 코딩할 수도 있다. 비디오 코더는 또한 이스케이프 샘플의 칼라 값 (들) 을 코딩할 수도 있다. 이 예에서는, 명시적인 신택스를 이용하여 시그널링될 단지 2개의 가능한 모드들 (예컨대, 상측 복사 모드 또는 인덱스 모드 (또한, 인덱스 모드 또는 값 모드로서 지칭됨)) 만이 있다. 예를 들어, 샘플이 인덱스 모드에서 코딩되고 인덱스 모드에 대한 인덱스가 이스케이프 인덱스 (예컨대, 위에서 언급한 팔레트에 대한 추가적인 인덱스) 와 동일하면, 비디오 코더는 코딩될 샘플을 이스케이프 샘플로서 추론할 수도 있다. 일부의 경우, 어떤 런도 이스케이프 샘플들로 시그널링되지 않는다.

[0042] 팔레트-코딩된 블록에 있어서, 하나 이상의 신택스 엘리먼트들은 블록-레벨 (예컨대, CU 레벨 또는 LCU 레벨) 에서, 블록의 임의의 샘플이 팔레트에 포함되지 않은 샘플의 칼라 값에 기초하여 코딩되는지, 예컨대, 이스케이프 샘플로서 코딩되는지 여부를 표시할 수도 있다. 하나 이상의 신택스 엘리먼트들은 블록-레벨 이스케이프 신택스로서 지칭될 수도 있다. 예를 들어, 블록-레벨 신택스는 CU 또는 LCU 와 같은 비디오 데이터의 블록을 코딩하는 동안 코딩되거나 또는 결정되는 신택스를 지칭할 수도 있다. 블록-레벨 신택스는 헤더에, 또는 블록과 연관된 다른 데이터 (예컨대, 블록의 특성을 기술하는 블록 이전 또는 이후에 코딩되는 데이터) 와 함께, 포함될 수도 있다. 이에 반해, 블록-레벨 신택스가 아닌 다른 신택스는 슬라이스 헤더에, 또는 비디오 데이터의 개개의 픽셀들과 함께, 포함될 수도 있다.

- [0043] 일 예에서, 비디오 코더는 블록의 임의의 샘플이 이스케이프 모드에서 코딩되는지 여부를 표시하는 (블록-레벨 이스케이프 플래그로서 지칭될 수도 있는) 플래그를 코딩하거나 및/또는 결정하도록 구성될 수도 있다. 예를 들어, 0의 플래그 값은 어떤 블록의 샘플들도 이스케이프 모드를 이용하여 코딩되지 않는다는 것을 표시할 수도 있다. 즉, 블록의 모든 샘플들의 값이 블록을 코딩하기 위한 팔레트에 포함된 칼라 값에 기초하여 결정될 수도 있다. 1의 플래그 값은 블록의 적어도 하나의 샘플이 이스케이프 모드를 이용하여 코딩된다는 것을 표시할 수도 있다. 즉, 적어도 하나의 샘플의 값이 이스케이프 샘플로서 코딩된다. 따라서, 플래그는 블록의 적어도 하나의 샘플이 블록을 코딩하기 위한 팔레트에 포함되지 않은 칼라 값을 가지는지 여부를, 비디오 데이터의 블록의 모든 샘플들에 대해, 표시할 수도 있다.
- [0044] 따라서, 일부 예들에서, CU 레벨 이스케이프 플래그는 현재의 CU가 이스케이프 픽셀을 가질 수도 있는지 여부를 규정하는 비트스트림으로 시그널링될 수도 있다. 예를 들어, CU 레벨 이스케이프 플래그는 이스케이프 값들이 특정의 CU에 대해 인에이블되는지 여부를 표시할 수도 있다. 위에서 언급한 바와 같이, CU에서의 이스케이프 샘플의 존재는 CU에 대한 팔레트 인덱스들의 개수에 영향을 미칠 수도 있다. 예를 들어, CU에 대한 팔레트는 예컨대, 인덱스가 {0, 1, ..., N} 중에서 선택될 수 있도록, 가장 큰 인덱스 N을 갖는 인덱스들의 개수를 가질 수도 있다.
- [0045] 그러나, 일부의 경우, 현재의 블록에 이스케이프 픽셀이 있을 수도 있다는 것을 CU 레벨 이스케이프 플래그가 표시하면, 현재의 블록에서의 가능한 인덱스 값들은 {0, 1, ..., N, N+1}일 수도 있으며, 여기서, N+1과 동일한 인덱스는 현재의 픽셀이 이스케이프 픽셀이라는 것을 표시한다. 이러한 경우들에서, 코딩 중인 특정의 샘플에 있어서, 인덱스 N+1을 가지는 선택스 엘리먼트들은 양자화된 픽셀 값들일 수도 있다. 위에서 언급한 바와 같이, N+1 미만인 인덱스는 현재의 픽셀이 그 인덱스와 연관된 팔레트로부터의 칼라로 표현된다는 것을 표시할 수도 있다.
- [0046] 본 개시물의 기법들은 일반적으로 이스케이프 픽셀 코딩에 관한 것이다. 이스케이프 픽셀은 루마 샘플, 또는 블록에서의 이스케이프 픽셀 위치 및 4:4:4, 4:2:2, 4:2:0와 같은 크로마 포맷에 의존한 루마 샘플과 2개의 크로마 샘플들 사이의 조합, 및/또는 단색을 포함할 수도 있다. 칼라 성분 샘플들은 팔레트 모드의 부분으로서 코딩될 원래 픽셀을 나타내며, 시그널링된 값은 양자화될 수 있다. 루마 및 크로마 샘플들은 원래 픽셀이 가질 수도 있는 최대 가능한 값을 나타내는 상이한 비트 심도들을 가질 수도 있다. 팔레트 이스케이프 값은 이스케이프 코딩된 픽셀 값의 칼라 성분 당 시그널링된 값들을 나타내는 `Palette_escape_val`로 표현될 수도 있으며, 이 값은 이스케이프 픽셀을 복원하기 위해 역양자화될 수 있다.
- [0047] 일부 예들에서, 손실 코딩된 (`cu_transquant_bypass_flag`가 거짓일 때) 코딩 유닛들(CU)에 대한 차수 3의 지수 Golomb 코드가 `Palette_escape_val` 2진화에 이용될 수도 있으며 어떤 상한도 포함하지 않을 수도 있다. 예를 들어, 표준 제안 문서, 2015년 6월 19-26일, 폴란드, 바르샤바, Bin Li 등의, "On the Palette Escape Pixel Coding", JCTVC-U0052 (이하, JCTVC-U0052)는, 상한을 포함하지 않는 코드를 기술하고 있다.
- [0048] 이러한 경우들에서, `Palette_escape_val`은 디코더가 디코딩하도록 요구되는 임의의 큰 값을 취할 수도 있다. 또한, 도 2 및 도 3과 관련하여 아래에서 더 자세히 설명하는 바와 같이, 컨텍스트-적응 2진 산술 코딩(CABAC) 엔진은, 무한히 큰 선택스 엘리먼트들을 파싱하는데 문제가 있을 수도 있다. 예를 들어, 비제한된 큰 값들은 이스케이프 복원 프로세스 동안 오버플로우를 초래할 수도 있다. 본 개시물의 기법들은 비트스트림으로 시그널링될 수 있는 `Palette_escape_val`의 값을 규범적으로 제한하기 위해 이용될 수도 있다. 이것은 예를 들어, 적합한 비트스트림이 어떤 값보다 큰 `Palette_escape_val`을 포함하지 않을 것이라는 비트스트림 제약에 의해 정의될 수도 있고 또는 2진화 이후 어떤 개수보다 큰 빈들로 표현될 수 있다. 다른 예들에서, 이러한 제약은 또한 양자화된 신호 이스케이프 값들, 복원된 이스케이프 값들, 비양자화된 이스케이프 값들, 및/또는 이스케이프 값들, 및/또는 임의의 다른 형태의 이스케이프 값들에 적용될 수도 있다.
- [0049] 본 개시물의 다른 양태들은 양자화 파라미터들(QPs)에 관한 것이다. 예를 들어, 어떤 값, 예를 들어, 4보다 작은 QP의 사용은 입력 값이 감소되기 보다는 확장되는 것을 초래할 수도 있다. QP가 그 어떤 값(상기 예에서는 4)과 동일할 때, 명목상, 대응하는 양자화기 단계-사이즈(`qStep`)가 1과 동일할 수도 있으며, 이는 양자화 프로세스가 동일한 비수정된 값을 그의 입력 값으로서 출력한다는 것을 의미한다. QP가 어떤 값, 예를 들어, 4 미만일 때, 입력 값을 양자화하는 대신, 양자화 프로세스는 입력 값을 확장할, 즉, 입력 값을 `qStep < 1`로 나눔으로써, 입력 값을 확장할 수도 있다. 이것은 (예컨대, 입력 값(들)이 변환 계수들일 때) 입력 값의 정확도를 증가시키기 위해 사용될 수도 있다. 그러나, 1과 동일한 `qStep`을 갖는 이스케이프 코딩된 픽셀은 이미 임의의 왜곡 또는 에러 없이 입력 값의 전체 정밀도를 표현할 수도 있다.

이들의 경우, 정밀도를 더 증가시키는 것은 증가된 정확도의 관점에서 어떤 향상도 초래하지 않을 수도 있다.

이것은 이스케이프 코딩된 픽셀들에 대해서 뿐만 아니라, 변환 스킵이 이용될 때, 어떤 값, 예를 들어, 4 만인 QP들에 대해서도 발생할 수도 있다.

[0050] 본 개시물의 예들은 QP 의 범위에 관한 제한을 포함한다. 예를 들어, 더 낮은 QP 값들은 1 의 qStep 에 공칭적으로 대응하는 QP 값으로 클리핑될 수도 있다. 다른 예에서, 더 낮은 QP 값들은 이러한 시나리오에서 사용되는 것이 제한될 수도 있다.

[0051] 도 1 은 본 개시물의 기법들을 이용할 수도 있는 예시적인 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 사용될 때, 용어 "비디오 코더" 는 비디오 인코더들 및 비디오 디코더들 양쪽을 포괄적으로 지칭한다. 본 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 비디오 인코딩 또는 비디오 디코딩을 포괄적으로 지칭할 수도 있다. 비디오 코딩 시스템 (10) 의 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물에서 설명된 다양한 예들에 따라 팔레트-기반의 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스들의 예들을 나타낸다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트-기반의 코딩 또는 비-팔레트 기반의 코딩을 이용하여, HEVC 코딩에서의 CU들 또는 PU들과 같은, 비디오 데이터의 여러 블록들을 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트 기반의 코딩 모드들은 HEVC 에 의해 규정된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간 코딩 모드들 또는 인트라-예측 공간 코딩 모드들을 지칭할 수도 있다.

[0052] 도 1 에 나타난 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 및 목적지 디바이스 (14) 를 포함한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 발생한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치로서 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 발생된 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.

[0053] 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예컨대, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 자동차용 컴퓨터들, 또는 기타 등등을 포함한, 광범위한 디바이스들을 포함할 수도 있다.

[0054] 목적지 디바이스 (14) 는 소스 디바이스 (12) 로부터 채널 (16) 을 통해서 인코딩된 비디오 데이터를 수신할 수도 있다. 채널 (16) 은 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시키는 것이 가능한 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 일 예에서, 채널 (16) 은 소스 디바이스 (12) 로 하여금, 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 무선 통신 프로토콜과 같은 통신 표준에 따라서, 인코딩된 비디오 데이터를 변조할 수도 있으며, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 근거리 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 용이하게 하는 다른 장비를 포함할 수도 있다.

[0055] 또 다른 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 발생하는 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 저장 매체에, 예컨대, 디스크 액세스 또는 카드 액세스를 통해서, 액세스할 수도 있다. 저장 매체는 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 로컬-엑세스되는 데이터 저장 매체들을 포함할 수도 있다.

[0056] 추가 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 발생된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해서 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신가능한 서버의 형태일 수도 있다. 예시적인 파일 서버들은 (예컨대, 웹사이트용) 웹 서버들, 파일 전송 프로토콜 (FTP) 서버들, NAS (network attached storage) 디바이스들, 및 로컬 디스크 드라이브들을 포함

한다.

- [0057] 목적지 디바이스 (14) 는 인코딩된 비디오 데이터에 인터넷 접속과 같은 표준 데이터 접속을 통해서 액세스할 수도 있다. 데이터 접속들의 예시적인 유형들은 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, DSL, 케이블 모뎀, 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 양쪽의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다.
- [0058] 본 개시물의 기법들은 무선 애플리케이션들 또는 환경들에 한정되지 않는다. 이 기법들은 다양한 멀티미디어 애플리케이션들, 예컨대, 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 예컨대, 인터넷을 통한 위성 텔레비전 송신들, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들의 지원 하에, 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.
- [0059] 도 1 에 예시된 비디오 코딩 시스템 (10) 은 단지 일 예이며, 본 개시물의 기법들은 인코딩 디바이스와 디코딩 디바이스 사이의 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 환경들 (예컨대, 비디오 인코딩 또는 비디오 디코딩) 에 적용할 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 추출되어, 네트워크 등을 통해서 스트리밍된다. 비디오 인코딩 디바이스는 데이터를 인코딩하여 메모리에 저장할 수도 있거나, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 추출하여 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은 서로 통신하지 않지만 간단히 데이터를 메모리로 인코딩하거나 및/또는 메모리로부터 데이터를 추출하여 디코딩하는 디바이스들에 의해 수행된다.
- [0060] 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, 이전에-캡처된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하는 비디오 공급 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 이런 비디오 데이터의 소스들의 조합을 포함할 수도 있다.
- [0061] 비디오 인코더 (20) 는 비디오 소스 (18) 로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 출력 인터페이스 (22) 를 통해서 목적지 디바이스 (14) 로 직접 송신한다. 다른 예들에서, 인코딩된 비디오 데이터는 또한 디코딩 및/또는 플레이백을 위해 목적지 디바이스 (14) 에 의한 추후 액세스를 위해 저장 매체 또는 파일 서버 상으로 저장될 수도 있다.
- [0062] 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 인코딩된 비디오 데이터를 채널 (16) 을 통해서 수신할 수도 있다. 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 또는 그 외부에 있을 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32) 는 다양한 디스플레이 디바이스들, 예컨대 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스를 포함할 수도 있다.
- [0063] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 집적 회로들 (ASICs), 필드-프로그래밍가능 게이트 어레이들 (FPGAs), 이산 로직, 하드웨어, 또는 임의의 이들의 조합들과 같은, 다양한 적합한 회로 중 임의의 회로로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현되면, 디바이스는 소프트웨어용 명령들을 적합한 비일시성 컴퓨터-판독가능 저장 매체에 저장할 수도 있으며, 본 개시물의 기법들을 수행하기 위해 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 이용하여 실행할 수도 있다. (하드웨어, 소프트웨어, 하드웨어와 소프트웨어의 조합 등을 포함한) 진술한 것 중 임의의 것이 하나 이상의 프로세서들로 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 개별 디바이스에서 결합된 인코더/디코더 (CODEC) 의 일부분으로서 통합될 수도 있다.
- [0064] 본 개시물은 일반적으로 어떤 정보를 비디오 디코더 (30) 와 같은 또 다른 디바이스로 "시그널링하는" 또는 "송

신하는" 비디오 인코더 (20) 를 참조할 수도 있다. 용어 "시그널링하는" 또는 "송신하는" 은 일반적으로, 압축된 비디오 데이터를 디코딩하는데 사용되는 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 이런 통신은 실시간 또는 거의-실시간으로 일어날 수도 있다. 대안적으로, 이런 통신은 어떤 기간에 걸쳐서 일어날 수도 있으며, 예컨대 인코딩 시에 선택스 엘리먼트들을 컴퓨터-관독가능 저장 매체에 인코딩된 비트스트림으로 저장할 때에 발생할 수도 있으며, 이 선택스 엘리먼트들은 그후 이 매체에 저장되어진 후 언제라도 디코딩 디바이스에 의해 취출될 수도 있다.

[0065] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 위에서 언급된 HEVC 표준과 같은, 비디오 압축 표준에 따라서 동작하며, HEVC 에 설명되어 있다. 기본 HEVC 표준 외에도, 스케일러블 비디오 코딩, 멀티뷰 비디오 코딩, 및 HEVC 에 대한 3D 코딩 확장판들 뿐만 아니라, 소위, 범위 확장판들 (RExt) 및 스크린 콘텐츠 코딩 (SCC) 이 있다. 게다가, 예컨대, 본 개시물에서 설명하는 바와 같은, 팔레트-기반의 코딩 모드들이 HEVC 표준의 확장판에 제공될 수도 있다. 일부 예들에서, 팔레트-기반의 코딩에 대해 본 개시물에서 설명되는 기법들은 ITU-T-H.264/AVC 표준과 같은 다른 비디오 코딩 표준들, 또는 미래의 표준들에 따라서 동작하도록 구성된 인코더들 및 디코더들에 적용될 수도 있다. 따라서, HEVC 코덱에서의 코딩 유닛들 (CUs) 또는 예측 유닛들 (PUs) 의 코딩을 위한 팔레트-기반의 코딩 모드의 적용은 예의 목적을 위해 설명된다.

[0066] HEVC 및 다른 비디오 코딩 표준들에서, 비디오 시퀀스는 일반적으로 일련의 픽처들을 포함한다. 픽처들은 또한 "프레임들" 로서 지칭될 수도 있다. 픽처는 S_L , S_{Cb} 및 S_{Cr} 로 표기되는, 3개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 색차 샘플들의 2차원 어레이이다. S_{Cr} 은 Cr 색차 샘플들의 2차원 어레이이다. 색차 샘플들은 또한 본원에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우, 픽처는 단색일 수도 있으며 단지 루마 샘플들의 어레이를 포함할 수도 있다.

[0067] 픽처의 인코딩된 표현을 발생하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛들 (CTUs) 의 세트를 발생시킬 수도 있다. CTU들의 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들일 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로서 지칭될 수도 있다. HEVC 의 CTU들은 H.264/AVC 와 같은, 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 특정의 사이즈에 반드시 제한되지 않으며, 하나 이상의 코딩 유닛들 (CUs) 을 포함할 수도 있다. 슬라이스는 래스터 스캔으로 연속적으로 순서화된 CTU들의 정수를 포함할 수도 있다.

[0068] 코딩된 CTU 를 발생시키기 위해, 비디오 인코더 (20) 는 코딩 트리 블록들을 코딩 블록들, 따라서 이름 "코딩 트리 유닛들" 로 분할하기 위해 CTU 의 코딩 트리 블록들에 관해 쿼드-트리 파티셔닝을 회귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다. CU 는 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 코딩 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록일 수도 있다. CU 의 예측 유닛 (PU) 은 루마 샘플들의 예측 블록, 픽처의 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용되는 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 발생시킬 수도 있다.

[0069] 비디오 인코더 (20) 는 PU 에 대한 예측 블록들을 발생시키기 위해 인트라 예측 또는 인터 예측을 이용할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 발생시키기 위해 인트라 예측을 이용하면, 비디오 인코더 (20) 는 PU 와 연관되는 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 발생시킬 수도 있다.

[0070] 비디오 인코더 (20) 가 PU 의 예측 블록들을 발생시키기 위해 인터 예측을 이용하면, 비디오 인코더 (20) 는 PU 와 연관되는 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여, PU 의 예측 블록들을 발생시킬 수도 있다. 비디오 인코더 (20) 는 PU 의 예측 블록들을 발생시키기 위해 단방향-예측 또는 양방향-예측을 이용할 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 발생시키기 위해 단방향-예측을 이용할 때, PU 는 단일 모션 벡터 (MV) 를 가질 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 발생시키기 위해 양방향-예측을 이용할 때, PU 는 2개의 MV들을 가질 수도 있다.

[0071] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 루마, Cb, 및 Cr 블록들을 발생시킨 후, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 발생시킬 수도 있다. CU 의 루마 잔차 블록에서의 각각의 샘플

은 CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 원래 루마 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낸다. 게다가, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 발생시킬 수도 있다. CU 의 Cb 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 원래 Cb 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한 CU 에 대한 Cr 잔차 블록을 발생시킬 수도 있다. CU 의 Cr 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 원래 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다.

[0072] 더욱이, 비디오 인코더 (20) 는 쿼드-트리 파티셔닝을 이용하여, CU 의 루마, Cb, 및 Cr 잔차 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형의 블록일 수도 있다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용되는 선택스 구조들일 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관되는 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다.

[0073] 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용하여, TU 에 대한 루마 계수 블록을 발생시킬 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용하여, TU 에 대한 Cb 계수 블록을 발생시킬 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용하여, TU 에 대한 Cr 계수 블록을 발생시킬 수도 있다.

[0074] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록, 또는 Cr 계수 블록) 을 발생시킨 후, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로 변환 계수들이 변환 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감소시키도록 양자화되어 추가적인 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들에 관해 컨텍스트-적응 2진 산술 코딩 (CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피-인코딩된 선택스 엘리먼트들을 비트스트림으로 출력할 수도 있다.

[0075] 비디오 인코더 (20) 는 엔트로피-인코딩된 선택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 코딩된 픽처들 및 연관되는 데이터의 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 비트스트림은 네트워크 추상화 계층 (NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하며, 미가공 바이트 시퀀스 페이로드 (Rbsp; raw byte sequence payload) 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 선택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 규정된 NAL 유닛 유형 코드는 NAL 유닛의 형태를 나타낸다. Rbsp 는 NAL 유닛 내에 캡슐화된 정수의 바이트들을 포함하는 선택스 구조일 수도 있다. 일부의 경우, Rbsp 는 제로 비트들을 포함한다.

[0076] 상이한 유형들의 NAL 유닛들이 상이한 유형들의 Rbsp들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 유형은 픽처 파라미터 세트 (PPS) 에 대한 Rbsp 를 캡슐화할 수도 있으며, NAL 유닛의 제 2 유형은 코딩된 슬라이스에 대한 Rbsp 를 캡슐화할 수도 있으며, NAL 유닛의 제 3 유형은 SEI 에 대한 Rbsp 를 캡슐화할 수도 있으며, 기타 등등을 캡슐화할 수도 있다. (파라미터 세트들 및 SEI 메시지들에 대한 Rbsp들과는 반대로) 비디오 코딩 데이터에 대한 Rbsp들을 캡슐화하는 NAL 유닛들은 비디오 코딩 계층 (VCL) NAL 유닛들로서 지칭될 수도 있다.

[0077] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 발생하는 비트스트림을 수신할 수도 있다. 게다가, 비디오 디코더 (30) 는 비트스트림을 파싱하여, 비트스트림으로부터 선택스 엘리먼트들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 선택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하는 프로세스는 일반적으로 비디오 인코더 (20) 에 의해 수행되는 프로세스와 반대일 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재의 CU 의 PU들에 대한 예측 블록들을 결정하기 위해 PU들의 MV들을 이용할 수도 있다. 게다가, 비디오 디코더 (30) 는 현재의 CU 의 TU들과 연관되는 변환 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 변환 계수 블록들에 관해 역변환들을 수행하여, 현재의 CU 의 TU들과 연관되는 변환 블록들을 복원할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 PU들에 대한 예측 블록들의 샘플들을 현재의 CU 의 TU들의 변환 블록들의 대응하는 샘플들에 가산함으로써, 현재의 CU 의 코딩 블록들을 복원할 수도 있다. 픽처의 각각의 CU 에 대

해 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 그 픽처를 복원할 수도 있다.

[0078] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트-기반의 코딩을 수행하도록 구성될 수도 있다. 예를 들어, 팔레트-기반의 코딩에서, 위에서 설명된 인트라-예측 또는 인터-예측 코딩 기법들을 수행하는 대신, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 소위 팔레트를, 특정의 영역 (예컨대, 주어진 블록) 의 비디오 데이터를 나타내는 칼라들의 테이블로서 코딩할 수도 있다. 블록에서의 각각의 픽셀은 픽셀의 칼라를 나타내는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 픽셀 값을 팔레트에서의 적합한 값에 관련시키는 인덱스를 코딩할 수도 있다.

[0079] 상기 예에서, 비디오 인코더 (20) 는 블록에 대한 팔레트를 결정하고, 각각의 픽셀의 값을 표현하기 위해 팔레트에 엔트리를 로케이트하고, 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 팔레트 뿐만 아니라, 블록의 픽셀들에 대한 팔레트 인덱스들을 획득할 수도 있다.

비디오 디코더 (30) 는 픽셀들의 팔레트 인덱스들을 팔레트의 엔트리들에 관련시켜, 블록의 픽셀 값들을 복원할 수도 있다.

[0080] 위에서 언급한 바와 같이, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 상이한 팔레트 코딩 모드들의 수를 이용하여 팔레트의 팔레트 인덱스들을 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이스케이프 모드, 상측 복사 모드 (또한, 상부복사 모드로서 지칭됨), 또는 인덱스 모드 (또한, 값 모드 또는 좌측 모드로서 지칭됨) 를 이용하여, 블록의 팔레트 인덱스들을 코딩할 수도 있다. 일반적으로, "이스케이프 모드" 를 이용하여 샘플을 코딩하는 것은 일반적으로 블록을 코딩하기 위한 팔레트에 표현된 대응하는 칼라를 갖지 않을 수도 있는 블록의 샘플을 코딩하는 것을 지칭할 수도 있다. 위에서 언급한 바와 같이, 이러한 샘플들은 이스케이프 샘플들 또는 이스케이프 픽셀들로서 지칭될 수도 있다.

[0081] 이스케이프 시그널링과 관련하여, 비디오 인코더 (20) 는 블록의 특정의 샘플 값에 대해, 추가적인 샘플이 이스케이프 샘플 (예컨대, 블록을 코딩하기 위한 팔레트에 표현된 칼라 값을 가지지 않는 샘플) 로서 코딩된다는 것을 표시하기 위해 추가적인 팔레트 인덱스를 표현하는 데이터를 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한 이스케이프 샘플의 칼라 값(들) 을 인코딩할 수도 있다. 따라서, 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 상측 복사 모드와 인덱스 모드 사이를 식별하는 선택을 코딩할 수도 있다. 게다가, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 따라서, 코딩될 픽셀의 인덱스 값에 기초하여 (예컨대, 인덱스 값이 팔레트에서의 N 개의 엔트리들에 포함되는지 여부 또는 인덱스 값이 추가적인 인덱스 N+1 인지 여부에 기초하여) 이스케이프 코딩된 픽셀과 인덱스 모드 (또는, 상측 복사 모드) 사이를 식별할 수도 있다.

[0082] 이스케이프 모드에 있어서, 비디오 인코더 (20) 는 비트스트림으로 시그널링하며, 비디오 디코더 (30) 는 비트스트림으로, 명시적으로 시그널링된 픽셀 값을 수신한다. 일부 예들에서, 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 도 5 와 관련하여 아래에서 설명되는 기법들과 같은, 이스케이프 픽셀 코딩에 대해 본원에서 설명하는 기법들의 임의의 조합을 수행하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 그렇지 않고 비제한될 때에 과잉 문제를 일으킬 수도 있는 시그널링된 비트스트림 이스케이프 픽셀 값들에 대한 규범적인 제한을 도입할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 16, 32, 또는 다른 개수의 bin들과 같은, 최대 비트 심도에 기초하여, 이스케이프 픽셀 값들을 인코딩할 수도 있다. 이와 유사하게, 비디오 디코더 (30) 는 비트스트림으로부터 이스케이프 값을 과잉하고 최대 비트 심도에 기초하여 이스케이프 값을 디코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 양자화된 신호 이스케이프 값들, 복원된 이스케이프 값들, 비양자화된 이스케이프 값들, 및/또는 임의의 다른 형태의 이스케이프 값들에 대한 규범적인 제한을 도입할 수도 있다.

[0083] 더욱이, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 양자화 파라미터를 제한하도록 구성될 수도 있다. 즉, 아래에서 더 자세히 설명하는 바와 같이, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이스케이프 픽셀들을 코딩하기 위한 QP 를 결정할 때 본 개시물의 기법들을 구현할 수도 있다.

[0084] 도 2 는 본 개시물의 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 도 2 는 설명의 목적들을 위해 제공되며 본 개시물에서 넓게 예시되고 설명된 바와 같은 기법들의 한정으로 간주되지 않아야 한다. 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 상황에서 비디오 인코더 (20) 를 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에도 적용가능할 수도 있다.

[0085] 비디오 인코더 (20) 는 본 개시물에서 설명된 다양한 예들에 따라 팔레트-기반의 비디오 코딩을 위한 기법들을

수행하도록 구성될 수도 있는 디바이스의 일 예를 나타낸다. 예를 들어, 비디오 인코더 (20) 는 팔레트-기반의 코딩 또는 비-팔레트 기반의 코딩을 이용하여, HEVC 코딩에서의 CU들 또는 PU들과 같은, 비디오 데이터의 다양한 블록들을 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트 기반의 코딩 모드들은 HEVC 에 의해 규정된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간 코딩 모드들 또는 인트라-예측 공간 코딩 모드들을 지칭할 수도 있다. 비디오 인코더 (20) 는, 일 예에서, 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 발생시키고, 비디오 데이터의 블록에서의 적어도 일부 픽셀 로케이션들의 픽셀 값들을 표시하기 위해 그 팔레트에서 픽셀 값들을 선택하고, 그리고 비디오 데이터의 블록에서의 픽셀 로케이션들 중 적어도 일부를 그 팔레트에서의 선택된 픽셀 값들에 각각 대응하는 팔레트에서의 엔트리들과 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 시그널링된 정보는 비디오 데이터를 디코딩하기 위해 비디오 디코더 (30) 에 의해 사용될 수도 있다.

[0086] 도 2 의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 유닛 (100), 비디오 데이터 메모리 (101), 잔차 발생 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 프로세싱 유닛 (110), 복원 유닛 (112), 필터 유닛 (114), 디코딩된 픽처 버퍼 (116), 및 엔트로피 인코딩 유닛 (118) 을 포함한다. 예측 프로세싱 유닛 (100) 은 인터-예측 프로세싱 유닛 (120) 및 인트라-예측 프로세싱 유닛 (126) 을 포함한다. 인터-예측 프로세싱 유닛 (120) 은 모션 추정 유닛 및 모션 보상 유닛 (미도시) 을 포함한다. 비디오 인코더 (20) 는 또한 본 개시물에서 설명된 팔레트-기반의 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.

[0087] 비디오 데이터 메모리 (101) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩된 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (101) 에 저장된 비디오 데이터는 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. 디코딩된 픽처 버퍼 (116) 는 예컨대, 인트라- 또는 인터-코딩 모드들에서 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩할 때에 사용하기 위해 참조 비디오 데이터를 저장하는 참조 픽처 메모리 일 수도 있다. 비디오 데이터 메모리 (101) 및 디코딩된 픽처 버퍼 (116) 는 동기 DRAM (SDRAM) 을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은, 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (101) 및 디코딩된 픽처 버퍼 (116) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (101) 는 비디오 인코더 (20) 의 다른 컴포넌트들과의 온칩, 또는 그들 컴포넌트들에 대한 오프-칩일 수도 있다.

[0088] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터의 픽처의 슬라이스에서의 각각의 CTU 를 인코딩할 수도 있다. CTU들의 각각은 픽처의 동일-사이즈의 루마 코딩 트리 블록들 (CTB들) 및 대응하는 CTB들과 연관될 수도 있다. CTU 를 인코딩하는 것의 일부분으로서, 예측 프로세싱 유닛 (100) 은 쿼드-트리 파티셔닝을 수행하여, CTU 의 CTB들을 점차적으로-더 작은 블록들로 분할할 수도 있다. 더 작은 블록은 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 CTU 와 연관되는 CTB 를 4개의 동일-사이즈로된 서브-블록들로 파티셔닝하고, 서브-블록들 중 하나 이상을 4개의 동일-사이즈로된 서브-블록들로, 그리고 기타 등등으로 파티셔닝할 수도 있다.

[0089] 비디오 인코더 (20) 는 CU들의 인코딩된 표현들 (즉, 코딩된 CU들) 을 발생시키기 위해 CTU 의 CU들을 인코딩할 수도 있다. CU 를 인코딩하는 것의 부분으로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU들 중에서 CU 와 연관되는 코딩 블록들을 파티셔닝할 수도 있다. 따라서, 각각의 PU 는 루마 예측 블록 및 대응하는 크로마 예측 블록들과 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다양한 사이즈들을 가지는 PU들을 지원할 수도 있다. 위에서 나타난 바와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있으며, PU 의 사이즈는 루마 PU 의 예측 블록의 사이즈를 지칭할 수도 있다. 특정의 CU 의 사이즈가 $2N \times 2N$ 이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인트라 예측에 대해서는 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들을, 그리고 인터 예측에 대해서는 $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, 또는 등등의 대칭 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한 인터 예측에 대해서 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

[0090] 인터-예측 프로세싱 유닛 (120) 은 CU 의 각각의 PU 에 대해 인터 예측을 수행함으로써, PU 에 대한 예측 데이터를 발생시킬 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 있는지에 따라서, CU 의 PU 에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들은 인트라 예측된다.

따라서, PU 가 I 슬라이스에 있으면, 인터-예측 유닛 (121) 은 PU 에 대해 인터 예측을 수행하지 않는다.
따라서, I-모드에서 인코딩되는 블록들에 대해, 예측된 블록은 동일한 프레임 내에서 이전에-인코딩된 이웃하는 블록들로부터 공간 예측을 이용하여 형성된다.

[0091] PU 가 P 슬라이스 내에 있을 때, 인터-예측 프로세싱 유닛 (120) 의 모션 추정 유닛은 참조 픽처들의 리스트 (예컨대, "RefPicList0") 에서의 참조 픽처들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있다. PU 에 대한 참조 영역은 PU 의 샘플 블록들에 가장 가깝게 대응하는 샘플 블록들을 포함하는, 참조 픽처 내, 영역일 수도 있다. 모션 추정 유닛은 PU 에 대한 참조 영역을 포함하는 참조 픽처의 RefPicList0 에서의 위치를 나타내는 참조 인덱스를 발생시킬 수도 있다. 게다가, 모션 추정 유닛은 PU 의 코딩 블록과, 참조 영역과 연관되는 참조 로케이션 사이의 공간 변위를 나타내는 MV 를 발생시킬 수도 있다. 예를 들어, MV 는 현재의 디코딩된 픽처에서의 좌표들로부터 참조 픽처에서의 좌표들까지 오프셋을 제공하는 2차원 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 표시되는 참조 로케이션에서의 실제 또는 내삽된 샘플들에 기초하여, PU 의 예측 블록들을 발생시킬 수도 있다.

[0092] PU 가 B 슬라이스 내에 있을 때, 인터-예측 프로세싱 유닛 (120) 의 모션 추정 유닛은 PU 에 대해 단방향-예측 또는 양방향-예측을 수행할 수도 있다. PU 에 대한 단방향-예측을 수행하기 위해, 모션 추정 유닛은 RefPicList0 또는 제 2 참조 픽처 리스트 ("RefPicList1") 의 참조 픽처들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있다. 모션 추정 유닛은 참조 영역을 포함하는 참조 픽처의 RefPicList0 또는 RefPicList1 에서의 위치를 나타내는 참조 인덱스, PU 의 예측 블록과 참조 영역과 연관되는 참조 로케이션 사이의 공간 변위를 나타내는 MV, 및 참조 픽처가 RefPicList0 또는 RefPicList1 에 있는지 여부를 나타내는 하나 이상의 예측 방향 표시자들을, PU 의 모션 정보로서 출력할 수도 있다. 인터-예측 프로세싱 유닛 (120) 의 모션 보상 유닛은 PU 의 모션 벡터에 의해 표시되는 참조 영역에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여, PU 의 예측 블록들을 발생시킬 수도 있다.

[0093] PU 에 대한 양방향 인터예측을 수행하기 위해, 모션 추정 유닛은 RefPicList0 에서의 참조 픽처들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있으며, 또한 RefPicList1 에서의 참조 픽처들을 PU 에 대한 또 다른 참조 영역에 대해 탐색할 수도 있다. 모션 추정 유닛은 참조 영역들을 포함하는 참조 픽처들의 RefPicList0 및 RefPicList1 에서의 위치들을 나타내는 참조 픽처 인덱스들을 발생시킬 수도 있다. 게다가, 모션 추정 유닛은 참조 영역들과 연관되는 참조 로케이션과 PU 의 샘플 블록 사이의 공간 변위들을 나타내는 MV들을 발생시킬 수도 있다. PU 의 모션 정보는 PU 의 참조 인덱스들 및 MV들을 포함할 수도 있다. 인터-예측 프로세싱 유닛 (120) 의 모션 보상 유닛은 PU 의 모션 벡터들에 의해 표시되는 참조 영역들에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여, PU 의 예측 블록들을 발생시킬 수도 있다.

[0094] 비디오 인코더 (20) 는 팔레트-기반의 코딩을 수행하도록 구성될 수도 있다. HEVC 프레임워크에 있어서, 일 예로서, 팔레트-기반의 코딩 기법들은 코딩 유닛 (CU) 모드로서 사용되도록 구성될 수도 있다. 다른 예들에서, 팔레트-기반의 코딩 기법들은 HEVC 의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 따라서, CU 모드의 상황에서 (본 개시물 전반에 걸쳐) 본원에서 설명되는 개시된 프로세스들 모두는, 추가적으로 또는 대안적으로, PU 에 적용될 수도 있다. 그러나, 이들 HEVC-기반의 예들은, 이러한 기법들이 독립적으로 또는 다른 기준 또는 앞으로 개발될 시스템들/표준들의 일부분으로서 동작하도록 적용될 수도 있기 때문에, 본원에서 설명하는 팔레트-기반의 코딩 기법들의 제한 또는 한정으로서, 간주되지 않아야 한다. 이들 경우들에서, 팔레트 코딩을 위한 유닛은 정사각형 블록들, 비-정사각형, 직사각형의 블록들 또는 심지어 비-직사각형 형태의 영역들일 수 있다.

[0095] 팔레트-기반 인코딩 유닛 (122) 은, 예를 들어, 팔레트-기반의 인코딩 모드가 예컨대, CU 또는 PU 에 대해 선택될 때 팔레트-기반의 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 인코딩 유닛 (122) 은 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 발생시키고, 비디오 데이터의 블록의 적어도 일부 위치들의 픽셀 값들을 표시하기 위해 그 팔레트에서 픽셀 값들을 선택하고, 그리고 비디오 데이터의 블록의 위치들 중 적어도 일부를 선택된 픽셀 값들에 각각 대응하는 팔레트에서의 엔트리들과 연관시키는 정보를 시그널링하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 인코딩 유닛 (122) 에 의해 수행되는 것으로 설명되지만, 이러한 기능들 중 일부 또는 모두는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0096] 일부 예들에서, 본 개시물의 양태들에 따르면, 팔레트-기반 인코딩 유닛 (122) 은 비트스트림으로 시그널링될 수도 있는 시그널링된 비트스트림 이스케이프 픽셀 값들 (예컨대, Palette_escape_val) 에 대한 규범적인 제한

을 도입하도록 구성될 수도 있다. 이 규범적인 제한은 예를 들어, 적합한 비트스트림이 어떤 값보다 큰 이스케이프 픽셀 값, 예컨대, `Palette_escape_val` 을 포함하지 않을 수도 있다는 비트스트림 제약에 의해 정의되거나, 또는 엔트로피 코딩 2진화, 예컨대, CABAC 코딩을 위한 2진화 후 어떤 개수 보다 많은 bin들로 표현될 수도 있다. 일부 예들에서, 팔레트-기반 인코딩 유닛 (122) 은 아래에서 설명되는, 엔트로피 인코딩 유닛 (118) 을 이용하여 본 개시물의 기법들을 시행할 수도 있다.

[0097] 예를 들어, bin들의 개수는 16 또는 32 개의 bin들 내로, 또는 일반적으로는, 특정의 비디오 코딩 아키텍처가 지원할 수 있는 임의의 최대 개수의 bin들로 제한될 수도 있다. 일부 예들에서, bin들의 최대 개수는 코딩될 비디오 데이터와 연관된 특정의 레벨 또는 프로파일에 기초하여 변환될 수도 있다.

[0098] 비트스트림으로 시그널링될 수 있는 이스케이프 픽셀 값, 예컨대, `Palette_escape_val` 에 대한 최대 값이 제한될 때, 이것은 선택스 엘리먼트 표현 (예컨대, 2진화) 에서의 bin들의 개수에 대한 한계에 대응할 수도 있다. 다른 예에서, 선택스 엘리먼트 표현에서 bin들의 개수에 관해 한계를 부과하는 것은 비트스트림으로 시그널링될 수 있는 `Palette_escape_val` 에 대한 최대 값에 관한 제한에 대응한다.

[0099] 인트라-예측 프로세싱 유닛 (126) 은 PU 에 대해 인트라 예측을 수행함으로써, PU 에 대한 예측 데이터를 발생시킬 수도 있다. PU 에 대한 예측 데이터는 PU 에 대한 예측 블록들 및 다양한 선택스 엘리먼트들을 포함할 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU들에 대해 인트라 예측을 수행할 수도 있다.

[0100] PU 상에 인트라 예측을 수행하기 위해, 인트라-예측 프로세싱 유닛 (126) 은 다수의 인트라 예측 모드들을 이용하여, PU 에 대한 다수의 예측 데이터의 세트들을 발생시킬 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 이웃하는 PU들의 샘플 블록들로부터의 샘플들을 이용하여, PU 에 대한 예측 블록을 발생시킬 수도 있다. 이웃하는 PU들은 PU들, CU들, 및 CTU들에 대해 좌-우, 상-하 인코딩 순서를 가정하면, PU 의 상측에, 우상부에, 좌상부에, 또는 좌측에 있을 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 다수의 인트라 예측 모드들, 예컨대, 33개의 방향 인트라 예측 모드들을 이용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 수는 PU 와 연관되는 영역의 사이즈에 의존할 수도 있다.

[0101] 예측 프로세싱 유닛 (100) 은 PU들에 대해 인트라-예측 프로세싱 유닛 (120) 에 의해 발생된 예측 데이터, 또는 PU들에 대해 인트라-예측 프로세싱 유닛 (126) 에 의해 발생된 예측 데이터 중에서, CU 의 PU들에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여, CU 의 PU들에 대한 예측 데이터를 선택한다. 선택된 예측 데이터의 예측 블록들은 본원에서 선택된 예측 블록들로서 지칭될 수도 있다.

[0102] 잔차 발생 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 발생시킬 수도 있다. 예를 들어, 잔차 발생 유닛 (102) 은 잔차 블록들에서의 각각의 샘플이 CU 의 코딩 블록에서의 샘플과 CU 의 PU 의 대응하는 선택된 예측 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 갖도록, CU 의 잔차 블록들을 발생시킬 수도 있다.

[0103] 변환 프로세싱 유닛 (104) 은 쿼드-트리 파티셔닝을 수행하여, CU 와 연관되는 잔차 블록들을 CU 의 TU들과 연관되는 변환 블록들로 파티셔닝할 수도 있다. 따라서, TU 는 루마 변환 블록 및 2개의 크로마 변환 블록들과 연관될 수도 있다. CU 의 TU들의 루마 및 크로마 변환 블록들의 사이즈들 및 위치들은 CU 의 PU들의 예측 블록들의 사이즈들 및 위치들에 기초하거나 또는 기초하지 않을 수도 있다. "잔차 쿼드-트리" (RQT) 로서 알려진 쿼드-트리 구조는 그 영역들 각각과 연관되는 노드들을 포함할 수도 있다. CU 의 TU들은 RQT 의 리프 노드들에 대응할 수도 있다.

[0104] 변환 프로세싱 유닛 (104) 은 하나 이상의 변환들을 TU 의 변환 블록들에 적용함으로써, CU 의 각각의 TU 에 대한 변환 계수 블록들을 발생시킬 수도 있다. 변환 프로세싱 유닛 (104) 은 다양한 변환들을 TU 와 연관되는 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 변환 블록에 이산 코사인 변환 (DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (104) 은 변환 블록에 변환들을 적용하지 않는다. 이러한 예들에서, 변환 블록은 변환 계수 블록으로서 취급될 수도 있다.

[0105] 양자화 유닛 (106) 은 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 그 변환 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안 m-비트 변환 계수로 절사될 수도 있으며, 여기서, n 은 m 보다 더 크다. 양자화 유닛 (106) 은 CU

와 연관되는 양자화 파라미터 (QP) 값에 기초하여, CU 의 TU 와 연관되는 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 CU 와 연관되는 QP 값을 조정함으로써 CU 와 연관되는 계수 블록들에 적용되는 양자화의 정도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있으며, 따라서 양자화된 변환 계수들은 원래 정밀도들보다 낮은 정밀도를 가질 수도 있다. 일부 예들에서, 본 개시물의 양태들에 따르면, 양자화 유닛 (106) 은 이스케이프 픽셀들에 대한 QP 를 결정하기 위해 본원에서 설명되는 기법들을 적용할 수도 있다.

[0106] 역양자화 유닛 (108) 및 역변환 프로세싱 유닛 (110) 은 역양자화 및 역변환들을 계수 블록에 각각 적용하여, 계수 블록으로부터 잔차 블록을 복원할 수도 있다. 복원 유닛 (112) 은 그 복원된 잔차 블록을 예측 프로세싱 유닛 (100) 에 의해 발생되는 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 가산하여, TU 와 연관되는 복원된 변환 블록을 생성할 수도 있다. 이러한 방법으로 CU 의 각각의 TU 에 대한 변환 블록들을 복원함으로써, 비디오 인코더 (20) 는 CU 의 코딩 블록들을 복원할 수도 있다.

[0107] 필터 유닛 (114) 은 CU 와 연관되는 코딩 블록들에서 블록킹 아티팩트들을 감소시키기 위해, 하나 이상의 디블록킹 동작들을 수행할 수도 있다. 디코딩된 픽처 버퍼 (116) 은 필터 유닛 (114) 이 복원된 코딩 블록들에 대해 하나 이상의 디블록킹 동작들을 수행한 후 그 복원된 코딩 블록들을 저장할 수도 있다. 인터-예측 프로세싱 유닛 (120) 은 그 복원된 코딩 블록들을 포함하는 참조 픽처를 이용하여, 다른 픽처들의 PU들에 대해 인터 예측을 수행할 수도 있다. 게다가, 인트라-예측 프로세싱 유닛 (126) 은 디코딩된 픽처 버퍼 (116) 에서의 복원된 코딩 블록들을 이용하여, CU 와 동일한 픽처에서의 다른 PU들에 대해 인트라 예측을 수행할 수도 있다.

[0108] 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 있으며 예측 프로세싱 유닛 (100) 으로부터 선택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (118) 은 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여, 엔트로피-인코딩된 데이터를 발생시킬 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 컨텍스트-적응 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, V2V (variable-to-variable) 길이 코딩 동작, 선택스-기반 컨텍스트-적응 2진 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-Golomb 인코딩 동작, 또는 또다른 유형의 엔트로피 인코딩 동작들을 데이터에 대해 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩 유닛 (118) 에 의해 발생된 엔트로피-인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들어, 비트스트림은 CU 에 대한 잔차 쿼드트리 (RQT) 를 나타내는 데이터를 포함할 수도 있다.

[0109] 일부 예들에서, 본 개시물의 양태들에 따르면, 팔레트-기반 인코딩 유닛 (122) 및/또는 엔트로피 인코딩 유닛 (118) 은 비트스트림으로 시그널링될 수도 있는 시그널링된 비트스트림 이스케이프 픽셀 값들 (예컨대, `Palette_escape_val`) 에 대한 규범적인 제한을 도입하도록 구성될 수도 있다. 이것은 예를 들어, 적합한 비트스트림이 어떤 값보다 큰 `Palette_escape_val` 을 포함하지 않을 수도 있거나 또는 2진화 후 어떤 개수 보다 많은 bin들로 표현되지 않을 수도 있다는 비트스트림 제약에 의해 정의될 수도 있다. 예를 들어, 인코더 (20) 는 bin들의 개수를, 일 예에서는, 16 개의 bin들, 또는 다른 예에서는 32개로, 또는 일반적으로는, 특정의 아키텍처가 지원할 수 있는 임의의 최대 개수의 bin들로 제한할 수도 있다. 따라서, 인코더 (20) 는 특정의 개수보다 크기 않은 bin들을 가지는 이스케이프 값을 시그널링할 수도 있다. 디코더 (30) 는 설명된 방법으로 제한된 비트스트림을 수신하도록 구성될 수도 있으며, 디코더 (30) 는 인코더 (20) 에 의해 부과된 제한들에 따라서 비트스트림을 디코딩할 수도 있다.

[0110] 비트스트림으로 시그널링될 수 있는 `Palette_escape_val` 에 대한 최대 값이 제한될 때, 이것은 선택스 엘리먼트 표현 (2진화) 에서의 bin들의 개수에 관한 한계에 대응할 수도 있다. 다른 예에서, 선택스 엘리먼트 표현에서 bin들의 개수에 관해 한계를 부과하는 것은 비트스트림으로 시그널링될 수 있는 `Palette_escape_val` 에 대한 최대 값에 관한 제한에 대응한다. 일부 양태들이 `Palette_escape_val` 을 참조하여 위에서 설명되지만, 본 개시물의 하나 이상의 양태들에 따른 다른 예들에서는, 양자화된 신호 이스케이프 값들, 복원된 이스케이프 값들, 비양자화된 이스케이프 값들, 및/또는 임의의 다른 형태의 이스케이프 값들에 적용되는 제한이 도입될 수도 있다.

[0111] 도 3 은 본 개시물의 기법들을 구현하도록 구성된 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 3 은 설명의 목적들을 위해 제공되며 본 개시물에 넓게 예시되고 설명된 것과 같은 기법들에 한정하는 것이 아니다. 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 상황에서의 비디오 디코더 (30) 를 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에도 적용가능할 수도 있다.

- [0112] 비디오 디코더 (30) 는 본 개시물에서 설명된 다양한 예들에 따라 팔레트-기반의 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스의 일 예를 나타낸다. 예를 들어, 비디오 디코더 (30) 는 팔레트-기반의 코딩 또는 비-팔레트 기반의 코딩을 이용하여 HEVC 코딩에서의 CU들 또는 PU들과 같은, 비디오 데이터의 다양한 블록들을 선택적으로 디코딩하도록 구성될 수도 있다. 비-팔레트 기반의 코딩 모드들은 HEVC 에 의해 규정된 다양한 코딩 모드들과 같은, 다양한 인터-예측 시간 코딩 모드들 또는 인트라-예측 공간 코딩 모드들을 지칭할 수도 있다. 비디오 디코더 (30) 는, 일 예에서, 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 (예컨대, 비트스트림에서의 시그널링된 또는 예측된 정보에 기초하여) 발생시키고, 비디오 데이터의 블록에서의 적어도 일부 픽셀 로케이션들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신하고, 그 정보에 기초하여 팔레트에서 픽셀 값들을 선택하고, 그리고 팔레트에서의 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 복원하도록 구성될 수도 있다.
- [0113] 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), 복원 유닛 (158), 필터 유닛 (160), 및 디코딩된 픽처 버퍼 (162) 를 포함한다. 예측 프로세싱 유닛 (152) 은 모션 보상 유닛 (164) 및 인트라-예측 프로세싱 유닛 (166) 을 포함한다. 비디오 디코더 (30) 는 또한 본 개시물에서 설명된 팔레트-기반의 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.
- [0114] 비디오 데이터 메모리 (151) 는 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될, 인코딩된 비디오 비트스트림과 같은, 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (151) 에 저장된 비디오 데이터는 예를 들어, 컴퓨터-판독가능 매체로부터, 예컨대, 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해서, 또는 물리적인 데이터 저장 매체들에 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (151) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (CPB) 를 형성할 수도 있다. 디코딩된 픽처 버퍼 (162) 는 예컨대, 인트라- 또는 인터-코딩 모드들에서 비디오 디코더 (30) 에 의해 비디오 데이터를 디코딩할 때에 사용하기 위해 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (151) 및 디코딩된 픽처 버퍼 (162) 는 동기 DRAM (SDRAM) 을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은, 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (151) 및 디코딩된 픽처 버퍼 (162) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 여러 예들에서, 비디오 데이터 메모리 (151) 는 비디오 디코더 (30) 의 다른 컴포넌트들과의 온칩, 또는 그들 컴포넌트들에 대한 오프-칩일 수도 있다.
- [0115] 예컨대, 비디오 데이터 메모리 (151) 에 의해 형성된, 코딩된 픽처 버퍼 (CPB) 는 비트스트림의 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신하여 저장할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 CPB 로부터 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신하고 NAL 유닛들을 파싱하여, 선택스 엘리먼트들을 디코딩할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 NAL 유닛들에서의 엔트로피-인코딩된 선택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 복원 유닛 (158), 및 필터 유닛 (160) 은 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여, 디코딩된 비디오 데이터를 발생시킬 수도 있다.
- [0116] 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 것의 부분으로서, 엔트로피 디코딩 유닛 (150) 은 코딩된 슬라이스 NAL 유닛들로부터 선택스 엘리먼트들을 추출하여 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관련된 선택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 선택스 엘리먼트들은 슬라이스를 포함하는 픽처와 연관되는 PPS 를 식별하는 선택스 엘리먼트를 포함할 수도 있다.
- [0117] 비트스트림으로부터 선택스 엘리먼트들을 디코딩하는 것에 더해서, 비디오 디코더 (30) 는 비파티셔닝된 CU 에 대해 복원 동작을 수행할 수도 있다. 비-파티셔닝된 CU 에 대해 복원 동작을 수행하기 위해, 비디오 디코더 (30) 는 CU 의 각각의 TU 에 대해 복원 동작을 수행할 수도 있다. CU 의 각각의 TU 에 대해 복원 동작을 수행함으로써, 비디오 디코더 (30) 는 CU 의 잔차 블록들을 복원할 수도 있다.
- [0118] CU 의 TU 에 대해 복원 동작을 수행하는 것의 부분으로서, 역양자화 유닛 (154) 은 TU 와 연관되는 계수 블록들을 역양자화할 수도 있다, 즉, 양자화 해제할 수도 있다. 역양자화 유닛 (154) 은 TU 의 CU 와 연관되는 QP

값을 이용하여, 적용할 역양자화 유닛 (154) 에 대한 양자화의 정도와, 이와 유사하게, 역양자화의 정도를 결정할 수도 있다. 즉, 압축 비, 즉, 원래 시퀀스 및 압축된 시퀀스를 표현하는데 사용되는 비트수의 비는, 변환 계수들을 양자화할 때 사용되는 QP 의 값을 조정함으로써 제어될 수도 있다. 압축 비는 또한 채용되는 엔트로피 코딩의 방법에 의존할 수도 있다. 일부 예들에서, 본 개시물의 양태들에 따르면, 역양자화 유닛 (154) 은 이스케이프 픽셀들에 대한 QP 를 결정하기 위해 본원에서 설명되는 기법들을 적용할 수도 있다.

[0119] 역양자화 유닛이 계수 블록을 역양자화한 후, 역변환 프로세싱 유닛 (156) 은 TU 와 연관되는 잔차 블록을 발생하기 위해, 하나 이상의 역변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (156) 은 역 DCT, 역 정수 변환, 역 Karhunen-Loeve 변환 (KLT), 역 회전 변환, 역 방향 변환, 또는 또 다른 역변환을 계수 블록에 적용할 수도 있다.

[0120] PU 가 인트라 예측을 이용하여 인코딩되면, 인트라-예측 프로세싱 유닛 (166) 은 인트라 예측을 수행하여, PU 에 대한 예측 블록들을 발생시킬 수도 있다. 인트라-예측 프로세싱 유닛 (166) 은 공간적으로-이웃하는 PU 들의 예측 블록들에 기초하여, PU 에 대한 예측 루마, Cb 및 Cr 블록들을 발생시키기 위해, 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.

[0121] 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. 더욱이, PU 가 인트라 예측을 이용하여 인코딩되면, 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 발생시킬 수도 있다.

[0122] 복원 유닛 (158) 은 CU 의 루마, Cb, 및 Cr 코딩 블록들을 복원하기 위해, 적용가능한 경우, CU 의 TU들과 연관되는 루마, Cb, 및 Cr 변환 블록들, 및 CU 의 PU들의 예측 루마, Cb, 및 Cr 블록들, 즉, 인트라-예측 데이터 또는 인트라-예측 데이터를 이용할 수도 있다. 예를 들어, 복원 유닛 (158) 은 루마, Cb 및 Cr 변환 블록들의 샘플들을 예측 루마, Cb 및 Cr 블록들의 대응하는 샘플들에 추가하여, CU 의 루마, Cb 및 Cr 코딩 블록들을 복원할 수도 있다.

[0123] 필터 유닛 (160) 은 CU 의 루마, Cb 및 Cr 코딩 블록들과 연관되는 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행할 수도 있다. 비디오 디코더 (30) 는 CU 의 루마, Cb, 및 Cr 코딩 블록들을 디코딩된 픽처 버퍼 (162) 에 저장할 수도 있다. 디코딩된 픽처 버퍼 (162) 는 후속 모션 보상, 인트라 예측, 및 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에의 프리젠테이션을 위해, 참조 픽처들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 디코딩된 픽처 버퍼 (162) 에서의 루마, Cb, 및 Cr 블록들에 기초하여, 다른 CU들의 PU들에 대해 인트라 예측 또는 인트라 예측 동작들을 수행할 수도 있다.

[0124] 비디오 디코더 (30) 는 팔레트-기반의 코딩을 수행하도록 구성될 수도 있다. 팔레트-기반 디코딩 유닛 (165) 은, 예를 들어, 팔레트-기반의 디코딩 모드가 예컨대, CU 또는 PU 에 대해 선택될 때 팔레트-기반의 디코딩을 수행할 수도 있다. 예를 들어, 팔레트-기반 디코딩 유닛 (165) 은 픽셀 값들을 표시하는 엔트리들을 가지는 팔레트를 발생시키고, 비디오 데이터의 블록에서의 적어도 일부 픽셀 위치들을 팔레트에서의 엔트리들과 연관시키는 정보를 수신하고, 그 정보에 기초하여 팔레트에서 픽셀 값들을 선택하고, 그리고 팔레트에서의 선택된 픽셀 값들에 기초하여 블록의 픽셀 값들을 복원하도록 구성될 수도 있다. 다양한 기능들이 팔레트-기반 디코딩 유닛 (165) 에 의해 수행되는 것으로 설명되지만, 이러한 기능들 중 일부 또는 모두는 다른 프로세싱 유닛들, 또는 상이한 프로세싱 유닛들의 조합에 의해 수행될 수도 있다.

[0125] 팔레트-기반 디코딩 유닛 (165) 은 팔레트 코딩 모드 정보를 수신하고, 팔레트 코딩 모드가 그 블록에 적용된다고 팔레트 코딩 모드 정보가 표시할 때 상기 동작을 수행할 수도 있다. 팔레트 코딩 모드가 블록에 적용되지 않는다고 팔레트 코딩 모드 정보가 표시할 때, 또는 다른 모드 정보가 상이한 모드의 사용을 표시할 때, 예측 프로세싱 유닛 (152) 은, 비디오 데이터의 블록을 예컨대, 이러한 HEVC 인트라-예측 또는 인트라-예측 코딩 모드와 같은, 비-팔레트 기반의 코딩 모드를 이용하여 디코딩한다. 비디오 데이터의 블록은 예를 들어, HEVC 코딩 프로세스에 따라서 발생된 CU 또는 PU 일 수도 있다. 팔레트-기반의 코딩 모드는 복수의 상이한 팔레트-기반의 코딩 모드들 중 하나를 포함할 수도 있거나, 또는 단일 팔레트-기반의 코딩 모드가 있을 수도 있다.

[0126] 일부 예들에서, 본 개시물의 양태들에 따르면, 팔레트-기반 디코딩 유닛 (165) 및/또는 엔트로피 디코딩 유닛

(150) 은 비트스트림으로 시그널링될 수도 있는 시그널링된 비트스트림 이스케이프 픽셀 값들 (예컨대, `Palette_escape_val`) 에 대한 규범적인 제한을 도입하도록 구성될 수도 있다. 이것은 예를 들어, 적합한 비트스트림이 어떤 값보다 큰 `Palette_escape_val` 을 포함하지 않을 수도 있거나 또는 2진화 이후 어떤 개수 보다 큰 bin들로 표현되지 않을 수도 있다는 비트스트림 제약에 의해 정의될 수도 있다. 예를 들어, bin들의 개수는 일 예에서는 16 개의 bin들, 또는 다른 예에서는 32 개의 bin들 내로, 또는 일반적으로는, 특정의 아키텍처가 지원할 수 있는 임의의 최대 개수의 bin들로 제한될 수도 있다.

[0127] 비트스트림으로 시그널링될 수 있는 `Palette_escape_val` 에 대한 최대 값이 제한될 때, 이것은 그 값의 신택스 엘리먼트 표현 (2진화) 에서의 bin들의 개수에 관한 한계에 대응할 수도 있다. 다른 예에서, 신택스 엘리먼트 표현에서 bin들의 개수에 대해 한계를 부과하는 것은 비트스트림으로 시그널링될 수 있는 `Palette_escape_val` 에 대한 최대 값에 관한 제한에 대응한다.

[0128] `Palette_escape_val` 이 최대 허용된 값보다 큰 경우, 비디오 디코더 (30) 는 에러가 발생했다고 결정할 수도 있다. 이러한 예에서, 플래그가 에러를 시그널링하는데 사용될 수도 있다. 대안적으로, 추가 실시형태들에서, 디코더 (30) 는 검출된 조건을 바이트 정렬이 손실되었을 수도 있다는 표시로서 간주할 수도 있으며, 비트스트림의 이후의 위치에서의 바이트 정렬의 검출 때까지 모든 비트스트림 데이터를 폐기할 수도 있다.

[0129] 일 예에서, 비디오 인코더 (20), 팔레트-기반 인코딩 유닛 (122) 및/또는 엔트로피 인코딩 유닛 (118) 은 비트스트림으로 시그널링된 칼라 성분 입력 비트심도에 기초하여 이스케이프 픽셀 값들을 인코딩할 수도 있다. 이러한 예에서, 주어진 칼라 성분에 대한 이스케이프 값들은 0 내지 $(1 \ll \text{비트심도}) - 1$ 을 포함한 범위 내에 들도록 제한될 수도 있다. 이 수식에서, 비트심도는 문제가 되고 있는 칼라 성분에 대한 비트심도일 수도 있다. 이러한 비트심도는 인코더에 대한 입력 파라미터일 수도 있으며, 인코더에 의해 발생되거나, 또는 다른 방법으로 결정된 비트스트림에서의 시퀀스 파라미터 세트 (SPS) 로 시그널링될 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 이스케이프 픽셀 값들을, 이들이 (즉, 입력 비트심도보다는 오히려) 출력 비트심도에 기초하거나, 또는 프로세싱을 위해 인코더 또는 디코더에 의해 사용되는 임의의 내부 비트심도에 기초하는 최대 값을 초과하지 않도록, 인코딩할 수도 있다. 비디오 디코더 (30), 팔레트-기반 디코딩 유닛 (165), 및/또는 엔트로피 디코딩 유닛 (150) 은 비트스트림을 수신하고, 이스케이프 픽셀 값들의 이러한 제한에 기초하여 비트스트림을 디코딩할 수도 있다.

[0130] 다른 예에서, 비디오 인코더 (20), 팔레트-기반 인코딩 유닛 (122) 및/또는 엔트로피 인코딩 유닛 (118) 은 하나 이상의 칼라 성분들 (예컨대, 루마 또는 크로마 칼라 성분들) 에 대한 코덱 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 에 의해 지원되는 최대 비트심도에 대응하는 칼라 성분 비트심도에 기초하여 이스케이프 픽셀 값들을 인코딩할 수도 있다. 이러한 예에서, 비트심도는 스크린 콘텐츠 코딩 프로파일과 같은 어떤 프로파일에 대한 하나의 칼라 성분, 또는 모든 칼라 성분들, 및/또는 어떤 레벨들에 대한 지원되는 최대 비트심도에 대응할 수도 있다. 비디오 디코더 (30), 팔레트-기반 디코딩 유닛 (165), 및/또는 엔트로피 디코딩 유닛 (150) 은 비트스트림을 수신하고 이스케이프 픽셀 값들의 이러한 제한에 기초하여 비트스트림을 디코딩할 수도 있다. 일부 예들이 `Palette_escape_val` 의 값들에 대한 제한을 적용하거나 및/또는 강제하는 것을 참조하여 설명되었지만, 본 개시물의 하나 이상의 양태들에 따른 다른 예들에서는, 유사한 제한이 양자화된 신호 이스케이프 값들, 복원된 이스케이프 값들, 비양자화된 이스케이프 값들, 및/또는 임의의 다른 형태의 이스케이프 값들에 적용될 수도 있는 것으로 이해되어야 한다.

[0131] 다른 예들은 이스케이프 값들 또는 이스케이프 값들에 대한 최대 값에 대한 양자화의 효과들을 고려한다. 예를 들어, 비디오 인코더 (20) 및/또는 양자화 유닛 (106) 은 값들을 양자화 단계 사이즈 (qStep) 로 나눈 후 그 결과를 사사오입함으로써 이스케이프 값들과 같은 값들을 양자화할 수도 있다. 비디오 디코더 (30) 및 역양자화 유닛 (154) 은 이러한 값들을 qStep 값으로 곱함으로써 값들을 역양자화할 수도 있다. qStep 값은 양자화 파라미터의 함수로서 결정된다. 종종, qStep 값은 1 보다 클 수도 있어, 양자화 (예컨대, qStep 으로 나누는 것) 는 이스케이프 값과 같은 양자화된 값을 표현하는데 사용되는 비트수 감소를 초래할 수도 있다. 그러나, 일부 양자화 파라미터 값들에 대해, qStep 은 1 미만일 수도 있는데, 이는 양자화가 이스케이프 값을 표현하는데 사용되는 비트수를 감소시키지 않으며, 오히려, 양자화가 이스케이프 값을 표현하는데 사용되는 비트수를 확장하거나 또는 증가시킨다는 것을 의미할 수도 있다. 다시 말해서, qStep 이 1 미만인 경우, 그 값을 qStep 으로 나누는 양자화 프로세스는 그 값을 1 보다 큰 수로 효과적으로 곱하는 것이다.

[0132] 따라서, 다른 예들에서, 비디오 인코더 (20), 팔레트-기반 인코딩 유닛 (122), 엔트로피 인코딩 유닛 (118), 및/또는 양자화 유닛 (106) 은 칼라 성분 비트심도에 기초하여서 뿐만 아니라, 양자화 파라미터에 기초하여, 이스

케이프 픽셀 값들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20), 팔레트-기반 인코딩 유닛 (122) 및/또는 엔트로피 인코딩 유닛 (118)은 이스케이프 픽셀 값들에 대한 최대 값을 결정할 수도 있으며, 여기서, 최대 값은 컴포넌트 비트심도 및 양자화 파라미터 양쪽에 기초하여 결정된다. 일 예에서, 1 미만인 qStep에 의한 양자화에 기인하는 임의의 증가된 정확도는 주어진 칼라 성분에 대한 이스케이프 값들을 0 내지 $((1 \ll \text{비트심도}) - 1) * k$ 를 포함한 범위 내에 들도록 제한함으로써 수용될 수 있으며, 여기서, k는 양자화 파라미터 및/또는 qStep 값에 기초한 확장 인자이며, 비트심도는 문제가 되는 칼라 성분에 대한 (또는, 다른 칼라 성분에 대한, 또는 모든 칼라 성분들에 대한) 비트심도이다. 다른 예에서, 양자화에 기인하는 임의의 증가된 정확도는 주어진 칼라 성분에 대한 이스케이프 값들을 0 내지 $((1 \ll \text{비트심도} + b) - 1)$ 을 포함한 범위 내에 들도록 제한함으로써 수용될 수 있으며, 여기서, b는 양자화 파라미터 및/또는 qStep 값에 기초한 확장 인자이며, 비트심도는 문제가 되는 칼라 성분에 대한 (또는, 다른 칼라 성분에 대한, 또는 모든 칼라 성분들에 대한) 비트심도이다. 비디오 디코더 (30), 팔레트-기반 디코딩 유닛 (165), 엔트로피 디코딩 유닛 (150), 및/또는 역양자화 유닛 (154)은 비트스트림을 수신하고 이스케이프 픽셀 값들의 제한에 기초하여 비트스트림을 디코딩할 수도 있다.

[0133] 다른 예들에서, 1 미만인 qStep에 의한 양자화에 기인하는 증가된 정확도를 수용하기 보다는, 비디오 인코더 (20) 및/또는 양자화 유닛 (106)은 qStep 값들이 1보다 작지 않게 보장하도록 양자화 파라미터의 값들을 수정하거나 또는 제한할 수도 있다. 위에서 설명한 바와 같이, qStep이 1 미만인 경우, 비디오 인코더 (20) 및/또는 양자화 유닛 (106)에 의한 양자화는 양자화될 값의 확장을 초래할 수도 있다. 그러나, 일부의 경우, 비트스트림으로 코딩된 이스케이프 값들은 이미 임의의 왜곡 또는 에러 없이 이스케이프 값의 전체 정밀도를 표현할 수도 있다. 따라서, 이스케이프 값의 전체 정밀도가 비트스트림으로 이미 코딩되어 있을 수도 있기 때문에, 이스케이프 값을 1 미만인 qStep 값으로 양자화하는 것은 비생산적일 수도 있다. 이러한 이스케이프 값들을 확장하는 것은 코드 컴포넌트 값들의 증가된 정확도의 관점에서 어떤 향상도 제공하지 않을 수도 있으며, 비트스트림 사이즈를 바람직하지 않게 증가시킬 수도 있다.

[0134] 따라서, 일부 예들에서, 이스케이프 값을 인코딩하거나 또는 디코딩할 때, qStep 값은 1보다 작지 않도록 제한될 수도 있다. qStep 값이 양자화 파라미터 (QP)의 함수일 수도 있기 때문에, 이러한 제한은 일부 예들에서, QP에 대한 하한을 결정함으로써 수행될 수 있으며, 여기서, 하한은 1의 qStep 값에 대응하며, QP의 더 높은 값들은 1보다 큰 qStep 값들에 대응한다. 이스케이프 값을 인코딩하거나 또는 디코딩할 때, QP 값은 QP 하한보다 작지 않도록 제한될 수 있다. 다시 말해서, 이스케이프 픽셀들을 코딩하는 양자화 또는 역양자화 프로세스들에서 QP 값을 이용하기 전에, 하한 미만인 임의의 QP 값들은 하한과 동일하게 수정됨으로써, qStep이 1보다 작지 않도록 보장할 수도 있다. 이러한 예에서, 비디오 인코더 (20) 및/또는 양자화 유닛 (106)은 수정된 QP를 적용함으로써 이스케이프 값들과 같은 값들을 양자화할 수도 있으며, 비디오 디코더 (30) 및 역양자화 유닛 (154)은 수정된 QP를 적용함으로써 그 값들을 역양자화할 수도 있다.

[0135] 다른 예에서, 비디오 인코더 (20), 팔레트-기반 인코딩 유닛 (122) 및/또는 엔트로피 인코딩 유닛 (118)은 이스케이프 픽셀 값들을, 이들이 정밀도 정보 (예컨대, `extended_precision_processing_flag`)에 기초하고, 그리고 또한 칼라 성분 비트심도에 기초하는 최대 값으로 제한되도록, 인코딩할 수도 있다. 이러한 예에 대응하는 구체적인 예시적인 계산들은 아래에서 추가로 설명된다. 비디오 디코더 (30), 팔레트-기반 디코딩 유닛 (165) 및/또는 엔트로피 디코딩 유닛 (150)은 비트스트림을 수신하고 비트스트림을 정밀도 정보 (예컨대, `extended_precision_processing_flag`)에 기초하여, 그리고 또한 칼라 성분 비트심도에 기초하여 결정된 최대 값에 따라서, 디코딩할 수도 있다.

[0136] 추가적인 예들에서, 이스케이프 값들은 변환 계수들로서 취급될 수 있으며, 변환 계수들에 부과되는 모든 제한들이 이스케이프 값들의 값을 제한하기 위해 적용될 수도 있다. 또한 추가적인 예들에서, 이스케이프 값들의 2진화는 `extended_precision_processing_flag`에 기초하여 제어될 수도 있다. `extended_precision_processing_flag`의 값에 의존하여, 이스케이프 값들의 2진화는 상이하게 유도될 수도 있다. 상이한 2진화 프로세스들에 대응하는 구체적인 예들은 아래에서 추가로 설명된다.

[0137] 도 4는 본 개시물의 기법들에 따른, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 일 예를 예시하는 개념도이다. 도 4의 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. 제 1 CU (180) 및 제 2 CU (188)는 팔레트 모드 (PAL)를 이용하여 코딩된다. 아래에서 더욱더 자세하게 설명되는 바와 같이, 그리고 본 개시물의 기법들에 따르면, 제 2 팔레트들 (192)은 제 1 팔레트들 (184)에 기초한다. 픽처 (178)는 또한 인

트라-예측 코딩 모드로 코딩되는 블록 (196), 및 인터-예측 코딩 모드로 코딩되는 블록 (200) 을 포함한다.

- [0138] 도 4 의 기법들은 설명의 목적들을 위해 비디오 인코더 (20) (도 1 및 도 2) 및 비디오 디코더 (30) (도 1 및 도 3) 의 상황에서, 그리고 HEVC 비디오 코딩 표준과 관련하여 설명된다. 그러나, 본 개시물의 기법들이 이러한 방법으로 제한되지 않으며 다른 비디오 코딩 프로세스들 및/또는 표준들에서의 다른 비디오 코딩 프로세서들 및/또는 디바이스들에 의해 적용될 수도 있는 것으로 이해되어야 한다.
- [0139] 일반적으로, 팔레트는 현재 코딩중인 CU, 도 4 의 예에서는, 제 2 CU (188) 에 지배적이거나 및/또는 대표적인 픽셀 값들의 수를 지칭한다. 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 은 다수의 팔레트들을 포함하는 것으로 도시된다. 일부 예들에서, 본 개시물의 양태들에 따르면, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 CU 의 각각의 칼라 성분에 대해 별개로 팔레트들을 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 의 루마 (Y) 성분에 대한 팔레트, CU 의 크로마 (U) 성분에 대한 다른 팔레트, 및 CU 의 크로마 (V) 성분에 대한 또 다른 팔레트를 인코딩할 수도 있다. 이 예에서, Y 팔레트의 엔트리들은 CU 의 픽셀들의 Y 값들을 나타낼 수도 있으며, U 팔레트의 엔트리들은 CU 의 픽셀들의 U 값들을 나타낼 수도 있으며, V 팔레트의 엔트리들은 CU 의 픽셀들의 V 값들을 나타낼 수도 있다.
- [0140] 다른 예들에서, 비디오 인코더 (20) 는 CU 의 모든 칼라 성분들에 대해 단일 팔레트를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 Y_i , U_i , 및 V_i 를 포함하는, 삼중 값인 i -번째 엔트리를 가지는 팔레트를 인코딩할 수도 있다. 이 경우, 팔레트는 픽셀들의 컴포넌트들의 각각에 대한 값들을 포함한다. 따라서, 다수의 개개의 팔레트들을 가지는 팔레트들의 세트로서의 제 1 팔레트들 (184) 및 제 2 팔레트들 (192) 의 표현은 단지 일 예이며 한정하려는 것이 아니다.
- [0141] 도 4 의 예에서, 제 1 팔레트들 (184) 각각은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3 을 각각 가지는 3개의 엔트리들 (202-206) 을 포함한다. 엔트리들 (202-206) 은 인덱스 값들을 픽셀 값 A, 픽셀 값 B, 및 픽셀 값 C 를 각각 포함하는 픽셀 값들에 관련시킨다. 앞에서 설명한 바와 같이, 제 1 CU (180) 의 실제 픽셀 값들을 코딩하기 보다는, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 인덱스들 1-3 을 이용하여 블록의 픽셀들을 코딩하기 위해 팔레트-기반의 코딩을 이용할 수도 있다. 즉, 제 1 CU (180) 의 각각의 픽셀 위치에 대해, 비디오 인코더 (20) 는 픽셀에 대한 인덱스 값을 인코딩할 수도 있으며, 인덱스 값은 제 1 팔레트들 (184) 중 하나 이상에서의 픽셀 값과 연관된다. 비디오 디코더 (30) 는 비트스트림으로부터 인덱스 값들을 획득하고, 인덱스 값들 및 제 1 팔레트들 (184) 중 하나 이상을 이용하여 픽셀 값들을 복원할 수도 있다. 따라서, 제 1 팔레트들 (184) 은 비디오 인코더 (20) 에 의해, 팔레트-기반의 디코딩에서 비디오 디코더 (30) 에 의한 이용을 위해, 인코딩된 비디오 데이터 비트스트림으로 송신된다.
- [0142] 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 결정되는 하나 이상의 블록들을 로케이트할 수도 있다. 예측의 목적들을 위해 사용될 엔트리들의 조합은 예측자 팔레트로서 지칭될 수도 있다.
- [0143] 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 3개의 엔트리들 (208-212) 을 포함한다. 엔트리들 (208-212) 은 팔레트 인덱스들을 픽셀 값 A, 픽셀 값 B, 및 픽셀 값 D 를 각각 포함하는 픽셀 값들에 관련시킨다. 이 예에서, 비디오 인코더 (20) 는 (예측자 팔레트가 블록들의 수의 엔트리들을 포함할 수도 있지만, 예측자 팔레트를 표현하는) 제 1 팔레트들 (184) 의 어느 엔트리들이 제 2 팔레트들 (192) 에 포함되는지를 표시하는 하나 이상의 선택스 엘리먼트들을 코딩할 수도 있다.
- [0144] 도 4 의 예에서, 하나 이상의 선택스 엘리먼트들이 벡터 (216) 로서 예시된다. 벡터 (216) 는 다수의 연관된 빈들 (또는, 비트들) 을 가지며, 각각의 빈은 그 빈과 연관된 팔레트 예측자가 현재의 팔레트의 엔트리를 예측하는데 사용되는지 여부를 표시한다. 예를 들어, 벡터 (216) 는 제 1 팔레트들 (184) 의 첫번째 2개의 엔트리들 (202 및 204) 이 제 2 팔레트들 (192) 에 포함되며 (벡터 (216) 에서 "1" 의 값), 반면 제 1 팔레트들 (184) 의 제 3 엔트리가 제 2 팔레트들 (192) 에 포함되지 않는다 (벡터 (216) 에서 "0" 의 값) 는 것을 표시한다. 도 4 의 예에서, 벡터는 불린 벡터이다. 벡터는 팔레트 예측 벡터로서 지칭될 수도 있다.
- [0145] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트 예측을 수행할 때 (팔레트 예측자 테이블로서 또한 지칭될 수도 있는) 팔레트 예측자 리스트를 결정할 수도 있다. 팔레트 예측자 리스트는 현재의 블록을 코딩하기 위한 팔레트의 하나 이상의 엔트리들을 예측하는데 사용되는 하나 이상의 이웃하는 블록들의

팔레트들로부터의 엔트리들을 포함할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 리스트를 동일한 방법으로 구성할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 어느 팔레트 예측자 리스트의 엔트리들이 현재의 블록을 코딩하기 위한 팔레트에 포함되어야 하는지를 표시하기 위해 (벡터 (216) 와 같은) 데이터를 코딩할 수도 있다.

[0146] 도 5 는 본 개시물의 기법들에 따른, 픽셀들의 블록에 대한 팔레트로의 팔레트 인덱스들을 결정하는 일 예를 예시하는 개념도이다. 예를 들어, 도 5 는 팔레트 인덱스들과 연관된 픽셀들의 개별 위치들을 팔레트들 (244) 의 엔트리에 관련시키는 팔레트 인덱스들의 맵 (240) 을 포함한다. 예를 들어, 인덱스 1 은 값 A 와 연관되며, 인덱스 2 는 값 B 와 연관되며, 인덱스 3 은 값 C 와 연관된다. 게다가, 이스케이프 샘플들이 암시적인 이스케이프 시그널링을 이용하여 표시될 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한 인덱스 4 와 연관된 맵 (240) 의 샘플들이 이스케이프 샘플들임을 표시할 수도 있는, 도 5 에 인덱스 4 로서 예시된, 추가적인 인덱스를 팔레트들 (244) 에 추가할 수도 있다. 이 경우, 비디오 인코더 (20) 는 픽셀 값이 팔레트들 (244) 에 포함되지 않으면 맵 (240) 에서의 위치에 대한 실제 픽셀 값 (또는, 그의 양자화된 버전) 의 표시를 인코딩할 수도 있다 (비디오 디코더 (30) 는 인코딩된 비트스트림으로부터, 획득할 수도 있다).

[0147] 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) 엔트리가 CU 의 (i, j) 위치에 대응한다고 가정하자. 비디오 인코더 (20) 는 엔트리가 연관된 인덱스 값을 가지는지 여부를 표시하는 맵의 각각의 엔트리 (즉, 각각의 픽셀 위치) 에 대한 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 에서 (i, j) 로케이션에서의 픽셀 값이 팔레트들 (244) 에서의 값들 중 하나임을 표시하기 위해 1 의 값을 가지는 플래그를 인코딩할 수도 있다. 비디오 인코더 (20) 는 이러한 예에서, 또한 팔레트에서의 그 픽셀 값을 표시하여 비디오 디코더로 하여금 픽셀 값을 복원가능하게 하기 위해 (도 5 의 예에서 값들 1-3 으로 나타낸) 팔레트 인덱스를 인코딩할 수도 있다.

[0148] 팔레트들 (244) 이 단일 엔트리 및 연관된 픽셀 값을 포함하는 경우에, 비디오 인코더 (20) 는 인덱스 값의 시그널링을 스킵할 수도 있다. 비디오 인코더 (20) 는 CU 에서 (i, j) 로케이션에서의 픽셀 값이 팔레트들 (244) 에서의 값들 중 하나가 아니라는 것을 표시하기 위해 0 의 값을 갖도록 플래그를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20) 는 또한 픽셀 값을 복원할 때에 비디오 디코더 (30) 에 의한 사용을 위해 픽셀 값의 표시를 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 픽셀 값을 명시적으로 코딩할 수도 있다. 일부의 경우, 픽셀 값은 손실 방식으로 코딩될 수도 있다.

[0149] CU 의 하나의 위치에서의 픽셀의 값은 CU 의 다른 위치들에서의 하나 이상의 다른 픽셀들의 값들의 표시를 제공할 수도 있다. 예를 들어, (하나 보다 많은 픽셀 값이 단일 인덱스 값에 맵핑될 수도 있는 손실 코딩의 경우에) CU 의 이웃하는 픽셀 위치들이 동일한 픽셀 값을 가지거나 또는 동일한 인덱스 값에 맵핑될 수도 있는 상대적으로 높은 확률이 존재할 수도 있다.

[0150] 따라서, 비디오 인코더 (20) 는 함께 코딩되는, 주어진 스캔 순서에서 연속된 픽셀들 또는 인덱스 값들의 수를 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 위에서 언급한 바와 같이, 팔레트 인덱스들의 스트링 (또는, 팔레트 인덱스들에 의해 표시되는 픽셀 값들) 은 본원에서 런 (run) 으로서 지칭될 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터, 런을 표시하는 신택스 엘리먼트들을 획득하고 그 데이터를 이용하여 동일한 픽셀 또는 인덱스 값을 가지는 연속된 로케이션들의 개수를 결정할 수도 있다.

[0151] 위에서 언급한 바와 같이, 런들은 상측 복사 모드 또는 인덱스 모드와 함께 이용될 수도 있다. 일 예에서, 예시의 목적을 위해, 맵 (240) 의 로우들 (264 및 268) 을 고려하자. 수평한, 좌우 스캔 방향을 가정하면, 로우 (264) 는 "1" 의 3개의 팔레트 인덱스들, "2" 의 2개의 팔레트 인덱스들, 및 "3" 의 3개의 팔레트 인덱스들을 포함한다. 로우 (268) 는 "1" 의 5개의 팔레트 인덱스들, "3" 의 2개의 팔레트 인덱스들, 및 이스케이프 샘플로서 지칭될 수도 있는, (인덱스 4 로 표현된) 팔레트들 (244) 에 포함되지 않은 하나의 샘플을 포함한다.

[0152] 이 예에서, 비디오 인코더 (20) 는 상측 복사 모드를 이용하여, 로우 (268) 에 대한 데이터를 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 로우 (268) 의 제 1 위치 (로우 (268) 의 최좌측 위치) 가 로우 (264) 의 제 1 위치와 동일하다는 것을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한 로우 (268) 에서 스캔 방향으로 2개의 연속된 엔트리들의 다음 런이 로우 (264) 의 제 1 위치와 동일하다는 것을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다.

- [0153] 로우 (264) 의 제 1 위치 및 (위에서 언급된) 2개의 엔트리들의 런을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩한 후, 비디오 인코더 (20) 는 값 모드를 이용하여, (좌측으로부터 우측으로) 로우 (268) 에서의 제 4 및 제 5 위치들을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 제 4 위치에 대해 1 의 값을 표시하는 하나 이상의 신택스 엘리먼트들 및 1 의 런 (예컨대, 값 모드) 을 표시하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있다. 따라서, 비디오 인코더 (20) 는 다른 라인을 참조함이 없이 이들 2개의 위치들을 인코딩한다.
- [0154] 비디오 인코더 (20) 는 그후 상부 로우 (264) 에 대한 상측 복사 모드를 이용하여 로우 (268) 에서 3 의 인덱스 값을 가지는 제 1 위치를 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 상측 복사 모드 및 1 의 런을 시그널링할 수도 있다. 따라서, 비디오 인코더 (20) 는 라인의 다른 값들에 대한 라인의 팔레트 인덱스들 또는 코딩 픽셀 값들 사이를, 예컨대, 런, 다른 라인 (또는, 칼럼) 의 값들에 대한 라인의 팔레트 인덱스들 또는 코딩 픽셀 값들, 또는 이들의 조합을 이용하여, 선택할 수도 있다. 비디오 인코더 (20) 는 일부 예들에서, 레이트/왜곡 최적화를 수행하여 선택을 행할 수도 있다.
- [0155] 비디오 인코더 (20) 는 그후 팔레트들 (244) 에 포함되지 않은, (좌측으로부터 우측으로) 로우 (268) 의 최종 샘플에 대한 이스케이프 샘플을 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 로우 (268) 의 최종 위치를 이스케이프 샘플로서 인코딩할 수도 있다. 즉, 비디오 인코더 (20) 는 로우 (268) 의 최종 위치가 이스케이프 샘플 (예컨대, 인덱스 4) 이라는 표시 뿐만 아니라, 샘플 값의 표시를 인코딩할 수도 있다. 비디오 디코더 (30) 는 인코딩된 비트스트림으로부터 위에서 설명한 신택스를 획득하고 이러한 신택스를 이용하여 로우 (268) 를 복원할 수도 있다.
- [0156] 일 예에서, 이스케이프 샘플을 시그널링하거나 또는 코딩하기 위해, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 추가적인 인덱스를 팔레트들 (244) (엔트리 인덱스 4) 에 추가할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 팔레트들 (244) 에 대한 추가적인 인덱스를 이용하여 샘플이 이스케이프 샘플, 예컨대, 인덱스 4 로 코딩된다는 것을 표시할 수도 있다. 그러나, 추가적인 인덱스는 연관된 칼라 값을 갖지 않는다. 오히려, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한 추가적인 인덱스와 연관된 각각의 샘플에 대한 칼라 값들을 코딩한다. 샘플이 이스케이프 샘플로서 코딩되지 않으면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 palette_mode 신택스 엘리먼트와 같은, 그 모드가 상측 복사 모드 또는 인덱스 모드인지 여부를 표시하는 데이터를 코딩할 수도 있다.
- [0157] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 데이터의 블록의 모든 샘플들에 대해, 블록의 적어도 하나의 샘플이 블록에 대한 칼라들의 팔레트에 포함되지 않은 칼라 값에 기초하여 코딩되는지 여부를 표시하는 하나 이상의 블록-레벨 신택스 엘리먼트들을 코딩하도록 구성될 수도 있다. 도 5 의 예에 있어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 맵 (240) 의 적어도 하나의 샘플이 이스케이프 샘플, 즉, 로우 (268) 의 최종 샘플로서 코딩된다는 것을 표시하는, 맵 (240) 과 연관된 하나 이상의 신택스 엘리먼트들을 코딩할 수도 있다.
- [0158] 일 예에서, 하나 이상의 신택스 엘리먼트들은 블록-레벨 이스케이프 플래그일 수도 있다. 예를 들어, 비디오 인코더 (20) 는 맵 (240) 이 이스케이프 샘플로서 코딩된 샘플을 포함한다는 것을 표시하기 위해 1 의 값을 가지는 이스케이프 플래그를 인코딩할 수도 있다. 이와 유사하게, 비디오 디코더 (30) 는 맵 (240) 이 이스케이프 샘플로서 코딩된 샘플을 포함한다는 것을 표시하는, 1 의 값을 가지는 이스케이프 플래그를 디코딩할 수도 있다. 따라서, 이스케이프 플래그에 따라서 맵 (240) 을, 비디오 인코더 (20) 는 인코딩할 수도 있으며, 비디오 디코더 (30) 는 디코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이스케이프 샘플들로서 코딩된 샘플들을 표현하는데 사용될 수도 있는 팔레트들 (244) 에 인덱스 4 를 추가할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 맵 (240) 의 코딩 동안 이러한 추가적인 인덱스를 이용할 수도 있다.
- [0159] 일 예로서, 비디오 디코더 (30) 는 비트스트림으로부터 맵 (240) 과 연관된 팔레트 코딩된 블록에 대한 데이터를 파싱할 수도 있다. 비디오 디코더 (30) 는 블록에서의 하나 이상의 샘플들이 이스케이프 픽셀들로서 코딩될 수 있는지 여부를 표시하는, 블록에 대한 블록-레벨 이스케이프 플래그를 디코딩할 수도 있다. 이스케이프 샘플들로서 코딩된 샘플들이 있을 수도 있다는 것을 블록-레벨 이스케이프 플래그가 표시하면, 비디오 디코더 (30) 는 인덱스 4 를 팔레트들 (244) 에 추가할 수도 있다. 비디오 디코더 (30) 는 또한, 다른 샘플들의 런에서 코딩되지 않은 샘플들 (예컨대, 로우 (264) 및 로우 (268) 과 관련하여 위에서 설명한 것들과 같은, 명시적으로 코딩중인 샘플들) 에 대해, (예컨대, palette_mode 플래그와 같은) 팔레트 코딩 모드를 표시하는 하

나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다.

- [0160] 상기 예에서, 비디오 디코더 (30) 는 또한 샘플들에 대한 팔레트 인덱스를 디코딩할 수도 있다. 비디오 디코더 (30) 는 디코딩된 팔레트 인덱스를 이용하여, 디코딩된 샘플에 대한 샘플 값을 디코딩하는 방법을 결정할 수도 있다. 예를 들어, 샘플이 4 의 인덱스 (예컨대, 이스케이프 샘플들과 연관된 인덱스) 를 가지면, 비디오 디코더 (30) 는 샘플이 이스케이프 코딩된 샘플이라고 결정할 수도 있으며, 샘플에 대한 샘플 값을 디코딩할 수도 있다. 샘플이 4 이외의 임의의 인덱스를 가지면, 비디오 디코더 (30) 는 디코딩된 인덱스에 기초하여 팔레트들 (244) 로부터 적합한 샘플 값을 결정할 수도 있다. 비디오 디코더 (30) 는 또한 주어진 스캔 순서에서 현재의 샘플로 코딩중인 다른 샘플들의 런을 표시하는 하나 이상의 선택스 엘리먼트들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 팔레트들 (244) 의 사이즈, 예컨대, 4개의 인덱스들과 동일한 입력 파라미터들을 가지는 트렁케이트된 2진 코드를 이용하여 인덱스를 디코딩할 수도 있다. 결과적인 트렁케이트된 2진 코드는 접두부 및 접미부 (예컨대, 트레일링 1 또는 0) 을 포함할 수도 있다.
- [0161] 본 개시물의 기법들은 (예컨대, 비디오 인코더 (20) 에 의해) 비트스트림으로 인코딩되거나 또는 (예컨대, 비디오 디코더 (30) 에 의해) 인코딩된 비트스트림으로부터 디코딩될 수도 있는, `Palette_escape_val` 의 값을 규범적으로 제한하는 것을 포함할 수도 있다. 이러한 제한은 예를 들어, 적합한 비트스트림이 어떤 값보다 큰 `Palette_escape_val` 을 포함하지 않아야 하거나 또는 2진화 이후 어떤 개수 보다 큰 bin들로 표현되지 않아야 한다는 비트스트림 제약에 의해 정의될 수 있다.
- [0162] 이러한 제한은 시그널링된 이스케이프 값들 (즉, `Palette_escape_val`) 뿐만 아니라, 양자화된 신호 이스케이프 값들, 복원된 이스케이프 값들, 비양자화된 이스케이프 값들, 및/또는 임의의 다른 형태의 이스케이프 값들에도 적용할 수도 있다. (복원된 이스케이프 값들은 양자화된 이스케이프 값들에 역양자화를 적용함으로써 결정될 수도 있다.) 예시의 목적을 위해, 값들을 제한하는 하나 이상의 기법들이 `Palette_escape_val` 또는 다른 값들을 참조하여 설명될 수도 있으나, 동일한 또는 유사한 기법들이 양자화된 신호 이스케이프 값들, 복원된 이스케이프 값들, 비양자화된 이스케이프 값들, 및/또는 이스케이프 값들과 같은, 다른 양들에 적용될 수도 있다. 따라서, 값을 주어진 최대 값으로 규범적으로 제한하는 본원에서 설명되는 임의의 기법은 시그널링된 이스케이프 값들 (즉, `Palette_escape_val`) 뿐만 아니라, 양자화된 신호 이스케이프 값들, 복원된 이스케이프 값들, 비양자화된 이스케이프 값들, 및/또는 임의의 다른 형태의 이스케이프 값들에도 적용할 수도 있다.
- [0163] 예를 들어, bin들의 개수는 16 개의 bin들 이내 (즉, 작거나 또는 동일하게), 또는 32 개의 bin들 이내, 또는 일반적으로, 특정의 아키텍처가 지원할 수 있는 임의의 최대 개수의 bin들로 제한될 수도 있다.
- [0164] 비트스트림으로 시그널링될 수 있는 `Palette_escape_val` 에 대한 최대 값이 제한될 때, 이것은 선택스 엘리먼트 표현 (2진화) 에서의 bin들의 개수에 관한 한계에 대응한다. 대안적으로, 선택스 엘리먼트 표현에서 bin들의 개수에 관해 한계를 부과하는 것은 비트스트림으로 시그널링될 수 있는 `Palette_escape_val` 에 대한 최대 값에 관한 제한에 대응한다.
- [0165] `Palette_escape_val` 선택스 엘리먼트의 값은 예를 들어, 적어도 다음 중 하나에 의해 제한될 수 있다:
- [0166] 칼라 성분 당 시퀀스 파라미터 세트 (SPS) 로 시그널링될 수도 있는 입력 비트심도가 알려져 있기 때문에, 칼라 성분의 최대 가능한 값도 또한 알려져 있을 수도 있다. 이러한 칼라 성분의 최대 가능한 값은 $(1 \ll \text{bitDepth}) - 1$ 로서 계산될 수도 있으며, 여기서, `bitDepth` 는 칼라 성분 비트심도이고, ' $\ll$ ' 연산자는 2 의 보수 정수 표현의 산술적 좌측 시프트를 나타낸다. 복원된 이스케이프 코딩된 샘플 칼라 성분 값은 0 과 $(1 \ll \text{bitDepth}) - 1$ 을 포함한 이들 사이의 범위일 수도 있다. 따라서, 이스케이프 픽셀 샘플 값의 양자화된 표현일 수도 있는, `Palette_escape_val` 은 입력된 또는 복원된 픽셀 값들과 동일한 범위 내로 제한될 수도 있다. 다시 말해서, 적합한 비트스트림은 $(1 \ll \text{bitDepth}) - 1$ 보다 큰 `Palette_escape_val` 값들을 포함하지 않을 수도 있으며, 여기서, `bitDepth` 는 칼라 성분 입력 비트심도이다. 추가적인 예들에서, `bitDepth` 는 또한 출력 비트심도, 또는 프로세싱에 사용되는 임의의 내부 비트심도일 수도 있다.
- [0167] 위에서 설명된 것에 추가하여, `Palette_escape_val` 의 범위는 예를 들어, QP 값들이 4 보다 작을 때 정확도 증가를 수용하기 위해 확장될 수도 있다. 이 경우, `Palette_escape_val` 값들의 범위는 0 과 $((1 \ll \text{bitDepth}) - 1) * k$ 사이에서 정의될 수 있으며, 여기서, `k` (> 1) 는 확장 인자이다. 예를 들어, `Palette_escape_val` 은 $1/q\text{Step}0$ 으로서 정의될 수도 있으며, 여기서, `qStep0` 은 0 과 동일한 QP 에 대한 양자화 단계이다. 대안적인 예에서, `Palette_escape_val` 은 임의의 다른 미리 정의된 QP 값일 수도 있다.
- [0168] 다른 예에서, QP 가 4 미만인 경우와 같은, 어떤 QP 값들과 연관된 범위 증가는, 표현된 `Palette_escape_val` 값

들을 0 과 $(1 \ll (\text{bitDepth} + b)) - 1$ 사이의 범위 이내로 제한함으로써 해결될 수도 있으며, 여기서, b 는 예를 들어, $1/\text{qStep0}$ 과 연관된 정확도 증가를 나타내는 자연수이다. 일 예에서, b 의 값은 1 과 동일하게 설정된다. 그러나, 다른 예들에서, b 의 값은 다른 값으로 설정된다.

[0169] 대안적으로, 본 개시물의 다른 예에서, 유사한 양자화 프로세스가 이스케이프 샘플들에 적용되는 것과 같은 변환 계수들에 사용될 수도 있다. 이러한 예에서, 신택스 엘리먼트 `Palette_escape_val` 은 변환 계수로서 취급될 수도 있으며, 변환 계수들에 대해 부과되는 제한들이 이 신택스 엘리먼트의 시그널링된 값을 제한하기 위해 적용될 수도 있다. 좀더 구체적으로, 추가적인 예에서, 이스케이프 픽셀 코딩은 옵션적으로 동일한 Rice 파라미터 적용을 이용하는 것과 함께, 크기, 시그널링, 및 2진화에 대한 제한을 포함하는 것과 같은, 프로세스의 모든 양태들이 차용되는 변환 계수 코딩의 변환 스킵 부분을 따를 수 있다. 일부 예들에서, 프로세스의 단지 일부 양태들만이 차용되며; 일 예에서, 예를 들어, 단지 크기에 관한 제한만이 재사용된다. 다른 예에서, 계수 코딩에 사용되는 2진화가 크기 제한과 함께 사용된다. 고정된 Rice 파라미터가 Rice 파라미터의 적용 대신 사용될 수도 있다. 다른 예에서, 사용된 고정된 Rice 파라미터는 3 일 수도 있다. 대안적인 예들에서, 고정된 Rice 파라미터는 3 보다 크거나 또는 작을 수도 있다.

[0170] 일 예에서, 변수들 `Palette_escape_val_MinY`, `Palette_escape_val_MinC`, `Palette_escape_val_MaxY` 및 `Palette_escape_val_MaxC` 는 다음과 같이 유도될 수도 있다:

[0171] $\text{Palette_escape_val_Min}_Y = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepth}_Y + 6) : 15))$

[0172] $\text{Palette_escape_val_Min}_C = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepth}_C + 6) : 15))$

[0173] $\text{Palette_escape_val_Max}_Y = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepth}_Y + 6) : 15)) - 1$

[0174] $\text{Palette_escape_val_Max}_C = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepth}_C + 6) : 15)) - 1$

[0175] 여기서, 스테이트먼트 $x ? y : z$ 는, x 가 참이거나 또는 0 과 동일하지 않으면, 스테이트먼트는 y 의 값에 대해 평가하고; 그렇지 않으면, 스테이트먼트는 z 의 값에 대해 평가한다는 것을 나타낸다. `extended_precision_processing_flag` 는 확장된 동적 범위가 사용될 때는 참 (TRUE) 으로 설정되고 확장된 동적 범위가 사용되지 않을 때는 거짓 (FALSE) 으로 설정된다.

[0176] 본 개시물의 일 예에서, 인코더 (20) 는 신택스 엘리먼트 `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` 의 값을, 0 과 동일한 `cIdx` 에 대해서는, `Palette_escape_val_MinY` 내지 `Palette_escape_val_MaxY` 를 포함한 범위로, 그리고, 0 과 동일하지 않은 `cIdx` 에 대해서는, `Palette_escape_val_MinC` 내지 `Palette_escape_val_MaxC` 를 포함한 범위로, 제약할 수도 있다. `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` 는 (현재의 팔레트에서의 i -번째 엘리먼트를 규정하는 `CurrentPaletteEntries[ cIdx ][ i ]` 에 의해 표현되는 어레이에 대한 인덱스인 팔레트 인덱스를 규정하는) `PaletteIndexMap[ xC ][ yC ]` 가 (현재의 코딩 유닛에 대한 팔레트 인덱스에 대한 최대 가능한 값을 규정하는) `MaxPaletteIndex` 와 동일하고 `Palette_escape_val_present_flag` 가 1 과 동일한 샘플의 이스케이프 값을 규정한다. 어레이 인덱스 `cIdx` 는 칼라 성분을 규정한다. 어레이 인덱스 `cIdx` 는 칼라 성분을 규정할 수도 있다. 어레이 인덱스들 `xC` 및 `yC` 는 픽처의 좌상부 루마 샘플에 대한 샘플의 로케이션 (`xC`, `yC`) 를 규정할 수도 있다. 디코더 (30) 는 인코더에 의해 부과되는 제약들을 조건으로 하여 비트스트림을 수신할 수도 있으며, 디코더 (30) 는 제약들에 기초하여 `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` 를 복원할 수도 있다.

[0177] 대안적인 예에서, 인코더 (20) 는 `Palette_escape_val` 또는 `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` 의 범위를, 0 내지 $(1 \ll \text{bitDepthMax}) - 1$ 을 포함한 범위로 제한할 수도 있다. `bitDepthMax` 는 최대 비트심도, 예를 들어, 루마 또는 크로마 성분들 또는 양쪽에 대한 코덱에 의해 지원되는 최대 비트심도, 예를 들어, 8, 9, 10, 11, 12, 13, 14, 15, 16, 등과 같은 `bitDepthMax` 값들일 수도 있다. 일 예에서, `bitDepthMax` 는 스크린 콘텐츠 코딩 (SCC) 프로파일과 같은 어떤 프로파일에 대한 모든 칼라 성분들, 및/또는 어떤 레벨들 중에서 지원되는 최대 비트심도일 수 있다. 다른 예에서, `bitDepthMax` 는 항상 15 와 동일하다.

- [0178] 본 개시물의 추가적인 예에서, $\text{PaletteEscapeVal}[cIdx][xC][yC]$ 의 값은 0 과 동일한 $cIdx$ 에 대해서는, 0 내지 $(1 \ll \text{bitdepthMax}) - 1$ 을 포함한 범위로, 그리고, 0 과 동일하지 않은 $cIdx$ 에 대해서는, 0 내지 $(1 \ll \text{bitdepthMax}) - 1$ 을 포함한 범위로, 제약될 수도 있다.
- [0179] 대안적인 예에서, $\text{Palette_escape_val}$ 또는 $\text{PaletteEscapeVal}[cIdx][xC][yC]$ 의 범위는 0 내지 $(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{비트심도} + 6) : 15)) - 1$ 을 포함한 범위로 제한된다. 범위는 비트심도, 및 아래에서 정의된 바와 같은 $\text{extended_precision_processing_flag}$ 에 기초할 수도 있다:
- [0180] 변수들 $\text{Palette_escape_val_Min}_Y$, $\text{Palette_escape_val_Min}_C$, $\text{Palette_escape_val_Max}_Y$ 및 $\text{Palette_escape_val_Max}_C$ 는 다음과 같이 유도될 수도 있다:
- [0181] $\text{Palette_escape_val_Min}_Y = 0$
- [0182] $\text{Palette_escape_val_Min}_C = 0$
- [0183] $\text{Palette_escape_val_Max}_Y = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15)) - 1$
- [0184] $\text{Palette_escape_val_Max}_C = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15)) - 1$
- [0185] 추가적인 예에서, $\text{PaletteEscapeVal}[cIdx][xC][yC]$ 의 값은 0 과 동일한 $cIdx$ 에 대해서는, $\text{Palette_escape_val_Min}_Y$ 내지 $\text{Palette_escape_val_Max}_Y$ 를 포함한 범위 내로, 그리고, 0 과 동일하지 않은 $cIdx$ 에 대해서는, $\text{Palette_escape_val_Min}_C$ 내지 $\text{Palette_escape_val_Max}_C$ 를 포함한 범위 내로, 제약될 수도 있다.
- [0186] 추가적인 대안적인 예에서, $\text{Palette_escape_val}$ 의 2진화는 아래에서 설명되는 바와 같은 $\text{extended_precision_processing_flag}$ 에 기초하여 제어된다:
- [0187] $\text{extended_precision_processing_flag}$ 가 0 과 동일한 예에서, $\text{Palette_escape_val}$ 의 2진화는 k-차수 지수-Golomb (EGk) 2진화 프로세스를 호출함으로써 유도될 수도 있다. 일 예에서, k 는 3 과 동일하게 설정될 수도 있다. 대안적인 예에서, k 는 임의의 값, 예컨대, 0, 1, 2, 3, 4, 5, 6, 등으로 설정될 수도 있다.
- [0188] $\text{extended_precision_processing_flag}$ 가 1 과 동일한 예에서, $\text{Palette_escape_val}$ 의 2진화는 제한된 k-차수 EGk 2진화 프로세스를 호출함으로써 유도된다. 예시적인 예에서, $\text{Palette_escape_val}$ 의 2진화는 3 과 동일하게 설정된 변수 riceParam 및 칼라 성분 $cIdx$ 로 수행된다. 예를 들어, 이용되는 2진화 프로세스는 k-차수 Exp-Golomb (EGk) 2진화 프로세스, 및/또는 2015년 6월 19일 - 6월 26일, 폴란드, 바르샤바, Rajan Joshi 등의, "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 4", JCTVC-U1005 에서 설명된 바와 같은, 제한된 k-차수 Exp-Golomb (EGk) 2진화 프로세스일 수도 있다.
- [0189] 본 개시물의 어떤 예들에서, 4 보다 작은 양자화 파라미터 (QP) 는, 값 확장의 문제를 제기할 수도 있다. 이러한 예에서, QP 가 4 와 동일할 때, 명목상, 대응하는 양자화기 단계-사이즈 (qStep) 는 1 과 동일하다. 그러나, QP 가 4 미만일 때, 입력 값을 양자화하는 대신, 양자화 프로세스는 입력 값을, 즉, 그것을 $qStep < 1$ 으로 나눔으로써, 확장하는 것이다. 이것은 변환 계수들의 정확도를 증가시키기 위해 사용될 수도 있다. 그러나, 어떤 예들에서, 1 과 동일한 qStep 을 갖는 이스케이프 코딩된 픽셀들은 이미 임의의 왜곡 또는 에러 없이 입력 값의 전체 정밀도를 표현한다. 따라서, 이러한 예들에서, 추가적인 확장은 단지 메모리를 낭비할 수도 있다.
- [0190] 어떤 QP 값들과 연관되는 값 확장을 해결할 수도 있는 일 예에서는, QP 값들이 4 보다 작은 예에서, QP 값은 이스케이프 픽셀들에 대한 원하는 QP 값들 범위 내로 클리핑될 수 있다. 이러한 예에서, 예를 들어, 이 경우, 양자화 및 역양자화 프로세스들에 사용되는 원하는 QP 값들 범위의 하한은 4 와 동일하게 설정될 수도 있다. 따라서, 이스케이프 픽셀 코딩 또는 변환 스킵을 위한 양자화 또는 역양자화 프로세스들에서 QP 값을 이용하기 전에, QP 값은 먼저 원하는 QP 범위 내로 클리핑될 수도 있다. 예를 들어, 0 과 동일한 QP 값이 사용되면, QP 값은 4 의 값으로 클리핑될 수도 있으며, 그후, 클리핑된 QP 값이 관련된 코딩 또는 디코딩 프로세스들에서 사용된다. 8 이외의 비트심도들에 대해, 4 와 동일한 것으로 간주될 수도 있는 QP 는 $4 - 6 * (\text{bitDepth})$

- 8) 로 규정될 수도 있다는 점에 유의해야 한다. 10의 비트심도에 대해, QP 값은 아래로부터 -8의 값까지 클리핑될 수도 있다. 일 예에서, 더 낮은 QP 값들은 1의 qStep에 명목상 대응하는 QP 값으로 클리핑될 수도 있다.

[0191] 추가적인 예에서, QP 범위의 제한은 또한 변환 스킵, 즉, 변환 프로세싱 유닛 (104)의 변환 단계를 바이패스하는 프로세스에도 적용가능하다.

[0192] 도 6은 본 개시물의 하나 이상의 양태들에 따른, 이스케이프 픽셀 값들이 제한되는 예시적인 인코딩 프로세스를 예시하는 플로우 차트이다. 도 6의 프로세스는 예시의 목적을 위해 비디오 인코더 (20)에 의해 수행되는 것으로 일반적으로 설명되지만, 다양한 다른 프로세서들 중 임의의 하나 이상이 또한 도 6에 나타난 프로세스를 실행할 수도 있다.

[0193] 도 6의 예에서, 비디오 인코더 (20)는 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들에 대한 최대 값을 결정할 수도 있으며, 여기서, 이스케이프 값들은 인코딩될 비디오 데이터의 블록에서의 샘플들을 나타낸다 (602). 예를 들어, 비디오 인코더 (20)는 루마 또는 크로마 성분에 대한 시그널링된 입력 비트심도에 기초하여, 이스케이프 값들에 대한 최대 값을 결정할 수도 있다. 다른 예에서, 비디오 인코더 (20)는 인코더에 의해 지원되는 최대 비트심도에 기초하여 이스케이프 값들에 대한 최대 값을 결정할 수도 있다.

[0194] 비디오 인코더 (20)는 하나 이상의 이스케이프 값들을 최대 값보다 크지 않게 제한할 수도 있다 (604). 예를 들어, 비디오 인코더 (20)는 시그널링된 `Palette_escape_val`에 기초하여 결정된 복원된 이스케이프 값의 값이 결정된 최대 값보다 작거나 또는 동일하도록, 하나 이상의 이스케이프 값들을 규범적으로 제한할 수도 있다. 비디오 인코더 (20)는 인코딩된 비디오 데이터 비트스트림에서 이스케이프 값들을 인코딩할 수도 있다 (606).

[0195] 도 7은 본 개시물의 하나 이상의 양태들에 따른, 이스케이프 픽셀 값들이 제한되는 예시적인 디코딩 프로세스를 예시하는 플로우 차트이다. 도 7의 프로세스는 예시의 목적을 위해 비디오 디코더 (30)에 의해 수행되는 것으로 일반적으로 설명되지만, 다양한 다른 프로세서들 중 임의의 하나 이상이 또한 도 7에 나타난 프로세스를 실행할 수도 있다.

[0196] 도 7의 예에서, 비디오 디코더 (30)는 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들이 최대 값보다 크지 않게 제약된 비트스트림을 수신할 수도 있으며, 여기서, 비트스트림은 디코딩될 비디오 데이터의 블록에서의 샘플을 나타내는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함한다 (702). 예를 들어, 비디오 디코더 (30)는 이스케이프 값들이 루마 또는 크로마 성분에 대한 시그널링된 입력 비트심도에 기초하여 결정된 최대 값보다 크지 않게 제약된 비트스트림을 수신할 수도 있다. 다른 예에서, 비디오 디코더 (30)는 이스케이프 값들이 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 지원되는 최대 비트심도에 기초하여 결정된 최대 값보다 크지 않게 제약된 비트스트림을 수신할 수도 있다.

[0197] 비디오 디코더 (30)는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 이용하여 비디오 데이터의 블록에서의 샘플을 복원할 수도 있다 (704). 예를 들어, 비디오 디코더 (30)는 시그널링된 최대 비트심도에 기초하여 비트스트림으로부터 이스케이프 값들을 결정하고 샘플을 복원할 수도 있다. 다른 예에서, 비디오 디코더 (30)는 인코더 및/또는 디코더에 의해 지원되는 최대 비트심도에 기초하여 비트스트림으로부터 이스케이프 값들을 결정할 수도 있다.

[0198] 도 8은 본 개시물의 하나 이상의 양태들에 따른, 하한이 양자화 파라미터에 적용되는 예시적인 인코딩 프로세스를 예시하는 플로우 차트이다. 도 8의 프로세스는 예시의 목적을 위해 비디오 인코더 (20)에 의해 수행되는 것으로 일반적으로 설명되지만, 다양한 다른 프로세서들 중 임의의 하나 이상이 또한 도 8에 나타난 프로세스를 실행할 수도 있다.

[0199] 도 8의 예에서, 비디오 인코더 (20)는 비디오 데이터에 대한 양자화 파라미터 값을 결정할 수도 있다 (802). 예를 들어, 비디오 인코더 (20)는 비디오 인코더 (20)에 의해 사용되는 세팅에 기초하여 및/또는 인코딩될 콘텐츠에 기초하여, 양자화 파라미터 값을 결정할 수도 있다.

[0200] 비디오 인코더 (20)는 양자화 파라미터 하한을 결정할 수도 있다 (804). 예를 들어, 비디오 인코더 (20)는 1의 qStep에 대응하는 양자화 파라미터를 결정할 수도 있으며, 비디오 인코더 (20)는 그 양자화 파라미터를 하한으로서 선택할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 4의 양자화 파라미터를 양자화

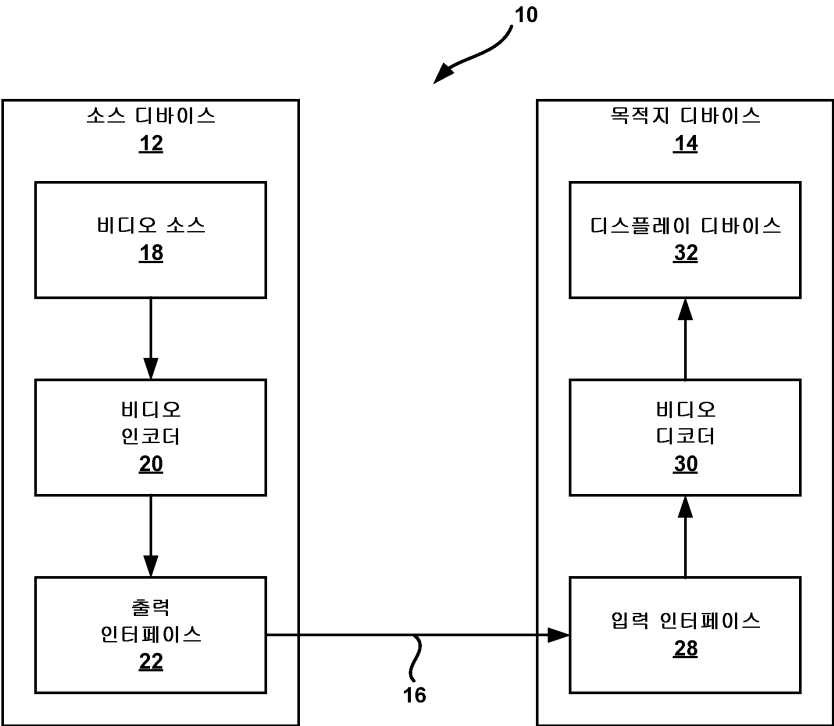
파라미터에 대한 하한으로서 결정할 수도 있다.

- [0201] 비디오 인코더 (20) 는 양자화 파라미터 값이 양자화 파라미터 하한 미만이면 양자화 파라미터 하한에 따라서, 비디오 데이터를 인코딩하기 위한 팔레트 모드에서 사용되는 이스케이프 값들을 양자화할 수도 있으며, 여기서, 이스케이프 값들은 인코딩될 비디오 데이터의 블록에서의 샘플들을 나타낸다 (806). 예를 들어, 하한이 4 로 결정되는 경우, 비디오 인코더 (20) 는 4 미만인 양자화 파라미터가 조우되면, 이스케이프 값들에 대한 또는 변환 스킵에 대한 양자화 프로세스들에서 양자화 파라미터로서 사용하기 전에, 양자화 파라미터를 4 의 값으로 수정하거나 또는 클리핑할 수도 있다. 비디오 인코더 (20) 는 인코딩된 비디오 데이터 비트스트림에서 이스케이프 값들을 인코딩할 수도 있다 (808).
- [0202] 도 9 는 본 개시물의 하나 이상의 양태들에 따른, 하한이 양자화 파라미터에 적용되는 예시적인 디코딩 프로세스를 예시하는 플로우 차트이다. 도 9 의 프로세스는 예시의 목적을 위해 비디오 디코더 (30) 에 의해 수행되는 것으로 일반적으로 설명되지만, 다양한 다른 프로세서들 중 임의의 하나 이상이 또한 도 9 에 나타난 프로세스를 실행할 수도 있다.
- [0203] 도 9 의 예에서, 비디오 디코더 (30) 는 비디오 데이터를 디코딩하기 위한 팔레트 모드에서 사용되는 적어도 하나의 이스케이프 값을 결정하기 위한 정보를 포함하는 비트스트림을 수신할 수도 있으며, 여기서, 비트스트림은 적어도 하나의 이스케이프 값이 양자화 파라미터 하한 이상인 양자화 파라미터에 따라서 양자화되게 보장하는 제한을 준수하도록 제한된다 (902). 예를 들어, 비디오 디코더 (30) 는 이스케이프 값들이 1 의 qStep 에 대응하는 것보다 작지 않은 양자화 파라미터로 양자화되도록 제약된 비트스트림을 수신할 수도 있다. 일부 예들에서, 예를 들어, 8 과 동일한 비트심도 값들에 대해, 양자화 파라미터 하한은 4 일 수도 있다. 따라서, 이러한 예에서, 비디오 디코더 (30) 는 이스케이프 값들이 4 보다 작지 않은 양자화 파라미터로 양자화되도록 제약된 비트스트림을 수신할 수도 있다.
- [0204] 비디오 디코더 (30) 는 제한에 기초하여 비트스트림으로부터 적어도 하나의 이스케이프 값을 결정할 수도 있다 (904). 예를 들어, 비디오 디코더 (30) 는 이스케이프 값들이 4 보다 작지 않은 양자화 파라미터로 양자화되도록 제약되어 있는 비트스트림으로부터 이스케이프 값들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 적어도 하나의 이스케이프 값을 이용하여 비디오 데이터의 샘플을 복원할 수도 있다 (906).
- [0205] 임의의 플로우차트들 또는 흐름도들을 포함하여, 본원에서 설명되는 프로세스들, 장치들, 및 다른 예들 또는 예시들에 있어서, 본원에서 설명되는 기법들 중 임의의 기법에 포함된 어떤 동작들, 행위들, 단계들, 또는 이벤트들은 상이한 시퀀스로 수행될 수 있거나, 함께 추가되거나, 병합되거나, 또는 제외될 수도 있다 (예컨대, 모든 설명된 행위들 또는 이벤트들이 본 기법들의 실시예에 필요하지는 않다). 더욱이, 어떤 예들에서, 동작들, 행위들, 단계들, 또는 이벤트들은 순차적으로 보다는, 동시에, 예컨대, 멀티-쓰레드된 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해서 수행될 수도 있다. 또한, 어떤 동작들, 행위들, 단계들, 또는 이벤트들은 설명된 자동적으로 수행되는 것으로 구체적으로 식별되지 않더라도 자동적으로 수행될 수도 있다. 또한, 자동적으로 수행되는 것으로 설명된 어떤 동작들, 행위들, 단계들, 또는 이벤트들은 대안적으로 자동적으로 수행되지 않을 수도 있지만, 오히려, 일부 예들에서, 이러한 동작들, 행위들, 단계들, 또는 이벤트들은 입력 또는 다른 이벤트에 응답하여 수행될 수도 있다.
- [0206] 그 예에 따라서, 본원에서 설명되는 기법들 중 임의의 기법의 어떤 행위들 또는 이벤트들이 상이한 시퀀스로 수행될 수 있으며, 추가되거나, 병합되거나, 또는 모두 제외시킬 수도 있는 (예컨대, 모든 설명되는 행위들 또는 이벤트들이 기법들의 실시예에 필수적인 것은 아닌) 것으로 인식되어야 한다. 더욱이, 어떤 예들에서, 행위들 또는 이벤트들은 순차적으로 보다는, 동시에, 예컨대, 멀티-쓰레드된 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해서 수행될 수도 있다. 게다가, 본 개시물의 어떤 양태들이 명료성의 목적들을 위해 단일 모듈 또는 유닛에 의해 수행되는 것으로 설명되지만, 본 개시물의 기법들은 비디오 코더와 연관되는 유닛들 또는 모듈들의 결합에 의해 수행될 수도 있는 것으로 이해되어야 한다.
- [0207] 본 개시물의 어떤 양태들은 예시의 목적을 위해 개발중인 HEVC 표준에 대해서 설명되었다. 그러나, 본 개시물에서 설명하는 기법들은 아직 개발되지 않은 다른 표준 또는 독점 비디오 코딩 프로세스들을 포함한, 다른 비디오 코딩 프로세스들에 유용할 수도 있다.
- [0208] 위에서 설명된 기법들은 비디오 인코더 (20) (도 1 및 도 2) 및/또는 비디오 디코더 (30) (도 1 및 도 3) 에 의해 수행될 수도 있으며, 이 비디오 인코더와 디코더는 비디오 코더로서 일반적으로 지칭될 수도 있다. 이와 유사하게, 비디오 코딩은 적용가능한 경우, 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.

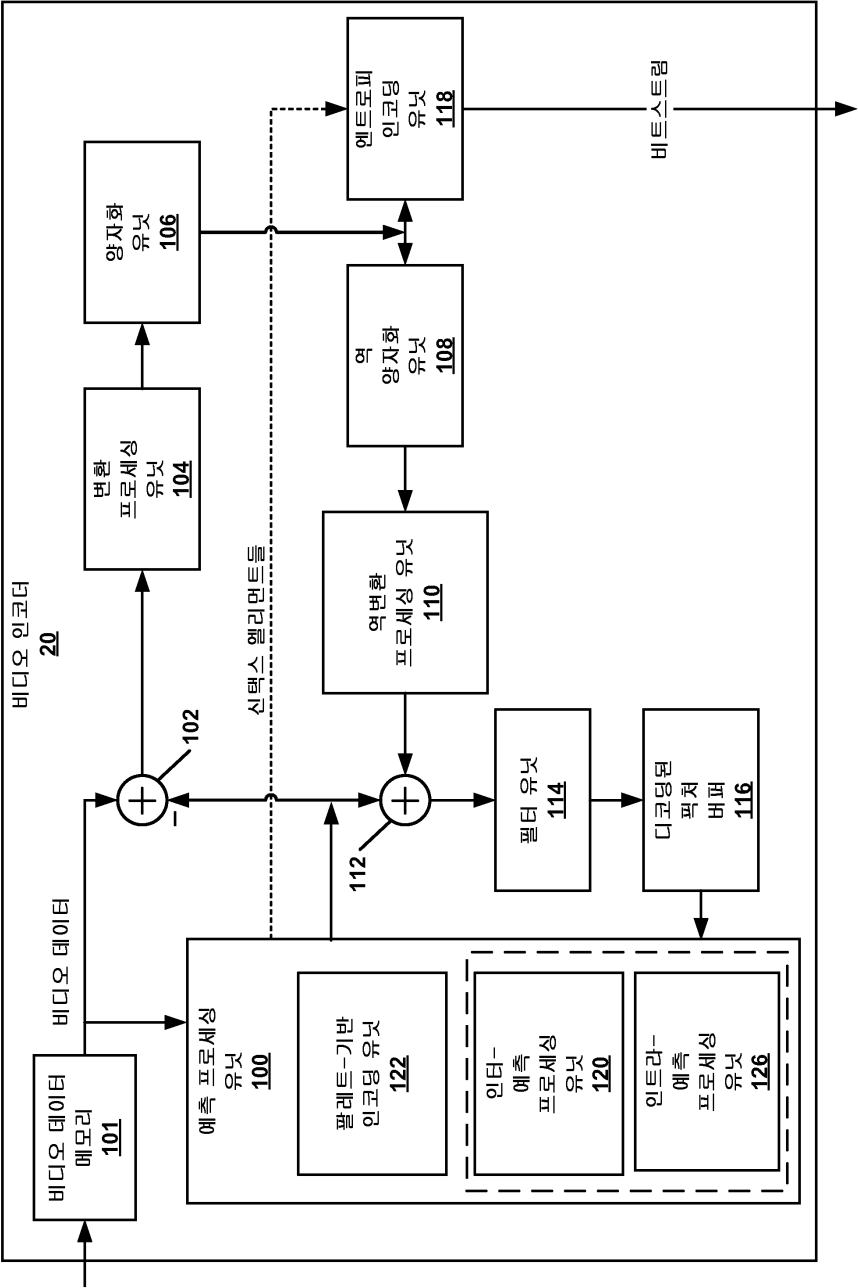
- [0209] 본 기법들의 다양한 양태들의 특징의 조합들이 위에서 설명되지만, 이들 조합들은 단지 본 개시물에서 설명하는 기법들의 예들을 예시하기 위해 제공된다. 따라서, 본 개시물의 기법들은 이들 예시적인 조합들에 한정되지 않아야 하며 본 개시물에서 설명하는 기법들의 다양한 양태들의 임의의 상상가능한 조합을 포괄할 수도 있다.
- [0210] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 또는 컴퓨터-판독가능 매체를 통해서 송신될 수도 있으며, 하드웨어-기반의 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터-판독가능 매체는 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터-판독가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라서 한 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방법으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비일시성 유형의 컴퓨터-판독가능 저장 매체, 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명하는 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독가능 매체를 포함할 수도 있다.
- [0211] 일 예로서, 이에 한정하지 않고, 이런 컴퓨터-판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광디스크 스토리지, 자기디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터-판독가능 매체로 적절히 지칭된다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 무선 기술들, 예컨대 적외선, 라디오, 및 마이크로파를 이용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들 예컨대 적외선, 라디오, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체 및 데이터 저장 매체는 접속부들, 반송파들, 신호들, 또는 다른 일시성 매체를 포함하지 않고, 그 대신, 비-일시성 유형의 저장 매체로 송신되는 것으로 해석되어야 한다. 디스크 (disk) 및 디스크 (disc) 는, 본원에서 사용할 때, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루-레이 디스크를 포함하며, 디스크들 (disks) 은 데이터를 자기적으로 보통 재생하지만, 디스크들 (discs) 은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 결합들이 또한 컴퓨터-판독가능 매체들의 범위 내에 포함되어야 한다.
- [0212] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서" 는, 본원에서 사용할 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능은 전용 하드웨어 및/또는 인코딩 및 디코딩을 위해 구성되는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.
- [0213] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트) 를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 다양한 컴포넌트들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 대신, 위에서 설명한 바와 같이, 다양한 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

도면

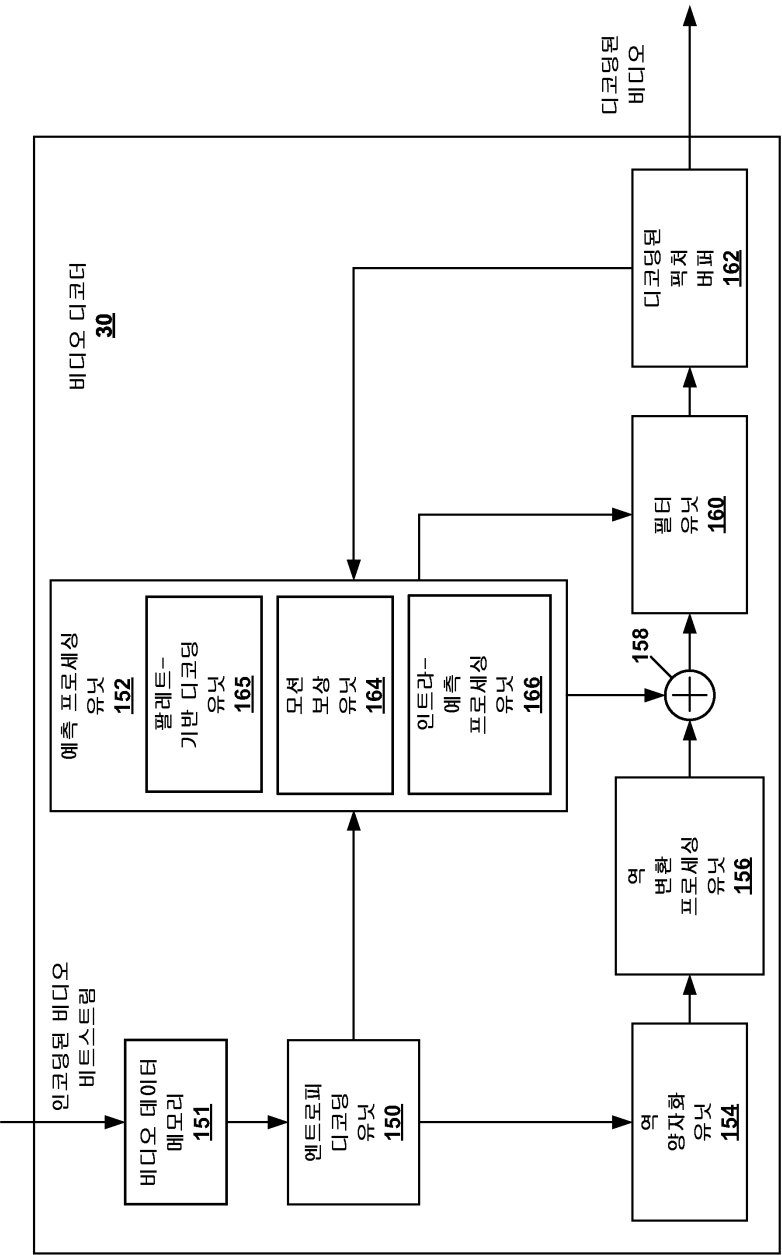
도면1



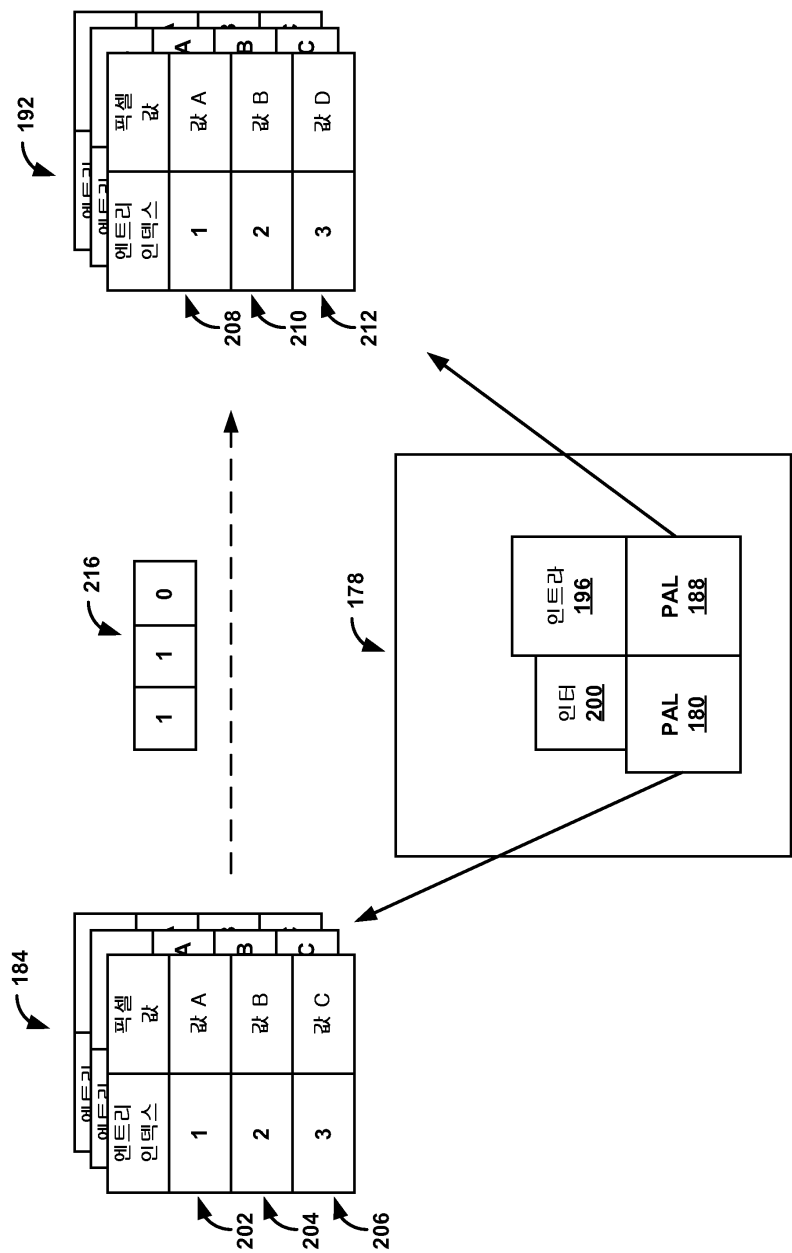
도면2



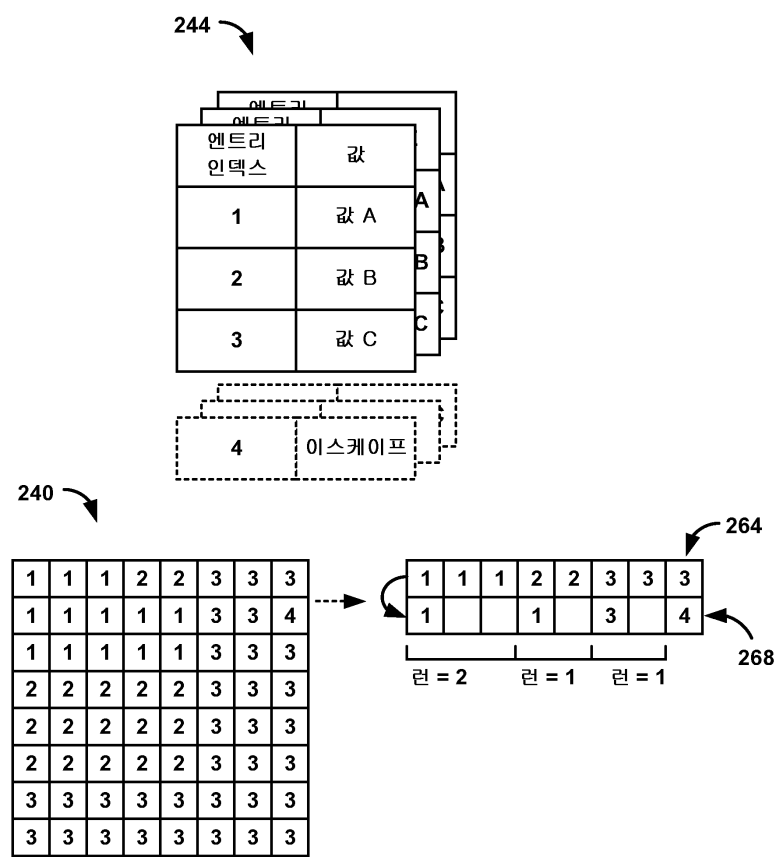
도면3



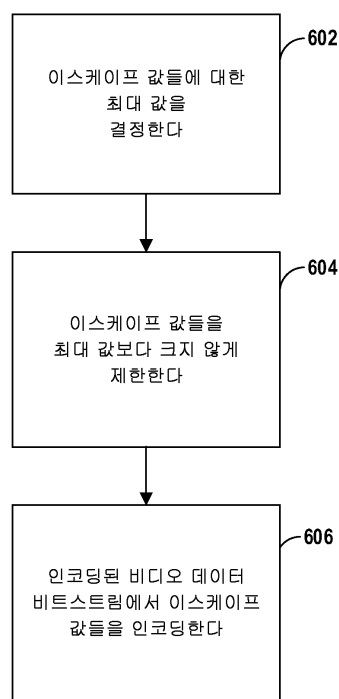
도면4



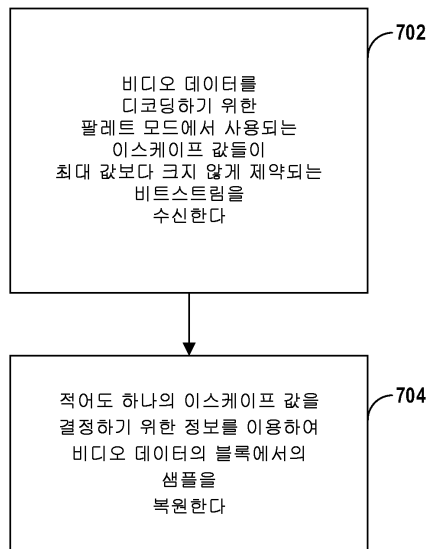
도면5



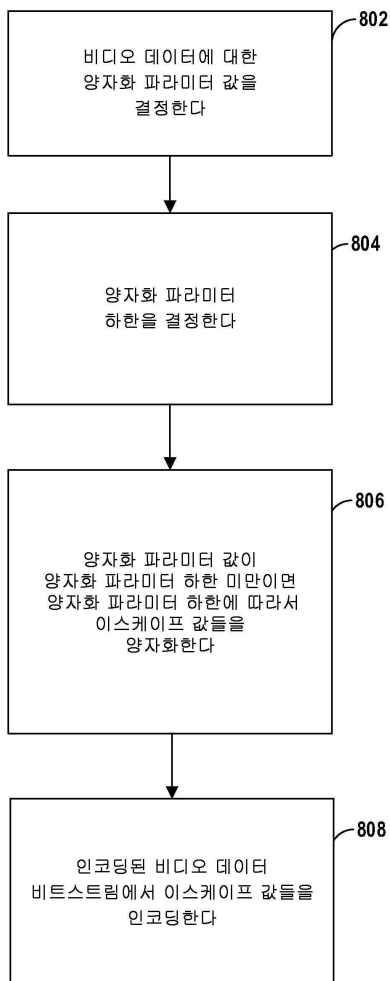
도면6



도면7



도면8



도면9

