(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년08월07일
(11) 등록번호 10-2692622
(24) 등록일자 2024년08월01일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) *H04N 19/176* (2014.01)
H04N 19/186 (2014.01) *H04N 19/44* (2014.01)
H04N 19/60 (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/176 (2015.01)
(21) 출원번호 10-2021-7025187
(22) 출원일자(국제) 2020년10월19일
심사청구일자 2021년08월09일
(85) 번역문제출일자 2021년08월09일
(65) 공개번호 10-2021-0107871
(43) 공개일자 2021년09월01일
(86) 국제출원번호 PCT/US2020/056275
(87) 국제공개번호 WO 2021/080904
국제공개일자 2021년04월29일
(30) 우선권주장
62/924,674 2019년10월22일 미국(US)
17/072,980 2020년10월16일 미국(US)
(56) 선행기술조사문헌
Benjamin Bross, et al., Versatile Video Coding (Draft 7), JVET-P2001-v9, 2019.10.17.*
Helmrich Christian, #371 (Avoid reading unused slice-level elements for joint coding of chroma residuals(JCCR) ~), https://jvet.hhi.fraunhofer.de/trac/vvc/ticket/371, 2019.07.24.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
텐센트 아메리카 엘엘씨
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
(72) 발명자
오옌, 청
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
리, 샹
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
류, 산
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
(74) 대리인
양영준, 김연송, 백만기

전체 청구항 수 : 총 9 항 심사관 : 이남숙

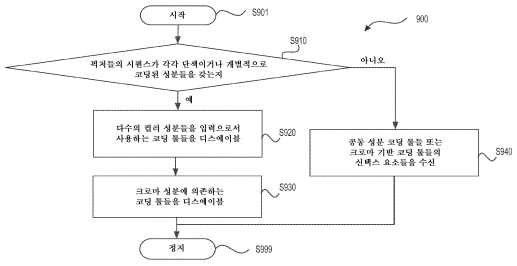
(54) 발명의 명칭 **비디오 성분을 단색 비디오로서 인코딩하기 위한 코딩 툴들의 시그널링**

(57) 요 약

비디오 디코더에서 수행되는 비디오 디코딩 방법이 개시된다. 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함하는지를 표시하는 신택스 요소가 코딩된 비디오의 비트스트림으로부터 수신될 수 있다. 신택스 요소의 값을 추론함으로써, 신택스 요소가 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러

(뒷면에 계속)

대 표 도



성분을 포함한다는 것을 표시할 때 코딩 툴이 디스에이블될 수 있다. 코딩 툴은 픽처의 다수의 컬러 성분들을 입력으로서 사용하거나 픽처의 크로마 성분에 의존한다. 디스에이블된 코딩 툴들의 예들은 크로마 잔차들의 공동 코딩, 액티브 컬러 변환(ACT), 또는 크로마 성분에 대한 블록-기반 델타 펄스 코드 변조(BDPCM)를 포함할 수 있다.

(52) CPC특허분류
H04N 19/186 (2015.01)
H04N 19/44 (2015.01)
H04N 19/60 (2015.01)

명 세 서

청구범위

청구항 1

비디오 디코더에서 수행되는 비디오 디코딩 방법으로서,

코딩된 비디오의 비트스트림으로부터 제1 신택스 요소를 수신하는 단계 - 상기 제1 신택스 요소는 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함하는지를 표시함 - ; 및

상기 제1 신택스 요소가 상기 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함한다는 것을 표시하는 것에 응답하여 픽처의 다수의 컬러 성분들을 입력으로서 사용하거나 픽처의 크로마 성분에 의존하는 제1 코딩 툴을 디스에이블하기 위해 제2 신택스 요소의 값을 추론하는 단계 - 상기 제1 신택스 요소 및 상기 제2 신택스 요소는 상이한 신택스 요소들이며, 상기 제2 신택스 요소는 크로마 잔차들의 공동 코딩(joint coding of chroma residuals)이 인에이블되어 있는지를 표시함 - 를 포함하고,

상기 제1 신택스 요소가 상기 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함한다는 것을 표시하는 것에 응답하여, 액티브 컬러 변환(ACT)이 인에이블되어 있는지를 표시하기 위한 제3 신택스 요소의 값이 0인 것으로 추론되어, 상기 픽처의 다수의 컬러 성분들을 입력으로서 사용하는 제2 코딩 툴을 디스에이블하는 단계를 추가로 포함하는, 방법.

청구항 2

삭제

청구항 3

제1항에 있어서, 상기 픽처의 상기 다수의 컬러 성분들을 입력으로서 사용하는 상기 제1 코딩 툴을 디스에이블하기 위해 상기 제2 신택스 요소의 값을 추론하는 단계는:

상기 제2 신택스 요소의 상기 값이 상기 크로마 잔차들의 공동 코딩이 디스에이블되어 있음을 표시하는 것으로 추론하는 것을 포함하는, 방법.

청구항 4

삭제

청구항 5

제1항에 있어서, 입력으로서 상기 픽처의 크로마 성분에 의존하는 제3 코딩 툴을 디스에이블하기 위해 제4 신택스 요소의 값을 추론하는 단계를 추가로 포함하고, 상기 제4 신택스 요소는 크로마 성분에 대한 블록-기반 델타 펄스 코드 변조(block-based delta pulse code modulation, BDPCM)가 인에이블되어 있는지를 표시하고, 상기 제4 신택스 요소의 상기 값은 0인 것으로 추론되는, 방법.

청구항 6

제1항에 있어서, 상기 픽처의 상기 다수의 컬러 성분들을 입력으로서 사용하거나 상기 픽처의 상기 크로마 성분에 의존하는 상기 제1 코딩 툴을 디스에이블하기 위해 상기 제2 신택스 요소의 상기 값을 추론하는 단계는:

상기 제1 신택스 요소가 상기 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함한다는 것을 표시할 때 변수의 값을 0인 것으로 결정하는 단계 - 상기 변수는 상기 픽처들의 시퀀스의 크로마 어레이 타입을 표시함 - ; 및

상기 변수의 상기 값을 0인 것으로 결정하는 것에 응답하여, 다음의 신택스 요소들:

상기 크로마 잔차들의 공동 코딩이 인에이블되어 있는지를 표시하는 상기 제2 신택스 요소,

ACT가 인에이블되어 있는지를 표시하는 제3 신택스 요소, 또는

크로마 성분에 대한 BDPCM이 인에이블되어 있는지를 표시하는 제4 신택스 요소

중 하나의 신택스 요소의 값을 0인 것으로 추론하는 단계

를 포함하는, 방법.

청구항 7

제1항에 있어서,

상기 픽처들의 시퀀스가 단색이 아니고 개별적으로 코딩되지 않은 3개의 컬러 성분을 포함하는 것으로 결정될 때 상기 크로마 잔차들의 공동 코딩이 인에이블되어 있는지를 표시하는 상기 제2 신택스 요소를 수신하는 단계를 추가로 포함하는, 방법.

청구항 8

제1항에 있어서,

상기 픽처들의 시퀀스가 단색이 아니고 개별적으로 코딩되지 않은 3개의 컬러 성분을 포함하는 것으로 결정될 때 상기 픽처들의 시퀀스의 크로마 어레이 타입을 표시하는 변수의 값을 결정하는 단계; 및

상기 변수의 상기 값이 0이 아닌 것으로 결정될 때 다음의 신택스 요소들:

상기 크로마 잔차들의 공동 코딩이 인에이블되어 있는지를 표시하는 상기 제2 신택스 요소,

ACT가 인에이블되어 있는지를 표시하는 제3 신택스 요소, 또는

크로마 성분에 대한 BDPCM이 인에이블되어 있는지를 표시하는 제4 신택스 요소

중 하나의 신택스 요소를 수신하는 단계

를 추가로 포함하는, 방법.

청구항 9

제1항에 있어서,

상기 픽처들의 시퀀스가 단색이 아니고 개별적으로 코딩되지 않은 3개의 컬러 성분을 포함하는 것으로 결정될 때 상기 픽처들의 시퀀스의 크로마 어레이 타입을 표시하는 변수의 값을 결정하는 단계; 및

상기 변수의 상기 값이 0이 아닌 것으로 결정될 때, 그리고 상기 픽처들의 시퀀스에 대해 무손실 모드가 인에이블되어 있을 때, 크로마 성분에 대한 BDPCM이 인에이블되어 있는지를 표시하는 신택스 요소를 수신하는 단계

를 추가로 포함하는, 방법.

청구항 10

비디오 디코딩 장치로서,

제1항, 제3항, 제5항 내지 제9항 중 어느 한 항의 방법을 수행하도록 구성된 회로를 포함하는, 비디오 디코딩 장치.

청구항 11

비일시적 컴퓨터 판독가능 매체로서,

프로세서에 의해 실행될 때, 상기 프로세서로 하여금 제1항, 제3항, 제5항 내지 제9항 중 어느 한 항의 비디오 디코딩 방법을 수행하게 하는 명령어들이 저장된 비일시적 컴퓨터 판독가능 매체.

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

발명의 설명

기 술 분 야

[0001] 인용에 의한 통합

[0002] 이 본 개시내용은, 2019년 10월 22일자로 출원된 미국 가출원 제62/924,674호, "Signaling of Video Coding Tools for the Encoding of a Video Component as Monochrome Video"에 대한 우선권의 이익을 주장하는, 2020년 10월 16일자로 출원된 미국 특허 출원 제17/072,980호, "Signaling of Coding Tools for Encoding a Video Component as Monochrome Video"에 대한 우선권의 이익을 주장한다. 이전 출원들의 개시내용들은 그 전체가 본 명세서에 참조로 포함된다.

[0003] 기술 분야

[0004] 본 개시내용은 일반적으로 비디오 코딩에 관련된 실시예들을 설명한다.

배 경 기 술

[0005] 본 명세서에 제공된 배경 설명은 본 개시내용의 맥락을 일반적으로 제시하기 위한 것이다. 본 배경기술 부분에 설명되어 있는 현재 등록된 발명자들의 연구 및 출원 시점에 종래 기술로서 달리 간주되지 않을 수 있는 설명의 양태는 명시적으로도 암시적으로도 본 개시내용에 대한 종래 기술로 인정되지 않는다.

[0006] 비디오 코딩 및 디코딩은 모션 보상을 동반한 인터-픽처 예측(inter-picture prediction)을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처들을 포함할 수 있고, 각각의 픽처는, 예를 들어, 1920x1080 루미넌스 샘플들 및 연관된 크로미넌스 샘플들의 공간적 차원(spatial dimension)을 갖는다. 이 일련의 픽처들은, 예를 들어, 초당 60 픽처 또는 60Hz의, 고정 또는 가변 픽처 레이트(비공식적으로 프레임 레이트로도 알려져 있음)를 가질 수 있다. 압축되지 않은 비디오는 상당한 비트레이트 요건들을 갖는다. 예를 들어, 샘플당 8 비트에서의 1080p60 4:2:0 비디오(60Hz 프레임 레이트에서의 1920x1080 루미넌스 샘플 해상도)는 1.5 Gbit/s에 근접한 대역폭을 요구한다. 1 시간 분량의 이러한 비디오는 600 기가바이트를 초과하는 저장 공간을 필요로 한다.

[0007] 비디오 코딩 및 디코딩의 한 가지 목적은, 압축을 통한, 입력 비디오 신호에서의 중복성(redundancy)의 감소일 수 있다. 압축은 앞서 설명한 대역폭 또는 저장 공간 요건들을, 일부 경우들에서는, 2 자릿수 이상 감소시키는 데 도움이 될 수 있다. 무손실 및 손실 압축 양자 모두뿐만 아니라 이들의 조합이 이용될 수 있다. 무손실 압축은 압축된 원래 신호로부터 원래 신호의 정확한 사본이 재구성될 수 있는 기법들을 지칭한다. 손실 압축을 사용할 때, 재구성된 신호는 원래 신호와 동일하지 않을 수 있지만, 원래 신호와 재구성된 신호 사이의 왜곡은 재구성된 신호가 의도된 응용에 유용할 정도로 충분히 작다. 비디오의 경우, 손실 압축이 널리 이용된다. 용인되는 왜곡의 양은 응용에 의존하며; 예를 들어, 특정 소비자 스트리밍 응용들의 사용자들은 텔레비전 배포 응용들의 사용자들보다 더 높은 왜곡을 용인할 수 있다. 달성가능한 압축비는 더 높은 허용가능한/용인가능한 왜곡이 더 높은 압축비를 산출할 수 있다는 사실을 반영할 수 있다.

[0008] 비디오 인코더 및 디코더는, 예를 들어, 모션 보상, 변환, 양자화, 및 엔트로피 코딩을 포함한, 몇가지 광범위한 카테고리로부터의 기법들을 이용할 수 있다.

[0009] 비디오 코덱 기술들은 인트라 코딩(intra coding)으로 알려진 기법들을 포함할 수 있다. 인트라 코딩에서, 샘플 값들은 이전에 재구성된 참조 픽처들로부터의 샘플들 또는 다른 데이터를 참조하지 않고 표현된다. 일부 비디오 코덱들에서, 픽처는 샘플들의 블록들로 공간적으로 세분된다. 샘플들의 모든 블록들이 인트라 모드에서 코딩될 때, 그 픽처는 인트라 픽처(intra picture)일 수 있다. 인트라 픽처들 및 그것들의 파생물들, 예컨대, 독립 디코더 리프레시 픽처들(independent decoder refresh pictures)은 디코더 상태를 리셋하기 위해 사용될 수 있고, 따라서 코딩된 비디오 비트스트림 및 비디오 세션에서 첫번째 픽처로서 또는 스틸 이미지(still image)로서 사용될 수 있다. 인트라 블록의 샘플들은 변환에 노출될 수 있고, 변환 계수들은 엔트로피 코딩 전에 양자화될 수 있다. 인트라 예측은 사전-변환 도메인(pre-transform domain)에서 샘플 값들을 최소화하는 기법일 수 있다. 일부 경우들에서, 변환 후의 DC 값이 더 작을수록, 그리고 AC 계수들이 더 작을수록, 엔트로피 코딩 후의 블록을 나타내기 위해 주어진 양자화 스텝 크기(quantization step size)에서 요구되는 비트들이 더 적다.

[0010] 예를 들어, MPEG-2 세대 코딩 기술로부터 알려진 것과 같은 전통적인 인트라 코딩은 DC 예측 외에 인트라 예측을 사용하지 않는다. 그러나, 일부 더 새로운 비디오 압축 기술들은, 예를 들어, 공간적으로 이웃하는, 그리고 디코딩 순서에서 선행하는 데이터 블록들의 인코딩/디코딩 동안 획득된 주위의 샘플 데이터 및/또는 메타데이터로부터 시도하는 기법들을 포함한다. 이러한 기법들은 이후 "인트라 예측(intra prediction)" 기법들로 불린다. 적어도 일부 경우들에서, 인트라 예측은 참조 픽처들로부터가 아니라 재구성 중인 현재 픽처로부터의 참조 데이터만을 사용한다는 점에 유의한다.

[0011] 많은 상이한 형태의 인트라 예측이 있을 수 있다. 이러한 기법들 중 하나보다 많은 기법이 주어진 비디오 코딩 기술에서 사용될 수 있는 경우, 사용 중인 기법은 인트라 예측 모드에서 코딩될 수 있다. 특정 경우들에서, 모드들은 서브모드들 및/또는 파라미터들을 가질 수 있고, 이들은 개별적으로 코딩되거나 모드 코드워드에 포함될 수 있다. 주어진 모드/서브모드/파라미터 조합에 사용할 코드워드는 인트라 예측을 통해 코딩 효율 이득에 영향을 미칠 수 있고, 코드워드들을 비트스트림으로 변환하기 위해 사용되는 엔트로피 코딩 기술도 그렇게 할 수 있다.

[0012] 인트라 예측의 특정 모드가 H.264와 함께 도입되었고, H.265에서 개선되었으며, JEM(joint exploration model), VVC(versatile video coding), 및 BMS(benchmark set)와 같은 더 새로운 코딩 기술들에서 추가로 개선되었다. 이미 이용가능한 샘플들에 속하는 이웃 샘플 값들을 사용하여 예측자 블록(predictor block)이 형성될 수 있다. 이웃 샘플들의 샘플 값들은 방향에 따라 예측자 블록 내로 복사된다. 사용중인 방향에 대한 참조는 비트스트림에서 코딩될 수 있거나, 자체적으로 예측될 수 있다.

[0013] 도 1a를 참조하면, 하부 우측에 H.265의 33개의 가능한 예측자 방향(35개의 인트라 모드 중 33개의 각도 모드에 대응함)으로부터 알려진 9개의 예측자 방향의 서브세트가 도시되어 있다. 화살표들이 수렴(converge)하는 포인트(101)는 예측되고 있는 샘플을 나타낸다. 화살표들은 샘플이 예측되고 있는 방향을 나타낸다. 예를 들어, 화살표(102)는 샘플(101)이 샘플 또는 샘플들로부터 상부 우측으로, 수평으로부터 45도 각도로 예측되는 것을 표시한다. 유사하게, 화살표(103)는 샘플(101)이 샘플 또는 샘플들로부터 샘플(101)의 하부 좌측으로, 수평으로부터 22.5도 각도로 예측되는 것을 표시한다.

[0014] 계속 도 1a를 참조하면, 상단 좌측에, 4x4 샘플들의 정사각형 블록(104)(굵은 파선으로 표시됨)이 묘사되어 있다. 정사각형 블록(104)은 16개의 샘플을 포함하며, 각각의 샘플은 "S", Y 차원에서의 위치(예를 들어, 행 인덱스), 및 X 차원에서의 위치(예를 들어, 열 인덱스)로 라벨링되어 있다. 예를 들어, 샘플 S21은 Y 차원에서의

(상단으로부터) 2번째 샘플 및 X 차원에서의 (좌측으로부터) 1번째 샘플이다. 유사하게, 샘플 S44는 블록(104)에서 Y 차원과 X 차원 둘 다에서의 4번째 샘플이다. 블록이 크기가 4x4 샘플이므로, S44는 하단 우측에 있다. 유사한 넘버링 방식을 따르는 참조 샘플들이 추가로 도시되어 있다. 블록(104)에 대해 참조 샘플이 R, 그의 Y 위치(예를 들어, 행 인덱스) 및 X 위치(열 인덱스)로 라벨링된다. H.264와 H.265 양자 모두에서, 예측 샘플들은 재구성중인 블록에 이웃하며; 따라서, 음의 값들이 사용될 필요가 없다.

[0015] 인트라 픽처 예측은 시그널링된 예측 방향에 의해 적절하게 이웃 샘플들로부터 참조 샘플 값들을 복사함으로써 작동할 수 있다. 예를 들어, 코딩된 비디오 비트스트림은, 이 블록에 대해, 화살표(102)와 일치하는 예측 방향을 표시하는 - 즉, 샘플들이 예측 샘플 또는 샘플들로부터 상부 우측으로, 수평으로부터 45도 각도로 예측되는 - 시그널링을 포함한다고 가정한다. 그 경우, 샘플들 S41, S32, S23, 및 S14는 동일한 참조 샘플 R05로부터 예측된다. 이어서, 샘플 S44는 참조 샘플 R08로부터 예측된다.

[0016] 특정 경우에, 다수의 참조 샘플들의 값들은, 특히, 방향들이 45도로 균등하게 나누어지지 않을 때, 참조 샘플을 계산하기 위해, 예를 들어, 보간을 통해 조합될 수 있다.

[0017] 비디오 코딩 기술이 발전함에 따라 가능한 방향의 수가 증가하였다. H.264(2003년)에서, 9개의 상이한 방향이 표현될 수 있다. 그것은 H.265(2013년)에서 33개로 증가하였고, JEM/VVC/BMS는 공개 시점에 최대 65개의 방향을 지원할 수 있다. 가장 가능성 있는 방향들을 식별하기 위한 실험들이 수행되었고, 엔트로피 코딩에서의 특정 기법들이 적은 수의 비트들로 이러한 가능성 있는 방향들을 나타내기 위해 사용되어, 가능성이 적은 방향들에 대한 특정 페널티를 용인한다. 또한, 방향들 자체가 이웃하는, 이미 디코딩된, 블록들에서 사용되는 이웃 방향들로부터 때때로 예측될 수 있다.

[0018] HEVC에서 사용되는 인트라 예측 모드들이 도 1b에 예시되어 있다. HEVC에서, 총 35개의 인트라 예측 모드가 존재하며, 그 중에서 모드 10은 수평 모드이고, 모드 26은 수직 모드이고, 모드 2, 모드 18 및 모드 34는 대각선 모드들이다. 인트라 예측 모드들은 3개의 최고 확률 모드(most probable mode, MPM) 및 32개의 나머지 모드에 의해 시그널링된다.

[0019] 도 1c는 VVC에서 사용되는 인트라 예측 모드들을 예시한다. VVC에서는, 도 1c에 도시된 바와 같이 총 95개의 인트라 예측 모드가 존재하며, 여기서 모드 18은 수평 모드이고, 모드 50은 수직 모드이고, 모드 2, 모드 34 및 모드 66은 대각선 모드들이다. 모드 -1 내지 모드 -14 및 모드 67 내지 모드 80은 광각 인트라 예측(Wide-Angle Intra Prediction, WAIP) 모드들이라고 불린다.

[0020] 방향을 표현하는 코딩된 비디오 비트스트림 내의 인트라 예측 방향 비트들의 맵핑은 비디오 코딩 기술마다 상이할 수 있으며; 예를 들어, 인트라 예측 모드에 대한 예측 방향의 간단한 직접 맵핑들로부터, 코드워드들, MPM들을 수반하는 복잡한 적응적 방식들, 및 유사한 기법들에 이르기까지 다양할 수 있다. 그러나, 모든 경우에, 특정한 다른 방향들보다 비디오 콘텐츠에서 일어날 가능성이 통계적으로 낮은 특정 방향들이 있을 수 있다. 비디오 압축의 목표는 중복성의 감소이므로, 잘 동작하는 비디오 코딩 기술에서, 이러한 가능성이 적은 방향들은 가능성이 많은 방향들보다 많은 수의 비트들로 표현될 것이다.

[0021] 비디오 코딩 및 디코딩은 모션 보상을 동반한 인터-픽처 예측(inter-picture prediction)을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처들을 포함할 수 있고, 각각의 픽처는, 예를 들어, 1920x1080 루미넌스 샘플들 및 연관된 크로미넌스 샘플들의 공간적 차원(spatial dimension)을 갖는다. 이 일련의 픽처들은, 예를 들어, 초당 60 픽처 또는 60Hz의, 고정 또는 가변 픽처 레이트(비공식적으로 프레임 레이트로도 알려져 있음)를 가질 수 있다. 압축되지 않은 비디오는 상당한 비트레이트 요건들을 갖는다. 예를 들어, 샘플당 8 비트에서의 1080p60 4:2:0 비디오(60Hz 프레임 레이트에서의 1920x1080 루미넌스 샘플 해상도)는 1.5 Gbit/s에 근접한 대역폭을 요구한다. 1 시간 분량의 이러한 비디오는 600 기가바이트를 초과하는 저장 공간을 필요로 한다.

[0022] 모션 보상은 손실 압축 기법일 수 있고, 이전에 재구성된 픽처 또는 그의 일부(참조 픽처)로부터의 샘플 데이터의 블록이, 모션 벡터(motion vector)(이후 MV)에 의해 표시된 방향으로 공간적으로 시프트된 이후에, 새롭게 재구성된 픽처 또는 픽처 부분의 예측에 사용되는 기법들과 관련될 수 있다. 일부 경우들에서, 참조 픽처는 현재 재구성 중인 픽처와 동일할 수 있다. MV들은 2개의 차원 X 및 Y, 또는 3개의 차원을 가질 수 있고, 제3 차원은 사용중인 참조 픽처의 표시이다(후자는, 간접적으로, 시간 차원일 수 있다).

[0023] 일부 비디오 압축 기법들에서, 샘플 데이터의 특정 영역에 적용가능한 MV는 다른 MV들로부터, 예를 들어 재구성 중인 영역에 공간적으로 인접한 샘플 데이터의 다른 영역과 관련되고, 디코딩 순서로 그 MV에 선행하는 것들로

부터 예측될 수 있다. 그렇게 함으로써 MV를 코딩하기 위해 요구되는 데이터의 양을 실질적으로 감소시킬 수 있고, 그에 의해 중복성을 제거하고 압축을 증가시킨다. MV 예측은, 예를 들어, 카메라로부터 도출된 입력 비디오 신호(자연 비디오(natural video)라고 알려짐)를 코딩할 때, 단일 MV가 적용가능한 영역보다 더 큰 영역들이 유사한 방향으로 움직이는 통계적 가능성이 있기 때문에 효과적으로 작동할 수 있고, 따라서, 일부 경우들에는 이웃 영역의 MV들로부터 도출된 유사한 모션 벡터를 사용하여 예측될 수 있다. 그 결과, 주어진 영역에 대해 발견되는 MV가 주위의 MV들로부터 예측된 MV와 유사하거나 동일하게 되고, 그것은 결국, 엔트로피 코딩 후에, MV를 직접 코딩하는 경우에 사용되는 것보다 더 적은 수의 비트들로 표현될 수 있다. 일부 경우들에서, MV 예측은 원래 신호(즉: 샘플 스트림)로부터 도출된 신호(즉: MV들)의 무손실 압축의 예일 수 있다. 다른 경우들에서, MV 예측 자체는, 예를 들어, 수 개의 주위의 MV들로부터 예측자를 계산할 때의 반올림 오류들 때문에, 손실성일 수 있다.

[0024] 다양한 MV 예측 메커니즘들이 H.265/HEVC (ITU-T Rec. H.265, "High Efficiency Video Coding", December 2016)에 설명되어 있다. H.265가 제안하는 많은 MV 예측 메커니즘들 중에서, 여기서는 이후 "공간적 병합(spatial merge)"이라고 지칭되는 기법이 설명된다.

[0025] 도 1d를 참조하면, 현재 블록(110)은 공간적으로 시프트된 동일한 크기의 이전 블록으로부터 예측가능한 것으로 모션 검색 프로세스 동안 인코더에 의해 발견된 샘플들을 포함한다. 그 MV를 직접 코딩하는 대신에, MV는 하나 이상의 참조 픽처와 연관된 메타데이터로부터, 예를 들어, 가장 최근의(디코딩 순서로) 참조 픽처로부터, A0, A1, 및 B0, B1, B2(각각, 102 내지 106)로 나타내어진 5개의 주위 샘플 중 어느 하나와 연관된 MV를 사용하여 도출될 수 있다. H.265에서, MV 예측은 이웃 블록이 사용하고 있는 동일한 참조 픽처로부터의 예측자들을 사용할 수 있다. 후보 리스트를 형성하는 순서는 A0→B0→B1→A1→B2일 수 있다.

발명의 내용

[0026] 본 개시내용의 양태들은 비디오 디코더에서 수행되는 비디오 디코딩 방법을 제공한다. 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함하는지를 표시하는 신택스 요소가 코딩된 비디오의 비트스트림으로부터 수신될 수 있다. 신택스 요소의 값을 추론함으로써, 신택스 요소가 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함한다는 것을 표시할 때 코딩 툴이 디스에이블될 수 있다. 코딩 툴은 픽처의 다수의 컬러 성분들을 입력으로서 사용하거나 픽처의 크로마 성분에 의존한다.

[0027] 일 실시예에서, 디스에이블된 코딩 툴은 크로마 잔차들의 공동 코딩(joint coding of chroma residuals), 액티브 컬러 변환(active color transform, ACT), 또는 크로마 성분에 대한 블록-기반 델타 펄스 코드 변조(block-based delta pulse code modulation, BDPCM)의 코딩 툴 중 하나이다.

[0028] 일 실시예에서, 크로마 잔차들의 공동 코딩이 인에이블되어 있는지를 표시하는 신택스 요소의 값은 0인 것으로 추론된다. 일 실시예에서, ACT가 인에이블되어 있는지를 표시하는 신택스 요소의 값은 0인 것으로 추론될 수 있다. 일 실시예에서, 크로마 성분에 대한 BDPCM이 인에이블되어 있는지를 표시하는 신택스 요소의 값은 0인 것으로 추론될 수 있다.

[0029] 일 실시예에서, 신택스 요소가 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함한다는 것을 표시할 때 변수의 값은 0인 것으로 결정된다. 변수는 픽처들의 시퀀스의 크로마 어레이 타입을 표시한다. 변수의 값을 0인 것으로 결정하는 것에 응답하여, 다음의 신택스 요소들: 크로마 잔차들의 공동 코딩이 인에이블되어 있는지를 표시하는 신택스 요소, ACT가 인에이블되어 있는지를 표시하는 신택스 요소, 또는 크로마 성분에 대한 BDPCM이 인에이블되어 있는지를 표시하는 신택스 요소 중 하나의 신택스 요소의 값은 0인 것으로 추론될 수 있다.

[0030] 일부 실시예들에서, 픽처들의 시퀀스가 단색이 아니고 개별적으로 코딩되지 않은 3개의 컬러 성분을 포함한다고 결정하는 것에 응답하여, 크로마 잔차들의 공동 코딩이 인에이블되어 있는지를 표시하는 신택스 요소가 수신될 수 있거나; ACT가 인에이블되어 있는지를 표시하는 신택스 요소가 수신될 수 있거나; 또는 크로마 성분에 대한 BDPCM이 인에이블되어 있는지를 표시하는 신택스 요소가 수신될 수 있다.

[0031] 일 실시예에서, 픽처들의 시퀀스가 단색이 아니고 개별적으로 코딩되지 않은 3개의 컬러 성분을 포함한다고 결정하는 것에 응답하여, 픽처들의 시퀀스의 크로마 어레이 타입을 표시하는 변수의 값이 결정될 수 있다. 변수의 값이 0이 아닌 것으로 결정될 때 다음의 신택스 요소들: 크로마 잔차들의 공동 코딩이 인에이블되어 있는지를 표시하는 신택스 요소, ACT가 인에이블되어 있는지를 표시하는 신택스 요소, 또는 크로마 성분에 대한 BDPCM

이 인에이블되어 있는지를 표시하는 신택스 요소 중 하나의 신택스 요소가 수신될 수 있다.

[0032] 일 실시예에서, 픽처들의 시퀀스가 단색이 아니고 개별적으로 코딩되지 않은 3개의 컬러 성분을 포함한다고 결정될 때 픽처들의 시퀀스의 크로마 어레이 타입을 표시하는 변수의 값이 결정될 수 있다. 변수의 값이 0이 아닌 것으로 결정될 때, 그리고 픽처들의 시퀀스에 대해 무손실 모드가 인에이블되어 있을 때, 크로마 성분에 대한 BDPCM이 인에이블되어 있는지를 표시하는 신택스 요소가 수신될 수 있다.

[0033] 본 개시내용의 양태들은 회로를 포함하는 비디오 디코딩 장치를 제공한다. 회로는 코딩된 비디오의 비트스트림으로부터 신택스 요소를 수신하도록 구성될 수 있다. 신택스 요소는 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함하는지를 표시한다. 회로는 신택스 요소가 픽처들의 시퀀스가 단색이거나 개별적으로 코딩된 3개의 컬러 성분을 포함한다는 것을 표시할 때 픽처의 다수의 컬러 성분들을 입력으로서 사용하거나 픽처의 크로마 성분에 의존하는 코딩 툴을 디스에이블하기 위해 신택스 요소의 값을 추론하도록 추가로 구성될 수 있다.

[0034] 본 개시내용의 양태들은, 프로세서에 의해 실행될 때, 프로세서로 하여금 비디오 디코딩 방법을 수행하게 하는 명령어들이 저장된 비일시적 컴퓨터 판독가능 매체를 제공한다.

도면의 간단한 설명

[0035] 개시된 주제의 추가의 특징들, 본질 및 다양한 이점들이 다음의 상세한 설명 및 첨부 도면들로부터 더 명백할 것이다.

도 1a는 인트라 예측 모드들의 예시적인 서브세트의 개략도이다.

도 1b는 예시적인 인트라 예측 방향들의 도면이다.

도 1c는 예시적인 인트라 예측 방향들의 도면이다.

도 1d는 일 예에서 현재 블록 및 그 주위의 공간적 병합 후보들의 개략도이다.

도 2는 일 실시예에 따른 통신 시스템의 단순화된 블록도의 개략도이다.

도 3은 일 실시예에 따른 통신 시스템의 단순화된 블록도의 개략도이다.

도 4는 일 실시예에 따른 디코더의 단순화된 블록도의 개략도이다.

도 5는 일 실시예에 따른 인코더의 단순화된 블록도의 개략도이다.

도 6은 다른 실시예에 따른 인코더의 블록도를 도시한다.

도 7은 다른 실시예에 따른 디코더의 블록도를 도시한다.

도 8은 디코더에 의해 수행되는 프로세스의 실시예의 도면이다.

도 9는 디코더에 의해 수행되는 다른 프로세스의 실시예의 도면이다.

도 10은 본 개시내용의 실시예에 따른 컴퓨터 시스템의 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0036] I. 비디오 인코더 및 디코더 시스템들

[0037] 도 2는 본 개시내용의 일 실시예에 따른 통신 시스템(200)의 단순화된 블록도를 예시한다. 통신 시스템(200)은, 예를 들어, 네트워크(250)를 통해, 서로 통신할 수 있는 복수의 단말 디바이스를 포함한다. 예를 들어, 통신 시스템(200)은 네트워크(250)를 통해 상호접속되는 제1 쌍의 단말 디바이스들(210 및 220)을 포함한다. 도 2의 예에서, 제1 쌍의 단말 디바이스들(210 및 220)은 데이터의 단방향 송신을 수행한다. 예를 들어, 단말 디바이스(210)는 네트워크(250)를 통해 다른 단말 디바이스(220)로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스(210)에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림의 형태로 송신될 수 있다. 단말 디바이스(220)는 네트워크(250)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구하고 복구된 비디오 데이터에 따라 비디오 픽처들을 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 응용들(media serving applications) 등에서 일반적일 수 있다.

[0038] 다른 예에서, 통신 시스템(200)은, 예를 들어, 영상 회의 동안 발생할 수 있는 코딩된 비디오 데이터의 양방향 송신을 수행하는 제2 쌍의 단말 디바이스들(230 및 240)을 포함한다. 데이터의 양방향 송신을 위해, 일 예에서, 단말 디바이스들(230 및 240) 중의 각각의 단말 디바이스는 네트워크(250)를 통해 단말 디바이스들(230 및 240) 중의 다른 단말 디바이스로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 단말 디바이스들(230 및 240) 중의 각각의 단말 디바이스는 또한 단말 디바이스들(230 및 240) 중의 다른 단말 디바이스에 의해 송신된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구할 수 있고, 복구된 비디오 데이터에 따라 액세스가능한 디스플레이 디바이스에서 비디오 픽처들을 디스플레이할 수 있다.

[0039] 도 2의 예에서, 단말 디바이스들(210, 220, 230 및 240)은 서버들, 개인용 컴퓨터들 및 스마트 폰들로서 예시될 수 있지만, 본 개시내용의 원리들은 이에 제한되지 않는다. 본 개시내용의 실시예들은 랩톱 컴퓨터들, 태블릿 컴퓨터들, 미디어 플레이어들 및/또는 전용 영상 회의 장비에서 응용된다. 네트워크(250)는 예를 들어 유선(와이어드) 및/또는 무선 통신 네트워크들을 포함하여, 단말 디바이스들(210, 220, 230 및 240) 사이에 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크들을 나타낸다. 통신 네트워크(250)는 회선 교환 및/또는 패킷 교환 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 통신 네트워크들, 로컬 영역 네트워크들, 광역 네트워크들 및/또는 인터넷을 포함한다. 본 논의의 목적을 위해, 네트워크(250)의 아키텍처 및 토폴로지는 아래에서 본 명세서에서 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.

[0040] 도 3은, 개시된 주제를 위한 응용에 대한 예로서, 스트리밍 환경에서의 비디오 인코더 및 비디오 디코더의 배치를 예시한다. 개시된 주제는, 예를 들어, 영상 회의, 디지털 TV, CD, DVD, 메모리 스틱 등을 포함하는 디지털 미디어 상의 압축된 비디오의 저장 등을 포함하여, 다른 비디오 가능 응용들에 동등하게 적용가능할 수 있다.

[0041] 스트리밍 시스템은, 예를 들어 압축되지 않은 비디오 픽처들의 스트림(302)을 생성하는 비디오 소스(301), 예를 들어 디지털 카메라를 포함할 수 있는 캡처 서브시스템(313)을 포함할 수 있다. 일 예에서, 비디오 픽처들의 스트림(302)은 디지털 카메라에 의해 촬영되는 샘플들을 포함한다. 인코딩된 비디오 데이터(304)(또는 코딩된 비디오 비트스트림)와 비교할 때 높은 데이터 용량을 강조하기 위해 굵은 라인으로 묘사된 비디오 픽처들의 스트림(302)은 비디오 소스(301)에 결합된 비디오 인코더(303)를 포함하는 전자 디바이스(320)에 의해 처리될 수 있다. 비디오 인코더(303)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 양태들을 가능하게 하거나 구현하기 위해 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 비디오 픽처들의 스트림(302)과 비교할 때 낮은 데이터 용량을 강조하기 위해 얇은 라인으로서 묘사된 인코딩된 비디오 데이터(304)(또는 인코딩된 비디오 비트스트림(304))는 장래의 사용을 위해 스트리밍 서버(305) 상에 저장될 수 있다. 도 3에서의 클라이언트 서브시스템들(306 및 308)과 같은 하나 이상의 스트리밍 클라이언트 서브시스템들은 스트리밍 서버(305)에 액세스하여 인코딩된 비디오 데이터(304)의 사본들(307 및 309)을 검색할 수 있다. 클라이언트 서브시스템(306)은, 예를 들어, 전자 디바이스(330) 내에 비디오 디코더(310)를 포함할 수 있다. 비디오 디코더(310)는 인코딩된 비디오 데이터의 유입 사본(307)을 디코딩하고 디스플레이(312)(예를 들어, 디스플레이 스크린) 또는 다른 렌더링 디바이스(묘사되지 않음) 상에 렌더링될 수 있는 비디오 픽처들의 유출 스트림(311)을 생성한다. 일부 스트리밍 시스템들에서, 인코딩된 비디오 데이터(304, 307, 및 309)(예를 들어, 비디오 비트스트림들)는 특정 비디오 코딩/압축 표준들에 따라 인코딩될 수 있다. 그러한 표준들의 예들은 ITU-T 권고안(Recommendation) H.265를 포함한다. 일 예에서, 개발 중인 비디오 코딩 표준은 다용도 비디오 코딩(Versatile Video Coding, VVC)으로서 비공식적으로 알려져 있다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.

[0042] 전자 디바이스들(320 및 330)은 다른 컴포넌트들(도시되지 않음)을 포함할 수 있다는 점에 유의한다. 예를 들어, 전자 디바이스(320)는 비디오 디코더(도시되지 않음)도 포함할 수 있고 전자 디바이스(330)는 비디오 인코더(도시되지 않음)도 포함할 수 있다.

[0043] 도 4는 본 개시내용의 일 실시예에 따른 비디오 디코더(410)의 블록도를 도시한다. 비디오 디코더(410)는 전자 디바이스(430)에 포함될 수 있다. 전자 디바이스(430)는 수신기(431)(예를 들어, 수신 회로)를 포함할 수 있다. 비디오 디코더(410)는 도 3의 예에서의 비디오 디코더(310) 대신에 사용될 수 있다.

[0044] 수신기(431)는 비디오 디코더(410)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스를 수신할 수 있으며; 동일한 또는 다른 실시예에서는, 한 번에 하나의 코딩된 비디오 시퀀스를 수신할 수 있으며, 여기서 각각의 코딩된 비디오 시퀀스의 디코딩은 다른 코딩된 비디오 시퀀스들과 독립적이다. 코딩된 비디오 시퀀스는, 인코딩된 비디오 데이터를 저장하는 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는, 채널(401)로부터 수신될

수 있다. 수신기(431)는 인코딩된 비디오 데이터를 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림들과 함께 수신할 수 있고, 이들은 그것들 각각의 사용 엔티티들(묘사되지 않음)에 포워딩될 수 있다. 수신기(431)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터를 방지하기 위해, 수신기(431)와 엔트로피 디코더/파서(420)(이후 "파서(420)") 사이에 버퍼 메모리(415)가 결합될 수 있다. 특정 응용들에서, 버퍼 메모리(415)는 비디오 디코더(410)의 일부이다. 다른 응용들에서, 그것은 비디오 디코더(410)(묘사되지 않음) 외부에 있을 수 있다. 또 다른 응용들에서, 예를 들어 네트워크 지터를 방지하기 위해, 비디오 디코더(410) 외부의 버퍼 메모리(묘사되지 않음), 그리고 추가로, 예를 들어 재생 타이밍을 핸들링하기 위해, 비디오 디코더(410) 내부의 다른 버퍼 메모리(415)가 존재할 수 있다. 수신기(431)가 충분한 대역폭 및 제어가능성의 저장/포워드 디바이스로부터, 또는 등시 동기식 네트워크(isosynchronous network)로부터 데이터를 수신하고 있을 때, 버퍼 메모리(415)는 필요하지 않을 수 있거나, 작을 수 있다. 인터넷과 같은 베스트 에포트 패킷 네트워크들(best effort packet networks) 상에서의 사용을 위해, 버퍼 메모리(415)가 요구될 수 있고, 비교적 클 수 있으며, 유리하게는 적응적 크기일 수 있고, 비디오 디코더(410) 외부의 운영 체제 또는 유사한 요소들(묘사되지 않음)에서 적어도 부분적으로 구현될 수 있다.

[0045] 비디오 디코더(410)는 코딩된 비디오 시퀀스로부터 심벌들(421)을 재구성하기 위해 파서(420)를 포함할 수 있다. 해당 심벌들의 카테고리들은 비디오 디코더(410)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 4에 도시된 바와 같이, 전자 디바이스(430)의 일체 부분(integral part)은 아니지만 전자 디바이스(430)에 결합될 수 있는 렌더링 디바이스(412)(예를 들어, 디스플레이 스크린)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함한다. 렌더링 디바이스(들)에 대한 제어 정보는 SEI(Supplemental Enhancement Information) 메시지들 또는 VUI(Video Usability Information) 파라미터 세트 프래그먼트들(묘사되지 않음)의 형태로 될 수 있다. 파서(420)는 수신되는 코딩된 비디오 시퀀스를 파싱/엔트로피 디코딩할 수 있다. 코딩된 비디오 시퀀스의 코딩은 비디오 코딩 기술 또는 표준에 따를 수 있고, 가변 길이 코딩, 허프만 코딩(Huffman coding), 맥락 민감성(context sensitivity)을 갖거나 갖지 않는 산술 코딩 등을 포함하는 다양한 원리들을 따를 수 있다. 파서(420)는, 코딩된 비디오 시퀀스로부터, 그룹에 대응하는 적어도 하나의 파라미터에 기초하여, 비디오 디코더 내의 픽셀들의 서브그룹들 중 적어도 하나에 대한 서브그룹 파라미터들의 세트를 추출할 수 있다. 서브그룹들은 픽처 그룹(Group of Pictures, GOP)들, 픽처들, 타일들, 슬라이스들, 매크로블록들, 코딩 유닛(Coding Unit, CU)들, 블록들, 변환 유닛(Transform Unit, TU)들, 예측 유닛(Prediction Unit, PU)들 등을 포함할 수 있다. 파서(420)는 또한 코딩된 비디오 시퀀스로부터 변환 계수들, 양자화기 파라미터 값들, 모션 벡터들 등과 같은 정보를 추출할 수 있다.

[0046] 파서(420)는 버퍼 메모리(415)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여, 심벌들(421)을 생성할 수 있다.

[0047] 심벌들(421)의 재구성은 코딩된 비디오 픽처 또는 그의 부분들의 타입(예컨대: 인터 및 인트라 픽처, 인터 및 인트라 블록), 및 다른 인자들에 따라 다수의 상이한 유닛들을 수반할 수 있다. 어느 유닛들이 수반되는지, 그리고 그 방식은 파서(420)에 의해 코딩된 비디오 시퀀스로부터 파싱된 서브그룹 제어 정보에 의해 제어될 수 있다. 파서(420)와 아래의 다수의 유닛 사이의 이러한 서브그룹 제어 정보의 흐름은 명확성을 위해 묘사되어 있지 않다.

[0048] 이미 언급된 기능 블록들 이외에, 비디오 디코더(410)는 아래에 설명되는 바와 같이 개념적으로 다수의 기능 유닛으로 세분될 수 있다. 상업적 제약 하에서 동작하는 실제 구현에서, 이들 유닛 중 다수는 서로 밀접하게 상호작용하고, 적어도 부분적으로 서로 통합될 수 있다. 그러나, 개시된 주제를 설명하기 위해서는, 아래의 기능 유닛들로의 개념적 세분이 적절하다.

[0049] 제1 유닛은 스케일러/역변환 유닛(451)이다. 스케일러/역변환 유닛(451)은, 파서(420)로부터의 심벌(들)(421)로서, 어느 변환을 사용할지, 블록 크기, 양자화 인자, 양자화 스케일링 행렬들(quantization scaling matrices) 등을 포함한, 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 스케일러/역변환 유닛(451)은 집계기(aggregator)(455)에 입력될 수 있는 샘플 값들을 포함하는 블록들을 출력할 수 있다.

[0050] 일부 경우들에서, 스케일러/역변환(451)의 출력 샘플들은 인트라 코딩된 블록; 즉, 이전에 재구성된 픽처들로부터의 예측 정보를 사용하는 것이 아니고, 현재 픽처의 이전에 재구성된 부분들로부터의 예측 정보를 사용할 수 있는 블록에 관련될 수 있다. 그러한 예측 정보는 인트라 픽처 예측 유닛(452)에 의해 제공될 수 있다. 일부 경우들에서, 인트라 픽처 예측 유닛(452)은 현재 픽처 버퍼(458)로부터 페치된 주위의 이미 재구성된 정보를 사용하여, 재구성 중인 블록의 동일한 크기 및 형상의 블록을 생성한다. 현재 픽처 버퍼(458)는, 예를 들어, 부

분적으로 재구성된 현재 픽처 및/또는 완전히 재구성된 현재 픽처를 버퍼링한다. 집계기(455)는, 일부 경우들에서, 샘플당 기준으로, 인트라 예측 유닛(452)이 생성한 예측 정보를 스케일러/역변환 유닛(451)에 의해 제공된 출력 샘플 정보에 추가한다.

[0051] 다른 경우들에서, 스케일러/역변환 유닛(451)의 출력 샘플들은 인터 코딩되고, 잠재적으로 모션 보상된 블록에 관련될 수 있다. 그러한 경우에, 모션 보상 예측 유닛(453)은 참조 픽처 메모리(457)에 액세스하여 예측에 사용되는 샘플들을 페치할 수 있다. 블록에 관련된 심벌들(421)에 따라 페치된 샘플들을 모션 보상한 후에, 이들 샘플은 집계기(455)에 의해 스케일러/역변환 유닛(451)의 출력(이 경우 잔차 샘플들 또는 잔차 신호라고 불림)에 추가되어 출력 샘플 정보를 생성할 수 있다. 모션 보상 예측 유닛(453)이 예측 샘플들을 페치하는 참조 픽처 메모리(457) 내의 어드레스들은, 예를 들어 X, Y, 및 참조 픽처 컴포넌트들을 가질 수 있는 심벌들(421)의 형태로 모션 보상 예측 유닛(453)에 이용가능한 모션 벡터들에 의해 제어될 수 있다. 모션 보상은 또한 서브샘플 정확한 모션 벡터들이 사용중일 때 참조 픽처 메모리(457)로부터 페치된 샘플 값들의 보간, 모션 벡터 예측 메커니즘 등을 포함할 수 있다.

[0052] 집계기(455)의 출력 샘플들에 대해 루프 필터 유닛(456) 내의 다양한 루프 필터링 기법들이 수행될 수 있다. 비디오 압축 기술들은, 파서(420)로부터의 심벌들(421)로서 루프 필터 유닛(456)에 이용가능하게 되고 코딩된 비디오 시퀀스(코딩된 비디오 비트스트림이라고도 지칭됨)에 포함된 파라미터들에 의해 제어되지만, 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서로) 부분들의 디코딩 동안 획득된 메타-정보에 응답할 뿐만 아니라, 이전에 재구성된 및 루프-필터링된 샘플 값들에 응답할 수도 있는 인-루프 필터(in-loop filter) 기술들을 포함할 수 있다.

[0053] 루프 필터 유닛(456)의 출력은 렌더링 디바이스(412)에 출력될 뿐만 아니라 장래의 인터-픽처 예측에서 사용하기 위해 참조 픽처 메모리(457)에 저장될 수도 있는 샘플 스트림일 수 있다.

[0054] 특정 코딩된 픽처들은, 완전히 재구성되면, 장래의 예측을 위한 참조 픽처들로서 사용될 수 있다. 예를 들어, 현재 픽처에 대응하는 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 참조 픽처로서 식별되면(예를 들어, 파서(420)에 의해), 현재 픽처 버퍼(458)는 참조 픽처 메모리(457)의 일부가 될 수 있고, 다음 코딩된 픽처의 재구성에 착수하기 전에 새로운 현재 픽처 버퍼가 재할당될 수 있다.

[0055] 비디오 디코더(410)는 ITU-T Rec. H.265와 같은 표준 또는 미리 결정된 비디오 압축 기술에 따라 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스가 비디오 압축 기술 또는 표준의 신택스와 비디오 압축 기술 또는 표준에서 문서화된 프로파일들을 둘 다 고수한다는 점에서, 코딩된 비디오 시퀀스는 사용중인 비디오 압축 기술 또는 표준에 의해 특정된 신택스를 준수할 수 있다. 구체적으로, 프로파일은 비디오 압축 기술 또는 표준에서 이용가능한 모든 툴들로부터 해당 프로파일 하에서 사용하기 위해 이용가능한 유일한 툴들로서 특정 툴들을 선택할 수 있다. 또한 준수를 위해, 코딩된 비디오 시퀀스의 복잡도가 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 경계 내에 있는 것이 필요할 수 있다. 일부 경우들에서, 레벨들은 최대 픽처 크기, 최대 프레임 레이트, 최대 재구성 샘플 레이트(예를 들어, 초당 메가샘플수로 측정됨), 최대 참조 픽처 크기 등을 제한한다. 레벨들에 의해 설정된 한계들은, 일부 경우들에서, HRD(Hypothetical Reference Decoder) 사양들 및 코딩된 비디오 시퀀스에서 시그널링된 HRD 버퍼 관리를 위한 메타데이터를 통해 추가로 제한될 수 있다.

[0056] 일 실시예에서, 수신기(431)는 인코딩된 비디오와 함께 추가적인(중복) 데이터를 수신할 수 있다. 이 추가적인 데이터는 코딩된 비디오 시퀀스(들)의 일부로서 포함될 수 있다. 이 추가적인 데이터는 데이터를 적절히 디코딩하고/하거나 원래의 비디오 데이터를 더 정확하게 재구성하기 위해 비디오 디코더(410)에 의해 사용될 수 있다. 추가적인 데이터는 예를 들어, 시간적, 공간적, 또는 신호 잡음 비(SNR) 향상 계층들, 중복 슬라이스들, 중복 픽처들, 순방향 오류 정정 코드들 등의 형태로 될 수 있다.

[0057] 도 5는 본 개시내용의 일 실시예에 따른 비디오 인코더(503)의 블록도를 도시한다. 비디오 인코더(503)는 전자 디바이스(520)에 포함된다. 전자 디바이스(520)는 송신기(540)(예를 들어, 송신 회로)를 포함한다. 비디오 인코더(503)는 도 3의 예에서의 비디오 인코더(303) 대신에 사용될 수 있다.

[0058] 비디오 인코더(503)는 비디오 인코더(503)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(501)(도 5의 예에서는 전자 디바이스(520)의 일부가 아님)로부터 비디오 샘플들을 수신할 수 있다. 다른 예에서, 비디오 소스(501)는 전자 디바이스(520)의 일부이다.

[0059] 비디오 소스(501)는, 임의의 적합한 비트 심도(예를 들어: 8 비트, 10 비트, 12 비트, …), 임의의 색공간(예를 들어, BT.601 Y CrCB, RGB, …), 및 임의의 적합한 샘플링 구조(예를 들어, Y CrCb 4:2:0, Y CrCb 4:4:4)일 수

있는 디지털 비디오 샘플 스트림의 형태로 비디오 인코더(503)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(501)는 이전에 준비된 비디오를 저장하는 저장 디바이스일 수 있다. 영상 회의 시스템에서, 비디오 소스(501)는 비디오 시퀀스로서 로컬 이미지 정보를 캡처하는 카메라일 수 있다. 비디오 데이터는 순차적으로 볼 때 모션을 부여하는 복수의 개별 픽처로서 제공될 수 있다. 픽처들 자체는 픽셀들의 공간적 어레이로서 조직될 수 있고, 여기서 각각의 픽셀은 사용중인 샘플링 구조, 색 공간 등에 의존하여 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 픽셀들과 샘플들 사이의 관계를 쉽게 이해할 수 있다. 아래의 설명은 샘플들에 초점을 맞춘다.

[0060] 일 실시예에 따르면, 비디오 인코더(503)는 소스 비디오 시퀀스의 픽처들을 실시간으로 또는 응용에 의해 요구되는 임의의 다른 시간 제약들 하에서 코딩된 비디오 시퀀스(543)로 코딩 및 압축할 수 있다. 적절한 코딩 속도를 시행하는 것이 제어기(550)의 하나의 기능이다. 일부 실시예들에서, 제어기(550)는 아래 설명되는 바와 같이 다른 기능 유닛들을 제어하고 다른 기능 유닛들에 기능적으로 결합된다. 결합은 명확성을 위해 묘사되어 있지 않다. 제어기(550)에 의해 설정된 파라미터들은 레이트 제어 관련 파라미터들(픽처 스킵, 양자화기, 레이트-왜곡 최적화 기법들의 람다 값, …), 픽처 크기, 픽처 그룹(GOP) 레이아웃, 최대 모션 벡터 검색 범위 등을 포함할 수 있다. 제어기(550)는 특정 시스템 설계에 대해 최적화된 비디오 인코더(503)에 관련된 다른 적합한 기능들을 갖도록 구성될 수 있다.

[0061] 일부 실시예들에서, 비디오 인코더(503)는 코딩 루프에서 동작하도록 구성된다. 과도하게 단순화된 설명으로서, 일 예에서, 코딩 루프는 소스 코더(530)(예를 들어, 코딩될 입력 픽처, 및 참조 픽처(들)에 기초하여 심벌 스트림과 같은 심벌들을 생성하는 것을 담당함), 및 비디오 인코더(503)에 임베드된 (로컬) 디코더(533)를 포함할 수 있다. 디코더(533)는 (원격) 디코더가 또한 생성하는 것과 유사한 방식으로 샘플 데이터를 생성하기 위해 심벌들을 재구성한다(심벌들과 코딩된 비디오 비트스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문임). 재구성된 샘플 스트림(샘플 데이터)은 참조 픽처 메모리(534)에 입력된다. 심벌 스트림의 디코딩이 디코더 위치(로컬 또는 원격)와는 독립적으로 비트-정확한 결과들(bit-exact results)을 야기하기 때문에, 참조 픽처 메모리(534) 내의 콘텐츠도 또한 로컬 인코더와 원격 인코더 사이에서 비트 정확(bit exact)하다. 다시 말해서, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "보는" 것과 정확히 동일한 샘플 값들을 참조 픽처 샘플들로서 "본다". 참조 픽처 동기성(reference picture synchronicity)의 이러한 기본적인 원리(그리고 결과적인 드리프트, 예를 들어, 채널 오류들 때문에 동기성이 유지될 수 없는 경우)는 일부 관련 기술들에서도 사용된다.

[0062] "로컬" 디코더(533)의 동작은 도 4와 관련하여 위에서 이미 상세히 설명한 비디오 디코더(410)와 같은 "원격" 디코더와 동일할 수 있다. 그러나, 또한 도 4를 간단히 참조하면, 심벌들이 이용가능하고 엔트로피 코더(545) 및 파서(420)에 의한 코딩된 비디오 시퀀스로의 심벌들의 인코딩/디코딩이 무손실일 수 있기 때문에, 버퍼 메모리(415), 및 파서(420)를 포함한, 비디오 디코더(410)의 엔트로피 디코딩 부분들은 로컬 디코더(533)에서 완전히 구현되지 않을 수 있다.

[0063] 이 시점에서 이루어질 수 있는 관찰은, 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한 필연적으로, 대응하는 인코더에서, 실질적으로 동일한 기능 형태로 존재할 필요가 있다는 점이다. 이러한 이유로, 개시된 주제는 디코더 동작에 초점을 맞춘다. 인코더 기술들은 포괄적으로 설명된 디코더 기술들의 역(inverse)이기 때문에 그것들에 대한 설명은 축약될 수 있다. 특정 영역들에서만 더 상세한 설명이 요구되고 아래에 제공된다.

[0064] 동작 동안, 일부 예들에서, 소스 코더(530)는, "참조 픽처"로 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 픽처를 참조하여 예측적으로 입력 픽처를 코딩하는, 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(532)은 입력 픽처의 픽셀 블록들과 입력 픽처에 대한 예측 참조(들)로서 선택될 수 있는 참조 픽처(들)의 픽셀 블록들 사이의 차이들을 코딩한다.

[0065] 로컬 비디오 디코더(533)는, 소스 코더(530)에 의해 생성된 심벌들에 기초하여, 참조 픽처들로서 지정될 수 있는 픽처들의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(532)의 동작들은 유리하게는 손실 프로세스들일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 5에 도시되지 않음)에서 디코딩될 수 있는 경우, 재구성된 비디오 시퀀스는 전형적으로 일부 오류들을 갖는 소스 비디오 시퀀스의 복제본(replica)일 수 있다. 로컬 비디오 디코더(533)는 참조 픽처들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고 재구성된 참조 픽처들이 참조 픽처 캐시(534)에 저장되게 할 수 있다. 이러한 방식으로, 비디오 인코더(503)는 (송신 오류들이 없이) 원단(far-end) 비디오 디코더에 의해 획득될 재구성된 참조 픽처로서 공통 콘텐

츠를 갖는 재구성된 참조 픽처들의 사본들을 로컬로 저장할 수 있다.

[0066] 예측자(535)는 코딩 엔진(532)에 대한 예측 검색들을 수행할 수 있다. 즉, 코딩될 새로운 픽처에 대해, 예측자(535)는 새로운 픽처들에 대한 적절한 예측 참조로서 역할을 할 수 있는 참조 픽처 모션 벡터들, 블록 형상들 등과 같은 특정 메타데이터 또는 샘플 데이터(후보 참조 픽셀 블록들로서)에 대해 참조 픽처 메모리(534)를 검색할 수 있다. 예측자(535)는 적절한 예측 참조들을 찾기 위해 샘플 블록-바이-픽셀 블록(sample block-by-pixel block) 기준으로 동작할 수 있다. 일부 경우들에서, 예측자(535)에 의해 획득된 검색 결과들에 의해 결정된 바와 같이, 입력 픽처는 참조 픽처 메모리(534)에 저장된 다수의 참조 픽처로부터 인출된 예측 참조들을 가질 수 있다.

[0067] 제어기(550)는, 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하여, 소스 코더(530)의 코딩 동작을 관리할 수 있다.

[0068] 전술한 모든 기능 유닛들의 출력은 엔트로피 코더(545)에서 엔트로피 코딩을 거칠 수 있다. 엔트로피 코더(545)는 다양한 기능 유닛들에 의해 생성된 심벌들을, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술들에 따라 심벌들을 무손실 압축함으로써, 코딩된 비디오 시퀀스로 변환한다.

[0069] 송신기(540)는, 인코딩된 비디오 데이터를 저장할 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는, 통신 채널(560)을 통한 송신을 준비하기 위해 엔트로피 코더(545)에 의해 생성된 코딩된 비디오 시퀀스(들)를 버퍼링할 수 있다. 송신기(540)는 비디오 코더(503)로부터의 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림(소스들이 도시되지 않음)과 병합할 수 있다.

[0070] 제어기(550)는 비디오 인코더(503)의 동작을 관리할 수 있다. 코딩 동안, 제어기(550)는, 각자의 픽처에 적용될 수 있는 코딩 기법들에 영향을 미칠 수 있는, 특정 코딩된 픽처 타입을 각각의 코딩된 픽처에 할당할 수 있다. 예를 들어, 픽처들은 종종 다음 픽처 타입들 중 하나로서 할당될 수 있다:

[0071] 인트라 픽처(Intra Picture)(I 픽처)는 예측의 소스로서 시퀀스 내의 임의의 다른 픽처를 사용하지 않고 코딩 및 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은, 예를 들어, 독립 디코더 리프레시(Independent Decoder Refresh)("IDR") 픽처들을 포함한, 상이한 타입들의 인트라 픽처들을 허용한다. 본 기술분야의 통상의 기술자는 I 픽처들의 해당 변형들 및 그것들 각자의 응용들 및 특징들을 인식한다.

[0072] 예측 픽처(predictive picture)(P 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 하나의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측(intra prediction) 또는 인터 예측(inter prediction)을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다.

[0073] 양방향 예측 픽처(bi-directionally predictive picture)(B 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 2개의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다. 유사하게, 다중-예측 픽처들은 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.

[0074] 소스 픽처들은 일반적으로 복수의 샘플 블록(예를 들어, 각각 4x4, 8x8, 4x8, 또는 16x16 샘플들의 블록들)으로 공간적으로 세분되고 블록-바이-블록(block-by-block) 기준으로 코딩될 수 있다. 블록들은 블록들의 각자의 픽처들에 적용되는 코딩 할당에 의해 결정된 다른(이미 코딩된) 블록들을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 픽처들의 블록들은 비예측적으로 코딩될 수 있거나 그것들은 동일한 픽처의 이미 코딩된 블록들을 참조하여 예측적으로 코딩될 수 있다(공간적 예측 또는 인트라 예측). P 픽처의 픽셀 블록들은, 하나의 이전에 코딩된 참조 픽처를 참조하여 공간적 예측을 통해 또는 시간적 예측을 통해, 예측적으로 코딩될 수 있다. B 픽처들의 블록들은, 1개 또는 2개의 이전에 코딩된 참조 픽처를 참조하여 공간적 예측을 통해 또는 시간적 예측을 통해, 예측적으로 코딩될 수 있다.

[0075] 비디오 인코더(503)는 ITU-T Rec. H.265와 같은 미리 결정된 비디오 코딩 기술 또는 표준에 따라 코딩 동작들을 수행할 수 있다. 그 동작에서, 비디오 인코더(503)는, 입력 비디오 시퀀스에서 시간적 및 공간적 중복성들을 활용하는 예측 코딩 동작들을 포함한, 다양한 압축 동작들을 수행할 수 있다. 따라서, 코딩된 비디오 데이터는 사용중인 비디오 코딩 기술 또는 표준에 의해 특정된 신택스를 준수할 수 있다.

[0076] 일 실시예에서, 송신기(540)는 인코딩된 비디오와 함께 추가적인 데이터를 송신할 수 있다. 소스 코더(530)는 코딩된 비디오 시퀀스의 일부로서 그러한 데이터를 포함할 수 있다. 추가적인 데이터는 시간적/공간적/SNR 향상 계층들, 중복 픽처들 및 슬라이스들과 같은 다른 형태들의 중복 데이터, SEI 메시지들, VUI 파라미터 세트

프래그먼트들 등을 포함할 수 있다.

[0077] 비디오는 시간적 시퀀스에서 복수의 소스 픽처들(비디오 픽처들)로서 캡처될 수 있다. 인트라-픽처 예측(종종 인트라 예측으로 축약됨)은 주어진 픽처에서 공간적 상관을 이용하고, 인터-픽처 예측은 픽처들 사이의 (시간적 또는 다른) 상관을 이용한다. 일 예에서, 현재 픽처라고 지칭되는 인코딩/디코딩 중인 특정 픽처가 블록들로 파티셔닝된다. 현재 픽처 내의 블록이 비디오 내의 이전에 코딩되고 여전히 버퍼링된 참조 픽처 내의 참조 블록과 유사할 때, 현재 픽처 내의 블록은 모션 벡터라고 지칭되는 벡터에 의해 코딩될 수 있다. 모션 벡터는 참조 픽처 내의 참조 블록을 가리키고, 다수의 참조 픽처가 사용중인 경우, 참조 픽처를 식별하는 제3 차원을 가질 수 있다.

[0078] 일부 실시예들에서, 인터-픽처 예측에서 양예측(bi-prediction) 기법이 사용될 수 있다. 양예측 기법에 따르면, 둘 다 비디오에서 디코딩 순서가 현재 픽처에 앞서는(그러나, 디스플레이 순서는, 각각 과거 및 미래에 있을 수 있는) 제1 참조 픽처 및 제2 참조 픽처와 같은 2개의 참조 픽처가 사용된다. 현재 픽처 내의 블록은 제1 참조 픽처 내의 제1 참조 블록을 가리키는 제1 모션 벡터, 및 제2 참조 픽처 내의 제2 참조 블록을 가리키는 제2 모션 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 예측될 수 있다.

[0079] 또한, 코딩 효율을 개선하기 위해 인터-픽처 예측에서 병합 모드 기법이 사용될 수 있다.

[0080] 본 개시내용의 일부 실시예들에 따르면, 인터-픽처 예측들 및 인트라-픽처 예측들과 같은 예측들이 블록들의 유닛으로 수행된다. 예를 들어, HEVC 표준에 따르면, 비디오 픽처들의 시퀀스 내의 픽처는 압축을 위해 코딩 트리 유닛들(CTU)로 파티셔닝되고, 픽처 내의 CTU들은 64x64 픽셀들, 32x32 픽셀들, 또는 16x16 픽셀들과 같은 동일한 크기를 갖는다. 일반적으로, CTU는 3개의 코딩 트리 블록(coding tree block)(CTB)을 포함하는데, 이는 하나의 루마 CTB 및 2개의 크로마 CTB이다. 각각의 CTU는 하나 또는 다수의 코딩 유닛(CU)으로 재귀적으로 쿼드트리 분할(recursively quadtree split)될 수 있다. 예를 들어, 64x64 픽셀들의 CTU는 64x64 픽셀들의 하나의 CU, 또는 32x32 픽셀들의 4개의 CU, 또는 16x16 픽셀들의 16개의 CU로 분할될 수 있다. 일 예에서, 각각의 CU는, 인터 예측 타입 또는 인트라 예측 타입과 같은, CU에 대한 예측 타입을 결정하기 위해 분석된다. CU는 시간적 및/또는 공간적 예측성에 의존하여 하나 이상의 예측 유닛(PU)으로 분할된다. 일반적으로, 각각의 PU는 루마 예측 블록(PB), 및 2개의 크로마 PB를 포함한다. 일 실시예에서, 코딩(인코딩/디코딩)에서의 예측 동작은 예측 블록의 유닛으로 수행된다. 예측 블록의 예로서 루마 예측 블록을 사용하여, 예측 블록은, 8x8 픽셀들, 16x16 픽셀들, 8x16 픽셀들, 16x8 픽셀들 등과 같은, 픽셀들에 대한 값들(예를 들어, 루마 값들)의 행렬을 포함한다.

[0081] 도 6은 본 개시내용의 다른 실시예에 따른 비디오 인코더(603)의 도면을 도시한다. 비디오 인코더(603)는 비디오 픽처들의 시퀀스에서 현재 비디오 픽처 내의 샘플 값들의 처리 블록(예를 들어, 예측 블록)을 수신하고, 처리 블록을 코딩된 비디오 시퀀스의 일부인 코딩된 픽처 내에 인코딩하도록 구성된다. 일 예에서, 비디오 인코더(603)는 도 3의 예에서의 비디오 인코더(303) 대신에 사용된다.

[0082] HEVC 예에서, 비디오 인코더(603)는 8x8 샘플들 등의 예측 블록과 같은 처리 블록에 대한 샘플 값들의 행렬 등을 수신한다. 비디오 인코더(603)는 처리 블록이, 예를 들어, 레이트-왜곡 최적화를 사용하여 인트라 모드, 인터 모드, 또는 양예측 모드 중 어느 것을 사용하여 최선으로 코딩되는지를 결정한다. 처리 블록이 인트라 모드로 코딩되어야 할 때, 비디오 인코더(603)는 인트라 예측 기법을 사용하여 처리 블록을 코딩된 픽처 내에 인코딩할 수 있으며; 처리 블록이 인터 모드 또는 양예측 모드로 코딩되어야 할 때, 비디오 인코더(603)는 인터 예측 또는 양예측 기법을 각각 사용하여 처리 블록을 코딩된 픽처 내에 인코딩할 수 있다. 특정 비디오 코딩 기술들에서, 병합 모드는 예측자들 외부의 코딩된 모션 벡터 성분의 이점 없이 하나 이상의 모션 벡터 예측자로부터 모션 벡터가 도출되는 인터 픽처 예측 서브모드일 수 있다. 특정 다른 비디오 코딩 기술들에서, 대상 블록에 적용가능한 모션 벡터 성분이 존재할 수 있다. 일 예에서, 비디오 인코더(603)는 처리 블록들의 모드를 결정하기 위한 모드 결정 모듈(도시되지 않음)과 같은 다른 컴포넌트들을 포함한다.

[0083] 도 6의 예에서, 비디오 인코더(603)는 도 6에 도시된 바와 같이 함께 결합된 인터 인코더(inter encoder)(630), 인트라 인코더(intra encoder)(622), 잔차 계산기(residue calculator)(623), 스위치(626), 잔차 인코더(624), 일반 제어기(621), 및 엔트로피 인코더(625)를 포함한다.

[0084] 인터 인코더(630)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 블록을 참조 픽처들 내의 하나 이상의 참조 블록(예를 들어, 이전 픽처들 및 나중 픽처들 내의 블록들)과 비교하고, 인터 예측 정보(예를 들어,

인터 인코딩 기법에 따른 중복 정보의 설명, 모션 벡터들, 병합 모드 정보)를 생성하고, 임의의 적합한 기법을 사용하여 인터 예측 정보에 기초하여 인터 예측 결과들(예를 들어, 예측된 블록)을 계산하도록 구성된다. 일부 예들에서, 참조 픽처들은 인코딩된 비디오 정보에 기초하여 디코딩되는 디코딩된 참조 픽처들이다.

[0085] 인트라 인코더(622)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 일부 경우들에서 블록을 동일한 픽처 내의 이미 코딩된 블록들과 비교하고, 변환 후 양자화된 계수들을 생성하고, 일부 경우들에서 또한 인트라 예측 정보(예를 들어, 하나 이상의 인트라 인코딩 기법에 따른 인트라 예측 방향 정보)를 수신하도록 구성된다. 일 예에서, 인트라 인코더(622)는 또한 동일한 픽처 내의 참조 블록들 및 인트라 예측 정보에 기초하여 인트라 예측 결과들(예를 들어, 예측 블록)을 계산한다.

[0086] 일반 제어기(621)는 일반 제어 데이터를 결정하고 일반 제어 데이터에 기초하여 비디오 인코더(603)의 다른 컴포넌트들을 제어하도록 구성된다. 일 예에서, 일반 제어기(621)는 블록의 모드를 결정하고, 모드에 기초하여 스위치(626)에 제어 신호를 제공한다. 예를 들어, 모드가 인트라 모드일 때, 일반 제어기(621)는 잔차 계산기(623)에 의한 사용을 위해 인트라 모드 결과를 선택하도록 스위치(626)를 제어하고, 인트라 예측 정보를 선택하고 인트라 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(625)를 제어하며; 모드가 인터 모드일 때, 일반 제어기(621)는 잔차 계산기(623)에 의한 사용을 위해 인터 예측 결과를 선택하도록 스위치(626)를 제어하고, 인터 예측 정보를 선택하고 인터 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(625)를 제어한다.

[0087] 잔차 계산기(623)는 수신된 블록과 인트라 인코더(622) 또는 인터 인코더(630)로부터 선택된 예측 결과들 사이의 차이(잔차 데이터)를 계산하도록 구성된다. 잔차 인코더(624)는 잔차 데이터에 기초하여 동작하여 잔차 데이터를 인코딩하여 변환 계수들을 생성하도록 구성된다. 일 예에서, 잔차 인코더(624)는 잔차 데이터를 공간적 도메인으로부터 주파수 도메인으로 변환하고, 변환 계수들을 생성하도록 구성된다. 그 후 변환 계수들에 대해 양자화 처리를 수행하여 양자화된 변환 계수들을 획득한다. 다양한 실시예들에서, 비디오 인코더(603)는 잔차 디코더(628)를 또한 포함한다. 잔차 디코더(628)는 역변환을 수행하고, 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(622) 및 인터 인코더(630)에 의해 적합하게 사용될 수 있다. 예를 들어, 인터 인코더(630)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있고, 인트라 인코더(622)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있다. 디코딩된 블록들은 디코딩된 픽처들을 생성하기 위해 적합하게 처리되고 디코딩된 픽처들은 메모리 회로(도시되지 않음)에 버퍼링되고 일부 예들에서 참조 픽처들로서 사용될 수 있다.

[0088] 엔트로피 인코더(625)는 인코딩된 블록을 포함하도록 비트스트림을 포맷하도록 구성된다. 엔트로피 인코더(625)는 HEVC 표준과 같은 적합한 표준에 따라 다양한 정보를 포함하도록 구성된다. 일 예에서, 엔트로피 인코더(625)는 일반 제어 데이터, 선택된 예측 정보(예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 잔차 정보, 및 다른 적합한 정보를 비트스트림 내에 포함시키도록 구성된다. 개시된 주제에 따르면, 인터 모드 또는 양예측 모드의 병합 서브모드에서 블록을 코딩할 때, 잔차 정보가 존재하지 않는다는 점에 유의한다.

[0089] 도 7은 본 개시내용의 다른 실시예에 따른 비디오 디코더(710)의 도면을 도시한다. 비디오 디코더(710)는 코딩된 비디오 시퀀스의 일부인 코딩된 픽처들을 수신하고, 코딩된 픽처들을 디코딩하여 재구성된 픽처들을 생성하도록 구성된다. 일 예에서, 비디오 디코더(710)는 도 3의 예에서의 비디오 디코더(310) 대신에 사용된다.

[0090] 도 7의 예에서, 비디오 디코더(710)는 도 7에 도시된 바와 같이 함께 결합된 엔트로피 디코더(771), 인터 디코더(780), 잔차 디코더(773), 재구성 모듈(774), 및 인트라 디코더(772)를 포함한다.

[0091] 엔트로피 디코더(771)는, 코딩된 픽처로부터, 코딩된 픽처가 구성되는 신택스 요소들을 나타내는 특정 심벌들을 재구성하도록 구성될 수 있다. 그러한 심벌들은, 예를 들어, 블록이 코딩되는 모드(예컨대, 예를 들어, 인트라 모드, 인터 모드, 양예측(bi-predicted) 모드, 후자의 둘은 병합 서브모드 또는 다른 서브모드에서임), 인트라 디코더(772) 또는 인터 디코더(780) 각각에 의한 예측을 위해 사용되는 특정 샘플 또는 메타데이터를 식별할 수 있는 예측 정보(예컨대, 예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 예를 들어, 양자화된 변환 계수들의 형태로 된 잔차 정보 등을 포함할 수 있다. 일 예에서, 예측 모드가 인터 또는 양예측 모드일 때, 인터 예측 정보가 인터 디코더(780)에 제공되고; 예측 타입이 인트라 예측 타입일 때, 인트라 예측 정보가 인트라 디코더(772)에 제공된다. 잔차 정보에 대해 역양자화가 수행될 수 있고 이는 잔차 디코더(773)에 제공된다.

[0092] 인터 디코더(780)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기초하여 인터 예측 결과들을 생성하도록 구성된다.

[0093] 인트라 디코더(772)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기초하여 예측 결과들을 생성하도록 구성된다.

[0094] 잔차 디코더(773)는 역양자화를 수행하여 탈양자화된 변환 계수들을 추출하고, 탈양자화된 변환 계수들을 처리하여 잔차를 주파수 도메인으로부터 공간적 도메인으로 변환하도록 구성된다. 잔차 디코더(773)는 또한(양자화기 파라미터(QP)를 포함하도록) 특정 제어 정보를 요구할 수 있고, 그 정보는 엔트로피 디코더(771)에 의해 제공될 수 있다(이는 단지 저용량 제어 정보일 수 있으므로 데이터 경로가 묘사되지 않음).

[0095] 재구성 모듈(774)은, 공간적 도메인에서, 잔차 디코더(773)에 의해 출력된 잔차와 예측 결과들(경우에 따라 인터 또는 인트라 예측 모듈에 의해 출력된 것)을 조합하여 재구성된 블록을 형성하도록 구성하고, 재구성된 블록은 재구성된 픽처의 일부일 수 있고, 재구성된 픽처는 결국 재구성된 비디오의 일부일 수 있다. 시각적 품질을 개선하기 위해 디블로킹 동작 등과 같은 다른 적합한 동작들이 수행될 수 있다는 점에 유의한다.

[0096] 비디오 인코더들(303, 503, 및 603), 및 비디오 디코더들(310, 410, 및 710)은 임의의 적합한 기법을 사용하여 구현될 수 있다는 점에 유의한다. 일 실시예에서, 비디오 인코더들(303, 503, 및 603), 및 비디오 디코더들(310, 410, 및 710)은 하나 이상의 집적 회로를 사용하여 구현될 수 있다. 다른 실시예에서, 비디오 인코더들(303, 503, 및 603), 및 비디오 디코더들(310, 410, 및 710)은 소프트웨어 명령어들을 실행하는 하나 이상의 프로세서를 사용하여 구현될 수 있다.

[0097] II. 비디오 컬러 성분들을 개별적으로 코딩

[0098] 비디오 코딩 기술들은, 인코딩될 비디오 시퀀스가 다수의 컬러 평면(예를 들어, 1개의 루마 성분과 2개의 크로마 성분)을 갖는다고 통상적으로 가정한다. 특정 코딩 툴들을 이용함으로써, 컬러 평면들은 공동으로 인코딩될 수 있다. 예를 들어, 동일한 픽처의 루마 및 크로마 성분들은 동일한 파티셔닝 트리(partitioning tree)를 공유할 수 있다. 코딩된 루마 및 크로마 성분들은 동일한 CU로 조직될 수 있다. 크로마 성분들의 코딩은 예측을 위해 루마 성분의 픽셀 값들 또는 잔차 값들을 참조할 수 있다(예를 들어, 교차 성분 선형 모델(cross component linear model, CCLM)). 처리 단계는 3개의 루마 및 크로마 성분을 입력으로서 사용할 수 있다(예를 들어, 액티브 컬러 변환(ACT)). 또는, 2개의 크로마 성분이 공동으로 코딩될 수 있다(예를 들어, 크로마 잔차들의 공동 코딩(joint coding of chroma residuals, JCCR)).

[0099] 그러나, 일부 응용에서는, 비디오가 단색이거나, 또는 비디오의 다수의 컬러 평면이 독립적으로 인코딩되도록 요구된다. 예를 들어, 4:4:4 크로마 포맷을 갖는 비디오의 3개의 컬러 성분은 개별적으로 그리고 독립적으로 코딩되도록 요구될 수 있다. 예를 들어, 비디오의 각각의 컬러 성분은 단색 비디오로서 취급된다. 비디오가 코딩되는 동안 이들 컬러 성분들 사이에는 종속성이 없다. 다수의 성분들에 의존(예를 들어, ACT 및 JCCR)하거나 크로마 성분들에 대해 동작(예를 들어, 블록 기반 델타(또는 차분) 펄스 코드 변조(BDPCM))하는 코딩 툴들이 이용되지 않는다. 비디오의 코딩은 루마 성분에 대해 동작하는 단색 코딩 툴들에 기초한다.

[0100] 상이한 크로마 포맷들을 갖는 비디오들 및 하나 이상의 단색 성분을 포함하는 비디오들의 코딩을 지원하기 위해, 일부 실시예들에서, 2개의 신택스 요소가 표 1에 도시된 바와 같이 정의된다.

표 1

[0101]

chroma_format_idc	separate_colour_plane_flag	크로마 포맷
0	0	**단색**
1	0	4:2:0
2	0	4:2:2
3	0	4:4:4
3	1	4:4:4

[0102] 신택스 요소 chroma_format_idc는 다수의 크로마 포맷들에 대한 인덱스를 제공한다. 정의된 크로마 포맷들은 상이한 크로마 성분 샘플링 구조들에 대응한다. 구체적으로, 단색 샘플링에서는, 명목상 루마 어레이로 간주되는 단 하나의 샘플 어레이가 있다. 4:2:0 샘플링에서, 2개의 크로마 어레이 각각은 루마 어레이의 높이의 1/2 및 폭의 1/2을 가질 수 있다. 4:2:2 샘플링에서, 2개의 크로마 어레이 각각은 루마 어레이의 높이와 동일한 높

이 및 폭의 1/2을 가질 수 있다. 본 개시내용에서 표기법 및 용어의 편의를 위해, 이들 어레이들과 연관된 변수들 및 용어들은 루마 및 크로마로 지칭된다. 2개의 크로마 어레이는 사용 중인 실제 컬러 표현 방법에 관계없이 Cb 및 Cr로 지칭된다. 사용 중인 실제 컬러 표현 방법은 비트스트림으로 송신되는 신택스에 표시될 수 있다.

[0103] 신택스 요소 separate_colour_plane_flag는 비디오 시퀀스의 컬러 성분들이 개별적으로 코딩되도록 요구되는지를 표시한다. 예를 들어, separate_colour_plane_flag가 1인 것은 4:4:4 크로마 포맷의 3개의 컬러 성분이 개별적으로 코딩될 수 있다는 것을 특정한다. separate_colour_plane_flag가 0인 것은 컬러 성분들이 개별적으로 코딩되지 않는다는 것을 특정한다. separate_colour_plane_flag가 존재하지 않을 때, 그것은 0인 것으로 추론된다.

[0104] separate_colour_plane_flag가 1일 때, 코딩된 픽처는 3개의 개별 성분으로 구성되고, 그 각각은 하나의 컬러 평면(예를 들어, Y, Cb, 또는 Cr)의 코딩된 샘플들로 구성되고, 단색 코딩 신택스를 사용한다. 이 경우, 각각의 컬러 평면은 특정 colour_plane_id 값과 연관된다. 디코딩 프로세스들에서는 상이한 colour_plane_id 값들을 갖는 컬러 평면들 사이에 종속성이 없다. 예를 들어, 하나의 colour_plane_id 값을 갖는 단색 픽처의 디코딩 프로세스는 인터 또는 인트라 예측을 위해 상이한 colour_plane_id 값들을 갖는 단색 픽처들로부터의 어떠한 데이터도 사용하지 않는다.

[0105] 4:4:4 샘플링에서, 2개의 크로마 어레이 각각은 루마 어레이와 동일한 높이 및 폭을 가지며, separate_colour_plane_flag의 값에 따라, 이하가 적용될 수 있다. separate_colour_plane_flag가 0이면, 3개의 컬러 평면은 단색 샘플링된 픽처들로서 개별적으로 처리되지 않는다. 그렇지 않으면(separate_colour_plane_flag가 1이면), 3개의 컬러 평면은 단색 샘플링된 픽처들로서 개별적으로 처리된다.

[0106] 일 예에서, 신택스 요소들 chroma_format_idc 및 separate_colour_plane_flag는 표 2에 도시된 바와 같이 시퀀스 파라미터 세트(SPS)에서 시그널링된다. 표 2의 행 11에서, chroma_format_idc가 시그널링된다. 행 12에서, chroma_forma_idc가 4:4:4 크로마 포맷 샘플링 구조를 표시하는지가 검증된다. 행 13에서, chroma_format_idc가 3의 값을 가질 때, 표 2의 SPS를 참조하는 비디오 시퀀스의 성분들이 개별적으로 코딩되는지를 표시하기 위해 separate_colour_plane_flag가 시그널링된다.

[0107] 비디오가 단색 비디오이거나 또는 각각의 성분이 단색인 것처럼 비디오의 각각의 컬러 성분을 인코딩하도록 요구될 때, 공동 컬러 평면 코딩 툴들 또는 크로마 성분 기반 코딩 툴들은 적용가능하지 않고 디스에이블될 수 있다. 그러나, 표 2에 도시된 바와 같이, 이들 적용가능하지 않은 코딩 툴들을 제어하는 수 개의 신택스 요소는 컬러 성분들의 개별 인코딩이 인에이블(또는 요구)되어 있는지와는 독립적으로 시그널링된다. 그 결과, 단색 비디오들에 적용가능하지 않은 일부 코딩 툴들은, 단색 비디오들과 상이한 컬러 평면들의 개별 인코딩이 현재 비디오에 대해 채용되어 바람직하지 않은 충돌들을 야기할 때 여전히 인에이블될 수 있다.

[0108] 구체적으로, 표 2의 행 86에서, 신택스 요소 sps_joint_cbcr_enabled_flag는 행 13에서 시그널링된 sperate_colour_plane_flag에 대한 종속성 없이 시그널링된다. sps_joint_cbcr_enabled_flag는 크로마 잔차들의 공동 코딩(JCCR) 툴이 비디오를 코딩하기 위해 인에이블되어 있는지를 표시할 수 있다. CU의 2개의 크로마 성분이 공동으로 코딩되므로, JCCR 코딩 툴은 단색 코딩 툴이 아니다. sps_joint_cbcr_enabled_flag가 0인 것은 크로마 잔차들의 공동 코딩이 디스에이블되어 있다는 것을 특정한다. sps_joint_cbcr_enabled_flag가 1인 것은 크로마 잔차들의 공동 코딩이 인에이블되어 있다는 것을 특정한다.

[0109] 행들 104-105에서, BDPCM이 인에이블되고 크로마 포맷이 4:4:4일 때, 신택스 요소 sps_bdpcm_chroma_enabled_flag는 행 13에서 시그널링된 sperate_colour_plane_flag에 대한 종속성 없이 시그널링된다. sperate_colour_plane_flag는 크로마에 대한 BDPCM의 툴이 비디오를 코딩하기 위해 인에이블되어 있는지를 표시할 수 있다. 크로마에 대한 BDPCM은 크로마 성분에 적용되는 코딩 툴이고, 따라서 비디오가 단색이거나, 또는 각각의 성분이 단색인 것처럼 비디오의 각각의 컬러 성분을 인코딩하도록 요구되는 경우 디스에이블될 수 있다.

[0110] 시맨틱에 관하여, sps_bdpcm_chroma_enabled_flag가 1인 것은 intra_bdpcm_chroma_flag가 인트라 코딩 유닛들에 대한 코딩 유닛 신택스에 존재할 수 있다는 것을 특정한다. sps_bdpcm_chroma_enabled_flag가 0인 것은 intra_bdpcm_chroma_flag가 인트라 코딩 유닛들에 대한 코딩 유닛 신택스에 존재하지 않는다는 것을 특정한다. 존재하지 않을 때, sps_bdpcm_chroma_enabled_flag의 값은 0인 것으로 추론된다. intra_bdpcm_chroma_flag가 1인 것은 BDPCM이 현재 크로마 코딩 블록들에 적용된다는 것, 즉, 변환이 스킵된다는 것을 특정하며, 인트라 크

로마 예측 모드는 intra_bdpcm_chroma_dir_flag에 의해 특정된다. intra_bdpcm_chroma_flag가 0인 것은 BDPCM이 현재 크로마 코딩 블록들에 적용되지 않는다는 것을 특정한다. intra_bdpcm_chroma_flag가 존재하지 않을 때, 그것은 0인 것으로 추론된다.

[0111] 표 2의 행들 142 및 144에서, 크로마 포맷이 4:4:4일 때, 신택스 요소 sps_act_enabled_flag는 행 13에서 시그널링된 sperate_colour_plane_flag에 대한 종속성 없이 시그널링된다. sps_act_enabled_flag는 ACT 툴이 비디오를 코딩하기 위해 인에이블되어 있는지를 표시할 수 있다. 예를 들어, 원래 컬러 공간에서의 컬러 포맷(예를 들어, RGB)은 3개의 컬러 성분 사이에 높은 상관들을 가질 수 있다. 컬러 공간 변환을 수행함으로써, 컬러 포맷을 원래 컬러 공간으로부터 타겟 컬러 공간으로 변환하여 3개의 컬러 성분 사이의 중복성을 감소시킬 수 있다. 예를 들어, HEVC 또는 VCC에서, ACT는 잔차 블록들을 RGB 컬러 공간으로부터 YCgCo 컬러 공간으로 변환하기 위해 공간적 잔차 도메인에서 수행될 수 있다. 3개의 성분의 잔차 블록들이 입력으로서 사용된다. 따라서, ACT는 단색 비디오 또는 개별적으로 처리된 컬러 성분들을 갖는 비디오에 적용가능하지 않다.

[0112] 시맨틱에 관하여, sps_act_enabled_flag는 적응적 컬러 변환이 인에이블되어 있는지 여부를 특정한다. sps_act_enabled_flag가 1이면, 적응적 컬러 변환이 사용될 수 있고 플래그 cu_act_enabled_flag는 코딩 유닛 신택스에 존재할 수 있다. sps_act_enabled_flag가 0이면, 적응적 컬러 변환이 사용되지 않고 cu_act_enabled_flag는 코딩 유닛 신택스에 존재하지 않는다. sps_act_enabled_flag가 존재하지 않을 때, 그것은 0인 것으로 추론된다.

[0113] [표 2]

1	seq_parameter_set_rbsp( ) {	디스크립터
2	**sps_decoding_parameter_set_id**	u(4)
3	**sps_video_parameter_set_id**	u(4)
4	**sps_max_sub_layers_minus1**	u(3)
5	**sps_reserved_zero_4bits**	u(4)
6	**sps_ptl_dpb_hrd_params_present_flag**	u(1)
7	if(sps_ptl_dpb_hrd_params_present_flag)	
8	profile_tier_level(1, sps_max_sub_layers_minus1)	
9	**gdr_enabled_flag**	u(1)
10	**sps_seq_parameter_set_id**	u(4)
11 **	**chroma_format_idc**	u(2)
12 **	if(chroma_format_idc == 3)	
13 **	**separate_colour_plane_flag**	u(1)
14	**ref_pic_resampling_enabled_flag**	u(1)
15	**sps_seq_parameter_set_id**	ue(v)
16	**chroma_format_idc**	ue(v)
17	**pic_width_max_in_luma_samples**	ue(v)
18	**pic_height_max_in_luma_samples**	ue(v)
19	**sps_log2_ctu_size_minus5**	u(2)
20	**subpics_present_flag**	u(1)
21	**sps_num_subpics_minus1**	u(8)
22	for(i = 0; i <= sps_num_subpics_minus1; i++) {	
23	**subpic_ctu_top_left_x**[i]	u(v)
24	**subpic_ctu_top_left_y**[i]	u(v)
25	**subpic_width_minus1**[i]	u(v)
26	**subpic_height_minus1**[i]	u(v)
27	**subpic_treated_as_pic_flag**[i]	u(1)
28	**loop_filter_across_subpic_enabled_flag**[i]	u(1)
29	}	
30	}	
31	**sps_subpic_id_present_flag**	u(1)
32	if(sps_subpics_id_present_flag) {	
33	**sps_subpic_id_signalling_present_flag**	u(1)
34	if(sps_subpics_id_signalling_present_flag) {	
35	**sps_subpic_id_len_minus1**	ue(v)
36	for(i = 0; i <= sps_num_subpics_minus1; i++)	
37	**sps_subpic_id**[i]	u(v)
38	}	
39	}	
40	**bit_depth_minus8**	ue(v)
41	**min_qp_prime_ts_minus4**	ue(v)

[0114]

42	**sps_weighted_pred_flag**	u(1)
43	**sps_weighted_bipred_flag**	u(1)
44	**log2_max_pic_order_cnt_lsb_minus4**	u(4)
45	**sps_poc_msb_flag**	u(1)
46	if(sps_poc_msb_flag)	
47	**poc_msb_len_minus1**	**ue(v)**
48	if(sps_max_sub_layers_minus1 > 0)	
49	**sps_sub_layer_dpb_params_flag**	u(1)
50	if(sps_ptl_dpb_hrd_params_present_flag)	
51	dpb_parameters(0, sps_max_sub_layers_minus1, sps_sub_layer_dpb_params_flag)	
52	**long_term_ref_pics_flag**	u(1)
53	**inter_layer_ref_pics_present_flag**	u(1)
54	**sps_idr_rpl_present_flag**	u(1)
55	**rpl1_same_as_rpl0_flag**	u(1)
56	for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
57	**num_ref_pic_lists_in_sps[i]**	ue(v)
58	for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
59	ref_pic_list_struct(i, j)	
60	}	
61	if(ChromaArrayType != 0)	
62	**qtbtt_dual_tree_intra_flag**	u(1)
63	**log2_min_luma_coding_block_size_minus2**	ue(v)
64	**partition_constraints_override_enabled_flag**	u(1)
65	**sps_log2_diff_min_qt_min_cb_intra_slice_luma**	ue(v)
66	**sps_log2_diff_min_qt_min_cb_inter_slice**	ue(v)
67	**sps_max_mtt_hierarchy_depth_inter_slice**	ue(v)
68	**sps_max_mtt_hierarchy_depth_intra_slice_luma**	ue(v)
69	if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
70	**sps_log2_diff_max_bt_min_qt_intra_slice_luma**	ue(v)
71	**sps_log2_diff_max_tt_min_qt_intra_slice_luma**	ue(v)
72	}	
73	if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
74	**sps_log2_diff_max_bt_min_qt_inter_slice**	ue(v)
75	**sps_log2_diff_max_tt_min_qt_inter_slice**	ue(v)
76	}	
77	if(qtbtt_dual_tree_intra_flag) {	
78	**sps_log2_diff_min_qt_min_cb_intra_slice_chroma**	ue(v)
79	**sps_max_mtt_hierarchy_depth_intra_slice_chroma**	ue(v)
80	if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
81	**sps_log2_diff_max_bt_min_qt_intra_slice_chroma**	ue(v)
82	**sps_log2_diff_max_tt_min_qt_intra_slice_chroma**	ue(v)
83	}	

[0115]

84	}	
85	**sps_max_luma_transform_size_64_flag**	u(1)
86 **	**sps_joint_cbcr_enabled_flag**	u(1)
87	if(ChromaArrayType != 0) {	
88	**same_qp_table_for_chroma**	u(1)
89	numQpTables = same_qp_table_for_chroma ? 1 : (sps_joint_cbcr_enabled_flag ? 3 : 2)	
90	for(i = 0; i < numQpTables; i++) {	
91	**qp_table_start_minus26**[i]	se(v)
92	**num_points_in_qp_table_minus1**[i]	ue(v)
93	for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	
94	**delta_qp_in_val_minus1**[i][j]	ue(v)
95	**delta_qp_diff_val**[i][j]	ue(v)
96	}	
97	}	
98	}	
99	**sps_sao_enabled_flag**	u(1)
100	**sps_alf_enabled_flag**	u(1)
101	**sps_transform_skip_enabled_flag**	u(1)
102	if(sps_transform_skip_enabled_flag)	
103	**sps_bdpcm_enabled_flag**	u(1)
104 **	if(sps_bdpcm_enabled_flag && chroma_format_idc == 3)	
105 **	**sps_bdpcm_chroma_enabled_flag**	u(1)
106	**sps_ref_wraparound_enabled_flag**	u(1)
107	if(sps_ref_wraparound_enabled_flag)	
108	**sps_ref_wraparound_offset_minus1**	ue(v)
109	**sps_temporal_mvp_enabled_flag**	u(1)
110	if(sps_temporal_mvp_enabled_flag)	
111	**sps_sbtmvp_enabled_flag**	u(1)
112	**sps_amvr_enabled_flag**	u(1)
113	**sps_bdof_enabled_flag**	u(1)
114	if(sps_bdof_enabled_flag)	
115	**sps_bdof_pic_present_flag**	u(1)
116	**sps_smvd_enabled_flag**	u(1)
117	**sps_dmvr_enabled_flag**	u(1)
118	if(sps_dmvr_enabled_flag)	
119	**sps_dmvr_pic_present_flag**	u(1)
120	**sps_mmvd_enabled_flag**	u(1)
121	**sps_isp_enabled_flag**	u(1)
122	**sps_mrl_enabled_flag**	u(1)
123	**sps_mip_enabled_flag**	u(1)
124	if(ChromaArrayType != 0)	
125	**sps_cclm_enabled_flag**	u(1)

[0116]

126	if(sps_cclm_enabled_flag && chroma_format_idc == 1) [Ed. (JC): should sps_cclm_colocated_chroma_flag also be signalled for 422 case since it's used in the decoding process, to be confirmed]	
127	**sps_cclm_colocated_chroma_flag**	u(1)
128	**sps_mts_enabled_flag**	u(1)
129	if(sps_mts_enabled_flag) {	
130	**sps_explicit_mts_intra_enabled_flag**	u(1)
131	**sps_explicit_mts_inter_enabled_flag**	u(1)
132	}	
133	**sps_sbt_enabled_flag**	u(1)
134	**sps_affine_enabled_flag**	u(1)
135	if(sps_affine_enabled_flag) {	
136	**sps_affine_type_flag**	u(1)
137	**sps_affine_amvr_enabled_flag**	u(1)
138	**sps_affine_prof_enabled_flag**	u(1)
139	if(sps_affine_prof_enabled_flag)	
140	**sps_prof_pic_present_flag**	u(1)
141	}	
142 **	if(chroma_format_idc == 3) {	
143	**sps_palette_enabled_flag**	u(1)
144 **	**sps_act_enabled_flag**	u(1)
145	}	
146	**sps_bcw_enabled_flag**	u(1)
147	**sps_ibc_enabled_flag**	u(1)
148	**sps_ciip_enabled_flag**	u(1)
149	if(sps_mmvd_enabled_flag)	
150	**sps_fpel_mmvd_enabled_flag**	u(1)
151	**sps_triangle_enabled_flag**	u(1)
152	**sps_lmcs_enabled_flag**	u(1)
153	**sps_lfnst_enabled_flag**	u(1)
154	**sps_ladf_enabled_flag**	u(1)
155	if(sps_ladf_enabled_flag) {	
156	**sps_num_ladf_intervals_minus2**	u(2)
157	**sps_ladf_lowest_interval_qp_offset**	se(v)
158	for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
159	**sps_ladf_qp_offset**[i]	se(v)
160	**sps_ladf_delta_threshold_minus1**[i]	ue(v)
161	}	
162	}	
163	**sps_scaling_list_enabled_flag**	u(1)
164	**sps_loop_filter_across_virtual_boundaries_disabled_present_flag**	u(1)
165	if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
166	**sps_num_ver_virtual_boundaries**	u(2)

[0117]

167	for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
168	**sps_virtual_boundaries_pos_x[i]**	u(13)
169	**sps_num_hor_virtual_boundaries**	u(2)
170	for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
171	**sps_virtual_boundaries_pos_y[i]**	u(13)
172	}	
173	if(sps_ptl_dpb_hrd_params_present_flag) {	
174	**sps_general_hrd_params_present_flag**	u(1)
175	if(sps_general_hrd_params_present_flag) {	
176	general_hrd_parameters( )	
177	if(sps_max_sub_layers_minus1 > 0)	
178	**sps_sub_layer_cpb_params_present_flag**	u(1)
179	firstSubLayer = sps_sub_layer_cpb_params_present_flag ? 0 : sps_max_sub_layers_minus1	
180	ols_hrd_parameters(firstSubLayer, sps_max_sub_layers_minus1)	
181	}	
182	}	
183	**vui_parameters_present_flag**	u(1)
184	if(vui_parameters_present_flag)	
185	vui_parameters( )	
186	**sps_extension_flag**	**u(1)**
187	if(sps_extension_flag)	
188	while(more_rbsp_data( ))	
189	**sps_extension_data_flag**	u(1)
190	rbsp_trailing_bits( )	
191	}	

[0118]

[0119] III. 단색 비디오 또는 개별적으로 코딩된 성분들을 갖는 비디오에 적용가능하지 않은 코딩 툴들을 디스에이블

[0120] 일부 실시예들에서, 단색 비디오의 코딩 및 예를 들어, 4:4:4 크로마 포맷 비디오의 3개의 컬러 성분의 개별 코딩을 지원하기 위해, 크로마 어레이 타입을 표시하는 변수가 정의된다. 변수는 ChromaArrayType으로 표시된다. 변수 ChromaArrayType는, 비디오가 단색일 때 그리고 비디오의 컬러 성분들이 개별적으로 그리고 독립적으로 인코딩되도록 요구될 때 적용가능하지 않은 코딩 툴들을 디스에이블하는데 사용될 수 있다. separate_colour_plane_flag의 값에 따라, 변수 ChromaArrayType의 값은 다음과 같이 할당될 수 있다:

[0121] - separate_colour_plane_flag가 0이면, ChromaArrayType는 chroma_format_idc(예를 들어, 0, 1, 2, 또는 3)와 동일하게 설정된다.

[0122] - 그렇지 않으면(separate_colour_plane_flag가 1이면), ChromaArrayType는 0으로 설정된다.

[0123] ChromaArrayType가 0일 때, sps_joint_cbcr_enabled_flag, sps_act_enabled_flag, sps_bdpcm_chroma_enabled_flag 등에 의해 인에이블된 코딩 툴들은 디스에이블될 수 있다.

[0124] 표 3은 표 2에 도시된 SPS 신택스의 수정된 버전을 도시한다. 표 3의 행들 84-85에서, ChromaArrayType가 0이 아닌 값을 가질 때 sps_joint_cbcr_enabled_flag가 시그널링된다. ChromaArrayType가 0일 때(이는 표 2의 SPS를 참조하는 현재 비디오가 단색이거나 개별적으로 인코딩된 성분들을 포함한다는 것을 표시함), sps_joint_cbcr_enabled_flag는 시그널링되지 않고, 0인 것으로 추론될 수 있다. 따라서, 크로마 잔차들의 공동 코딩은 디스에이블될 수 있다. 표 2의 예들과 비교하여, sps_joint_cbcr_enabled_flag의 시맨틱은 다음과 같이 수정될 수 있다: sps_joint_cbcr_enabled_flag가 0인 것은 크로마 잔차들의 공동 코딩이 디스에이블되어 있다는 것을 특정한다. sps_joint_cbcr_enabled_flag가 1인 것은 크로마 잔차들의 공동 코딩이 인에이블되어 있다는 것을 특정한다. sps_joint_cbcr_enabled_flag가 존재하지 않을 때, 그것은 0인 것으로 추론된다.

[0125] 표 3의 행들 102-103에서, BDPCM이 인에이블되고 ChromaArrayType가 0이 아닌 값을 가질 때 sps_bdpcm_chroma_enabled_flag가 시그널링된다. ChromaArrayType가 0일 때, sps_bdpcm_chroma_enabled_flag는 시그널링되지 않고, 0인 것으로 추론될 수 있다. 따라서, 크로마에 대한 BDPCM은 디스에이블될 수 있다. sps_bdpcm_chroma_enabled_flag의 시맨틱은 표 2의 예에서와 동일할 수 있다.

[0126] 표 3의 행들 140, 142, 및 143에서, 비디오가 4:4:4 크로마 포맷을 갖고 ChromaArrayType가 0이 아닌 값을 가질 때 sps_act_enabled_flag가 시그널링된다. ChromaArrayType가 0일 때, sps_act_enabled_flag는 시그널링되

지 않고, 0인 것으로 추론될 수 있다. 따라서, ACT는 디스에이블될 수 있다. sps_act_enabled_flag의 시맨틱은 표 2의 예에서와 동일할 수 있다.

[0127] [표 3]

[0128]

1	seq_parameter_set_rbsp( ) {	디스크립터
2	**sps_decoding_parameter_set_id**	u(4)
3	**sps_video_parameter_set_id**	u(4)
4	**sps_max_sub_layers_minus1**	u(3)
5	**sps_reserved_zero_4bits**	u(4)
6	**sps_ptl_dpb_hrd_params_present_flag**	u(1)
7	if(sps_ptl_dpb_hrd_params_present_flag)	
8	profile_tier_level(1, sps_max_sub_layers_minus1)	
9	**gdr_enabled_flag**	u(1)
10	**sps_seq_parameter_set_id**	u(4)
11 **	**chroma_format_idc**	u(2)
12 **	if(chroma_format_idc == 3)	
13 **	**separate_colour_plane_flag**	u(1)
14	**ref_pic_resampling_enabled_flag**	u(1)
15	**pic_width_max_in_luma_samples**	ue(v)
16	**pic_height_max_in_luma_samples**	ue(v)
17	**sps_log2_ctu_size_minus5**	u(2)
18	**subpics_present_flag**	u(1)
19	**sps_num_subpics_minus1**	u(8)
20	for(i = 0; i <= sps_num_subpics_minus1; i++) {	
21	**subpic_ctu_top_left_x**[i]	u(v)
22	**subpic_ctu_top_left_y**[i]	u(v)
23	**subpic_width_minus1**[i]	u(v)
24	**subpic_height_minus1**[i]	u(v)
25	**subpic_treated_as_pic_flag**[i]	u(1)
26	**loop_filter_across_subpic_enabled_flag**[i]	u(1)
27	}	
28	}	
29	**sps_subpic_id_present_flag**	u(1)
30	if(sps_subpics_id_present_flag) {	
31	**sps_subpic_id_signalling_present_flag**	u(1)
32	if(sps_subpics_id_signalling_present_flag) {	
33	**sps_subpic_id_len_minus1**	ue(v)
34	for(i = 0; i <= sps_num_subpics_minus1; i++)	
35	**sps_subpic_id**[i]	u(v)
36	}	
37	}	
38	**bit_depth_minus8**	ue(v)
39	**min_qp_prime_ts_minus4**	ue(v)
40	**sps_weighted_pred_flag**	u(1)
41	**sps_weighted_bipred_flag**	u(1)
42	**log2_max_pic_order_cnt_lsb_minus4**	u(4)
43	**sps_poc_msb_flag**	u(1)
44	if(sps_poc_msb_flag)	
45	**poc_msb_len_minus1**	ue(v)
46	if(sps_max_sub_layers_minus1 > 0)	
47	**sps_sub_layer_dpb_params_flag**	u(1)
48	if(sps_ptl_dpb_hrd_params_present_flag)	
49	dpb_parameters(0, sps_max_sub_layers_minus1, sps_sub_layer_dpb_params_flag)	

[0129]

50	**long_term_ref_pics_flag**	u(1)
51	**inter_layer_ref_pics_present_flag**	u(1)
52	**sps_idr_rpl_present_flag**	u(1)
53	**rpl1_same_as_rpl0_flag**	u(1)
54	for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
55	**num_ref_pic_lists_in_sps**[i]	ue(v)
56	for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
57	ref_pic_list_struct(i, j)	
58	}	
59	if(ChromaArrayType != 0)	
60	**qtbtt_dual_tree_intra_flag**	u(1)
61	**log2_min_luma_coding_block_size_minus2**	ue(v)
62	**partition_constraints_override_enabled_flag**	u(1)
63	**sps_log2_diff_min_qt_min_cb_intra_slice_luma**	ue(v)
64	**sps_log2_diff_min_qt_min_cb_inter_slice**	ue(v)
65	**sps_max_mtt_hierarchy_depth_inter_slice**	ue(v)
66	**sps_max_mtt_hierarchy_depth_intra_slice_luma**	ue(v)
67	if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
68	**sps_log2_diff_max_bt_min_qt_intra_slice_luma**	ue(v)
69	**sps_log2_diff_max_tt_min_qt_intra_slice_luma**	ue(v)
70	}	
71	if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
72	**sps_log2_diff_max_bt_min_qt_inter_slice**	ue(v)
73	**sps_log2_diff_max_tt_min_qt_inter_slice**	ue(v)
74	}	
75	if(qtbtt_dual_tree_intra_flag) {	
76	**sps_log2_diff_min_qt_min_cb_intra_slice_chroma**	ue(v)
77	**sps_max_mtt_hierarchy_depth_intra_slice_chroma**	ue(v)
78	if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
79	**sps_log2_diff_max_bt_min_qt_intra_slice_chroma**	ue(v)
80	**sps_log2_diff_max_tt_min_qt_intra_slice_chroma**	ue(v)
81	}	
82	}	
83	**sps_max_luma_transform_size_64_flag**	u(1)
84 **	if(ChromaArrayType != 0) {	u(1)
85 **	**sps_joint_cbcr_enabled_flag**	
86	**same_qp_table_for_chroma**	u(1)
87	numQpTables = same_qp_table_for_chroma ? 1 : (sps_joint_cbcr_enabled_flag ? 3 : 2)	
88	for(i = 0; i < numQpTables; i++) {	
89	**qp_table_start_minus26**[i]	se(v)
90	**num_points_in_qp_table_minus1**[i]	ue(v)
91	for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	

[0130]

92	**delta_qp_in_val_minus1**[i][j]	ue(v)
93	**delta_qp_diff_val**[i][j]	ue(v)
94	}	
95	}	
96	}	
97	**sps_sao_enabled_flag**	u(1)
98	**sps_alf_enabled_flag**	u(1)
99	**sps_transform_skip_enabled_flag**	u(1)
100	if(sps_transform_skip_enabled_flag)	
101	**sps_bdpcm_enabled_flag**	u(1)
102 **	if(sps_bdpcm_enabled_flag && ChromaArrayType!=0)	
103 **	**sps_bdpcm_chroma_enabled_flag**	u(1)
104	**sps_ref_wraparound_enabled_flag**	u(1)
105	if(sps_ref_wraparound_enabled_flag)	
106	**sps_ref_wraparound_offset_minus1**	ue(v)
107	**sps_temporal_mvp_enabled_flag**	u(1)
108	if(sps_temporal_mvp_enabled_flag)	
109	**sps_sbtmvp_enabled_flag**	u(1)
110	**sps_amvr_enabled_flag**	u(1)
111	**sps_bdof_enabled_flag**	u(1)
112	if(sps_bdof_enabled_flag)	
113	**sps_bdof_pic_present_flag**	u(1)
114	**sps_smvd_enabled_flag**	u(1)
115	**sps_dmvr_enabled_flag**	u(1)
116	if(sps_dmvr_enabled_flag)	
117	**sps_dmvr_pic_present_flag**	u(1)
118	**sps_mmvd_enabled_flag**	u(1)
119	**sps_isp_enabled_flag**	u(1)
120	**sps_mrl_enabled_flag**	u(1)
121	**sps_mip_enabled_flag**	u(1)
122	if(ChromaArrayType != 0)	
123	**sps_cclm_enabled_flag**	u(1)
124	if(sps_cclm_enabled_flag && chroma_format_idc == 1) [Ed. (JC): should sps_cclm_colocated_chroma_flag also be signalled for 422 case since it's used in the decoding process, to be confirmed]	
125	**sps_cclm_colocated_chroma_flag**	u(1)
126	**sps_mts_enabled_flag**	u(1)
127	if(sps_mts_enabled_flag) {	
128	**sps_explicit_mts_intra_enabled_flag**	u(1)
129	**sps_explicit_mts_inter_enabled_flag**	u(1)
130	}	
131	**sps_sbt_enabled_flag**	u(1)
132	**sps_affine_enabled_flag**	u(1)

[0131]

133	if(sps_affine_enabled_flag) {	
134	**sps_affine_type_flag**	u(1)
135	**sps_affine_amvr_enabled_flag**	u(1)
136	**sps_affine_prof_enabled_flag**	u(1)
137	if(sps_affine_prof_enabled_flag)	
138	**sps_prof_pic_present_flag**	u(1)
139	}	
140 **	if(chroma_format_idc = = 3) {	
141	**sps_palette_enabled_flag**	u(1)
142 **	if(ChromaArrayType != 0)	
143 **	**sps_act_enabled_flag**	u(1)
144	}	
145	**sps_bcw_enabled_flag**	u(1)
146	**sps_ibc_enabled_flag**	u(1)
147	**sps_ciip_enabled_flag**	u(1)
148	if(sps_mmvd_enabled_flag)	
149	**sps_fpel_mmvd_enabled_flag**	u(1)
150	**sps_triangle_enabled_flag**	u(1)
151	**sps_lmcs_enabled_flag**	u(1)
152	**sps_lfnst_enabled_flag**	u(1)
153	**sps_ladf_enabled_flag**	u(1)
154	if(sps_ladf_enabled_flag) {	
155	**sps_num_ladf_intervals_minus2**	u(2)
156	**sps_ladf_lowest_interval_qp_offset**	se(v)
157	for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
158	**sps_ladf_qp_offset**[i]	se(v)
159	**sps_ladf_delta_threshold_minus1**[i]	ue(v)
160	}	
161	}	
162	**sps_scaling_list_enabled_flag**	u(1)
163	**sps_loop_filter_across_virtual_boundaries_disabled_present_flag**	u(1)
164	if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
165	**sps_num_ver_virtual_boundaries**	u(2)
166	for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
167	**sps_virtual_boundaries_pos_x**[i]	u(13)
168	**sps_num_hor_virtual_boundaries**	u(2)
169	for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
170	**sps_virtual_boundaries_pos_y**[i]	u(13)
171	}	
172	if(sps_ptl_dpb_hrd_params_present_flag) {	
173	**sps_general_hrd_params_present_flag**	u(1)
174	if(sps_general_hrd_params_present_flag) {	
175	general_hrd_parameters( )	

[0132]

176	if(sps_max_sub_layers_minus1 > 0)	
177	**sps_sub_layer_cpb_params_present_flag**	u(1)
178	firstSubLayer = sps_sub_layer_cpb_params_present_flag ? 0 : sps_max_sub_layers_minus1	
179	ols_hrd_parameters(firstSubLayer, sps_max_sub_layers_minus1)	
180	}	
181	}	
182	**vui_parameters_present_flag**	u(1)
183	if(vui_parameters_present_flag)	
184	vui_parameters( )	
185	**sps_extension_flag**	u(1)
186	if(sps_extension_flag)	
187	while(more_rbsp_data( ))	
188	**sps_extension_data_flag**	u(1)
189	rbsp_trailing_bits( )	
190	}	

[0133]

[0134] 일부 실시예들에서, 단색 비디오 또는 개별적으로 인코딩된 성분들을 포함하는 비디오에 적용가능하지 않은 코딩 툴들을 디스에이블하기 위해 대안적인 구현들이 이용된다.

[0135] 일 실시예에서, (표 3에서 복사한) 다음의 표 4의 sps_act_enabled_flag 신택스는 chroma_format_idc가 3이고 ChromaArrayType가 0일 때 sps_act_enabled_flag의 값을 0으로 설정하기 위해 표 5에 나타낸 대안적인 신택스로 표현될 수 있다. chroma_format_idc==3 및 separate_colour_plan_flag==0은 ChromaArrayType가 0이 아님을 암시하기 때문에, 표 4 및 표 5의 신택스는 동일한 효과를 가질 수 있다. sps_act_enabled_flag의 시그널링은 행 141에서의 sps_palette_enabled_flag의 시그널링과 독립적이라는 점에 유의한다.

표 4

140 **	if(chroma_format_idc = = 3) {	
141	**sps_palette_enabled_flag**	u(1)
142 **	if(ChromaArrayType != 0)	
143 **	**sps_act_enabled_flag**	u(1)
144	}	

[0136]

표 5

140 **	if(chroma_format_idc = = 3) {	
141	**sps_palette_enabled_flag**	u(1)
142 **	if(separate_colour_plane_flag== 0)	
143 **	**sps_act_enabled_flag**	u(1)
144	}	

[0137]

[0138] 일 실시예에서, (표 3에서 복사한) 다음의 표 6의 sps_bdpcm_chroma_enabled_flag 신택스는 ChromaArrayType가 0일 때 sps_bdpcm_chroma_enabled_flag의 값을 0으로 설정하기 위해 표 7에 나타낸 바와 같은 대안적인 신택스로 표현될 수 있다. 표 7에서, ChromaArrayType가 0이거나 무손실에 대한 BDPCM을 지원하는 sps_transquant_bypass_flag가 0일 때 sps_bdpcm_chroma_enabled_flag의 값은 0인 것으로 추론된다. sps_transquant_bypass_flag가 1일 때, sps_transquant_bypass 플래그는 변환 및 양자화 바이패스가 CU 레벨에서 활성화되어야 한다는 것을 표시한다. 그렇지 않고, sps_transquant_bypass_flag가 0이면, 변환 및 양자화 바이패스는 활성화되지 않아야 한다. sps_transquant_bypass_flag는 SPS에서 시그널링될 수 있거나 다른 SPS 레벨 무손실 코딩 표시 플래그에 의해 표시된 바와 같이 무손실 코딩에 대해 1인 것으로 추론될 수 있다.

표 6

[0139]

102 **	if(sps_bdpcm_enabled_flag && ChromaArrayType!=0)	
103 **	**sps_bdpcm_chroma_enabled_flag**	u(1)

표 7

[0140]

102 **	if(sps_bdpcm_enabled_flag && sps_transquant_bypass_flag &&ChromaArrayType!=0)	
103 **	**sps_bdpcm_chroma_enabled_flag**	u(1)

[0141] 일 실시예에서, 표 7의 신택스가 사용된다. 그러나, 위의 실시예와 달리, 신택스 요소 sps_transquant_bypass_flag의 시맨틱은 다음과 같이 정의된다. sps_transquant_bypass_flag가 1일 때, sps_transquant_bypass 플래그는 변환 및 양자화 바이패스가 CU 레벨에서 (활성화되어야만 하는 대신에) 활성화될 수도 있다는 것을 표시한다. 그렇지 않고, sps_transquant_bypass_flag가 0이면, 변환 및 양자화 바이패스는 활성화되지 않아야 한다. sps_transquant_bypass_flag는 SPS에서 시그널링될 수 있거나 다른 SPS 레벨 무손실 코딩 표시 플래그에 의해 표시된 바와 같이 무손실 코딩에 대해 1인 것으로 추론될 수 있다.

[0142] 도 8은 코딩된 비디오의 비트스트림에서 교차 성분 코딩 툴들 또는 크로마 성분 기반 코딩 툴들의 플래그들을 수신하는 예시적인 프로세스(800)를 도시한다. 프로세스(800)는 디코더에서 수행될 수 있다. 프로세스(800)는 (S801)로부터 시작하여 (S810)으로 진행할 수 있다.

[0143] (S810)에서, 시퀀스의 픽처들 각각이 단색이거나 개별적으로 인코딩된 성분들을 갖는지를 표시하는 신택스 요소가 비트스트림에서 수신될 수 있다. 예를 들어, 신택스 요소는 chroma_format_idc, 또는 separate_colour_plane_flag일 수 있다. chroma_format_idc가 0인 것은 픽처들이 단색이라는 것을 표시할 수 있다. separate_colour_plane_flag가 1인 것은 픽처들 각각이 개별적으로 코딩된 성분들을 갖는 것을 표시할 수 있다. 두 경우 모두에서, ChromaArrayType는 0의 값을 가질 수 있다.

[0144] 예로서, 픽처들의 시퀀스는 표 3의 SPS를 참조한다. 표 3의 행 11에서 시그널링된 chroma_format_idc가 수신된다. chroma_format_idc가 0의 값을 가질 때, 픽처들이 단색인 것으로 결정될 수 있다. chroma_format_idc가 0일 때, separate_colour_plane_flag는 행 13에서 시그널링되지 않을 수 있고, 일 예에서 0인 것으로 추론될 수 있다. 따라서, ChromaArrayType는 현재 경우에 대해 0인 chroma_format_idc와 동일하게 설정될 수 있다.

[0145] chroma_format_idc가 3의 값을 가질 때(이는 시퀀스의 픽처들의 4:4:4 크로마 포맷을 표시함), separate_colour_plane_flag는 행 13에서 수신될 수 있다. separate_colour_plane_flag가 1의 값을 갖는 경우(이는 픽처들이 성분들을 개별적으로 코딩하도록 요구된다는 것을 표시함), 픽처들이 개별적으로 인코딩된 성분들을 갖는 것으로 결정될 수 있다. ChromaArrayType는 0으로 설정될 수 있다.

[0146] 다른 경우들에서, 행 11에서 수신된 chroma_format_idc가 1 또는 2의 값을 가질 때, 또는 행 11에서 수신된 chroma_format_idc가 3의 값을 갖지만 separate_colour_plane_flag가 0의 값을 가질 때, 픽처들 각각은 단색이 아니고 개별적으로 코딩되지 않은 성분들을 포함하는 것으로 결정될 수 있다. 공동 성분 코딩 툴들 또는 크로마 기반 코딩 툴들이 픽처들에 적용될 수 있다. 표 3의 행 11에서 수신된 chroma_format_idc가 1 또는 2의 값을 갖는 경우(픽처들은 단색이 아님), separate_colour_plane_flag는 0으로서 추론될 수 있다. 따라서, ChromeArrayType은 1 또는 2(0이 아님)인 chroma_format_idc의 값을 취할 수 있다. 수신된 chroma_format_idc가 3의 값을 갖지만 separate_colour_plane_flag가 0의 값을 갖는 시나리오에 대해, ChromeArrayType은 3(0이 아님)인 chroma_format_idc의 값을 여전히 취할 수 있다.

[0147] 픽처들이 단색이거나 개별적으로 인코딩된 성분들을 포함하는 것으로 결정되면, 또는 ChromaArrayType가 0인 것으로 결정되면, (S820) 내지(S840)의 단계들이 수행될 수 있다. 공동 성분 코딩 툴 또는 크로마 기반 코딩 툴의 제어를 위한 신택스 요소들은 이들 코딩 툴을 디스에이블하기 위해 0인 것으로 추론될 수 있다. 구체적으로, sps_joint_cbcr_enabled_flag, sps_bdpcm_chroma_enabled_flag, 및 sps_act_enabled_flag는 각각 0인 것으로 추론된다.

[0148] 픽처들이 단색이 아니고 개별적으로 인코딩되지 않은 성분들을 포함하는 것으로 결정되면, 또는 chromeArrayType가 0이 아닌 것으로 결정되면, (S850) 내지 (S870)의 단계들이 수행될 수 있다. 공동 성분 코딩 툴 또는 크로마 기반 코딩 툴의 제어를 위한 신택스 요소들은 비트스트림으로부터 수신될 수 있다. 구체적으로, sps_joint_cbcr_enabled_flag, sps_bdpcm_chroma_enabled_flag, 및 sps_act_enabled_flag는 연속적으로 수신될 수 있다.

[0149] (S840) 또는 (S870) 후에, 프로세스(800)는 (S899)로 진행하여 (S899)에서 종료할 수 있다.

[0150] 도 9는 본 개시내용의 일 실시예에 따른 단색 비디오 또는 개별적으로 인코딩된 성분들을 포함하는 비디오에 적용가능하지 않은 코딩 툴들을 디스에이블하는 프로세스(900)를 도시한다. 프로세스(900)는 디코더(710)와 같은 디코더에서 수행될 수 있다. 프로세스(900)는 (S901)로부터 시작하여 (S910)으로 진행할 수 있다.

[0151] (S910)에서, 픽처들의 시퀀스가 각각 단색이거나 개별적으로 인코딩된 성분들을 갖는지를 표시하는 신택스 요소가 비트스트림에서 수신될 수 있다. 예를 들어, 신택스 요소는 표 3의 예에서 chroma_format_idc, 또는 separate_colour_plane_flag일 수 있다. chroma_format_idc가 0인 것은 픽처들이 단색이라는 것을 표시할 수 있다. separate_colour_plane_flag가 1인 것은 픽처들 각각이 개별적으로 코딩된 성분들을 갖는 것을 표시할 수 있다. 두 경우 모두에서(chroma_format_idc가 0이거나 separate_colour_plane_flag가 1임), 변수 ChromaArrayType는 0의 값을 가질 수 있다.

[0152] 픽처들의 시퀀스가 각각 단색이거나 개별적으로 인코딩된 성분들을 갖는다고 결정될 때, 단계들 (S920) 및 (S930)이 수행될 수 있다. (S920)에서, 예를 들어, 각각의 대응하는 코딩 툴을 제어하는 신택스 요소의 값을 추론함으로써, 픽처들의 다수의 성분들을 입력으로서 사용하는 코딩 툴들이 디스에이블될 수 있다. 이러한 코딩 툴들의 예들은 ACT, 크로마 잔차들의 공동 코딩 등을 포함할 수 있다.

[0153] (S930)에서, 예를 들어, 각각의 대응하는 코딩 툴을 제어하는 신택스 요소의 값을 추론함으로써, 픽처들의 크로마 성분에 의존하는 코딩 툴들이 디스에이블될 수 있다. 이러한 코딩 툴들의 예들은 크로마에 대한 BDPCM을 포함할 수 있다. 이어서 프로세스(900)는 (S999)로 진행하여 (S999)에서 종료할 수 있다.

[0154] (S910)에서 픽처들의 시퀀스가 각각 단색이 아니고 개별적으로 인코딩된 성분들을 갖지 않는다고 결정될 때, 단계 (S940)이 수행될 수 있다. (S940)에서, 공동 성분 코딩 툴들 또는 크로마 성분 기반 코딩 툴들을 인에이블하기 위한 신택스 요소들이 비트 스트림으로부터 수신될 수 있다. 이들 코딩 툴들을 인에이블하기 위한 신택스 요소들이 비트스트림에서 시그널링되는지는 비트스트림에서 송신되는 다른 조건들 또는 다른 신택스 요소들에 의존할 수 있다. 이어서 프로세스(900)는 (S999)로 진행하여 (S999)에서 종료할 수 있다.

[0155] IV. 컴퓨터 시스템

[0156] 위에서 설명된 기법들은 컴퓨터 판독가능 명령어들을 사용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 판독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 10은 개시된 주제의 특정 실시예들을 구현하기에 적합한 컴퓨터 시스템(1000)을 도시한다.

[0157] 컴퓨터 소프트웨어는, 하나 이상의 컴퓨터 중앙 처리 유닛(CPU), 그래픽 처리 유닛(GPU) 등에 의해, 직접, 또는 해석, 마이크로-코드 실행 등을 통해 실행될 수 있는 명령어들을 포함하는 코드를 생성하기 위해 어셈블리, 컴파일(compilation), 링킹(linking), 또는 유사한 메커니즘들이 수행될 수 있는 임의의 적합한 머신 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.

[0158] 명령어들은, 예를 들어, 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게이밍 디바이스, 사물 인터넷 디바이스 등을 포함하여, 다양한 타입의 컴퓨터들 또는 그것의 컴포넌트들 상에서 실행될 수 있다.

[0159] 컴퓨터 시스템(1000)에 대한 도 10에 도시된 컴포넌트들은 사실상 예시적인 것이고, 본 개시내용의 실시예들을 구현하는 컴퓨터 소프트웨어의 사용 또는 기능성의 범위에 대한 임의의 제한을 암시하도록 의도되지 않는다. 컴포넌트들의 구성이 컴퓨터 시스템(1000)의 예시적인 실시예에서 예시된 컴포넌트들 중 임의의 하나 또는 이들의 조합과 관련하여 임의의 종속성 또는 요건을 갖는 것으로 해석되어서도 안 된다.

[0160] 컴퓨터 시스템(1000)은 특정 휴먼 인터페이스 입력 디바이스들을 포함할 수 있다. 그러한 휴먼 인터페이스 입력 디바이스는, 예를 들어, 촉각 입력(예컨대: 키스트로크, 스와이프, 데이터 글러브 움직임), 오디오 입력(예컨대: 음성, 손뼉), 시각적 입력(예컨대, 제스처), 후각적 입력(묘사되지 않음)을 통한 하나 이상의 인간 사용자에 의한 입력에 응답할 수 있다. 휴먼 인터페이스 디바이스들은 또한 오디오(예컨대: 음성, 음악, 주변 사운드), 이미지들(예컨대: 스캐닝된 이미지들, 스틸 이미지 카메라로부터 획득된 사진 이미지들), 비디오(예컨대 2

차원 비디오, 입체적 비디오를 포함하는 3차원 비디오)와 같은, 인간에 의한 의식적인 입력과 반드시 직접적으로 관련되지는 않는 특정 미디어를 캡처하기 위해 사용될 수 있다.

[0161] 입력 휴먼 인터페이스 디바이스들은: 키보드(1001), 마우스(1002), 트랙패드(1003), 터치 스크린(1010), 데이터-글러브(도시되지 않음), 조이스틱(1005), 마이크로폰(1006), 스캐너(1007), 카메라(1008) 중 하나 이상(각각의 하나만이 묘사됨)을 포함할 수 있다.

[0162] 컴퓨터 시스템(1000)은 특정 휴먼 인터페이스 출력 디바이스들을 또한 포함할 수 있다. 그러한 휴먼 인터페이스 출력 디바이스들은, 예를 들어, 촉각 출력, 사운드, 광, 및 냄새/맛을 통해 하나 이상의 인간 사용자의 감각들을 자극하고 있을 수 있다. 그러한 휴먼 인터페이스 출력 디바이스들은 촉각 출력 디바이스들(예를 들어 터치-스크린(1010), 데이터-글러브(도시되지 않음), 또는 조이스틱(1005)에 의한 촉각 피드백, 그러나 입력 디바이스들로서 역할을 하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(예컨대: 스피커들(1009), 헤드폰들(묘사되지 않음)), 시각적 출력 디바이스들(예컨대, 각각 터치-스크린 입력 능력이 있거나 없고, 각각 촉각 피드백 능력이 있거나 없는, CRT 스크린들, LCD 스크린들, 플라즈마 스크린들, OLED 스크린들을 포함하는 스크린들(1010) - 이들 중 일부는 스테레오그래픽 출력과 같은 수단을 통해 2차원 시각적 출력 또는 3차원을 초과한 출력을 출력할 수 있음 - ; 가상 현실 안경(묘사되지 않음), 홀로그래픽 디스플레이들 및 스모크 탱크들(묘사되지 않음)), 및 프린터들(묘사되지 않음)을 포함할 수 있다.

[0163] 컴퓨터 시스템(1000)은 인간 액세스가능한 저장 디바이스들 및 그의 연관된 매체들, 예컨대 CD/DVD 등의 매체(1021)를 갖는 CD/DVD ROM/RW(1020)를 포함하는 광학 매체, 썸-드라이브(thumb-drive)(1022), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(1023), 테이프 및 플로피 디스크(묘사되지 않음)와 같은 레거시 자기 매체, 보안 동글(묘사되지 않음)과 같은 특수화된 ROM/ASIC/PLD 기반 디바이스들 등을 또한 포함할 수 있다.

[0164] 본 기술분야의 통상의 기술자들은 또한, 현재 개시된 주제와 관련하여 사용되는 용어 "컴퓨터 판독가능 매체"가 송신 매체, 반송파들, 또는 다른 일시적 신호들을 포함하지 않는다는 것을 이해해야 한다.

[0165] 컴퓨터 시스템(1000)은 하나 이상의 통신 네트워크(1055)에 대한 인터페이스(1054)를 또한 포함할 수 있다. 네트워크들은 예를 들어 무선, 유선, 광학일 수 있다. 네트워크들은 추가로 로컬, 광역, 대도시, 차량 및 산업, 실시간, 지연-허용(delay-tolerant) 등일 수 있다. 네트워크들의 예들은 로컬 영역 네트워크들, 예컨대 이더넷, 무선 LAN들, GSM, 3G, 4G, 5G, LTE 등을 포함하는 셀룰러 네트워크들, 케이블 TV, 위성 TV 및 지상파 브로드캐스트 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크들, CANBus를 포함하는 차량 및 산업 등을 포함한다. 특정 네트워크들은 일반적으로 특정 범용 데이터 포트들 또는 주변 버스들(1049)(예컨대, 예를 들어, 컴퓨터 시스템(1000)의 USB 포트들)에 부착되는 외부 네트워크 인터페이스 어댑터들을 요구하며; 다른 것들은 일반적으로 아래에 설명되는 바와 같은 시스템 버스(예를 들어, PC 컴퓨터 시스템으로의 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템으로의 셀룰러 네트워크 인터페이스)로의 부착에 의해 컴퓨터 시스템(1000)의 코어에 통합된다. 이들 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(1000)은 다른 엔티티들과 통신할 수 있다. 그러한 통신은 단방향성 수신 전용(예를 들어, 브로드캐스트 TV), 단방향성 송신 전용(예를 들어, CANbus 대 특정 CANbus 디바이스들), 또는 예를 들어 로컬 영역 또는 광역 디지털 네트워크들을 사용하는 다른 컴퓨터 시스템들과의 양방향성일 수 있다. 위에서 설명한 바와 같은 네트워크들 및 네트워크 인터페이스들 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0166] 전술한 휴먼 인터페이스 디바이스들, 인간-액세스가능한 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(1000)의 코어(1040)에 부착될 수 있다.

[0167] 코어(1040)는 하나 이상의 중앙 처리 유닛(CPU)(1041), 그래픽 처리 유닛(GPU)(1042), 필드 프로그래머블 게이트 영역(FPGA)(1043)의 형태로 된 특수화된 프로그래머블 처리 유닛, 특정 태스크들에 대한 하드웨어 가속기(1044), 그래픽 어댑터(1050) 등을 포함할 수 있다. 이들 디바이스는, 판독 전용 메모리(ROM)(1045), 랜덤 액세스 메모리(1046), 내부 비-사용자 액세스가능 하드 드라이브들, SSD들 등과 같은 내부 대용량 스토리지(1047)와 함께, 시스템 버스(1048)를 통해 접속될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(1048)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형태로 액세스가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(1048)에 직접, 또는 주변 버스(1049)를 통해 부착될 수 있다. 일 예에서, 스크린(1010)은 그래픽 어댑터(1050)에 접속될 수 있다. 주변 버스를 위한 아키텍처들은 PCI, USB 등을 포함한다.

[0168] CPU들(1041), GPU들(1042), FPGA들(1043), 및 가속기들(1044)은, 조합하여, 전술한 컴퓨터 코드를 구성할 수

있는 특정 명령어들을 실행할 수 있다. 그 컴퓨터 코드는 ROM(1045) 또는 RAM(1046)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(1046)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 스토리지(1047)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은, 하나 이상의 CPU(1041), GPU(1042), 대용량 스토리지(1047), ROM(1045), RAM(1046) 등과 밀접하게 연관될 수 있는, 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

[0169] 컴퓨터 판독가능 매체는 다양한 컴퓨터 구현 동작들을 수행하기 위한 컴퓨터 코드를 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시내용의 목적을 위해 특별히 설계되고 구성된 것들일 수 있거나, 또는 그것들은 컴퓨터 소프트웨어 기술분야의 기술자들에게 잘 알려져 있고 이용가능한 종류의 것일 수 있다.

[0170] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(1000), 및 구체적으로 코어(1040)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형의(tangible) 컴퓨터 판독가능 매체에 구현된 소프트웨어를 실행하는 결과로서 기능성을 제공할 수 있다. 그러한 컴퓨터 판독가능 매체는 위에 소개된 바와 같은 사용자-액세스가능한 대용량 스토리지뿐만 아니라, 코어-내부 대용량 스토리지(1047) 또는 ROM(1045)과 같은 비일시적인 본질의 것인 코어(1040)의 특정 스토리지와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 그러한 디바이스들에 저장되고 코어(1040)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 필요에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(1040) 및 구체적으로 그 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(1046)에 저장된 데이터 구조들을 정의하는 것 및 소프트웨어에 의해 정의된 프로세스들에 따라 그러한 데이터 구조들을 수정하는 것을 포함하여, 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하게 할 수 있다. 추가로 또는 대안으로서, 컴퓨터 시스템은, 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하기 위해 소프트웨어 대신에 또는 그와 함께 동작할 수 있는, 회로(예를 들어: 가속기(1044))에 하드와이어링되거나 다른 방식으로 구현된 로직의 결과로서 기능성을 제공할 수 있다. 소프트웨어에 대한 참조는, 적절한 경우, 로직을 포함할 수 있고, 그 반대도 가능하다. 컴퓨터 판독가능 매체에 대한 참조는, 적절한 경우, 실행을 위한 소프트웨어를 저장하는 회로(예컨대 집적 회로(IC)), 또는 실행을 위한 로직을 구현하는 회로, 또는 둘 다를 포함할 수 있다. 본 개시내용은 하드웨어 및 소프트웨어의 임의의 적합한 조합을 포함한다.

[0171] 부록 A: 두문자어들

[0172] VTM: Versatile Video Coding Test Model

[0173] SPS: sequence parameter set

[0174] BDPCM: Block-based Delta Pulse Code Modulation

[0175] ACT: Adaptive Colour Transform

[0176] JEM: joint exploration model

[0177] VVC: versatile video coding

[0178] BMS: benchmark set

[0179] MV: Motion Vector

[0180] HEVC: High Efficiency Video Coding

[0181] SEI: Supplementary Enhancement Information

[0182] VUI: Video Usability Information

[0183] GOPs: Groups of Pictures

[0184] TUs: Transform Units

[0185] PUs: Prediction Units

[0186] CTUs: Coding Tree Units

[0187] CTBs: Coding Tree Blocks

[0188] PBs: Prediction Blocks

[0189] HRD: Hypothetical Reference Decoder

[0190] SNR: Signal Noise Ratio

[0191] CPUs: Central Processing Units

[0192] GPUs: Graphics Processing Units

[0193] CRT: Cathode Ray Tube

[0194] LCD: Liquid-Crystal Display

[0195] OLED: Organic Light-Emitting Diode

[0196] CD: Compact Disc

[0197] DVD: Digital Video Disc

[0198] ROM: Read-Only Memory

[0199] RAM: Random Access Memory

[0200] ASIC: Application-Specific Integrated Circuit

[0201] PLD: Programmable Logic Device

[0202] LAN: Local Area Network

[0203] GSM: Global System for Mobile communications

[0204] LTE: Long-Term Evolution

[0205] CANBus: Controller Area Network Bus

[0206] USB: Universal Serial Bus

[0207] PCI: Peripheral Component Interconnect

[0208] FPGA: Field Programmable Gate Areas

[0209] SSD: solid-state drive

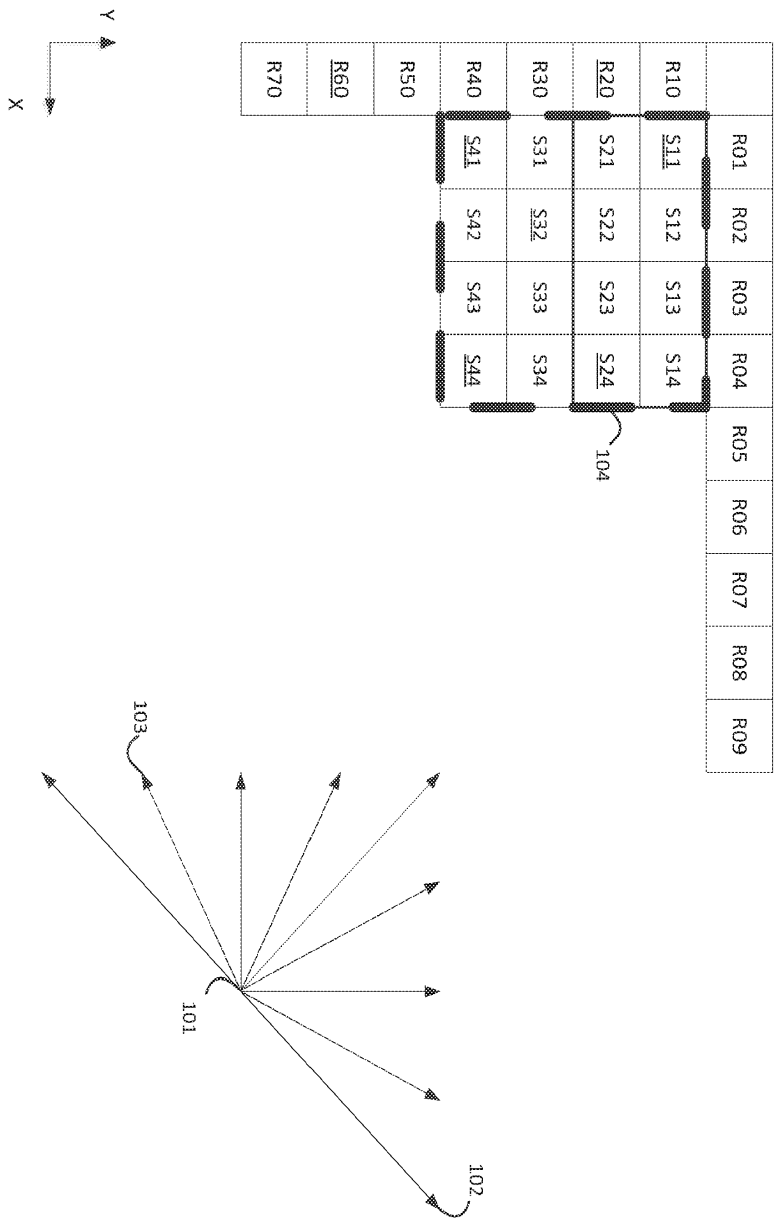
[0210] IC: Integrated Circuit

[0211] CU: Coding Unit

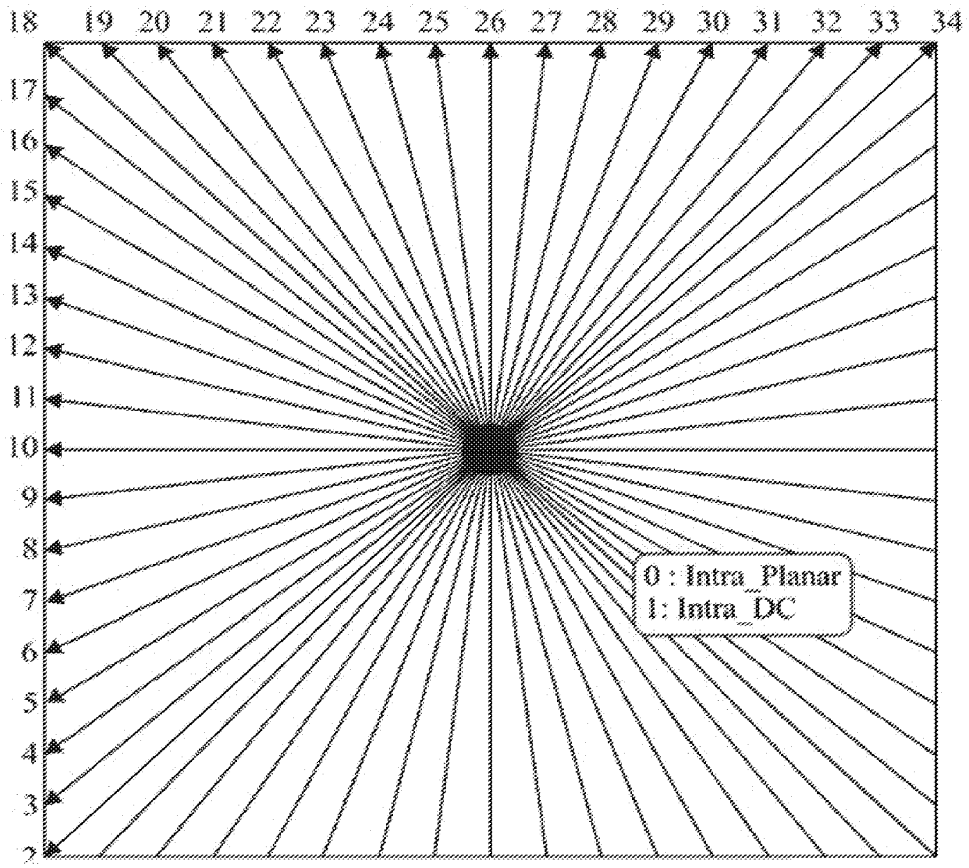
[0212] 본 개시내용이 여러 예시적인 실시예들을 설명하였지만, 본 개시내용의 범위 내에 속하는 변경들, 치환들, 및 다양한 대체 균등물들이 존재한다. 따라서, 본 기술분야의 통상의 기술자들은, 비록 본 명세서에 명시적으로 도시되거나 설명되지는 않았지만, 본 개시내용의 원리들을 구현하고 따라서 그것의 진의 및 범위 내에 있는, 다수의 시스템들 및 방법들을 고안할 수 있을 것이라는 점이 인정될 것이다.

도면

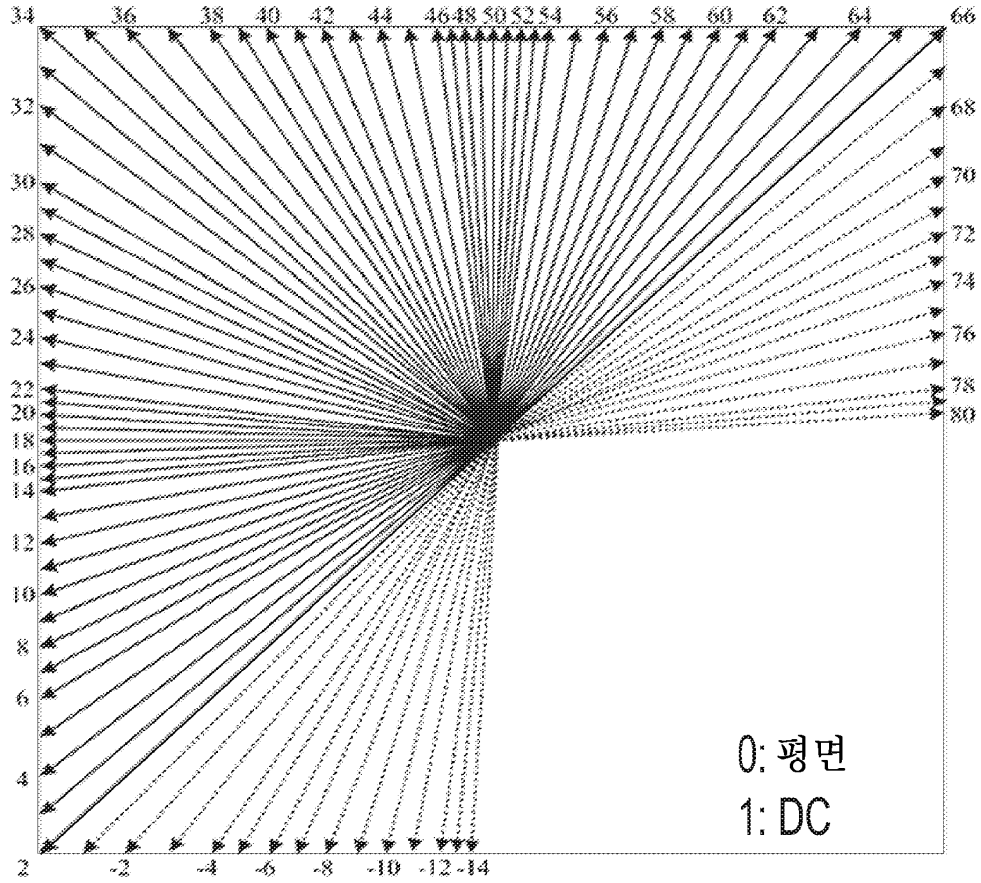
도면1a


Y
X
R70
R60
R50
R40
R30
R20
R10
R01
R02
R03
R04
R05
R06
R07
R08
R09
S41
S31
S21
S11
S42
S32
S22
S12
S43
S33
S23
S13
S44
S34
S24
S14
104
103
101
102

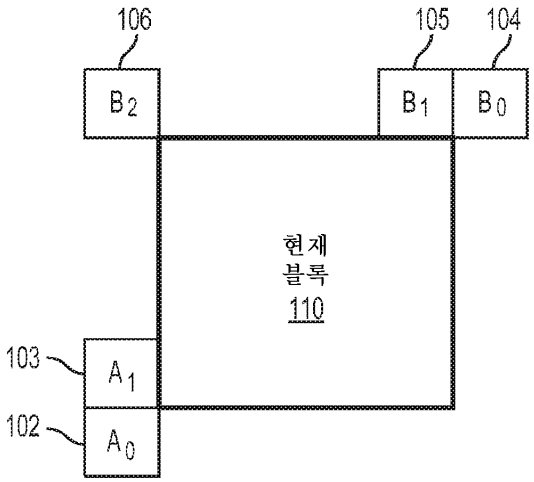
도면1b


18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
0 : Intra_Planar
1 : Intra_DC

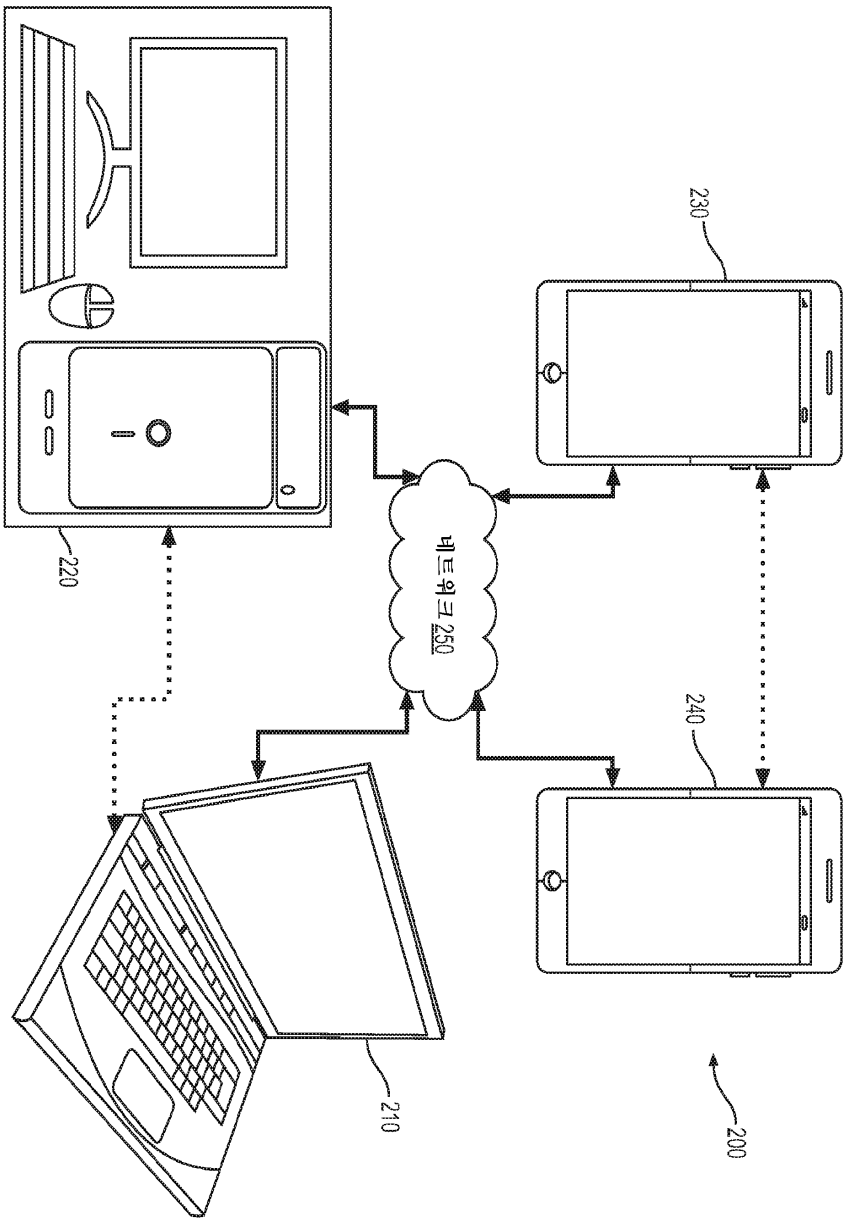
도면1c


34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
32
68
30
70
28
72
26
74
24
76
22
78
20
80
18
16
14
12
10
8
6
4
2
-2
-4
-6
-8
-10
-12
-14
0: 평면
1: DC

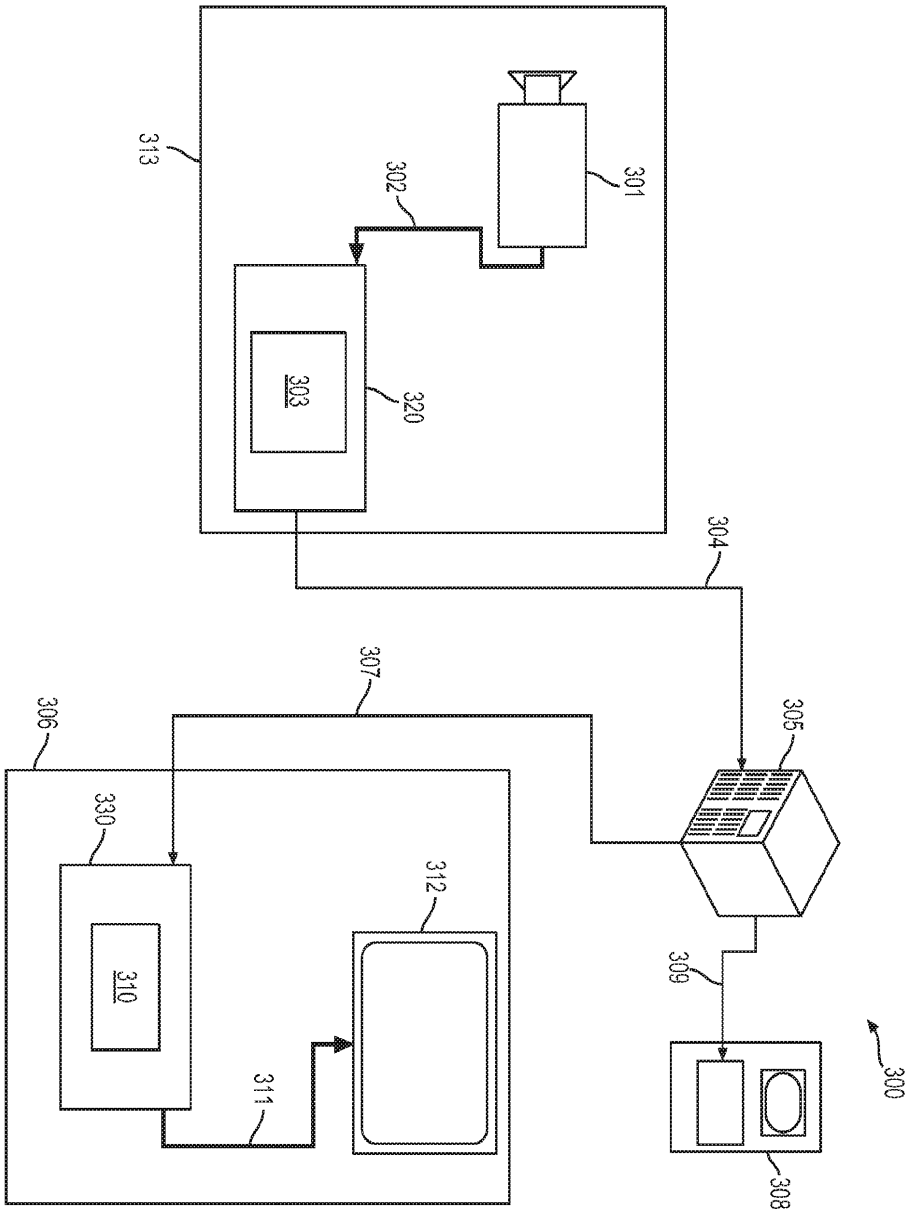
도면1d


106
105
104
B2
B1
B0
현재
블록
110
103
A1
102
A0

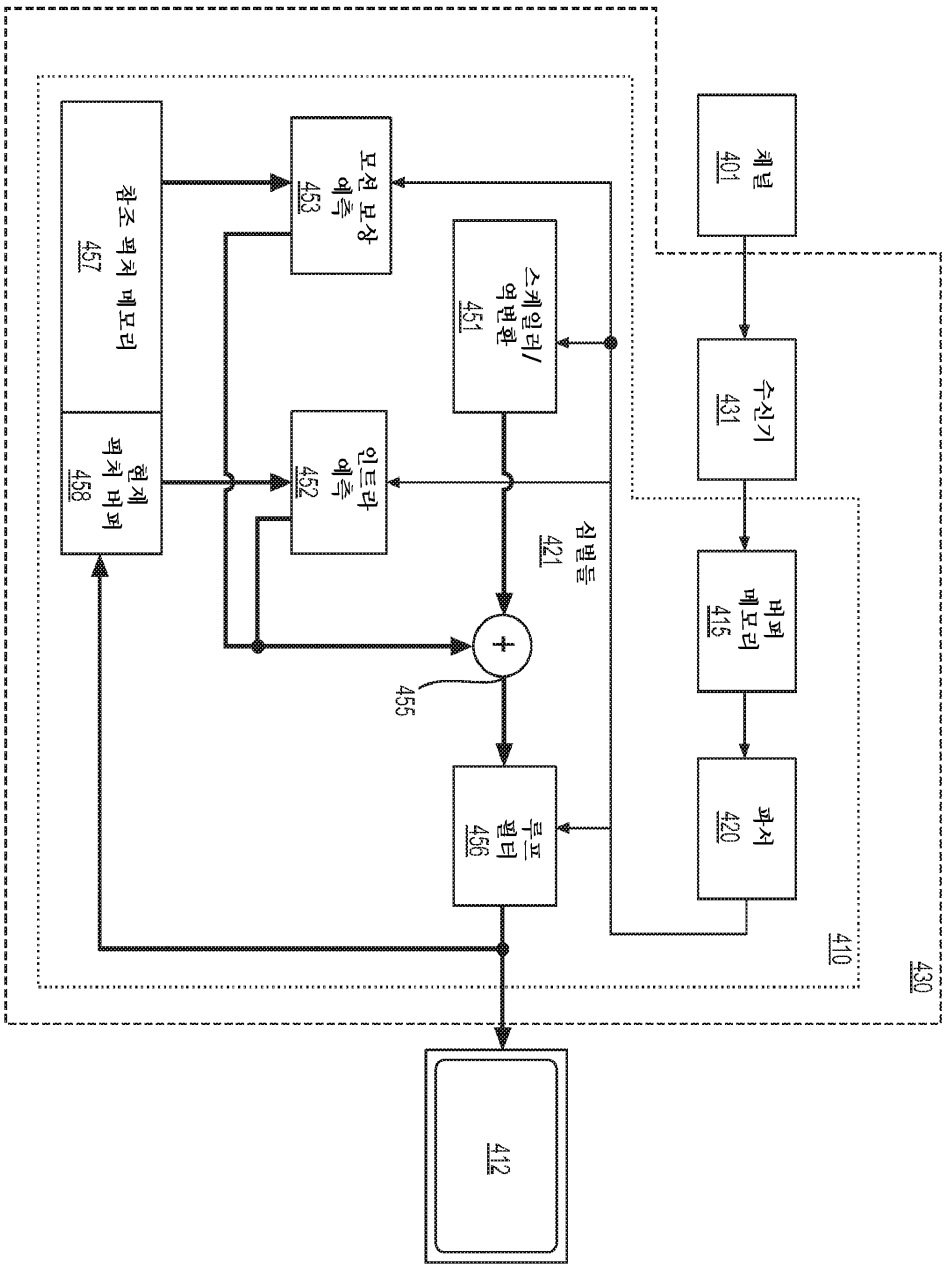
도면2



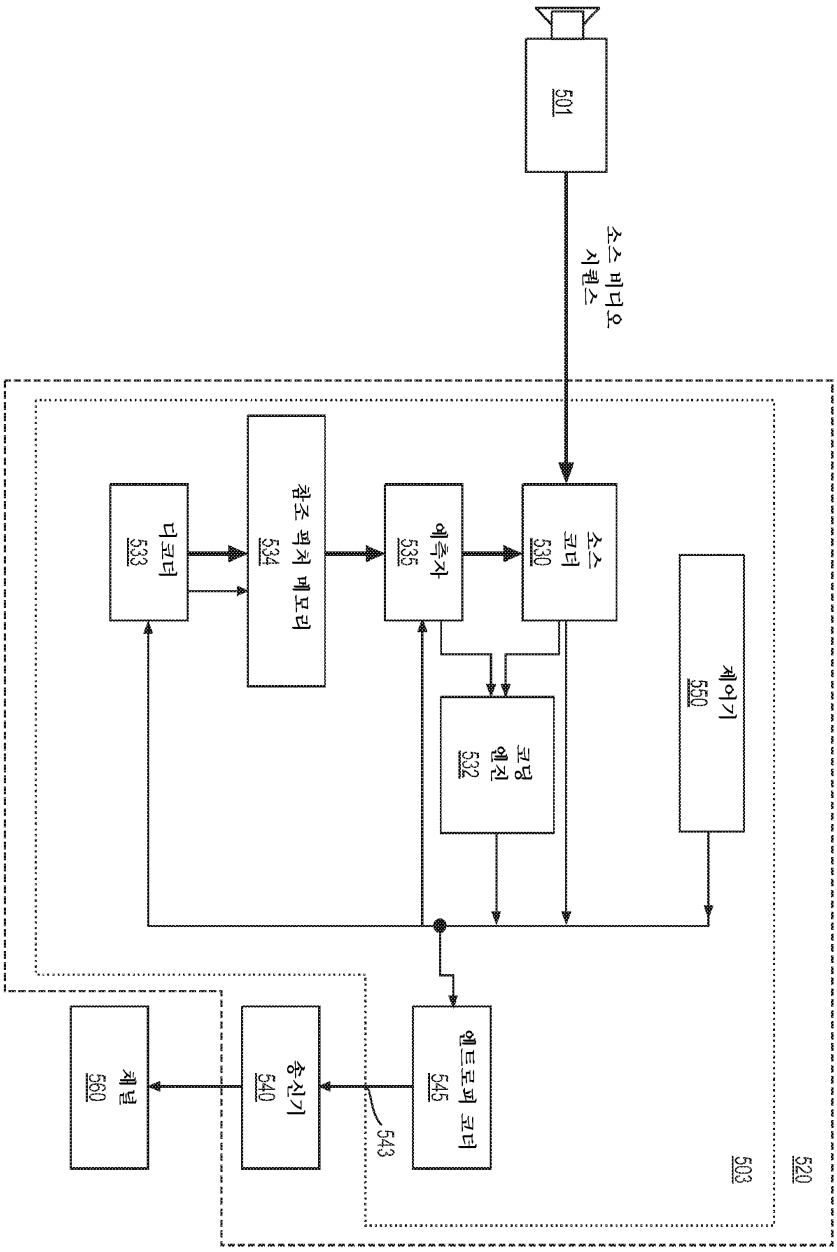
도면3



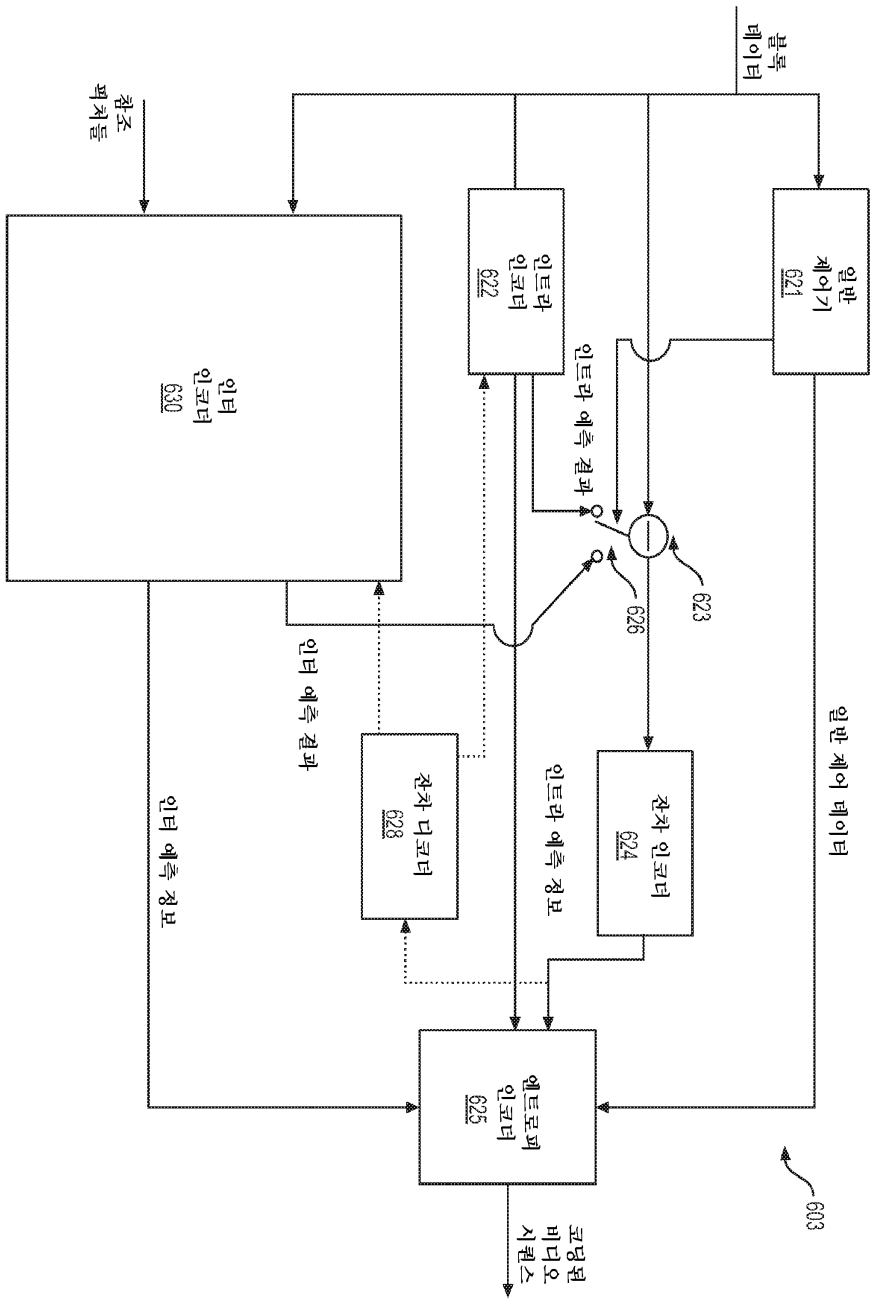
도면4


채널
401
수신기
431
버퍼
메모리
415
파서
420
410
430
스케일러/
역변환
451
모션 보상
예측
453
참조 픽처 메모리
457
현재
픽처 버퍼
458
인트라
예측
452
심벌들
421
455
루프
필터
456
412

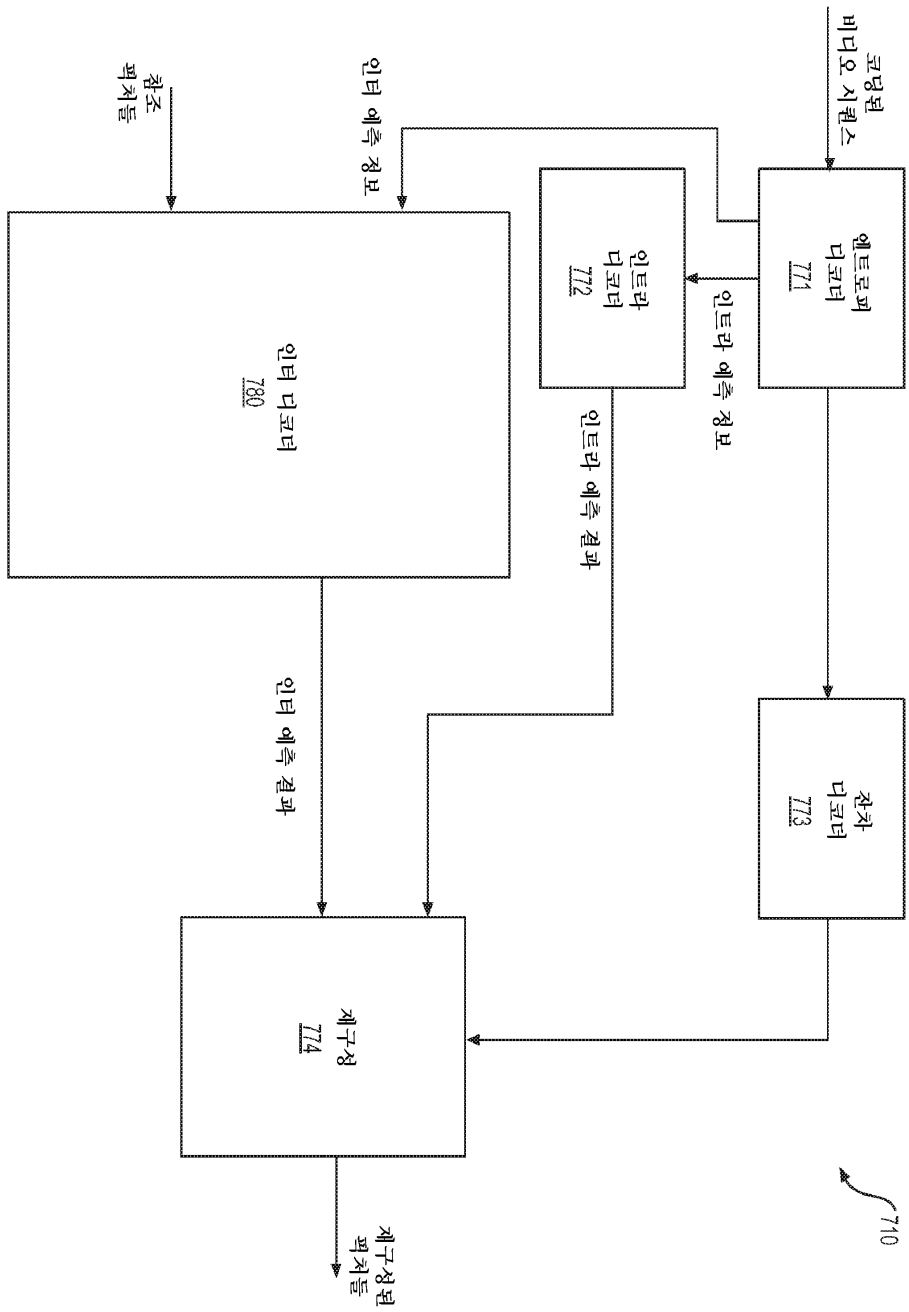
도면5


501
소스 비디오 시퀀스
520
503
디코더 533
참조 픽처 메모리 534
예측자 535
소스 코더 530
코딩 엔진 532
제어기 550
엔트로피 코더 545
543
송신기 540
채널 560

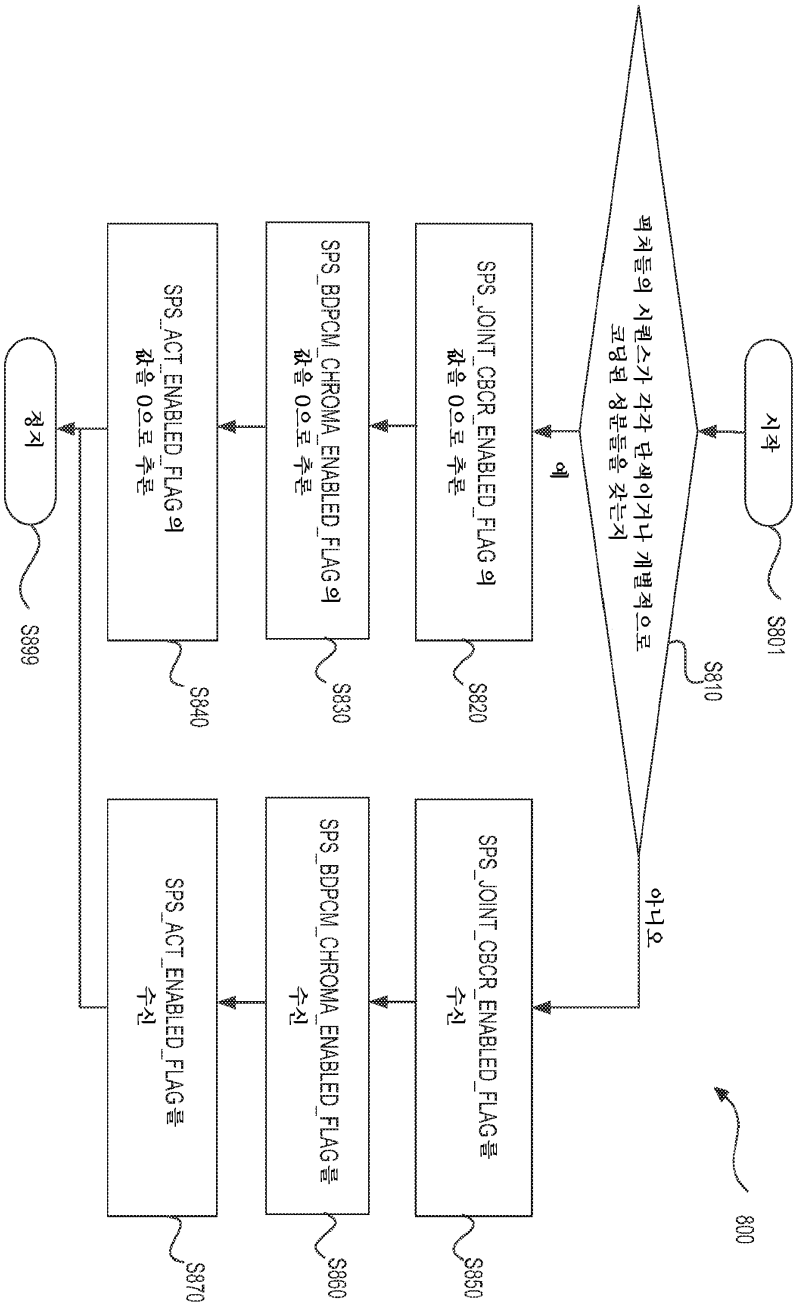
도면6


블록 데이터
참조 픽처들
일반 제어기 621
인트라 인코더 622
인터 인코더 630
인트라 예측 결과
623
626
인터 예측 결과
잔차 디코더 628
잔차 인코더 624
인트라 예측 정보
일반 제어 데이터
인터 예측 정보
엔트로피 인코더 625
603
코딩된 비디오 시퀀스

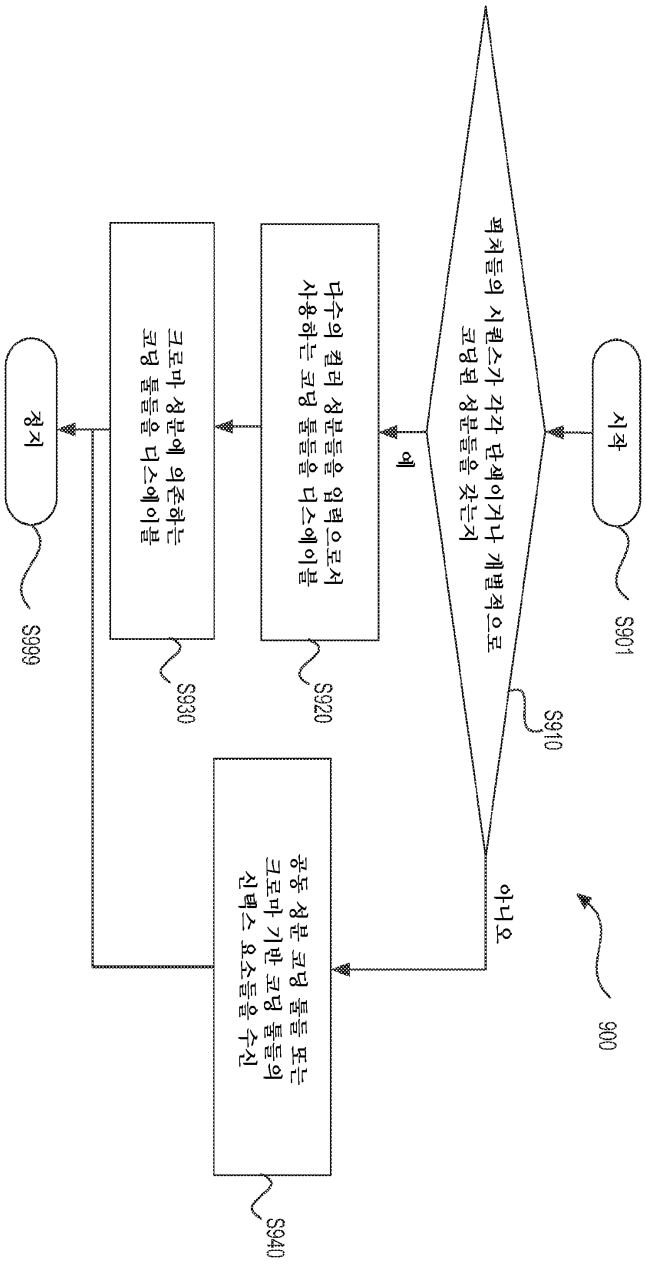
도면7


코딩된
비디오 시퀀스
엔트로피
디코더
771
인트라
디코더
772
인터 예측 정보
인트라 예측 정보
참조
픽처들
인터 디코더
780
인트라 예측 결과
인터 예측 결과
잔차
디코더
773
재구성
774
710
재구성된
픽처들

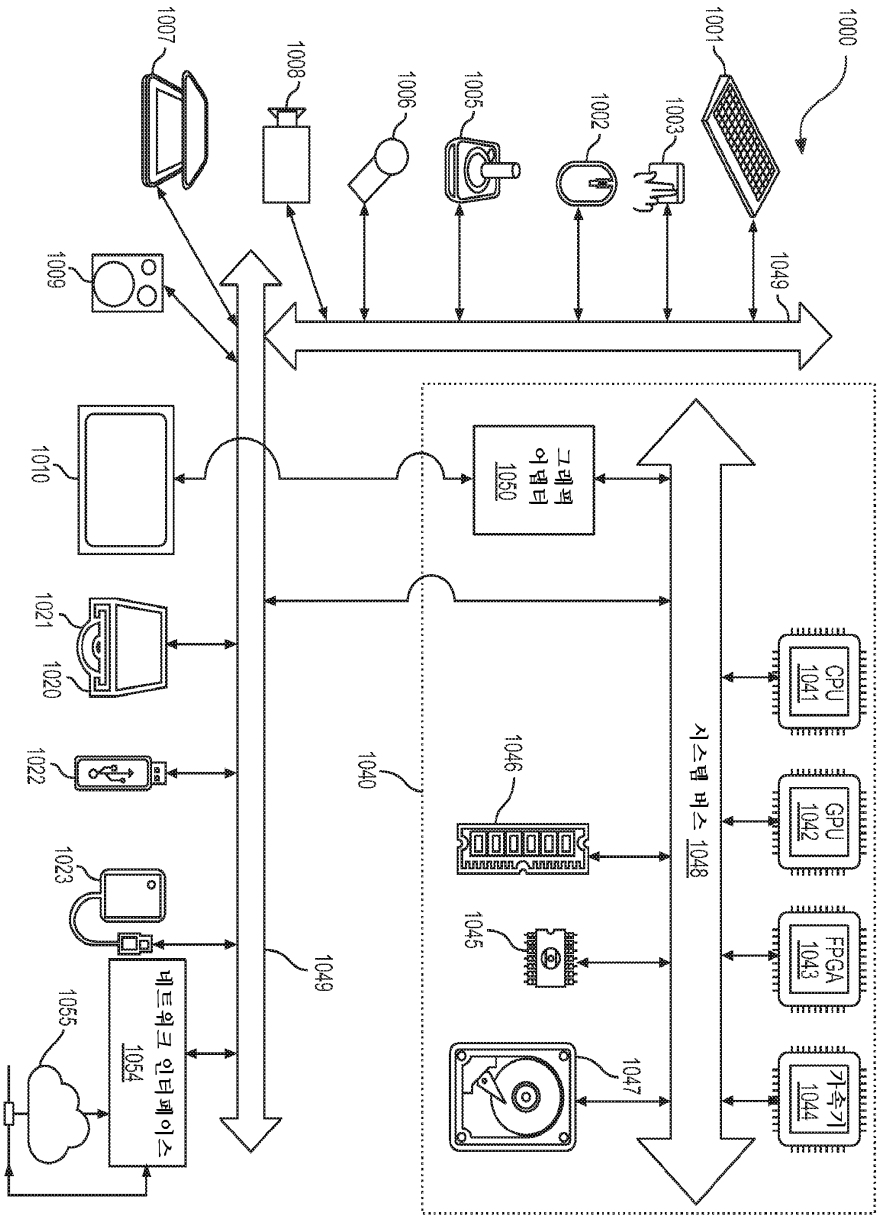
도면8


시작
S801
픽처들의 시퀀스가 각각 단색이거나 개별적으로 코딩된 성분들을 갖는지
S810
예
아니오
SPS_JOINT_CBCR_ENABLED_FLAG 의 값을 0으로 추론
S820
SPS_BDPCM_CHROMA_ENABLED_FLAG 의 값을 0으로 추론
S830
SPS_ACT_ENABLED_FLAG 의 값을 0으로 추론
S840
SPS_JOINT_CBCR_ENABLED_FLAG를 수신
S850
SPS_BDPCM_CHROMA_ENABLED_FLAG를 수신
S860
SPS_ACT_ENABLED_FLAG를 수신
S870
종지
S899
800

도면9


시작
S901
픽처들의 시퀀스가 각각 단색이거나 개별적으로 코딩된 성분들을 갖는지
S910
예
다수의 컬러 성분들을 입력으로서 사용하는 코딩 툴들을 디스에이블
S920
크로마 성분에 의존하는 코딩 툴들을 디스에이블
S930
아니오
공동 성분 코딩 툴들 또는 크로마 기반 코딩 툴들의 신택스 요소들을 수신
S940
종지
S999
900

도면10


1000
1001
1002
1003
1005
1006
1007
1008
1009
1010
1020
1021
1022
1023
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1054
1055
CPU 1041
GPU 1042
FPGA 1043
가속기 1044
시스템 버스 1048
그래픽 어댑터 1050
네트워크 인터페이스 1054