



등록특허 10-2506619



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

(45) 공고일자 2023년03월07일  
(11) 등록번호 10-2506619  
(24) 등록일자 2023년02월28일

- (51) 국제특허분류(Int. Cl.)  
*HO4N 19/132* (2014.01) *HO4N 19/105* (2014.01)  
*HO4N 19/159* (2014.01) *HO4N 19/176* (2014.01)  
*HO4N 19/44* (2014.01) *HO4N 19/517* (2014.01)  
*HO4N 19/593* (2014.01) *HO4N 19/96* (2014.01)

- (52) CPC특허분류  
*HO4N 19/132* (2015.01)  
*HO4N 19/105* (2015.01)
- (21) 출원번호 10-2021-7016000

- (22) 출원일자(국제) 2020년01월06일  
 심사청구일자 2021년05월26일

- (85) 번역문제출일자 2021년05월26일

- (65) 공개번호 10-2021-0077766

- (43) 공개일자 2021년06월25일

- (86) 국제출원번호 PCT/US2020/012367

- (87) 국제공개번호 WO 2020/150026

- 국제공개일자 2020년07월23일

- (30) 우선권주장

62/792,888 2019년01월15일 미국(US)

16/533,719 2019년08월06일 미국(US)

- (56) 선행기술조사문헌

KR1020170019364 A\*

Luong Phan Van, et al.

CE8-related: Restrictions for the search area  
 of the IBC blocks in CPR, JVET-L0404-v2,  
 2018-10-02, pp. 1-6\*

\*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 20 항

심사관 : 황수진

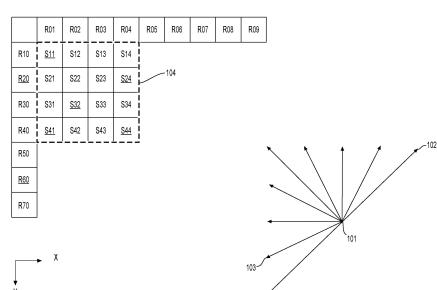
- (54) 발명의 명칭 **비디오 코딩을 위한 방법 및 장치**

**(57) 요 약**

본 개시내용의 양태들은 비디오 인코딩/디코딩을 위한 방법들 및 장치들을 제공한다. 일부 예들에서, 비디오 디코딩을 위한 장치는 수신 회로 및 처리 회로를 포함한다. 예를 들어, 처리 회로는 코딩된 비디오 비트스트림으로부터 현재 코딩 트리 유닛(coding tree unit, CTU) 내의 현재 블록의 예측 정보를 디코딩한다. 예측 정보는

(뒷면에 계속)

**대 표 도**



인트라 블록 카피 모드를 지시한다. 현재 CTU의 크기는 재구성된 샘플들을 저장하기 위한 참조 샘플 메모리의 최대 크기보다 작다. 처리 회로는 현재 블록과 동일한 픽처 내의 참조 블록을 가리키는 블록 벡터를 결정한다. 참조 블록은 참조 샘플 메모리에 버퍼링된 재구성된 샘플들을 갖는다. 그 후, 처리 회로는 참조 샘플 메모리로부터 검색되는 참조 블록의 재구성된 샘플들에 기초하여 현재 블록의 적어도 한 샘플을 재구성한다.

(52) CPC특허분류

*H04N 19/159* (2015.01)

*H04N 19/176* (2015.01)

*H04N 19/44* (2015.01)

*H04N 19/517* (2015.01)

*H04N 19/593* (2015.01)

*H04N 19/96* (2015.01)

---

## 명세서

### 청구범위

#### 청구항 1

디코더에서의 비디오 디코딩을 위한 방법으로서,

코딩된 비디오 비트스트림으로부터 현재 코딩 트리 유닛(coding tree unit, CTU) 내의 현재 블록의 예측 정보를 디코딩하는 단계- 상기 예측 정보는 인트라 블록 카피 모드를 지시하고, 상기 현재 CTU의 크기는 재구성된 샘플들을 저장하기 위한 참조 샘플 메모리의 최대 크기보다 작고, 상기 최대 크기는 상기 현재 CTU의 크기의 N배이고, N은 4 이상의 양수이고, Log2 도메인 CtbLog2SizeY에서 현재 CTU 크기가 사용될 때, 참조 샘플 메모리 버퍼에 참조 샘플 레이터를 저장할 수 있는 CTU들의 수는 MaxCtbLog2SizeY와 CtbLog2SizeY 사이의 관계에 따라 가변임 -;

상기 현재 블록과 동일한 픽처 내의 참조 블록을 가리키는 블록 벡터를 결정하는 단계- 상기 참조 블록은 상기 참조 샘플 메모리에 버퍼링된 재구성된 샘플들을 가짐 -; 및

상기 참조 샘플 메모리로부터 검색되는 상기 참조 블록의 상기 재구성된 샘플들에 기초하여 상기 현재 블록의 적어도 한 샘플(at least a sample)을 재구성하는 단계

를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

상기 현재 CTU와 동일한 CTU 행에 있고 상기 현재 CTU의 제(N-1) 좌측 CTU로부터 인접한 좌측 CTU까지의 영역에 위치하는 상기 참조 블록을 가리키는 상기 블록 벡터를 결정하는 단계를 추가로 포함하는, 방법.

#### 청구항 3

제2항에 있어서,

상기 참조 블록의 상단 경계가 상기 동일한 CTU 행에 있는지를 체크하는 단계;

상기 참조 블록의 하단 경계가 상기 동일한 CTU 행에 있는지를 체크하는 단계;

상기 참조 블록의 좌측 경계가 제N 좌측 CTU의 우측에 있는지를 체크하는 단계; 및

상기 참조 블록의 우측 경계가 상기 현재 CTU의 좌측에 있는지를 체크하는 단계를 추가로 포함하는 방법.

#### 청구항 4

제1항에 있어서,

상기 참조 블록이 상기 현재 CTU와 동일한 CTU 행에 있는 제N 좌측 CTU에 적어도 부분적으로 있는지를 체크하는 단계를 추가로 포함하는, 방법.

#### 청구항 5

제4항에 있어서,

상기 참조 블록의 좌측 경계가 상기 제N 좌측 CTU에 있는지를 체크하는 단계를 추가로 포함하는 방법.

#### 청구항 6

제4항에 있어서,

상기 참조 블록이 상기 제N 좌측 CTU에 적어도 부분적으로 있을 때, 상기 현재 CTU에서, 상기 참조 블록의 병치된 블록이 적어도 부분적으로 재구성되는지를 결정하는 단계를 추가로 포함하는 방법.

**청구항 7**

제6항에 있어서,

상기 병치된 블록의 좌측-상단 코너가 재구성되었는지를 결정하는 단계를 추가로 포함하는 방법.

**청구항 8**

제6항에 있어서,

상기 현재 CTU 내의 상기 병치된 블록이 적어도 부분적으로 재구성될 때 상기 참조 블록을 가리키는 상기 블록 벡터를 무효화하는 단계를 추가로 포함하는 방법.

**청구항 9**

제4항에 있어서,

상기 참조 블록을 포함하는 상기 제N 좌측 CTU에서 참조 블록 영역을 결정하는 단계;

상기 현재 CTU에서, 상기 참조 블록 영역의 병치된 블록 영역이 적어도 부분적으로 재구성되었는지를 결정하는 단계; 및

상기 현재 CTU 내의 상기 병치된 블록 영역이 적어도 부분적으로 재구성될 때 상기 참조 블록을 가리키는 상기 블록 벡터를 무효화하는 단계를 추가로 포함하는 방법.

**청구항 10**

비디오 디코딩을 위한 장치로서,

처리 회로를 포함하고, 상기 처리 회로는:

코딩된 비디오 비트스트림으로부터 현재 코딩 트리 유닛(CTU) 내의 현재 블록의 예측 정보를 디코딩하고- 상기 예측 정보는 인트라 블록 카피 모드를 지시하고, 상기 현재 CTU의 크기는 재구성된 샘플들을 저장하기 위한 참조 샘플 메모리의 최대 크기보다 작고, 상기 최대 크기는 상기 현재 CTU의 크기의 N배이고, N은 4 이상의 양수이고, Log2 도메인 CtbLog2SizeY에서 현재 CTU 크기가 사용될 때, 참조 샘플 메모리 버퍼에 참조 샘플 데이터를 저장할 수 있는 CTU들의 수는 MaxCtbLog2SizeY와 CtbLog2SizeY 사이의 관계에 따라 가변임 -;

상기 현재 블록과 동일한 꽉쳐 내의 참조 블록을 가리키는 블록 벡터를 결정하고- 상기 참조 블록은 상기 참조 샘플 메모리에 버퍼링된 재구성된 샘플들을 가짐 -;

상기 참조 샘플 메모리로부터 검색되는 상기 참조 블록의 상기 재구성된 샘플들에 기초하여 상기 현재 블록의 적어도 한 샘플을 재구성하도록 구성되는 장치.

**청구항 11**

제10항에 있어서, 상기 처리 회로는 추가로:

상기 현재 CTU와 동일한 CTU 행에 있고 상기 현재 CTU의 제(N-1) 좌측 CTU로부터 인접한 좌측 CTU까지의 영역에 위치하는 상기 참조 블록을 가리키는 상기 블록 벡터를 결정하도록 구성되는, 장치.

**청구항 12**

제11항에 있어서, 상기 처리 회로는 추가로:

상기 참조 블록의 상단 경계가 상기 동일한 CTU 행에 있는지를 체크하고;

상기 참조 블록의 하단 경계가 상기 동일한 CTU 행에 있는지를 체크하고;

상기 참조 블록의 좌측 경계가 제N 좌측 CTU의 우측에 있는지를 체크하고;

상기 참조 블록의 우측 경계가 상기 현재 CTU의 좌측에 있는지를 체크하도록 구성되는 장치.

**청구항 13**

제10항에 있어서, 상기 처리 회로는 추가로:

상기 참조 블록이 상기 현재 CTU와 동일한 CTU 행에 있는 제N 좌측 CTU에 적어도 부분적으로 있는지를 체크하도록 구성되는, 장치.

#### 청구항 14

제13항에 있어서, 상기 처리 회로는 추가로:

상기 참조 블록의 좌측 경계가 상기 제N 좌측 CTU에 있는지를 체크하도록 구성되는 장치.

#### 청구항 15

제13항에 있어서, 상기 처리 회로는 추가로:

상기 참조 블록이 상기 제N 좌측 CTU에 적어도 부분적으로 있을 때, 상기 현재 CTU에서, 상기 참조 블록의 병치된 블록이 적어도 부분적으로 재구성되는지를 결정하도록 구성되는 장치.

#### 청구항 16

제15항에 있어서, 상기 처리 회로는 추가로:

상기 병치된 블록의 좌측-상단 코너가 재구성되었는지를 결정하도록 구성되는 장치.

#### 청구항 17

제15항에 있어서, 상기 처리 회로는 추가로:

상기 현재 CTU 내의 상기 병치된 블록이 적어도 부분적으로 재구성될 때 상기 참조 블록을 가리키는 상기 블록 백터를 무효화하도록 구성되는 장치.

#### 청구항 18

제13항에 있어서, 상기 처리 회로는 추가로:

상기 참조 블록을 포함하는 상기 제N 좌측 CTU에서 참조 블록 영역을 결정하고;

상기 현재 CTU에서, 상기 참조 블록 영역의 병치된 블록 영역이 적어도 부분적으로 재구성되는지를 결정하고;

상기 현재 CTU 내의 상기 병치된 블록 영역이 적어도 부분적으로 재구성될 때 상기 참조 블록을 가리키는 상기 블록 백터를 무효화하도록 구성되는 장치.

#### 청구항 19

명령어들을 저장하는 비일시적 컴퓨터 관독가능 매체로서,

상기 명령어들은 비디오 디코딩을 위해 컴퓨터에 의해 실행될 때, 상기 컴퓨터로 하여금:

코딩된 비디오 비트스트림으로부터 현재 코딩 트리 유닛(CTU) 내의 현재 블록의 예측 정보를 디코딩하는 것- 상기 현재 CTU의 크기는 재구성된 샘플들을 저장하기 위한 참조 샘플 메모리의 최대 크기보다 작고, 상기 최대 크기는 상기 현재 CTU의 크기의 N배이고, N은 4 이상의 양수이고, Log2 도메인 CtbLog2SizeY에서 현재 CTU 크기가 사용될 때, 참조 샘플 메모리 버퍼에 참조 샘플 데이터를 저장할 수 있는 CTU들의 수는 MaxCtbLog2SizeY와 CtbLog2SizeY 사이의 관계에 따라 가변이고, 상기 예측 정보는 인트라 블록 카피 모드를 지시함 -;

상기 현재 블록과 동일한 픽처 내의 참조 블록을 가리키는 블록 백터를 결정하는 것- 상기 참조 블록은 상기 참조 샘플 메모리에 버퍼링된 재구성된 샘플들을 가짐 -; 및

상기 참조 샘플 메모리로부터 검색되는 상기 참조 블록의 상기 재구성된 샘플들에 기초하여 상기 블록의 적어도 한 샘플을 재구성하는 것을 수행하게 하는 비일시적 컴퓨터 관독가능 매체.

#### 청구항 20

제19항에 있어서, 상기 명령어들은 추가로 상기 컴퓨터로 하여금:

상기 현재 CTU와 동일한 CTU 행에 있고 상기 현재 CTU의 제(N-1) 좌측 CTU로부터 인접한 좌측 CTU까지의 영역에 위치하는 상기 참조 블록을 가리키는 상기 블록 벡터를 결정하는 것을 수행하게 하는, 비일시적 컴퓨터 판독가능 매체.

### 발명의 설명

#### 기술 분야

[0001] 본 출원은, 2019년 1월 15일자로 출원된 미국 가출원 제62/792,888호, "SEARCH RANGE ADJUSTMENT WITH VARIABLE CTU SIZE FOR INTRA PICTURE BLOCK COMPENSATION"에 대한 우선권의 이익을 주장하는, 2019년 8월 6일자로 출원된 미국 특허 출원 제16/533,719호, "METHOD AND APPARATUS FOR VIDEO CODING"의 우선권의 이익을 주장한다. 앞선 출원들의 전체 개시내용들은 그 전체가 참조로서 포함된다.

[0002] 본 개시내용은 일반적으로 비디오 코딩에 관련된 실시예들을 설명한다.

#### 배경기술

[0003] 본 명세서에 제공되는 배경기술 설명은, 본 개시내용의 맥락을 일반적으로 제시하기 위한 것이다. 현재 호명된 발명자들의 연구 - 그 연구가 이 배경기술 부분에서 설명되는 한 - 뿐만 아니라 출원 시에 종래 기술로서의자격이 없을 수 있는 설명의 양태들은 명백하게도 또는 암시적으로도 본 개시내용에 대한 종래 기술로서 인정되지 않는다.

[0004] 비디오 코딩 및 디코딩은 모션 보상을 갖는 인터-픽처 예측(inter-picture prediction)을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처들을 포함할 수 있고, 각각의 픽처는, 예를 들어, 1920x1080 루미넌스 샘플들 및 연관된 크로미넌스 샘플들의 공간 차원을 갖는다. 이 일련의 픽처들은, 예를 들어, 초당 60개 픽처 또는 60Hz의, 고정된 또는 가변 픽처 레이트(비공식적으로 프레임 레이트로도 알려져 있음)를 가질 수 있다. 압축되지 않은 비디오는 상당한 비트레이트 요건들을 갖는다. 예를 들어, 샘플당 8비트에서의 1080p60 4:2:0 비디오(60 Hz 프레임 레이트에서의 1920x1080 루미넌스 샘플 해상도)는 1.5Gbit/s 대역폭에 가까울 것을 요구한다. 그러한 비디오의 시간은 600기가바이트보다 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 카피가 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 사용된다. 용인되는 왜곡의 양은 애플리케이션에 의존하는데; 예를 들어, 특정 소비자 스트리밍 애플리케이션들의 사용자들은 텔레비전 배포 애플리케이션들의 사용자들보다 더 높은 왜곡을 용인할 수 있다. 달성 가능한 압축비는: 더 높은 허용 가능한/용인 가능한 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 포함하는, 수개의 넓은 카테고리로부터의 기법들을 활용할 수 있다.

[0007] 비디오 코덱 기술들은 인트라 코딩으로 알려진 기법들을 포함할 수 있다. 인트라 코딩에서, 샘플 값들은 이전에 재구성된 참조 픽처들로부터의 샘플들 또는 다른 데이터를 참조하지 않고 표현된다. 일부 비디오 코덱들에서, 픽처는 샘플들의 블록들로 공간적으로 세분된다. 샘플들의 모든 블록들이 인트라 모드에서 코딩될 때, 그 픽처는 인트라 픽처일 수 있다. 인트라 픽처들 및 독립적인 디코더 리프레시 픽처들과 같은 그들의 파생들은 디코더 상태를 리셋하기 위해 사용될 수 있고, 따라서 코딩된 비디오 비트스트림 및 비디오 세션에서의 제1 픽처로서 또는 정지 이미지로서 사용될 수 있다. 인트라 블록의 샘플들은 변환에 노출될 수 있고, 변환 계수들은 엔트로피 코딩 전에 양자화될 수 있다. 인트라 예측은 사전 변환 도메인에서 샘플 값들을 최소화하는 기법일 수 있다. 일부 경우들에서, 변환 후의 DC 값이 더 작을수록, 그리고 AC 계수들이 더 작을수록, 엔트로피 코딩 후에 블록을 표현하기 위해 주어진 양자화 스텝 사이즈에서 요구되는 비트들이 더 적다.

[0008] 예를 들면, MPEG-2세대 코딩 기술들로부터 알려진 것과 같은 전통적인 인트라 코딩은 인트라 예측을 사용하지 않는다. 그러나, 일부 더 새로운 비디오 압축 기술들은, 예를 들어, 데이터의 공간적으로 이웃하는, 그리고 디코딩 순서에서 선행하는 블록들의 인코딩/디코딩 동안 획득된 주위의 샘플 데이터 및/또는 메타데이터로부터 시

도하는 기법들을 포함한다. 이러한 기법들은 이후 "인트라 예측" 기법들로 불린다. 적어도 일부 경우들에서, 인트라 예측은 참조 픽처들로부터가 아니라 재구성 하의 현재 픽처로부터의 참조 데이터만을 사용한다는 점에 유의한다.

[0009] 많은 상이한 형태들의 인트라 예측이 있을 수 있다. 주어진 비디오 코딩 기술에서 그러한 기법들 중 하나보다 많은 기법이 사용될 수 있을 때, 사용 중인 기법은 인트라 예측 모드에서 코딩될 수 있다. 특정 경우들에서, 모드들은 서브모드들 및/또는 파라미터들을 가질 수 있고, 이들은 개별적으로 코딩되거나 모드 코드워드에 포함될 수 있다. 주어진 모드/서브모드/파라미터 조합에 사용할 코드워드는 인트라 예측을 통해 코딩 효율 이득에 영향을 미칠 수 있고, 따라서 코드워드들을 비트스트림으로 변환하기 위해 사용되는 엔트로피 코딩 기술에 영향을 미칠 수 있다.

[0010] 인트라 예측의 특정 모드가 H.264로 도입되었고, H.265에서 정제되고, JEM(joint exploration model), VVC(versatile video coding), 및 BMS(benchmark set)와 같은 더 새로운 코딩 기술들에서 추가로 정제되었다. 예측자 블록은 이미 이용가능한 샘플들에 속하는 이웃 샘플 값들을 사용하여 형성될 수 있다. 이웃 샘플들의 샘플 값들은 방향에 따라 예측자 블록 내로 카페된다. 사용 중인 방향에 대한 참조는 비트스트림에서 코딩될 수 있거나, 그 자체로 예측될 수 있다.

[0011] 도 1을 참조하면, 우측 아래 도시된 것은 (35개의 인트라 모드의 33개의 각도 모드들에 대응하는) H.265의 33개의 가능한 예측자 방향으로부터 알려진 9개의 예측자 방향의 서브세트이다. 화살표들이 수렴하는 포인트(101)는 예측되는 샘플을 나타낸다. 화살표들은 샘플이 예측되고 있는 방향을 나타낸다. 예를 들어, 화살표(102)는 샘플(101)이 하나의 샘플 또는 샘플들로부터 우측 상부로, 수평으로부터 45도 각도로 예측되는 것을 나타낸다. 유사하게, 화살표(103)는 샘플(101)이 하나의 샘플 또는 샘플들로부터 샘플(101)의 좌측 하부로, 수평으로부터 22.5도 각도로 예측되는 것을 나타낸다.

[0012] 계속 도 1을 참조하면, 좌측 상부에서, 4x4 샘플의 정사각형 블록(104)(파선, 굵은 라인으로 지시됨)이 도시된다. 정사각형 블록(104)은 16개의 샘플을 포함하고, 각각은 "S", Y 차원에서의 그의 위치(예를 들어, 행 인덱스) 및 X 차원에서의 그의 위치(예를 들어, 열 인덱스)로 라벨링되어 있다. 예를 들어, 샘플 S21은 Y 차원에서의 (상단으로부터) 2번째 샘플 및 X 차원에서의 (좌측으로부터) 1번째 샘플이다. 유사하게, 샘플 S44는 Y 및 X 차원 둘 다에서 블록(104) 내의 네 번째 샘플이다. 블록의 크기가 4x4 샘플이므로, S44는 우측 하단에 있다. 유사한 넘버링 스킁을 따르는 참조 샘플들이 추가로 도시되어 있다. 참조 샘플은 블록(104)에 대한 R, 그의 Y 위치(예를 들어, 행 인덱스) 및 X 위치(열 인덱스)로 라벨링된다. H.264 및 H.265 둘 다에서, 예측 샘플들은 재구성 하의 블록에 이웃하는데; 그러므로, 음의 값들이 사용될 필요가 없다.

[0013] 인트라 픽처 예측은 시그널링된 예측 방향에 의해 적절하게 이웃 샘플들로부터 참조 샘플 값들을 카페함으로써 작동할 수 있다. 예를 들어, 코딩된 비디오 비트스트림은, 이 블록에 대해, 화살표(102)와 일치하는 예측 방향을 지시하는 시그널링을 포함한다고 가정한다 - 즉, 샘플들이 예측 샘플 또는 샘플들로부터 우측 상부로, 수평으로부터 45도 각도로 예측됨. 그 경우, 샘플들 S41, S32, S23 및 S14가 동일한 참조 샘플 R05로부터 예측된다. 그 후 샘플 S44가 참조 샘플 R08로부터 예측된다.

[0014] 특정 경우들에서, 다수의 참조 샘플의 값들은, 참조 샘플을 계산하기 위해, 예를 들어, 보간을 통해 조합될 수 있다; 특히, 방향들이 45도씩 균등하게 나누어지지 않을 때.

[0015] 가능한 방향들의 수는 비디오 코딩 기술이 개발됨에 따라 증가하였다. H.264(2003년)에서, 9개의 상이한 방향이 표현될 수 있었다. H.265(2013년)에서 33으로 증가되었고, 본 개시내용의 시점에서 JEM/VVC/BMS는 최대 65개의 방향을 지원할 수 있다. 가능성성이 높은 방향들을 식별하기 위한 실험들이 수행되었고, 엔트로피 코딩의 특정 기법들은 가능성이 낮은 방향에 대한 특정 페널티를 수용하여 가능성이 있는 방향들을 적은 수의 비트로 표현하는 데 사용된다. 또한, 방향들 자체는 때때로 이웃하는 이미 디코딩된 블록들에서 사용되는 이웃하는 방향들로부터 예측될 수 있다.

[0016] 도 2는 시간 경과에 따라 증가하는 예측 방향들의 수를 예시하기 위해 JEM에 따라 65개의 인트라 예측 방향을 묘사하는 개략도(201)를 도시한다.

[0017] 방향을 표현하는 코딩된 비디오 비트스트림 내의 인트라 예측 방향 비트들의 매핑은 비디오 코딩 기술마다 상이할 수 있고; 예를 들어, 예측 방향의 단순한 직접 매핑들에서 인트라 예측 모드, 코드워드들, 최고 확률 모드들을 수반하는 복잡한 적응 스킁들 및 유사한 기법들에 이르기까지 다양할 수 있다. 그러나, 모든 경우에, 특정한 다른 방향들보다 비디오 콘텐츠에서 통계적으로 발생할 가능성이 낮은 특정 방향들이 있을 수 있다. 비디오

압축의 목표는 중복성의 감소이기 때문에, 잘 작동하는 비디오 코딩 기술에서, 그러한 가능성이 낮은 방향들은 가능성이 더 높은 방향들보다 더 많은 수의 비트로 표현될 것이다.

### 발명의 내용

[0018]

본 개시내용의 양태들은 비디오 인코딩/디코딩을 위한 방법들 및 장치들을 제공한다. 일부 예들에서, 비디오 디코딩을 위한 장치는 수신 회로 및 처리 회로를 포함한다. 예를 들어, 처리 회로는 코딩된 비디오 비트스트림으로부터 현재 코딩 트리 유닛(coding tree unit, CTU) 내의 현재 블록의 예측 정보를 디코딩한다. 예측 정보는 인트라 블록 카피 모드를 지시한다. 현재 CTU의 크기는 재구성된 샘플들을 저장하기 위한 참조 샘플 메모리의 최대 크기보다 작다. 처리 회로는 현재 블록과 동일한 퍽처 내의 참조 블록을 가리키는 블록 벡터를 결정한다. 참조 블록은 참조 샘플 메모리에 버퍼링된 재구성된 샘플들을 갖는다. 그 후, 처리 회로는 참조 샘플 메모리로부터 검색되는 참조 블록의 재구성된 샘플들에 기초하여 현재 블록의 적어도 한 샘플(at least a sample)을 재구성한다.

[0019]

일부 실시예들에서, 처리 회로는 현재 CTU와 동일한 CTU 행에 있고 현재 CTU의 제(N-1) 좌측 CTU로부터 인접한 좌측 CTU까지의 영역에 위치하는 참조 블록을 가리키는 블록 벡터를 결정하고, 참조 샘플 메모리의 최대 크기는 현재 CTU의 크기의 N배이고, N은 일 예에서 1보다 큰 양의 정수이다.

[0020]

일부 예들에서, 처리 회로는 참조 블록의 상단 경계가 동일한 CTU 행에 있는지를 체크한다. 또한, 처리 회로는 참조 블록의 하단 경계가 동일한 CTU 행에 있는지를 체크한다. 그 후, 처리 회로는 참조 블록의 좌측 경계가 제N 좌측 CTU의 우측에 있고, 참조 블록의 우측 경계가 현재 CTU의 좌측에 있는지를 체크한다.

[0021]

일부 실시예들에서, 처리 회로는 참조 블록이 현재 CTU와 동일한 CTU 행에 있는 제N 좌측 CTU에 적어도 부분적으로 있는지를 체크하고, 참조 샘플 메모리의 최대 크기는 현재 CTU의 크기의 N배이고, N은 1보다 큰 양수이다.

[0022]

또한, 처리 회로는 참조 블록의 좌측 경계가 제N 좌측 CTU에 있는지를 체크한다. 참조 블록이 제N 좌측 CTU에 적어도 부분적으로 있을 때, 처리 회로는 현재 CTU에서, 참조 블록의 병치된 블록이 적어도 부분적으로 재구성되는지를 결정한다. 일 예에서, 처리 회로는 병치된 블록의 좌측-상단 코너가 재구성되었는지를 결정한다. 처리 회로는 현재 CTU 내의 병치된 블록이 적어도 부분적으로 재구성될 때 참조 블록을 가리키는 블록 벡터를 무효화한다.

[0023]

일부 예들에서, 처리 회로는 참조 블록을 포함하는 제N 좌측 CTU 내의 참조 블록 영역을 결정하고, 현재 CTU 내에서, 참조 블록 영역의 병치된 블록 영역이 적어도 부분적으로 재구성되는지를 결정한다. 그 후, 처리 회로는 현재 CTU 내의 병치된 블록 영역이 적어도 부분적으로 재구성될 때 참조 블록을 가리키는 블록 벡터를 무효화한다.

[0024]

본 개시내용의 양태들은 또한, 비디오 디코딩을 위해 컴퓨터에 의해 실행될 때, 컴퓨터로 하여금 비디오 디코딩을 위한 방법을 수행하게 하는 명령어들을 저장하는 비일시적 컴퓨터 판독가능 매체를 제공한다.

### 도면의 간단한 설명

[0025]

개시된 주제의 추가의 특징들, 본질 및 다양한 이점들이 다음의 상세한 설명 및 첨부 도면들로부터 더 명백할 것이다:

도 1은 인트라 예측 모드들의 예시적인 서브세트의 개략 예시이다.

도 2는 예시적인 인트라 예측 방향들의 예시이다.

도 3은 일 실시예에 따른 통신 시스템(300)의 단순화된 블록도의 개략 예시이다.

도 4는 일 실시예에 따른 통신 시스템(400)의 단순화된 블록도의 개략 예시이다.

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

도 6은 일 실시예에 따른 인코더의 단순화된 블록도의 개략 예시이다.

도 7은 다른 실시예에 따른 인코더의 블록도를 도시한다.

도 8은 다른 실시예에 따른 디코더의 블록도를 도시한다.

도 9는 본 개시내용의 일 실시예에 따른 인트라 블록 카피의 예를 도시한다.

도 10a 내지 도 10d는 본 개시내용의 일 실시예에 따른 인트라 블록 카피 모드에 대한 유효 검색 범위들의 예들을 도시한다.

도 11은 본 개시내용의 일부 실시예들에 따른 병치된 블록들의 예들을 도시한다.

도 12는 본 개시내용의 일부 실시예들에 따른 병치된 블록들의 예들을 도시한다.

도 13은 본 개시내용의 일부 실시예들에 따른 프로세스 예를 약술하는 흐름도를 도시한다.

도 14는 일 실시예에 따른 컴퓨터 시스템의 개략 예시이다.

### 발명을 실시하기 위한 구체적인 내용

[0026] 도 3은 본 개시내용의 일 실시예에 따른 통신 시스템(300)의 단순화된 블록도를 예시한다. 통신 시스템(300)은, 예를 들어, 네트워크(350)를 통해, 서로 통신할 수 있는 복수의 단말 디바이스를 포함한다. 예를 들어, 통신 시스템(300)은 네트워크(350)를 통해 상호접속되는 제1 쌍의 단말 디바이스들(310 및 320)을 포함한다. 도 3의 예에서, 제1 쌍의 단말 디바이스들(310 및 320)은 데이터의 단방향 송신을 수행한다. 예를 들어, 단말 디바이스(310)는 네트워크(350)를 통해 다른 단말 디바이스(320)로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스(310)에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림의 형식으로 송신될 수 있다. 단말 디바이스(320)는 네트워크(350)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구하고 복구된 비디오 데이터에 따라 비디오 픽처들을 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 애플리케이션들 등에서 일반적일 수 있다.

[0027] 다른 예에서, 통신 시스템(300)은, 예를 들어, 영상회의 동안 발생할 수 있는 코딩된 비디오 데이터의 양방향 송신을 수행하는 제2 쌍의 단말 디바이스들(330 및 340)을 포함한다. 데이터의 양방향 송신을 위해, 일 예에서, 단말 디바이스들(330 및 340) 중의 각각의 단말 디바이스는 네트워크(350)를 통해 단말 디바이스들(330 및 340) 중의 다른 단말 디바이스로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 단말 디바이스들(330 및 340) 중의 각각의 단말 디바이스는 또한 단말 디바이스들(330 및 340) 중의 다른 단말 디바이스에 의해 송신된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구할 수 있고, 복구된 비디오 데이터에 따라 액세스 가능한 디스플레이 디바이스에서 비디오 픽처들을 디스플레이할 수 있다.

[0028] 도 3의 예에서, 단말 디바이스들(310, 320, 330 및 340)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 애플리케이션을 찾는다. 네트워크(350)는 예를 들어 와이어라인(유선) 및/또는 무선 통신 네트워크들을 포함하여, 단말 디바이스들(310, 320, 330 및 340) 사이에서 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크들을 표현한다. 통신 네트워크(350)는 회로-스위칭 및/또는 패킷-스위칭 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 전기 통신 네트워크들, 로컬 영역 네트워크들, 광역 네트워크들 및/또는 인터넷을 포함한다. 본 논의의 목적을 위해, 네트워크(350)의 아키텍처 및 토플로지는 아래에서 본 명세서에서 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.

[0029] 도 4는, 개시된 주제를 위한 애플리케이션에 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 애플리케이션들에 동등하게 적용가능할 수 있다.

[0030] 스트리밍 시스템은, 예를 들어 압축되지 않은 비디오 픽처들의 스트림(402)을 생성하는 비디오 소스(401), 예를 들어 디지털 카메라를 포함할 수 있는 캡처 서브시스템(413)을 포함할 수 있다. 일 예에서, 비디오 픽처들의 스트림(402)은 디지털 카메라에 의해 취해지는 샘플들을 포함한다. 인코딩된 비디오 데이터(404)(또는 코딩된 비디오 비트스트림들)와 비교할 때 많은 데이터 용량을 강조하기 위해 짧은 라인으로 묘사된 비디오 픽처들의 스트림(402)은 비디오 소스(401)에 결합된 비디오 인코더(403)를 포함하는 전자 디바이스(420)에 의해 처리될 수 있다. 비디오 인코더(403)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 양태들을 가능하게 하거나 구현하기 위해 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 비디오 픽처들의 스트림(402)과 비교할 때 적은 데이터 용량을 강조하기 위해 짧은 라인으로서 묘사된 인코딩된 비디오 데이터(404)(또는

인코딩된 비디오 비트스트림(404)는 미래의 사용을 위해 스트리밍 서버(405) 상에 저장될 수 있다. 도 4에서의 클라이언트 서브시스템들(406 및 408)과 같은 하나 이상의 스트리밍 클라이언트 서브시스템들은 스트리밍 서버(405)에 액세스하여 인코딩된 비디오 데이터(404)의 카피들(407 및 409)을 검색할 수 있다. 클라이언트 서브시스템(406)은, 예를 들어, 전자 디바이스(430) 내에 비디오 디코더(410)를 포함할 수 있다. 비디오 디코더(410)는 인코딩된 비디오 데이터의 착신 카피(407)를 디코딩하고 디스플레이(412)(예를 들어, 디스플레이 스크린) 또는 다른 렌더링 디바이스(묘사되지 않음) 상에 렌더링될 수 있는 비디오 픽처들의 발신 스트림(411)을 생성한다. 일부 스트리밍 시스템들에서, 인코딩된 비디오 데이터(404, 407, 및 409)(예를 들어, 비디오 비트스트림들)는 특정 비디오 코딩/압축 표준들에 따라 인코딩될 수 있다. 이러한 표준들의 예들은 ITU-T 권고안(Recommendation) H.265를 포함한다. 일 예에서, 개발 하의 비디오 코딩 표준은 VVC(Versatile Video Coding)로서 널리 알려져 있다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.

[0031] 전자 디바이스들(420 및 430)은 다른 컴포넌트들(도시되지 않음)을 포함할 수 있다는 점에 유의한다. 예를 들어, 전자 디바이스(420)는 비디오 디코더(도시되지 않음)도 포함할 수 있고 전자 디바이스(430)는 비디오 인코더(도시되지 않음)도 포함할 수 있다.

[0032] 도 5은 본 개시내용의 일 실시예에 따른 비디오 디코더(510)의 블록도를 도시한다. 비디오 디코더(510)는 전자 디바이스(530)에 포함될 수 있다. 전자 디바이스(530)는 수신기(531)(예를 들어, 수신 회로)를 포함할 수 있다. 비디오 디코더(510)는 도 4의 예에서의 비디오 디코더(410) 대신에 사용될 수 있다.

[0033] 수신기(531)는 비디오 디코더(510)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스를 수신할 수 있는데; 동일한 또는 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스를 수신할 수 있다 - 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(501)로부터 수신될 수 있다. 수신기(531)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(531)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터를 방지하기 위해, 수신기(531)와 엔트로피 디코더/파서(520)(이후 "파서(520)") 사이에 베퍼 메모리(515)가 결합될 수 있다. 특정 애플리케이션들에서, 베퍼 메모리(515)는 비디오 디코더(510)의 일부이다. 다른 것들에서, 그것은 비디오 디코더(510)(묘사되지 않음) 외부에 있을 수 있다. 또 다른 것들에서, 예를 들어 네트워크 지터를 방지하기 위해, 비디오 디코더(510) 외부의 베퍼 메모리(묘사되지 않음), 그리고 추가로, 예를 들어 재생 타이밍을 핸들링하기 위해, 비디오 디코더(510) 내부의 다른 베퍼 메모리(515)가 존재할 수 있다. 수신기(531)가 충분한 대역폭 및 제어가능성의 저장/포워드 디바이스로부터, 또는 동기식 네트워크(isynchronous network)로부터 데이터를 수신하고 있을 때, 베퍼 메모리(515)는 필요하지 않을 수 있거나, 작을 수 있다. 인터넷과 같은 최선 노력 패킷 네트워크들 상에서의 사용을 위해, 베퍼 메모리(515)는 요구될 수 있고, 비교적 클 수 있고, 유리하게는 적응적 크기일 수 있고, 비디오 디코더(510) 외부의 운영 체제 또는 유사한 요소들(묘사되지 않음)에서 적어도 부분적으로 구현될 수 있다.

[0034] 비디오 디코더(510)는 코딩된 비디오 시퀀스로부터 심별들(521)을 재구성하기 위해 파서(520)를 포함할 수 있다. 이러한 심별들의 카테고리들은 비디오 디코더(510)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 5에 도시된 바와 같이, 전자 디바이스(530)의 일체 부분(integral part)은 아니지만 전자 디바이스(530)에 결합될 수 있는 렌더링 디바이스(512)(예를 들어, 디스플레이 스크린)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함한다. 렌더링 디바이스(들)에 대한 제어 정보는 SEI 메시지(Supplemental Enhancement Information) 또는 VUI(Video Usability Information) 파라미터 세트 프래그먼트들(묘사되지 않음)의 형식일 수 있다. 파서(520)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, 맥락 민감성(context sensitivity)을 갖거나 갖지 않는 산술 코딩 등을 포함하는 다양한 원리들을 따를 수 있다. 파서(520)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 유닛들(Coding Units, CUs), 블록들, 변환 유닛들(Transform Units, TUs), 예측 유닛들(Prediction Units, PUs) 등을 포함할 수 있다. 파서(520)는 또한 코딩된 비디오 시퀀스로부터 변환 계수들, 양자화기 파라미터 값들, 모션 벡터들 등과 같은 정보를 추출할 수 있다.

[0035] 파서(520)는 베퍼 메모리(515)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여, 심

별들(521)을 생성할 수 있다.

[0036] 심별들(521)의 재구성은 코딩된 비디오 픽처 또는 그것의 부분들의 타입(예컨대: 인터 및 인트라 픽처, 인터 및 인트라 블록), 및 다른 인자들에 의존하여 다수의 상이한 유닛들을 수반할 수 있다. 어느 유닛들이 수반되는지, 그리고 어떻게 되는지는 파서(520)에 의해 코딩된 비디오 시퀀스로부터 형성된 서브그룹 제어 정보에 의해 제어될 수 있다. 파서(520)와 아래의 다수의 유닛 사이의 그러한 서브그룹 제어 정보의 흐름은 명확성을 위해 묘사되어 있지 않다.

[0037] 이미 언급된 기능 블록들 이외에, 비디오 디코더(510)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.

[0038] 제1 유닛은 스케일러/역변환 유닛(551)이다. 스케일러/역변환 유닛(551)은, 파서(520)로부터의 심별(들)(521)로서, 어느 변환을 사용할지, 블록 크기, 양자화 인자, 양자화 스케일링 행렬들 등을 포함하여, 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 스케일러/역변환 유닛(551)은 집계기(aggregator)(555)에 입력될 수 있는 샘플 값들을 포함하는 블록들을 출력할 수 있다.

[0039] 일부 경우들에서, 스케일러/역변환(551)의 출력 샘플들은 인트라 코딩된 블록에 관련될 수 있는데; 즉, 이전에 재구성된 픽처들로부터의 예측 정보를 사용하는 것이 아니고, 현재 픽처의 이전에 재구성된 부분들로부터의 예측 정보를 사용할 수 있는 블록. 그러한 예측 정보는 인트라 픽처 예측 유닛(552)에 의해 제공될 수 있다. 일부 경우들에서, 인트라 픽처 예측 유닛(552)은 현재 픽처 베퍼(558)로부터 폐치된 주위의 이미 재구성된 정보를 사용하여, 재구성 하의 블록의 동일한 크기 및 형상의 블록을 생성한다. 현재 픽처 베퍼(558)는, 예를 들어, 부분적으로 재구성된 현재 픽처 및/또는 완전히 재구성된 현재 픽처를 베퍼링한다. 집계기(555)는, 일부 경우들에서, 샘플당 기준으로, 인트라 예측 유닛(552)이 생성한 예측 정보를 스케일러/역변환 유닛(551)에 의해 제공된 출력 샘플 정보에 추가한다.

[0040] 다른 경우들에서, 스케일러/역변환 유닛(551)의 출력 샘플들은 인터 코딩되고, 잠재적으로 모션 보상된 블록에 관련될 수 있다. 그러한 경우에, 모션 보상 예측 유닛(553)은 참조 픽처 메모리(557)에 액세스하여 예측에 사용되는 샘플들을 폐치할 수 있다. 블록에 관련된 심별들(521)에 따라 폐치된 샘플들을 모션 보상한 후에, 이들 샘플은 집계기(555)에 의해 스케일러/역변환 유닛(551)의 출력(이 경우 잔차 샘플들 또는 잔차 신호라고 불림)에 추가되어 출력 샘플 정보를 생성할 수 있다. 모션 보상 예측 유닛(553)이 예측 샘플들을 폐치하는 참조 픽처 메모리(557) 내의 어드레스들은, 예를 들어 X, Y, 및 참조 픽처 컴포넌트들을 가질 수 있는 심별들(521)의 형식으로 모션 보상 예측 유닛(553)에 이용가능한 모션 벡터들에 의해 제어될 수 있다. 모션 보상은 또한 서브 샘플 정확한 모션 벡터들이 사용 중일 때 참조 픽처 메모리(557)로부터 폐치된 샘플 값들의 보간, 모션 벡터 예측 메커니즘 등을 포함할 수 있다.

[0041] 집계기(555)의 출력 샘플들에 대해 루프 필터 유닛(556) 내의 다양한 루프 필터링 기법들이 수행될 수 있다. 비디오 압축 기술들은, 파서(520)로부터의 심별들(521)로서 루프 필터 유닛(556)에 이용가능하게 되고 코딩된 비디오 시퀀스(코딩된 비디오 비트스트림이라고도 지칭됨)에 포함된 파라미터들에 의해 제어되지만, 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서에서) 부분들의 디코딩 동안 획득된 메타-정보에 응답할 뿐만 아니라, 이전에 재구성된 및 루프-필터링된 샘플 값들에 응답할 수도 있는 인-루프 필터(in-loop filter) 기술들을 포함할 수 있다.

[0042] 루프 필터 유닛(556)의 출력은 렌더링 디바이스(512)에 출력될 뿐만 아니라 미래의 인터-픽처 예측에서 사용하기 위해 참조 픽처 메모리(557)에 저장될 수도 있는 샘플 스트림일 수 있다.

[0043] 특정 코딩된 픽처들은, 완전히 재구성되면, 미래 예측을 위한 참조 픽처들로서 사용될 수 있다. 예를 들어, 현재 픽처에 대응하는 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 참조 픽처로서 식별되면(예를 들어, 파서(520)에 의해), 현재 픽처 베퍼(558)는 참조 픽처 메모리(557)의 일부가 될 수 있고, 다음의 코딩된 픽처의 재구성을 개시하기 전에 새로운 현재 픽처 베퍼가 재할당될 수 있다.

[0044] 비디오 디코더(510)는 ITU-T Rec. H.265와 같은 표준에서의 미리 결정된 비디오 압축 기술에 따라 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스는 비디오 압축 기술 또는 표준의 구문, 또는 비디오 압축 기술 또는 표준에서 문서화된 프로파일들 둘 다를 고수한다는 점에서, 코딩된 비디오 시퀀스는 사용 중인 비디오 압축 기술 또는 표준에 의해 특정된 구문을 따를 수 있다. 구체적으로, 프로파일은 비디오 압축 기술 또는 표준에서

이용가능한 모든 툴들로부터 해당 프로파일 하에서 사용하기 위해 이용가능한 유일한 툴들로서 특정 툴들을 선택할 수 있다. 또한 준수를 위해 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 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.

[0045] 일 실시예에서, 수신기(531)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(510)에 의해 사용될 수 있다. 추가적인 데이터는 예를 들어, 시간, 공간, 또는 신호 잡음 비(SNR) 향상 계층들, 중복 슬라이스들, 중복 픽처들, 순방향 오류 정정 코드들 등의 형식일 수 있다.

[0046] 도 6은 본 개시내용의 일 실시예에 따른 비디오 인코더(603)의 블록도를 도시한다. 비디오 인코더(603)는 전자 디바이스(620)에 포함된다. 전자 디바이스(620)는 송신기(640)(예를 들어, 송신 회로)를 포함한다. 비디오 인코더(603)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용될 수 있다.

[0047] 비디오 인코더(603)는 비디오 인코더(603)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(601)(도 6의 예에서는 전자 디바이스(620)의 일부가 아님)로부터 비디오 샘플들을 수신할 수 있다. 다른 예에서, 비디오 소스(601)는 전자 디바이스(620)의 일부이다.

[0048] 비디오 소스(601)는, 임의의 적합한 비트 심도(예를 들어: 8 비트, 10 비트, 12 비트, ...), 임의의 색공간(예를 들어, BT.601 Y CrCb, RGB, ...), 및 임의의 적합한 샘플링 구조(예를 들어, Y CrCb 4:2:0, Y CrCb 4:4:4) 일 수 있는 디지털 비디오 샘플 스트림의 형식으로 비디오 인코더(603)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(601)는 이전에 준비된 비디오를 저장하는 저장 디바이스 일 수 있다. 영상회의 시스템에서, 비디오 소스(601)는 비디오 시퀀스로서 로컬 이미지 정보를 캡처하는 카메라일 수 있다. 비디오 데이터는 순차적으로 보여질 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 의존하여 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 픽셀들과 샘플들 사이의 관계를 쉽게 이해할 수 있다. 아래의 설명은 샘플들에 초점을 맞춘다.

[0049] 일 실시예에 따르면, 비디오 인코더(603)는 소스 비디오 시퀀스의 픽처들을 실시간으로 또는 애플리케이션에 의해 요구되는 임의의 다른 시간 제약들 하에서 코딩된 비디오 시퀀스(643)로 코딩 및 압축할 수 있다. 적절한 코딩 속도를 시행하는 것이 제어기(650)의 하나의 기능이다. 일부 실시예들에서, 제어기(650)는 아래 설명되는 바와 같이 다른 기능 유닛들을 제어하고 다른 기능 유닛들에 기능적으로 결합된다. 결합은 명확성을 위해 묘사되어 있지 않다. 제어기(650)에 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. 제어기(650)는 특정 시스템 설계에 대해 최적화된 비디오 인코더(603)에 관련된 다른 적합한 기능들을 갖도록 구성될 수 있다.

[0050] 일부 실시예들에서, 비디오 인코더(603)는 코딩 루프에서 동작하도록 구성된다. 과도하게 단순화된 설명으로서, 일 예에서, 코딩 루프는 소스 코더(630)(예를 들어, 코딩될 입력 픽처, 및 참조 픽처(들)에 기초하여 심벌 스트림과 같은 심벌들을 생성하는 것을 담당함), 및 비디오 인코더(603)에 임베드된 (로컬) 디코더(633)를 포함할 수 있다. 디코더(633)는 (원격) 디코더가 또한 생성하는 것과 유사한 방식으로 샘플 데이터를 생성하기 위해 심벌들을 재구성한다(심벌들과 코딩된 비디오 비트스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문에). 재구성된 샘플 스트림(샘플 데이터)은 참조 픽처 메모리(634)에 입력된다. 심벌 스트림의 디코딩이 디코더 위치(로컬 또는 원격)와는 독립적으로 비트-정확한 결과들을 야기하기 때문에, 참조 픽처 메모리(634) 내의 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 샘플로서 "본다". 참조 픽처 동기성의 이 기본적인 원리(그리고 결과적인 드리프트, 예를 들어, 채널 오류들 때문에 동기성이 유지될 수 없는 경우)는 일부 관련 기술들에서도 사용된다.

[0051] "로컬" 디코더(633)의 동작은 도 5과 관련하여 위에서 이미 상세히 설명된 비디오 디코더(510)와 같은 "원격" 디코더와 동일할 수 있다. 그러나, 또한 도 5를 간략히 참조하면, 심벌들이 이용가능하고 엔트로피 코더(645) 및 파서(520)에 의한 코딩된 비디오 시퀀스로의 심벌들의 인코딩/디코딩이 무손실일 수 있기 때문에, 버퍼 메모

리(515), 및 파서(520)를 포함하는, 비디오 디코더(510)의 엔트로피 디코딩 부분들은 로컬 디코더(633)에서 완전히 구현되지 않을 수 있다.

[0052] 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.

[0053] 동작 동안, 일부 예들에서, 소스 코더(630)는, "참조 픽처"로 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 픽처를 참조하여 예측적으로 입력 픽처를 코딩하는, 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(632)은 입력 픽처의 픽셀 블록들과 입력 픽처에 대한 예측 참조(들)로서 선택될 수 있는 참조 픽처(들)의 픽셀 블록들 사이의 차이들을 코딩한다.

[0054] 로컬 비디오 디코더(633)는, 소스 코더(630)에 의해 생성된 심벌들에 기초하여, 참조 픽처들로서 지정될 수 있는 픽처들의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(632)의 동작들은 유리하게는 순실 프로세스들일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 6에 도시되지 않음)에서 디코딩될 수 있는 경우, 재구성된 비디오 시퀀스는 전형적으로 일부 오류들을 갖는 소스 비디오 시퀀스의 복제본일 수 있다. 로컬 비디오 디코더(633)는 참조 픽처들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고 재구성된 참조 픽처들이 참조 픽처 캐시(634)에 저장되게 할 수 있다. 이러한 방식으로, 비디오 인코더(603)는 (송신 오류들이 없이) 원단(far-end) 비디오 디코더에 의해 획득될 재구성된 참조 픽처들로서 공통 콘텐츠를 갖는 재구성된 참조 픽처들의 카피들을 국부적으로 저장할 수 있다.

[0055] 예측기(635)는 코딩 엔진(632)에 대한 예측 검색들을 수행할 수 있다. 즉, 코딩될 새로운 픽처에 대해, 예측기(635)는 새로운 픽처들에 대한 적절한 예측 참조로서 역할을 할 수 있는 참조 픽처 모션 벡터들, 블록 형상들 등과 같은 특정 메타데이터 또는 샘플 데이터(후보 참조 픽셀 블록들로서)에 대해 참조 픽처 메모리(634)를 검색할 수 있다. 예측기(635)는 적절한 예측 참조들을 찾기 위해 샘플 블록 바이 픽셀 블록(sample block-by-pixel block) 기준으로 동작할 수 있다. 일부 경우들에서, 예측기(635)에 의해 획득된 검색 결과들에 의해 결정된 바와 같이, 입력 픽처는 참조 픽처 메모리(634)에 저장된 다수의 참조 픽처로부터 인출된 예측 참조들을 가질 수 있다.

[0056] 제어기(650)는, 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하여, 소스 코더(630)의 코딩 동작을 관리할 수 있다.

[0057] 전술한 모든 기능 유닛들의 출력에 대해 엔트로피 코더(645)에서 엔트로피 코딩이 실시될 수 있다. 엔트로피 코더(645)는 다양한 기능 유닛들에 의해 생성된 심벌들을, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술들에 따라 심벌들을 무손실 압축함으로써, 코딩된 비디오 시퀀스로 변환한다.

[0058] 송신기(640)는, 인코딩된 비디오 데이터를 저장할 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는, 통신 채널(660)을 통한 송신을 준비하기 위해 엔트로피 코더(645)에 의해 생성된 코딩된 비디오 시퀀스(들)를 버퍼링할 수 있다. 송신기(640)는 비디오 코더(603)로부터의 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림들(소스들이 도시되지 않음)과 병합할 수 있다.

[0059] 제어기(650)는 비디오 인코더(603)의 동작을 관리할 수 있다. 코딩 동안, 제어기(650)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 픽처 타입들 중 하나로서 할당될 수 있다:

[0060] 인트라 픽처(I 픽처)는 예측의 소스로서 시퀀스 내의 임의의 다른 픽처를 사용하지 않고 코딩되고 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은, 예를 들어, "IDR"(Independent Decoder Refresh) 픽처들을 포함하는, 상이한 타입의 인트라 픽처들을 허용한다. 본 기술분야의 통상의 기술자는 I 픽처들의 해당 변형들 및 그 것들 각각의 애플리케이션들 및 특징들을 인식한다.

[0061] 예측 픽처(P 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 하나의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다.

[0062] 양방향 예측 픽처(B 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 2개의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다. 유사하게, 다

중-예측 픽처들은 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.

[0063] 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 비예측적으로 코딩될 수 있거나 그것들은 동일한 픽처의 이미 코딩된 블록들을 참조하여 예측적으로 코딩될 수 있다(공간 예측 또는 인트라 예측). P 픽처의 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.

[0064] 비디오 인코더(603)는 ITU-T Rec. H.265와 같은 미리 결정된 비디오 코딩 기술 또는 표준에 따라 코딩 동작들을 수행할 수 있다. 그것의 동작에서, 비디오 인코더(603)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.

[0065] 일 실시예에서, 송신기(640)는 인코딩된 비디오와 함께 추가적인 데이터를 송신할 수 있다. 소스 코더(630)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 메시지들, VUI 파라미터 세트 프래그먼트들 등을 포함할 수 있다.

[0066] 비디오는 시간 시퀀스에서 복수의 소스 픽처(비디오 픽처)로서 캡처될 수 있다. 인트라-픽처 예측(종종 인트라 예측으로 축약됨)은 주어진 픽처에서 공간 상관을 사용하고, 인터-픽처 예측은 픽처들 사이의 (시간 또는 다른) 상관을 사용한다. 일 예에서, 현재 픽처라고 지칭되는, 인코딩/디코딩 하의 특정 픽처가 블록들로 분할된다. 현재 픽처 내의 블록이 비디오 내의 이전에 코딩되고 여전히 버퍼링된 참조 픽처 내의 참조 블록과 유사할 때, 현재 픽처 내의 블록은 모션 벡터라고 지칭되는 벡터에 의해 코딩될 수 있다. 모션 벡터는 참조 픽처 내의 참조 블록을 가리키고, 다수의 참조 픽처가 사용 중인 경우, 참조 픽처를 식별하는 제3의 차원을 가질 수 있다.

[0067] 일부 실시예들에서, 인터-픽처 예측에서 양방향 예측(bi-prediction) 기법이 사용될 수 있다. 양방향 예측 기법에 따르면, 둘 다 비디오 내의 현재 픽처에 디코딩 순서에서 앞서는(그러나, 디스플레이 순서에서, 과거 및 미래에 각각 있을 수 있는) 제1 참조 픽처 및 제2 참조 픽처와 같은 2개의 참조 픽처가 사용된다. 현재 픽처 내의 블록은 제1 참조 픽처 내의 제1 참조 블록을 가리키는 제1 모션 벡터, 및 제2 참조 픽처 내의 제2 참조 블록을 가리키는 제2 모션 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 예측될 수 있다.

[0068] 또한, 코딩 효율을 개선하기 위해 인터-픽처 예측에서 병합 모드 기법이 사용될 수 있다.

[0069] 본 개시내용의 일부 실시예들에 따르면, 인터-픽처 예측들 및 인트라-픽처 예측들과 같은 예측들이 블록들의 유닛으로 수행된다. 예를 들어, HEVC 표준에 따르면, 비디오 픽처들의 시퀀스 내의 픽처는 압축을 위해 코딩 트리 유닛들(CTU)로 분할되고, 픽처 내의 CTU들은 64x64 픽셀들, 32x32 픽셀들, 또는 16x16 픽셀들과 같은 동일한 크기를 갖는다. 일반적으로, CTU는 3개의 코딩 트리 블록(CTB)을 포함하는데, 이는 하나의 루마 CTB 및 2개의 크로마 CTB이다. 각각의 CTU는 하나 또는 다수의 코딩 유닛(CU)들로 재귀적으로 쿼드트리 분열될 수 있다. 예를 들어, 64x64 픽셀들의 CTU는 64x64 픽셀들의 하나의 CU, 또는 32x32 픽셀들의 4개의 CU, 또는 16x16 픽셀들의 16개의 CU로 분열될 수 있다. 일 예에서, 각각의 CU는, 인터 예측 타입 또는 인트라 예측 타입과 같은, CU에 대한 예측 타입을 결정하기 위해 분석된다. CU는 시간 및/또는 공간 예측성에 의존하여 하나 이상의 예측 유닛(PU)으로 분열된다. 일반적으로, 각각의 PU는 루마 예측 블록(PB), 및 2개의 크로마 PB를 포함한다. 일 실시예에서, 코딩(인코딩/디코딩)에서의 예측 동작은 예측 블록의 유닛으로 수행된다. 예측 블록의 예로서 루마 예측 블록을 사용하여, 예측 블록은, 8x8 픽셀들, 16x16 픽셀들, 8x16 픽셀들, 16x8 픽셀들 등과 같은, 픽셀들에 대한 값들(예를 들어, 루마 값들)의 행렬을 포함한다.

[0070] 도 7은 본 개시내용의 다른 실시예에 따른 비디오 인코더(703)의 도면을 도시한다. 비디오 인코더(703)는 비디오 픽처들의 시퀀스에서 현재 비디오 픽처 내의 샘플 값들의 처리 블록(예를 들어, 예측 블록)을 수신하고, 처리 블록을 코딩된 비디오 시퀀스의 일부인 코딩된 픽처 내에 인코딩하도록 구성된다. 일 예에서, 비디오 인코더(703)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용된다.

[0071] HEVC 예에서, 비디오 인코더(703)는 8x8 샘플들 등의 예측 블록과 같은 처리 블록에 대한 샘플 값들의 행렬 등

을 수신한다. 비디오 인코더(703)는 처리 블록이, 예를 들어, 레이트-왜곡 최적화를 사용하여 인트라 모드, 인터 모드, 또는 양방향 예측 모드 중 어느 것을 사용하여 최선으로 코딩되는지를 결정한다. 처리 블록이 인트라 모드로 코딩되어야 할 때, 비디오 인코더(703)는 인트라 예측 기법을 사용하여 처리 블록을 코딩된 픽처 내에 인코딩할 수 있고; 처리 블록이 인터 모드 또는 양방향 예측 모드로 코딩되어야 할 때, 비디오 인코더(703)는 인터 예측 또는 양방향 예측 기법을 각각 사용하여 처리 블록을 코딩된 픽처 내에 인코딩할 수 있다. 특정 비디오 코딩 기술들에서, 병합 모드는 예측자들 외부의 코딩된 모션 벡터 성분의 혜택 없이 하나 이상의 모션 벡터 예측자들로부터 모션 벡터가 도출되는 인터 픽처 예측 서브모드일 수 있다. 특정 다른 비디오 코딩 기술들에서, 대상 블록에 적용 가능한 모션 벡터 컴포넌트가 존재할 수 있다. 일 예에서, 비디오 인코더(703)는 처리 블록들의 모드를 결정하기 위한 모드 결정 모듈(도시되지 않음)과 같은 다른 컴포넌트들을 포함한다.

[0072] 도 7의 예에서, 비디오 인코더(703)는 도 7에 도시된 바와 같이 함께 결합된 인터 인코더(730), 인트라 인코더(722), 잔차 계산기(723), 스위치(726), 잔차 인코더(724), 일반 제어기(721), 및 엔트로피 인코더(725)를 포함한다.

[0073] 인터 인코더(730)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 블록을 참조 픽처들 내의 하나 이상의 참조 블록(예를 들어, 이전 픽처들 및 나중 픽처들 내의 블록들)과 비교하고, 인터 예측 정보(예를 들어, 인터 인코딩 기법에 따른 중복 정보의 설명, 모션 벡터들, 병합 모드 정보)를 생성하고, 임의의 적합한 기법을 사용하여 인터 예측 정보에 기초하여 인터 예측 결과들(예를 들어, 예측된 블록)을 계산하도록 구성된다. 일부 예들에서, 참조 픽처들은 인코딩된 비디오 정보에 기초하여 디코딩된 참조 픽처들이다.

[0074] 인트라 인코더(722)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 일부 경우들에서 블록을 동일한 픽처 내의 이미 코딩된 블록들과 비교하고, 변환 후 양자화된 계수들을 생성하고, 일부 경우들에서 또한 인트라 예측 정보(예를 들어, 하나 이상의 인트라 인코딩 기법에 따라 인트라 예측 방향 정보)를 수신하도록 구성된다. 일 예에서, 인트라 인코더(722)는 또한 동일한 픽처 내의 참조 블록들 및 인트라 예측 정보에 기초하여 인트라 예측 결과들(예를 들어, 예측 블록)을 계산한다.

[0075] 일반 제어기(721)는 일반 제어 데이터를 결정하고 일반 제어 데이터에 기초하여 비디오 인코더(703)의 다른 컴포넌트들을 제어하도록 구성된다. 일 예에서, 일반 제어기(721)는 블록의 모드를 결정하고, 모드에 기초하여 스위치(726)에 제어 신호를 제공한다. 예를 들어, 모드가 인트라 모드일 때, 일반 제어기(721)는 잔차 계산기(723)에 의한 사용을 위해 인트라 모드 결과를 선택하도록 스위치(726)를 제어하고, 인트라 예측 정보를 선택하고 인트라 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(725)를 제어하고; 모드가 인터 모드일 때, 일반 제어기(721)는 잔차 계산기(723)에 의한 사용을 위해 인터 예측 결과를 선택하도록 스위치(726)를 제어하고, 인터 예측 정보를 선택하고 인터 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(725)를 제어한다.

[0076] 잔차 계산기(723)는 수신된 블록과 인트라 인코더(722) 또는 인터 인코더(730)로부터 선택된 예측 결과들 사이의 차이(잔차 데이터)를 계산하도록 구성된다. 잔차 인코더(724)는 잔차 데이터에 기초하여 동작하여 잔차 데이터를 인코딩하여 변환 계수들을 생성하도록 구성된다. 일 예에서, 잔차 인코더(724)는 잔차 데이터를 공간 도메인으로부터 주파수 도메인으로 변환하고, 변환 계수들을 생성하도록 구성된다. 그 후 변환 계수들에 대해 양자화 처리를 수행하여 양자화된 변환 계수들을 획득한다. 다양한 실시예들에서, 비디오 인코더(703)는 잔차 디코더(728)를 또한 포함한다. 잔차 디코더(728)는 역변환을 수행하고, 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(722) 및 인터 인코더(730)에 의해 적합하게 사용될 수 있다. 예를 들어, 인터 인코더(730)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있고, 인트라 인코더(722)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있다. 디코딩된 블록들은 디코딩된 픽처들을 생성하기 위해 적합하게 처리되고 디코딩된 픽처들은 메모리 회로(도시되지 않음)에 버퍼링되고 일부 예들에서 참조 픽처들로서 사용될 수 있다.

[0077] 엔트로피 인코더(725)는 인코딩된 블록을 포함하도록 비트스트림을 포맷하도록 구성된다. 엔트로피 인코더(725)는 HEVC 표준과 같은 적합한 표준에 따라 다양한 정보를 포함하도록 구성된다. 일 예에서, 엔트로피 인코더(725)는 일반 제어 데이터, 선택된 예측 정보(예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 잔차 정보, 및 다른 적합한 정보를 비트스트림 내에 포함시키도록 구성된다. 개시된 주제에 따르면, 인터 모드 또는 양방향 예측 모드의 병합 서브모드에서 블록을 코딩할 때, 잔차 정보가 존재하지 않는다는 점에 유의한다.

[0078] 도 8은 본 개시내용의 다른 실시예에 따른 비디오 디코더(810)의 도면을 도시한다. 비디오 디코더(810)는 코딩된 비디오 시퀀스의 일부인 코딩된 픽처들을 수신하고, 코딩된 픽처들을 디코딩하여 재구성된 픽처들을 생성하

도록 구성된다. 일 예에서, 비디오 디코더(810)는 도 4의 예에서의 비디오 디코더(410) 대신에 사용된다.

[0079] 도 8의 예에서, 비디오 디코더(810)는 도 8에 도시된 바와 같이 함께 결합된 엔트로피 디코더(871), 인터 디코더(880), 잔차 디코더(873), 재구성 모듈(874), 및 인트라 디코더(872)를 포함한다.

[0080] 엔트로피 디코더(871)는, 코딩된 픽처로부터, 코딩된 픽처가 구성되는 구문 요소들을 나타내는 특정 심벌들을 재구성하도록 구성될 수 있다. 그러한 심벌들은, 예를 들어, 블록이 코딩되는 모드(예를 들어, 인트라 모드, 인터 모드, 양방향 예측(bi-predicted) 모드, 병합 서브모드 또는 다른 서브모드에서 후자 두 개 등), 인트라 디코더(872) 또는 인터 디코더(880) 각각에 의한 예측을 위해 사용되는 특정 샘플 또는 메타데이터를 식별할 수 있는 예측 정보(예를 들어, 인트라 예측 정보 또는 인터 예측 정보 등), 예를 들어, 양자화된 변환 계수들의 형식으로 된 잔차 정보 등을 포함할 수 있다. 일 예에서, 예측 모드가 인터 또는 양방향 예측 모드일 때, 인터 예측 정보가 인터 디코더(880)에 제공되고; 예측 타입이 인트라 예측 타입일 때, 인트라 예측 정보가 인트라 디코더(872)에 제공된다. 잔차 정보에 대해 역양자화가 수행될 수 있고 이는 잔차 디코더(873)에 제공된다.

[0081] 인터 디코더(880)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기초하여 인터 예측 결과들을 생성하도록 구성된다.

[0082] 인트라 디코더(872)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기초하여 예측 결과들을 생성하도록 구성된다.

[0083] 잔차 디코더(873)는 역양자화를 수행하여 탈양자화된 변환 계수들을 추출하고, 탈양자화된 변환 계수들을 처리하여 잔차를 주파수 도메인으로부터 공간 도메인으로 변환하도록 구성된다. 잔차 디코더(873)는 또한(양자화기 파라미터(QP)를 포함하도록) 특정 제어 정보를 요구할 수 있고, 그 정보는 엔트로피 디코더(871)에 의해 제공될 수 있다(이는 단지 저용량 제어 정보일 수 있으므로 데이터 경로가 묘사되지 않음).

[0084] 재구성 모듈(874)은, 공간 도메인에서, 잔차 디코더(873)에 의해 출력된 잔차와 예측 결과들(경우에 따라 인터 또는 인트라 예측 모듈에 의해 출력된 것)을 조합하여 재구성된 블록을 형성하도록 구성하고, 재구성된 블록은 재구성된 픽처의 일부일 수 있고, 재구성된 픽처는 결국 재구성된 비디오의 일부일 수 있다. 시각적 품질을 개선하기 위해 디블로킹 동작 등과 같은 다른 적합한 동작들이 수행될 수 있다는 점에 유의한다.

[0085] 비디오 인코더들(403, 603, 및 703), 및 비디오 디코더들(410, 510, 및 810)은 임의의 적합한 기법을 사용하여 구현될 수 있다는 점에 유의한다. 일 실시예에서, 비디오 인코더들(403, 603, 및 703), 및 비디오 디코더들(410, 510, 및 810)은 하나 이상의 집적 회로를 사용하여 구현될 수 있다. 다른 실시예에서, 비디오 인코더들(403, 603, 및 603), 및 비디오 디코더들(410, 510, 및 810)은 소프트웨어 명령어들을 실행하는 하나 이상의 프로세서를 사용하여 구현될 수 있다.

[0086] 본 개시내용의 양태들은 인트라 픽처 블록 보상을 위한 인코딩/디코딩 기법들, 구체적으로 가변 CTU 크기를 갖는 검색 범위 조정을 위한 기법들을 제공한다.

[0087] 블록 기반 보상은 인터 예측 및 인트라 예측에 사용될 수 있다. 인터 예측에 대해, 상이한 픽처로부터의 블록 기반 보상은 모션 보상으로 알려져 있다. 인트라 예측의 경우, 블록 기반 보상은 또한 동일한 픽처 내의 이전에 재구성된 영역으로부터 행해질 수 있다. 동일한 픽처 내의 재구성된 영역으로부터의 블록 기반 보상은 인트라 픽처 블록 보상, 현재 픽처 참조(CPR) 또는 인트라 블록 카피(IBC)로 지칭된다. 동일한 픽처 내의 현재 블록과 참조 블록 사이의 오프셋을 지시하는 변위 벡터는 블록 벡터(또는 줄여서 BV)로서 지칭된다. 임의의 값(x 또는 y 방향에서 포지티브 또는 네거티브)일 수 있는 모션 보상에서의 모션 벡터와는 상이하게, 블록 벡터는 참조 블록이 이용가능하고 이미 재구성되는 것을 보장하기 위해 몇 가지 제약을 갖는다. 또한, 일부 예들에서, 병렬 처리 고려사항을 위해, 타일 경계 또는 파면 래더 형상 경계(wavefront ladder shape boundary)인 일부 참조 영역이 배제된다.

[0088] 블록 벡터의 코딩은 명시적이거나 암시적일 수 있다. (인터 코딩에서 AMVP(advanced motion vector prediction) 모드로 지칭되는) 명시적 모드에서, 블록 벡터와 그의 예측자 사이의 차이가 시그널링되고; 암시적 모드에서, 블록 벡터는 병합 모드에서 모션 벡터와 유사한 방식으로 예측자(블록 벡터 예측자라고 지칭됨)로부터 복구된다. 일부 구현들에서, 블록 벡터의 해상도는 정수 위치들로 제한되고; 다른 시스템들에서, 블록 벡터는 분수 위치들을 가리키도록 허용된다.

[0089] 일부 예들에서, 블록 레벨에서의 인트라 블록 카피의 사용은 IBC 플래그로서 지칭되는 블록 레벨 플래그를 사용하여 시그널링될 수 있다. 일 실시예에서, IBC 플래그는 현재 블록이 병합 모드에서 코딩되지 않을 때 시그널

링된다. 다른 예들에서, 블록 레벨에서 인트라 블록 카피의 사용은 참조 인덱스 접근법에 의해 시그널링된다. 디코딩 하의 현재 픽처는 그 후 참조 픽처로서 취급된다. 일 예에서, 이러한 참조 픽처는 참조 픽처들의 리스트의 마지막 위치에 놓인다. 이 특수 참조 픽처는 또한 DPB(decoded picture buffer)와 같은 베퍼 내의 다른 시간 참조 픽처들과 함께 관리된다.

[0090] 풀립된 인트라 블록 카피(참조 블록은 현재 블록을 예측하기 위해 사용되기 전에 수평으로 또는 수직으로 풀립됨), 또는 라인 기반 인트라 블록 카피(MxN 코딩 블록 내의 각각의 보상 유닛은 Mx1 또는 1xN 라인임)와 같은, 인트라 블록 카피에 대한 일부 변형들이 또한 존재한다.

[0091] 도 9는 본 개시내용의 일 실시예에 따른 인트라 블록 카피의 예를 도시한다. 현재 픽처(900)는 디코딩 하에 있다. 현재 픽처(900)는 재구성된 영역(910)(회색 영역) 및 디코딩될 영역(920)(백색 영역)을 포함한다. 현재 블록(930)은 디코더에 의해 재구성 하에 있다. 현재 블록(930)은 재구성된 영역(910)에 있는 참조 블록(940)으로부터 재구성될 수 있다. 참조 블록(940)과 현재 블록(930) 사이의 위치 오프셋은 블록 벡터(950)(또는 BV(950))로서 지정된다.

[0092] 일부 예들(예를 들어, VVC)에서, 인트라 블록 카피 모드의 검색 범위는 현재 CTU 내에 있도록 제약된다. 그 다음, 인트라 블록 카피 모드에 대한 참조 샘플들을 저장하기 위한 메모리 요건은 샘플들의 1(가장 큰) CTU 크기이다. 일 예에서, (가장 큰) CTU는 128x128 샘플들의 크기를 갖는다. 일부 예들에서, CTU는 64x64 샘플들의 크기를 각각 갖는 4개의 블록 영역으로 나누어진다. 따라서, 일부 실시예들에서, 전체 메모리(예를 들어, 메인 스토리지보다 빠른 액세스 속도를 갖는 캐시 메모리)는 128x128 크기의 샘플들을 저장할 수 있고, 전체 메모리는 64x64 영역과 같은 현재 블록에 재구성된 샘플들을 저장하는 기준의 참조 샘플 메모리 부분, 및 64x64 크기의 3개의 다른 영역의 샘플들을 저장하는 추가적인 메모리 부분을 포함한다. 따라서, 일부 예들에서, 인트라 블록 카피 모드의 유효 검색 범위는 좌측 CTU의 일부 부분으로 확장되는 한편 참조 픽셀들을 저장하기 위한 전체 메모리 요건은 변경되지 않은 채로 유지된다(예를 들어, 1 CTU 크기, 총 64x64 참조 샘플 메모리의 4배).

[0093] 일부 실시예들에서, 좌측 CTU로부터의 저장된 참조 샘플들을 현재 CTU로부터의 재구성된 샘플들로 업데이트하기 위해 업데이트 프로세스가 수행된다. 구체적으로는, 일부 예들에서, 업데이트 프로세스는 64x64 루마 샘플 단위로 행해진다. 일 실시예에서, CTU 크기 메모리 내의 4개의 64x64 블록 영역 각각에 대해, 좌측 CTU로부터의 영역들 내의 참조 샘플들은 현재 CTU의 동일 영역 내의 블록들 중 임의의 블록이 코딩되고 있거나 코딩 완료될 때까지 CPR 모드로 현재 CTU 내의 코딩 블록을 예측하는데 사용될 수 있다.

[0094] 도 10a 내지 도 10d는 본 개시내용의 일 실시예에 따른 인트라 블록 카피 모드에 대한 유효 검색 범위들의 예들을 도시한다. 일부 예들에서, 인코더/디코더는 128x128 샘플들과 같은, 하나의 CTU의 샘플들을 저장할 수 있는 캐시 메모리를 포함한다. 또한, 도 10a 내지 도 10d의 예들에서, 예측을 위한 현재 블록 영역은 64x64 샘플의 크기를 갖는다. 예들은 다른 적절한 크기들의 현재 블록 영역에 대해 적절하게 수정될 수 있다는 점에 유의한다.

[0095] 도 10a 내지 도 10d 각각은 현재 CTU(1020) 및 좌측 CTU(1010)를 도시한다. 좌측 CTU(1010)는 4개의 블록 영역(1011-1014)을 포함하고, 각각의 블록 영역은 64x64 샘플의 샘플 크기를 갖는다. 현재 CTU(1020)는 4개의 블록 영역(1021-1024)을 포함하고, 각각의 블록 영역은 64x64 샘플의 샘플 크기를 갖는다. 현재 CTU(1020)는 재구성 하의 현재 블록 영역(라벨 "Curr"로 도시되고 수직 스트라이프 패턴을 가짐)을 포함하는 CTU이다. 좌측 CTU(1010)는 현재 CTU(1020)의 좌측에서의 바로 이웃이다. 도 10a 내지 도 10d에서, 회색 블록들은 이미 재구성된 블록 영역들이고, 백색 블록들은 재구성될 블록 영역들이라는 점에 유의한다.

[0096] 도 10a에서, 재구성 하의 현재 블록 영역은 블록 영역(1021)이다. 캐시 메모리는 재구성된 샘플들을 블록 영역들(1012, 1013 및 1014)에 저장하고, 캐시 메모리는 현재 블록 영역(1021)의 재구성된 샘플들을 저장하는 데 사용될 것이다. 도 10a의 예에서, 현재 블록 영역(1021)에 대한 유효 검색 범위는 캐시 메모리에 저장된 재구성된 샘플들을 갖는 좌측 CTU(1010) 내의 블록 영역들(1012, 1013 및 1014)을 포함한다. 일 실시예에서, 블록 영역(1011)의 재구성된 샘플들은 캐시 메모리보다 느린 액세스 속도를 갖는 메인 메모리에 저장된다(예를 들어, 블록 영역(1021)의 재구성 전에 캐시 메모리로부터 메인 메모리로 카피된다).

[0097] 도 10b에서, 재구성 하의 현재 블록 영역은 블록 영역(1022)이다. 캐시 메모리는 블록 영역들(1013, 1014 및 1021)에 재구성된 샘플들을 저장하고, 캐시 메모리는 현재 블록 영역(1022)의 재구성된 샘플들을 저장하는 데 사용될 것이다. 도 10b의 예에서, 현재 블록 영역(1022)에 대한 유효 검색 범위는 캐시 메모리에 저장된 재구성된 샘플들을 갖는 현재 CTU(1020)에서 좌측 CTU(1010 및 1021) 내의 블록 영역들(1013 및 1014)을 포함한다.

일 실시예에서, 블록 영역(1012)의 재구성된 샘플들은 캐시 메모리보다 느린 액세스 속도를 갖는 메인 메모리에 저장된다(예를 들어, 블록 영역(1022)의 재구성 전에 캐시 메모리로부터 메인 메모리로 카피된다).

[0098] 도 10c에서, 재구성 하의 현재 블록 영역은 블록 영역(1023)이다. 캐시 메모리는 재구성된 샘플들을 블록 영역들(1014, 1021, 및 1022)에 저장하고, 캐시 메모리는 현재 블록 영역(1023)의 재구성된 샘플들을 저장하는 데 사용될 것이다. 도 10c의 예에서, 현재 블록(1023)에 대한 유효 검색 범위는 캐시 메모리에 저장된 재구성된 샘플들을 갖는 현재 CTU(1020)에서 좌측 CTU(1010, 1021, 및 1022) 내의 블록 영역들(1014)을 포함한다. 일 실시예에서, 블록 영역(1013)의 재구성된 샘플들은 캐시 메모리보다 느린 액세스 속도를 갖는 메인 메모리에 저장된다(예를 들어, 블록 영역(1023)의 재구성 전에 캐시 메모리로부터 메인 메모리로 카피된다).

[0099] 도 10d에서, 재구성 하의 현재 블록 영역은 블록 영역(1024)이다. 캐시 메모리는 재구성된 샘플들을 블록 영역들(1021, 1022 및 1023)에 저장하고, 캐시 메모리는 현재 블록 영역(1024)의 재구성된 샘플들을 저장하는 데 사용될 것이다. 도 10d의 예에서, 현재 블록 영역(1024)에 대한 유효 검색 범위는 캐시 메모리에 저장된 재구성된 샘플들을 갖는 현재 CTU(1020) 내의 블록들(1021, 1022 및 1023)을 포함한다. 일 실시예에서, 블록 영역(1014)의 재구성된 샘플들은 캐시 메모리보다 느린 액세스 속도를 갖는 메인 메모리에 저장된다(예를 들어, 블록 영역(1024)의 재구성 이전에 캐시 메모리로부터 메인 메모리로 카피된다).

[0100] 위의 예들에서, 캐시 메모리는 1 (가장 큰) CTU 크기에 대한 전체 메모리 공간을 갖는다. 예들은 다른 적합한 CTU 크기들에 적합하게 조정될 수 있다. 일부 예에서 캐시 메모리는 참조 샘플 메모리라고 지칭된다는 점에 유의한다.

[0101] 제안된 방법들은 개별적으로 사용되거나 임의의 순서로 조합될 수 있다. 또한, 방법들(또는 실시예들), 인코더 및 디코더 각각은 처리 회로(예를 들어, 하나 이상의 프로세서 또는 하나 이상의 집적 회로)에 의해 구현될 수 있다. 하나의 예에서, 하나 이상의 프로세서는 비일시적 컴퓨터 판독가능 매체에 저장되는 프로그램을 실행한다. 이하에서, 블록이라는 용어는 예측 블록, 코딩 블록, 또는 코딩 유닛, 즉 CU로서 해석될 수 있다.

[0102] 본 개시내용의 양태들은 CTU 크기가 예를 들어 최대 CTU 크기보다 작도록 변화될 때 검색 범위 조정을 위한 기법들을 제공한다. 구현에서, 미래의 인트라 블록 카피 참조를 위해 이전에 코딩된 CU들의 참조 샘플들을 저장하는 지정된 메모리는 참조 샘플 메모리라고 지칭(일부 예들에서는 캐시 메모리라고 지칭)된다. 본 개시내용에서는, 특정 참조 영역 제약들 하에서 인트라 블록 카피 성능을 개선하기 위한 방법들이 제안된다. 더 구체적으로, 검색을 위한 참조 샘플 메모리의 크기가 제한된다. 이하의 논의에서, 참조 샘플 메모리의 크기는 (대응하는 크로마 샘플들과 함께) 128x128 루마 샘플들이 되도록 고정된다. 일부 예들에서(예를 들어, VVC 표준에서), 참조 샘플들의 하나의 최대 CTU 크기가 지정된 메모리 크기로서 간주된다. 제안된 방법들은 CTU 크기에 대한 64x64 루마 샘플들(플러스 대응하는 크로마 샘플들) 및 메모리 크기에 대한 128x128 루마 샘플(및 대응하는 크로마 샘플들)과 같은 상이한 메모리 크기/CTU 크기 조합들로 추가로 확장될 수 있다.

[0103] 본 개시내용의 일 양태에 따르면, 본 개시내용에서의 병치된 블록들은 동일한 크기들 및 동일한 형상을 갖는 블록들의 쌍을 지칭하고, 병치된 블록들 중 하나는 이전에 코딩된 CTU 내에 있고, 병치된 블록들 중 다른 하나는 현재 CTU 내에 있고, 쌍 내의 하나의 블록은 쌍 내의 다른 블록의 대조 블록(collated block)이라고 지칭된다. 또한, 메모리 베퍼 크기가 최대 크기(예를 들어, 128x128)의 CTU를 저장하도록 설계될 때, 이전 CTU는 일 예에서 현재 CTU의 좌측에 하나의 CTU 폭 루마 샘플 오프셋을 갖는 CTU를 지칭한다. 또한, 이들 2개의 병치된 블록은 각각 그들 자신의 CTU의 좌측-상단 코너에 대해 동일한 위치 오프셋 값들을 갖는다. 즉, 병치된 블록들은 픽처의 좌측-상단 코너에 대해 동일한 y 좌표를 갖지만, 일부 예들에서는 서로에 대해 x 좌표들에서의 CTU 폭 차이를 갖는 2개의 블록이다.

[0104] 도 11은 본 개시내용의 일부 실시예들에 따른 병치된 블록들의 예들을 도시한다. 도 11의 예에서, 디코딩 동안 현재 CTU 및 좌측 CTU가 도시되어 있다. 재구성된 영역은 회색으로 도시되고, 재구성될 영역은 백색으로 도시된다. 도 11은 디코딩 동안 인트라 블록 카피 모드에서 현재 블록에 대한 좌측 CTU 내의 참조 블록들의 3개의 예를 도시한다. 3개의 예는 참조 블록 1, 참조 블록 2 및 참조 블록 3으로서 도시되어 있다. 도 11은 또한 참조 블록 1에 대한 병치된 블록 1, 참조 블록 2에 대한 병치된 블록 2 및 참조 블록 3에 대한 병치된 블록 3을 도시한다. 도 11의 예에서, 참조 샘플 메모리 크기는 CTU 크기이다. 현재 CTU 및 좌측 CTU의 재구성된 샘플들은 상보적인 방식으로 참조 샘플 메모리에 저장된다. 현재 CTU의 재구성된 샘플이 참조 샘플 메모리에 기입될 때, 재구성된 샘플은 좌측 CTU 내의 대조 샘플 대신에 기입된다. 일 예에서, 참조 블록 3에 대해, 병치된 블록 3은 아직 재구성되지 않은 현재 CTU 내에 있기 때문에, 참조 블록 3은 참조 샘플 메모리로부터 발견될 수 있다. 참조 샘플 메모리는 좌측 CTU로부터의 참조 블록 3의 샘플들을 여전히 저장하고, 고속 속도로 액세스되어 참조

블록 3의 샘플들을 검색할 수 있고, 참조 블록 3은 일 예에서 인트라 블록 카피 모드에서 현재 블록을 재구성하는 데 사용될 수 있다.

[0105] 다른 예에서, 참조 블록 1에 대해, 현재 CTU 내의 병치된 블록 1은 재구성 완료되었고, 따라서 참조 샘플 메모리는 병치된 블록 1의 샘플들을 저장하고, 참조 블록 1의 샘플들은 예를 들어 참조 샘플 메모리에 비해 비교적 높은 지연을 갖는 오프-칩 스토리지에 저장되었다. 따라서, 일 예에서, 참조 블록 1은 참조 샘플 메모리에서 발견될 수 없고, 참조 블록 1은 일 예에서 인트라 블록 카피 모드에서 현재 블록을 재구성하는데 사용될 수 없다.

[0106] 유사하게, 다른 예에서, 참조 블록 2에 대해, 병치된 블록 2의 일부가 재구성되었고, 따라서 참조 샘플 메모리는 병치된 블록 2의 샘플들을 저장하도록 업데이트되었다. 따라서, 일 예에서, 참조 블록 2는 인트라 블록 카피 모드에서 현재 블록을 재구성하기 위한 유효 참조 블록일 수 없다.

[0107] 일반적으로, 인트라 블록 카피 모드에서, 이전에 디코딩된 CTU 내의 참조 블록에 대해, 현재 CTU 내의 병치된 블록이 아직 재구성되지 않았을 때, 참조 블록의 샘플들은 참조 샘플 메모리에서 이용가능하고, 참조 샘플 메모리는 인트라 블록 카피 모드에서 재구성을 위한 참조로서 사용하기 위해 참조 블록의 샘플들을 검색하기 위해 액세스될 수 있다.

[0108] 위의 예들에서, 참조 블록의 좌측-상단 코너의 병치된 샘플이라고도 지칭되는, 현재 CTU 내의 대조 블록의 좌측-상단 코너 샘플이 체크된다는 점에 유의한다. 현재 CTU 내의 병치된 샘플이 아직 재구성되지 않았을 때, 참조 블록에 대한 나머지 샘플들은 모두 인트라 블록 카피에서 참조로서 사용하는데 이용가능할 것이다.

[0109] 또한, 위의 예들에서, 참조 샘플 메모리의 크기는 하나의 CTU의 크기이고, 이전에 디코딩된 CTU는 현재 CTU의 바로 좌측에 있는 CTU를 의미한다는 점에 유의한다.

[0110] 본 개시내용의 일 양태에 따르면, 참조 샘플 메모리의 크기는 하나의 CTU의 크기보다 클 수 있다.

[0111] 도 12는 본 개시내용의 일부 실시예들에 따른 병치된 블록들의 예들을 도시한다. 도 12의 예에서, 참조 샘플 메모리는 CTU 크기의 N배(N은 2 이상인 정수임)를 갖도록 구성되고, 따라서 참조 샘플 메모리는 도 12에 도시된 바와 같이 현재 CTU(1210), 좌측 끝(the furthest left) CTU(1230), 및 현재 CTU(1210)와 좌측 끝 CTU(1230) 사이에 있는 좌측-중간(mid-left) CTU(들)(1220)와 같은 N+1개의 CTU로부터의 재구성된 샘플들을 저장할 수 있다. 좌측 CTU(들)의 수는 N-1과 동일하다. 현재 CTU(1210) 및 좌측 끝 CTU(1230)의 재구성된 샘플들은 상보적인 방식으로 참조 샘플 메모리에 저장된다. 현재 CTU(1210)의 재구성된 샘플을 참조 샘플 메모리에 저장하기 위해, 현재 CTU(1210)의 재구성된 샘플은 좌측 끝 CTU(1230)의 대조 샘플 대신에 기입된다. 재구성된 영역은 회색으로 도시되고, 재구성될 영역은 백색으로 도시된다. 일부 실시예들에서, 좌측 CTU들은 바로 좌측 CTU로부터 좌측 끝 CTU로 넘버링된다. 예를 들어, 현재 CTU(120)에 바로 좌측 CTU는 제1 좌측 CTU이고, 좌측 끝 CTU(1230)는 제N 좌측 CTU이고, 좌측 끝 CTU의 우측의 CTU는 제(N-1) 좌측 CTU이다.

[0112] 도 12의 예에서, 참조 샘플 메모리 크기는 CTU 크기의 N배이고, 따라서 좌측-중간 CTU(들)(1220)의 샘플들은 모두 현재 CTU(1210) 내의 샘플들의 재구성 동안 이용가능하다는 점에 유의한다. 그러나, 좌측 끝 CTU(1230)의 샘플들은 참조 샘플 메모리에서 부분적으로 이용가능하고, 도 11의 예에 도시된 바와 같은 유사한 제약들 하에 있다. 예를 들어, 디코딩 동안 인트라 블록 카피 모드에서 현재 블록에 대한 좌측 끝 CTU(1230)에서의 참조 블록들의 3개의 예가 참조 블록 1, 참조 블록 2 및 참조 블록 3으로서 도시된다. 도 12는 또한 참조 블록 1에 대한 병치된 블록 1, 참조 블록 2에 대한 병치된 블록 2 및 참조 블록 3에 대한 병치된 블록 3을 도시한다. 이러한 경우에, 한 쌍의 병치된 블록들 또는 샘플들 사이의 x 좌표 오프셋은 CTU 폭의 N배일 것이다. 도 12의 다른 설명은 도 11과 유사하며, 위에서 제공되었고 명확함을 위해 여기서는 생략될 것이다.

[0113] 일부 예들에서, 좌측 끝 CTU(1230)의 재구성 샘플들의 부분 및 현재 CTU(1210)의 재구성된 샘플들의 부분은 참조 샘플 메모리에서 상보적이라는 점에 유의한다. 현재 CTU(1210)의 샘플이 재구성될 때, 현재 CTU(1210)의 재구성된 샘플은 좌측 끝 CTU(1230)의 재구성된 샘플 대신에 참조 샘플 메모리에 기입된다. 좌측 끝 CTU(1230)의 재구성된 샘플은 참조 샘플 메모리 내의 현재 CTU(1210)의 재구성된 샘플로 오버라이트되기 전에 메인 메모리에 기입되었다.

[0114] 도 12의 예는 참조 샘플 메모리 크기가 CTU 크기와 동일하거나 그보다 큰 시나리오들에 사용될 수 있다는 점에 유의한다. 예를 들어, 참조 샘플 메모리 크기가 CTU 크기의 4배일 때, 좌측-중간 CTU(들)(1220)는 3개의 CTU를 포함하고, 참조 샘플 메모리 크기가 CTU 크기의 16배일 때, 좌측-중간 CTU(들)(1220)는 15개의 CTU를 포함한

다. 일 예에서, 참조 샘플 메모리 크기가 CTU 크기와 동일할 때, 좌측-중간 CTU(1220)가 없다.

[0115] 본 개시내용의 양태들은 현재 CTU 크기가 최대 CTU 크기 미만이고 참조 샘플 메모리 크기가 1 최대 CTU 크기와 동일할 때 인트라 블록 카피 모드에서의 검색 범위 조정을 위한 기법들을 제공한다. 그 후, 참조 샘플 메모리는 다수의 이전에 디코딩된 CTU들을 버퍼링할 수 있다. 참조 블록이 바로 좌측 CTU로부터의 것일 때, 이용가능성에 대한 어떠한 추가적인 조건 체크들도 필요하지 않다. 이 경우에 인트라 블록 카피 참조를 위해 전체 좌측 CTU가 이용가능하다. 일부 실시예들에서, 현재 CTU 크기가 최대 CTU 크기와 동일한 경우 및 현재 CTU 크기가 최대 CTU 크기보다 작은 경우에 대해 통합 조건 체크(unified condition check)가 사용된다.

[0116] 실시예들의 이하의 설명에서 다양한 파라미터들이 사용된다.

[0117] MaxCtbLog2SizeY는, log2 도메인에서, CTU가 정사각형 형상을 가질 때 일 측(높이 또는 폭)에서의 최대 허용 CTU 크기를 나타낸다. 예를 들어, 최대 허용 CTU가 128x128 투마 샘플 높이(high)일 때, MaxCtbLog2SizeY는 7과 동일하다.

[0118] CtbLog2SizeY는, log2 도메인에서, CTU가 정사각형 형상을 가질 때 일 측(높이 또는 폭)에서의 CTU 크기를 나타낸다.

[0119] cbHeight는 코딩 블록(현재 블록이라고도 함)의 높이를 나타내고; cbWidth는 코딩 블록의 폭을 나타낸다. 코딩 블록의 좌측-상단 코너의 위치는 (xCb, yCb)로서 표기되고, 코딩 블록의 우측 상단 코너의 위치는 (xCb+cbWidth-1, yCb)로서 표기되고, 코딩 블록의 좌측 하단 코너의 위치는 (xCb, yCb+cbHeight-1)로서 표기된다.

[0120] mvL0은 블록 벡터를 나타내고, mvL0[0]은 1/16 웨 해상도에서 블록 벡터 mvL0의 x 성분을 나타내고, mvL0[1]은 1/16 웨 해상도에서 블록 벡터 mvL0의 y 성분을 나타낸다. 따라서, x 성분의 정수 값은 mvL0[0]을 4 비트만큼 우측 시프트함으로써 획득되고, y 성분의 정수 값은 mvL0[1]을 4 비트만큼 우측 시프트함으로써 획득된다.

[0121] 본 개시내용의 일부 양태들에 따르면, 블록 벡터 mvL0은, mvL0이 인트라 블록 카피 모드에서 참조 블록을 가리키는 유효 블록 벡터이고 참조 블록이 참조 샘플 메모리에 완전히 저장되는지를 결정하기 위해 2-단계(two-step) 제약 프로세스를 사용하여 제약된다. 따라서, 참조 샘플 메모리는 메인 메모리(예를 들어, 오프-침 메모리)로부터 참조 샘플들을 검색하지 않고 액세스될 수 있다.

[0122] 제1 단계에서, 블록 벡터 mvL0은 블록 벡터가 참조 샘플 메모리 내의 재구성된 샘플들의 전부 또는 일부를 갖는 임의의 CTU들을 포함하는 CTU 기반 검색 범위를 가리키는 잠재적인 유효 블록 벡터에 있는지를 결정하기 위해 체크된다. 예를 들어, CTU 기반 검색 범위는 현재 CTU, 좌측 끝 CTU, 및 현재 CTU와 좌측 끝 CTU 사이의 임의의 좌측 CTU들을 포함한다. 일 예에서, CTU 크기는 최대 허용 CTU 크기와 동일하고, 그 다음에 CTU 기반 검색 범위는 현재 CTU 및 좌측 끝 CTU(도 11에서 좌측 CTU라고 지칭됨)를 포함하고, 현재 CTU와 좌측 끝 CTU 사이에는 좌측-중간 CTU가 없다.

[0123] 블록 벡터 mvL0이 잠재적인 유효 블록 벡터일 때, 블록 벡터 mvL0은 제2 단계에서 블록 벡터가 유효 블록 벡터 인지를 결정하기 위해 추가로 체크된다. 예를 들어, 제2 단계에서, 블록 벡터가 좌측-중간 CTU 내의 참조 블록을 가리킬 때, 블록 벡터는 유효 블록 벡터이다. 제2 단계에서, 블록 벡터가 좌측 끝 CTU 내의 참조 블록을 가리킬 때, 참조 블록의 병치된 블록이 재구성되었는지를 결정하기 위해 블록 벡터 mvL0이 추가로 체크된다.

[0124] 제1 실시예에서, CTU 크기는 최대 허용된 CTU 크기와 동일하고, 따라서 참조 샘플 메모리에 저장된 재구성된 샘플들은 현재 CTU 또는 좌측 끝 CTU(도 11에서 좌측 CTU)로부터의 것이다.

[0125] 제1 단계에서, mvL0은 블록 벡터가 현재 CTU 및 좌측 끝 CTU를 포함하는 CTU 기반 검색 범위를 가리키는지를 결정하기 위해 체크된다. 일부 예들에서, 동일한 y 값들을 갖는 CTU들은 CTU 행을 형성하고, 동일한 x 값들을 갖는 CTU들은 CTU 열을 형성한다. 일부 예에서, 블록 벡터 mvL0에 대한 제약은 수학식 1 내지 수학식 4에 의해 표현된다.

## 수학식 1

$$( yCb + ( mvL0[ 1 ] >> 4 ) ) >> CtbLog2SizeY = yCb >> CtbLog2SizeY$$

[0126]

## 수학식 2

[0127]  $( yCb + ( mvL0[ 1 ] >> 4 ) + cbHeight - 1 ) >> CtbLog2SizeY = yCb >> CtbLog2SizeY$

## 수학식 3

[0128]  $( xCb + ( mvL0[ 0 ] >> 4 ) ) >> CtbLog2SizeY >= ( xCb >> CtbLog2SizeY ) - 1$

## 수학식 4

[0129]  $( xCb + ( mvL0[ 0 ] >> 4 ) + cbWidth - 1 ) >> CtbLog2SizeY <= ( xCb >> CtbLog2SizeY )$

[0130] 수학식 1이 충족될 때, 참조 블록의 상단은 현재 블록과 동일한 CTU 행에 있다. 수학식 2가 충족될 때, 참조 블록의 하단은 현재 블록과 동일한 CTU 행에 있다. 수학식 3이 충족될 때, 일 예에서, 참조 블록의 좌측은 현재 블록과 동일한 CTU 열에 있거나, 현재 블록으로부터 바로 좌측 CTU 열에 있다. 수학식 4가 충족될 때, 일 예에서 참조 블록의 우측은 현재 블록과 동일한 CTU 열에 있거나, 현재 블록으로부터 좌측 CTU 열에 있다. 따라서, 일 예에서 수학식 1 내지 수학식 4가 충족될 때, 참조 블록은 CTU 기반 검색 범위 내에 있다.

[0131] 제2 단계에서, 참조 블록이 좌측 끝 CTU에 있을 때, 예를 들어, 수학식 5가 충족될 때, 참조 블록의 병치된 블록이 재구성되었는지를 결정하기 위해 블록 벡터 mvL0이 체크된다.

## 수학식 5

[0132]  $( xCb + ( mvL[ 0 ] >> 4 ) ) >> CtbLog2SizeY = ( xCb >> CtbLog2SizeY ) - 1$

[0133] 일 예에서, 현재 블록의 좌측-상단 코너는  $(xCb, yCb)$ 에 있고, 참조 블록의 좌측-상단 코너는  $(xCb + (mvL[0] >> 4), yCb + (mvL[1] >> 4))$ 에 있고, 참조 블록에 대한 병치된 블록의 좌측-상단 코너는  $(xCb + (mvL[0] >> 4) + (1 << CtbLog2SizeY), yCb + (mvL[1] >> 4))$ 에 있다. 일 예에서, 병치된 블록의 좌측-상단 코너는 꽉쳐의 재구성 프로세스를 추적하는 맵 상에서 체크하기 위해 사용된다. 맵 상의 위치가 일 예에서 병치된 블록이 재구성되지 않았음을 지시하는 "거짓"일 때, 따라서 참조 블록은 참조 샘플 메모리에서 이용가능하고, 인트라 블록 카피 모드에서 현재 블록을 재구성하는데 사용될 수 있다. 그 다음, 블록 벡터는 유효 블록 벡터이다. 그러나, 맵 상의 위치가 일 예에서 병치된 블록의 적어도 일부가 재구성되었다는 것을 지시하는 "참"일 때, 따라서 병치된 블록의 샘플들은 참조 블록의 샘플들 대신에 참조 샘플 메모리에 저장되었고, 참조 블록의 샘플들은 참조 샘플 메모리에서 이용가능하지 않다. 그러면, 블록 벡터 mvL0은 유효 블록 벡터가 아니다.

[0134] 제1 실시예에서, 제2 단계는 병치된 블록에 기초하고, 제1 실시예에서의 프로세스는 코딩 유닛(CU) 기반 업데이트 프로세스로서 지칭된다.

[0135] 제2 실시예에서, 제2 단계는 블록 영역에 기초한다. 예를 들어, CTU가 128x128 샘플들을 가질 때, CTU는 64x64 샘플들을 각각 갖는 4개의 블록 영역으로 나뉘어 질 수 있다. 유사하게, 제2 실시예에서, CTU 크기는 최대 허용된 CTU 크기와 동일하고, 따라서 참조 샘플 메모리에 저장된 재구성된 샘플들은 현재 CTU 또는 좌측 끝 CTU(도 11에서 좌측 CTU)로부터의 것이다.

[0136] 제1 단계에서, 블록 벡터 mvL0가, 이를테면 제1 실시예에서와 같이 유사한 방식으로 수학식 1 내지 수학식 4를 사용하여, 현재 CTU 및 좌측 끝 CTU를 포함하는 CTU 기반 검색 범위를 가리키는지를 결정하기 위해 블록 벡터 mvL0이 체크된다.

[0137] 제2 단계에서, 참조 블록이 좌측 끝 CTU에 있을 때, 예를 들어, 수학식 5가 충족될 때, 참조 블록에 대한 병치

된 블록 영역(예를 들어, 64x64 블록 영역)이 재구성되었는지를 결정하기 위해 블록 벡터 mvL0이 체크된다.

[0138] 일 예에서, 현재 블록의 좌측-상단 코너는 (xCb, yCb)에 있고, 그 후 참조 블록의 좌측-상단 코너는 (xCb+(mvL[0]>>4), yCb+(mvL[1]>>4))에 있고, 참조 블록에 대한 병치된 블록 영역의 좌측-상단 코너는 (((xCb+(mvL[0]>>4)+(1<<CtbLog2SizeY))>>(CtbLog2SizeY-1))<<(CtbLog2SizeY-1), ((yCb+(mvL[1]>>4))>>(CtbLog2SizeY-1))<<(CtbLog2SizeY-1))에 있다. 일 예에서, 병치된 블록 영역의 좌측-상단 코너는 꼭처의 재구성 프로세스를 추적하는 맵 상에서 체크하기 위해 사용된다. 맵 상의 위치가 일 예에서 병치된 블록 영역이 재구성되지 않았음을 지시하는 "거짓"일 때, 따라서 참조 블록은 참조 샘플 메모리에서 이용가능하고, 인트라 블록 카피 모드에서 현재 블록을 재구성하는데 사용될 수 있다. 그 다음, 블록 벡터는 유효 블록 벡터이다. 그러나, 맵 상의 위치가 일 예에서 병치된 블록 영역이 재구성되었거나 예에서 부분적으로 재구성되었음을 지시하는 "참"일 때, 블록 벡터 mvL0은 유효 블록 벡터가 아니다.

[0139] 제2 실시예에서, 제2 단계는 병치된 블록 영역에 기초하고, 제2 실시예에서의 프로세스는 블록 영역 기반 업데이트 프로세스라고 지칭된다. 제2 실시예에서, 현재 CTU 내의 64x64 블록 영역(병치된 블록 영역)의 임의의 샘플이 재구성되었을 때, (현재 샘플의) 병치된 샘플이 속하는 참조 샘플 메모리 내의 대응하는 영역은 인트라 블록 카피 참조를 위해 이용가능하지 않을 것이다.

[0140] 본 개시내용의 양태에 따르면, CTU 크기 변경은 통상적으로 폭 및/또는 높이가 2배가 되거나 절반으로 감소되면서 발생한다. 일 예에서, 폭 및 높이가 절반으로 감소될 때, 1 최대 CTU 크기의 참조 샘플 메모리는 4개의 CTU에 대한 샘플들을 저장할 수 있다.

[0141] 일부 실시예들에서, Log2 도메인 CtbLog2SizeY에서 현재 CTU 크기가 사용될 때, 참조 샘플 메모리 범위에 참조 샘플 데이터를 저장할 수 있는 CTU들의 수는 MaxCtbLog2SizeY와 CtbLog2SizeY 사이의 관계에 따라 가변일 것이다.

[0142] 제3 실시예에서, CTU 크기는 Log2 도메인에서 CtbLog2SizeY이고, 따라서 참조 샘플 메모리에 저장된 재구성된 샘플들은 현재 CTU, 좌측 끝 CTU, 및 현재 CTU와 좌측 끝 CTU 사이에 위치되는 좌측-중간 CTU(들) 중 임의의 것으로부터 비롯된다.

[0143] 제1 단계에서, 블록 mvL0은 블록 벡터 mvL0이 현재 CTU, 좌측 끝 CTU 및 좌측-중간 CTU(들)를 포함하는 CTU 기반 검색 범위를 가리키는지를 결정하기 위해 체크된다. 일부 예들에서, 블록 벡터 mvL0에 대한 제약은 수학식 6 내지 수학식 9에 의해 표현된다.

### 수학식 6

$$( yCb + ( mvL0[ 1 ] >> 4 ) ) >> CtbLog2SizeY = yCb >> CtbLog2SizeY$$

### 수학식 7

$$( yCb + ( mvL0[ 1 ] >> 4 ) + cbHeight - 1 ) >> CtbLog2SizeY = yCb >> CtbLog2SizeY$$

### 수학식 8

$$( xCb + ( mvL0[ 0 ] >> 4 ) ) >> CtbLog2SizeY >= ( xCb >> CtbLog2SizeY ) - 1 << ( 2*(MaxCtbLog2SizeY - CtbLog2SizeY) )$$

## 수학식 9

$$[0147] (xCb + (mvL0[0] >> 4) + cbWidth - 1) >> CtbLog2SizeY \leq (xCb >> CtbLog2SizeY)$$

[0148] 수학식 6이 충족될 때, 참조 블록의 상단은 현재 블록과 동일한 CTU 행에 있다. 수학식 7이 충족될 때, 참조 블록의 하단은 현재 블록과 동일한 CTU 행에 있다. 수학식 8이 충족될 때, 참조 블록의 좌측은 일 예에서 현재 CTU, 좌측-중간 CTU(들) 및 좌측 끝 CTU 중 하나와 동일한 CTU 열에 있다. 수학식 9가 충족될 때, 참조 블록의 우측은 현재 CTU 중 하나와 동일한 CTU 열에 있거나, 또는 현재 CTU의 좌측에 있는 CTU(예를 들어, 좌측-중간 CTU(들), 좌측 끝 CTU 등)에 있다. 따라서, 일 예에서 수학식 6 내지 수학식 9가 충족될 때, 참조 블록은 CTU 기반 검색 범위 내에 있다.

[0149] 블록 벡터 mvL0이 잠재적인 유효 블록 벡터일 때, 블록 벡터 mvL0은 블록 벡터가 유효 블록 벡터인지를 결정하기 위해 제2 단계에서 추가로 체크된다. 예를 들어, 제2 단계에서, 블록 벡터가 좌측-중간 CTU에서의 참조 블록을 가리킬 때, 예를 들어, 수학식 10이 충족되지 않을 때, 블록 벡터는 유효 블록 벡터이다. 제2 단계에서, 참조 블록이 좌측 끝 CTU에 있을 때, 예를 들어, 수학식 10이 충족될 때, 일 예에서, 블록 벡터 mvL0가 추가 체크되어, 참조 블록의 병치된 블록이 재구성되었는지 여부를 결정한다.

## 수학식 10

$$[0150] (xCb + (mvL[0] >> 4)) >> CtbLog2SizeY = (xCb >> CtbLog2SizeY) - (1 << (2 * (MaxCtbLog2SizeY - CtbLog2SizeY)))$$

[0151] 일 예에서, 현재 블록의 좌측-상단 코너는  $(xCb, yCb)$ 에 있고, 이어서 참조 블록의 좌측-상단 코너는  $(xCb + (mvL[0] >> 4), yCb + (mvL[1] >> 4))$ 에 있고, 참조 블록에 대한 병치된 블록의 좌측-상단 코너는  $(xCb + (mvL[0] >> 4) + (1 << (2 * (MaxCtbLog2SizeY - CtbLog2SizeY))), yCb + (mvL[1] >> 4))$ 에 있다. 일 예에서, 병치된 블록의 좌측-상단 코너는 꽉치의 재구성 프로세스를 추적하는 맵 상에서 체크하기 위해 사용된다. 맵 상의 위치가 일 예에서 병치된 블록이 재구성되지 않았음을 지시하는 "거짓"일 때, 따라서 참조 블록은 참조 샘플 메모리에서 이용가능하고, 인트라 블록 카피 모드에서 현재 블록을 재구성하는데 사용될 수 있다. 그 다음, 블록 벡터는 유효 블록 벡터이다. 그러나, 맵 상의 위치가 일 예에서 병치된 블록의 적어도 일부가 재구성되었다는 것을 지시하는 "참"일 때, 따라서 병치된 블록의 샘플들은 참조 블록의 샘플들 대신에 참조 샘플 메모리에 저장되었고, 참조 블록의 샘플들은 참조 샘플 메모리에서 이용가능하지 않다. 그러면, 블록 벡터 mvL0은 유효 블록 벡터가 아니다.

[0152] 제3 실시예에서, 제2 단계는 병치된 블록에 기초하고, 제3 실시예에서의 프로세스는 CU 기반 업데이트 프로세스라고 지칭된다.

[0153] 제4 실시예에서, 제2 단계는 블록 영역에 기초한다. 예를 들어, CTU 크기가 Log2 도메인에서 CtbLog2SizeY이고, 블록 영역 크기가 Log2 도메인에서 CtbLog2SizeY-1일 때, CTU는 동일한 크기의 4개의 블록 영역으로 나누어질 수 있다. 유사하게, 제4 실시예에서, 따라서 참조 샘플 메모리에 저장된 재구성된 샘플들은 현재 CTU, 좌측 끝 CTU 및 현재 CTU와 좌측 끝 CTU 사이의 좌측-중간 CTU(들)로부터의 것이다.

[0154] 제1 단계에서, 블록 벡터 mvL0이, 이를테면 제3 실시예에서와 같은 유사한 방식으로 수학식 6 내지 수학식 9를 사용하여 현재 CTU, 좌측 끝 CTU, 및 현재 CTU와 좌측 끝 CTU 사이의 좌측-중간 CTU(들)를 포함하는 CTU 기반 검색 범위를 가리키는 잠재적인 유효 블록 벡터인지를 결정하기 위해 블록 벡터 mvL0이 체크된다.

[0155] 블록 벡터 mvL0이 잠재적인 유효 블록 벡터일 때, 블록 벡터 mvL0은 블록 벡터가 유효 블록 벡터인지를 결정하기 위해 제2 단계에서 추가로 체크된다. 예를 들어, 제2 단계에서, 블록 벡터가 좌측-중간 CTU에서의 참조 블록을 가리킬 때, 예를 들어, 수학식 10이 충족되지 않을 때, 블록 벡터는 유효 블록 벡터이다. 제2 단계에서, 참조 블록이 좌측 끝 CTU에 있을 때, 예를 들어, 수학식 10이 충족될 때, 참조 블록에 대한 병치된 블록 영역이 재구성되었는지 여부를 결정하기 위해 블록 벡터 mvL0이 체크된다.

[0156] 일 예에서, 현재 블록의 좌측-상단 코너는  $(xCb, yCb)$ 에 있고, 그 후 참조 블록의 좌측-상단 코너는  $(xCb + (mvL[0] \gg 4), yCb + (mvL[1] \gg 4))$ 에 있고, 참조 블록에 대한 병치된 블록 영역의 좌측-상단 코너는  $((xCb + (mvL[0] \gg 4) + (1 \ll 4^{\text{MaxCtbLog2SizeY}} - CtbLog2SizeY)) \gg (CtbLog2SizeY - 1) \ll (CtbLog2SizeY - 1), ((yCb + (mvL[1] \gg 4)) \gg (CtbLog2SizeY - 1)) \ll (CtbLog2SizeY - 1))$ 에 있다. 일 예에서, 병치된 블록 영역의 좌측-상단 코너는 꽉쳐의 재구성 프로세스를 추적하는 맵 상에서 체크하기 위해 사용된다. 맵 상의 위치가 일 예에서 병치된 블록 영역이 재구성되지 않았음을 지시하는 "거짓"일 때, 따라서 참조 블록은 참조 샘플 메모리에서 이용가능하고, 인트라 블록 카피 모드에서 현재 블록을 재구성하는데 사용될 수 있다. 그 다음, 블록 벡터는 유효 블록 벡터이다. 그러나, 맵 상의 위치가 일 예에서 병치된 블록 영역이 재구성되었거나 부분적으로 재구성되었음을 지시하는 "참"일 때, 블록 벡터  $mvL0$ 은 유효 블록 벡터가 아니다.

[0157] 제4 실시예에서, 제2 단계는 병치된 블록 영역에 기초하고, 제4 실시예에서의 프로세스는 블록 영역 기반 업데이트 프로세스라고 지칭된다. 제4 실시예에서, 현재 CTU 내의 (참조 블록에 대한) 병치된 블록 영역의 임의의 샘플이 재구성되었을 때, 병치된 블록 영역의 샘플들을 저장하는 참조 샘플 메모리 내의 대응하는 영역은 인트라 블록 카피 참조를 위해 이용가능하지 않을 것이다.

[0158] 본 개시내용의 일부 양태에 따르면, 좌측 끝 CTU는 검색 영역으로부터 의도적으로 배제되고, 그 후 블록 벡터  $mvL0$ 은 1-단계 제약 프로세스를 사용하여 제약되어  $mvL0$ 이 인트라 블록 카피 모드에서 참조 블록을 가리키는 유효 블록 벡터이고 참조 블록이 참조 샘플 메모리에 완전히 저장되는지를 결정한다.

[0159] 제5 실시예에서, 제3 실시예와 유사하게, CTU 크기는  $\text{Log2 도메인에서 } CtbLog2SizeY$ 이고, 따라서 참조 샘플 메모리에 저장된 재구성된 샘플들은 현재 CTU, 좌측 끝 CTU, 및 현재 CTU와 좌측 끝 CTU 사이에 위치되는 좌측-중간 CTU(들) 중 임의의 것으로부터 비롯된다. 제5 실시예에서, 블록 벡터  $mvL0$ 이 현재 CTU 및 좌측-중간 CTU(들)를 포함하는 검색 범위를 가리키는지를 결정하기 위해 1-단계에서 블록  $mvL0$ 이 체크된다. 제5 실시예에서 좌측 끝 CTU가 검색 범위로부터 배제된다는 점에 유의한다. 일부 예들에서, 블록 벡터  $mvL0$ 에 대한 제약은 수학식 11 내지 수학식 14에 의해 표현된다:

### 수학식 11

$$(yCb + (mvL0[1] \gg 4)) \gg CtbLog2SizeY = yCb \gg CtbLog2SizeY$$

### 수학식 12

$$(yCb + (mvL0[1] \gg 4) + cbHeight - 1) \gg CtbLog2SizeY = yCb \gg CtbLog2SizeY$$

### 수학식 13

$$(xCb + (mvL0[0] \gg 4)) \gg CtbLog2SizeY > (xCb \gg CtbLog2SizeY) - 1 \ll (2 * (\text{MaxCtbLog2SizeY} - CtbLog2SizeY))$$

### 수학식 14

$$(xCb + (mvL0[0] \gg 4) + cbWidth - 1) \gg CtbLog2SizeY > (xCb \gg CtbLog2SizeY) - 1 \ll (2 * (\text{MaxCtbLog2SizeY} - CtbLog2SizeY))$$

[0164] 수학식 11이 충족될 때, 참조 블록의 상단은 현재 블록과 동일한 CTU 행에 있다. 수학식 12가 충족될 때, 참조 블록의 하단은 현재 블록과 동일한 CTU 행에 있다. 수학식 13이 충족될 때, 참조 블록의 좌측은 일 예에서 현

재 CTU 및 중간 좌측 CTU(들) 중 하나와 동일한 CTU 열에 있다. 수학식 14가 충족될 때, 참조 블록의 우측은 일 예에서 현재 CTU 및 중간 좌측 CTU(들) 중 하나와 동일한 CTU 열에 있다. 따라서, 수학식 11 내지 수학식 14가 일 예에서 충족될 때, 참조 블록은 검색 범위 내에 있고, 블록 벡터 mvL0은 유효 블록 벡터이다.

[0165] 도 13은 본 개시내용의 일 실시예에 따른 프로세스(1300)를 약술하는 흐름도를 도시한다. 프로세스(1300)는, 재구성 하의 블록에 대한 예측 블록을 생성하기 위해, 인트라 모드에서 코딩된 블록의 재구성에서 사용될 수 있다. 다양한 실시예들에서, 프로세스(1300)는, 단말 디바이스들(310, 320, 330 및 340) 내의 처리 회로, 비디오 인코더(403)의 기능들을 수행하는 처리 회로, 비디오 디코더(410)의 기능들을 수행하는 처리 회로, 비디오 디코더(510)의 기능들을 수행하는 처리 회로, 비디오 인코더(603)의 기능들을 수행하는 처리 회로 등과 같은 처리 회로에 의해 실행된다. 일부 실시예들에서, 프로세스(1300)는 소프트웨어 명령어들로 구현되고, 따라서 처리 회로가 소프트웨어 명령어들을 실행할 때, 처리 회로는 프로세스(1300)를 수행한다. 프로세스는 (S1301)에서 시작되어 (S1310)으로 진행한다.

[0166] (S1310)에서, 현재 CTU에서 현재 블록의 예측 정보가 코딩된 비디오 비트스트림으로부터 디코딩된다. 예측 정보는 인트라 블록 카피 모드를 지시한다. 현재 CTU의 크기는 참조 샘플 메모리의 저장 용량에 대응하는 최대 크기보다 작다. 일부 예들에서, 참조 샘플 메모리는 코딩된 비디오 비트스트림으로부터의 재구성된 샘플들을 저장하기 위한 메인 메모리보다 더 빠른 액세스 속도를 갖는다. 예를 들어, 참조 샘플 메모리는 디코더 회로와 동일한 칩 상에 있는 온-칩 메모리이고, 메인 메모리는 디코더 회로를 갖는 칩 외부에 있는 오프-칩 메모리이다. 참조 샘플 메모리는 일부 예들에서 오프-칩 메모리를 사용하여 구현될 수 있다는 점에 유의한다.

[0167] (S1320)에서, 블록 벡터가 결정된다. 블록 벡터는 현재 블록과 동일한 광역 내의 참조 블록을 가리키고, 참조 블록은 참조 샘플 메모리에 버퍼링된 재구성된 샘플들을 갖는다. 일부 실시예들에서, 검색 영역은 현재 CTU, 중간-좌측 CTU(들) 및 좌측 끝 CTU와 같은, 참조 샘플 메모리에 버퍼링된 재구성된 샘플들을 갖는 CTU들을 포함하도록 정의된다. 좌측 끝 CTU는 참조 샘플 메모리에서 현재 CTU의 재구성된 샘플로 오버라이트된 적어도 하나의 재구성된 샘플을 갖는다.

[0168] (S1330)에서, 현재 블록은 참조 샘플 메모리로부터 검색되는 참조 블록의 재구성된 샘플들에 기초하여 재구성된다. 예를 들어, 참조 샘플 메모리는 참조 블록의 재구성된 샘플들을 검색하기 위해 액세스되고, 이후 현재 블록의 샘플들은 참조 샘플 메모리로부터 검색되는 재구성된 샘플들에 기초하여 재구성된다. 이어서, 프로세스는 (S1399)로 진행하고 종료된다.

[0169] 위에서 설명된 기법들은 컴퓨터 판독가능 명령어들을 사용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 판독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 14는 개시된 주제의 특정 실시예들을 구현하기에 적합한 컴퓨터 시스템(1400)을 도시한다.

[0170] 컴퓨터 소프트웨어는, 하나 이상의 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 사용하여 코딩될 수 있다.

[0171] 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.

[0172] 컴퓨터 시스템(1400)에 대한 도 14에 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(1400)의 예시적인 실시예에서 예시된 컴포넌트들 중 임의의 하나 또는 이들의 조합과 관련하여 임의의 종속성 또는 요건을 갖는 것으로 해석되어서도 안 된다.

[0173] 컴퓨터 시스템(1400)은 특정 휴면 인터페이스 입력 디바이스들을 포함할 수 있다. 그러한 휴면 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. 휴면 인터페이스 디바이스들은 또한 오디오(예컨대: 음성, 음악, 주변 사운드), 이미지들(예컨대: 스캐닝된 이미지들, 스템 이미지 카메라로부터 획득된 사진 이미지들), 비디오(예컨대 2 차원 비디오, 입체적 비디오를 포함하는 3차원 비디오)와 같은, 인간에 의한 의식적인 입력과 반드시 직접적으로 관련되는 것은 아닌 특정 미디어를 캡처하기 위해 사용될 수 있다.

[0174] 입력 휴면 인터페이스 디바이스들은: 키보드(1401), 마우스(1402), 트랙패드(1403), 터치 스크린(1410), 데이터

-글리브(도시되지 않음), 조이스틱(1405), 마이크로폰(1406), 스캐너(1407), 카메라(1408) 중 하나 이상(각각의 하나만이 묘사됨)을 포함할 수 있다.

[0175] 컴퓨터 시스템(1400)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(1410), 데이터-글리브(도시되지 않음), 또는 조이스틱(1405)에 의한 촉각 피드백이지만, 입력 디바이스들로서 역할하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(예컨대: 스피커들(1409), 헤드폰들(묘사되지 않음)), 시각적 출력 디바이스들(예컨대 CRT 스크린들, LCD 스크린들, 폴라즈마 스크린들, OLED 스크린들을 포함하는 스크린들(1410), 각각은 터치-스크린 입력 능력이 있거나 없고, 각각은 촉각 피드백 능력이 있거나 없고- 이들 중 일부는 스테레오그래픽 출력; 가장 현실 안경(묘사되지 않음), 홀로그래픽 디스플레이들 및 연기 텅크들(묘사되지 않음)), 및 프린터들(묘사되지 않음)과 같은 수단을 통해 2차원 시각적 출력 또는 3개보다 많은 차원의 출력을 출력할 수 있다.

[0176] 컴퓨터 시스템(1400)은 인간 액세스 가능한 저장 디바이스들 및 그것들과 연관된 매체들, 예컨대 CD/DVD 등의 매체(1421)를 갖는 CD/DVD ROM/RW(1420)를 포함하는 광학 매체, 썬-드라이브(1422), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(1423), 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.

[0177] 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.

[0178] 컴퓨터 시스템(1400)은 하나 이상의 통신 네트워크에 대한 인터페이스를 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, 예컨대 이더넷, 무선 LAN들, GSM, 3G, 4G, 5G, 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(1449)(예컨대, 예를 들어, 컴퓨터 시스템(1400)의 USB 포트들)에 부착된 외부 네트워크 인터페이스 어댑터들을 요구하고; 다른 것들은 일반적으로 아래에 설명되는 바와 같은 시스템 버스로의 부착에 의해 컴퓨터 시스템(1400)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(1400)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. 위에서 설명된 바와 같은 네트워크들 및 네트워크 인터페이스들 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0179] 전술한 휴먼 인터페이스 디바이스들, 인간-액세스 가능한 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(1400)의 코어(1440)에 부착될 수 있다.

[0180] 코어(1440)는 하나 이상의 중앙 처리 유닛(CPU)(1441), 그래픽 처리 유닛(GPU)(1442), 필드 프로그래머블 게이트 영역(FPGA)(1443)의 형식으로 특수화된 프로그래머블 처리 유닛들, 특정 태스크들에 대한 하드웨어 가속기들(1444) 등을 포함할 수 있다. 이들 디바이스는, 판독 전용 메모리(ROM)(1445), 랜덤 액세스 메모리(1446), 내부 비-사용자 액세스 가능 하드 드라이브들, SSD들 등과 같은 내부 대용량 저장소(1447)와 함께, 시스템 버스(1448)를 통해 연결될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(1448)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형식으로 액세스 가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(1448)에 직접, 또는 주변 버스(1449)를 통해 부착될 수 있다. 주변 버스를 위한 아키텍처들은 PCI, USB 등을 포함한다.

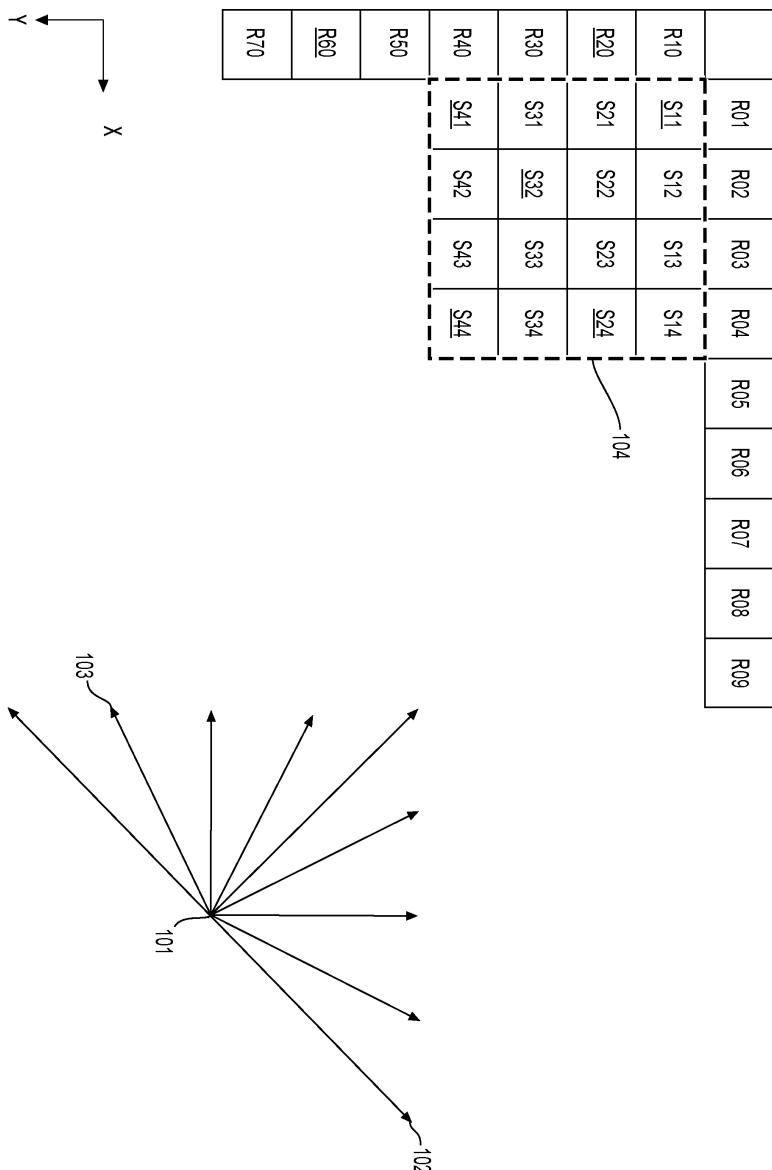
[0181] CPU들(1441), GPU들(1442), FPGA들(1443), 및 가속기들(1444)은, 조합하여, 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령어들을 실행할 수 있다. 해당 컴퓨터 코드는 ROM(1445) 또는 RAM(1446)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(1446)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 저장소(1447)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은, 하나 이상의 CPU(1441), GPU(1442), 대용량 저장소(1447), ROM(1445), RAM(1446) 등과 밀접하게 연관될 수 있는, 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

- [0182] 컴퓨터 관독가능 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 이용가능한 종류의 것일 수 있다.
- [0183] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(1400), 및 구체적으로 코어(1440)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형의(tangible) 컴퓨터 관독가능 매체에 구현된 소프트웨어를 실행하는 결과로서 기능성을 제공할 수 있다. 그러한 컴퓨터 관독가능 매체는 위에 소개된 바와 같은 사용자-액세스 가능한 대용량 저장소뿐만 아니라, 코어-내부 대용량 저장소(1447) 또는 ROM(1445)과 같은 비일시적인 본질의 것인 코어(1440)의 특정 저장소와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 그러한 디바이스들에 저장되고 코어(1440)에 의해 실행될 수 있다. 컴퓨터 관독가능 매체는 특정 필요에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(1440) 및 구체적으로 그 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(1446)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(1444))에 하드웨어링되거나 다른 방식으로 구현된 로직의 결과로서 기능성을 제공할 수 있다. 소프트웨어에 대한 참조는, 적절한 경우, 로직을 포함할 수 있고, 그 반대도 가능하다. 컴퓨터 관독가능 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.
- [0184] 부록 A: 두문자어들
- [0185] JEM: 조인트 탐색 모델
- [0186] VVC: 다목적 비디오 코딩
- [0187] BMS: 벤치마크 세트
- [0188] MV: 모션 벡터
- [0189] HEVC: 고효율 비디오 코딩
- [0190] SEI: 보조 향상 정보
- [0191] VUI: 비디오 사용성 정보
- [0192] GOP들: 픽처들의 그룹들
- [0193] TU들: 변환 유닛들
- [0194] PU들: 예측 유닛들
- [0195] CTU들: 코딩 트리 유닛들
- [0196] CTB들: 코딩 트리 블록들
- [0197] PB들: 예측 블록들
- [0198] HRD: 가상 참조 디코더
- [0199] SNR: 신호 잡음비
- [0200] CPU들: 중앙 처리 유닛들
- [0201] GPU들: 그래픽 처리 유닛들
- [0202] CRT: 음극선관
- [0203] LCD: 액정 디스플레이
- [0204] OLED: 유기 발광 다이오드

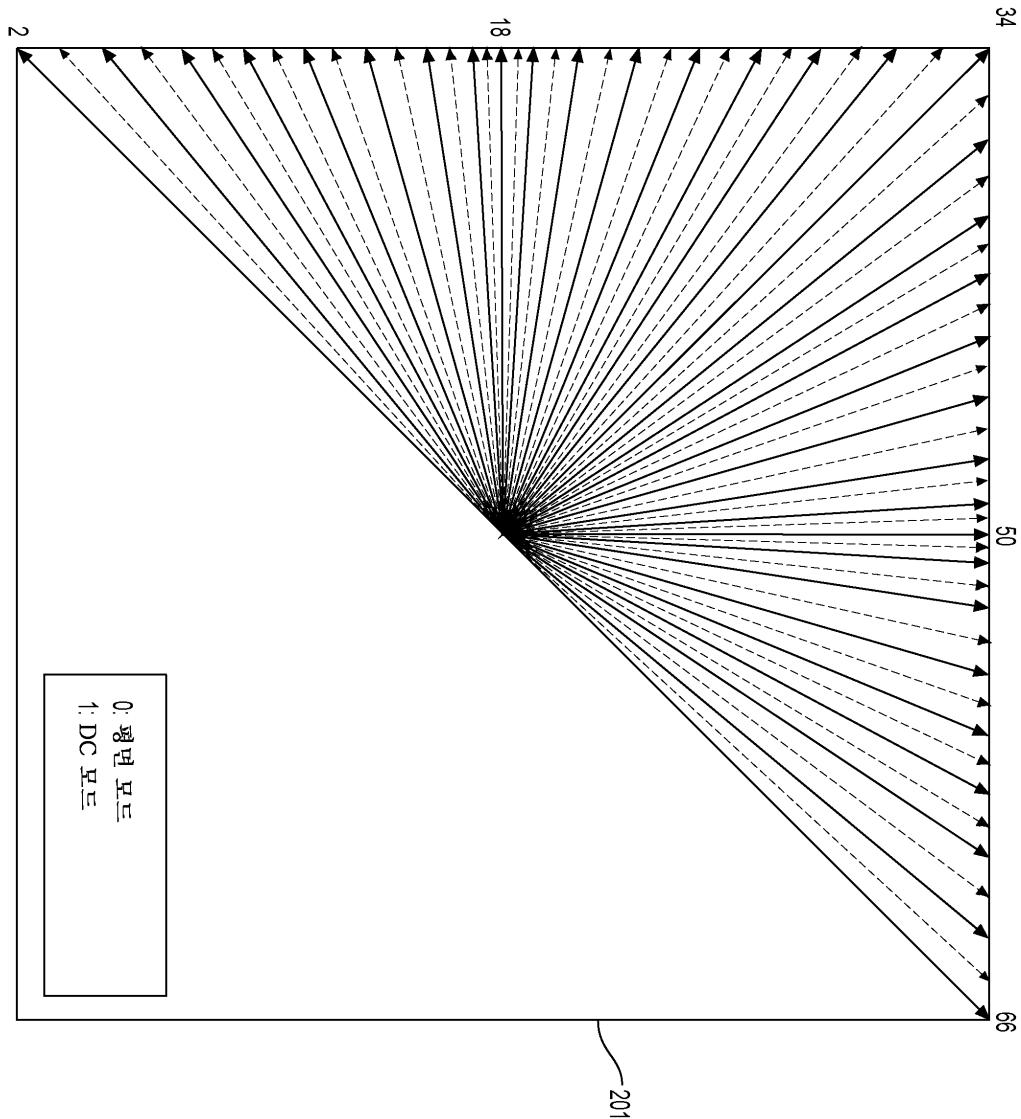
[0205]	CD: 콤팩트 디스크
[0206]	DVD: 디지털 비디오 디스크
[0207]	ROM: 판독 전용 메모리
[0208]	RAM: 랜덤 액세스 메모리
[0209]	ASIC: 애플리케이션-특정 집적 회로
[0210]	PLD: 프로그래머블 로직 디바이스
[0211]	LAN: 로컬 영역 네트워크
[0212]	GSM: 모바일 통신을 위한 글로벌 시스템
[0213]	LTE: 롱 텀 에볼루션
[0214]	CANBus: 제어기 영역 네트워크 버스
[0215]	USB: 범용 직렬 버스
[0216]	PCI: 주변 컴포넌트 인터커넥트
[0217]	FPGA: 필드 프로그래머블 게이트 영역들
[0218]	SSD: 솔리드 스테이트 드라이브
[0219]	IC: 집적 회로
[0220]	CU: 코딩 유닛
[0221]	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

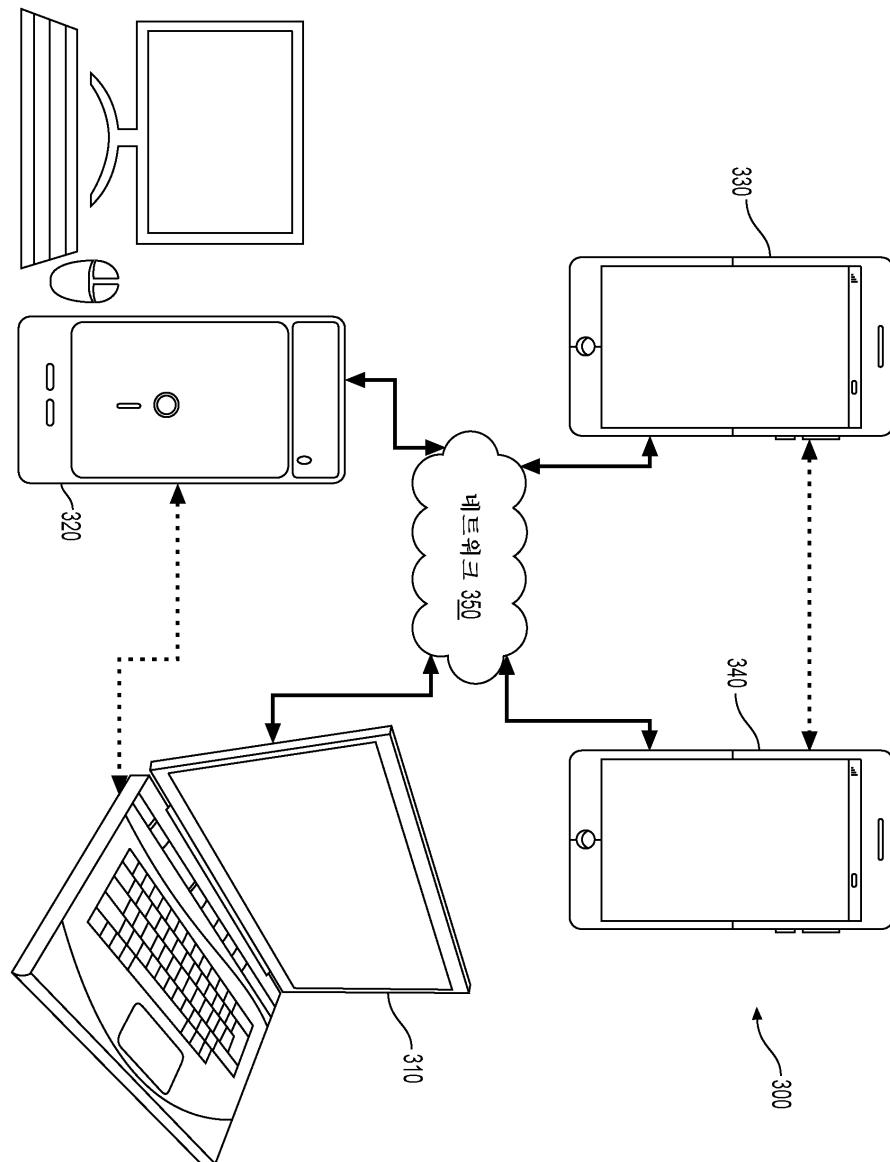
도면1



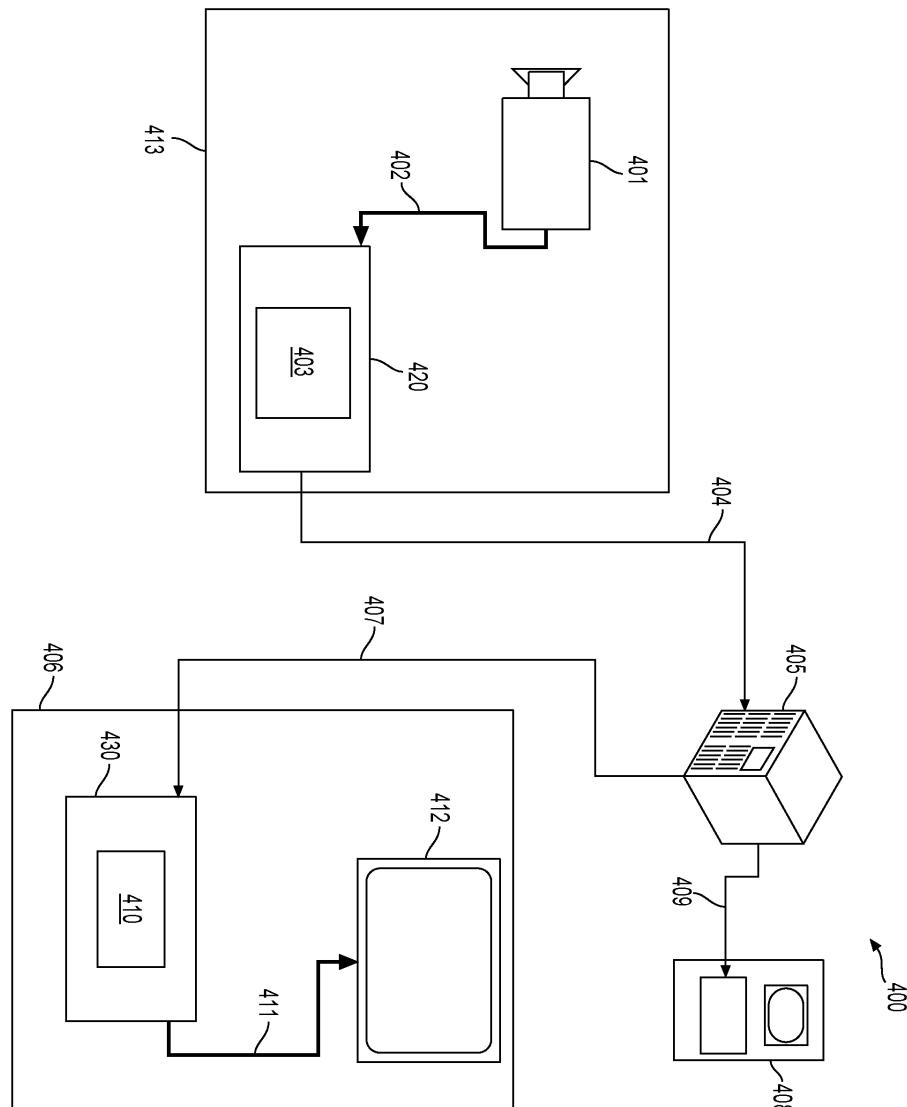
도면2



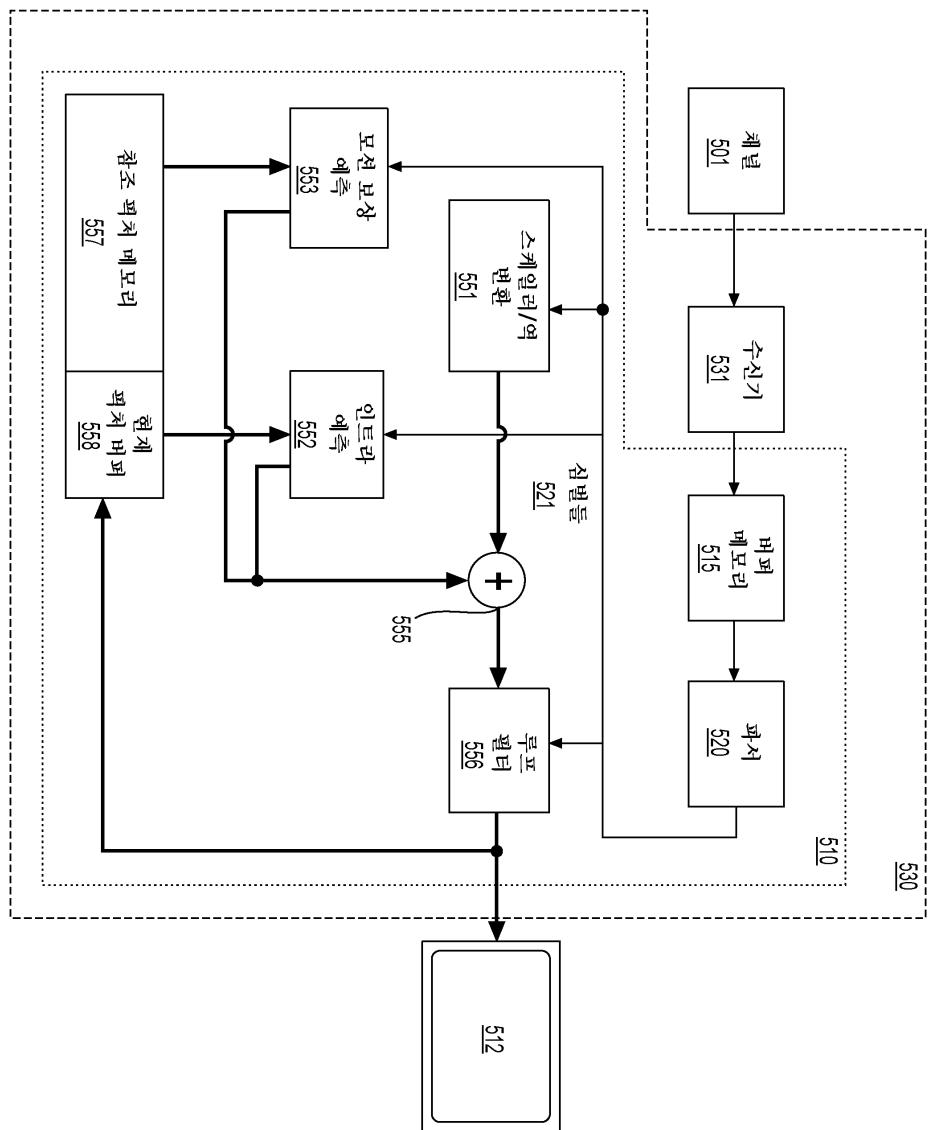
도면3



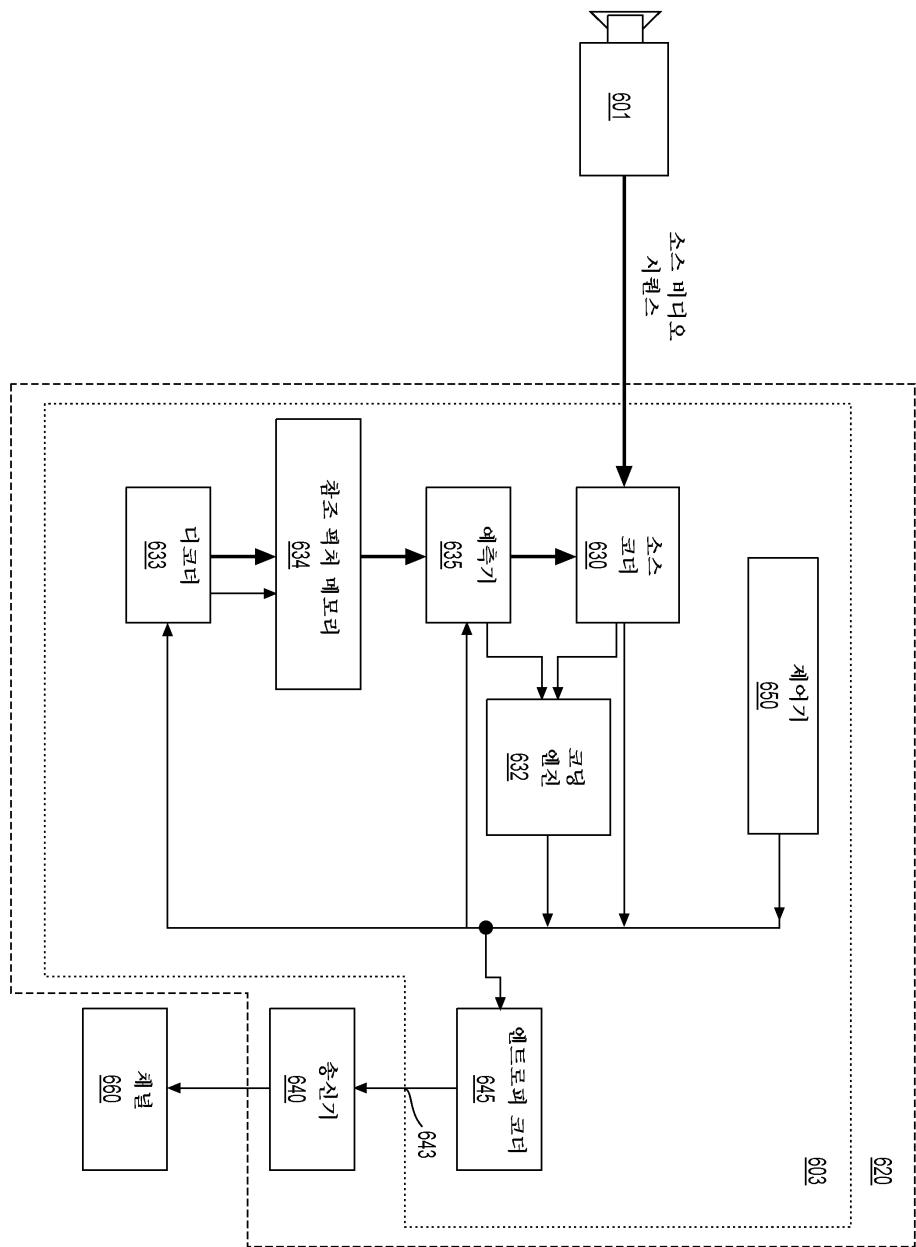
도면4



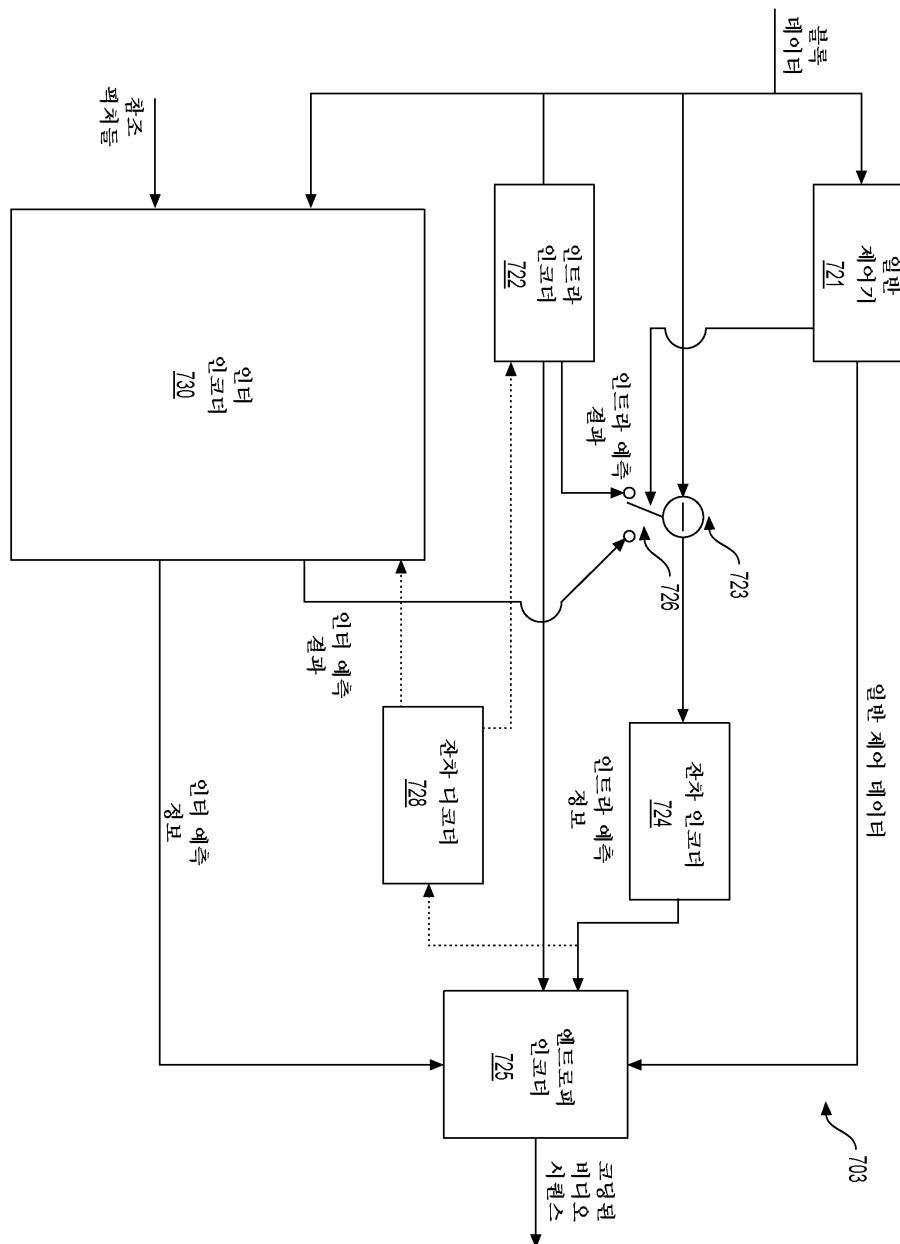
## 도면5



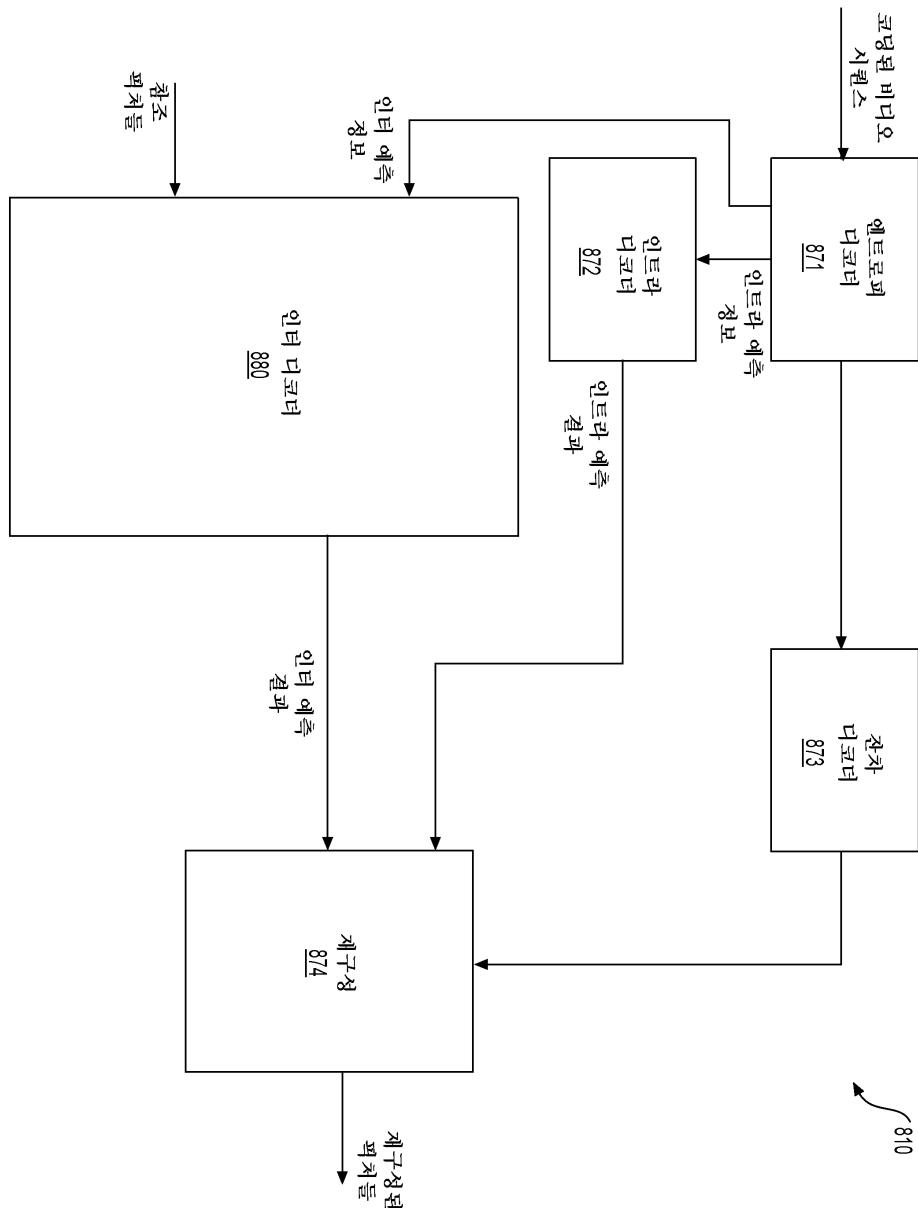
도면6



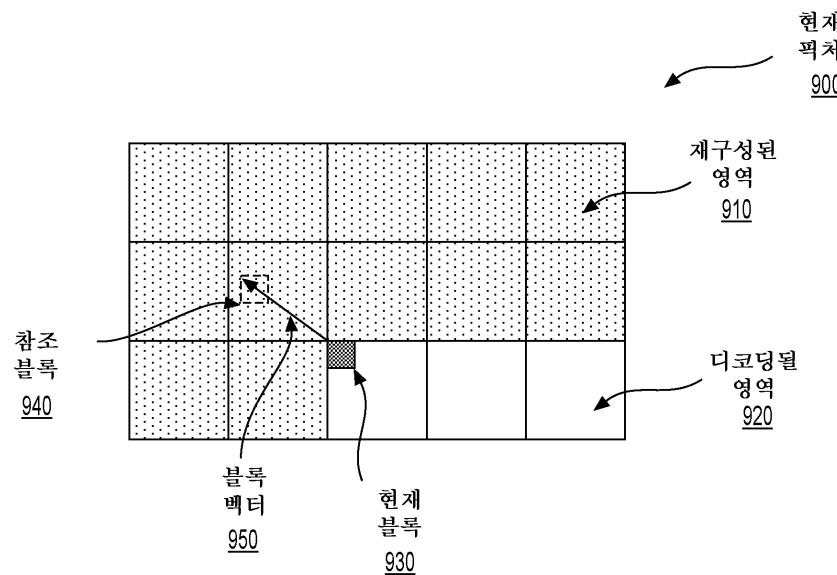
도면7



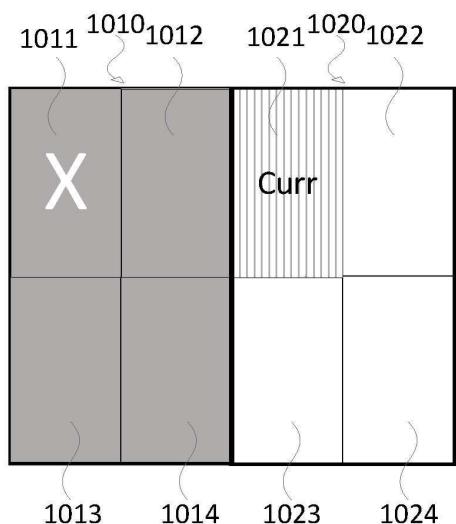
## 도면8



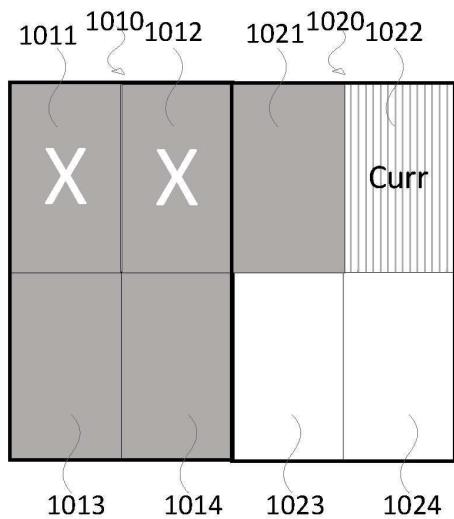
도면9



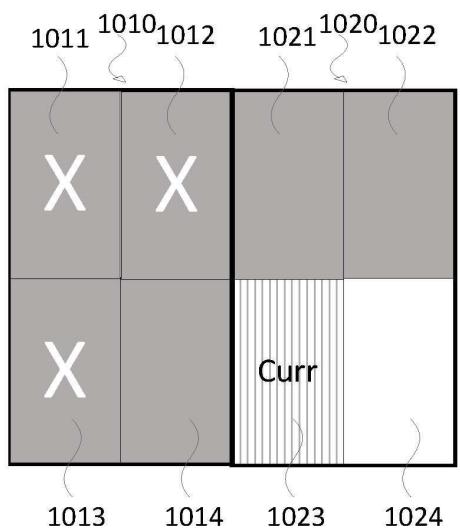
도면10a



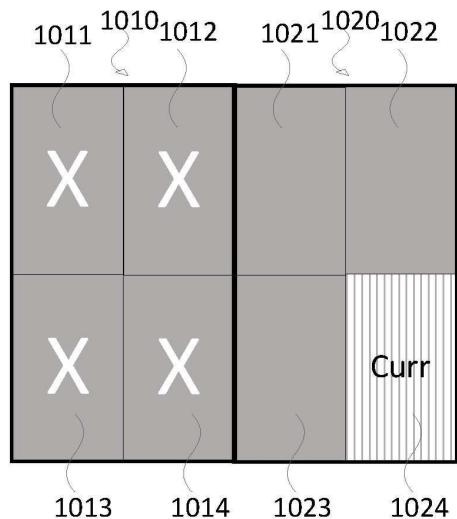
도면 10b



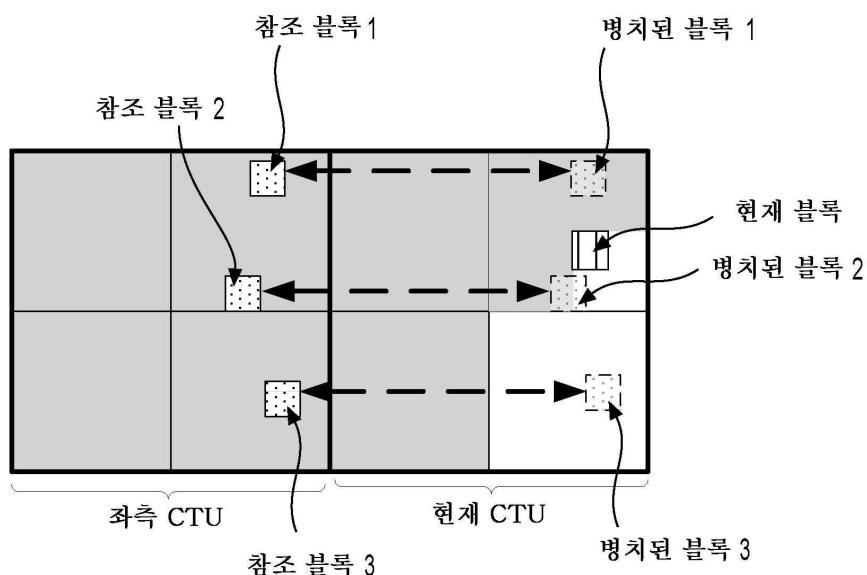
도면 10c



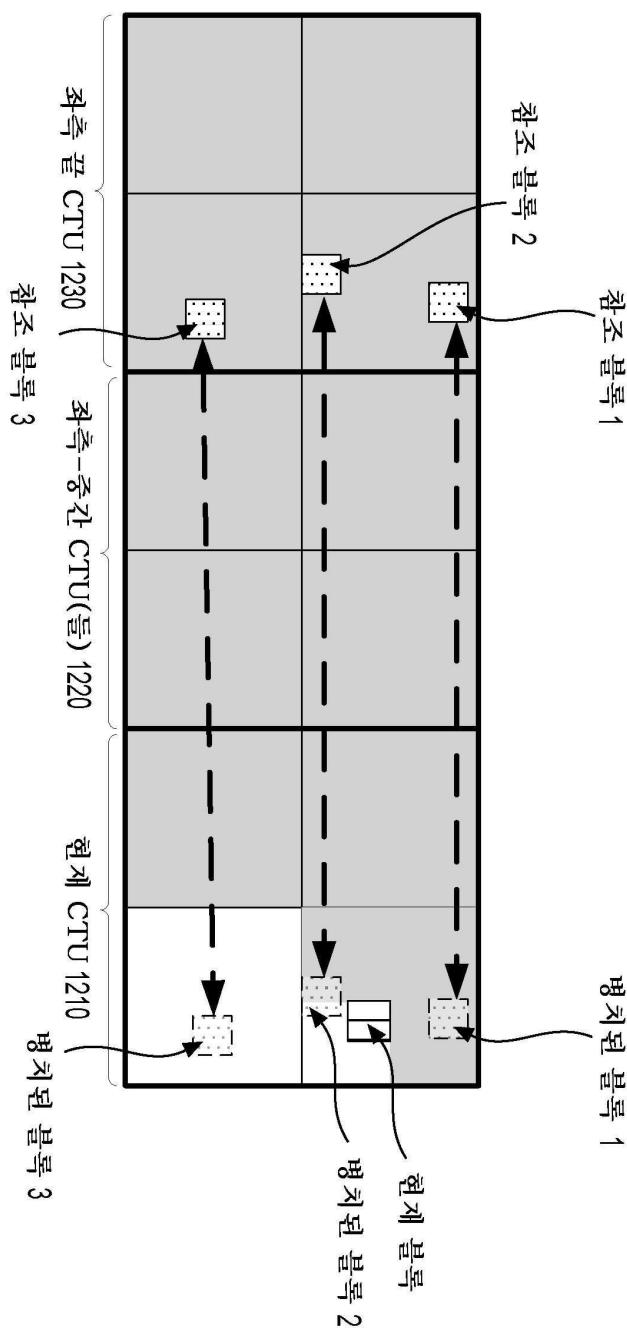
도면10d



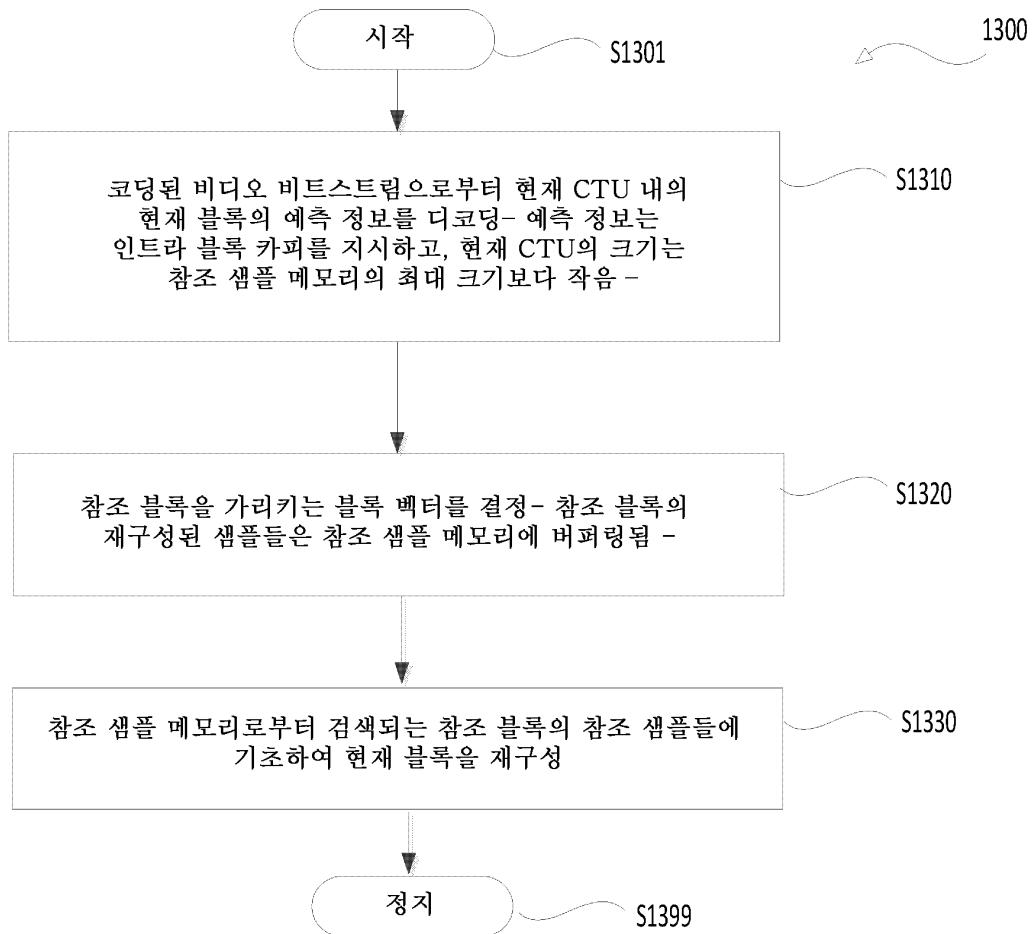
도면11



도면12



## 도면13



## 도면14

