



등록특허 10-2283407



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2021년07월28일  
(11) 등록번호 10-2283407  
(24) 등록일자 2021년07월23일

- (51) 국제특허분류 (Int. Cl.)  
*H04N 19/51* (2014.01) *H04N 19/11* (2014.01)  
*H04N 19/12* (2014.01) *H04N 19/124* (2014.01)  
*H04N 19/167* (2014.01) *H04N 19/176* (2014.01)  
*H04N 19/52* (2014.01) *H04N 19/583* (2014.01)
- (52) CPC특허분류  
*H04N 19/51* (2015.01)  
*H04N 19/11* (2015.01)
- (21) 출원번호 10-2016-7021098
- (22) 출원일자(국제) 2014년12월22일  
 심사청구일자 2019년11월21일
- (85) 번역문제출일자 2016년08월01일
- (65) 공개번호 10-2016-0105855
- (43) 공개일자 2016년09월07일
- (86) 국제출원번호 PCT/US2014/071780
- (87) 국제공개번호 WO 2015/102975  
 국제공개일자 2015년07월09일
- (30) 우선권주장  
 61/923,628 2014년01월03일 미국(US)  
 14/222,580 2014년03월21일 미국(US)
- (56) 선행기술조사문헌  
 JP2000102016 A\*  
 US20130089266 A1\*  
 \*는 심사관에 의하여 인용된 문헌

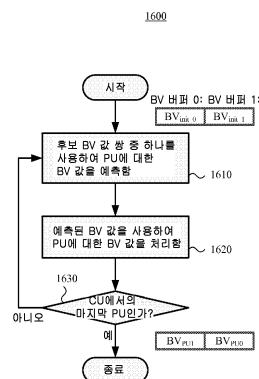
전체 청구항 수 : 총 20 항

심사관 : 박상철

(54) 발명의 명칭 중첩 구역 내에서의 재구성된 샘플 값의 블록 벡터 예측 및 추정에서의 혁신

**(57) 요 약**

인트라 BC(block copy) 예측 모드에 대한 인코더측 옵션에서의 혁신은 인코딩의 레이트 왜곡 성능 및/또는 계산 효율의 면에서 보다 효과적인 인트라 BC 예측을 용이하게 한다. 예를 들어, 혁신의 일부는 블록 벡터 추정 동안 현재 블록의 중첩 구역 내에서의 샘플 값을 추정하는 것에 관한 것이다. 다른 혁신은 "평평" 접근법을 사용하여 인코딩 또는 디코딩 동안 BV(block vector) 값을 예측하는 것에 관한 것이다.

**대 표 도** - 도16

(52) CPC특허분류

*HO4N 19/12* (2015.01)

*HO4N 19/124* (2015.01)

*HO4N 19/167* (2015.01)

*HO4N 19/176* (2015.01)

*HO4N 19/52* (2015.01)

*HO4N 19/583* (2015.01)

---

## 명세서

### 청구범위

#### 청구항 1

컴퓨팅 디바이스에 있어서,

픽처(picture)에 대한 인코딩된 데이터를 생성하기 위한 동작들을 수행하도록 구성된 인코더; 및

비트스트림의 일부로서 출력하기 위해 상기 인코딩된 데이터를 저장하도록 구성된 버퍼

를 포함하고, 상기 동작들은,

상기 픽처의 현재 블록의 중첩 구역 내에서의 재구성된 샘플 값들을 추정하는 동작 - 상기 현재 블록은 복수의 보다 작은 블록들을 가지며, 상기 중첩 구역은, 상기 복수의 보다 작은 블록들 중 적어도 하나에 대한 잠재적인 인트라 예측 영역(potential intra-prediction region)들에 있는, 상기 현재 블록의 일부를 커버(cover)하고, 상기 중첩 구역 내에서의 재구성된 샘플 값들을 추정하는 동작은, 상기 중첩 구역 밖으로부터의 재구성된 샘플 값들을 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값들로서 각각 사용하지 않고, 상기 중첩 구역 내로부터의 대응하는 원래의 샘플 값들을 사용해 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값들을 각각 결정하는 동작을 포함함 -; 및

블록 벡터(block vector; BV) 추정을 수행해 상기 현재 블록에 대한 BV 값을 결정하는 동작 - 상기 BV 값은, 상기 현재 블록부터, 예측을 위해 사용되는 샘플 값들을 포함하는 상기 픽처의 영역까지의 변위를 나타내고, 상기 BV 추정은 상기 현재 블록의 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값을 중 적어도 일부를 사용함 -

을 포함하는 것인, 컴퓨팅 디바이스.

#### 청구항 2

제1항에 있어서,

상기 중첩 구역 내로부터의 상기 대응하는 원래의 샘플 값들은, 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값들로서 각각 사용되는 것인, 컴퓨팅 디바이스.

#### 청구항 3

제1항에 있어서,

상기 중첩 구역 내로부터의 대응하는 원래의 샘플 값들을 사용해 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값들을 각각 결정하는 동작은,

주파수 변환, 양자화, 역양자화, 및 역 주파수 변환을 사용하여 상기 중첩 구역 내로부터의 상기 대응하는 원래의 샘플 값들을 처리하는 동작; 및

상기 처리된 샘플 값들을 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값들로서 각각 사용하는 동작을 포함하는 것인, 컴퓨팅 디바이스.

#### 청구항 4

제1항에 있어서,

상기 중첩 구역 내로부터의 대응하는 원래의 샘플 값들을 사용해 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값들을 각각 결정하는 동작은,

적어도 하나의 BV 예측자(BV predictor)를 사용하여 상기 중첩 구역 내에서의 샘플 값들을 예측하는 동작;

상기 예측된 샘플 값들과, 상기 중첩 구역으로부터의 상기 대응하는 원래의 샘플 값을 사이의 차이들에 기초하

여 잔차 값(residual value)들을 결정하는 동작;

주파수 변환, 양자화, 역양자화, 및 역 주파수 변환을 사용하여 상기 잔차 값을 처리하는 동작; 및

상기 처리의 결과들을 상기 예측된 샘플 값을 결합시키는 동작

을 포함하고,

상기 결합의 결과들은, 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값을 각각 사용되는 것인, 컴퓨팅 디바이스.

## 청구항 5

제1항에 있어서,

상기 BV 추정은, 상기 현재 블록에 대한 하나 이상의 후보 BV 값을 각각에 대해,

상기 후보 BV 값에 따라 상기 복수의 보다 작은 블록들과 후보 인트라 예측 영역들 간에 각각 블록 정합 동작을 수행하는 동작을 포함하는 것인, 컴퓨팅 디바이스.

## 청구항 6

제1항에 있어서, 상기 BV 추정은,

상기 복수의 보다 작은 블록들 각각에 대해, 상기 보다 작은 블록에 대한 BV 값을 결정하는 동작; 및

상기 복수의 보다 작은 블록들에 대한 BV 값을 상기 현재 블록에 대한 BV 값으로 합성하는 동작

을 포함하는 것인, 컴퓨팅 디바이스.

## 청구항 7

제1항에 있어서, 상기 동작들은,

상기 현재 블록에 대한 BV 값을 사용해 상기 현재 블록에 대한 인트라 블록 복사(block copy; BC) 예측을 수행하는 동작을 더 포함하고, 상기 인트라 BC 예측을 수행하는 동작은, 상기 복수의 보다 작은 블록들에 대해 블록 단위로 BC 동작들을 수행하는 동작을 포함하는 것인, 컴퓨팅 디바이스.

## 청구항 8

제1항에 있어서,

상기 현재 블록은 코딩 단위(coding unit)의 코딩 블록(coding block)이고, 상기 복수의 보다 작은 블록들 각각은 변환 단위(transform unit)의 변환 블록(transform block)인, 컴퓨팅 디바이스.

## 청구항 9

비디오 인코더 또는 이미지 인코더를 구현하는 컴퓨팅 디바이스에서의 방법에 있어서,

인코딩된 데이터를 생성하기 위해 꾹처를 인코딩하는 단계; 및

비트스트림의 일부로서 상기 인코딩된 데이터를 출력하는 단계

를 포함하고,

상기 인코딩하는 단계는,

상기 꾹처의 현재 블록의 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 단계 - 상기 현재 블록은 복수의 보다 작은 블록들을 가지며, 상기 중첩 구역은, 상기 복수의 보다 작은 블록들 중 적어도 하나에 대한 잠재적인 인트라 예측 영역들에 있는, 상기 현재 블록의 일부를 커버하고, 상기 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 단계는,

상기 중첩 구역 내로부터의 대응하는 원래의 샘플 값을 사용해, 상기 중첩 구역 내에서의 상기 재구성된 샘플 값을 대한 추정된 값을 각각 결정하는 단계; 및

상기 중첩 구역 밖으로부터의 재구성된 샘플 값을 상기 중첩 구역 내에서의 상기 재구성된

샘플 값들로서 각각 사용하지 않고, 상기 추정된 값들을 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용하는 단계를 포함함 -; 및

블록 벡터(block vector; BV) 추정을 수행해 상기 현재 블록에 대한 BV 값을 결정하는 단계 - 상기 BV 값은, 상기 현재 블록부터, 예측을 위해 사용되는 샘플 값들을 포함하는 상기 픽처의 영역까지의 변위를 나타내고, 상기 BV 추정은 상기 현재 블록의 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값을 중 적어도 일부를 사용함 -

을 포함하는 것인, 컴퓨팅 디바이스에서의 방법.

#### 청구항 10

제9항에 있어서,

상기 추정된 값들은 상기 중첩 구역 내로부터의 상기 대응하는 원래의 샘플 값들이고,

상기 중첩 구역 내로부터의 상기 대응하는 원래의 샘플 값들은, 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용되는 것인, 컴퓨팅 디바이스에서의 방법.

#### 청구항 11

제9항에 있어서,

상기 중첩 구역 내에서의 상기 재구성된 샘플 값들에 대한 추정된 값들을 각각 결정하는 단계는, 주파수 변환, 양자화, 역양자화, 및 역 주파수 변환을 사용하여 상기 중첩 구역으로부터의 상기 대응하는 원래의 샘플 값들을 처리하는 단계를 포함하고,

상기 처리된 샘플 값들은, 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용되는 것인, 컴퓨팅 디바이스에서의 방법.

#### 청구항 12

제9항에 있어서,

상기 중첩 구역 내에서의 상기 재구성된 샘플 값들에 대한 추정된 값들을 각각 결정하는 단계는,

적어도 하나의 BV 예측자를 사용하여 상기 중첩 구역 내에서의 샘플 값들을 예측하는 단계;

상기 예측된 샘플 값들과, 상기 중첩 구역으로부터의 상기 대응하는 원래의 샘플 값들 사이의 차이들에 기초하여 잔차 값을 결정하는 단계;

주파수 변환, 양자화, 역양자화, 및 역 주파수 변환을 사용하여 상기 잔차 값을 처리하는 단계; 및

상기 처리의 결과들을 상기 예측된 샘플 값들과 결합시키는 단계

를 포함하고,

상기 결합의 결과들은 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용되는 것인, 컴퓨팅 디바이스에서의 방법.

#### 청구항 13

제9항에 있어서,

상기 BV 추정은, 상기 현재 블록에 대한 하나 이상의 후보 BV 값 각각에 대해,

상기 후보 BV 값에 따라 상기 복수의 보다 작은 블록들과 후보 인트라 예측 영역들 간에 각각 블록 정합 동작을 수행하는 단계를 포함하는 것인, 컴퓨팅 디바이스에서의 방법.

#### 청구항 14

제9항에 있어서, 상기 BV 추정은,

상기 복수의 보다 작은 블록들 각각에 대해, 상기 보다 작은 블록에 대한 BV 값을 결정하는 단계; 및

상기 복수의 보다 작은 블록들에 대한 BV 값들을 상기 현재 블록에 대한 BV 값으로 합성하는 단계를 포함하는 것인, 컴퓨팅 디바이스에서의 방법.

### 청구항 15

제9항에 있어서,

상기 인코딩하는 단계는 상기 현재 블록에 대한 BV 값을 사용하여 상기 현재 블록에 대한 인트라 블록 복사(BC) 예측을 수행하는 단계를 더 포함하고, 상기 인트라 BC 예측을 수행하는 단계는 상기 복수의 보다 작은 블록들에 대해 블록 단위로 BC 동작들을 수행하는 단계를 포함하는 것인, 컴퓨팅 디바이스에서의 방법.

### 청구항 16

제9항에 있어서,

상기 현재 블록은 코딩 단위의 코딩 블록이고, 상기 복수의 보다 작은 블록들 각각은 변환 단위의 변환 블록인 것인, 컴퓨팅 디바이스에서의 방법.

### 청구항 17

컴퓨팅 디바이스에 의해 프로그램되었을 때 상기 컴퓨팅 디바이스로 하여금 동작들을 수행하게 하기 위한 컴퓨터 실행 가능 명령어들을 저장한 하나 이상의 컴퓨터 판독 가능 매체에 있어서,

상기 하나 이상의 컴퓨터 판독 가능 매체는 휴발성 메모리, 비휘발성 메모리, 자기 디스크, CD-ROM, 및 DVD로 이루어지는 그룹으로부터 선택된 것이고, 상기 동작들은,

인코딩된 데이터를 생성하기 위해 빅처를 인코딩하는 동작; 및

비트스트림의 일부로서 상기 인코딩된 데이터를 출력하는 동작

을 포함하고, 상기 인코딩하는 동작은,

상기 빅처의 현재 블록의 중첩 구역 내에서의 재구성된 샘플 값들을 추정하는 동작 - 상기 현재 블록은 복수의 보다 작은 블록들을 가지며, 상기 중첩 구역은, 상기 복수의 보다 작은 블록들 중 적어도 하나에 대한 잠재적인 인트라 예측 영역들에 있는 상기 현재 블록의 일부를 커버하고, 상기 중첩 구역 내에서의 재구성된 샘플 값들을 추정하는 동작은,

상기 중첩 구역 내로부터의 대응하는 원래의 샘플 값들을 사용해, 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들에 대한 추정된 값들을 각각 결정하는 동작; 및

상기 중첩 구역 밖으로부터의 재구성된 샘플 값들을 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용하지 않고, 상기 추정된 값들을 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용하는 동작을 포함함 -; 및

블록 벡터(BV) 추정을 수행해 상기 현재 블록에 대한 BV 값을 결정하는 동작 - 상기 BV 값은, 상기 현재 블록부터, 예측을 위해 사용된 샘플 값들을 포함하는 상기 빅처의 영역까지의 변위를 나타내고, 상기 BV 추정은 상기 현재 블록의 상기 중첩 구역 내에서의 상기 추정된 재구성된 샘플 값들 중 적어도 일부를 사용함 - 을 포함하는 것인, 컴퓨터 판독 가능 매체.

### 청구항 18

제17항에 있어서,

상기 추정된 값들은 상기 중첩 구역 내로부터의 상기 대응하는 원래의 샘플 값들이고,

상기 중첩 구역 내로부터의 상기 대응하는 원래의 샘플 값들은, 상기 중첩 구역 내에서 상기 재구성된 샘플 값들로서 각각 사용되는 것인, 컴퓨터 판독 가능 매체.

### 청구항 19

제17항에 있어서,

상기 중첩 구역 내에서의 상기 재구성된 샘플 값들에 대한 추정된 값들을 각각 결정하는 동작은, 주파수 변환, 양자화, 역양자화, 및 역 주파수 변환을 사용하여 상기 중첩 구역으로부터의 상기 대응하는 원래의 샘플 값들을 처리하는 동작을 포함하고,

상기 처리된 샘플 값들은, 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용되는 것인, 컴퓨터 판독 가능 매체.

## 청구항 20

제17항에 있어서, 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들에 대한 추정된 값들을 각각 결정하는 동작은,

적어도 하나의 BV 예측자를 사용하여 상기 중첩 구역 내에서의 샘플 값들을 예측하는 동작;

상기 예측된 샘플 값들과, 상기 중첩 구역으로부터 상기 대응하는 원래의 샘플 값들 사이의 차이들에 기초하여 잔차 값들을 결정하는 동작;

주파수 변환, 양자화, 역양자화, 및 역 주파수 변환을 사용하여 상기 잔차 값들을 처리하는 동작; 및

상기 처리의 결과들을 상기 예측된 샘플 값들과 결합시키는 동작

을 포함하고,

상기 결합의 결과들은 상기 중첩 구역 내에서의 상기 재구성된 샘플 값들로서 각각 사용되는 것인, 컴퓨터 판독 가능 매체.

## 발명의 설명

### 기술 분야

### 배경 기술

[0001] 엔지니어는 디지털 비디오의 비트 레이트(bit rate)를 감소시키기 위해 압축(compression)(소스 코딩(source coding) 또는 소스 인코딩(source encoding)이라고도 불리움)을 사용한다. 압축은 비디오 정보를 보다 낮은 비트 레이트 형태로 변환함으로써 비디오 정보를 저장하고 전송하는 비용을 감소시킨다. 압축 해제(decompression)(디코딩(decoding)이라고도 불리움)는 압축된 형태로부터 원래 정보(original information)의 베전을 재구성한다. "코덱"은 인코더/디코더 시스템이다.

[0002] 지난 20년에 걸쳐, ITU-T H.261, H.262(MPEG-2 또는 ISO/IEC 13818-2), H.263 및 H.264(MPEG-4 AVC 또는 ISO/IEC 14496-10) 표준, MPEG-1(ISO/IEC 11172-2) 및 MPEG-4 Visual(ISO/IEC 14496-2) 표준, 및 SMPTE 421M(VC-1) 표준을 비롯한, 다양한 비디오 코덱 표준이 채택되었다. 보다 최근에, HEVC 표준(ITU-T H.265 또는 ISO/IEC 23008-2)이 승인되었다. (예컨대, 스케일러블 비디오 코딩/디코딩(scalable video coding/decoding)에 대한, 샘플 비트 심도(sample bit depth) 또는 크로마 샘플링 레이트(chroma sampling rate)의 면에서 보다 높은 충실도(fidelity)를 갖는 비디오의 코딩/디코딩에 대한, 또는 멀티뷰 코딩/디코딩(multi-view coding/decoding)에 대한) HEVC 표준에 대한 확장이 현재 개발 중이다. 비디오 코덱 표준은 전형적으로, 특정의 특징이 인코딩 및 디코딩에서 사용될 때 비트스트림에서의 파라미터를 상세히 기술하는, 인코딩된 비디오 비트스트림의 구문(syntax)에 대한 옵션을 정의한다. 많은 경우에, 비디오 코덱 표준은 또한 디코더가 디코딩에서 부합하는 결과를 달성하기 위해 수행해야만 하는 디코딩 동작에 관한 상세를 제공한다. 코덱 표준 이외에, 다양한 독점적 코덱 포맷(proprietary codec format)은 인코딩된 비디오 비트스트림의 구문에 대한 다른 옵션 및 대응하는 디코딩 동작을 정의한다.

[0003] 인트라 BC(block copy: 블록 복사)는 HEVC 확장을 위해 개발 중인 예측 모드이다. 인트라 BC 예측 모드에 있어서, 픽처의 현재 블록의 샘플 값이 동일한 픽처에서의 이전에 재구성된 샘플 값을 사용하여 예측된다. BV(block vector: 블록 벡터)는 현재 블록부터 예측을 위해 사용된 이전에 재구성된 샘플 값을 포함하는 픽처의 영역까지의 변위(displacement)를 나타낸다. BV가 비트스트림에서 신호된다. 인트라 BC 예측은 인트라 픽처 예측의 일 형태이다 - 픽처의 블록에 대한 인트라 BC 예측은 동일한 픽처 내의 샘플 값 이외의 어떤 샘플 값도

사용하지 않는다 -.

[0004] HEVC 표준에 대한 초안 확장에 현재 명시되고 HEVC 표준에 대한 초안 확장에 대해 일부 참조 소프트웨어 (reference software)에서 구현되는 바와 같이, 인트라 BC 예측 모드는 몇 가지 문제점이 있다. 예를 들어, 인트라 BC 예측을 어떻게 사용할지에 관한 인코더측 결정이 효과적으로 행해지지 않는다. 다른 예로서, 많은 시나리오에서 BV 값이 효율적으로 신호되지 않는다.

### 발명의 내용

[0005] 요약하면, 상세한 설명은 인트라 BC(block copy) 예측 모드에 대한 인코더측 동작에서의 혁신을 제시한다. 예를 들어, 혁신의 일부는 BV(block vector) 추정 동안 현재 블록의 중첩 구역(overlap area) 내에서의 재구성된 샘플 값을 추정하는 방식에 관한 것이다. 다른 혁신은, 인코더 또는 디코더가 블록에 대한 BV 값을 예측할 때 후보 BV 값 쌍 중에서 선택하는, "핑퐁(ping-pong)" 접근법을 사용하여 인코딩 또는 디코딩 동안 BV 값을 예측하는 것에 관한 것이다.

[0006] 본원에 기술되는 혁신의 일 양태에 따르면, 인코더 또는 디코더는 현재 빙처의 CU(coding unit: 코딩 단위)를 처리하고, 여기서 CU는 4 개의 PU(prediction unit: 예측 단위)를 갖는다. CU는, 일반적으로, 인코딩 및 디코딩하기 위한 하나 이상의 블록의 세트이고, PU들 각각은 예측 정보 및/또는 예측 처리를 신호하기 위한 CU 내의 하나 이상의 블록의 세트이다. 4 개의 PU 각각에 대해, 인코더 또는 디코더는, (1) 후보 BV 값 쌍 중 하나를 사용하여 BV 값을 예측하고 - 여기서 플래그 값은 후보 BV 값 쌍 중에서의 선택을 나타냄 -, (2) 예측된 BV 값을 사용하여 PU에 대한 BV 값을 처리한다. 후보 BV 값 쌍이 CU의 상이한 PU에 대해 상이할 수 있다. 예를 들어, 첫 번째 PU에 대해, 후보 BV 값 쌍은 제1 초기 후보 BV 값( $BV_{init\_0}$ 이라고 표시됨) 및 제2 초기 후보 BV 값( $BV_{init\_1}$ 이라고 표시됨)을 포함한다. 두 번째 PU에 대해, 후보 BV 값 쌍은 첫 번째 PU에 대한 BV 값 및  $BV_{init\_0}$ 을 포함한다. 세 번째 PU에 대해, 후보 BV 값 쌍은 현재 CU의 첫 번째 및 두 번째 PU에 대한 BV 값을 포함한다. 마지막으로, 네 번째 PU에 대해, 후보 BV 값 쌍은 현재 CU의 두 번째 및 세 번째 PU에 대한 BV 값을 포함한다. 그렇지만, 현재 빙처의 후속 CU에 대해, 후보 BV 값 쌍은 (현재 CU의 첫 번째 및 두 번째 PU에 대한 BV 값이 현재 CU의 세 번째 및 네 번째 PU에 대한 BV 값보다 먼저 처리되었더라도) 현재 CU의 첫 번째 및 두 번째 PU에 대한 BV 값을 포함한다.

[0007] 본원에 기술되는 혁신의 다른 양태에 따르면, 인코더 또는 디코더는 현재 빙처의 다수의 CU를 처리한다. CU들 중 적어도 하나는 단일의 PU를 갖고, CU들 중 적어도 하나는 다수의 PU를 갖는다. 다수의 CU 중 주어진 CU에 대해, 주어진 CU가 단일의 PU를 갖는 경우, 인코더 또는 디코더는 (1) 후보 BV 값 쌍(제1 BV 버퍼에 저장된 제1 초기 후보 BV 값 및 제2 BV 버퍼에 저장된 제2 초기 후보 BV 값을 포함함) 중 하나를 사용하여 BV 값을 예측하고 - 여기서 플래그 값은 후보 BV 값 쌍 중에서의 선택을 나타냄 -, (2) 예측된 BV 값을 사용하여 단일의 PU에 대한 BV 값을 처리한다. 그렇게 함에 있어서, 인코더 또는 디코더는 PU에 대한 BV 값이 제1 BV 버퍼에 저장된 BV 값과 같은지에 따라 제1 BV 버퍼 및 제2 BV 버퍼를 선택적으로 업데이트한다.

[0008] 그렇지 않은 경우(주어진 CU가 다수의 PU를 가짐), PU들 각각에 대해, 인코더 또는 디코더는 (1) 후보 BV 값 쌍 중 하나를 사용하여 BV 값을 예측하고 - 여기서 플래그 값은 후보 BV 값 쌍 중에서의 선택을 나타냄 -, (2) 예측된 BV 값을 사용하여 PU에 대한 BV 값을 처리한다. 후보 BV 값 쌍이 상이한 PU에 대해 상이할 수 있다. PU가 주어진 CU의 첫 번째 또는 두 번째 PU인 경우, 인코더 또는 디코더는 PU에 대한 BV 값이 제1 BV 버퍼에 저장된 BV 값과 같은지에 따라 제1 BV 버퍼 및 제2 BV 버퍼를 선택적으로 업데이트한다. 주어진 CU의 세 번째 PU 또는 네 번째 PU에 대해, 인코더는 BV 버퍼를 업데이트하는 것을 건너뛴다.

[0009] 본원에 기술되는 혁신의 다른 양태에 따르면, 인코더는 빙처의 현재 블록의 중첩 구역 내에서의 재구성된 샘플 값을 추정한다. 현재 블록은 다수의 보다 작은 블록을 갖는다. 중첩 구역은 다수의 보다 작은 블록 중 적어도 하나에 대한 잠재적인 인트라 예측 영역(intra-prediction region)에 있는 현재 블록의 일부를 커버한다. 인코더는 이어서 현재 블록에 대한 BV 값을 결정하기 위해 BV 추정을 수행한다. BV 추정은 현재 블록의 중첩 구역 내에서의 추정된 재구성된 샘플 값을 중 적어도 일부를 사용한다.

[0010] BV 예측 및 인트라 BC 예측 모드에 대한 인코더측 옵션에 대한 혁신은 방법의 일부로서, 방법을 수행하도록 구성된 컴퓨팅 디바이스의 일부로서, 또는 컴퓨팅 디바이스로 하여금 방법을 수행하게 하기 위한 컴퓨터 실행 가능 명령어를 저장하는 유형적 컴퓨터 판독 가능 매체의 일부로서 구현될 수 있다. 다양한 혁신들이 결합하여 또는 개별적으로 사용될 수 있다.

[0011] 본 발명의 전술한 목적, 특징 및 장점 그리고 다른 목적, 특징 및 장점이 첨부 도면을 참조하여 계속되는 이하의 발명을 실시하기 위한 구체적인 내용으로부터 보다 명백하게 될 것이다.

### 도면의 간단한 설명

[0012] 도 1은 일부 기술된 실시예가 구현될 수 있는 예시적인 컴퓨팅 시스템을 나타낸 도면. 도 2a 및 도 2b는 일부 기술된 실시예가 구현될 수 있는 예시적인 네트워크 환경을 나타낸 도면. 도 3은 일부 기술된 실시예가 구현될 수 있는 예시적인 인코더 시스템을 나타낸 도면. 도 4는 일부 기술된 실시예가 구현될 수 있는 예시적인 디코더 시스템을 나타낸 도면. 도 5a 및 도 5b는 일부 기술된 실시예가 구현될 수 있는 예시적인 비디오 인코더를 나타낸 도면. 도 6은 일부 기술된 실시예가 구현될 수 있는 예시적인 비디오 디코더를 나타낸 도면. 도 7은 픽처의 블록에 대한 인트라 BC 예측을 나타낸 도면. 도 8은 BV 값에 대한 검색 범위(search range)에 관한 예시적인 제약조건을 나타낸 도면. 도 9는 BV 값의 선택에 관한 하나 이상의 제약조건 하에서, 인트라 BC 예측 모드를 사용해 인코딩하는 일반화된 기법을 나타낸 플로우차트. 도 10은 픽처의 블록에 대한 예시적인 z-스캔 순서(z-scan order)를 나타낸 도면. 도 11은 BV 범퍼에서의 후보 BV 값을 나타낸 도면이고, 도 12는 다른 데이터 구조에서의 후보 BV 값을 나타낸 도면. 도 13은 예시적인 평평 접근법에 따른, 2Nx2N CU의 2Nx2N PU에 대한 BV 예측의 일 예를 나타낸 도면. 도 14는 예시적인 평평 접근법에 따른, 2Nx2N CU의 NxN PU에 대한 BV 예측의 예를 나타낸 도면. 도 15는 예시적인 평평 접근법에 따른, 2Nx2N CU의 Nx2N PU 또는 2NxN PU에 대한 BV 예측의 예를 나타낸 도면. 도 16은 예시적인 평평 접근법에 따른, 다수의 PU를 갖는 CU에 대한 BV 값을 예측하는 예시적인 기법을 나타낸 플로우차트. 도 17은 예시적인 평평 접근법에 따른, CU에 대한 BV 값을 예측하는 예시적인 기법을 나타낸 플로우차트. 도 18은 BV 추정 동안 블록들을 보다 큰 블록으로 선택적으로 병합하는 일반화된 기법을 나타낸 플로우차트. 도 19는 BV 추정 동안 블록들을 보다 큰 블록으로 선택적으로 병합하는 것의 장점을 나타낸 도면. 도 20은 블록에 대해 BV 추정과 블록 분할 결정을 동시에 수행하는 일반화된 기법을 나타낸 플로우차트. 도 21은 블록에 대해 후보 BV 값과 블록 분할 결정을 동시에 평가하는 예시적인 기법을 나타낸 플로우차트. 도 22a 및 도 22b는 BV 추정 동안 현재 블록의 중첩 구역을 나타낸 도면. 도 23은 인코더가 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 예시적인 BV 추정 기법을 나타낸 플로우차트. 도 24는 BV 추정 동안 YUV 4:2:2 포맷의 비디오에 대한 최소 변환 크기를 갖는 블록의 중첩 구역을 나타낸 도면.

### 발명을 실시하기 위한 구체적인 내용

[0013] 발명을 실시하기 위한 구체적인 내용은 인코딩 및 디코딩 동안의 BV(block vector) 예측에서의 그리고 디코딩 동안의 인트라 BC(block copy) 예측 모드에 대한 인코더측 결정에서의 혁신을 제시한다. 상세하게는, 발명을 실시하기 위한 구체적인 내용은 BV 추정 동안 현재 블록의 중첩 구역 내에서 샘플 값을 추정하는 것에 대한 혁신을 제시한다. 발명을 실시하기 위한 구체적인 내용은 또한, 인코더 또는 디코더가 블록에 대한 BV 값을 예측할 때 후보 BV 값 쌍 중에서 선택하는, "평평" 접근법을 사용하는 인코딩 또는 디코딩 동안의 BV 예측에 대한 혁신을 제시한다.

[0014] 본원에 기술되는 동작이 여러 곳에서 비디오 인코더 또는 비디오 디코더에 의해 수행되는 것으로 기술되어 있지

만, 많은 경우에, 동작이 다른 유형의 미디어 처리 도구(예컨대, 영상 인코더 또는 영상 디코더)에 의해 수행될 수 있다.

[0015] 본원에 기술되는 혁신들 중 일부는 HEVC 표준에 특유한 구문 요소(syntax element) 및 동작을 참조하여 설명된다. 예를 들어, HEVC 표준의 초안 버전 JCTVC-01005 - "High Efficiency Video Coding (HEVC) Range Extensions Text Specification: Draft 5," JCTVC-01005-v3, November 2013 - 이 참조된다. 본원에 기술되는 혁신은 또한 다른 표준 또는 포맷에 대해 구현될 수 있다.

[0016] 보다 일반적으로, 본원에 기술되는 예에 대한 다양한 대안이 가능하다. 예를 들어, 본원에 기술되는 방법들 중 일부가 기술된 방법 동작들의 순서를 변경하는 것에 의해, 특정의 방법 동작을 분할, 반복 또는 생략하는 것에 의해, 기타에 의해 수정될 수 있다. 개시되는 기술의 다양한 양태가 결합하여 또는 개별적으로 사용될 수 있다. 상이한 실시예가 기술되는 혁신들 중 하나 이상을 사용한다. 본원에 기술되는 혁신들 중 일부는 배경 기술에서 살펴본 문제점들 중 하나 이상을 해결한다. 전형적으로, 주어진 기법/도구가 이러한 문제점들 모두를 해결하지는 않는다.

## I. 예시적인 컴퓨팅 시스템

[0018] 도 1은 기술되는 혁신들 중 몇몇이 구현될 수 있는 적당한 컴퓨팅 시스템(100)의 일반화된 예를 나타낸 것이다. 혁신들이 다양한 범용 또는 특수 목적 컴퓨팅 시스템에서 구현될 수 있기 때문에, 컴퓨팅 시스템(100)은 용도 또는 기능의 범주에 관한 어떤 제한을 암시하는 것으로 의도되어 있지 않다.

[0019] 도 1을 참조하면, 컴퓨팅 시스템(100)은 하나 이상의 처리 유닛(110, 115) 및 메모리(120, 125)를 포함한다. 처리 유닛(110, 115)은 컴퓨터 실행 가능 명령어를 실행한다. 처리 유닛은 범용 CPU(central processing unit), ASIC(application-specific integrated circuit) 내의 프로세서, 또는 임의의 다른 유형의 프로세서일 수 있다. 다중 처리 시스템(multi-processing system)에서는, 처리 능력을 증가시키기 위해 다수의 처리 유닛이 컴퓨터 실행 가능 명령어를 실행한다. 예를 들어, 도 1은 중앙 처리 유닛(central processing unit)(110)은 물론, 그래픽 처리 유닛 또는 코프로세싱 유닛(coprocessing unit)(115)을 나타내고 있다. 유형적 메모리(tangible memory)(120, 125)는 처리 유닛(들)에 의해 액세스 가능한, 휘발성 메모리(예컨대, 레지스터, 캐시, RAM), 비휘발성 메모리(예컨대, ROM, EEPROM, 플래시 메모리 등), 또는 이 둘의 어떤 조합일 수 있다. 메모리(120, 125)는 BV 예측 및/또는 BV 추정 동안 현재 블록의 중첩 구역 내에서 샘플 값을 추정하는 것, 또는 인트라 BC 예측 모드에 대한 다른 인코더측 옵션에 대한 하나 이상의 혁신을, 처리 유닛(들)에 의한 실행에 적당한 컴퓨터 실행 가능 명령어의 형태로, 구현하는 소프트웨어(180)를 저장한다.

[0020] 컴퓨팅 시스템은 부가의 특징을 가질 수 있다. 예를 들어, 컴퓨팅 시스템(100)은 저장소(140), 하나 이상의 입력 디바이스(150), 하나 이상의 출력 디바이스(160), 및 하나 이상의 통신 연결(170)을 포함한다. 버스, 제어기 또는 네트워크와 같은 상호연결 메커니즘(도시되지 않음)은 컴퓨팅 시스템(100)의 구성요소들을 상호연결시킨다. 전형적으로, 운영 체제 소프트웨어(도시되지 않음)는 컴퓨팅 시스템(100)에서 실행 중인 다른 소프트웨어에 대한 운영 환경을 제공하고, 컴퓨팅 시스템(100)의 구성요소의 활동을 조정한다.

[0021] 유형적 저장소(140)는 이동식 또는 비이동식일 수 있고, 정보를 저장하는 데 사용될 수 있고 컴퓨팅 시스템(100) 내에서 액세스될 수 있는, 자기 디스크, 자기 테이프 또는 카세트, CD-ROM, DVD, 또는 임의의 다른 매체를 포함한다. 저장소(140)는 BV 예측 및/또는 BV 추정 동안 현재 블록의 중첩 구역 내에서 샘플 값을 추정하는 것, 또는 인트라 BC 예측 모드에 대한 다른 인코더측 옵션에 대한 하나 이상의 혁신을 구현하는 소프트웨어(180)에 대한 명령어를 저장한다.

[0022] 입력 디바이스(들)(150)는 컴퓨팅 시스템(100)에 입력을 제공하는 키보드, 마우스, 펜, 또는 트랙볼과 같은 터치 입력 디바이스, 음성 입력 디바이스, 스캐닝 디바이스, 또는 다른 디바이스일 수 있다. 비디오의 경우, 입력 디바이스(들)(150)는 아날로그 또는 디지털 형태로 비디오 입력을 받아들이는 카메라, 비디오 카드, TV 투너 카드, 또는 유사한 디바이스, 또는 비디오 입력을 컴퓨팅 시스템(100) 내로 읽어들이는 CD-ROM 또는 CD-RW일 수 있다. 출력 디바이스(들)(160)는 컴퓨팅 시스템(100)으로부터의 출력을 제공하는 디스플레이, 프린터, 스피커, CD 라이터(CD-writer), 또는 다른 디바이스일 수 있다.

[0023] 통신 연결(들)(170)은 통신 매체를 통한 다른 컴퓨팅 엔터티와의 통신을 가능하게 한다. 통신 매체는 컴퓨터 실행 가능 명령어, 오디오 또는 비디오 입력 또는 출력, 또는 다른 데이터와 같은 정보를 피변조 데이터 신호(modulated data signal)로 전달한다. 피변조 데이터 신호는 신호의 특성들 중 하나 이상이 정보를 그 신호에 인코딩하는 방식으로 설정되거나 변경된 신호이다. 제한이 아닌 예로서, 통신 매체는 전기, 광, RF, 또는 다른

반송파를 사용할 수 있다.

[0024] 혁신이 일반적으로 컴퓨터 관독 가능 매체와 관련하여 기술되어 있을 수 있다. 컴퓨터 관독 가능 매체는 컴퓨팅 환경 내에서 액세스될 수 있는 임의의 이용 가능한 유형적 매체이다. 제한이 아닌 예로서, 컴퓨팅 시스템(100)에서, 컴퓨터 관독 가능 매체는 메모리(120, 125), 저장소(140), 및 상기한 것들 중 임의의 것의 조합을 포함한다.

[0025] 혁신이 일반적으로 컴퓨팅 시스템에서 실제 또는 가상의 대상 프로세서 상에서 실행되는, 프로그램 모듈에 포함된 것과 같은, 컴퓨터 실행 가능 명령어와 관련하여 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정의 작업을 수행하거나 특정의 추상 데이터 형식을 구현하는 루틴, 프로그램, 라이브러리, 객체, 클래스, 구성요소, 데이터 구조 등을 포함한다. 프로그램 모듈의 기능이 다양한 실시예에서 원하는 바에 따라 프로그램 모듈들 간에 결합되거나 분할될 수 있다. 프로그램 모듈에 대한 컴퓨터 실행 가능 명령어는 로컬 또는 분산 컴퓨팅 시스템 내에서 실행될 수 있다.

[0026] "시스템"과 "디바이스"라는 용어는 본원에서 서로 바꾸어 사용될 수 있다. 문맥이 명백히 달리 나타내지 않는 한, 어느 용어도 컴퓨팅 시스템 또는 컴퓨팅 디바이스의 유형에 대한 어떤 제한을 암시하지 않는다. 일반적으로, 컴퓨팅 시스템 또는 컴퓨팅 디바이스는 로컬이거나 분산되어 있을 수 있고, 본원에 기술되는 기능을 구현하는 소프트웨어를 갖는 특수 목적 하드웨어 및/또는 범용 하드웨어의 임의의 조합을 포함할 수 있다.

[0027] 개시되는 방법이 또한 개시되는 방법들 중 임의의 것을 수행하도록 구성된 특수 컴퓨팅 하드웨어를 사용하여 구현될 수 있다. 예를 들어, 개시되는 방법은 개시되는 방법들 중 임의의 것을 구현하도록 특수 설계되거나 구성된 집적 회로(예컨대, ASIC DSP(digital signal process unit)와 같은 ASIC, GPU(graphics processing unit), 또는 FPGA(field programmable gate array)와 같은 PLD(programmable logic device))에 의해 구현될 수 있다.

[0028] 제시를 위해, 발명을 실시하기 위한 구체적인 내용은 컴퓨팅 시스템에서의 컴퓨터 동작을 기술하기 위해 "결정 한다" 및 "사용한다"와 같은 용어를 사용한다. 이 용어들은 컴퓨터에 의해 수행되는 동작에 대한 상위 레벨 추상화(high-level abstraction)이고, 사람에 의해 수행되는 동작과 혼동되어서는 안된다. 이 용어들에 대응하는 실제 컴퓨터 동작은 구현에 따라 다르다. 본원에서 사용되는 바와 같이, "최적화\*"라는 용어(최적화 및 최적화하는과 같은 변형을 포함함)는 주어진 결정 범주 하에서 옵션들 중의 선택을 지칭하고, 최적화된 선택이 확장된 결정 범주에 대한 "최상의" 또는 "최적의" 선택임을 암시하지 않는다.

## II. 예시적인 네트워크 환경

[0030] 도 2a 및 도 2b는 비디오 인코더(220) 및 비디오 디코더(270)를 포함하는 예시적인 네트워크 환경(201, 202)을 나타낸 것이다. 인코더(220) 및 디코더(270)는 적절한 통신 프로토콜을 사용하여 네트워크(250)를 통해 연결된다. 네트워크(250)는 인터넷 또는 다른 컴퓨터 네트워크를 포함할 수 있다.

[0031] 도 2a에 도시된 네트워크 환경(201)에서, 각각의 RTC(real-time communication: 실시간 통신) 도구(210)는 양방향 통신을 위해 인코더(220) 및 디코더(270) 둘 다를 포함한다. 주어진 인코더(220)는 HEVC 표준(H.265라고도 알려져 있음), SMPTE 421M 표준, ISO-IEC 14496-10 표준(H.264 또는 AVC라고도 알려져 있음), 다른 표준, 또는 독점적 포맷(proprietary format)의 변형 또는 확장과 호환되는 출력을 생성할 수 있고, 대응하는 디코더(270)는 인코더(220)부터 인코딩된 데이터를 받는다. 양방향 통신은 화상 회의, 화상 전화 통화, 또는 다른 양자간 또는 다자간 통신 시나리오의 일부일 수 있다. 도 2a에서의 네트워크 환경(201)이 2 개의 실시간 통신 도구(210)를 포함하지만, 네트워크 환경(201)이 그 대신에 다자간 통신에 참여하는 3 개 이상의 실시간 통신 도구(210)를 포함할 수 있다.

[0032] 실시간 통신 도구(210)는 인코더(220)에 의한 인코딩을 관리한다. 도 3은 실시간 통신 도구(210)에 포함될 수 있는 예시적인 인코더 시스템(300)을 나타낸 것이다. 대안적으로, 실시간 통신 도구(210)는 다른 인코더 시스템을 사용한다. 실시간 통신 도구(210)는 디코더(270)에 의한 디코딩도 관리한다. 도 4는 실시간 통신 도구(210)에 포함될 수 있는 예시적인 디코더 시스템(400)을 나타낸 것이다. 대안적으로, 실시간 통신 도구(210)는 다른 디코더 시스템을 사용한다.

[0033] 도 2b에 도시된 네트워크 환경(202)에서, 인코딩 도구(212)는 디코더(270)를 포함하는 다수의 재생 도구(214)로 전달하기 위한 비디오를 인코딩하는 인코더(220)를 포함한다. 비디오가 인코딩되어 하나의 장소로부터 하나 이상의 다른 장소로 송신되는 비디오 감시 시스템, 웹 카메라 모니터링 시스템, 원격 테스크톱 회의 프레젠테이션 또는 다른 시나리오를 위해 단방향 통신이 제공될 수 있다. 도 2b에서의 네트워크 환경(202)이 2 개의 재생 도구(214)를 포함하지만, 네트워크 환경(202)은 더 많거나 더 적은 수의 재생 도구(214)를 포함할 수 있다. 일반

적으로, 재생 도구(214)는 재생 도구(214)가 수신할 비디오 스트림을 결정하기 위해 인코딩 도구(212)와 통신한다. 재생 도구(214)는 스트림을 수신하고, 수신된 인코딩된 데이터를 적절한 기간 동안 버퍼링하며, 디코딩 및 재생을 시작한다.

[0034] 도 3은 인코딩 도구(212)에 포함될 수 있는 예시적인 인코더 시스템(300)을 나타낸 것이다. 대안적으로, 인코딩 도구(212)는 다른 인코더 시스템을 사용한다. 인코딩 도구(212)는 또한 하나 이상의 재생 도구(214)와의 연결을 관리하기 위한 서버측 제어기 논리(server-side controller logic)를 포함할 수 있다. 도 4는 재생 도구(214)에 포함될 수 있는 예시적인 디코더 시스템(400)을 나타낸 것이다. 대안적으로, 재생 도구(214)는 다른 디코더 시스템을 사용한다. 재생 도구(214)는 또한 인코딩 도구(212)와의 연결을 관리하기 위한 클라이언트측 제어기 논리(client-side controller logic)를 포함할 수 있다.

### III. 예시적인 인코더 시스템

[0035] 도 3은 일부 기술된 실시예가 구현될 수 있는 예시적인 인코더 시스템(300)의 블록도이다. 인코더 시스템(300)은 실시간 통신을 위한 저-레이턴시(low-latency) 인코딩 모드, 트랜스코딩 모드(transcoding mode), 및 파일 또는 스트림으로부터 재생을 위한 미디어를 생성하기 위한 고-레이턴시(higher-latency) 인코딩 모드와 같은 다수의 인코딩 모드 중 임의의 것에서 동작할 수 있는 범용 인코딩 도구일 수 있거나, 하나의 이러한 인코딩 모드를 위해 구성된 특수 목적 인코딩 도구일 수 있다. 인코더 시스템(300)은 운영 체제 모듈로서, 애플리케이션 라이브러리의 일부로서 또는 독립형 애플리케이션으로서 구현될 수 있다. 일반적으로, 인코더 시스템(300)은 비디오 소스(310)로부터 소스 비디오 프레임들(311)의 시퀀스를 수신하고 인코딩된 데이터를 채널(390)로의 출력으로서 생성한다. 채널로 출력되는 인코딩된 데이터는 인트라 BC 예측 모드를 사용하여 인코딩된 콘텐츠를 포함할 수 있다.

[0036] 비디오 소스(310)는 카메라, 투너 카드, 저장 매체, 또는 다른 디지털 비디오 소스일 수 있다. 비디오 소스(310)는, 예를 들어, 초당 30 프레임의 프레임 레이트(frame rate)로 비디오 프레임들의 시퀀스를 생성한다. 본원에서 사용되는 바와 같이, "프레임"이라는 용어는 일반적으로 소스, 코딩된 또는 재구성된 영상 데이터를 지칭한다. 순차 주사 비디오(progressive-scan video)의 경우, 프레임은 순차 주사 비디오 프레임(progressive-scan video frame)이다. 인터레이스 비디오(interlaced video)의 경우, 예시적인 실시예에서, 인터레이스 비디오 프레임(interlaced video frame)은 인코딩 이전에 디인터레이스(de-interlace)된다. 대안적으로, 2 개의 상보적인 인터레이스 비디오 필드가 함께 단일의 비디오 프레임으로서 인코딩되거나 2 개의 개별적으로 인코딩된 필드로서 인코딩된다. 순차 주사 비디오 프레임 또는 비월 주사 비디오 프레임(interlaced-scan video frame)을 나타내는 것 이외에, "프레임" 또는 "픽처"라는 용어는 단일의 쌍이 아닌 비디오 필드(single non-paired video field), 상보적인 비디오 필드 쌍, 주어진 때의 비디오 객체를 표현하는 비디오 객체 평면(video object plane), 또는 보다 큰 영상에서의 관심 영역을 나타낼 수 있다. 비디오 객체 평면 또는 영역은 장면의 다수의 객체 또는 영역을 포함하는 보다 큰 영상의 일부일 수 있다.

[0037] 도착하는 소스 프레임(311)은 다수의 프레임 버퍼 저장 구역(321, 322, ..., 32n)을 포함하는 소스 프레임 임시 메모리 저장 구역(320)에 저장된다. 프레임 버퍼(321, 322 등)는 소스 프레임 저장 구역(320)에 하나의 소스 프레임을 보유한다. 소스 프레임들(311) 중 하나 이상이 프레임 버퍼(321, 322 등)에 저장된 후에, 프레임 선택기(frame selector)(330)는 소스 프레임 저장 구역(320)으로부터 개개의 소스 프레임을 선택한다. 프레임이 인코더(340)에 입력하기 위해 프레임 선택기(330)에 의해 선택되는 순서는 프레임이 비디오 소스(310)에 의해 생성되는 순서와 상이할 수 있다(예컨대, 어떤 나중의 프레임이 먼저 인코딩될 수 있게 하고 따라서 시간상 역방향 예측(temporally backward prediction)을 용이하게 하기 위해, 어떤 프레임의 인코딩이 순서가 지연될 수 있다). 인코더(340)보다 앞에, 인코더 시스템(300)은 인코딩 이전에 선택된 프레임(331)의 전처리(pre-processing)(예컨대, 필터링)를 수행하는 전처리기(pre-processor)(도시되지 않음)를 포함할 수 있다. 전처리는 또한 주 성분(primary component)(예컨대, 루마(luma) 성분) 및 보조 성분(secondary component)(예컨대, 적색에 대한 색차(chroma difference) 성분 및 청색에 대한 색차 성분)로의 색 공간 변환(color space conversion) 및 (예컨대, 크로마 성분(chroma component)의 공간 분해능을 감소시키기 위한) 재샘플링 처리를 포함할 수 있다. 전형적으로, 인코딩 이전에, 비디오는 YUV와 같은 색 공간으로 변환되었고, 여기서 루마(luma)(Y) 성분의 샘플 값은 밝기 또는 휘도 값을 나타내고, 크로마(chroma)(U, V) 성분의 샘플 값은 색차 값(color-difference value)을 나타낸다. 크로마 샘플 값은 (예컨대, YUV 4:2:0 포맷 또는 YUV 4:2:2에 대해) 보다 낮은 크로마 샘플링 레이트로 서브샘플링될 수 있거나, 크로마 샘플 값은 (예컨대, YUV 4:4:4 포맷에 대해) 루마 샘플 값과 동일한 분해능을 가질 수 있다. YUV 4:2:0 포맷에서, 크로마 성분은 수평으로 2배만큼 그리고 수직으로 2배만큼 다운샘플링된다. YUV 4:2:2 포맷에서, 크로마 성분은 수평으로 2배만큼 다운샘플링된다.

다. 또는, 비디오가 다른 포맷(예컨대, RGB 4:4:4 포맷)으로 인코딩될 수 있다.

[0039] 인코더(340)는, 코딩된 프레임(341)을 생성하기 위해, 선택된 프레임(331)을 인코딩하고, 또한 MMCO(memory management control operation: 메모리 관리 제어 동작) 신호(342) 또는 RPS(reference picture set: 참조 픽처 세트) 정보를 생성한다. 현재 프레임이 인코딩된 첫 번째 프레임이 아닌 경우, 그의 인코딩 프로세스를 수행할 때, 인코더(340)는 디코딩된 프레임 임시 메모리 저장 구역(360)에 저장된 하나 이상의 이전에 인코딩된/디코딩된 프레임(369)을 사용할 수 있다. 이러한 저장된 디코딩된 프레임(369)은 현재 소스 프레임(331)의 콘텐츠의 인터 프레임 예측(inter-frame prediction)을 위한 참조 프레임(reference frame)으로서 사용된다. MMCO/RPS 정보(342)는 어느 재구성된 프레임이 참조 프레임으로서 사용될 수 있고 따라서 프레임 저장 구역에 저장되어야만 하는지를 디코더에게 알려준다.

[0040] 일반적으로, 인코더(340)는 타일들로 분할(partition)하는 것, 인트라 예측 추정 및 예측, 움직임 추정 및 보상, 주파수 변환, 양자화 및 엔트로피 코딩과 같은 인코딩 작업을 수행하는 다수의 인코딩 모듈을 포함한다. 인코더(340)에 의해 수행되는 정확한 동작은 압축 포맷(compression format)에 따라 변할 수 있다. 출력되는 인코딩된 데이터의 포맷은 HEVC 포맷(H.265), WMV(Windows Media Video) 포맷, VC-1 포맷, MPEG-x 포맷(예컨대, MPEG-1, MPEG-2, 또는 MPEG-4), H.26x 포맷(예컨대, H.261, H.262, H.263, H.264), 또는 다른 포맷의 변형 또는 확장일 수 있다.

[0041] 인코더(340)는 프레임을 동일한 크기 또는 상이한 크기의 다수의 타일로 분할할 수 있다. 예를 들어, 인코더(340)는 프레임을, 프레임 경계와 함께, 프레임 내의 타일의 수평 및 수직 경계를 정의하는 타일 행 및 타일 열을 따라 분할하고, 여기서 각각의 타일은 직사각형 영역이다. 타일은 종종 병렬 처리를 위한 옵션을 제공하기 위해 사용된다. 프레임은 또한 하나 이상의 슬라이스로서 구성될 수 있고, 여기서 슬라이스는 프레임 전체 또는 프레임의 영역일 수 있다. 슬라이스는 프레임 내의 다른 슬라이스와 독립적으로 디코딩될 수 있고, 이는 오류 내성(error resilience)을 개선시킨다. 슬라이스 또는 타일의 내용이 인코딩 및 디코딩을 위해 블록 또는 다른 샘플 세트로 추가로 분할된다.

[0042] HEVC 표준에 따른 구문에 대해, 인코더는 프레임(또는 슬라이스 또는 타일)의 내용을 코딩 트리 단위(coding tree unit)로 분할한다. CTU(coding tree unit: 코딩 트리 단위)는 루마 CTB(coding tree block: 코딩 트리 블록)로서 구성되는 루마 샘플 값 및 2 개의 크로마 CTB로서 구성되는 대응하는 크로마 샘플 값을 포함한다. CTU(및 그의 CTB)의 크기는 인코더에 의해 선택되고, 예를 들어, 64x64, 32x32 또는 16x16 샘플 값일 수 있다. CTU는 하나 이상의 코딩 단위를 포함한다. CU(coding unit)는 루마 CB(coding block: 코딩 블록) 및 2 개의 대응하는 크로마 CB를 가진다. 예를 들어, 하나의 64x64 루마 CTB 및 2 개의 64x64 크로마 CTB를 갖는 CTU(YUV 4:4:4 포맷)는 4 개의 CU로 분할될 수 있고, 각각의 CU는 하나의 32x32 루마 CB 및 2 개의 32x32 크로마 CB를 포함하고, 각각의 CU는 어쩌면 보다 작은 CU로 추가로 분할된다. 또는, 다른 예로서, 하나의 64x64 루마 CTB 및 2 개의 32x32 크로마 CTB를 갖는 CTU(YUV 4:2:0 포맷)는 4 개의 CU로 분할될 수 있고, 각각의 CU는 하나의 32x32 루마 CB 및 2 개의 16x16 크로마 CB를 포함하고, 각각의 CU는 어쩌면 보다 작은 CU로 추가로 분할된다. CU의 가장 작은 허용 가능한 크기(예컨대, 8x8, 16x16)는 비트스트림에서 신호될 수 있다.

[0043] 일반적으로, CU는 인터(inter) 또는 인트라(intra)와 같은 예측 모드를 갖는다. CU는 예측 정보(예측 모드 상세, 변위 값, 기타 등등) 및/또는 예측 처리를 신호하기 위한 하나 이상의 예측 단위를 포함한다. PU(prediction unit)는 루마 PB(prediction block: 예측 블록) 및 2 개의 크로마 PB를 가진다. 인트라 예측된 CU에 대해, PU는, CU가 가장 작은 크기(예컨대, 8x8)를 갖지 않는 한, CU와 동일한 크기를 갖는다. 그 경우에, CU에 대한 구문 요소에 의해 나타내는 바와 같이, CU는 4 개의 보다 작은 PU(예컨대, 가장 작은 CU 크기가 8x8인 경우 각각이 4x4임)로 분할될 수 있거나, PU가 가장 작은 CU 크기를 가질 수 있다. CU는 또한 잔차 코딩/디코딩(residual coding/decoding)을 위한 하나 이상의 변환 단위(transform unit)를 가지며, 여기서 TU(transform unit: 변환 단위)는 하나의 TB(transform block: 변환 블록) 및 2 개의 크로마 TB를 갖는다. 인트라 예측된 CU에서의 PU는 단일의 TU(크기가 PU와 같음) 또는 다수의 TU를 포함할 수 있다. 본원에서 사용되는 바와 같이, "블록"이라는 용어는, 문맥에 따라, CB, PB, TB 또는 다른 샘플 값 세트를 나타낼 수 있다. 인코더는 비디오를 CTU, CU, PU, TU, 기타로 어떻게 분할할지를 결정한다.

[0044] 도 3으로 돌아가서, 인코더는 소스 프레임(331)의 인트라 코딩된 블록을 소스 프레임(331) 내의 다른 이전에 재구성된 샘플 값으로부터의 예측으로 표현한다. 인트라 BC 예측에 있어서, 인트라 픽처 추정기(intra-picture estimator)는 다른 이전에 재구성된 샘플 값에 대한 블록의 변위를 추정한다. 인트라 프레임 예측 참조 영역(intra-frame prediction reference region)(또는 간단히 인트라 예측 영역)은 블록에 대한 BC 예측 값을 발생

시키기 위해 사용되는 프레임 내의 샘플 영역이다. 인트라 프레임 예측 영역은 BV(block vector) 값(BV 추정에서 결정됨)으로 표시될 수 있다. 블록에 대한 인트라 공간 예측(intra spatial prediction)에 대해, 인트라 뷍처 추정기는 이웃하는 재구성된 샘플 값의 블록 내로의 외삽(extrapolation)을 추정한다. 인트라 뷍처 추정기는 엔트로피 코딩되어 있는, (인트라 BC 예측을 위한 BV 값 또는 인트라 공간 예측을 위한 예측 모드(방향)와 같은) 예측 정보를 출력할 수 있다. 인트라 프레임 예측 예측기(intra-frame prediction predictor)는 인트라 예측 값을 결정하기 위해 예측 정보를 적용한다.

[0045] 인코더(340)는 소스 프레임(331)의 인터 프레임 코딩된, 예측된 블록을 참조 프레임으로부터의 예측으로 표현한다. 움직임 추정기(motion estimator)는 하나 이상의 참조 프레임(369)에 대한 블록의 움직임을 추정한다. 다수의 참조 프레임이 사용될 때, 다수의 참조 프레임은 상이한 시간 방향 또는 동일한 시간 방향으로부터의 것일 수 있다. 움직임 보상된 예측 참조 영역(motion-compensated prediction reference region)은 현재 프레임의 샘플 블록에 대한 움직임 보상된 예측 값을 발생시키는 데 사용되는 참조 프레임(들) 내의 샘플 영역이다. 움직임 추정기는 엔트로피 코딩되어 있는, MV(motion vector: 움직임 벡터) 정보와 같은, 움직임 정보를 출력한다. 움직임 보상기(motion compensator)는 인터 프레임 예측을 위한 움직임 보상된 예측 값을 결정하기 위해 참조 프레임(369)에 MV를 적용한다.

[0046] 인코더는 블록의 예측 값(인트라 또는 인터)과 대응하는 원래 값 사이의 차이(있는 경우)를 결정할 수 있다. 이 예측 잔차 값(prediction residual value)은 주파수 변환, 양자화 및 엔트로피 인코딩을 사용하여 추가로 인코딩된다. 예를 들어, 인코더(340)는 비디오의 뷍처, 타일, 슬라이스 및/또는 다른 부분에 대해 QP(quantization parameter: 양자화 파라미터)에 대한 값을 설정하고, 그에 따라 변환 계수를 양자화한다. 인코더(340)의 엔트로피 코더(entropy coder)는 양자화된 변환 계수 값을 물론, 특정의 보조 정보(side information)(예컨대, MV 정보, BV 값, QP 값, 모드 결정, 파라미터 선택)를 압축한다. 전형적인 엔트로피 코딩 기법은 지수 골롬 코딩(Exponential-Golomb coding), 골롬 라이스 코딩(Golomb-Rice coding), 산술 코딩(arithmetic coding), 차분 코딩(differential coding), 허프만 코딩(Huffman coding), 런 길이 코딩(run length coding), V2V(variable-length-to-variable-length) 코딩, V2F(variable-length-to-fixed-length) 코딩, LZ(Lempel-Ziv) 코딩, 사전 코딩(dictionary coding), PIPE(probability interval partitioning entropy coding), 및 이들의 조합을 포함한다. 엔트로피 코더는 상이한 종류의 정보에 대해 상이한 코딩 기법을 사용할 수 있고, (예컨대, 골롬 라이스 코딩에 뒤이어서 산술 코딩을 적용하는 것에 의해) 다수의 기법을 결합하여 적용할 수 있으며, 특정의 코딩 기법 내에서 다수의 코드 테이블 중에서 선택할 수 있다.

[0047] 디코딩된 프레임(451)에서의 블록 경계 행 및/또는 열에 걸친 불연속을 평활화하기 위해 적응적 블록화 제거 필터(adaptive deblocking filter)가 인코더(340) 내의 움직임 보상 루프 내에 포함된다. (도시되지 않은, 링잉 제거 필터링(de-ringing filtering), ALF(adaptive loop filtering), 또는 SAO(sample-adaptive offset) 필터링과 같은) 다른 필터링이 대안적으로 또는 그에 부가하여 루프내 필터링 동작으로서 적용될 수 있다.

[0048] 코딩된 프레임(341) 및 MMCO/RPS 정보(342)(또는 프레임에 대한 의존 관계 및 정렬 구조를 인코더(340)에서 이미 알고 있기 때문에, MMCO/RPS 정보(342)와 동등한 정보)는 디코딩 프로세스 에뮬레이터(350)에 의해 처리된다. 디코딩 프로세스 에뮬레이터(350)는 디코더의 기능 중 일부(예를 들어, 참조 프레임을 재구성하는 디코딩 작업)를 구현한다. MMCO/RPS 정보(342)와 부합하는 방식으로, 디코딩 프로세스 에뮬레이터(350)는 주어진 코딩된 프레임(342)이 재구성되어 인코딩될 후속 프레임의 인터 프레임 예측에서 참조 프레임으로서 사용하기 위해 저장될 필요가 있는지를 결정한다. 코딩된 프레임(341)이 저장될 필요가 있는 경우, 디코딩 프로세스 에뮬레이터(350)는 코딩된 프레임(341)을 수신하고 대응하는 디코딩된 프레임(351)을 생성하는 디코더에 의해 수행될 디코딩 프로세스를 모델링한다. 그렇게 함에 있어서, 인코더(340)가 디코딩된 프레임 저장 구역(360)에 저장된 디코딩된 프레임(들)(369)을 사용할 때, 디코딩 프로세스 에뮬레이터(350)는 또한 디코딩 프로세스의 일부로서 저장 구역(360)으로부터의 디코딩된 프레임(들)(369)을 사용한다.

[0049] 디코딩된 프레임 임시 메모리 저장 구역(360)은 다수의 프레임 버퍼 저장 구역(361, 362, ..., 36n)을 포함한다. MMCO/RPS 정보(342)와 부합하는 방식으로, 디코딩 프로세스 에뮬레이터(350)는 참조 프레임으로서 사용하기 위해 인코더(340)에 의해 더 이상 필요로 하지 않는 프레임을 갖는 임의의 프레임 버퍼(361, 362 등)를 식별하기 위해 저장 구역(360)의 내용을 관리한다. 디코딩 프로세스를 모델링한 후에, 디코딩 프로세스 에뮬레이터(350)는 새로 디코딩된 프레임(351)을 이러한 방식으로 식별된 프레임 버퍼(361, 362 등)에 저장한다.

[0050] 코딩된 프레임(341) 및 MMCO/RPS 정보(342)는 임시 코딩된 데이터 구역(temporary coded data area)(370)에 버퍼링된다. 코딩된 데이터 구역(370)에 통합되어 있는 코딩된 데이터는, 기본 코딩된 비디오 비트스트림

(elementary coded video bitstream)의 구문의 일부로서, 하나 이상의 픽쳐에 대한 인코딩된 데이터를 포함한다. 코딩된 데이터 구역(370)에 통합되어 있는 코딩된 데이터는 또한 코딩된 비디오 데이터에 관한 미디어 메타데이터를 (예컨대, 하나 이상의 SEI(supplemental enhancement information) 메시지 또는 VUI(video usability information) 메시지에 하나 이상의 파라미터로서) 포함할 수 있다.

[0051] 임시 코딩된 데이터 구역(370)으로부터의 통합된 데이터(371)는 채널 인코더(channel encoder)(380)에 의해 처리된다. 채널 인코더(380)는 (예컨대, ITU-T H.222.0 | ISO/IEC 13818-1과 같은 미디어 프로그램 스트림 또는 전송 스트림 포맷 또는 IETF RFC 3550과 같은 인터넷 실시간 전송 프로토콜 포맷에 따라) 미디어 스트림으로서 전송 또는 저장하기 위해 통합된 데이터를 패킷화(packetize) 및/또는 다중화(multiplex)할 수 있고, 이 경우에 채널 인코더(380)는 미디어 전송 스트림의 구문의 일부로서 구문 요소를 추가할 수 있다. 또는, 채널 인코더(380)는 (예컨대, ISO/IEC 14496-12와 같은 미디어 컨테이너 포맷(media container format)에 따라) 파일로서 저장하기 위해 통합된 데이터를 구성(organize)할 수 있고, 이 경우에 채널 인코더(380)는 미디어 저장 파일의 구문의 일부로서 구문 요소를 추가할 수 있다. 또는, 보다 일반적으로, 채널 인코더(380)는 하나 이상의 미디어 시스템 다중화 프로토콜 또는 전송 프로토콜을 구현할 수 있고, 이 경우에 채널 인코더(380)는 프로토콜(들)의 구문의 일부로서 구문 요소를 추가할 수 있다. 채널 인코더(380)는 출력에 대한 저장소, 통신 연결, 또는 다른 채널을 나타내는 출력을 채널(390)에 제공한다. 채널 인코더(380) 또는 채널(390)은 또한 FEC(forward-error correction: 순방향 오류 정정) 인코딩 및 아날로그 신호 변조와 같은 다른 요소(도시되지 않음)를 포함할 수 있다.

#### IV. 예시적인 디코더 시스템

[0053] 도 4는 일부 기술된 실시예가 구현될 수 있는 예시적인 디코더 시스템(400)의 블록도이다. 디코더 시스템(400)은 실시간 통신을 위한 저-레이턴시 디코딩 모드 및 파일 또는 스트림으로부터 미디어 재생을 위한 고-레이턴시 디코딩 모드와 같은 다수의 디코딩 모드 중 임의의 것에서 동작할 수 있는 범용 디코딩 도구일 수 있거나, 하나의 이러한 디코딩 모드를 위해 구성된 특수 목적 디코딩 도구일 수 있다. 디코더 시스템(400)은 운영 체제 모듈로서, 애플리케이션 라이브러리의 일부로서 또는 독립형 애플리케이션으로서 구현될 수 있다. 일반적으로, 디코더 시스템(400)은 채널(410)로부터 코딩된 데이터를 수신하고 출력 목적지(490)에 대한 출력으로서 재구성된 프레임들을 생성한다. 코딩된 데이터는 인트라 BC 예측 모드를 사용하여 인코딩된 콘텐츠를 포함할 수 있다.

[0054] 디코더 시스템(400)은 입력으로서의 코딩된 데이터에 대한 저장소, 통신 연결, 또는 다른 채널을 나타낼 수 있는 채널(410)을 포함한다. 채널(410)은 채널 코딩되어 있는 코딩된 데이터를 생성한다. 채널 디코더(420)는 코딩된 데이터를 처리할 수 있다. 예를 들어, 채널 디코더(420)는 (예컨대, ITU-T H.222.0 | ISO/IEC 13818-1과 같은 미디어 프로그램 스트림 또는 전송 스트림 포맷 또는 IETF RFC 3550과 같은 인터넷 실시간 전송 프로토콜 포맷에 따라) 미디어 스트림으로서 전송 또는 저장하기 위해 통합된 데이터를 역패킷화(de-packetize) 및/또는 역다중화(demultiplex)하고, 이 경우에 채널 디코더(420)는 미디어 전송 스트림의 구문의 일부로서 추가된 구문 요소를 파싱할 수 있다. 또는, 채널 디코더(420)는 (예컨대, ISO/IEC 14496-12와 같은 미디어 컨테이너 포맷에 따라) 파일로서 저장하기 위해 통합되어 있는 코딩된 비디오 데이터를 분리시키고, 이 경우에 채널 디코더(420)는 미디어 저장 파일의 구문의 일부로서 추가된 구문 요소를 파싱할 수 있다. 또는, 보다 일반적으로, 채널 디코더(420)는 하나 이상의 미디어 시스템 역다중화 프로토콜 또는 전송 프로토콜을 구현할 수 있고, 이 경우에 채널 디코더(420)는 프로토콜(들)의 구문의 일부로서 추가된 구문 요소를 파싱할 수 있다. 채널(410) 또는 채널 디코더(420)는 또한, 예컨대, FEC 디코딩 및 아날로그 신호 복조를 위한 다른 요소(도시되지 않음)를 포함할 수 있다.

[0055] 채널 디코더(420)로부터 출력되는 코딩된 데이터(421)는, 충분한 양의 이러한 데이터가 수신될 때까지, 임시 코딩된 데이터 구역(430)에 저장된다. 코딩된 데이터(421)는 코딩된 프레임(431) 및 MMCO/RPS 정보(432)를 포함한다. 코딩된 데이터 구역(430) 내의 코딩된 데이터(421)는, 기본 코딩된 비디오 비트스트림의 구문의 일부로서, 하나 이상의 픽쳐에 대한 코딩된 데이터를 포함한다. 코딩된 데이터 구역(430) 내의 코딩된 데이터(421)는 또한 (예컨대, 하나 이상의 SEI 메시지 또는 VUI 메시지에서의 하나 이상의 파라미터로서) 인코딩된 비디오 데이터에 관련된 미디어 메타데이터를 포함할 수 있다.

[0056] 일반적으로, 코딩된 데이터 구역(430)은 코딩된 데이터(421)가 디코더(450)에 의해 사용될 때까지 이러한 코딩된 데이터(421)를 일시적으로 저장한다. 그 시점에서, 코딩된 프레임(431) 및 MMCO/RPS 정보(432)에 대한 코딩된 데이터가 코딩된 데이터 구역(430)으로부터 디코더(450)로 전송된다. 디코딩이 계속됨에 따라, 새로운 코딩

된 데이터가 코딩된 데이터 구역(430)에 추가되고, 코딩된 데이터 구역(430)에 남아 있는 가장 오래된 코딩된 데이터가 디코더(450)로 전송된다.

[0057] 디코더(450)는 코딩된 프레임(431)을 디코딩하여 대응하는 디코딩된 프레임(451)을 생성한다. 적절한 경우, 그의 디코딩 프로세스를 수행할 때, 디코더(450)는 하나 이상의 이전에 디코딩된 프레임(469)을 인터 프레임 예측을 위한 참조 프레임으로서 사용할 수 있다. 디코더(450)는 디코딩된 프레임 임시 메모리 저장 구역(460)으로부터 이러한 이전에 디코딩된 프레임(469)을 읽는다. 일반적으로, 디코더(450)는 엔트로피 디코딩, 인트라 프레임 예측, 움직임 보상된 인터 프레임 예측, 역양자화(inverse quantization), 역 주파수 변환(inverse frequency transform), 및 타일의 병합과 같은 디코딩 작업을 수행하는 다수의 디코딩 모듈을 포함한다. 디코더(450)에 의해 수행되는 정확한 동작은 압축 포맷에 따라 변할 수 있다.

[0058] 예를 들어, 디코더(450)는 압축된 프레임 또는 프레임 시퀀스에 대한 인코딩된 데이터를 수신하고, 디코딩된 프레임(451)을 포함하는 출력을 생성한다. 디코더(450)에서, 버퍼는 압축된 프레임에 대한 인코딩된 데이터를 수신하고, 적절한 때에, 수신된 인코딩된 데이터를 엔트로피 디코더에 이용 가능하게 만든다. 엔트로피 디코더는, 전형적으로 인코더에서 수행된 엔트로피 인코딩의 역을 적용하여, 엔트로피 코딩된 양자화된 데이터는 물론 엔트로피 코딩된 보조 정보를 엔트로피 디코딩한다. 움직임 보상기는 재구성되는 프레임의 인터 코딩된 블록의 움직임 보상된 예측 값을 형성하기 위해 움직임 정보를 하나 이상의 참조 프레임에 적용한다. 인트라 프레임 예측 모듈은 이웃하는 이전에 재구성된 샘플값으로부터 현재 블록의 샘플값을 공간적으로 예측할 수 있거나, 인트라 BC 예측에 대해서는, 프레임에서의 인트라 프레임 예측 영역의 이전에 재구성된 샘플 값을 사용해 현재 블록의 샘플 값을 예측할 수 있다. 인트라 프레임 예측 영역은 BV 값으로 표시될 수 있다. 디코더(450)는 또한 예측 잔차 값을 재구성한다. 역양자화기는 엔트로피 디코딩된 데이터를 역양자화한다. 예를 들어, 디코더(450)는 비디오의 픽처, 타일, 슬라이스 및/또는 다른 부분에 대해 QP에 대한 값을 비트스트림에서의 구문 요소에 기초하여 설정하고, 그에 따라 변환 계수를 역양자화한다. 역 주파수 변환기는 양자화된 주파수 영역 데이터를 공간 영역 데이터로 변환시킨다. 인터 프레임 예측된 블록에 대해, 디코더(450)는 재구성된 예측 잔차 값을 움직임 보상된 예측 값을 결합시킨다. 디코더(450)는 이와 유사하게 예측 잔차 값을 인트라 프레임 예측으로부터의 예측 값을 결합시킬 수 있다. 디코딩된 프레임(451)에서의 블록 경계 행 및/또는 열에 걸친 불연속을 평활화하기 위해 적응적 블록화 제거 필터가 비디오 디코더(450)에서의 움직임 보상 루프 내에 포함된다. (도시되지 않은, 링잉 제거 필터링, ALF, 또는 SAO 필터링과 같은) 다른 필터링이 대안적으로 또는 그에 부가하여 루프내 필터링 동작으로서 적용될 수 있다.

[0059] 디코딩된 프레임 임시 메모리 저장 구역(460)은 다수의 프레임 버퍼 저장 구역(461, 462, ..., 46n)을 포함한다. 디코딩된 프레임 저장 구역(460)은 디코딩된 픽처 버퍼(decoded picture buffer)의 일 예이다. 디코더(450)는 디코딩된 프레임(451)을 저장할 수 있는 프레임 버퍼(461, 462 등)를 식별하기 위해 MMCO/RPS 정보(432)를 사용한다. 디코더(450)는 디코딩된 프레임(451)을 그 프레임 버퍼에 저장한다.

[0060] 출력 시퀀서(output sequencer)(480)는 출력 순서에서 생성될 다음 프레임이 디코딩된 프레임 저장 구역(460)에서 이용 가능할 때를 식별한다. 출력 순서에서 생성될 다음 프레임(481)이 디코딩된 프레임 저장 구역(460)에서 이용 가능할 때, 그것이 출력 시퀀서(480)에 의해 읽혀지고 출력 목적지(490)(예컨대, 디스플레이)로 출력된다. 일반적으로, 프레임이 디코딩된 프레임 저장 구역(460)으로부터 출력 시퀀서(480)에 의해 출력되는 순서는 프레임이 디코더(450)에 의해 디코딩되는 순서와 상이할 수 있다.

## V. 예시적인 비디오 인코더

[0062] 도 5a 및 도 5b는 일부 기술된 실시예가 구현될 수 있는 일반화된 비디오 인코더(500)의 블록도이다. 인코더(500)는 현재 프레임(505)을 포함하는 비디오 픽처 시퀀스를 입력 비디오 신호(505)로서 수신하고, 코딩된 비디오 비트스트림(595)에서의 인코딩된 데이터를 출력으로서 생성한다.

[0063] 인코더(500)는 블록 기반(block-based)이고, 구현에 의존하는 블록 포맷을 사용한다. 블록이 상이한 스테이지에서(예컨대, 예측, 주파수 변환 및/또는 엔트로피 인코딩 스테이지에서) 추가로 세분화될 수 있다. 예를 들어, 픽처가 64x64 블록, 32x32 블록 또는 16x16 블록으로 나누어질 수 있고, 이들이 차례로 코딩 및 디코딩을 위해 보다 작은 샘플 값을 블록으로 나누어질 수 있다. HEVC 표준에 대한 인코딩의 구현에서, 인코더는 픽처를 CTU(CTB), CU(CB), PU(PB) 및 TU(TB)로 분할한다.

[0064] 인코더(500)는 인트라 픽처 코딩(intra-picture coding) 및/또는 인터 픽처 코딩(inter-picture coding)을 사용하여 픽처를 압축한다. 인코더(500)의 구성요소들 중 다수는 인트라 픽처 코딩 및 인터 픽처 코딩 둘 다를

위해 사용된다. 그 구성요소들에 의해 수행되는 정확한 동작은 압축되는 정보의 유형에 따라 달라질 수 있다.

[0065] 타일화 모듈(tiling module)(510)은, 선택적으로, 픽처를 동일한 크기 또는 상이한 크기의 다수의 타일로 분할한다. 예를 들어, 타일화 모듈(510)은 픽처를, 픽처 경계와 함께, 픽처 내의 타일의 수평 및 수직 경계를 정의하는 타일 행 및 타일 열을 따라 분할하고, 여기서 각각의 타일은 직사각형 영역이다.

[0066] 일반 인코딩 제어(520)는 입력 비디오 신호(505)에 대한 픽처는 물론 인코더(500)의 다양한 모듈로부터의 피드백(도시되지 않음)을 수신한다. 전체적으로, 일반 인코딩 제어(520)는, 인코딩 동안 코딩 파라미터를 설정하고 변경하기 위해, 제어 신호(도시되지 않음)를 다른 모듈(타일화 모듈(510), 변환기/스케일러/양자화기(530), 스케일러/역변환기(535), 인트라 픽처 추정기(540), 움직임 추정기(550) 및 인트라/인터 스위치(intra/inter switch) 등)에 제공한다. 상세하게는, 일반 인코딩 제어(520)는 인코딩 동안 인트라 BC 예측을 사용할지 여부 및 어떻게 사용할지를 결정할 수 있다. 일반 인코딩 제어(520)는 또한 인코딩 동안 중간 결과를 평가할 수 있다(예컨대, 레이트 왜곡 분석(rate-distortion analysis)을 수행함). 일반 인코딩 제어(520)는, 대응하는 디코더가 일관성 있는 결정을 할 수 있도록, 인코딩 동안 행해진 결정을 나타내는 일반 제어 데이터(522)를 생성한다. 일반 제어 데이터(522)는 헤더 포맷터/엔트로피 코더(header formatter/entropy coder)(590)에 제공된다.

[0067] 현재 픽처가 인터 픽처 예측을 사용하여 예측되는 경우, 움직임 추정기(550)는 하나 이상의 참조 픽처에 대한 입력 비디오 신호(505)의 현재 픽처의 샘플 값 블록의 움직임을 추정한다. 디코딩된 픽처 버퍼(570)는 참조 픽처로서 사용하기 위해 하나 이상의 재구성된 이전에 코딩된 픽처(reconstructed previously coded picture)를 버퍼링한다. 다수의 참조 픽처가 사용될 때, 다수의 참조 픽처는 상이한 시간 방향 또는 동일한 시간 방향으로 부터의 것일 수 있다. 움직임 추정기(550)는 MV 데이터와 같은 움직임 데이터(552), 병합 모드 인덱스 값(merge mode index value) 및 참조 픽처 선택 데이터를 보조 정보로서 생성한다. 움직임 데이터(552)는 헤더 포맷터/엔트로피 코더(590)에는 물론 움직임 보상기(555)에도 제공된다.

[0068] 움직임 보상기(555)는 디코딩된 픽처 버퍼(570)로부터의 재구성된 참조 픽처(들)에 MV를 적용한다. 움직임 보상기(555)는 현재 픽처에 대한 움직임 보상된 예측을 생성한다.

[0069] 인코더(500) 내의 별도의 경로에서, 인트라 픽처 추정기(540)는 입력 비디오 신호(505)의 현재 픽처의 샘플 값 블록에 대한 인트라 픽처 예측을 어떻게 수행할지를 결정한다. 현재 픽처는 전체 또는 일부가 인트라 픽처 코딩을 사용하여 코딩될 수 있다. 현재 픽처의 재구성(538)의 값을 사용하여, 인트라 공간 예측에 대해, 인트라 픽처 추정기(540)는 현재 픽처의 이웃하는 이전에 재구성된 샘플 값으로부터 현재 픽처의 현재 블록의 샘플 값을 어떻게 공간적으로 예측할지를 결정한다. 또는, BV 값을 사용하는 인트라 BC 예측에 있어서, 인트라 픽처 추정기(540)는 현재 픽처 내의 상이한 후보 영역에 대한 현재 블록의 샘플 값의 변위를 추정한다. 이하에서 기술되는 바와 같이, BV 예측에 대한 평균 접근법을 사용하여 BV 값이 예측될 수 있다. 인트라 BC 예측에 있어서, 인트라 예측 추정기(540)는 이하에서 기술되는 하나 이상의 제약조건을 사용하여 BV 선택 프로세스를 제약할 수 있고, BV 추정을 위해 현재 블록의 중첩 구역에서의 재구성된 샘플 값을 추정할 수 있다.

[0070] 인트라 픽처 추정기(540)는, 인트라 예측이 공간 예측 또는 인트라 BC 예측을 사용하는지를 나타내는 정보(예컨대, 인트라 블록마다의 플래그 값), (인트라 공간 예측에 대한) 예측 모드 방향, 및 (인트라 BC 예측에 대한) BV 값을 보조 정보로서 생성한다. 인트라 예측 데이터(542)는 헤더 포맷터/엔트로피 코더(590)에는 물론 인트라 픽처 예측기(545)에도 제공된다.

[0071] 인트라 예측 데이터(542)에 따라, 인트라 픽처 예측기(545)는 현재 픽처의 이웃하는 이전에 재구성된 샘플 값으로부터 현재 픽처의 현재 블록의 샘플 값을 공간적으로 예측한다. 또는 인트라 BC 예측에 있어서, 인트라 픽처 예측기(545)는 현재 블록에 대한 BV 값에 의해 표시되는, 인트라 예측 영역의 이전에 재구성된 샘플 값을 사용하여 현재 블록의 샘플 값을 예측한다. 픽처에 대한 크로마 데이터가 루마 데이터와 동일한 분해능을 가질 때(예컨대, 포맷이 YUV 4:4:4 포맷 또는 RGB 4:4:4 포맷일 때), 크로마 블록에 대해 적용되는 BV 값은 루마 블록에 대해 적용되는 BV 값과 동일할 수 있다. 다른 한편으로, 픽처에 대한 크로마 데이터가 루마 데이터에 비해 감소된 분해능을 가질 때(예컨대, 포맷이 YUV 4:2:0 포맷 또는 YUV 4:2:2 포맷일 때), (예컨대, YUV 4:2:0 포맷에 대해, BV 값의 수직 및 수평 성분을 2로 나누고 이를 정수 값으로 버림(truncate)하거나 반올림(round)하는 것에 의해; YUV 4:2:2 포맷에 대해, BV 값의 수평 성분을 2로 나누고 이를 정수 값으로 버림하거나 반올림하는 것에 의해) 크로마 분해능에서의 차이에 대해 조절하기 위해 크로마 블록에 대해 적용되는 BV 값이 스케일링 다운(scale down)되고 어쩌면 반올림될 수 있다.

[0072] 인트라/인터 스위치는 주어진 블록에 대한 예측(558)으로서 사용하기 위해 움직임 보상된 예측 또는 인트라 픽

처 예측의 값을 선택한다. 예측(558)의 블록과 입력 비디오 신호(505)의 원래의 현재 픽처의 대응하는 부분 사이의 차이(있는 경우)는 잔차(518)의 값을 제공한다. 현재 픽처의 재구성 동안, 재구성된 잔차 값은, 비디오 신호(505)로부터의 원래 콘텐츠의 재구성(538)을 생성하기 위해, 예측(558)과 결합된다. 그렇지만, 손실 압축에서, 일부 정보가 비디오 신호(505)로부터 여전히 손실된다.

[0073] 변환기/스케일러/양자화기(530)에서, 주파수 변환기는 공간 영역 비디오 데이터를 주파수 영역(즉, 스팩트럼, 변환) 데이터로 변환시킨다. 블록 기반 비디오 코딩에 있어서, 주파수 변환기는 예측 잔차 데이터(또는 예측(558)이 널(null)인 경우, 샘플 값 데이터)의 블록에 이산 코사인 변환(DCT), 그의 정수 근사화(integer approximation), 또는 다른 유형의 순방향 블록 변환(forward block transform)(예컨대, 이산 사인 변환(discrete sine transform) 또는 그의 정수 근사화)을 적용하여, 주파수 변환 계수 블록을 생성한다. 인코더(500)는 또한 이러한 변환 단계가 생략된다는 것을 나타낼 수 있다. 스케일러/양자화기는 변환 계수를 스케일링하고 양자화한다. 예를 들어, 양자화기는 프레임 단위로, 타일 단위로, 슬라이스 단위로, 블록 단위로, 주파수 특정(frequency-specific) 단위로 또는 다른 단위로 변하는 양자화 계단 크기(quantization step size)로 데드존 스칼라 양자화(dead-zone scalar quantization)를 주파수 영역 데이터에 적용한다. 양자화된 변환 계수 데이터(532)는 헤더 포맷터/엔트로피 코더(590)에 제공된다.

[0074] 스케일러/역변환기(535)에서, 스케일러/역양자화기는 양자화된 변환 계수에 대해 역스케일링 및 역양자화를 수행한다. 역 주파수 변환기는 역 주파수 변환을 수행하여, 재구성된 예측 잔차 값 블록 또는 재구성된 샘플 값 블록을 생성한다. 인코더(500)는 재구성된 잔차 값을 예측(558)의 값(예컨대, 움직임 보상된 예측 값, 인트라 픽처 예측 값)과 결합하여 재구성(538)을 형성한다.

[0075] 인트라 픽처 예측의 경우, 재구성(538)의 값은 인트라 픽처 추정기(540) 및 인트라 픽처 예측기(545)에 피드백 될 수 있다. 또한, 재구성(538)의 값이 후속 픽처의 움직임 보상된 예측을 위해 사용될 수 있다. 재구성(538)의 값이 추가로 필터링될 수 있다. 필터링 제어(560)는, 비디오 신호(505)의 주어진 픽처에 대해, 재구성(538)의 값에 대해 블록화 제거 필터링 및 SAO 필터링을 어떻게 수행할지를 결정한다. 필터링 제어(560)는 헤더 포맷터/엔트로피 코더(590) 및 병합기/필터(들)(565)에 제공되는 필터 제어 데이터(562)를 생성한다.

[0076] 병합기/필터(들)(565)에서, 인코더(500)는 상이한 타일로부터의 콘텐츠를 픽처의 재구성된 버전에 병합한다. 인코더(500)는, 프레임에서의 경계에 걸쳐 불연속을 적응적으로 평활화하기 위해, 필터 제어 데이터(562)에 따라 블록화 제거 필터링 및 SAO 필터링을 선택적으로 수행한다. (도시되어 있지 않은, 렇잉 제거 필터링 또는 ALF와 같은) 다른 필터링이 대안적으로 또는 그에 부가하여 적용될 수 있다. 인코더(500)의 설정에 따라, 타일 경계가 선택적으로 필터링되거나 전혀 필터링되지 않을 수 있고, 인코더(500)는 이러한 필터링이 적용되었는지 여부를 나타내는 구문을 코딩된 비트스트림 내에 제공할 수 있다. 디코딩된 픽처 버퍼(570)는 후속하는 움직임 보상된 예측에서 사용하기 위해 재구성된 현재 픽처를 버퍼링한다.

[0077] 헤더 포맷터/엔트로피 코더(590)는 일반 제어 데이터(522), 양자화된 변환 계수 데이터(532), 인트라 예측 데이터(542), 움직임 데이터(552) 및 필터 제어 데이터(562)를 포맷 지정하고 그리고/또는 엔트로피 코딩한다. 예를 들어, 헤더 포맷터/엔트로피 코더(590)는 BV 값에 대한 구문 요소와 같은 다양한 구문 요소의 엔트로피 코딩을 위해 컨텍스트 적응적 이진 산술 코딩(context-adaptive binary arithmetic coding)을 사용한다. 헤더 포맷터/엔트로피 코더(590)는 BV 값을 인코딩할 때, 이하에서 기술하는 바와 같이, BV 예측에 대한 평평 접근법을 사용할 수 있다.

[0078] 헤더 포맷터/엔트로피 코더(590)는 인코딩된 데이터를 코딩된 비디오 비트스트림(595)으로 제공한다. 코딩된 비디오 비트스트림(595)의 포맷은 HEVC 포맷, WMV(Windows Media Video) 포맷, VC-1 포맷, MPEG-x 포맷(예컨대, MPEG-1, MPEG-2, 또는 MPEG-4), H.26x 포맷(예컨대, H.261, H.262, H.263, H.264), 또는 다른 포맷의 변형 또는 확장일 수 있다.

[0079] 구현 및 원하는 압축 유형에 따라, 인코더의 모듈이 추가되고, 생략되며, 다수의 모듈로 분할되고, 다른 모듈과 결합되며, 그리고/또는 유사한 모듈로 대체될 수 있다. 대안의 실시예에서, 상이한 모듈 및/또는 다른 구성의 모듈을 갖는 인코더가 기술되는 기법들 중 하나 이상을 수행한다. 인코더의 구체적인 실시예는 전형적으로 인코더(500)의 변형 또는 보완된 버전을 사용한다. 인코더(500) 내의 모듈들 간의 도시된 관계는 인코더에서의 정보의 일반적인 흐름을 나타내고; 간단함을 위해, 다른 관계는 도시되어 있지 않다.

## VI. 예시적인 비디오 디코더

[0080] 도 6은 일부 기술된 실시예가 구현될 수 있는 일반화된 디코더(600)의 블록도이다. 디코더(600)는 인코딩된 데

이터를 코딩된 비디오 비트스트림(605)으로 수신하고, 재구성된 비디오(695)에 대한 픽처를 포함하는 출력을 생성한다. 코딩된 비디오 비트스트림(605)의 포맷은 HEVC 포맷, WMV(Windows Media Video) 포맷, VC-1 포맷, MPEG-x 포맷(예컨대, MPEG-1, MPEG-2, 또는 MPEG-4), H.26x 포맷(예컨대, H.261, H.262, H.263, H.264), 또는 다른 포맷의 변형 또는 확장일 수 있다.

[0082] 디코더(600)는 블록 기반이고, 구현에 의존하는 블록 포맷을 사용한다. 블록이 상이한 스테이지에서 추가로 세분될 수 있다. 예를 들어, 픽처가 64x64 블록, 32x32 블록 또는 16x16 블록으로 나누어질 수 있고, 이들이 차례로 보다 작은 샘플 값 블록으로 나누어질 수 있다. HEVC 표준에 대한 디코딩의 구현에서, 픽처가 CTU(CTB), CU(CB), PU(PB) 및 TU(TB)로 분할된다.

[0083] 디코더(600)는 인트라 픽처 디코딩(intra-picture decoding) 및/또는 인터 픽처 디코딩(inter-picture decoding)을 사용하여 픽처를 압축 해제한다. 디코더(600)의 구성요소들 중 다수는 인트라 픽처 디코딩 및 인터 픽처 디코딩 둘 다를 위해 사용된다. 그 구성요소들에 의해 수행되는 정확한 동작은 압축 해제되는 정보의 유형에 따라 달라질 수 있다.

[0084] 베퍼는 인코딩된 데이터를 코딩된 비디오 비트스트림(605)으로 수신하고 수신된 인코딩된 데이터를 파서/엔트로피 디코더(610)에 이용 가능하게 만든다. 파서/엔트로피 디코더(610)는, 전형적으로 인코더(500)에서 수행된 엔트로피 코딩의 역(예컨대, 컨텍스트 적응적 이진 산술 디코딩)을 적용하여, 엔트로피 코딩된 데이터를 엔트로피 디코딩한다. 파싱 및 엔트로피 디코딩의 결과로서, 파서/엔트로피 디코더(610)는 일반 제어 데이터(622), 양자화된 변환 계수 데이터(632), 인트라 예측 데이터(642), 움직임 데이터(652) 및 필터 제어 데이터(662)를 생성한다. 상세하게는, 인트라 예측 데이터(642)에 대해, 파서/엔트로피 디코더(610)는, 예컨대, 컨텍스트 적응적 이진 산술 디코딩을 사용해, BV 값에 대한 구문 요소를 엔트로피 디코딩한다. 파서/엔트로피 디코더(610)는 BV 값을 디코딩할 때, 이하에서 기술하는 바와 같이, BV 예측에 대한 평평 접근법을 사용할 수 있다.

[0085] 일반 디코딩 제어(620)는, 디코딩 동안 디코딩 파라미터를 설정하고 변경하기 위해, 일반 제어 데이터(622)를 수신하고 제어 신호를 다른 모듈(도시되지 않음)(스케일러/역변환기(635), 인트라 픽처 예측기(645), 움직임 보상기(655) 및 인트라/인터 스위치 등)에 제공한다.

[0086] 현재 픽처가 인터 픽처 예측을 사용하여 예측되는 경우, 움직임 보상기(655)는 MV 데이터와 같은 움직임 데이터(652), 참조 픽처 선택 데이터 및 병합 모드 인덱스 값을 수신한다. 움직임 보상기(655)는 디코딩된 픽처 베퍼(670)로부터의 재구성된 참조 픽처(들)에 MV를 적용한다. 움직임 보상기(655)는 현재 픽처의 인터 코딩된 블록에 대한 움직임 보상된 예측을 생성한다. 디코딩된 픽처 베퍼(670)는 참조 픽처로서 사용하기 위해 하나 이상의 이전에 재구성된 픽처를 저장한다.

[0087] 디코더(600) 내의 별도의 경로에서, 인트라 프레임 예측 예측기(645)는, 인트라 예측이 공간 예측 또는 인트라 BC 예측을 사용하는지를 나타내는 정보(예컨대, 인트라 블록마다의 플래그 값), (인트라 공간 예측에 대한) 예측 모드 방향, 및 (인트라 BC 예측에 대한) BV 값과 같은, 인트라 예측 데이터(642)를 수신한다. 인트라 공간 예측의 경우, 현재 픽처의 재구성(638)의 값을 사용하여, 예측 모드 데이터에 따라, 인트라 픽처 예측기(645)는 현재 픽처의 이웃하는 이전에 재구성된 샘플 값을 사용하여, 예측 모드 데이터에 따라, 인트라 픽처 예측기(645)는 현재 블록의 샘플 값을 공간적으로 예측한다. 또는, BV 값을 사용하는 인트라 BC 예측에 있어서, 인트라 픽처 예측기(645)는 현재 블록에 대한 BV 값에 의해 표시되는, 인트라 예측 영역의 이전에 재구성된 샘플 값을 사용하여 현재 블록의 샘플 값을 예측한다.

[0088] 인트라/인터 스위치는 주어진 블록에 대한 예측(658)으로서 사용하기 위해 움직임 보상된 예측 또는 인트라 픽처 예측의 값을 선택한다. 예를 들어, HEVC 구문을 따를 때, 인트라/인터 스위치는 인트라 예측된 CU 및 인터 예측된 CU를 포함할 수 있는 픽처의 CU에 대해 인코딩된 구문 요소에 기초하여 제어될 수 있다. 디코더(600)는 비디오 신호로부터의 콘텐츠의 재구성(638)을 생성하기 위해 예측(658)을 재구성된 잔차 값을 결합시킨다.

[0089] 잔차를 재구성하기 위해, 스케일러/역변환기(635)는 양자화된 변환 계수 데이터(632)를 수신하고 처리한다. 스케일러/역변환기(635)에서, 스케일러/역양자화기는 양자화된 변환 계수에 대해 역스케일링 및 역양자화를 수행한다. 역 주파수 변환기는 역 주파수 변환을 수행하여, 재구성된 예측 잔차 값 블록 또는 재구성된 샘플 값을 블록을 생성한다. 예를 들어, 역 주파수 변환기는 주파수 변환 계수에 역 블록 변환을 적용하여, 샘플 값을 데이터 또는 예측 잔차 데이터를 생성한다. 역 주파수 변환은 역 DCT, 그의 정수 근사화, 또는 다른 유형의 역 주파수 변환(예컨대, 역 이산 사인 변환 또는 그의 정수 근사화)일 수 있다.

[0090] 인트라 픽처 예측의 경우, 재구성(638)의 값이 인트라 픽처 예측기(645)에 피드백될 수 있다. 인터 픽처 예측의 경우, 재구성(638)의 값이 추가로 필터링될 수 있다. 병합기/필터(들)(665)에서, 디코더(600)는 상이한 타

일로부터의 콘텐츠를 픽처의 재구성된 버전으로 병합한다. 디코더(600)는, 프레임에서의 경계에 걸쳐 불연속을 적응적으로 평활화하기 위해, 필터 제어 데이터(662) 및 필터 적용에 대한 규칙에 따라 블록화 제거 필터링 및 SAO 필터링을 선택적으로 수행한다. (도시되지 않은, 링잉 제거 필터링 또는 ALF와 같은) 다른 필터링이 대안적으로 또는 그에 부가하여 적용될 수 있다. 디코더(600)의 설정 또는 인코딩된 비트스트림 내의 구문 표시에 따라, 타일 경계가 선택적으로 필터링되거나 전혀 필터링되지 않을 수 있다. 디코딩된 픽처 버퍼(670)는 후속하는 움직임 보상된 예측에서 사용하기 위해 재구성된 현재 픽처를 버퍼링한다.

[0091] 디코더(600)는 또한 후처리 필터(post-processing filter)를 포함할 수 있다. 후처리 필터(608)는 링잉 제거 필터링, 적응 Wiener 필터링, 필름 그레이 재현 필터링(film-grain reproduction filtering), SAO 필터링 또는 다른 종류의 필터링을 포함할 수 있다.

[0092] 구현 및 원하는 압축 해제 유형에 따라, 디코더의 모듈이 추가되고, 생략되며, 다수의 모듈로 분할되고, 다른 모듈과 결합되며, 그리고/또는 유사한 모듈로 대체될 수 있다. 대안의 실시예에서, 상이한 모듈 및/또는 다른 구성의 모듈을 갖는 디코더는 기술된 기법들 중 하나 이상을 수행한다. 디코더의 구체적인 실시예는 전형적으로 디코더(600)의 변형 또는 보완된 버전을 사용한다. 디코더(600) 내의 모듈들 간의 도시된 관계는 디코더에서의 정보의 일반적인 흐름을 나타내고; 간단함을 위해, 다른 관계는 도시되어 있지 않다.

## VII. 인트라 블록 복사 예측 모드

[0094] 이 섹션은 인트라 BC(block copy) 예측 모드의 다양한 특징을 제시한다. 특징들 중 일부 특징은 BV(block vector) 값의 선택에 관한 것인 반면, 다른 특징은 BV 값의 인코딩/디코딩에 관한 것이다. 이 특징들은 인코딩 및 디코딩의 레이트 왜곡 성능 및/또는 계산 효율의 면에서 보다 효과적인 인트라 BC 예측을 용이하게 할 수 있다. 상세하게는, 인트라 BC 예측은 화면 포착 콘텐츠(screen capture content)와 같은 특정의 "인위적으로" 생성된 비디오 콘텐츠를 인코딩할 때 레이트 왜곡 성능을 향상시킬 수 있다. 화면 포착 콘텐츠는 전형적으로 반복된 구조(예컨대, 그래픽, 텍스트 문자)를 포함하며, 이는 인트라 BC 예측이 성능을 향상시킬 기회를 제공한다. 화면 포착 콘텐츠는 보통 높은 크로마 샘플링 분해능을 갖는 포맷(예컨대, YUV 4:4:4 또는 RGB 4:4:4)으로 인코딩되지만, 보다 낮은 크로마 샘플링 분해능을 갖는 포맷(예컨대, YUV 4:2:0, YUV 4:2:2)으로도 인코딩될 수 있다.

### A. 인트라 BC 예측 모드 - 서문.

[0096] 인트라 BC 예측 모드에 있어서, 픽처의 현재 블록의 샘플 값이 동일한 픽처에서의 샘플 값을 사용하여 예측된다. BV는 현재 블록부터 예측을 위해 사용된 샘플 값을 포함하는 픽처의 영역까지의 변위를 나타낸다. 전형적으로, 예측을 위해 사용되는 샘플 값은 이전에 재구성된 샘플 값이다. BV가 비트스트림에서 신호된다. 인트라 BC 예측은 인트라 픽처 예측의 일 형태이다 - 픽처의 블록에 대한 인트라 BC 예측은 동일한 픽처 내의 샘플 값 이외의 어떤 샘플 값도 사용하지 않는다 -.

[0097] 도 7은 현재 프레임(710)의 현재 블록(730)에 대한 인트라 BC 예측을 나타낸 것이다. 현재 블록은 CU(coding unit)의 CB(coding block: 코딩 블록), PU(prediction unit)의 PB(prediction block), TU(transform unit)의 TB(transform block), 또는 다른 블록일 수 있다. 현재 블록의 크기는 64x64, 32x32, 16x16, 8x8 또는 어떤 다른 크기일 수 있다. 보다 일반적으로, 현재 블록의 크기는  $m \times n$ 이고, 여기서  $m$  및  $n$  각각은 정수이며, 여기서  $m$  및  $n$ 은 서로 같을 수 있거나 상이한 값을 가질 수 있다. 대안적으로, 현재 블록은 어떤 다른 형상(예컨대, 비직사각형 형상을 갖는 코딩된 비디오 객체의 구역)을 가질 수 있다.

[0098] BV(740)는 현재 블록(730)부터 예측을 위해 사용된 샘플 값을 포함하는 픽처의 영역(750)까지의 변위(또는 오프셋)를 나타낸다. 현재 블록의 좌측 상단 위치가 현재 프레임에서 위치( $x_0, y_0$ )에 있는 것으로 가정하고, 인트라 예측 영역의 좌측 상단 위치가 현재 프레임에서 위치( $x_1, y_1$ )에 있는 것으로 가정한다. BV는 변위( $x_1 - x_0, y_1 - y_0$ )를 나타낸다. 예를 들어, 현재 블록의 좌측 상단 위치가 위치(320, 256)에 있고, 인트라 예측 영역의 좌측 상단 위치가 위치(295, 270)에 있는 경우, BV 값은 (-25, 14)이다. 이 예에서, 마이너스 수평 변위는 현재 블록의 좌측에 있는 위치를 나타내고, 마이너스 수직 변위는 현재 블록의 위쪽에 있는 위치를 나타낸다.

[0099] 일부 예시적인 구현에서, 인트라 예측된 영역(750)은 현재 블록(730)과 동일한 슬라이스 및 타일 내에 있도록 제약된다. 이러한 인트라 BC 예측은 다른 슬라이스 또는 타일에서의 샘플 값을 사용하지 않는다. 인트라 예측된 영역(750)의 위치가 (예컨대, 검색 범위에 대한, 인터 코딩된 블록의 재구성된 샘플 값의 사용에 관한) 하나 이상의 다른 제약조건을 받을 수 있다.

- [0100] 인트라 BC 예측의 예측 모드를 갖는 블록은 CB, PB 또는 다른 블록일 수 있다. 블록이 CB이 일 때, 블록에 대한 BV는 CU 레벨에서 신호될 수 있다(그리고 CU 내의 다른 CB는 동일한 BV 또는 그의 스케일링된 범위를 사용한다). 또는, 블록이 PB이 일 때, 블록에 대한 BV는 PU 레벨에서 신호될 수 있다(그리고 PU 내의 다른 PB는 동일한 BV 또는 그의 스케일링된 범위를 사용한다). 보다 일반적으로, 인트라 BC 예측 블록에 대한 BV는 블록에 대한 적절한 구문 레벨에서 신호된다.
- [0101] 인트라 BC 예측 모드에 따른 예측의 블록 복사 동작은 CB(BV가 CB마다 신호될 때) 또는 PB(BV가 PB마다 신호될 때)의 레벨에서 수행될 수 있다. 예를 들어, 16x16 CB가 단일의 16x16 PB를 갖는 것으로 가정한다. 16x16 영역을 블록 복사하기 위해 (PB에 대한) BV가 적용된다. 인트라 예측 영역이 예측되는 16x16 블록과 중첩하지 않도록 제약될 때, BV는 수평으로 또는 수직으로 적어도 16의 크기(절댓값)를 갖는다.
- [0102] 대안적으로, PB 또는 CB에 대한 BV가 신호될 때에도, PB 또는 CB 내에서의 TB의 레벨에서 블록 복사 동작이 수행될 수 있다. 이러한 방식으로, TB에 대해 적용되는 BV가 동일한 PB 또는 CB에서의 다른 TB의 위치를 참조할 수 있다. 예를 들어, 16x16 CB가 단일의 16x16 PB를 갖지만 잔차 코딩/디코딩을 위해 16 개의 4x4 TB로 분할되는 것으로 가정한다. 제1 TB에 대한 4x4 영역을 라스터 스캔 순서로 블록 복사하기 위해 (PB에 대한) BV가 적용되고, 이어서 제2 TB에 대한 4x4 영역을 라스터 스캔 순서로 블록 복사하기 위해 동일한 BV가 적용되며, 이하 마찬가지이다. TB에 대한 BC 동작에서 사용되는 4x4 영역은, 잔차 값을 동일한 CB 내의 이전에 재구성된 TB에 대한 예측된 값과 결합시킨 후에, 그 이전에 재구성된 TB에서의 위치를 포함할 수 있다. (BV는 예측되고 있는 동일한 TB에서의 위치를 여전히 참조하지 않는다.) TB 레벨에서 BC 동작을 적용하는 것은 비교적 작은 크기를 갖는 BV의 사용을 용이하게 한다.
- [0103] CU의 크로마 블록에 대한 인트라 BC 예측 동작은 일반적으로 CU의 루마 블록에 대한 인트라 BC 예측 동작에 대응한다. 보통, 크로마 PB 및 크로마 TB의 세그먼트화는 CU에서의 루마 PB 및 루마 TB의 세그먼트화에 바로 대응한다. 비디오의 포맷이 YUV 4:4:4일 때, 크로마 PB 및 TB의 크기는 대응하는 루마 PB 및 TB의 크기와 일치한다. 비디오의 포맷이 YUV 4:2:0일 때, 크로마 PB 및 TB는 폭과 높이가 대응하는 루마 PB 및 TB의 1/2이다. 그렇지만, 루마 TB가 최소 변환 크기를 갖는 경우, 그 최소 변환 크기를 갖는 단일의 크로마 TB가 사용된다. 비디오의 포맷이 YUV 4:2:2일 때, 크로마 PB 및 TB는 폭이 대응하는 루마 PB 및 TB의 1/2이다.
- [0104] 일부 구현에서, 인트라 BC 예측된 CU에 대해, 크로마 데이터가 루마 데이터에 비해 감소된 분해능을 가질 때(예컨대, 포맷이 YUV 4:2:0 포맷 또는 YUV 4:2:2 포맷일 때) 어쩌면 스케일링 및 반올림 후에, PU에서의 크로마 블록에 대한 인트라 BC 예측은 PU에서의 루마 블록에 대한 인트라 BC 예측과 동일한 BV 값을 사용한다. 대안적으로, PU의 루마 블록 및 크로마 블록에 대해 상이한 BV 값이 신호될 수 있다.
- [0105] 일부 구현에서, PU의 루마 블록의 예측 모드가 인트라 BC 예측인 경우, PU의 크로마 블록에 대한 예측 모드도 인트라 BC 예측된다. 예를 들어, PU에 대한 예측 모드가 신호된다. 대안적으로, 예측 모드가 PU의 루마 블록 또는 크로마 블록에 대해 인트라 BC 예측일 수 있지만, 둘 다에 대해 그렇지는 않다.
- [0106] **B. 인트라 BC 예측 모드에 대한 BV 검색을 제약하기**
- [0107] 일부 예시적인 구현에서, 인코더는 하나 이상의 제약조건에 따라 BV 범위를 제한한다. BV 범위를 제한하는 것에 의해, 인코딩 및 디코딩 동안 인트라 BC 예측을 위한 고속 메모리 액세스에 의해 참조되는 재구성된 샘플 값의 구역이 감소될 수 있고, 이는 구현 비용을 줄이는 경향이 있다.
- [0108] 도 8은 BV 값에 대한 검색 범위에 관한 예시적인 제약조건을 나타낸 것이다. 현재 프레임(810)의 현재 블록(830)에 부가하여, 도 8은 2 개의 CTB에 의해 정의되는 검색 범위(820, 822)를 나타내고 있다. 현재 CTB(820)는 현재 CTU의 일부이고, 현재 블록(830)을 포함한다. CTB(822)가 그의 왼쪽에 있는 경우, 현재 CTB(820)는 현재 블록(830)에 대한 허용 가능 BV가 발견될 수 있는 검색 범위를 정의한다. BV(842, 844)는 검색 범위 밖에 있는 영역을 참조하고, 따라서 그 BV 값(842, 844)은 허용되지 않는다.
- [0109] 일부 예시적인 구현에서, 현재 블록의 BV 값에 대한 검색 범위는 현재 CTB와 그의 왼쪽에 있는 CTB이다. 예를 들어, CTB는 64x64, 32x32 또는 16x16 샘플 값의 크기를 가질 수 있고, 이는 128x64, 64x32 또는 32x16 샘플 값의 검색 범위를 산출한다. 현재 CTB 및 좌측 CTB에서의 샘플 값만이 현재 블록에 대한 인트라 BC 예측을 위해 사용된다. 이것은 검색 프로세스를 제약함으로써 인코더 구현을 단순화시킨다. 이는 또한 디코더가 인트라 예측을 위해 고속 메모리에 버퍼링하는 샘플 값의 개수를 제한함으로써 디코더 구현을 단순화시킨다. 다른 제약조건은 인트라 예측이 다른 슬라이스 또는 타일로부터의 샘플 값을 참조할 수 없다는 것이다. 좌측 상단 위치가  $(x_0, y_0)$ 에 있는 현재  $m \times n$  블록 및 각각이  $CTB_{sizeY} \times CTB_{sizeY}$  차원을 가지는 CTB(들)에 대해, 인코더는 다

음과 같이 수평 성분  $BV[0]$  및 수직 성분  $BV[1]$ 을 가지는 2차원 BV에 대해 이 제약조건을 검사할 수 있다.

[0110]  $\bullet \quad BV[0] \geq -((x_0 \% CTB_{sizeY}) + CTB_{sizeY})$

[0111]  $\bullet \quad BV[1] \geq -(y_0 \% CTB_{sizeY})$

[0112]  $\bullet \quad$  위치  $(x_0, y_0)$ ,  $(x_0+BV[0], y_0+BV[1])$  및  $(x_0+BV[0]+m-1, y_0+BV[1]+n-1)$ 에 있는 샘플 값은 동일한 슬라이스에 있어야 한다.

[0113]  $\bullet \quad$  위치  $(x_0, y_0)$ ,  $(x_0+BV[0], y_0+BV[1])$  및  $(x_0+BV[0]+m-1, y_0+BV[1]+n-1)$ 에 있는 샘플 값은 동일한 타일에 있어야 한다.

[0114] 도 9는 BV 값의 선택에 관한 하나 이상의 제약조건 하에서, 인트라 BC 예측 모드를 사용해 인코딩하는 기법(900)을 나타낸 것이다. 도 3 또는 도 5a 및 도 5b를 참조하여 기술된 것과 같은 인코더는 기법(900)을 수행할 수 있다.

[0115] 우선, 인코더는 픽처의 현재 블록에 대한 BV를 결정한다(910). 현재 블록은 CB, PB 또는 다른 블록일 수 있다. BV는 픽처 내의 영역까지의 범위를 나타낸다. BV를 결정할 시에, 인코더는 하나 이상의 제약조건을 검사한다.

[0116] 하나의 가능한 제약조건에 따라, 인코더는 인트라 BC 예측을 위해 사용되는 샘플 값의 범위를 검사한다. 인코더는 후보 인트라 예측 영역이 현재 CTB 및 하나 이상의 다른 CTB(예컨대, 현재 CTB의 왼쪽에 있는 CTB)에 의해 정의된 범위 내에 있는지를 검사할 수 있다. 예를 들어, BV가 제1 성분  $BV[0]$  및 제2 성분  $BV[1]$ 을 갖고, 현재 블록이 좌측 상단 위치를 위치  $(x_0, y_0)$ 에 가지며, CTB(들) 각각이 폭  $CTB_{width}$  및 높이  $CTB_{height}$ 를 가질 때,  $BV[0] \geq -((x_0 \% CTB_{width}) + CTB_{width})$ 이고  $BV[1] \geq -(y_0 \% CTB_{height})$ 이면 제약조건이 충족된다. 인코더도 이와 유사하게 검색 범위 내에서의  $BV[0]$  및  $BV[1]$ 의 값에 대한 상한(upper limit)을 검사할 수 있다:  $BV[0] < (CTB_{width} - m - (x_0 \% CTB_{width}))$ 이고  $BV[1] < (CTB_{height} - n - (y_0 \% CTB_{height}))$ 이다. 대안적으로, 검색 범위는 보다 많은 또는 보다 적은 수의 CTB를 포함하거나, 검색 범위가 어떤 다른 방식으로 정의된다.

[0117] 다른 가능한 제약조건에 따라, 인코더는 검색을 현재 슬라이스 및 타일로 제한한다(즉, 현재 블록 및 영역은 픽처의 1 개 이하의 슬라이스 및 픽처의 1 개 이하의 타일의 일부이다). 인코더는 현재 블록의 좌측 상단 위치, 후보 인트라 예측 영역의 좌측 상단 위치 및 후보 인트라 예측 영역의 우측 하단 위치가 단일의 슬라이스 및 단일의 타일의 일부인지를 검사할 수 있다. 예를 들어,  $(x_0, y_0)$ ,  $(x_0 + BV[0], y_0 + BV[1])$  및  $(x_0 + BV[0] + m - 1, y_0 + BV[1] + n - 1)$ 이 단일의 슬라이스 및 단일의 타일의 일부이면 제약조건이 충족된다.

[0118] 대안적으로, 인코더는 다른 및/또는 부가의 제약조건을 검사한다.

[0119] 인코더는 BV를 사용하여 현재 블록에 대한 인트라 BC 예측을 수행한다(920). 예를 들어, 인코더는 현재 블록 전체에 대해 인트라 BC 예측을 수행한다. 또는, 인코더는 현재 블록과 연관된 다수의 블록에 대해(예컨대, 다수의 TB에 대해 TB 단위로 - 여기서 TB는 BV를 갖는 현재 PB와 연관되어 있음-) 인트라 BC 예측을 수행한다.

[0120] 인코더는 BV를 인코딩한다(930). 예를 들어, 인코더는 이하에서 기술되는 바와 같이 BV를 인코딩한다(930). 인코더는 다른 인트라 BC 예측 모드 블록에 대해 기법(900)을 반복할 수 있다.

[0121] 인트라 BC 예측을 위해, 인코더 및 디코더는 재구성된 샘플 값을 사용한다. 재구성되지 않은 샘플 값을 아직 인코딩되어 재구성되지 않은 픽처의 일부로서 존재할 수 있다. 인트라 BC 예측을 위해 재구성되지 않은 샘플 값을 사용하는 것을 피하기 위해, 인코더는, BV에 따른 인트라 BC 예측을 위해 실제의 이전에 재구성된 샘플 값을 만이 사용되도록, BV의 허용 가능한 값에 대한 제약조건을 설정할 수 있다.

[0122] 일부 예시적인 구현에서, 인코더는 현재 블록 및 후보 인트라 예측 영역의 우측 하단 위치를 포함하는 블록의 z-스캔 순서를 고려함으로써 BV 값을 검사한다. 보다 구체적으로는, 인코더는 위치  $(x_0 + BV[0] + m - 1, y_0 + BV[1] + n - 1)$ 을 포함하는 블록의 z-스캔 순서가  $(x_0, y_0)$ 을 포함하는 블록의 z-스캔 순서보다 작은지를 검사한다. 그러한 경우, 인트라 예측 영역의 우측 하단 위치를 포함하는 블록은 이전에 재구성되었다(그리고 따라서 나머지 인트라 예측 영역을 갖는다). BV는 또한 인트라 예측 영역이 현재 블록과 중첩하지 않도록 보장하는 조건  $BV[0] + m \leq 0$  및  $BV[1] + n \leq 0$  중 적어도 하나를 충족시킨다.

[0123] z-스캔 순서는 꾹처를 분할하는 블록들의 순차적으로 명시된 순서를 따른다. 도 10은 현재 블록(1030) 및 후보 BV에 대한 인트라 예측 영역의 우측 하단 위치를 포함할 수 있는 블록에 대한 예시적인 z-스캔 순서(1000)를 나타낸 것이다. 현재 블록(1030)은 CB, PB 또는 다른 블록일 수 있다. z-스캔 순서는 일반적으로, 상단부터 하단까지 연속적인 행에서 반복하여, 행에서 좌에서 우로 순차적으로 블록에 할당된다. 블록이 분할될 때, z-스캔 순서는 분할된 블록 내에서 재귀적으로 할당된다. HEVC 표준에 대한 인코딩/디코딩의 구현의 경우에, z-스캔 순서는 CTB 래스터 스캔 패턴에 의해(상단부터 하단까지 연속적인 CTB 행에서 반복하여, CTB 행에서 좌에서 우로) CTB에서 CTB로 진행한다. CTB가 분할되는 경우, z-스캔 순서는 분할된 CTB 내에서 쿼드트리(quadtreen)의 CB에 대한 래스터 스캔 패턴을 따른다. 그리고, CB가 (예컨대, 다수의 CB로 또는 다수의 PB로) 분할되는 경우, z-스캔 순서는 분할된 CB 내에서 블록에 대한 래스터 스캔 패턴을 따른다.

[0124] 대안적으로, 인트라 BC 예측이 TB 단위로 수행될 수 있을 때, 인코더 및 디코더는 인트라 예측 영역과 현재 블록(TB) 사이의 가능한 중첩이 있는지를 검사할 수 있고, 이어서 현재 TB가 인트라 BC 예측 동작의 적용을 위해 보다 작은 TB로 분할되어야만 하는지를 결정하기 위해 검사의 결과를 사용할 수 있다. 현재 TB가  $m \times n$ 의 크기 - 여기서  $m$  및  $n$ 은 서로 같을 수 있거나 상이한 값을 가질 수 있음 - 를 갖는 것으로 가정한다.  $BV[0] > -m$ 이고  $BV[1] > -n$ 이면, 인트라 예측 영역은 현재  $m \times n$  TB와 중첩하고, 이는 현재  $m \times n$  TB가 인트라 BC 예측 동작의 적용을 위해 보다 작은 TB로 분할되지 않는 한 문제가 된다. 이와 같이,  $BV[0] > -m$ 이고  $BV[1] > -n$ 이면, 인코더 및 디코더는 현재 TB를 보다 작은 TB로 분할한다. 분할 이후에  $m$  및  $n$ 의 보다 작은 값에 대해서도  $BV[0] > -m$ 이고  $BV[1] > -n$ 이면 추가로 분할될 수 있는 보다 작은 TB에 대해 동일한 조건이 검사(예컨대, 재귀적으로 검사)된다.

[0125] 예를 들어, PB에 대한 BV가 (-9, -5)이고 현재 TB가 32x32 블록인 것으로 가정한다. 인코더 및 디코더는  $-9 > -32$ 이고  $-5 > -32$ 인 것으로 결정하고, 이는 인트라 예측 영역(인트라 예측 영역의 좌측 상단 코너가 -9, -5만큼 변위됨)이 현재 32x32 TB와 중첩할 것임을 나타낸다. 인코더 및 디코더는 32x32 TB를 4 개의 16x16 TB로 분할한다. 16x16 TB 각각에 대해, 인코더 및 디코더는  $-9 > -16$ 이고  $-5 > -16$ 인 것으로 결정하고, 이는 인트라 예측 영역(인트라 예측 영역의 좌측 상단 코너가 -9, -5만큼 변위됨)이 현재 16x16 TB와 중첩할 것임을 나타낸다. 인코더 및 디코더는 각각의 16x16 TB를, 연속적으로, 4 개의 8x8 TB로 분할한다. 8x8 TB에 대해, (-9, -5)의 BV는 문제가 되지 않으며, 따라서 8x8 TB가 강제로 추가로 분할되지 않는다.

[0126] 이 시나리오에서, TB의 BV 값 및 크기로 인해 TB가 분할될 때, 인코더는 그렇지 않았으면 현재 TB를 보다 작은 TB로 분할해야만 하는지를 신호하게 될 플래그 값을 신호하는 것을 생략할 수 있다. 인코딩된 데이터의 비트스트림이 현재 TB를 보다 작은 TB로 분할하라고 디코더에 지시하는 플래그 값을 갖지 않는다. 그 대신에, 디코더는 TB가 TB의 BV 값 및 크기로 인해 분할되어야만 하는 것으로 추론할 수 있다. 이것은 그렇지 않았으면 TB를 분할하는 것에 관한 정보를 신호하는 데 소비될 비트를 절감할 수 있다.

### C. BV 값의 인코딩 및 디코딩.

[0127] 전체적으로, 인트라 BC 예측을 사용하여 인코딩되는 블록에 대한 BV 값은 상당한 수의 비트를 소비할 수 있다. 비트 레이트를 감소시키기 위해 BV 값이 엔트로피 인코딩될 수 있다. BV 값에 대한 비트 레이트를 추가로 감소시키기 위해, 인코더는 BV 값의 예측을 사용할 수 있다. BV 값은 종종 중복성을 나타낸다 - 주어진 블록에 대한 BV 값이 꾹처에서의 이전 블록의 BV 값과 종종 유사하거나 심지어 동일하다 -. BV 예측에 있어서, 주어진 블록에 대한 BV 값이 BV 예측자(BV predictor)를 사용하여 예측된다. 주어진 블록에 대한 BV 값과 BV 예측자 사이의 차이(또는 BV 차분(BV differential))가 이어서 엔트로피 코딩된다. 전형적으로, BV 값 및 BV 예측자의 수평 성분 및 수직 성분에 대해 BV 차분이 계산된다. BV 예측이 잘 동작할 때, BV 차분은 효율적인 엔트로피 코딩을 지원하는 확률 분포를 갖는다. 인코더 및 디코더는 BV 값을 인코딩/디코딩할 때 기본적인 BV 예측 및/또는 병합 모드/BV 경쟁(BV competition)을 사용할 수 있다.

[0128] 인코딩 및 디코딩 동안 기본적인 BV 예측에 있어서, 현재 블록에 대한 BV 값이 하나 이상의 이전 블록의 BV 값에 기초하여 예측될 수 있다. 예를 들어, 이웃 블록(예컨대, 현재 블록의 좌측에 있는 블록)의 BV 값이 현재 블록의 BV 값에 대한 BV 예측자를 결정하는 데 사용될 수 있다. 현재 블록의 BV 값이 (-80, -24)이고 BV 예측자가 (-80, -32)인 경우, (0, 8)의 BV 차분이 엔트로피 인코딩된다. 또는, 현재 블록의 BV 값에 대한 BV 예측자가 다수의 이웃 블록(예컨대, 현재 블록의 좌측, 위쪽 및 좌측 위쪽에 있는 블록)의 BV 값들의 성분별 중간값(median) 또는 평균(average)일 수 있다.

[0129] 디코딩 동안, 디코더는 BV 값에 대한 엔트로피 코딩된 BV 차분(있는 경우)을 수신하여 엔트로피 디코딩한다. 디코더는 또한 BV 값에 대한 BV 예측자를 결정한다. (디코더에 의해 결정된 BV 예측자는 인코더에 의해 결정된

BV 예측자와 동일하다.) 재구성된 BV 차이(BV difference)(있는 경우)가 이어서 BV 예측자와 결합된다.

[0131] 인코더 및 디코더는, 이전 블록으로부터의 실제 BV 값이 이용 가능하지 않을 때(예컨대, 주어진 CTU에서 첫 번째 인트라 BC 예측된 블록에 대한 BV 값을 결정할 때), 기본 BV 예측자(default BV predictor)를 사용할 수 있다. 예를 들어, 기본 BV 예측자는 (0, 0)일 수 있다. 또는, 기본 BV 예측자는 (-W, 0) 또는 (-2\*W, 0)일 수 있고, 여기서 W는 현재 블록의 폭이다. 또는, 기본 BV 예측자는 (0, -H) 또는 (0, -2\*H)일 수 있고, 여기서 H는 현재 블록의 높이이다. 영 값 기본 BV 예측자와 비교하여, 영이 아닌 성분을 갖는 기본 BV 예측자가 현재 블록의 BV 값에 더 가까운 경향이 있으며, 그 결과 BV 차분의 보다 효율적인 엔트로피 코딩이 얻어진다. 또는, 기본 BV 예측자의 성분이 고정된 영이 아닌 값(예컨대, 8 또는 16)을 가질 수 있다.

[0132] 어쨌든, 현재 블록에 대해 가능한 BV 값이 특정 검색 범위(예컨대, 이전 섹션에서 기술된 바와 같이, 현재 CTB 및 현재 CTB의 좌측에 있는 CTB) 내에 있도록 제약되고 BV 값이 현재 블록과 그의 인트라 예측 영역 간의 중첩을 야기할 수 없을 때, 어떤 경우에, 현재 블록에 대한 BV 값이 하나의 가능한 값만을 가질 수 있다. 예를 들어, (1) 현재 CTB가 현재 블록으로서 단일의 CB를 갖고, (2) BV 값에 대한 검색 범위가 현재 CTB 및 그의 좌측에 있는 CTB이며, (3) 현재 블록과 그의 인트라 예측 영역 간의 중첩이 허용되지 않는 경우, 단일의 CB에 대한 BV 값은 (-W, 0)의 BV 값을 가져야만 한다. 다른 BV 값은 검색 범위 밖에 있는 위치를 참조하거나 현재 블록 내의 위치를 참조한다. 이 상황에서, 인코더는 현재 블록에 대한 BV 값(또는 BV 차분)을 신호하는 것을 생략할 수 있다. 디코더는 현재 블록에 대한 임의의 BV 값(또는 BV 차분)을 파싱하고 디코딩하는 일 없이 상황을 식별하고 (-W, 0)의 BV 값을 추론하기 위해 조건을 검사할 수 있다.

[0133] 또는, 인코더 및 디코더는 현재 블록에 공간적으로 이웃하는 재구성된 블록(예컨대, 현재 블록의 좌측에 있는 블록, 현재 블록의 위쪽에 있는 블록 등)에 대해 사용된 BV 값들 중에서 현재 블록에 대한 하나 이상의 후보 BV 값을 결정한다. 후보 BV 값(들)은 또한 현재 블록에 시간적으로 이웃하는 재구성된 블록에 대해 사용된 하나 이상의 BV 값을 포함할 수 있고, 여기서 시간적으로 이웃하는 블록은 다른 빙처에서의 현재 블록에 대응하는 위치(예컨대, 동일한 위치 또는 중첩하는 위치)에 있다. 중복적인 BV 값을 제거하기 위해 인코딩 및 디코딩 동안 규칙에 의해 후보 BV 값(들)의 리스트가 결정될 수 있다. 인코딩 동안, 인코더는 후보 BV 값(들) 중 어느 것을 현재 블록에 대한 BV 예측자로서 사용할지를 나타내는 하나 이상의 구문 요소를 신호할 수 있다. 일부 모드에서, 현재 블록의 BV 값을 이웃의 BV 값과 효과적으로 "병합"하여 후보 BV 값을 제공하는 그 BV 예측자가 현재 블록에 대한 BV 값으로서 사용될 수 있다. 또는, 인코더가 BV 값 및 BV 예측자에 기초하여 BV 차이를 결정하고 인코딩할 수 있다. 디코딩 동안, 디코더는 후보 BV 값(들) 중 어느 것을 현재 블록에 대한 BV 예측자로서 사용할지를 나타내는 하나 이상의 구문 요소를 수신할 수 있다. 일부 모드에서, 현재 블록의 BV 값을 이웃의 BV 값과 효과적으로 "병합"하여 후보 BV 값을 제공하는 그 BV 예측자가 현재 블록에 대한 BV 값으로서 사용될 수 있다. 또는, 디코더가 BV 차이를 수신하여 디코딩할 수 있고, 디코더는 BV 값을 재구성하기 위해 BV 차이를 BV 예측자와 결합시킨다. (규칙에 의해 선택되는) BV 예측자가 현재 블록의 BV 값으로서 사용되고 현재 블록에 대한 어떤 잔차 값도 신호되지 않는 BV "생략(skip)" 또는 BV "직접(direct)" 모드가 제공될 수 있다. 인코더 또는 디코더가 블록에 대한 BV 값을 예측할 때 후보 BV 값 쌍 중에서 선택하는, 평庸 접근법을 사용하는 BV 예측의 예가 이하에서 기술된다.

#### D. 인트라 BC 예측의 특징들을 결합시키는 예시적인 구현.

[0134] 살펴본 바와 같이, 인트라 BC 예측의 전술한 특징들이 따로따로 그리고 개별적으로 사용될 수 있다. 또는, 인트라 BC 예측의 전술한 특징들이 결합하여 사용될 수 있다.

[0135] 예를 들어, 일반적으로 HEVC 구문을 따르는 하나의 결합된 구현에서, (CU 또는 CU의 일부일 수 있는) PU에 대해 BV 값이 신호된다. PU는 하나 이상의 TU를 포함할 수 있다. 인트라 BC 예측 프로세스는, PU에 대해 신호된 BV 값을 사용하여, TB의 레벨에서 TB 단위로 행해진다. (모든 TB가 동일한 BV 값을 사용하고, 현재 TB에 대한 인트라 BC 예측은 동일한 CU에서의 다른 이전의 TB의 재구성된 샘플 값을 사용할 수 있다.) BV 값이 하나 이상의 이웃하는 PU의 BV 값을 사용하여(예컨대, BV 예측에 대한 평庸 접근법을 사용하여) 예측될 수 있다. (a) 인코더가 아직 인코딩/재구성되지 않은 구역 내에 있는 임의의 샘플 값이 참조되게 할 BV 값을 선택하지 못하도록 (즉, 현재 TB에 대한 인트라 예측 영역의 샘플 값이 디코딩/비트스트림 순서에서 현재 TB에 선행하는 다른 TB에 의해 커버되는 구역에 있어야 함; 즉, 주어진 TB에 대해, BV 값이 TB 밖에 있는 영역을 참조하도록 제약됨); (b) (예컨대, BV 값에 따른 참조를 현재 CTB 및 현재 CTB의 왼쪽에 있는 하나의 또는 2 개의 CTB 내에 있도록 제약함으로써) 디코더에서의 필요한 메모리 용량을 감소시키기 위해; 그리고 (c) BV 값에 따른 참조가 현재 슬라이스 밖에, 현재 타일 밖에 또는 빙처 밖에 있지 않도록 하기 위해 BV 값의 선택이 제약된다.

- [0137] **VIII. 블록 벡터 예측에서의 혁신.**
- [0138] 본원에 기술되는 혁신의 일 양태에 따르면, 인코더 및 디코더는 "평퐁" 접근법에 따라 BV(block vector) 예측을 사용한다. 상세하게는, BV 예측에 대한 예시적인 평퐁 접근법은, 일부 시나리오에서 화면 콘텐츠 비디오를 인코딩할 때, 코딩 효율을 개선시킨다.
- [0139] 예시적인 평퐁 접근법에서, 인코더 및 디코더는 상태 정보로서 후보 BV 값 쌍을 유지한다. 도 11은 예시적인 평퐁 접근법에서 사용될 수 있는 BV 베퍼(1100) 내의 후보 BV 값 쌍을 나타낸 것이다. BV 베퍼(1100)에서, BV 베퍼 0은  $BV_{init\_0}$ 이라고 표시된 제1 초기 후보 BV 값을 저장한다. BV 베퍼 1은  $BV_{init\_1}$ 이라고 표시된 제2 초기 후보 BV 값을 저장한다. 일반적으로, BV 베퍼 0에 대한 BV 값 변수는 PBV0로 표시될 수 있고, BV 베퍼 1에 대한 BV 값 변수는 PBV1로 표시될 수 있다.
- [0140] 도 12는 예시적인 평퐁 접근법에 따른, 후보 BV 값 쌍을 저장할 수 있는 다른 데이터 구조(1200)를 나타낸 것이다. 데이터 구조(1200)는  $BV_{init\_0}$ 이라고 표시되고 인덱스 값 idx0과 연관되어 있는 제1 초기 후보 BV 값을 저장한다. 제2 초기 후보 BV 값( $BV_{init\_1}$ )은 데이터 구조(1200)에서 인덱스 값 idx1과 연관되어 있다. 일반적으로, idx0과 연관된 BV 값 변수는 PBV0로 표시될 수 있고, idx1과 연관된 BV 값 변수는 PBV1로 표시될 수 있다.
- [0141] 대안적으로, 인코더 및 디코더는 평퐁 접근법에 따라 BV 예측을 위한 후보 BV 값 쌍을 추적하기 위해 다른 데이터 구조를 사용한다.
- [0142] 예시적인 평퐁 접근법에 따르면, 후보 BV 값 쌍 중에서, 인코더는 PU의 BV 값에 대해 사용할 BV 예측자를 선택한다. 예를 들어, PU가 선택된 후보 BV 값에 의한 인트라 BC 예측을 사용하여 인코딩될 때, 인코더는 참조된 인트라 예측 영역이 어떤 메트릭(예컨대, 절대차의 합(sum of absolute difference), 평균 제곱 오차(mean squared error))에 의해 PU와 가장 가깝게 일치하는 후보 BV 값을 선택한다. 또는, PU가 인트라 BC 예측을 위해 사용될 BV 값(BV 추정을 통해 식별됨)을 가질 때, 인코더는 PU에 대한 BV 값과 가장 가깝게 일치하는 후보 BV 값을 선택한다. 이러한 선택의 결과, 가장 작은 BV 차분이 얻어지고, 이는 엔트로피 코딩의 효율을 개선시키는 경향이 있다.
- [0143] BV 값에 대해, 인코더는, 후보 BV 값 쌍 중 어느 것이 BV 예측자로서 사용되어야 하는지를 디코더에 알려주기 위해, 후보 BV 값 쌍 중 선택된 후보 BV 값을 나타내는 플래그 값을 신호한다. 플래그 값은 엔트로피 코딩되거나 고정 길이 값으로서 신호될 수 있다. 플래그 값이 (조건에 따라) 비트스트림에서 별도의 구문 요소로서 신호될 수 있거나, 플래그 값이 비트스트림에서 다른 구문 요소와 함께 신호될 수 있다.
- [0144] 후보 BV 값 쌍에서의 각각의 후보 BV 값이 똑같이 BV 예측자일 가능성이 있을 때, 플래그 값이 효율적이게도 고정 길이 값(예컨대, 1 비트)으로서, 또는 바이패스 모드(bypass mode)에서 이진 산술 코딩(binary arithmetic coding)을 사용하여(2 개의 가능한 값이 확률이 같은 것으로 가정함) 신호될 수 있다. 제1 값(예컨대, 0)은 제1 후보 BV 값의 선택을 나타내고, 제2 값(예컨대, 1)은 제2 후보 BV 값의 선택을 나타낸다. 후보 BV 값 쌍 중 하나의 후보 BV 값이 다른 후보 BV 값보다 가능성이 더 높지만 코딩 효율이 감소될 수 있을 때, 플래그 값이 또한 고정 길이 값으로서 신호될 수 있다.
- [0145] 쌍에서의 후보 BV 값을 중 하나의 후보 BV 값이 BV 예측자로서 선택될 가능성이 더 높을 때, 플래그 값을 신호하기 위해 컨텍스트 기반 산술 코딩(context-based arithmetic coding)을 사용하는 것이 보다 효율적일 수 있다. 예를 들어, 플래그 값에 대한 확률 내용(probability content)이, 슬라이스에 대해 코딩 또는 디코딩이 시작될 때, 2 개의 가능한 플래그 값에 대해 똑같은 확률로 초기화되고, 이어서 상이한 플래그 값의 출현을 고려하기 위해 코딩/디코딩 동안 업데이트될 수 있다. BV 예측자 플래그 값에 대한 개별적인 컨텍스트를 갖는 것이 코딩 효율을 개선시킬 수는 있지만, 컨텍스트 모델링 등으로 인해 인코딩 및 디코딩의 복잡성을 증가시킨다. 전형적으로, 보다 빈번히 선택될 것으로 예상되는 후보 BV 값이 제1 BV 베퍼(예컨대, BV 베퍼 0)에 저장되고 제1 플래그 값으로 표시되며, 덜 빈번히 선택될 것으로 예상되는 후보 BV 값이 다른 BV 베퍼(예컨대, BV 베퍼 1)에 저장되고 다른 플래그 값으로 표시된다. 이와 같이, 제1 BV 베퍼를 나타내는 플래그 값은, 평균적으로, 산술 코딩에서 보다 적은 수의 소수 부분 비트(fractional bit)를 사용하여 인코딩될 수 있다. 그렇지만, 어떤 내용에 대해서는, 보다 흔한 후보 BV 값이 다른 BV 베퍼에 저장될 수 있고, 컨텍스트 모델링(context modeling)이 BV 예측자 분포의 변화를 고려한다.
- [0146] 디코더는 BV 값에 대한 플래그 값을 파싱/디코딩하고, 후보 BV 값 쌍 중 어느 것이 BV 값에 대한 BV 예측자인지 를 결정하기 위해 플래그 값을 사용한다. 디코더는 선택된 후보 BV 값을 BV 예측자로서 사용한다.

[0147] 인코더 및 디코더는 어느 BV 값이 후보 BV 값 쌍에 있는지를 추적하고 업데이트한다. 이전의 평평 접근법에서 는, 새로운 BV 값이 생성(또는 재구성)될 때, 쌍에서의 보다 오래된 후보 BV 값(PBV1)이 쌍에서의 보다 새로운 BV 값(PBV0)으로 대체되고, 쌍에서의 보다 새로운 BV 값(PBV0)이 방금 생성된/재구성된 BV 값(BV<sub>new</sub>)에 의해 대체된다. (즉, PBV1 = PBV0이고, PBV0 = BV<sub>new</sub>이다.) 각각의 BV 값이 생성되거나 재구성될 때 후보 BV 값 쌍이 업데이트된다.

[0148] 본원에 기술되는 예시적인 평평 접근법에서, 인코더 및 디코더는 BV 값이 생성되거나 재구성된 후에 BV 값을 포함하도록 후보 BV 값 쌍을 자동으로 업데이트할 수 있다. 그렇지만, 이 결과, 2 개의 동일한 후보 BV 값을 가지는 후보 BV 값 쌍이 얻어질 수 있다. 이 상황에서, (양쪽 후보 BV 값이 똑같다는 것을 인식하는) 인코더는 후보 BV 값 쌍의 선택을 나타내는 플래그 값을 신호하는 것을 생략할 수 있고, (양쪽 후보 BV 값이 똑같다는 것을 인식하는) 디코더는 플래그 값을 파싱하는 것을 생략할 수 있고, 그 대신에 똑같은 후보 BV 값 중 어느 하나를 BV 예측자로서 사용한다.

[0149] 또는, 본원에 기술되는 예시적인 평평 접근법에서, 인코더 및 디코더는 똑같은 후보 BV 값을 가지는 것을 피하기 위해 후보 BV 값 쌍(및 BV 버퍼)을 선택적으로 업데이트할 수 있다. 인코더 또는 디코더는, 새로운 BV 값이 쌍에 남아 있게 될 후보 BV 값과 같지 않을 때, 후보 BV 값 쌍을 업데이트한다. 새로운 BV 값이 쌍에 남아 있게 될 후보 BV 값과 같다면, 인코더 또는 디코더는 업데이트 동작을 생략한다. 예를 들, 인코더 또는 디코더는 다음과 같이 새로운 BV 값(BV<sub>new</sub>)을 포함하도록 후보 BV 값 쌍(PBV0 및 PBV1)을 선택적으로 업데이트한다.

[0150] if (BV<sub>new</sub> != PBV0) {

[0151] PBV1 = PBV0;

[0152] PBV0 = BV<sub>new</sub>;

[0153] } else {

[0154] do nothing;

[0155] }

[0156] 조건 BV<sub>new</sub> != PBV0을 검사하는 것에 의해, 인코더 및 디코더는 PBV0이 PBV1과 같은 상황 - 이 경우에 BV 예측자 를 나타내는 플래그 값이 낭비됨 - 을 피한다. 다른 한편으로, 조건을 검사하는 것은 BV 예측에 복잡성을 부가하고, 이는 자원 요구사항이 주어진 경우 타당하지 않을 수 있거나 코딩 효율에서의 이득에 의해 정당화되지 않을 수 있다.

[0157] 예를 들어, PBV0이 (-12, 0)이고 PBV1이 (-16, 0)인 것으로 가정한다. BV<sub>new</sub>가 (-17, 0)인 경우, (-12, 0)은 PBV1에 대한 (-16, 0)을 대체하고, (-17, 0)은 PBV0에 대한 (-12, 0)을 대체한다. PBV1로부터의 이전 BV 값(-16, 0)은 폐기된다. 다른 한편으로, BV<sub>new</sub>가 (-12, 0)인 경우, PBV0 및 PBV1은 변하지 않는데, 그 이유는 BV<sub>new</sub> 가 PBV0과 일치하기 때문이다. BV<sub>new</sub>가 PBV1과 일치하면(이 예에서, 둘 다 (-16, 0)임), PBV0과 PBV1의 순서가 효과적으로 스위칭된다 - (-12, 0)이 PBV1에 대한 (-16, 0)을 대체하고 (BV<sub>new</sub>에 대한) (-16, 0)이 PBV0에 대한 (-12, 0)을 대체한다 -.

[0158] 본원에 기술되는 예시적인 평평 접근법에서, 4 개의 PU를 가지는 CU에 대해, 인코더 및 디코더는 BV 예측을 위해 사용되는 후보 BV 값 쌍을 업데이트할 때 상이한 패턴을 사용한다. 개념적으로, 상이한 패턴은, PU-레벨 업데이트가 아니라, 후보 BV 값 쌍의 CU-레벨 업데이트를 사용한다. CU에 대해 BV 예측이 완료된 후에, 후보 BV 값 쌍은 CU의 처음 2 개의 PU의 BV 값 - BV 버퍼 0 내의 PU1에 대한 BV 값, 및 BV 버퍼 1 내의 PU0에 대한 BV 값 - 을 포함한다. 이와 달리, 이전의 평평 접근법에 따르면, CU에 대해 BV 예측이 완료된 후에, 후보 BV 값 쌍은, PU3 및 PU2일 수 있는, CU의 마지막 2 개의 PU의 BV 값을 포함한다.

[0159] 대안적으로, 인코더 및 디코더는 BV 예측을 위해 사용되는 후보 BV 값 쌍을 업데이트할 때 상이한 패턴을 사용할 수 있다. 예를 들어, 플래그 값이 고정 길이 값을 사용하여 코딩되는 경우, 인코더 및 디코더는 2 개의 BV 버퍼를 방금 생성되거나 재구성된 BV 값으로 업데이트하는 것 사이에서 "토글(교번)" 할 수 있다. 이러한 방식으로, 인코더 및 디코더는 (보다 새로운 후보 BV 값을 보다 오래된 후보 BV 값의 지점으로 천이시키고 이어서 새로운 BV 값을 보다 새로운 후보 BV 값의 이전 지점에 추가하는 것 대신에) 보다 오래된 후보 BV 값을 대체하

기 위해 단일의 동작을 사용할 수 있지만, 인코더 및 디코더는 어느 BV 버퍼가 다음에 업데이트되어야만 하는지를 추적할 필요가 있다. 또한, 토글링은 컨텍스트 적응적 이진 산술 코딩을 방해함으로써 플래그 값에 대한 코딩 효율을 해칠 수 있다.

[0160] 실제 BV 값이 (예컨대, CTU의 첫 번째 CU를 코딩 또는 디코딩하기 전에) 후보 BV 값으로서 사용될 수 없을 때, 후보 BV 값 쌍이 기본값으로 초기화될 수 있다. 예를 들어, 후보 BV 값 둘 다가 (-2W, 0)로 초기화되고, 여기서 W는 CU의 폭이다. 또는, 후보 BV 값 중 하나가 (-2W, 0)로 초기화될 수 있고, 다른 후보 BV 값이 (0, -2H)로 초기화될 수 있으며, 여기서 H는 CU의 높이이다(그리고 W와 동일할 수 있다). 또는, 후보 BV 값이 (-W, 0) 또는 (0, -H)와 같은 다른 값으로 초기화될 수 있다. 또는, 간단함을 위해, 실제 CU 크기에 의존하는 기본값을 사용하는 대신에, 후보 BV 값이 (-8, 0), (-16, 0), (0, -8) 또는 (0, -16)과 같은 고정 값으로 또는 비트스트림에서 신호되는 값으로 초기화될 수 있다. 이것은, 예를 들어, 인트라 BC 예측에 대해 크기 8 또는 16의 CU가 가장 흔할 때, 적절할 수 있다. 어쨌든, BV 예측의 시작 후에, 실제 BV 값이 후보 BV 값 쌍에서의 기본 BV 값을 대체한다.

[0161] 하기의 예 모두에서, 달리 언급되지 않는 한, BV 버퍼 0은 BV 버퍼 1에 저장된 다른 후보 BV 값보다 선택될 가능성이 더 높은 "선후된(favored)" 후보 BV 값을 저장한다. 컨텍스트 적응적 이진 산술 코딩은 보다 흔한 BV 예측자 플래그 값이 플래그 값에 대한 코딩 효율을 개선시킬 확률이 더 높다는 것을 이용할 수 있다.

#### A. 2Nx2N CU의 2Nx2N PU에 대한 예시적인 BV 예측.

[0163] 도 13은 예시적인 평庸 접근법에 따른, 2Nx2N CU의 2Nx2N PU에 대한 BV 예측의 일 예를 나타낸 것이다. 2Nx2N CU는 단일의 2Nx2N PU(PU0) - 그의 BV가 후보 BV 값 쌍을 사용하여 예측됨 - 를 포함한다.

[0164] 도 13에서, BV 버퍼 0은 처음에 제1 초기 후보 BV 값(BV<sub>init\_0</sub>)(PBV0)을 저장하고 있으며, BV 버퍼 1은 처음에 제2 초기 후보 BV 값(BV<sub>init\_1</sub>)(PBV1)을 저장하고 있다. 단일의 2Nx2N PU(PU0)에 대한 BV 값은, 2Nx2N PU에 대한 플래그 값으로 표시된 바와 같이, PBV0 또는 PBV1 중 어느 하나를 사용하여 예측된다. PU0에 대한 BV 값이 (예컨대, 인코딩 또는 디코딩 동안) 선택된 BV 예측자를 사용하여 처리된다.

[0165] 후보 BV 값 쌍이 이어서 다음과 같이 업데이트된다. PBV1이 PBV0로 대체되고, PBV0이 PU0에 대한 BV 값으로 대체된다. 즉, PBV1 = PBV0이고, PBV0 = BV PU0이다. 도 13에서, BV 버퍼 1은 BV<sub>init\_0</sub>을 저장하도록 업데이트되고, BV 버퍼 0은 PU0에 대한 BV 값(BV<sub>PU0</sub>)을 저장하도록 업데이트된다. 대안적으로, 앞서 설명된 바와 같이, 인코더 또는 디코더는, 새로운 BV 값(BV<sub>new</sub>)(여기서, BV<sub>PU0</sub>)이 PBV0과 같은지에 따라, 후보 BV 값 쌍(및 BV 버퍼)을 선택적으로 업데이트한다.

#### B. 2Nx2N CU의 NxN PU에 대한 예시적인 BV 예측.

[0167] 도 14는 예시적인 평庸 접근법에 따른, 2Nx2N CU의 NxN PU에 대한 BV 예측의 일 예를 나타낸 것이다. 2Nx2N CU는 4 개의 NxN PU(PU0, PU1, PU2 및 PU3) - 그의 BV 값은 CU에 대한 BV 예측 프로세스 동안 업데이트되는 후보 BV 값 쌍을 사용하여 예측됨 - 를 포함한다.

[0168] 도 14에서, BV 버퍼 0은 처음에 제1 초기 후보 BV 값(BV<sub>init\_0</sub>)(PBV0)을 저장하고 있으며, BV 버퍼 1은 처음에 제2 초기 후보 BV 값(BV<sub>init\_1</sub>)(PBV1)을 저장하고 있다. 첫 번째 NxN PU(PU0)에 대한 BV 값은, PU0에 대한 플래그 값으로 표시된 바와 같이, PBV0 또는 PBV1 중 어느 하나를 사용하여 예측된다. PU0에 대한 BV 값(BV<sub>PU0</sub>)이 (예컨대, 인코딩 또는 디코딩 동안) 선택된 BV 예측자를 사용하여 처리된다.

[0169] 후보 BV 값 쌍이 이어서 PBV1을 PBV0으로 대체하는 것에 의해 그리고 PBV0을 PU0에 대한 BV 값으로 대체하는 것에 의해 업데이트된다. 즉, PBV1 = PBV0이고, PBV0 = BV<sub>PU0</sub>이다. 도 14에서, BV 버퍼 1은 BV<sub>init\_0</sub>을 저장하도록 업데이트되고, BV 버퍼 0은 BV<sub>PU0</sub>을 저장하도록 업데이트된다. 대안적으로, 앞서 설명된 바와 같이, 인코더 또는 디코더는, 새로운 BV 값(BV<sub>new</sub>)(여기서, BV<sub>PU0</sub>)이 PBV0과 같은지에 따라, 후보 BV 값 쌍(및 BV 버퍼)을 선택적으로 업데이트한다.

[0170] 다음에, 두 번째 NxN PU(PU1)에 대한 BV 값이, PU1에 대한 플래그 값으로 나타낸 바와 같이, PBV0(도 14에서, 그의 좌측 이웃의 BV<sub>PU0</sub>) 또는 PBV1(도 14에서, BV<sub>init\_0</sub>) 중 어느 하나를 사용하여 예측된다. PU1에 대한 BV 값

( $BV_{PU1}$ )이 (예컨대, 인코딩 또는 디코딩 동안) 선택된 BV 예측자를 사용하여 처리된다.

[0171] 후보 BV 값 쌍이 이어서 PBV1을 PBV0으로 대체하는 것에 의해 그리고 PBV0을 PU1에 대한 BV 값으로 대체하는 것에 의해 업데이트된다. 즉,  $PBV1 = PBV0$ 이고,  $PBV0 = BV_{PU1}$ 이다. 도 14에서, BV 버퍼 1은  $BV_{PU0}$ 을 저장하도록 업데이트되고, BV 버퍼 0은  $BV_{PU1}$ 을 저장하도록 업데이트된다. 대안적으로, 앞서 설명된 바와 같이, 인코더 또는 디코더는, 새로운 BV 값( $BV_{new}$ )(여기서,  $BV_{PU1}$ )이 PBV0과 같은지에 따라, 후보 BV 값 쌍(및 BV 버퍼)을 선택적으로 업데이트한다.

[0172] 다음에, 세 번째 NxN PU(PU2)에 대한 BV 값이, PU2에 대한 플래그 값으로 나타낸 바와 같이, PBV0(도 14에서, 그의 우측 상부 이웃의  $BV_{PU1}$ ) 또는 PBV1(도 14에서, 그의 상부 이웃의  $BV_{PU0}$ ) 중 어느 하나를 사용하여 예측된다. 이 플래그 값에 대해, BV 버퍼 0이 아니라 BV 버퍼 1이 선호된 후보 BV 값을 저장한다. 이 조절은  $BV_{PU0}$ (상부 이웃)이  $BV_{PU1}$ (우측 상부 이웃)보다 더 유망한 후보라는 것을 고려한다. PU2에 대한 BV 값( $BV_{PU2}$ )이 (예컨대, 인코딩 또는 디코딩 동안) 선택된 BV 예측자를 사용하여 처리된다. 그 후에, BV 버퍼 0 내의 후보 BV 값(도 14에서,  $BV_{PU1}$ ) 및 BV 버퍼 1 내의 후보 BV 값(도 14에서,  $BV_{PU0}$ )은 변하지 않는다.

[0173] 마지막으로, 네 번째 NxN PU(PU3)에 대한 BV 값이, PU3에 대한 플래그 값으로 나타낸 바와 같이, 다른 장소에 버퍼링되는 PU2에 대한 BV 값( $BV_{PU2}$ ) 또는 PBV1(도 14에서, 그의 상부 이웃의  $BV_{PU1}$ ) 중 어느 하나를 사용하여 예측된다. 이 플래그 값에 대해, BV 버퍼 0이 또다시 선호된 후보 BV 값을 저장하지만, 다른 후보 BV 값이 BV 버퍼 1에 저장되지 않는다. PU3에 대한 BV 값( $BV_{PU3}$ )이 (예컨대, 인코딩 또는 디코딩 동안) 선택된 BV 예측자를 사용하여 처리된다. 그 후에, BV 버퍼 0 및 BV 버퍼 1 내의 후보 BV 값은 변하지 않는다. 도 14에서, CU의 4 개의 NxN PU에 대한 BV 예측의 끝에서, 후보 BV 값 쌍은  $BV_{PU1}$ (BV 버퍼 0에 있음) 및  $BV_{PU0}$ (BV 버퍼 1에 있음)이다.

[0174] 이와 달리, 이전의 평평 접근법에 따르면, 2Nx2N CU의 NxN PU에 대한 BV 예측에 대해, BV 값이 생성되거나 재구성될 때, BV 버퍼 0 내의 후보 BV 값은 BV 버퍼 1로 이동되고, 방금 생성/재구성된 BV 값은 BV 버퍼 0에 저장된다. 2Nx2N CU의 4 개의 NxN PU에 대한 BV 예측의 끝에서, 후보 BV 값 쌍은  $BV_{PU3}$ (BV 버퍼 0에 있음) 및  $BV_{PU2}$ (BV 버퍼 1에 있음)일 것이다. 실제로는, 이전의 평평 접근법에 대한 몇 가지 단점이 있다. 다음 CU가 단일의 2Nx2N PU를 갖는 경우, 현재 CU의  $BV_{PU0}$ 이 보통 최상의 BV 예측자일 것이지만, 현재 CU의  $BV_{PU0}$ 이 후보 BV 값을 중 하나로서 유지되지 않는다. 또는, 다음 CU가 2 개의 Nx2N PU를 갖는 경우, 현재 CU의  $BV_{PU0}$  및  $BV_{PU1}$ 이 보통, 각각, 다음 CU의 PU0 및 PU1에 대한 최상의 BV 예측자일 것이지만, 현재 CU의  $BV_{PU0}$  및  $BV_{PU1}$ 이 후보 BV 값으로서 유지되지 않는다.

[0175] 그렇지만, 본원에 기술되는 예시적인 평평 접근법에 따르면, (도 14에 도시된 바와 같은) 2Nx2N CU의 NxN PU에 대한 BV 예측 후에, 후보 BV 값 쌍은  $BV_{PU1}$ (BV 버퍼 0에 있음) 및  $BV_{PU0}$ (BV 버퍼 1에 있음)이다. 이것은 다음 CU에 대한 BV 예측에 가장 유용할 가능성이 있는 후보 BV 값을 유지한다.

### C. 2Nx2N CU의 Nx2N PU 또는 2NxN PU에 대한 예시적인 BV 예측.

[0176] 도 15는 예시적인 평평 접근법에 따른, 2Nx2N CU의 2 개의 Nx2N PU 또는 2 개의 2NxN PU에 대한 BV 예측의 일 예를 나타낸 것이다. 2Nx2N CU는 2 개의 Nx2N PU(좌측 PU0 및 우측 PU1) 또는 2 개의 2NxN PU(상부 PU0 및 하부 PU1) - 그의 BV 값은 CU에 대한 BV 예측 프로세스 동안 업데이트되는 후보 BV 값 쌍을 사용하여 예측됨 - 를 포함한다.

[0178] 도 15에서, BV 버퍼 0은 처음에 제1 초기 후보 BV 값( $BV_{init\_0}$ )(PBV0)을 저장하고 있으며, BV 버퍼 1은 처음에 제2 초기 후보 BV 값( $BV_{init\_1}$ )(PBV1)을 저장하고 있다. 첫 번째 PU(PU0)에 대한 BV 값은, PU0에 대한 플래그 값으로 표시된 바와 같이, PBV0 또는 PBV1 중 어느 하나를 사용하여 예측된다. PU0에 대한 BV 값( $BV_{PU0}$ )이 (예컨대, 인코딩 또는 디코딩 동안) 선택된 BV 예측자를 사용하여 처리된다.

[0179] 후보 BV 값 쌍이 이어서 PBV1을 PBV0으로 대체하는 것에 의해 그리고 PBV0을 PU0에 대한 BV 값으로 대체하는 것에 의해 업데이트된다. 즉,  $PBV1 = PBV0$ 이고,  $PBV0 = BV_{PU0}$ 이다. 도 15에서, BV 버퍼 1은  $BV_{init\_0}$ 을 저장하도록

업데이트되고, BV 버퍼 0은  $BV_{PU0}$ 을 저장하도록 업데이트된다. 대안적으로, 앞서 설명된 바와 같이, 인코더 또는 디코더는, 새로운 BV 값( $BV_{new}$ )(여기서,  $BV_{PU0}$ )이 PBV0과 같은지에 따라, 후보 BV 값 쌍(및 BV 버퍼)을 선택적으로 업데이트한다.

[0180] 다음에, 두 번째 PU(PU1)에 대한 BV 값이, PU1에 대한 플래그 값으로 나타낸 바와 같이, PBV0(도 15에서, 그의 이웃의  $BV_{PU0}$ ) 또는 PBV1(도 15에서,  $BV_{init\_0}$ ) 중 어느 하나를 사용하여 예측된다. PU1에 대한 BV 값( $BV_{PU1}$ )이 (예컨대, 인코딩 또는 디코딩 동안) 선택된 BV 예측자를 사용하여 처리된다.

[0181] 후보 BV 값 쌍이 이어서 PBV1을 PBV0으로 대체하는 것에 의해 그리고 PBV0을 PU1에 대한 BV 값으로 대체하는 것에 의해 업데이트된다. 즉,  $PBV1 = PBV0$ 이고,  $PBV0 = BV_{PU1}$ 이다. 도 15에서, BV 버퍼 1은  $BV_{PU0}$ 을 저장하도록 업데이트되고, BV 버퍼 0은  $BV_{PU1}$ 을 저장하도록 업데이트된다. 대안적으로, 앞서 설명된 바와 같이, 인코더 또는 디코더는, 새로운 BV 값( $BV_{new}$ )(여기서,  $BV_{PU1}$ )이 PBV0과 같은지에 따라, 후보 BV 값 쌍(및 BV 버퍼)을 선택적으로 업데이트한다.

#### D. 평평 접근법에 따른 예시적인 BV 예측 기법.

[0183] 도 16은 예시적인 평평 접근법에 따른, 다수의 PU를 갖는 CU에 대한 BV 값을 예측하는 예시적인 기법을 나타낸 것이다. CU는, 일반적으로, 인코딩 및 디코딩하기 위한 하나 이상의 블록의 세트이고, PU들 각각은 예측 정보 및/또는 예측 처리를 신호하기 위한 CU 내의 하나 이상의 블록의 세트이다. 도 3 또는 도 5a 및 도 5b를 참조하여 기술된 것과 같은 영상 인코더 또는 비디오 인코더, 또는 다른 인코더는 기법(1600)을 수행할 수 있다. 또는, 도 4 또는 도 6을 참조하여 기술된 것과 같은 영상 디코더 또는 비디오 디코더, 또는 다른 디코더는 기법(1600)을 수행할 수 있다.

[0184] 우선, 후보 BV 값 쌍은 제1 초기 후보 BV 값( $BV_{init\_0}$ )(PBV0) 및 제2 초기 후보 BV 값( $BV_{init\_1}$ )(PBV1) - 각각, BV 버퍼 0 및 BV 버퍼 1에 저장되어 있음 - 이다.

[0185] 현재 빙처의 현재 CU의 다수의 PU 각각에 대해, 인코더 또는 디코더는 후보 BV 값 쌍 중 하나를 사용하여 PU에 대한 BV 값을 예측한다(1610). 플래그 값은 후보 BV 값 쌍 중에서의 선택을 나타낸다. 후보 BV 값 쌍이 CU의 상이한 PU에 대해 상이할 수 있다. 예를 들어, 현재 CU가 4 개의 PU를 포함할 때, 첫 번째 PU에 대해, 후보 BV 값 쌍(PBV0 및 PBV1)은, 각각,  $BV_{init\_0}$  및  $BV_{init\_1}$ 일 수 있다. 두 번째 PU에 대해, 후보 BV 값 쌍(PBV0 및 PBV1)은, 각각, 첫 번째 PU에 대한 BV( $BV_{PU0}$ ) 및  $BV_{init\_0}$ 일 수 있다. 세 번째 PU에 대해, 후보 BV 값 쌍(PBV0 및 PBV1)은, 각각, 두 번째 PU에 대한 BV( $BV_{PU1}$ ) 및  $BV_{PU0}$ 일 수 있다. 마지막으로, 네 번째 PU에 대해, 후보 BV 값 쌍은  $BV_{PU1}$  및 세 번째 PU에 대한 BV 값( $BV_{PU2}$ )일 수 있다.

[0186] 인코더 또는 디코더는 이어서 PU에 대한 예측된 BV 값을 사용하여 PU에 대한 BV 값을 처리한다(1620). 예를 들어, 인코딩 동안, 인코더는 PU에 대한 BV 값 및 예측된 BV 값을 사용하여 BV 차분 값을 결정한다. 또는, 디코딩 동안, 디코더는 예측된 BV 값에 적어도 부분적으로 기초하여 PU에 대한 BV 값을 재구성한다.

[0187] BV 버퍼가 자동으로 또는 선택적으로(새로운 BV 값( $BV_{new}$ )이 PBV0과 같은지에 따라) 업데이트될 수 있다. 예를 들어, (도 14에 도시된 바와 같이) 현재 CU가 4 개의 PU를 포함할 때, 첫 번째 PU(PU0)에 대한 BV 예측 이전에, BV 버퍼 0은  $BV_{init\_0}$ 을 저장하고, BV 버퍼 1은  $BV_{init\_1}$ 을 저장한다. 두 번째 PU(PU1)에 대한 BV 예측 이전에, BV 버퍼 0은  $BV_{PU0}$ 을 저장하고, BV 버퍼 1은  $BV_{init\_1}$ 을 저장한다. 세 번째 PU(PU2)에 대한 BV 예측 이전에, BV 버퍼 0은  $BV_{PU1}$ 을 저장하고, BV 버퍼 1은  $BV_{PU0}$ 을 저장한다. PU2에 대한 처리 후에, BV 버퍼 0은  $BV_{PU1}$ 을 유지하고, BV 버퍼 1은  $BV_{PU0}$ 을 유지한다.

[0188] 현재 CU에 대한 BV 예측 후에, BV 버퍼에 대한 업데이트가 일어날 때, 후보 BV 값 쌍은 현재 CU의  $BV_{PU1}$  및  $BV_{PU0}$ 이다. 이들은 현재 빙처의 후속 CU에 대한 BV 예측을 위한 초기 후보 BV 값일 것이다. 이와 같이, 현재 CU가 4 개의 PU를 포함할 때, 후보 BV 값 쌍은, 현재 CU의 첫 번째 및 두 번째 PU에 대한 BV 값이 현재 CU의 세 번째 및 네 번째 PU에 대한 BV 값보다 먼저 처리되었더라도, 현재 CU의 첫 번째 및 두 번째 PU에 대한 BV 값을 포함할 수 있다.

[0189] 도 17은 BV 버퍼의 선택적인 업데이트를 갖는, 예시적인 평평 접근법에 따른, CU에 대한 BV 값을 예측하는 다른

예시적인 기법을 나타낸 것이다. 도 3 또는 도 5a 및 도 5b를 참조하여 기술된 것과 같은 영상 인코더 또는 비디오 인코더, 또는 다른 인코더는 기법(1700)을 수행할 수 있다. 또는, 도 4 또는 도 6을 참조하여 기술된 것과 같은 영상 디코더 또는 비디오 디코더, 또는 다른 디코더는 기법(1700)을 수행할 수 있다.

[0190] 인코더 또는 디코더는 현재 꽂처의 다수의 CU를 처리한다. 예를 들어, 인코더는 인코딩의 일부로서 다수의 CU를 처리하고 인코딩된 데이터를 비트스트림의 일부로서 출력한다. 또는, 디코더는 디코딩의 일부로서 다수의 CU를 처리하고 재구성된 샘플 값을 출력한다. 다수의 CU의 임의의 주어진 CU는 단일의 PU 또는 다수의 PU를 포함할 수 있다.

[0191] 우선, 후보 BV 값 쌍은 제1 초기 후보 BV 값( $BV_{init\_0}$ )(PBV0) 및 제2 초기 후보 BV 값( $BV_{init\_1}$ )(PBV1) - 각각, BV 버퍼 0 및 BV 버퍼 1에 저장되어 있음 - 이다.

[0192] 다수의 CU 중 주어진 CU에 대해, 인코더 또는 디코더는 주어진 CU가 단일의 PU를 갖는지 다수의 PU를 갖는지를 검사한다(1710). 주어진 CU가 단일의 PU를 갖는 경우, 인코더 또는 디코더는 후보 BV 값 쌍(PBV0 및 PBV1)(도 17에서,  $BV_{init\_0}$  및  $BV_{init\_1}$ ) 중 하나를 사용하여 단일의 PU에 대한 BV 값을 예측하고(1720), 여기서 플래그 값은 후보 BV 값 쌍 중에서의 선택을 나타낸다. 인코더 또는 디코더는 예측된 BV 값을 사용하여 단일의 PU에 대한 BV 값을 처리한다(1730). 인코더 또는 디코더는 PU에 대한 BV 값( $BV_{PU0}$ )이 BV 버퍼 0에 저장된 BV 값( $BV_{init\_0}$ )과 같은지에 따라 BV 버퍼 0 및 BV 버퍼 1을 선택적으로 업데이트한다. 예를 들어, 인코더 또는 디코더는  $BV_{PU0}$ 과  $BV_{init\_0}$ 를 비교한다.  $BV_{PU0}$ 이 BV 버퍼 0에 저장된 BV 값( $BV_{init\_0}$ )과 상이한 경우, 인코더 또는 디코더는 BV 버퍼 0으로부터의 BV 값( $BV_{init\_0}$ )을 BV 버퍼 1에 저장하고,  $BV_{PU0}$ 을 BV 버퍼 0에 저장한다.

[0193] 처리(1730) 후에, BV 버퍼에 대한 업데이트가 일어날 때, BV 버퍼 0은 단일의 PU에 대한 BV 값( $BV_{PU0}$ )을 저장하고, BV 버퍼 1은  $BV_{init\_0}$ 을 저장한다. 인코더 또는 디코더는 꽂처의 다음 CU를 계속할지를 검사(1740)한다. 그러한 경우, 베파링된 BV 값 쌍은 다음 CU에 대한 초기 후보 BV 값( $BV_{init\_0}$  및  $BV_{init\_1}$ )으로서 취급된다.

[0194] 그렇지 않고, 주어진 CU가 다수의 PU를 가질 때, 인코더 또는 디코더는 주어진 CU의 PU에 대한 BV 예측 및 처리를 수행한다. 주어진 CU의 다수의 PU 각각에 대해, 인코더 또는 디코더는 후보 BV 값 쌍(PBV0 및 PBV1)(처음에,  $BV_{init\_0}$  및  $BV_{init\_1}$ 이지만 나중에 업데이트됨) 중 하나를 사용하여 PU에 대한 BV 값을 예측한다(1750). PU에 대한 플래그 값은 후보 BV 값 쌍 중에서의 선택을 나타낸다. 인코더 또는 디코더는 예측된 BV 값을 사용하여 PU에 대한 BV 값을 처리한다(1760). PU가 주어진 CU의 첫 번째 또는 두 번째 PU인 경우, 인코더 또는 디코더는 PU에 대한 BV 값( $BV_{PUx}$ )이 BV 버퍼 0에 저장된 BV 값(PBV0)과 같은지에 따라 BV 버퍼 0 및 BV 버퍼 1을 선택적으로 업데이트한다. 예를 들어, 인코더 또는 디코더는  $BV_{PUx}$ 와 PBV0을 비교한다.  $BV_{PUx}$ 가 PBV0과 상이한 경우, 인코더 또는 디코더는 PBV0을 BV 버퍼 1에 저장하고,  $BV_{PUx}$ 를 BV 버퍼 0에 저장한다.

[0195] 인코더 또는 디코더는 PU가 주어진 CU에서의 마지막 PU인지를 검사(1770)한다. 그렇지 않은 경우, 인코더 또는 디코더는 주어진 CU의 다음 PU에 대한 BV 예측을 계속한다(1750). 예를 들어, PU는 2Nx2N CU의 4 개의 NxN 블록, 2 개의 Nx2N 블록 또는 2 개의 2NxN 블록이다.

[0196] 주어진 CU의 마지막 PU에 대한 처리(1760) 후에, BV 버퍼에 대한 양 업데이트가 일어날 때, BV 버퍼 0은 주어진 CU의 두 번째 PU에 대한 BV 값( $BV_{PU1}$ )을 저장하고, BV 버퍼 1은 주어진 CU의 첫 번째 PU에 대한 BV 값( $BV_{PU0}$ )을 저장한다. 인코더 또는 디코더는 꽂처의 다음 CU를 계속할지를 검사(1740)한다. 그러한 경우, 베파링된 BV 값 쌍( $BV_{PU1}$  및  $BV_{PU0}$ )은 다음 CU에 대한 초기 후보 BV 값( $BV_{init\_0}$  및  $BV_{init\_1}$ )으로서 취급된다.

## IX. 인트라 블록 복사 예측 모드에 대한 인코더측 옵션.

[0198] 이 섹션은 인트라 BC(block copy) 예측에 대한 인코더측 옵션에 대한 다양한 혁신을 제시한다. 혁신들 중 일부는 BV(block vector) 추정과 블록에 대한 블록 분할 결정을 동시에 수행하는 것에 관한 것이다. 다른 혁신은 BV 추정 동안 블록들을 보다 큰 블록으로 선택적으로 병합하는 것에 관한 것이다. 또 다른 혁신은 BV 추정 동안 현재 블록의 중첩 구역 내에서 샘플 값을 추정하는 것에 관한 것이다. 일반적으로, 이 혁신은 인트라 BC 예측의 코딩 효율을 개선시킨다.

[0199] BV 추정의 일부로서, 인코더는 몇 가지 접근법 중 임의의 것을 사용할 수 있다. 인코더는, 겸색 범위에서 허용

된 모든 후보 BV 값을 평가하는, 전체 검색(full search)을 사용할 수 있다. 또는, 인코더는, 검색 범위에서 허용된 후보 BV 값 중 일부만을 평가하는, 부분 검색(partial search)을 사용할 수 있다. 예를 들어, 인코더는 현재 블록에 대한 예측된(예컨대, 하나 이상의 이웃 블록의 BV 값에 기초하여 예측된) BV 값에서 부분 검색을 시작할 수 있다. 부분 검색을 위한 시작 위치에서 후보 BV 값을 계산한 후에, 인코더는 (예컨대, 나선형 검색 패턴 또는 어떤 다른 패턴에 따라) 시작 위치로부터의 증가하는 거리에서 하나 이상의 다른 후보 BV 값을 평가할 수 있다. 주어진 후보 BV 값을 평가할 때, 인코더는 인트라 예측 영역과 현재 블록 내의 모든 샘플 값을 비교할 수 있다. 또는, 인코더는 샘플 값들의 서브셋(즉, 값이 평가되는 서브샘플(sub-sample))을 평가할 수 있다. 왜곡 비용(distortion cost)을 결정하기 위해 인트라 예측 영역과 현재 블록 사이에서 샘플 값을 비교할 때, 인코더는 평균 제곱 오차, 차 제곱의 합, 절대차의 합, 또는 어떤 다른 왜곡 척도를 계산할 수 있다. 인코더는 또한 후보 BV 값의 인코딩과 연관된 레이트 비용(rate cost)을 결정할 수 있다.

[0200]

인코딩 동안, 다양한 스테이지에서, 인코더는 다양한 옵션의 레이트 비용과 왜곡 비용이 평가되는 RDO(rate-distortion optimization)를 사용할 수 있다. 일반적으로, 옵션에 대한 레이트 왜곡 비용은  $D + \lambda R$ (또는  $R + \lambda D$ )에 의해 주어지고, 여기서  $R$ 은 레이트 비용을 인코딩된 데이터의 비트로 표현하고,  $D$ 는 평균 제곱 오차 또는 지각 왜곡 척도(perceptual distortion measure)와 같은 메트릭을 사용하여 왜곡 비용을 표현하며,  $\lambda$ 는 왜곡 비용( $D$ )과 비교하여 레이트 비용( $R$ )을 가중하는 라그랑지 승수(Lagrangian multiplier)(가중 파라미터의 예)이다. 인코더는 전형적으로 가장 낮은 레이트 왜곡 비용을 제공하는 옵션을 선택한다.

[0201]

#### A. BV 추정 동안 블록들을 보다 큰 블록으로 선택적으로 병합하기

[0202]

BV 추정 동안, 인코더는 블록들을 보다 큰 블록으로 병합할 수 있다. 이러한 방식으로, 인코더는 BV 정보를 신호하는 데 사용되는 비트 수를 감소시킬 수 있다. 상세하게는, 인트라 BC 예측이 TB 단위로 수행될 때, 인코더는 BV 추정을 CB 단위로 수행하고, 이어서 작은 CB들을 보다 큰 CB로 병합할 수 있고, TB와 그의 인트라 예측 영역 사이의 중첩을 피하기 위해 인트라 BC 예측 동안 CB를 보다 작은 TB로 분할하는 것에 의존한다.

[0203]

도 18은 BV 추정 동안 블록들을 보다 큰 블록으로 선택적으로 병합하는 기법(1800)을 나타낸 것이다. 기법(1800)은 도 3 또는 도 5a 및 도 5b를 참조하여 기술된 것과 같은 인코더에 의해 또는 다른 인코더에 의해 수행될 수 있다.

[0204]

이 기법에 따르면, 인코더가 인트라 BC 예측을 사용하여 빅처에 대한 데이터를 인코딩할 때, 제1 크기를 갖는 다수의 블록 각각에 대해, 인코더는 BV 추정을 사용하여 BV 값을 식별한다(1810). 제1 크기는 8x8, 16x16, 32x32 또는 어떤 다른 크기일 수 있다. 일반적으로, 제1 크기를 갖는 블록들 각각에 대해, BV 추정은 후보 BV 값에 대한 비용을 결정하는 것을 포함한다. 비용은 레이트 비용 및/또는 왜곡 비용을 포함할 수 있다. 식별된 BV 값은 제1 크기를 갖는 블록과 중첩하지 않는 인트라 예측 영역을 참조한다.

[0205]

인코더는 블록들 중 2 개 이상을 제1 크기보다 더 큰 제2 크기를 갖는 블록으로 선택적으로 병합한다(1820). 제2 크기는 16x16, 32x32, 64x64 또는 어떤 다른 크기일 수 있다. 예를 들어, 인코더는 제1 크기를 갖는 다수의 블록 중에서 2 개 이상의 인접 블록의 BV 값을 비교하고, 비교된 BV 값이 같으면, 2 개 이상의 인접 블록을 제2 크기를 갖는 블록으로 병합한다. 제2 크기를 갖는 블록은 똑같은 BV 값을 할당받는다. 제1 크기를 갖는 블록과 달리, BV 값은 제1 크기를 갖는 블록과 중첩하는 인트라 예측 영역을 참조할 수 있다.

[0206]

인코더는 다른 블록에 대해 기법(1800)을 반복할 수 있다. 궁극적으로, 인코더는 빅처에 대한 인코딩된 데이터를 출력한다.

[0207]

일부 예시적인 구현에서, 제1 크기를 갖는 다수의 블록 각각은 CB이고, 제2 크기를 갖는 블록도 CB이다. 그렇지만, 인트라 BC 예측이 TB 단위로 수행된다. BV 추정은 제1 크기를 갖는 다수의 블록 각각에 대해 검색 범위의 전체 검색을 사용할 수 있다. 또는, BV 추정은, 예를 들어, 블록에 대한 예측된 BV 값에서 시작하여, 어쩌면 평가된 후보 BV 값에 대한 비용에 적어도 부분적으로 기초하여 부분 검색을 끝내는, 제1 크기를 갖는 다수의 블록 각각에 대해 검색 범위의 부분 검색을 사용할 수 있다.

[0208]

도 19는 BV 추정 동안 블록들을 보다 큰 블록으로 선택적으로 병합하는 것(1900)의 장점을 나타낸 것이다. 도 19에서, 0, 1, 2 및 3으로 번호가 매겨진 4 개의 블록(연한 실선으로 표시됨)은, 각각, 대응하는 인트라 예측 영역(0', 1', 2' 및 3')을 참조하는 BV 값을 갖는다. 즉, BV 값은 블록 0에 대한 인트라 예측 영역 0'을 참조하고, 블록 1에 대한 인트라 예측 영역 1'을 참조하며, 이하 마찬가지이다. 인트라 예측 영역은 점선으로 표시되어 있다. 블록들(0, 1, 2 또는 3) 중 어느 것도 그의 대응하는 인트라 예측 영역과 중첩하지 않는다. 그렇지만, 블록들(0, 1, 2 및 3)을 포함하는 보다 큰 블록(진한 실선으로 표시됨)은 그의 대응하는 인트라 예측 영

역과 중첩한다. 인코더가 보다 큰 블록에 대한 BV 추정을 수행하는 경우, 인트라 예측 영역과 블록 간의 중첩이 금지될 때, 인코더는 도 19에 도시된 BV 값을 허용 가능 BV 값을으로서 식별할 수 없을 것이다. 이와 달리, 인코더가 보다 작은 블록(0, 1, 2 및 3)에 대해 BV 추정을 수행할 때, 인코더는 4 개의 블록 전부에 대해 동일한 BV 값을 식별하고, 인코더는 4 개의 블록을 동일한 BV 값을 갖는 보다 큰 블록으로 병합할 수 있다. 인코딩 또는 디코딩 동안, 실제의 인트라 BC 예측 동작을 위해 보다 큰 블록이 보다 작은 블록으로 분할될 수 있다. 예를 들어, 주어진 크기를 가지는 CB가 TB 단위로 인트라 BC 예측 동작을 수행하기 위해 보다 작은 TB로 분할되고, TB와 그의 인트라 예측 영역 사이에 중첩이 없다.

#### [0209] B. 동시적인 블록 베터 추정과 블록 분할 결정.

인코더는 현재 블록에 대해 BV 추정과 블록 분할 결정을 동시에 수행할 수 있다. 상세하게는, 인트라 BC 예측 동작이 TB 단위로 수행될 때, 인코더는 현재 블록에 대한 후보 BV 값과 가능한 블록 분할 결정을 동시에 평가할 수 있다.

도 20은 블록에 대해 BV 추정과 블록 분할 결정을 동시에 수행하는 일반화된 기법을 나타낸 플로우차트이다. 기법(2000)은 도 3 또는 도 5a 및 도 5b를 참조하여 기술된 것과 같은 인코더에 의해 또는 다른 인코더에 의해 수행될 수 있다.

인코더가 인트라 BC 예측을 사용하여 빙처에 대한 데이터를 인코딩할 때, 인코더는 현재 블록에 대한 데이터를 인코딩(2010)한다. 인코딩(2010)의 일부로서, 인코더는 BV 추정 동작 및 블록 분할 결정 동작을 동시에 수행한다. 일부 예시적인 구현에서, 현재 블록은 크기가 TB에 대응하는 CB이고, 인트라 BC 예측이 TB 단위로 수행된다. 예를 들어, 인코딩(2010)의 일부로서, 인코더는 현재 블록에 대한 후보 BV 값 및 가능한 블록 분할 결정을 동시에 평가하기 위해 도 21에 도시된 기법(2100)을 수행한다.

도 21은 블록에 대해 후보 BV 값과 블록 분할 결정을 동시에 평가하는 예시적인 기법을 나타낸 플로우차트이다. 기법(2100)은 도 3 또는 도 5a 및 도 5b를 참조하여 기술된 것과 같은 인코더에 의해 또는 다른 인코더에 의해 수행될 수 있다.

인코더는 BV 추정 동작을 사용하여 현재 블록에 대한 후보 BV 값을 식별한다(2110). 현재 블록은 제1 크기를 갖는다. 예를 들어, 제1 크기는 8x8, 16x16, 32x32, 64x64 또는 어떤 다른 크기이다.

인코더는 현재 블록을 분할할지를 검사(2120)한다. 예를 들어, 인코더는, 현재 블록에 대한 후보 BV 값을 사용한 인트라 BC 예측의 결과, 현재 블록과 후보 BV 값에 의해 참조되는 인트라 예측 영역 사이에 중첩이 발생하는지를 평가한다.

검사(2120)의 결과에 따라, 인코더는 현재 블록을 다수의 블록 - 각각이 제1 크기보다 더 작은 제2 크기를 가짐 - 으로 선택적으로 분할한다(2130). 예를 들어, 제2 크기는 4x4, 8x8, 16x16, 32x32 또는 어떤 다른 크기이다. 이어서(현재 블록이 분할됨), 인코더는 보다 작은 블록들 각각에 대해 기법(2100)을 반복한다(2140). 즉, 제2 크기를 가지는 블록들 각각에 대해, 인코더는 식별하는 것, 평가하는 것, 선택적으로 분할하는 것 등을 반복하고, 제2 크기를 가지는 블록이 현재 블록으로서 취급된다.

다른 한편으로, 현재 블록이 분할되지 않은 경우, 인코더는 현재 블록을 인코딩하고(2150), 현재 블록을 인코딩하는 비용을 측정한다(2160). 예를 들어, 현재 블록을 인코딩할 때, 인코더는 (a) 인트라 BC 예측을 사용하여 현재 블록의 샘플 값을 예측하고, (b) 예측된 샘플 값을 및 현재 블록의 샘플 값을 사용하여 현재 블록에 대한 잔차 값을 결정하며, (c) 선택적으로, 변환 계수를 생성하기 위해 잔차 값을 주파수 변환을 적용하고, 변환 계수를 양자화하며, (d) 선택적으로, 변환 계수를 역양자화하고, 잔차 값을 재구성하기 위해 역 주파수 변환을 적용하며, (e) 잔차 값을 예측된 샘플 값을 결합시킨다. 측정된 비용은 레이트 왜곡 비용, 레이트 비용 또는 왜곡 비용일 수 있다. 레이트 비용은 BV 값을 신호하는 비용과 블록 분할 결정 정보를 신호하는 비용 둘 다를 고려할 수 있다.

대안적으로, 인코더는 어떤 다른 방식으로 현재 블록에 대해 후보 BV 값 및 가능한 블록 분할 결정을 평가한다. 예를 들어, 인트라 예측 영역과 블록 사이의 중복을 고려하는 것에 부가하여, 인코더는 또한 블록이 이미 가장 작은 허용 가능 크기를 갖는지도 고려한다. 그러한 경우, 블록이 분할되지 않지만, 후보 BV 값이 허용되지 않는다.

도 20으로 돌아가서, 인코딩(2010)의 일부로서, 인코더는 후보 BV 값에 대한 비용을 기록할 수 있다. 인코더는 이어서 하나 이상의 다른 후보 BV 값을 평가할 수 있다. 예를 들어, 인코더는 다른 후보 BV 값에 대해 기법

(2100)을 반복한다. 궁극적으로, 인코더는 하나 이상의 BV를 선택하고, 가장 낮은 비용(예컨대, 레이트 왜곡 비용, 레이트 비용 또는 왜곡 비용)을 가져오는 현재 블록(및 현재 블록 내의 블록)에 대한 블록 분할 결정을 행한다.

[0220] 인코더는 블록에 대한 인코딩된 데이터를 출력(2020)한다. 인코더는 다른 블록에 대해 기법(2000)을 반복할 수 있다.

### C. 중첩 구역 내에서의 재구성된 샘플 값을 추정하기

[0222] 본원에 기술되는 일부 예에서, BV가 CU의 레벨에서 신호되고 CB 레벨에서 블록에 적용된다. CB와 대응하는 인트라 예측 영역 사이의 중첩이 허용되지 않는다. 본원에 기술되는 다른 예에서, 인코더는 BV 값을 보다 작은 블록에 TB 단위로 적용하지만, BV 값은 (예컨대, CU에 대한) 상위 구문 레벨에서 신호된다. 예를 들어, 32x32 CB를 갖는 CU에 대해 BV 값이 신호되지만, BV 값은 4x4 TB 또는 8x8 TB의 레벨에서 블록에 적용된다. 이것은, TB 레벨에서의 블록과 복사되는 인트라 예측 영역 사이에 여전히 중첩을 갖지 않으면서, BV 값이 보다 작은 크기를 가질 수 있게 한다.

[0223] 주어진 블록이 인트라 BC 예측 동작을 위해 보다 작은 블록으로 분할될 때, 주어진 블록은 주어진 블록의 샘플 값이 보다 작은 블록들 중 하나 이상에 대한 인트라 예측 영역의 일부일 수 있는 중첩 구역을 포함한다. 이 섹션은 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 접근법을 기술한다. 상세하게는, 이 섹션은, 블록에 대한 BV 값이 보다 작은 블록에 TB 단위로 적용될 때, 블록의 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 접근법을 기술한다.

[0224] 도 22a는 크기  $2m \times 2n$ 을 가지는 현재 블록(2210)을 나타낸 것이다. 현재 블록(2210)은 인트라 BC 동작을 위한 4 개의  $m \times n$  블록(2221, 2222, 2223, 2224)을 포함한다. 예를 들어, 현재 블록(2210)은 CU의 8x8 블록이고, 4 개의 보다 작은 블록((2221, 2222, 2223, 2224))은 4x4 TB를 갖는 4x4 블록이다. 또는, 현재 블록(2210)은 CU의 16x16 블록이고, 4 개의 보다 작은 블록((2221, 2222, 2223, 2224))은 8x8 TB를 갖는 8x8 블록이다. 인트라 BC 예측 동작이  $m \times n$  블록에 지그재그 스캔 순서(도 22a에서, 블록 0, 블록 1, 블록 2, 이어서 블록 3)로 적용될 때, 첫 번째  $m \times n$  블록(2221)에 대한 인트라 예측 영역은 현재  $2m \times 2n$  블록(2210) 밖에 있다. 그렇지만, 두 번째  $m \times n$  블록(2222)에 대해, 인트라 예측 영역은 첫 번째  $m \times n$  블록(2221)과 부분적으로 또는 완전히 중첩할 수 있다. 세 번째  $m \times n$  블록(2223)에 대해, 인트라 예측 영역은 첫 번째  $m \times n$  블록(2221) 또는 두 번째  $m \times n$  블록(2222)과 부분적으로 또는 완전히 중첩할 수 있다. 그리고 네 번째  $m \times n$  블록(2224)에 대해, 인트라 예측 영역은 현재 블록(2210)의 처음 3 개의  $m \times n$  블록(2221, 2222, 2223)과 부분적으로 또는 완전히 중첩할 수 있다.

[0225] 도 22b는 현재 블록(2210)의 중첩 구역(2230)을 나타내고 있다. 중첩 구역(2230)은 처음 3 개의  $m \times n$  블록(2221, 2222, 2223)을 지그재그 스캔 순서로 커버하는데, 그 이유는 이 3 개의  $m \times n$  블록에서의 샘플 값이 현재 블록(2210)의 적어도 하나의  $m \times n$  블록에 대한 인트라 예측 영역의 일부일 수 있기 때문이다. 다른 크기의 보다 작은 블록(예컨대, 16x16 블록 또는 32x32 블록의 4x4 블록)에 대해, 중첩 구역은 현재 블록을 훨씬 더 많이 커버할 수 있을 것이다.

[0226] BV 추정 동안, 인코더가 인트라 BC 예측 동작을 위해 보다 작은 블록으로 분할되는 현재 블록에 대한 후보 BV 값을 평가할 때, 인코더가 각자의 보다 작은 블록에 걸쳐 지그재그 순서로 후보 BV 값을 적용할 때 인코더는 중첩 구역 내에서의 실제의 재구성된 샘플 값을 계산할 수 있다. 이것은 계산 집중적일 수 있는데, 그 이유는 후보 BV 값에 따라 보다 작은 블록마다 적용되는 인코딩 동작 및 재구성 동작을 수반하기 때문이다. 그 대신에, 인코더는 중첩 구역 내에서의 재구성된 샘플 값을 추정할 수 있다. 상세하게는, 인트라 BC 예측 동작이 보다 작은 블록에 대해 TB 단위로 수행될 때, 인코더는 다수의 보다 작은 블록을 포함하는 현재 블록의 중첩 구역 내에서의 재구성된 샘플 값을 추정할 수 있다.

[0227] 도 23은 인코더가 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 예시적인 BV 추정 기법을 나타낸 플로우차트이다. 기법(2300)은 도 3 또는 도 5a 및 도 5b를 참조하여 기술된 것과 같은 인코더에 의해 또는 다른 인코더에 의해 수행될 수 있다.

[0228] 꼭지의 현재 블록에 대해, 인코더는 현재 블록의 중첩 구역 내에서의 재구성된 샘플 값을 추정한다(2310). 현재 블록은 다수의 보다 작은 블록을 갖는다. 예를 들어, 현재 블록은 제1 크기(예컨대, 32x32, 16x16 또는 8x8)를 가지며, 다수의 블록 각각은 제1 크기보다 더 작은 제2 크기(예컨대, 16x16, 8x8 또는 4x4)를 갖는다. 현재 블록은 CU의 블록일 수 있고, 이 경우에 다수의 보다 작은 블록 각각은 TU의 TB의 레벨에 있을 수 있다. 중첩 구

역은 다수의 보다 작은 블록 중 적어도 하나에 대한 잠재적인 인트라 예측 영역(intra-prediction region)에 있는 현재 블록의 일부를 커버한다.

[0229] 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 제1 접근법에 따르면, 인코더는 대응하는 원래의 샘플 값을, 각각, 중첩 구역 내에서의 재구성된 샘플 값으로서 사용한다. 현재 블록 밖에서, BV 추정은 꽉쳐 내의 원래의 샘플 값을 사용할 수 있다. 대안적으로, 현재 블록 밖에서, BV 추정은 꽉쳐 내에서의 재구성된 샘플 값(즉, 현재 블록 밖에 있는 재구성된 샘플 값; 현재 블록의 중첩 구역 내에서의 원래의 샘플 값)을 사용할 수 있다. 이 경우에, 실제의 재구성된 샘플 값이 BV 추정 동안 이용 가능하지 않을 때 원래의 샘플 값이 인트라 예측 영역에서 사용될 수 있지만, 그렇지 않은 경우, 실제의 재구성된 샘플 값이 사용된다.

[0230] 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 제2 접근법에 따르면, 인코더는 중첩 구역 내에서의 원래의 샘플 값을 처리한다. 상세하게는, 인코더는 중첩 구역 내에서의 원래의 샘플에 주파수 변환, 양자화, 역양자화 및 역 주파수 변환을 적용하여, 중첩 구역 내에서의 재구성된 샘플 값의 근사치를 산출한다. 양자화 및 역양자화는 현재 블록 주위의 이웃에서 적용된(또는 적용될 것으로 예상되는) 양자화 파라미터를 사용할 수 있다.

[0231] 중첩 구역 내에서의 재구성된 샘플 값을 추정하는 제3 접근법에 따르면, 인코더는 중첩 구역 내에서의 재구성된 샘플 값을 추정할 때 인트라 BC 예측 동작을 사용한다. 인코더는 적어도 하나의 BV 예측자(꽉쳐에서의 이전 블록(예컨대, 이웃 블록)에 대한 BV 값 또는 기본 BV 예측자 등)를 사용하여 중첩 구역 내에서의 샘플 값을 예측한다. 인코더는 중첩 구역 내에서의 예측된 샘플 값과 대응하는 원래의 샘플 값 사이의 차이에 기초하여 잔차 값을 결정한다. 잔차 값에 대해, 인코더는 주파수 변환, 양자화, 역양자화 및 역 주파수 변환을 적용한다. 양자화 및 역양자화는 현재 블록 주위의 이웃에서 적용된(또는 적용될 것으로 예상되는) 양자화 파라미터를 사용할 수 있다. 인코더는 처리의 결과(즉, 재구성된 잔차 값)를 예측된 샘플 값과 결합시켜, 중첩 구역 내에서의 재구성된 잔차 값의 근사치를 산출한다.

[0232] 제3 접근법의 하나의 변형에서, 인코더는 적어도 하나의 BV 예측자(꽉쳐에서의 이전 블록(예컨대, 이웃 블록)에 대한 BV 값 또는 기본 BV 예측자 등)를 사용하여 중첩 구역 내에서의 샘플 값을 예측하지만, 잔차 값의 처리를 생략한다. 중첩 구역에서의 추정된 재구성된 샘플 값이 예측된 샘플 값이다.

[0233] 제3 접근법의 다른 변형에서, 인코더는 중첩 구역에 대한 잔차 값이 상당할 것인지(예컨대, 변환, 양자화 등 후에 영이 아닌 변환 계수값을 가지는지)를 추정할 수 있다. 예를 들어, 인코더는 잔차 값이 작은 크기를 가지는지를 개별적으로 또는 전체적으로 평가하거나, 잔차 값에 대한 어떤 다른 조건을 검사할 수 있다. 잔차 값이 상당하지 않은 경우, 인코더는 잔차 처리를 생략하고, 단순히 예측된 샘플 값을 중첩 구역에서의 추정된 재구성된 샘플 값으로서 사용할 수 있다. 그렇지 않은 경우(잔차 값이 상당한 경우), 인코더는 잔차 값을 인코딩하고 이어서 재구성하며, 이어서 이를 예측된 샘플 값과 결합시켜 앞서 기술된 중첩 구역에서의 추정된 재구성된 샘플 값을 생성할 수 있다.

[0234] 인코더는 현재 블록에 대한 BV 값을 결정하기 위해 BV 추정을 수행한다(2320). BV 추정은 현재 블록의 중첩 구역 내에서의 추정된 재구성된 샘플 값을 중 적어도 일부를 사용한다. BV 추정은 또한 꽉쳐 내에서 현재 블록 밖에 있는 재구성된 샘플 값(또는 하나의 접근법에서, 원래의 샘플 값)을 사용한다.

[0235] BV 추정은, 현재 블록에 대한 하나 이상의 후보 BV 값 각각에 대해, 후보 BV 값에 따라 보다 작은 블록 각각과 후보 인트라 예측 영역 사이의 블록 정합 동작을 수행하는 것을 포함할 수 있다. 예를 들어, 4 개의 보다 작은 블록을 갖는 현재 블록에 대한 주어진 후보 BV 값에 대해, 인코더는 주어진 후보 BV 값에 따라 4 개의 보다 작은 블록과 그의 대응하는 후보 인트라 예측 영역 사이의 블록 정합 동작을 수행한다.

[0236] 대안적으로, BV 추정은 (1) 다수의 보다 작은 블록 각각에 대해, 보다 작은 블록에 대한 BV 값을 결정하는 것, 그리고 이어서 (2) 다수의 보다 작은 블록에 대한 BV 값들을 현재 블록에 대한 BV 값으로 합성하는 것을 포함할 수 있다. 예를 들어, 4 개의 보다 작은 블록을 갖는 현재 블록에 대해, 인코더는 첫 번째 보다 작은 블록에 대한 BV 값, 두 번째 보다 작은 블록에 대한 BV 값, 세 번째 보다 작은 블록에 대한 BV 값 및 네 번째 보다 작은 블록에 대한 BV 값을, 연속적으로, 결정한다. 첫 번째 보다 작은 블록 후에, BV 추정은 현재 블록의 중첩 구역에서의 인트라 예측 영역의 평가를 포함할 수 있다. 인코더는 이어서 4 개의 보다 작은 블록에 대한 BV 값을 현재 블록에 대한 BV 값으로 합성한다. 예를 들어, 인코더는 (a) 다수의 보다 작은 블록에 대한 BV 값을 평균하거나, (b) 다수의 보다 작은 블록에 대한 BV 값을 중간값을 계산하거나, (c) 다수의 보다 작은 블록에 대한 BV 값을 주위의 이웃에서 현재 블록에 대한 BV 값을 식별하거나, (d) 다수의 보다 작은 블록에 대한 BV 값을 주위의 이웃에서 후보 BV 값에 대한 RDO를 사용하여 현재 블록에 대한 BV 값을 식별하거나, (e) 어떤 다른 방식

으로 다수의 보다 작은 블록에 대한 BV 값들을 현재 블록에 대한 BV 값으로 합성할 수 있다.

[0237] 현재 블록에 대한 BV 값이 BV 추정에 의해 식별된 후에, 인코더는 현재 블록에 대한 BV 값을 사용하여 현재 블록에 대한 인트라 BC 예측을 수행할 수 있다. 그렇게 함에 있어서, 인코더는 다수의 보다 작은 블록에 대해 BC 동작을 블록 단위로 수행한다. 이와 같이, 인코더는 현재 블록의 레벨에서 BV 추정을 수행할 수 있지만, 보다 작은 블록의 레벨에서 BV 보상을 수행할 수 있다. 예를 들어, 인코더는 8x8 CU의 8x8 블록에 대해 BV 추정을 수행할 수 있지만, 4x4 블록에 대해 지그재그 스캔 순서로 TB 단위로 인트라 BC 동작(및 재구성 동작)을 수행할 수 있다. 4x4 블록은 (8x8 CU의) 동일한 BV 값을 공유하고, 인코더는 연속적인 4x4 잔차 값 블록에 대해 4x4 변환, 양자화, 역양자화, 4x4 역변환 등을 수행한다. (TB 0에 대한) 첫 번째 4x4 블록이 재구성되고, 이어서 (TB 1에 대한) 두 번째 4x4 블록에 대한 인트라 예측 영역에서 사용될 수 있다. 두 번째 4x4 블록이 재구성되고, 이어서 (TB 2에 대한) 세 번째 4x4 블록에 대한 인트라 예측 영역에서 사용될 수 있다. (TB 3에 대한) 네 번째 4x4 블록에 대해, 처음 3 개의 4x4 블록 중 임의의 것이 인트라 예측 영역에서 사용될 수 있다.

[0238] 디코더측에서, 앞서 기술된 바와 같이 디코더는 BV 값의 결과로서 인트라 예측 영역과 현재 블록 사이에 중첩이 생기는지를 검사할 수 있다. 그러한 경우, (TB 레벨에서의) 블록이 (보다 작은 TB의 레벨에서의) 보다 작은 블록으로 분할되고, 이어서 4 개의 보다 작은 블록이 (인트라 BC 예측 및 재구성 동작을 사용하여) 지그재그 순서로 연속적으로 디코딩된다. 이와 같이, 보다 작은 블록들 중 하나에 대한 재구성된 샘플 값이 지그재그 순서에서 다음의 보다 작은 블록에 대해 인트라 예측 영역에서 사용 가능하다. TU의 변환 블록 플래그가 0인 경우, 인트라 BC 예측이 CU 레벨에서만 적용되고, 크로마 역상관(chroma de-correlation)이 디스에이블된다.

#### D. 대안 및 변형.

[0240] 전술한 예들 중 다수가 루마 블록에 대한 BV 값 및 인트라 BC 예측 동작을 기술하고 있다. YUV 4:4:4 포맷의 비디오에 대해, 크로마 성분은 루마 성분과 동일한 분해능을 갖는다. 현재 루마 블록에 대해 사용되는 BV 값이 대응하는 크로마 블록에 대해 사용될 수 있다. 현재 루마 블록이 인트라 BC 예측 동작을 위해 보다 작은 루마 블록으로 분할될 때, 대응하는 크로마 블록도 인트라 BC 예측 동작을 위해 동일한 방식으로 보다 작은 크로마 블록으로 분할된다.

[0241] YUV 4:2:0 포맷의 비디오에 대해, 크로마 성분은 수평으로 2배만큼 그리고 수직으로 2배만큼 다운샘플링된다. 예를 들어, 현재 루마 블록이 16x16의 크기를 갖는 경우, 대응하는 크로마 블록은 8x8의 크기를 갖는다. 현재 루마 블록에 대해 사용되는 BV 값이, 적절한 스케일링 및 반올림(및/또는 벼름) 동작 후에, 대응하는 크로마 블록에 대해 사용될 수 있다. 일반적으로, 현재 루마 블록이 인트라 BC 예측 동작을 위해 보다 작은 루마 블록으로 분할될 때, 대응하는 크로마 블록도 인트라 BC 예측 동작을 위해 동일한 방식으로 보다 작은 크로마 블록으로 분할된다.

[0242] 그렇지만, 일부 구현에서, 이것이 항상 그렇지만은 않다. 일부 구현에서, YUV 4:2:0 포맷의 비디오의 루마 TB가 최소 변환 크기를 갖는 경우, 그 최소 변환 크기를 가지는 크로마 TB가 그 루마 TB 및 동일한 CB에 대한 그의 3 개의 이웃하는 루마 TB에 대한 크로마 대응물에 대해 사용된다. 예를 들어, 현재 루마 TB가 4x4의 최소 크기를 갖는 경우(그리고 8x8 루마 CB와 연관되어 있는 경우), 대응하는 크로마 CB는 4x4의 크기를 갖는다. 각각의 4x4 크로마 CB는 단일의 4x4 크로마 TB를 갖는다. 크기 2x2를 갖는 크로마 TB는, YUV 4:2:0 포맷에 대한 다운샘플링에 의해 예상될 수 있는 바와 같이, 지원되지 않는다. 이러한 구현에서, 보다 작은 루마 블록이 최소 변환 크기를 가지는 경우, 인트라 BC 예측 동작을 위해 현재 루마 블록을 보다 작은 루마 블록으로 분할하는 것이 지원되지 않는데, 그 이유는, 대응하는 크로마 블록에 적용될 때, 분할의 결과로서 최소 변환 크기보다 더 작은 크로마 블록이 생길 것이기 때문이다. 이러한 구현에서, 인트라 BC 예측 동작을 위해 현재 루마 블록(및 대응하는 크로마 블록)을 보다 작은 블록으로 분할하는 것이 보다 큰 크기에 대해 여전히 허용된다.

[0243] YUV 4:2:2 포맷의 비디오에 대해, 크로마 성분은 수평으로 2배만큼 다운샘플링된다. 예를 들어, 현재 루마 블록이 16x16의 크기를 갖는 경우, 대응하는 크로마 블록은 8x16의 크기를 갖는다. 현재 루마 블록에 대해 사용되는 BV 값이, 적절한 스케일링 및 반올림(및/또는 벼름) 동작 후에, 대응하는 크로마 블록에 대해 사용될 수 있다. 일반적으로, 현재 루마 블록이 인트라 BC 예측 동작을 위해 보다 작은 루마 블록으로 분할될 때, 대응하는 크로마 블록도 인트라 BC 예측 동작을 위해 동일한 방식으로 보다 작은 크로마 블록으로 분할될 수 있다.

[0244] 그렇지만, 일부 구현에서, 이것이 항상 그렇지만은 않다. 일부 구현에서, YUV 4:2:2 포맷의 비디오의 루마 TB가 최소 변환 크기를 갖는 경우, 그 최소 변환 크기를 가지는 크로마 TB가 그 루마 TB 및 동일한 CB에서의 이웃하는 루마 TB에 대한 크로마 대응물에 대해 사용된다. 예를 들어, 현재 루마 TB가 4x4의 최소 크기를 갖는 경

우(그리고 8x8 CB와 연관되어 있는 경우), 대응하는 4x8 크로마 CB 각각은 2 개의 4x4 크로마 TB를 갖는다. 크기 2x4를 갖는 크로마 TB는, YUV 4:2:2 포맷에 대한 다운샘플링에 의해 예상될 수 있는 바와 같이, 지원되지 않는다. 그 대신에, 4x8 크로마 CB의 4x4 크로마 TB 각각은 2 개의 2x4 크로마 구역에 대한 잔차 값을 포함한다. 이러한 구현에서, 앞서 기술된 바와 같이 인트라 BC 예측 동작을 위해 현재 루마 블록을 보다 작은 루마 블록으로 분할하는 것(앞서 기술된 바와 같이, 그에 대응하여 인트라 BC 예측 동작을 위해 크로마 블록을 보다 작은 크로마 블록으로 분할하는 것)이 최소 변환 크기보다 더 큰 블록 크기에 대해 허용된다. 보다 작은 블록이 최소 변환 크기를 갖는 경우, 잔차 값의 크로마 블록이 도 24에 도시된 바와 같이 재배열되는 경우 인트라 BC 예측 동작을 위해 현재 루마 블록을 보다 작은 루마 블록으로 분할하는 것이 지원된다.

[0245] 도 24는 YUV 4:2:2 포맷의 비디오에 대한 4x4의 최소 변환 크기를 갖는 블록을 나타내고 있다. 8x8 루마 블록(2410)은 0부터 3까지 번호가 부여된 4 개의 4x4 루마 블록(2421, 2422, 2423, 2424)을 포함한다. 대응하는 4x8 크로마 블록(2430)은 수평으로 2배 만큼 다운샘플링되는 대응하는 크로마 성분을 포함한다. 2x4의 변환 크기가 지원되지 않는다.

[0246] HEVC 표준의 현재 초안에 따르면, 4x8 크로마 블록에 대해, 2x4 블록 0과 2x4 블록 1이 잔차 코딩 및 디코딩을 위해 4x4 변환 크기를 갖는 4x4 크로마 TB로서 함께 처리되고, 2x4 블록 2와 2x4 블록 3이 잔차 코딩 및 디코딩을 위해 4x4 변환 크기를 갖는 4x4 크로마 TB로서 함께 처리된다. 그 결과, 2x4 블록 1이 재구성될 때까지 2x4 블록 0이 재구성되지 않을 것이기 때문에, 2x4 블록 1에 대한 인트라 BC 예측 동작은 2x4 블록 0에서의 재구성된 샘플 값을 참조할 수 없다. 이와 유사하게, 2x4 블록 3에 대한 인트라 BC 예측 동작은 2x4 블록 2에서의 재구성된 샘플 값을 참조할 수 없다.

[0247] 도 24에 도시된 바와 같이, 4x8 크로마 블록(2440)의 2x4 블록이 잔차 코딩 및 디코딩을 위해 재배열될 수 있다. 이 경우에, 2x4 블록 0과 2x4 블록 2가 잔차 코딩 및 디코딩을 위해 4x4 변환 크기를 갖는 4x4 크로마 TB로서 함께 처리되고, 2x4 블록 1과 2x4 블록 3이 잔차 코딩 및 디코딩을 위해 4x4 변환 크기를 갖는 4x4 크로마 TB로서 함께 처리된다. 4x8 크로마 블록(2430)의 "재배열되지 않은" 2x4 블록에서 인트라 BC 예측 동작이 여전히 행해진다. 4x8 크로마 블록이 디코딩될 때, 2x4 블록 0과 2x4 블록 2에 대한 인트라 BC 예측이 먼저 행해질 수 있고, 이어서 2x4 블록 0과 2x4 블록 2에 대한 잔차 디코딩 및 2x4 블록 0과 2x4 블록 2에 대한 샘플 값을 재구성이 행해질 수 있다. 이어서, 2x4 블록 1과 2x4 블록 3이 재구성된다. 이와 같이, 2x4 블록 1이 재구성되기 전에 2x4 블록 0이 재구성될 수 있고, 따라서 2x4 블록 1에 대한 인트라 BC 예측 동작은 2x4 블록 0에서의 재구성된 샘플 값을 참조할 수 있다. 이와 유사하게, 2x4 블록 3에 대한 인트라 BC 예측 동작은 2x4 블록 2에서의 재구성된 샘플 값을 참조할 수 있다. 잔차 값의 2x4 크로마 블록의 이러한 재배열이 인코더에서 그리고 디코더에서 일어난다.

[0248] 4x8 크로마 블록(2440)이 잔차 코딩 및 디코딩을 위한 재배열된 2x4 블록을 가지는 경우, 회색으로 도시된 바와 같이, 루마 블록 및 대응하는 크로마 블록의 좌반부에 대해 중첩 구역이 지원된다. BV 추정 동안, 인코더는(앞서 기술된 바와 같이) 중첩 구역에서의 재구성된 샘플 값을 추정하고 중첩 구역을 부분적으로 또는 완전히 커버하는 인트라 예측 영역을 참조하는 후보 BV 값을 평가할 수 있다.

[0249] 이 접근법에서, 영이 아닌 수평 BV 성분을 갖는 BV 값이 허용되지만, BV 값이 영 값 수직 BV 성분을 갖는다. 이것은, 예를 들어, 전형적으로 샘플 값에서 수평 연속성(horizontal continuity)을 나타내는 화면 포착 콘텐츠에 대해 적절하다.

[0250] 예시적인 구현에서, 인트라 BC 예측은 루마 블록에 대해 그리고 크로마 블록에 대해 정수 샘플 정밀도(integer-sample precision)를 갖는 BV 값을 사용한다. BV 값에 대해 소수 변위(fractional displacement)(및 재구성된 샘플 값들 사이의 소수 보간(fractional interpolation))가 사용되지 않는다. 픽처에 대한 크로마 데이터가 픽처에 대한 루마 데이터에 비해 감소된 분해능을 가질 때(예컨대, 포맷이 YUV 4:2:0 포맷 또는 YUV 4:2:2 포맷일 때), 루마 블록으로부터의 BV 값이, 대응하는 크로마 블록에 적용되기 전에, 크로마 분해능에서의 차이에 대해 조절하기 위해 스케일링 다운될 수 있지만, 정수 샘플 정밀도를 갖는 값으로 반올림 및/또는 버림된다.

[0251] 대안적으로, BV 값은 루마 블록에 대해 정수 샘플 정밀도를 갖지만, 대응하는 크로마 블록에 대해 소수 샘플 정밀도(fractional-sample precision)를 가질 수 있다. 상세하게는, 픽처에 대한 크로마 데이터가 픽처에 대한 루마 데이터에 비해 감소된 분해능을 가질 때, 루마 블록으로부터의 BV 값이, 대응하는 크로마 블록에 적용되기 전에, 크로마 분해능에서의 차이에 대해 조절하기 위해 스케일링 다운될 수 있다. YUV 4:2:0 포맷에 대해, BV 값의 수직 성분 및 수평 성분이 2로 나누어지고, 버림되거나 반올림되어, 1/2 샘플 정밀도(half-sample precision)와 같은 정수 샘플 정밀도 또는 소수 샘플 정밀도 중 어느 하나를 갖는 값을 산출할 수 있다. 예를

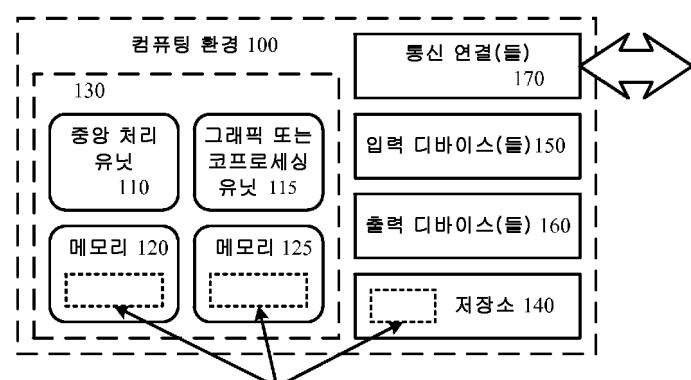
들어, 루마 블록에 대한 3의 BVy 성분 값이 대응하는 크로마 블록에 대해 1.5의 BVy 성분 값으로 스케일링된다. YUV 4:2:2 포맷에 대해, BV 값의 수평 성분이 2로 나누어지고, 버림되거나 반올림되어, 1/2 샘플 위치 정밀도 (half-sample position precision)와 같은 정수 정밀도 또는 소수 샘플 정밀도 중 어느 하나를 갖는 값을 산출할 수 있다. BV 값이 소수 샘플 정밀도를 가질 때, 재구성된 샘플 값들 사이의 보간은 이중 선형 보간 (bilinear interpolation), 이중 큐빅 보간 (bicubic interpolation) 또는 다른 형태의 보간을 사용할 수 있다.

[0252] 대안적으로, BV 값은 루마 블록 및 대응하는 크로마 블록에 대한 소수 샘플 예측 (fractional-sample prediction)을 가질 수 있다. 다시 말하지만, BV 값이 소수 샘플 정밀도를 가질 때, 재구성된 샘플 값들 사이의 보간은 이중 선형 보간, 이중 큐빅 보간 또는 다른 형태의 보간을 사용할 수 있다.

[0253] 개시된 발명의 원리가 적용될 수 있는 많은 가능한 실시예를 바탕으로, 예시된 실시예가 본 발명의 바람직한 예에 불과하고 본 발명의 범주를 제한하는 것으로서 해석되어서는 안된다는 것을 잘 알 것이다. 오히려, 본 발명의 범주는 이하의 청구범위에 의해 한정된다. 따라서, 이 청구범위의 범주 내에 속하는 모든 것을 본 발명으로서 청구한다.

## 도면

### 도면1



블록 벡터 예측 및/또는 인트라 블록 복사 예측의 인코더측 옵션에 대한 하나 이상의 혁신을 구현하는 소프트웨어 (180)

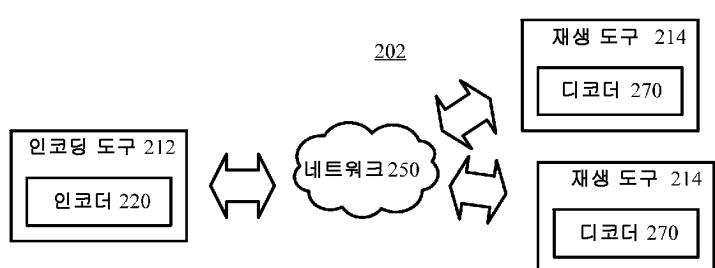
### 도면2a

201

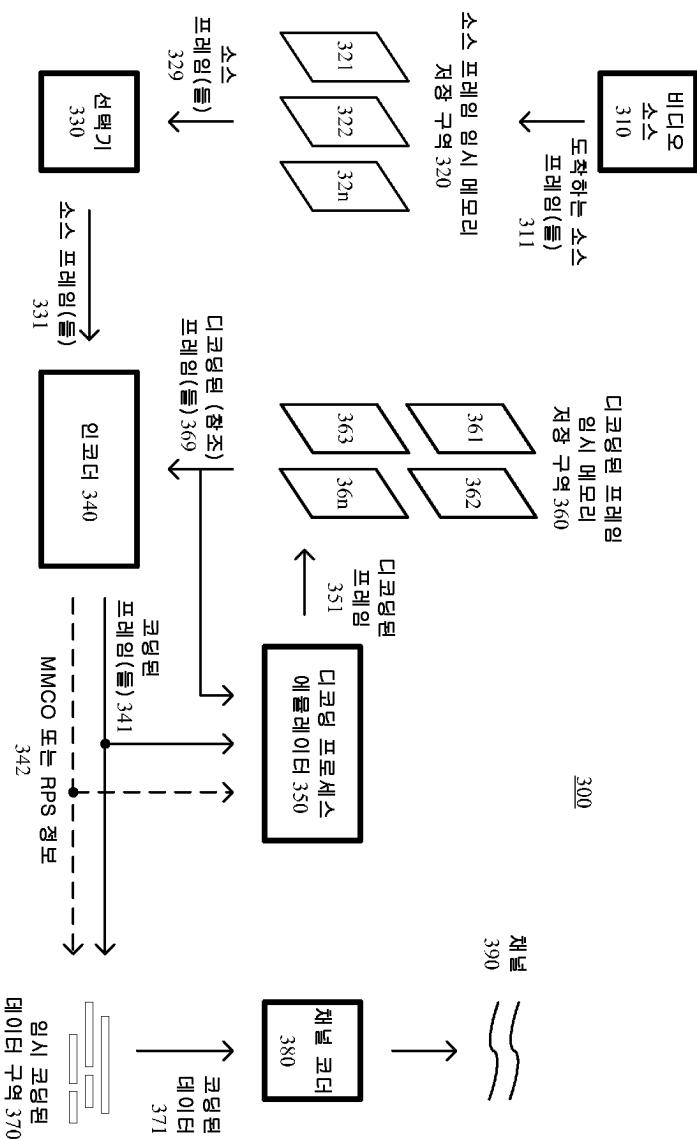


### 도면2b

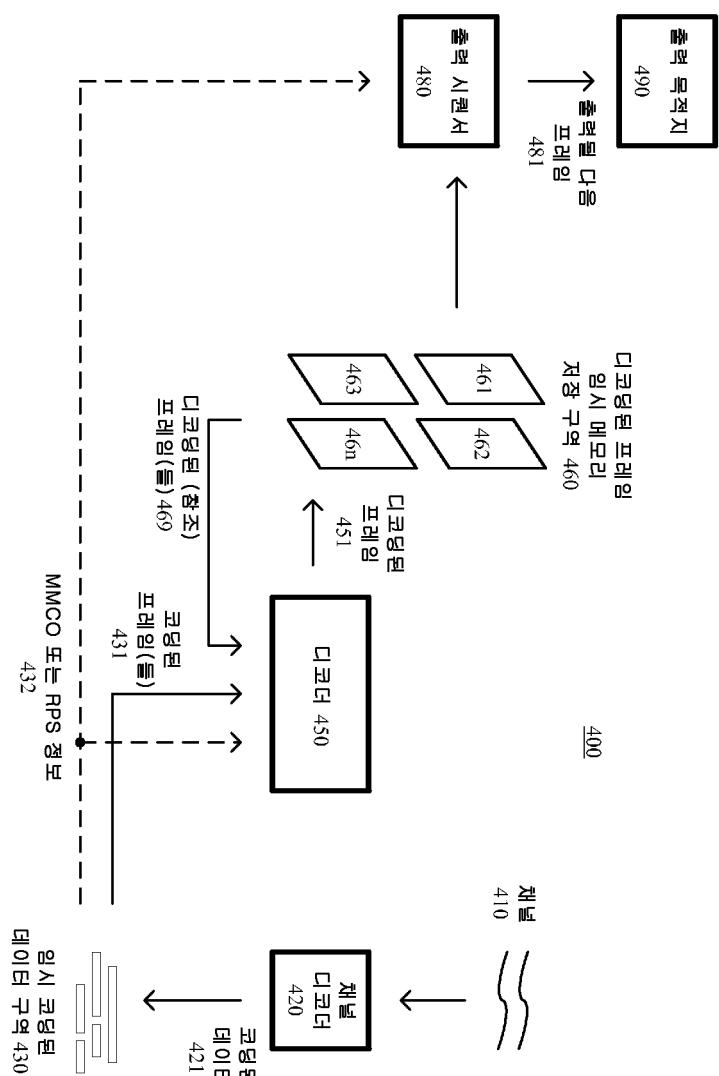
202



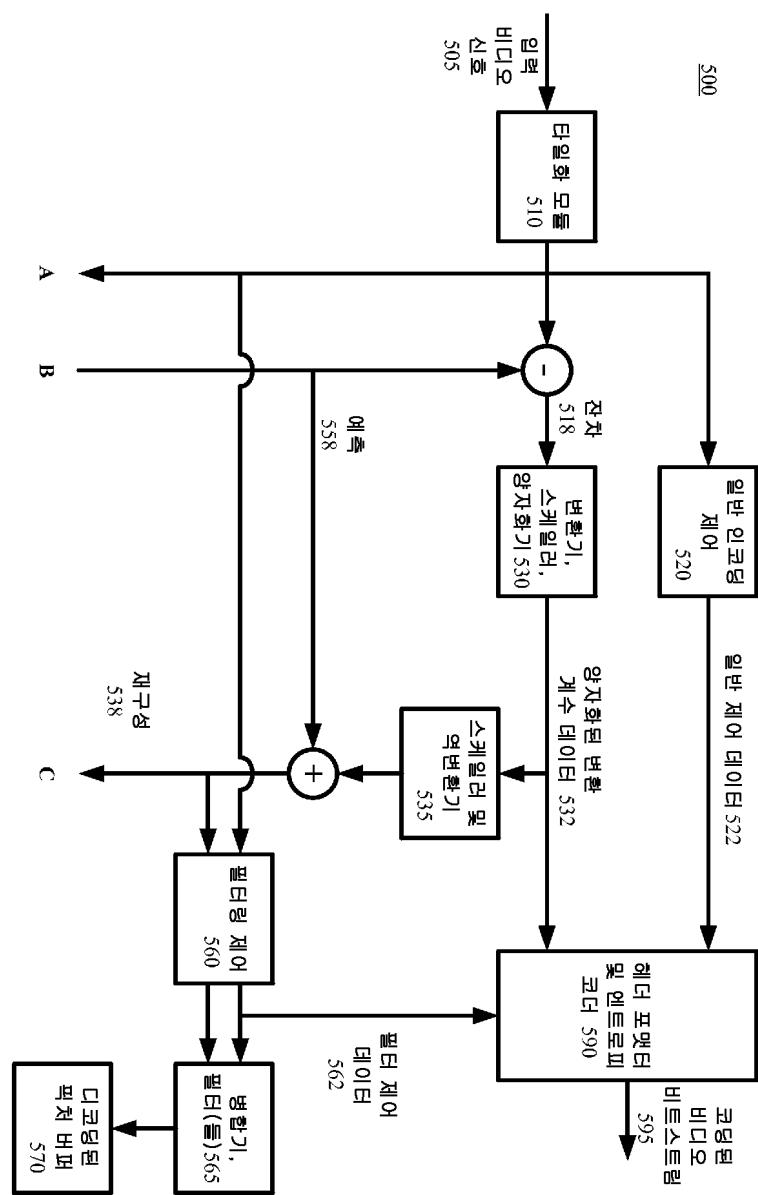
### 도면3



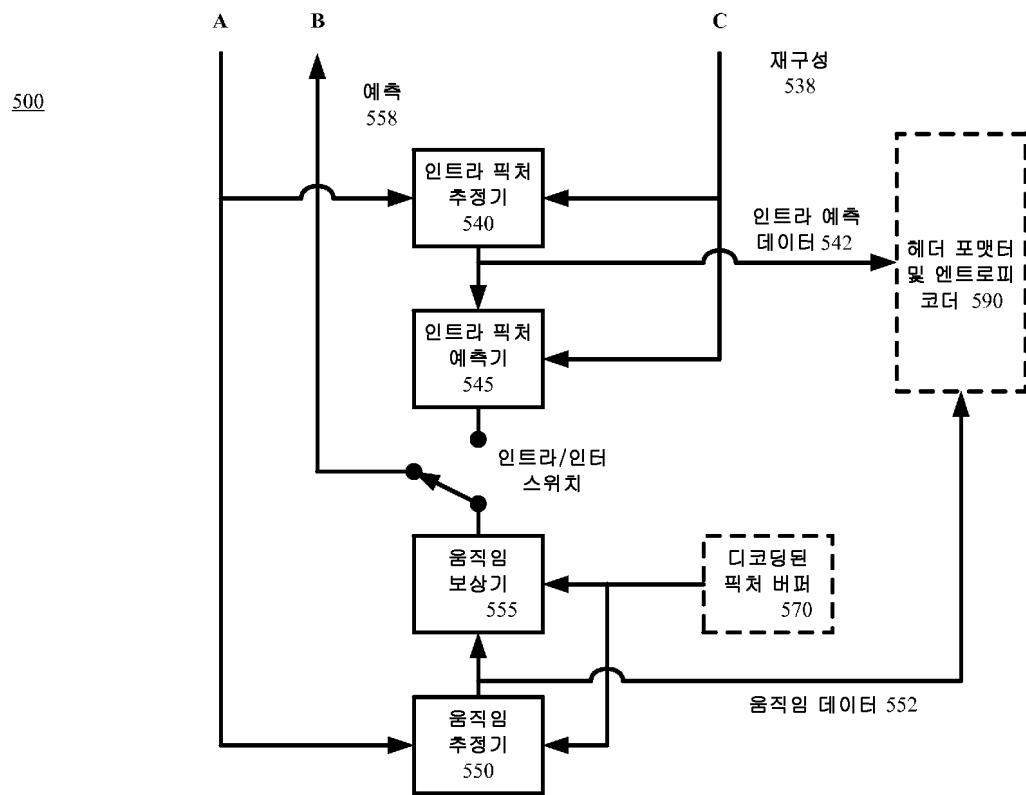
도면4



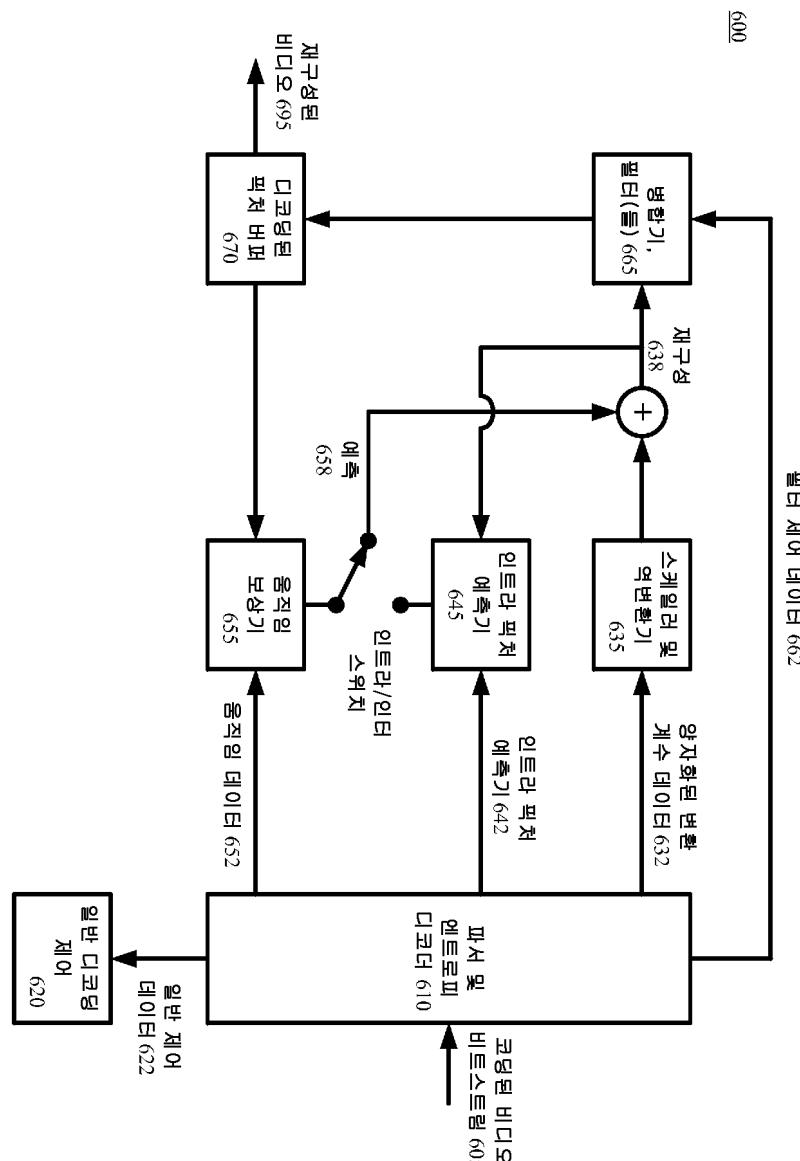
도면 5a



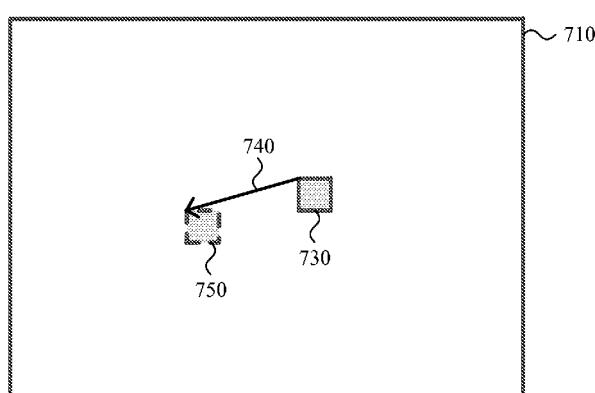
## 도면5b



도면6

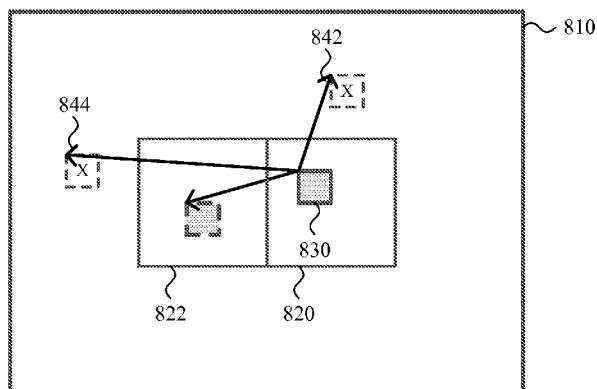


도면7



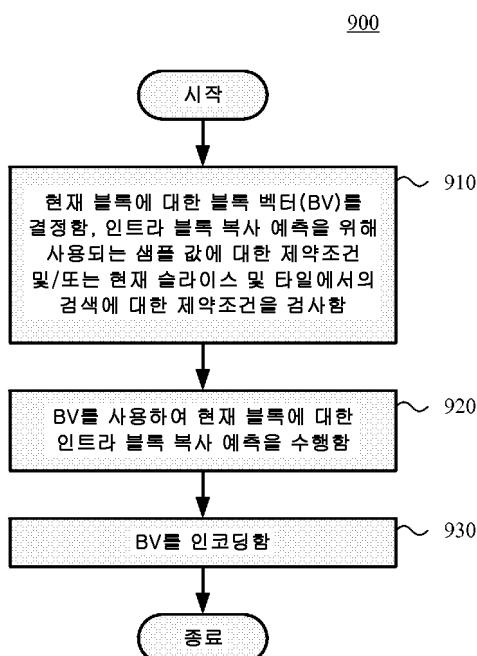
현재 프레임(710)에서 영역(750)까지의 변화를 나타내는, 현재 프레임(710)에 대한 블록(730)에 대한 블록(740)

## 도면8

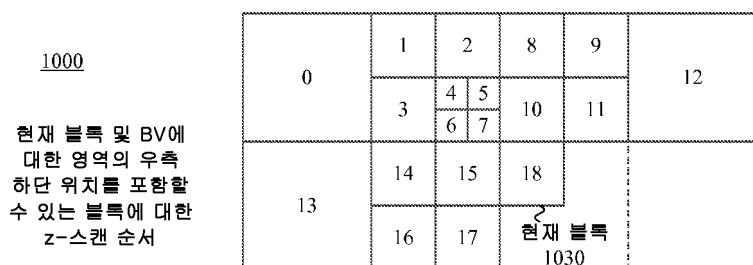


현재 프레임(810)의 현재 블록(830)에 대한 검색 범위 밖에 있는 영역까지의 범위를 나타내는 후보 블록 벡터(842, 844)

## 도면9



## 도면10



## 도면11

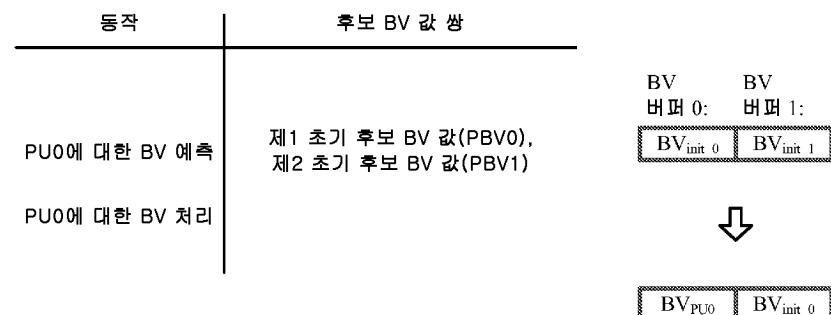
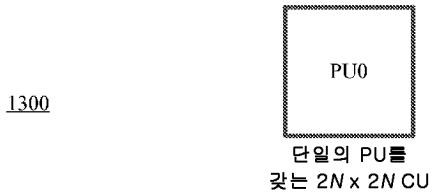


## 도면12

1200

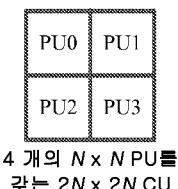
인덱스 값:	BV 예측자:
<b>idx 0 :</b>	제1 후보 BV 값 ( $BV_{init\_0}$ )
<b>idx 1 :</b>	제2 후보 BV 값 ( $BV_{init\_1}$ )

## 도면13



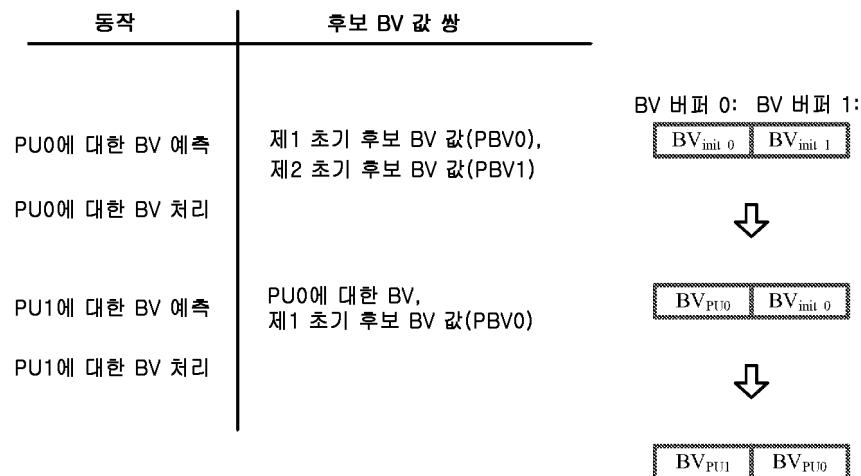
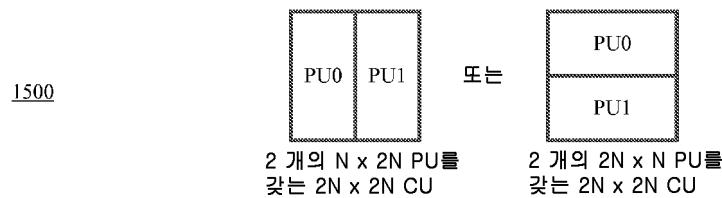
## 도면14

1400

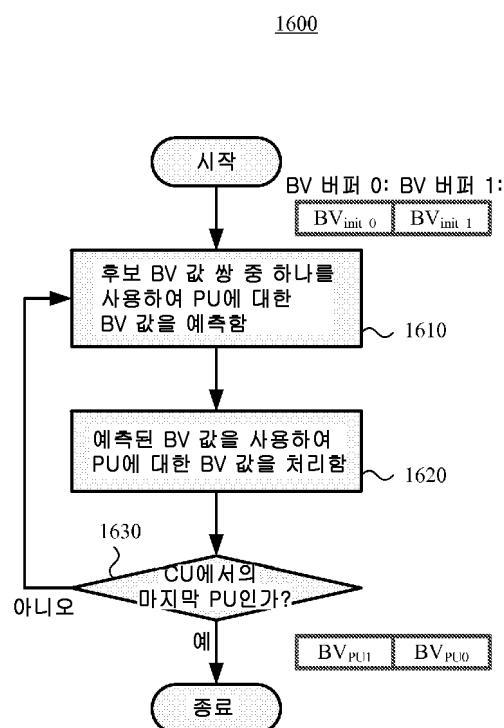


동작	후보 BV 값 쌍
PU0에 대한 BV 예측	제1 초기 후보 BV 값(PBV0), 제2 초기 후보 BV 값(PBV1)
PU0에 대한 BV 처리	 BV <sub>init 0</sub> BV <sub>init 1</sub>
PU1에 대한 BV 예측	PU0에 대한 BV, 제1 초기 후보 BV 값(PBV1)
PU1에 대한 BV 처리	 BV <sub>PU0</sub> BV <sub>init 0</sub>
PU2에 대한 BV 예측	PU1에 대한 BV, PU0에 대한 BV
PU2에 대한 BV 처리	 BV <sub>PU1</sub> BV <sub>PU0</sub>
PU3에 대한 BV 예측	PU1에 대한 BV, PU2에 대한 BV
PU3에 대한 BV 처리	 BV <sub>PU1</sub> BV <sub>PU0</sub>

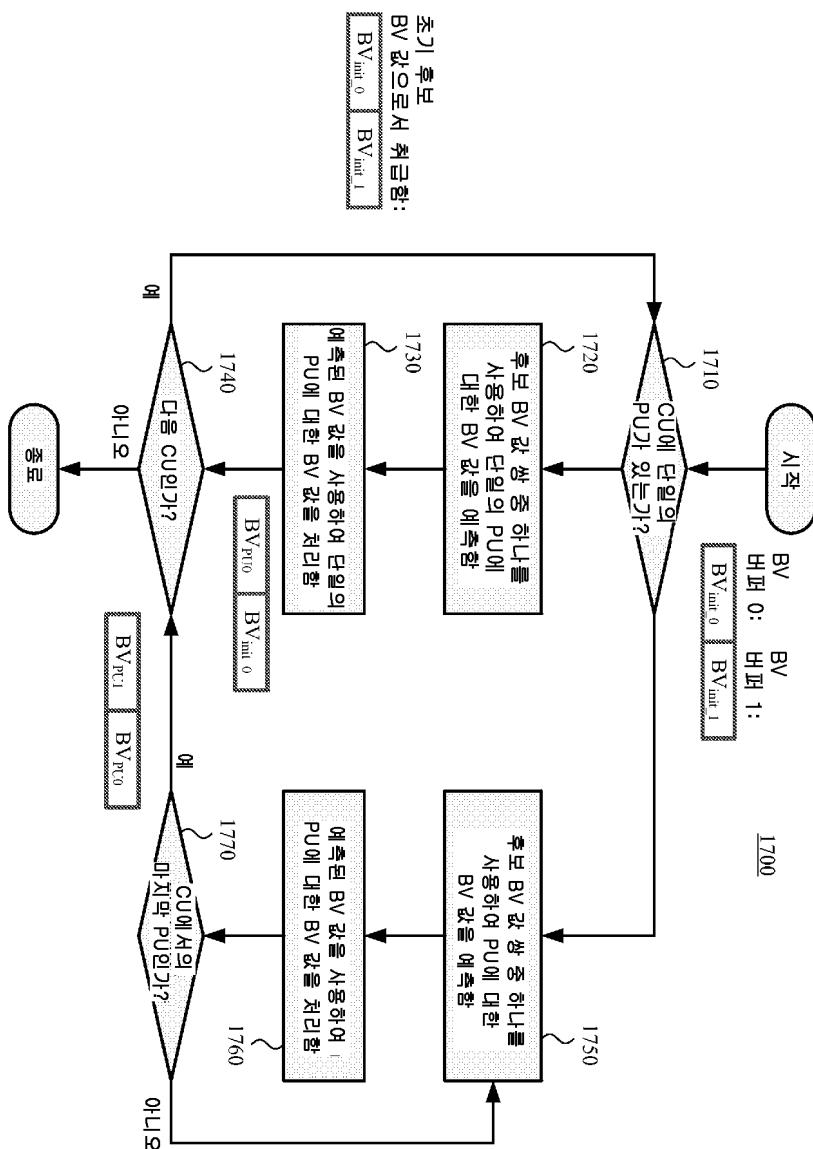
## 도면15



## 도면16

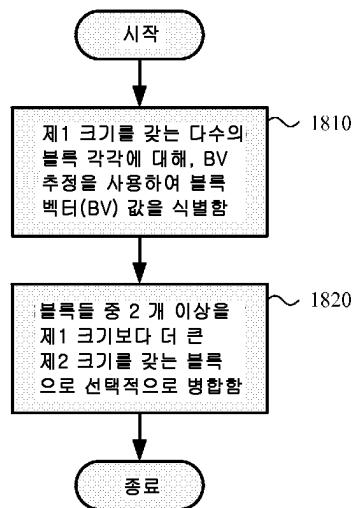


## 도면17



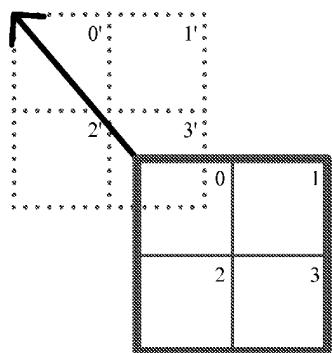
## 도면18

1800



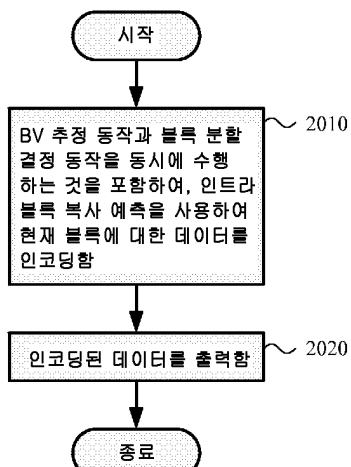
## 도면19

1900

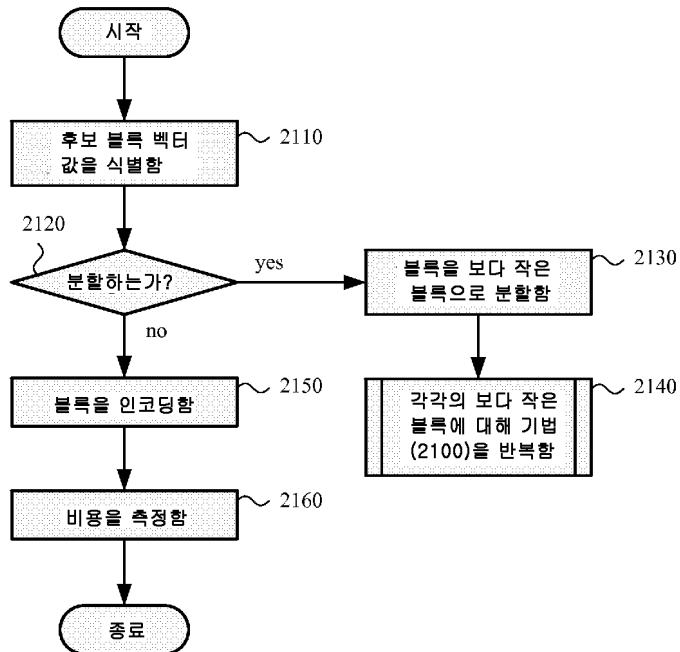


## 도면20

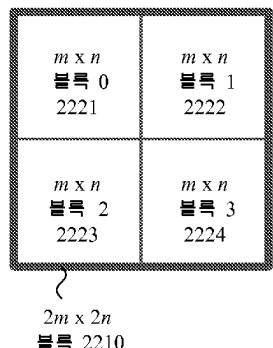
2000



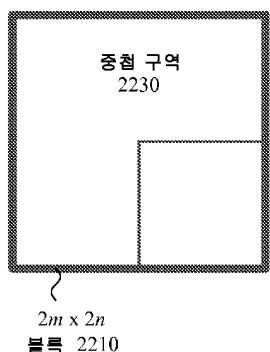
## 도면21

2100

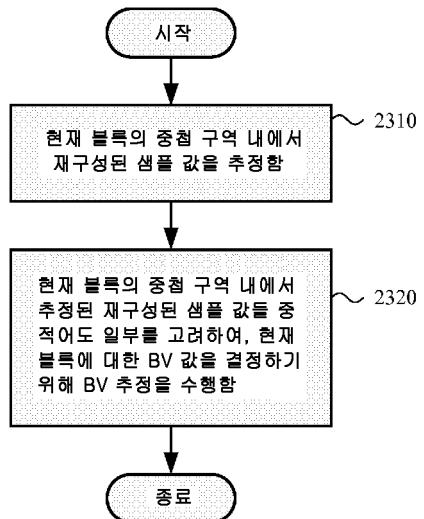
## 도면22a



## 도면22b



## 도면23

2300

## 도면24

