



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2024년05월23일

(11) 등록번호 10-2668065

(24) 등록일자 2024년05월17일

(51) 국제특허분류(Int. Cl.)

H04N 19/70 (2014.01) H04N 19/119 (2014.01)

H04N 19/176 (2014.01) H04N 19/436 (2014.01)

H04N 19/463 (2014.01)

(52) CPC특허분류

H04N 19/70 (2015.01)

H04N 19/119 (2015.01)

(21) 출원번호 10-2022-7001087

(22) 출원일자(국제) 2020년05월28일

심사청구일자 2022년01월12일

(85) 번역문제출일자 2022년01월12일

(65) 공개번호 10-2022-0019289

(43) 공개일자 2022년02월16일

(86) 국제출원번호 PCT/JP2020/021183

(87) 국제공개번호 WO 2020/255667

국제공개일자 2020년12월24일

(30) 우선권주장

JP-P-2019-115751 2019년06월21일 일본(JP)

(56) 선행기술조사문헌

비특허문헌 1 (BENJAMIN BROSS VERSATILE VIDEO

CODING (DRAFT 5) JVET\_N1001\_V8.DOCX

JVET\_N1001 VERSION 8 2019.06.11)\*

비특허문헌 2 (YE\_KUI WANG AHG12:

MISCELLANEOUS AHG12 TOPICS JVET\_00145\_V1.DOCX

JVET\_00145 VERSION 1 2019.06.20)\*

\*는 심사관에 의하여 인용된 문헌

(73) 특허권자

캐논 가부시끼가이샤

일본 도쿄도 오오따꾸 시모마루쵸 3쵸메 30방 2고

(72) 발명자

오카와 고지

일본 1468501 도쿄도 오오따꾸 시모마루쵸 3쵸메

30방 2고 캐논 가부시끼가이샤 내

시마 마사토

일본 1468501 도쿄도 오오따꾸 시모마루쵸 3쵸메

30방 2고 캐논 가부시끼가이샤 내

(74) 대리인

장수길, 이중희

전체 청구항 수 : 총 5 항

심사관 : 김혜린

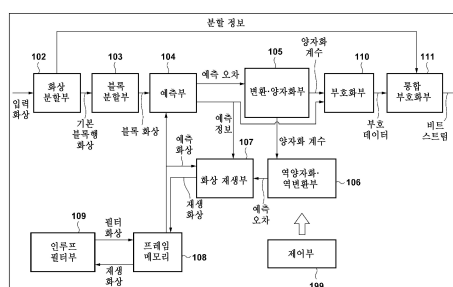
(54) 발명의 명칭 화상 부호화 장치, 화상 복호 장치, 화상 부호화 방법, 화상 복호 방법, 컴퓨터 프로그램을 저장한 비일시적 컴퓨터-판독가능 저장 매체

## (57) 요약

화상을 복수의 타일로 분할하고, 해당 타일을, 해당 타일 이하의 크기의 복수의 블록을 포함하는 블록행을 1개 이상 포함하는 직사각형 영역으로 분할한다. 화상에 있어서의 슬라이스에 포함되는 직사각형 영역의 수를 나타내는 정보에 기초하여, 해당 슬라이스에 포함되는 직사각형 영역을 특정한다. 해당 특정된 직사각형 영역에 있

(뒷면에 계속)

## 대표도



어서의 수직 방향에 있어서의 블록의 수에 기초하여, 슬라이스에 있어서의 블록행의 부호 데이터의 선두 위치를 특정하기 위한 정보의 수를 특정한다. 특정한 수의 정보와, 슬라이스에 포함되는 직사각형 영역의 수를 나타내는 정보와, 블록행의 부호 데이터를 적어도 다중화한, 비트 스트림을 생성한다.

(52) CPC특허분류

*H04N 19/176* (2015.01)

*H04N 19/436* (2015.01)

*H04N 19/463* (2015.01)

---

## 명세서

### 청구범위

#### 청구항 1

삭제

#### 청구항 2

삭제

#### 청구항 3

삭제

#### 청구항 4

복수의 블록으로 형성된 블록행을 1개 이상 포함하는 직사각형 영역을 포함하는 화상을 부호화하여 얻어진 비트 스트림으로부터 화상을 복호하는 화상 복호 장치로서,

병렬 처리를 가능하게 하는 것과 관련된 `entropy_coding_sync_enabled_flag`를 복호하고, 슬라이스가 직사각형인 제1 모드가 사용되는지 또는 슬라이스와 관련되고 상기 제1 모드와 상이한 제2 모드가 사용되는지를 나타내는 `rect_slice_flag`를 상기 비트 스트림의 픽처 파라미터 세트로부터 복호하고, 상기 화상 내 직사각형 영역의 수직 방향으로 블록의 수에 대응되는 제1 정보를 복호하고, 상기 화상 내 슬라이스에 포함되는 직사각형 영역의 수의 제2 정보를 상기 비트스트림의 슬라이스 헤더로부터 복호하고, 상기 화상이 복수의 직사각형 영역을 포함하고 `rect_slice_flag`가 상기 제2 모드가 사용되는 것을 나타내는 경우 상기 슬라이스 헤더로부터 상기 슬라이스의 `address`를 복호하는 복호 유닛과,

상기 `entropy_coding_sync_enabled_flag`의 값이 1이고 상기 `rect_slice_flag`가 상기 제2 모드가 사용되는 것을 나타내는 경우, 상기 슬라이스의 `address`, 상기 직사각형 영역의 수직 방향으로 블록의 수에 대응되는 상기 제1 정보, 및 상기 슬라이스에 포함되는 직사각형 영역의 수의 상기 제2 정보에 기초하여, 상기 슬라이스에 있어서의 블록행의 부호 데이터의 선두 위치를 특정하기 위한 정보의 수를 특정하는 특정 유닛을 구비하고,

상기 복호 유닛은, 상기 선두 위치를 특정하기 위한 정보의 수와, 상기 선두 위치를 특정하기 위한 정보에 적어도 기초하여 상기 블록행의 부호 데이터를 복호하는, 화상 복호 장치.

#### 청구항 5

제4항에 있어서,

상기 복호 유닛은, 상기 직사각형 영역에 있어서의 블록행들의 부호 데이터를 병렬로 복호하는, 화상 복호 장치.

#### 청구항 6

제4항에 있어서,

상기 화상은, 동화상에 있어서의 각 프레임의 화상 또는 정지 화상인, 화상 복호 장치.

#### 청구항 7

삭제

#### 청구항 8

복수의 블록으로 형성된 블록행을 1개 이상 포함하는 직사각형 영역을 포함하는 화상을 부호화하여 얻어진 비트 스트림으로부터 화상을 복호하는 화상 복호 방법으로서,

병렬 처리를 가능하게 하는 것과 관련된 `entropy_coding_sync_enabled_flag`를 복호하고,

슬라이스가 직사각형인 제1 모드가 사용되는지 또는 슬라이스와 관련되고 상기 제1 모드와 상이한 제2 모드가 사용되는지를 나타내는 `rect_slice_flag`를 상기 비트 스트림의 픽처 파라미터 세트로부터 복호하고,

상기 화상 내 직사각형 영역의 수직 방향으로 블록의 수에 대응되는 제1 정보를 복호하고,

상기 화상 내 슬라이스에 포함되는 직사각형 영역의 수의 제2 정보를 상기 비트스트림의 슬라이스 헤더로부터 복호하고,

상기 화상이 복수의 직사각형 영역을 포함하고 `rect_slice_flag`가 상기 제2 모드가 사용되는 것을 나타내는 경우 상기 슬라이스 헤더로부터 상기 슬라이스의 `address`를 복호하고,

상기 `entropy_coding_sync_enabled_flag`의 값이 1이고 상기 `rect_slice_flag`가 상기 제2 모드가 사용되는 것을 나타내는 경우, 상기 슬라이스의 `address`, 상기 직사각형 영역의 수직 방향으로 블록의 수에 대응되는 상기 제1 정보, 및 상기 슬라이스에 포함되는 직사각형 영역의 수의 상기 제2 정보에 기초하여, 상기 슬라이스에 있어서의 블록행의 부호 데이터의 선두 위치를 특정하기 위한 정보의 수를 특정하는 것을 포함하고,

상기 블록행의 부호 데이터는, 상기 선두 위치를 특정하기 위한 정보의 수와, 상기 선두 위치를 특정하기 위한 정보에 적어도 기초하여 복호되는,

화상 복호 방법.

## 청구항 9

삭제

## 청구항 10

복수의 블록으로 형성된 블록행을 1개 이상 포함하는 직사각형 영역을 포함하는 화상을 부호화하여 얻어진 비트 스트림으로부터 화상을 복호하는 화상 복호 방법을 컴퓨터에게 실행시키기 위한, 컴퓨터 프로그램을 저장한 비일시적 컴퓨터-판독가능 저장 매체로서, 상기 화상 복호 방법은,

병렬 처리를 가능하게 하는 것과 관련된 `entropy_coding_sync_enabled_flag`를 복호하고,

슬라이스가 직사각형인 제1 모드가 사용되는지 또는 슬라이스와 관련되고 상기 제1 모드와 상이한 제2 모드가 사용되는지를 나타내는 `rect_slice_flag`를 상기 비트 스트림의 픽처 파라미터 세트로부터 복호하고,

상기 화상 내 직사각형 영역의 수직 방향으로 블록의 수에 대응되는 제1 정보를 복호하고,

상기 화상 내 슬라이스에 포함되는 직사각형 영역의 수의 제2 정보를 상기 비트스트림의 슬라이스 헤더로부터 복호하고,

상기 화상이 복수의 직사각형 영역을 포함하고 `rect_slice_flag`가 상기 제2 모드가 사용되는 것을 나타내는 경우 상기 슬라이스 헤더로부터 상기 슬라이스의 `address`를 복호하고,

상기 `entropy_coding_sync_enabled_flag`의 값이 1이고 상기 `rect_slice_flag`가 상기 제2 모드가 사용되는 것을 나타내는 경우, 상기 슬라이스의 `address`, 상기 직사각형 영역의 수직 방향으로 블록의 수에 대응되는 상기 제1 정보, 및 상기 슬라이스에 포함되는 직사각형 영역의 수의 상기 제2 정보에 기초하여, 상기 슬라이스에 있어서의 블록행의 부호 데이터의 선두 위치를 특정하기 위한 정보의 수를 특정하는 것을 포함하고,

상기 블록행의 부호 데이터는, 상기 선두 위치를 특정하기 위한 정보의 수와, 상기 선두 위치를 특정하기 위한 정보에 적어도 기초하여 복호되는,

컴퓨터 프로그램을 저장한 비일시적 컴퓨터-판독가능 저장 매체.

## 청구항 11

삭제

## 청구항 12

삭제

## 청구항 13

삭제

## 청구항 14

삭제

## 발명의 설명

### 기술 분야

[0001] 본 발명은, 화상의 부호화/복호 기술에 관한 것이다.

### 배경 기술

[0002] 동화상의 압축 기록의 부호화 방식으로서, HEVC(High Efficiency Video Coding) 부호화 방식(이하, HEVC라 기재함)이 알려져 있다. HEVC에서는 부호화 효율 향상을 위해, 종래의 매크로 블록( $16 \times 16$ 화소)보다 큰 사이즈의 기본 블록이 채용되었다. 이 큰 사이즈의 기본 블록은 CTU(Coding Tree Unit)라 불리며, 그 사이즈는 최대  $64 \times 64$ 화소이다. CTU는 예측이나 변환을 행하는 단위가 되는 서브블록으로 더 분할된다.

[0003] 또한 HEVC에서는, 픽처를 복수의 타일 또는 슬라이스로 분할하여 부호화하는 것이 가능하다. 각 타일간 또는 슬라이스간에는 데이터의 의존성이 적으며, 병렬로 부호화·복호화 처리를 실시할 수 있다. 멀티코어의 CPU 등에 의해 병렬로 처리를 실행하여, 처리 시간을 단축할 수 있는 것을, 타일, 슬라이스 분할의 큰 이점 중 하나로 서 들 수 있다.

[0004] 또한, 각 슬라이스는 HEVC에 채용되어 있는 종래의 2차 산술 부호화의 방법에 의해 부호화된다. 즉, 각 선택 요소가 2차화되어, 2차 신호가 생성된다. 각선택 요소에는, 미리 발생 확률이 테이블(이하, 발생 확률 테이블)로서 부여되고, 2차 신호는 발생 확률 테이블에 기초하여 산술 부호화된다. 이 발생 확률 테이블은 복호 시에는 복호 정보로서, 이어지는 부호의 복호에 사용된다. 부호화 시에는 부호화 정보로서, 이어지는 부호화에 사용된다. 그리고 부호화가 행해질 때마다, 부호화된 2차 신호가 발생 확률이 높은 쪽의 심볼인지 여부를 나타내는 통계 정보에 기초하여 발생 확률 테이블이 갱신된다.

[0005] 또한 HEVC에는, Wavefront Parallel Processing(이하, WPP)이라 불리는 엔트로피 부호화·복호화를 병렬로 처리하기 위한 방법이 있다. WPP에서는, 미리 지정된 위치의 블록을 부호화 처리한 시점에서의 발생 확률의 테이블을, 다음 행의 좌측 끝의 블록에 적용함으로써, 부호화 효율의 저하를 억제한 후에 행 단위로의 블록의 병렬 부호화 처리가 가능해진다. 블록행 단위로의 병렬 처리를 가능하게 하기 위해, 슬라이스 헤더에는 비트 스트림 중의 각 블록행의 선두 위치를 나타내는 entry\_point\_offset\_minus1 및 그 수를 나타내는 num\_entry\_point\_offsets가 부호화된다. 특허문헌 1에서는, WPP와 관련된 기술이 개시되어 있다.

[0006] 근년, HEVC의 후계로서 더욱 고효율의 부호화 방식의 국제 표준화를 행하는 활동이 개시되었다. JVET(Joint Video Experts Team)가 ISO/IEC와 ITU-T의 사이에 설립되어, VVC(Versatile Video Coding) 부호화 방식(이하, VVC)으로서 표준화가 진행되고 있다. VVC에서는, 타일을 복수의 블록행으로 구성되는 직사각형(브릭)으로 더 분할하는 것이 검토되고 있다. 그리고 슬라이스는 1개 이상의 브릭을 포함하도록 구성된다.

## 선행기술문헌

### 특허문헌

[0007] (특허문헌 0001) 일본 특허 공개 제2014-11638호 공보

## 발명의 내용

### 해결하려는 과제

[0008] VVC에 있어서는, 슬라이스를 구성하는 브릭이 미리 도출 가능하고, 또한 그 브릭에 내포되는 기본 블록행의 수도 다른 선택으로부터 도출 가능하다. 그 때문에, 해당 슬라이스에 속하는 기본 블록행의 선두 위치를 나타내

는 entry\_point\_offset\_minus1의 수를, num\_entry\_point\_offset을 사용하지 않고 도출하는 것이 가능하다. 그 때문에 num\_entry\_point\_offset은 용장한 선택스가 된다. 본 발명에서는, 용장한 선택스를 저감시킴으로써 비트 스트림의 부호량을 저감시키는 기술을 제공한다.

### 과제의 해결 수단

[0009] 본 발명의 일 양태는, 화상을 복수의 타일로 분할하고, 해당 타일을, 해당 타일 이하의 크기의 복수의 블록을 포함하는 블록행을 1개 이상 포함하는 직사각형 영역으로 분할하는 분할 수단과, 상기 화상에 있어서의 슬라이스에 포함되는 직사각형 영역의 수를 나타내는 정보에 기초하여 해당 슬라이스에 포함되는 상기 직사각형 영역을 특정하고, 해당 특정된 상기 직사각형 영역에 있어서의 수직 방향에 있어서의 블록의 수에 기초하여 상기 슬라이스에 있어서의 상기 블록행의 부호 데이터의 선두 위치를 특정하기 위한 정보의 수를 특정하는 특정 수단과, 상기 특정 수단에 의해 특정한 수의 상기 선두 위치를 특정하기 위한 상기 정보와, 상기 슬라이스에 포함되는 직사각형 영역의 수를 나타내는 상기 정보와, 상기 블록행의 부호 데이터를 적어도 다중화한, 비트 스트림을 생성하는 생성 수단을 구비하는 것을 특징으로 한다.

### 발명의 효과

[0010] 본 발명의 구성에 따르면, 용장한 선택스를 저감시킴으로써 비트 스트림의 부호량을 저감시킬 수 있다.

[0011] 본 발명의 그 밖의 특징 및 이점은, 첨부 도면을 참조로 한 이하의 설명에 의해 명백해질 것이다. 또한, 첨부 도면에 있어서는, 동일하거나 혹은 마찬가지로의 구성에는 동일한 참조 번호를 붙인다.

### 도면의 간단한 설명

[0012] 첨부 도면은 명세서에 포함되고, 그 일부를 구성하며, 본 발명의 실시 형태를 나타내고, 그 서술과 함께 본 발명의 원리를 설명하기 위해 사용된다.

도 1은 화상 부호화 장치의 기능 구성 예를 도시하는 블록도.

도 2는 화상 복호 장치의 기능 구성 예를 도시하는 블록도.

도 3은 화상 부호화 장치에 의한 입력 화상의 부호화 처리의 흐름도.

도 4는 화상 복호 장치에 의한 비트 스트림의 복호 처리의 흐름도.

도 5는 컴퓨터 장치의 하드웨어 구성 예를 도시하는 블록도.

도 6은 비트 스트림의 포맷의 일례를 도시하는 도면.

도 7은 입력 화상의 분할 예를 도시하는 도면.

도 8a는 입력 화상의 분할 예를 도시하는 도면.

도 8b는 입력 화상의 분할 예를 도시하는 도면.

도 9a는 타일과 슬라이스의 관계를 도시하는 도면.

도 9b는 타일과 슬라이스의 관계를 도시하는 도면.

### 발명을 실시하기 위한 구체적인 내용

[0013] 이하, 첨부 도면을 참조하여 실시 형태를 상세하게 설명한다. 또한, 이하의 실시 형태는 특허 청구의 범위에 관한 발명을 한정하는 것은 아니다. 실시 형태에는 복수의 특징이 기재되어 있지만, 이들 복수의 특징 전부가 발명에 필수적인 것만은 아니며, 또한 복수의 특징은 임의로 조합되어도 된다. 또한, 첨부 도면에 있어서는 동일 혹은 마찬가지로의 구성에 동일한 참조 번호를 붙이고 중복된 설명은 생략한다.

[0014] [제1 실시 형태]

[0015] 우선, 본 실시 형태에 관한 화상 부호화 장치의 기능 구성 예에 대해 도 1의 블록도를 사용하여 설명한다. 화상 분할부(102)에는, 부호화 대상이 되는 입력 화상이 입력된다. 입력 화상은 동화상을 구성하는 각 프레임의 화상이어도 되고, 정지 화상이어도 된다. 화상 분할부(102)는, 입력 화상을 「1개 혹은 복수의 타일」로 분할한다. 타일은 입력 화상 내의 직사각형 영역을 덮는, 연속되는 기본 블록의 집합이다. 화상 분할부(102)는,

각각의 타일을 1개 혹은 복수의 브릭으로 더 분할한다. 브릭은, 타일 내의 1개 혹은 복수의 기본 블록의 행(기본 블록행)으로 구성되는 직사각형 영역(타일 이하의 크기의 복수의 블록을 포함하는 블록행을 1개 이상 포함하는 직사각형 영역)이다. 화상 분할부(102)는 또한 입력 화상을, 「1개 또는 복수의 타일」 혹은 「1개의 타일 내의 1개 이상의 브릭」으로 구성되는 슬라이스로 분할한다. 슬라이스는 부호화의 기본 단위이며, 슬라이스마다 슬라이스의 종류를 나타내는 정보 등의 헤더 정보가 부가된다. 입력 화상을 4개의 타일, 4개의 슬라이스, 11개의 브릭으로 분할하는 예를 도 7에 도시한다. 좌측 상단의 타일은 1개의 브릭, 좌측 하단의 타일은 2개의 브릭, 우측 상단의 타일은 5개의 브릭, 우측 하단의 타일은 3개의 브릭으로 각각 분할되어 있다. 그리고 좌측의 슬라이스는 3개의 브릭, 우측 상단의 슬라이스는 2개의 브릭, 우측 중앙의 슬라이스는 3개의 브릭, 우측 하단의 슬라이스는 3개의 브릭을 포함하도록 구성되어 있다. 화상 분할부(102)는, 이와 같이 하여 분할한 타일, 브릭, 슬라이스의 각각에 대해, 크기에 관한 정보를 분할 정보로서 출력한다.

- [0016] 블록 분할부(103)는, 화상 분할부(102)로부터 출력된 기본 블록행의 화상(기본 블록행 화상)을 복수의 기본 블록으로 분할하고, 기본 블록 단위의 화상(블록 화상)을 후단으로 출력한다.
- [0017] 예측부(104)는, 기본 블록 단위의 화상을 서브블록으로 분할하고, 서브블록 단위로 프레임 내 예측인 인트라 예측이나 프레임간 예측인 인터 예측 등을 행하여, 예측 화상을 생성한다. 브릭에 걸친 인트라 예측(다른 브릭의 블록의 화소를 사용한 인트라 예측), 브릭에 걸친 움직임 벡터의 예측(다른 브릭의 블록의 움직임 벡터를 사용한 움직임 벡터의 예측)은 행해지지 않는다. 또한 예측부(104)는, 입력된 화상과 예측 화상으로부터 예측 오차를 산출하여 출력한다. 또한 예측부(104)는, 예측에 필요한 정보(예측 정보), 예를 들어 서브블록 분할 방법, 예측 모드나 움직임 벡터 등의 정보도 예측 오차와 함께 출력한다.
- [0018] 변환·양자화부(105)는, 예측 오차를 서브블록 단위로 직교 변환하여 변환 계수를 구하고, 해당 구한 변환 계수를 양자화하여 양자화 계수를 구한다. 역양자화·역변환부(106)는, 변환·양자화부(105)로부터 출력된 양자화 계수를 역양자화하여 변환 계수를 재생하고, 또한 해당 재생된 변환 계수를 역직교 변환하여 예측 오차를 재생한다.
- [0019] 프레임 메모리(108)는, 재생된 화상을 저장해 두는 메모리로서 기능한다. 화상 재생부(107)는, 예측부(104)로부터 출력된 예측 정보에 기초하여 프레임 메모리(108)를 적절하게 참조하여 예측 화상을 생성하고, 해당 예측 화상과 입력된 예측 오차로부터 재생 화상을 생성하여 출력한다.
- [0020] 인루프 필터부(109)는, 재생 화상에 대해 디블로킹 필터나 샘플 어댑티브 오프셋 등의 인루프 필터 처리를 행하고, 해당 인루프 필터 처리가 실시된 화상(필터 화상)을 출력한다.
- [0021] 부호화부(110)는, 변환·양자화부(105)로부터 출력된 양자화 계수 및 예측부(104)로부터 출력된 예측 정보를 부호화함으로써 부호 데이터(부호화 데이터)를 생성하고, 해당 생성된 부호 데이터를 출력한다.
- [0022] 통합 부호화부(111)는, 화상 분할부(102)로부터 출력된 분할 정보를 사용하여 헤더 부호 데이터를 생성하고, 해당 생성된 헤더 부호 데이터와, 부호화부(110)로부터 출력된 부호 데이터를 포함하는 비트 스트림을 생성하여 출력한다. 제어부(199)는, 화상 부호화 장치 전체의 동작 제어를 행하는 것이며, 상기한 화상 부호화 장치의 각 기능부의 동작 제어를 행한다.
- [0023] 다음으로, 도 1에 도시한 구성을 갖는 화상 부호화 장치에 의한 입력 화상에 대한 부호화 처리에 대해 설명한다. 본 실시 형태에서는, 설명을 용이하게 하기 위해 인트라 예측 부호화의 처리만을 설명하지만, 이것에 한정되지 않고 인터 예측 부호화의 처리에 있어서도 적용 가능하다. 또한 본 실시 형태에서는 구체적인 설명을 행하기 위해, 블록 분할부(103)는, 화상 분할부(102)로부터 출력된 기본 블록행 화상을 「64×64화소의 사이즈를 갖는 기본 블록」을 단위로 분할하는 것으로서 설명한다.
- [0024] 화상 분할부(102)는 입력 화상을 타일, 브릭, 슬라이스로 분할한다. 화상 분할부(102)에 의한 입력 화상의 분할 예를 도 8a, 도 8b에 도시한다. 본 실시 형태에서는 도 8a에 도시하는 바와 같이, 1152×1152화소의 사이즈를 갖는 입력 화상을 9개의 타일(1개의 타일의 사이즈는 384×384화소)로 분할한다. 각 타일에는 좌측 상단으로부터 래스터순으로 ID(타일 ID)가 부여되며, 좌측 상단의 타일의 타일 ID는 0, 우측 하단의 타일의 타일 ID는 8이다.
- [0025] 또한, 입력 화상에 있어서의 타일, 브릭, 슬라이스의 분할 예를 도 8b에 도시한다. 도 8b에 도시하는 바와 같이, 타일 ID=0의 타일 및 타일 ID=7의 타일은 2개의 브릭(각각의 브릭의 사이즈는 384×192화소)으로 분할된다. 타일 ID=2의 타일은 2개의 브릭(상측의 브릭의 사이즈는 384×128화소, 하측의 브릭의 사이즈는 384×256화소)으로 분할된다. 타일 ID=3의 타일은 3개의 브릭(각각의 브릭의 사이즈는 384×128화소)으로 분할된다.

타일 ID=1, 4, 5, 6, 8의 타일은 브릭으로 분할되어 있지 않고(1개의 타일을 1개의 브릭으로 분할하는 것과 등가), 그 결과, 타일=브릭으로 되어 있다. 각 브릭에는 래스터순의 타일 중에서 위로부터 차례로 ID가 부여된다. 도 8b에 도시하는 BID가 브릭의 ID이다. 또한 입력 화상은, 슬라이스 0 내지 4로 분할되어 있다. 슬라이스 0은 BID=0에 대응하는 브릭을 포함하는 슬라이스, 슬라이스 1은 BID=1에 대응하는 브릭을 포함하는 슬라이스, 슬라이스 2는 BID=2에 대응하는 브릭을 포함하는 슬라이스이다. 슬라이스 3은 BID=3 내지 10에 대응하는 브릭을 포함하는 슬라이스, 슬라이스 4는 BID=11 내지 13에 대응하는 브릭을 포함하는 슬라이스이다. 또한, 각 슬라이스에도, 래스터순의 슬라이스 중에서 위로부터 차례로 ID가 부여되어 있으며, 예를 들어 슬라이스 0은 ID=0의 슬라이스, 슬라이스 4는 ID=4의 슬라이스를 가리키는 것으로 한다.

[0026] 그리고 화상 분할부(102)는, 분할된 타일, 브릭, 슬라이스의 각각에 대해, 크기에 관한 정보를 분할 정보로서 통합 부호화부(111)에 출력한다. 또한 화상 분할부(102)는, 각 브릭을 기본 블록행 화상으로 분할하고, 해당 분할된 기본 블록행 화상을 블록 분할부(103)에 출력한다.

[0027] 블록 분할부(103)는, 화상 분할부(102)로부터 출력된 기본 블록행 화상을 복수의 기본 블록으로 분할하고, 기본 블록 단위의 화상인 블록 화상(64×64화소)을, 후단의 예측부(104)에 대해 출력한다.

[0028] 예측부(104)는, 기본 블록 단위의 화상을 서브블록으로 분할하고, 서브블록 단위로 수평 예측이나 수직 예측 등의 인트라 예측 모드를 결정하고, 해당 결정된 인트라 예측 모드 및 부호화 완료된 화소로부터 예측 화상을 생성한다. 또한 예측부(104)는, 입력된 화상과 예측 화상으로부터 예측 오차를 산출하고, 해당 산출된 예측 오차를 변환·양자화부(105)에 대해 출력한다. 또한 예측부(104)는, 서브블록 분할 방법이나 인트라 예측 모드 등의 정보를 예측 정보로서, 부호화부(110) 및 화상 재생부(107)에 대해 출력한다.

[0029] 변환·양자화부(105)는, 예측부(104)로부터 출력된 예측 오차를 서브블록 단위로 직교 변환(서브블록의 사이즈에 대응한 직교 변환 처리)하여 변환 계수(직교 변환 계수)를 구한다. 그리고 변환·양자화부(105)는, 당해 구한 변환 계수를 양자화하여 양자화 계수를 구한다. 그리고 변환·양자화부(105)는, 당해 구한 양자화 계수를 부호화부(110) 및 역양자화·역변환부(106)에 대해 출력한다.

[0030] 역양자화·역변환부(106)는, 변환·양자화부(105)로부터 출력된 양자화 계수를 역양자화하여 변환 계수를 재생하고, 또한 해당 재생된 변환 계수를 역직교 변환하여 예측 오차를 재생한다. 그리고 역양자화·역변환부(106)는, 해당 재생된 예측 오차를 화상 재생부(107)에 대해 출력한다.

[0031] 화상 재생부(107)는, 예측부(104)로부터 출력된 예측 정보에 기초하여 프레임 메모리(108)를 적절하게 참조하여 예측 화상을 생성하고, 해당 예측 화상과, 역양자화·역변환부(106)로부터 입력된 예측 오차로부터 재생 화상을 생성한다. 그리고 화상 재생부(107)는, 해당 생성된 재생 화상을 프레임 메모리(108)에 저장한다.

[0032] 인루프 필터부(109)는, 프레임 메모리(108)로부터 재생 화상을 판독하고, 해당 판독된 재생 화상에 대해 디블로킹 필터나 샘플 어댑티브 오프셋 등의 인루프 필터 처리를 행한다. 그리고 인루프 필터부(109)는, 해당 인루프 필터 처리가 실시된 화상을 다시 프레임 메모리(108)에 저장(재저장)한다.

[0033] 부호화부(110)는, 변환·양자화부(105)로부터 출력된 양자화 계수 및 예측부(104)로부터 출력된 예측 정보를 엔트로피 부호화함으로써 부호 데이터를 생성한다. 엔트로피 부호화의 방법은 특별히 지정하지 않지만, 곱셈 부호화, 산술 부호화, 허프만 부호화 등을 사용할 수 있다. 그리고 부호화부(110)는, 해당 생성된 부호 데이터를 통합 부호화부(111)에 대해 출력한다.

[0034] 통합 부호화부(111)는, 화상 분할부(102)로부터 출력된 분할 정보를 사용하여 헤더 부호 데이터를 생성하고, 해당 생성된 헤더 부호 데이터와, 부호화부(110)로부터 출력된 부호 데이터를 다중화함으로써 비트 스트림을 생성하여 출력한다. 비트 스트림의 출력처는 특정한 출력처에 한정되는 것은 아니며, 화상 부호화 장치의 내부 혹은 외부의 메모리에 출력(저장)해도 되고, LAN이나 인터넷 등의 네트워크를 통해 화상 부호화 장치와 통신 가능한 외부 장치에 대해 송신해도 된다.

[0035] 다음으로, 통합 부호화부(111)가 출력하는 비트 스트림(화상 부호화 장치가 부호화하는 VVC에 의한 부호 데이터)의 포맷의 일례를 도 6에 도시한다. 도 6의 비트 스트림에는, 시퀀스의 부호화에 관계되는 정보가 포함된 헤더 정보인 시퀀스 파라미터 세트(PPS)가 포함되어 있다. 또한 도 6의 비트 스트림에는, 픽처의 부호화에 관계되는 정보가 포함된 헤더 정보인 픽처 파라미터 세트(PPS)가 포함되어 있다. 또한 도 6의 비트 스트림에는, 슬라이스의 부호화에 관계되는 정보가 포함된 헤더 정보인 슬라이스 헤더(SLH)가 포함되어 있다. 또한 도 6의 비트 스트림에는, 각 브릭(도 6에서는 브릭 0 내지 브릭 (N-1))의 부호 데이터가 포함되어 있다.

- [0036] SPS에는 화상 사이즈 정보와 기본 블록 데이터 분할 정보가 포함되어 있다. PPS에는, 타일의 분할 정보인 타일 데이터 분할 정보와, 브릭의 분할 정보인 브릭 데이터 분할 정보와, 슬라이스의 분할 정보인 슬라이스 데이터 분할 정보 0과, 기본 블록행 데이터 동기화 정보가 포함되어 있다. SLH에는, 슬라이스 데이터 분할 정보 1과 기본 블록행 데이터 위치 정보가 포함되어 있다.
- [0037] 우선, SPS에 대해 설명한다. SPS에는 화상 사이즈 정보로서 정보 601인 `pic_width_in_luma_samples` 및 정보 602인 `pic_height_in_luma_samples`가 포함되어 있다. `pic_width_in_luma_samples`는 입력 화상의 수평 방향의 사이즈(화소수)를 나타내고 있고, `pic_height_in_luma_samples`는 입력 화상의 수직 방향의 사이즈(화소수)를 나타내고 있다. 본 실시 형태에서는, 입력 화상으로서 도 8a, 도 8b의 입력 화상을 사용하므로 `pic_width_in_luma_samples`=1152, `pic_height_in_luma_samples`=1152가 된다. 또한 SPS에는 기본 블록 데이터 분할 정보로서 정보 603인 `log2_ctu_size_minus2`가 포함되어 있다. `log2_ctu_size_minus2`는, 기본 블록의 사이즈를 나타낸다. 기본 블록의 수직 방향 및 수평 방향의 화소수는  $1 \ll (\log2\_ctu\_size\_minus2 + 2)$ 로 나타난다. 본 실시 형태에서는 기본 블록의 사이즈는  $64 \times 64$ 화소이므로, `log2_ctu_size_minus2`의 값은 4가 된다.
- [0038] 다음으로, PPS에 대해 설명한다. PPS에는 타일 데이터 분할 정보로서 정보 604 내지 607이 포함되어 있다. 정보 604는, 입력 화상이 복수의 타일로 분할되어 부호화되어 있는지 여부를 나타내는 `single_tile_in_pic_flag`이다. `single_tile_in_pic_flag`=1인 경우는, 입력 화상이 복수의 타일로 분할되어 부호화되어 있지 않은 것을 나타낸다. 한편, `single_tile_in_pic_flag`=0인 경우는, 입력 화상이 복수의 타일로 분할되어 부호화되어 있는 것을 나타낸다.
- [0039] 정보 605는, `single_tile_in_pic_flag`=0인 경우에 타일 데이터 분할 정보에 포함되는 정보이다. 정보 605는, 각 타일이 동일한 사이즈를 갖는지 여부를 나타내는 `uniform_tile_spacing_flag`이다. `uniform_tile_spacing_flag`=1인 경우는, 각 타일이 동일한 사이즈를 갖는 것을 나타내고, `uniform_tile_spacing_flag`=0인 경우는, 사이즈가 동일하지 않은 타일이 존재하는 것을 나타낸다.
- [0040] 정보 606 및 정보 607은, `uniform_tile_spacing_flag`=1인 경우에 타일 데이터 분할 정보에 포함되는 정보이다. 정보 606은, (타일의 수평 방향의 기본 블록수-1)을 나타내는 `tile_cols_width_minus1`이다. 정보 607은, (타일의 수직 방향의 기본 블록수-1)을 나타내는 `tile_rows_height_minus1`이다. 입력 화상의 수평 방향의 타일수는, 입력 화상의 수평 방향의 기본 블록수를 타일의 수평 방향의 기본 블록수로 제산한 경우의 몫으로서 얻어진다. 이 제산에 의해 나머지가 발생한 경우는, 몫에 1을 더한 수를 「입력 화상의 수평 방향의 타일수」로 한다. 또한 입력 화상의 수직 방향의 타일수는, 입력 화상의 수직 방향의 기본 블록수를 타일의 수직 방향의 기본 블록수로 제산한 경우의 몫으로서 얻어진다. 이 제산에 의해 나머지가 발생한 경우는, 몫에 1을 더한 수를 「입력 화상의 수직 방향의 타일수」로 한다. 또한, 입력 화상 내의 총 타일수는, 입력 화상의 수평 방향의 타일수 $\times$ 입력 화상의 수직 방향의 타일수를 계산함으로써 구할 수 있다.
- [0041] 또한, `uniform_tile_spacing_flag`=0인 경우, 다른 것과 사이즈가 다른 타일이 포함되어 있으므로, 입력 화상의 수평 방향에 있어서의 타일의 수, 입력 화상의 수직 방향에 있어서의 타일의 수, 각 타일의 종횡의 사이즈를 부호로 한다.
- [0042] 또한 PPS에는 브릭 데이터 분할 정보로서 정보 608 내지 613이 포함되어 있다. 정보 608은, `brick_splitting_present_flag`이다. `brick_splitting_present_flag`=1인 경우는, 입력 화상 내에 있어서의 1개 이상의 타일이 복수의 브릭으로 분할되어 있는 것을 나타낸다. 한편, `brick_splitting_present_flag`=0인 경우는, 입력 화상 내에 있어서의 각각의 타일이 단일의 브릭으로 구성되어 있는 것을 나타낸다.
- [0043] 정보 609는, `brick_splitting_present_flag`=1인 경우에 브릭 데이터 분할 정보에 포함되는 정보이다. 정보 609는, 각각의 타일에 대해, 해당 타일이 복수의 브릭으로 분할되어 있는지 여부를 나타내는 `brick_split_flag[]`이다.  $i$ 번째의 타일이 복수의 브릭으로 분할되어 있는지를 나타내는 `brick_split_flag[]`를 `brick_split_flag[i]`라 표기한다. `brick_split_flag[i]`=1인 경우,  $i$ 번째의 타일이 복수의 브릭으로 분할되어 있는 것을 나타내고, `brick_split_flag[i]`=0인 경우,  $i$ 번째의 타일이 단일의 브릭으로 구성되어 있는 것을 나타낸다.
- [0044] 정보 610은, `brick_split_flag[i]`=1인 경우에,  $i$ 번째의 타일을 구성하는 각각의 브릭의 사이즈가 동일한지 여부를 나타내는 `uniform_brick_spacing_flag[i]`이다. 모든  $i$ 에 대해 `brick_split_flag[i]`=0이면, 정보 610은 브릭 데이터 분할 정보에는 포함되어 있지 않다. 정보 610은, `brick_split_flag[i]`=1을 충족하는  $i$ 에 대해,

uniform\_brick\_spacing\_flag[i]를 포함한다. uniform\_brick\_spacing\_flag[i]=1인 경우는, i번째의 타일을 구성하는 각각의 브릭의 사이즈가 동일한 것을 나타낸다. 한편, uniform\_brick\_spacing\_flag[i]=0인 경우는, i번째의 타일을 구성하는 브릭 중 다른 것과 사이즈가 다른 브릭이 존재하는 것을 나타낸다.

[0045] 정보 611은, uniform\_brick\_spacing\_flag[i]=1인 경우에 브릭 데이터 분할 정보에 포함되는 정보이다. 정보 611은, (i번째의 타일에 있어서의 브릭의 수직 방향의 기본 블록수-1)을 나타내는 brick\_height\_minus1[i]이다.

[0046] 또한, 브릭의 수직 방향의 기본 블록수는, 해당 브릭의 수직 방향의 화소수를, 기본 블록의 수직 방향의 화소수 (본 실시 형태에서는 64화소)로 제산함으로써 구할 수 있다. 또한, 타일을 구성하는 브릭수는, 타일의 수직 방향의 기본 블록수를 브릭의 수직 방향의 기본 블록수로 제산한 경우의 몫으로서 얻어진다. 이 제산에 의해 나머지가 발생한 경우는, 몫에 1을 더한 수를 「타일을 구성하는 브릭수」로 한다. 예를 들어 타일의 수직 방향의 기본 블록수가 10이고, brick\_height\_minus1의 값이 2라고 가정한다. 이때, 이 타일은 위로부터 차례로, 기본 블록행 수가 3인 브릭, 기본 블록행 수가 3인 브릭, 기본 블록행 수가 3인 브릭, 기본 블록행 수가 1인 브릭의 4개의 브릭으로 분할되게 된다.

[0047] 정보 612는, uniform\_brick\_spacing\_flag[i]=0을 충족하는 i에 대해 (i번째의 타일을 구성하는 브릭수-1)을 나타내는 num\_brick\_rows\_minus1[i]이다.

[0048] 또한, 본 실시 형태에서는, uniform\_brick\_spacing\_flag[i]=0인 경우는, (i번째의 타일을 구성하는 브릭수-1)을 나타내는 num\_brick\_rows\_minus1[i]를 브릭 데이터 분할 정보에 포함하였다. 그러나 이것에 한정되는 것은 아니다.

[0049] 예를 들어 brick\_split\_flag[i]=1의 시점에서 i번째의 타일을 구성하는 브릭의 수가 2 이상이라고 상정한다. 그리고 (해당 타일을 구성하는 브릭의 수-2)를 나타내는 num\_brick\_rows\_minus2[i]를 num\_brick\_rows\_minus1[i] 대신에 부호화해도 된다. 이와 같이 함으로써, 해당 타일을 구성하는 브릭의 수를 나타내는 선택스의 비트수를 저감시킬 수 있다. 예를 들어, 해당 타일이 2개의 브릭에 의해 구성되고, num\_brick\_rows\_minus1[i]를 곱셈 부호화하는 경우, 「1」을 나타내는 「010」의 3비트의 데이터가 부호화된다. 한편, (해당 타일을 구성하는 브릭의 수-2)를 나타내는 num\_brick\_rows\_minus2[i]를 곱셈 부호화하는 경우, 0을 나타내는 「0」의 1비트의 데이터가 부호화된다.

[0050] 정보 613은 uniform\_brick\_spacing\_flag[i]=0을 충족하는 i에 대해 (i번째의 타일에 있어서의 j번째의 브릭의 수직 방향의 기본 블록수-1)을 나타내는 brick\_row\_height\_minus1[i][j]이다. brick\_row\_height\_minus1[i][j]는 num\_brick\_rows\_minus1[i]의 수만큼 부호화된다. 또한, 상술한 num\_brick\_rows\_minus2[i]를 사용한 경우는, brick\_row\_height\_minus1[i][j]는 num\_brick\_rows\_minus2[i]+1의 수만큼 부호화된다. 타일에 있어서의 하단의 브릭의 수직 방향의 기본 블록수는, 해당 타일의 수직 방향의 기본 블록수로부터 「brick\_row\_height\_minus1+1」의 총합을 감산함으로써 구할 수 있다. 예를 들어, 타일의 수직 방향의 기본 블록수=10, num\_brick\_rows\_minus1=3, brick\_row\_height\_minus1=2, 1, 2라고 가정한다. 이때, 해당 타일에 있어서의 하단의 브릭의 수직 방향의 기본 블록수는 10-(3+2+3)=2가 된다.

[0051] 또한 PPS에는 슬라이스 데이터 분할 정보 0으로서 정보 614 내지 618이 포함되어 있다. 정보 614는, single\_brick\_per\_slice\_flag이다. single\_brick\_per\_slice\_flag=1인 경우, 입력 화상 내의 모든 슬라이스가 단일의 브릭으로 구성되어 있는 것을 나타낸다. 즉, 각각의 슬라이스가 1개만의 브릭으로 구성되는 것을 나타낸다. 한편, single\_brick\_per\_slice\_flag=0인 경우, 입력 화상에 있어서 1개 이상의 슬라이스가 복수의 브릭으로 구성되어 있는 것을 나타낸다.

[0052] 정보 615는 rect\_slice\_flag이며, single\_brick\_per\_slice\_flag=0인 경우에 슬라이스 데이터 분할 정보 0에 포함되는 정보이다. rect\_slice\_flag는, 슬라이스가 포함하는 타일이 래스터순인지 직사각형인지를 나타낸다. 도 9a는 rect\_slice\_flag=0인 경우에 있어서의 타일과 슬라이스의 관계를 도시하고 있고, 슬라이스 내의 타일이 래스터순으로 부호화되는 것을 나타내고 있다. 한편, 도 9b는 rect\_slice\_flag=1인 경우에 있어서의 타일과 슬라이스의 관계를 도시하고 있고, 슬라이스 내의 복수의 타일이 직사각형상인 것을 나타내고 있다.

[0053] 정보 616은 num\_slices\_in\_pic\_minus1이고, rect\_slice\_flag=1 또한 single\_brick\_per\_slice\_flag=0인 경우에 슬라이스 데이터 분할 정보 0에 포함되는 정보이다. num\_slices\_in\_pic\_minus1은, (입력 화상에 있어서의 슬라이스의 수-1)을 나타낸다.

[0054] 정보 617은, 입력 화상에 있어서의 각각의 슬라이스에 대해, 해당 슬라이스 (i번째의 슬라이스)의 좌측 상단의

브릭의 인덱스를 나타내는 `top_left_brick_idx[i]`이다.

- [0055] 정보 618은, 입력 화상 중  $i$ 번째의 슬라이스에 있어서의 좌측 상단의 브릭의 인덱스와 우측 하단의 브릭의 인덱스의 차분을 나타내는 `bottom_right_brick_idx_delta[i]`이다. 여기서, 「입력 화상 중  $i$ 번째의 슬라이스에 있어서의 좌측 상단의 브릭」이란, 해당 슬라이스에 있어서 최초로 처리되는 브릭이다. 또한, 「입력 화상 중  $i$ 번째의 슬라이스에 있어서의 우측 하단의 브릭」이란, 해당 슬라이스에 있어서 마지막에 처리되는 브릭이다. 단, 프레임 내의 최초의 슬라이스의 좌측 상단의 브릭의 인덱스는 0으로 정해져 있으므로, 최초의 슬라이스의 `top_left_brick_idx[0]`은 부호화되지 않는다.
- [0056] 또한 PPS에는 기본 블록행 데이터 동기화 정보로서 정보 619가 부호화되어 포함되어 있다. 정보 619는 `entropy_coding_sync_enabled_flag`이다. `entropy_coding_sync_enabled_flag=1`인 경우는, 위에 인접하는 기본 블록행의 소정 위치의 기본 블록을 처리한 시점에서의 발생 확률의 테이블을 좌측 끝의 블록에 적용한다. 이에 의해, 기본 블록행 단위로 엔트로피 부호화·복호화의 병렬 처리가 가능해진다.
- [0057] 다음으로, SLH에 대해 설명한다. SLH에는 슬라이스 데이터 분할 정보 1로서 정보 620 내지 621이 부호화되어 포함되어 있다. 정보 620은, `rect_slice_flag=1` 혹은 입력 화상 중의 브릭의 수가 2 이상인 경우에 슬라이스 데이터 분할 정보 1에 포함되는 `slice_address`이다. `slice_address`는, `rect_slice_flag=0`인 경우는 슬라이스의 선두 BID를 나타내고, `rect_slice_flag=1`인 경우는 현재의 슬라이스의 번호를 나타낸다.
- [0058] 정보 621은, `rect_slice_flag=0` 또한 `single_brick_per_slice_flag=0`인 경우에 슬라이스 데이터 분할 정보 1에 포함되는 `num_bricks_in_slice_minus1`이다. `num_bricks_in_slice_minus1`은 (슬라이스 중의 브릭수-1)을 나타낸다.
- [0059] SLH에는 기본 블록행 데이터 위치 정보로서 정보 622가 포함되어 있다. 정보 622는 `entry_point_offset_minus1[]`이다. `entry_point_offset_minus1[]`은, `entropy_coding_sync_enabled_flag=1`인 경우에, (슬라이스 중의 기본 블록행 수-1)의 수만큼 기본 블록행 데이터 위치 정보에 부호화되어 포함된다.
- [0060] `entry_point_offset_minus1[]`은, 기본 블록행의 부호 데이터의 엔트리 포인트, 즉 기본 블록행의 부호 데이터의 선두 위치를 나타낸다. `entry_point_offset_minus1[j-1]`은  $j$ 번째의 기본 블록행의 부호 데이터의 엔트리 포인트를 나타낸다. 0번째의 기본 블록행의 부호 데이터의 선두 위치는, 해당 기본 블록행이 속하는 슬라이스의 부호 데이터의 선두 위치와 동일하므로 생략된다. 그리고  $\{(j-1)$ 번째의 기본 블록행의 부호 데이터의 크기-1 $\}$ 이 `entry_point_offset_minus1[j-1]`로서 부호화된다.
- [0061] 다음으로, 본 실시 형태에 화상 부호화 장치에 의한 입력 화상의 부호화 처리(도 6의 구성을 갖는 비트 스트림의 생성 처리)에 대해, 도 3의 흐름도에 따라서 설명한다.
- [0062] 먼저, 스텝 S301에서는, 화상 분할부(102)는 입력 화상을 타일, 브릭, 슬라이스로 분할한다. 그리고 화상 분할부(102)는, 분할된 타일, 브릭, 슬라이스의 각각에 대해, 크기에 관한 정보를 분할 정보로서 통합 부호화부(111)에 출력한다. 또한 화상 분할부(102)는, 각 브릭을 기본 블록행 화상으로 분할하고, 해당 분할된 기본 블록행 화상을 블록 분할부(103)에 출력한다.
- [0063] 스텝 S302에서는, 블록 분할부(103)는, 기본 블록행 화상을 복수의 기본 블록으로 분할하고, 기본 블록 단위의 화상인 블록 화상을, 후단의 예측부(104)에 대해 출력한다.
- [0064] 스텝 S303에서는, 예측부(104)는, 블록 분할부(103)로부터 출력된 기본 블록 단위의 화상을 서브블록으로 분할하고, 서브블록 단위로 인트라 예측 모드를 결정하고, 해당 결정된 인트라 예측 모드 및 부호화 완료된 화소로부터 예측 화상을 생성한다. 또한 예측부(104)는, 입력된 화상과 예측 화상으로부터 예측 오차를 산출하고, 해당 산출된 예측 오차를 변환·양자화부(105)에 대해 출력한다. 또한 예측부(104)는, 서브블록 분할 방법이나 인트라 예측 모드 등의 정보를 예측 정보로서, 부호화부(110) 및 화상 재생부(107)에 대해 출력한다.
- [0065] 스텝 S304에서는, 변환·양자화부(105)는, 예측부(104)로부터 출력된 예측 오차를 서브블록 단위로 직교 변환하여 변환 계수(직교 변환 계수)를 구한다. 그리고 변환·양자화부(105)는, 해당 구한 변환 계수를 양자화하여 양자화 계수를 구한다. 그리고 변환·양자화부(105)는, 해당 구한 양자화 계수를 부호화부(110) 및 역양자화·역변환부(106)에 대해 출력한다.
- [0066] 스텝 S305에서는, 역양자화·역변환부(106)는, 변환·양자화부(105)로부터 출력된 양자화 계수를 역양자화하여 변환 계수를 재생하고, 또한 해당 재생된 변환 계수를 역직교 변환하여 예측 오차를 재생한다. 그리고 역양자

화·역변환부(106)는, 해당 재생된 예측 오차를 화상 재생부(107)에 대해 출력한다.

- [0067] 스텝 S306에서는, 화상 재생부(107)는, 예측부(104)로부터 출력된 예측 정보에 기초하여 프레임 메모리(108)를 적절하게 참조하여 예측 화상을 생성하고, 해당 예측 화상과, 역양자화·역변환부(106)로부터 입력된 예측 오차로부터 재생 화상을 생성한다. 그리고 화상 재생부(107)는, 해당 생성된 재생 화상을 프레임 메모리(108)에 저장한다.
- [0068] 스텝 S307에서는, 부호화부(110)는, 변환·양자화부(105)로부터 출력된 양자화 계수 및 예측부(104)로부터 출력된 예측 정보를 엔트로피 부호화함으로써 부호 데이터를 생성한다.
- [0069] 여기서, `entropy_coding_sync_enabled_flag=1`인 경우, 위에 인접하는 기본 블록행의 소정의 위치의 기본 블록을 처리한 시점에서의 발생 확률 테이블을, 다음 기본 블록행의 좌측 끝의 기본 블록을 처리하기 전에 적용한다. 본 실시 형태에서는, `entropy_coding_sync_enabled_flag=1`인 것으로 하여 설명한다.
- [0070] 스텝 S308에서는, 제어부(199)는, 슬라이스 내의 모든 기본 블록의 부호화가 완료되었는지 여부를 판단한다. 이 판단 결과, 슬라이스 내의 모든 기본 블록의 부호화가 완료된 경우에는, 처리는 스텝 S309로 진행한다. 한편, 슬라이스 내의 기본 블록 중 아직 부호화되어 있지 않은 기본 블록(미부호화의 기본 블록)이 남아 있는 경우에는, 해당 미부호화의 기본 블록을 부호화하기 위해, 처리는 스텝 S303으로 진행한다.
- [0071] 스텝 S309에서는, 통합 부호화부(111)는, 화상 분할부(102)로부터 출력된 분할 정보를 사용하여 헤더 부호 데이터를 생성하고, 해당 생성된 헤더 부호 데이터와, 부호화부(110)로부터 출력된 부호 데이터를 포함하는 비트 스트림을 생성하여 출력한다.
- [0072] 입력 화상을 도 8a, 도 8b에 도시하는 바와 같이 분할한 경우, 타일 데이터 분할 정보의, `single_tile_in_pic_flag`는 0, `uniform_tile_spacing_flag`는 1이 된다. 또한 `tile_cols_width_minus1`은 5, `tile_rows_height_minus1`은 5이다.
- [0073] 브릭 데이터 분할 정보의 `brick_splitting_present_flag`는 1이다. 또한, 타일 ID=1, 4, 5, 6, 8에 대응하는 타일은 브릭으로 분할되어 있지 않다. 그런데 `brick_split_flag[1]`, `brick_split_flag[4]`, `brick_split_flag[5]`, `brick_split_flag[6]`, `brick_split_flag[8]`은 0이다. 또한 타일 ID=0, 2, 3, 7에 대응하는 타일은 브릭으로 분할되어 있다. 그런데 `brick_split_flag[0]`, `brick_split_flag[2]`, `brick_split_flag[3]`, `brick_split_flag[7]`은 1이다.
- [0074] 또한, 타일 ID=0, 3, 7에 대응하는 타일은 모두 동일 사이즈의 브릭으로 분할되어 있다. 그런데 `uniform_brick_spacing_flag[0]`, `uniform_brick_spacing_flag[3]`, `uniform_brick_spacing_flag[7]`은 모두 1이다. 타일 ID=2에 대응하는 타일에 대해서는, BID=3의 브릭의 사이즈는, BID=4의 브릭의 사이즈와는 다르다. 그런데 `uniform_brick_spacing_flag[2]`는 0이다.
- [0075] `brick_height_minus1[0]`은 2이고, `brick_height_minus1[3]`은 1이다. 또한, `brick_height_minus1[7]`은 2이다. 또한, `brick_height_minus1`은 `uniform_brick_spacing`이 1인 경우에 부호화된다.
- [0076] `brick_row_height_minus1[2][0]`은 1이다. 또한, `num_brick_rows_minus1[2]`가 아니라, 대신에 상술한 `num_brick_rows_minus2[2]`의 선택스를 부호화하는 경우는, 값은 0이 된다.
- [0077] 또한, 본 실시 형태에서는, 1개의 슬라이스가 복수의 브릭을 포함하므로, 슬라이스 데이터 분할 정보 0에 있어서의 `single_brick_per_slice_flag`는 0이 된다. 또한 본 실시 형태에서는, 슬라이스가 복수의 타일을 래스터순으로 내포하므로(슬라이스는 래스터순으로 배열되는 타일을 포함함), `rect_slice_flag`는 0이다.
- [0078] 이 경우, SLH에 있어서의 슬라이스 데이터 분할 정보로서, 먼저 슬라이스 중의 최초의 BID(해당 슬라이스 중의 브릭의 BID 중 최초의 BID)가 `slice_address`로서 부호화된다.
- [0079] 슬라이스 0에 대해서는, 해당 슬라이스 0 중의 최초의 브릭의 BID인 「0」이 `slice_address`로서 부호화된다. 슬라이스 1에 대해서는, 해당 슬라이스 1 중의 최초의 브릭의 BID인 「1」이 `slice_address`로서 부호화된다. 슬라이스 2에 대해서는, 해당 슬라이스 2 중의 최초의 브릭의 BID인 「2」가 `slice_address`로서 부호화된다. 슬라이스 3에 대해서는, 해당 슬라이스 3 중의 최초의 브릭의 BID인 「3」이 `slice_address`로서 부호화된다. 슬라이스 4에 대해서는, 해당 슬라이스 4 중의 최초의 브릭의 BID인 「11」이 `slice_address`로서 부호화된다.
- [0080] 그리고 `num_bricks_in_slice_minus1`로서 각 슬라이스가 몇 개의 브릭을 내포하고 있는지가 부호화된다. 슬라이스 0에 포함되는 브릭은 BID=0에 대응하는 브릭뿐(단일의 브릭뿐)이므로, 슬라이스 0에 대응하는

num\_bricks\_in\_slice\_minus1은 0이 된다. 슬라이스 1에 포함되는 브릭은 BID=1에 대응하는 브릭뿐(단일의 브릭뿐)이므로, 슬라이스 1에 대응하는 num\_bricks\_in\_slice\_minus1은 0이 된다. 슬라이스 2에 포함되는 브릭은 BID=2에 대응하는 브릭뿐(단일의 브릭뿐)이므로, 슬라이스 2에 대응하는 num\_bricks\_in\_slice\_minus1은 0이 된다. 슬라이스 3에 포함되는 브릭은 BID=3 내지 10에 대응하는 브릭(8개의 브릭)이므로, 슬라이스 3에 대응하는 num\_bricks\_in\_slice\_minus1은 7이 된다. 슬라이스 4에 포함되는 브릭은 BID=11 내지 13에 대응하는 브릭(3개의 브릭)이므로, 슬라이스 4에 대응하는 num\_bricks\_in\_slice\_minus1은 2가 된다.

- [0081] 다음으로, 기본 블록행 데이터 위치 정보는, 부호화부(110)로부터 보내진, (슬라이스 중의 ((j-1)번째의 기본 블록행의 부호 데이터의 크기-1)의 값을 entry\_point\_offset\_minus1[j-1]로서 부호화한다. 슬라이스 중의 entry\_point\_offset\_minus1의 수는, (슬라이스 중의 기본 블록행 수-1)과 동등하다.
- [0082] 슬라이스 0은 단일의 브릭으로 구성되어 있고(대응하는 num\_bricks\_in\_slice\_minus1=0), brick\_height\_minus1[0]=2이다. 그런데 해당 브릭의 기본 블록행의 수는, brick\_height\_minus1[0]+1=3이 된다(j의 범위는 0 내지 1).
- [0083] 슬라이스 1은 단일의 브릭으로 구성되어 있고(대응하는 num\_bricks\_in\_slice\_minus1=0), brick\_height\_minus1[0]=2, tile\_rows\_height\_minus1=5이다. 그런데 해당 브릭의 기본 블록행의 수는, tile\_rows\_height\_minus1-brick\_height\_minus1[0]=3이 된다(j의 범위는 0 내지 1).
- [0084] 슬라이스 2는 단일의 브릭으로 구성되어 있고(대응하는 num\_bricks\_in\_slice\_minus1=0), brick\_height\_minus1[1]=5이다. 그런데 해당 브릭의 기본 블록행의 수는, brick\_height\_minus1[1]+1=6이 된다(j의 범위는 0 내지 4).
- [0085] 슬라이스 3의 경우, 대응하는 slice\_address=3, 대응하는 num\_bricks\_in\_slice\_minus1=7이다. 그런데 슬라이스 3은 BID=3 내지 10(=slice\_address+num\_bricks\_in\_slice\_minus1)에 대응하는 브릭(8개의 브릭)으로 구성되어 있다. 슬라이스 3에 있어서의 기본 블록행의 수는, BID=3 내지 10 각각에 대응하는 브릭의 기본 블록행의 수의 합계값이 된다.
- [0086] BID=3에 대응하는 브릭의 기본 블록행의 수는, brick\_row\_height\_minus1[2][0]=1이므로, brick\_row\_height\_minus1[2][0]+1=2이다.
- [0087] BID=4에 대응하는 브릭의 기본 블록행의 수는, 다음과 같이 하여 도출할 수 있다. brick\_row\_height\_minus1[2][0]=1, num\_brick\_rows\_minus1[2]=1, tile\_rows\_height\_minus1=5이다. 그런데 BID=4에 대응하는 브릭의 기본 블록행의 수는, tile\_rows\_height\_minus1-brick\_row\_height\_minus1[2][0]=4이다.
- [0088] BID=5 내지 7의 각각에 대응하는 브릭의 기본 블록행의 수는, brick\_height\_minus1[3]=1인 점에서, brick\_height\_minus1[3]+1=2인 것을 알 수 있다.
- [0089] BID=8 내지 10의 각각에 대응하는 브릭의 기본 블록행의 수는, tile\_rows\_height\_minus1=5인 점에서, tile\_rows\_height\_minus1+1=6인 것을 알 수 있다.
- [0090] 따라서, 슬라이스 3에 있어서의 기본 블록행의 수는, BID=3 내지 10의 각각에 대응하는 브릭의 기본 블록행의 수의 합계값(2+4+2+2+2+6+6+6)=30으로 도출할 수 있다. j의 범위는 0 내지 28이 된다.
- [0091] 슬라이스 4의 경우, 대응하는 slice\_address=11, 대응하는 num\_bricks\_in\_slice\_minus1=2이다. 그런데 슬라이스 4는, BID=11 내지 13(=slice\_address+num\_bricks\_in\_slice\_minus1)에 대응하는 브릭(3개의 브릭)으로 구성되어 있다. 슬라이스 4에 있어서의 기본 블록행의 수는, BID=11 내지 13의 각각에 대응하는 브릭의 기본 블록행의 수의 합계값이 된다.
- [0092] BID=11 내지 12의 각각에 대응하는 브릭의 기본 블록행의 수는, brick\_height\_minus1[7]=2인 점에서, brick\_height\_minus1[7]+1=3인 것을 알 수 있다.
- [0093] BID=13에 대응하는 브릭의 기본 블록행의 수는, tile\_rows\_height\_minus1=5인 점에서, tile\_rows\_height\_minus1+1=6인 것을 알 수 있다.
- [0094] 따라서, 슬라이스 4에 있어서의 기본 블록행의 수는, BID=11 내지 13의 각각에 대응하는 브릭의 기본 블록행의 수의 합계값(3+3+6)=12로 도출할 수 있다. j의 범위는 0 내지 10이 된다.

- [0095] 이러한 처리에 의해, 각 슬라이스에 있어서의 기본 블록행 수가 확정된다. 본 실시 형태에서는, 다른 선택스로부터 entry\_point\_offset\_minus1의 수를 도출할 수 있으므로, 종래와 같이 num\_entry\_point\_offset를 부호화하여 헤더에 포함할 필요는 없다. 따라서, 본 실시 형태에 따르면, 비트 스트림의 데이터양을 삭감할 수 있다.
- [0096] 스텝 S310에서는, 제어부(199)는, 입력 화상에 있어서의 모든 기본 블록의 부호화가 완료되었는지 여부를 판단한다. 이 판단의 결과, 입력 화상에 있어서의 모든 기본 블록의 부호화가 완료된 경우에는, 처리는 스텝 S311로 진행한다. 한편, 입력 화상에 있어서 아직 부호화되어 있지 않은 기본 블록이 남아 있는 경우에는 처리는 스텝 S303으로 진행하고, 아직 부호화되어 있지 않은 기본 블록에 대해 이후의 처리를 행한다.
- [0097] 스텝 S311에서는, 인루프 필터부(109)는, 스텝 S306에서 생성된 재생 화상에 대해 인루프 필터 처리를 행하고, 해당 인루프 필터 처리가 실시된 화상을 출력한다.
- [0098] 이와 같이 본 실시 형태에 따르면, 브릭이 갖는 기본 블록행의 부호 데이터의 선두 위치를 나타내는 정보가 몇 개 부호화되어 있는지를 나타내는 정보를 부호화하여 비트 스트림에 포함하지 않아도 되며, 해당 정보가 도출 가능한 비트 스트림을 생성할 수 있다.
- [0099] [제2 실시 형태]
- [0100] 본 실시 형태에서는, 제1 실시 형태에 관한 화상 부호화 장치에 의해 생성된 비트 스트림을 복호하는 화상 복호 장치에 대해 설명한다. 또한, 비트 스트림의 구성 등, 제1 실시 형태와 공통되는 요건에 대해서는 제1 실시 형태에서 설명한 바와 같으므로, 설명은 생략한다.
- [0101] 본 실시 형태에 관한 화상 복호 장치의 기능 구성 예에 대해, 도 2의 블록도를 사용하여 설명한다. 분리 복호부(202)는, 제1 실시 형태에 관한 화상 부호화 장치에 의해 생성된 비트 스트림을 취득한다. 비트 스트림의 취득 방법은 특정한 취득 방법에 한정되지 않는다. 예를 들어, LAN이나 인터넷 등의 네트워크를 통해 화상 부호화 장치로부터 직접적 혹은 간접적으로 비트 스트림을 취득해도 되고, 화상 복호 장치의 내부 혹은 외부에 보존해 둔 비트 스트림을 취득해도 된다. 그리고 분리 복호부(202)는, 해당 취득된 비트 스트림으로부터, 복호 처리에 관한 정보나 계수에 관한 부호 데이터를 분리하여, 복호부(203)로 보낸다. 또한 분리 복호부(202)는, 비트 스트림의 헤더 부호 데이터를 복호한다. 본 실시 형태에서는, 타일, 브릭, 슬라이스, 기본 블록의 크기 등의 화상의 분할에 관한 헤더 정보를 복호하여 분할 정보를 생성하고, 해당 생성된 분할 정보를 화상 재생부(205)에 출력한다. 즉 분리 복호부(202)는, 도 1의 통합 부호화부(111)와 반대의 동작을 행한다.
- [0102] 복호부(203)는, 분리 복호부(202)로부터 출력된 부호 데이터를 복호하여 양자화 계수 및 예측 정보를 재생한다. 역양자화·역변환부(204)는, 양자화 계수에 대해 역양자화를 행하여 변환 계수를 생성하고, 해당 생성된 변환 계수에 대해 역직교 변환을 행함으로써 예측 오차를 재생한다.
- [0103] 프레임 메모리(206)는, 재생된 픽처의 화상 데이터를 저장하기 위한 메모리이다. 화상 재생부(205)는, 입력된 예측 정보에 기초하여 프레임 메모리(206)를 적절하게 참조하여 예측 화상을 생성한다. 그리고 화상 재생부(205)는, 해당 생성된 예측 화상과 역양자화·역변환부(204)에서 재생된 예측 오차로부터 재생 화상을 생성한다. 그리고 화상 재생부(205)는, 재생 화상에 대해, 분리 복호부(202)로부터 입력된 분할 정보에 기초하여 타일, 브릭, 슬라이스의 입력 화상 중의 위치를 특정하여 출력한다.
- [0104] 인루프 필터부(207)는, 상기한 인루프 필터부(109)와 마찬가지로, 재생 화상에 대해 디블로킹 필터 등의 인루프 필터 처리를 행하고, 인루프 필터 처리가 실시된 화상을 출력한다. 제어부(299)는, 화상 복호 장치 전체의 동작 제어를 행하는 것이며, 상기한 화상 복호 장치의 각 기능부의 동작 제어를 행한다.
- [0105] 다음으로, 도 2에 도시한 구성을 갖는 화상 복호 장치에 의한 비트 스트림의 복호 처리에 대해 설명한다. 이하에서는, 비트 스트림을 프레임 단위로 화상 복호 장치에 입력하는 것으로서 설명하는데, 1프레임분의 정지 화상의 비트 스트림을 화상 복호 장치에 입력하도록 해도 된다. 또한, 본 실시 형태에서는 설명을 용이하게 하기 위해 인트라 예측 복호 처리만을 설명하는데, 이것에 한정되지 않고 인터 예측 복호 처리에 있어서도 적용 가능하다.
- [0106] 분리 복호부(202)는, 입력된 비트 스트림으로부터, 복호 처리에 관한 정보나 계수에 관한 부호 데이터를 분리하여, 복호부(203)로 보낸다. 또한 분리 복호부(202)는, 비트 스트림의 헤더의 부호 데이터를 복호한다. 보다 구체적으로는 분리 복호부(202)는, 도 6에 있어서의 기본 블록 데이터 분할 정보, 타일 데이터 분할 정보, 브릭 데이터 분할 정보, 슬라이스 데이터 분할 정보 0, 기본 블록행 데이터 동기화 정보, 기본 블록행 데이터 위치 정보 등을 복호하여 분할 정보를 생성한다. 그리고 분리 복호부(202)는, 해당 생성된 분할 정보를 화상 재생부

(205)에 출력한다. 또한 분리 복호부(202)는, 픽처 데이터의 기본 블록 단위의 부호 데이터를 재생하여, 복호부(203)에 출력한다.

[0107] 복호부(203)는, 분리 복호부(202)로부터 출력된 부호 데이터를 복호하여 양자화 계수 및 예측 정보를 재생한다. 재생된 양자화 계수는 역양자화·역변환부(204)에 출력되고, 재생된 예측 정보는 화상 재생부(205)에 출력된다.

[0108] 역양자화·역변환부(204)는, 입력된 양자화 계수에 대해 역양자화를 행하여 변환 계수를 생성하고, 해당 생성된 변환 계수에 대해 역직교 변환을 행함으로써 예측 오차를 재생한다. 재생된 예측 오차는 화상 재생부(205)에 출력된다.

[0109] 화상 재생부(205)는, 분리 복호부(202)로부터 입력된 예측 정보에 기초하여 프레임 메모리(206)를 적절하게 참조하여 예측 화상을 생성한다. 그리고 화상 재생부(205)는, 해당 생성된 예측 화상과 역양자화·역변환부(204)에서 재생된 예측 오차로부터 재생 화상을 생성한다. 그리고 화상 재생부(205)는, 재생 화상에 대해, 분리 복호부(202)로부터 입력된 분할 정보에 기초하여, 예를 들어 도 7과 같은 타일, 브릭, 슬라이스의 형상 및 입력 화상 중의 위치를 특정하여 프레임 메모리(206)에 출력(저장)한다. 프레임 메모리(206)에 저장된 화상은 예측 시의 참조에 사용된다.

[0110] 인루프 필터부(207)는, 프레임 메모리(206)로부터 판독한 재생 화상에 대해 디블로킹 필터 등의 인루프 필터 처리를 행하고, 인루프 필터 처리가 실시된 화상을 프레임 메모리(206)에 출력(저장)한다.

[0111] 제어부(299)는, 프레임 메모리(206)에 저장된 재생 화상을 출력한다. 재생 화상의 출력처는 특정한 출력처에 한정되지 않는다. 예를 들어 제어부(299)는, 화상 복호 장치가 갖는 표시 장치에 해당 재생 화상을 출력하여 해당 표시 장치에 해당 재생 화상을 표시시켜도 된다. 또한 예를 들어 제어부(299)는, LAN이나 인터넷 등의 네트워크를 통해 재생 화상을 외부의 장치에 대해 송신해도 된다.

[0112] 다음으로, 본 실시 형태에 관한 화상 복호 장치에 의한 비트 스트림의 복호 처리(도 6의 구성을 갖는 비트 스트림의 복호 처리)에 대해, 도 4의 흐름도에 따라서 설명한다.

[0113] 스텝 S401에서는, 분리 복호부(202)는, 입력된 비트 스트림으로부터, 복호 처리에 관한 정보나 계수에 관한 부호 데이터를 분리하여, 복호부(203)로 보낸다. 또한 분리 복호부(202)는, 비트 스트림의 헤더의 부호 데이터를 복호한다. 보다 구체적으로는, 분리 복호부(202)는, 도 6에 있어서의 기본 블록 데이터 분할 정보, 타일 데이터 분할 정보, 브릭 데이터 분할 정보, 슬라이스 데이터 분할 정보, 기본 블록행 데이터 동기화 정보, 기본 블록행 데이터 위치 정보 등을 복호하여 분할 정보를 생성한다. 그리고 분리 복호부(202)는, 해당 생성된 분할 정보를 화상 재생부(205)에 출력한다. 또한 분리 복호부(202)는, 픽처 데이터의 기본 블록 단위의 부호 데이터를 재생하여, 복호부(203)에 출력한다.

[0114] 본 실시 형태에서는, 비트 스트림의 부호화원인 입력 화상의 분할은 도 8a, 도 8b에 도시한 분할로 한다. 비트 스트림의 부호화원인 입력 화상이나 그 분할에 관한 정보는 분할 정보로부터 도출할 수 있다.

[0115] 화상 사이즈 정보에 포함되어 있는 `pic_width_in_luma_samples`로부터, 입력 화상의 수평 방향의 사이즈(가로 사이즈)가 1152화소인 것을 특정할 수 있다. 또한, 화상 사이즈 정보에 포함되어 있는 `pic_height_in_luma_samples`로부터, 입력 화상의 수직 방향의 사이즈(세로 사이즈)가 1152화소인 것을 특정할 수 있다.

[0116] 또한, 기본 블록 데이터 분할 정보의 `log2_ctu_size_minus2=4`인 점에서, 기본 블록의 사이즈를,  $1 < \log_2 \text{ctu\_size\_minus2} + 2$ 로부터,  $64 \times 64$ 화소로 도출할 수 있다.

[0117] 또한, 타일 데이터 분할 정보의 `single_tile_in_pic_flag=0`인 점에서, 입력 화상이 복수의 타일로 분할되어 있는 것을 특정할 수 있다. 그리고 `uniform_tile_spacing_flag=1`인 점에서, 각 타일이 (단부를 제외하고)동일한 사이즈를 갖는 것을 특정할 수 있다.

[0118] 또한, `tile_cols_width_minus1=5`, `tile_rows_height_minus1=5`인 점에서, 각 타일이  $6 \times 6$ 개의 기본 블록으로 구성되어 있는 것을 특정할 수 있다. 즉, 각 타일이  $384 \times 384$ 화소로 구성되어 있는 것을 특정할 수 있다. 입력 화상이  $1152 \times 1152$ 화소인 점에서, 해당 입력 화상은 수평 방향으로 3개, 수직 방향으로 3개의 9개의 타일로 분할되어 부호화되어 있는 것을 알 수 있다.

[0119] 또한, 브릭 데이터 분할 정보의 `brick_splitting_present_flag=1`인 점에서, 입력 화상 중의 적어도 1개의 타일이 복수의 브릭으로 분할되어 있는 것을 특정할 수 있다.

- [0120] 또한, `brick_split_flag[1]`, `brick_split_flag[4]`, `brick_split_flag[5]`, `brick_split_flag[6]`, `brick_split_flag[8]`은 0이다. 이에 의해, 타일 ID=1, 4, 5, 6, 8에 대응하는 타일은 브릭으로 분할되어 있지 않은 것을 특정할 수 있다. 본 실시 형태에서는, 모든 타일의 기본 블록행 수가 6인 점에서, 타일 ID=1, 4, 5, 6, 8에 대응하는 타일의 브릭의 기본 블록행 수는 6인 것을 알 수 있다.
- [0121] 한편, `brick_split_flag[0]`, `brick_split_flag[2]`, `brick_split_flag[3]`, `brick_split_flag[7]`은 1이다. 이에 의해, 타일 ID=0, 2, 3, 7에 대응하는 타일은 브릭으로 분할되어 있는 것을 특정할 수 있다. 또한, `uniform_brick_spacing_flag[0]`, `uniform_brick_spacing_flag[3]`, `uniform_brick_spacing_flag[7]`은 모두 1이다. 이에 의해, 타일 ID=0, 3, 7에 대응하는 타일은 모두 동일 사이즈의 브릭으로 분할되어 있는 것을 특정할 수 있다.
- [0122] 또한, `brick_height_minus1[0]`, `brick_height_minus1[7]`은 모두 2이다. 그런데, 타일 ID=0에 대응하는 타일 및 타일 ID=7에 대응하는 타일은 모두, 타일 내의 브릭의 수직 방향의 기본 블록수가 3인 것을 특정할 수 있다. 또한, 타일 ID=0에 대응하는 타일 및 타일 ID=7에 대응하는 타일은 모두, 해당 타일 내의 브릭수가 2 (=타일의 기본 블록행 수(6)/타일 내의 브릭의 수직 방향의 기본 블록수(3))인 것을 특정할 수 있다.
- [0123] 또한, `brick_height_minus1[3]`은 1이다. 그런데, 타일 ID=3에 대응하는 타일 내의 브릭의 수직 방향의 기본 블록수가 2인 것을 특정할 수 있다. 또한, 타일 ID=3에 대응하는 타일 내의 브릭수가 3(=타일의 기본 블록행 수(6)/타일 내의 브릭의 수직 방향의 기본 블록수(2))인 것을 특정할 수 있다.
- [0124] 타일 ID=2에 대응하는 타일에 대해서는, `num_brick_rows_minus1[2]=1`인 점에서, 해당 타일은 2개의 브릭으로 구성되어 있는 것을 특정할 수 있다. 또한, `uniform_brick_spacing_flag[2]=0`이다. 이에 의해, 타일 ID=2에 대응하는 타일은, 다른 것과 사이즈가 다른 브릭이 존재하는 것을 특정할 수 있다. 그리고 `brick_row_height_minus1[2][0]=1`, `brick_row_height_minus1[2][1]=3`, 모든 타일의 수직 방향의 기본 블록수는 6이다. 이에 의해, 타일 ID=2에 대응하는 타일은 위로부터 차례로, 수직 방향의 기본 블록수가 2인 브릭, 수직 방향의 기본 블록수가 4인 브릭으로 구성되어 있는 것을 특정할 수 있다. 또한, `brick_row_height_minus1[2][1]=3`은 부호화하지 않도록 해도 된다. 타일에 있어서의 브릭의 수가 2개인 경우, 타일의 높이와, 타일 내의 첫 번째 브릭의 높이(`brick_row_height_minus1[2][0]=1`)로부터 두 번째 브릭의 높이를 구하는 것이 가능하기 때문이다.
- [0125] 또한, 슬라이스 데이터 분할 정보 0에 있어서의 `single_brick_per_slice_flag=0`인 점에서, 적어도 1개의 슬라이스가 복수의 브릭으로 구성되어 있는 것을 특정할 수 있다. 본 실시 형태에서는, `uniform_brick_spacing_flag[i]=0`인 경우는, (i번째의 타일을 구성하는 브릭수-1)을 나타내는 `num_brick_rows_minus1[i]`를 브릭 데이터 분할 정보에 포함하였다. 그러나 이것에 한정되는 것은 아니다.
- [0126] 예를 들어 `brick_split_flag[i]=1`인 시점에서 i번째의 타일을 구성하는 브릭의 수가 2 이상이라고 상정한다. 그리고 (해당 타일을 구성하는 브릭의 수-2)를 나타내는 `num_brick_rows_minus2[i]`를 `num_brick_rows_minus1[i]` 대신에 복호해도 된다. 이와 같이 함으로써, 해당 타일을 구성하는 브릭의 수를 나타내는 선택스의 비트수를 저감시킨 비트 스트림을 복호할 수 있다.
- [0127] 다음으로, 각 브릭의 좌측 상단 및 우측 하단의 경계의 좌표를 구한다. 좌표는 입력 화상의 좌측 상단을 원점으로 하고, 기본 블록의 수평 위치와 수직 위치로 표시된다. 예를 들어, 좌측으로부터 3번째, 위로부터 2번째의 기본 블록의 좌측 상단의 경계의 좌표는 (3, 2), 우측 하단의 경계의 좌표는 (4, 3)이 된다.
- [0128] 타일 ID=0에 대응하는 타일 내의 BID=0의 브릭의 좌측 상단의 경계의 좌표는 (0, 0)이다. BID=0의 브릭의 기본 블록행 수는 3이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (3, 3)이 된다.
- [0129] 타일 ID=0에 대응하는 타일 내의 BID=1의 브릭의 좌측 상단의 경계의 좌표는 (0, 3)이다. BID=1의 브릭의 기본 블록행 수는 3이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (6, 6)이 된다.
- [0130] 타일 ID=1에 대응하는 타일(BID=2의 브릭)의 좌측 상단의 경계의 좌표는 (6, 0)이다. BID=2의 브릭의 기본 블록행 수는 6이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (12, 6)이 된다.
- [0131] 타일 ID=2에 대응하는 타일 내의 BID=3의 브릭의 좌측 상단의 경계의 좌표는 (12, 0)이다. BID=3의 브릭의

기본 블록행 수는 2이므로, 우측 하단의 경계의 좌표는 (18, 2)가 된다.

- [0132] 타일 ID=2에 대응하는 타일 내의 BID=4의 브릭의 좌측 상단의 경계의 좌표는 (12, 2)이다. BID=4의 브릭의 기본 블록행 수는 4이므로, 우측 하단의 경계의 좌표는 (18, 6)이 된다.
- [0133] 타일 ID=3에 대응하는 타일 내의 BID=5의 브릭의 좌측 상단의 경계의 좌표는 (0, 6)이다. BID=5의 브릭의 기본 블록행 수는 2이므로, 우측 하단의 경계의 좌표는 (6, 8)이 된다.
- [0134] 타일 ID=3에 대응하는 타일 내의 BID=6의 브릭의 좌측 상단의 경계의 좌표는 (0, 8)이다. BID=6의 브릭의 기본 블록행 수는 2이므로, 우측 하단의 경계의 좌표는 (6, 10)이 된다.
- [0135] 타일 ID=3에 대응하는 타일 내의 BID=7의 브릭의 좌측 상단의 경계의 좌표는 (0, 10)이다. BID=7의 브릭의 기본 블록행 수는 2이므로, 우측 하단의 경계의 좌표는 (6, 12)가 된다.
- [0136] 타일 ID=4에 대응하는 타일(BID=8의 브릭)의 좌측 상단의 경계의 좌표는 (6, 6)이다. BID=8의 브릭의 기본 블록행 수는 6이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (12, 12)가 된다.
- [0137] 타일 ID=5에 대응하는 타일(BID=9의 브릭)의 좌측 상단의 경계의 좌표는 (12, 6)이다. BID=9의 브릭의 기본 블록행 수는 6이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (18, 12)가 된다.
- [0138] 타일 ID=6에 대응하는 타일(BID=10의 브릭)의 좌측 상단의 경계의 좌표는 (0, 12)이다. BID=10의 브릭의 기본 블록행 수는 6이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (6, 18)이 된다.
- [0139] 타일 ID=7에 대응하는 타일 내의 BID=11의 브릭의 좌측 상단의 경계의 좌표는 (6, 12)이다. BID=11의 브릭의 기본 블록행 수는 3이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (12, 15)가 된다.
- [0140] 타일 ID=7에 대응하는 타일 내의 BID=12의 브릭의 좌측 상단의 경계의 좌표는 (6, 15)이다. BID=12의 브릭의 기본 블록행 수는 3이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (12, 18)이 된다.
- [0141] 타일 ID=8에 대응하는 타일(BID=13의 브릭)의 좌측 상단의 경계의 좌표는 (12, 12)이다. BID=13의 브릭의 기본 블록행 수는 6이고, 모든 타일의 수평 방향의 기본 블록수는 6이므로, 우측 하단의 경계의 좌표는 (18, 18)이 된다.
- [0142] 다음으로, 각 슬라이스에 포함되는 브릭을 특정한다. 슬라이스 0에 대해서는, slice\_address는 0, num\_bricks\_in\_slice\_minus1은 0이므로, 슬라이스 0은 BID=0의 브릭만으로 구성되어 있는 것을 알 수 있다.
- [0143] 슬라이스 1에 대해서는, slice\_address는 1, num\_bricks\_in\_slice\_minus1은 0이므로, 슬라이스 1은 BID=1의 브릭만으로 구성되어 있는 것을 알 수 있다.
- [0144] 슬라이스 2에 대해서는, slice\_address는 2, num\_bricks\_in\_slice\_minus1은 0이므로, 슬라이스 2는 BID=2의 브릭만으로 구성되어 있는 것을 알 수 있다.
- [0145] 슬라이스 3에 대해서는, slice\_address는 3, num\_bricks\_in\_slice\_minus1은 7이다. 그런데 슬라이스 3은 BID=3 내지 10(=slice\_address+num\_bricks\_in\_slice\_minus1)의 브릭으로 구성되어 있는 것을 알 수 있다.
- [0146] 슬라이스 4에 대해서는, slice\_address는 11, num\_bricks\_in\_slice\_minus1은 2이다. 그런데 슬라이스 4는 BID=11 내지 13(=slice\_address+num\_bricks\_in\_slice\_minus1)의 브릭으로 구성되어 있는 것을 알 수 있다.
- [0147] 또한, entropy\_coding\_sync\_enabled\_flag가 1이다. 따라서, (슬라이스 중의 (j-1)번째의 기본 블록행의 부호 데이터의 크기-1)이 entry\_point\_offset\_minus1[j-1]로서 부호화되어 있는 것을 알 수 있다. 그 개수는, (처리할 슬라이스의 기본 블록행 수-1)이다.
- [0148] 슬라이스 0에 대해서는, 해당 슬라이스 0을 구성하는 BID=0의 브릭의 기본 블록행 수=3이므로, 3-1=2가 되고, j의 범위는 0 내지 1이 된다. 슬라이스 1에 대해서는, 해당 슬라이스 1을 구성하는 BID=1의 브릭의 기본 블록행 수=3이므로, 3-1=2가 되고, j의 범위는 0 내지 1이 된다. 슬라이스 2에 대해서는, 해당 슬라이스 2를 구성하는 BID=2의 브릭의 기본 블록행 수=6이므로, 6-1=5가 되고, j의 범위는 0 내지 4가 된다. 슬라이스

스 3에 대해서는, 해당 슬라이스 3을 구성하는 BID=3 내지 10의 브릭의 기본 블록행 수의 합계값은  $2+4+2+2+2+6+6+6=30$ 이므로, j의 범위는 0 내지 28이 된다.

- [0149] 슬라이스 4에 대해서는, 해당 슬라이스 4를 구성하는 BID=11 내지 13의 브릭의 기본 블록행 수의 합계값은  $3+3+6=12$ 이므로, j의 범위는 0 내지 10이 된다.
- [0150] 이상에 의해, 종래와 같이 num\_entry\_point\_offset를 부호화하지 않아도, 다른 선택스로부터 entry\_point\_offset\_minus1의 수를 도출할 수 있다. 그리고 각 기본 블록행의 데이터의 선두 위치를 알 수 있으므로, 기본 블록행마다 병렬하여 복호 처리를 행할 수 있다. 분리 복호부(202)로부터 도출된 분할 정보는 화상 재생부(205)로 보내지고, 스텝 S404에서 입력 화상 내에 있어서의 처리 대상의 위치 특징에 사용된다.
- [0151] 스텝 S402에서는, 복호부(203)는, 분리 복호부(202)에서 분리된 부호 데이터를 복호하여 양자화 계수 및 예측 정보를 재생한다. 스텝 S403에서는, 역양자화·역변환부(204)는, 입력된 양자화 계수에 대해 역양자화를 행하여 변환 계수를 생성하고, 해당 생성된 변환 계수에 대해 역직교 변환을 행함으로써 예측 오차를 재생한다.
- [0152] 스텝 S404에서는, 화상 재생부(205)는, 복호부(203)로부터 입력된 예측 정보에 기초하여 프레임 메모리(206)를 적절하게 참조하여 예측 화상을 생성한다. 그리고 화상 재생부(205)는, 해당 생성된 예측 화상과 역양자화·역변환부(204)에서 재생된 예측 오차로부터 재생 화상을 생성한다. 그리고 화상 재생부(205)는, 재생 화상에 대해, 분리 복호부(202)로부터 입력된 분할 정보에 기초하여 타일, 브릭, 슬라이스의 입력 화상 중의 위치를 특정하여 그 위치에 합성하고, 프레임 메모리(206)에 출력(저장)한다.
- [0153] 스텝 S405에서는, 제어부(299)는, 입력 화상의 모든 기본 블록을 복호하였는지 여부를 판단한다. 이 판단의 결과, 입력 화상의 모든 기본 블록을 복호한 경우에는, 처리는 스텝 S406으로 진행한다. 한편, 입력 화상에 있어서 아직 복호되어 있지 않은 기본 블록이 남아 있는 경우에는, 처리는 스텝 S402로 진행하여, 아직 복호되어 있지 않은 기본 블록에 대해 복호 처리를 행한다.
- [0154] 스텝 S406에서는, 인루프 필터부(207)는, 프레임 메모리(206)로부터 판독한 재생 화상에 대해 인루프 필터 처리를 행하고, 인루프 필터 처리가 실시된 화상을 프레임 메모리(206)에 출력(저장)한다.
- [0155] 이와 같이 본 실시 형태에 따르면, 제1 실시 형태에 관한 화상 부호화 장치에 의해 생성된 「브릭이 갖는 기본 블록행의 선두 위치를 나타내는 정보가 몇 개 부호화되어 있는지를 나타내는 정보를 포함하지 않는 비트 스트림」으로부터 입력 화상을 복호할 수 있다.
- [0156] 또한, 제1 실시 형태에 관한 화상 부호화 장치와 제2 실시 형태에 관한 화상 복호 장치는 별개의 장치여도 된다. 또한, 제1 실시 형태에 관한 화상 부호화 장치와 제2 실시 형태에 관한 화상 복호 장치를 하나의 장치로 통합해도 된다.
- [0157] [제3 실시 형태]
- [0158] 도 1이나 도 2에 도시한 각 기능부는 하드웨어로 실장해도 되지만, 일부를 소프트웨어로 실장해도 된다. 후자의 경우, 프레임 메모리(108)나 프레임 메모리(206)를 제외한 각 기능부를 소프트웨어(컴퓨터 프로그램)로 실장해도 된다. 이러한 컴퓨터 프로그램을 실행 가능한 컴퓨터 장치는, 상기한 화상 부호화 장치나 화상 복호 장치에 적용 가능하다.
- [0159] 상기한 화상 부호화 장치나 화상 복호 장치에 적용 가능한 컴퓨터 장치의 하드웨어 구성에 대해, 도 5의 블록도를 사용하여 설명한다. 또한, 도 5에 도시한 하드웨어 구성은, 상기한 화상 부호화 장치나 화상 복호 장치에 적용 가능한 컴퓨터 장치의 하드웨어 구성의 일례에 불과하며, 적절하게 변경/변형이 가능하다.
- [0160] CPU(501)는 RAM(502)이나 ROM(503)에 저장되어 있는 컴퓨터 프로그램이나 데이터를 사용하여 각종 처리를 실행한다. 이에 의해 CPU(501)는 컴퓨터 장치 전체의 동작 제어를 행함과 함께, 상기한 화상 부호화 장치나 화상 복호 장치가 행하는 것으로서 설명한 각 처리를 실행 혹은 제어한다. 즉 CPU(501)는, 도 1이나 도 2에 나타낸 각 기능부(프레임 메모리(108)나 프레임 메모리(206)를 제외함)로서 기능할 수 있다.
- [0161] RAM(502)은, ROM(503)이나 외부 기억 장치(506)로부터 로드된 컴퓨터 프로그램이나 데이터를 저장하기 위한 에어리어, I/F(507)를 통해 외부로부터 수신한 데이터를 저장하기 위한 에어리어를 갖는다. 또한 RAM(502)은, CPU(501)가 각종 처리를 실행할 때에 사용하는 워크 에어리어를 갖는다. 이와 같이 RAM(502)은, 각종에어리어를 적절하게 제공할 수 있다. ROM(503)에는, 컴퓨터 장치의 설정 데이터나 기동 프로그램 등이 저장되어 있다.
- [0162] 조작부(504)는, 키보드, 마우스, 터치 패널 화면 등의 유저 인터페이스이며, 유저가 조작함으로써 각종 지시를

CPU(501)에 대해 입력할 수 있다.

- [0163] 표시부(505)는, 액정 화면이나 터치 패널 화면 등에 의해 구성되어 있고, CPU(501)에 의한 처리 결과를 화상이나 문자 등에 의해 표시할 수 있다. 또한, 표시부(505)는, 화상이나 문자를 투영하는 프로젝터 등의 장치여도 된다.
- [0164] 외부 기억 장치(506)는, 하드디스크 드라이브 장치 등의 대용량 정보 기억 장치이다. 외부 기억 장치(506)에는, OS(오퍼레이팅 시스템)나, 상기한 화상 부호화 장치나 화상 복호 장치가 행하는 것으로서 상술한 각 처리를 CPU(501)에 실행 혹은 제어시키기 위한 컴퓨터 프로그램이나 데이터가 보존되어 있다.
- [0165] 외부 기억 장치(506)에 보존되어 있는 컴퓨터 프로그램에는, 도 1이나 도 2에 있어서 프레임 메모리(108)나 프레임 메모리(206)를 제외한 각 기능부의 기능을 CPU(501)에 실행 혹은 제어시키기 위한 컴퓨터 프로그램이 포함되어 있다. 또한, 외부 기억 장치(506)에 보존되어 있는 데이터에는, 상기한 설명에 있어서 기지의 정보로서 설명한 것이나, 부호화나 복호와 관련된 다양한 정보가 포함되어 있다.
- [0166] 외부 기억 장치(506)에 보존되어 있는 컴퓨터 프로그램이나 데이터는, CPU(501)에 의한 제어에 따라서 적절하게 RAM(502)에 로드되어, CPU(501)에 의한 처리 대상이 된다.
- [0167] 도 1의 화상 부호화 장치가 갖는 프레임 메모리(108)나 도 2의 화상 부호화 장치가 갖는 프레임 메모리(206)는, 상기한 RAM(502)이나 외부 기억 장치(506) 등의 메모리 장치를 사용하여 실장 가능하다.
- [0168] I/F(507)는, 외부의 장치와의 사이의 데이터 통신을 행하기 위한 인터페이스이다. 예를 들어, 컴퓨터 장치를 화상 부호화 장치에 적용한 경우, 화상 부호화 장치는 생성된 비트 스트림을 I/F(507)를 통해 외부에 출력할 수 있다. 또한, 컴퓨터 장치를 화상 복호 장치에 적용한 경우, 화상 복호 장치는 비트 스트림을 I/F(507)를 통해 수신할 수 있다. 또한 화상 복호 장치는, 비트 스트림을 복호한 결과를 I/F(507)를 통해 외부로 송신할 수 있다. CPU(501), RAM(502), ROM(503), 조작부(504), 표시부(505), 외부 기억 장치(506), I/F(507)는 모두 버스(508)에 접속되어 있다.
- [0169] 또한, 상기한 설명에 있어서 사용한 구체적인 수치는, 구체적인 설명을 행하기 위해 사용한 것이지, 상기한 각 실시 형태가 이들의 수치에 한정되는 것을 의도한 것은 아니다. 또한, 이상 설명한 각 실시 형태의 일부 혹은 전부를 적절하게 조합해도 상관없다. 또한, 이상 설명한 각 실시 형태의 일부 혹은 전부를 선택적으로 사용해도 상관없다.
- [0170] (그 밖의 실시예)
- [0171] 본 발명은, 상술한 실시 형태의 하나 이상의 기능을 실현하는 프로그램을, 네트워크 또는 기억 매체를 통해 시스템 또는 장치에 공급하고, 그 시스템 또는 장치의 컴퓨터에 있어서의 하나 이상의 프로세서가 프로그램을 관독하여 실행하는 처리로도 실현 가능하다. 또한, 하나 이상의 기능을 실현하는 회로(예를 들어, ASIC)에 의해서도 실현 가능하다.
- [0172] 발명은 상기 실시 형태에 제한되는 것은 아니며, 발명의 정신 및 범위로부터 이탈하지 않고 다양한 변경 및 변형이 가능하다. 따라서, 발명의 범위를 명확하게 하기 위해 청구항을 첨부한다.
- [0173] 본원은, 2019년 6월 21일에 제출된 일본 특허 출원 제2019-115751호를 기초로 하여 우선권을 주장하는 것이며, 그 기재 내용의 전부를 본 명세서에 원용한다.

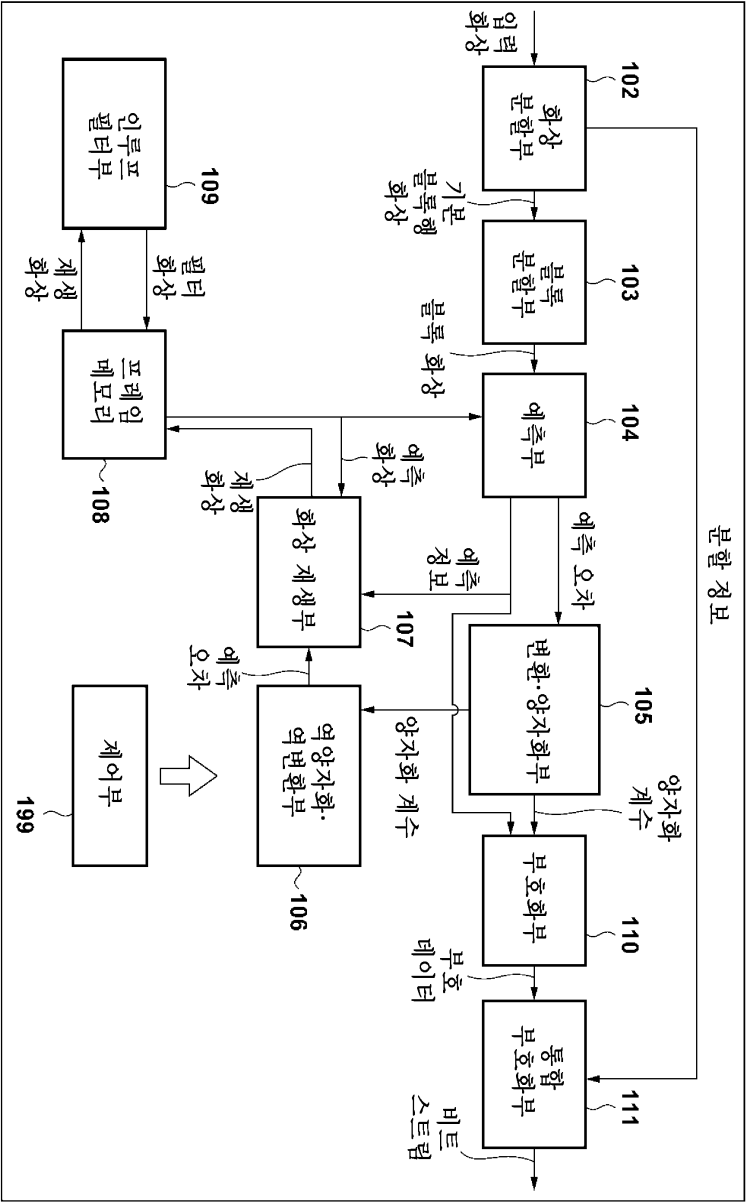
## 부호의 설명

- [0174] 102: 화상 분할부
- 103: 블록 분할부
- 104: 예측부
- 105: 변환·양자화부
- 106: 역양자화·역변환부
- 107: 화상 재생부
- 108: 프레임 메모리

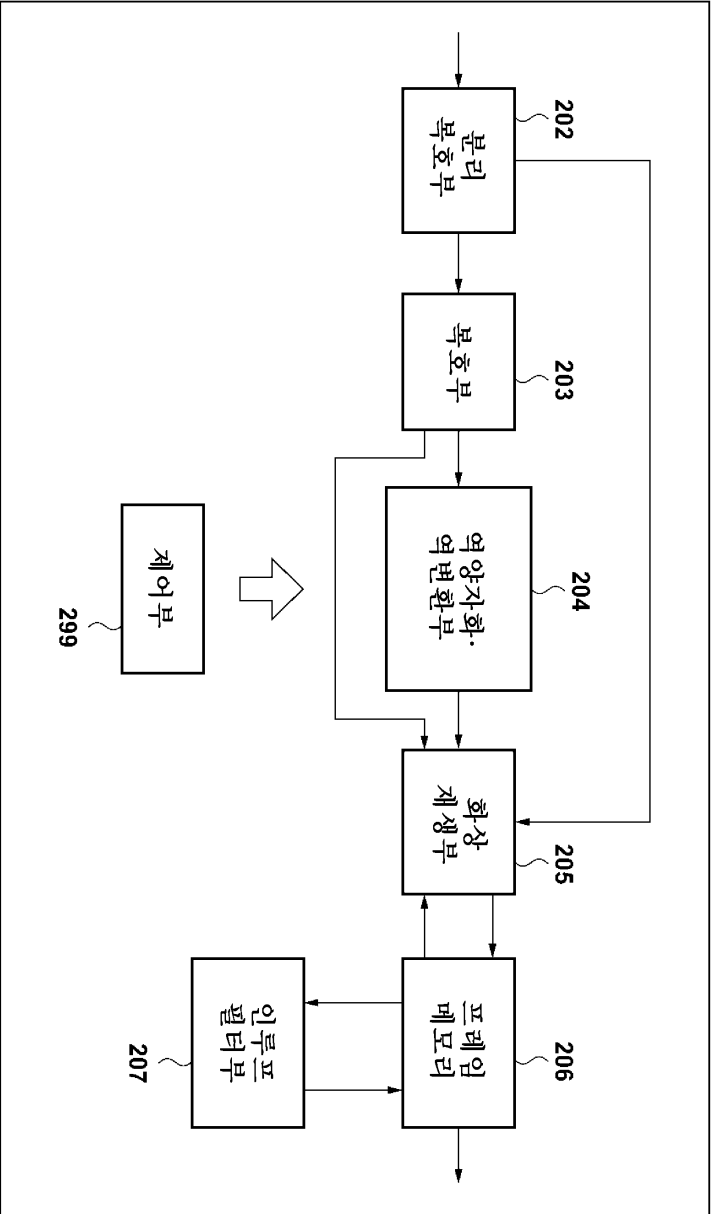
- 109: 인루프 필터부
- 110: 부호화부
- 111: 통합 부호화부

도면

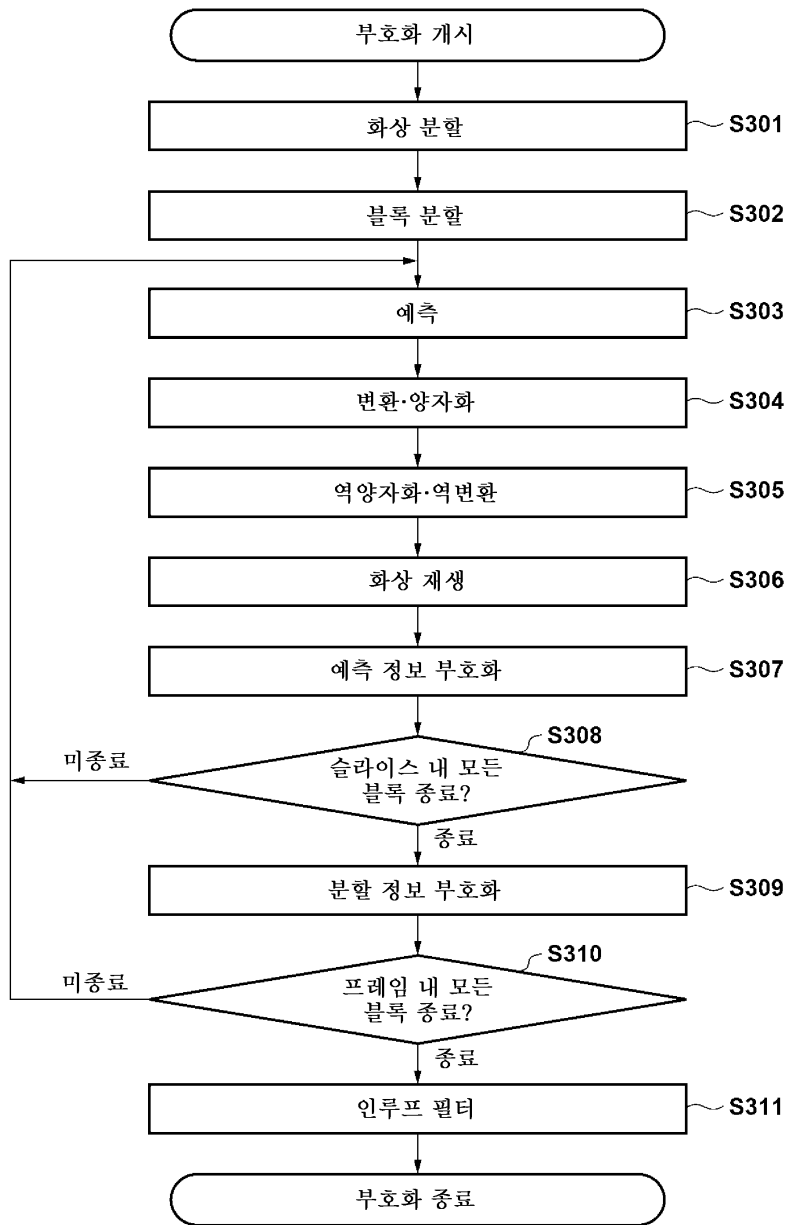
도면1



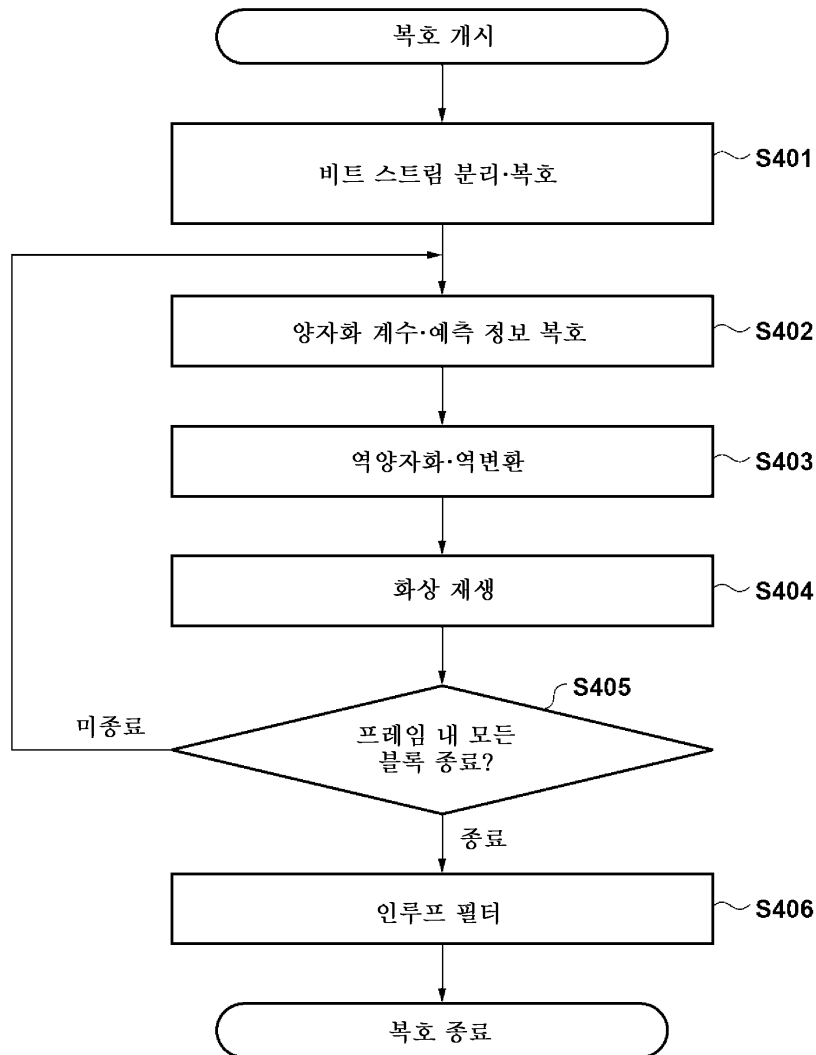
도면2



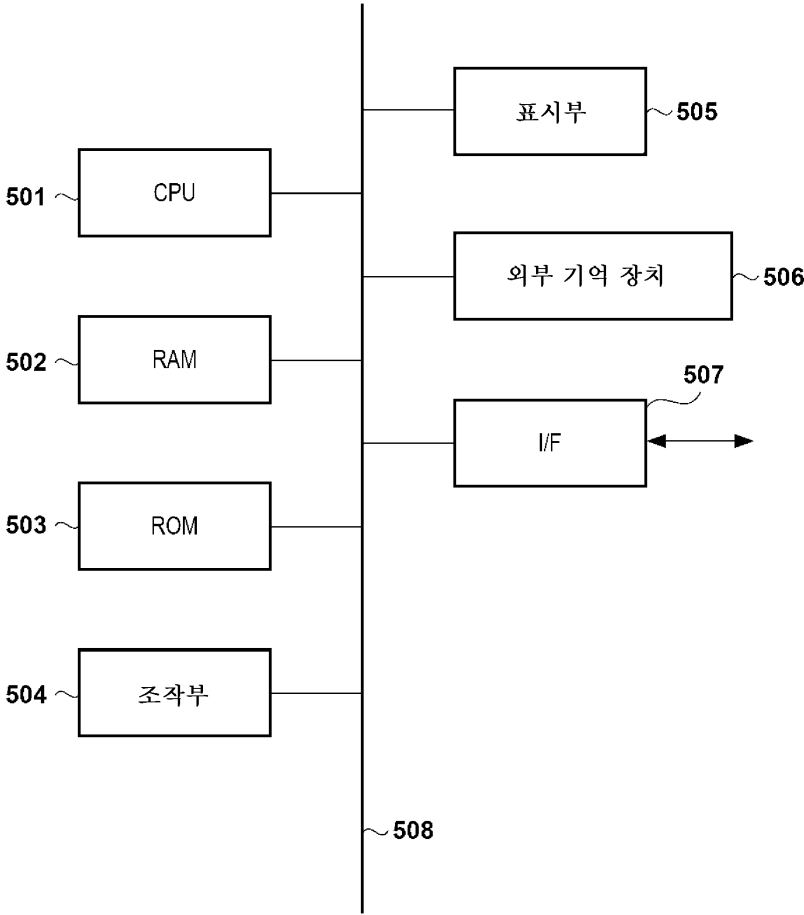
도면3

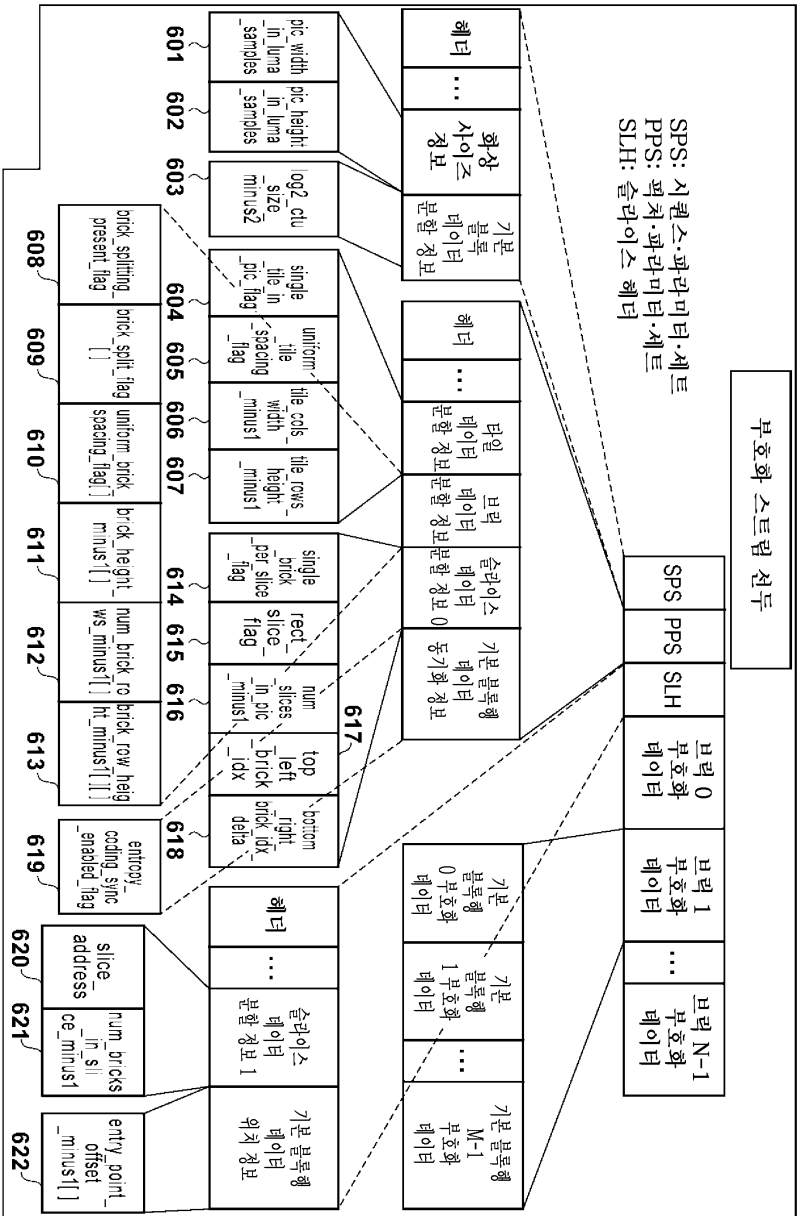


도면4



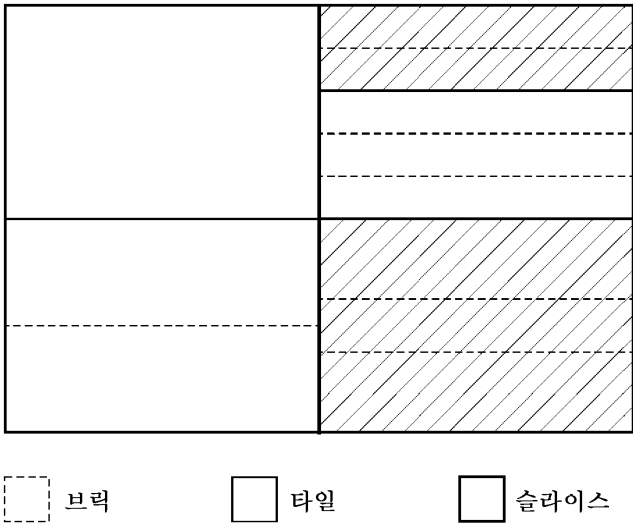
도면5



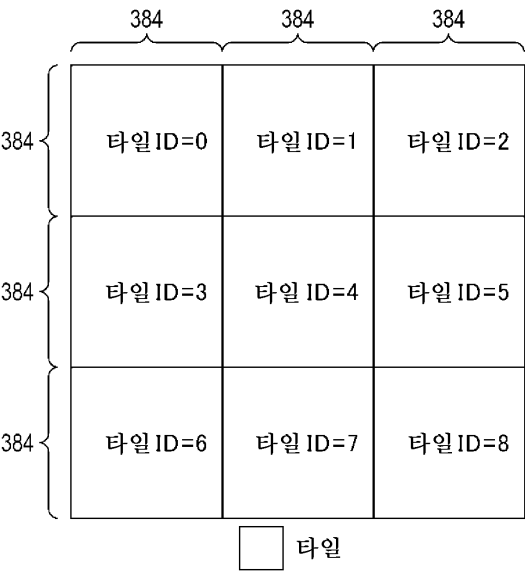


도면6

도면7



도면8a





도면9b

