(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년04월22일
(11) 등록번호 10-2798404
(24) 등록일자 2025년04월16일

(51) 국제특허분류(Int. Cl.)
*H04N 19/563* (2014.01) *H04N 19/172* (2014.01)
*H04N 19/523* (2014.01) *H04N 19/583* (2014.01)
*H04N 19/70* (2014.01) *H04N 19/86* (2014.01)
(52) CPC특허분류
*H04N 19/563* (2015.01)
*H04N 19/172* (2015.01)
(21) 출원번호 10-2024-7012957(분할)
(22) 출원일자(국제) 2020년09월17일
심사청구일자 2024년04월18일
(85) 번역문제출일자 2024년04월18일
(65) 공개번호 10-2024-0056776
(43) 공개일자 2024년04월30일
(62) 원출원 특허 10-2021-7027148
원출원일자(국제) 2020년09월17일
심사청구일자 2021년08월25일
(86) 국제출원번호 PCT/US2020/051192
(87) 국제공개번호 WO 2021/055552
국제공개일자 2021년03월25일
(30) 우선권주장
62/903,635 2019년09월20일 미국(US)
17/019,692 2020년09월14일 미국(US)
(56) 선행기술조사문헌
KR1020160034998 A*
KR1020190080805 A*
Philippe Hanhart et al., CE13: PERP with horizontal geometry padding of reference pictures , JVET of ITU-T and ISO/IEC, JVET-L0231-v4(2018.10.11.)*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
**텐센트 아메리카 엘엘씨**
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
(72) 발명자
**최, 병두**
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
**웽거, 스테판**
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 10 항 심사관 : 김성권

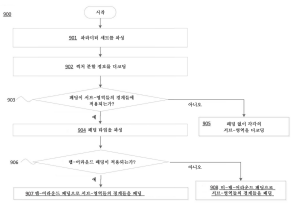
(54) 발명의 명칭 **비디오 스트림 내의 서브-영역 분할들을 이용한 패딩 처리를 위한 방법**

## (57) 요 약

비디오 코딩을 위한 방법, 컴퓨터 프로그램, 및 컴퓨터 시스템이 제공된다. 랩-어라운드 패딩 프로세스에 대응하는 코딩된 구문 요소들이 디코딩된다. 랩-어라운드 패딩 프로세스를 이용하여 적어도 하나의 코딩된 현재 픽처가 재구성된다. 상기 구문 요소들은 랩-어라운드 처리를 위한 오프셋 값; 또는 좌측 및 우측 패딩 폭 정보를

(뒷면에 계속)

## 대 표 도



지시한다. 플래그가 랩-어라운드 패딩 프로세스에 대응하는 구문 요소들이 파라미터 세트에 존재하는지를 지시한다. 랩-어라운드 패딩 프로세스에 대응하는 구문 요소들을 클리핑 프로세스로 해석함으로써 참조 픽처 내의 모션 보상된 예측을 위한 픽셀 위치가 결정된다.

(52) CPC특허분류
*H04N 19/523* (2015.01)
*H04N 19/583* (2015.01)
*H04N 19/70* (2015.01)
*H04N 19/86* (2015.01)

# 명 세 서

## 청구범위

### 청구항 1

프로세서에 의해 실행가능한 비디오 디코딩의 방법으로서,

서브픽처가 디코딩 프로세스에서 픽처로서 취급된다는 것을 지시하는 제1 플래그를 포함하는 랩-어라운드 패딩 프로세스에 대응하는 코딩된 구문 요소들을 디코딩하는 단계;

상기 코딩된 구문 요소들에 포함된 제2 플래그를 디코딩하는 단계 - 상기 제2 플래그는 상기 랩-어라운드 패딩 프로세스가 적용되는지 여부를 지시함 -; 및

상기 제2 플래그가 상기 랩-어라운드 패딩 프로세스가 적용되는 것을 지시하는 경우, 상기 서브픽처의 경계에서 상기 랩-어라운드 패딩 프로세스를 적용하는 것에 의해 상기 랩-어라운드 패딩 프로세스를 이용하여 적어도 하나의 코딩된 현재 픽처를 재구성하는 단계 - 상기 서브픽처의 경계는 상기 적어도 하나의 코딩된 현재 픽처로부터 분할된 직사각형 영역이고, 상기 서브픽처의 경계는 상기 적어도 하나의 코딩된 현재 픽처의 경계와는 다르고, 상기 랩-어라운드 패딩 프로세스의 오프셋 값은 루마 샘플에서의 픽처 폭에서, 루마 샘플에서 좌측 패딩 영역의 폭 및 루마 샘플에서 우측 패딩 영역의 폭을 감산하는 것에 의해 도출됨 -

를 포함하는, 방법.

### 청구항 2

제1항에 있어서,

상기 랩-어라운드 패딩 프로세스에 대응하는 구문 요소들을 클리핑 프로세스로 해석함으로써 참조 픽처 내의 모션 보상된 예측을 위한 픽셀 위치가 결정되는, 방법.

### 청구항 3

제2항에 있어서,

상기 결정된 픽셀 위치에 기초하여 상기 모션 보상된 예측을 위해 분수 픽셀이 보간되는, 방법.

### 청구항 4

제1항에 있어서,

상기 랩-어라운드 패딩 프로세스가 상기 적어도 하나의 코딩된 현재 픽처의 경계에서 적용되는, 방법.

### 청구항 5

제1항에 있어서,

상기 랩-어라운드 패딩 프로세스가 수평 경계에서, 수직 경계에서, 또는 상기 수직 경계와 상기 수평 경계 둘 다에서 적용되는, 방법.

### 청구항 6

제5항에 있어서,

하이 레벨 구문 구조에서의 플래그는 상기 랩-어라운드 패딩 프로세스가 상기 수평 경계에서, 상기 수직 경계에서, 또는 상기 수직 경계와 상기 수평 경계 둘 다에서 적용되는지를 지시하는, 방법.

### 청구항 7

프로세서에 의해 실행가능한 비디오 인코딩의 방법으로서,

서브픽처가 디코딩 프로세스에서 픽처로서 취급된다는 것을 지시하는 제1 플래그를 포함하는 랩-어라운드 패딩

프로세스에 대응하는 제1 구문 요소들을 인코딩하는 단계; 및

상기 랩-어라운드 패딩 프로세스가 적용되는지 여부를 지시하는 제2 플래그를 포함하는 제2 구문 요소들을 인코딩하는 단계

를 포함하고,

상기 제2 플래그는 상기 랩-어라운드 패딩 프로세스가 적용되는 것을 지시하고, 적어도 하나의 현재 픽처는 상기 서브픽처의 경계에서 상기 랩-어라운드 패딩 프로세스를 적용하는 것에 의해 상기 랩-어라운드 패딩 프로세스를 이용하여 재구성되도록 코딩되며, 상기 서브픽처의 경계는 적어도 하나의 현재 픽처로부터 분할된 직사각형 영역이고, 상기 서브픽처의 경계는 상기 적어도 하나의 현재 픽처의 경계와는 다르고, 상기 랩-어라운드 패딩 프로세스의 오프셋 값은 루마 샘플에서의 픽처 폭에서, 루마 샘플에서 좌측 패딩 영역의 폭 및 루마 샘플에서 우측 패딩 영역의 폭을 감산하는 것에 의해 도출되는, 방법.

**청구항 8**

비디오 코딩을 위한 컴퓨터 시스템으로서,

컴퓨터 프로그램 코드를 저장하도록 구성된 하나 이상의 컴퓨터 판독가능 비-일시적 저장 매체; 및

제1항 내지 제7항 중 어느 한 항의 방법을 수행하도록, 상기 컴퓨터 프로그램 코드에 액세스하고 상기 컴퓨터 프로그램 코드에 의해 지시된 대로 동작하도록 구성된 하나 이상의 컴퓨터 프로세서

를 포함하는 컴퓨터 시스템.

**청구항 9**

비디오 코딩을 위한 컴퓨터 프로그램이 저장되어 있는 비-일시적 컴퓨터 판독가능 매체로서, 상기 컴퓨터 프로그램은 하나 이상의 컴퓨터 프로세서로 하여금 제1항 내지 제7항 중 어느 한 항의 방법을 수행하게 하도록 구성되는, 비-일시적 컴퓨터 판독가능 매체.

**청구항 10**

제7항의 방법에 따라 생성된 비트스트림을 저장하는 컴퓨터 판독가능 저장 매체.

**청구항 11**

삭제

**청구항 12**

삭제

**청구항 13**

삭제

**청구항 14**

삭제

**청구항 15**

삭제

## *발명의 설명*

### *기 술 분 야*

[0001] 관련 출원의 상호 참조

[0002] 본 출원은 2019년 9월 20일자로 출원된 미국 가특허 출원 제62/903,635호, 및 2020년 9월 14일자로 출원된 미국 특허 출원 제17/019,692호로부터 우선권을 주장하고, 이들 전체는 본 명세서에 포함된다.

[0003] 본 개시내용은 일반적으로 데이터 처리 분야에 관한 것이고, 더 특정하게는 비디오 인코딩 및 디코딩에 관한 것이다.

## 배 경 기 술

[0004] 모션 보상을 갖는 인터-픽처 예측(inter-picture prediction)을 이용한 비디오 코딩 및 디코딩이 수십년 동안 알려져 있다. 압축되지 않은 디지털 비디오는 일련의 픽처들로 구성될 수 있고, 각각의 픽처는, 예를 들어, 1920 x 1080 루미넌스 샘플들 및 연관된 크로미넌스 샘플들의 공간 차원을 갖는다. 이 일련의 픽처들은, 예를 들어, 초당 60개 픽처 또는 60 Hz의, 고정된 또는 가변 픽처 레이트(비공식적으로 프레임 레이트로도 알려져 있음)를 가질 수 있다. 압축되지 않은 비디오는 상당한 비트레이트 요건들을 갖는다. 예를 들어, 샘플 당 8 비트에서의 1080p60 4:2:0 비디오(60 Hz 프레임 레이트에서의 1920x1080 루미넌스 샘플 해상도)는 1.5 Gbit/s 대역폭에 가까울 것을 요구한다. 1 시간 분량의 그러한 비디오는 600 GByte보다 많은 저장 공간을 필요로 한다.

[0005] 비디오 코딩 및 디코딩의 하나의 목적은, 압축을 통한, 입력 비디오 신호에서의 중복성의 감소일 수 있다. 압축은 전술한 대역폭 또는 저장 공간 요건들을, 일부 경우들에서, 2 자릿수 이상 감소시키는 데 도움이 될 수 있다. 무손실 및 손실 압축 둘 다뿐만 아니라 이들의 조합이 이용될 수 있다. 무손실 압축은 압축된 원래 신호로부터 원래 신호의 정확한 사본이 재구성될 수 있는 기법들을 지칭한다. 손실 압축을 이용할 때, 재구성된 신호는 원래 신호와 동일하지 않을 수 있지만, 원래 신호와 재구성된 신호 사이의 왜곡은 재구성된 신호를 의도된 응용에 유용하게 만들 정도로 충분히 작다. 비디오의 경우, 손실 압축이 널리 이용된다. 용인되는 왜곡의 양은 응용에 의존한다; 예를 들어, 특정 소비자 스트리밍 응용들의 사용자들은 텔레비전 기여 응용들의 사용자들보다 더 높은 왜곡을 용인할 수 있다. 달성가능한 압축비는 더 높은 허용가능한/용인가능한 왜곡이 더 높은 압축비를 산출할 수 있다는 것을 반영할 수 있다.

[0006] 비디오 인코더 및 디코더는, 예를 들어, 모션 보상, 변환, 양자화, 및 엔트로피 코딩을 포함한, 여러 광범위한 카테고리로부터의 기법들을 이용할 수 있고, 이들 중 일부가 아래에 소개될 것이다.

[0007] 코딩된 비디오 비트스트림을 패킷 네트워크들을 통한 전송을 위한 패킷들로 분할하는 개념이 수십년 동안 사용되어 왔다. 초기에, 비디오 코딩 표준들 및 기술들은 그것들의 대부분이 보트 지향 전송(bot-oriented transport), 및 정의된 비트스트림들에 최적화되었다. 예를 들어 실시간 전송 프로토콜(Real-time Transport Protocol, RTP) 페이로드 포맷들로 특정된 시스템 계층 인터페이스들에서 패킷화가 발생하였다. 인터넷을 통한 비디오의 대용량 사용에 적합한 인터넷 연결의 출현으로, 비디오 코딩 표준들은 VCL(video coding layer) 및 NAL(network abstraction layer)의 개념적 구별을 통해 그 현저한 사용 사례를 반영하였다. NAL 유닛들이 2003년 H.264에 도입되었고, 그 후 약간의 수정만을 가하여 특정 비디오 코딩 표준들 및 기술들에서 유지되었다.

[0008] NAL 유닛은, 많은 경우에, 코딩된 비디오 시퀀스의 모든 선행 NAL 유닛들을 반드시 디코딩하지 않고도 디코더가 작용할 수 있는 가장 작은 엔티티로 간주될 수 있다. NAL 유닛들이 특정 오류 회복력 기술들뿐만 아니라 특정 비트스트림 조작 기법들이 선택적 포워딩 유닛(SFU)들 또는 멀티포인트 제어 유닛(MCU)들과 같은 미디어 인식 네트워크 요소(MANE)들에 의한 비트스트림 프루닝을 포함하는 것을 가능하게 하는 한.

[0009] 도 1은 H.264(101) 및 H.265(102)에 따른 NAL 유닛 헤더들의 구문 다이어그램의 관련 부분들을 묘사하고, 양쪽 모두의 경우에 그것들 각각의 확장들 중 어떠한 것도 없다. 양쪽 모두의 경우에, forbidden_zero_bit는 특정 시스템 계층 환경들에서 시작 코드 에뮬레이션 방지를 위해 사용되는 제로 비트이다. nal_unit_type 구문 요소는, 예를 들어, 특정 슬라이스 타입들, 파라미터 세트 타입들, SEI(Supplementary Enhancement Information) 메시지 등 중 하나일 수 있는, NAL 유닛이 반송하는 데이터의 타입을 지칭한다. H.265 NAL 유닛 헤더는, NAL 유닛이 속하는 코딩된 픽처의 공간/SNR 및 시간 계층을 지시하는, nuh_layer_id 및 nuh_temporal_id_plus1을 추가로 포함한다.

[0010] NAL 유닛 헤더는, 예를 들어, 다른 NAL 유닛 헤더, 파라미터 세트 등과 같은 비트스트림 내의 다른 데이터에 대한 어떠한 파싱 종속성도 갖지 않는 쉽게 파싱 가능한 고정 길이 코드워드들만을 포함한다는 것을 알 수 있다. NAL 유닛 헤더들은 NAL 유닛 내의 첫 번째 옥텟이기 때문에, MANE들은 쉽게 그것들을 추출하고, 그것들을 파싱하고, 그것들에 작용할 수 있다. 다른 하이 레벨 구문 요소들, 예를 들어 슬라이스 또는 타일 헤더들은, 대조적으로, MANE들에 의해 덜 쉽게 액세스 가능한데, 그 이유는 그것들이 파라미터 세트 맥락 및/또는 가변 길이

또는 산술 코딩된 코드포인트들의 처리를 유지하는 것을 필요로 할 수 있기 때문이다.

[0011] 도 1에 도시된 바와 같은 NAL 유닛 헤더들은 NAL 유닛을 복수의 NAL 유닛(예컨대, 예를 들어, 다수의 타일 또는 슬라이스를 포함하는 것 - 이들 중 적어도 일부는 개별 NAL 유닛들에 패킷화됨)으로 구성된 코딩된 픽처에 연관시킬 수 있는 정보를 포함하지 않는다는 것을 또한 알 수 있다.

[0012] RTP(RFC 3550), MPEG-시스템 표준, ISO 파일 포맷 등과 같은 특정 전송 기술들은 종종 MANE들에 의해 쉽게 액세스 가능할 수 있고 그것들 각각의 전송 유닛들을 코딩된 픽처들과 연관시키는 데 도움을 줄 수 있는 프레젠테이션 시간(MPEG 및 ISO 파일 포맷의 경우에) 또는 캡처 시간(RTP의 경우에)과 같은 타이밍 정보의 형식으로 되어 있는 특정 정보를 포함할 수 있다. 그러나, 이들 정보의 의미론은 전송/저장 기술마다 다를 수 있고, 비디오 코딩에서 사용되는 픽처 구조와 직접적인 관계를 갖지 않을 수 있다. 따라서, 이들 정보는 기껏해야 휴리스틱(heuristics)일 수 있고 또한 NAL 유닛 스트림 내의 NAL 유닛들이 동일한 코딩된 픽처에 속하는지 여부를 식별하는 데 특히 적합하지 않을 수 있다.

## 발명의 내용

[0013] 실시예들은 비디오 코딩을 위한 방법, 시스템, 및 컴퓨터 판독가능 매체에 관한 것이다. 일 양태에 따르면, 비디오 코딩을 위한 방법이 제공된다. 이 방법은 랩-어라운드 패딩 프로세스에 대응하는 코딩된 구문 요소들을 디코딩하는 단계를 포함할 수 있다. 랩-어라운드 패딩 프로세스를 이용하여 적어도 하나의 코딩된 현재 픽처가 재구성된다. 상기 구문 요소들은 랩-어라운드 처리를 위한 오프셋 값; 또는 좌측 및 우측 패딩 폭 정보를 지시한다. 랩-어라운드 패딩 프로세스에 대응하는 구문 요소들을 클리핑 프로세스로 해석함으로써 참조 픽처 내의 모션 보상된 예측을 위한 픽셀 위치가 결정될 수 있다.

[0014] 다른 양태에 따르면, 비디오 코딩을 위한 컴퓨터 시스템이 제공된다. 이 컴퓨터 시스템은 하나 이상의 프로세서, 하나 이상의 컴퓨터 판독가능 메모리, 하나 이상의 컴퓨터 판독가능 유형(tangible) 저장 디바이스, 및 하나 이상의 메모리 중 적어도 하나를 통해 상기 하나 이상의 프로세서 중 적어도 하나에 의해 실행하기 위해 상기 하나 이상의 저장 디바이스 중 적어도 하나에 저장된 프로그램 명령어들을 포함할 수 있고, 그에 의해 상기 컴퓨터 시스템은 방법을 수행할 수 있다. 이 방법은 랩-어라운드 패딩 프로세스에 대응하는 코딩된 구문 요소들을 디코딩하는 단계를 포함할 수 있다. 랩-어라운드 패딩 프로세스를 이용하여 적어도 하나의 코딩된 현재 픽처가 재구성된다. 상기 구문 요소들은 랩-어라운드 처리를 위한 오프셋 값; 또는 좌측 및 우측 패딩 폭 정보를 지시한다.

[0015] 또 다른 양태에 따르면, 비디오 코딩을 위한 컴퓨터 판독가능 매체가 제공된다. 이 컴퓨터 판독가능 매체는 하나 이상의 컴퓨터 판독가능 저장 디바이스 및 상기 하나 이상의 유형 저장 디바이스 중 적어도 하나에 저장된 프로그램 명령어들을 포함할 수 있고, 이 프로그램 명령어들은 프로세서에 의해 실행가능하다. 상기 프로그램 명령어들은 랩-어라운드 패딩 프로세스에 대응하는 코딩된 구문 요소들을 디코딩하는 단계를 따라서 포함할 수 있는 방법을 수행하기 위해 프로세서에 의해 실행가능하다. 랩-어라운드 패딩 프로세스를 이용하여 적어도 하나의 코딩된 현재 픽처가 재구성된다. 상기 구문 요소들은 랩-어라운드 처리를 위한 오프셋 값; 또는 좌측 및 우측 패딩 폭 정보를 지시한다.

## 도면의 간단한 설명

[0016] 이들 및 다른 목적들, 특징들 및 이점들은, 첨부 도면들과 관련하여 읽혀질, 예시적인 실시예들에 대한 다음의 상세한 설명으로부터 명백해질 것이다. 도면들의 다양한 특징들은 일정한 비율로 되어 있지 않은데 그 이유는 예시들은 상세한 설명과 관련하여 본 기술분야의 통상의 기술자의 이해를 용이하게 하는 데 있어서 명확성을 위한 것이기 때문이다. 도면들에서:

도 1은 H.264 및 H.265에 따른 NAL 유닛 헤더들의 개략 예시이다.

도 2는 일 실시예에 따른 통신 시스템의 단순화된 블록도의 개략 예시이다.

도 3은 일 실시예에 따른 통신 시스템의 단순화된 블록도의 개략 예시이다.

도 4는 일 실시예에 따른 디코더의 단순화된 블록도의 개략 예시이다.

도 5는 일 실시예에 따른 인코더의 단순화된 블록도의 개략 예시이다.

도 6은 일 실시예에 따른 오프셋 시그널링을 위한 구문 요소들의 개략 예시이다.

도 7은 일 실시예에 따른 인코더의 패딩 폭 시그널링을 위한 구문 요소들의 개략 예시이다.

도 8은 일 실시예에 따른 각각의 서브-영역(타일 또는 서브-픽처)의 패딩 처리의 단순화된 블록도의 개략 예시이다.

도 9는 일 실시예에 따른 각각의 서브-픽처의 패딩 오프셋 시그널링을 위한 구문 요소들의 개략 예시이다.

도 10은 일 실시예에 따른 각각의 서브-픽처의 통합된 패딩 오프셋 시그널링을 위한 구문 요소들의 개략 예시이다.

도 11은 일 실시예에 따른 컴퓨터 시스템의 개략 예시이다.

## *발명을 실시하기 위한 구체적인 내용*

[0017] 청구된 구조들 및 방법들의 상세한 실시예들이 본 명세서에 개시된다; 그러나, 개시된 실시예들은 다양한 형식들로 구현될 수 있는 청구된 구조들 및 방법들을 예시하는 것에 불과하다는 것을 이해할 수 있다. 그러나, 해당 구조들 및 방법들은 많은 상이한 형식으로 구현될 수 있고 본 명세서에서 제시되는 예시적인 실시예들로 제한되는 것으로 해석되어서는 안 된다. 오히려, 이들 예시적인 실시예는 본 개시내용이 철저하고 완전하도록 그리고 본 기술분야의 통상의 기술자들에게 그 범위를 충분히 전달하도록 제공된다. 설명에서, 잘 알려진 특징들 및 기법들의 세부 사항들은 제시된 실시예들을 불필요하게 모호하게 하는 것을 피하기 위해 생략될 수 있다.

[0018] 실시예들은 일반적으로 데이터 처리 분야에 관한 것이고, 더 특정하게는 비디오 인코딩 및 디코딩에 관한 것이다. 다음에 설명된 예시적인 실시예들은, 여러 다른 것들 중에서도 특히, 코딩된 구문 요소들을 디코딩하는 것에 기초하여 랩-어라운드 패딩 프로세스를 이용하여 코딩된 현재 픽처를 재구성하는 시스템, 방법 및 컴퓨터 프로그램을 제공한다. 따라서, 일부 실시예들은 압축된 이미지의 재구성에 이용될 구문 요소들의 코딩을 허용함으로써 컴퓨팅 분야를 개선하는 능력을 가진다.

[0019] 이전에 기술된 바와 같이, 비디오 코딩 및 디코딩의 하나의 목적은, 압축을 통한, 입력 비디오 신호에서의 중복성의 감소일 수 있다. 압축은 전술한 대역폭 또는 저장 공간 요건들을, 일부 경우들에서, 2 자릿수 이상 감소시키는 데 도움이 될 수 있다. 무손실 및 손실 압축 둘 다뿐만 아니라 이들의 조합이 이용될 수 있다. 무손실 압축은 압축된 원래 신호로부터 원래 신호의 정확한 사본이 재구성될 수 있는 기법들을 지칭한다. 손실 압축을 이용할 때, 재구성된 신호는 원래 신호와 동일하지 않을 수 있지만, 원래 신호와 재구성된 신호 사이의 왜곡은 재구성된 신호를 의도된 응용에 유용하게 만들 정도로 충분히 작다. 비디오의 경우, 손실 압축이 널리 이용된다. 용인되는 왜곡의 양은 응용에 의존한다; 예를 들어, 특정 소비자 스트리밍 응용들의 사용자들은 텔레비전 기여 응용들의 사용자들보다 더 높은 왜곡을 용인할 수 있다. 달성가능한 압축비는 더 높은 허용가능한/용인가능한 왜곡이 더 높은 압축비를 산출할 수 있다는 것을 반영할 수 있다. 그러나, 픽처가 하나 이상의 서브-영역(타일, 슬라이스 또는 서브-픽처)으로 분할될 때, 각각의 서브-영역의 경계 처리는 코딩 효율 및 주관적인 시각적 품질에 영향을 미칠 수 있다. 각각의 서브-영역 경계에서의 경계 처리의 적응적 제어가 360 미디어 처리에서의 중요 요인이다. 따라서, 코딩된 구문 요소들을 디코딩하는 것에 기초하여 랩-어라운드 패딩 프로세스를 이용하여 코딩된 현재 픽처를 재구성하는 것이 유리할 수 있다.

[0020] 양태들은 다양한 실시예들에 따른 방법들, 장치들(시스템들), 및 컴퓨터 판독가능 매체들의 흐름도 예시들 및/또는 블록도들을 참조하여 본 명세서에서 설명된다. 흐름도 예시들 및/또는 블록도들의 각각의 블록, 및 흐름도 예시들 및/또는 블록도들 내의 블록들의 조합들은 컴퓨터 판독가능 프로그램 명령어들에 의해 구현될 수 있다는 것을 이해할 것이다.

[0021] 다음에 설명된 예시적인 실시예들은 코딩된 구문 요소들을 디코딩하는 것에 기초하여 랩-어라운드 패딩 프로세스를 이용하여 코딩된 현재 픽처를 재구성하는 시스템, 방법 및 컴퓨터 프로그램을 제공한다.

[0022] 도 2는 본 개시내용의 일 실시예에 따른 통신 시스템(200)의 단순화된 블록도를 예시한다. 시스템(200)은 네트워크(250)를 통해 상호연결되는 적어도 2개의 단말(210-220)을 포함할 수 있다. 데이터의 단방향 송신을 위해, 제1 단말(210)은 네트워크(250)를 통해 다른 단말(220)로 송신하기 위해 로컬 위치에서 비디오 데이터를 코딩할 수 있다. 제2 단말(220)은 네트워크(250)로부터 다른 단말의 코딩된 비디오 데이터를 수신하고, 코딩된 데이터를 디코딩하고 복구된 비디오 데이터를 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 응용들 등에서 일반적일 수 있다.

[0023] 도 2는, 예를 들어, 영상 회의 동안 발생할 수 있는 코딩된 비디오의 양방향 송신을 지원하기 위해 제공되는 제

2 쌍의 단말들(230, 240)을 예시한다. 데이터의 양방향 송신을 위해, 각각의 단말(230, 240)은 네트워크(250)를 통해 다른 단말로 송신하기 위해 로컬 위치에서 캡처된 비디오 데이터를 코딩할 수 있다. 각각의 단말(230, 240)은 또한 다른 단말에 의해 송신된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 데이터를 디코딩할 수 있고, 복구된 비디오 데이터를 로컬 디스플레이 디바이스에서 디스플레이할 수 있다.

[0024] 도 2에서, 단말들(210-240)은 서버들, 개인용 컴퓨터들 및 스마트 폰들로서 예시될 수 있지만, 본 개시내용의 원리들은 그렇게 제한되지 않는다. 본 개시내용의 실시예들은 랩톱 컴퓨터들, 태블릿 컴퓨터들, 미디어 플레이어들 및/또는 전용 영상 회의 장비를 이용한 응용을 찾는다. 네트워크(250)는 예를 들어 유선 및/또는 무선 통신 네트워크들을 포함하여, 단말들(210-240) 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.

[0025] 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 인에이블 응용들에 동등하게 적용가능할 수 있다.

[0026] 스트리밍 시스템은, 예를 들어 압축되지 않은 비디오 샘플 스트림(302)을 생성하는 비디오 소스(301), 예를 들어 디지털 카메라를 포함할 수 있는 캡처 서브시스템(313)을 포함할 수 있다. 인코딩된 비디오 비트스트림들과 비교할 때 많은 데이터 용량을 강조하기 위해 굵은 라인으로 묘사된, 샘플 스트림(302)은 카메라(301)에 결합된 인코더(303)에 의해 처리될 수 있다. 인코더(303)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 양태들을 가능하게 하거나 구현하기 위해 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 샘플 스트림과 비교할 때 적은 데이터 용량을 강조하기 위한 얇은 라인으로 묘사된, 인코딩된 비디오 비트스트림(304)은 미래의 사용을 위해 스트리밍 서버(305) 상에 저장될 수 있다. 하나 이상의 스트리밍 클라이언트(306, 308)는 스트리밍 서버(305)에 액세스하여 인코딩된 비디오 비트스트림(304)의 사본들(307, 309)을 검색할 수 있다. 클라이언트(306)는 인코딩된 비디오 비트스트림의 착신 사본(307)을 디코딩하고 디스플레이(312) 또는 다른 렌더링 디바이스(묘사되지 않음) 상에 렌더링될 수 있는 발신 비디오 샘플 스트림(311)을 생성하는 비디오 디코더(310)를 포함할 수 있다. 일부 스트리밍 시스템들에서, 비디오 비트스트림들(304, 307, 309)은 특정 비디오 코딩/압축 표준에 따라 인코딩될 수 있다. 해당 표준들의 예들은 ITU-T 권고안(Recommendation) H.265를 포함한다. Versatile Video Coding 또는 VVC로서 비공식적으로 알려진 비디오 코딩 표준이 개발 중이다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.

[0027] 도 4는 본 개시내용의 일 실시예에 따른 비디오 디코더(310)의 기능 블록도일 수 있다.

[0028] 수신기(410)가 디코더(310)에 의해 디코딩될 하나 이상의 코덱 비디오 시퀀스를 수신할 수 있다; 동일한 또는 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스를 수신할 수 있고, 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(412)로부터 수신될 수 있다. 수신기(410)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(410)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터를 방지하기 위해, 수신기(410)와 엔트로피 디코더/파서(420)(이후 "파서") 사이에 버퍼 메모리(415)가 결합될 수 있다. 수신기(410)가 충분한 대역폭 및 제어가능성의 저장/포워드 디바이스로부터, 또는 동기식 네트워크(isosynchronous network)로부터 데이터를 수신하고 있을 때, 버퍼(415)는 필요하지 않을 수 있거나, 작을 수 있다. 인터넷과 같은 최선 노력 패킷 네트워크들 상에서의 사용을 위해, 버퍼(415)가 요구될 수 있고, 비교적 클 수 있고, 유리하게는 적응적 크기일 수 있다.

[0029] 비디오 디코더(310)는 엔트로피 코딩된 비디오 시퀀스로부터 심벌들(421)을 재구성하기 위해 파서(420)를 포함할 수 있다. 해당 심벌들의 카테고리들은 디코더(310)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 3에 도시된 바와 같이, 디코더의 일체 부분(integral part)은 아니지만 디코더에 결합될 수 있는 디스플레이(312)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함한다. 렌더링 디바이스(들)에 대한 제어 정보는 SEI(Supplementary Enhancement Information) 메시지 또는 VUI(Video Usability Information) 파라미터 세트

프래그먼트들(묘사되지 않음)의 형식일 수 있다. 파서(420)는 수신된 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, 맥락 민감성(context sensitivity)을 갖거나 갖지 않는 산술 코딩 등을 포함하는, 본 기술분야의 통상의 기술자에게 잘 알려진 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 그룹들(Groups of Pictures, GOPs), 픽처들, 타일들, 슬라이스들, 매크로블록들, 코딩 단위들(Coding Units, CUs), 블록들, 변환 단위들(Transform Units, TUs), 예측 단위들(Prediction Units, PUs) 등을 포함할 수 있다. 엔트로피 디코더/파서는 또한 코딩된 비디오 시퀀스로부터 변환 계수들, 양자화기 파라미터 값들, 모션 벡터들 등과 같은 정보를 추출할 수 있다.

[0030] 파서(420)는 버퍼(415)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여, 심벌들(421)을 생성할 수 있다.

[0031] 심벌들(421)의 재구성은 코딩된 비디오 픽처 또는 그것의 부분들의 타입(예컨대: 인터 및 인트라 픽처, 인터 및 인트라 블록), 및 다른 요인들에 따라 다수의 상이한 유닛들을 수반할 수 있다. 어느 유닛들이 수반되는지, 그리고 어떻게 되는지는 파서(420)에 의해 코딩된 비디오 시퀀스로부터 파싱된 서브그룹 제어 정보에 의해 제어될 수 있다. 파서(420)와 아래의 다수의 유닛 사이의 그러한 서브그룹 제어 정보의 흐름은 명확성을 위해 묘사되어 있지 않다.

[0032] 이미 언급된 기능 블록들 이외에, 디코더(310)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 위해, 아래의 기능 유닛들로의 개념적 세분이 적절하다.

[0033] 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 팩터, 양자화 스케일링 행렬들 등을 포함하여, 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 그것은 집계기(455)에 입력될 수 있는, 샘플 값들을 포함하는 블록들을 출력할 수 있다.

[0034] 일부 경우들에서, 스케일러/역변환(451)의 출력 샘플들은 인트라-코딩된 블록에 관련될 수 있다; 즉, 이전에 재구성된 픽처들로부터의 예측 정보를 이용하는 것이 아니고, 현재 픽처의 이전에 재구성된 부분들로부터의 예측 정보를 이용할 수 있는 블록. 그러한 예측 정보는 인트라 픽처 예측 유닛(452)에 의해 제공될 수 있다. 일부 경우들에서, 인트라 픽처 예측 유닛(452)은 현재(부분적으로 재구성된) 픽처(456)로부터 페치된 주위의 이미 재구성된 정보를 이용하여, 재구성 중인 블록의 동일한 크기 및 형상의 블록을 생성한다. 집계기(455)는, 일부 경우들에서, 샘플 당 기준으로, 인트라 예측 단위(452)가 생성한 예측 정보를 스케일러/역변환 유닛(451)에 의해 제공된 출력 샘플 정보에 추가한다.

[0035] 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)이 참조 픽처 메모리(457)에 액세스하여 예측을 위해 사용되는 샘플들을 페치할 수 있다. 블록에 관련된 심벌들(421)에 따라 페치된 샘플들을 모션 보상한 후에, 이들 샘플은 집계기(455)에 의해 스케일러/역변환 유닛의 출력(이 경우 잔차 샘플들 또는 잔차 신호라고 불림)에 추가되어 출력 샘플 정보를 생성할 수 있다. 모션 보상 예측 유닛이 예측 샘플들을 페치하는 참조 픽처 메모리 내의 어드레스들은, 예를 들어 X, Y, 및 참조 픽처 컴포넌트들을 가질 수 있는 심벌들(421)의 형식으로 모션 보상 예측 유닛이 이용할 수 있는 모션 벡터들에 의해 제어될 수 있다. 모션 보상은 또한 서브-샘플 정확한 모션 벡터들이 사용 중일 때 참조 픽처 메모리로부터 페치된 샘플 값들의 보간, 모션 벡터 예측 메커니즘 등을 포함할 수 있다.

[0036] 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))이 이용할 수 있게 되고 코딩된 비디오 비트스트림에 포함된 파라미터들에 의해 제어되지만, 또한 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서로) 부분들의 디코딩 동안 획득된 메타-정보에 응답할 수 있을 뿐만 아니라, 이전에 재구성된 및 루프-필터링된 샘플 값들에 응답할 수도 있는 인-루프 필터(in-loop filter) 기술들을 포함할 수 있다.

[0037] 루프 필터 유닛(456)의 출력은 렌더링 디바이스(312))에 출력될 뿐만 아니라 미래의 인터-픽처 예측에서 사용하기 위해 참조 픽처 메모리(456)에 저장될 수도 있는 샘플 스트림일 수 있다.

[0038] 특정 코딩된 픽처들은, 완전히 재구성되면, 미래 예측을 위한 참조 픽처들로서 사용될 수 있다. 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 참조 픽처로서 식별되면(예를 들어, 파서(420)에 의해), 현재 참조 픽처(456))는 참조 픽처 버퍼(457)의 일부가 될 수 있고, 다음의 코딩된 픽처의 재구성을 개시하기 전에 새로운 현재 픽처 메모리가 재할당될 수 있다.

[0039] 비디오 디코더(420)는 ITU-T Rec. H.265와 같은 표준에 문서화될 수 있는 미리 결정된 비디오 압축 기술에 따라 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스는, 비디오 압축 기술 문서 또는 표준에서 그리고 구체적으로 그 내부의 프로파일들의 문서에서 특정된 바와 같은, 비디오 압축 기술 또는 표준의 구문을 고수한다는 점에서, 코딩된 비디오 시퀀스는 사용 중인 비디오 압축 기술 또는 표준에 의해 특정된 구문을 따를 수 있다. 또한 준수를 위해 필요한 것은 코딩된 비디오 시퀀스의 복잡성이 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 경계 내에 있는 것일 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.

[0040] 일 실시예에서, 수신기(410)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(420)에 의해 사용될 수 있다. 추가적인 데이터는 예를 들어, 시간, 공간, 또는 SNR 향상 계층들, 중복 슬라이스들, 중복 픽처들, 순방향 오류 정정 코드들 등의 형식일 수 있다.

[0041] 도 5는 본 개시내용의 일 실시예에 따른 비디오 인코더(303)의 기능 블록도일 수 있다.

[0042] 인코더(303)는 인코더(303)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(301)(인코더의 일부가 아님)로부터 비디오 샘플들을 수신할 수 있다.

[0043] 비디오 소스(301)는, 임의의 적합한 비트 심도(예를 들어: 8 비트, 10 비트, 12개의 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 형식으로 인코더(303)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(301)는 이전에 준비된 비디오를 저장하는 저장 디바이스일 수 있다. 영상 회의 시스템에서, 비디오 소스(303)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 공간 어레이로서 조직될 수 있고, 여기서 각각의 픽셀은 사용 중인 샘플링 구조, 색 공간 등에 따라 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 픽셀들과 샘플들 간의 관계를 쉽게 이해할 수 있다. 아래의 설명은 샘플들에 초점을 맞춘다.

[0044] 일 실시예에 따르면, 인코더(303)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 컨트롤러(550)의 하나의 기능이다. 컨트롤러는 아래에 설명되는 바와 같이 다른 기능 유닛들을 제어하고 이들 유닛들에 기능적으로 결합된다. 결합은 명확성을 위해 묘사되어 있지 않다. 컨트롤러에 의해 설정된 파라미터들은 레이트 제어 관련 파라미터들(픽처 스킵, 양자화기, 레이트-왜곡 최적화 기법들의 람다 값들, ...), 픽처 크기, 픽처 그룹(GOP) 레이아웃, 최대 모션 벡터 검색 범위 등을 포함할 수 있다. 본 기술분야의 통상의 기술자는 컨트롤러(550)의 다른 기능들을 쉽게 식별할 수 있는데 그 이유는 그것들이 특정 시스템 설계에 대해 최적화된 비디오 인코더(303)에 관련될 수 있기 때문이다.

[0045] 일부 비디오 인코더들은 본 기술분야의 통상의 기술자가 "코딩 루프"로서 쉽게 인식하는 것에서 동작한다. 과도하게 단순화된 설명으로서, 코딩 루프는, 코딩될 입력 픽처, 및 참조 픽처(들)에 기초하여 심벌들을 생성하는 것을 담당하는, 인코더(530)(이후 "소스 코더")의 인코딩 부분, 및 인코더(303)에 임베드된 (로컬) 디코더(533)로 구성될 수 있는데, 상기 디코더는 (원격) 디코더가 또한 생성하는 샘플 데이터를 생성하기 위해 심벌들을 재구성한다(심벌들과 코딩된 비디오 비트스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문에). 재구성된 샘플 스트림은 참조 픽처 메모리(534)에 입력된다. 심벌 스트림의 디코딩이 디코더 위치(로컬 또는 원격)와는 독립적으로 비트-정확한 결과들을 야기하기 때문에, 참조 픽처 버퍼 콘텐츠도 또한 로컬 인코더와 원격 인코더 사이에서 비트 정확하다. 다시 말해서, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "보는" 것과 정확히 동일한 샘플 값들을 참조 픽처 샘플들로서 "본다". 참조 픽처 동기성의 이 기본적인 원리(그리고 결과적인 드리프트, 예를 들어, 채널 오류들 때문에 동기성이 유지

될 수 없는 경우)는 본 기술분야의 통상의 기술자에게 잘 알려져 있다.

[0046] "로컬" 디코더(533)의 동작은 도 4와 관련하여 위에서 이미 상세히 설명된 "원격" 디코더(310)와 동일할 수 있다. 그러나, 또한 도 4를 잠시 참조하면, 심벌들이 이용가능하고 엔트로피 코더(545) 및 파서(420)에 의한 코딩된 비디오 시퀀스로의 심벌들의 인코딩/디코딩이 무손실일 수 있기 때문에, 채널(412), 수신기(410), 버퍼(415), 및 파서(420)를 포함하는, 디코더(310)의 엔트로피 디코딩 부분들은 로컬 디코더(533)에서 완전히 구현되지 않을 수 있다.

[0047] 이 시점에서 이루어질 수 있는 관찰은, 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한 필연적으로, 대응하는 인코더에서, 실질적으로 동일한 기능 형식으로 존재할 필요가 있다는 점이다. 이러한 이유로, 개시된 주제는 디코더 동작에 초점을 맞춘다. 인코더 기술들은 포괄적으로 설명된 디코더 기술들의 역이기 때문에 그것들에 대한 설명은 축약될 수 있다. 특정 영역들에서만 더 상세한 설명이 요구되고 아래에 제공된다.

[0048] 동작의 일부로서, 소스 코더(530)는, "참조 프레임들"로서 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 프레임들을 참조하여 예측적으로 입력 프레임을 코딩하는, 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(532)은 입력 프레임의 픽셀 블록들과 입력 프레임에 대한 예측 참조(들)로서 선택될 수 있는 참조 프레임(들)의 픽셀 블록들 간의 차이들을 코딩한다.

[0049] 로컬 비디오 디코더(533)는, 소스 코더(530)에 의해 생성된 심벌들에 기초하여, 참조 프레임들로서 지정될 수 있는 프레임들의 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 있을 때, 재구성된 비디오 시퀀스는 전형적으로 일부 오류들을 갖는 소스 비디오 시퀀스의 복제본일 수 있다. 로컬 비디오 디코더(533)는 참조 프레임들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고 재구성된 참조 프레임들이 참조 픽처 캐시(534)에 저장되게 할 수 있다. 이러한 방식으로, 인코더(303)는 (송신 오류들이 없이) 원단(far-end) 비디오 디코더에 의해 획득될 재구성된 참조 프레임들로서 공통 콘텐츠를 갖는 재구성된 참조 프레임들의 사본들을 로컬로 저장할 수 있다.

[0050] 예측기(535)는 코딩 엔진(532)에 대한 예측 검색들을 수행할 수 있다. 즉, 코딩될 새로운 프레임에 대해, 예측기(535)는 새로운 픽처들에 대한 적절한 예측 참조로서의 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. 예측기(535)는 적절한 예측 참조들을 찾기 위해 샘플 블록 바이 픽셀 블록(sample block-by-pixel block) 기준으로 동작할 수 있다. 일부 경우들에서, 예측기(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.

[0051] 컨트롤러(550)는, 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하여, 비디오 코더(530)의 코딩 동작을 관리할 수 있다.

[0052] 전술한 모든 기능 유닛들의 출력은 엔트로피 코더(545)에서 엔트로피 코딩을 겪을 수 있다. 엔트로피 코더는 다양한 기능 유닛들에 의해 생성된 심벌들을, 예를 들어, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 본 기술분야의 통상의 기술자에게 알려진 기술들에 따라 심벌들을 무손실 압축함으로써, 코딩된 비디오 시퀀스로 변환한다.

[0053] 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(530)로부터의 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림(소스들이 도시되지 않음)과 병합할 수 있다.

[0054] 컨트롤러(550)는 인코더(303)의 동작을 관리할 수 있다. 코딩 동안, 컨트롤러(550)는, 각각의 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 프레임 타입들 중 하나로서 할당될 수 있다:

[0055] 인트라 픽처(I 픽처)는 예측의 소스로서 시퀀스 내의 임의의 다른 프레임을 사용하지 않고 코딩되고 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은, 예를 들어, 독립적인 디코더 리프레시(Independent Decoder Refresh) 픽처들을 포함하는, 상이한 타입의 인트라 픽처들을 허용한다. 본 기술분야의 통상의 기술자는 I 픽

처들의 해당 변형들 및 그것들 각각의 응용들 및 특징들을 인식한다.

[0056] 예측 픽처(P 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 하나의 모션 벡터 및 참조 인덱스를 이용하여 인트라 예측 또는 인터 예측을 이용하여 코딩되고 디코딩될 수 있는 것일 수 있다.

[0057] 양방향 예측 픽처(B 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 2개의 모션 벡터 및 참조 인덱스를 이용하여 인트라 예측 또는 인터 예측을 이용하여 코딩되고 디코딩될 수 있는 것일 수 있다. 유사하게, 다중-예측 픽처들은 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.

[0058] 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 각각의 픽처들에 적용되는 코딩 할당에 의해 결정된 다른(이미 코딩된) 블록들을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 픽처들의 블록들은 비-예측적으로 코딩될 수 있거나 그것들은 동일한 픽처의 이미 코딩된 블록들을 참조하여 예측적으로 코딩될 수 있다(공간 예측 또는 인트라 예측). P 픽처들의 픽셀 블록들은, 하나의 이전에 코딩된 참조 픽처를 참조하여 공간 예측을 통해 또는 시간 예측을 통해, 예측적으로 코딩될 수 있다. B 픽처들의 블록들은, 하나 또는 2개의 이전에 코딩된 참조 픽처를 참조하여 공간 예측을 통해 또는 시간 예측을 통해, 예측적으로 코딩될 수 있다.

[0059] 비디오 코더(303)는 ITU-T Rec. H.265와 같은 미리 결정된 비디오 코딩 기술 또는 표준에 따라 코딩 동작들을 수행할 수 있다. 그것의 동작 중에, 비디오 코더(303)는, 입력 비디오 시퀀스에서 시간 및 공간 중복성을 이용하는 예측 코딩 동작들을 포함하여, 다양한 압축 동작들을 수행할 수 있다. 따라서, 코딩된 비디오 데이터는 사용 중인 비디오 코딩 기술 또는 표준에 의해 특정된 구문을 따를 수 있다.

[0060] 일 실시예에서, 송신기(540)는 인코딩된 비디오와 함께 추가적인 데이터를 송신할 수 있다. 비디오 코더(530)는 코딩된 비디오 시퀀스의 일부로서 그러한 데이터를 포함할 수 있다. 추가적인 데이터는 시간/공간/SNR 향상 계층들, 중복 픽처들 및 슬라이스들과 같은 다른 형식의 중복 데이터, SEI(Supplementary Enhancement Information) 메시지들, VUI(Visual Usability Information) 파라미터 세트 프래그먼트들 등을 포함할 수 있다.

[0061] 일 실시예에서, 360 비디오는 카메라들의 세트 또는 다수의 렌즈를 갖는 카메라 디바이스에 의해 캡처된다. 카메라들은 전형적으로 카메라 세트의 중심점 주위의 전방향들을 커버한다. 동일한 타임 인스턴스의 이미지들은 스티칭되고, 아마도 회전되고, 프로젝션되고, 픽처 상에 매핑된다. 패킹된 픽처들은 코딩된 비디오 비트스트림에 코딩된 것으로서 인코딩되고, 특정 미디어 컨테이너 파일 포맷에 따라 스트리밍된다. 파일은 프로젝션 및 패킹 정보와 같은 메타데이터를 포함한다.

[0062] 일 실시예에서, 360 비디오는 등장방형 프로젝션(equirectangular projection, ERP)을 이용하여 2D 비디오에 프로젝션될 수 있다. ERP 프로젝션은 심(seam) 아티팩트들을 야기할 수 있다. 패딩된 ERP(PERP) 포맷은 ERP 픽처의 좌측 및 우측 경계들을 포함하는 재구성된 뷰포트들에서 심 아티팩트들을 효과적으로 감소시킬 수 있다. 그러나, 패딩 및 블렌딩은 심 문제를 완전히 해결하기에 충분하지 않을 수 있다.

[0063] 일 실시예에서, 심 아티팩트를 감소시키기 위해 ERP 또는 PERP에 대해 수평 지오메트리 패딩이 적용될 수 있다. PERP에 대한 패딩 프로세스는, 오프셋이 패딩된 영역들의 크기를 고려하기 위해 픽처 폭 대신에 패딩되지 않은 ERP 폭에 기초할 수 있다는 점을 제외하고는, ERP에 대한 것과 동일할 수 있다. 참조 블록이 좌측(우측) 참조 픽처 경계 밖에 있다면, 그것은 ERP 폭만큼 우측(좌측)으로 시프트된 "랩-어라운드된" 참조 블록으로 대체될 수 있다. 전통적인 반복적 패딩은 수직 방향으로 이용될 수 있다. 좌측 및 우측 패딩된 영역들의 블렌딩은, 후처리 동작으로서, 루프 밖에서 유지된다. 후처리로서 심 아티팩트를 제거하기 위해 오버래핑된 영역(overwrapped region)이 블렌딩될 수 있다.

[0064] 일 실시예에서, ERP 및 PERP 포맷들에 대한 참조 픽처들의 수평 지오메트리 패딩을 가능하게 하는 구문이 도 6에 도시되어 있다.

[0065] **sps_ref_wraparound_enabled_flag**(602)가 1과 동등한 것은 인터 예측을 위해 수평 랩-어라운드 모션 보상이 사용된다는 것을 특정한다. sps_ref_wraparound_enabled_flag가 0과 동등한 것은 이 모션 보상 방법이 적용되지 않는다는 것을 특정한다.

[0066] **ref_wraparound_offset**(603)은 수평 랩-어라운드 위치를 계산하기 위해 사용되는 루마 샘플들 내의 오프셋을 특정한다. ref_wraparound_offset는 pic_width_in_luma_samples - 1보다 커야 하고,

pic_width_in_luma_samples보다 크지 않아야 하고, MinCbSizeY의 정수 배수이어야 한다.

[0067] 일 실시예에서, ERP 및 PERP 포맷들에 대한 참조 픽처들의 수평 지오메트리 패딩을 가능하게 하는 구문 요소들이 도 7에 도시되어 있다.

[0068] **sps_ref_wraparound_enabled_flag**(702)가 1과 동등한 것은 인터 예측을 위해 수평 랩-어라운드 모션 보상이 사용된다는 것을 특정한다. sps_ref_wraparound_enabled_flag가 0과 동등한 것은 이 모션 보상 방법이 적용되지 않는다는 것을 특정한다.

[0069] **left_wraparound_padding_width**(703)는 루마 샘플들에서 좌측 패딩 영역의 폭을 특정한다. ref_wraparound_offset는 0보다 크거나 그와 동등해야 하고, pic_width_in_luma_samples / 2보다 크지 않아야 하고, MinCbSizeY의 정수 배수이어야 한다.

[0070] **right_wraparound_padding_width**(704)는 루마 샘플들에서 우측 패딩 영역의 폭을 특정한다. ref_wraparound_offset는 0보다 크거나 그와 동등해야 하고, pic_width_in_luma_samples / 2보다 크지 않아야 하고, MinCbSizeY의 정수 배수이어야 한다.

[0071] 이제 도 8을 참조하면, 비디오 데이터를 인코딩하기 위한 방법(900)의 단계들을 예시하는 동작 흐름도가 묘사되어 있다.

[0072] 901에서, 방법(900)은 파라미터 세트를 파싱하는 단계를 포함한다.

[0073] 902에서, 방법(900)은 픽처 분할 정보를 디코딩하는 단계를 포함한다.

[0074] 903에서, 방법(900)은 패딩이 서브-영역들의 경계들에 적용되는지를 결정하는 단계를 포함한다.

[0075] 904에서, 방법(900)은 패딩이 서브-영역들의 경계들에 적용되면 패딩 타입을 파싱하는 단계를 포함한다.

[0076] 905에서, 방법(900)은 패딩이 서브-영역들의 경계들에 적용되지 않으면 패딩 없이 각각의 서브-영역을 디코딩하는 단계를 포함한다.

[0077] 906에서, 방법(900)은 랩-어라운드 패딩이 적용되는지를 결정하는 단계를 포함한다.

[0078] 907에서, 방법(900)은 랩-어라운드 패딩이 적용되면 랩-어라운드 패딩으로 서브-영역들의 경계들을 패딩하는 단계를 포함한다.

[0079] 908에서, 방법(900)은 랩-어라운드 패딩이 적용되지 않으면 비-랩-어라운드 패딩으로 서브-영역들의 경계들을 패딩하는 단계를 포함한다.

[0080] 일 실시예에서, 랩-어라운드 오프셋 값은 다음의 도출 프로세스에 의해 획득될 수 있다:

```
if ref_wraparound_offset is present
    wrapAroundOffset = ref_wraparound_offset
else if left_wraparound_padding_width and right_wraparound_padding_width are
present
    wrapAroundOffset = pic_width_in_luma_samples –
( left_wraparound_padding_width + right_wraparound_padding_width )
else
    wrapAroundOffset = pic_width_in_luma_samples
```

[0081]

[0082] 일 실시예에서, ERP 및 PERP 포맷들에 대한 참조 픽처들의 수평 지오메트리 패딩을 가능하게 하기 위해, 루마 및 크로마 샘플 보간 프로세스들이 수정될 수 있다.

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{그 외} \end{cases}$$

$$\text{ClipH}(o, W, x) = \begin{cases} (o - (-x) \,\%\, o) \,\%\, o & ; \quad x < 0 \\ (x - W) \,\%\, o + W - o & ; \quad x > P - 1 \\ x & ; \quad \text{그 외} \end{cases}$$

[0083]

[0084] 루마 샘플 보간 프로세스. 이 프로세스에 대한 입력들은 풀-샘플 단위의 루마 위치( $xInt_L$, $yInt_L$ ), 분수-샘플 단위의 루마 위치( $xFrac_L$, $yFrac_L$ ), 및 루마 참조 샘플 어레이 $refPicLX_L$을 포함할 수 있다. 이 프로세스의 출력들은 예측된 루마 샘플 값 $predSampleLX_L$을 포함할 수 있다.

[0085] 변수 shift1, shift2 및 shift3은 다음과 같이 도출된다. 변수 shift1은 Min( 4, $BitDepth_Y$ - 8 )과 동등하게 설정되고, 변수 shift2는 6과 동등하게 설정되고, 변수 shift3은 Max( 2, 14 - $BitDepth_Y$ )와 동등하게 설정된다. 변수 picW는 pic_width_in_luma_samples와 동등하게 설정되고 변수 picH는 pic_height_in_luma_samples와 동등하게 설정된다. 변수 xOffset는 wrapAroundOffset와 동등하게 설정된다. $xFrac_L$ 또는 $yFrac_L$과 동등한 각각의 1/16 분수 샘플 위치 p에 대한 루마 보간 필터 계수들 $f_L[ p ]$가 특정될 수 있다.

[0086] 예측된 루마 샘플 값 $predSampleLX_L$은 다음과 같이 도출된다. $xFrac_L$ 및 $yFrac_L$ 둘 다가 0과 동등하면, 다음이 적용된다. sps_ref_wraparound_enabled_flag가 0과 동등하면, $predSampleLX_L$의 값은 $predSampleLX_L$ = $refPicLX_L$[ Clip3( 0, picW - 1, $xInt_L$ ) ][ Clip3( 0, picH - 1, $yInt_L$ ) ] << shift3으로서 도출된다. 그렇지 않으면, $predSampleLX_L$의 값은 다음과 같이 도출된다: $predSampleLX_L$ = $refPicLX_L$[ ClipH( xOffset, picW, $xInt_L$ ) ][ Clip3( 0, picH - 1, $yInt_L$ ) ] << shift3. 그렇지 않고 $xFrac_L$이 0과 동등하지 않고 $yFrac_L$이 0과 동등하면, 다음이 적용된다. $yPos_L$의 값은 $yPos_L$ = Clip3( 0, picH - 1, $yInt_L$ )로서 도출된다.

[0087] sps_ref_wraparound_enabled_flag가 0과 동등하면, $predSampleLX_L$의 값은 다음과 같이 도출된다:

$$\begin{aligned} predSampleLX_L = \\ (\, & f_L[\, xFrac_L \,][\, 0 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L - 3 \,) \,][\, yPos_L \,] + \\ & f_L[\, xFrac_L \,][\, 1 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L - 2 \,) \,][\, yPos_L \,] + \\ & f_L[\, xFrac_L \,][\, 2 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L - 1 \,) \,][\, yPos_L \,] + \\ & f_L[\, xFrac_L \,][\, 3 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L \,) \,][\, yPos_L \,] + \\ & f_L[\, xFrac_L \,][\, 4 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L + 1 \,) \,][\, yPos_L \,] + \\ & f_L[\, xFrac_L \,][\, 5 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L + 2 \,) \,][\, yPos_L \,] + \\ & f_L[\, xFrac_L \,][\, 6 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L + 3 \,) \,][\, yPos_L \,] + \\ & f_L[\, xFrac_L \,][\, 7 \,] * refPicLX_L[\, \text{Clip3}(\, 0, picW - 1, xInt_L + 4 \,) \,][\, yPos_L \,] \,) >> shift1 \end{aligned}$$

[0088]

[0089] 그렇지 않으면, $predSampleLX_L$의 값은 다음과 같이 도출된다:

$$predSampleLX_L = ( f_L[ xFrac_L ][ 0 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 3 ) ][ yPos_L ] + f_L[ xFrac_L ][ 1 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 2 ) ][ yPos_L ] + f_L[ xFrac_L ][ 2 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 1 ) ][ yPos_L ] + f_L[ xFrac_L ][ 3 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L ) ][ yPos_L ] + f_L[ xFrac_L ][ 4 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 1 ) ][ yPos_L ] + f_L[ xFrac_L ][ 5 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 2 ) ][ yPos_L ] + f_L[ xFrac_L ][ 6 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 3 ) ][ yPos_L ] + f_L[ xFrac_L ][ 7 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 4 ) ][ yPos_L ] ) >> shift.$$

[0090]

[0091] 그렇지 않고 $xFrac_L$이 0과 동등하고 $yFrac_L$이 0과 동등하지 않으면, $predSampleLX_L$의 값은 다음과 같이 도출된다. sps_ref_wraparound_enabled_flag가 0과 동등하면, $xPos_L$의 값은 $xPos_L = Clip3( 0, picW - 1, xInt_L )$로서 도출된다. 그렇지 않으면, $xPos_L$의 값은 다음과 같이 도출된다: $xPos_L = ClipH( xOffset, picW, xInt_L )$.

[0092] 예측된 루마 샘플 값 $predSampleLX_L$은 다음과 같이 도출된다:

$$predSampleLX_L = ( f_L[ yFrac_L ][ 0 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L - 3 ) ] + f_L[ yFrac_L ][ 1 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L - 2 ) ] + f_L[ yFrac_L ][ 2 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L - 1 ) ] + f_L[ yFrac_L ][ 3 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L ) ] + f_L[ yFrac_L ][ 4 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L + 1 ) ] + f_L[ yFrac_L ][ 5 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L + 2 ) ] + f_L[ yFrac_L ][ 6 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L + 3 ) ] + f_L[ yFrac_L ][ 7 ] * refPicLX_L[ xPos_L ][ Clip3( 0, picH - 1, yInt_L + 4 ) ] ) >> shift1.$$

[0093]

[0094] 그렇지 않고 $xFrac_L$이 0과 동등하지 않고 $yFrac_L$이 0과 동등하지 않으면, $predSampleLX_L$의 값은 다음과 같이 도출된다. sps_ref_wraparound_enabled_flag가 0과 동등하면, n = 0..7인 샘플 어레이 temp[ n ]이 다음과 같이 도출된다:

$$yPos_L = Clip3( 0, picH - 1, yInt_L + n - 3 )$$

$$temp[ n ] = ( f_L[ xFrac_L ][ 0 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L - 3 ) ][ yPos_L ] + f_L[ xFrac_L ][ 1 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L - 2 ) ][ yPos_L ] + f_L[ xFrac_L ][ 2 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L - 1 ) ][ yPos_L ] + f_L[ xFrac_L ][ 3 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L ) ][ yPos_L ] + f_L[ xFrac_L ][ 4 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L + 1 ) ][ yPos_L ] + f_L[ xFrac_L ][ 5 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L + 2 ) ][ yPos_L ] + f_L[ xFrac_L ][ 6 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L + 3 ) ][ yPos_L ] + f_L[ xFrac_L ][ 7 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L + 4 ) ][ yPos_L ] ) >> shift1.$$

[0095]

[0096] 그렇지 않으면, n = 0..7인 샘플 어레이 temp[ n ]이 다음과 같이 도출된다:

$$yPos_L = Clip3( 0, picH - 1, yInt_L + n - 3 )$$

$$\begin{aligned} temp[ n ] = ( & f_L[ xFrac_L ][ 0 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 3 ) ][ yPos_L ] + \\ & f_L[ xFrac_L ][ 1 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 2 ) ][ yPos_L ] + \\ & f_L[ xFrac_L ][ 2 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 1 ) ][ yPos_L ] + \\ & f_L[ xFrac_L ][ 3 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L ) ][ yPos_L ] + \\ & f_L[ xFrac_L ][ 4 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 1 ) ][ yPos_L ] + \\ & f_L[ xFrac_L ][ 5 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 2 ) ][ yPos_L ] + \\ & f_L[ xFrac_L ][ 6 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 3 ) ][ yPos_L ] + \\ & f_L[ xFrac_L ][ 7 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 4 ) ][ yPos_L ] ) >> shift1. \end{aligned}$$

[0097]

[0098] 예측된 루마 샘플 값 $predSampleLX_L$은 다음과 같이 도출된다:

$$\begin{aligned} predSampleLX_L = ( & f_L[ yFrac_L ][ 0 ] * temp[ 0 ] + \\ & f_L[ yFrac_L ][ 1 ] * temp[ 1 ] + \\ & f_L[ yFrac_L ][ 2 ] * temp[ 2 ] + \\ & f_L[ yFrac_L ][ 3 ] * temp[ 3 ] + \\ & f_L[ yFrac_L ][ 4 ] * temp[ 4 ] + \\ & f_L[ yFrac_L ][ 5 ] * temp[ 5 ] + \\ & f_L[ yFrac_L ][ 6 ] * temp[ 6 ] + \\ & f_L[ yFrac_L ][ 7 ] * temp[ 7 ] ) >> shift2. \end{aligned}$$

[0099]

[0100] 크로마 샘플 보간 프로세스. 이 프로세스에 대한 입력들은 풀-샘플 단위의 크로마 위치( $xInt_C$, $yInt_C$ ), 1/32 분수-샘플 단위의 크로마 위치( $xFrac_C$, $yFrac_C$ ), 및 크로마 참조 샘플 어레이 $refPicLX_C$를 포함할 수 있다. 이 프로세스의 출력들은 예측된 크로마 샘플 값 $predSampleLX_C$를 포함할 수 있다.

[0101] 변수 shift1, shift2 및 shift3은 다음과 같이 도출된다. 변수 shift1은 Min( 4, $BitDepth_C$ - 8 )과 동등하게 설정되고, 변수 shift2는 6과 동등하게 설정되고, 변수 shift3은 Max( 2, 14 - $BitDepth_C$ )와 동등하게 설정된다. 변수 $picW_C$는 pic_width_in_luma_samples / SubWidthC와 동등하게 설정되고 변수 $picH_C$는 pic_height_in_luma_samples / SubHeightC와 동등하게 설정된다. 변수 $xOffset_C$는 wrapAroundOffset / SubWidthC와 동등하게 설정된다. $xFrac_C$ 또는 $yFrac_C$와 동등한 각각의 1/32 분수 샘플 위치 p에 대한 크로마 보간 필터 계수들 $f_C$[ p ]가 특정될 수 있다.

[0102] 예측된 크로마 샘플 값 $predSampleLX_C$는 다음과 같이 도출된다. $xFrac_C$ 및 $yFrac_C$ 둘 다가 0과 동등하면, 다음이 적용된다. sps_ref_wraparound_enabled_flag가 0과 동등하면, $predSampleLX_C$의 값은 다음과 같이 도출된다:

$$predSampleLX_C = refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C ) ][ Clip3( 0, picH_C - 1, yInt_C ) ] << shift3.$$

[0103]

[0104] 그렇지 않으면, $predSampleLX_C$의 값은 $predSampleLX_C$ = $refPicLX_C$[ ClipH( $xOffset_C$, $picW_C$, $xInt_C$ ) ][ Clip3( 0, $picH_C$ - 1, $yInt_C$ ) ] << shift3으로서 도출된다.

[0105] 그렇지 않고 $xFrac_C$가 0과 동등하지 않고 $yFrac_C$가 0과 동등하면, 다음이 적용된다. $yPos_C$의 값은 $yPos_C$ = Clip3( 0, $picH_C$ - 1, $yInt_C$ )로서 도출된다. sps_ref_wraparound_enabled_flag가 0과 동등하면,

$predSampleLX_C$의 값은 다음과 같이 도출된다:

$$predSampleLX_C = ( f_C[ xFrac_C ][ 0 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C - 1 ) ][ yInt_C ] + f_C[ xFrac_C ][ 1 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C ) ][ yInt_C ] + f_C[ xFrac_C ][ 2 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C + 1 ) ][ yInt_C ] + f_C[ xFrac_C ][ 3 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C + 2 ) ][ yInt_C ] ) >> shift1.$$

[0106]

[0107] 그렇지 않으면, $predSampleLX_C$의 값은 다음과 같이 도출된다:

$$predSampleLX_C = ( f_C[ xFrac_C ][ 0 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C - 1 ) ][ yPos_C ] + f_C[ xFrac_C ][ 1 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C ) ][ yPos_C ] + f_C[ xFrac_C ][ 2 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C + 1 ) ][ yPos_C ] + f_C[ xFrac_C ][ 3 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C + 2 ) ][ yPos_C ] ) >> shift1.$$

[0108]

[0109] 그렇지 않고 $xFrac_C$가 0과 동등하고 $yFrac_C$가 0과 동등하지 않으면, $predSampleLX_C$의 값은 다음과 같이 도출된다. sps_ref_wraparound_enabled_flag가 0과 동등하면, $xPos_C$의 값은 다음과 같이 도출된다: $xPos_C = Clip3( 0, picW_C - 1, xInt_C )$. 그렇지 않으면, $xPos_C$의 값은 다음과 같이 도출된다: $xPos_C = ClipH( xOffset_C, picW_C, xInt_C )$.

[0110] 예측된 크로마 샘플 값 $predSampleLX_C$는 다음과 같이 도출된다:

$$predSampleLX_C = ( f_C[ yFrac_C ][ 0 ] * refPicLX_C[ xPos_C ][ Clip3( 0, picH_C - 1, yInt_C - 1 ) ] + f_C[ yFrac_C ][ 1 ] * refPicLX_C[ xPos_C ][ Clip3( 0, picH_C - 1, yInt_C ) ] + f_C[ yFrac_C ][ 2 ] * refPicLX_C[ xPos_C ][ Clip3( 0, picH_C - 1, yInt_C + 1 ) ] + f_C[ yFrac_C ][ 3 ] * refPicLX_C[ xPos_C ][ Clip3( 0, picH_C - 1, yInt_C + 2 ) ] ) >> shift1.$$

[0111]

[0112] 그렇지 않고 $xFrac_C$가 0과 동등하지 않고 $yFrac_C$가 0과 동등하지 않으면, $predSampleLX_C$의 값은 다음과 같이 도출된다. sps_ref_wraparound_enabled_flag가 0과 동등하면, n = 0..3인 샘플 어레이 temp[ n ]이 다음과 같이 도출된다:

$$yPos_C = Clip3( 0, picH_C - 1, yInt_C + n - 1 )$$
$$temp[ n ] = ( f_C[ xFrac_C ][ 0 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C - 1 ) ][ yPos_C ] + f_C[ xFrac_C ][ 1 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C ) ][ yPos_C ] + f_C[ xFrac_C ][ 2 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C + 1 ) ][ yPos_C ] + f_C[ xFrac_C ][ 3 ] * refPicLX_C[ Clip3( 0, picW_C - 1, xInt_C + 2 ) ][ yPos_C ] ) >> shift1.$$

[0113]

[0114] 그렇지 않으면, n = 0..3인 샘플 어레이 temp[ n ]이 다음과 같이 도출된다:

$$yPos_C = Clip3( 0, picH_C - 1, yInt_C + n - 1 )$$
$$temp[ n ] = ( f_C[ xFrac_C ][ 0 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C - 1 ) ][ yPos_C ] + f_C[ xFrac_C ][ 1 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C ) ][ yPos_C ] + f_C[ xFrac_C ][ 2 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C + 1 ) ][ yPos_C ] + f_C[ xFrac_C ][ 3 ] * refPicLX_C[ ClipH( xOffset_C, picW_C, xInt_C + 2 ) ][ yPos_C ] ) >> shift1.$$

[0115]

[0116] 예측된 크로마 샘플 값 $predSampleLX_C$는 다음과 같이 도출된다:

$$predSampleLX_C = ( f_C[ yFrac_C ][ 0 ] * temp[ 0 ] + f_C[ yFrac_C ][ 1 ] * temp[ 1 ] + f_C[ yFrac_C ][ 2 ] * temp[ 2 ] + f_C[ yFrac_C ][ 3 ] * temp[ 3 ] ) >> shift2.$$

[0117]

[0118] 일 실시예에서, sps_ref_wraparound_enabled_flag가 0과 동등하거나 존재하지 않으면, 전통적인 반복적 패딩이 적용될 수 있다. 그렇지 않으면, 랩-어라운드 패딩이 적용될 수 있다.

[0119] 일 실시예에서, 랩-어라운드 패딩은 수평 경계와 수직 경계 둘 다에서 적용될 수 있다. 하이 레벨 구문 구조에서의 플래그가 랩-어라운드 패딩이 수평 및 수직 둘 다에서 적용된다는 것을 지시할 수 있다.

[0120] 일 실시예에서, 랩-어라운드 패딩은 타일 또는 타일 그룹 경계들에서 적용될 수 있다. 하이 레벨 구문 구조에서의 플래그가 랩-어라운드 패딩이 수평 및 수직 둘 다에서 적용된다는 것을 지시할 수 있다.

[0121] 일 실시예에서, 참조 픽처는 모션 보상된 예측에 대해 현재 픽처와 동일할 수 있다. 랩-어라운드 패딩은, 현재 픽처가 참조일 때, 현재 픽처의 경계에서 적용될 수 있다.

[0122] 일 실시예에서, 픽처가 하나 이상의 슬라이스의 직사각형 영역인 하나 이상의 서브-픽처로 분할될 때, 각각의 서브-픽처의 경계는 픽처 경계로서 핸들링되거나 그렇지 않을 수 있다. 서브-픽처를 픽처로서 취급하는 것은 각각의 서브-픽처의 경계가 모션 보상된 예측을 위해 패딩될 수 있다는 것을 지시한다.

[0123] 도 9를 참조하면, SPS(또는 임의의 다른 파라미터 세트)에서, subpic_treated_as_pic_flag[ i ]가 1과 동등한 것은 CVS 내의 각각의 코딩된 픽처의 i 번째 서브픽처가 디코딩 프로세스에서 픽처로서 취급된다는 것을 특정한다. subpic_treated_as_pic_flag[ i ]가 0과 동등한 것은 CVS 내의 각각의 코딩된 픽처의 i 번째 서브픽처가 디코딩 프로세스에서 픽처로서 취급되지 않는다는 것을 특정한다. 존재하지 않을 때, subpic_treated_as_pic_flag[ i ]의 값은 0과 동등한 것으로 추론된다.

[0124] 동일한 실시예에서, subpic_treated_as_pic_flag[ i ]가 1과 동등하여, i 번째 서브-픽처가 픽처로서 취급된다는 것을 지시할 때, 플래그 subpic_ref_wraparound_enabled_flag[ i ]가 시그널링된다. subpic_ref_wraparound_enabled_flag[ i ]가 1과 동등하다. subpic_ref_wraparound_enabled_flag[ i ]가 1과 동등한 것은 각각의 코딩된 픽처의 i 번째 서브-픽처의 경계에서의 인터 예측에서 수평 랩-어라운드 모션 보상이 적용된다는 것을 특정한다. sps_ref_wraparound_enabled_flag[ i ]가 0과 동등한 것은 각각의 코딩된 픽처의 i 번째 서브-픽처의 경계에서 수평 랩-어라운드 모션 보상이 적용되지 않는다는 것을 특정한다.

[0125] 동일한 실시예에서, subpic_ref_wraparound_offset_minus1[ i ] + 1은 MinCbSizeY 루마 샘플들의 단위로 수평 랩-어라운드 위치를 계산하기 위해 사용되는, 각각의 코딩된 픽처의 i 번째 서브-픽처의 오프셋을 특정한다.

[0126] 동일한 실시예에서, 보간 필터링 프로세스는 다음과 같다. 루마 샘플 보간 필터링 프로세스. 이 프로세스에 대한 입력들은 풀-샘플 단위의 루마 위치( $xInt_L$, $yInt_L$ ), 분수-샘플 단위의 루마 위치( $xFrac_L$, $yFrac_L$ ), 참조 픽처의 상기 좌측 상단 루마 샘플에 대한 참조 샘플 패딩에 대한 경계 블록의 좌측 상단 샘플을 특정하는 풀-샘플 단위의 루마 위치( $xSbInt_L$, $ySbInt_L$ ), 루마 참조 샘플 어레이 $refPicLX_L$, 절반 샘플 보간 필터 인덱스 hpelIfIdx, 현재 서브블록의 폭을 특정하는 변수 sbWidth, 현재 서브블록의 높이를 특정하는 변수 sbHeight, 및 현재 픽처의 좌측 상단 루마 샘플에 대한 현재 서브블록의 좌측 상단 샘플을 특정하는 루마 위치( xSb, ySb )를 포함할 수 있다. 이 프로세스의 출력들은 예측된 루마 샘플 값 $predSampleLX_L$을 포함할 수 있다.

[0127] 변수 shift1, shift2 및 shift3은 다음과 같이 도출된다. 변수 shift1은 $Min( 4, BitDepth_Y - 8 )$과 동등하게 설정되고, 변수 shift2는 6과 동등하게 설정되고, 변수 shift3은 $Max( 2, 14 - BitDepth_Y )$와 동등하게 설정된다. 변수 picW는 pic_width_in_luma_samples와 동등하게 설정되고 변수 picH는 pic_height_in_luma_samples와 동등하게 설정된다.

[0128] $xFrac_L$ 또는 $yFrac_L$과 동등한 각각의 1/16 분수 샘플 위치 p에 대한 루마 보간 필터 계수들 $f_L[ p ]$가 다음과 같이 도출된다. MotionModelIdc[ xSb ][ ySb ]가 0보다 크고, sbWidth 및 sbHeight 둘 다가 4와 동등하면, 루마

보간 필터 계수들 $f_L[\ p\ ]$가 특정될 수 있다. 그렇지 않으면, 루마 보간 필터 계수들 $f_L[\ p\ ]$는 hpelIfIdx에 따라 표 8-11에 특정된다.

[0129] 풀-샘플 단위의 루마 위치( $xInt_i$, $yInt_i$ )는 i = 0...7에 대해 다음과 같이 도출된다. subpic_treated_as_pic_flag[ SubPicIdx ]가 1과 동등하면, 다음이 적용된다:

$$xInt_i = Clip3(\ SubPicLeftBoundaryPos,\ SubPicRightBoundaryPos,$$
$$subpic_ref_wraparound_enabled_flag[\ SubPicIdx\ ]$$
$$ClipH(\ (\ subpic_ref_wraparound_offset_minus1[\ SubPicIdx\ ] + 1\ ) * MinCbSizeY,\ SubPicRight$$
$$BoundaryPos,\ xInt_L + i - 3\ ) :$$
$$xInt_L + i - 3\ ).$$

[0130] $$yInt_i = Clip3(\ SubPicTopBoundaryPos,\ SubPicBotBoundaryPos,\ yInt_L + i - 3\ ).$$

[0131] 그렇지 않으면(subpic_treated_as_pic_flag[ SubPicIdx ]가 0과 동등하면), 다음이 적용된다:

$$xInt_i = Clip3(\ 0,\ picW - 1,\ sps_ref_wraparound_enabled_flag$$
$$ClipH(\ (\ sps_ref_wraparound_offset_minus1 + 1\ ) * MinCbSizeY,\ picW,\ xInt_L + i - 3\ ) :$$
$$xInt_L + i - 3\ )$$

[0132] $$yInt_i = Clip3(\ 0,\ picH - 1,\ yInt_L + i - 3\ )$$

[0133] 풀-샘플 단위의 루마 위치들은 i = 0...7에 대해 다음과 같이 추가로 수정된다:

[0134] $$xInt_i = Clip3(\ xSbInt_L - 3,\ xSbInt_L + sbWidth + 4,\ xInt_i\ )$$
$$yInt_i = Clip3(\ ySbInt_L - 3,\ ySbInt_L + sbHeight + 4,\ yInt_i\ )$$

[0135] 예측된 루마 샘플 값 $predSampleLX_L$은 다음과 같이 도출된다. $xFrac_L$ 및 $yFrac_L$ 둘 다가 0과 동등하면, $predSampleLX_L$의 값은 다음과 같이 도출된다:

[0136] $$predSampleLX_L = refPicLX_L[\ xInt_3\ ][\ yInt_3\ ] << shift3.$$

[0137] 그렇지 않고, $xFrac_L$이 0과 동등하지 않고 $yFrac_L$이 0과 동등하면, $predSampleLX_L$의 값은 $predSampleLX_L$ =

[0138] $$(\ \sum_{i=0}^{7} f_L[\ xFrac_L\ ][\ i\ ] * refPicLX_L[\ xInt_i\ ][\ yInt_3\ ]\ ) >> shift1$$ 로서 도출된다.

[0139] 그렇지 않고, $xFrac_L$이 0과 동등하고 $yFrac_L$이 0과 동등하지 않으면, $predSampleLX_L$의 값은 $predSampleLX_L$ =

[0140] $$(\ \sum_{i=0}^{7} f_L[\ yFrac_L\ ][\ i\ ] * refPicLX_L[\ xInt_3\ ][\ yInt_i\ ]\ ) >> shift1$$ 로서 도출된다.

[0141] 그렇지 않고 $xFrac_L$이 0과 동등하지 않고 $yFrac_L$이 0과 동등하지 않으면, $predSampleLX_L$의 값은 다음과 같이 도출된다:

[0142] n = 0..7인 샘플 어레이 temp[ n ]은 다음과 같이 도출된다: temp[ n ] =

[0143] $$(\ \sum_{i=0}^{7} f_L[\ xFrac_L\ ][\ i\ ] * refPicLX_L[\ xInt_i\ ][\ yInt_n\ ]\ ) >> shift1.$$

[0144] 예측된 루마 샘플 값 $predSampleLX_L$은 $predSampleLX_L$ =

[0145] $$(\ \sum_{i=0}^{7} f_L[\ yFrac_L\ ][\ i\ ] * temp[\ i\ ]\ ) >> shift2$$ 로서 도출된다.

[0146] 동일한 또는 다른 실시예에서, 도 10을 참조하면, SPS(또는 임의의 다른 파라미터 세트)에서, all_subpic_treated_as_pic_flag가 1과 동등한 것은 CVS 내의 각각의 코딩된 픽처의 임의의 서브-픽처가 디코딩 프로세스에서 픽처로서 취급된다는 것을 특정한다. subpic_treated_as_pic_flag[ i ]가 0과 동등한 것은

CVS 내의 각각의 코딩된 픽처의 임의의 서브픽처가 디코딩 프로세스에서 픽처로서 취급되지 않는다는 것을 특정한다.

[0147] 동일한 실시예에서, all_subpic_treated_as_pic_flag[ i ]가 1과 동등할 때, 플래그 all_subpic_ref_wraparound_enabled_flag가 시그널링된다. all_subpic_ref_wraparound_enabled_flag가 1과 동등한 것은 각각의 코딩된 픽처의 임의의 서브-픽처의 경계에서의 인터 예측에서 수평 랩-어라운드 모션 보상이 적용된다는 것을 특정한다. all_sps_ref_wraparound_enabled_flag가 0과 동등한 것은 각각의 코딩된 픽처의 임의의 서브-픽처의 경계에서 수평 랩-어라운드 모션 보상이 적용되지 않는다는 것을 특정한다.

[0148] 동일한 실시예에서, all_subpic_ref_wraparound_offset_minus1[ i ] + 1은 MinCbSizeY 루마 샘플들의 단위로 수평 랩-어라운드 위치를 계산하기 위해 사용되는, 각각의 코딩된 픽처의 임의의 서브-픽처의 오프셋을 특정한다. 동일한 실시예에서, subpic_treated_as_pic_flag[ i ]가 존재하지 않을 때, subpic_treated_as_pic_flag[ i ]의 값은 all_subpic_treated_as_pic_flag와 동등한 것으로 추론된다.

[0149] 동일한 실시예에서, subpic_ref_wraparound_enabled_flag[ i ]가 존재하지 않을 때, subpic_ref_wraparound_enabled_flag[ i ]의 값은 all_subpic_ref_wraparound_enabled_flag와 동등한 것으로 추론된다.

[0150] 동일한 실시예에서, subpic_ref_wraparound_offset_minus1[ i ]가 존재하지 않을 때, subpic_ref_wraparound_offset_minus1[ i ]의 값은 all_subpic_ref_wraparound_offset_minus1과 동등한 것으로 추론된다.

[0151] 위에서 설명된 랩-어라운드 패딩 프로세스에 대한 기법들은 컴퓨터 판독가능 명령어들을 이용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 판독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 11은 개시된 주제의 특정 실시예들을 구현하기에 적합한 컴퓨터 시스템(800)을 도시한다.

[0152] 컴퓨터 소프트웨어는, 컴퓨터 중앙 처리 유닛(CPU)들, 그래픽 처리 유닛(GPU)들 등에 의해, 직접, 또는 해석, 마이크로-코드 실행 등을 통해 실행될 수 있는 명령어들을 포함하는 코드를 생성하기 위해 어셈블리, 컴파일, 링킹, 또는 유사한 메커니즘들이 수행될 수 있는 임의의 적합한 머신 코드 또는 컴퓨터 언어를 이용하여 코딩될 수 있다.

[0153] 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.

[0154] 컴퓨터 시스템(800)에 대한 도 11에 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(800)의 예시적인 실시예에서 예시된 컴포넌트들 중 임의의 하나 또는 이들의 조합과 관련하여 임의의 종속성 또는 요건을 갖는 것으로 해석되어서도 안 된다.

[0155] 컴퓨터 시스템(800)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 2차원 비디오, 입체적 비디오를 포함하는 3차원 비디오)와 같은, 인간에 의한 의식적인 입력과 반드시 직접적으로 관련되는 것은 아닌 특정 미디어를 캡처하기 위해 사용될 수 있다.

[0156] 입력 휴먼 인터페이스 디바이스들은: 키보드(801), 마우스(802), 트랙패드(803), 터치 스크린(810), 데이터-글러브(804), 조이스틱(805), 마이크로폰(806), 스캐너(807), 카메라(808) 중 하나 이상(각각의 하나만이 묘사됨)을 포함할 수 있다.

[0157] 컴퓨터 시스템(800)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(810), 데이터-글러브(804), 또는 조이스틱(805)에 의한 촉각 피드백이지만, 입력 디바이스들로서 역할을 하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(예컨대: 스피커들(809), 헤드폰들(묘사되지 않음)), 시각적 출력 디바이스들(예컨대 CRT 스크린들, LCD 스크린들, 플라즈마 스크린들, OLED

스크린들을 포함하는 스크린들(810) - 각각은 터치-스크린 입력 능력이 있거나 없고, 각각은 촉각 피드백 능력이 있거나 없고, 이들 중 일부는 스테레오그래픽 출력과 같은 수단을 통해 2차원 시각적 출력 또는 3개보다 많은 차원의 출력을 출력할 수 있음 -; 가상 현실 안경(묘사되지 않음), 홀로그래픽 디스플레이들 및 연기 탱크들(묘사되지 않음)), 및 프린터들(묘사되지 않음)을 포함할 수 있다.

[0158] 컴퓨터 시스템(800)은 인간 액세스 가능한 저장 디바이스들 및 그것들과 연관된 매체들, 예컨대 CD/DVD 등의 매체(821)를 갖는 CD/DVD ROM/RW(820)를 포함하는 광학 매체, 썸-드라이브(822), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(823), 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.

[0159] 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.

[0160] 컴퓨터 시스템(800)은 하나 이상의 통신 네트워크에 대한 인터페이스를 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(849)(예컨대, 예를 들어, 컴퓨터 시스템(800)의 USB 포트들)에 접속된 외부 네트워크 인터페이스 어댑터들을 요구한다; 다른 것들은 일반적으로 아래에 설명되는 바와 같은 시스템 버스로의 접속에 의해 컴퓨터 시스템(800)의 코어에 통합된다(예를 들어, PC 컴퓨터 시스템으로의 이더넷 인터페이스는 또는 스마트폰 컴퓨터 시스템으로의 셀룰러 네트워크 인터페이스). 이들 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(800)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 또는 광역 디지털 네트워크들을 사용하는 다른 컴퓨터 시스템들과의 양방향성일 수 있다. 위에 설명된 바와 같은 해당 네트워크들 및 네트워크 인터페이스들 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0161] 전술한 휴먼 인터페이스 디바이스들, 인간-액세스 가능한 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(800)의 코어(840)에 접속될 수 있다.

[0162] 코어(840)는 하나 이상의 중앙 처리 유닛(CPU)(841), 그래픽 처리 유닛(GPU)(842), 필드 프로그래머블 게이트 영역(FPGA)(843)의 형식으로 특수화된 프로그래머블 처리 유닛들, 특정 태스크들(844)에 대한 하드웨어 가속기들 등을 포함할 수 있다. 이들 디바이스는, 판독 전용 메모리(ROM)(845), 랜덤 액세스 메모리(846), 내부 비-사용자 액세스 가능 하드 드라이브들, SSD들 등과 같은 내부 대용량 저장소(847)와 함께, 시스템 버스(848)를 통해 연결될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(848)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형식으로 액세스 가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(848)에 직접, 또는 주변 버스(849)를 통해 접속될 수 있다. 주변 버스를 위한 아키텍처들은 PCI, USB 등을 포함한다.

[0163] CPU들(841), GPU들(842), FPGA들(843), 및 가속기들(844)은, 조합하여, 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령어들을 실행할 수 있다. 해당 컴퓨터 코드는 ROM(845) 또는 RAM(846)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(846)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 저장소(847)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은, 하나 이상의 CPU(841), GPU(842), 대용량 저장소(847), ROM(845), RAM(846) 등과 밀접하게 연관될 수 있는, 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

[0164] 컴퓨터 판독가능 매체는 다양한 컴퓨터 구현 동작들을 수행하기 위한 컴퓨터 코드를 그 위에 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 그들이 이용할 수 있는 종류의 것일 수 있다.

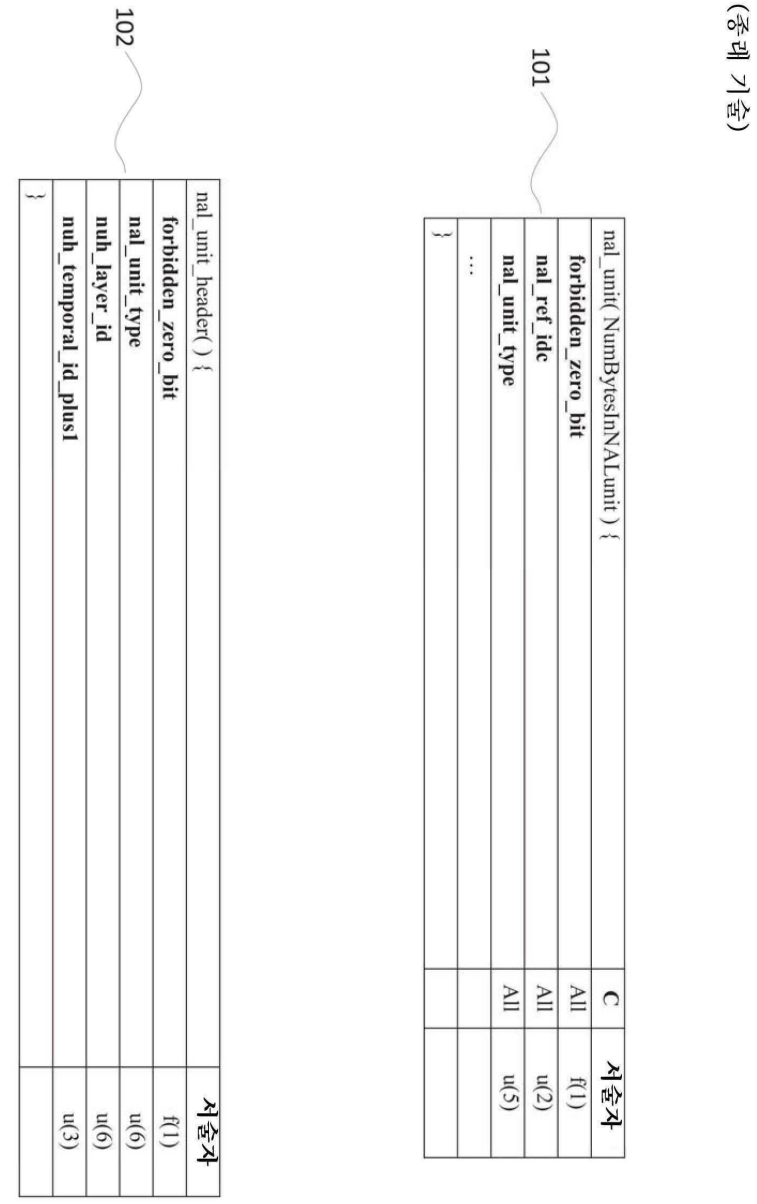
[0165] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(800), 및 구체적으로 코어(840)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형(tangible) 컴퓨터 판독가능 매체에 구현된 소프트웨어를 실행하는 결과로서 기능성을 제공할 수 있다. 그러한 컴퓨터 판독가능 매체는 위에 소개된 바와 같은 사용자-액세스 가능한 대용량 저장소뿐만 아니라, 코어-내부 대용량 저장소(847) 또는 ROM(845)과 같은 비-일시적인

성질의 것인 코어(840)의 특정 저장소와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 그러한 디바이스들에 저장되고 코어(840)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 요구에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(840) 및 구체적으로 그 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(846)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 소프트웨어와 함께 동작할 수 있는, 회로(예를 들어: 가속기(844))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.

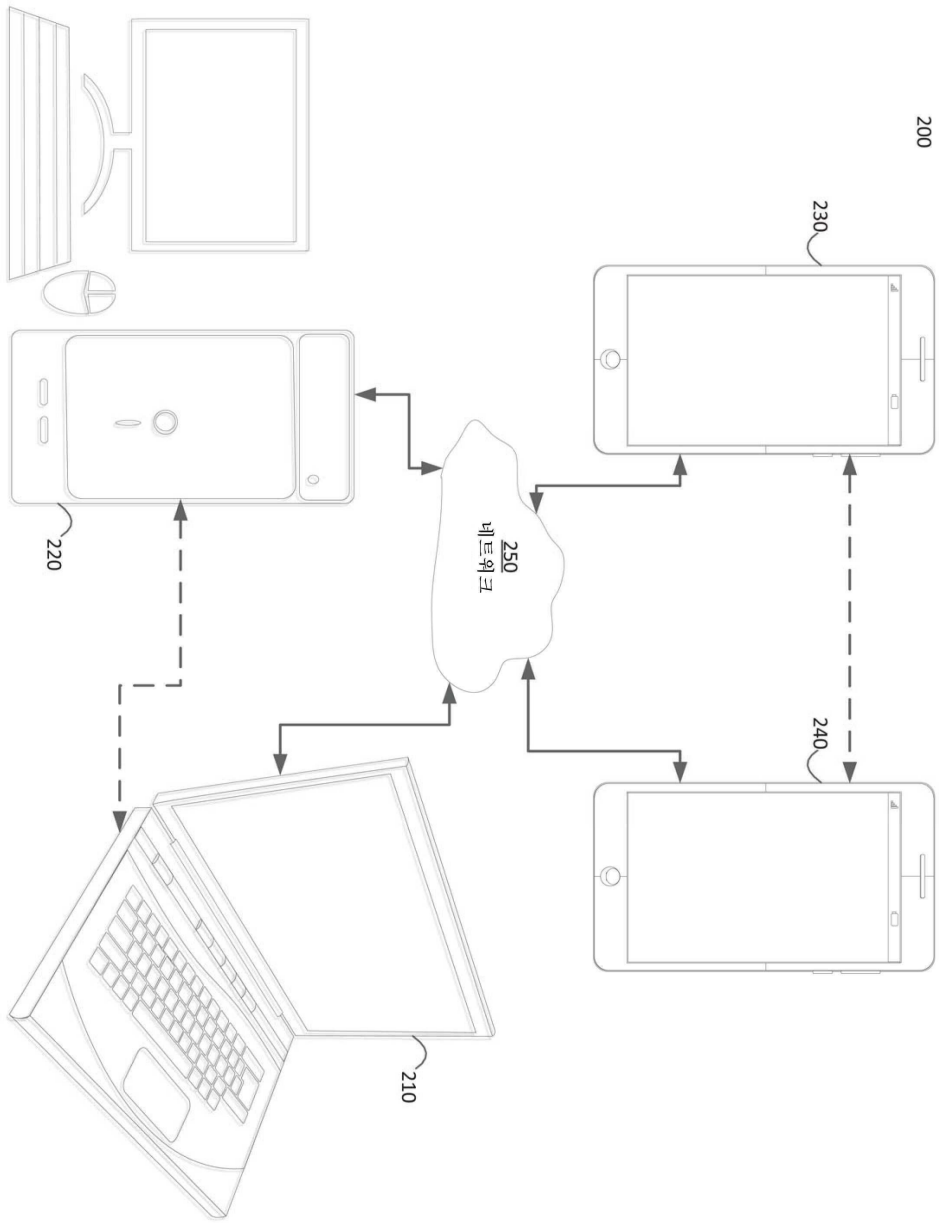
[0166] 본 개시내용이 여러 예시적인 실시예들을 설명하였지만, 본 개시내용의 범위 내에 속하는 변경들, 치환들, 및 다양한 대체 균등물들이 존재한다. 따라서, 본 기술분야의 통상의 기술자들은, 비록 본 명세서에 명시적으로 도시되거나 설명되지는 않았지만, 본 개시내용의 원리들을 구현하고 따라서 그것의 진의 및 범위 내에 있는, 다수의 시스템들 및 방법들을 고안할 수 있을 것이라는 점이 인정될 것이다.

도면

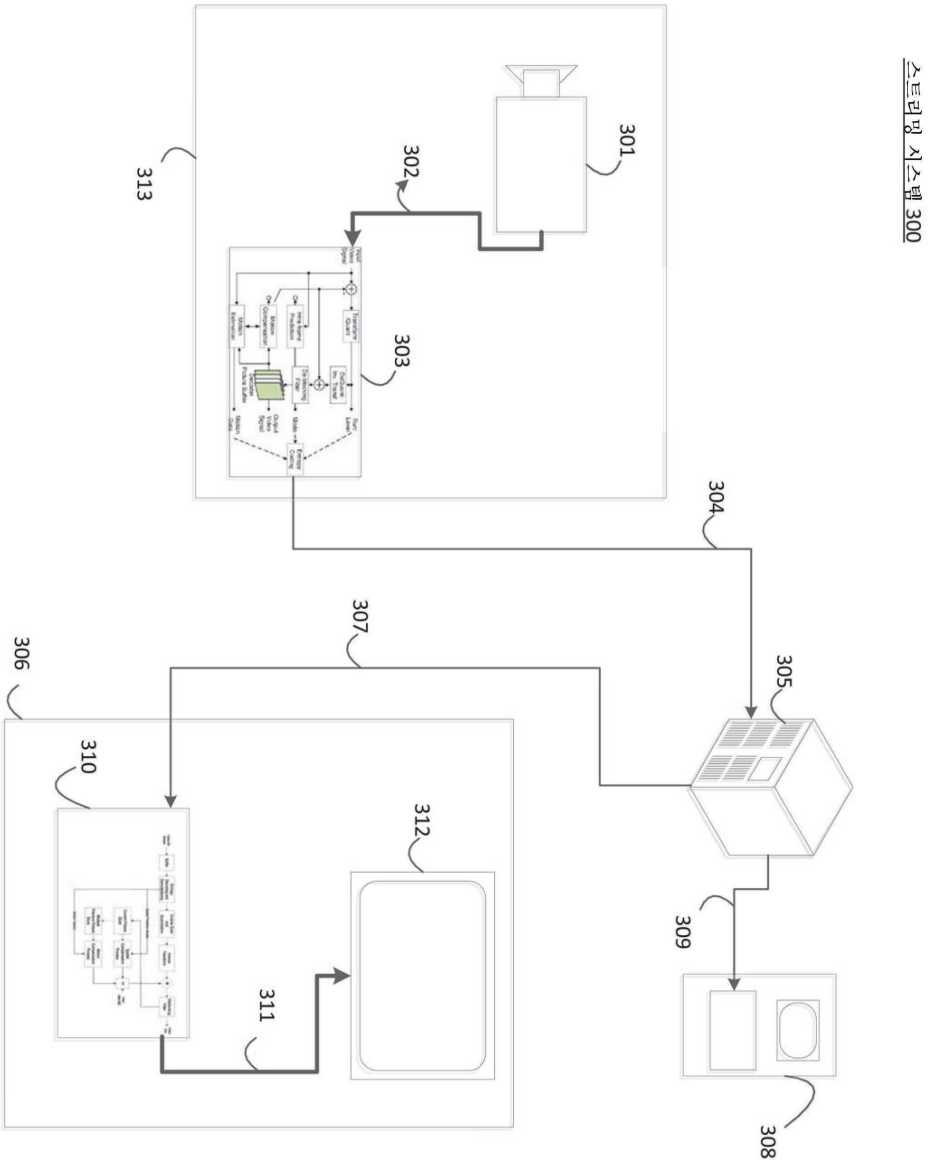
도면1


(종래 기술)
101
nal_unit( NumBytesInNALunit ) {
forbidden_zero_bit
nal_ref_idc
nal_unit_type
...
}
C
All
All
All
서술자
f(1)
u(2)
u(5)
102
nal_unit_header( ) {
forbidden_zero_bit
nal_unit_type
nuh_layer_id
nuh_temporal_id_plus1
}
서술자
f(1)
u(6)
u(6)
u(3)

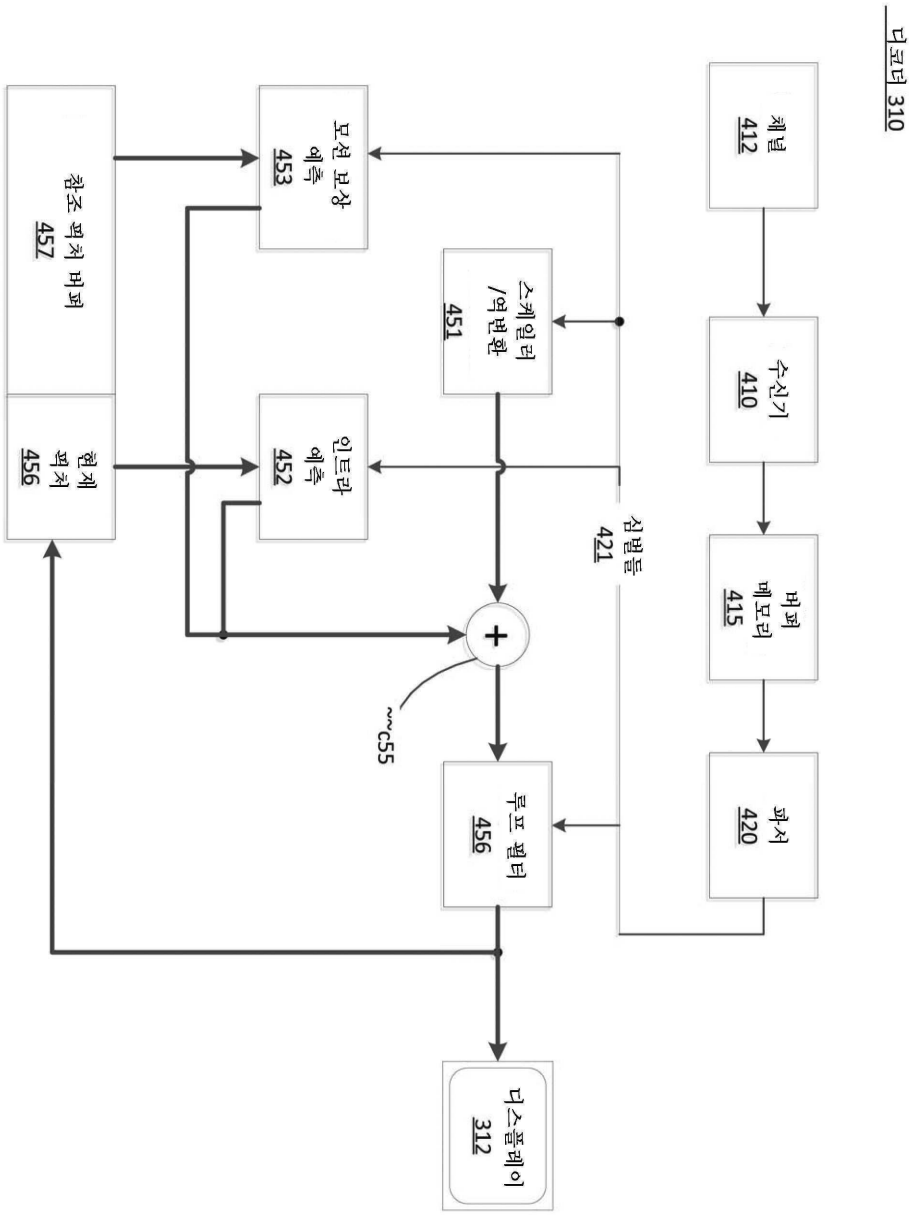
도면2



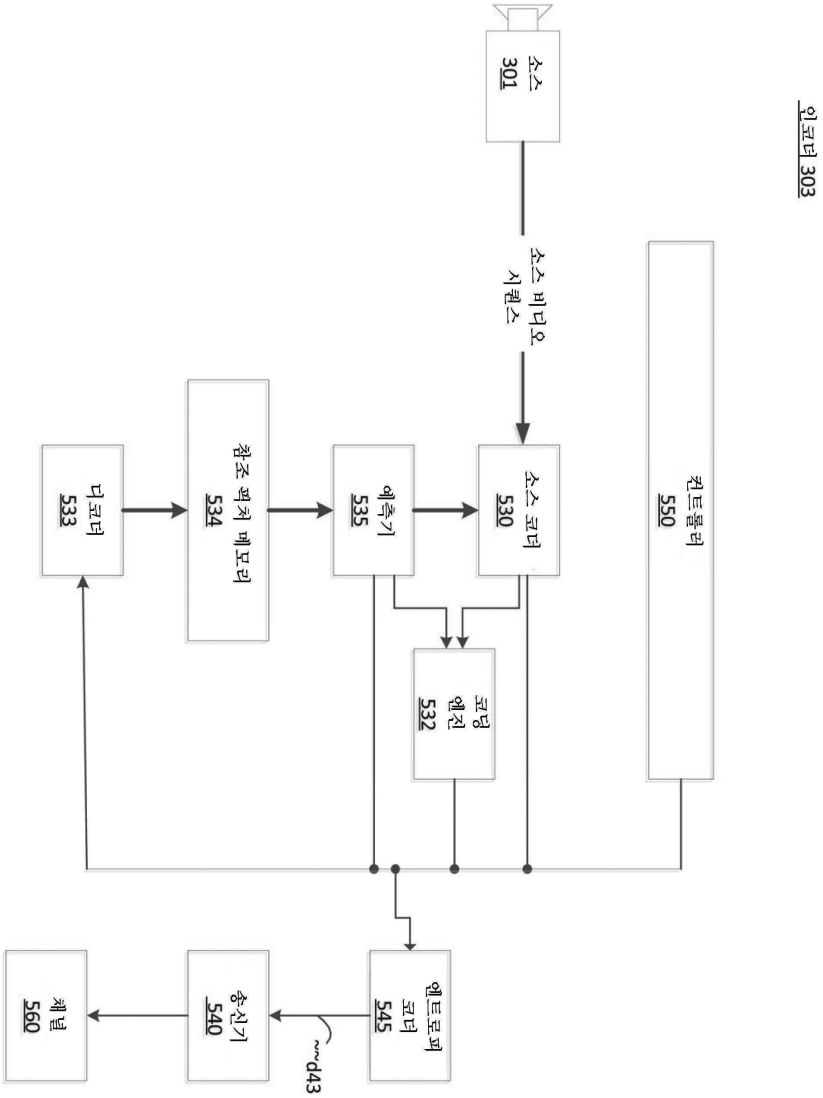
도면3



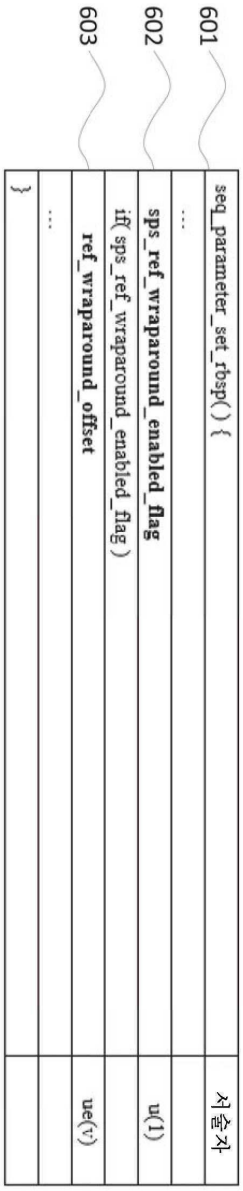
도면4


디코더 310
채널 412
수신기 410
버퍼 메모리 415
파서 420
심벌들 421
스케일러 /역변환 451
모션 보상 예측 453
참조 픽처 버퍼 457
현재 픽처 456
인트라 예측 452
C55
루프 필터 456
디스플레이 312

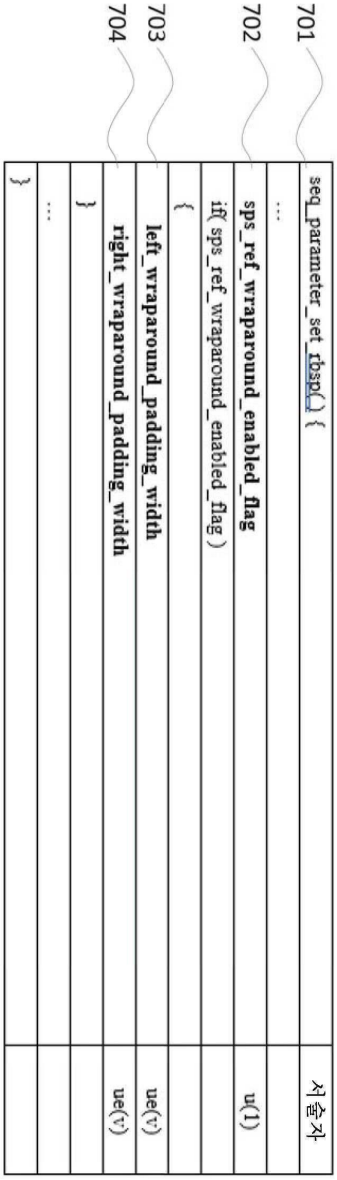
도면5


인코더 303
소스
301
소스 비디오 시퀀스
소스 코더
530
컨트롤러
550
예측기
535
참조 픽처 메모리
534
디코더
533
코딩 엔진
532
엔트로피 코더
545
송신기
540
채널
560
~d43

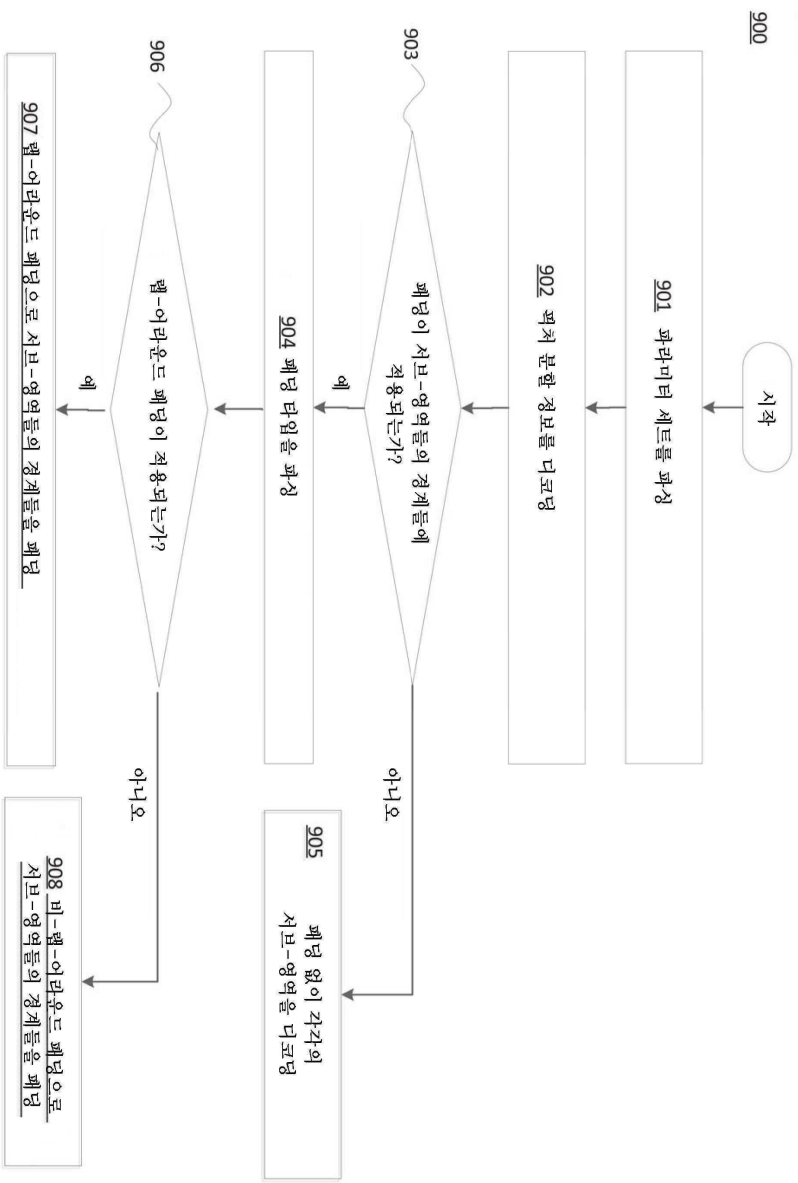
도면6


601
602
603
seq_parameter_set_rbsp( ) {
...
sps_ref_wraparound_enabled_flag
if( sps_ref_wraparound_enabled_flag )
ref_wraparound_offset
...
}
서술자
u(1)
ue(v)

도면7


701
702
703
704
seq_parameter_set_rbsp( ) {
...
sps_ref_wraparound_enabled_flag
if( sps_ref_wraparound_enabled_flag )
{
left_wraparound_padding_width
right_wraparound_padding_width
}
...
}
서술자
u(1)
ue(v)
ue(v)

도면8


900
시작
901 파라미터 세트를 파싱
902 픽처 분할 정보를 디코딩
903
패딩이 서브-영역들의 경계들에 적용되는가?
예
904 패딩 타입을 파싱
906
랩-어라운드 패딩이 적용되는가?
예
907 랩-어라운드 패딩으로 서브-영역들의 경계들을 패딩
아니오
908 비-랩-어라운드 패딩으로 서브-영역들의 경계들을 패딩
아니오
905 패딩 없이 각각의 서브-영역을 디코딩

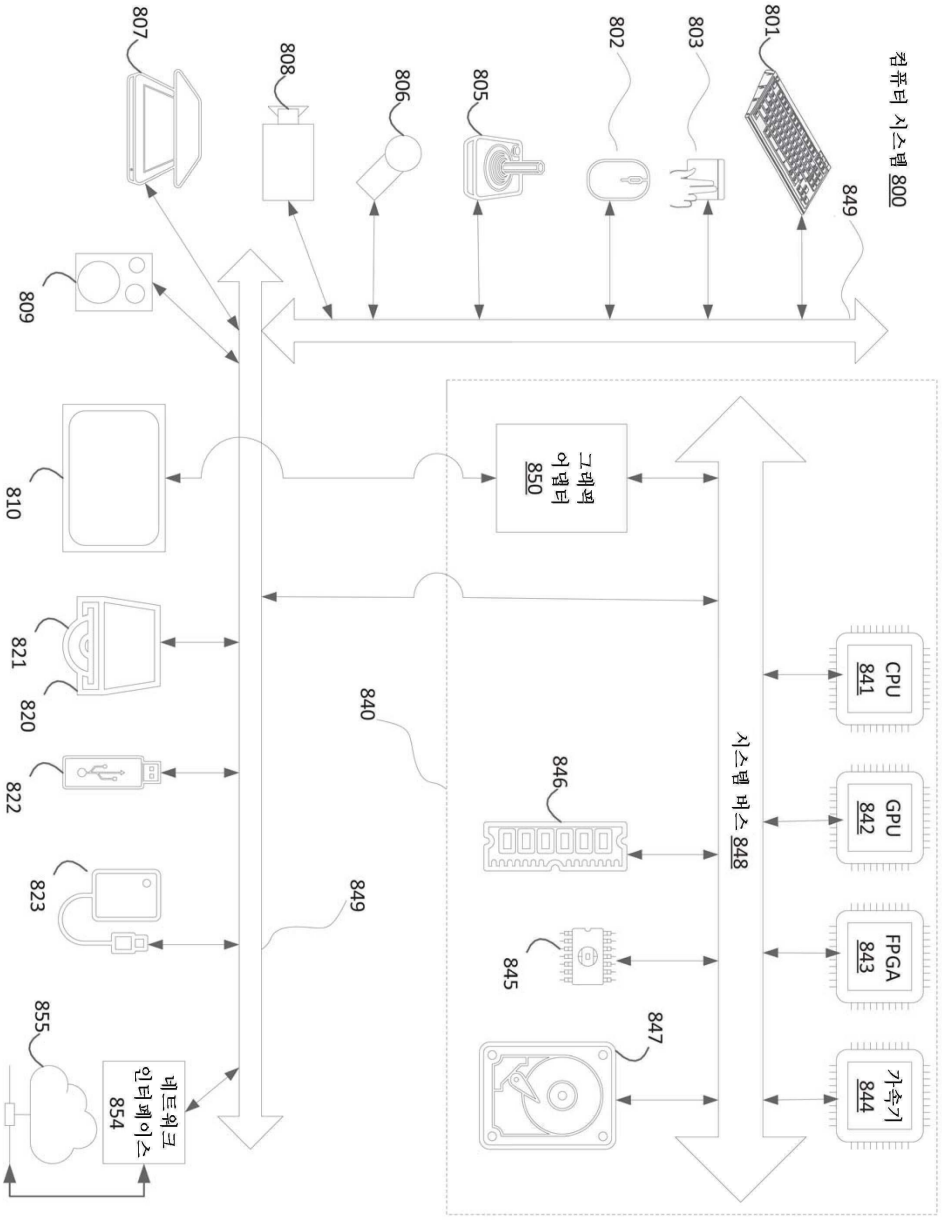
도면9

| seq_parameter_set_rbsp() { | | 서술자 |
|---|---|---|
| … | | |
| **sps_ref_wraparound_enabled_flag** | | u(1) |
| if( sps_ref_wraparound_enabled_flag ) | | |
| **sps_ref_wraparound_offset_minus1** | | ue(v) |
| **subpics_present_flag** | | u(1) |
| if( subpics_present_flag ) { | | |
| **max_subpics_minus1** | | u(8) |
| **subpic_grid_col_width_minus1** | | u(v) |
| **subpic_grid_row_height_minus1** | | u(v) |
| for( i = 0; i < NumSubPicGridRows; i++ ) | | |
| for( j = 0; j < NumSubPicGridCols; j++ ) | | |
| **subpic_grid_idx**[ i ][ j ] | | u(v) |
| for( i = 0; i <= NumSubPics; i++ ) { | | |
| **subpic_treated_as_pic_flag**[ i ] | | u(1) |
| if( subpic_treated_as_pic_flag[ i ] ) { | | |
| **subpic_ref_wraparound_enabled_flag**[ i ] | | u(1) |
| if( subpic_ref_wraparound_enabled_flag[ i ] ) | | |
| **subpic_ref_wraparound_offset_minus1**[ i ] | | u(1) |
| } | | |
| **loop_filter_across_subpic_enabled_flag**[ i ] | | u(1) |
| } | | |
| } | | |
| … | | |
| } | | |

도면10

| | 서술자 |
|---|---|
| seq_parameter_set_rbsp( ) { | |
| ... | |
| **sps_ref_wraparound_enabled_flag** | u(1) |
| if( sps_ref_wraparound_enabled_flag ) | |
| **sps_ref_wraparound_offset_minus1** | ue(v) |
| **subpics_present_flag** | u(1) |
| if( subpics_present_flag ) { | |
| **max_subpics_minus1** | u(8) |
| **subpic_grid_col_width_minus1** | u(v) |
| **subpic_grid_row_height_minus1** | u(v) |
| for( i = 0; i < NumSubPicGridRows; i++ ) | |
| for( j = 0; j < NumSubPicGridCols; j++ ) | |
| **subpic_grid_idx**[ i ][ j ] | u(v) |
| **all_sub_pic_treated_as_pic_flag** | u(1) |
| if( all_sub_pic_treated_as_pic_flag ) { | |
| **all_subpic_ref_wraparound_enabled_flag**[ i ] | u(1) |
| if( all_subpic_ref_wraparound_enabled_flag[ i ] ) | |
| **all_subpic_ref_wraparound_offset_minus1**[ i ] | ue(v) |
| } | |
| for( i = 0; i <= NumSubPics; i++ ) { | |
| if( !all_sub_pic_treated_as_pic_flag ) { | |
| **subpic_treated_as_pic_flag**[ i ] | u(1) |
| if( subpic_treated_as_pic_flag[ i ] ) { | |
| **subpic_ref_wraparound_enabled_flag**[ i ] | u(1) |
| if( subpic_ref_wraparound_enabled_flag[ i ] ) | |
| **subpic_ref_wraparound_offset_minus1**[ i ] | ue(v) |
| } | |
| } | |
| **loop_filter_across_subpic_enabled_flag**[ i ] | u(1) |
| } | |
| } | |
| ... | |
| } | |

도면11


컴퓨터 시스템 800
801
802
803
805
806
807
808
809
810
820
821
822
823
840
845
846
847
849
849
850
841
842
843
844
848
854
855