



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2024년12월27일  
(11) 등록번호 10-2746924  
(24) 등록일자 2024년12월20일

- (51) 국제특허분류(Int. Cl.)  
H04N 19/563 (2014.01) H04N 19/172 (2014.01)  
H04N 19/192 (2014.01) H04N 19/597 (2014.01)  
H04N 19/70 (2014.01)
- (52) CPC특허분류  
H04N 19/563 (2015.01)  
H04N 19/172 (2015.01)
- (21) 출원번호 10-2024-7011356(분할)
- (22) 출원일자(국제) 2019년12월27일  
심사청구일자 2024년05월07일
- (85) 번역문제출일자 2024년04월04일
- (65) 공개번호 10-2024-0051290
- (43) 공개일자 2024년04월19일
- (62) 원출원 특허 10-2021-7016175  
원출원일자(국제) 2019년12월27일  
심사청구일자 2021년05월27일
- (86) 국제출원번호 PCT/US2019/068678
- (87) 국제공개번호 WO 2020/142360  
국제공개일자 2020년07월09일
- (30) 우선권주장  
62/787,063 2018년12월31일 미국(US)  
16/710,936 2019년12월11일 미국(US)
- (56) 선행기술조사문헌  
KR1020130070646 A  
Jianle Chen et al., Algorithm description for Versatile Video Coding and Test Model 3 (VTM 3), JVET of ITU-T and ISO/IEC, JVET-L1002-v2(2018.12.24.)  
Philippe Hanhart et al., CE13: PERP with horizontal geometry padding of reference pictures (Test 3.3), JVET of ITU-T and ISO/IEC, JVET-L0231-v4(2018.10.11.)

- (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  
평, 웨이웨이  
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 12 항

심사관 : 김성권

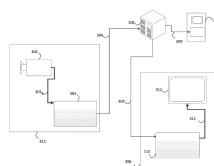
(54) 발명의 명칭 **전방향 미디어 코딩을 위한 랩-어라운드 패딩 방법, 디바이스, 및 컴퓨터 판독가능 매체**

**(57) 요약**

비디오 디코딩을 위해 코딩된 현재 픽처를 재구성하는 것은, 현재 픽처에 대응하는 픽처 파티셔닝 정보를 디코딩하는 것; 픽처 파티셔닝 정보를 사용하여 현재 픽처의 서브-영역들에 패딩이 적용되는지를 결정하는 것; 패딩이 적용되지 않는다는 결정에 기초하여, 복수의 서브-영역을 패딩하지 않고 서브-영역들을 디코딩하는 것; 패딩이

(뒷면에 계속)

**대표도**



적용된다는 결정에 기초하여, 픽처 파티셔닝 정보를 사용하여 패딩이 랩-어라운드 패딩을 포함하는지를 결정하는 것; 패딩이 랩-어라운드 패딩을 포함하지 않는다는 결정에 기초하여, 서브-영역들에 반복 패딩을 적용하고, 반복 패딩을 사용하여 서브-영역들을 디코딩하는 것; 패딩이 랩-어라운드 패딩을 포함한다는 결정에 기초하여, 랩-어라운드 패딩을 서브-영역들에 적용하고, 랩-어라운드 패딩을 사용하여 서브-영역들을 디코딩하는 것; 및 디코딩된 서브-영역들에 기초하여 현재 픽처를 재구성하는 것을 포함한다.

(52) CPC특허분류

*H04N 19/192* (2015.01)

*H04N 19/597* (2015.01)

*H04N 19/70* (2015.01)

## 명세서

### 청구범위

#### 청구항 1

비디오 코딩을 위해 현재 픽처를 인코더에 의해 인코딩하는 방법으로서,

상기 현재 픽처의 복수의 서브-영역에 패딩이 적용되는지를 나타내는 제1 결정을 하는 단계;

패딩이 적용된다는 것을 나타내는 상기 제1 결정에 기초하여, 상기 패딩이 랩-어라운드 패딩을 포함하는지를 나타내는 제2 결정을 하는 단계;

상기 패딩이 상기 랩-어라운드 패딩을 포함한다는 것을 나타내는 상기 제2 결정에 기초하여, 상기 랩-어라운드 패딩에 기초하여 상기 복수의 서브-영역을 인코딩하는 단계;

상기 제1 결정 및 상기 제2 결정에 기초하여 픽처 파티셔닝 정보를 생성하는 단계 - 상기 픽처 파티셔닝 정보는 상기 현재 픽처의 현재 블록에 대응하는 참조 블록을 선택하는 데 사용되는 랩-어라운드 위치를 계산하는 데 사용되는 루마 샘플들에서의 좌측 랩-어라운드 패딩 폭 및 루마 샘플들에서의 우측 랩-어라운드 패딩 폭을 직접 지정함 -; 및

상기 인코딩된 복수의 서브-영역 및 상기 픽처 파티셔닝 정보에 기초하여 상기 현재 픽처를 인코딩하는 단계를 포함하고,

상기 랩-어라운드 위치를 계산하는 데 사용되는 랩-어라운드 오프셋 값은:

(i) 상기 픽처 파티셔닝 정보에 오프셋 값이 존재하는 경우, 상기 픽처 파티셔닝 정보로부터의 상기 오프셋 값으로 지정되고,

(ii) 상기 조건 (i)이 충족되지 않고 루마 샘플들에서의 좌측 및 우측 패딩 영역의 폭들이 존재하는 경우, {루마 샘플들에서의 폭 - (루마 샘플들에서의 좌측 패딩 영역의 폭 + 루마 샘플들에서의 우측 패딩 영역의 폭)}의 값으로 지정되고,

(iii) 상기 조건 (i) 및 상기 조건 (ii) 중 어느 것도 충족되지 않는 경우, 루마 샘플들에서의 폭의 값으로 지정되는 방법.

#### 청구항 2

제1항에 있어서, 상기 픽처 파티셔닝 정보는 상기 현재 픽처에 대응하는 픽처 파라미터 세트에 포함되는 방법.

#### 청구항 3

제2항에 있어서, 상기 픽처 파티셔닝 정보는 상기 픽처 파라미터 세트에 포함된 적어도 하나의 플래그를 포함하는 방법.

#### 청구항 4

제1항에 있어서, 상기 복수의 서브-영역은 브릭, 타일, 슬라이스, 타일 그룹, 서브-픽처, 또는 서브-계층 중에서 적어도 하나를 포함하는 방법.

#### 청구항 5

제1항에 있어서, 상기 패딩은 복수의 서브-영역 중의 서브-영역의 경계에 적용되는 방법.

#### 청구항 6

제5항에 있어서, 상기 경계는 상기 서브-영역의 수직 경계인 방법.

#### 청구항 7

제5항에 있어서, 상기 경계는 상기 서브-영역의 수평 경계인 방법.

**청구항 8**

제1항에 있어서, 상기 패딩은 복수의 서브-영역 중의 서브-영역의 수직 경계 및 상기 서브-영역의 수평 경계에 적용되는 방법.

**청구항 9**

제1항에 있어서, 상기 패딩이 랩-어라운드 패딩을 포함하지 않는다는 것을 나타내는 상기 제2 결정에 기초하여, 상기 복수의 서브-영역에 반복 패딩을 적용하고, 상기 반복 패딩을 사용하여 상기 복수의 서브-영역을 인코딩하는 단계를 더 포함하는 방법.

**청구항 10**

제1항에 있어서, 상기 픽처 파티셔닝 정보는 좌측 패딩 폭 정보 및 우측 패딩 폭 정보를 나타내는 방법.

**청구항 11**

비디오 코딩을 위해 현재 픽처를 인코딩하기 위한 디바이스로서, 프로그램 코드를 저장하도록 구성되는 적어도 하나의 메모리; 및 상기 프로그램 코드를 판독하고 상기 프로그램 코드에 의해 지시된 대로 동작하도록 구성되는 적어도 하나의 프로세서

를 포함하고, 상기 프로그램 코드는 제1항 내지 제10항 중 어느 한 항에 따른 방법을 실행하기 위한 명령어들을 포함하는 디바이스.

**청구항 12**

명령어들이 저장된 비일시적 컴퓨터 판독가능 매체로서, 상기 명령어들은, 비디오 코딩을 위해 현재 픽처를 인코딩하기 위한 디바이스의 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서로 하여금, 제1항 내지 제10항 중 어느 한 항에 따른 방법을 수행하게 하는 하나 이상의 명령어를 포함하는 비일시적 컴퓨터 판독가능 매체.

**발명의 설명**

**기술 분야**

[0001] 본 출원은 35 U.S.C. § 119로부터 미국 특허 상표청에 2018년 12월 31일자로 출원된 미국 가출원 제62/787,063호, 및 미국 특허 상표청에 2019년 12월 11일자로 출원된 미국 출원 제16/710,936호에 대한 우선권을 주장하며, 그 개시내용들은 그 전체가 본 명세서에 참고로 포함된다.

[0002] 개시된 주제는 비디오 코딩 및 디코딩에 관한 것으로, 특히, 360 전방향 미디어 코딩을 위한 랩-어라운드 패딩 처리의 포함에 관한 것이다.

**배경 기술**

[0003] 모션 보상(motion compensation)을 갖는 인터-픽처 예측(inter-picture prediction)을 사용하는 비디오 코딩 및 디코딩의 예들은 수십년 동안 알려져 왔다. 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 조건들을 갖는다. 예를 들어, 샘플당 8 비트에서의 1080p60 4:2:0 비디오(60 Hz 프레임 레이트에서의 1920x1080 루미넌스 샘플 해상도)는 1.5 Gbit/s에 근접한 대역폭을 요구한다. 한 시간의 그러한 비디오는 600 GByte보다 많은 저장 공간을 요구한다.

[0004] 비디오 코딩 및 디코딩의 하나의 목적은, 압축을 통한, 입력 비디오 신호에서의 중복성(redundancy)의 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 사용할 때, 재구성된 신호는 원래 신호와 동일하지 않을 수 있지만, 원래 신호와 재구성된 신호 사이의 왜곡은 재구성된 신호를 의도된 애플리케이션에 유용하게 만들 정도로 충분히 작다. 비디오의 경우, 손실 압축이 널리 이용된다. 용인되는 왜곡의 양은 애플리케이션에 의존하는데; 예를 들어, 특정 소비자 스트리밍 애플리케이션들의 사용자들은 텔레비전 기어 애플리케이션들의 사용자들보다 더 높은 왜곡을 용인할 수 있다. 달성가능한 압축비는: 더 높은 허용가능한/용인가능한 왜곡이 더 높은 압축비를 산출할 수 있다는 것을 반영할 수 있다.

[0005] 비디오 인코더 및 디코더는, 예를 들어, 모션 보상, 변환, 양자화, 및 엔트로피 코딩을 포함한, 몇가지 광범위한 카테고리로부터의 기법들을 활용할 수 있으며, 이들 중 일부는 아래에서 소개될 것이다.

[0006] 코딩된 비디오 비트스트림을 패킷 네트워크들을 통한 수송을 위한 패킷들로 분할하는 예들은 수십년 동안 사용되어 왔다. 초기에, 비디오 코딩 표준들 및 기술들은 그들의 대부분이 보트-지향성 수송(bot-oriented transport) 및 정의된 비트스트림들에 최적화되었다. 패킷화는 예를 들어 RTP(Real-time Transport Protocol) 페이로드 포맷들로 지정된 시스템 계층 인터페이스들에서 발생하였다. 인터넷을 통한 비디오의 대량-사용에 적합한 인터넷 접속의 출현으로, 비디오 코딩 표준들은 VCL(video coding layer) 및 NAL(network abstraction layer)의 개념적 차별화를 통해 현저한 사용 사례를 반영하였다. NAL 유닛들은 2003년 H.264에 도입되었고, 그 이후로 약간의 수정만으로 특정 비디오 코딩 표준들 및 기술들에서 유지되었다.

[0007] NAL 유닛은, 많은 경우에, 코딩된 비디오 시퀀스의 모든 선행 NAL 유닛을 반드시 디코딩하지 않고도 디코더가 작용할 수 있는 가장 작은 엔티티로 볼 수 있다. 지금까지, NAL 유닛들은 SFU(Selective Forwarding Unit)들 또는 MCU(Multipoint Control Unit)들과 같은 MANE(Media Aware Network Element)들에 의해, 비트스트림 프루닝을 포함하는, 특정 비트스트림 조작 기법들뿐만 아니라 특정 에러 복원 기술들을 가능하게 한다.

[0008] 도 1은, 두 경우 모두 그들 각자의 어떠한 확장도 없는, H.264(101) 및 H.265(102)에 따른 NAL 유닛 헤더들의 선택스 다이어그램의 관련 부분들을 묘사한다. 두 경우 모두에서, forbidden\_zero\_bit는 특정 시스템 계층 환경들에서 시작 코드 에물레이션 방식을 위해 사용되는 제로 비트이다. nal\_unit\_type 선택스 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 공간적/SNR 및 시간적 계층을 나타내는 nuh\_layer\_id 및 nuh\_temporal\_id\_plus1을 추가로 포함한다.

[0009] NAL 유닛 헤더는, 예를 들어, 다른 NAL 유닛 헤더들, 파라미터 세트들 등과 같은 비트스트림 내의 다른 데이터에 대한 어떠한 파싱 의존성도 갖지 않는, 용이하게 파싱가능한 고정 길이 코드워드들만을 포함한다는 것을 알 수 있다. NAL 유닛 헤더들은 NAL 유닛 내의 첫 번째 옥텟들이기 때문에, MANE들은 이들을 쉽게 추출하고, 이들을 파싱하며, 이들에 작용할 수 있다. 이와 달리, 다른 고레벨 선택스 요소들, 예를 들어 슬라이스 또는 타일 헤더들은 파라미터 세트 컨텍스트 유지 및/또는 가변 길이 또는 산술 코딩된 코드포인트들의 처리를 요구할 수 있으므로 MANE들에 쉽게 액세스할 수 없다.

[0010] 또한, 도 1에 도시된 바와 같은 NAL 유닛 헤더들은 NAL 유닛을 복수의 NAL 유닛(이를테면, 예를 들어, 다수의 타일 또는 슬라이스를 포함하고, 이들 중 적어도 일부는 개별 NAL 유닛들로 패킷화됨)으로 구성된 코딩된 픽처에 연관시킬 수 있는 정보를 포함하지 않는다는 것을 알 수 있다.

[0011] RTP(RFC 3550), MPEG-시스템 표준들, ISO 파일 포맷들 등과 같은 특정 수송 기술들은 종종 MANE에 의해 쉽게 액세스될 수 있고 그들의 각자의 수송 유닛들을 코딩된 픽처들과 연관시키는 데 도움을 줄 수 있는 제시 시간(MPEG 및 ISO 파일 포맷의 경우에) 또는 캡처 시간(RTP의 경우에)과 같은 타이밍 정보의 형태로 되어 있는 특정의 정보를 포함할 수 있다. 그러나, 이러한 정보의 시맨틱은 수송/저장 기술마다 다를 수 있으며, 비디오 코딩에서 사용되는 픽처 구조와 직접적인 관계를 갖지 않을 수 있다. 따라서, 이들 정보는 기껏해야 휴리스틱일 수 있고, 또한 NAL 유닛 스트림 내의 NAL 유닛들이 동일한 코딩된 픽처에 속하는지 여부를 식별하는 데 특히 적합하지 않을 수 있다.

**발명의 내용**

[0012] 일 실시예에서, 적어도 하나의 프로세서를 사용하여 비디오 디코딩을 위해 코딩된 현재 픽처를 재구성하는 방법이 제공되고, 이러한 방법은 현재 픽처에 대응하는 픽처 파티셔닝 정보를 디코딩하는 단계; 픽처 파티셔닝 정보

를 사용하여 현재 픽처의 복수의 서브-영역에 패딩이 적용되는지를 결정하는 단계; 패딩이 적용되지 않는다는 결정에 기초하여, 복수의 서브-영역을 패딩하지 않고 복수의 서브-영역을 디코딩하는 단계; 패딩이 적용된다는 결정에 기초하여, 픽처 파티셔닝 정보를 사용하여 패딩이 랩-어라운드 패딩을 포함하는지를 결정하는 단계; 패딩이 랩-어라운드 패딩을 포함하지 않는다는 결정에 기초하여, 복수의 서브-영역에 반복 패딩을 적용하고, 반복 패딩을 사용하여 복수의 서브-영역을 디코딩하는 단계; 패딩이 랩-어라운드 패딩을 포함한다는 결정에 기초하여, 랩-어라운드 패딩을 복수의 서브-영역에 적용하고, 랩-어라운드 패딩을 사용하여 복수의 서브-영역을 디코딩하는 단계; 및 디코딩된 복수의 서브-영역에 기초하여 현재 픽처를 재구성하는 단계를 포함한다.

[0013] 일 실시예에서, 비디오 디코딩을 위해 코딩된 현재 픽처를 재구성하기 위한 디바이스가 제공되는데, 이 디바이스는 프로그램 코드를 저장하도록 구성되는 적어도 하나의 메모리; 및 프로그램 코드를 판독하고, 프로그램 코드에 의해 지시되는 바와 같이 동작하도록 구성되는 적어도 하나의 프로세서를 포함하며, 프로그램 코드는 적어도 하나의 프로세서로 하여금 현재 픽처에 대응하는 픽처 파티셔닝 정보를 디코딩하게 하도록 구성되는 제1 디코딩 코드; 적어도 하나의 프로세서로 하여금 픽처 파티셔닝 정보를 사용하여 현재 픽처의 복수의 서브-영역에 패딩이 적용되는지를 결정하게 하도록 구성되는 제1 결정 코드; 적어도 하나의 프로세서로 하여금, 패딩이 적용되지 않는다는 결정에 기초하여, 복수의 서브-영역을 패딩하지 않고 복수의 서브-영역을 디코딩하게 하도록 구성되는 제2 디코딩 코드; 패딩이 적용된다는 결정에 기초하여, 픽처 파티셔닝 정보를 사용하여 패딩이 랩-어라운드 패딩을 포함하는지를 결정하도록 구성되는 제2 결정 코드; 적어도 하나의 프로세서로 하여금, 패딩이 랩-어라운드 패딩을 포함하지 않는다는 결정에 기초하여, 복수의 서브-영역에 반복 패딩을 적용하게 하고, 반복 패딩을 사용하여 복수의 서브-영역을 디코딩하게 하도록 구성되는 제1 반복 코드; 적어도 하나의 프로세서로 하여금, 패딩이 랩-어라운드 패딩을 포함한다는 결정에 기초하여, 랩-어라운드 패딩을 복수의 서브-영역에 적용하게 하고, 랩-어라운드 패딩을 사용하여 복수의 서브-영역을 디코딩하게 하도록 구성되는 제2 반복 코드; 및 적어도 하나의 프로세서로 하여금 디코딩된 복수의 서브-영역에 기초하여 현재 픽처를 재구성하게 하도록 구성되는 재구성 코드를 포함한다.

[0014] 일 실시예에서, 명령어들을 저장하는 비일시적 컴퓨터 판독가능 매체가 제공되며, 이러한 명령어들은, 비디오 디코딩을 위해 코딩된 현재 픽처를 재구성하기 위한 디바이스의 하나 이상의 프로세서에 의해 실행될 때, 하나 이상의 프로세서로 하여금 현재 픽처에 대응하는 픽처 파티셔닝 정보를 디코딩하게 하고; 픽처 파티셔닝 정보를 사용하여 현재 픽처의 복수의 서브-영역에 패딩이 적용되는지 결정하게 하고; 패딩이 적용되지 않는다는 결정에 기초하여, 복수의 서브-영역을 패딩하지 않고 복수의 서브-영역을 디코딩하게 하고; 패딩이 적용된다는 결정에 기초하여, 픽처 파티셔닝 정보를 사용하여 패딩이 랩-어라운드 패딩을 포함하는지를 결정하게 하고; 패딩이 랩-어라운드 패딩을 포함하지 않는다는 결정에 기초하여, 복수의 서브-영역에 반복 패딩을 적용하게 하고, 반복 패딩을 사용하여 복수의 서브-영역을 디코딩하게 하고; 패딩이 랩-어라운드 패딩을 포함한다는 결정에 기초하여, 랩-어라운드 패딩을 복수의 서브-영역에 적용하게 하고, 랩-어라운드 패딩을 사용하여 복수의 서브-영역을 디코딩하게 하고; 디코딩된 복수의 서브-영역에 기초하여 현재 픽처를 재구성하게 하는 하나 이상의 명령어를 포함한다.

**도면의 간단한 설명**

[0015] 개시된 주제의 추가의 특징들, 본질 및 다양한 이점들이 다음의 상세한 설명 및 첨부 도면들로부터 더 명백할 것이다.

- 도 1은 H.264 및 H.265에 따른 NAL 유닛 헤더들의 개략적인 예시이다.
- 도 2는 일 실시예에 따른 통신 시스템의 단순화된 블록 다이어그램의 개략적인 예시이다.
- 도 3은 일 실시예에 따른 통신 시스템의 단순화된 블록 다이어그램의 개략적인 예시이다.
- 도 4는 일 실시예에 따른 디코더의 단순화된 블록 다이어그램의 개략적인 예시이다.
- 도 5는 일 실시예에 따른 인코더의 단순화된 블록 다이어그램의 개략적인 예시이다.
- 도 6은 일 실시예에 따른 오프셋 시그널링을 위한 신택스 요소들의 개략적인 예시이다.
- 도 7은 일 실시예에 따른 인코더의 패딩 폭 시그널링을 위한 신택스 요소들의 개략적인 예시이다.
- 도 8은 일 실시예에 따른 비디오 디코딩을 위해 코딩된 현재 픽처를 재구성하기 위한 예시적인 프로세스의 흐름도이다.

도 9는 일 실시예에 따른 컴퓨터 시스템의 개략적인 예시이다.

**발명을 실시하기 위한 구체적인 내용**

- [0016] 360도 비디오는 ERP(equirectangular projection)과 같은 3D 대 2D 투영 방법(3D-to-2D projection method)을 사용하여 2D 비디오에 맵핑된다. 투영된 비디오는 전통적인 2D 비디오 코더에 의해 인코딩 및 디코딩되고, 2D 비디오를 3D 표면 상에 재-투영함으로써 렌더링된다. 그 다음에, 독립적으로 코딩된 영역들을 스티칭함으로써 재-투영 프로세스로부터 심 시각적 아티팩트들(seam visual artifacts)이 초래된다.
- [0017] 도 2는 본 개시내용의 일 실시예에 따른 통신 시스템(200)의 단순화된 블록 다이어그램을 예시한다. 시스템(200)은 네트워크(250)를 통해 상호접속되는 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 서버 애플리케이션들(media serving applications) 등에서 일반적일 수 있다.
- [0018] 도 2는, 예를 들어, 영상회의(videoconferencing) 동안 발생할 수 있는 코딩된 비디오의 양방향 송신을 지원하기 위해 제공되는 제2 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.
- [0019] 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 플레이어 및/또는 전용 영상 회의 장비가 있는 애플리케이션을 찾는다. 네트워크(250)는, 예를 들어, 와이어라인(wireline) 및/또는 무선 통신 네트워크들을 포함하여, 단말들(210-240) 사이에서 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크를 표현한다. 통신 네트워크(250)는 회선-교환(circuit-switched) 및/또는 패킷-교환(packet-switched) 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 원거리통신(telecommunications) 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.
- [0020] 도 3은, 개시된 주제를 위한 애플리케이션에 대한 예로서, 스트리밍 환경에서의 비디오 인코더 및 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 애플리케이션들에 동등하게 적용가능할 수 있다.
- [0021] 스트리밍 시스템은, 예를 들어, 압축되지 않은 비디오 샘플 스트림(302)을 생성하는, 비디오 소스(301), 예를 들어 디지털 카메라를 포함할 수 있는 캡처 서브시스템(capture subsystem)(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(incoming) 사본(307)을 디코딩하고 디스플레이(312) 또는 다른 렌더링 디바이스(묘사되지 않음) 상에 렌더링될 수 있는 발신(outgoing) 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 표준들에 따라 인코딩될 수 있다. 이러한 표준들의 예들은 ITU-T 권고안(Recommendation) H.265를 포함한다. 비공식적으로 다용도 비디오 코딩(Versatile Video Coding)(VVC)로서 알려진 비디오 코딩 표준이 개발 중이다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.
- [0022] 도 4는 본 개시내용의 일 실시예에 따른 비디오 디코더(310)의 기능적 블록 다이어그램일 수 있다.
- [0023] 수신기(410)는 디코더(310)에 의해 디코딩될 하나 이상의 코덱 비디오 시퀀스를 수신할 수 있고; 동일한 또는 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스를 수신할 수 있고, 각각의 코딩된 비디오 시퀀스의 디

코딩은 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 그들 각자의 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 동시성 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.

[0024] 비디오 디코더(310)는 엔트로피 코딩된 비디오 시퀀스로부터 심벌들(421)을 재구성하기 위해 파서(420)를 포함할 수 있다. 이러한 심벌들의 카테고리들은 디코더(310)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 4에 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 메시지(Supplementary Enhancement Information) 또는 VUI(Video Usability Information) 파라미터 세트 프래그먼트들(묘사되지 않음)의 형식일 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, 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 GOP들(Groups of Pictures), 픽처들, 서브-픽처들, 타일들, 슬라이스들, 브릭들, 매크로블록들, CTU들(Coding Tree Units), CU들(Coding Units), 블록들, TU들(Transform Units), PU들(Prediction Units) 등을 포함할 수 있다. 타일은 픽처에서 특정 타일 열 및 행 내의 CU/CTU들의 직사각형 영역을 나타낼 수 있다. 브릭은 특정 타일 내의 CU/CTU 행들의 직사각형 영역을 나타낼 수 있다. 슬라이스는 NAL 유닛에 포함된 픽처의 하나 이상의 브릭을 나타낼 수 있다. 서브-픽처는 픽처에서 하나 이상의 슬라이스의 직사각형 영역을 나타낼 수 있다. 엔트로피 디코더/파서는 또한 코딩된 비디오 시퀀스로부터 변환 계수들, 양자화기 파라미터 값들, 모션 벡터들 등과 같은 정보를 추출할 수 있다.

[0025] 파서(420)는 버퍼(415)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여, 심벌들(421)을 생성할 수 있다.

[0026] 심벌들(421)의 재구성은 코딩된 비디오 픽처 또는 그의 부분들(이를테면: 인터 및 인트라 픽처, 인터 및 인트라 블록)의 타입, 및 다른 인자들에 의존하는 다수의 상이한 유닛을 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.

[0027] 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 설명할 목적으로, 아래의 기능 유닛들로의 개념적 세분이 적절하다.

[0028] 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 등을 포함하여, 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 그것은 집계기(455)에 입력될 수 있는, 샘플 값들을 포함하는 블록들을 출력할 수 있다.

[0029] 일부 경우들에서, 스케일러/역변환(451)의 출력 샘플들은 인트라 코딩된 블록; 즉: 이전에 재구성된 픽처들로부터의 예측 정보를 사용하는 것이 아니라, 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(부분적으로 재구성된) 픽처(458)으로부터 폐지된 주위의 이미 재구성된 정보를 사용하여, 재구성 중인 블록의 동일한 크기 및 형상의 블록을 생성한다. 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.

- [0030] 다른 경우들에서, 스케일러/역변환 유닛(451)의 출력 샘플들은 인트라 코딩되고, 잠재적으로 모션 보상된 블록에 관련될 수 있다. 이 경우에, 모션 보상 예측 유닛(453)은 참조 픽처 메모리(457)에 액세스하여 예측에 사용되는 샘플을 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 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 유닛에 이용가능한 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.
- [0031] 집계기(455)의 출력 샘플들은 루프 필터 유닛(456)에서 다양한 루프 필터링 기법들의 대상이 될 수 있다. 비디오 압축 기술들은, 과사(420)로부터의 심벌들(421)로서 루프 필터 유닛(456)에 이용가능하게 되고 코딩된 비디오 비트스트림에 포함된 파라미터들에 의해 제어되지만, 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서에서) 부분들의 디코딩 동안 획득된 메타-정보에 즉각 응답할 뿐만 아니라, 이전에 재구성되고 루프-필터링된 샘플 값들에 즉각 응답할 수도 있는 인-루프 필터(in-loop filter) 기술들을 포함할 수 있다.
- [0032] 루프 필터 유닛(456)의 출력은 렌더링 디바이스(312)에 출력될 뿐만 아니라 미래의 인트라-픽처 예측에서 사용하기 위해 참조 픽처 메모리(457)에 저장될 수도 있는 샘플 스트림일 수 있다.
- [0033] 특정 코딩된 픽처들은, 일단 완전히 재구성되면, 미래 예측을 위한 참조 픽처들로서 사용될 수 있다. 일단 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 참조 픽처로서 식별되면(예를 들어, 과사(420)에 의해), 현재 참조 픽처(458)는 참조 픽처 버퍼(457)의 일부가 될 수 있고, 다음의 코딩된 픽처의 재구성을 개시하기 전에 새로운 현재 픽처 메모리가 재할당될 수 있다.
- [0034] 비디오 디코더(310)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, 이것이 비디오 압축 기술 문서 또는 표준에서, 구체적으로는 그 안에 있는 프로파일 문서에서 지정된 바와 같이, 비디오 압축 기술 또는 표준의 선택스를 고수한다는 점에서, 사용되는 비디오 압축 기술 또는 표준에 의해 지정된 선택스를 준수할 수 있다. 또한 준수를 위해 필요한 것은 코딩된 비디오 시퀀스의 복잡도가 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 바와 같은 경계들 내에 있는 것일 수 있다. 일부 경우들에서, 레벨들은 최대 픽처 크기, 최대 프레임 레이트, 최대 재구성 샘플 레이트(예를 들어, 초당 메가샘플들로 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.
- [0035] 일 실시예에서, 수신기(410)는 인코딩된 비디오와 함께 추가적인(중복) 데이터를 수신할 수 있다. 추가적인 데이터는 코딩된 비디오 시퀀스(들)의 부분으로서 포함될 수 있다. 추가적인 데이터는 데이터를 적절히 디코딩하고/하거나 원래의 비디오 데이터를 더 정확하게 재구성하기 위해 비디오 디코더(310)에 의해 사용될 수 있다. 추가적인 데이터는, 예를 들어, 시간적, 공간적, 또는 SNR 향상 계층들, 중복 슬라이스들, 중복 픽처들, 순방향 에러 정정 코드들 등의 형식일 수 있다.
- [0036] 도 5는 본 개시내용의 일 실시예에 따른 비디오 인코더(303)의 기능적 블록 다이어그램일 수 있다.
- [0037] 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.
- [0038] 비디오 소스(301)는, 임의의 적합한 비트 심도(예를 들어: 8 비트, 10 비트, 12 비트, ...), 임의의 색 공간(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 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. 영상회의 시스템에서, 비디오 소스(301)는 비디오 시퀀스로서 로컬 이미지 정보를 캡처하는 카메라일 수 있다. 비디오 데이터는 순차적으로 시청할 때 모션을 부여하는 복수의 개별 픽처로서 제공될 수 있다. 픽처들 자체는 픽셀들의 공간적 어레이로서 조직될 수 있고, 여기서 각각의 픽셀은 사용 중인 샘플링 구조, 색 공간 등에 의존하여 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 픽셀들과 샘플들 사이의 관계를 용이하게 이해할 수 있다. 아래의 설명은 샘플들에 초점을 맞춘다.

- [0039] 일 실시예에 따르면, 인코더(303)는 소스 비디오 시퀀스의 픽처들을 실시간으로 또는 애플리케이션에 의해 요구되는 바와 같은 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. 제어기는 아래 설명된 바와 같이 다른 기능적 유닛들을 제어하고, 이러한 유닛들에 기능적으로 결합된다. 결합은 명료성을 위해 묘사되지 않는다. 제어기에 의해 설정된 파라미터들은 레이트 제어 관련 파라미터들(픽처 스킵, 양자화기, 레이트-왜곡 최적화 기법들의 램다 값, ...), 픽처 크기, 픽처 그룹(GOP) 레이아웃, 최대 모션 벡터 검색 범위 등을 포함할 수 있다. 본 기술 분야의 통상의 기술자는 제어기(550)의 다른 기능들을 쉽게 식별할 수 있는데 그 이유는 그것들이 특정 시스템 설계에 대해 최적화된 비디오 인코더(303)에 관련될 수 있기 때문이다.
- [0040] 일부 비디오 인코더들은 본 기술분야의 통상의 기술자가 "코딩 루프(coding loop)"로서 쉽게 인식하는 것에서 동작한다. 과도하게 단순화된 설명으로서, 코딩 루프는 인코더(530)의 인코딩 부분(이후 "소스 코더")(코딩될 입력 픽처, 및 참조 픽처(들)에 기초하여 심벌들을 생성하는 것을 담당함), 및 (원격) 디코더가 또한 생성할 샘플 데이터를 생성하기 위해 심벌들을 재구성하는 인코더(303)에 내장된 (로컬) 디코더(533)로 구성될 수 있다 (심벌들과 코딩된 비디오 비트스트림 사이의 임의의 압축은 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문에). 재구성된 샘플 스트림은 참조 픽처 메모리(534)에 입력된다. 심벌 스트림의 디코딩은 디코더 위치(로컬 또는 원격)와는 독립적으로 비트-정확한 결과들(bit-exact results)로 이어지기 때문에, 참조 픽처 버퍼 콘텐츠도 또한 로컬 인코더와 원격 인코더 사이에서 비트 정확(bit exact)하다. 다시 말해서, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "볼(would see)" 것과 정확히 동일한 샘플 값들을 참조 픽처 샘플로서 "본다(sees)". 참조 픽처 동기성(reference picture synchronicity)의 이러한 기본적인 원리(및, 예를 들어, 채널 에러들 때문에, 동기성이 유지될 수 없는 경우, 결과적인 드리프트)는 본 기술분야의 통상의 기술자에게 잘 알려져 있다.
- [0041] "로컬" 디코더(533)의 동작은 도 4와 관련하여 위에서 이미 상세히 설명된 "원격" 디코더(310)와 동일할 수 있다. 그러나, 간단히 도 4를 또한 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.
- [0042] 이 시점에서 이루어질 수 있는 관찰은, 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한, 대응하는 인코더에서, 실질적으로 동일한 기능적 형식으로 반드시 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.
- [0043] 그 동작의 일부로서, 소스 코더(530)는 "참조 프레임들"로서 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 프레임들을 참조하여 예측적으로 입력 프레임을 코딩하는 모션 보상 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 사이의 차이들을 코딩한다.
- [0044] 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 에러들을 갖는 소스 비디오 시퀀스의 복제본(replica)일 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 에러들이 없이) 원단(far-end) 비디오 디코더에 의해 획득될 재구성된 참조 프레임들로서 공통 콘텐츠를 갖는 재구성된 참조 프레임들의 사본들을 로컬로 저장할 수 있다.
- [0045] 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 참조로서 역할할 수 있는 참조 픽처 모션 벡터들, 블록 형상들과 같은 특정 메타데이터 또는 샘플 데이터(후보 참조 픽셀 블록들로서)를 참조 픽처 메모리(534)에서 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.

- [0046] 제어기(550)는, 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하여, 소스 코더(530)의 코딩 동작들을 관리할 수 있다.
- [0047] 전술한 모든 기능 유닛들의 출력은 엔트로피 코더(545)에서 엔트로피 코딩의 대상이 될 수 있다. 엔트로피 코더는, 예를 들어, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등으로서 본 기술분야의 통상의 기술자에게 알려진 기술들에 따라 심벌들을 무손실 압축함으로써 다양한 기능 유닛들에 의해 생성된 심벌들을 코딩된 비디오 시퀀스로 변환한다.
- [0048] 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 소스 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.
- [0049] 제어기(550)는 인코더(303)의 동작을 관리할 수 있다. 코딩 동안, 제어기(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 이하의 프레임 타입들 중 하나로서 할당될 수 있다:
- [0050] 인트라 픽처(Intra Picture)(I 픽처)는 예측의 소스로서 시퀀스 내의 임의의 다른 프레임을 사용하지 않고 코딩 및 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은, 예를 들어, 독립적 디코더 리프레시 픽처들(Independent Decoder Refresh Pictures)을 포함하는, 상이한 타입들의 인트라 픽처들을 허용한다. 본 기술분야의 통상의 기술자는 I 픽처들의 이러한 변형들 및 그들 각자의 애플리케이션들 및 특징들을 인식한다.
- [0051] 예측 픽처(Predictive picture)(P 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 하나의 모션 벡터 및 참조 인덱스를 사용하는 인트라 예측(intra prediction) 또는 인터 예측(inter prediction)을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다.
- [0052] 양방향 예측 픽처(Bi-directionally Predictive Picture)(B 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 2개의 모션 벡터 및 참조 인덱스를 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다. 유사하게, 다수의-예측 픽처들은 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.
- [0053] 소스 픽처들은 흔히 복수의 샘플 블록(예를 들어, 각각 4x4, 8x8, 4x8, 또는 16x16 샘플들의 블록들)로 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 비예측적으로 코딩될 수 있거나 그들은 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.
- [0054] 비디오 인코더(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 인코더(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 지정된 선택어(syntax)를 준수할 수 있다.
- [0055] 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 시간/공간/SNR 향상 계층들, 중복 픽처들 및 슬라이스들과 같은 다른 형태들의 중복 데이터, SEI(Supplementary Enhancement Information) 메시지들, VUI(Visual Usability Information) 파라미터 세트 프래그먼트들 등을 포함할 수 있다.
- [0056] 도 6-7을 참조하면, 일 실시예에서, 360 비디오는 카메라들의 세트 또는 다수의 렌즈를 갖는 카메라 디바이스에 의해 캡처된다. 카메라들은 카메라 세트의 중심점 주위의 전방향들을 커버할 수 있다. 동일한 시간 인스턴스의 이미지들은 스티칭되고, 가능하게는 회전되고, 투영되고, 픽처 상에 매핑된다. 패킹된 픽처들은 코딩된 비디오 비트스트림으로 코딩된 것으로서 인코딩되고, 특정 미디어 컨테이너 파일 포맷에 따라 스트리밍된다. 파

일은 투영 및 패킹 정보와 같은 메타데이터를 포함한다.

- [0057] 일 실시예에서, 360 비디오는 ERP(equirectangular projection)을 사용하여 2D 비디오에 투영될 수 있다. ERP 투영은 심 아티팩트들(seam artifacts)을 초래할 수 있다. 패딩된 ERP(PERP) 포맷은 ERP 픽처의 좌측 및 우측 경계들을 포괄하는 재구성된 뷰포트들에서 심 아티팩트들을 효과적으로 감소시킬 수 있다. 그러나, 패딩 및 블렌딩은 심 이슈(seam issue)를 완전히 해결하기에 충분하지 않을 수 있다.
- [0058] 일 실시예에서, 심 아티팩트를 감소시키기 위해 ERP 또는 PERP에 대해 수평 지오메트리 패딩(horizontal geometry padding)이 적용될 수 있다. PERP에 대한 패딩 프로세스는, 오프셋이 패딩된 영역들의 사이즈를 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 외측에 있는 경우, 그것은 ERP 폭만큼 우측(좌측)으로 시프트된 "랩-어라운드" 참조 블록으로 대체될 수 있다. 전통적인 반복 패딩은 수직 방향으로 이용될 수 있다. 좌측 및 우측 패딩된 영역들의 혼합은 사후-처리 동작으로서 루프에서 제외된다.
- [0059] 일 실시예에서, ERP 및 PERP 포맷들에 대한 참조 픽처들의 수평 지오메트리 패딩을 가능하게 하는 선택스, 예를 들어, seq\_parameter\_set\_rbsp()(601)가 도 6에 도시된다.
- [0060] 일 실시예에서, sps\_ref\_wraparound\_enabled\_flag(602)가 1이라는 것은 수평 랩-어라운드 모션 보상이 인터 예측을 위해 사용되는 것을 지정한다. 일 실시예에서, sps\_ref\_wraparound\_enabled\_flag(602)가 0이라는 것은 이 모션 보상 방법이 적용되지 않음을 지정한다.
- [0061] 일 실시예에서, ref\_wraparound\_offset(603)이 수평 랩-어라운드 위치를 계산하기 위해 사용되는 루마 샘플들 내의 오프셋을 지정한다. 일 실시예에서, ref\_wraparound\_offset(603)은 pic\_width\_in\_luma\_samples-1보다 크야 하고, pic\_width\_in\_luma\_samples보다 크지 않아야 하고, MinCbSizeY의 정수배이어야 한다.
- [0062] 일 실시예에서, ERP 및 PERP 포맷들에 대한 참조 픽처들의 수평 지오메트리 패딩을 가능하게 하는 선택스, 예를 들어, seq\_parameter\_set\_rbsp()(701)가 도 7에 도시된다.
- [0063] 일 실시예에서, sps\_ref\_wraparound\_enabled\_flag(702)가 1이라는 것은 수평 랩-어라운드 모션 보상이 인터 예측을 위해 사용되는 것을 지정한다. sps\_ref\_wraparound\_enabled\_flag(702)가 0이라는 것은 이 모션 보상 방법이 적용되지 않음을 지정한다.
- [0064] 일 실시예에서, left\_wraparound\_padding\_width(703)는 루마 샘플들에서 좌측 패딩 영역의 폭을 지정한다. 일 실시예에서, ref\_wraparound\_offset은 0 이상이어야 하고, pic\_width\_in\_luma\_samples/2보다 크지 않아야 하고, MinCbSizeY의 정수배이어야 한다.
- [0065] 일 실시예에서, right\_wraparound\_padding\_width(704)는 루마 샘플들에서 우측 패딩 영역의 폭을 지정한다. 일 실시예에서, ref\_wraparound\_offset은 0 이상이어야 하고, pic\_width\_in\_luma\_samples/2보다 크지 않아야 하고, MinCbSizeY의 정수배이어야 한다.
- [0066] 일 실시예에서, 랩-어라운드 오프셋 값은 아래의 도출 프로세스에 의해 획득될 수 있다:  
 ref\_wraparound\_offset 이 존재하는 경우  

$$\text{wrapAroundOffset} = \text{ref\_wraparound\_offset}$$
 그렇지 않고 left\_wraparound\_padding\_width 및 right\_wraparound\_padding\_width 이 존재하는 경우  

$$\text{wrapAroundOffset} = \text{pic\_width\_in\_luma\_samples} - (\text{left\_wraparound\_padding\_width} + \text{right\_wraparound\_padding\_width})$$
 그렇지 않으면  

$$\text{wrapAroundOffset} = \text{pic\_width\_in\_luma\_samples}$$
- [0067]
- [0068] 일 실시예에서, ERP 및 PERP 포맷들에 대한 참조 픽처들의 수평 지오메트리 패딩을 가능하게 하기 위해, 루마 및 크로마 샘플 보간 프로세스들이 수정될 수 있다.

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

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

[0069]

[0070] 일 실시예에 따른 루마 샘플 보간 프로세스의 예, 및 일 실시예에 따른 크로마 샘플 보간 프로세스의 예가 아래에 설명된다:

[0071] **루마 샘플 보간 프로세스**

[0072] 이 프로세스에 대한 입력들은 다음과 같다:

- [0073] - 전체-샘플 유닛들(full-sample units)에서의 루마 위치 (xInt<sub>L</sub>, yInt<sub>L</sub>),
- [0074] - 부분-샘플 유닛들(fractional-sample units)에서의 루마 위치 (xFrac<sub>L</sub>, yFrac<sub>L</sub>),
- [0075] - 루마 참조 샘플 어레이 refPicL<sub>X<sub>L</sub></sub>.

[0076] 이 프로세스의 출력은 예측된 루마 샘플 값 predSampleL<sub>X<sub>L</sub></sub>이다

[0077] 변수 shift1, shift2 및 shift3은 다음과 같이 도출된다:

- [0078] - 변수 shift1은 Min(4, BitDepth<sub>Y</sub> - 8)과 동일하게 설정되고, 변수 shift2는 6과 동일하게 설정되고, 변수 shift3은 Max(2, 14 - BitDepth<sub>Y</sub>)와 동일하게 설정된다.

- [0079] - 변수 picW는 pic\_width\_in\_luma\_samples와 동일하게 설정되고, 변수 picH는 pic\_height\_in\_luma\_samples와 동일하게 설정된다.

- [0080] - 변수 xOffset는 wrapAroundOffset과 동일하게 설정된다.

[0081] xFrac<sub>L</sub> 또는 yFrac<sub>L</sub>과 동일한 각각의 1/16 부분 샘플 위치 p에 대한 루마 보간 필터 계수들 f<sub>L</sub>[p]는 아래에 지정된다.

[0082] 예측된 루마 샘플 값 predSampleL<sub>X<sub>L</sub></sub>은 다음과 같이 도출된다:

- [0083] - xFrac<sub>L</sub> 및 yFrac<sub>L</sub> 둘 모두 0인 경우, 다음이 적용된다:

- [0084] - sps\_ref\_wraparound\_enabled\_flag가 0인 경우, predSampleL<sub>X<sub>L</sub></sub>의 값은 다음과 같이 도출된다:

[0085]  $\text{predSampleL}_{X_L} = \text{refPicL}_{X_L} [ \text{Clip3}(0, \text{picW} - 1, x\text{Int}_L) ] [ \text{Clip3}(0, \text{picH} - 1, y\text{Int}_L) ] \ll \text{shift3}$

- [0086] - 그렇지 않으면, predSampleL<sub>X<sub>L</sub></sub>의 값은 다음과 같이 도출된다:

[0087]  $\text{predSampleL}_{X_L} = \text{refPicL}_{X_L} [ \text{ClipH}(x\text{Offset}, \text{picW}, x\text{Int}_L) ] [ \text{Clip3}(0, \text{picH} - 1, y\text{Int}_L) ] \ll \text{shift3}$

- [0088] - 그렇지 않고 xFrac<sub>L</sub>이 0이 아니고 yFrac<sub>L</sub>이 0인 경우, 다음이 적용된다:

- [0089] - yPos<sub>L</sub>의 값은 다음과 같이 도출된다:

[0090]  $y\text{Pos}_L = \text{Clip3}(0, \text{picH} - 1, y\text{Int}_L)$

[0091] - sps\_ref\_wraparound\_enabled\_flag가 0인 경우, predSampleLX<sub>L</sub>의 값은 다음과 같이 도출된다:

$$\begin{aligned} \text{predSampleLX}_L = & ( f_L[ \text{xFrac}_L ][ 0 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L - 3 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 1 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L - 2 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 2 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L - 1 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 3 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 4 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L + 1 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 5 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L + 2 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 6 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L + 3 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 7 ] * \text{refPicLXL}[ \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L + 4 ) ][ \text{yPos}_L ] ) \gg \text{shift1} \end{aligned}$$

[0092] - 그렇지 않으면, predSampleLX<sub>L</sub>의 값은 다음과 같이 도출된다:

$$\begin{aligned} \text{predSampleLX}_L = & ( f_L[ \text{xFrac}_L ][ 0 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L - 3 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 1 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L - 2 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 2 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L - 1 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 3 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 4 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L + 1 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 5 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L + 2 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 6 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L + 3 ) ][ \text{yPos}_L ] + \\ & f_L[ \text{xFrac}_L ][ 7 ] * \text{refPicLXL}[ \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L + 4 ) ][ \text{yPos}_L ] ) \gg \text{shift1} \end{aligned}$$

[0093] - 그렇지 않고 xFrac<sub>L</sub>이 0이 아니고 yFrac<sub>L</sub>이 0이 아닌 경우, predSampleLX<sub>L</sub>의 값은 다음과 같이 도출된다:

[0094] - sps\_ref\_wraparound\_enabled\_flag가 0인 경우, xPos<sub>L</sub>의 값은 다음과 같이 도출된다:

$$\text{xPos}_L = \text{Clip3}( 0, \text{picW} - 1, \text{xInt}_L )$$

[0095] - 그렇지 않으면, xPos<sub>L</sub>의 값은 다음과 같이 도출된다:

$$\text{xPos}_L = \text{ClipH}( \text{xOffset}, \text{picW}, \text{xInt}_L )$$

[0096] - 예측된 루마 샘플 값 predSampleLX<sub>L</sub>은 다음과 같이 도출된다:

$$\begin{aligned} \text{predSampleLX}_L = & ( f_L[ \text{yFrac}_L ][ 0 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L - 3 ) ] + \\ & f_L[ \text{yFrac}_L ][ 1 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L - 2 ) ] + \\ & f_L[ \text{yFrac}_L ][ 2 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L - 1 ) ] + \\ & f_L[ \text{yFrac}_L ][ 3 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L ) ] + \\ & f_L[ \text{yFrac}_L ][ 4 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L + 1 ) ] + \\ & f_L[ \text{yFrac}_L ][ 5 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L + 2 ) ] + \\ & f_L[ \text{yFrac}_L ][ 6 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L + 3 ) ] + \\ & f_L[ \text{yFrac}_L ][ 7 ] * \text{refPicLXL}[ \text{xPos}_L ][ \text{Clip3}( 0, \text{picH} - 1, \text{yInt}_L + 4 ) ] ) \gg \text{shift1} \end{aligned}$$

[0097] - 그렇지 않고 xFrac<sub>L</sub>이 0이 아니고 yFrac<sub>L</sub>이 0이 아닌 경우, predSampleLX<sub>L</sub>의 값은 다음과 같이

도출된다:

[0103] - sps\_ref\_wraparound\_enabled\_flag가 0인 경우, n = 0..7인 샘플 어레이 temp[n]은 다음과 같이 도출된다:

[0104] 
$$yPos_L = Clip3( 0, picH - 1, yInt_L + n - 3 )$$

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

[0105] 
$$f_L[ xFrac_L ][ 7 ] * refPicLX_L[ Clip3( 0, picW - 1, xInt_L + 4 ) ][ yPos_L ] ) \gg shift1$$

[0106] - 그렇지 않으면, n=0..7인 샘플 어레이 temp[n]은 다음과 같이 도출된다:

[0107] 
$$yPos_L = Clip3( 0, picH - 1, yInt_L + n - 3 )$$

$$temp[ n ] = ( f_L[ xFrac_L ][ 0 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 3 ) ][ yPos_L ] + f_L[ xFrac_L ][ 1 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 2 ) ][ yPos_L ] + f_L[ xFrac_L ][ 2 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L - 1 ) ][ yPos_L ] + f_L[ xFrac_L ][ 3 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L ) ][ yPos_L ] + f_L[ xFrac_L ][ 4 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 1 ) ][ yPos_L ] + f_L[ xFrac_L ][ 5 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 2 ) ][ yPos_L ] + f_L[ xFrac_L ][ 6 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 3 ) ][ yPos_L ] +$$

[0108] 
$$f_L[ xFrac_L ][ 7 ] * refPicLX_L[ ClipH( xOffset, picW, xInt_L + 4 ) ][ yPos_L ] ) \gg shift1$$

[0109] - 예측된 루마 샘플 값 predSampleLX\_L은 다음과 같이 도출된다:

$$predSampleLX_L = ( f_L[ yFrac_L ][ 0 ] * temp[ 0 ] + f_L[ yFrac_L ][ 1 ] * temp[ 1 ] + f_L[ yFrac_L ][ 2 ] * temp[ 2 ] + f_L[ yFrac_L ][ 3 ] * temp[ 3 ] + f_L[ yFrac_L ][ 4 ] * temp[ 4 ] + f_L[ yFrac_L ][ 5 ] * temp[ 5 ] + f_L[ yFrac_L ][ 6 ] * temp[ 6 ] + f_L[ yFrac_L ][ 7 ] * temp[ 7 ] ) \gg shift2$$

[0110] **크로마 샘플 보간 프로세스**

[0111] 이 프로세스에 대한 입력들은 다음과 같다:

[0112] - 전체-샘플 유닛들에서의 크로마 위치 (xInt<sub>c</sub>, yInt<sub>c</sub>),

[0113] - 1/32 부분-샘플 유닛들에서의 크로마 위치 (xFrac<sub>c</sub>, yFrac<sub>c</sub>),

[0114] - 크로마 참조 샘플 어레이 refPicLX<sub>c</sub>.

[0115] 이 프로세스의 출력은 예측된 크로마 샘플 값 predSampleLX<sub>c</sub>이다

[0116] 변수 shift1, shift2 및 shift3은 다음과 같이 도출된다:

[0118] - 변수  $shift1$ 은  $\text{Min}(4, \text{BitDepth}_c - 8)$ 과 동일하게 설정되고, 변수  $shift2$ 는 6과 동일하게 설정되고, 변수  $shift3$ 은  $\text{Max}(2, 14 - \text{BitDepth}_c)$ 와 동일하게 설정된다.

[0119] - 변수  $picW_c$ 는  $\text{pic\_width\_in\_luma\_samples}/\text{SubWidth}_C$ 와 동일하게 설정되고, 변수  $picH_c$ 는  $\text{pic\_height\_in\_luma\_samples}/\text{SubHeight}_C$ 와 동일하게 설정된다.

[0120] - 변수  $xOffset_c$ 는  $\text{wrapAroundOffset}/\text{SubWidth}_C$ 와 동일하게 설정된다.

[0121]  $xFrac_c$  또는  $yFrac_c$ 와 동일한 각각의 1/32 부분 샘플 위치  $p$ 에 대한 루마 보간 필터 계수들  $f_c[p]$ 가 아래에 지정된다.

[0122] 예측된 크로마 샘플 값  $\text{predSampleLX}_c$ 는 다음과 같이 도출된다:

[0123] -  $xFrac_c$  및  $yFrac_c$  둘 모두가 0인 경우, 이하가 적용된다:

[0124] -  $\text{sps\_ref\_wraparound\_enabled\_flag}$ 가 0인 경우,  $\text{predSampleLX}_c$ 의 값은 다음과 같이 도출된다:

[0125] 
$$\text{predSampleLX}_c = \text{refPicLX}_c[ \text{Clip3}(0, \text{picW}_c - 1, xInt_c) ][ \text{Clip3}(0, \text{picH}_c - 1, yInt_c) ] \ll \text{shift3}$$

[0126] - 그렇지 않으면,  $\text{predSampleLX}_c$ 의 값은 다음과 같이 도출된다:

[0127] 
$$\text{predSampleLX}_c = \text{refPicLX}_c[ \text{ClipH}(xOffset_c, \text{picW}_c, xInt_c) ][ \text{Clip3}(0, \text{picH}_c - 1, yInt_c) ] \ll \text{shift3}$$

[0128] - 그렇지 않고  $xFrac_c$ 가 0이 아니고  $yFrac_c$ 가 0인 경우, 다음이 적용된다:

[0129] -  $yPos_c$ 의 값은 다음과 같이 도출된다:

[0130] 
$$yPos_c = \text{Clip3}(0, \text{picH}_c - 1, yInt_c)$$

[0131] -  $\text{sps\_ref\_wraparound\_enabled\_flag}$ 가 0인 경우,  $\text{predSampleLX}_c$ 의 값은 다음과 같이 도출된다:

[0132] 
$$\begin{aligned} \text{predSampleLX}_c = & ( f_c[xFrac_c][0] * \text{refPicLX}_c[ \text{Clip3}(0, \text{picW}_c - 1, xInt_c - 1) ][ yInt_c ] + \\ & f_c[xFrac_c][1] * \text{refPicLX}_c[ \text{Clip3}(0, \text{picW}_c - 1, xInt_c) ][ yInt_c ] + \\ & f_c[xFrac_c][2] * \text{refPicLX}_c[ \text{Clip3}(0, \text{picW}_c - 1, xInt_c + 1) ][ yInt_c ] + \\ & f_c[xFrac_c][3] * \text{refPicLX}_c[ \text{Clip3}(0, \text{picW}_c - 1, xInt_c + 2) ][ yInt_c ] ) \gg \text{shift1} \end{aligned}$$

[0133] - 그렇지 않으면,  $\text{predSampleLX}_c$ 의 값은 다음과 같이 도출된다:

[0134] 
$$\begin{aligned} \text{predSampleLX}_c = & ( f_c[xFrac_c][0] * \text{refPicLX}_c[ \text{ClipH}(xOffset_c, \text{picW}_c, xInt_c - 1) ][ yPos_c ] + \\ & f_c[xFrac_c][1] * \text{refPicLX}_c[ \text{ClipH}(xOffset_c, \text{picW}_c, xInt_c) ][ yPos_c ] + \\ & f_c[xFrac_c][2] * \text{refPicLX}_c[ \text{ClipH}(xOffset_c, \text{picW}_c, xInt_c + 1) ][ yPos_c ] + \\ & f_c[xFrac_c][3] * \text{refPicLX}_c[ \text{ClipH}(xOffset_c, \text{picW}_c, xInt_c + 2) ][ yPos_c ] ) \gg \text{shift1} \end{aligned}$$

[0135] - 그렇지 않고  $xFrac_c$ 가 0이고  $yFrac_c$ 가 0이 아닌 경우,  $\text{predSampleLX}_c$ 의 값은 다음과 같이 도출된다:

[0136] -  $\text{sps\_ref\_wraparound\_enabled\_flag}$ 가 0인 경우,  $xPos_c$ 의 값은 다음과 같이 도출된다:

[0137] 
$$xPos_c = \text{Clip3}(0, \text{picW}_c - 1, xInt_c)$$

[0138] - 그렇지 않으면,  $xPosC$ 의 값은 다음과 같이 도출된다:

$$[0139] \quad xPosC = ClipH(xOffsetC, picWC, xIntC)$$

[0140] - 예측된 크로마 샘플 값  $predSampleLXC$ 는 다음과 같이 도출된다:

$$[0141] \quad \begin{aligned} predSampleLXC = & (fc[yFracC][0] * refPicLXC[xPosC][Clip3(0, picHC - 1, yIntC - 1)]) + \\ & fc[yFracC][1] * refPicLXC[xPosC][Clip3(0, picHC - 1, yIntC)] + \\ & fc[yFracC][2] * refPicLXC[xPosC][Clip3(0, picHC - 1, yIntC + 1)] + \\ & fc[yFracC][3] * refPicLXC[xPosC][Clip3(0, picHC - 1, yIntC + 2)]) \gg shift1 \end{aligned}$$

[0142] - 그렇지 않고  $xFracC$ 가 0이 아니고  $yFracC$ 가 0이 아닌 경우,  $predSampleLXC$ 의 값은 다음과 같이 도출된다:

[0143] -  $sps\_ref\_wraparound\_enabled\_flag$ 가 0인 경우,  $n=0..3$ 인 샘플 어레이  $temp[n]$ 은 다음과 같이 도출된다:

$$[0144] \quad yPosC = Clip3(0, picHC - 1, yIntC + n - 1)$$

$$[0145] \quad \begin{aligned} temp[n] = & (fc[xFracC][0] * refPicLXC[Clip3(0, picWC - 1, xIntC - 1)][yPosC] + \\ & fc[xFracC][1] * refPicLXC[Clip3(0, picWC - 1, xIntC)][yPosC] + \\ & fc[xFracC][2] * refPicLXC[Clip3(0, picWC - 1, xIntC + 1)][yPosC] + \\ & fc[xFracC][3] * refPicLXC[Clip3(0, picWC - 1, xIntC + 2)][yPosC]) \gg shift1 \end{aligned}$$

[0146] - 그렇지 않으면,  $n=0..3$ 인 샘플 어레이  $temp[n]$ 은 다음과 같이 도출된다:

$$[0147] \quad yPosC = Clip3(0, picHC - 1, yIntC + n - 1)$$

$$[0148] \quad \begin{aligned} temp[n] = & (fc[xFracC][0] * refPicLXC[ClipH(xOffsetC, picWC, xIntC - 1)][yPosC] + \\ & fc[xFracC][1] * refPicLXC[ClipH(xOffsetC, picWC, xIntC)][yPosC] + \\ & fc[xFracC][2] * refPicLXC[ClipH(xOffsetC, picWC, xIntC + 1)][yPosC] + \\ & fc[xFracC][3] * refPicLXC[ClipH(xOffsetC, picWC, xIntC + 2)][yPosC]) \gg shift1 \end{aligned}$$

[0149] - 예측된 크로마 샘플 값  $predSampleLXC$ 는 다음과 같이 도출된다:

$$[0150] \quad \begin{aligned} predSampleLXC = & (fc[yFracC][0] * temp[0] + \\ & fc[yFracC][1] * temp[1] + \\ & fc[yFracC][2] * temp[2] + \\ & fc[yFracC][3] * temp[3]) \gg shift2 \end{aligned}$$

[0151] 일 실시예에서,  $sps\_ref\_wraparound\_enabled\_flag(602)$  또는  $sps\_ref\_wraparound\_enabled\_flag(702)$ 가 0이거나 존재하지 않으면, 종래의 반복 패딩이 적용될 수 있다. 그렇지 않으면, 램-어라운드 패딩이 적용될 수 있다.

[0152] 일 실시예에서, 램-어라운드 패딩은 수평 경계 및 수직 경계 둘 모두에서 적용될 수 있다. 고레벨 신텍스 구조에서의 플래그는 램-어라운드 패딩이 수평 및 수직 둘 모두에서 적용된다는 것을 나타낼 수 있다.

[0153] 일 실시예에서, 램-어라운드 패딩은 브릭, 타일, 슬라이스 또는 서브-픽처 경계들에서 적용될 수 있다. 일 실시예에서, 램-어라운드 패딩은 타일 그룹 경계들에서 적용될 수 있다. 고레벨 신텍스 구조에서의 플래그는 램-어라운드 패딩이 수평 및 수직 둘 모두에서 적용된다는 것을 나타낼 수 있다.

[0154] 일 실시예에서, 참조 픽처는 모션 보상된 예측에 대해 현재 픽처와 동일할 수 있다. 램-어라운드 패딩은 현재

픽처가 참조일 때 현재 픽처의 경계에서 적용될 수 있다.

- [0155] 도 8은 비디오 디코딩을 위해 코딩된 현재 픽처를 재구성하기 위한 예시적인 프로세스(800)의 흐름도이다. 일부 구현들에서, 도 8의 하나 이상의 프로세스 블록은 디코더(310)에 의해 수행될 수 있다. 일부 구현들에서, 도 8의 하나 이상의 프로세스 블록은 인코더(303)와 같이, 디코더(310)로부터 분리되거나 또는 이를 포함하는 다른 디바이스 또는 디바이스들의 그룹에 의해 수행될 수 있다.
- [0156] 도 8에 도시된 바와 같이, 프로세스(800)는 현재 픽처에 대응하는 픽처 파티셔닝 정보를 디코딩하는 것(블록 810)을 포함할 수 있다.
- [0157] 도 8에 추가로 도시된 바와 같이, 프로세스(800)는 픽처 파티셔닝 정보를 사용하여 현재 픽처의 복수의 서브-영역에 패딩이 적용되는지를 결정하는 것(블록 820)을 포함할 수 있다.
- [0158] 도 8에 추가로 도시된 바와 같이, 패딩이 적용되지 않는다는 결정에 기초하여, 프로세스(800)는 복수의 서브-영역을 패딩하지 않고 복수의 서브-영역을 디코딩하는 것(블록 830)을 포함할 수 있다. 그 다음에, 프로세스(800)는 디코딩된 복수의 서브-영역에 기초하여 현재 픽처를 재구성하는 것(블록 870)으로 진행할 수 있다.
- [0159] 도 8에 추가로 도시된 바와 같이, 패딩이 적용된다는 결정에 기초하여, 프로세스(800)는 픽처 파티셔닝 정보를 사용하여 패딩이 랩-어라운드 패딩을 포함하는지를 결정하는 것(블록 840)을 포함할 수 있다.
- [0160] 도 8에 추가로 도시된 바와 같이, 패딩이 랩-어라운드 패딩을 포함하지 않는다는 결정에 기초하여, 프로세스(800)는 반복 패딩을 복수의 서브-영역에 적용하는 것, 및 반복 패딩을 사용하여 복수의 서브-영역을 디코딩하는 것(블록 850)을 포함할 수 있다. 그 다음에, 프로세스(800)는 디코딩된 복수의 서브-영역에 기초하여 현재 픽처를 재구성하는 것(블록 870)으로 진행할 수 있다.
- [0161] 도 8에 추가로 도시된 바와 같이, 패딩이 랩-어라운드 패딩을 포함한다는 결정에 기초하여, 프로세스(800)는 랩-어라운드 패딩을 복수의 서브-영역에 적용하는 것, 및 랩-어라운드 패딩을 사용하여 복수의 서브-영역을 디코딩하는 것(블록 860)을 포함할 수 있다. 그 다음에, 프로세스(800)는 디코딩된 복수의 서브-영역에 기초하여 현재 픽처를 재구성하는 것(블록 870)으로 진행할 수 있다.
- [0162] 일 실시예에서, 픽처 파티셔닝 정보는 현재 픽처에 대응하는 픽처 파라미터 세트에 포함될 수 있다.
- [0163] 일 실시예에서, 픽처 파티셔닝 정보는 픽처 파라미터 세트에 포함된 적어도 하나의 플래그를 포함한다.
- [0164] 일 실시예에서, 복수의 서브-영역은 브릭, 타일, 슬라이스, 타일 그룹, 서브-픽처, 또는 서브-계층 중에서 적어도 하나를 포함한다.
- [0165] 일 실시예에서, 패딩은 복수의 서브-영역 중의 서브-영역의 경계에 적용될 수 있다.
- [0166] 일 실시예에서, 경계는 서브-영역의 수직 경계일 수 있다.
- [0167] 일 실시예에서, 경계는 서브-영역의 수평 경계일 수 있다.
- [0168] 일 실시예에서, 패딩은 복수의 서브-영역 중의 서브-영역의 수직 경계 및 서브-영역의 수평 경계에 적용될 수 있다.
- [0169] 일 실시예에서, 픽처 파티션 정보는 랩-어라운드 패딩에 대한 오프셋 값을 나타낼 수 있다.
- [0170] 일 실시예에서, 픽처 파티션 정보는 좌측 패딩 폭 정보 및 우측 패딩 폭 정보를 나타낼 수 있다.
- [0171] 도 8은 프로세스(800)의 예시적인 블록들을 도시하지만, 일부 구현들에서, 프로세스(800)는 도 8에 도시된 것들에 추가적인 블록들, 더 적은 블록들, 상이한 블록들 또는 상이하게 배열된 블록들을 포함할 수 있다. 추가로 또는 대안으로, 프로세스(800)의 블록들 중 2개 이상이 병렬로 수행될 수 있다.
- [0172] 또한, 제안된 방법들은 처리 회로(예를 들어, 하나 이상의 프로세서 또는 하나 이상의 집적 회로)에 의해 구현될 수 있다. 하나의 예에서, 하나 이상의 프로세서는 제안된 방법들 중 하나 이상을 수행하기 위해 비일시적 컴퓨터 판독가능 매체에 저장되는 프로그램을 실행한다.
- [0173] 위에서 설명된 기법들은 컴퓨터 판독가능 명령어들을 사용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 판독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 9는 개시된 주제의 특정 실시예들을 구현하기에 적합한 컴퓨터 시스템(900)을 도시한다.

- [0174] 컴퓨터 소프트웨어는, 컴퓨터 중앙 처리 유닛(CPU), 그래픽 처리 유닛(GPU) 등에 의해, 직접 실행될 수 있거나, 또는 해석, 마이크로-코드 실행 등을 통해 실행될 수 있는 명령어들을 포함하는 코드를 생성하기 위해, 어셈블리, 컴파일레이션, 링킹, 또는 유사한 메커니즘들의 대상이 될 수 있는 임의의 적합한 머신 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.
- [0175] 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.
- [0176] 컴퓨터 시스템(900)에 대한 도 9에 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(900)의 예시적인 실시예에 예시된 컴포넌트들 중 어느 하나 또는 조합과 관련 있는 임의의 의속성 또는 요건을 갖는 것으로 해석되어서는 안된다.
- [0177] 컴퓨터 시스템(900)은 특정 인간 인터페이스 입력 디바이스들을 포함할 수 있다. 그러한 인간 인터페이스 입력 디바이스는, 예를 들어, 촉각 입력(이를테면: 키스트로크, 스위이프, 데이터 글러브 움직임), 오디오 입력(이를테면: 음성, 손뼉), 시각적 입력(이를테면, 제스처), 후각적 입력(묘사되지 않음)을 통한 하나 이상의 인간 사용자에게 의한 입력에 즉각 응답할 수 있다. 인간 인터페이스 디바이스들은 또한 오디오(이를테면: 음성, 음악, 주변 사운드), 이미지들(이를테면: 스캐닝된 이미지들, 스틸 이미지 카메라로부터 획득된 사진 이미지들), 비디오(이를테면 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.
- [0178] 입력 인간 인터페이스 디바이스들은: 키보드(901), 마우스(902), 트랙패드(903), 터치 스크린(910)과 연관된 그래픽 어댑터(950), 데이터-글러브, 조이스틱(905), 마이크로폰(906), 스캐너(907), 카메라(908) 중 하나 이상(각각의 하나만이 묘사됨)을 포함할 수 있다.
- [0179] 컴퓨터 시스템(900)은 또한 특정한 인간 인터페이스 출력 디바이스들을 포함할 수 있다. 그러한 인간 인터페이스 출력 디바이스들은, 예를 들어, 촉각 출력, 사운드, 광, 및 냄새/맛을 통해 하나 이상의 인간 사용자의 감각들을 자극하고 있을 수 있다. 그러한 인간 인터페이스 출력 디바이스들은 촉각 출력 디바이스들(예를 들어 터치-스크린(910), 데이터-글러브, 또는 조이스틱(905)에 의한 촉각 피드백이지만, 입력 디바이스들로서 역할하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(이를테면: 스피커들(909), 헤드폰들(묘사되지 않음)), 시각적 출력 디바이스들(이를테면 CRT(cathode ray tube) 스크린들, LCD(liquid-crystal display) 스크린들, 플라즈마 스크린들, OLED(organic light-emitting diode) 스크린들을 포함하는 스크린들(910), 각각은 터치-스크린 입력 능력이 있거나 없고, 각각은 촉각 피드백 능력이 있거나 없고- 이들 중 일부는 스테레오그래픽 출력; 가상 현실 안경(묘사되지 않음), 홀로그래픽 디스플레이들 및 연기 탱크들(묘사되지 않음)과 같은 수단을 통해 2차원 시각적 출력 또는 3차원 이상의 출력을 출력할 수 있음), 및 프린터들(묘사되지 않음)을 포함할 수 있다.
- [0180] 컴퓨터 시스템(900)은 인간 액세스가능 저장 디바이스들 및 그들과 연관된 매체, 이를테면 CD/DVD 등의 매체(921)를 갖는 CD/DVD ROM/RW(920)를 포함하는 광학 매체, 썸-드라이브(922), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(923), 테이프 및 플로피 디스크(묘사되지 않음)와 같은 레거시 자기 매체, 보안 동글들(묘사되지 않음)과 같은 특수화된 ROM/ASIC/PLD 기반 디바이스들 등을 또한 포함할 수 있다.
- [0181] 본 기술분야의 통상의 기술자들은 현재 개시된 주제와 관련하여 사용되는 용어 "컴퓨터 판독가능 매체(computer readable media)"가 송신 매체, 반송파들, 또는 다른 일시적 신호들을 포괄하지 않는다는 점을 또한 이해할 것이다.
- [0182] 컴퓨터 시스템(900)은 하나 이상의 통신 네트워크(955)에 대한 인터페이스(들)를 또한 포함할 수 있다. 네트워크들은 예를 들어 무선, 와이어라인, 광학일 수 있다. 네트워크들은 추가로 로컬, 광역, 대도시, 차량 및 산업, 실시간, 지연-허용(delay-tolerant) 등일 수 있다. 네트워크들의 예들은 로컬 영역 네트워크들, 이를테면 이더넷, 무선 LAN들, GSM(global systems for mobile communications), 제3 세대(3G), 제4 세대(4G), 제5 세대(5G), 롱-텀 에볼루션(LTE) 등을 포함하는 셀룰러 네트워크들, 케이블 TV, 위성 TV 및 지상파 브로드캐스트 TV를 포함하는 TV 와이어라인 또는 무선 광역 디지털 네트워크들, CANBus를 포함하는 차량 및 산업 등을 포함한다. 특정 네트워크들은 일반적으로 특정 범용 데이터 포트들 또는 주변 버스들(949)(예를 들어, 컴퓨터 시스템(900)의 범용 직렬 버스(USB) 포트들 등)에 부착된 외부 네트워크 인터페이스 어댑터들(954)을 요구하고; 다른 것들은 일반적으로 아래에 설명되는 바와 같은 시스템 버스로의 부착에 의해 컴퓨터 시스템(900)의 코어에 통합

된다(예를 들어, PC 컴퓨터 시스템 내로의 이더넷 인터페이스는 또는 스마트폰 컴퓨터 시스템 내로의 셀룰러 네트워크 인터페이스). 예로서, 네트워크(955)는 네트워크 인터페이스(954)를 사용하여 주변 버스(949)에 접속될 수 있다. 이러한 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(900)은 다른 엔티티들과 통신할 수 있다. 그러한 통신은 단방향성 수신 전용(예를 들어, 브로드캐스트 TV), 단방향성 전신 전용(예를 들어, CANbus 대 특정 CANbus 디바이스들), 또는 예를 들어 로컬 또는 광역 디지털 네트워크들을 사용하는 다른 컴퓨터 시스템들과의 양방향성일 수 있다. 위에서 설명된 바와 같은 네트워크들 및 네트워크 인터페이스들(954) 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0183] 전술한 인간 인터페이스 디바이스들, 인간-액세스가능 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(900)의 코어(940)에 부착될 수 있다.

[0184] 코어(940)는 하나 이상의 CPU(Central Processing Unit)(941), GPU(Graphics Processing Unit)(942), FPGA(Field Programmable Gate Area)(943)의 형식으로 특수화된 프로그램가능 처리 유닛들, 특정 태스크들에 대한 하드웨어 가속기들(944) 등을 포함할 수 있다. 이러한 디바이스들은, ROM(Read-only memory)(945), RAM(Random-access memory)(946), 내부 비-사용자 액세스가능 하드 드라이브들, SSD(solid-state drive)들 등과 같은 내부 대용량 저장소(947)와 함께, 시스템 버스(1248)를 통해 접속될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(1248)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형식으로 액세스가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(1248)에 직접, 또는 주변 버스(949)를 통해 부착될 수 있다. 주변 버스를 위한 아키텍처들은 PCI(peripheral component interconnect), USB 등을 포함한다.

[0185] CPU들(941), GPU들(942), FPGA들(943), 및 가속기들(944)은, 조합하여, 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령어들을 실행할 수 있다. 그러한 컴퓨터 코드는 ROM(945) 또는 RAM(946)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(946)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 저장소(947)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은, 하나 이상의 CPU(941), GPU(942), 대용량 저장소(947), ROM(945), RAM(946) 등과 밀접하게 연관될 수 있는, 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

[0186] 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 그들은 컴퓨터 소프트웨어 기술분야의 통상의 기술자들에게 잘 알려져 있고 이용가능한 종류의 것일 수 있다.

[0187] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(900), 및 구체적으로 코어(940)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형의(tangible) 컴퓨터 판독가능 매체로 구현된 소프트웨어를 실행하는 결과로서 기능을 제공할 수 있다. 그러한 컴퓨터 판독가능 매체는 위에 소개된 바와 같은 사용자-액세스가능 대용량 저장소뿐만 아니라, 코어-내부 대용량 저장소(947) 또는 ROM(945)과 같은 비일시적인 본질의 것인 코어(940)의 특정 저장소와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 그러한 디바이스들에 저장되고 코어(940)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 필요에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(940) 및 구체적으로 그 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(946)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(944))에 하드와이어링되거나 다른 방식으로 구현된 로직의 결과로서 기능을 제공할 수 있다. 소프트웨어에 대한 참조는, 적절한 경우, 로직을 포괄할 수 있고, 그 반대도 가능하다. 컴퓨터 판독가능 매체에 대한 참조는, 적절한 경우, 실행을 위한 소프트웨어를 저장하는 회로(이러한 경우 집적 회로(IC)), 또는 실행을 위한 로직을 구현하는 회로, 또는 둘 다를 포괄할 수 있다. 본 개시내용은 하드웨어 및 소프트웨어의 임의의 적합한 조합을 포괄한다.

[0188] 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

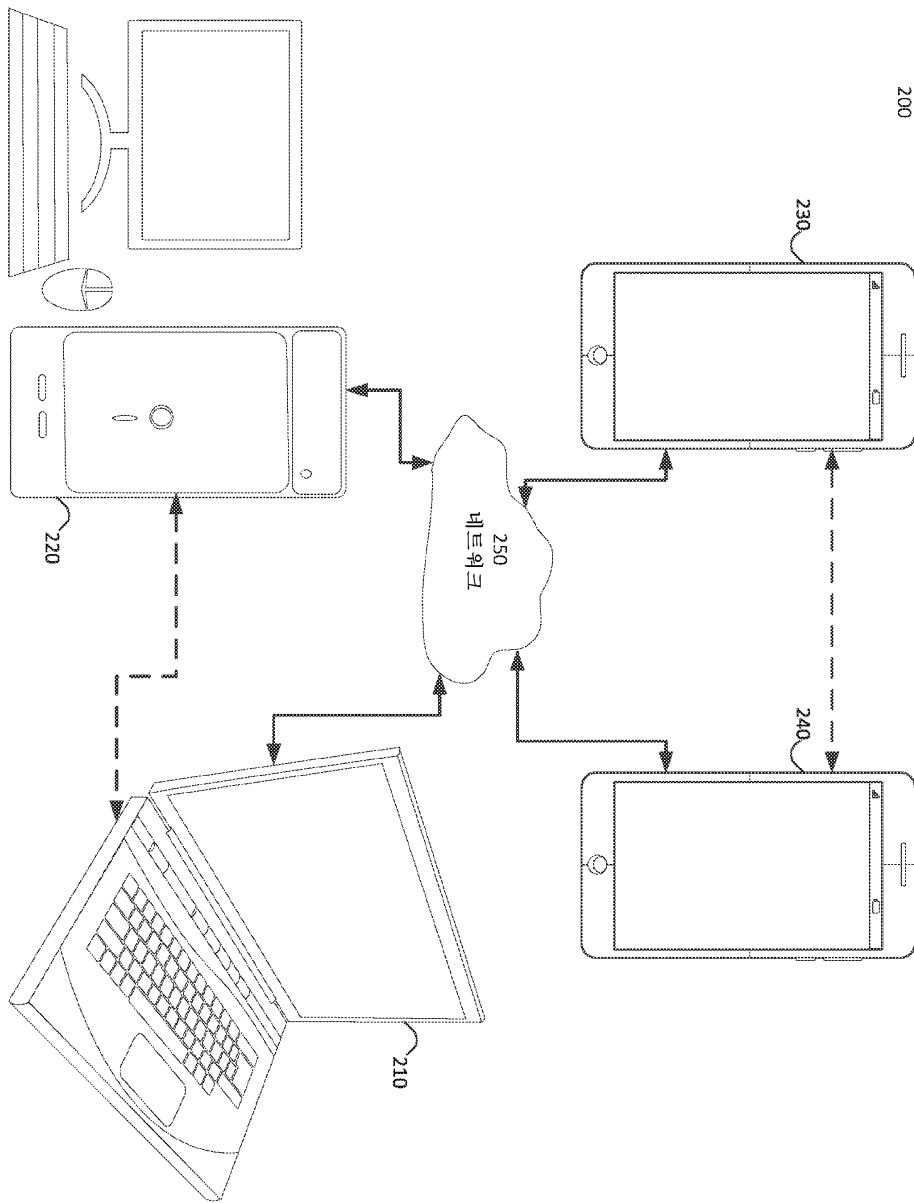
101

nal_unit(NumbaBytesInNALunit) {	C	다스크립터
forbidden_zero_bit	모두	1(1)
nal_ref_idc	모두	n(2)
nal_unit_type	모두	n(5)
...		
}		

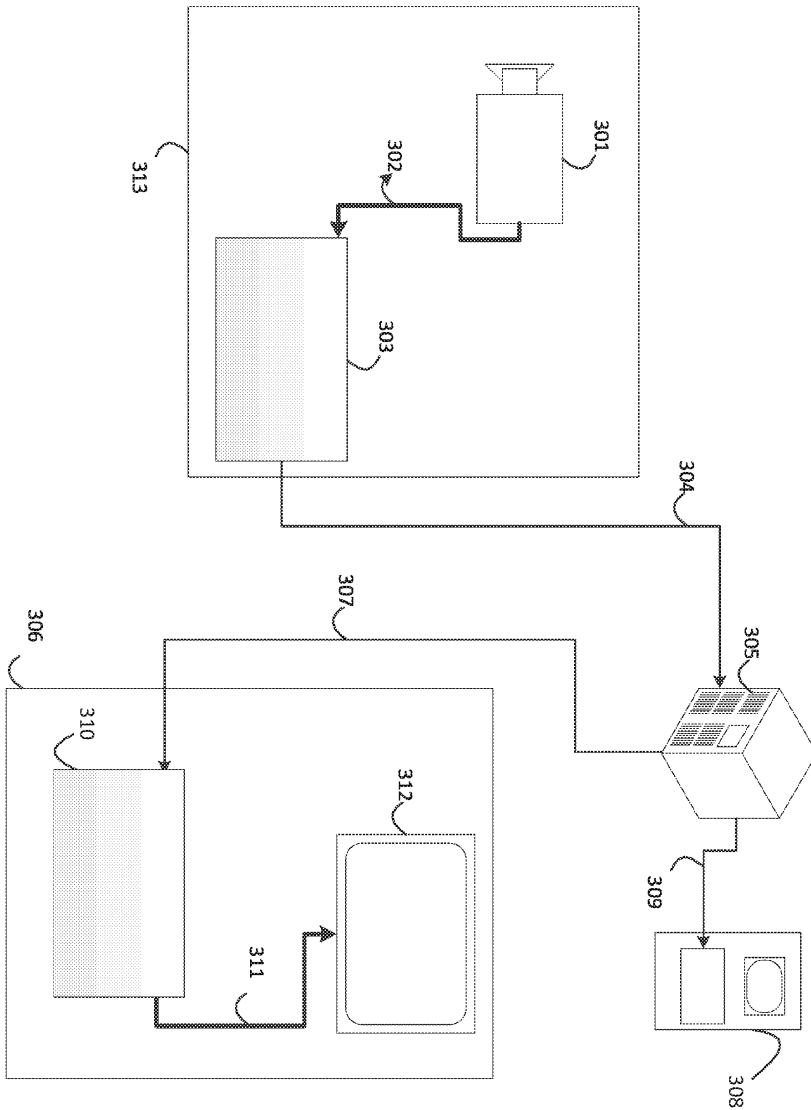
102

nal_unit_header() {	다스크립터
forbidden_zero_bit	1(1)
nal_unit_type	n(6)
nal_layer_id	n(6)
nal_temporal_id_plus1	n(3)
}	

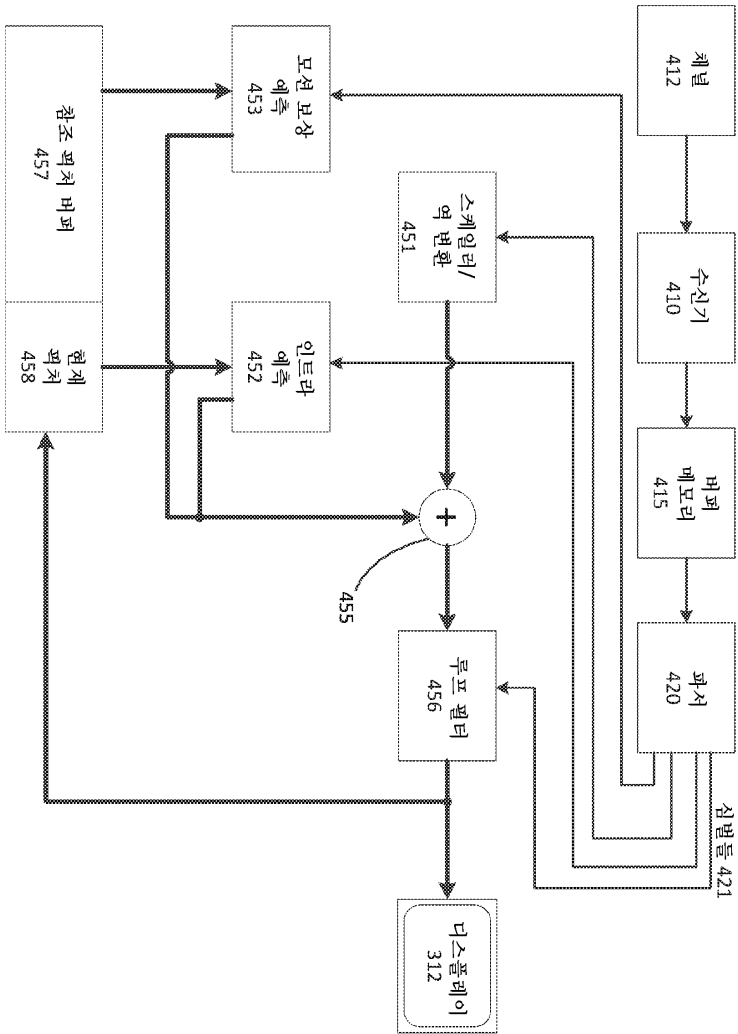
도면2



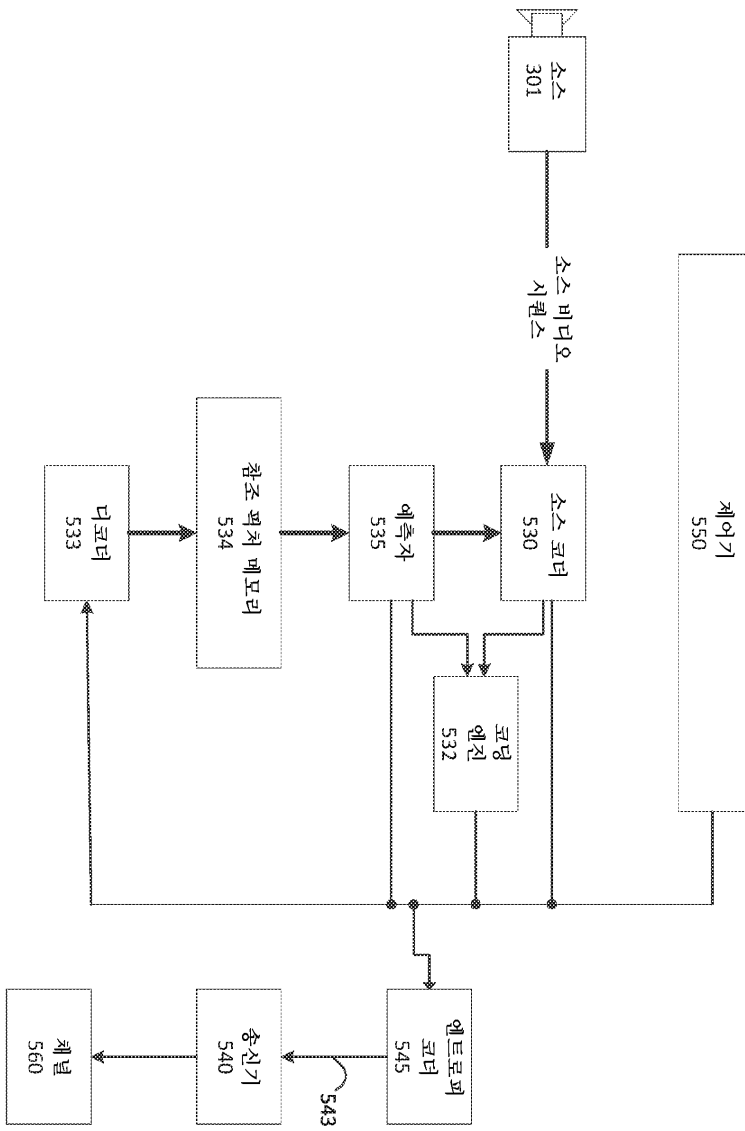
도면3



도면 4



도면5



인코더 303

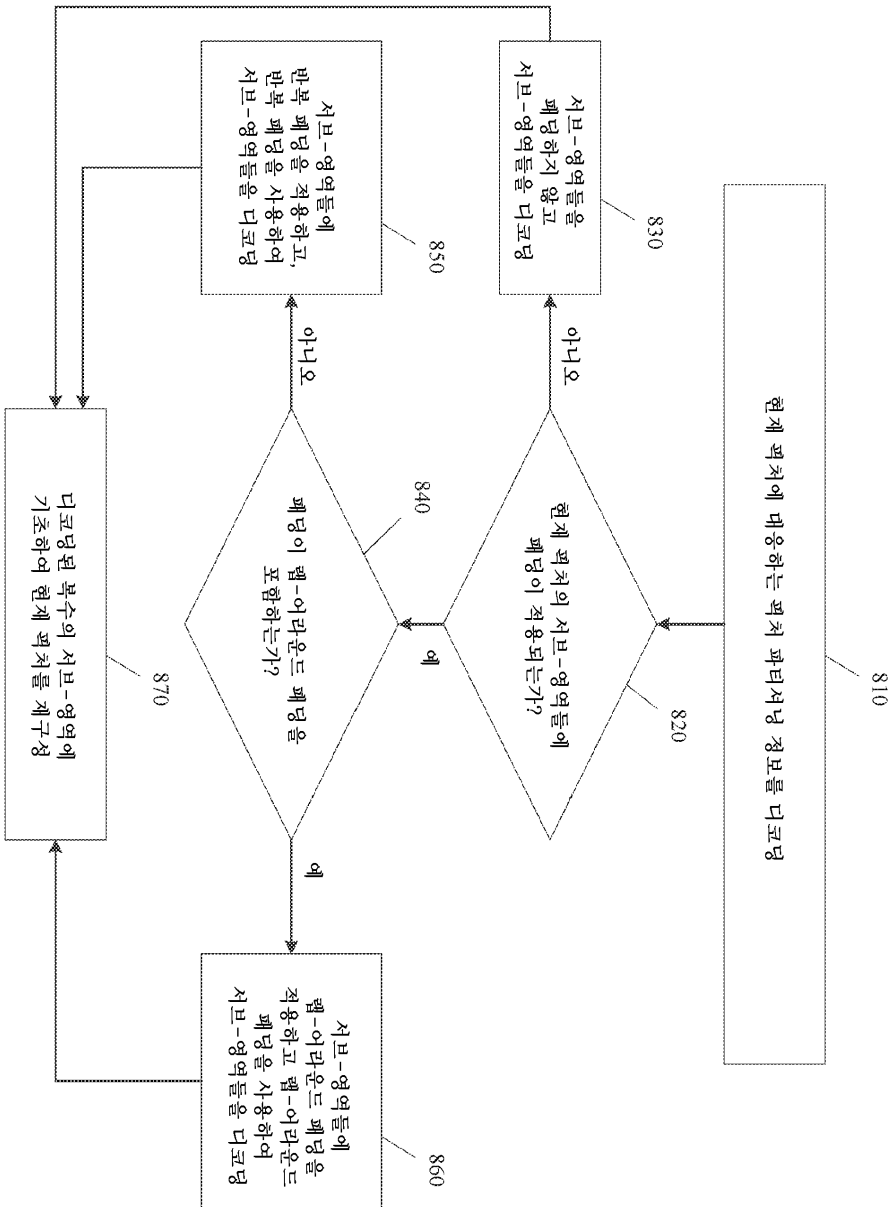
도면6

601	seq_parameter_set_top() { ... }	미스크웨어
602	sps_ref_wraparound_enabled_flag (!sps_ref_wraparound_enabled_flag)	u(1)
603	ref_wraparound_offset ...	ue(v)

도면7

701	seq_parameter_set_flag() {	다스크워드
	...	
702	sps_ref_wraparound_enabled_flag	w(1)
	if (sps_ref_wraparound_enabled_flag)	
	{	
703	left_wraparound_padding_width	ue(v)
704	right_wraparound_padding_width	ue(v)
	}	
	...}	
	}	

도면8



도면9

