(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년04월26일
(11) 등록번호 10-2661461
(24) 등록일자 2024년04월23일

(51) 국제특허분류(Int. Cl.)
H04N 19/124 (2014.01) *H04N 19/11* (2014.01)
H04N 19/132 (2014.01) *H04N 19/157* (2014.01)
H04N 19/176 (2014.01) *H04N 19/186* (2014.01)
H04N 19/60 (2014.01) *H04N 19/70* (2014.01)
(52) CPC특허분류
H04N 19/124 (2015.01)
H04N 19/11 (2015.01)
(21) 출원번호 10-2022-7005848
(22) 출원일자(국제) 2020년08월26일
심사청구일자 2022년02월22일
(85) 번역문제출일자 2022년02월22일
(65) 공개번호 10-2022-0038724
(43) 공개일자 2022년03월29일
(86) 국제출원번호 PCT/KR2020/011383
(87) 국제공개번호 WO 2021/040400
국제공개일자 2021년03월04일
(30) 우선권주장
62/891,951 2019년08월26일 미국(US)
(56) 선행기술조사문헌
KR1020160031495 A*
KR1020160135262 A*
KR1020180056687 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
엘지전자 주식회사
서울특별시 영등포구 여의대로 128 (여의도동)
(72) 발명자
자오지에
서울시 서초구 양재대로11길 19 LG전자 특허센터
김승환
서울시 서초구 양재대로11길 19 LG전자 특허센터
파루리시탈
서울시 서초구 양재대로11길 19 LG전자 특허센터
(74) 대리인
인비전 특허법인

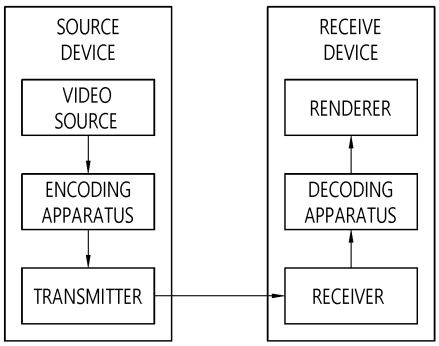
전체 청구항 수 : 총 4 항 심사관 : 김성권

(54) 발명의 명칭 **팔레트 모드 기반 영상 또는 비디오 코딩**

(57) 요 약

본 문서의 개시에 따르면, 팔레트 모드에서의 양자화된 이스케이프 값에 대한 스케일링 과정에서 사용되는 양자화 파라미터를 도출함에 있어서, 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출될 수 있다. 이를 통하여 비디오/영상 코딩을 위하여 시그널링되어야 하는 데이터량을 줄이고, 팔레트 모드에서 이스케이프 코딩을 효율적으로 수행할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
H04N 19/132 (2015.01)
H04N 19/157 (2015.01)
H04N 19/176 (2015.01)
H04N 19/186 (2015.01)
H04N 19/60 (2015.01)
H04N 19/70 (2015.01)

명 세 서

청구범위

청구항 1

디코딩 장치에 의하여 수행되는 영상 디코딩 방법에 있어서,

비트스트림으로부터 팔레트 모드에서의 양자화된 이스케이프 값 정보를 포함하는 영상 정보를 획득하는 단계;

상기 양자화된 이스케이프 값 정보를 기반으로 현재 블록에 대한 이스케이프 값을 도출하는 단계; 및

상기 이스케이프 값을 기반으로 복원 샘플들을 생성하는 단계를 포함하며,

상기 이스케이프 값은 상기 양자화된 이스케이프 값 정보 및 양자화 파라미터를 기반으로 도출되고,

상기 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출되며, 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보는 상기 영상 정보에 포함되는 SPS(Sequence Parameter Set)로부터 획득되는 신택스 엘리먼트(syntax element)이며,

상기 양자화 파라미터는 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보로부터 도출되는 최소 양자화 파라미터 값보다 크거나 같은 값을 갖는 것을 특징으로 하는 영상 디코딩 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

인코딩 장치에 의하여 수행되는 영상 인코딩 방법에 있어서,

현재 블록에 대한 팔레트 모드에서의 이스케이프 값을 도출하는 단계;

상기 이스케이프 값을 기반으로 양자화된 이스케이프 값을 도출하는 단계; 및

상기 양자화된 이스케이프 값에 대한 정보를 포함하는 영상 정보를 인코딩하는 단계를 포함하며,

상기 양자화된 이스케이프 값은 상기 현재 블록 내의 상기 이스케이프 값에 대한 양자화 파라미터를 기반으로 도출되고,

상기 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출되며, 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보는 SPS(Sequence Parameter Set)에 포함되는 신택스 엘리먼트(syntax

element)이며, 상기 SPS는 상기 영상 정보에 포함되며,

상기 양자화 파라미터는 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보로부터 도출되는 최소 양자화 파라미터 값보다 크거나 같은 값을 갖는 것을 특징으로 하는 영상 인코딩 방법.

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

특정 방법에 의해 생성된 비트스트림을 저장하는 컴퓨터 판독 가능한 저장 매체에 있어서, 상기 특정 방법은,

현재 블록에 대한 팔레트 모드에서의 이스케이프 값을 도출하는 단계, 상기 이스케이프 값을 기반으로 양자화된 이스케이프 값을 도출하는 단계 및 상기 양자화된 이스케이프 값에 대한 정보를 포함하는 영상 정보를 인코딩하여 상기 비트스트림을 생성하는 단계를 포함하며,

상기 양자화된 이스케이프 값은 상기 현재 블록 내의 상기 이스케이프 값에 대한 양자화 파라미터를 기반으로 도출되고,

상기 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출되며, 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보는 SPS(Sequence Parameter Set)에 포함되는 신택스 엘리먼트(syntax element)이며, 상기 SPS는 상기 영상 정보에 포함되며,

상기 양자화 파라미터는 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보로부터 도출되는 최소 양자화 파라미터 값보다 크거나 같은 값을 갖는 것을 특징으로 하는 컴퓨터 판독 가능한 저장 매체.

청구항 16

영상에 대한 데이터를 전송하는 방법에 있어서,

비트스트림을 획득하되, 상기 비트스트림은 현재 블록에 대한 팔레트 모드에서의 이스케이프 값을 도출하는 단계, 상기 이스케이프 값을 기반으로 양자화된 이스케이프 값을 도출하는 단계, 상기 비트스트림을 생성하기 위하여 상기 양자화된 이스케이프 값에 대한 정보를 포함하는 영상 정보를 인코딩하는 단계를 기반으로 생성되는 단계; 및

상기 비트스트림을 포함하는 상기 데이터를 전송하는 단계를 포함하되,

상기 양자화된 이스케이프 값은 상기 현재 블록 내의 상기 이스케이프 값에 대한 양자화 파라미터를 기반으로 도출되고,

상기 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출되며, 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보는 SPS(Sequence Parameter Set)에 포함되는 신택스 엘리먼트(syntax

element)이며, 상기 SPS는 상기 영상 정보에 포함되며,

상기 양자화 파라미터는 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보로부터 도출되는 최소 양자화 파라미터 값보다 크거나 같은 값을 갖는 것을 특징으로 하는 전송 방법.

발명의 설명

기 술 분 야

[0001] 본 기술은 비디오 또는 영상 코딩에 관한 것이며, 예를 들어 팔레트 모드 기반 코딩 기술에 관한 것이다.

배 경 기 술

[0002] 최근 4K 또는 8K 이상의 UHD(Ultra High Definition) 영상/비디오와 같은 고해상도, 고품질의 영상/비디오에 대한 수요가 다양한 분야에서 증가하고 있다. 영상/비디오 데이터가 고해상도, 고품질이 될수록 기존의 영상/비디오 데이터에 비해 상대적으로 전송되는 정보량 또는 비트량이 증가하기 때문에 기존의 유무선 광대역 회선과 같은 매체를 이용하여 영상 데이터를 전송하거나 기존의 저장 매체를 이용해 영상/비디오 데이터를 저장하는 경우, 전송 비용과 저장 비용이 증가된다.

[0003] 또한, 최근 VR(Virtual Reality), AR(Artificial Realtiy) 컨텐츠나 홀로그램 등의 실감 미디어(Immersive Media)에 대한 관심 및 수요가 증가하고 있으며, 게임 영상과 같이 현실 영상과 다른 영상 특성을 갖는 영상/비디오에 대한 방송이 증가하고 있다.

[0004] 이에 따라, 상기와 같은 다양한 특성을 갖는 고해상도 고품질의 영상/비디오의 정보를 효과적으로 압축하여 전송하거나 저장하고, 재생하기 위해 고효율의 영상/비디오 압축 기술이 요구된다.

[0005] 또한, 상당한 양의 텍스트 및 그래픽을 포함하는 컴퓨터 생성 비디오(computer generated video)와 같은 스크린 컨텐츠에 대한 코딩 효율성을 향상시키기 위해 팔레트 모드 코딩 기술에 관한 논의가 있다. 이러한 기술을 효율적으로 적용하기 위하여 관련된 정보를 코딩하고 시그널링하는 방안이 필요하다.

발명의 내용

해결하려는 과제

[0006] 본 문서의 기술적 과제는 비디오/영상 코딩 효율을 높이는 방법 및 장치를 제공함에 있다.

[0007] 본 문서의 다른 기술적 과제는 팔레트 모드 코딩에서 효율을 높이는 방법 및 장치를 제공함에 있다.

[0008] 본 문서의 또 다른 기술적 과제는 팔레트 모드 코딩에서 사용되는 다양한 정보를 효율적으로 구성하고 시그널링하는 방법 및 장치를 제공함에 있다.

[0009] 본 문서의 또 다른 기술적 과제는 팔레트 모드에서 이스케이프 코딩을 효율적으로 적용하는 방법 및 장치를 제공함에 있다.

과제의 해결 수단

[0010] 본 문서의 일 실시예에 따르면, 팔레트 모드에서의 양자화된 이스케이프 값에 대한 스케일링 과정에서 사용되는 양자화 파라미터를 도출함에 있어서, 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출될 수 있다. 상기 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 값보다 크거나 같은 값을 가질 수 있다.

[0011] 본 문서의 일 실시예에 따르면, 팔레트 모드에서의 양자화된 이스케이프 값의 범위를 비트심도에 기반하여 제한할 수 있다. 예를 들어, 루마 성분에 대한 양자화된 이스케이프 값 정보의 범위는 0에서 $(1 << BitDepth_Y) - 1$ 사이의 값을 갖고, 크로마 성분에 대한 상기 양자화된 이스케이프 값 정보의 범위는 0에서 $(1 << BitDepth_C) - 1$ 사이의 값을 가질 수 있다.

[0012] 본 문서의 일 실시예에 따르면, 팔레트 테이블의 최대 인덱스에 관한 팔레트 사이즈 정보를 정의하고 상기 팔레트 사이즈 정보를 SPS(sequence parameter set)를 통하여 시그널링할 수 있다.

[0013] 본 문서의 일 실시예에 따르면, 디코딩 장치에 의하여 수행되는 비디오/영상 디코딩 방법을 제공한다. 상기 비

디오/영상 디코딩 방법은 본 문서의 실시예들에서 개시된 방법을 포함할 수 있다.

[0014] 본 문서의 일 실시예에 따르면, 비디오/영상 디코딩을 수행하는 디코딩 장치를 제공한다. 상기 디코딩 장치는 본 문서의 실시예들에서 개시된 방법을 수행할 수 있다.

[0015] 본 문서의 일 실시예에 따르면, 인코딩 장치에 의하여 수행되는 비디오/영상 인코딩 방법을 제공한다. 상기 비디오/영상 인코딩 방법은 본 문서의 실시예들에서 개시된 방법을 포함할 수 있다.

[0016] 본 문서의 일 실시예에 따르면, 비디오/영상 인코딩을 수행하는 인코딩 장치를 제공한다. 상기 인코딩 장치는 본 문서의 실시예들에서 개시된 방법을 수행할 수 있다.

[0017] 본 문서의 일 실시예에 따르면, 본 문서의 실시예들 중 적어도 하나에 개시된 비디오/영상 인코딩 방법에 따라 생성된 인코딩된 비디오/영상 정보가 저장된 컴퓨터 판독 가능한 디지털 저장 매체를 제공한다.

[0018] 본 문서의 일 실시예에 따르면, 디코딩 장치에 의하여 본 문서의 실시예들 중 적어도 하나에 개시된 비디오/영상 디코딩 방법을 수행하도록 야기하는 인코딩된 정보 또는 인코딩된 비디오/영상 정보가 저장된 컴퓨터 판독 가능한 디지털 저장 매체를 제공한다.

발명의 효과

[0019] 본 문서는 다양한 효과를 가질 수 있다. 예를 들어, 본 문서의 일 실시예에 따르면 전반적인 영상/비디오 압축 효율을 높일 수 있다. 또한, 본 문서의 일 실시예에 따르면 팔레트 모드 코딩에서 효율을 향상시킬 수 있다. 또한, 본 문서의 일 실시예에 따르면 팔레트 모드 코딩에서 사용되는 다양한 정보를 효율적으로 구성하고 시그널링할 수 있다. 또한, 본 문서의 일 실시예에 따르면 팔레트 모드에서 이스케이프 코딩을 효율적으로 적용함으로써, 이스케이프 샘플에 대한 정확도 및 코딩 효율을 향상시킬 수 있다.

[0020] 본 문서의 구체적인 실시예를 통해 얻을 수 있는 효과는 이상에서 나열된 효과로 제한되지 않는다. 예를 들어, 관련된 기술분야의 통상의 지식을 자긴 자(a person having ordinary skill in the related art)가 본 문서로부터 이해하거나 유도할 수 있는 다양한 기술적 효과가 존재할 수 있다. 이에 따라 본 문서의 구체적인 효과는 본 문서에 명시적으로 기재된 것에 제한되지 않고, 본 문서의 기술적 특징으로부터 이해되거나 유도될 수 있는 다양한 효과를 포함할 수 있다.

도면의 간단한 설명

[0021] 도 1은 본 문서의 실시예들에 적용될 수 있는 비디오/영상 코딩 시스템의 예를 개략적으로 나타낸다.

도 2는 본 문서의 실시예들이 적용될 수 있는 비디오/영상 인코딩 장치의 구성을 개략적으로 설명하는 도면이다.

도 3은 본 문서의 실시예들이 적용될 수 있는 비디오/영상 디코딩 장치의 구성을 개략적으로 설명하는 도면이다.

도 4는 본 문서의 실시예(들)이 적용 가능한 개략적인 비디오/영상 인코딩 절차의 예를 나타낸다.

도 5는 본 문서의 실시예(들)이 적용 가능한 개략적인 비디오/영상 디코딩 절차의 예를 나타낸다.

도 6은 팔레트 코딩의 기본적인 구조를 설명하기 위한 일예를 나타낸다.

도 7은 팔레트 인덱스 맵을 코딩하기 위해 사용되는 수평 및 수직 트래버스 스캔 방법을 설명하기 위한 일예를 나타낸다.

도 8은 팔레트 모드 기반 코딩 방법의 일예를 설명하기 위한 도면이다.

도 9는 본 문서의 실시예(들)에 따른 비디오/영상 인코딩 방법의 일 예를 개략적으로 나타낸다.

도 10은 본 문서의 실시예(들)에 따른 비디오/영상 디코딩 방법의 일 예를 개략적으로 나타낸다.

도 11은 본 문서에서 개시된 실시예들이 적용될 수 있는 컨텐츠 스트리밍 시스템의 예를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0022] 본 문서는 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고

상세하게 설명하고자 한다. 그러나, 이는 본 문서를 특정 실시예에 한정하려고 하는 것이 아니다. 본 문서에서 상용하는 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 문서의 기술적 사상을 한정하려는 의도로 사용되는 것은 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 문서에서 "포함하다" 또는 "가지다" 등의 용어는 문서상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부품 도는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0023] 한편, 본 문서에서 설명되는 도면상의 각 구성들은 서로 다른 특징적인 기능들에 관한 설명의 편의를 위해 독립적으로 도시된 것으로서, 각 구성들이 서로 별개의 하드웨어나 별개의 소프트웨어로 구현된다는 것을 의미하지는 않는다. 예컨대, 각 구성 중 두 개 이상의 구성이 합쳐져 하나의 구성을 이룰 수도 있고, 하나의 구성이 복수의 구성으로 나뉘어질 수도 있다. 각 구성이 통합 및/또는 분리된 실시예도 본 문서의 본질에서 벗어나지 않는 한 본 문서의 권리범위에 포함된다.

[0024] 본 문서에서 "A 또는 B(A or B)"는 "오직 A", "오직 B" 또는 "A와 B 모두"를 의미할 수 있다. 달리 표현하면, 본 문서에서 "A 또는 B(A or B)"는 "A 및/또는 B(A and/or B)"으로 해석될 수 있다. 예를 들어, 본 문서에서 "A, B 또는 C(A, B or C)"는 "오직 A", "오직 B", "오직 C", 또는 "A, B 및 C의 임의의 모든 조합(any combination of A, B and C)"를 의미할 수 있다.

[0025] 본 문서에서 사용되는 슬래쉬(/)나 쉼표(comma)는 "및/또는(and/or)"을 의미할 수 있다. 예를 들어, "A/B"는 "A 및/또는 B"를 의미할 수 있다. 이에 따라 "A/B"는 "오직 A", "오직 B", 또는 "A와 B 모두"를 의미할 수 있다. 예를 들어, "A, B, C"는 "A, B 또는 C"를 의미할 수 있다.

[0026] 본 문서에서 "적어도 하나의 A 및 B(at least one of A and B)"는, "오직 A", "오직 B" 또는 "A와 B 모두"를 의미할 수 있다. 또한, 본 문서에서 "적어도 하나의 A 또는 B(at least one of A or B)"나 "적어도 하나의 A 및/또는 B(at least one of A and/or B)"라는 표현은 "적어도 하나의 A 및 B(at least one of A and B)"와 동일하게 해석될 수 있다.

[0027] 또한, 본 문서에서 "적어도 하나의 A, B 및 C(at least one of A, B and C)"는, "오직 A", "오직 B", "오직 C", 또는 "A, B 및 C의 임의의 모든 조합(any combination of A, B and C)"를 의미할 수 있다. 또한, "적어도 하나의 A, B 또는 C(at least one of A, B or C)"나 "적어도 하나의 A, B 및/또는 C(at least one of A, B and/or C)"는 "적어도 하나의 A, B 및 C(at least one of A, B and C)"를 의미할 수 있다.

[0028] 또한, 본 문서에서 사용되는 괄호는 "예를 들어(for example)"를 의미할 수 있다. 구체적으로, "예측(인트라 예측)"로 표시된 경우, "예측"의 일례로 "인트라 예측"이 제안된 것일 수 있다. 달리 표현하면 본 문서의 "예측"은 "인트라 예측"으로 제한(limit)되지 않고, "인트라 예측"이 "예측"의 일례로 제안될 것일 수 있다. 또한, "예측(즉, 인트라 예측)"으로 표시된 경우에도, "예측"의 일례로 "인트라 예측"이 제안된 것일 수 있다.

[0029] 본 문서는 비디오/영상 코딩에 관한 것이다. 예를 들어 본 문서에서 개시된 방법/실시예는 VVC (versatile video coding) 표준에 개시되는 방법에 적용될 수 있다. 또한, 본 문서에서 개시된 방법/실시예는 EVC (essential video coding) 표준, AV1 (AOMedia Video 1) 표준, AVS2 (2nd generation of audio video coding standard) 또는 차세대 비디오/영상 코딩 표준(ex. H.267 or H.268 등)에 개시되는 방법에 적용될 수 있다.

[0030] 본 문서에서는 비디오/영상 코딩에 관한 다양한 실시예들을 제시하며, 다른 언급이 없는 한 상기 실시예들은 서로 조합되어 수행될 수도 있다.

[0031] 본 문서에서 비디오(video)는 시간의 흐름에 따른 일련의 영상(image)들의 집합을 의미할 수 있다. 픽처(picture)는 일반적으로 특정 시간대의 하나의 영상을 나타내는 단위를 의미하며, 슬라이스(slice)/타일(tile)은 코딩에 있어서 픽처의 일부를 구성하는 단위이다. 슬라이스/타일은 하나 이상의 CTU(coding tree unit)을 포함할 수 있다. 하나의 픽처는 하나 이상의 슬라이스/타일로 구성될 수 있다. 타일은 픽너 내 특정 타일 열 및 특정 타일 열 이내의 CTU들의 사각 영역이다(A tile is a rectangular region of CTUs within a particular tile column and a particular tile row in a picture). 상기 타일 열은 CTU들의 사각 영역이고, 상기 사각 영역은 상기 픽처의 높이와 동일한 높이를 갖고, 너비는 픽처 파라미터 세트 내의 신택스 요소들에 의하여 명시될 수 있다(The tile column is a rectangular region of CTUs having a height equal to the height of the picture and a width specified by syntax elements in the picture parameter set). 상기 타일 행은 CTU들의 사각 영역이고, 상기 사각 영역은 픽처 파라미터 세트 내의 신택스 요소들에 의하여 명시되는 너비를 갖고, 높

이는 상기 픽처의 높이와 동일할 수 있다(The tile row is a rectangular region of CTUs having a height specified by syntax elements in the picture parameter set and a width equal to the width of the picture). 타일 스캔은 픽처를 파티셔닝하는 CTU들의 특정 순차적 오더링을 나타낼 수 있고, 상기 CTU들은 타일 내 CTU 래스터 스캔으로 연속적으로 정렬될 수 있고, 픽처 내 타일들은 상기 픽처의 상기 타일들의 래스터 스캔으로 연속적으로 정렬될 수 있다(A tile scan is a specific sequential ordering of CTUs partitioning a picture in which the CTUs are ordered consecutively in CTU raster scan in a tile whereas tiles in a picture are ordered consecutively in a raster scan of the tiles of the picture). 슬라이스는 단일 NAL 유닛에 배타적으로 담겨질 수 있는, 정수개의 완전한 타일들 또는 픽처의 타일 내의 정수개의 연속적인 완전한 CTU 행들을 포함할 수 있다(A slice includes an integer number of complete tiles or an integer number of consecutive complete CTU rows within a tile of a picture that may be exclusively contained in a single NAL unit).

[0032] 한편, 하나의 픽처는 둘 이상의 서브픽처로 구분될 수 있다. 서브픽처는 픽처 내 하나 이상의 슬라이스들의 사각 리전일 수 있다(an rectangular region of one or more slices within a picture).

[0033] 픽셀(pixel) 또는 펠(pel)은 하나의 픽처(또는 영상)을 구성하는 최소의 단위를 의미할 수 있다. 또한, 픽셀에 대응하는 용어로서 '샘플(sample)'이 사용될 수 있다. 샘플은 일반적으로 픽셀 또는 픽셀의 값을 나타낼 수 있으며, 루마(luma) 성분의 픽셀/픽셀값만을 나타낼 수도 있고, 크로마(chroma) 성분의 픽셀/픽셀 값만을 나타낼 수도 있다. 또는 샘플은 공간 도메인에서의 픽셀값을 의미할 수도 있고, 이러한 픽셀값이 주파수 도메인으로 변환되면 주파수 도메인에서의 변환 계수를 의미할 수도 있다.

[0034] 유닛(unit)은 영상 처리의 기본 단위를 나타낼 수 있다. 유닛은 픽처의 특정 영역 및 해당 영역에 관련된 정보 중 적어도 하나를 포함할 수 있다. 하나의 유닛은 하나의 루마 블록 및 두개의 크로마(ex. cb, cr) 블록을 포함할 수 있다. 유닛은 경우에 따라서 블록(block) 또는 영역(area) 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, MxN 블록은 M개의 열과 N개의 행으로 이루어진 샘플들(또는 샘플 어레이) 또는 변환 계수(transform coefficient)들의 집합(또는 어레이)을 포함할 수 있다.

[0035] 또한, 본 문서에서 양자화/역양자화 및/또는 변환/역변환 중 적어도 하나는 생략될 수 있다. 양자화/역양자화가 생략되는 경우, 양자화된 변환 계수는 변환 계수라고 불릴 수 있다. 변환/역변환이 생략되는 경우, 변환 계수는 계수 또는 레지듀얼 계수라고 불릴 수도 있고, 또는 표현의 통일성을 위하여 변환 계수라고 여전히 불릴 수도 있다.

[0036] 본 문서에서 양자화된 변환 계수 및 변환 계수는 각각 변환 계수 및 스케일링된(scaled) 변환 계수라고 지칭될 수 있다. 이 경우 레지듀얼 정보는 변환 계수(들)에 관한 정보를 포함할 수 있고, 변환 계수(들)에 관한 정보는 레지듀얼 코딩 신택스를 통하여 시그널링될 수 있다. 레지듀얼 정보(또는 변환 계수(들)에 관한 정보)를 기반으로 변환 계수들이 도출될 수 있고, 변환 계수들에 대한 역변환(스케일링)을 통하여 스케일링된 변환 계수들이 도출될 수 있다. 스케일링된 변환 계수들에 대한 역변환(변환)을 기반으로 레지듀얼 샘플들이 도출될 수 있다. 이는 본 문서의 다른 부분에서도 마찬가지로 적용/표현될 수 있다.

[0037] 본 문서에서 하나의 도면 내에서 개별적으로 설명되는 기술적 특징은, 개별적으로 구현될 수도 있고, 동시에 구현될 수도 있다.

[0038] 이하, 첨부한 도면들을 참조하여, 본 문서의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 이하, 도면상의 동일한 구성 요소에 대해서는 동일한 참조 부호를 사용하고 동일한 구성 요소에 대해서 중복된 설명은 생략될 수 있다.

[0039] 도 1은 본 문서의 실시예들에 적용될 수 있는 비디오/영상 코딩 시스템의 예를 개략적으로 나타낸다.

[0040] 도 1을 참조하면, 비디오/영상 코딩 시스템은 제1 장치(소스 디바이스) 및 제2 장치(수신 디바이스)를 포함할 수 있다. 소스 디바이스는 인코딩된 비디오(video)/영상(image) 정보 또는 데이터를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여 수신 디바이스로 전달할 수 있다.

[0041] 상기 소스 디바이스는 비디오 소스, 인코딩 장치, 전송부를 포함할 수 있다. 상기 수신 디바이스는 수신부, 디코딩 장치 및 렌더러를 포함할 수 있다. 상기 인코딩 장치는 비디오/영상 인코딩 장치라고 불릴 수 있고, 상기 디코딩 장치는 비디오/영상 디코딩 장치라고 불릴 수 있다. 송신기는 인코딩 장치에 포함될 수 있다. 수신기는 디코딩 장치에 포함될 수 있다. 렌더러는 디스플레이부를 포함할 수도 있고, 디스플레이부는 별개의 디바이스

또는 외부 컴포넌트로 구성될 수도 있다.

[0042] 비디오 소스는 비디오/영상의 캡쳐, 합성 또는 생성 과정 등을 통하여 비디오/영상을 획득할 수 있다. 비디오 소스는 비디오/영상 캡쳐 디바이스 및/또는 비디오/영상 생성 디바이스를 포함할 수 있다. 비디오/영상 캡쳐 디바이스는 예를 들어, 하나 이상의 카메라, 이전에 캡쳐된 비디오/영상을 포함하는 비디오/영상 아카이브 등을 포함할 수 있다. 비디오/영상 생성 디바이스는 예를 들어 컴퓨터, 타블렛 및 스마트폰 등을 포함할 수 있으며 (전자적으로) 비디오/영상을 생성할 수 있다. 예를 들어, 컴퓨터 등을 통하여 가상의 비디오/영상이 생성될 수 있으며, 이 경우 관련 데이터가 생성되는 과정으로 비디오/영상 캡쳐 과정이 갈음될 수 있다.

[0043] 인코딩 장치는 입력 비디오/영상을 인코딩할 수 있다. 인코딩 장치는 압축 및 코딩 효율을 위하여 예측, 변환, 양자화 등 일련의 절차를 수행할 수 있다. 인코딩된 데이터(인코딩된 비디오/영상 정보)는 비트스트림(bitstream) 형태로 출력될 수 있다.

[0044] 전송부는 비트스트림 형태로 출력된 인코딩된 비디오/영상 정보 또는 데이터를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여 수신 디바이스의 수신부로 전달할 수 있다. 디지털 저장 매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장 매체를 포함할 수 있다. 전송부는 미리 정해진 파일 포멧을 통하여 미디어 파일을 생성하기 위한 엘리먼트를 포함할 수 있고, 방송/통신 네트워크를 통한 전송을 위한 엘리먼트를 포함할 수 있다. 수신부는 상기 비트스트림을 수신/추출하여 디코딩 장치로 전달할 수 있다.

[0045] 디코딩 장치는 인코딩 장치의 동작에 대응하는 역양자화, 역변환, 예측 등 일련의 절차를 수행하여 비디오/영상을 디코딩할 수 있다.

[0046] 렌더러는 디코딩된 비디오/영상을 렌더링할 수 있다. 렌더링된 비디오/영상은 디스플레이부를 통하여 디스플레이될 수 있다.

[0047] 도 2는 본 문서의 실시예들이 적용될 수 있는 비디오/영상 인코딩 장치의 구성을 개략적으로 설명하는 도면이다. 이하 인코딩 장치라 함은 영상 인코딩 장치 및/또는 비디오 인코딩 장치를 포함할 수 있다.

[0048] 도 2를 참조하면, 인코딩 장치(200)는 영상 분할부(image partitioner, 210), 예측부(predictor, 220), 레지듀얼 처리부(residual processor, 230), 엔트로피 인코딩부(entropy encoder, 240), 가산부(adder, 250), 필터링부(filter, 260) 및 메모리(memory, 270)를 포함하여 구성될 수 있다. 예측부(220)는 인터 예측부(221) 및 인트라 예측부(222)를 포함할 수 있다. 레지듀얼 처리부(230)는 변환부(transformer, 232), 양자화부(quantizer 233), 역양자화부(dequantizer 234), 역변환부(inverse transformer, 235)를 포함할 수 있다. 레지듀얼 처리부(230)은 감산부(subtractor, 231)를 더 포함할 수 있다. 가산부(250)는 복원부(reconstructor) 또는 복원 블록 생성부(recontructged block generator)로 불릴 수 있다. 상술한 영상 분할부(210), 예측부(220), 레지듀얼 처리부(230), 엔트로피 인코딩부(240), 가산부(250) 및 필터링부(260)는 실시예에 따라 하나 이상의 하드웨어 컴포넌트(예를 들어 인코더 칩셋 또는 프로세서)에 의하여 구성될 수 있다. 또한 메모리(270)는 DPB(decoded picture buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구성될 수도 있다. 상기 하드웨어 컴포넌트는 메모리(270)을 내/외부 컴포넌트로 더 포함할 수도 있다.

[0049] 영상 분할부(210)는 인코딩 장치(200)에 입력된 입력 영상(또는, 픽쳐, 프레임)를 하나 이상의 처리 유닛(processing unit)으로 분할할 수 있다. 일 예로, 상기 처리 유닛은 코딩 유닛(coding unit, CU)이라고 불릴 수 있다. 이 경우 코딩 유닛은 코딩 트리 유닛(coding tree unit, CTU) 또는 최대 코딩 유닛(largest coding unit, LCU)으로부터 QTBTTT (Quad-tree binary-tree ternary-tree) 구조에 따라 재귀적으로(recursively) 분할될 수 있다. 예를 들어, 하나의 코딩 유닛은 쿼드 트리 구조, 바이너리 트리 구조, 및/또는 터너리 구조를 기반으로 하위(deeper) 뎁스의 복수의 코딩 유닛들로 분할될 수 있다. 이 경우 예를 들어 쿼드 트리 구조가 먼저 적용되고 바이너리 트리 구조 및/또는 터너리 구조가 나중에 적용될 수 있다. 또는 바이너리 트리 구조가 먼저 적용될 수도 있다. 더 이상 분할되지 않는 최종 코딩 유닛을 기반으로 본 문서에 따른 코딩 절차가 수행될 수 있다. 이 경우 영상 특성에 따른 코딩 효율 등을 기반으로, 최대 코딩 유닛이 바로 최종 코딩 유닛으로 사용될 수 있고, 또는 필요에 따라 코딩 유닛은 재귀적으로(recursively) 보다 하위 뎁스의 코딩 유닛들로 분할되어 최적의 사이즈의 코딩 유닛이 최종 코딩 유닛으로 사용될 수 있다. 여기서 코딩 절차라 함은 후술하는 예측, 변환, 및 복원 등의 절차를 포함할 수 있다. 다른 예로, 상기 처리 유닛은 예측 유닛(PU: Prediction Unit) 또는 변환 유닛(TU: Transform Unit)을 더 포함할 수 있다. 이 경우 상기 예측 유닛 및 상기 변환 유닛은 각각 상술한 최종 코딩 유닛으로부터 분할 또는 파티셔닝될 수 있다. 상기 예측 유닛은 샘플 예측의 단위일 수 있고, 상기 변환 유닛은 변환 계수를 유도하는 단위 및/또는 변환 계수로부터 레지듀얼 신호(residual signal)를 유도하는 단

위일 수 있다.

[0050] 유닛은 경우에 따라서 블록(block) 또는 영역(area) 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, MxN 블록은 M개의 열과 N개의 행으로 이루어진 샘플들 또는 변환 계수(transform coefficient)들의 집합을 나타낼 수 있다. 샘플은 일반적으로 픽셀 또는 픽셀의 값을 나타낼 수 있으며, 휘도(luma) 성분의 픽셀/픽셀값만을 나타낼 수도 있고, 채도(chroma) 성분의 픽셀/픽셀 값만을 나타낼 수도 있다. 샘플은 하나의 픽처(또는 영상)을 픽셀(pixel) 또는 펠(pel)에 대응하는 용어로서 사용될 수 있다.

[0051] 인코딩 장치(200)는 입력 영상 신호(원본 블록, 원본 샘플 어레이)에서 인터 예측부(221) 또는 인트라 예측부(222)로부터 출력된 예측 신호(예측된 블록, 예측 샘플 어레이)를 감산하여 레지듀얼 신호(residual signal, 잔여 블록, 잔여 샘플 어레이)를 생성할 수 있고, 생성된 레지듀얼 신호는 변환부(232)로 전송된다. 이 경우 도시된 바와 같이 인코더(200) 내에서 입력 영상 신호(원본 블록, 원본 샘플 어레이)에서 예측 신호(예측 블록, 예측 샘플 어레이)를 감산하는 유닛은 감산부(231)라고 불릴 수 있다. 예측부는 처리 대상 블록(이하, 현재 블록이라 함)에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부는 현재 블록 또는 CU 단위로 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있다. 예측부는 각 예측모드에 대한 설명에서 후술하는 바와 같이 예측 모드 정보 등 예측에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(240)로 전달할 수 있다. 예측에 관한 정보는 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.

[0052] 인트라 예측부(222)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 상기 참조되는 샘플들은 예측 모드에 따라 상기 현재 블록의 주변(neighbor)에 위치할 수 있고, 또는 떨어져서 위치할 수도 있다. 인트라 예측에서 예측 모드들은 복수의 비방향성 모드와 복수의 방향성 모드를 포함할 수 있다. 비방향성 모드는 예를 들어 DC 모드 및 플래너 모드(Planar 모드)를 포함할 수 있다. 방향성 모드는 예측 방향의 세밀한 정도에 따라 예를 들어 33개의 방향성 예측 모드 또는 65개의 방향성 예측 모드를 포함할 수 있다. 다만, 이는 예시로서 설정에 따라 그 이상 또는 그 이하의 개수의 방향성 예측 모드들이 사용될 수 있다. 인트라 예측부(222)는 주변 블록에 적용된 예측 모드를 이용하여, 현재 블록에 적용되는 예측 모드를 결정할 수도 있다.

[0053] 인터 예측부(221)는 참조 픽처 상에서 움직임 벡터에 의해 특정되는 참조 블록(참조 샘플 어레이)을 기반으로, 현재 블록에 대한 예측된 블록을 유도할 수 있다. 이때, 인터 예측 모드에서 전송되는 움직임 정보의 양을 줄이기 위해 주변 블록과 현재 블록 간의 움직임 정보의 상관성에 기초하여 움직임 정보를 블록, 서브블록 또는 샘플 단위로 예측할 수 있다. 상기 움직임 정보는 움직임 벡터 및 참조 픽처 인덱스를 포함할 수 있다. 상기 움직임 정보는 인터 예측 방향(L0 예측, L1 예측, Bi 예측 등) 정보를 더 포함할 수 있다. 인터 예측의 경우에, 주변 블록은 현재 픽처 내에 존재하는 공간적 주변 블록(spatial neighboring block)과 참조 픽처에 존재하는 시간적 주변 블록(temporal neighboring block)을 포함할 수 있다. 상기 참조 블록을 포함하는 참조 픽처와 상기 시간적 주변 블록을 포함하는 참조 픽처는 동일할 수도 있고, 다를 수도 있다. 상기 시간적 주변 블록은 동일 위치 참조 블록(collocated reference block), 동일 위치 CU(colCU) 등의 이름으로 불릴 수 있으며, 상기 시간적 주변 블록을 포함하는 참조 픽처는 동일 위치 픽처(collocated picture, colPic)라고 불릴 수도 있다. 예를 들어, 인터 예측부(221)는 주변 블록들을 기반으로 움직임 정보 후보 리스트를 구성하고, 상기 현재 블록의 움직임 벡터 및/또는 참조 픽처 인덱스를 도출하기 위하여 어떤 후보가 사용되는지를 지시하는 정보를 생성할 수 있다. 다양한 예측 모드를 기반으로 인터 예측이 수행될 수 있으며, 예를 들어 스킵 모드와 머지 모드의 경우에, 인터 예측부(221)는 주변 블록의 움직임 정보를 현재 블록의 움직임 정보로 이용할 수 있다. 스킵 모드의 경우, 머지 모드와 달리 레지듀얼 신호가 전송되지 않을 수 있다. 움직임 정보 예측(motion vector prediction, MVP) 모드의 경우, 주변 블록의 움직임 벡터를 움직임 벡터 예측자(motion vector predictor)로 이용하고, 움직임 벡터 차분(motion vector difference)을 시그널링함으로써 현재 블록의 움직임 벡터를 지시할 수 있다.

[0054] 예측부(220)는 후술하는 다양한 예측 방법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부는 하나의 블록에 대한 예측을 위하여 인트라 예측 또는 인터 예측을 적용할 수 있을 뿐 아니라, 인트라 예측과 인터 예측을 동시에 적용할 수 있다. 이는 combined inter and intra prediction (CIIP)라고 불릴 수 있다. 또한, 예측부는 블록에 대한 예측을 위하여 인트라 블록 카피(intra block copy, IBC) 예측 모드에 기반할 수도 있고 또는 팔레트 모드(palette mode)에 기반할 수도 있다. 상기 IBC 예측 모드 또는 팔레트 모드는 예를 들어 SCC(screen content coding) 등과 같이 게임 등의 컨텐츠 영상/동영상 코딩을 위하여 사용될 수 있다. IBC는 기본적으로 현재 픽처 내에서 예측을 수행하나 현재 픽처 내에서 참조 블록을 도출하는 점에서 인터 예측과 유사하게 수행될 수 있다. 즉, IBC는 본 문서에서 설명되는 인터 예측 기법들 중 적어도 하나를 이용할 수 있다. 팔

레트 모드는 인트라 코딩 또는 인트라 예측의 일 예로 볼 수 있다. 팔레트 모드가 적용되는 경우 팔레트 테이블 및 팔레트 인덱스에 관한 정보를 기반으로 픽처 내 샘플 값을 시그널링할 수 있다.

[0055] 상기 예측부 (인터 예측부(221) 및/또는 상기 인트라 예측부(222) 포함)를 통해 생성된 예측 신호는 복원 신호를 생성하기 위해 이용되거나 레지듀얼 신호를 생성하기 위해 이용될 수 있다. 변환부(232)는 레지듀얼 신호에 변환 기법을 적용하여 변환 계수들(transform coefficients)를 생성할 수 있다. 예를 들어, 변환 기법은 DCT(Discrete Cosine Transform), DST(Discrete Sine Transform), KLT(Karhunen-Loeve Transform), GBT(Graph-Based Transform), 또는 CNT(Conditionally Non-linear Transform) 중 적어도 하나를 포함할 수 있다. 여기서, GBT는 픽셀 간의 관계 정보를 그래프로 표현한다고 할 때 이 그래프로부터 얻어진 변환을 의미한다. CNT는 이전에 복원된 모든 픽셀(all previously reconstructed pixel)를 이용하여 예측 신호를 생성하고 그에 기초하여 획득되는 변환을 의미한다. 또한, 변환 과정은 정사각형의 동일한 크기를 갖는 픽셀 블록에 적용될 수도 있고, 정사각형이 아닌 가변 크기의 블록에도 적용될 수 있다.

[0056] 양자화부(233)는 변환 계수들을 양자화하여 엔트로피 인코딩부(240)로 전송되고, 엔트로피 인코딩부(240)는 양자화된 신호(양자화된 변환 계수들에 관한 정보)를 인코딩하여 비트스트림으로 출력할 수 있다. 상기 양자화된 변환 계수들에 관한 정보는 레지듀얼 정보라고 불릴 수 있다. 양자화부(233)는 계수 스캔 순서(scan order)를 기반으로 블록 형태의 양자화된 변환 계수들을 1차원 벡터 형태로 재정렬할 수 있고, 상기 1차원 벡터 형태의 양자화된 변환 계수들을 기반으로 상기 양자화된 변환 계수들에 관한 정보를 생성할 수도 있다. 엔트로피 인코딩부(240)는 예를 들어 지수 골롬(exponential Golomb), CAVLC(context-adaptive variable length coding), CABAC(context-adaptive binary arithmetic coding) 등과 같은 다양한 인코딩 방법을 수행할 수 있다. 엔트로피 인코딩부(240)는 양자화된 변환 계수들 외 비디오/이미지 복원에 필요한 정보들(예컨대 신택스 요소들(syntax elements)의 값 등)을 함께 또는 별도로 인코딩할 수도 있다. 인코딩된 정보(ex. 인코딩된 비디오/영상 정보)는 비트스트림 형태로 NAL(network abstraction layer) 유닛 단위로 전송 또는 저장될 수 있다. 상기 비디오/영상 정보는 어댑테이션 파라미터 세트(APS), 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS) 또는 비디오 파라미터 세트(VPS) 등 다양한 파라미터 세트에 관한 정보를 더 포함할 수 있다. 또한 상기 비디오/영상 정보는 일반 제한 정보(general constraint information)을 더 포함할 수 있다. 본 문서에서 인코딩 장치에서 디코딩 장치로 전달/시그널링되는 정보 및/또는 신택스 요소들은 비디오/영상 정보에 포함될 수 있다. 상기 비디오/영상 정보는 상술한 인코딩 절차를 통하여 인코딩되어 상기 비트스트림에 포함될 수 있다. 상기 비트스트림은 네트워크를 통하여 전송될 수 있고, 또는 디지털 저장매체에 저장될 수 있다. 여기서 네트워크는 방송망 및/또는 통신망 등을 포함할 수 있고, 디지털 저장매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장매체를 포함할 수 있다. 엔트로피 인코딩부(240)로부터 출력된 신호는 전송하는 전송부(미도시) 및/또는 저장하는 저장부(미도시)가 인코딩 장치(200)의 내/외부 엘리먼트로서 구성될 수 있고, 또는 전송부는 엔트로피 인코딩부(240)에 포함될 수도 있다.

[0057] 양자화부(233)로부터 출력된 양자화된 변환 계수들은 예측 신호를 생성하기 위해 이용될 수 있다. 예를 들어, 양자화된 변환 계수들에 역양자화부(234) 및 역변환부(235)를 통해 역양자화 및 역변환을 적용함으로써 레지듀얼 신호(레지듀얼 블록 or 레지듀얼 샘플들)를 복원할 수 있다. 가산부(155)는 복원된 레지듀얼 신호를 인터 예측부(221) 또는 인트라 예측부(222)로부터 출력된 예측 신호에 더함으로써 복원(reconstructed) 신호(복원 픽처, 복원 블록, 복원 샘플 어레이)가 생성될 수 있다. 스킵 모드가 적용된 경우와 같이 처리 대상 블록에 대한 레지듀얼이 없는 경우, 예측된 블록이 복원 블록으로 사용될 수 있다. 가산부(250)는 복원부 또는 복원 블록 생성부라고 불릴 수 있다. 생성된 복원 신호는 현재 픽처 내 다음 처리 대상 블록의 인트라 예측을 위하여 사용될 수 있고, 후술하는 바와 같이 필터링을 거쳐서 다음 픽처의 인터 예측을 위하여 사용될 수도 있다.

[0058] 한편 픽처 인코딩 및/또는 복원 과정에서 LMCS (luma mapping with chroma scaling)가 적용될 수도 있다.

[0059] 필터링부(260)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(260)은 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정된 복원 픽처를 메모리(270), 구체적으로 메모리(270)의 DPB에 저장할 수 있다. 상기 다양한 필터링 방법은 예를 들어, 디블록킹 필터링, 샘플 적응적 오프셋(sample adaptive offset), 적응적 루프 필터(adaptive loop filter), 양방향 필터(bilateral filter) 등을 포함할 수 있다. 필터링부(260)은 각 필터링 방법에 대한 설명에서 후술하는 바와 같이 필터링에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(240)로 전달할 수 있다. 필터링 관한 정보는 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.

[0060] 메모리(270)에 전송된 수정된 복원 픽처는 인터 예측부(221)에서 참조 픽처로 사용될 수 있다. 인코딩 장치는

이를 통하여 인터 예측이 적용되는 경우, 인코딩 장치(100)와 디코딩 장치에서의 예측 미스매치를 피할 수 있고, 부호화 효율도 향상시킬 수 있다.

[0061] 메모리(270) DPB는 수정된 복원 픽처를 인터 예측부(221)에서의 참조 픽처로 사용하기 위해 저장할 수 있다. 메모리(270)는 현재 픽처 내 움직임 정보가 도출된(또는 인코딩된) 블록의 움직임 정보 및/또는 이미 복원된 픽처 내 블록들의 움직임 정보를 저장할 수 있다. 상기 저장된 움직임 정보는 공간적 주변 블록의 움직임 정보 또는 시간적 주변 블록의 움직임 정보로 활용하기 위하여 인터 예측부(221)에 전달할 수 있다. 메모리(270)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(222)에 전달할 수 있다.

[0062] 도 3은 본 문서의 실시예들이 적용될 수 있는 비디오/영상 디코딩 장치의 구성을 개략적으로 설명하는 도면이다. 이하 디코딩 장치라 함은 영상 디코딩 장치 및/또는 비디오 디코딩 장치를 포함할 수 있다.

[0063] 도 3을 참조하면, 디코딩 장치(300)는 엔트로피 디코딩부(entropy decoder, 310), 레지듀얼 처리부(residual processor, 320), 예측부(predictor, 330), 가산부(adder, 340), 필터링부(filter, 350) 및 메모리(memoery, 360)를 포함하여 구성될 수 있다. 예측부(330)는 인터 예측부(331) 및 인트라 예측부(332)를 포함할 수 있다. 레지듀얼 처리부(320)는 역양자화부(dequantizer, 321) 및 역변환부(inverse transformer, 321)를 포함할 수 있다. 상술한 엔트로피 디코딩부(310), 레지듀얼 처리부(320), 예측부(330), 가산부(340) 및 필터링부(350)는 실시예에 따라 하나의 하드웨어 컴포넌트(예를 들어 디코더 칩셋 또는 프로세서)에 의하여 구성될 수 있다. 또한 메모리(360)는 DPB(decoded picture buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구성될 수도 있다. 상기 하드웨어 컴포넌트는 메모리(360)을 내/외부 컴포넌트로 더 포함할 수도 있다.

[0064] 비디오/영상 정보를 포함하는 비트스트림이 입력되면, 디코딩 장치(300)는 도 2의 인코딩 장치에서 비디오/영상 정보가 처리된 프로세스에 대응하여 영상을 복원할 수 있다. 예를 들어, 디코딩 장치(300)는 상기 비트스트림으로부터 획득한 블록 분할 관련 정보를 기반으로 유닛들/블록들을 도출할 수 있다. 디코딩 장치(300)는 인코딩 장치에서 적용된 처리 유닛을 이용하여 디코딩을 수행할 수 있다. 따라서 디코딩의 처리 유닛은 예를 들어 코딩 유닛일 수 있고, 코딩 유닛은 코딩 트리 유닛 또는 최대 코딩 유닛으로부터 쿼드 트리 구조, 바이너리 트리 구조 및/또는 터너리 트리 구조를 따라서 분할될 수 있다. 코딩 유닛으로부터 하나 이상의 변환 유닛이 도출될 수 있다. 그리고, 디코딩 장치(300)를 통해 디코딩 및 출력된 복원 영상 신호는 재생 장치를 통해 재생될 수 있다.

[0065] 디코딩 장치(300)는 도 2의 인코딩 장치로부터 출력된 신호를 비트스트림 형태로 수신할 수 있고, 수신된 신호는 엔트로피 디코딩부(310)를 통해 디코딩될 수 있다. 예를 들어, 엔트로피 디코딩부(310)는 상기 비트스트림을 파싱하여 영상 복원(또는 픽처 복원)에 필요한 정보(ex. 비디오/영상 정보)를 도출할 수 있다. 상기 비디오/영상 정보는 어댑테이션 파라미터 세트(APS), 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS) 또는 비디오 파라미터 세트(VPS) 등 다양한 파라미터 세트에 관한 정보를 더 포함할 수 있다. 또한 상기 비디오/영상 정보는 일반 제한 정보(general constraint information)을 더 포함할 수 있다. 디코딩 장치는 상기 파라미터 세트에 관한 정보 및/또는 상기 일반 제한 정보를 더 기반으로 픽처를 디코딩할 수 있다. 본 문서에서 후술되는 시그널링/수신되는 정보 및/또는 신택스 요소들은 상기 디코딩 절차를 통하여 디코딩되어 상기 비트스트림으로부터 획득될 수 있다. 예컨대, 엔트로피 디코딩부(310)는 지수 골롬 부호화, CAVLC 또는 CABAC 등의 코딩 방법을 기초로 비트스트림 내 정보를 디코딩하고, 영상 복원에 필요한 신택스 엘리먼트의 값, 레지듀얼에 관한 변환 계수의 양자화된 값 들을 출력할 수 있다. 보다 상세하게, CABAC 엔트로피 디코딩 방법은, 비트스트림에서 각 구문 요소에 해당하는 빈을 수신하고, 디코딩 대상 구문 요소 정보와 주변 및 디코딩 대상 블록의 디코딩 정보 혹은 이전 단계에서 디코딩된 심볼/빈의 정보를 이용하여 문맥(context) 모델을 결정하고, 결정된 문맥 모델에 따라 빈(bin)의 발생 확률을 예측하여 빈의 산술 디코딩(arithmetic decoding)를 수행하여 각 구문 요소의 값에 해당하는 심볼을 생성할 수 있다. 이때, CABAC 엔트로피 디코딩 방법은 문맥 모델 결정 후 다음 심볼/빈의 문맥 모델을 위해 디코딩된 심볼/빈의 정보를 이용하여 문맥 모델을 업데이트할 수 있다. 엔트로피 디코딩부(310)에서 디코딩된 정보 중 예측에 관한 정보는 예측부(인터 예측부(332) 및 인트라 예측부(331))로 제공되고, 엔트로피 디코딩부(310)에서 엔트로피 디코딩이 수행된 레지듀얼 값, 즉 양자화된 변환 계수들 및 관련 파라미터 정보는 레지듀얼 처리부(320)로 입력될 수 있다. 레지듀얼 처리부(320)는 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플들, 레지듀얼 샘플 어레이)를 도출할 수 있다. 또한, 엔트로피 디코딩부(310)에서 디코딩된 정보 중 필터링에 관한 정보는 필터링부(350)으로 제공될 수 있다. 한편, 인코딩 장치로부터 출력된 신호를 수신하는 수신부(미도시)가 디코딩 장치(300)의 내/외부 엘리먼트로서 더 구성될 수 있고, 또는 수신부는 엔트로피 디코딩부(310)의 구성요소일 수도 있다. 한편, 본 문서에 따른 디코딩 장치는 비디오/영상/픽처 디코딩 장치라고 불릴 수 있고, 상기 디코딩 장치는 정보 디코더(비디오/영상/픽처 정보 디코더) 및 샘플 디코더(비디오/영상/픽처 샘플 디코더)로 구분할 수도 있다. 상기 정보 디코더는 상기 엔트로피 디코딩부(310)를 포함할 수 있고, 상기 샘

플 디코더는 상기 역양자화부(321), 역변환부(322), 가산부(340), 필터링부(350), 메모리(360), 인터 예측부(332) 및 인트라 예측부(331) 중 적어도 하나를 포함할 수 있다.

[0066] 역양자화부(321)에서는 양자화된 변환 계수들을 역양자화하여 변환 계수들을 출력할 수 있다. 역양자화부(321)는 양자화된 변환 계수들을 2차원의 블록 형태로 재정렬할 수 있다. 이 경우 상기 재정렬은 인코딩 장치에서 수행된 계수 스캔 순서를 기반하여 재정렬을 수행할 수 있다. 역양자화부(321)는 양자화 파라미터(예를 들어 양자화 스텝 사이즈 정보)를 이용하여 양자화된 변환 계수들에 대한 역양자화를 수행하고, 변환 계수들(transform coefficient)를 획득할 수 있다.

[0067] 역변환부(322)에서는 변환 계수들를 역변환하여 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플 어레이)를 획득하게 된다.

[0068] 예측부는 현재 블록에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부는 엔트로피 디코딩부(310)로부터 출력된 상기 예측에 관한 정보를 기반으로 상기 현재 블록에 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있고, 구체적인 인트라/인터 예측 모드를 결정할 수 있다.

[0069] 예측부(320)는 후술하는 다양한 예측 방법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부는 하나의 블록에 대한 예측을 위하여 인트라 예측 또는 인터 예측을 적용할 수 있을 뿐 아니라, 인트라 예측과 인터 예측을 동시에 적용할 수 있다. 이는 combined inter and intra prediction (CIIP)라고 불릴 수 있다. 또한, 예측부는 블록에 대한 예측을 위하여 인트라 블록 카피(intra block copy, IBC) 예측 모드에 기반할 수도 있고 또는 팔레트 모드(palette mode)에 기반할 수도 있다. 상기 IBC 예측 모드 또는 팔레트 모드는 예를 들어 SCC(screen content coding) 등과 같이 게임 등의 컨텐츠 영상/동영상 코딩을 위하여 사용될 수 있다. IBC는 기본적으로 현재 픽처 내에서 예측을 수행하나 현재 픽처 내에서 참조 블록을 도출하는 점에서 인터 예측과 유사하게 수행될 수 있다. 즉, IBC는 본 문서에서 설명되는 인터 예측 기법들 중 적어도 하나를 이용할 수 있다. 팔레트 모드는 인트라 코딩 또는 인트라 예측의 일 예로 볼 수 있다. 팔레트 모드가 적용되는 경우 팔레트 테이블 및 팔레트 인덱스에 관한 정보가 상기 비디오/영상 정보에 포함되어 시그널링될 수 있다.

[0070] 인트라 예측부(331)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 상기 참조되는 샘플들은 예측 모드에 따라 상기 현재 블록의 주변(neighbor)에 위치할 수 있고, 또는 떨어져서 위치할 수도 있다. 인트라 예측에서 예측 모드들은 복수의 비방향성 모드와 복수의 방향성 모드를 포함할 수 있다. 인트라 예측부(331)는 주변 블록에 적용된 예측 모드를 이용하여, 현재 블록에 적용되는 예측 모드를 결정할 수도 있다.

[0071] 인터 예측부(332)는 참조 픽처 상에서 움직임 벡터에 의해 특정되는 참조 블록(참조 샘플 어레이)을 기반으로, 현재 블록에 대한 예측된 블록을 유도할 수 있다. 이때, 인터 예측 모드에서 전송되는 움직임 정보의 양을 줄이기 위해 주변 블록과 현재 블록 간의 움직임 정보의 상관성에 기초하여 움직임 정보를 블록, 서브블록 또는 샘플 단위로 예측할 수 있다. 상기 움직임 정보는 움직임 벡터 및 참조 픽처 인덱스를 포함할 수 있다. 상기 움직임 정보는 인터 예측 방향(L0 예측, L1 예측, Bi 예측 등) 정보를 더 포함할 수 있다. 인터 예측의 경우에, 주변 블록은 현재 픽처 내에 존재하는 공간적 주변 블록(spatial neighboring block)과 참조 픽처에 존재하는 시간적 주변 블록(temporal neighboring block)을 포함할 수 있다. 예를 들어, 인터 예측부(332)는 주변 블록들을 기반으로 움직임 정보 후보 리스트를 구성하고, 수신한 후보 선택 정보를 기반으로 상기 현재 블록의 움직임 벡터 및/또는 참조 픽처 인덱스를 도출할 수 있다. 다양한 예측 모드를 기반으로 인터 예측이 수행될 수 있으며, 상기 예측에 관한 정보는 상기 현재 블록에 대한 인터 예측의 모드를 지시하는 정보를 포함할 수 있다.

[0072] 가산부(340)는 획득된 레지듀얼 신호를 예측부(인터 예측부(332) 및/또는 인트라 예측부(331) 포함)로부터 출력된 예측 신호(예측된 블록, 예측 샘플 어레이)에 더함으로써 복원 신호(복원 픽처, 복원 블록, 복원 샘플 어레이)를 생성할 수 있다. 스킵 모드가 적용된 경우와 같이 처리 대상 블록에 대한 레지듀얼이 없는 경우, 예측된 블록이 복원 블록으로 사용될 수 있다.

[0073] 가산부(340)는 복원부 또는 복원 블록 생성부라고 불릴 수 있다. 생성된 복원 신호는 현재 픽처 내 다음 처리 대상 블록의 인트라 예측을 위하여 사용될 수 있고, 후술하는 바와 같이 필터링을 거쳐서 출력될 수도 있고 또는 다음 픽처의 인터 예측을 위하여 사용될 수도 있다.

[0074] 한편, 픽처 디코딩 과정에서 LMCS (luma mapping with chroma scaling)가 적용될 수도 있다.

[0075] 필터링부(350)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(350)는 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정

된 복원 픽처를 메모리(360), 구체적으로 메모리(360)의 DPB에 전송할 수 있다. 상기 다양한 필터링 방법은 예를 들어, 디블록킹 필터링, 샘플 적응적 오프셋(sample adaptive offset), 적응적 루프 필터(adaptive loop filter), 양방향 필터(bilateral filter) 등을 포함할 수 있다.

[0076] 메모리(360)의 DPB에 저장된 (수정된) 복원 픽처는 인터 예측부(332)에서 참조 픽쳐로 사용될 수 있다. 메모리(360)는 현재 픽처 내 움직임 정보가 도출된(또는 디코딩된) 블록의 움직임 정보 및/또는 이미 복원된 픽처 내 블록들의 움직임 정보를 저장할 수 있다. 상기 저장된 움직임 정보는 공간적 주변 블록의 움직임 정보 또는 시간적 주변 블록의 움직임 정보로 활용하기 위하여 인터 예측부(332)에 전달할 수 있다. 메모리(360)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(331)에 전달할 수 있다.

[0077] 본 문서에서, 인코딩 장치(200)의 필터링부(260), 인터 예측부(221) 및 인트라 예측부(222)에서 설명된 실시예들은 각각 디코딩 장치(300)의 필터링부(350), 인터 예측부(332) 및 인트라 예측부(331)에도 동일 또는 대응되도록 적용될 수 있다.

[0078] 상술한 바와 같이 비디오 코딩을 수행함에 있어 압축 효율을 높이기 위하여 예측을 수행한다. 이를 통하여 코딩 대상 블록인 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록을 생성할 수 있다. 여기서 상기 예측된 블록은 공간 도메인(또는 픽셀 도메인)에서의 예측 샘플들을 포함한다. 상기 예측된 블록은 인코딩 장치 및 디코딩 장치에서 동일하게 도출되며, 상기 인코딩 장치는 원본 블록의 원본 샘플 값 자체가 아닌 상기 원본 블록과 상기 예측된 블록 간의 레지듀얼에 대한 정보(레지듀얼 정보)를 디코딩 장치로 시그널링함으로써 영상 코딩 효율을 높일 수 있다. 디코딩 장치는 상기 레지듀얼 정보를 기반으로 레지듀얼 샘플들을 포함하는 레지듀얼 블록을 도출하고, 상기 레지듀얼 블록과 상기 예측된 블록을 합하여 복원 샘플들을 포함하는 복원 블록을 생성할 수 있고, 복원 블록들을 포함하는 복원 픽처를 생성할 수 있다.

[0079] 상기 레지듀얼 정보는 변환 및 양자화 절차를 통하여 생성될 수 있다. 예를 들어, 인코딩 장치는 상기 원본 블록과 상기 예측된 블록 간의 레지듀얼 블록을 도출하고, 상기 레지듀얼 블록에 포함된 레지듀얼 샘플들(레지듀얼 샘플 어레이)에 변환 절차를 수행하여 변환 계수들을 도출하고, 상기 변환 계수들에 양자화 절차를 수행하여 양자화된 변환 계수들을 도출하여 관련된 레지듀얼 정보를 (비트스트림을 통하여) 디코딩 장치로 시그널링할 수 있다. 여기서 상기 레지듀얼 정보는 상기 양자화된 변환 계수들의 값 정보, 위치 정보, 변환 기법, 변환 커널, 양자화 파라미터 등의 정보를 포함할 수 있다. 디코딩 장치는 상기 레지듀얼 정보를 기반으로 역양자화/역변환 절차를 수행하고 레지듀얼 샘플들(또는 레지듀얼 블록)을 도출할 수 있다. 디코딩 장치는 예측된 블록과 상기 레지듀얼 블록을 기반으로 복원 픽처를 생성할 수 있다. 인코딩 장치는 또한 이후 픽처의 인터 예측을 위한 참조를 위하여 양자화된 변환 계수들을 역양자화/역변환하여 레지듀얼 블록을 도출하고, 이를 기반으로 복원 픽처를 생성할 수 있다.

[0080] 도 4는 본 문서의 실시예(들)이 적용 가능한 개략적인 비디오/영상 인코딩 절차의 예를 나타낸다. 도 4에서 S400은 도 2에서 상술한 인코딩 장치의 예측부(220)에서 수행될 수 있고, S410은 레지듀얼 처리부(230)에서 수행될 수 있고, S420은 엔트로피 인코딩부(240)에서 수행될 수 있다. S400은 본 문서에서 설명된 인터/인트라 예측 절차를 포함할 수 있고, S410은 본 문서에서 설명된 레지듀얼 처리 절차를 포함할 수 있고, S420은 본 문서에서 설명된 정보 인코딩 절차를 포함할 수 있다.

[0081] 도 4를 참조하면, 비디오/영상 인코딩 절차는 도 2에 대한 설명에서 나타난 바와 같이 개략적으로 픽처 복원을 위한 정보(ex. 예측 정보, 레지듀얼 정보, 파티셔닝 정보 등)을 인코딩하여 비트스트림 형태로 출력하는 절차뿐 아니라, 현재 픽처에 대한 복원 픽처를 생성하는 절차 및 복원 픽처에 인루프 필터링을 적용하는 절차(optional)를 포함할 수 있다. 인코딩 장치는 역양자화부(234) 및 역변환부(235)를 통하여 양자화된 변환 계수로부터 (수정된) 레지듀얼 샘플들을 도출할 수 있으며, S400의 출력인 예측 샘플들과 상기 (수정된) 레지듀얼 샘플들을 기반으로 복원 픽처를 생성할 수 있다. 이렇게 생성된 복원 픽처는 상술한 디코딩 장치에서 생성한 복원 픽처와 동일할 수 있다. 상기 복원 픽처에 대한 인루프 필터링 절차를 통하여 수정된 복원 픽처가 생성될 수 있으며, 이는 복호 픽처 버퍼 또는 메모리(270)에 저장될 수 있으며, 디코딩 장치에서의 경우와 마찬가지로, 이후 픽처의 인코딩시 인터 예측 절차에서 참조 픽처로 사용될 수 있다. 상술한 바와 같이 경우에 따라서 상기 인루프 필터링 절차의 일부 또는 전부는 생략될 수 있다. 상기 인루프 필터링 절차가 수행되는 경우, (인루프) 필터링 관련 정보(파라미터)가 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있고, 디코딩 장치는 상기 필터링 관련 정보를 기반으로 인코딩 장치와 동일한 방법으로 인루프 필터링 절차를 수행할 수 있다.

[0082] 이러한 인루프 필터링 절차를 통하여 블록킹 아티팩트(artifact) 및 링잉(ringing) 아티팩트 등 영상/동영상 코

딩시 발생하는 노이즈를 줄일 수 있으며, 주관적/객관적 비주얼 퀄리티를 높일 수 있다. 또한, 인코딩 장치와 디코딩 장치에서 둘 다 인루프 필터링 절차를 수행함으로서, 인코딩 장치와 디코딩 장치는 동일한 예측 결과를 도출할 수 있으며, 픽처 코딩의 신뢰성을 높이고, 픽처 코딩을 위하여 전송되어야 하는 데이터량을 줄일 수 있다.

[0083] 상술한 바와 같이 디코딩 장치뿐 아니라 인코딩 장치서도 픽처 복원 절차가 수행될 수 있다. 각 블록 단위로 인트라 예측/인터 예측에 기반하여 복원 블록이 생성될 수 있으며, 복원 블록들을 포함하는 복원 픽처가 생성될 수 있다. 현재 픽처/슬라이스/타일 그룹이 I 픽처/슬라이스/타일 그룹인 경우 상기 현재 픽처/슬라이스/타일 그룹에 포함되는 블록들은 인트라 예측만을 기반으로 복원될 수 있다. 한편, 현재 픽처/슬라이스/타일 그룹이 P 또는 B 픽처/슬라이스/타일 그룹인 경우 상기 현재 픽처/슬라이스/타일 그룹에 포함되는 블록들은 인트라 예측 또는 인터 예측을 기반으로 복원될 수 있다. 이 경우 현재 픽처/슬라이스/타일 그룹 내 일부 블록들에 대하여는 인터 예측이 적용되고, 나머지 일부 블록들에 대하여는 인트라 예측이 적용될 수도 있다. 픽처의 컬러 성분은 루마 성분 및 크로마 성분을 포함할 수 있으며, 본 문서에서 명시적으로 제한하지 않으면 본 문서에서 제안되는 방법들 및 실시예들은 루마 성분 및 크로마 성분에 적용될 수 있다.

[0084] 도 5는 본 문서의 실시예(들)이 적용 가능한 개략적인 비디오/영상 디코딩 절차의 예를 나타낸다. 도 5에서 S500은 도 3에서 상술한 디코딩 장치의 엔트로피 디코딩부(310)에서 수행될 수 있고, S510은 예측부(330)에서 수행될 수 있고, S520은 레지듀얼 처리부(320)에서 수행될 수 있고, S530은 가산부(340)에서 수행될 수 있고, S540은 필터링부(350)에서 수행될 수 있다. S500은 본 문서에서 설명된 정보 디코딩 절차를 포함할 수 있고, S510은 본 문서에서 설명된 인터/인트라 예측 절차를 포함할 수 있고, S520은 본 문서에서 설명된 레지듀얼 처리 절차를 포함할 수 있고, S530은 본 문서에서 설명된 블록/픽처 복원 절차를 포함할 수 있고, S540은 본 문서에서 설명된 인루프 필터링 절차를 포함할 수 있다.

[0085] 도 5를 참조하면, 픽처 디코딩 절차는 도 3에 대한 설명에서 나타난 바와 같이 개략적으로 비트스트림으로부터 (디코딩을 통한) 영상/비디오 정보 획득 절차(S500), 픽처 복원 절차(S510~S530) 및 복원된 픽처에 대한 인루프 필터링 절차(S540)를 포함할 수 있다. 상기 픽처 복원 절차는 본 문서에서 설명된 인터/인트라 예측(S510) 및 레지듀얼 처리(S520, 양자화된 변환 계수에 대한 역양자화, 역변환) 과정을 거쳐서 획득한 예측 샘플들 및 레지듀얼 샘플들을 기반으로 수행될 수 있다. 상기 픽처 복원 절차를 통하여 생성된 복원 픽처에 대한 인루프 필터링 절차를 통하여 수정된(modified) 복원 픽처가 생성될 수 있으며, 상기 수정된 복원 픽처가 디코딩된 픽처로서 출력될 수 있고, 또한 디코딩 장치의 복호 픽처 버퍼 또는 메모리(360)에 저장되어 이후 픽처의 디코딩시 인터 예측 절차에서 참조 픽처로 사용될 수 있다.

[0086] 경우에 따라서 상기 인루프 필터링 절차는 생략될 수 있으며, 이 경우 상기 복원 픽처가 디코딩된 픽처로서 출력될 수 있고, 또한 디코딩 장치의 복호 픽처 버퍼 또는 메모리(360)에 저장되어 이후 픽처의 디코딩시 인터 예측 절차에서 참조 픽처로 사용될 수 있다. 상기 인루프 필터링 절차(S540)는 상술한 바와 같이 디블록킹 필터링 절차, SAO(sample adaptive offset) 절차, ALF(adaptive loop filter) 절차 및/또는 바이래터럴 필터(bi-lateral filter) 절차 등을 포함할 수 있고, 그 일부 또는 전부가 생략될 수 있다. 또한, 상기 디블록킹 필터링 절차, SAO(sample adaptive offset) 절차, ALF(adaptive loop filter) 절차 및 바이래터럴 필터(bi-lateral filter) 절차들 중 하나 또는 일부가 순차적으로 적용될 수 있고, 또는 모두가 순차적으로 적용될 수도 있다. 예를 들어, 복원 픽처에 대하여 디블록킹 필터링 절차가 적용된 후 SAO 절차가 수행될 수 있다. 또는 예를 들어 복원 픽처에 대하여 디블록킹 필터링 절차가 적용된 후 ALF 절차가 수행될 수 있다. 이는 인코딩 장치에서도 마찬가지로 수행될 수 있다.

[0087] 상술한 바와 같이 인코딩 장치는 인트라/인터/IBC 예측 등을 통하여 예측된 블록(예측 샘플들)을 기반으로 레지듀얼 블록(레지듀얼 샘플들)을 도출할 수 있고, 도출된 레지듀얼 샘플들에 변환 및 양자화를 적용하여 양자화된 변환 계수들을 도출할 수 있다. 양자화된 변환 계수들에 대한 정보(레지듀얼 정보)는 레지듀얼 코딩 신택스에 포함되어 인코딩 후 비트스트림 형태로 출력될 수 있다. 디코딩 장치는 상기 비트스트림으로부터 상기 양자화된 변환 계수들에 대한 정보(레지듀얼 정보)를 획득하고, 디코딩하여 양자화된 변환 계수들을 도출할 수 있다. 디코딩 장치는 양자화된 변환 계수들을 기반으로 역양자화/역변환을 거쳐서 레지듀얼 샘플들을 도출할 수 있다. 상술한 바와 같이 상기 양자화/역양자화 및/또는 변환/역변환 중 적어도 하나는 생략될 수 있다. 상기 변환/역변환이 생략되는 경우, 상기 변환 계수는 계수 또는 레지듀얼 계수 라고 불릴 수도 있고, 또는 표현의 통일성을 위하여 변환 계수라고 여전히 불릴 수도 있다. 상기 변환/역변환의 생략 여부는 transform_skip_flag를 기반으로 시그널링될 수 있다. 예를 들어, transform_skip_flag의 값이 1인 경우 상기 변환/역변환이 생략되는 것을

나타낼 수 있고, 이를 변환 스킵 모드로 지칭할 수 있다.

[0088] 일반적으로 비디오/영상 코딩에서는 양자화율을 변화시킬 수 있으며, 변화된 양자화율을 이용하여 압축률을 조절할 수 있다. 구현 관점에서는 복잡도를 고려하여 양자화율을 직접 사용하는 대신 양자화 파라미터(quantization parameter, QP)가 사용될 수 있다. 예를 들어, 0부터 63까지의 정수 값의 양자화 파라미터가 사용될 수 있으며, 각 양자화 파라미터 값은 실제 양자화율에 대응될 수 있다. 또한, 예를 들어, 루마 성분(루마 샘플)에 대한 양자화 파라미터(QP_Y)와 크로마 성분(크로마 샘플)에 대한 양자화 파라미터(QP_C)는 다르게 설정될 수 있다.

[0089] 양자화 과정은 변환 계수(C)를 입력으로 하고, 양자화율(Q_{step})로 나누어서, 이를 기반으로 양자화된 변환 계수(C`)을 얻을 수 있다. 이 경우, 계산 복잡도를 고려하여 양자화율에 스케일을 곱하여 정수 형태로 만들고, 스케일 값에 해당하는 값만큼 쉬프트 연산을 수행할 수 있다. 양자화율과 스케일 값의 곱을 기반으로 양자화 스케일(quantization scale)이 도출될 수 있다. 즉, QP에 따라 상기 양자화 스케일이 도출될 수 있다. 예를 들어, 상기 변환 계수(C)에 상기 양자화 스케일을 적용하여, 이를 기반으로 양자화된 변환 계수(C`)가 도출될 수도 있다.

[0090] 역양자화 과정은 양자화 과정의 역과정으로 양자화된 변환 계수(C`)에 양자화율(Q_{step})을 곱하여, 이를 기반으로 복원된 변환 계수(C``)를 얻을 수 있다. 이 경우 상기 양자화 파라미터에 따라 레벨 스케일(level scale)이 도출될 수 있으며, 상기 양자화된 변환 계수(C`)에 상기 레벨 스케일을 적용하여, 이를 기반으로 복원된 변환 계수(C``)가 도출될 수 있다. 복원된 변환 계수(C``)는 변환 및/또는 양자화 과정에서의 손실(loss)로 인하여 최초 변환 계수(C)와 다소 차이가 있을 수 있다. 따라서, 인코딩 장치에서도 디코딩 장치에서와 동일하게 역양자화을 수행한다.

[0091] 한편, 예측을 수행함에 있어 팔레트 코딩(palette coding)에 기반할 수 있다. 팔레트 코딩은 적은 수의 고유한 색상 값을 포함하는 블록들을 나타내는데 유용한 기술이다. 블록에 대해 예측 및 변환을 적용하는 대신, 팔레트 모드는 각 샘플의 컬러 값을 나타내기 위해 인덱스를 시그널링한다. 이 팔레트 모드는 비디오 메모리 버퍼 공간을 절약하는데 유용하다. 팔레트 모드(예컨대, MODE_PLT)를 사용하여 블록이 코딩될 수 있다. 이와 같이 인코딩된 블록을 디코딩하기 위해서, 디코더는 팔레트 컬러 및 인덱스를 디코딩해야 한다. 팔레트 컬러는 팔레트 테이블에 의해 나타낼 수 있고 팔레트 테이블 코딩 도구에 의해 인코딩될 수 있다.

[0092] 도 6은 팔레트 코딩의 기본적인 구조를 설명하기 위한 일예를 나타낸다.

[0093] 도 6을 참조하면, 이미지(600)는 히스토그램(610)으로 나타낼 수 있다. 이때 주된 컬러 값들은 일반적으로 컬러 인덱스에 매핑되며(620) 이미지는 컬러 인덱스 맵을 사용하여 코딩될 수 있다(630).

[0094] 팔레트 코딩은 (인트라) 팔레트 모드 또는 (인트라) 팔레트 코딩 모드 등으로 불릴 수 있다. 팔레트 코딩 또는 팔레트 모드에 따라 현재 블록이 복원될 수 있다. 팔레트 코딩은 인트라 코딩의 일예로 볼 수 있으며, 또는 인트라 예측 방법 중 하나로 볼 수도 있다. 다만, 상술한 스킵 모드와 유사하게 해당 블록에 대한 별도의 레지듀얼 값은 시그널링되지 않을 수 있다.

[0095] 예를 들어, 팔레트 모드는 상당한 양의 텍스트 및 그래픽을 포함하는 컴퓨터 생성 비디오(computer generated video)와 같은 스크린 컨텐츠에 대한 코딩 효율성을 향상시키는데 사용될 수 있다. 일반적으로 스크린 컨텐츠의 로컬 영역에는 날카로운 엣지(sharp edge)에 의해 구분되는 몇 가지 색상들이 있다. 이러한 특성을 활용하기 위해서, 팔레트 모드는 팔레트 테이블의 컬러 엔트리(color entries)를 가리키는 인덱스(indexes)를 기반으로 블록의 샘플들을 나타낼 수 있다.

[0096] 예를 들어, 팔레트 테이블에 대한 정보가 시그널링될 수 있다. 팔레트 테이블은 각각의 컬러에 대응하는 인덱스 값을 포함할 수 있다. 팔레트 인덱스 예측 데이터가 수신될 수 있고, 비디오 데이터의 픽셀을 팔레트 테이블의 컬러 인덱스에 매핑하는 팔레트 인덱스 맵(palette index map)의 적어도 일부에 대한 인덱스 값을 나타내는 데이터를 포함할 수 있다. 팔레트 인덱스 예측 데이터는 팔레트 인덱스 맵의 적어도 일부에 대한 인덱스 값과 런 값(run values)을 연관시키는 런 값 데이터(run value data)를 포함할 수 있다. 런 값은 이스케이프 컬러 인덱스(escape color index)와 연관될 수 있다. 팔레트 인덱스 맵은 마지막 인덱스 값을 기반으로 팔레트 인덱스 예측 데이터의 인덱스 값을 조정할지 여부를 결정함으로써 적어도 부분적으로 팔레트 인덱스 예측 데이터로부터 생성될 수 있다. 픽처 내의 현재 블록은 팔레트 인덱스 맵에 따라 재구성될 수 있다.

[0097] 팔레트 모드를 사용하는 경우, CU의 픽셀 값들은 대표 컬러 값들의 세트로 나타낼 수 있다. 이러한 세트는 팔레

트로서 지칭될 수 있다. 팔레트 내 컬러 값에 가까운 값을 가진 픽셀의 경우, 팔레트 내 컬러 값에 대응하는 팔레트 인덱스가 시그널링될 수 있다. 팔레트 이외의 컬러 값을 가진 픽셀의 경우, 그 픽셀은 이스케이프 심볼(escape symbol)로 표시될 수 있고 양자화된 픽셀 값이 직접적으로 시그널링될 수 있다. 본 문서에서, 픽셀 또는 픽셀 값은 샘플 또는 샘플 값으로서 지칭될 수 있다.

[0098] 팔레트 모드로 인코딩된 블록을 디코딩하기 위해서, 디코더는 팔레트 컬러 및 인덱스를 디코딩해야한다. 팔레트 컬러는 팔레트 테이블로 나타낼 수 있고 팔레트 테이블 코딩 도구로 인코딩될 수 있다. 이스케이프 플래그가 각 CU에 대해 시그널링되어 현재 CU에 이스케이프 심볼이 존재하는지 여부를 나타낼 수 있다. 이스케이프 심볼이 존재하는 경우, 팔레트 테이블이 1만큼 증가되고 마지막 인덱스가 이스케이프 모드로 할당될 수 있다. CU 내의 모든 픽셀들의 팔레트 인덱스는 팔레트 인덱스 맵을 형성하고 팔레트 인덱스 맵 코딩 도구에 의해 인코딩될 수 있다.

[0099] 예를 들어, 팔레트 테이블의 코딩을 위해 팔레트 예측자(palette predictor)가 유지될 수 있다. 예측자가 0으로 리셋되는 각 슬라이스의 시작 부분에서 예측자가 초기화될 수 있다. 팔레트 예측자의 각 엔트리에 대해, 재사용 플래그(reuse flag)가 시그널링되어 현재 팔레트의 일부인지 여부를 나타낼 수 있다. 재사용 플래그는 0의 run-length 코딩을 사용하여 전송될 수 있다. 그 후, 새로운 팔레트 엔트리의 개수는 0 order의 지수 골롬(exponential Golomb) 코딩을 사용하여 시그널링될 수 있다. 마지막으로, 새 팔레트 엔트리에 대한 컴포넌트 값(component value)이 시그널링될 수 있다. 현재 CU를 인코딩한 후, 팔레트 예측자는 현재 팔레트를 사용하여 업데이트될 수 있고, 현재 팔레트에서 재사용되지 않는 이전 팔레트 예측자의 엔트리는 허용되는 최대 크기에 도달할 때까지 새 팔레트 예측자의 끝에 추가될 수 있다(palette stuffing, 팔레트 채우기).

[0100] 예를 들어, 팔레트 인덱스 맵을 코딩하기 위해 수평 및 수직 트래버스 스캔(traverse scans)을 사용하여 인덱스를 코딩할 수 있다. 스캔 순서(scan order)는 플래그 정보(예컨대, palette_transpose_flag)를 사용하여 비트스트림으로부터 명시적으로 시그널링될 수 있다.

[0101] 도 7은 팔레트 인덱스 맵을 코딩하기 위해 사용되는 수평 및 수직 트래버스 스캔 방법을 설명하기 위한 일예를 나타낸다.

[0102] 도 7의 (a)는 수평 트래버스 스캔을 사용하여 팔레트 인덱스 맵을 코딩하는 일예를 나타내고, 도 7의 (b)는 수직 트래버스 스캔을 사용하여 팔레트 인덱스 맵을 코딩하는 일예를 나타낸다.

[0103] 도 7의 (a)에 도시된 바와 같이, 수평 스캔을 사용하는 경우 현재 블록(즉, 현재 CU) 내의 첫번째 행(최상단 행)에 있는 샘플들부터 마지막 행(최하단 행)에 있는 샘플들까지 수평 방향으로 스캔하여 팔레트 인덱스를 코딩할 수 있다.

[0104] 도 7의 (b)에 도시된 바와 같이, 수직 스캔을 사용하는 경우 현재 블록(즉, 현재 CU) 내의 첫번째 열(가장 좌측 열)에 있는 샘플들부터 마지막 열(가장 우측 열)에 있는 샘플들까지 수직 방향으로 스캔하여 팔레트 인덱스를 코딩할 수 있다.

[0105] 한편, 팔레트 인덱스는 두 가지 팔레트 샘플 모드를 사용하여 코딩될 수 있는데, 예컨대 "INDEX" 모드 및 "COPY_ABOVE" 모드를 사용할 수 있다. 이러한 팔레트 모드는 "INDEX" 모드인지 "COPY_ABOVE" 모드인지를 나타내는 플래그를 사용하여 시그널링될 수 있다. 이때 수평 스캔이 사용되는 경우에는 상단 행(top row)을 제외하고 플래그를 시그널링할 수 있고, 수직 스캔이 사용되는 경우 또는 이전 모드가 "COPY_ABOVE" 모드인 경우에는 첫번째 열(first column)을 제외하고 플래그를 시그널링할 수 있다. "COPY_ABOVE" 모드에서는 위에 있는 행의 샘플의 팔레트 인덱스가 복사될 수 있다. "INDEX" 모드에서는 팔레트 인덱스가 명시적으로 시그널링될 수 있다. "INDEX" 모드 및 "COPY_ABOVE" 모드 둘 다에 대해, 동일한 모드를 사용하여 코딩된 픽셀 개수를 나타내는 런 값이 시그널링될 수 있다.

[0106] 인덱스 맵에 대한 인코딩 순서는 다음과 같다. 먼저, CU에 대한 인덱스 값의 개수가 시그널링될 수 있다. 그 다음에 TB(truncated binary) 코딩을 사용하여 전체 CU에 대한 실제 인덱스 값들을 시그널링할 수 있다. 인덱스 개수와 인덱스 값들은 모두 바이패스 모드(bypass mode)로 코딩될 수 있다. 이는 인덱스 관련 바이패스 빈(index-related bypass bins)이 함께 그룹화될 수 있다. 다음으로, 팔레트 모드(INDEX" 모드 또는 "COPY_ABOVE" 모드) 및 런이 인터리브 방식(interleaved manner)으로 시그널링될 수 있다. 마지막으로, 전체 CU에 대한 이스케이프 샘플들에 대응하는 컴포넌트 이스케이프 값들이 함께 그룹화되고 바이패스 모드로 코딩될 수 있다. 인덱스 값들을 시그널링한 후에 추가 신택스 요소인 last_run_type_flag가 시그널링될 수 있다. 이 신택스 요소는 인덱스 개수와 함께, 블록 내의 마지막 런에 대응하는 런 값을 시그널링할 필요가 없다.

[0107] 한편, VVC 표준에서는 루마와 크로마에 대한 코딩 유닛 파티셔닝을 분리하는 I 슬라이스에 대해 듀얼 트리가 사용(enabled)될 수 있다. 팔레트 코딩(팔레트 모드)은 루마(Y 성분) 및 크로마(Cb 및 Cr 성분)에 개별적으로 또는 함께 적용될 수 있다. 듀얼 트리가 사용되지 않으면(disable), 팔레트 코딩(팔레트 모드)은 루마(Y 성분) 및 크로마(Cb 및 Cr 성분)에 함께 적용될 수 있다.

[0108] 도 8은 팔레트 모드 기반 코딩 방법의 일예를 설명하기 위한 도면이다.

[0109] 도 8을 참조하면, 디코딩 장치는 비트스트림 및/또는 이전 팔레트 정보를 기반으로 팔레트 정보를 획득할 수 있다(S800).

[0110] 일 실시예로, 디코딩 장치는 비트스트림으로부터 팔레트 인덱스 정보, 트래버스 방향(스캔 순서) 정보, CU 내의 샘플들을 순회하며 각 샘플 위치에 대한 팔레트 모드 정보 및 각 팔레트 모드의 연속 길이(run-length) 정보를 수신할 수 있다.

[0111] 디코딩 장치는 팔레트 정보를 기반으로 팔레트를 구성할 수 있다(S810).

[0112] 일 실시예로, 디코딩 장치는 팔레트 예측자를 구성할 수 있다. 이전 블록에서 사용한 팔레트 정보를 추후 발생할 다음 팔레트 CU(즉, 팔레트 모드로 코딩되는 CU)를 위해 저장할 수 있는데, 이를 팔레트 예측자 엔트리로 정의할 수 있다. 그리고, 디코딩 장치는 신규 팔레트 엔트리 정보를 수신하고, 현재 CU에 대한 팔레트를 구성할 수 있다. 예컨대, 디코딩 장치는 수신된 팔레트 예측자 재사용 정보와 현재 CU에서 사용될 새로운 팔레트 엔트리 정보를 수신한 뒤, 이 두 엔트리 정보를 결합하여 현재 CU를 대표하는 한 개의 팔레트로 구성할 수 있다.

[0113] 디코딩 장치는 팔레트 기반 현재 블록 내 샘플값(샘플 예측값)을 도출할 수 있다(S820).

[0114] 일 실시예로, 디코딩 장치는 트래버스 방향(스캔 순서) 정보를 기반으로 CU 내 샘플들을 수평 방향 혹은 수직 방향으로 순회하면서 상기 획득된 팔레트 정보로부터 샘플들을 구성할 수 있다. 만일 팔레트 모드 정보가 COPY_ABOVE 모드를 나타내는 경우, 수직 방향 스캔에서는 좌측 샘플 위치의 인덱스 정보를 복사하고 수평 방향 스캔에서는 상단 샘플 위치의 인덱스 정보를 복사하여 CU 내 각 샘플 값을 도출할 수 있다. 즉, CU 내 각 샘플의 인덱스 정보를 기반으로 상기 구성된 팔레트 테이블로부터 각 샘플의 컬러 값을 도출하여, CU 내 예측 샘플들을 도출할 수 있다. 그리고, 디코딩 장치는 팔레트 정보를 이용한 CU 내 각 샘플 정보를 재구성하고 팔레트 예측자를 업데이트 할 수 있다.

[0115] 한편, 상술한 팔레트 코딩(팔레트 모드 혹은 팔레트 코딩 모드)은 현재 CU에 대해 팔레트 모드로 코딩된 것인지 여부를 지시하고 팔레트 모드를 적용하여 코딩하기 위한 정보를 시그널링할 수 있다.

[0116] 일예로, 팔레트 코딩 모드의 가용 여부에 관한 정보가 다음 표 1과 같이 SPS(sequence parameter set)을 통하여 시그널링될 수 있다.

표 1

[0117]

seq_parameter_set_rbsp( ) {	**Descriptor**
…	
sps_decoding_parameter_set_id	u(4)
if(chroma_format_idc = = 3)	
sps_palette_enabled_flag	u(1)
…	
}	

[0118] 상기 표 1의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 2와 같이 나타낼 수 있다.

표 2

sps_palette_enabled_flag equal to 1 specifies that pred_mode_plt_flag may be present in the coding unit syntax. sps_palette_enabled_flag equal to 0 specifies that pred_mode_plt_flag is not present in the coding unit syntax. When sps_palette_enabled_flag is not present, it is inferred to be equal to 0. Additionally, during the decoding of the coding unit (CU), a flag is enabled to indicate if the current CU is coded in the palette mode.

[0119]

[0120] 상기 표 1 및 표 2를 참조하면, SPS에서 sps_palette_enabled_flag 신택스 요소가 파싱/시그널링될 수 있다. sps_palette_enabled_flag 신택스 요소는 팔레트 코딩 모드가 가용한지 여부를 나타낼 수 있다. 예를 들어, sps_palette_enabled_flag의 값이 1인 경우 팔레트 코딩 모드가 가용한 것을 나타낼 수 있고, 이때 코딩 유닛 신택스에서 현재 코딩 유닛에 대해 팔레트 코딩 모드를 적용하는지 여부를 나타내는 정보(예: pred_mode_plt_flag)를 파싱/시그널링할 수 있다. 또는, sps_palette_enabled_flag의 값이 0인 경우 팔레트 코딩 모드가 가용하지 않는 것을 나타낼 수 있고, 이때 코딩 유닛 신택스에서 현재 코딩 유닛에 대해 팔레트 코딩 모드를 적용하는지 여부를 나타내는 정보(예: pred_mode_plt_flag)를 파싱/시그널링하지 않을 수 있다.

[0121] 또한, 일예로, 팔레트 코딩 모드의 가용 여부에 관한 정보(예: sps_palette_enabled_flag)를 기반으로, 팔레트 모드를 적용하여 코딩할지 여부에 관한 정보가 시그널링될 수 있으며, 다음 표 3에서와 같이 코딩 유닛 신택스를 통하여 시그널링될 수 있다.

표 3

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	**Descriptor**
chType = treeType = = DUAL_TREE_CHROMA? 1 : 0	
if(slice_type != I || sps_ibc_enabled_flag || sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !((cbWidth = = 4 && cbHeight = = 4) || modeType = = MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] = = 0 && slice_type != I && !(cbWidth = = 4 && cbHeight = = 4) && modeType = = MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if(((slice_type = = I && cu_skip_flag[x0][y0] = =0) || (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA || (cbWidth = = 4 && cbHeight = = 4 && cu_skip_flag[x0][y0] = = 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
if((((slice_type = = I || (cbWidth = = 4 && cbHeight = = 4) || sps_ibc_enabled_flag) && CuPredMode[x0][y0] = = MODE_INTRA) || (slice_type != I && !(cbWidth = = 4 && cbHeight = = 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && && cu_skip_flag[x0][y0] = = 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA || CuPredMode[chType][x0][y0] = = MODE_PLT) {	
if(treeType = = SINGLE_TREE || treeType = = DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
if(treeType = = DUAL_TREE_LUMA)	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 1)	
else /* SINGLE_TREE */	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 3)	
}	
…	
}	

[0122]

[0123] 상기 표 3의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 4와 같이 나타낼 수 있다.

표 4

pred_mode_plt_flag specifies the use of palette mode in the current coding unit. pred_mode_plt_flag equal to 1 indicates that palette mode is applied in the current coding unit. pred_mode_plt_flag equal to 0 indicates that palette mode is not applied in the current coding unit. When pred_mode_plt_flag is not present, it is inferred to be equal to 0. When pred_mode_plt_flag is equal to 1, the variable CuPredMode[x][y] is set to be equal to MODE_PLT for x = x0..x0 + cbWidth − 1 and y = y0..y0 + cbHeight − 1.

[0124]

[0125] 상기 표 3 및 표 4를 참조하면, 코딩 유닛 신택스에서 pred_mode_plt_flag 신택스 요소가 파싱/시그널링될 수 있다. pred_mode_plt_flag 신택스 요소는 현재 코딩 유닛에 팔레트 모드가 적용되는지 여부를 나타낼 수 있다. 예를 들어, pred_mode_plt_flag의 값이 1인 경우 팔레트 모드가 현재 코딩 유닛에 적용되는 것을 지시하고, pred_mode_plt_flag의 값이 0인 경우 현재 코딩 유닛에 팔레트 모드가 적용되지 않음을 지시할 수 있다.

[0126] 이때, pred_mode_plt_flag는 팔레트 코딩 모드의 가용 여부에 관한 정보(예: sps_palette_enabled_flag)를 기반으로 파싱/시그널링될 수 있다. 예를 들어, sps_palette_enabled_flag의 값이 1인 경우(즉, 팔레트 코딩 모드가 가용한 경우), pred_mode_plt_flag가 파싱/시그널링될 수 있다.

[0127] 또한, pred_mode_plt_flag를 기반으로 현재 코딩 유닛에 대해 팔레트 모드를 적용하여 코딩을 수행할 수 있다. 예를 들어, pred_mode_plt_flag의 값이 1인 경우 palette_coding() 신택스를 파싱/시그널링함으로써, 현재 코딩 유닛에 대해 팔레트 모드를 적용하여 복원 샘플을 생성할 수 있다.

[0128] 일예로, 다음 표 5는 팔레트 코딩 신택스를 나타낸다.

표 5

palette_coding(x0, y0, cbWidth, cbHeight, startComp, numComps) {	Descriptor
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
for(predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[startComp] && !palettePredictionFinished && NumPredictedPaletteEntries[startComp] < palette_max_size; predictorEntryIdx++) {	
palette_predictor_run	ae(v)
if(palette_predictor_run != 1) {	
if(palette_predictor_run > 1)	
predictorEntryIdx += palette_predictor_run − 1	
PalettePredictorEntryReuseFlags[predictorEntryIdx] = 1	
NumPredictedPaletteEntries++	
} else	
palettePredictionFinished = 1	
}	
if(NumPredictedPaletteEntries < palette_max_size)	
num_signalled_palette_entries	ae(v)
for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++)	
for(i = 0; i < num_signalled_palette_entries; i++)	
new_palette_entries[cIdx][i]	ae(v)
if(CurrentPaletteSize[startComp] > 0)	
palette_escape_val_present_flag	ae(v)
if(MaxPaletteIndex > 0) {	
num_palette_indices_minus1	ae(v)
adjust = 0	
for(i = 0; i <= num_palette_indices_minus1; i++) {	
if(MaxPaletteIndex − adjust > 0) {	
palette_idx_idc	ae(v)
PaletteIndexIdc[i] = palette_idx_idc	
}	
adjust = 1	
}	
copy_above_indices_for_final_run_flag	ae(v)
palette_transpose_flag	ae(v)
}	
if(treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
}	
if(treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag) {	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag)	

[0129]

cu_chroma_qp_offset_idx	ae(v)
}	
}	
remainingNumIndices = num_palette_indices_minus1 + 1	
PaletteScanPos = 0	
log2CbWidth = Log2(cbWidth)	
log2CbHeight = Log2(cbHeight)	
while(PaletteScanPos < cbWidth*cbHeightt) {	
xC = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][0]	
yC = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][1]	
if(PaletteScanPos > 0) {	
xcPrev = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos − 1][0]	
ycPrev = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos − 1][1]	
}	
PaletteRunMinus1 = cbWidth * cbHeight − PaletteScanPos − 1	
RunToEnd = 1	
CopyAboveIndicesFlag[xC][yC] = 0	
if(MaxPaletteIndex > 0)	
if(((!palette_transpose_flag && yC > 0) || (palette_transpose_flag && xC > 0)) && CopyAboveIndicesFlag[xcPrev][ycPrev] == 0)	
if(remainingNumIndices > 0 && PaletteScanPos < cbWidth* cbHeight − 1) {	
copy_above_palette_indices_flag	ae(v)
CopyAboveIndicesFlag[xC][yC] = copy_above_palette_indices_flag	
} else {	
if(PaletteScanPos == cbWidth * cbHeight − 1 && remainingNumIndices > 0)	
CopyAboveIndicesFlag[xC][yC] = 0	
else	
CopyAboveIndicesFlag[xC][yC] = 1	
}	
if(CopyAboveIndicesFlag[xC][yC] == 0) {	
currNumIndices = num_palette_indices_minus1 + 1 − remainingNumIndices	
PaletteIndexMap[xC][yC] = PaletteIndexIdc[currNumIndices]	
}	
if(MaxPaletteIndex > 0) {	
if(CopyAboveIndicesFlag[xC][yC] == 0)	
remainingNumIndices − = 1	
if(remainingNumIndices > 0 || CopyAboveIndicesFlag[xC][yC] != copy_above_indices_for_final_run_flag) {	
PaletteMaxRunMinus1 = cbWidth * cbHeight − PaletteScanPos − 1 − remainingNumIndices − copy_above_indices_for_final_run_flag	
RunToEnd = 0	
if(PaletteMaxRunMinus1 > 0) {	
palette_run_prefix	ae(v)

[0130]

if((palette_run_prefix > 1) && (PaletteMaxRunMinus1 != (1 << (palette_run_prefix – 1))))	
palette_run_suffix	ae(v)
}	
}	
}	
runPos = 0	
while (runPos <= PaletteRunMinus1) {	
xR = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][0]	
yR = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][1]	
if(CopyAboveIndicesFlag[xC][yC] == 0) {	
CopyAboveIndicesFlag[xR][yR] = 0	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xC][yC]	
} else {	
CopyAboveIndicesFlag[xR][yR] = 1	
if(!palette_transpose_flag)	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR][yR − 1]	
else	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR − 1][yR]	
}	
runPos++	
PaletteScanPos ++	
}	
}	
if(palette_escape_val_present_flag) {	
for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++)	
for(sPos = 0; sPos < cbWidth* cbHeight; sPos++) {	
xC = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][0]	
yC = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][1]	
if(PaletteIndexMap[cIdx][xC][yC] == MaxPaletteIndex) {	
palette_escape_val	ae(v)
PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val	
}	
}	
}	
}	

[0131]

[0132] 상기 표 5의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 6과 같이 나타낼 수 있다.

표 6

In the following semantics, the array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture. The array indices xC, yC specify the location (xC, yC) of the sample relative to the top-left luma sample of the picture. The array index startComp specifies the first colour component of the current palette table. startComp equal to 0 indicates the Y component; startComp equal to 1 indicates the Cb component; startComp equal to 2 indicates the Cr component. numComps specifies the number of colour components in the current palette table.

The predictor palette consists of palette entries from previous coding units that are used to predict the entries in the current palette.

The variable PredictorPaletteSize[startComp] specifies the size of the predictor palette for the first colour component of the current palette table startComp. PredictorPaletteSize is derived as specified in Section 1.4

The variable PalettePredictorEntryReuseFlags[i] equal to 1 specifies that the i-th entry in the predictor palette is reused in the current palette. PalettePredictorEntryReuseFlags[i] equal to 0 specifies that the i-th entry in the predictor palette is not an entry in the current palette. All elements of the array PalettePredictorEntryReuseFlags[i] are initialized to 0.

palette_predictor_run is used to determine the number of zeros that precede a non-zero entry in the array PalettePredictorEntryReuseFlags.

It is a requirement of bitstream conformance that the value of palette_predictor_run shall be in the range of 0 to (PredictorPaletteSize − predictorEntryIdx), inclusive, where predictorEntryIdx corresponds to the current position in the array PalettePredictorEntryReuseFlags. The variable NumPredictedPaletteEntries specifies the number of entries in the current palette that are reused from the predictor palette. The value of NumPredictedPaletteEntries shall be in the range of 0 to palette_max_size, inclusive.

num_signalled_palette_entries specifies the number of entries in the current palette that are explicitly signalled for the first colour component of the current palette table startComp.

When num_signalled_palette_entries is not present, it is inferred to be equal to 0.

The variable CurrentPaletteSize[startComp] specifies the size of the current palette for the first colour component of the current palette table startComp and is derived as follows:

[0133]

CurrentPaletteSize[startComp] = NumPredictedPaletteEntries + num_signalled_palette_entries

The value of CurrentPaletteSize[startComp] shall be in the range of 0 to palette_max_size, inclusive.

new_palette_entries[cIdx][i] specifies the value for the i-th signalled palette entry for the colour component cIdx.

The variable PredictorPaletteEntries[cIdx][i] specifies the i-th element in the predictor palette for the colour component cIdx.

The variable CurrentPaletteEntries[cIdx][i] specifies the i-th element in the current palette for the colour component cIdx and is derived as follows:

```
numPredictedPaletteEntries = 0
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++ )
    if( PalettePredictorEntryReuseFlags[ i ] ) {
        for( cIdx =startComp; cIdx < ( startComp + numComps ); cIdx++ )
            CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] =
PredictorPaletteEntries[ cIdx ][ i ]
        numPredictedPaletteEntries++
        }

for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++)
    for( i = 0; i < num_signalled_palette_entries[startComp]; i++ )
        CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] =
new_palette_entries[ cIdx ][ i ]
```

palette_escape_val_present_flag equal to 1 specifies that the current coding unit contains at least one escape coded sample. escape_val_present_flag equal to 0 specifies that there are no escape coded samples in the current coding unit. When not present, the value of palette_escape_val_present_flag is inferred to be equal to 1.

[0134]

The variable MaxPaletteIndex specifies the maximum possible value for a palette index for the current coding unit. The value of MaxPaletteIndex is set equal to CurrentPaletteSize[startComp] − 1 + palette_escape_val_present_flag.

num_palette_indices_minus1 plus 1 is the number of palette indices explicitly signalled or inferred for the current block.

When num_palette_indices_minus1 is not present, it is inferred to be equal to 0.

palette_idx_idc is an indication of an index to the palette table, CurrentPaletteEntries. The value of palette_idx_idc shall be in the range of 0 to MaxPaletteIndex, inclusive, for the first index in the block and in the range of 0 to (MaxPaletteIndex − 1), inclusive, for the remaining indices in the block.

When palette_idx_idc is not present, it is inferred to be equal to 0.

The variable PaletteIndexIdc[i] stores the i-th palette_idx_idc explicitly signalled or inferred. All elements of the array PaletteIndexIdc[i] are initialized to 0.

copy_above_indices_for_final_run_flag equal to 1 specifies that the palette indices of the last positions in the coding unit are copied from the palette indices in the row above if horizontal traverse scan is used or the palette indices in the left column if vertical traverse scan is used. copy_above_indices_for_final_run_flag equal to 0 specifies that the palette indices of the last positions in the coding unit are copied from PaletteIndexIdc[num_palette_indices_minus1].

When copy_above_indices_for_final_run_flag is not present, it is inferred to be equal to 0.

palette_transpose_flag equal to 1 specifies that vertical traverse scan is applied for scanning the indices for samples in the current coding unit. palette_transpose_flag equal to 0 specifies that horizontal traverse scan is applied for scanning the indices for samples in the current coding unit. When not present, the value of palette_transpose_flag is inferred to be equal to 0.

The array TraverseScanOrder specifies the scan order array for palette coding. TraverseScanOrder is assigned the horizontal scan order HorTravScanOrder if

[0135]

palette_transpose_flag is equal to 0 and TraverseScanOrder is assigned the vertical scan order VerTravScanOrder if if palette_transpose_flag is equal to 1.

copy_above_palette_indices_flag equal to 1 specifies that the palette index is equal to the palette index at the same location in the row above if horizontal traverse scan is used or the same location in the left column if vertical traverse scan is used. copy_above_palette_indices_flag equal to 0 specifies that an indication of the palette index of the sample is coded in the bitstream or inferred.

The variable CopyAboveIndicesFlag[xC][yC] equal to 1 specifies that the palette index is copied from the palette index in the row above (horizontal scan) or left column (vertical scan). CopyAboveIndicesFlag[xC][yC] equal to 0 specifies that the palette index is explicitly coded in the bitstream or inferred. The array indices xC, yC specify the location (xC, yC) of the sample relative to the top-left luma sample of the picture. The value of PaletteIndexMap[xC][yC] shall be in the range of 0 to (MaxPaletteIndex – 1), inclusive.

The variable PaletteIndexMap[xC][yC] specifies a palette index, which is an index to the array represented by CurrentPaletteEntries. The array indices xC, yC specify the location (xC, yC) of the sample relative to the top-left luma sample of the picture. The value of PaletteIndexMap[xC][yC] shall be in the range of 0 to MaxPaletteIndex, inclusive.

The variable adjustedRefPaletteIndex is derived as follows:

```
adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev =
x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos − 1 ][ 0 ]
    ycPrev =
y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos − 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ]  = =  0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] {
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC − 1 ]
```

[0136]

```
            else
                adjustedRefPaletteIndex = PaletteIndexMap[ xC − 1 ][ yC ]
        }
    }
```

When CopyAboveIndicesFlag[xC][yC] is equal to 0, the variable CurrPaletteIndex is derived as follows:

```
if( CurrPaletteIndex  >=  adjustedRefPaletteIndex )
    CurrPaletteIndex++
```

palette_run_prefix, when present, specifies the prefix part in the binarization of PaletteRunMinus1.

palette_run_suffix is used in the derivation of the variable PaletteRunMinus1. When not present, the value of palette_run_suffix is inferred to be equal to 0.

When RunToEnd is equal to 0, the variable PaletteRunMinus1 is derived as follows:

– If PaletteMaxRunMinus1 is equal to 0, PaletteRunMinus1 is set equal to 0.

– Otherwise (PaletteMaxRunMinus1 is greater than 0) the following applies:

  – If palette_run_prefix is less than 2, the following applies:

```
PaletteRunMinus1=palette_run_prefix
```

  – Otherwise (palette_run_prefix is greater than or equal to 2), the following applies:

```
PrefixOffset = 1  <<  ( palette_run_prefix − 1 )
PaletteRunMinus1 = PrefixOffset + palette_run_suffix
```

The variable PaletteRunMinus1 is used as follows:

– If CopyAboveIndicesFlag[xC][yC] is equal to 0, PaletteRunMinus1 specifies the number of consecutive locations minus 1 with the same palette index.

– Otherwise if palette_transpose_flag equal to 0, PaletteRunMinus1 specifies the number of consecutive locations minus 1 with the same palette index as used in the corresponding position in the row above.

[0137]

- Otherwise, PaletteRunMinus1 specifies the number of consecutive locations minus 1 with the same palette index as used in the corresponding position in the left column.

When RunToEnd is equal to 0, the variable PaletteMaxRunMinus1 represents the maximum possible value for PaletteRunMinus1 and it is a requirement of bitstream conformance that the value of PaletteMaxRunMinus1 shall be greater than or equal to 0.

palette_escape_val specifies the quantized escape coded sample value for a component.

The variable PaletteEscapeVal[cIdx][xC][yC] specifies the escape value of a sample for which PaletteIndexMap[xC][yC] is equal to MaxPaletteIndex and palette_escape_val_present_flag is equal to 1. The array index cIdx specifies the colour component. The array indices xC, yC specify the location (xC, yC) of the sample relative to the top-left luma sample of the picture.

It is a requirement of bitstream conformance that PaletteEscapeVal[cIdx][xC][yC] shall be in the range of 0 to (1 << ($BitDepth_Y$ + 1)) − 1, inclusive, for cIdx equal to 0, and in the range of 0 to (1 << ($BitDepth_C$ + 1)) − 1, inclusive, for cIdx not equal to 0.

[0138]

[0139] 상기 표 5 및 표 6을 참조하면, 현재 블록(즉, 현재 코딩 유닛)에 대해 팔레트 모드가 적용되는 경우 상기 표 5에서와 같은 팔레트 코딩 신택스(예: palette_coding())가 파싱/시그널링될 수 있다.

[0140] 예를 들어, 팔레트 엔트리 정보를 기반으로 팔레트 테이블을 구성할 수 있다. 팔레트 엔트리 정보는 palette_predictor_run, num_signalled_palette_entries, new_palette_entries 등의 신택스 요소들을 포함할 수 있다.

[0141] 또한, 팔레트 인덱스 정보를 기반으로 현재 블록에 대한 팔레트 인덱스 맵을 구성할 수 있다. 팔레트 인덱스 정보는 num_palette_indices_minus1, palette_idx_idc, copy_above_indices_for_final_run_flag, palette_transpose_flag 등의 신택스 요소들을 포함할 수 있다. 상기와 같은 팔레트 인덱스 정보를 기반으로, 트래버스 스캔 방향(수직 방향 혹은 수평 방향)에 따라 순회하면서 현재 블록 내의 샘플들에 대해 팔레트 인덱스 값(예: PaletteIndexIdc)을 도출하여 팔레트 인덱스 맵(예: PaletteIndexMap)을 구성할 수 있다.

[0142] 또한, 팔레트 인덱스 맵을 기반으로 팔레트 테이블 내의 팔레트 엔트리에 대한 샘플 값을 도출하고, 상기 팔레트 엔트리에 매핑된 샘플 값(즉, 컬러 값)을 기반으로 현재 블록의 복원 샘플들을 생성할 수 있다.

[0143] 또한, 현재 블록 내 이스케이프 값을 갖는 샘플이 존재하는 경우(즉, palette_escape_val_present_flag의 값이 1인 경우), 이스케이프 정보를 기반으로 현재 블록에 대한 이스케이프 값을 도출할 수 있다. 이스케이프 정보는 palette_escape_val_present_flag, palette_escape_val 등의 신택스 요소들을 포함할 수 있다. 예를 들어, 양자화된 이스케이프 값 정보(예: palette_escape_val)를 기반으로 현재 블록 내의 이스케이프 코딩된 샘플에 대한 이스케이프 값을 도출할 수 있다. 상기 이스케이프 값을 기반으로 현재 블록의 복원 샘플들을 생성할 수 있다.

[0144] 상술한 바와 같이 본 문서에서 개시되는 신택스 테이블 내 정보(신택스 요소)는 영상/비디오 정보에 포함될 수 있으며, 인코딩 장치에서 수행된 코딩 기법(팔레트 코딩 포함)에 따라 구성/인코딩되어 비트스트림 형태로 디코딩 장치로 전달될 수 있다. 디코딩 장치는 해당 신택스 테이블 내 정보(신택스 요소)을 파싱/디코딩할 수 있다. 디코딩 장치는 디코딩된 정보를 기반으로 팔레트 코딩 등의 코딩 기법을 수행할 수 있고, 이를 기반으로 블록/영상/비디오 복원(디코딩)절차를 수행할 수 있다. 이하 본 문서에서는 팔레트 코딩을 기반으로 블록/영상/비디오를 효율적으로 코딩하기 위한 신택스 테이블 및 신택스 요소를 제안한다.

[0145] 본 문서는 팔레트 모드 코딩에서 이스케이프 값을 효율적으로 코딩하고 시그널링하는 방안을 제안한다. 팔레트 모드에서 블록 내 주변 샘플들과 특성이 다른 값을 갖는 샘플에 대해 별도로 해당 샘플 값을 전송해 주기 위해 이스케이프 값이 사용될 수 있다. 이러한 이스케이프 값은 추가적인 데이터이므로 이를 절약하기 위해 양자화를 할 수 있다. 또한, 팔레트 모드의 이스케이프 코딩에서는 변환이 적용되지 않고 양자화된 이스케이프 값이 직접적으로 시그널링될 수 있다. 이는 코딩 유닛(CU)에 변환이 적용되지 않는 변환 스킵 모드와 유사한 것으로 고려

될 수 있다.

[0146] 현재 VVC 표준에서는 팔레트 모드에서의 이스케이프 값에 전체 범위의 양자화 파라미터(quantization parameters; QP) 값을 적용한다. 그러나, 본 문서에서는 팔레트 모드에서의 이스케이프 값 코딩을 위한 양자화 스텝 사이즈가 1보다 작아지는 것을 방지하기 위해서 QP 값의 범위를 제한하는 방안을 제안한다. 일 실시예로, 변환 스킵을 위한 최소 QP와 동일한 제한이 팔레트 모드에서의 이스케이프 값 코딩에 적용될 수 있다. 변환 스킵을 위한 최소 QP를 사용하여 팔레트 모드를 위한 최소 QP를 클리핑할 수 있다.

[0147] 일예로, 변환 스킵을 위한 최소 QP에 대한 정보는 다음 표 7과 같이 SPS(sequence parameter set)을 통하여 시그널링될 수 있다.

표 7

[0148]

seq_parameter_set_rbsp( ) {	**Descriptor**
sps_decoding_parameter_set_id	u(4)
…..	
min_qp_prime_ts_minus4	ue(v)
….	
if(chroma_format_idc = = 3)	
sps_palette_enabled_flag	u(1)
…	
}	

[0149] 상기 표 7의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 8과 같이 나타낼 수 있다.

표 8

[0150]

min_qp_prime_ts_minus4 specifies the minimum allowed quantization parameter for transform skip mode as follows: QpPrimeTsMin = 4 + min_qp_prime_ts_minus4 min_qp_prime_ts_minus4 is signalled in the SPS and is typically given by 6*(inputBitDepth – 8).

[0151] 상기 표 7 및 표 8을 참조하면, SPS에서 min_qp_prime_ts_minus4 신택스 요소가 파싱/시그널링될 수 있다. min_qp_prime_ts_minus4 신택스 요소는 변환 스킵 모드에 대해 허용되는 최소 양자화 파라미터를 나타낼 수 있다. 다시 말해, min_qp_prime_ts_minus4 신택스 요소를 기반으로 변환 스킵 모드에서의 최소 양자화 파라미터 값(예: QpPrimeTsMin)이 도출될 수 있다. 예를 들어, min_qp_prime_ts_minus4의 값에 4를 더함으로써 최소 양자화 파라미터 값(예: QpPrimeTsMin)을 도출할 수 있다.

[0152] 상술한 바와 같이 SPS에서 시그널링되는 min_qp_prime_ts_minus4 신택스 요소를 기반으로, 팔레트 모드의 이스케이프 값에 대한 QP는 다음 표 9에 개시된 알고리즘과 같이 도출될 수 있다. 즉, 팔레트 모드 기반 디코딩 과정에서 이스케이프 값 재구성을 위해 사용되는 QP 값은 다음 표 9에 개시된 알고리즘과 같이 도출될 수 있다.

표 9

- Otherwise (bIsEscapeSample is equal to 1 and cu_transquant_bypass_flag is equal to 0), the following ordered steps apply:
  1. The quantization parameter qP is derived as follows:
     - If cIdx is equal to 0,

       qP = Max(QpPrimeTsMin, Qp′Y)

     - Otherwise, if cIdx is equal to 1,

       qP = Max(QpPrimeTsMin, Qp′Cb)

     - Otherwise (cIdx is equal to 2),

       qP = Max(QpPrimeTsMin, Qp′Cr)

  2. The variables bitDepth is derived as follows:

     $bitDepth = (cIdx == 0) ? BitDepth_Y : BitDepth_C$

  3. The list levelScale[ ] is specified as levelScale[k] = { 40, 45, 51, 57, 64, 72 } with k = 0..5.
  4. The following applies:

     tmpVal = (PaletteEscapeVal[cIdx][xCb + xL][yCb + yL] * levelScale[qP%6]) << (qP / 6) + 32) >> 6

     recSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, tmpVal)

[0153]

[0154] 표 9를 참조하면, 팔레트 모드의 이스케이프 값이 존재하는 경우에 QP 값이 도출될 수 있다. 즉, 팔레트 모드의 이스케이프 값에 대한 QP는 상술한 min_qp_prime_ts_minus4 신택스 요소를 기반으로 도출되는 변환 스킵 모드에서의 최소 양자화 파라미터 값(예: QpPrimeTsMin)을 기반으로 도출될 수 있다. 예컨대, 상기 표 9에 나타낸 바와 같이, 팔레트 모드의 이스케이프 값에 대한 QP는 QpPrimeTsMin 및 양자화 파라미터 Qp(루마 성분인 경우 Qp'Y, 크로마 성분인 경우 Qp'Cb 또는 Qp'Cr) 중에서 큰 값으로 도출될 수 있다. 그리고, 팔레트 모드의 이스케이프 값에 대한 QP를 기반으로 이스케이프 값을 도출하여 블록 내의 샘플들을 복원할 수 있다.

[0155] 또한, 본 문서에서는 상기와 같이 팔레트 모드에서의 QP 범위가 변환 스킵 모드에서의 최소 양자화 파라미터 값(예: QpPrimeTsMin)보다 크거나 같은 경우로 제한될 때, 팔레트 모드에서 양자화된 이스케이프 값의 범위를 제한할 수 있다. 일 실시예로, 팔레트 모드에서 양자화된 이스케이프 값의 범위는 비트심도(bitdepth)를 기반으로 정해질 수 있으며, 예컨대 (1 << BitDepth) - 1 보다 크지 않도록 제한될 수 있다.

[0156] 일예로, 팔레트 모드에서 양자화된 이스케이프 값은 신택스 요소 palette_escape_val으로 나타낼 수 있다. 상기 신택스 요소 palette_escape_val는 다음 표 10에서와 같이 팔레트 코딩 신택스를 통하여 시그널링될 수 있다.

표 10

palette_coding(x0, y0, cbWidth, cbHeight, treeType) {	**Descriptor**
…..	
if(palette_escape_val_present_flag) {	
for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++) {	
for(sPos = minSubPos; sPos < maxSubPos; sPos++) {	
xC = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][0]	
yC = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][1]	
if(!(treeType = = SINGLE_TREE && cIdx != 0 && (xC % SubWidthC != 0 || yC % SubHeightC != 0))) {	
if(PaletteIndexMap[cIdx][xC][yC] = = MaxPaletteIndex) {	
palette_escape_val	ae(v)
PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val	
}	
}	
}	
}	
}	
}	
}	

[0157]

[0158] 상기 표 10의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 11에서와 같이 나타낼 수 있다.

표 11

palette_escape_val specifies the quantized escape coded sample value for a component.

The variable PaletteEscapeVal[cIdx][xC][yC] specifies the escape value of a sample for which PaletteIndexMap[xC][yC] is equal to MaxPaletteIndex and palette_escape_val_present_flag is equal to 1. The array index cIdx specifies the colour component. The array indices xC, yC specify the location (xC, yC) of the sample relative to the top-left luma sample of the picture.

It is a requirement of bitstream conformance that PaletteEscapeVal[cIdx][xC][yC] shall be in the range of 0 to (1 << ($BitDepth_Y$)) − 1, inclusive, for cIdx equal to 0, and in the range of 0 to (1 << ($BitDepth_C$)) − 1, inclusive, for cIdx not equal to 0.

[0159]

[0160] 상기 표 10 및 표 11을 참조하면, 팔레트 코딩 신택스에서 palette_escape_val 신택스 요소가 파싱/시그널링될 수 있다. palette_escape_val 신택스 요소는 양자화된 이스케이프 값을 나타낼 수 있다. 또한, 표 10에 나타낸 것과 같이 신택스 요소 palette_escape_val의 값이 PaletteEscapeVal로 설정될 수 있고, 이 PaletteEscapeVal은 팔레트 인덱스 맵(PaletteIndexMap)이 최대 팔레트 인덱스(MaxPaletteIndex)와 같고 palette_escape_val_present_flag의 값이 1인 샘플의 이스케이프 값을 나타낼 수 있다. 여기서 palette_escape_val_present_flag의 값이 1인 경우라 함은, 현재 CU에 적어도 하나의 이스케이프 코딩된 샘플(이스케이프 값)을 포함하는 것을 의미할 수 있다. 예를 들어, 루마 성분에 대해, PaletteEscapeVal은 0에서 (1 << ($BitDepth_Y$)) - 1 내의 범위로 제한될 수 있다. 크로마 성분에 대해, PaletteEscapeVal은 0에서 (1 << ($BitDepth_C$)) - 1 내의 범위로 제한될 수 있다.

[0161] 또한, 본 문서에서는 팔레트 사이즈를 정의하고 이를 시그널링하는 방안을 제안한다. 팔레트 사이즈는 팔레트 테이블의 엔트리 개수(즉, 팔레트 테이블의 인덱스 개수)를 나타낼 수 있다. 일 실시예로, 본 문서에서는 하나

이상의 상수(constant(s))로 팔레트 사이즈를 정의함으로써, 팔레트 내의 엔트리 개수를 나타낼 수 있다.

[0162] 일예로, 팔레트 사이즈는 신택스 요소 palette_max_size로 나타낼 수 있고, 신택스 요소 palette_max_size는 전체 시퀀스에 대해 동일하거나 또는 CU 사이즈(즉, CU 내의 픽셀 개수)에 따라 상이할 수도 있다. 예를 들어, 팔레트 사이즈(palette_max_size)는 팔레트 테이블의 최대 허용되는 인덱스를 나타낼 수 있고, 31로 정의될 수 있다. 다른 예로, 팔레트 사이즈(palette_max_size)는 팔레트 테이블의 최대 허용되는 인덱스를 나타낼 수 있고, CU 사이즈에 따라 다음 표 12에서와 같이 정의될 수 있다.

표 12

[0163]

```
if (cbWidth*cbHeight>=1024)
    palette_max_size is defined as 63
else if cbWidth*cbHeight>=256
    palette_max_size is defined as 31
else
    palette_max_size is defined as 15
```

[0164] 상기 표 12에 개시된 팔레트 사이즈 63, 31, 15 등과 CU 사이즈 1024, 256 등은 단지 예시로 사용된 것일 뿐이며, 다른 숫자로 변경될 수도 있다.

[0165] 일 실시예로, 팔레트 사이즈를 나타내는 정보(예: palette_max_size)는 다음 표 13에서와 같이 SPS를 통하여 시그널링될 수 있다.

표 13

[0166]

seq_parameter_set_rbsp( ) {	**Descriptor**
sps_decoding_parameter_set_id	u(4)
…..	
if(chroma_format_idc = = 3) {	
sps_palette_enabled_flag	u(1)
if (sps_palette_enabled_flag)	
palette_max_size	u(6)
}	
…	
}	

[0167] 상기 표 13의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 14와 같이 나타낼 수 있다.

표 14

[0168]

palette_max_size specifies the maximum allowed index of the palette table, and shall be in the range of 1 to 63 inclusive.

[0169] 상기 표 13 및 표 14를 참조하면, SPS에서 palette_max_size 신택스 요소가 파싱/시그널링될 수 있다. palette_max_size 신택스 요소는 팔레트 테이블의 최대 허용되는 인덱스를 나타낼 수 있고, 1에서 63 내의 범위로 제한될 수 있다.

[0170] 이때, palette_max_size 신택스 요소는 팔레트 모드가 가용한지(enabled) 여부를 지시하기 위한 정보인

sps_palette_enabled_flag 신택스 요소를 기반으로 파싱/시그널링될 수 있다. 예를 들어, sps_palette_enabled_flag의 값이 1인 경우(즉, 팔레트 모드가 가용한 것으로 지시하는 경우), palette_max_size 신택스 요소가 파싱/시그널링될 수 있다.

[0171] 또는, 일 실시예로, 팔레트 사이즈를 나타내는 정보(예: log2_palette_max_size)는 다음 표 15에서와 같이 SPS를 통하여 시그널링될 수 있다.

표 15

[0172]

seq_parameter_set_rbsp( ) {	**Descriptor**
sps_decoding_parameter_set_id	u(4)
…..	
if(chroma_format_idc = = 3) {	
sps_palette_enabled_flag	u(1)
if (sps_palette_enabled_flag)	
log2_palette_max_size	u(3)
}	
…	
}	

[0173] 상기 표 15의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 16에서와 같이 나타낼 수 있다.

표 16

[0174]

log2_palette_max_size specifies the log2 of (palette_max_size+1). palette_max_size=(1<< log2_palette_max_size)-1. palette_max_size specifies the the maximum allowed index of the palette table, and shall be in the range of 1 to 63 inclusive.

[0175] 상기 표 15 및 표 16을 참조하면, SPS에서 log2_palette_max_size 신택스 요소가 파싱/시그널링될 수 있다. log2_palette_max_size 신택스 요소는 팔레트 사이즈(즉, palette_max_size+1)의 log2 값을 나타낼 수 있다. 따라서, 팔레트 테이블의 최대 허용되는 인덱스를 나타내는 palette_max_size는, (1<< log2_palette_max_size)-1 로 계산하여 도출될 수 있고, 1에서 63 내의 범위로 제한될 수 있다.

[0176] 이때, log2_palette_max_size 신택스 요소는 팔레트 모드가 가용한지(enabled) 여부를 지시하기 위한 정보인 sps_palette_enabled_flag 신택스 요소를 기반으로 파싱/시그널링될 수 있다. 예를 들어, sps_palette_enabled_flag의 값이 1인 경우(즉, 팔레트 모드가 가용한 것으로 지시하는 경우), log2_palette_max_size 신택스 요소가 파싱/시그널링될 수 있다.

[0177] 또는, 일 실시예로, 팔레트 사이즈를 나타내는 정보(예: log2_palette_CU_size_TH1, log2_palette_max_size_TH1, log2_palette_max_size_default)는 다음 표 17에서와 같이 SPS를 통하여 시그널링될 수 있다.

표 17

seq_parameter_set_rbsp( ) {	Descriptor
sps_decoding_parameter_set_id	u(4)
…..	
if(chroma_format_idc = = 3) {	
sps_palette_enabled_flag	u(1)
if (sps_palette_enabled_flag) {	
log2_palette_CU_size_TH1	u(5)
log2_palette_max_size_TH1	u(3)
log2_palette_max_size_default	u(3)
}	
…	
}	

[0178]

[0179] 상기 표 17의 신택스에 포함된 신택스 요소(syntax elements)들에 대한 시맨틱스(semantics)는 다음 표 18에서와 같이 나타낼 수 있다.

표 18

log2_Palette_CU_size_TH1 specfies the log2 of the size limit of palette_max_size_TH1. Palette_CU_size_TH1=1<< log2_Palette_CU_size_TH1
log2_palette_max_size_TH1 specifies the log2 of (palette_max_size_TH1+1). palette_max_size_TH1=(1<< log2_palette_max_size_TH1)-1. palette_max_size_TH1 specifies the maximum allowed index of the palette table for CU with size greater than Palette_CU_size_TH1, and shall be in the range of 1 to 63 inclusive.
log2_palette_max_size_default specifies the log2 of (palette_max_size_default+1). palette_max_size_default=(1<< log2_palette_max_size_default)-1. palette_max_size_default specifies the maximum allowed index of the palette table, and shall be in the range of 1 to 63 inclusive.
Variable palette_max_size specifies the maximum allowed index of a palette table, and is derived as follows:
If (cbWidth*cbHeight >= Palette_CU_size_TH1)
palette_max_size = palette_max_size_TH1
else
palette_max_size = palette_max_size_default

[0180]

[0181] 상기 표 17 및 표 18을 참조하면, SPS에서 log2_palette_CU_size_TH1, log2_palette_max_size_TH1, log2_palette_max_size_default 신택스 요소들이 파싱/시그널링될 수 있다.

[0182] log2_palette_CU_size_TH1 신택스 요소는 palette_max_size_TH1의 사이즈 제한의 log2 값을 나타내고, palette_max_size_TH1는 1<< log2_Palette_CU_size_TH1로 도출될 수 있다.

[0183] log2_palette_max_size_TH1 신택스 요소는 (palette_max_size_TH1+1)의 log2 값을 나타내고, palette_max_size_TH1은 (1<< log2_palette_max_size_TH1)-1 로 도출될 수 있다. palette_max_size_TH1은 Palette_CU_size_TH1 보다 큰 사이즈를 가진 CU에 대해 팔레트 테이블의 최대 허용되는 인덱스를 나타내고, 1에서 63까지의 범위 내로 제한될 수 있다.

[0184] log2_palette_max_size_default 신택스 요소는 (palette_max_size_default+1)의 log2 값을 나타내고, palette_max_size_default는 (1<< log2_palette_max_size_default)-1 로 도출될 수 있다. palette_max_size_default는 팔레트 테이블의 최대 허용되는 인덱스를 나타내고, 1에서 63까지의 범위 내로 제한될 수 있다.

[0185] 이때, log2_palette_CU_size_TH1, log2_palette_max_size_TH1, log2_palette_max_size_default 신택스 요소들은 팔레트 모드가 가용한지(enabled) 여부를 지시하기 위한 정보인 sps_palette_enabled_flag 신택스 요소를 기반으로 파싱/시그널링될 수 있다. 예를 들어, sps_palette_enabled_flag의 값이 1인 경우(즉, 팔레트 모드가 가용한 것으로 지시하는 경우), log2_palette_CU_size_TH1, log2_palette_max_size_TH1, log2_palette_max_size_default 신택스 요소들이 파싱/시그널링될 수 있다.

[0186] 또한, palette_CU_size_TH 및 palette_max_size_TH의 하나 이상의 세트가 시그널링될 수 있고, palette_max_size를 나타내는데 사용될 수 있다.

[0187] 이하의 도면은 본 문서의 구체적인 일례를 설명하기 위해 작성되었다. 도면에 기재된 구체적인 장치의 명칭이나 구체적인 용어나 명칭(예컨대, 신택스/신택스 요소의 명칭 등)은 예시적으로 제시된 것이므로, 본 문서의 기술적 특징이 이하의 도면에 사용된 구체적인 명칭에 제한되지 않는다.

[0188] 도 9는 본 문서의 실시예(들)에 따른 비디오/영상 인코딩 방법의 일 예를 개략적으로 나타낸다.

[0189] 도 9에 개시된 방법은 도 2에서 개시된 인코딩 장치(200)에 의하여 수행될 수 있다. 구체적으로, 도 9의 단계 S900 ~ S910은 도 2에 개시된 예측부(220)에 의하여 수행될 수 있고, 도 9의 단계 S920은 도 2에 개시된 엔트로피 인코딩부(240)에 의하여 수행될 수 있다. 또한, 도 9에서 개시된 방법은 본 문서에서 상술한 실시예들을 포함하여 수행될 수 있다. 따라서, 도 9에서는 상술한 실시예들과 중복되는 내용에 관해서 구체적인 설명을 생략하거나 간단히 하기로 한다.

[0190] 도 9를 참조하면, 인코딩 장치는 현재 블록에 대한 팔레트 모드에서의 이스케이프 값을 도출할 수 있다(S900).

[0191] 일 실시예로, 인코딩 장치는 현재 블록에 대한 예측 모드를 결정하고 예측을 수행할 수 있다. 예를 들어, 인코딩 장치는 현재 블록에 인터 예측을 수행할지 또는 인트라 예측을 수행할지 여부를 결정할 수 있다. 또는 인코딩 장치는 현재 블록에 CIIP 모드, IBC 모드, 또는 팔레트 모드 등을 기반으로 예측을 수행할지 여부를 결정할 수 있다. 인코딩 장치는 RD 코스트 기반으로 예측 모드를 결정할 수 있다. 인코딩 장치는 결정된 예측 모드에 따라 예측을 수행하여 현재 블록에 대한 예측 샘플들을 도출할 수 있다. 또한, 인코딩 장치는 현재 블록에 적용된 예측과 관련된 정보(예컨대, 예측 모드 정보)를 생성하고 인코딩할 수 있다.

[0192] 현재 블록에 팔레트 모드 기반 예측을 수행하는 경우, 인코딩 장치는 상술한 실시예들에 개시된 팔레트 모드 코딩을 적용할 수 있다. 즉, 인코딩 장치는 현재 블록에 팔레트 모드 코딩을 적용하여, 팔레트 엔트리, 팔레트 인덱스, 이스케이프 값 등을 도출할 수 있다.

[0193] 일예로, 인코딩 장치는 현재 블록의 샘플 값들을 기반으로 팔레트 엔트리 정보를 생성할 수 있다. 즉, 인코딩 장치는 팔레트 테이블을 구성하기 위해 이전 팔레트 모드로 코딩된 블록에서 사용된 팔레트 예측자 엔트리 및 팔레트 엔트리 재사용 정보를 도출하고, 현재 블록에 대한 팔레트 엔트리를 도출할 수 있다. 예를 들어, 상기 표 5 및 표 6에서 개시된 바와 같이, 인코딩 장치는 팔레트 테이블을 구성하기 위해 사용되는 palette_predictor_run, num_signalled_palette_entries, new_palette_entries 등의 팔레트 엔트리 정보를 도출할 수 있다.

[0194] 또한, 인코딩 장치는 팔레트 엔트리 정보를 기반으로 현재 블록에 대한 팔레트 인덱스 정보를 생성할 수 있다. 즉, 인코딩 장치는 현재 블록의 샘플들을 트래버스 스캔 방향(수직 방향 혹은 수평 방향)에 따라 순회하면서 각 샘플의 팔레트 인덱스 값을 도출하고, 팔레트 인덱스 맵을 구성할 수 있다. 예를 들어, 상기 표 5 및 표 6에서 개시된 바와 같이, 인코딩 장치는 팔레트 인덱스 맵을 구성하기 위해 사용되는 palette_transpose_flag, palette_idx_idc, copy_above_indices_for_final_run_flag, num_palette_indices_minus1 등의 팔레트 엔트리 정보를 도출할 수 있다.

[0195] 여기서, 팔레트 테이블은 현재 블록 내의 샘플들에 대한 대표 컬러 값들(팔레트 엔트리들)을 포함하고, 각 컬러 값에 대응하는 팔레트 인덱스 값으로 구성될 수 있다. 즉, 인코딩 장치는 현재 블록 내의 각 샘플에 대해 팔레트 테이블 내의 엔트리(컬러 값)에 대응하는 팔레트 인덱스 값을 도출함으로써, 이를 디코딩 장치로 시그널링할 수 있다.

[0196] 인코딩 장치는 상기 팔레트 엔트리 정보 및 팔레트 인덱스 정보를 포함하는 영상 정보를 인코딩하고, 이를 디코딩 장치로 시그널링할 수 있다.

[0197] 또한, 현재 블록에 팔레트 모드 기반 예측을 수행함에 있어서, 인코딩 장치는 적어도 하나의 이스케이프 코딩된 샘플을 포함하는 현재 블록에 대하여 이스케이프 값을 도출할 수 있다.

[0198] 상술한 바와 같이 팔레트 모드에서 현재 블록 내 주변 샘플들과 특성이 다른 값을 갖는 샘플에 대해 별도로 해당 샘플 값을 전송해 주는 것이 코딩 효율 측면에서 효과적이기 때문에, 이 샘플 값을 이스케이프 값으로 시그널링할 수 있다. 이때, 이스케이프 값은 추가적인 데이터이므로 이를 절약하기 위해 양자화를 할 수 있다. 또한 팔레트 모드의 이스케이프 값은 변환이 적용되지 않고 양자화된 값이 직접적으로 시그널링 될 수 있다.

[0199] 인코딩 장치는 이스케이프 값을 기반으로 양자화된 이스케이프 값을 도출할 수 있다(S910).

[0200] 일 실시예로, 인코딩 장치는 이스케이프 값에 대한 양자화 파라미터를 기반으로 상기 이스케이프 값에 적용하여 양자화된 이스케이프 값을 도출할 수 있다.

[0201] 여기서, 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출될 수 있다. 예를 들어, 양자화 파라미터는 상기 표 7 내지 표 9에 개시된 변환 스킵 모드에 대한 최소 양자화 파라미터 정보(예: min_qp_prime_ts_minus4)를 기반으로 도출될 수 있다. 상술한 바와 같이 팔레트 모드의 이스케이프 값은 변환이 적용되지 않기 때문에, 변환 스킵 모드에서 사용되는 최소 양자화 파라미터 정보를 기반으로 양자화할 수 있다.

[0202] 구체적 예로, 상술한 표 9에서와 같이, 먼저 인코딩 장치는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보(예: min_qp_prime_ts_minus4)를 기반으로 최소 양자화 파라미터 값(예: QpPrimeTsMin)을 도출할 수 있다. 그리고, 인코딩 장치는 최소 양자화 파라미터 값(예: QpPrimeTsMin) 및 양자화 파라미터 Qp(루마 성분인 경우 Qp'Y, 크로마 성분인 경우 Qp'Cb 또는 Qp'Cr) 중에서 큰 값을 선택하여, 이를 팔레트 모드에서의 양자화 파라미터로 사용할 수 있다.

[0203] 다시 말해, 팔레트 모드에서의 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보(예: min_qp_prime_ts_minus4)로부터 도출되는 최소 양자화 파라미터 값(예: QpPrimeTsMin) 보다 크거나 같은 값을 가질 수 있다.

[0204] 인코딩 장치는 상기와 같이 도출된 팔레트 모드에서의 양자화 파라미터를 적용하여 양자화된 이스케이프 값을 도출할 수 있다. 인코딩 장치는 양자화된 이스케이프 값을 상기 표 5 및 표 6에서 개시된 바와 같이, palette_escape_val 신택스 요소로 생성하고, 이를 시그널링할 수 있다. 또한, 인코딩 장치는 현재 블록 내 이스케이프 값을 갖는 샘플이 존재함을 지시하기 위한 정보(예: palette_escape_val_present_flag)를 생성하고, 이를 시그널링할 수 있다.

[0205] 또한, 일 실시예에 있어서, 인코딩 장치는 양자화된 이스케이프 값을 특정 범위로 제한할 수 있다. 이스케이프 값은 주변 샘플들과는 특성이 다른 값을 갖기 때문에 양자화하여 직접적으로 시그널링하는데, 양자화에 의한 오차가 발생할 수 있다. 이러한 오차를 줄이고 보다 정확한 값을 코딩할 수 있도록 하기 위해 양자화된 이스케이프 값의 범위를 비트심도(bitdepth)에 기반하여 제한할 수 있다.

[0206] 예를 들어, 양자화된 이스케이프 값에 대한 정보의 범위는 상기 표 10 및 표 11에 개시된 바와 같이 비트심도를 기반으로 결정될 수 있고, 예컨대 (1 << BitDepth) - 1 보다 크지 않도록 제한될 수 있다. 또한, 비트심도는 루마 성분에 대한 비트심도 $BitDepth_Y$ 및 크로마 성분에 대한 비트심도 $BitDepth_C$를 포함할 수 있다. 이때, 루마 성분에 대한 양자화된 이스케이프 값 정보의 범위는 0에서 $(1 << BitDepth_Y) - 1$ 사이의 값을 갖고, 크로마 성분에 대한 상기 양자화된 이스케이프 값 정보의 범위는 0에서 $(1 << BitDepth_C) - 1$ 사이의 값을 가질 수 있다.

[0207] 또한, 일 실시예에 있어서, 인코딩 장치는 팔레트 테이블의 엔트리 개수(즉, 팔레트 테이블의 인덱스 개수)를 정의하고 이를 디코딩 장치로 시그널링할 수 있다. 즉, 인코딩 장치는 팔레트 테이블의 최대 인덱스에 관한 팔레트 사이즈 정보를 결정하고, 이를 시그널링할 수 있다. 팔레트 사이즈 정보는 기설정된 값이 거나 또는 코딩 유닛의 사이즈를 기반으로 결정될 수 있다.

[0208] 예를 들어, 팔레트 사이즈는 상기 표 12에 개시된 바와 같이 palette_max_size로 나타낼 수 있고, 전체 시퀀스에 대해 동일하거나 또는 CU 사이즈(즉, CU 내의 픽셀 개수)에 따라 상이하게 정해질 수 있다.

[0209] 또한, 예를 들어, 팔레트 사이즈는 상기 표 13 및 표 14에 개시된 바와 같이 palette_max_size로 나타낼 수 있고, SPS를 통하여 시그널링될 수 있다. 이때, 팔레트 사이즈(예: palette_max_size)는 팔레트 테이블의 최대 허용되는 인덱스를 나타낼 수 있고, 1에서 63 내의 범위로 제한될 수 있다. 또한, 팔레트 사이즈(예: palette_max_size)는 팔레트 모드가 가용한지(enabled) 여부를 지시하기 위한 정보(예: sps_palette_enabled_flag)를 기반으로 시그널링될 수도 있다.

[0210] 또한, 예를 들어, 팔레트 사이즈는 상기 표 15 및 표 16에 개시된 바와 같이 log2_palette_max_size로 나타낼 수 있고, SPS를 통하여 시그널링될 수 있다. 이때, 팔레트 사이즈(예: log2_palette_max_size)는 팔레트 사이즈(즉, palette_max_size+1)의 log2 값을 나타낼 수 있다. 따라서, 팔레트 테이블의 최대 허용되는 인덱스를 나타내는 palette_max_size는, (1<< log2_palette_max_size)-1 로 계산하여 도출될 수 있고, 1에서 63 내의 범위로 제한될 수 있다. 또한, 팔레트 사이즈(예: log2_palette_max_size)는 팔레트 모드가 가용한지(enabled) 여부를 지시하기 위한 정보(예: sps_palette_enabled_flag)를 기반으로 시그널링될 수도 있다.

[0211] 또한, 예를 들어, 팔레트 사이즈는 상기 표 17 및 표 18에 개시된 바와 같이 log2_palette_CU_size_TH1, log2_palette_max_size_TH1, log2_palette_max_size_default를 기반으로 도출될 수 있고, SPS를 통하여 시그널링될 수 있다. 팔레트 사이즈를 도출하고 시그널링하는 구체적인 실시예는 상기 표 17 및 표 18에서 상술한 바 있으므로, 여기서는 설명을 생략하도록 한다.

[0212] 인코딩 장치는 영상 정보(또는 비디오 정보)를 인코딩할 수 있다(S920). 여기서, 영상 정보는 상술한 팔레트 모드 코딩을 위해 사용되는 다양한 정보를 포함할 수 있다.

[0213] 일예로, 인코딩 장치는 양자화된 이스케이프 값에 대한 정보를 포함하는 영상 정보를 생성하고 인코딩할 수 있다. 또한, 인코딩 장치는 팔레트 엔트리 정보 및 팔레트 인덱스 정보를 포함하는 영상 정보를 생성하고 인코딩할 수 있다. 또한, 인코딩 장치는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 포함하는 영상 정보를 생성하고 인코딩할 수 있다. 이때, 영상 정보는 SPS를 포함할 수 있고, SPS는 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 포함할 수 있다.

[0214] 또한, 실시예에 따라, 인코딩 장치는 팔레트 모드의 가용 여부에 관한 정보를 기반으로, 상술한 팔레트 모드를 사용하여 현재 블록에 대한 코딩을 수행할지를 결정할 수 있다.

[0215] 예를 들어, 상기 표 1 내지 표 4에 개시된 바와 같이, 인코딩 장치는 팔레트 모드의 가용 여부를 결정하고, 상기 결정에 따라 팔레트 모드의 가용 여부에 관한 정보(예: sps_palette_enabled_flag)를 생성하고 SPS를 통하여 시그널링할 수 있다.

[0216] 또한, 인코딩 장치는 팔레트 모드의 가용 여부에 관한 정보(예: sps_palette_enabled_flag)를 기반으로, 현재 블록에 팔레트 모드를 적용하여 코딩할지 여부를 나타내는 정보(예: pred_mode_plt_flag)를 생성하고 코딩 유닛 신택스를 통하여 시그널링할 수 있다. 예를 들어, pred_mode_plt_flag의 값이 1인 경우 palette_coding() 신택스를 시그널링함으로써, 현재 블록에 대해 팔레트 모드를 적용하여 복원 샘플을 생성할 수 있다.

[0217] 상기 팔레트 모드의 가용 여부에 관한 정보(예: sps_palette_enabled_flag), 현재 블록에 팔레트 모드를 적용하여 코딩할지 여부를 나타내는 정보(예: pred_mode_plt_flag)는 영상 정보에 포함되어 인코딩될 수 있다.

[0218] 상술한 바와 같은 다양한 정보를 포함하는 영상 정보는 인코딩되어 비트스트림 형태로 출력될 수 있다. 비트스트림은 네트워크 또는 (디지털) 저장매체를 통하여 디코딩 장치로 전송될 수 있다. 여기서 네트워크는 방송망 및/또는 통신망 등을 포함할 수 있고, 디지털 저장매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장매체를 포함할 수 있다.

[0219] 도 10은 본 문서의 실시예(들)에 따른 비디오/영상 디코딩 방법의 일 예를 개략적으로 나타낸다.

[0220] 도 10에 개시된 방법은 도 3에서 개시된 디코딩 장치(300)에 의하여 수행될 수 있다. 구체적으로, 도 10의 단계 S1000은 도 3에 개시된 엔트로피 디코딩부(310)에 의하여 수행될 수 있고, 도 10의 단계 S1010 ~ S1020은 도 3에 개시된 예측부(330)에 의하여 수행될 수 있다. 또한, 도 10에서 개시된 방법은 본 문서에서 상술한 실시예들을 포함하여 수행될 수 있다. 따라서, 도 10에서는 상술한 실시예들과 중복되는 내용에 관해서 구체적인 설명을 생략하거나 간단히 하기로 한다.

[0221] 도 10을 참조하면, 디코딩 장치는 비트스트림으로부터 영상 정보(또는 비디오 정보)를 수신할 수 있다(S1000).

[0222] 디코딩 장치는 비트스트림을 파싱하여 영상 복원(또는 픽처 복원)에 필요한 정보(ex. 비디오/영상 정보)를 도출

할 수 있다. 이때, 영상 정보는 예측과 관련된 정보(예컨대, 예측 모드 정보)를 포함할 수 있다. 또한, 영상 정보는 상술한 팔레트 모드 코딩을 위해 사용되는 다양한 정보를 포함할 수 있다. 예를 들어, 영상 정보는 양자화된 이스케이프 값에 대한 정보, 팔레트 엔트리 정보, 팔레트 인덱스 정보, 변환 스킵 모드에 대한 최소 양자화 파라미터 정보 등을 포함할 수 있다. 즉, 영상 정보는 디코딩 과정에서 필요한 다양한 정보를 포함할 수 있고, 지수 골롬 부호화, CAVLC 또는 CABAC 등의 코딩 방법을 기초로 디코딩될 수 있다.

[0223] 일 실시예로, 디코딩 장치는 비트스트림으로부터 팔레트 모드에서의 양자화된 이스케이프 값 정보를 포함하는 영상 정보를 획득할 수 있다. 예를 들어, 양자화된 이스케이프 값 정보는 상기 표 5 및 표 6에서 개시된 바와 같이, palette_escape_val 신택스 요소일 수 있다. 이때, 양자화된 이스케이프 값 정보(예: palette_escape_val)은 현재 블록 내 이스케이프 값을 갖는 샘플이 존재하는지 여부를 나타내는 정보(예: palette_escape_val_present_flag)를 기반으로 획득될 수 있다. 예를 들어, 현재 블록 내 이스케이프 값을 갖는 샘플이 존재하는 경우(즉, palette_escape_val_present_flag의 값이 1인 경우), 디코딩 장치는 비트스트림으로부터 양자화된 이스케이프 값 정보(예: palette_escape_val)를 획득할 수 있다.

[0224] 디코딩 장치는 양자화된 이스케이프 값 정보를 기반으로 현재 블록에 대한 이스케이프 값을 도출할 수 있다(S1010).

[0225] 일 실시예로, 디코딩 장치는 양자화 파라미터를 기반으로 양자화된 이스케이프 값에 역양자화(스케일링 과정)을 수행하여 이스케이프 값을 도출할 수 있다.

[0226] 여기서, 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출될 수 있다. 예를 들어, 양자화 파라미터는 상기 표 7 내지 표 9에 개시된 변환 스킵 모드에 대한 최소 양자화 파라미터 정보(예: min_qp_prime_ts_minus4)를 기반으로 도출될 수 있다. 상술한 바와 같이 팔레트 모드의 이스케이프 값은 변환이 적용되지 않기 때문에, 변환 스킵 모드에서 사용되는 최소 양자화 파라미터 정보를 기반으로 양자화할 수 있다. 이때, 변환 스킵 모드에 대한 최소 양자화 파라미터 정보(예: min_qp_prime_ts_minus4)는 SPS로부터 파싱/시그널링될 수 있다.

[0227] 구체적 예로, 상술한 표 9에서와 같이, 먼저 디코딩 장치는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보(예: min_qp_prime_ts_minus4)를 기반으로 최소 양자화 파라미터 값(예: QpPrimeTsMin)을 도출할 수 있다. 그리고, 디코딩 장치는 최소 양자화 파라미터 값(예: QpPrimeTsMin) 및 양자화 파라미터 Qp(루마 성분인 경우 Qp'Y, 크로마 성분인 경우 Qp'Cb 또는 Qp'Cr) 중에서 큰 값을 선택하여, 이를 팔레트 모드에서의 양자화 파라미터로 사용할 수 있다.

[0228] 다시 말해, 팔레트 모드에서의 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보(예: min_qp_prime_ts_minus4)로부터 도출되는 최소 양자화 파라미터 값(예: QpPrimeTsMin) 보다 크거나 같은 값을 가질 수 있다.

[0229] 디코딩 장치는 상기와 같이 도출된 팔레트 모드에서의 양자화 파라미터를 기반으로 양자화된 이스케이프 값으로부터 이스케이프 값을 도출할 수 있다.

[0230] 또한, 일 실시예에 있어서, 디코딩 장치는 양자화된 이스케이프 값을 특정 범위로 제한할 수 있다. 이스케이프 값은 주변 샘플들과는 특성이 다른 값을 갖기 때문에 양자화하여 직접적으로 시그널링하는데, 양자화에 의한 오차가 발생할 수 있다. 이러한 오차를 줄이고 보다 정확한 값을 코딩할 수 있도록 하기 위해 양자화된 이스케이프 값의 범위를 비트심도(bitdepth)에 기반하여 제한할 수 있다.

[0231] 예를 들어, 양자화된 이스케이프 값에 대한 정보의 범위는 상기 표 10 및 표 11에 개시된 바와 같이 비트심도를 기반으로 결정될 수 있고, 예컨대 (1 << BitDepth) - 1 보다 크지 않도록 제한될 수 있다. 또한, 비트심도는 루마 성분에 대한 비트심도 $BitDepth_Y$ 및 크로마 성분에 대한 비트심도 $BitDepth_C$를 포함할 수 있다. 이때, 루마 성분에 대한 양자화된 이스케이프 값 정보의 범위는 0에서 $(1 << BitDepth_Y) - 1$ 사이의 값을 갖고, 크로마 성분에 대한 상기 양자화된 이스케이프 값 정보의 범위는 0에서 $(1 << BitDepth_C) - 1$ 사이의 값을 가질 수 있다.

[0232] 또한, 일 실시예에 있어서, 디코딩 장치는 팔레트 테이블의 엔트리 개수(즉, 팔레트 테이블의 인덱스 개수)를 포함하는 영상 정보를 획득할 수 있다. 즉, 디코딩 장치는 팔레트 테이블의 최대 인덱스에 관한 팔레트 사이즈 정보를 포함하는 영상 정보를 획득할 수 있다. 이때, 팔레트 사이즈 정보는 기설정된 값이 거나 또는 코딩 유닛의 사이즈를 기반으로 결정될 수 있다.

[0233] 예를 들어, 팔레트 사이즈는 상기 표 12에 개시된 바와 같이 palette_max_size로 나타낼 수 있고, 전체 시퀀스에 대해 동일하거나 또는 CU 사이즈(즉, CU 내의 픽셀 개수)에 따라 상이하게 정해질 수 있다.

[0234] 또한, 예를 들어, 팔레트 사이즈는 상기 표 13 및 표 14에 개시된 바와 같이 palette_max_size로 나타낼 수 있고, SPS를 통하여 파싱/시그널링될 수 있다. 이때, 팔레트 사이즈(예: palette_max_size)는 팔레트 테이블의 최대 허용되는 인덱스를 나타낼 수 있고, 1에서 63 내의 범위로 제한될 수 있다. 또한, 팔레트 사이즈(예: palette_max_size)는 팔레트 모드가 가용한지(enabled) 여부를 지시하기 위한 정보(예: sps_palette_enabled_flag)를 기반으로 파싱/시그널링될 수도 있다.

[0235] 또한, 예를 들어, 팔레트 사이즈는 상기 표 15 및 표 16에 개시된 바와 같이 log2_palette_max_size로 나타낼 수 있고, SPS를 통하여 파싱/시그널링될 수 있다. 이때, 팔레트 사이즈(예: log2_palette_max_size)는 팔레트 사이즈(즉, palette_max_size+1)의 log2 값을 나타낼 수 있다. 따라서, 팔레트 테이블의 최대 허용되는 인덱스를 나타내는 palette_max_size는, (1<< log2_palette_max_size)-1 로 계산하여 도출될 수 있고, 1에서 63 내의 범위로 제한될 수 있다. 또한, 팔레트 사이즈(예: log2_palette_max_size)는 팔레트 모드가 가용한지(enabled) 여부를 지시하기 위한 정보(예: sps_palette_enabled_flag)를 기반으로 파싱/시그널링될 수도 있다.

[0236] 또한, 예를 들어, 팔레트 사이즈는 상기 표 17 및 표 18에 개시된 바와 같이 log2_palette_CU_size_TH1, log2_palette_max_size_TH1, log2_palette_max_size_default를 기반으로 도출될 수 있고, SPS를 통하여 파싱/시그널링될 수 있다. 팔레트 사이즈를 도출하고 파싱/시그널링하는 구체적인 실시예는 상기 표 17 및 표 18에서 상술한 바 있으므로, 여기서는 설명을 생략하도록 한다.

[0237] 디코딩 장치는 이스케이프 값을 기반으로 복원 샘플들을 생성할 수 있다(S1020).

[0238] 일 실시예로, 디코딩 장치는 적어도 하나의 이스케이프 코딩된 샘플을 포함하는 현재 블록에 대하여, 이스케이프 값을 기반으로 복원 샘플들을 생성할 수 있다. 예를 들어, 현재 블록 내 이스케이프 값을 갖는 샘플이 존재하는 경우(즉, palette_escape_val_present_flag의 값이 1인 경우), 디코딩 장치는 상술한 바와 같이 이스케이프 값을 도출하여 이스케이프 코딩된 샘플의 복원 샘플을 생성할 수 있다.

[0239] 또한, 현재 블록에 팔레트 모드 기반 예측을 수행함에 있어서(즉, 현재 블록에 대해 팔레트 모드가 적용되는 경우), 현재 블록 내의 이스케이프 코딩된 샘플이 아닌 샘플들에 대해서, 디코딩 장치는 팔레트 엔트리 정보 및 팔레트 인덱스 정보를 포함하는 영상 정보를 획득하고, 이를 기반으로 복원 샘플들을 생성할 수 있다.

[0240] 일예로, 디코딩 장치는 팔레트 엔트리 정보를 기반으로 현재 블록에 대한 팔레트 테이블을 구성할 수 있다. 예를 들어, 팔레트 엔트리 정보는 상기 표 5 및 표 6에서 개시된 바와 같이, palette_predictor_run, num_signalled_palette_entries, new_palette_entries 등을 포함할 수 있다. 즉, 디코딩 장치는 팔레트 테이블을 구성하기 위해 이전 팔레트 모드로 코딩된 블록에서 사용된 팔레트 예측자 엔트리 및 팔레트 엔트리 재사용 정보를 도출하고, 현재 블록에 대한 팔레트 엔트리를 도출할 수 있다. 그리고 디코딩 장치는 이전 팔레트 예측자 엔트리와 현재 팔레트 엔트리를 기반으로 팔레트 테이블을 구성할 수 있다.

[0241] 또한, 디코딩 장치는 팔레트 인덱스 정보를 기반으로 현재 블록에 대한 팔레트 인덱스 맵을 구성할 수 있다. 예를 들어, 팔레트 인덱스 정보는 상기 표 5 및 표 6에서 개시된 바와 같이, 팔레트 인덱스 맵을 구성하기 위해 사용되는 palette_transpose_flag, palette_idx_idc, copy_above_indices_for_final_run_flag, num_palette_indices_minus1 등을 포함할 수 있다. 즉, 디코딩 장치는 트래버스 스캔 방향(수직 방향 혹은 수평 방향)을 나타내는 정보(예: palette_transpose_flag)를 기반으로 현재 블록의 샘플들을 순회하면서 각 샘플의 팔레트 인덱스 값을 나타내는 정보(예: palette_idx_idc)를 기반으로 팔레트 인덱스 맵(예: PaletteIndexMap)을 구성할 수 있다.

[0242] 또한, 디코딩 장치는 팔레트 인덱스 맵을 기반으로 팔레트 테이블 내의 팔레트 엔트리에 대한 샘플 값을 도출할 수 있다. 그리고 디코딩 장치는 팔레트 인덱스 맵 및 팔레트 엔트리에 대한 샘플 값을 기반으로 복원 샘플들을 생성할 수 있다.

[0243] 여기서, 팔레트 테이블은 현재 블록 내의 샘플들에 대한 대표 컬러 값들(팔레트 엔트리들)을 포함하고, 각 컬러 값에 대응하는 팔레트 인덱스 값으로 구성될 수 있다. 따라서, 디코딩 장치는 팔레트 인덱스 맵의 인덱스 값에 대응하는 팔레트 테이블 내의 엔트리의 샘플 값(즉, 컬러 값)을 도출하고, 이를 현재 블록의 복원 샘플 값으로 생성할 수 있다.

[0244] 또한, 실시예에 따라, 디코딩 장치는 팔레트 모드의 가용 여부에 관한 정보를 기반으로, 상술한 팔레트 모드를

사용하여 현재 블록에 대한 코딩을 수행할지를 결정할 수 있다.

[0245] 예를 들어, 상기 표 1 내지 표 4에 개시된 바와 같이, 디코딩 장치는 팔레트 모드의 가용 여부에 관한 정보(예: sps_palette_enabled_flag)를 포함하는 영상 정보를 획득하고, 이 정보를 기반으로 상기 팔레트 엔트리 정보, 펠레트 인덱스 정보, 양자화된 이스케이프 값 정보 등을 비트스트림으로부터 획득할 수 있다.

[0246] 또한, 예를 들어, 디코딩 장치는 팔레트 모드의 가용 여부에 관한 정보(예: sps_palette_enabled_flag)를 기반으로, 현재 블록에 팔레트 모드를 적용하여 코딩할지 여부를 나타내는 정보(예: pred_mode_plt_flag)를 비트스트림으로부터 획득할 수 있다. 예를 들어, pred_mode_plt_flag의 값이 1인 경우, 디코딩 장치는 palette_coding() 신택스를 더 획득할 수 있고, 상기 palette_coding() 신택스에 포함된 정보를 기반으로 현재 블록에 대해 팔레트 모드를 적용하여 복원 샘플을 도출할 수 있다.

[0247] 상술한 실시예에서, 방법들은 일련의 단계 또는 블록으로써 순서도를 기초로 설명되고 있지만, 본 문서의 실시예들은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 당업자라면 순서도에 나타내어진 단계들이 배타적이지 않고, 다른 단계가 포함되거나 순서도의 하나 또는 그 이상의 단계가 본 문서의 범위에 영향을 미치지 않고 삭제될 수 있음을 이해할 수 있을 것이다.

[0248] 상술한 본 문서에 따른 방법은 소프트웨어 형태로 구현될 수 있으며, 본 문서에 따른 인코딩 장치 및/또는 디코딩 장치는 예를 들어 TV, 컴퓨터, 스마트폰, 셋톱박스, 디스플레이 장치 등의 영상 처리를 수행하는 장치에 포함될 수 있다.

[0249] 본 문서에서 실시예들이 소프트웨어로 구현될 때, 상술한 방법은 상술한 기능을 수행하는 모듈(과정, 기능 등)로 구현될 수 있다. 모듈은 메모리에 저장되고, 프로세서에 의해 실행될 수 있다. 메모리는 프로세서 내부 또는 외부에 있을 수 있고, 잘 알려진 다양한 수단으로 프로세서와 연결될 수 있다. 프로세서는 ASIC(application-specific integrated circuit), 다른 칩셋, 논리 회로 및/또는 데이터 처리 장치를 포함할 수 있다. 메모리는 ROM(read-only memory), RAM(random access memory), 플래쉬 메모리, 메모리 카드, 저장 매체 및/또는 다른 저장 장치를 포함할 수 있다. 즉, 본 문서에서 설명한 실시예들은 프로세서, 마이크로 프로세서, 컨트롤러 또는 칩 상에서 구현되어 수행될 수 있다. 예를 들어, 각 도면에서 도시한 기능 유닛들은 컴퓨터, 프로세서, 마이크로 프로세서, 컨트롤러 또는 칩 상에서 구현되어 수행될 수 있다. 이 경우 구현을 위한 정보(ex. information on instructions) 또는 알고리즘이 디지털 저장 매체에 저장될 수 있다.

[0250] 또한, 본 문서가 적용되는 디코딩 장치 및 인코딩 장치는 멀티미디어 방송 송수신 장치, 모바일 통신 단말, 홈시네마 비디오 장치, 디지털 시네마 비디오 장치, 감시용 카메라, 비디오 대화 장치, 비디오 통신과 같은 실시간 통신 장치, 모바일 스트리밍 장치, 저장 매체, 캠코더, 주문형 비디오(VoD) 서비스 제공 장치, OTT 비디오(Over the top video) 장치, 인터넷 스트리밍 서비스 제공 장치, 3차원(3D) 비디오 장치, VR(virtual reality) 장치, AR(argumente reality) 장치, 화상 전화 비디오 장치, 운송 수단 단말 (ex. 차량(자율주행차량 포함) 단말, 비행기 단말, 선박 단말 등) 및 의료용 비디오 장치 등에 포함될 수 있으며, 비디오 신호 또는 데이터 신호를 처리하기 위해 사용될 수 있다. 예를 들어, OTT 비디오(Over the top video) 장치로는 게임 콘솔, 블루레이 플레이어, 인터넷 접속 TV, 홈시어터 시스템, 스마트폰, 태블릿 PC, DVR(Digital Video Recoder) 등을 포함할 수 있다.

[0251] 또한, 본 문서의 실시예(들)이 적용되는 처리 방법은 컴퓨터로 실행되는 프로그램의 형태로 생산될 수 있으며, 컴퓨터가 판독할 수 있는 기록 매체에 저장될 수 있다. 본 문서의 실시예(들)에 따른 데이터 구조를 가지는 멀티미디어 데이터도 또한 컴퓨터가 판독할 수 있는 기록 매체에 저장될 수 있다. 상기 컴퓨터가 판독할 수 있는 기록 매체는 컴퓨터로 읽을 수 있는 데이터가 저장되는 모든 종류의 저장 장치 및 분산 저장 장치를 포함한다. 상기 컴퓨터가 판독할 수 있는 기록 매체는, 예를 들어, 블루레이 디스크(BD), 범용 직렬 버스(USB), ROM, PROM, EPROM, EEPROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크 및 광학적 데이터 저장 장치를 포함할 수 있다. 또한, 상기 컴퓨터가 판독할 수 있는 기록 매체는 반송파(예를 들어, 인터넷을 통한 전송)의 형태로 구현된 미디어를 포함한다. 또한, 인코딩 방법으로 생성된 비트스트림이 컴퓨터가 판독할 수 있는 기록 매체에 저장되거나 유무선 통신 네트워크를 통해 전송될 수 있다.

[0252] 또한, 본 문서의 실시예(들)는 프로그램 코드에 의한 컴퓨터 프로그램 제품으로 구현될 수 있고, 상기 프로그램 코드는 본 문서의 실시예(들)에 의해 컴퓨터에서 수행될 수 있다. 상기 프로그램 코드는 컴퓨터에 의해 판독가능한 캐리어 상에 저장될 수 있다.

[0253] 도 11은 본 문서에서 개시된 실시예들이 적용될 수 있는 컨텐츠 스트리밍 시스템의 예를 나타낸다.

[0254] 도 11을 참조하면, 본 문서의 실시예들에 적용되는 컨텐츠 스트리밍 시스템은 크게 인코딩 서버, 스트리밍 서버, 웹 서버, 미디어 저장소, 사용자 장치 및 멀티미디어 입력 장치를 포함할 수 있다.

[0255] 상기 인코딩 서버는 스마트폰, 카메라, 캠코더 등과 같은 멀티미디어 입력 장치들로부터 입력된 컨텐츠를 디지털 데이터로 압축하여 비트스트림을 생성하고 이를 상기 스트리밍 서버로 전송하는 역할을 한다. 다른 예로, 스마트폰, 카메라, 캠코더 등과 같은 멀티미디어 입력 장치들이 비트스트림을 직접 생성하는 경우, 상기 인코딩 서버는 생략될 수 있다.

[0256] 상기 비트스트림은 본 문서의 실시예들에 적용되는 인코딩 방법 또는 비트스트림 생성 방법에 의해 생성될 수 있고, 상기 스트리밍 서버는 상기 비트스트림을 전송 또는 수신하는 과정에서 일시적으로 상기 비트스트림을 저장할 수 있다.

[0257] 상기 스트리밍 서버는 웹 서버를 통한 사용자 요청에 기초하여 멀티미디어 데이터를 사용자 장치에 전송하고, 상기 웹 서버는 사용자에게 어떠한 서비스가 있는지를 알려주는 매개체 역할을 한다. 사용자가 상기 웹 서버에 원하는 서비스를 요청하면, 상기 웹 서버는 이를 스트리밍 서버에 전달하고, 상기 스트리밍 서버는 사용자에게 멀티미디어 데이터를 전송한다. 이때, 상기 컨텐츠 스트리밍 시스템은 별도의 제어 서버를 포함할 수 있고, 이 경우 상기 제어 서버는 상기 컨텐츠 스트리밍 시스템 내 각 장치 간 명령/응답을 제어하는 역할을 한다.

[0258] 상기 스트리밍 서버는 미디어 저장소 및/또는 인코딩 서버로부터 컨텐츠를 수신할 수 있다. 예를 들어, 상기 인코딩 서버로부터 컨텐츠를 수신하게 되는 경우, 상기 컨텐츠를 실시간으로 수신할 수 있다. 이 경우, 원활한 스트리밍 서비스를 제공하기 위하여 상기 스트리밍 서버는 상기 비트스트림을 일정 시간동안 저장할 수 있다.

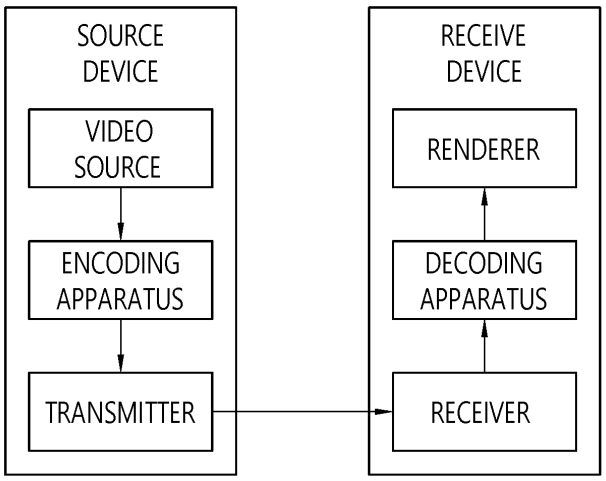
[0259] 상기 사용자 장치의 예로는, 휴대폰, 스마트 폰(smart phone), 노트북 컴퓨터(laptop computer), 디지털방송용 단말기, PDA(personal digital assistants), PMP(portable multimedia player), 네비게이션, 슬레이트 PC(slate PC), 태블릿 PC(tablet PC), 울트라북(ultrabook), 웨어러블 디바이스(wearable device, 예를 들어, 워치형 단말기 (smartwatch), 글래스형 단말기 (smart glass), HMD(head mounted display)), 디지털 TV, 데스크탑 컴퓨터, 디지털 사이니지 등이 있을 수 있다.

[0260] 상기 컨텐츠 스트리밍 시스템 내 각 서버들은 분산 서버로 운영될 수 있으며, 이 경우 각 서버에서 수신하는 데이터는 분산 처리될 수 있다.

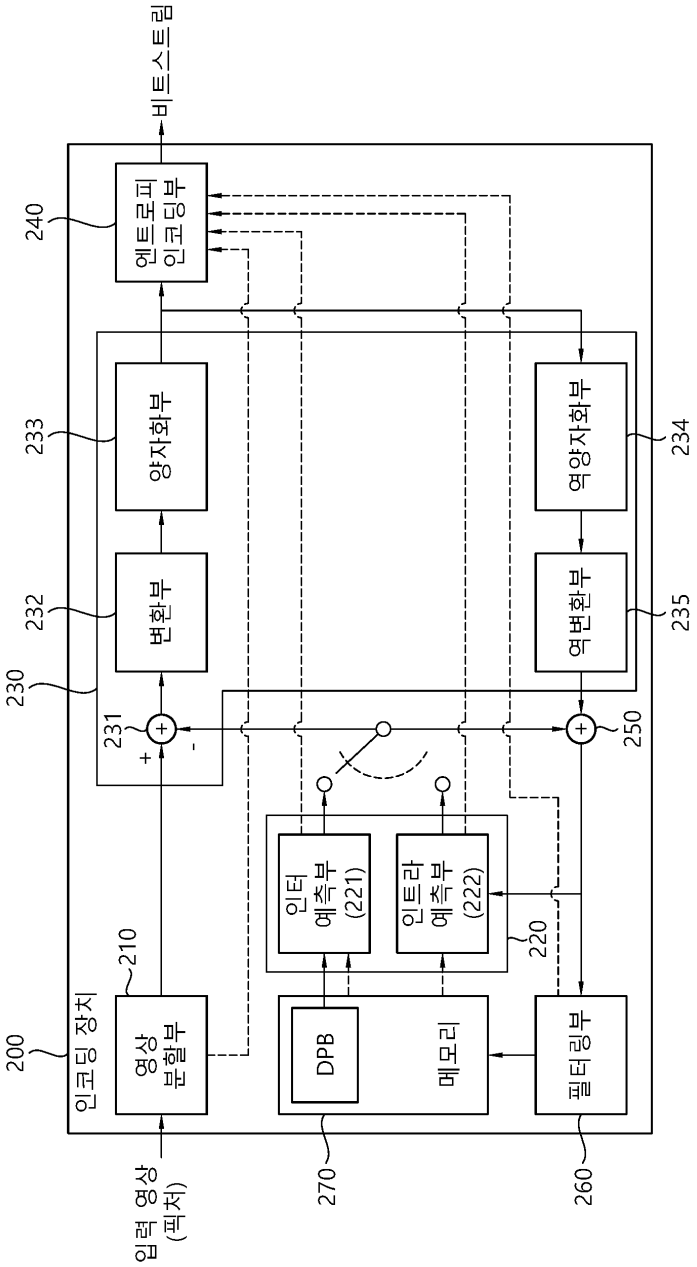
[0261] 본 문서에 기재된 청구항들은 다양한 방식으로 조합될 수 있다. 예를 들어, 본 문서의 방법 청구항의 기술적 특징이 조합되어 장치로 구현될 수 있고, 본 문서의 장치 청구항의 기술적 특징이 조합되어 방법으로 구현될 수 있다. 또한, 본 문서의 방법 청구항의 기술적 특징과 장치 청구항의 기술적 특징이 조합되어 장치로 구현될 수 있고, 본 문서의 방법 청구항의 기술적 특징과 장치 청구항의 기술적 특징이 조합되어 방법으로 구현될 수 있다.

도면

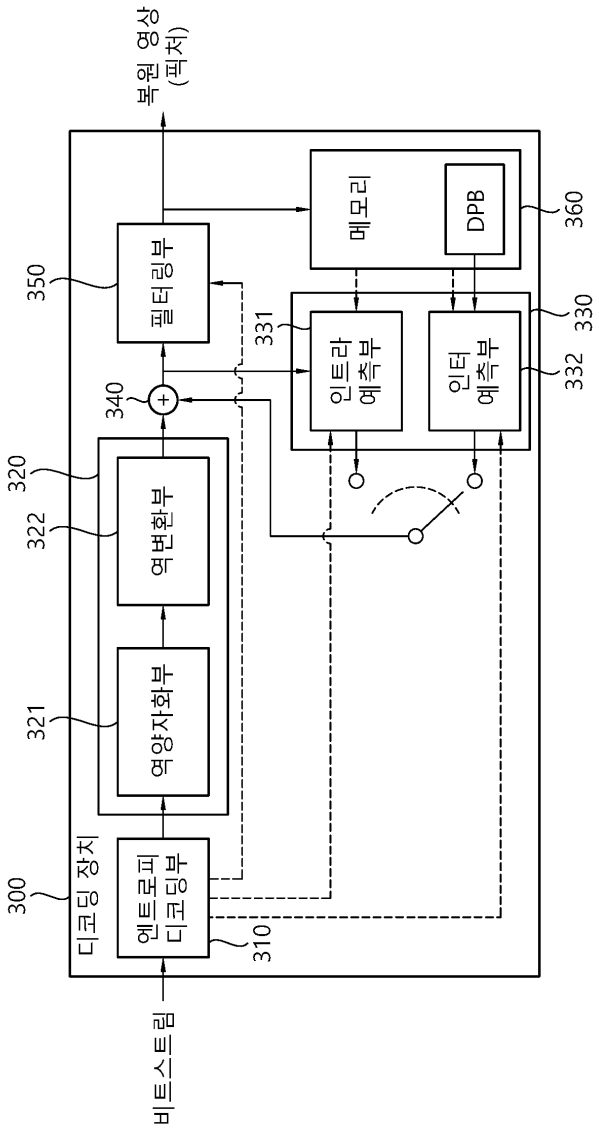
도면1


SOURCE
DEVICE
VIDEO
SOURCE
ENCODING
APPARATUS
TRANSMITTER
RECEIVE
DEVICE
RENDERER
DECODING
APPARATUS
RECEIVER

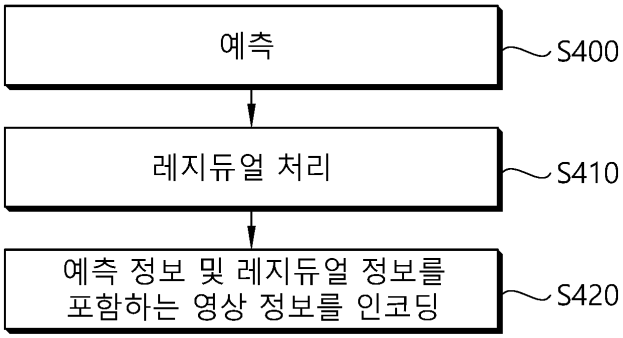
도면2


비트스트림
엔트로피 인코딩부
240
양자화부
233
변환부
232
230
231
+
-
역양자화부
234
역변환부
235
250
인터 예측부 (221)
인트라 예측부 (222)
220
210
인코딩 장치
200
영상 분할부
DPB
메모리
필터링부
입력 영상 (픽처)
270
260

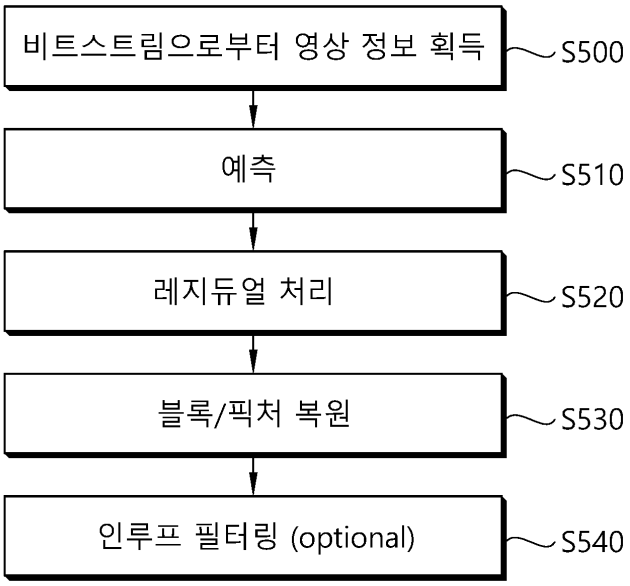
도면3


복원 영상
(픽처)
메모리
DPB
360
필터링부
350
331
인트라
예측부
인터
예측부
330
332
340
320
322
역변환부
321
역양자화부
디코딩 장치
300
엔트로피
디코딩부
310
비트스트림

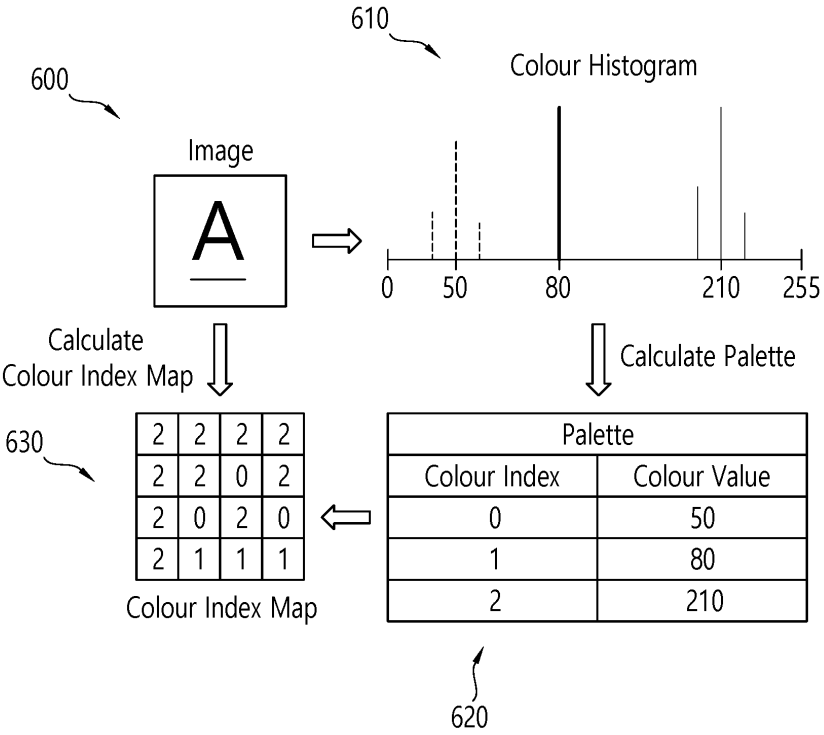
도면4


예측
S400
레지듀얼 처리
S410
예측 정보 및 레지듀얼 정보를
포함하는 영상 정보를 인코딩
S420

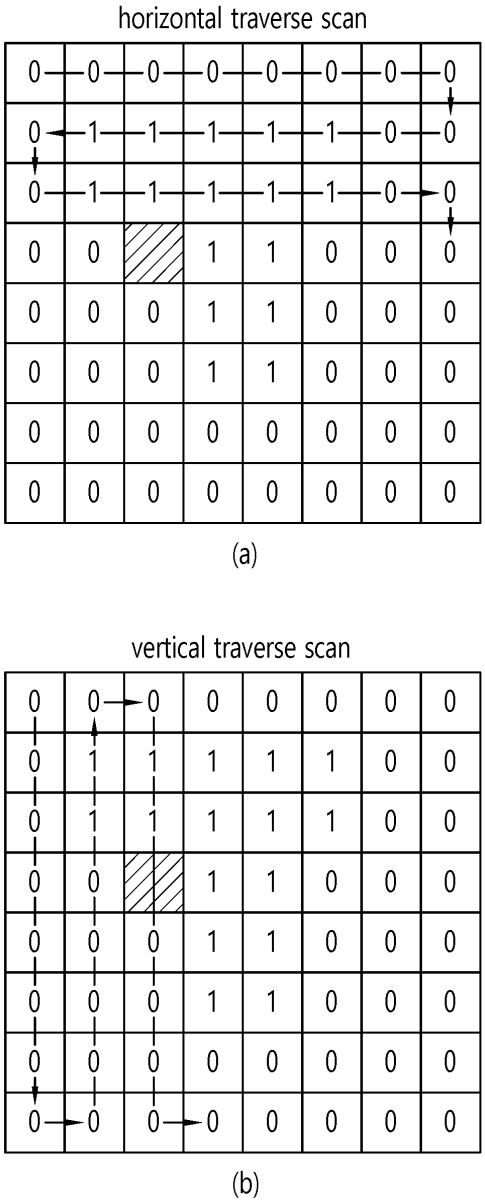
도면5


비트스트림으로부터 영상 정보 획득
S500
예측
S510
레지듀얼 처리
S520
블록/픽처 복원
S530
인루프 필터링 (optional)
S540

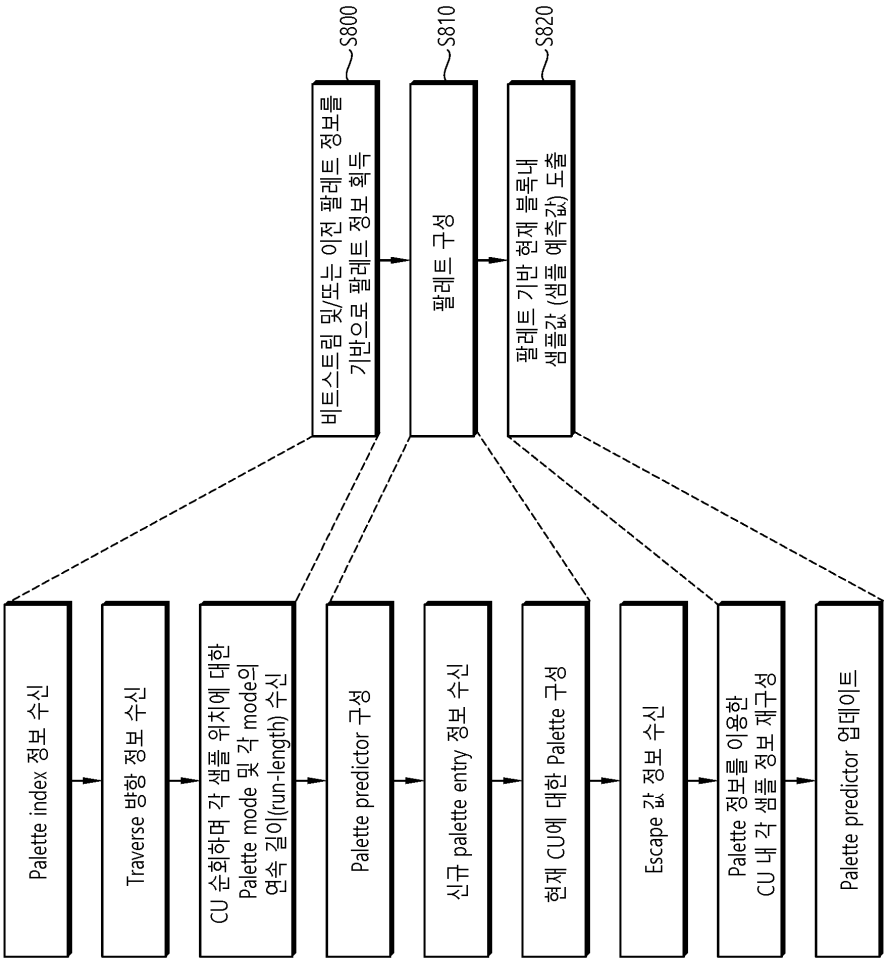
도면6


610
Colour Histogram
600
Image
A
0
50
80
210
255
Calculate
Colour Index Map
Calculate Palette
630
2 2 2 2
2 2 0 2
2 0 2 0
2 1 1 1
Colour Index Map
Palette
Colour Index
Colour Value
0
50
1
80
2
210
620

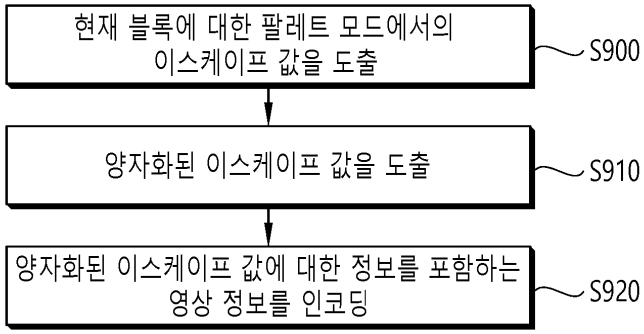
도면7


horizontal traverse scan
0 0 0 0 0 0 0 0
0 1 1 1 1 1 0 0
0 1 1 1 1 1 0 0
0 0 1 1 0 0 0
0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
(a)
vertical traverse scan
0 0 0 0 0 0 0 0
0 1 1 1 1 1 0 0
0 1 1 1 1 1 0 0
0 0 1 1 0 0 0
0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
(b)

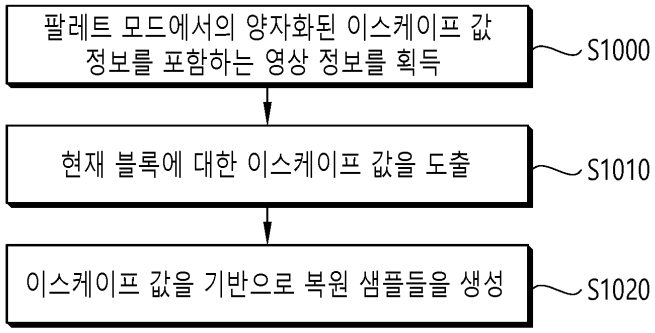
도면8


S800
S810
S820
비트스트림 및/또는 이전 팔레트 정보를 기반으로 팔레트 정보 획득
팔레트 구성
팔레트 기반 현재 블록내 샘플값 (샘플 예측값) 도출
Palette index 정보 수신
Traverse 방향 정보 수신
CU 순회하며 각 샘플 위치에 대한 Palette mode 및 각 mode의 연속 길이(run-length) 수신
Palette predictor 구성
신규 palette entry 정보 수신
현재 CU에 대한 Palette 구성
Escape 값 정보 수신
Palette 정보를 이용한 CU 내 각 샘플 정보 재구성
Palette predictor 업데이트

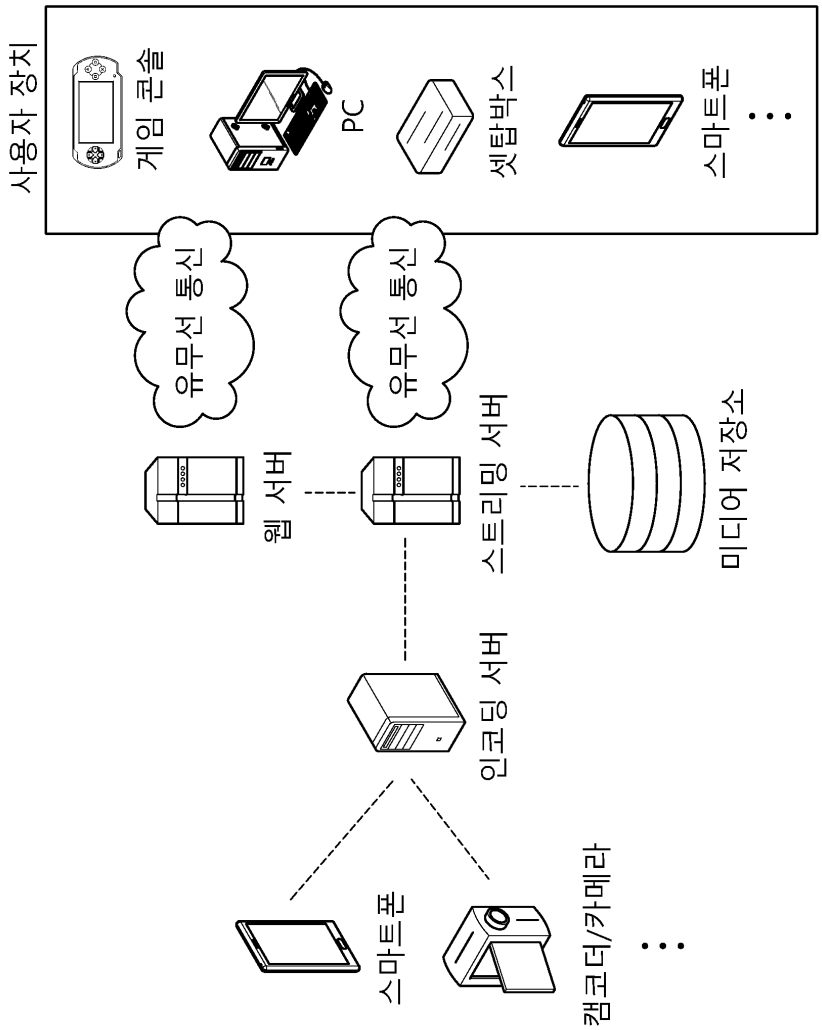
도면9


현재 블록에 대한 팔레트 모드에서의 이스케이프 값을 도출
S900
양자화된 이스케이프 값을 도출
S910
양자화된 이스케이프 값에 대한 정보를 포함하는 영상 정보를 인코딩
S920

도면10



도면11



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 16

【변경전】

영상에 대한 데이터를 전송하는 방법에 있어서,

비트스트림을 획득하되, 상기 비트스트림은 현재 블록에 대한 팔레트 모드에서의 이스케이프 값을 도출하는 단계, 상기 이스케이프 값을 기반으로 양자화된 이스케이프 값을 도출하는 단계, 상기 비트스크림을 생성하기 위하여 상기 양자화된 이스케이프 값에 대한 정보를 포함하는 영상 정보를 인코딩하는 단계를 기반으로 생성되는 단계; 및

상기 비트스트림을 포함하는 상기 데이터를 전송하는 단계를 포함하되,

상기 양자화된 이스케이프 값은 상기 현재 블록 내의 상기 이스케이프 값에 대한 양자화 파라미터를 기반으로 도출되고,

상기 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출되며, 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보는 SPS(Sequence Parameter Set)에 포함되는 신택스 엘리먼트(syntax element)이며, 상기 SPS는 상기 영상 정보에 포함되며,

상기 양자화 파라미터는 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보로부터 도출되는 최소 양자화 파라미터 값보다 크거나 같은 값을 갖는 것을 특징으로 하는 전송 방법.

【변경후】

영상에 대한 데이터를 전송하는 방법에 있어서,

비트스트림을 획득하되, 상기 비트스트림은 현재 블록에 대한 팔레트 모드에서의 이스케이프 값을 도출하는 단계, 상기 이스케이프 값을 기반으로 양자화된 이스케이프 값을 도출하는 단계, 상기 비트스트림을 생성하기 위하여 상기 양자화된 이스케이프 값에 대한 정보를 포함하는 영상 정보를 인코딩하는 단계를 기반으로 생성되는 단계; 및

상기 비트스트림을 포함하는 상기 데이터를 전송하는 단계를 포함하되,

상기 양자화된 이스케이프 값은 상기 현재 블록 내의 상기 이스케이프 값에 대한 양자화 파라미터를 기반으로 도출되고,

상기 양자화 파라미터는 변환 스킵 모드에 대한 최소 양자화 파라미터 정보를 기반으로 도출되며, 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보는 SPS(Sequence Parameter Set)에 포함되는 신택스 엘리먼트(syntax element)이며, 상기 SPS는 상기 영상 정보에 포함되며,

상기 양자화 파라미터는 상기 변환 스킵 모드에 대한 최소 양자화 파라미터 정보로부터 도출되는 최소 양자화 파라미터 값보다 크거나 같은 값을 갖는 것을 특징으로 하는 전송 방법.